ระบบตรวจวัดแก๊สแอมโมเนียแบบไร้สายโดยใช้เทคโนโลยี LoRa

WIRELESS AMMONIA DETECTION SYSTEM USING LoRa TECHNOLOGY

นายไตรภพ สวัสดิผล

นายเมืองแมน กฤษ์ปกรณ์เดช

นายอนัฐพงษ์ เตโช

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอัตโนมัติ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

ระบบตรวจวัดแก๊สแอมโมเนียแบบไร้สายโดยใช้เทคโนโลยี LoRa

WIRELESS AMMONIA DETECTION SYSTEM USING LoRa TECHNOLOGY

นายไตรภพ สวัสดิผล
นายเมืองแมน กฤษ์ปกรณ์เดช
นายอนัฐพงษ์ เตโช

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอัตโนมัติ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WIRELESS AMMONIA DETECTION SYSTEM USING LoRa TECHNOLOGY

Mr. Triphob Sawatdiphol
Mr. Muangman Kritpakorndech
Mr. Anatpong Techo

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN AUTOMATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2018

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สาขาวิชาวิศวกรรมอัตโนมัติ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

.....

หัวข้อปริญญานิพนธ์ ระบบตรวจวัดแก๊สแอมโมเนียแบบไร้สายโดยใช้เทคโนโลยี LoRa

Wireless Ammonia Detection System Using LoRa Technology

นักศึกษาผู้จัดทำ นายไตรภพ สวัสดิผล **รหัสนักศึกษา** 58010452

นายเมืองแมน กฤษ์ปรณ์เดช **รหัสนักศึกษา** 58011025

นายอนัฐพงษ์ เตโช **รหัสนักศึกษา** 58011394

ปริญญา วิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมอัตโนมัติ

ปีการศึกษา 2561

อาจารย์ผู้ควบคุมวิทยานิพนธ์	ลายมือชื่อ
ผศ.ดร.กฤษณ์ เสมอพิทักษ์	
ผศ.ดร.อภินัย ฤกษ์รัตน์	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบตรวจวัดแก๊สแอมโมเนียแบบไร้สายโดยใช้เทคโนโลยี LoRa		
นักศึกษาผู้จัดทำ	นายไตรภพ	สวัสดิผล	**รหัสนักศึกษา** 58010452
	นายเมืองแมน	กฤษ์ปกรณ์เดช	**รหัสนักศึกษา** 58011025
	นายอนัฐพงษ์	เตโช	**รหัสนักศึกษา** 58011394
อาจารย์ที่ปรึกษา	ผศ.ดร.กฤษณ์	เสมอพิทักษ์	
	ผศ.ดร.อภินัย	ฤกษ์รัตน์	
ปีการศึกษา	2561		

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เสนอวิธีการสร้างระบบตรวจวัดความเข้มข้นของแก๊สแอมโมเนียแบบ 3 ระดับสำหรับการแสดงผลระยะไกลบนแดชบอร์ดของระบบคลาวด์โดยการใช้เทคโนโลยีไร้สาย LoRa ซึ่งมีการสร้างเซนเซอร์โนดที่มีลักษณะเหมือนกันจำนวน 3 โนดสำหรับฟังก์ชันการวัดและแปลงสัญญาณ โดยค่าความเข้มข้นของแก๊สแอมโมเนียที่วัดได้จากเซนเซอร์แต่ละโนดจะถูกส่งไปยังกล่องรับสัญญาณที่มีหน้าที่ในการแปลงข้อมูลที่วัดได้จากชุดคำสั่ง LoRa มาเป็นโปรโตคอล MQTT เพื่อจัดเก็บบันทึกข้อมูลบนระบบคลาวด์ Ubidots ในการแสดงผลค่าความเข้มข้นของแก๊สแอมโมเนียที่วัดได้มีการใช้สีสามสี (สีเขียว สีเหลือง และสีแดง) แสดงค่าที่ต่างกันทั้ง 3 ช่วง (น้อยกว่า 25 ppm ค่าระหว่าง 25-100 ppm และมากกว่า 100 ppm) บนแดชบอร์ดที่สร้างขึ้น นอกจากนี้ยังมีการใช้แอพพลิเคชัน Line Notify สำหรับในการแจ้งเตือนขณะที่ความเข้มข้นของแก๊สแอมโมเนียที่วัดได้จากเซนเซอร์โนดมีค่ามากกว่าค่าความเข้มข้นที่กำหนด การแสดงผลข้อมูลที่วัดได้บนแดชบอร์ดของระบบคลาวด์ที่สร้างขึ้นทำงานได้อย่างถูกต้องซึ่งยืนยันได้จากผลการทดลองที่มีระยะการวัดแบบแนวสายตาที่ค่าต่าง ๆ ในช่วง 8.4 กิโลเมตร ยิ่งไปกว่านั้นในปริญญานิพนธ์นี้ยังได้นำเสนอผลการทดสอบการแจ้งเตือนผ่าน Line Notify

คำสำคัญ เทคโนโลยี LoRa, แก๊สแอมโมเนีย, การตรวจวัด, แดชบอร์ด, Ubidots

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Wireless Ammonia Detection System Using LoRa Technology
Authors	Mr.Triphob Sawatdiphol
	Mr.Muangman Kritpakorndech
	Mr.Anatpong Techo
Thesis Advisors	Asst. Prof. Dr. Krit Smerpitak
	Asst. Prof. Dr. Apinai Rerkratn
Academic Year	2018

ABSTRACT

This thesis presents an implementation method of a 3-level detection system of ammonia concentration for remote displays on a cloud dashboard by using LoRa wireless technology. Three identical sensor nodes for sensing and signal conditioning functions are built. The measured ammonia concentration values from each sensor node are sent to the receiving part, which converts the measured data from LoRa codes into MQTT protocol to record on the Ubidots cloud server. Three colors (green, yellow, and red) are utilized for displaying the measured ammonia concentration values in three different ranges (less than 5 ppm, between 25 and 100 ppm, greater than 100 ppm) on the created cloud dashboard. In addition, the 'Line Notify' application is used to alert when the measured ammonia concentration value from any sensor node is above the specific value. Experimental results from various line-of-sight distances within 8.4 km confirm that the proposed wireless detection system can display the measured data on cloud dashboard correctly. Moreover, the test results to show the alert message via 'Line Notify' are also included.

Keywords: LoRa technology, Ammonia Gas, Detection, Dashboard, Ubidots

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยความกรุณาจาก ผศ.ดร.กฤษณ์ เสมอพิทักษ์ และ ผศ.ดร.อภินัย ฤกษ์รัตน์ อาจารย์ที่ปรึกษา ที่ได้ให้คำแนะนำ ข้อเสนอแนะ ตลอดจนแก้ไขข้อบกพร่องต่าง ๆ มาโดยตลอดจนโครงงานเล่มนี้เสร็จสมบูรณ์ ขอขอบคุณ บิดา มารดา เพื่อน ๆ ที่คอยช่วยเหลือและเป็นกำลังใจที่ดีเสมอมา สุดท้ายนี้ขอขอบคุณอาจารย์และบุคลากร ภาควิชาวิศวกรรมอัตโนมัติทุกท่านตลอดระยะเวลา 4 ปี สำหรับความรู้ ประสบการณ์ คำปรึกษาคำแนะนำ ทั้งเรื่องการเรียนและแนวคิดในการดำเนินชีวิต ที่สามารถนำไปใช้ในการทำโครงงานได้และสำเร็จลุล่วงไปได้ด้วยดี จึงขอขอบคุณไว้ ณ ที่นี้ หากมีข้อผิดพลาดประการใดให้ถือเป็นความบกพร่องของคณะผู้จัดทำและขออภัยมา ณ ที่นี้ด้วย

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย ii

บทคัดย่อภาษาอังกฤษ iii

กิตติกรรมประกาศ iv

สารบัญ v

สารบัญตาราง viii

สารบัญรูป ix

บทที่1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์ 1

1.2 วัตถุประสงค์ของปริญญานิพนธ์ 2

1.3 ขอบเขตของปริญญานิพนธ์ 2

1.4 วิธีการดำเนินโครงงาน 2

1.5 ประโยชน์ที่คาดว่าจะได้รับ 3

บทที่ 2 ทฤษฎีหรืองานวิจัยที่เกี่ยวข้อง 4

2.1 กล่าวนำ 4

2.2 เซนเซอร์วัดค่าความเข้มข้นแอมโมเนีย 4

2.3 Arduino Plus LoRa SX1278 6

2.4 LoRa 7

2.4.1 RSSI 9

2.5 โมดูล LoRa 9

2.6 Arduino IDE 10

2.7 NodeMCU 11

2.8 Raspberry Pi 13

2.9 NODE RED & JSON 14

2.9.1 Flow-based Programming 14

2.9.2 Runtime/Editor 14

2.9.3 Built on Node.js 15

2.9.4 Social Development 15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.9.5 Node - RED Dashboard 16
2.9.6 JSON (JavaScript Object Notation) 17
2.10 IoT [2] 17
2.11 Cloud MQTT [10] 19

บทที่ 3 วิธีการดำเนินงาน 21
3.1 กล่าวนำ 21
3.2 โครงสร้างของระบบ 21
3.3 การสร้างเซนเซอร์โนด 22
3.3.1 อุปกรณ์สำหรับการจัดทำเซนเซอร์โนด 22
3.3.2 การเชื่อมต่อสายของเซนเซอร์โนด 22
3.4 การสร้างกล่องรับสัญญาณ 25
3.4.1 เตรียมอุปกรณ์สำหรับการสร้าง 25
3.4.2 การเชื่อมต่อสายของกล่องรับสัญญาณ 25
3.5 การเขียนชุดคำสั่ง 26
3.5.1 เขียนชุดคำสั่งที่ฝังเซนเซอร์โนด 26
3.5.2 เขียนชุดคำสั่งที่ฝังกล่องรับสัญญาณ 29
3.6 การลง Node - Red ฝังใน Raspberry Pi 30
3.6.1 ขั้นตอนการลง Node - Red ฝังใน Raspberry Pi 30
3.6.2 การสร้าง Flow ภายใน Node-Red 33
3.7 การออกแบบและสร้างแดชบอร์ด 34
3.7.1 ออกแบบและสร้างแดชบอร์ดแสดงข้อมูลภาพรวม 34
3.7.2 ออกแบบและสร้างแดชบอร์ดของแต่ละโนด 36

บทที่ 4 ผลการดำเนินงาน 38
4.1 กล่าวนำ 38
4.2 ผลการทดสอบเซ็นเซอร์โนด 38
4.3 ผลการทดสอบการสื่อสารของโมดูล LoRa 40
4.4 ผลการทดสอบข้อมูลระหว่างเซนเซอร์โนดและกล่องรับสัญญาณ 43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

4.5 ผลการแสดงผ่านเว็บเพจของ Ubidots 45
4.6 ผลการแจ้งเตือนค่าความเข้มข้นแก๊สแอมโมเนียผ่าน Line Notify 48

บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ 51
5.1 สรุปผลการดำเนินงาน 51
5.2 ข้อเสนอแนะ 51

เอกสารอ้างอิง 52

ภาคผนวก 53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

1.1 แผนการดำเนินงานโครงงาน 3

3.1 ตารางแสดงเงื่อนไขการสั่ง Output Status LED 36

4.1 ผลการทดลองปรับระดับแรงดันจาก Potentiometer 39

4.2 ผลการทดสอบการสื่อสารของโมดูล LoRa 43

4.3 ผลการทดสอบความถูกต้องของข้อมูลระหว่างเซนเซอร์โนดที่ 1 และกล่องรับสัญญาณ 43

4.4 ผลการทดสอบความถูกต้องของข้อมูลระหว่างเซนเซอร์โนดที่ 2 และกล่องรับสัญญาณ 44

4.5 ผลการทดสอบความถูกต้องของข้อมูลระหว่างเซนเซอร์โนดที่ 3 และกล่องรับสัญญาณ 45

4.6 ตารางสรุปผลค่าบนเว็บเพจ Ubidots หน้าภาพรวมแดชบอร์ด 46

4.7 ตารางสรุปผลค่าบนเว็บเพจ Ubidots หน้าแดชบอร์ดแต่ละเซนเซอร์โนด 47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 เซนเซอร์วัดค่าความเข้มข้นแอมโมเนีย Model MQ137 4
2.2 คุณสมบัติของเซนเซอร์ตรวจวัดค่าความเข้มข้นแอมโมเนีย 4
2.3 กราฟผลตอบสนองของเซนเซอร์วัดแก๊สแอมโมเนีย 5
2.4 กราฟความเข้มข้นของแก๊สแอมโมเนียด้วยโปรแกรม Excel 5
2.5 ระดับความอันตรายที่ระดับความเข้มข้นแอมโมเนียค่าต่าง ๆ 6
2.6 Arduino Plus LoRa SX1278 (ATmega328p) 6
2.7 Pinout บอร์ด Arduino Plus LoRa SX1278 7
2.8 Data Rate of LPWAN 8
2.9 Modulation Setting 8
2.10 โมดูล LoRa 9
2.11 โมดูล LoRa Pinout 10
2.12 โปรแกรม Arduino IDE 11
2.13 NodeMCU 11
2.14 ESP8266 (ESP-12E) 12
2.15 NodeMCU Pinout 12
2.16 Raspberry Pi 3 Model B+ 13
2.17 Browser-based flow editing 15
2.18 Flow ใน Node - RED 16
2.19 Node – RED Dashboard 16
2.20 JSON Format 17
2.21 IoT With Big Data 18
2.22 Cloud MQTT 19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ หน้า

3.1 โครงสร้างของระบบ 21
3.2 การเชื่อมต่อสายระหว่างโมดูล LoRa กับบอร์ด Arduino Plus LoRa SX1278 22
3.3 การเชื่อมต่อเซนเซอร์วัดความเข้มข้นแก๊สแอมโมเนียกับบอร์ด Arduino 23
3.4 เสาอากาศ LoRa 824MHz-2170MHz สำหรับขยายสัญญาณ 23
3.5 เซนเซอร์โนดวัดค่าความเข้มข้นแอมโมเนีย 24
3.6 เซนเซอร์โนดวัดค่าความเข้มข้นแอมโมเนียพร้อมใช้งาน 24
3.7 การเชื่อมต่อสายระหว่าง NodeMCU กับโมดูล LoRa 25
3.8 การเชื่อมต่อระหว่าง NodeMCU กับ Raspberry Pi 3 Model B+ 26
3.9 กล่องรับสัญญาณ 26
3.10 ภาพรวมของเซ็นเซอร์โนดกับกล่องรับสัญญาณ 27
3.11 ชุดคำสั่งตั้งค่า Parameter LoRa 27
3.12 ชุดคำสั่งอ่านข้อมูลจากเซนเซอร์วัดความเข้มข้นแอมโมเนีย 28
3.13 แสดงข้อมูลผ่าน Serial Monitor 28
3.14 ชุดคำสั่งเพื่อจัดการข้อมูลก่อนที่ทำการส่งข้อมูลจากเซนเซอร์โนด 28
3.15 การจัดข้อมูลในรูปแบบ JSON Format 29
3.16 ชุดคำสั่งสำหรับการส่งข้อมูลของเซนเซอร์โนด 29
3.17 ชุดคำสั่งตั้งค่า Parameter ฝั่งกล่องรับสัญญาณ 30
3.18 ชุดคำสั่งรับและส่งข้อมูลแบบ Serial Communication ของกล่องรับสัญญาณ 30
3.19 โปรแกรม AnyDesk 31
3.20 เมื่อเปิดโปรแกรม AnyDesk 31
3.21 คำสั่งสำหรับการตรวจสอบ มี-ไม่มี Node - RED 32
3.22 การเข้าใช้งาน Node – RED 32
3.23 ภาพรวมของ Flow ภายใน Node – RED 33
3.24 Flow จัดการรับและส่งข้อมูล 33
3.25 การเขียนโปรแกรมและเซ็ตตัวแปรภายใน Flow 33
3.26 การเขียน Flow แจ้งเตือนผ่าน Line Notify 34
3.27 การสร้าง Variable ภายใน Ubidots 34
3.28 หน้ารวมของการสร้างตัวแปรภายใน Ubidots 35
3.29 Ubidots Widgets 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.30 เครื่องมือ Gauge Indicator	35
3.31 เครื่องมือ On/Off Indicator	36
3.32 เครื่องมือ Slider Control	36
3.33 เครื่องมือ Historical Table	37
3.34 เครื่องมือ Line Chart (Trend)	37
3.35 เครื่องมือ Map	37
4.1 ผลการทดลอง Rs/Ro ที่ 0.59	38
4.2 ผลการทดลอง Rs/Ro ที่ 0.32	38
4.3 ผลการทดลอง Rs/Ro ที่ 0.22	38
4.4 ผลการทดลอง Rs/Ro ที่ 0.18	39
4.5 ผลการทดลอง Rs/Ro ที่ 0.15	39
4.6 จุดตั้งกล่องรับสัญญาณ ณ ชั้นดาดฟ้าตึก 12 ชั้นสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	40
4.7 RSSI ที่ระยะ 100 เมตร	40
4.8 RSSI ที่ระยะ 300 เมตร	41
4.9 RSSI ที่ระยะ 600 เมตร	41
4.10 RSSI ที่ระยะ 1000 เมตร	41
4.11 ภาพรวมการทดสอบระยไม่เกิน 1000 เมตร	41
4.12 จุดทดสอบที่ 8400 เมตรความสูง 30 เมตรที่ป่าในกรุงเขตประเวศ	42
4.13 RSSI ที่ระยะ 8400 เมตร	42
4.14 ภาพรวมการทดสอบระยะ 8400 เมตร	42
4.15 แดชบอร์ดแสดงข้อมูลในภาพรวม	46
4.16 แดชบอร์ดแต่ละเซนเซอร์โนด	46
4.17 ค่าความเข้มข้นแก๊สแอมโมเนีย < 25 ppm (ไฟสีเขียวติด)	47
4.18 ค่าความเข้มข้นแก๊สแอมโมเนีย >= 25 ppm และ < 100 ppm (ไฟสีเหลืองติด)	47
4.19 ค่าความเข้มข้นแก๊สแอมโมเนีย >= 100 ppm (ไฟสีแดงติด)	47
4.20 สถานะการเปิดใช้งานของแต่ละเซนเซอร์โนด	48
4.21 สถานะการปิดใช้งานของแต่ละเซนเซอร์โนด	48
4.22 การกำหนดค่าแจ้งเตือน 25 ppm	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ หน้า

4.23 การแจ้งเตือนผ่าน Line Notify เกิน 25 ppm 49
4.24 การกำหนดค่าแจ้งเตือน 50 ppm..... 49
4.25 การแจ้งเตือนผ่าน Line Notify เกิน 50 ppm 49
4.26 การกำหนดค่าแจ้งเตือน 100 ppm 50
4.27 การแจ้งเตือนผ่าน Line Notify เกิน 100 ppm 50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1
บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

แก๊สแอมโมเนียเป็นสารเคมีพื้นฐานที่มีประโยชน์อย่างมากและมีการใช้อย่างแพร่หลายทั้งในภาคเกษตรกรรมและอุตสาหกรรม ในขณะเดียวกันการใช้งานและการจัดเก็บก็อาจก่อให้เกิดความเสียหายต่อชีวิต ทรัพย์สิน และสิ่งแวดล้อมได้หากขาดความระมัดระวัง แก๊สแอมโมเนียสามารถเข้าสู่ร่างกายทางการหายใจและทางการสัมผัสไอระเหย แก๊สแอมโมเนียสามารถละลายในน้ำได้และให้สารละลายแอมโมเนียมไฮดรอกไซด์ออกมา สารละลายแอมโมเนียก่อให้เกิดความระคายเคืองอย่างมากต่อเยื่อเมือกบุผิว ตา และผิวหนัง อาการเฉียบพลันเมื่อสัมผัสไอระเหยของแก๊สทำให้ระคายเคืองที่เยื่อบุตา มีอาการน้ำตาไหล หนังตากระตุก และผิวหนังอาจไหม้ แอมโมเนียจะทำให้เกิดอาการปวดศีรษะ น้ำลายออกมาก ปวดแสบปวดร้อนบริเวณทรวงอก ประสาทหมกลิ่นเสียไป เหงื่อออกมาก คลื่นไส้ อาเจียน และอาจทำให้เกิดการระคาย ส่วนอาการแบบเรื้อรังอาจมีตามมาจากอาการพิษแบบเฉียบพลันแต่ค่อนข้างน้อย ส่วนใหญ่มักเกิดอาการระคายเคืองที่ทางเดินหายใจส่วนบน ซึ่งอาการต่างๆที่กล่าวมานั้นความรุนแรงของอาการที่เกิดขึ้นจะขึ้นอยู่กับระดับความเข้มข้นของแก๊สแอมโมเนียและระยะเวลาที่สัมผัสที่ได้รับสู่ร่างกาย[1] บริเวณที่มีแก๊สแอมโมเนียจึงควรมีเครื่องตรวจวัดแก๊สแอมโมเนียแต่เนื่องจากความเสี่ยงในการเข้าไปทำการตรวจวัดหรืออ่านค่าการวัดอาจจะทำให้เกิดความเสียหายต่อชีวิตได้ ซึ่งหากมีการวัดแก๊สแอมโมเนียจากระยะไกลนั้นจะเกิดประโยชน์ต่อทุกภาคการใช้งานได้เป็นอย่างมาก

จากที่ได้มีการกล่าวมาข้างต้นทำให้คณะผู้จัดทำมีความต้องการที่จะทำระบบตรวจวัดแก๊สแอมโมเนียแบบไร้สายในระยะไกลซึ่งทำให้ผู้ตรวจวัดไม่จำเป็นต้องเข้าไปอยู่ในบริเวณที่มีแก๊สแอมโมเนีย อีกทั้งในบางพื้นที่อาจอยู่ในบริวณที่ไม่มีการเข้าถึงของเครือข่ายอินเทอร์เน็ต (Internet) ด้วยเทคโลยี LoRa (Long Range) ซึ่งเป็นเทคโนโลยีการส่งข้อมูลแบบไร้สายในระยะไกล มีย่านความถี่ 920-925 MHz สามารถใช้งานได้ โดยที่ไม่ต้องขออนุญาต[3] เพื่อแก้ปัญหาเหล่านี้และตอบสนองตรงต่อความต้องการของผู้ใช้งานที่ต้องการใช้งานส่งข้อมูลได้โดยไม่จำเป็นต้องมีอินเทอร์เน็ตที่เซนเซอร์โนด (Sensor Node) ที่ใช้ตรวจวัดความเข้มข้นของแก๊สแอมโมเนียให้สามารถส่งข้อมูลแบบไร้สายในระยะไกลมายังกล่องรับสัญญาณ (Gateway) ซึ่งมีความสามารถเชื่อมต่อเข้ากับเครือข่ายอินเทอร์เน็ตทำให้สามารถประยุกต์เข้ากับแนวคิด Internet of Things (IoT) ซึ่งผู้ใช้ที่ต้องการทำการตรวจวัดและอ่านค่าความเข้มข้นของแก๊สแอมโมเนียสามารถตรวจสอบข้อมูลผ่านเครือข่ายอินเทอร์เน็ตได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของปริญญานิพนธ์

สร้างระบบตรวจวัดแก๊สแอมโมเนียแบบไร้สายเพื่อเฝ้าระวังค่าความเข้มข้นของแก๊สแอมโมเนียจากระยะไกลโดยใช้เทคโนโลยี LoRa และการ Monitor ค่าความเข้มข้นของแก๊สแอมโมเนียพร้อมทั้งมีการแจ้งเตือนความอันตราย 3 ระดับ บนเว็บเพจของระบบคลาวด์ Ubidots

1.3 ขอบเขตของปริญญานิพนธ์

1. สร้างชุดเซนเซอร์โนดเพื่อตรวจวัดแก๊สแอมโมเนียเพื่อส่งข้อมูลที่ตรวจวัดได้จากเซนเซอร์ไปยังกล่องรับสัญญาณทั้งหมดจำนวน 3 ชุด โดยแต่ละชุดประกอบด้วย เซนเซอร์ตรวจวัดค่าความเข้มข้นแอมโมเนีย Model MQ137 ซึ่งมีพิสัยการวัดตั้งแต่ 5-500 ppm NH3 และ Arduino Plus LoRa SX1278 (ATmega328p) ที่ใช้ในการประมวลผลและแปลงค่าที่วัดได้จากเซนเซอร์เป็นค่าความเข้มข้นของแอมโมเนียและส่งข้อมูลไปยังกล่องรับสัญญาณด้วยโมดูล LoRa (SX1276) ซึ่งมีการสื่อสารแบบคลื่นวิทยุที่ย่านความถี่ 920-925 MHz

2. สร้างกล่องรับสัญญาณเพื่อแปลงสัญญาณจากเทคโนโลยี LoRa เป็นโปรโตคอล MQTT ซึ่งกล่องรับสัญญาณนี้ประกอบด้วยโมดูล LoRa ใช้สำหรับแปลงสัญญาณคลื่นวิทยุเป็นข้อมูลได้ในระยะไม่เกิน 8.4 กิโลเมตร โดยมี NodeMCU เป็นตัวประมวลผลข้อมูลที่ได้รับและส่งต่อข้อมูลผ่าน Serial Communication ไปยัง Raspberry Pi 3 B+ ที่มีการติดตั้ง Node-Red สำหรับการจัดการและส่งข้อมูลไปยังระบบคลาวด์ Ubidots (Cloud Computing Service)

3. การแสดงผลข้อมูล ได้แก่ ค่าความเข้มข้นของแก๊สแอมโมเนีย ค่าแรงดันไฟฟ้าจากเซนเซอร์ ค่าความเข้มของสัญญาณ (Received Signal Strength Indicator : RSSI) และสถานะการเปิดใช้งานของฝั่งผู้ส่งและการกำหนดค่า Set point ของระดับความเข้มข้นของแก๊สแอมโมเนียผ่านเว็บเพจของระบบคลาวด์ Ubidots

4. การจัดทำระบบการแจ้งเตือนระดับความเข้มของแก๊สแอมโมเนียที่มีการกำหนดค่าแจ้งเตือนเกินที่ผู้ใช้กำหนดผ่านเว็บเพจของระบบคลาวด์ Ubidots และมีการเพิ่มฟังก์ชันการแจ้งเตือนผ่านทาง Line Notify โดยที่การแจ้งเตือนจะถูกส่งไปยังไลน์ไอดีที่ถูกระบุไว้ภายในส่วนของ Node-Red

1.4 วิธีการดำเนินโครงงาน

1. เลือกหัวข้อในการจัดทำโครงงาน
2. ศึกษาและทำความเข้าใจโครงงานในส่วนของ Hardware และ Software
3. ออกแบบการทำงานของระบบและการรับ-ส่งข้อมูลของเทคโนโลยี LoRa
4. เขียนชุดคำสั่งการทำงานของเซนเซอร์โนดและกล่องรับสัญญาณปรับแก้ตามปัญหาที่เจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ศึกษาการออกแบบและสร้างส่วนของการแสดงผลบนเว็บเพจของระบบคลาวด์ Ubidots เพื่อความสวยงามและเป็นมาตรฐาน

6. สร้างส่วนเก็บข้อมูลหรือ Database เพื่อเก็บเป็นฐานข้อมูลและเพื่อการนำข้อมูลมาใช้ได้ง่ายขึ้น

7. ทดสอบและติดตามการใช้งานของโครงงาน

จากขั้นตอนการดำเนินงานข้างต้น สามารถสรุปเป็นแผนการดำเนินงานดังตารางที่ 1.1

ตารางที่ 1.1 แผนการดำเนินงานโครงงาน

ลำดับ	แผนการดำเนินงาน	กันยายน				ตุลาคม				พฤศจิกายน				ธันวาคม			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	เลือกหัวข้อในการจัดทำ																
2	ศึกษาและทำความเข้าใจโครงงาน																
3	ออกแบบการทำงานของระบบ																
4	เขียนชุดคำสั่งการทำงานของเซนเซอร์โนดกล่องรับสัญญาณ																
5	ออกแบบและสร้างส่วนแสดงผลบนระบบคลาวด์ของ Ubidots																
6	สร้างส่วนเก็บข้อมูล																
7	ทดสอบและติดตามการใช้งานของโครงงาน																

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ผู้ใช้สามารถเฝ้าระวังค่าความเข้มข้นของแก๊สแอมโมเนียและสถานะการเปิดใช้งานของเซนเซอร์โนดแต่ละหน่วยผ่านบนเว็บเพจของระบบคลาวด์ Ubidots

2. ผู้ใช้สามารถกำหนดค่าแจ้งเตือนระดับค่าความเข้มข้นของแก๊สแอมโมเนียผ่านบนเว็บเพจของระบบคลาวด์ Ubidots สำหรับการแจ้งเตือนผ่าน Line Notify

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีหรืองานวิจัยที่เกี่ยวข้อง

2.1 กล่าวนำ

ในบทนี้จะอธิบายถึงแนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง จะกล่าวถึงแหล่งความรู้ที่เกี่ยวข้องซึ่งเป็นส่วนประกอบสำคัญ โดยจะนำแหล่งความรู้นั้นๆ มาศึกษาหาความรู้เพื่อที่จะนำมาประยุกต์ใช้ร่วมกันกับการทำโครงงานนี้

2.2 เซนเซอร์วัดค่าความเข้มข้นแอมโมเนีย [5]

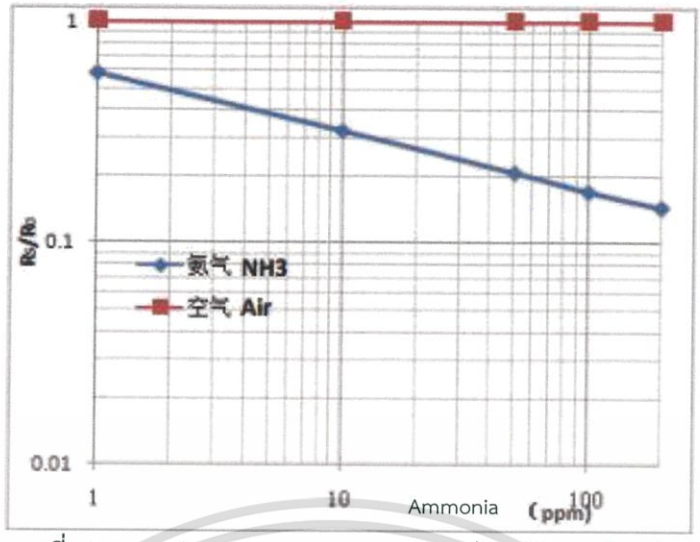
รูปที่ 2.1 เซนเซอร์วัดค่าความเข้มข้นแอมโมเนีย Model MQ137

ในการทำโครงงานครั้งนี้ผู้จัดทำเลือกใช้เซนเซอร์วัดค่าความเข้มข้นแอมโมเนียเนื่องจากแอมโมเนียนั้นมีความอันตรายต่อมนุษย์และสิ่งมีชีวิตอื่นๆ โดยส่วนต่อไปจะทำการกล่าวเกี่ยวกับคุณสมบัติของเซนเซอร์ตรวจวัดค่าความเข้มข้นแก๊สแอมโมเนียและหลักการทำงานต่างๆ

Model			MQ137
Sensor Type			Semiconductor
Standard Encapsulation			Bakelite, Metal cap
Target Gas			Ammonia Gas(NH3)
Detection range			5~500ppm NH3
Standard Circuit Conditions	Loop Voltage	V_c	≤24V DC
	Heater Voltage	V_H	5.0V±0.1V AC or DC
	Load Resistance	R_L	Adjustable
Sensor character under standard test conditions	Heater Resistance	R_H	29Ω±3Ω(room tem.)
	Heater consumption	P_H	≤900mW
	Sensitivity	S	Rs(in air)/Rs(50ppmNH_3)≥2
	Output Voltage	ΔVs	≥0.5V (in 50ppm NH_3)
	Concentration Slope	α	≤0.6(R_{200ppm}/R_{50ppm} NH_3)
Standard test conditions	Tem. Humidity		20℃±2℃; 55%±5%RH
	Standard test circuit		Vc:5.0V±0.1V; V_H: 5.0V±0.1V
	Preheat time		Over 48 hours

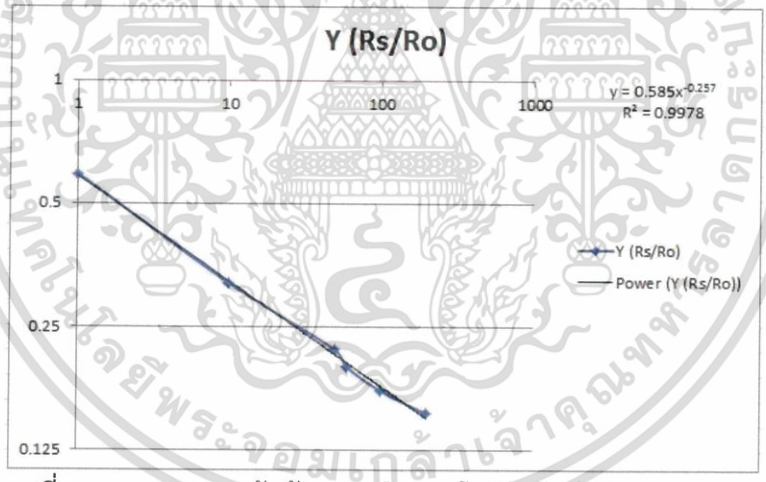
รูปที่ 2.2 คุณสมบัติของเซนเซอร์ตรวจวัดค่าความเข้มข้นแอมโมเนีย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 กราฟผลตอบสนองของเซนเซอร์วัดแก๊สแอมโมเนีย

จากรูปที่ 2.3 กราฟบอกถึงการทำงานของการวัดค่าแอมโมเนียภายในในอากาศและความไวต่อการสัมผัสของเซนเซอร์ในรูปแบบของกราฟ Logarithm และเนื่องจากผู้จัดทำต้องการสมการสำหรับการใช้คำนวณค่าความเข้มข้นของแอมโมเนีย จึงทำการสร้างกราฟในโปรแกรม Excel และได้ทำการถอดสมการได้ดังรูปที่ 2.4



รูปที่ 2.4 กราฟความเข้มข้นของแก๊สแอมโมเนียด้วยโปรแกรม Excel

สมการจากกราฟคือ $Y = 0.585x^{(-0.257)}$ (1)

โดยที่ Y คือ Rs/Ro

X คือ ค่าความเข้มข้นแก๊สแอมโมเนีย (ppm)

จากรูปที่ 2.4 นั้นเมื่อได้สมการแล้วจึงนำมาเขียนโปรแกรมคำนวณหาค่าแอมโมเนียจากการเปลี่ยนแปลงของอัตราส่วน Rs/Ro ที่เซนเซอร์วัดแอมโมเนียนั้นส่งข้อมูลมาซึ่งหน่วยที่ได้จากการคำนวณค่าแอมโมเนียนั้นมีหน่วยคือ ppm จากสมการได้ว่าความเข้มของแอมโมเนียในอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

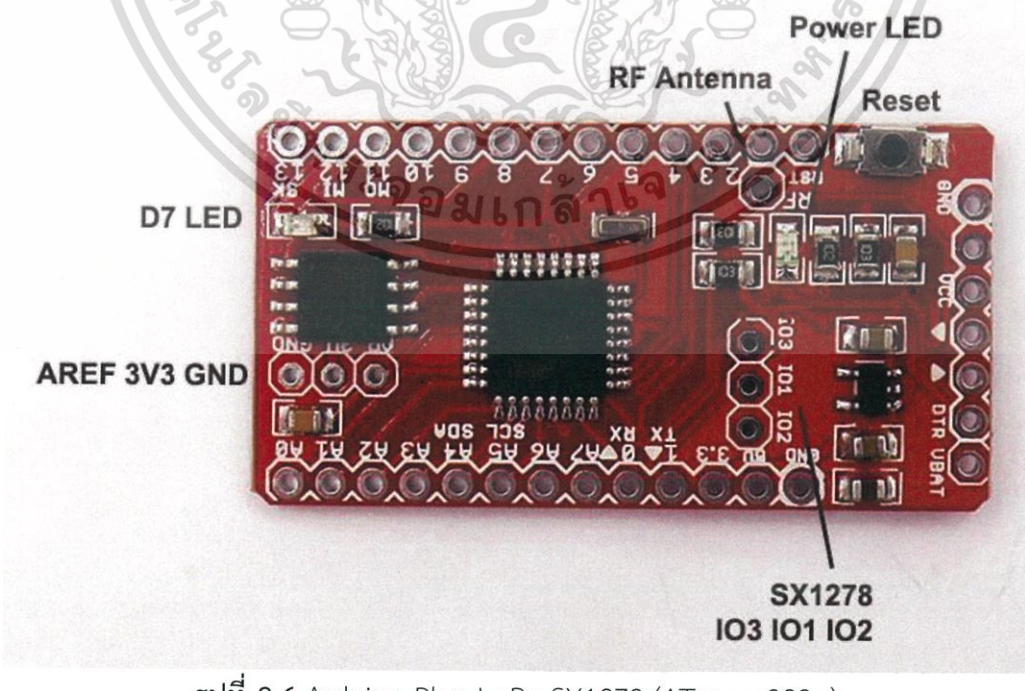
$$ppm = ((Rs/Ro)/0.585)^{(-1/0.257)}) \quad (2)$$

จากที่กล่าวมาในข้างต้นนั้นผู้จัดทำจึงอยากให้เห็นความสำคัญของความเข้มแอมโมเนียภายในอากาศที่เราใช้สูดหายใจดังรูปที่ 2.5

Intensity (ppm)	Effect	Exposure time
25	Most of people can smell.	8 hours
100	Not much affect and slight irritation.	Do not expose for a long time
400	Irritation the nose and throat.	30 minutes – 1 hour
700	Irritation the eyes.	30 minutes – 1 hour
1700	Seizures and violent irritation the eyes, nose and throat.	Can cause life if exposed more than 30 minutes
2000-5000	Irritation the throat and violent neck pain.	Can cause life if exposed more than 15 minutes
5000-10000	Muscle and respiratory system spasm causing lack of oxygen rapidly.	Can cause life if exposed more than 2-3 minutes

รูปที่ 2.5 ระดับความอันตรายที่ระดับความเข้มข้นแอมโมเนียค่าต่างๆ

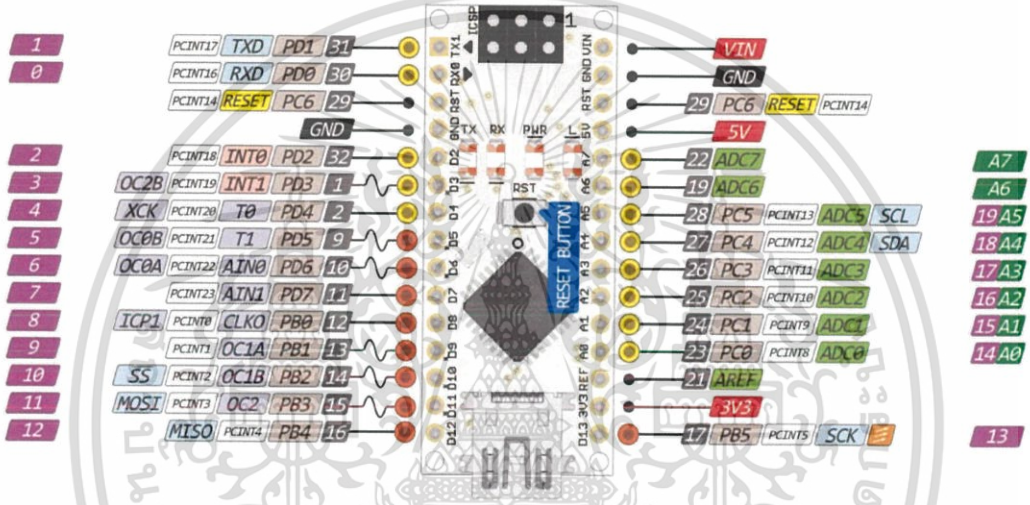
2.3 Arduino Plus LoRa SX1278 [6]



รูปที่ 2.6 Arduino Plus LoRa SX1278 (ATmega328p)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Arduino คือ บอร์ดไมโครคอนโทรลเลอร์สำเร็จรูปที่รวมเอาตัวไมโครคอนโทรลเลอร์และอุปกรณ์อื่นๆ ที่จำเป็นมาไว้ในบอร์ดเดียว และยังเปิดเผยข้อมูลทุกๆอย่าง ทั้งลายวงจรและตัวอย่างโปรแกรม ทำให้ผู้ใช้สามารถนำไปพัฒนาต่อได้ง่าย เพียงแค่มีบอร์ด Arduino กับคอมพิวเตอร์ ก็พร้อมใช้งานได้แล้ว โดยที่ไม่ต้องทำวงจรที่ซับซ้อน หรือการติดตั้งโปรแกรมที่ยุ่งยาก โดยทาง Arduino ได้ผลิตบอร์ดสำเร็จรูปออกมาหลายรุ่น หลายขนาด โดยแต่ละรุ่นก็มีข้อดีแตกต่างกันออกไป ซึ่งรุ่นที่ผู้จัดทำได้นำมาใช้คือบอร์ด Arduino Plus LoRa SX1278 ซึ่งเป็นรุ่นที่มีขนาดเล็กและนำมาประยุกต์ใช้งานกับโมดูล LoRa ได้สะดวกสบายเหมาะสำหรับผู้ที่ต้องการเริ่มต้นเขียนโปรแกรมบนไมโครคอนโทรลเลอร์ เพื่อศึกษา หรือเพื่อนำมาประยุกต์ใช้สร้างงานในด้านต่างๆ



รูปที่ 2.7 Pinout บอร์ด Arduino Plus LoRa SX1278

2.4 LoRa [3]

Long Range เป็นเทคโนโลยีการส่งข้อมูลแบบไร้สายในระยะไกล มีย่านความถี่ EU433 MHz, AS923 MHz และ 2.4 GHz ซึ่งเป็นย่านความถี่แบบ unlicensed band สามารถใช้งานได้ โดยที่ไม่ต้องขออนุญาต (ถ้ากำลังส่งไม่เกิน 500 mW) สาเหตุที่ LoRa ได้รับความนิยม เพราะมีจุดเด่น คือ กินกำลังไฟต่ำ สามารถทำงานโดยใช้แบตเตอรี่ได้หลายปี และยังส่งข้อมูลได้ไกล หลายกิโลเมตร นอกจากนี้ ยังสามารถเชื่อมต่อกับ Gateway เชื่อมต่อเข้ากับเครือข่าย Internet เรียกว่า LoRaWAN อีกด้วย

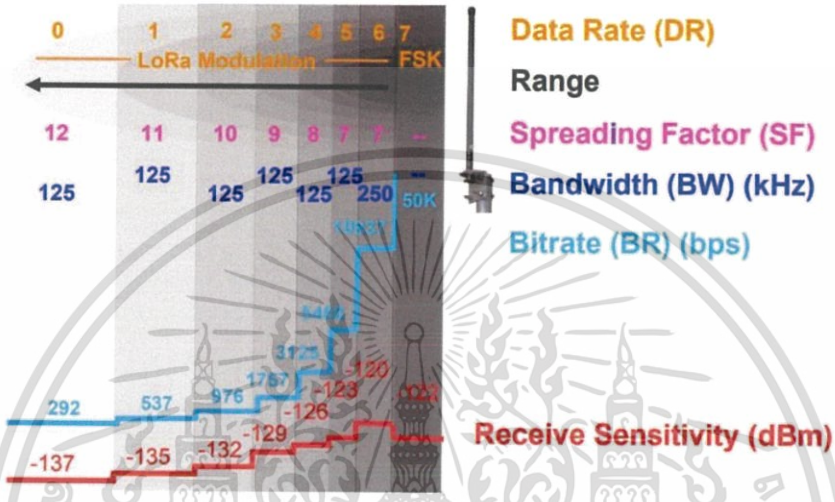
LPWAN (Low-Power Wide-Area Network) หรือเครือข่ายสื่อสารแบบกว้างที่เน้นใช้พลังงานต่ำ เป็นเทคโนโลยีที่ถือว่าเป็นหัวใจสำคัญสำหรับงานทางด้าน Internet of Things

จุดเด่นของ LPWAN เทคโนโลยี ได้แก่

- Secure Bidirectional Communication

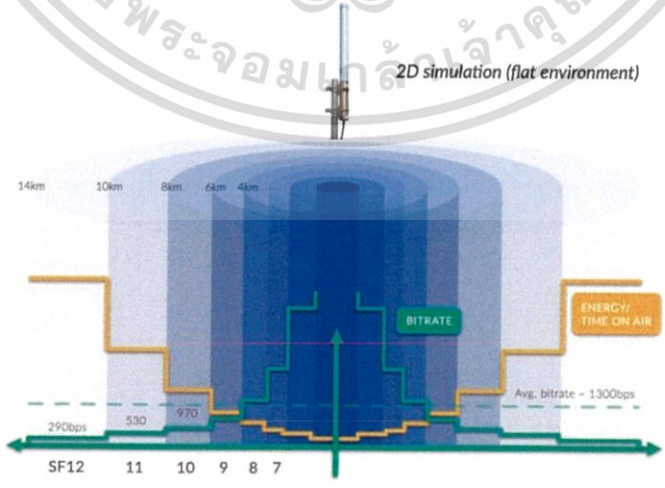
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Simple Star Network Topology (ไม่จำเป็นต้องมี network ที่ซับซ้อน เช่น Mesh หรือ Repeater)
- Low data rate
- Low cost
- Long battery life



รูปที่ 2.8 Data Rate of LPWAN

ตัวอย่างค่า Data Rate (DR) สังเกตจากรูปที่ 2.8 จะเห็นว่า DR เป็น 0 อุปกรณ์จะสามารถส่งข้อมูลได้ไกลที่สุด โดยสามารถส่งด้วย Bitrate ที่ต่ำที่สุดโดยการกำหนด Data Rate จะถูกกำหนดจาก Spreading Factor (SF) ตั้งแต่ 7– 12 โดยที่แบนวิดท์ (Bandwidth) ช่องสัญญาณและค่า SF ที่ปรับได้อาจจะเปลี่ยนแปลงตาม Frequency plan (ค่าความถี่ของสัญญาณในแต่ละโซนทวีป)



รูปที่ 2.9 Modulation Setting

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

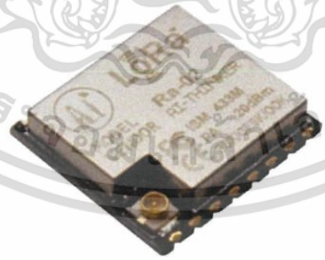
จากรูปที่ 2.9 จะเห็นว่าเมื่ออุปกรณ์เข้าใกล้ Gateway มากก็จะสามารถที่จะส่งข้อมูลด้วย Bitrate ที่สูงขึ้นได้ และการส่งข้อมูลจะเร็วขึ้นอีกด้วย รวมถึงพลังงานที่ใช้ในการส่งถ้าเทียบต่อขนาดของแพกเกตก็จะน้อยกว่าอุปกรณ์ที่อยู่ไกล Gateway อีกด้วย ซึ่งในระดับ LoRaWAN จะมีโหมด ADR (Adaptive Data Rate) ที่เซตในแพกเกตการส่งข้อมูลเพื่อให้การเชื่อมต่อระหว่าง Gateway และ Device สามารถปรับ Spreading Factor แบบอัตโนมัติเพื่อประสิทธิภาพในการส่งโดยดูจากระยะการเชื่อมต่อระหว่าง Gateway และ Device โดยสามารถเลือกได้ว่าต้องที่จะปรับเพื่อส่งข้อมูลได้เร็วที่สุดหรือปรับเพื่ออายุการใช้งานแบตเตอรี่ที่ยาวนานที่สุดเป็นต้น

2.4.1 RSSI [11]

RSSI (Received Signal Strength Indicator) เป็นการวัดความแรงหรือความเข้มของสัญญาณตัวรับ (พลังงานเป็นตัวสำคัญ) หน่วยเป็น dBm ยิ่งติดลบน้อยยิ่งดี โดยระดับความแรงของสัญญาณแบ่งเป็น 3 ระดับ

- สัญญาณมีความแรงระดับ -50 dBm ลงมาว่าเราอยู่ใกล้เสาส่งสัญญาณมากสัญญาณแรงมาก
- สัญญาณมีความแรงระดับ -60 ถึง -75 dBm ว่าสัญญาณแรง
- สัญญาณมีความแรงระดับที่ตัวเลขค่า (-) เพิ่มขึ้นไปเรื่อยๆ คุณภาพความแรงของสัญญาณก็จะลดลงจนถึงประมาณ -110 dBm เป็นต้นไป หมายความว่าคุณภาพสัญญาณต่ำ

2.5 โมดูล LoRa

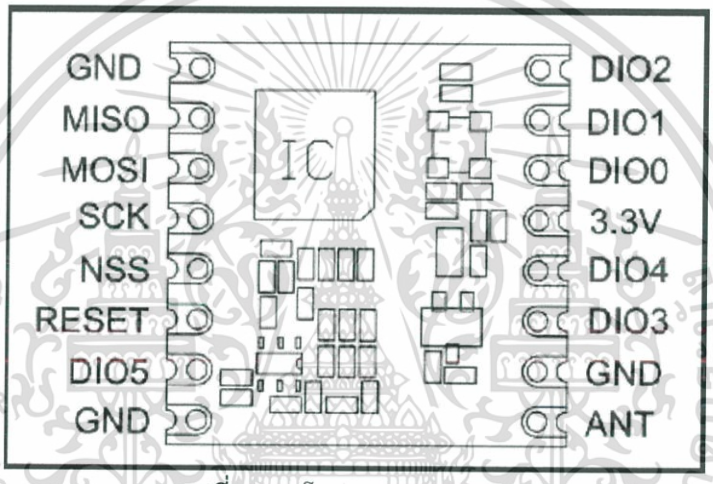


รูปที่ 2.10 โมดูล LoRa

จากรูปที่ 2.10 SX1276/77/78/79 ภายในประกอบด้วย LoRaTM Spread Modem ซึ่งมีความสามารถในการรับได้ดีตามการปรับ FSK หรือ OOK ที่อัตราการส่งข้อมูลสูงสุดของ LoRaTM ความไวคือ 8dB ดีกว่า FSK แต่การใช้รายการวัสดุที่มีต้นทุนต่ำพร้อมกับ 20ppm XTAL LoRaTM สามารถปรับปรุงความไวของผู้รับได้มากกว่า 20dB เมื่อเทียบกับ FSK LoRaTM ยังมีความก้าวหน้าที่สำคัญในการคัดสรรและประสิทธิภาพเพิ่มเติมปรับปรุงความน่าเชื่อถือในการสื่อสารเพื่อความยืดหยุ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สูงสุดผู้ใช้อาจตัดสินใจปรับการแพร่กระจายคลื่นความถี่แบนด์วิดท์ (BW) ปัจจัยการแพร่กระจาย (SF) และอัตราการแก้ไขข้อผิดพลาด (CR) ข้อดีอีกอย่างของการปรับการแพร่กระจายก็คือแต่ละปัจจัยการแพร่กระจายเป็นมุมฉาก ดังนั้นสัญญาณที่ส่งหลายข้อมูลสามารถครอบครองช่องทางเดียวกันโดยไม่รบกวนสิ่งนี้ยังอนุญาตให้อยู่ร่วมกันได้ง่ายกับระบบที่ใช้ FSK ที่มีอยู่ มาตรฐาน GFSK, FSK, OOK และ GMSK มีการปรับเพื่อให้เข้ากันได้กับระบบหรือมาตรฐานที่มีอยู่เช่น wireless MBUS และ IEEE 802.15.4g SX1276 และ SX1279 มีตัวเลือกแบนด์วิดธ์ตั้งแต่ 7.8 kHz ถึง 500 kHz พร้อมด้วยปัจจัยการกระจายตั้งแต่ 6 ถึง 12 และครอบคลุมคลื่นความถี่ที่มีอยู่ทั้งหมด SX1277 เสนอตัวเลือกแบนด์วิดท์และคลื่นความถี่เดียวกันกับปัจจัยการแพร่กระจายจาก 6 ถึง 9 SX1278 เสนอแบนด์วิดท์และตัวเลือกปัจจัยการแพร่กระจายแต่ครอบคลุมเฉพาะ UHF ที่ต่ำกว่า



รูปที่ 2.11 โมดูล LoRa Pinout

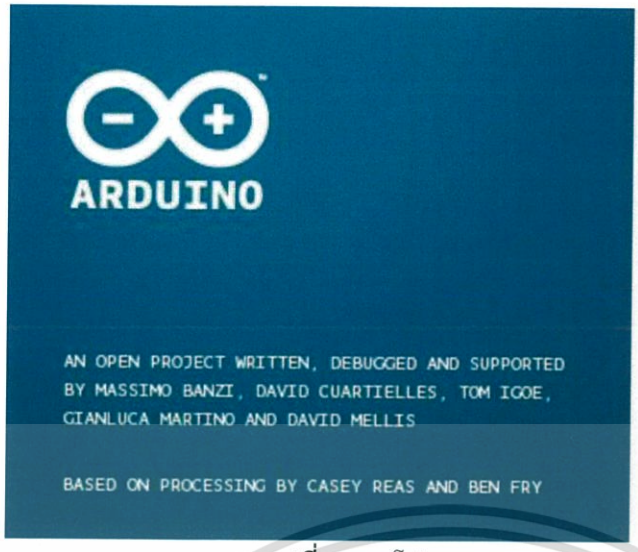
2.6 Arduino IDE [4]

Arduino IDE คือโปรแกรมสำหรับใช้เขียนโปรแกรม, คอมไพล์และดาวน์โหลดโปรแกรมลงบอร์ด Arduino หรือบอร์ดตัวอื่นๆ ที่คล้ายกัน เช่น Generic ESP8266 modules, NodeMCU หรือ WeMos D1 เป็นต้น

แนวคิดการใช้งานโปรแกรม Arduino IDE

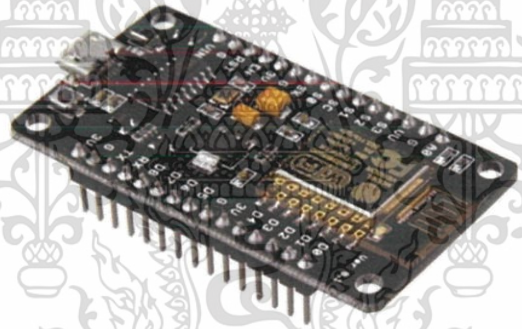
1. เขียนโปรแกรมด้วยภาษา C/C++ สำหรับ Arduino
2. คอมไพล์หรือแปลโปรแกรมภาษา C/C++ ให้เป็นภาษาสำหรับไมโครคอนโทรลเลอร์และบันทึกเป็น Intel Hex File
3. อัปโหลด Intel Hex File ลงบนไมโครคอนโทรลเลอร์ซึ่งอยู่บนบอร์ด Arduino ผ่านสาย USB หรือผ่าน Programmer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 โปรแกรม Arduino IDE

2.7 NodeMCU [8]



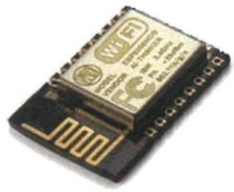
รูปที่ 2.13 NodeMCU

NodeMCU (โนด เอ็มซียู) คือ บอร์ดคล้าย Arduino ที่สามารถเชื่อมต่อกับ Wi-Fi ได้ สามารถเขียนโปรแกรมด้วย Arduino IDE ได้ เช่นเดียวกับ Arduino และบอร์ดมีราคาไม่แพง เหมาะสำหรับผู้เริ่มต้นศึกษาหรือทดลองใช้งานเกี่ยวกับ Arduino IoT อิเล็กทรอนิกส์ หรือแม้แต่การนำไปใช้จริงในโปรเจคต่างๆ

ภายในบอร์ดของ NodeMCU ประกอบไปด้วย ESP8266 (ไมโครคอนโทรลเลอร์ที่สามารถเชื่อมต่อ Wi-Fi ได้) พร้อมอุปกรณ์อำนวยความสะดวกต่างๆ เช่น พอร์ต micro USB สำหรับจ่ายไฟ ชิพสำหรับดาวน์โหลดโปรแกรมผ่านสาย USB ชิพแปลงแรงดันไฟฟ้าและขาสำหรับเชื่อมต่ออุปกรณ์ภายนอก เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ESP-12E

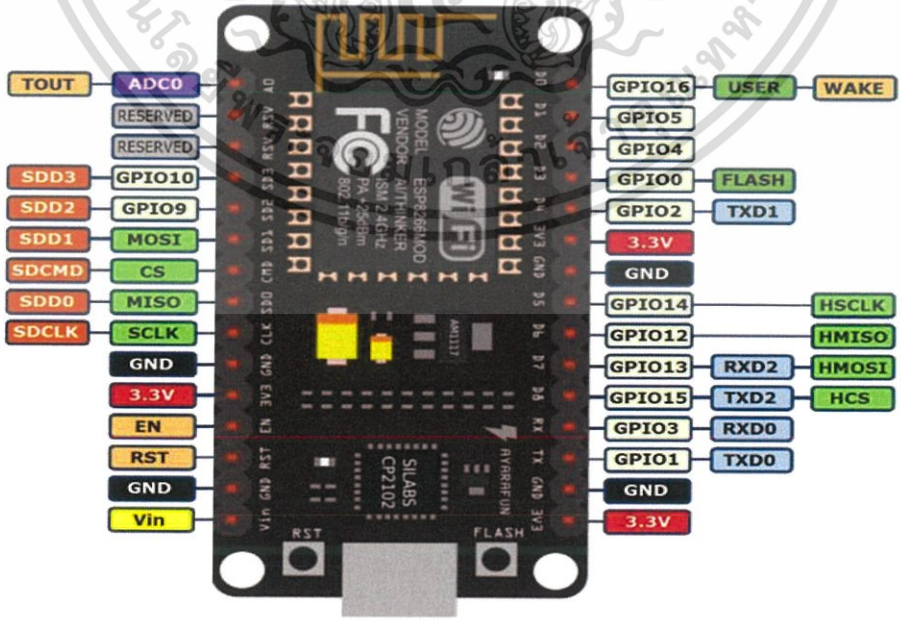


รูปที่ 2.14 ESP8266 (ESP-12E)

ESP8266 คือโมดูล Wi-Fi ที่มีความพิเศษตรงที่สามารถโปรแกรมลงไปได้ ทำให้สามารถนำไปใช้งานแทนไมโครคอนโทรลเลอร์ได้ และมีพื้นที่โปรแกรมที่มากถึง 4 MB ทำให้มีพื้นที่เหลือมากในการเขียนโปรแกรมลงไป

ESP8266 เป็นชื่อของชิปไอซีบนบอร์ดของโมดูล ซึ่งไอซี ESP8266 ไม่มีพื้นที่โปรแกรม (flash memory) ในตัว ทำให้ต้องใช้ไอซีภายนอก (external flash memory) ในการเก็บโปรแกรม ที่ใช้การเชื่อมต่อผ่านโปรโตคอล SPI ซึ่งสาเหตุนี้เองทำให้โมดูล ESP8266 มีพื้นที่โปรแกรมมากกว่าไอซีไมโครคอนโทรลเลอร์เบอร์อื่นๆ

ESP8266 ทำงานที่แรงดันไฟฟ้า 3.3 V - 3.6 V การนำไปใช้งานร่วมกับเซนเซอร์อื่นๆที่ใช้แรงดัน 5 V ต้องใช้วงจรแบ่งแรงดันมาช่วย เพื่อไม่ให้โมดูลพังเสียหาย กระแสที่โมดูลใช้งานสูงสุดคือ 200 mA ความถี่คริสตอล 40 MHz ทำให้เมื่อนำไปใช้งานอุปกรณ์ที่ทำงานรวดเร็วตามความถี่ เช่น LCD ทำให้การผลข้อมูลรวดเร็วกว่าไมโครคอนโทรลเลอร์ยอดนิยม Arduino มาก



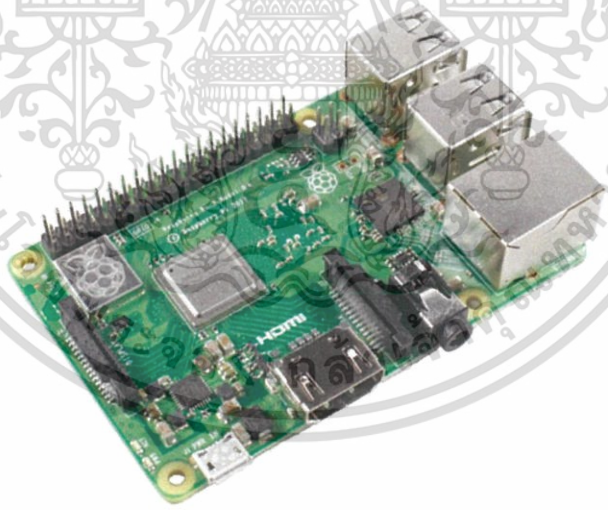
รูปที่ 2.15 NodeMCU Pinout

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 Raspberry Pi [7]

Raspberry Pi เป็นคอมพิวเตอร์ขนาดเล็กที่จัดอยู่ในกลุ่มคอมพิวเตอร์แบบฝังตัว (Embedded Computer) ถูกพัฒนาขึ้นในสหราชอาณาจักร (UK) โดย Raspberry Pi Foundation รูปแบบเดิมออกแบบเพื่อเป้าหมายในการส่งเสริมการเรียนการสอนวิทยาศาสตร์คอมพิวเตอร์ขั้นพื้นฐาน ในโรงเรียนและในประเทศกำลังพัฒนา ในปัจจุบันเป็นที่นิยมมากขึ้นกว่าเป้าหมายเดิมที่คาดการณ์ไว้ โดยขายนอกตลาดเป้าหมายสำหรับการใช้งาน เช่น หุ่นยนต์, Server, Internet of Thing ฯลฯ

Raspberry Pi ได้รับการเผยแพร่หลายรุ่นแล้ว ทุกรุ่นมีชิพ Broadcom เป็นหน่วยประมวลผลกลาง (CPU) และหน่วยประมวลผลกราฟิก (GPU) ในตัว มีความเร็วตั้งแต่ 700 MHz ถึง 1.4 GHz (1.4 GHz สำหรับ Raspberry Pi 3 รุ่น B+ โดยมี RAM ตั้งแต่ 256 MB ถึง 1 GB มีการ์ด Secure Digital (SD)) ใช้เพื่อจัดเก็บระบบปฏิบัติการและหน่วยความจำ ในขนาด SDHC หรือ MicroSDHC บอร์ดมีพอร์ต USB ตั้งแต่ 1 ถึง 4 พอร์ต มีเอาต์พุตวิดีโอ HDMI และวิดีโอคอมโพสิต โดยมีแจ็คขนาด 3.5 มม. สำหรับเอาต์พุตเสียง และ GPIO จำนวนหนึ่งซึ่งสนับสนุนโปรโตคอลทั่วไป เช่น i2C อีกทั้งในรุ่น B มีพอร์ต Ethernet และ Raspberry Pi Zero W มี Wi-Fi 802.11n และ Bluetooth ที่ติดตั้งอยู่ในตัว



รูปที่ 2.16 Raspberry Pi 3 Model B+

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 NODE RED & JSON

Node - Red เป็นเครื่องมือการเขียนโปรแกรมแบบ Flow-Based Programming เดิมทีพัฒนาโดยทีมงานบริการด้านเทคโนโลยี Emerging Technology ของไอบีเอ็มและตอนนี้เป็นส่วนหนึ่งของ JS Foundation ซึ่งยังเป็นเครื่องมือในการเขียนโปรแกรมสำหรับเชื่อมต่ออุปกรณ์ฮาร์ดแวร์ APIs และบริการออนไลน์ด้วยวิธีการใหม่ ๆ และน่าสนใจเข้าด้วยกัน โดยมีตัวแก้ไขบนเบราว์เซอร์ที่ช่วยให้สามารถเชื่อมต่อเข้าด้วยกันโดย Node ที่มีให้เลือกใช้งานอย่างหลากลาย รวมถึงจุดเด่นที่มี dashboard ที่สวยงามและมีประสิทธิภาพ ยิ่งทำให้ Node - Red เป็นเครื่องมือที่เหมาะสมมากในการแสดงผลหรือเป็น interface ที่เชื่อมต่อระหว่างผู้ใช้งาน

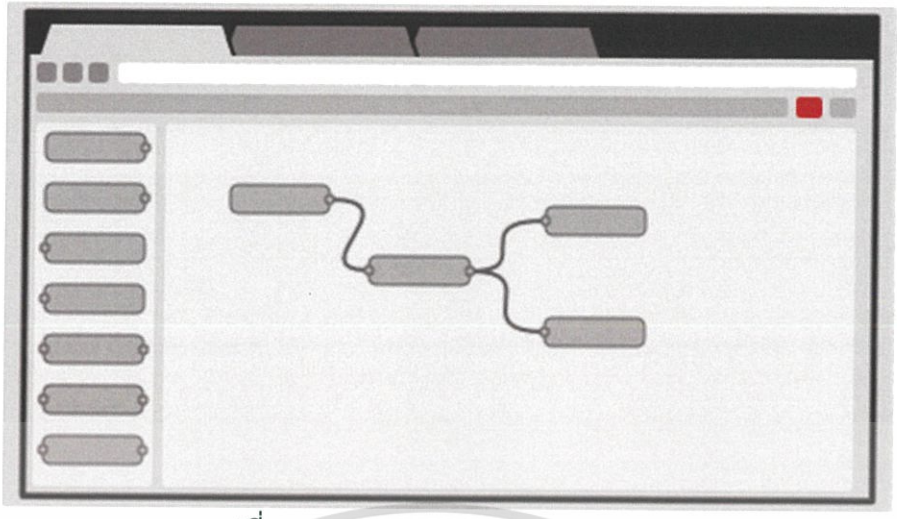
2.9.1 Flow-based Programming

การเขียนโปรแกรมแบบ Flow Flow-Based Programming และมี interface ในการใช้งานผ่านทาง Web browser ทั่วๆไป โดยการเขียนโปรแกรมนั้นเป็นรูปแบบควบคุมการไหลของข้อมูลเป็นวิธีการที่สามารถอธิบายพฤติกรรมของแอ็พพลิเคชันเป็นเครือข่ายกล่องดำหรือ Node ที่เรียกว่า Node - Red โดยแต่ละ Node มีหน้าที่ชัดเจน คือรับข้อมูลบางอย่าง กระทำการบางอย่างกับข้อมูลนั้น และส่งข้อมูลนั้นไปเครือข่ายเป็นผู้รับผิดชอบต่อการไหลของข้อมูลระหว่าง Node หากสามารถแบ่งปัญหาออกเป็นขั้นตอนได้ก็จะสามารถทำความเข้าใจได้ว่าแต่ละ Flow ทำหน้าที่อะไรโดยไม่จำเป็นต้องมีความเข้าใจโค้ดโปรแกรมใน Node แต่ละบรรทัด

2.9.2 Runtime/Editor

Node - Red นั้นทำงานบน Node.js ซึ่งเป็น Platform ตัวหนึ่งที่เขียนด้วย JavaScript สำหรับเป็น Web Server เพื่อเข้าถึงตัวแก้ไข Flow ภายในเบราว์เซอร์ที่สร้างแอปพลิเคชันโดยการลาก Node จากพาเลทไปที่พื้นที่ทำงานและเริ่มเชื่อมต่อกัน ด้วยการคลิกเพียงครั้งเดียวแอ็พพลิเคชันจะถูกนำไปรันไทม์พาเลท (Runtime Parate) ของ Node สามารถขยายได้ง่ายโดยการติดตั้ง Node ใหม่ที่สร้างขึ้นสามารถแชร์เป็นไฟล์ JSON ได้อย่างง่ายดาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 Browser-based flow editing

Node - Red มีตัวแก้ไข Flow บนพื้นฐานของเบราว์เซอร์ที่ช่วยให้สามารถเชื่อมต่อเข้าด้วยกันโดยใช้ Node ที่หลากหลายบนพาเลท สามารถรันไทม์ได้ในคลิกเดียว ฟังก์ชัน JavaScript สามารถสร้างขึ้นภายในตัวแก้ไขโดยใช้โปรแกรมแก้ไขข้อความที่มีรูปแบบ และมีไลบรารีในตัว

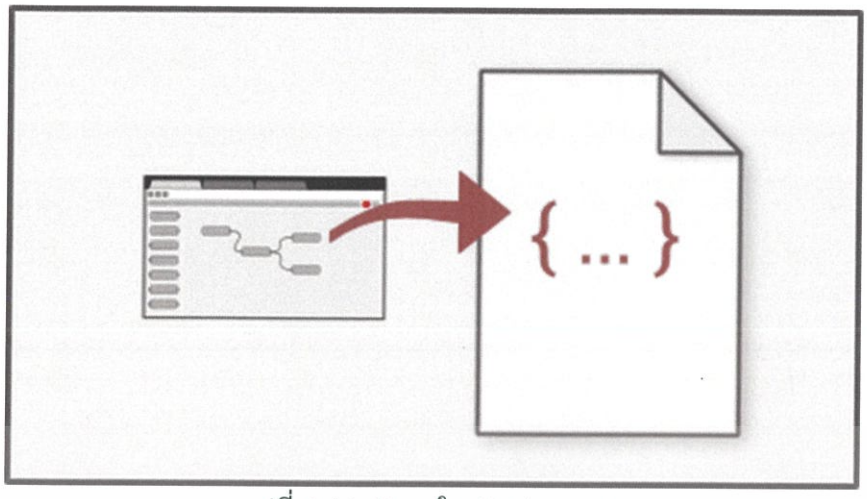
2.9.3 Built on Node.js

รันไทม์ที่มีโครงสร้างอย่างง่ายถูกสร้างขึ้นบน Node.js ทำให้เหมาะสำหรับการทำงานของเครือข่ายบนฮาร์ดแวร์ที่มีต้นทุนต่ำ เช่น Raspberry Pi รวมทั้งในระบบคลาวด์และด้วยโมดูลมากกว่า 225,000 โมดูลในที่เก็บในแพคเกจ จึงเป็นเรื่องง่ายที่จะขยายจำนวน Node เพื่อเพิ่มความสามารถใหม่ ๆให้กับมัน

2.9.4 Social Development

Flow ที่สร้างขึ้นใน Node - Red จะถูกจัดเก็บโดยใช้ JSON ซึ่งสามารถนำเข้าและส่งออกได้ง่ายเพื่อใช้ร่วมกับผู้อื่น ไลบรารีจะอยู่ในรูปแบบไลบรารีออนไลน์ช่วยให้สามารถแบ่งปัน Flow กับคนทั่วโลก

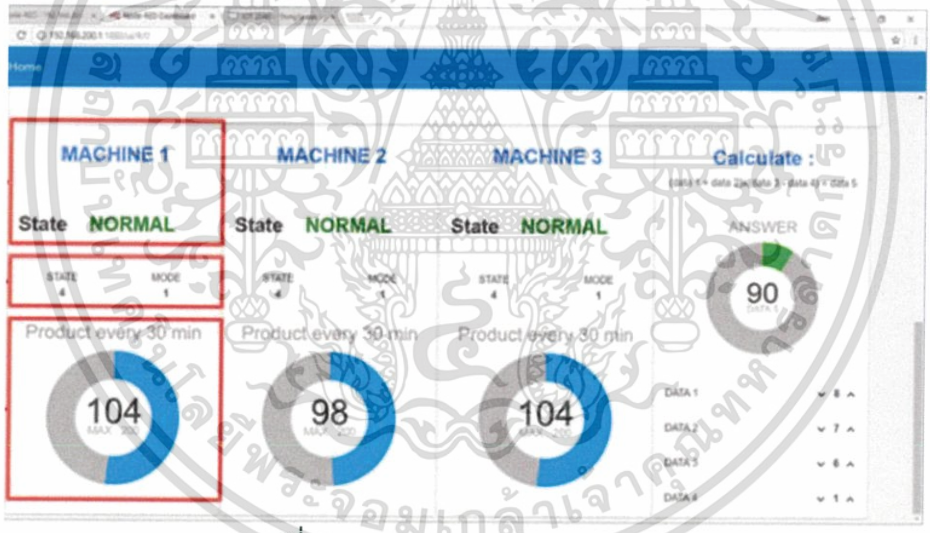
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 Flow ใน Node - RED

2.9.5 Node - RED Dashboard

การแสดงผลของ Node - RED อย่างหนึ่งก็คือ Dashboard เพราะเราสามารถใช้หน้า Dashboard นี้ในการผลต่างๆออกมาในรูปแบบ graphic ได้



รูปที่ 2.19 Node – RED Dashboard

ลักษณะการใช้งานหน้า dashboard จะประกอบด้วยส่วนสำคัญๆดังรูปข้างล่างคือ

- Theme : สีของหน้า dashboard ว่าเป็นโทน dark หรือ light
- Tab : หรือจะเรียกว่าหน้า Page ของ dashboard ก็ได้จะเข้าใจง่ายกว่า
- Group : คือส่วนย่อยๆในแต่ละ Page นั่นเอง
- Node : คือ part ของ dashboard ซึ่งเราต้องระบุว่าจะให้มันอยู่ใน Page ไหนและ Group ไหนด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.6 JSON (JavaScript Object Notation)

JSON (JavaScript Object Notation) คือ รูปแบบของข้อมูลที่ใช้สำหรับแลกเปลี่ยนข้อมูลที่มีขนาดเล็ก ซึ่งคนสามารถทำความเข้าใจได้ง่าย และสามารถถูกสร้างและอ่านโดยเครื่องได้ง่าย มันถูกกำหนดภายใต้ภาษา JavaScript (JavaScript Programming Language, Standard ECMA-262 3rd Edition – December 1999.) JSON เป็นรูปแบบข้อมูลตัวอักษรที่มีความเป็นอิสระอย่างสมบูรณ์ แต่จะมีหลักการการเขียนที่คุ้นเคยกับนักเขียนโปรแกรมภาษาต่างๆ ได้ ไม่ว่าจะเป็น ภาษา C, C++, C#, Java, JavaScript, Perl, Python และอื่นๆ คุณสมบัติเหล่านี้ทำให้ JSON เป็นภาษาแลกเปลี่ยนข้อมูลที่มีสมบูรณ์แบบ

JSON หรือ JavaScript Object Notation เป็นวิธีการที่ทำให้ JavaScript แลกเปลี่ยนข้อมูลกับ Server ได้อย่างง่าย รูปแบบของ JSON นั้น อาจทำให้หลายๆ กับมันไม่มากก็น้อย เช่น บางครั้งทำไมใช้ [] บางครั้งใช้ {} มันมีเหตุผลอะไร มีความหมายอย่างไร ถ้าจะให้อธิบายรูปแบบเป็นประโยคก็คือ JSON ถูกสร้างขึ้นจากชุดข้อมูลของ literal object notation ใน JavaScript JSON จะใช้ [] แทน array และใช้ {} แทน hash (หรือ associate array) แต่ละสมาชิกคั่นด้วย comma (,) และแต่ละ ชื่อสมาชิกคั่นด้วย colon (:)

```
{
  "todos": [
    {
      "id": 1,
      "title": "Todo 1",
      "completed": false
    },
```

รูปที่ 2.20 JSON Format

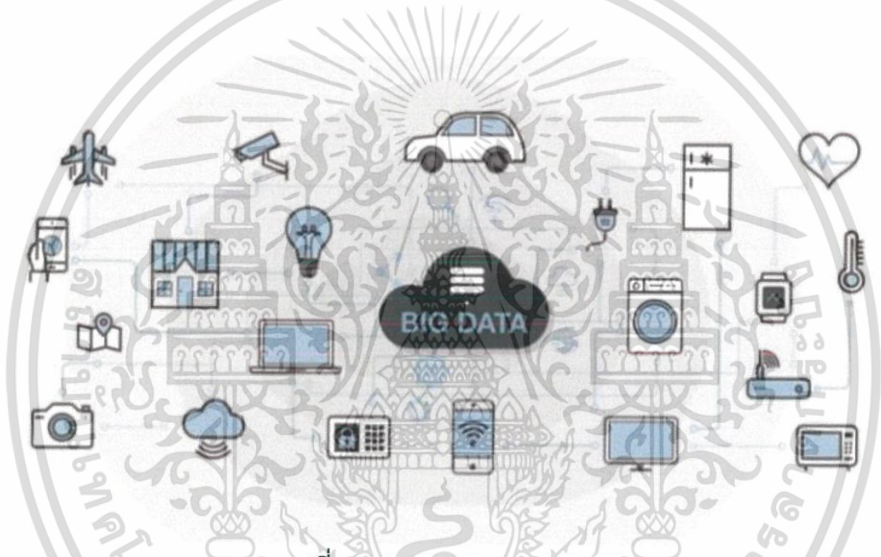
2.10 IoT [2]

Internet of Things (IoT) คือ การที่อุปกรณ์อิเล็กทรอนิกส์ต่างๆ สามารถเชื่อมโยงหรือส่งข้อมูลถึงกันได้ด้วยอินเทอร์เน็ต โดยไม่ต้องป้อนข้อมูล การเชื่อมโยงนี้ง่ายจนทำให้เราสามารถสั่งการควบคุมการใช้งานอุปกรณ์อิเล็กทรอนิกส์ต่างๆ ผ่านทางเครือข่ายอินเตอร์เน็ตได้ ไปจนถึงการเชื่อมโยงการใช้งานอุปกรณ์อิเล็กทรอนิกส์ต่างๆ ผ่านทางเครือข่ายอินเตอร์เน็ตเข้ากับการใช้งานอื่นๆ จนเกิดเป็นบรรดา Smart ต่างๆ ได้แก่ Smart Device, Smart Grid, Smart Home, Smart Network Smart Intelligent Transportation ทั้งหลายที่เราเคยได้ยินนั่นเอง ซึ่งแตกต่างจากในอดีตที่อุปกรณ์อิเล็กทรอนิกส์เป็นเพียงสื่อกลางในการส่งและข้อมูลเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล่าวได้ว่า Internet of Things นี้ได้แก่การเชื่อมโยงของอุปกรณ์อัจฉริยะทั้งหลายผ่านอินเทอร์เน็ตที่เรานึกออก เช่น แอปพลิเคชัน แว่นตากูเกิลกลาส รองเท้าวิ่งที่สามารถเชื่อมต่อข้อมูลการวิ่ง ทั้งความเร็ว ระยะทาง สถานที่ และสถิติได้

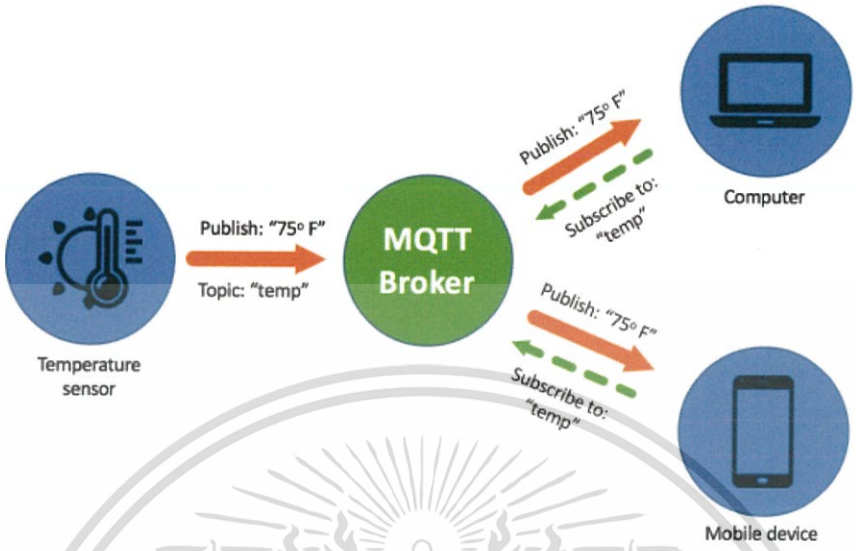
นอกจากนั้น Cloud Storage หรือ บริการรับฝากไฟล์และประมวลผลข้อมูลของคุณผ่านทางออนไลน์ หรือเรียกอีกอย่างว่า แหล่งเก็บข้อมูลบนก้อนเมฆ เป็นอีกสิ่งหนึ่งที่เราใช้งานบ่อยๆแต่ไม่รู้ว่าเป็นหนึ่งในรูปแบบของ Internet of Things สมัยนี้ผู้ใช้นิยมเก็บข้อมูลไว้ในก้อนเมฆมากขึ้น เนื่องจากมีข้อดีหลายประการ คือ ไม่ต้องกลัวข้อมูลสูญหายหรือถูกโจรกรรม ทั้งยังสามารถกำหนดให้เป็นแบบส่วนตัวหรือสาธารณะก็ได้ เข้าถึงข้อมูลได้ทุกที่ทุกเวลาด้วยอุปกรณ์อิเล็กทรอนิกส์ใดๆผ่านเครือข่ายอินเตอร์เน็ต แถมยังมีพื้นที่ใช้สอยมาก มีให้เลือกหลากหลาย ช่วยเราประหยัดค่าใช้จ่ายได้อีกด้วย เนื่องจากเราไม่ต้องเสียเงินซื้ออุปกรณ์จัดเก็บข้อมูล เช่น ฮาร์ดไดร์ฟ หรือ Flash drive ต่างๆ



รูปที่ 2.21 IoT With Big Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11 Cloud MQTT [10]



รูปที่ 2.22 Cloud MQTT

MQTT เป็นโปรโตคอลประยุกต์ที่ใช้โปรโตคอล TCP เป็นรากฐาน ออกแบบมาสำหรับงานที่ต้องการ ๆ สื่อสารแบบเรียลไทม์แบบไม่จำกัดแพลตฟอร์ม หมายถึงอุปกรณ์ทุกชิ้นสามารถสื่อสารกันได้ผ่าน MQTT

MQTT จะแบ่งเป็น 2 ฝั่ง คือฝั่งเซิร์ฟเวอร์มักจะเรียกว่า MQTT Broker ส่วนฝั่งผู้ใช้งานจะเรียกว่า MQTT Client ในการใช้งานด้าน IoT จะเกี่ยวข้องกับ MQTT Client เป็นหลัก โดยจะมี MQTT Broker ทั้งแบบฟรี และเสียเงินไว้รองรับอยู่แล้ว ทำให้การสื่อสารข้อมูลผ่าน MQTT จะใช้เซิร์ฟเวอร์ฟรี หรือ MQTT Broker ฟรี เหล่านั้นเป็นตัวกลาง

ลักษณะการใช้งาน MQTT อาจจะเปรียบเสมือนได้กับการใช้งานห้องแชท Line สำหรับอุปกรณ์ โดยอุปกรณ์แต่ละตัวจะมีชื่อเป็นของตนเอง มี Username Password เป็นของตัวเอง และอาจจะมีห้องลับเฉพาะของตนเอง ดังนั้นการใช้งาน MQTT ผู้เขียนจึงจะขอยกตัวอย่างของ MQTT เทียบกับห้องแชทได้ดังนี้

1. กลุ่มผู้ใช้ (User)

ใน MQTT จะแบ่งกลุ่มของผู้ใช้งานออกเป็น 2 ระดับ คือ

ระดับสูงสุด – สามารถที่จะรับ-ส่งข้อมูลกับอุปกรณ์ หรือช่องทางใด ๆ ก็ได้ในระบบ หรือเปรียบได้กับแอดมินที่สามารถเข้าไปดูข้อความได้ทุกห้องแม้จะเป็นห้องลับก็ตาม

ระดับทั่วไป – สามารถรับ-ส่งข้อมูลกับอุปกรณ์หรือช่องทางที่กำหนดไว้เฉพาะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เส้นทาง (Topic)

เส้นทาง เปรียบเหมือนกับหัวข้อ หรือห้องแชทที่ต้องการจะคุย และการคุยกันจะมีเฉพาะอุปกรณ์ที่อยู่ในห้องนั้น ๆ (Subscribe) ถึงจะสามารถได้รับข้อมูลที่มีการส่งไปในห้องนั้น ๆ ที่ถูกเรียกว่าเส้นทางเนื่องจากการใช้งานส่งข้อมูลและรับข้อมูลจะเหมือนกับเส้นทางในระบบไฟล์ เช่น /Room1/LED ซึ่งระบบเส้นทางนี้นอกจากอุปกรณ์จะสามารถรอการสนทนาในห้องตามเส้นทาง /Room1/LED ได้แล้ว ยังสามารถรอสนทนาเส้นทาง /Room1 ได้ด้วย หากเป็นการรอฟังในเส้นทาง (Subscribe) /Room1 จะหมายถึงการส่งข้อมูลใด ๆ ที่นำหน้าด้วย /Room1 เช่น /Room1/LED /Room1/Value ผู้ที่รอฟัง (Subscribe) /Room1 อยู่จะได้รับข้อมูลเหล่านั้นด้วยคุณภาพข้อมูล (QoS) แบ่งออกเป็น 3 ระดับดังนี้

QoS0 – ส่งข้อมูลเพียงครั้งเดียว ไม่สนใจว่าผู้รับจะได้รับหรือไม่

QoS1 – ส่งข้อมูลเพียงครั้งเดียว ไม่สนใจว่าผู้รับจะได้รับหรือไม่ แต่ให้จำค่าที่ส่งล่าสุดไว้ เมื่อมีการเชื่อมต่อใหม่จะได้รับข้อมูลครั้งล่าสุดอีกครั้ง

QoS2 – ส่งข้อมูลหลาย ๆ ครั้งจนกว่าปลายทางจะได้รับข้อมูล มีข้อเสียที่สามารถทำงานได้ช้ากว่า QoS0 และ QoS1

3. การส่งข้อมูล (Publish)

การส่งข้อมลในแต่ละครั้งจะต้องประกอบไปด้วยเส้นทาง (Topic) ข้อมูล และคุณภาพข้อมูล ซึ่งการส่งข้อมูลจะเรียกว่า Publish

4. การรับข้อมูล (Subscribe)

การรับข้อมูลในระบบ MQTT จะรับข้อมูลได้เฉพาะเมื่อมีการเรียกใช้การ Subscribe ไปยัง Topic ที่กำหนด อาจเปรียบได้กับการ Subscribe คือการเข้าไปนั่งรอเพื่อนในกลุ่ม Line ส่งแชทมาหา เมื่อมีการส่งข้อมูลเข้ามาจะเกิดสิ่งที่เรียกว่าเหตุการณ์ (Event)

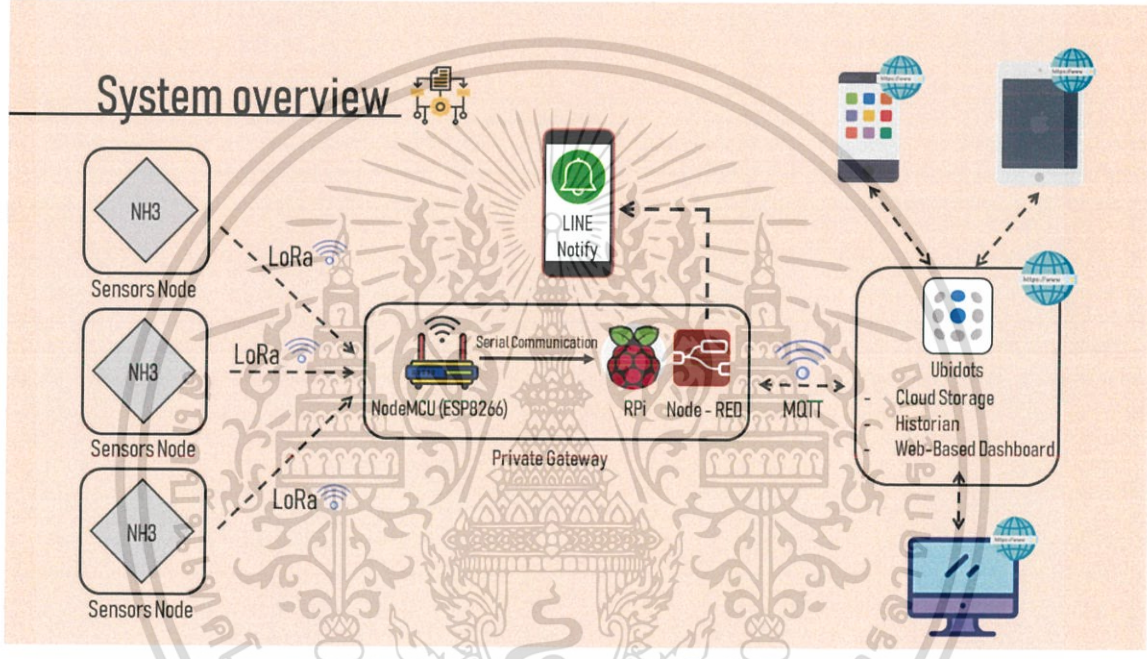
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีการดำเนินงาน

3.1 กล่าวนำ

ในบทนี้จะอธิบายถึงโครงสร้างของระบบ การสร้างเซนเซอร์โนด การสร้างกล่องรับสัญญาณ การเขียนชุดคำสั่ง ลง Node-Red และการออกแบบและสร้างแดชบอร์ด

3.2 โครงสร้างของระบบ



รูปที่ 3.1 โครงสร้างของระบบ

โครงสร้างของระบบจะประกอบด้วย 3 ส่วน ได้แก่

1) เซนเซอร์โนด : ตรวจวัดค่าความเข้มข้นแก๊สแอมโมเนียและส่งข้อมูลไปยังกล่องรับสัญญาณด้วยเทคโนยี LoRa
2) กล่องรับสัญญาณ : รับข้อมูลจากเซนเซอร์โนดแต่ละโนดและส่งข้อมูลไปยังส่วนแสดงผลด้วยโปรโตคอล MQTT
3) ส่วนแสดงผล : เว็บเพจของระบบคลาวด์ Ubidots

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

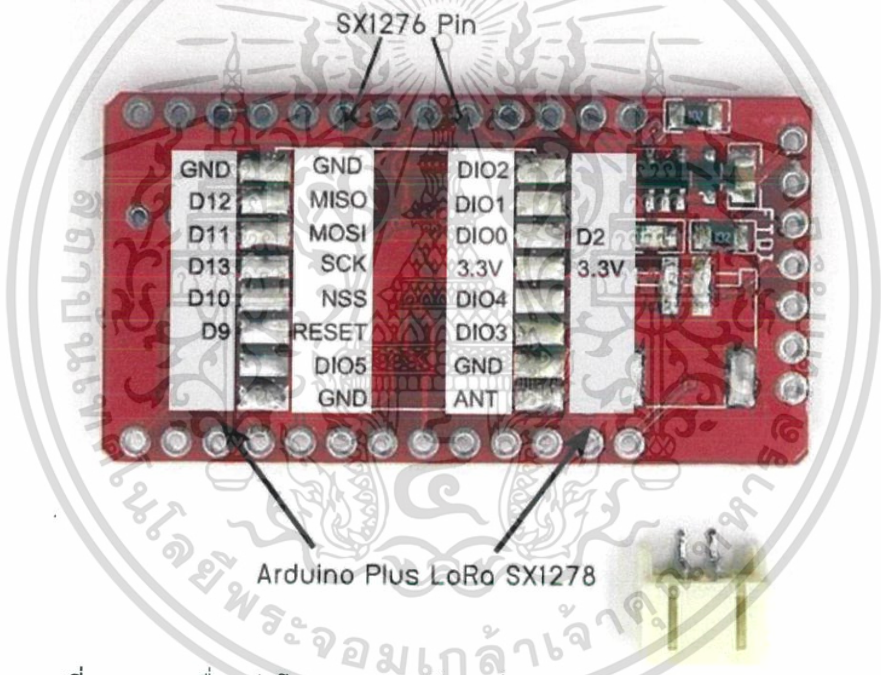
3.3 การสร้างเซนเซอร์โนด

3.3.1 อุปกรณ์สำหรับการจัดทำเซนเซอร์โนด

- โมดูล LoRa (SX1276)
- Arduino Plus LoRa SX1278 (ATmega328p)
- เซนเซอร์วัดค่าความเข้มข้นแอมโมเนีย (Model MQ137)
- เสาอากาศ LoRa 824MHz-2170MHz สำหรับขยายสัญญาณ

3.3.2 การเชื่อมต่อสายของเซนเซอร์โนด

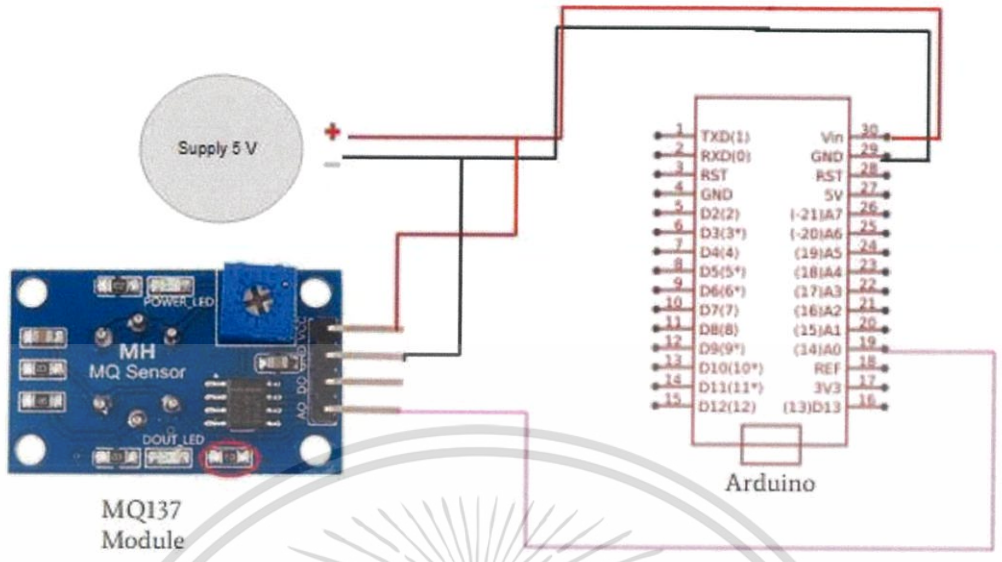
ส่วนนี้นำเสนอถึงการเชื่อมต่อสายต่างๆที่ต้องใช้ระหว่าง โมดูล LoRa บอร์ด Arduino Plus LoRa SX1278 (ATmega328p) และเซนเซอร์วัดค่าความเข้มข้นแอมโมเนีย (Model MQ137)



รูปที่ 3.2 การเชื่อมต่อโมดูล LoRa กับบอร์ด Arduino Plus LoRa SX1278

รูปที่ 3.2 แสดงการเชื่อมต่อสายระหว่างโมดูล LoRa กับบอร์ด Arduino Nano Plus LoRa SX1278 (ATmega328p) เมื่อทำการเชื่อมต่อสายครบ ส่วนต่อมาเริ่มการเชื่อมต่อสายระหว่างบอร์ด Arduino Plus LoRa SX1278 (ATmega328p) กับเซนเซอร์วัดค่าความเข้มข้นแอมโมเนียดังรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 การเชื่อมต่อเซนเซอร์วัดความเข้มข้นแก๊สแอมโมเนียกับบอร์ด Arduino

รูปที่ 3.4 เสาอากาศ LoRa 824MHz-2170MHz สำหรับขยายสัญญาณ

หลังจากทำการเชื่อมต่อสายระหว่างโมดูล LoRa กับบอร์ด Arduino Nano Plus LoRa SX1278 (ATmega328p) พร้อมทั้งเชื่อมต่อสายกับเซนเซอร์วัดความเข้มข้นแอมโมเนียเรียบร้อยจะออกมาดังรูปที่ 3.5 และ 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.5 เซนเซอร์โนดวัดค่าความเข้มข้นแอมโมเนีย

รูปที่ 3.6 เซนเซอร์โนดวัดค่าความเข้มข้นแอมโมเนียพร้อมใช้งาน

จากรูปที่ 3.5 และ 3.6 นั้นเป็นการประกอบ Hardware กับการเชื่อมต่อสายให้ครบตามที่ต้องการใช้และทำการเก็บรายละเอียดต่างๆ ลงภายในกล่องเพื่อความสวยงามและความปลอดภัยอีกขั้นหนึ่งของอุปกรณ์รวมทั้งมีการใช้เสาสัญญาณเพื่อการส่งข้อมูลได้เสถียรและมีประสิทธิภาพมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

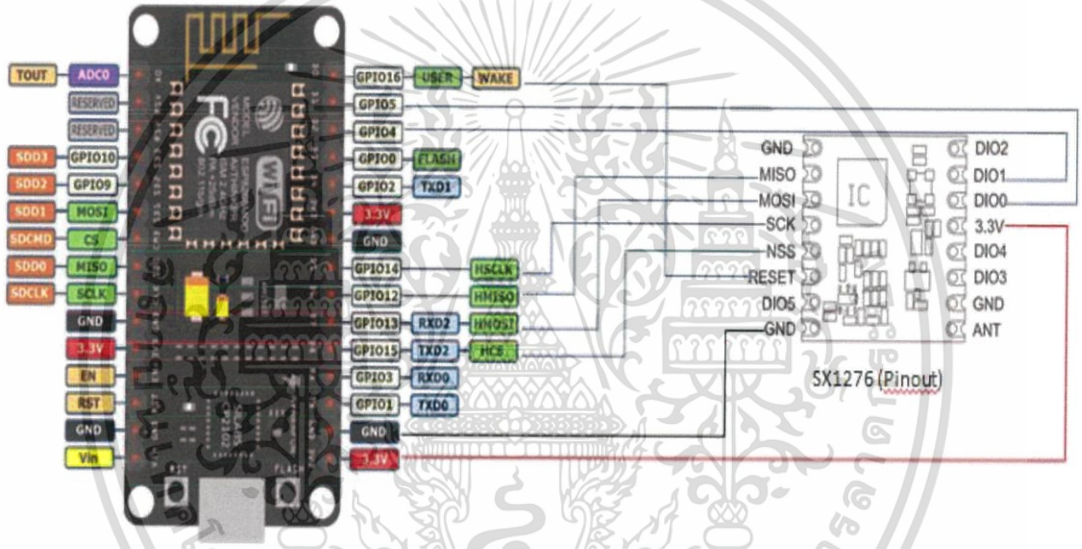
3.4 การสร้างกล่องรับสัญญาณ

3.4.1 เตรียมอุปกรณ์สำหรับการสร้าง

- โมดูล LoRa (SX1276)
- NodeMCU (ESP 8266)
- Raspberry Pi 3 Model B+

3.4.2 การเชื่อมต่อสายของกล่องรับสัญญาณ

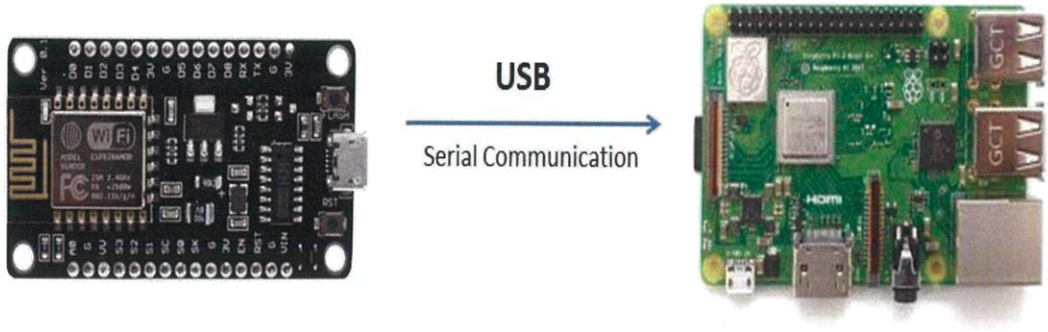
ส่วนนี้นำเสนอถึงการเชื่อมต่อสายต่างๆที่ต้องใช้ภายในกล่องรับสัญญาณระหว่าง โมดูล LoRa NodeMCU และ Raspberry Pi 3 Model B+



รูปที่ 3.7 การเชื่อมต่อสายระหว่าง NodeMCU กับโมดูล LoRa

จากรูปที่ 3.7 แสดงการเชื่อมต่อสายระหว่าง NodeMCU กับโมดูล LoRa เมื่อทำการเชื่อมต่อสายครบแล้ว จะทำการใช้สาย USB สำหรับการส่งข้อมูลแบบ Serial Communication ระหว่าง NodeMCU กับ Raspberry Pi Model B+ ดังรูปที่ 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 การเชื่อมต่อระหว่าง NodeMCU กับ Raspberry Pi 3 Model B+

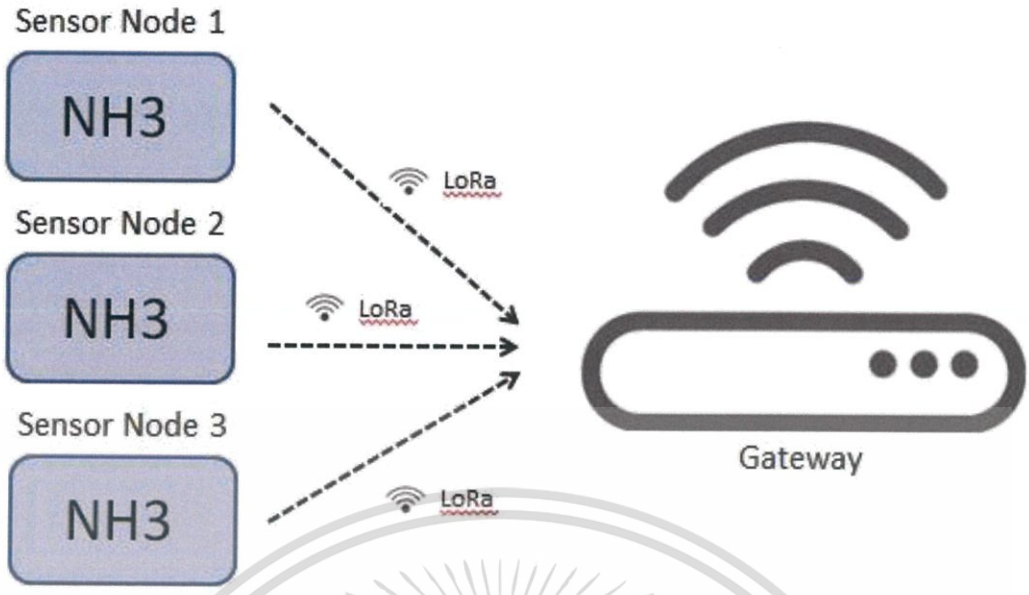
รูปที่ 3.9 กล่องรับสัญญาณ

3.5 การเขียนชุดคำสั่ง

3.5.1 เขียนชุดคำสั่งที่ฝั่งเซนเซอร์โนด

หลังจากการติดตั้งโปรแกรม Arduino IDE แล้วจึงเริ่มทำการเขียนโปรแกรมหรือชุดคำสั่งเพื่อให้เซนเซอร์โนดทำการส่งข้อมูลไปกล่องสัญญาณได้ โดยการส่งข้อมูลจะเป็นดังรูปที่ 3.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 ภาพรวมของเซ็นเซอร์โนดกับกล่องรับสัญญาณ

จากรูปที่ 3.10 เริ่มทำการเขียนโปรแกรมหรือชุดคำสั่งที่ฝั่งเซนเซอร์โนด โดยใช้โปรแกรม Arduino IDE ในการเขียนโปรแกรมซึ่งในส่วนนี้ชุดคำสั่งจะถูกประมวลผลโดยบอร์ด Arduino Plus LoRa SX1278 (ATmega328p)

ขั้นตอนการเขียนโปรแกรมภายในบอร์ด Arduino Plus LoRa SX1278 (ATmega328p)

1. เริ่มจากการเขียนโปรแกรมภาคส่ง โดยเริ่มจากการเซ็ต Parameter เพื่อให้โมดูล LoRa สามารถสื่อสารไปยังโมดูล LoRa อีกโมดูลซึ่งอยู่ที่กล่องรับสัญญาณ โดยจะต้องเขียนโปรแกรมหรือชุดคำสั่งเซ็ต Parameter ทั้งหมด 3 ค่า คือ

    1) Spreading Factor
    2) Signal bandwidth
    3) Frequency

โดยผู้จัดทำ Set ค่า Spreading Factor = 10 Signal Bandwidth = 31.25E3 และ Frequency = 915E6 Hz

```
void(LoRa_Parameters){
      LoRa.begin(915E6) LoRa.setSpreadingFactor(10);
      LoRa.setSignalBandwidth(31.25E3);
      LoRa.setTxPower(17);
}
```

รูปที่ 3.11 ชุดคำสั่งตั้งค่า Parameter LoRa

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เริ่มการเขียนโปรแกรมหรือชุดคำสั่งสำหรับการอ่านข้อมูลจากเซนเซอร์วัดความเข้มข้นแอมโมเนีย

```
void ReadSensor() {
      reading = analogRead(sensor_pin); //Read 0-1023 from sensor
      value = reading * 5.00 / 1023.00 ; //Convert to 0-5
      Rs = ((5.000/RL)-1.000)*value;
      Ratio = Rs/Ro;
      ppm = pow((1.7094*Ratio),(-1.000/0.257)); //Calculate PPM from equation
}
```

รูปที่ 3.12 ชุดคำสั่งอ่านข้อมูลจากเซนเซอร์วัดความเข้มข้นแอมโมเนีย

เมื่อเขียนโปรแกรมหรือชุดคำสั่งอ่านข้อมูลเสร็จสมบูรณ์แล้วจึงทำการดูผลผ่าน Serial Monitor เพื่อเป็นการเช็คความถูกต้องก่อนส่งข้อมูล

```
{"node" : "2","value":1.380544}
{"node" : "2","value":15.17375}
{"node" : "2","value":1.604784}
{"node" : "2","value":1.380544}
{"node" : "2","value":1.380544}
{"node" : "2","value":1.380544}
{"node" : "2","value":1.380544}
{"node" : "2","value":1.380544}
{"node" : "2","value":1.380544}
```

รูปที่ 3.13 แสดงข้อมูลผ่าน Serial Monitor

3. ขั้นตอนนี้เป็นขั้นตอนการเขียนโปรแกรมเพื่อจัดรูปแบบก่อนที่จะส่งข้อมูลออกจากฝั่งของเซนเซอร์โนด โดยการจัดข้อมูลนี้จะป็นในรูปแบบของ JSON Format

```
void JSON_Generate() {
      StaticJsonBuffer<250> jsonBuffer; //DynamicJsonBuffer jsonBuffer(100);
      JsonObject& root = jsonBuffer.createObject();
      root["node"] = "1";
      root["value"] = ppm;
      root["volt"] = value;
      root.printTo(Buffer);
      memset(sendvalue,0,sizeof(sendvalue));
      Buffer.toCharArray(sendvalue, sizeof(sendvalue));
      Serial.println(Buffer);
}
```

รูปที่ 3.14 ชุดคำสั่งเพื่อจัดการข้อมูลก่อนที่ทำการส่งข้อมูลเซนเซอร์โนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{"node" : "2","value":1.380544}
{"node" : "2","value":15.17375}
{"node" : "2","value":1.604784}
{"node" : "2","value":1.380544}
{"node" : "2","value":1.380544}
```

รูปที่ 3.15 การจัดข้อมูลในรูปแบบ JSON Format

4. เมื่อเขียนโปรแกรมจัดการข้อมูลในรูปแบบของ JSON Format เสร็จสมบูรณ์แล้ว เริ่มทำการเขียนโปรแกรมในส่วนของการจัดส่งข้อมูลเพื่อนำข้อมูลไปใช้ต่อในส่วนถัดไป

```
void JSON_Generate() {
      StaticJsonBuffer<250> jsonBuffer;
      JsonObject& root = jsonBuffer.createObject();
      root["node"] = nodeBuffer;
      root["value"] = valueBuffer;
      root["volt"] = voltBuffer;
      root["rssi"] = LoRa.packetRssi();
      root["digi"] = digiBuffer;
      root["active"] = "1";
      root.printTo(tonodered);
      Serial.print(tonodered);
      Serial.println(); tonodered = "";
}
```

รูปที่ 3.16 ชุดคำสั่งสำหรับการส่งข้อมูลของเซนเซอร์โนด

3.5.2 เขียนชุดคำสั่งที่ฝั่งกล่องรับสัญญาณ

ขั้นตอนการเขียนชุดคำสั่งลงใน NodeMCU

1. ในขั้นตอนแรกจะทำการเขียนโปรแกรมหรือชุดคำสั่งในส่วนของการเซ็ต Parameter ทั้ง 3 ค่าเหมือนกับฝั่งของเซนเซอร์โนด

- Spearding Factor = 10
- Signal Bandwidth = 31.25E3
- Frequency = 915E6

(ทั้ง 3 ค่าจะต้องเซ็ตให้ตรงกับฝั่งของเซนเซอร์โนด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void(LoRa_Parameters){
      LoRa.begin(915E6) LoRa.setSpreadingFactor(10);
      LoRa.setSignalBandwidth(31.25E3);
      LoRa.setTxPower(17);
}
```

รูปที่ 3.17 ชุดคำสั่งตั้งค่า Parameter ฝั่งกล่องรับสัญญาณ

2. เมื่อทำการเขียนโปรแกรมหรือชุดคำสั่งตั้งค่า Parameter เสร็จสมบูรณ์แล้วในส่วนถัดมาจะเป็นการเขียนโปรแกรมการรับข้อมูลจากเซนเซอร์โนดแต่ละโนด และเมื่อได้รับค่าจากเซนเซอร์โนดแล้วจะเขียนชุดคำสั่งสำหรับการส่งข้อมูลแบบ Serial Communication ไปยัง Node - Red ซึ่งอยู่ภายใน Raspberry Pi

```
void JSON_Generate() {
      StaticJsonBuffer<250> jsonBuffer;
      JsonObject& root = jsonBuffer.createObject();
      root["node"] = nodeBuffer;
      root["value"] = valueBuffer;
      root["volt"] = voltBuffer;
      root["rssi"] = LoRa.packetRssi();
      root["digi"] = digiBuffer;
      root["active"] = "1";
      root.printTo(tonodered);
      Serial.print(tonodered);
      Serial.println(); tonodered = "";
}
```

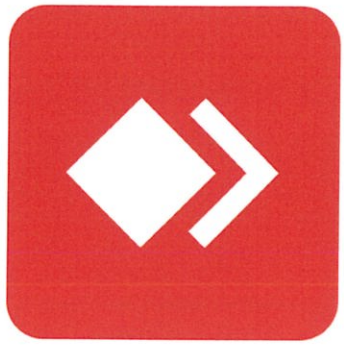
รูปที่ 3.18 ชุดคำสั่งรับและส่งข้อมูลแบบ Serial Communication ของกล่องรับสัญญาณ

3.6 การลง Node - Red ฝั่งใน Raspberry Pi

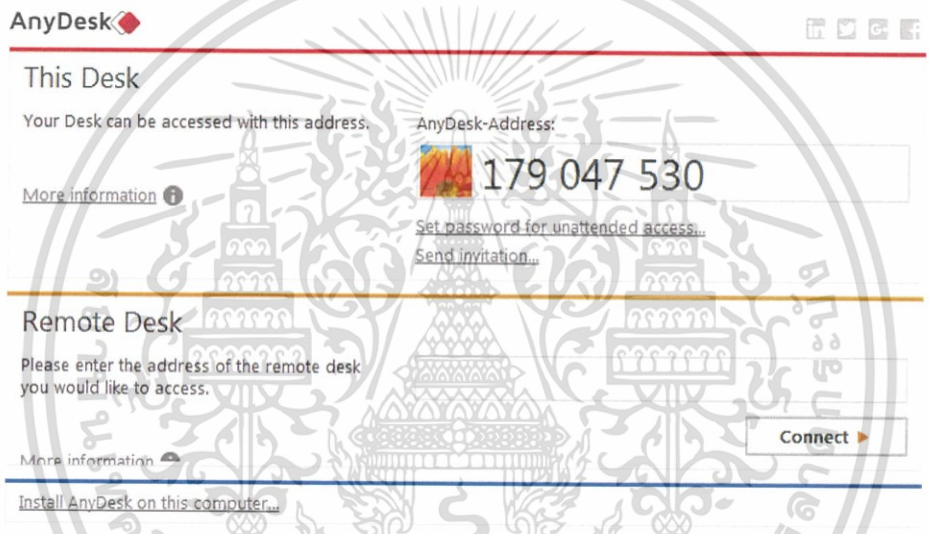
3.6.1 ขั้นตอนการลง Node - Red ฝั่งใน Raspberry Pi

1. ทำการ Download AnyDesk และติดตั้งภายใน Raspberry Pi ใช้สำหรับการ Remote Desktop หรือทำการต่อ HDMI กับหน้าจอแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 โปรแกรม AnyDesk

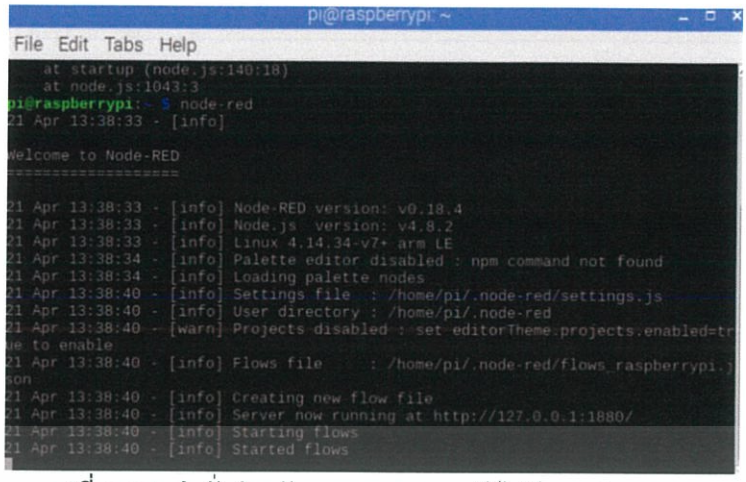


รูปที่ 3.20 เมื่อเปิดโปรแกรม AnyDesk

2.ทำการติดตั้ง Node - Red ฝังใน Raspberry Pi

ขั้นตอนนี้จะต้องทำการเช็คว่า Raspberry Pi มี Node - Red ด้วยคำสั่งดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.21 คำสั่งสำหรับการตรวจสอบ มี-ไม่มี Node - RED

ใช้ Code นี้ใน Terminal ภายใน Raspberry Pi และเมื่อ Run code เสร็จสมบูรณ์แล้วให้นำ code

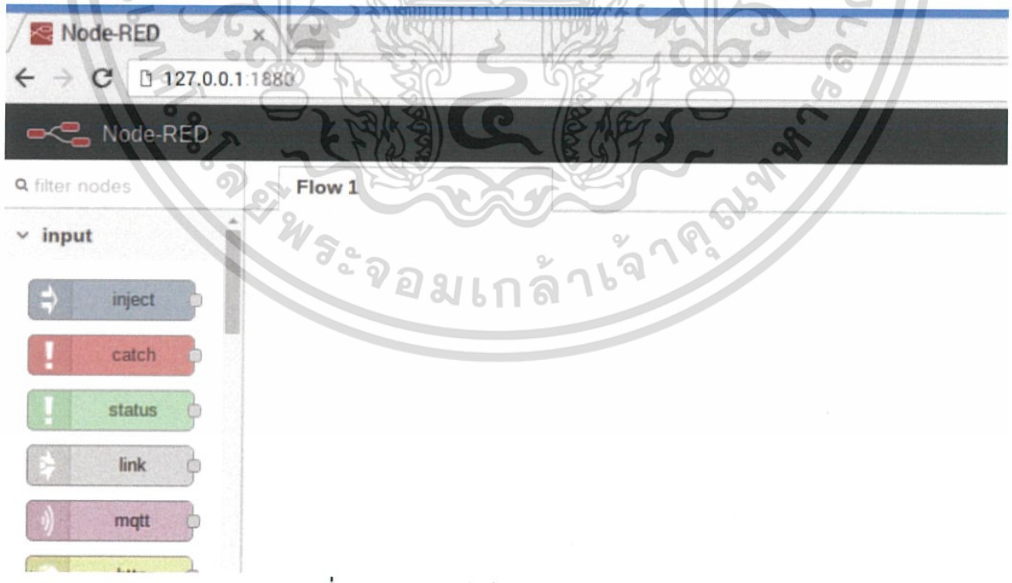
- sudo apt-get install build-essential

Run ใน Terminal ของ Raspberry Pi เมื่อเสร็จสิ้นแล้วให้ใช้คำสั่ง

- Node - Red-start

เมื่อเสร็จสิ้นคำสั่งถ้าต้องการ Auto boot Node - RED เมื่อทำการเปิดใช้งาน Raspberry Pi ให้ใช้คำสั่ง

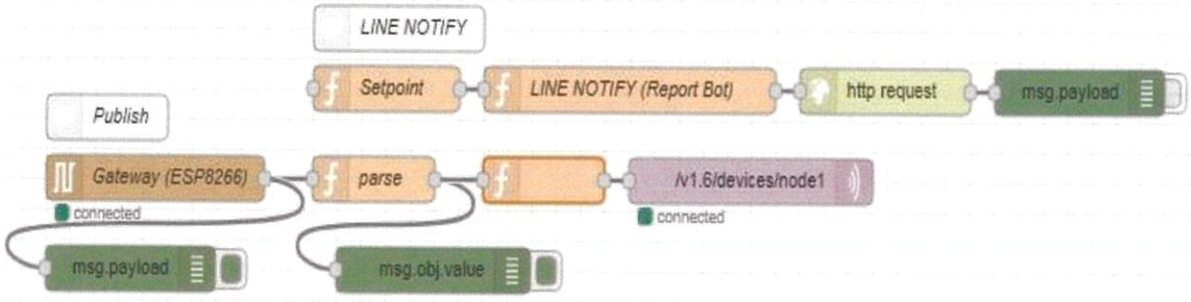
- sudo systemctl enable nodered.service
- เมื่อทำครบทุกขั้นตอนแล้วให้เปิด Node - RED ผ่าน Brower ด้วย localhost:1880



รูปที่ 3.22 การเข้าใช้งาน Node – RED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

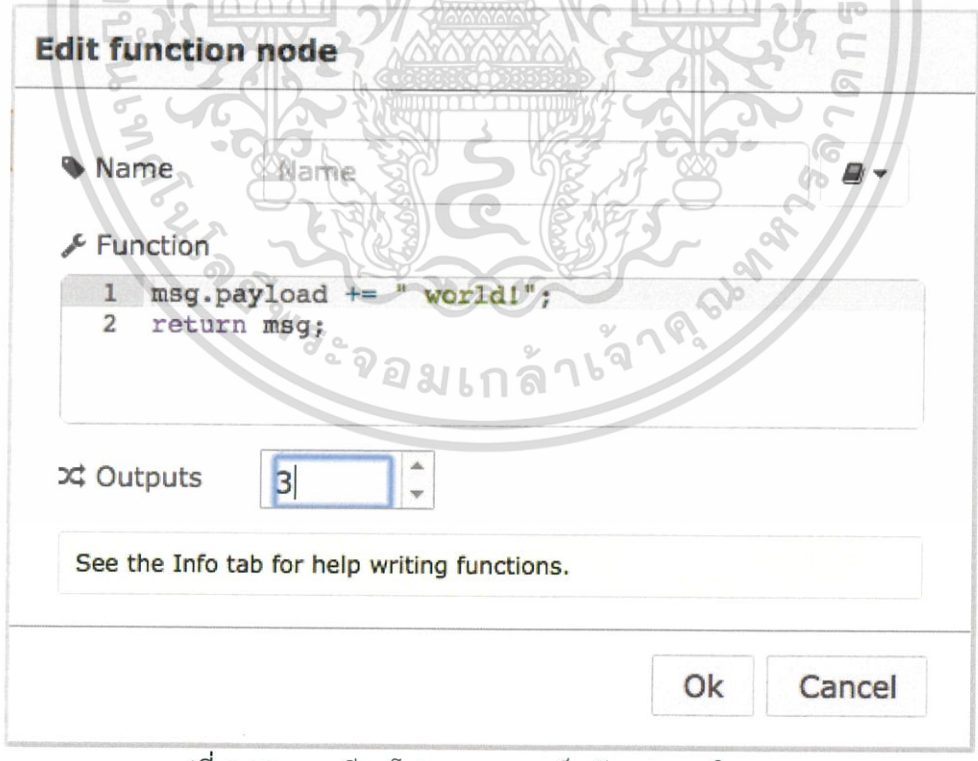
3.6.2 การสร้าง Flow ภายใน Node-Red



รูปที่ 3.23 ภาพรวมของ Flow ภายใน Node – RED

ภาพที่ 3.24 Flow จัดการรับและส่งข้อมูล

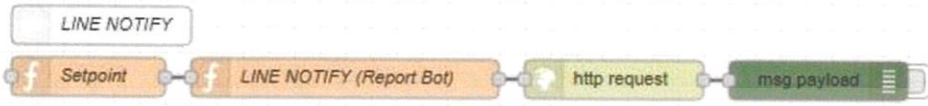
ในแต่ละ Flow นั้นจะต้องมีการตั้งค่า Parameter และการเขียนโปรแกรมของแต่ละ Flow ในรูปภาพ 3.25 นี้จะเป็นการตั้ง Parameter ของ Node Function



รูปที่ 3.25 การเขียนโปรแกรมและเซ็ตตัวแปรภายใน Flow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเนื่องด้วยผู้จัดทำต้องการทราบข้อมูลผ่านทาง Line Notify จึงทำการเขียน Flow ในการส่งข้อมูลทางไลน์เพื่อแจ้งข้อมูลอย่าง Real Time ให้ผู้ใช้สามารถได้ทราบข้อมูลอย่างถูกต้องและยังมีการเขียนโปรแกรมสำหรับการกำหนดค่าแจ้งเตือน สำหรับการแจ้งเตือนความผิดปกติของข้อมูลหรือข้อมูลนั้นมีค่าเกินค่าแจ้งเตือนที่กำหนด



รูปที่ 3.26 การเขียน Flow แจ้งเตือนผ่าน Line Notify

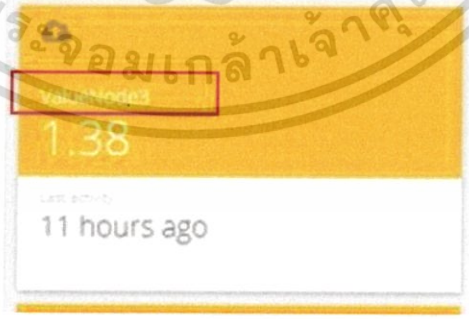
3.7 การออกแบบและสร้างแดชบอร์ด

ในส่วนนี้ผู้จัดทำเลือกใช้แดชบอร์ดของ Ubidots โดยขั้นตอนการออกแบบนี้จะมีการแสดงค่าความเข้มข้นแอมโมเนีย แรงดันไฟฟ้า และสถานะการเปิด-ปิดใช้งานของแต่ละโนด

3.7.1 ออกแบบและสร้างแดชบอร์ดแสดงข้อมูลภาพรวม

ในขั้นตอนนี้จะมีการสร้าง Variable ภายใน Ubidots เพื่อเก็บข้อมูลและผลทั้งยังสามารถเก็บข้อมูลเป็นรูปแบบของ Database ซึ่งในส่วนของการเก็บข้อมูลหรือส่วนของ Database นั้นทาง Ubidots จะมีการเก็บข้อมูลให้อย่างสำเร็จรูปโดยการเก็บข้อมูลได้มากหรือน้อยนั้นขึ้นอยู่กับการสมัครใช้ User กับทาง Ubidots

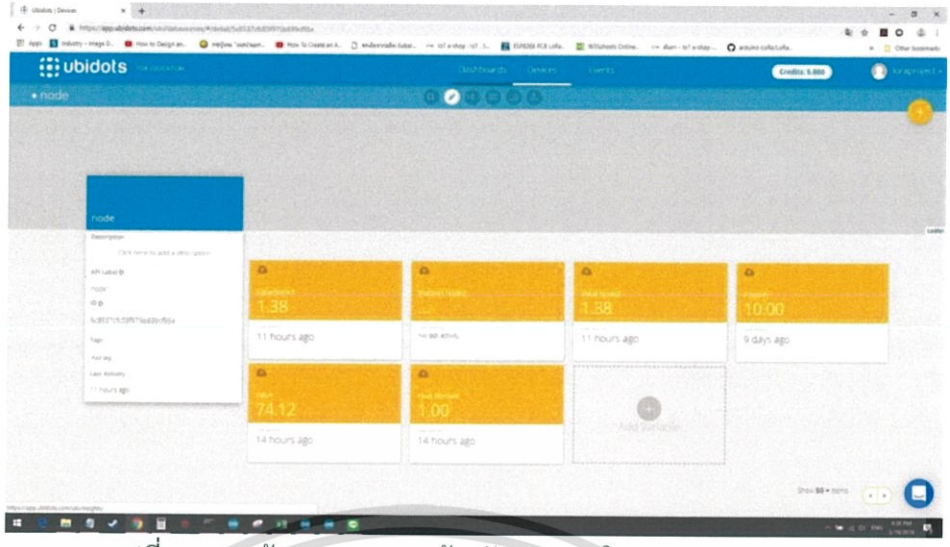
เริ่มจากการสร้างตัวแปรเพื่อเก็บข้อมูลที่ส่งมาจาก Node - Red ซึ่งชื่อตัวแปรที่สร้างนั้นมีความสำคัญมากเนื่องจากตัวแปรนี้จะต้องถูกเซ็ตให้เป็นชื่อตัวแปรเดียวกันกับ Node - Red



รูปที่ 3.27 การสร้าง Variable ภายใน Ubidots

จากรูปที่ 3.27 นั้นเมื่อเริ่มสร้างตัวแปรแต่ละตัวทางเว็บเพจของระบบคลาวด์ Ubidots จะแสดงให้เห็นว่าได้สร้างตัวแปรเสร็จสมบูรณ์แล้วดังรูปที่ 3.28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.28 หน้ารวมของการสร้างตัวแปรภายใน Ubidots

หลังจากสร้างตัวแปรที่ตั้งค่าครบตามต้องการแล้วเริ่มการออกแบบและสร้างหน้าแดชบอร์ด หรือหน้าจอแสดงผลด้วยเครื่องมือภายใน Ubidots

รูปที่ 3.29 Ubidots Widgets

จากรูปที่ 3.29 นั้นแสดงถึงเครื่องมือที่ทาง Ubidots นั้นมีให้ใช้ เช่น Chart Table Indicator Control เป็นต้น โดยคณะผู้จัดทำได้ออกแบบหน้าแดชบอร์ดแสดงข้อมูลภาพรวม มีทั้งหมด 5 ส่วนดังนี้

1. เกจแสดงผลค่าความเข้มข้นของแก๊สแอมโมเนียแต่ละโนด

รูปที่ 3.30 เครื่องมือ Gauge Indicator

2. สถานะความอันตรายระดับความเข้มข้นแก๊สแอมโมเนียแต่ละโนด : แสดงตามเงื่อนไขดังตารางที่ 3.1

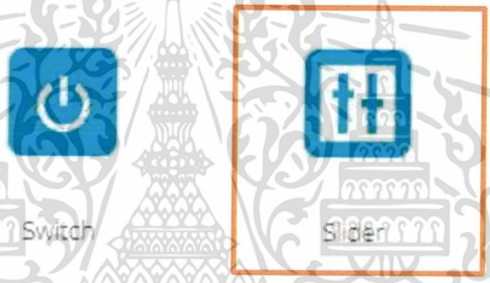
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.31 เครื่องมือ On/Off Indicator

3. Digital input : เครื่องมือ On/Off Indicator

4. สถานะการเปิด-ปิดใช้งานของแต่ละโนด : เครื่องมือ On/Off Indicator

5. Slider กำหนดค่าแจ้งเตือนการแจ้งเตือนค่าความเข้มข้นแก๊สแอมโมเนียผ่านทางไลน์ Notify



รูปที่ 3.32 เครื่องมือ Slider Control

ตารางที่ 3.1 ตารางเงื่อนไขการสั่ง Output status LED

เงื่อนไข	คำอธิบาย	Output LED	หมายเหตุ
1	เมื่อค่า ppm น้อยกว่า 25	สีเขียวติด	ค่าแอมโมเนียปกติ
2	เมื่อค่า ppm มากกว่า 25 และน้อยกว่า 100	สีเหลืองติด	ควรระวัง/เริ่มอันตราย
3	เมื่อค่า ppm มากกว่า 100	สีแดงติด	อันตราย

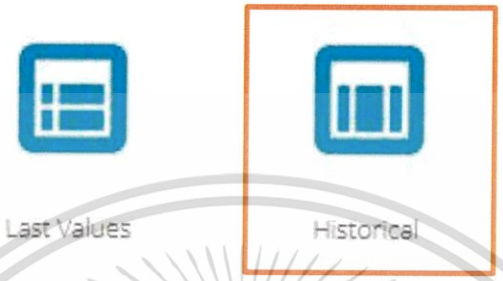
3.7.2 ออกแบบและสร้างแดชบอร์ดของแต่ละโนด

ในส่วนนี้นั้นจะมีขั้นตอนการทำแบบเดียวกันกับการออกแบบและสร้างแดชบอร์ดแสดงข้อมูลภาพรวม โดยแดชบอร์ดของแต่ละโนดนั้นจะถูกออกแบบให้เหมือนกันกับทุกโนดซึ่งจะประกอบด้วยการแสดงผลทั้งหมด 7 ส่วนประกอบด้วย

1. สถานะแจ้งเตือนระดับความอันตรายของแอโมเนียตามตารางที่ 3.1 : เครื่องมือ Gauge Indicator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เกจแสดงผลค่าความเข้มข้นแอมโมเนีย : เครื่องมือ Gauge Indicator
3. เกจแรงดันไฟฟ้าจากเซ็นเซอร์ : เครื่องมือ Gauge Indicator
4. RSSI : เครื่องมือ Gauge Indicator
5. Led Status Digital input (มีเฉพาะโนด Simulation) : เครื่องมือ On/Off Indicator
6. Data Historian ของค่าความเข้มข้นแอมโมเนีย

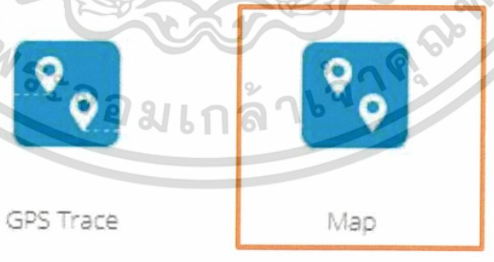


รูปที่ 3.33 เครื่องมือ Historical Table

7. Historian Chart ของค่าความเข้มข้นแอมโมเนีย

รูปที่ 3.34 เครื่องมือ Line Chart (Trend)

8. Location (ทำการ Fix ผ่าน Node-Red)



รูปที่ 3.35 เครื่องมือ Map

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

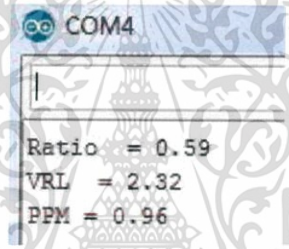
บทที่ 4
ผลการดำเนินงาน

4.1 กล่าวนำ

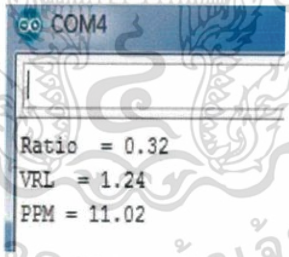
ในบทที่ 4 นี้จะอธิบายถึงผลการทดสอบเซนเซอร์โนด ผลการทดสอบการสื่อสารของโมดูล LoRa ผลการทดสอบข้อมูลระหว่างเซนเซอร์โนดทั้ง 3 ชุดกับกล่องรับสัญญาณ ผลการแสดงข้อมูลผ่านเว็บเพจของ Ubidots และผลการแจ้งเตือนค่าความเข้มข้นแก๊สแอมโมเนียผ่านแอปพลิเคชันไลน์

4.2 ผลการทดสอบเซ็นเซอร์โนด

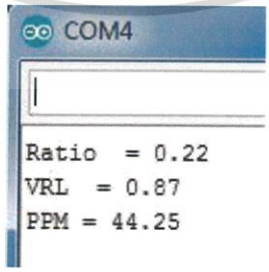
จากการทดลองการวัดค่าความเข้มข้นแก๊สแอมโมเนียจากเซนเซอร์เทียบกับรูปที่ 2.23 กราฟ Sensitivity ของเซนเซอร์ตรวจวัดแก๊สแอมโมเนียทั้งหมด 5 จุดทดสอบตามจุด Rs/Ro ต่าง ๆ ตามจริงจากกราฟซึ่งถูกระบุดังตารางที่ 4.1



รูปที่ 4.1 ผลการทดลอง Rs/Ro ที่ 0.59

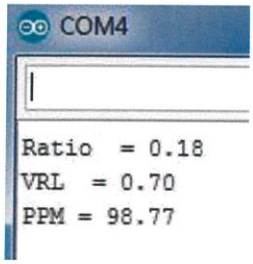


รูปที่ 4.2 ผลการทดลอง Rs/Ro ที่ 0.32

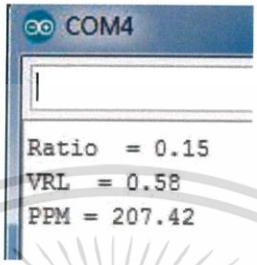


รูปที่ 4.3 ผลการทดลอง Rs/Ro ที่ 0.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 ผลการทดลอง Rs/Ro ที่ 0.18



รูปที่ 4.5 ผลการทดลอง Rs/Ro ที่ 0.15

ตารางที่ 4.1 ผลการทดลองปรับระดับแรงดันจาก Potentiometer

จุดที่	ค่า Plot จากกราฟ	ค่าที่ได้จากการคำนวณ (PPM)	Error (%)
1	Rs/Ro = 0.59, PPM = 1	0.96	4
2	Rs/Ro = 0.32, PPM = 10	11.02	10.2
3	Rs/Ro = 0.22, PPM = 50	44.25	11.5
4	Rs/Ro = 0.175, PPM = 100	98.77	1.23
5	Rs/Ro = 0.154, PPM = 200	207.42	3.71

จากตารางที่ 4.1 นั้นเป็นการนำ Potentiometer มาทำการปรับแรงดันแทนเซนเซอร์วัดค่าแก๊สแอมโมเนียจริงเนื่องจากผู้จัดทำไม่สามารถหาแก๊สแอมโมเนียบริสุทธิ์สำหรับการทดลองนี้ได้ ซึ่งจากผลการทดลองตามตารางที่ 4.1 นั้นทำให้มีค่า Error ต่ำสุดที่ 1.23 % และสูงสุด ณ ตามจุดทดลองคือ 11.5 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการทดสอบการสื่อสารของโมดูล LoRa

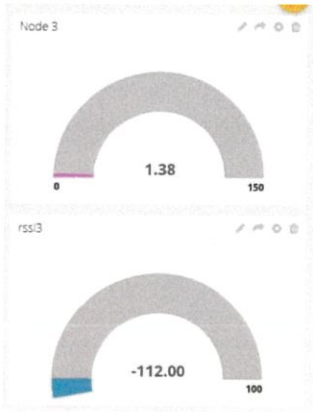
จากการสร้างเซนเซอร์โนดและกล่องรับสัญญาณนั้นคณะผู้จัดทำได้ทำการทดลองส่งข้อมูลเพื่อทดสอบการส่งข้อมูลของเซนเซอร์โนดกับกล่องรับสัญญาณเพื่อแสดงให้เห็นว่าทั้งเซนเซอร์โนดและกล่องรับสัญญาณนั้นสามารถสื่อสารกันได้จริงโดยการทดสอบที่ระยะต่างๆเพื่อดูความเข้มของสัญญาณหรือ RSSI โดยมีการตั้งกล่องรับสัญญาณไว้ ณ ชั้นดาดฟ้าตึก 12 ชั้นสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

รูปที่ 4.6 จุดตั้งกล่องรับสัญญาณ ณ ชั้นดาดฟ้าตึก 12 ชั้นสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

จากรูปที่ 4.6 ทำการติดตั้งกล่องรับสัญญาณสำหรับการทดสอบการสื่อสารของโมดูล LoRa ระหว่างเซนเซอร์โนดกับกล่องรับสัญญาณที่ระยะ 100 เมตร 300 เมตร 600 เมตร 1000 เมตรและที่ 8400 เมตร มีความเข้มสัญญาณดังนี้

รูปที่ 4.7 RSSI ที่ระยะ 100 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 RSSI ที่ระยะ 300 เมตร

รูปที่ 4.9 RSSI ที่ระยะ 600 เมตร

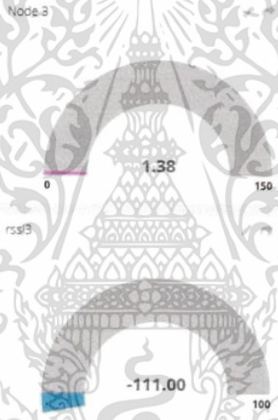
รูปที่ 4.10 RSSI ที่ระยะ 1000 เมตร

รูปที่ 4.11 ภาพรวมการทดสอบระยไม่เกิน 1000 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดสอบระยะความสามารถในการสื่อสารของโมดูล LoRa นั้นที่ 1000 เมตรยังสามารถส่งข้อมูลได้อย่างดีคณะผู้จัดทำจึงได้ทำการทดลองที่ระยะ 8400 เมตรเพื่อทดสอบว่าเซนเซอร์โนดกับกล่องรับสัญญาณยังสามารถสื่อสารกันได้

รูปที่ 4.12 จุดทดสอบที่ 8400 เมตรความสูง 30 เมตรที่ป่าในกรุงเขตประเวศ



รูปที่ 4.13 RSSI ที่ระยะ 8400 เมตร

รูปที่ 4.14 ภาพรวมการทดสอบระยะ 8400 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการทดสอบความสามารถในการสื่อสารที่ระยะ100 เมตร 300 เมตร 600 เมตร 1000 เมตรและที่ 8400 เมตรนั้นเทคโนโลยี LoRa ยังสามารถส่งข้อมูลได้โดยค่าความเข้มของสัญญาณนั้นขึ้นอยู่กับสิ่งกีดขวางซึ่งการส่งข้อมูลระหว่างเซนเซอร์โนดกับกล่องรับสัญญาณนั้นควรอยู่ในแนวเส้นสายตาเพื่อที่ให้สัญญาณของเทคโนยี LoRa นั้นมีความเสถียรมากยิ่งขึ้น

ตารางที่ 4.2 ผลการทดสอบการสื่อสารของโมดูล LoRa

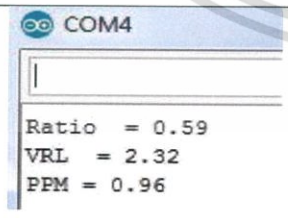
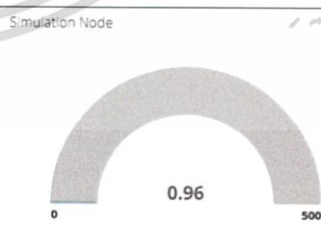
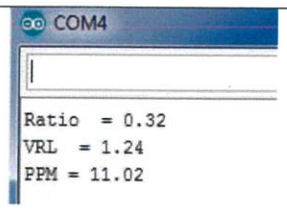
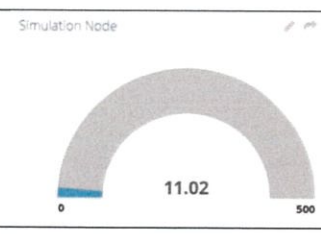
ลำดับ	คำอธิบาย	ผลที่ได้
1	ผลการทดสอบที่ระยะ 100 เมตร	ผ่าน
2	ผลการทดสอบที่ระยะ 300 เมตร	ผ่าน
3	ผลการทดสอบที่ระยะ 600 เมตร	ผ่าน
4	ผลการทดสอบที่ระยะ 1000 เมตร	ผ่าน
5	ผลการทดสอบที่ระยะ 8400 เมตร	ผ่าน

จากตารางที่ 4.2 นั้นแสดงผลการทดสอบการสื่อสารของโมดูล LoRa ที่ระยะต่าง ๆ ดังตารางซึ่งจากผลการทดสอบนั้นเซนเซอร์โนดและกล่องรับสัญญาณยังสามารถสื่อสารกันได้อย่างดีที่ระยะ 8400 เมตร

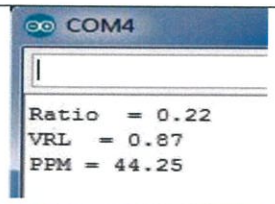
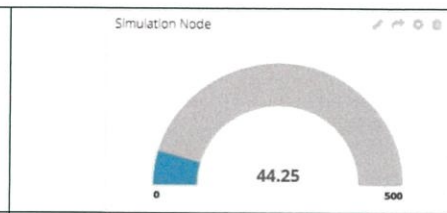
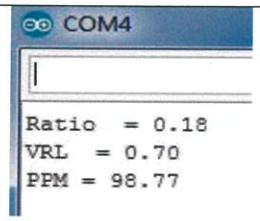
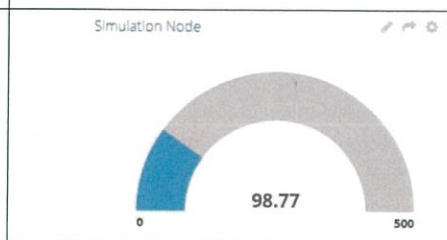
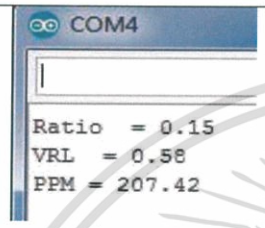
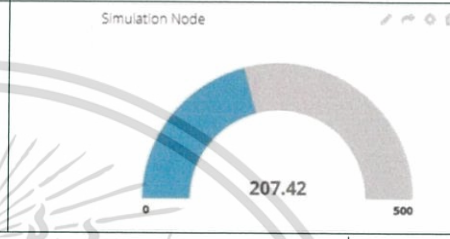
4.4 ผลการทดสอบข้อมูลระหว่างเซนเซอร์โนดและกล่องรับสัญญาณ

ส่วนนี้เป็นการทดสอบความถูกต้องของข้อมูลระหว่างเซนเซอร์โนดทั้ง 3 ชุด ไปยังกล่องรับสัญญาณด้วยค่าความเข้มแก๊สแอมโมเนียที่วัดแต่ละเซนเซอร์โนดเทียบกับค่าความเข้มข้นแก๊สแอมโมเนียบนเว็บเพจของ Ubidots

ตารางที่ 4.3 ผลการทดสอบความถูกต้องของข้อมูลระหว่างเซนเซอร์โนดที่ 1 และกล่องรับสัญญาณ

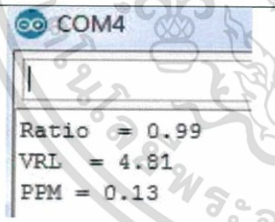
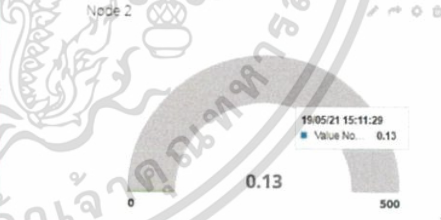
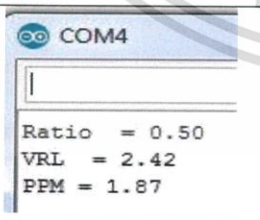
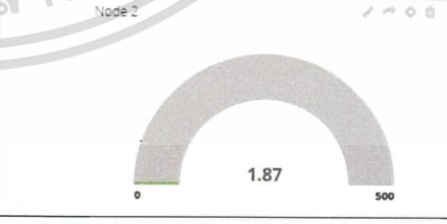
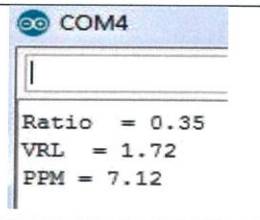
ค่าความเข้มข้นแก๊สแอมโมเนียจากเซนเซอร์โนดที่ 1	ค่าความเข้มข้นแก๊สแอมโมเนีย (ppm) บนเว็บเพจ Ubidots
	
	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการทดลองตารางที่ 4.3 ผลการทดสอบข้อมูลระหว่างเซนเซอร์โนดที่ 1 และกล่องรับสัญญาณสรุปได้ว่าค่าที่วัดได้จากเซนเซอร์และค่าที่แสดงบนเว็บเพจของ Ubidots นั้น สามารถส่งข้อมูลได้ถูกต้อง

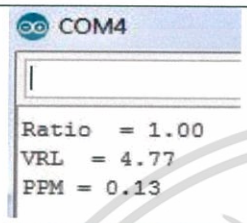
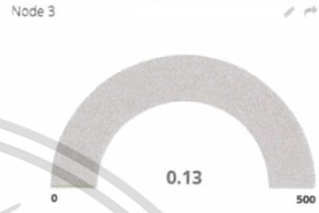
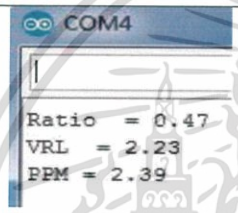
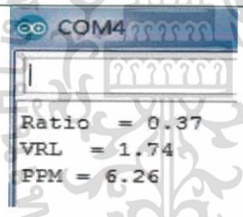
ตารางที่ 4.4 ผลการทดสอบความถูกต้องของข้อมูลระหว่างเซนเซอร์โนดที่ 2 และกล่องรับสัญญาณ

ค่าความเข้มข้นแก๊สแอมโมเนียจากเซนเซอร์โนดที่ 2	ค่าความเข้มข้นแก๊สแอมโมเนีย (ppm) บนเว็บเพจ Ubidots
	
	
	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการทดลองตารางที่ 4.4 ผลการทดสอบข้อมูลระหว่างเซนเซอร์โนดที่ 2 และกล่องรับสัญญาณสรุปได้ว่าค่าที่วัดได้จากเซนเซอร์และค่าที่แสดงบนเว็บเพจของ Ubiidots นั้น สามารถส่งข้อมูลได้ถูกต้อง

ตารางที่ 4.5 ผลการทดสอบความถูกต้องของข้อมูลระหว่างเซนเซอร์โนดที่ 3 และกล่องรับสัญญาณ

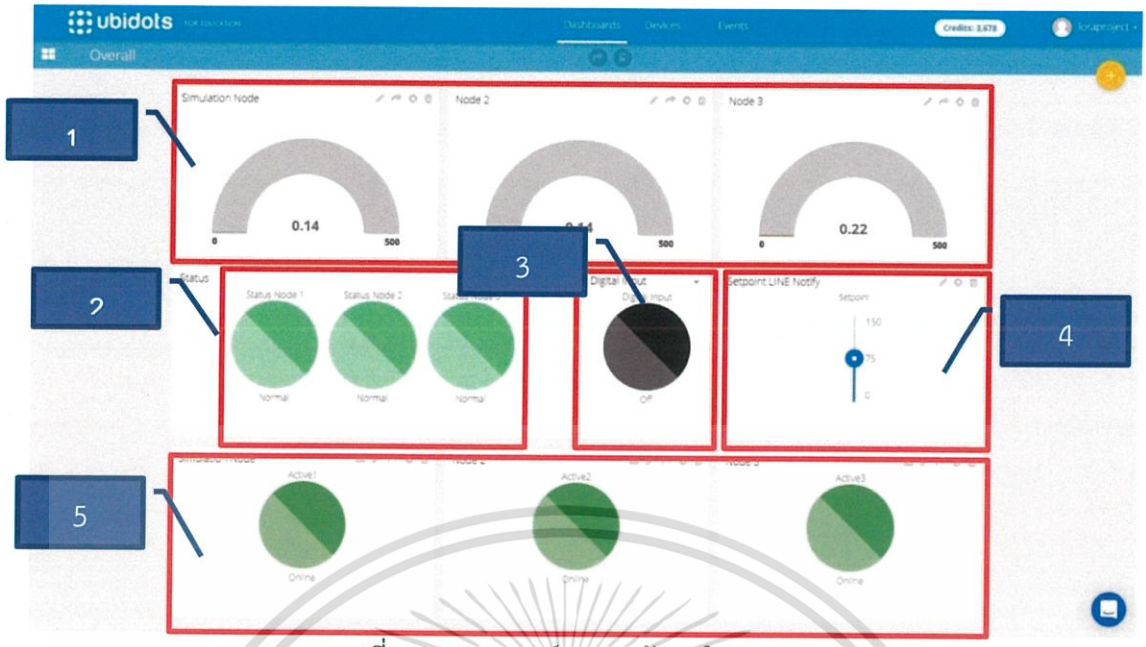
ค่าความเข้มข้นแก๊สแอมโมเนียจากเซนเซอร์โนดที่ 3	ค่าความเข้มข้นแก๊สแอมโมเนีย (ppm) บนเว็บเพจ Ubidots
	
	
	

จากผลการทดลองตารางที่ 4.5 ผลการทดสอบข้อมูลระหว่างเซนเซอร์โนดที่ 3 และกล่องรับสัญญาณสรุปได้ว่าค่าที่วัดได้จากเซนเซอร์และค่าที่แสดงบนเว็บเพจของ Ubidots นั้น สามารถส่งข้อมูลได้ถูกต้อง

4.5 ผลการแสดงผ่านเว็บเพจของ Ubidots

จากการออกแบบและสร้างแดชบอร์ดแสดงข้อมูลภาพรวมและแดชบอร์ดของแต่ละโนดนั้น เมื่อออกแบบและสร้างเสร็จสมบูรณ์แล้วได้ผลดังรูปที่ 4.15 และรูปที่ 4.16 ตามลำดับ

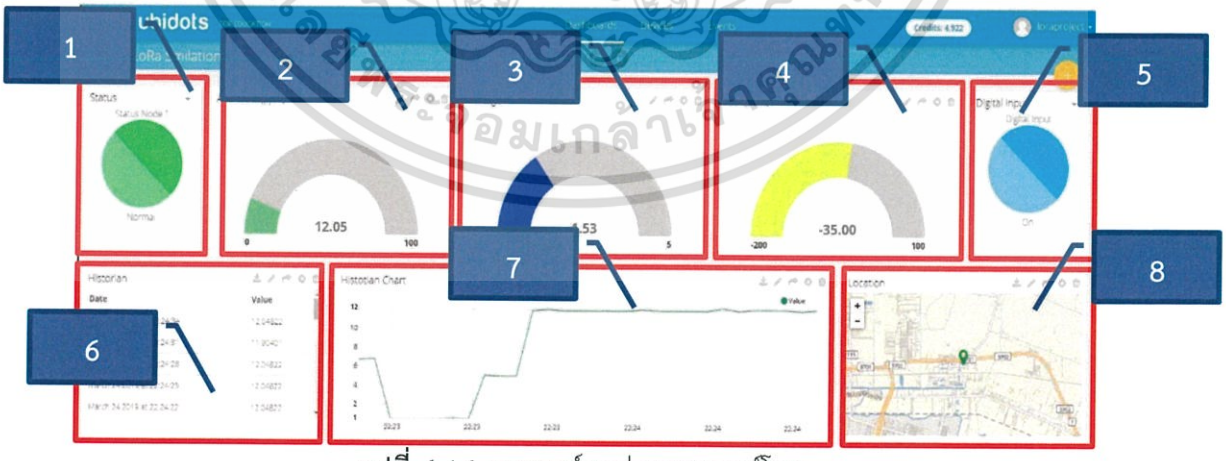
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 แดชบอร์ดแสดงข้อมูลในภาพรวม

ตารางที่ 4.6 ตารางสรุปผลค่าบนเว็บเพจ Ubidots หน้า Overall

ลำดับที่	คำอธิบาย	ผลที่ได้
1	เกจแสดงค่าความเข้มข้นแก๊สแอมโมเนียแต่ละเซนเซอร์โนด	ผ่าน
2	สถานะแสดงค่าความเข้มข้นแก๊สแอมโมเนียตามเงื่อนไข	ผ่าน
3	Digital Input	ผ่าน
4	Slide bar สำหรับกำหนดค่าแจ้งเตือนผ่าน Line Notify	ผ่าน
5	สถานะแสดงการเปิด - ปิดการใช้งานของแต่ละเซนเซอร์โนด	ผ่าน



รูปที่ 4.16 แดชบอร์ดแต่ละเซนเซอร์โนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.7 ตารางสรุปผลค่าบนเว็บเพจ Ubidots หน้าแดชบอร์ดแต่ละเซนเซอร์โนด

ลำดับที่	คำอธิบาย	ผลที่ได้
1	สถานะแสดงระดับความอันตรายของแอโมเนียตามตารางที่ 3.1	ผ่าน
2	เกจแสดงผลค่าความเข้มข้นแก๊สแอมโมเนีย	ผ่าน
3	เกจแรงดันไฟฟ้าที่วัดได้จากเซนเซอร์	ผ่าน
4	RSSI	ผ่าน
5	Led Status Digital input (มีเฉพาะโนด Simulation)	ผ่าน
6	Data Historian ของค่าความเข้มข้นแอมโมเนีย	ผ่าน
7	Historian Chart ของค่าความเข้มข้นแอมโมเนีย	ผ่าน
8	Location	ผ่าน

ส่วนต่อมาเป็นผลการทดลองตามเงื่อนไขของตารางที่ 3.1 เมื่อ PPM <25 (ไฟสีเขียวติด) 25>= PPM <100 (ไฟสีเหลืองติด) PPM > 100 (ไฟสีแดงติด) ดังรูปที่ 4.17 - 4.19

รูปที่ 4.17 ค่าความเข้มข้นแก๊สแอมโมเนีย < 25 ppm (ไฟสีเขียวติด)

รูปที่ 4.18 ค่าความเข้มข้นแก๊สแอมโมเนีย >= 25 ppm และ < 100 ppm (ไฟสีเหลืองติด)

รูปที่ 4.19 ค่าความเข้มข้นแก๊สแอมโมเนีย >= 100 ppm (ไฟสีแดงติด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.20 สถานะการเปิดใช้งานของแต่ละเซนเซอร์โนด

รูปที่ 4.21 สถานะการปิดใช้งานของแต่ละเซนเซอร์โนด

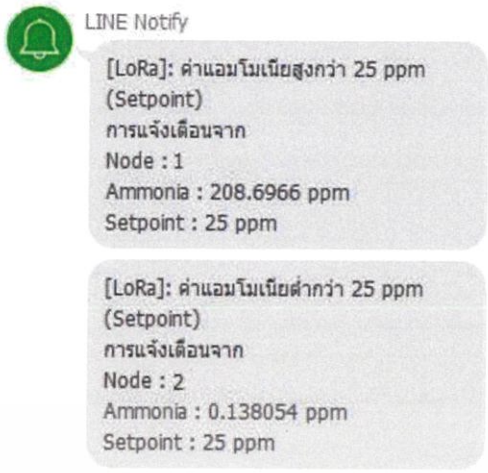
จากรูปที่ 4.20 และ 4.21 แสดงถึงสถานะการเปิด-ปิดการใช้งานของแต่ละเซนเซอร์โนดซึ่งส่วนแสดงผลส่วนนี้มีความสำคัญในการทำให้ผู้ใช้สามารถรู้ว่าเซนเซอร์โนดใดกำลังเปิดหรือปิดใช้งานหรือกำลังส่งข้อมูลหรือไม่ได้ส่งข้อมูลอยู่

4.6 ผลการแจ้งเตือนค่าความเข้มข้นแก๊สแอมโมเนียผ่าน Line Notify

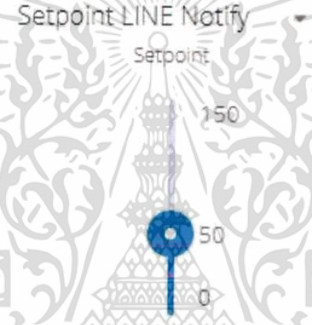
ส่วนนี้เป็นส่วนเพิ่มเติมของระบบการแจ้งเตือนซึ่งผู้จัดทำได้เล็งเห็นความสำคัญของการแจ้งเตือนความเข้มข้นของแก๊สแอมโมเนีย จึงได้นำมาประยุกต์ใช้กับ Node-Red เพื่อให้เป็นตัวจัดการข้อมูลส่งผ่านทางไลน์ของผู้ที่ถูกระบุไอดีไว้ภายใน Node-Red จึงได้รับการแจ้งเตือนโดยการแจ้งเตือนนั้นผู้ใช้ยังสามารถกำหนดค่าแจ้งเตือนด้วย Slide Bar ได้ผ่านทางเว็บเพจของระบบคลาวด์ Ubidots

รูปที่ 4.22 การกำหนดค่าแจ้งเตือน 25 ppm

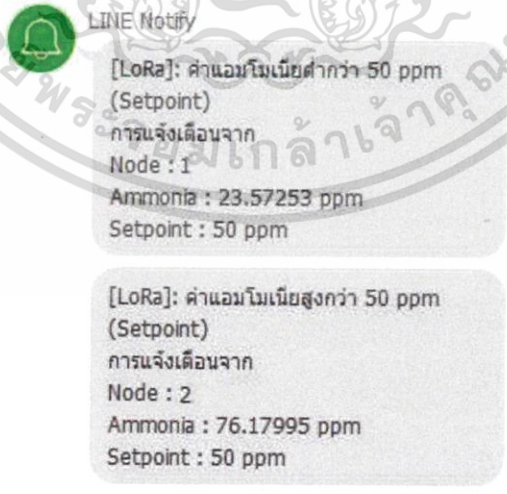
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.23 การแจ้งเตือนผ่าน Line Notify เกิน 25 ppm

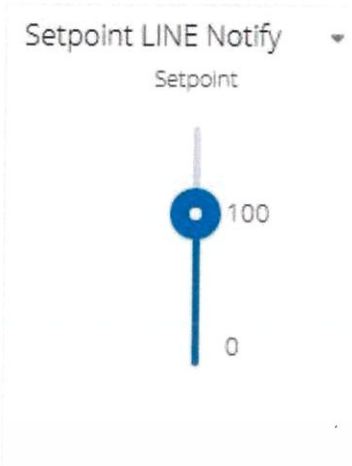


รูปที่ 4.24 การกำหนดค่าแจ้งเตือน 50 ppm

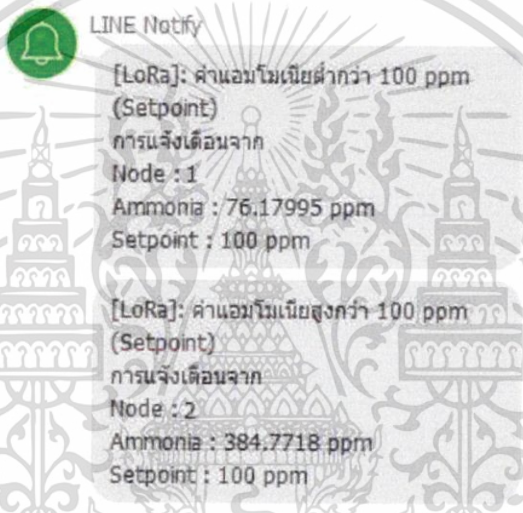


รูปที่ 4.25 การแจ้งเตือนผ่าน Line Notify เกิน 50 ppm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.26 การกำหนดค่าแจ้งเตือน 100 ppm



รูปที่ 4.27 การแจ้งเตือนผ่าน Line Notify เกิน 100 ppm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

จากการดำเนินโครงงานการทำระบบตรวจวัดแก๊สแอมโมเนียแบบไร้สายโดยใช้เทคโนโลยี LoRa ซึ่งทางคณะผู้จัดทำได้ทำการแบ่งโครงสร้างเป็น 3 ส่วนนั้นคือส่วนแรกเซนเซอร์โนด ซึ่งส่วนนี้จะมีการทดสอบโดยการใช้เซนเซอร์ตรวจวัดค่าความเข้มข้นของแก๊สแอมโมเนียและมีการประมวลผลด้วยบอร์ด Arduino Plus LoRa SX1278 และส่งข้อมูลแบบไร้สายระยะไกลไปยังกล่องรับสัญญาณ ส่วนที่สองนั้นคือส่วนของกล่องรับสัญญาณซึ่งมีหน้าที่เป็นโครงสร้างหลักในการรับข้อมูลและส่งข้อมูลเพื่อนำไปใช้ต่อหรือนำไปแสดงผล ซึ่งตัวประมวลผลคือ NodeMCU และถูกจัดการด้วยข้อมูลด้วย Node-Red ส่วนที่สามนั้นคือส่วนของการแสดงผลซึ่งคณะผู้จัดทำเลือกใช้เว็บเพจของระบบคลาวด์ Ubidots เป็นตัวแสดงผลและเก็บข้อมูล โดยในทั้งสามส่วนที่กล่าวมานั้นเชื่อมโยงเกี่ยวข้องกันทั้งหมดด้วยการเขียนโปรแกรมหรือชุดคำสั่งตั้งแต่ที่ฝั่งของเซนเซอร์โนดเพื่อที่จะอ่านค่าจากเซนเซอร์วัดความเข้มข้นแก๊สแอมโมเนียและนำข้อมูลส่งต่อในรูปแบบของ JSON ไปยังกล่องรับสัญญาณซึ่งภายในส่วนของกล่องรับสัญญาณนั้นก็มีการเขียนชุดคำสั่งไว้สำหรับการรับข้อมูลและจัดรูปแบบข้อมูลพร้อมทั้งส่งข้อมูลที่ต้องการขึ้นไปยัง Cloud Computing Service และคณะผู้จัดทำได้ใช้เว็บเพจของระบบคลาวด์ Ubidots เพื่อการแสดงผลผ่านระบบเครือข่าย ซึ่งจากผลการทดลองโครงงานนี้เป็นไปตามวัตถุประสงค์ของคณะผู้จัดทำทั้งในเรื่องของการส่งข้อมูลค่าความเข้มข้นแก๊สแอมโมเนียด้วยเทคโลยี LoRa การแสดงผลไปยังบนเว็บเพจ Ubidots และยังสามารถนำไปต่อยอดได้

5.2 ข้อเสนอแนะ

1. เทคโนโลยี LoRa นั้นสามารถนำประยุกต์เข้ากับเซนเซอร์ต่างๆได้ เช่น การวัดความด่างของน้ำและตาม Concept ของเทคโนโลยี LoRa ควรใช้เซนเซอร์ที่มีอัตราการกินไฟต่ำเพื่ออายุแบตเตอรี่ที่ยาวนานกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] “ความอันตรายของแก๊สแอมโมเนีย”

แหล่งที่มา :

http://envocc.ddc.moph.go.th/contents/view/68?fbclid=IwAR1Nyr_ukQXvjowX

[2] “IoT”

แหล่งที่มา : https://blog.sogoodweb.com/Article/Detail/59554

[3] “LoRa, LoRaWAN” เข้าถึงเมื่อ 10/03/2562

แหล่งที่มา : https://medium.com/deaware/lora-LoRaWAN

[4] “วิธีการใช้Arduino IDE พื้นฐาน” เข้าถึงเมื่อ 12/03/2562

แหล่งที่มา : https://poundxi.com

[5] “Model MQ137”

แหล่งที่มา : https://www.winsen-sensor.com/d/files/semiconductor/mq137.pdf

[6] “Arduino” เข้าถึงเมื่อ 12/03/2562

แหล่งที่มา : https://www.gravitechthai.com/guru2.php?p=215

[7] “Raspberry Pi” เข้าถึงเมื่อ 08/03/2562

แหล่งที่มา : https://www.arduinospro.com/article/51/raspberry-pi-tutorial-ep1-raspberry-pi

[8] “ESP 8266” เข้าถึงเมื่อ 15/03/2562

แหล่งที่มา : http://narong.ece.engr.tu.ac.th/ei444/document/ESP8266.pdf

[9] “MODBUS Protocol” เข้าถึงเมื่อ 15/03/2562

แหล่งที่มา : https://riverplusblog.com/2011/08/18/plc-protocol

[10] “ความรู้พื้นฐานเกี่ยวกับ Protocol MQTT” เข้าถึงเมื่อ 17/03/2562

แหล่งที่มา : https://www.ioxhop.com/article/74/ESP32

[11] “RSSI”

แหล่งที่มา : https://dict.drkrok.com/rssi/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดคำสั่งของเซนเซอร์โนด

```
#include <avr/sleep.h>
#include <avr/power.h>
#include <avr/wdt.h>
#include <ArduinoJson.h>
#include<LoRa.h>
#define RL 4.7
#define RO 15.10
#define sensor_pin A1     //Sensor Pin
unsigned long previousMillis = 0;
const long interval = 1000;
unsigned long currentMillis;
int timer=0;
String incoming;
int statee;
int reading;
float value;
volatile int f_wdt=1;
int f_cycle;
int second;
String Buffer;
char sendvalue[70];
float ppm;
float Rs;
float Ratio;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************************************************
 * Name:        ISR(WDT_vect)
 *
 * Returns:     Nothing.
 *
 * Parameters:  None.
 *
 * Description: Watchdog Interrupt Service. This
 *              is executed when watchdog timed out.
 *
 *****************************************************/
ISR(WDT_vect)
{
  if(f_wdt == 0)
  {
    f_wdt=1;
  }
  else
  {
   // Serial.println("WDT Overrun!!!");
  }
}
/*****************************************************
 * Name:        enterSleep
 *
 * Returns:     Nothing.
 *
 * Parameters:  None.
 *
 * Description: Enters the arduino into sleep mode.
 *
 *****************************************************/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void enterSleep(void)
{
  set_sleep_mode(SLEEP_MODE_PWR_DOWN);   /* EDIT: could also use
SLEEP_MODE_PWR_DOWN for lowest power consumption. */
  sleep_enable();
  /* Now enter sleep mode. */
  sleep_mode();
  /* The program will continue from here after the WDT timeout*/
  sleep_disable(); /* First thing to do is disable sleep. */
  /* Re-enable the peripherals. */
  power_all_enable();
}
void ReadSensor() {
  reading = analogRead(sensor_pin);   //Read 0-1023 from sensor
  value = reading * 5.00 / 1023.00 ;      //Convert to 0-5
  Rs = ((5.000/RL)-1.000)*value;
  Ratio = Rs/Ro;
  ppm = pow((1.7094*Ratio),(-1.000/0.257));
  delay(random(1500));
}
//Monitor via Serial Monitor
 /*
  Serial.print("Read : ");
  Serial.print(reading);
  Serial.print(" (");
  Serial.print(value);
  Serial.print(" ,");
  Serial.print(ppm);
  Serial.print(" ppm");
  Serial.println("). ");
 */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void JSON_Generate() {      //Convert Data to JSON Format
  StaticJsonBuffer<200> jsonBuffer;
  //DynamicJsonBuffer  jsonBuffer(100);
  JsonObject& root = jsonBuffer.createObject();
  root["node"] = "2";
  root["volt"] = value;
  root["value"] = ppm;
  root.printTo(Buffer);
  memset(sendvalue,0,sizeof(sendvalue));
  Buffer.toCharArray(sendvalue, sizeof(sendvalue));
  //Serial.print("JSON : ");
  Serial.println(Buffer);
}
void SendMsgtoGateway() {
  LoRa.beginPacket();
  LoRa.print(sendvalue);
  LoRa.endPacket();
  Buffer = "";
}
void recieveMessage(){
  Serial.println("Recieve Message...");
  bool check=false;
  while(!check) {
    currentMillis = millis();
    if(currentMillis - previousMillis >= interval){
    previousMillis = currentMillis;
    timer++;
   // Serial.println(timer);
    int packetSize = LoRa.parsePacket();
      if(packetSize)  {
      // Recieved a packet
      //Serial.println("Recieve packet '");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        incoming = "";
        while (LoRa.available()){
          incoming = LoRa.read();
        }
         // Serial.println(incoming);
          if(incoming == "49"){
            digitalWrite(6, HIGH);
            Serial.println("ON");
          }
          else if(incoming == "48"){
            digitalWrite(6, LOW);
            Serial.println("OFF");
          }
        }
      }
      if(timer >= 10){
      check = true;
      timer = 0;
    }
    }
        f_cycle = 0;
        f_wdt = 0;
        enterSleep();
}
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
    while (!Serial);
  Serial.println("LoRa Sender");
  //Set LoRa Parameter
  if (!LoRa.begin(915E6)) {
    Serial.println("Starting LoRa failed!");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    while (1);
  }
  pinMode(6, OUTPUT);
  LoRa.setSpreadingFactor(10);
  LoRa.setSignalBandwidth(31.25E3);
  LoRa.setTxPower(17);
  Serial.println("Initialising...");
  delay(100); //Allow for serial print to complete.
  /*** Setup the WDT ***/
  /* Clear the reset flag. */
  MCUSR &= ~(1<<WDRF);
  /* In order to change WDE or the prescaler, we need to
   * set WDCE (This will allow updates for 4 clock cycles).
   */
  WDTCSR |= (1<<WDCE) | (1<<WDE);
  /* set new watchdog timeout prescaler value */
  WDTCSR = 1<<WDP0 | 1<<WDP3; /* 8.0 seconds */
  /* Enable the WD interrupt (note no reset). */
  WDTCSR |= _BV(WDIE);
  Serial.println("Initialisation complete.");
  Serial.println("");
  delay(100); //Allow for serial print to complete.

}

void loop() {
   if(f_wdt == 1)
  {
   // Toggle the LED //
   //Serial.println("Wake Up...");
   //delay(100);
   f_cycle++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    second = f_cycle*8;
    //Serial.print("Count : ");
    //Serial.print(f_cycle);
    //Serial.print(" | ");
    //Serial.print(second);
    //Serial.println(" Seconds");
    //delay(100)
    if(f_cycle == 2){
      //Serial.println(F("Sending..."));
      //delay(100);
      ReadSensor();
      JSON_Generate();
      SendMsgtoGateway();
      //recieveMessage();
      //Serial.println("Send Successful...");
      //Serial.println();
      f_cycle = 0;
      f_wdt = 0;
      enterSleep();
    }
    else{
    // Don't forget to clear the flag. //
    f_wdt = 0;
    //Serial.println("Sleep...");
    //Serial.println("");
    //delay(100);
    // Re-enter sleep mode. //
    enterSleep();
    }
}
else
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    // Do nothing. //
  }
}
```

ชุดคำสั่งกล่องรับสัญญาณ

```
#include <ESP8266WiFi.h>
#include<LoRa.h>
#include <ArduinoJson.h>
String incoming;
String incoming_buffer;
char incomingBuffer[150];
int count=0;
bool check;
unsigned long previousMillis = 0;
const long interval = 1000;
unsigned long currentMillis;
int timer=0;
char recieveubi;
String tonodered;
int nodeBuffer;
float valueBuffer;
float voltBuffer;
int digiBuffer;
void LoRa_Setup() {
 //Serial.println("LoRa Receiver");
 LoRa.setPins(15,17,16);          //Set NSS(Chip Select)
 if (!LoRa.begin(915E6)) {        //Frequency (915 MHz)
   //Serial.println("Starting LoRa failed!");
   while (1);
 }
 LoRa.setSpreadingFactor(10);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  LoRa.setSignalBandwidth(31.25E3);
  LoRa.setTxPower(17);
}
void recieve_Node() {
  int packetSize = LoRa.parsePacket();
  if(packetSize) {
  incoming = "";
  while (LoRa.available()){
    incomingBuffer[count] = LoRa.read();
    count++;
  }
  Parse_JSON();
  JSON_Generate();
  count = 0;
  memset(incomingBuffer,0,sizeof(incomingBuffer));
  }
}
void Parse_JSON(){
  StaticJsonBuffer<200> jsonBuffer;
  JsonObject& root = jsonBuffer.parseObject(incomingBuffer);
  nodeBuffer = root["node"];
  valueBuffer = root["value"];
  voltBuffer = root["volt"];
  digiBuffer = root["digi"];
}
void JSON_Generate() {
  StaticJsonBuffer<200> jsonBuffer;
  JsonObject& root = jsonBuffer.createObject();
  root["node"] = nodeBuffer;
  root["value"] = valueBuffer;
  root["volt"] = voltBuffer;
  root["rssi"] = LoRa.packetRssi();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  root["digi"] = digiBuffer;
  root["active"] = "1";
  root.printTo(tonodered);
  Serial.print(tonodered);
  Serial.println();
  tonodered = "";

}
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  WiFi.softAPdisconnect (true);  //Disable AP
  pinMode(4, OUTPUT);
  LoRa_Setup();
}

void loop() {
  // put your main code here, to run repeatedly:
  recieve_Node();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้