

อุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IoT
MOBILE DUST MEASUREMENT DEVICE

นายวัชรกรณ ตั้งรัตนสมบูรณ์
นายชิษณุพงศ์ พฤชากิจ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ ภาควิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีและการจัดการอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2564
ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

อุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IoT

นายวัชรกรรณ์ ตั้งรัตนสมบูรณ์

นายชิษณุพงศ์ พุกษากิจ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ ภาควิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีและการจัดการอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ปีการศึกษา 2564

ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

MOBILE DUST MEASUREMENT DEVICE

MR.VADCHARAGORN TANGRATTANASOMBOON

MR.CHISANUPONG PHUESAKIT

PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE BACHELOR'S DEGREE OF INDUSTRIAL TECHNOLOGY
DEPARTMENT OF INFORMATION TECHNOLOGY
FACULTY OF INDUSTRIAL TECHNOLOGY AND MANAGEMENT
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK

2021

COPYRIGHT OF KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK



ใบรับรองปริญญาานิพนธ์

คณะเทคโนโลยีและการจัดการอุตสาหกรรม

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

เรื่อง อุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IoT

โดย นาย วัชรกรณ์ ตั้งรัตนสมบูรณ์

นาย ชิชณพงศ์ พุกษากิจ

ได้รับอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

_____ คณบดี

(ผู้ช่วยศาสตราจารย์ ดร.กฤษฎากร บุตดาจันทร์)

คณะกรรมการสอบปริญญาานิพนธ์

ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์นิมิต ศรีคำทา)

กรรมการ

(จำสืบัติรี นพเก้า ทองใบ)

กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.นัฐพันธ์ นาคพงษ์)

ชื่อ : นายวัชรกรณ ตั้งรัตนสมบุญ
นายชิษณุพงศ์ พฤชากิจ
ชื่อปริญญาบัตร : อุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IoT
สาขาวิชา : เทคโนโลยีสารสนเทศ
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
อาจารย์ที่ปรึกษาปริญญาบัตร : ผู้ช่วยศาสตราจารย์ ดร.นฤพนธ์ นาคพงษ์
ปีการศึกษา : 2564


บทคัดย่อ

ในปัจจุบัน การเดินทางมีความสำคัญอย่างมากในการใช้ชีวิตประจำวัน โดยส่วนใหญ่จะเป็นการเดินทางบก และอาจจำเป็นต้องเดินทางผ่านบริเวณที่มีการก่อสร้าง หรือโรงงานอุตสาหกรรม ทำให้ปริมาณฝุ่นในบริเวณนั้น ๆ เพิ่มขึ้น โดยฝุ่นที่เกิดขึ้นมีได้ตั้งแต่ PM 1.0 ถึง PM 10.0 แต่ฝุ่นที่ควรระวังมากที่สุดคือฝุ่นขนาดเล็กไม่เกิน 2.5 ไมครอน หาก PM 2.5 แพร่กระจายเข้าสู่ทางเดินหายใจไปจนถึงกระแสเลือด และสารอื่น ๆ ที่นำเข้ามาด้วย เช่น แคดเมียม ปรอท โลหะหนักและ สารก่อมะเร็งอื่น ๆ เข้าสู่ร่างกาย จะทำให้เกิดอันตรายต่อสุขภาพได้ จึงได้มีการสร้างอุปกรณ์วัดฝุ่นเคลื่อนที่ขึ้นมาเพื่อวัดฝุ่นตามเส้นทางต่าง ๆ เพื่อช่วยให้ผู้ใช้สามารถรู้ถึงปริมาณฝุ่นละอองในแต่ละบริเวณที่อุปกรณ์เดินทางไป และลดอันตรายจากฝุ่นในการเดินทาง

จากการพัฒนาอุปกรณ์วัดฝุ่นเคลื่อนที่ มีวัตถุประสงค์เพื่อ พัฒนาโครงสร้างของอุปกรณ์วัดฝุ่นเคลื่อนที่และแสดงผลบนเว็บไซต์ได้โดยแสดงตำแหน่งฝุ่นบนแผนที่ การพัฒนาครั้งนี้ คณะผู้จัดทำโครงงานระบบอุปกรณ์วัดฝุ่นเคลื่อนที่ เพื่ออำนวยความสะดวกให้แก่ผู้ใช้ทั่วไป โดยอุปกรณ์ที่ติดอยู่บนรถโดยสารหรือยานพาหนะต่าง ๆ เมื่อตรวจจับฝุ่นได้จะส่งค่าไปยัง Mongo DB และแสดงผลผ่านบน เว็บไซต์ได้

(ปริญญาบัตรมีจำนวนทั้งสิ้น 67 หน้า)

ส่วนสำคัญ : ฝุ่น ไมครอน สภาพอากาศ ฐานข้อมูล



อาจารย์ที่ปรึกษาปริญญาบัตร

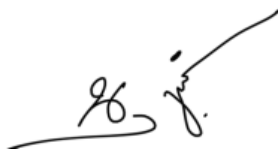
Name : Mr.Vadcharagorn Tangrattanasomboon
Mr.Chisanupong Phuesakit
Project Title : Mobile dust measurement device by IoT technology
Major Field : Information Technology King Mongkut's University of Technology
North Bangkok Prachinburi campus
Project Advisor : Assistant Professor Dr. Nuttapun Nakpong
YEAR : 2021

Abstracts

Nowadays, travel is very important in everyday life. Most of them will be on land. and may need to travel through the construction site or industrial plants causing the amount of dust in that area to increase. The dust that occurs can range from PM 1.0 to PM 10.0, but the dust that should be most careful is small dust, not more than 2.5 microns. If PM 2.5 spreads into the respiratory tract to The bloodstream and other substances brought with it, such as cadmium, mercury, heavy metals, and other carcinogens. enter the body can cause health hazards Therefore, a mobile dust measuring device was created to measure dust along various paths. To help users know the amount of dust in each area the device travels. and reduce the danger of dust from traveling

from the development of a mobile dust measuring device aimed at Developing the structure of the mobile dust measuring device and displaying it on the website by showing the dust location on the map. this development Project committee Mobile dust measurement system for convenience to general users A device attached to a bus or vehicle, when it detects dust, will send the value to the Mongo DB and display it on the website.

(Thesis, there are a total of 67 pages)



Project advisors

กิตติกรรมประกาศ

ปริญญานิพนธ์เรื่อง “อุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IoT” สำเร็จ ลงได้ด้วยดี ทางผู้จัดทำต้องขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.นัฐพันธ์ นาคพงษ์ ซึ่งเป็นอาจารย์ ที่ปรึกษาที่ได้กรุณาให้คำปรึกษาและแนะนำแนวทางในการจัดทำปริญญานิพนธ์ รวมทั้งคณาจารย์ภาควิชาเทคโนโลยีสารสนเทศ ที่ได้ให้ความรู้กับผู้จัดทำเพื่อนำมาประยุกต์ในการจัดทำโครงการนี้

ท้ายที่สุดนี้ ทางผู้จัดทำขอน้อมรำลึกพระคุณบิดา มารดา ผู้ซึ่งมีพระคุณอย่างสูงสุดที่ให้ความอุปการะผู้จัดทำมาโดยตลอด รวมทั้งผู้มีพระคุณทุกท่าน คณะผู้จัดทำรู้สึกซาบซึ้งเป็นอย่างยิ่งจึงใคร่ขอขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้ด้วย

วัชรกรรณ์ ตั้งรัตนสมบุรณ์

ชิษณุพงศ์ พฤษากิจ

สารบัญ

	หน้า
บทคัดย่อ	ข
ABSTRACTS	ค
กิตติกรรมประกาศ	ง
สารบัญ	จ
สารบัญภาพ	ช
สารบัญภาพ (ต่อ)	ซ
สารบัญตาราง	ณ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการวิจัย	2
1.3 ขอบเขตการวิจัย	2
1.4 นิยามศัพท์เฉพาะ	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	4
2.1 ทฤษฎีที่เกี่ยวข้อง	4
2.2 งานวิจัยที่เกี่ยวข้อง	19
2.3 บทสรุป	21
บทที่ 3 วิธีการดำเนินการ	22
3.1 กำหนดปัญหา	22
3.2 การรวบรวมข้อมูล	23
3.3 การวิเคราะห์	23
3.4 การออกแบบระบบ	27
3.5 การพัฒนา	31
3.6 การทดสอบระบบ	33

สารบัญ (ต่อ)

	หน้า
3.7 การนำไปใช้และการบำรุงรักษา	33
3.8 บทสรุป	33
บทที่ 4 ผลการดำเนินงาน	34
4.1 ผลลัพธ์ของอุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IoT	35
4.2 ผลการของการทดลอง	41
4.3 สรุปผลดำเนินงาน	43
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ	44
5.1 สรุปผลดำเนินงาน	44
5.2 ข้อจำกัดในการทำงาน	44
5.3 ปัญหาและอุปสรรค	44
5.4 ข้อเสนอแนะ	45
บรรณานุกรม	46
ภาคผนวก ก. คู่มือการใช้งานอุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IOT	49
ภาคผนวก ข. ใ้ดการทำงาน	51
ภาคผนวก ค. การต่อวงจร	65

สารบัญภาพ

ภาพที่	หน้า
2-1 เซ็นเซอร์วัดฝุ่น SENSIRION	5
2-2 การส่งข้อมูลของ NODE-RED	6
2-3 ฝุ่นละอองในเมืองใหญ่	7
2-4 ฝุ่น PM10 (น้ำเงิน) และฝุ่น PM2.5 (แดง) เทียบกับเส้นผม	8
2-5 ESP32 DAC EXAMPLE	9
2-6 โครงสร้างภาษา HTML	10
2-7 โครงสร้างภาษา JAVA	12
2-8 โครงสร้างภาษา ARDUINO (C/C++)	12
2-9 INTERNET OF THINGS (IOT)	13
2-10 การส่งผ่านข้อมูลผ่านระบบ MQTT	14
2-11 การส่งผ่านข้อมูลผ่านระบบ NB-IOT	15
2-12 การเก็บข้อมูลผ่านระบบ NB-IOT	16
2-13 SOLAR CELL	17
2-14 BATTERY	18
2-15 GPS MODULE GY-NEO6MV2	19
3-1 แสดงการทำงานของอุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IOT	23
3-2 BLOCK DIAGRAM อุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IOT	24
3-3 FLOWCHART อุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IOT	25
3-4 FLOWCHART การส่งข้อมูลอุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IOT ไปยังเว็บไซต์	26
3-5 แบบจำลองการติดตั้งอุปกรณ์ด้านหน้า	27
3-6 แบบจำลองการติดตั้งอุปกรณ์ด้านหลัง	27
3-7 แผนภาพจำลองการติดตั้งอุปกรณ์ภายใน	28
3-8 แผนผังเว็บไซต์	28
3-9 แบบจำลองการออกแบบเว็บไซต์	29

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3-10 แผนภาพจำลองการติดตั้งอุปกรณ์	30
3-11 แสดงการทำงานของระบบอุปกรณ์วัดฝุ่นเคลื่อนที่	30
3-12 โปรแกรม ARDUINO IDE	31
3-13 แสดง DATAFLOW การทำงานของ NODE-RED	31
3-14 การเก็บข้อมูลบน MONGO DB	32
3-15 ตัวอย่าง CODE HTML และ JAVA SCRIPT ในการพัฒนาเว็บไซต์	33
3-16 ตัวอย่าง การส่งข้อมูลจากอุปกรณ์ไปยังเว็บไซต์	33
4-1 อุปกรณ์วัดฝุ่น	35
4-2 อุปกรณ์วัดฝุ่นเคลื่อนที่ด้านหน้า	36
4-3 อุปกรณ์วัดฝุ่นเคลื่อนที่ด้านหลัง	36
4-4 อุปกรณ์วัดฝุ่นเคลื่อนที่ด้านซ้าย	37
4-5 อุปกรณ์วัดฝุ่นเคลื่อนที่ด้านขวา	37
4-6 อุปกรณ์วัดฝุ่นเคลื่อนที่ด้านบน	38
4-7 เว็บไซต์แสดงผลหน้า MAP	39
4-8 เว็บไซต์แสดงผลหน้า DATA LIST	40
4-9 เว็บไซต์แสดงผลหน้า ABOUT AIR QUALITY	40
4-10 เว็บไซต์แสดงผลหน้า CONTACT US	40
4-11 ทดสอบการวัดฝุ่นโดยการติดตั้งบนรถยนต์	41
4-12 ทดสอบการวัดฝุ่นขณะขับรถยนต์	41
4-13 ทดสอบการแสดงผลปริมาณฝุ่นที่วัดได้ขณะขับรถ	42
4-14 ทดสอบการแสดงผลปริมาณฝุ่นที่วัดได้ขณะขับรถ	42
ก-1 อุปกรณ์วัดฝุ่นเคลื่อนที่	50
ก-2 เว็บไซต์แสดงผลหน้า MAP	50
ค-1 การต่อวงจร	66

สารบัญตาราง

ตารางที่	หน้า
4-1 ตำแหน่งอุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IOT	35
ค-1 การเชื่อมต่อของอุปกรณ์	66

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบัน การเดินทางมีความสำคัญอย่างมากในการใช้ชีวิตประจำวัน โดยส่วนใหญ่จะเป็นการเดินทาง ทางบก ซึ่งตัวเลือกในการเดินทางส่วนใหญ่คือ รถยนต์ รถจักรยานยนต์ รถประจำทาง การใช้ยานพาหนะเหล่านี้ทำให้เกิดมลพิษต่าง ๆ ตามมาแต่มลพิษที่น่ากังวลที่สุดคือฝุ่นละออง เพราะฝุ่น ไม่มีกลิ่น และไม่สามารถมองเห็นได้ด้วยตาเปล่า ฝุ่นละอองไม่ได้มาจากยานพาหนะเพียงอย่างเดียว ฝุ่นส่วนใหญ่เกิดจากการกระทำของมนุษย์ เช่น การก่อสร้าง โรงงานอุตสาหกรรม ซึ่งการเดินทางบางครั้งอาจจำเป็นต้องเดินทางผ่านบริเวณที่มีการก่อสร้าง หรือโรงงานอุตสาหกรรม ทำให้ปริมาณฝุ่นในบริเวณนั้น ๆ เพิ่มขึ้น ฝุ่นที่เกิดขึ้นมีได้ตั้งแต่ PM 1.0 ถึง PM 10.0 แต่ฝุ่นที่ควรระวังมากที่สุดคือฝุ่นขนาดเล็กไม่เกิน 2.5 ไมครอน หรือ PM 2.5 เทียบได้ว่าขนาดประมาณ 1 ใน 25 ของเส้นผ่าศูนย์กลางเส้นผมมนุษย์ แม้จะเป็นขณจุมกของมนุษย์ที่ทำหน้าที่กรองฝุ่นก็ไม่สามารถกรองได้ หาก PM 2.5 แพร่กระจายเข้าสู่ทางเดินหายใจไปจนถึงกระแสเลือด สารอื่นที่ฝุ่นเป็นพาหะนำเข้ามาด้วย เช่น แคดเมียม ปรอท โลหะหนักและสารก่อมะเร็งอื่น ๆ เข้าสู่ร่างกาย จะทำให้เกิดอันตรายต่อสุขภาพได้

การสร้างอุปกรณ์วัดฝุ่นเคลื่อนที่ขึ้นมาเพื่อวัดฝุ่นตามเส้นทางต่าง ๆ เพื่อช่วยให้ผู้ใช้สามารถรู้ถึงปริมาณฝุ่นละอองในแต่ละบริเวณที่อุปกรณ์เดินทางไป และลดอันตรายจากฝุ่นในการเดินทาง แต่จากการพัฒนาในครั้งนี้เกิดข้อจำกัดคือมีเซ็นเซอร์วัดฝุ่นเพียงตัวเดียวทำให้การวัดอาจเกิดความคลาดเคลื่อน และข้อมูลอาจผิดพลาดได้ เช่น ฝุ่นจากควันเสียรถที่เกิดขึ้นจากเพียงชั่วขณะในบริเวณที่ฝุ่นมีปริมาณต่ำทำให้ข้อมูลผิดพลาด และยังมีข้อจำกัดด้านการเก็บข้อมูล ที่ยังใช้ฐานข้อมูลเป็น MySQL ซึ่งไม่เหมาะกับการเก็บข้อมูลด้านนี้ควรที่จะมีการรับส่งข้อมูลแบบ real-time เพื่อให้ข้อมูลมีความถูกต้องที่สุด และในส่วนของการแสดงผล การนำข้อมูลมาแสดงไม่เป็นที่พอใจนัก

จากข้อมูลที่กล่าวข้างต้น คณะผู้จัดทำโครงการพิเศษ จึงได้มีแนวคิดที่จะนำอุปกรณ์วัดฝุ่นเคลื่อนที่มาพัฒนาต่อ โดยนำอุปกรณ์วัดฝุ่นเคลื่อนที่มาพัฒนาด้านโครงสร้าง โดยเปลี่ยนตำแหน่งเซ็นเซอร์จากด้านหน้าเป็นด้านข้างเพื่อเพิ่มความถูกต้องของข้อมูล สิ่งต่อมาที่พัฒนาคือด้านการเก็บข้อมูล ซึ่งการเก็บข้อมูลในปัจจุบันใช้เป็นฐานข้อมูล MySQL ซึ่งผู้จัดทำโครงการพิเศษเห็นว่าเป็นการเก็บข้อมูลที่ไม่เหมาะสมกับการทำงานในด้าน IoT จึงได้มีการเปลี่ยนการเก็บข้อมูลจาก MySQL มาเป็น การเก็บข้อมูลแบบ NOSQL โดยใช้ MongoDB ในการเก็บข้อมูล และใช้ Node-Red ให้ส่งข้อมูลมาเพื่อแสดงผลใน Website ได้แบบ real time

1.2 วัตถุประสงค์ของการวิจัย

- 1.2.1 เพื่อพัฒนาโครงสร้างของอุปกรณ์วัดฝุ่นเคลื่อนที่
- 1.2.2 เพื่อพัฒนาการแสดงผลบนเว็บไซต์ได้
- 1.2.3 เพื่อพัฒนาการแสดงตำแหน่งฝุ่นบนแผนที่ได้
- 1.2.4 เพื่อพัฒนาระบบเก็บข้อมูล และการวิเคราะห์ ให้เห็นภาพข้อมูลได้ชัดเจนมากขึ้น

1.3 ขอบเขตการวิจัย

- 1.3.1 อุปกรณ์สามารถวัดฝุ่นได้
- 1.3.2 อุปกรณ์สามารถส่งข้อมูลขึ้นไปเก็บบนฐานข้อมูล MongoDB ได้
- 1.3.3 อุปกรณ์มี LED แสดง ดัชนีคุณภาพอากาศ เป็นสีตามระดับฝุ่นในบริเวณที่วัดได้
- 1.3.4 อุปกรณ์สามารถติดตั้งบนรถเมล์และรถยนต์ได้
- 1.3.5 อุปกรณ์มีขนาดเหมาะสมในการใช้งานและติดตั้งง่าย บริเวณหลังคารถ
- 1.3.6 มี GPS Module บอกตำแหน่งของการวัดฝุ่นในแต่ละจุด
- 1.3.7 หน้าเว็บแสดงผลสามารถดึงข้อมูลมาจากฐานข้อมูล MongoDB และแสดงข้อมูลได้
- 1.3.8 หน้าเว็บมีแผนที่สำหรับแสดงค่าฝุ่นเป็นจุดสีต่างๆตามปริมาณฝุ่นที่วัดได้ ตามเวลาฝุ่นต่อวันได้

1.4 นิยามศัพท์เฉพาะ

1.4.1 อุปกรณ์วัดฝุ่นเคลื่อนที่ หมายถึง อุปกรณ์ที่มีเซ็นเซอร์วัดฝุ่นติดตั้งอยู่ และอุปกรณ์ถูกติดตั้งอยู่บนหลังคารถ 4 ล้อขึ้นไป อุปกรณ์จะทำการวัดฝุ่น จากนั้นจะส่งข้อมูลขึ้นไปเก็บไว้บนฐานข้อมูล และนำมาแสดงผลบนหน้าเว็บ

1.4.2 NB-IoT หมายถึง มาตรฐานระบบโครงข่ายที่ใช้พลังงานต่ำ (Low Power Wide Area Network (LPWAN) ที่ถูกพัฒนามาเพื่อให้อุปกรณ์ต่างๆ สามารถเชื่อมต่อเข้าหากันได้โดยผ่านโครงข่ายของสัญญาณโทรศัพท์เคลื่อนที่ คลื่นมือถือที่ทำให้อุปกรณ์ต่างๆ เชื่อมต่อเข้าหากันได้โดยใช้พลังงานที่ต่ำ และข้อมูลที่รับส่งจะมีขนาดเล็ก [1]

1.4.3 MongoDB หมายถึง open-source document database ประเภทหนึ่ง โดยเป็น database แบบ NoSQL Database จะไม่มีการใช้คำสั่ง SQL ไม่เน้นในการสร้างความสัมพันธ์ของข้อมูลแต่จะเป็นรูปแบบโครงสร้างที่เจ้าของ NoSQL สร้างขึ้นมาเองและจัดเก็บข้อมูลเป็นแบบ JSON (JavaScript Object Notation) ซึ่งจะเก็บค่าเป็น key และ value [2]

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 ได้โครงสร้างอุปกรณ์วัดฝุ่นเคลื่อนที่มีประสิทธิภาพสูงขึ้น
- 1.5.2 ได้หน้าเว็บที่มีความถูกต้อง และมีประสิทธิภาพ
- 1.5.3 ได้ระบบเก็บข้อมูลที่มีประสิทธิภาพสูงขึ้น
- 1.5.4 ได้ระบบเก็บข้อมูลที่มีประสิทธิภาพสูงขึ้น

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ในการจัดทำโครงการพิเศษนี้ได้ศึกษาเอกสารและงานวิจัยที่เกี่ยวข้องและได้นำหลักการมาใช้ในการพัฒนาระบบวัดฝุ่นแบบเคลื่อนที่เพื่อเมืองอัจฉริยะตามหัวข้อดังต่อไปนี้

2.1 ทฤษฎีที่เกี่ยวข้อง

2.2 งานวิจัยที่เกี่ยวข้อง

2.3 บทสรุป

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 เซ็นเซอร์วัดฝุ่น

เซ็นเซอร์วัดฝุ่น (Dust Sensor) เป็นเซ็นเซอร์ที่ใช้ตรวจจับอนุภาคขนาดเล็กในอากาศ ใช้พลังงานต่ำที่เป็นแบบอนาล็อกมีหน่วยวัดมาตรฐานคือไมโครกรัมต่อลูกบาศก์เมตร มีเกณฑ์ที่ใช้แบ่งช่วงคุณภาพของอากาศเพื่อเปรียบเทียบกับค่าที่ได้จากเซ็นเซอร์ทำให้ง่ายต่อการวิเคราะห์ และสรุปผล

หลักการทำงานของ Optical Dust Sensor ตัวเครื่องจะเป็นกล่องขนาดเล็กที่มีรูตรงกลาง ข้างในเครื่องจะมีอุปกรณ์ปล่อยแสง LED และ อุปกรณ์ตรวจจับค่าการสะท้อนแสงถูกวางไว้ให้ลำแสงของทั้งสองอุปกรณ์ตัดกันตรงรูปพอดี เมื่อมีฝุ่นละอองตกเข้าไปภายในรูก็จะทำให้ค่าการสะท้อนของแสงที่วัดได้เปลี่ยน เนื่องจากฝุ่นไปทำให้แสงหักเห [3]

Sensirion Particulate Matter Sensor SPS30 สามารถวัดค่าความเข้มข้นของฝุ่นได้ตั้งแต่ 1 ไมครอน, 2.5 ไมครอน, 4 ไมครอนและ 10 ไมครอน

SPS30 มีอินเทอร์เฟซ 5 พินที่สามารถสื่อสารผ่านโปรโตคอลที่แตกต่างกันสองแบบ: UART และ I2C SPS30 ต้องใช้แหล่งจ่ายไฟ 5V แต่สามารถทำงานร่วมกับไมโครคอนโทรลเลอร์ 3.3V และ 5V



ภาพที่ 2-1 เซ็นเซอร์วัดฝุ่น Sensirion

ที่มา: <https://www.sensirion.com/en/environmental-sensors/particulate-matter-sensors-pm25/>

2.1.2 Node-RED

Node-RED มีแนวคิดการทำงานแบบ Flow-based programming for the Internet of Things โดย Node-RED สร้างบน Node.js ทำให้สามารถเชื่อมโยงกับโมดูล ที่กำเนิดบนโครงสร้างเดียวกันได้มากมาย คุณสามารถเพิ่มโมดูล Arduino โดยอาศัย Firmata ติดต่อกับฐานข้อมูลแบบ NoSQL อย่าง MongoDB และ เป็นเครื่องมือจัดการและจัดการเหตุการณ์ โดยผู้ใช้สามารถปรับแต่งและจัดการการเชื่อมต่อระหว่างฮาร์ดแวร์ต่าง ๆ และสร้างขั้นตอนการทำงานจากเบราว์เซอร์ของคอมพิวเตอร์เครื่องใดก็ได้ ไม่มีซอฟต์แวร์ที่มีราคาแพงหรือทำให้หน่วยความจำมากเกินไป และยังสามารถในการทำงานบนอุปกรณ์เช่น Raspberry Pi หรือ Arduino ฟังก์ชันทั้งหมดของ Node-RED นั้นไม่จำเป็นต้องป้อนรหัสทั้งหมด และทำงานกับโค้ดที่มีอยู่แล้ว โดยอินเทอร์เฟซผู้ใช้ของ Node-RED ดูเรียบง่ายและเปิดเผยไม่มีปัญหาในการพัฒนาโครงการ IoT ด้วย กระแส Node-RED สามารถแสดง JSON (JavaScript Object Notation) ทำให้สามารถส่งออกไปยังคลิปปอร์ดได้ง่ายหรือสามารถนำเข้าสู่ Node-RED หรือแชร์ทางออนไลน์ได้ [4]

หินอ่อน หรือวัตถุอื่น ๆ เช่น รั้วเหล็ก หลังคาสังกะสี รูปปั้น นอกจากฝุ่นละอองจะทำให้เกิดอาการระคายเคืองตาแล้ว ยังทำอันตรายต่อระบบหายใจเมื่อเราสูดเอาอากาศที่มีฝุ่นละอองเข้าไป โดยอาการระคายเคืองนั้นจะเกิดขึ้นตามส่วนต่าง ๆ ของระบบทางเดินหายใจขึ้นอยู่กับขนาดของฝุ่นละออง โดยฝุ่นที่มีขนาดใหญ่ร่างกายจะดักไว้ได้ที่ขนจมูก ส่วนฝุ่นที่มีขนาดเล็กนั้นสามารถเล็ดลอดเข้าไปในระบบหายใจ ทำให้ระคายเคืองแสบจมูก ไอ จาม มีเสมหะ หรือมีการสะสมของฝุ่นในถุงลมปอด ทำให้การทำงานของปอดเสื่อมลง โดยฝุ่นละอองนั้นจะแบ่งเป็นระดับออกไป เรียกว่า ดัชนีคุณภาพทางอากาศ โดยที่ระดับแรกจะมีดัชนีระดับฝุ่นที่ 0-50 จะอยู่ในเกณฑ์คุณภาพอากาศดี สามารถออกไปทำกิจกรรมต่าง ๆ ได้ ระดับ2จะมีดัชนีระดับฝุ่นที่ 51-100 จะเริ่มอยู่ในกลุ่มเสี่ยงที่ไวเป็นพิเศษต่อมลพิษในอากาศจากอนุภาคฝุ่นละออง ต่อมาในระดับที่3จะมีดัชนีระดับฝุ่นที่ 101-150 จะอยู่ในกลุ่มเสี่ยงหรือกลุ่มที่ไวต่อมลพิษรวมถึงผู้ที่เป็โรคปอดหรือโรคหัวใจ ผู้สูงอายุ เด็กและวัยรุ่น และตั้งแต่ช่วง 151 ขึ้นไป จะอยู่ในทุกกลุ่มช่วงอายุ ควรหลีกเลี่ยงกิจกรรมนอกบ้านทุกอย่าง ย้ายไปทำกิจกรรมภายในอาคารหรือเมื่อคุณภาพอากาศดีขึ้น หรือถ้าจำเป็นต้องออกไป ให้สวมหน้ากากป้องกันทางเดินหายใจที่เหมาะสม เมื่อสวมอย่างถูกต้องแล้ว หน้ากากป้องกันทางเดินหายใจจะสามารถกรองอนุภาคฝุ่นละอองที่สร้างมลพิษในอากาศได้สูงถึงร้อยละ 99 (หน้ากาก N95 กรองได้อย่างน้อยร้อยละ 95 ส่วนหน้ากาก N99 กรองได้ร้อยละ 99) แต่หน้ากากเหล่านี้ จะใช้การได้ดีก็ต่อเมื่อสวมอย่างถูกต้องเท่านั้น ดังนั้น ต้องแน่ใจว่าสวมตามวิธีการที่ระบุไว้และหมั่นตรวจสอบให้หน้ากากระชับกับหน้าอย่าง ซึ่งหน้ากากพวกนี้ไม่มีประสิทธิภาพเลยโดยสิ้นเชิงในการต้านมลพิษในอากาศจากอนุภาคฝุ่นละออง หน้ากาก N95 และ N99 มีจำหน่ายตามร้านค้าส่วนใหญ่ที่ขายผลิตภัณฑ์ปรับปรุงที่อยู่อาศัยและผลิตภัณฑ์เพื่อความปลอดภัย[5]

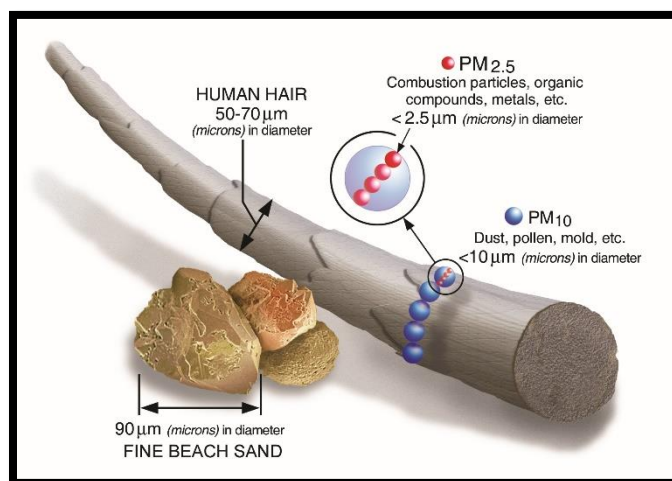


ภาพที่ 2-3 ฝุ่นละอองในเมืองใหญ่

ที่มา: https://www.khaosod.co.th/bbc-thai/news_1986432

2.1.4 PM 2.5

PM 2.5 คือ ฝุ่นละอองขนาดเล็กไม่เกิน 2.5 ไมครอน คือ มลพิษฝุ่นที่มีขนาดเล็กกว่า 1 ใน 25 ส่วนของเส้นผ่าศูนย์กลางของเส้นผมมนุษย์ กล่าวคือ เล็กจนสามารถเล็ดลอดจนจุกเข้าสู่ร่างกายได้และมีขนาดเพียงครึ่งหนึ่งของขนาดเม็ดเลือด (5 ไมครอน) ดังนั้นฝุ่นพิษจึงสามารถเข้าสู่เส้นเลือดฝอยและกระจายไปตามอวัยวะได้ ฝุ่นมีลักษณะที่ขรุขระคล้ายสำลี ฝุ่นเป็นพาหะนำสารอื่นเข้ามาด้วย เช่น แคดเมียม ปรอท โลหะหนัก ไฮโดรคาร์บอน และสารก่อมะเร็ง จำนวนมาก [6]



ภาพที่ 2-4 ฝุ่น PM10 (น้ำเงิน) และฝุ่น PM2.5 (แดง) เทียบกับเส้นผม

ที่มา: <http://www.kstronginsure.com/wp-content/>

2.1.5 ไมโครคอนโทรลเลอร์ DEVIO NB-DEVKIT I

ESP32 เป็นชื่อของไอซีไมโครคอนโทรลเลอร์ที่รองรับการเชื่อมต่อ WiFi ในตัว และ Bluetooth ในขณะนั้น ได้ออกมาปฏิวัติโลกของระบบสมองกลฝังตัว นั่นก็คือไอซีเบอร์ ESP8266 ที่ผลิตโดยบริษัท Espressif จากประเทศจีน ในช่วงเริ่มแรก ไอซี ESP8266 สามารถทำงานได้โดยใช้ การสื่อสารผ่าน UART เท่านั้น และพูดคุยสั่งงานผ่าน AT command ไม่สามารถอัปเดต หรือแก้ไขเฟิร์มแวร์ด้านในได้ แต่ต่อมาไม่นานบริษัท Espressif ก็ได้ออกไอซีเวอร์ชันใหม่มา ในครั้งนี้สามารถที่จะอัปเดตเฟิร์มแวร์ได้ และเราสามารถลงไปเขียนเฟิร์มแวร์เองได้ โดยในขณะนั้น การเขียนเฟิร์มแวร์จะใช้ภาษา C เพียงอย่างเดียว และใช้ ESP8266 SDK เป็นชุดซอฟต์แวร์พัฒนา ด้วยความยากของการทำงานภาษา C เพียงอย่างเดียว ทำให้ไม่ได้ ได้รับความนิยมเรื่องการพัฒนาเฟิร์มแวร์เองมากนัก ESP32 ยังมีเซ็นเซอร์ในตัวดังนี้ วงจร

2.1.7 Javascript

จาวาสคริปต์ (JavaScript) เป็นภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต ซึ่งเป็นภาษาสคริปต์เชิงวัตถุ สามารถใช้ร่วมกับภาษา HTML เพื่อการสร้างและพัฒนาเว็บไซต์ ทำให้เว็บไซต์มีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น ซึ่งมีวิธีการทำงานในลักษณะการแปลความและดำเนินงานไปที่ละคำสั่ง เรียกว่า (Object Oriented Programming) ที่มีเป้าหมายในการออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต สำหรับผู้เขียนภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ โดยทำงานร่วมกับภาษา HTML และภาษาจาวา (Java) ได้ทั้งฝั่งไคลเอนต์ (Client) และฝั่งเซิร์ฟเวอร์ (Server) ซึ่งมีลักษณะการเขียนแบบ โปรโตไทป์ (Prototyped-based Programming) ส่วนมากใช้ในหน้าเว็บเพื่อประมวลผลข้อมูลที่ฝั่งของผู้ใช้งาน แต่ก็ยังมีใช้เพื่อเพิ่มเติมความสามารถในการเขียนสคริปต์โดยฝังอยู่ในโปรแกรมอื่นๆ

จาวาสคริปต์ (JavaScript) ถูกพัฒนาขึ้นโดย Netscape Communications Corporation โดยใช้ชื่อว่า Live Script ออกมาพร้อมกับ Netscape Navigator 2.0 เพื่อใช้สร้างเว็บเพจ โดยติดต่อกับเซิร์ฟเวอร์แบบ Live Wire ต่อมาเน็ตสเคปจึงได้ร่วมมือกับ บริษัทซันไมโครซิสเต็มส์ปรับปรุงระบบของบราวเซอร์ เพื่อให้สามารถติดต่อใช้งานกับภาษาจาวาได้ และได้ปรับปรุง LiveScript ใหม่ เมื่อปี 2538 แล้วตั้งชื่อใหม่ว่า JavaScript แล้วตั้งชื่อใหม่ว่า JavaScript ซึ่งสามารถทำให้การสร้างเว็บเพจ มีลูกเล่นต่างๆ มากมาย และยังสามารถโต้ตอบกับผู้ใช้ได้อย่างทันที เช่น การใช้เมาส์คลิก หรือ การกรอกข้อความในฟอร์ม เป็นต้น

เนื่องจาก JavaScript ช่วยให้ผู้พัฒนาสามารถสร้างเว็บเพจได้ตรงกับความต้องการ และมีความน่าสนใจมากขึ้น ประกอบกับเป็นภาษาเปิด ที่ใครก็สามารถนำไปใช้ได้ ดังนั้นจึงได้รับความนิยมเป็นอย่างสูง มีการใช้งานอย่างกว้างขวาง รวมทั้งได้ถูกกำหนดให้เป็นมาตรฐานโดย ECMA การทำงานของ JavaScript จะต้องมีการแปลความคำสั่ง ซึ่งขั้นตอนนี้จะถูกจัดการโดยบราวเซอร์ (เรียกว่าเป็น client-side script) ดังนั้น JavaScript จึงสามารถทำงานได้ เฉพาะบนบราวเซอร์ที่สนับสนุน ซึ่งปัจจุบันบราวเซอร์เกือบทั้งหมดก็สนับสนุน JavaScript แล้ว อย่างไรก็ตาม สิ่งที่ต้องระวังคือ JavaScript มีการพัฒนาเป็นเวอร์ชันใหม่ๆ ออกมาด้วย ดังนั้นถ้านำโค้ดของเวอร์ชันใหม่ ไปรันบนบราวเซอร์รุ่นเก่าที่ยังไม่สนับสนุน ก็อาจจะทำให้เกิด error ได้ [9]

```
function onload(){
  var xlat;
  var xlng;
  client = new Paho.MQTT.Client("mqtt.netpie.io", 443, "15467b07-e274-410b-93e4-a34b22838b06");
  var options = {
    useSSL: true,
    userName: "3RNfq1Ry4pJw4V2Kv8EVLuwoDDPz1hb",
    password: "",
    onSuccess: onConnect,
    onFailure: doFail
  }
  client.connect(options);
}
```

ภาพที่ 2-7 โครงสร้างภาษา Java

2.1.8 ภาษา Arduino

ภาษาของ Arduino จะอ้างอิงตามภาษา C/C++ จึงอาจกล่าวได้ว่าการเขียนโปรแกรมสำหรับ Arduino (ซึ่งก็รวมถึง บอร์ด Arduino) ก็คือการเขียนโปรแกรมภาษา C โดยเรียกใช้ฟังก์ชันและไลบรารีที่ทาง Arduino ได้เตรียมไว้ให้แล้ว ซึ่งสะดวกและทำให้ผู้ที่ไม่มีความรู้ด้านไมโครคอนโทรลเลอร์อย่างลึกซึ้งสามารถเขียนโปรแกรมสั่งงานได้

โครงสร้างโปรแกรมของ Arduino แบ่งได้เป็นสองส่วนคือ void setup() และ void loop() โดยฟังก์ชัน setup() เมื่อโปรแกรมทำงานจะทำคำสั่งของฟังก์ชันนี้เพียงครั้งเดียว ใช้ในการกำหนดค่าเริ่มต้นของการทำงานส่วนฟังก์ชัน loop() เป็นส่วนทำงานโปรแกรมจะทำคำสั่งในฟังก์ชันนั้นต่อเนื่องกันตลอดเวลา โดยปกติใช้กำหนดโหมดการทำงานของขาต่างๆ กำหนดการสื่อสารแบบอนุกรม ฯลฯ ส่วนของ loop() เป็นโค้ดโปรแกรมที่ทำงาน เช่น อ่านค่าอินพุต ประมวลผล ส่งงานเอาต์พุต ฯลฯ โดยส่วนกำหนดค่าเริ่มต้น เช่น ตัวแปรจะต้องเขียนที่ส่วนหัวของโปรแกรมก่อนถึงตัวฟังก์ชัน นอกจากนั้นยังต้องคำนึงถึง ตัวพิมพ์ เล็ก-ใหญ่ ของตัวแปรและชื่อฟังก์ชันนั้นให้ถูกต้อง[10]

```
#include <Arduino.h>
#include <Stream.h>
#include <TinyGPS++.h>
#include <HardwareSerial.h>
#include <sps30.h>

#include "DHT.h"
#define DHTPIN 25
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
// The TinyGPS++ object
TinyGPSPlus gps;

// The serial connection to the GPS device
HardwareSerial ss(2);
HardwareSerial pma(1); // Rx, Tx

#define TX2_pin 17
#define RX2_pin 16

#define RXPin (14)
#define TXPin (13)
```

ภาพที่ 2-8 โครงสร้างภาษา Arduino (C/C++)

2.1.9 Internet of Things

Internet of Things หมายถึง การที่อุปกรณ์ต่าง ๆ สิ่งต่าง ๆ ได้ถูกเชื่อมโยงทุกสิ่งทุกอย่างสู่โลกอินเทอร์เน็ต ทำให้มนุษย์สามารถสั่งการควบคุมการใช้งานอุปกรณ์ต่าง ๆ ผ่านทางเครือข่ายอินเทอร์เน็ต เช่น การเปิด-ปิด อุปกรณ์เครื่องใช้ไฟฟ้า การสั่งการเปิดไฟฟ้าภายในบ้านด้วยการเชื่อมต่ออุปกรณ์ควบคุมเช่น มือถือผ่านทางอินเทอร์เน็ต ผ่านเครือข่ายอินเทอร์เน็ต IoT มีชื่อเรียกอีกอย่างว่า M2M ย่อมาจาก Machine to Machine คือเทคโนโลยีอินเทอร์เน็ตที่เชื่อมต่ออุปกรณ์กับเครื่องมือต่าง ๆ เข้าไว้ด้วยกันเทคโนโลยี IoT มีความจำเป็นต้องทำงานร่วมกับอุปกรณ์ประเภท RFID และ Sensors ซึ่งเปรียบเสมือนการเติมสมองให้กับอุปกรณ์ต่าง ๆ ที่ขาดไม่คือการเชื่อมต่ออินเทอร์เน็ตเพื่อให้อุปกรณ์สามารถรับส่งข้อมูลถึงกันได้ เทคโนโลยี IoT ประโยชน์ในหลายด้าน แต่ก็มาพร้อมกับความเสี่ยง เพราะหากระบบรักษาความปลอดภัยของอุปกรณ์ และเครือข่ายอินเทอร์เน็ตไม่ดีพอ ก็อาจทำให้มีผู้ไม่ประสงค์ดีเข้ามาขโมยข้อมูลหรือละเมิดความเป็นส่วนตัวของเราได้ [11]



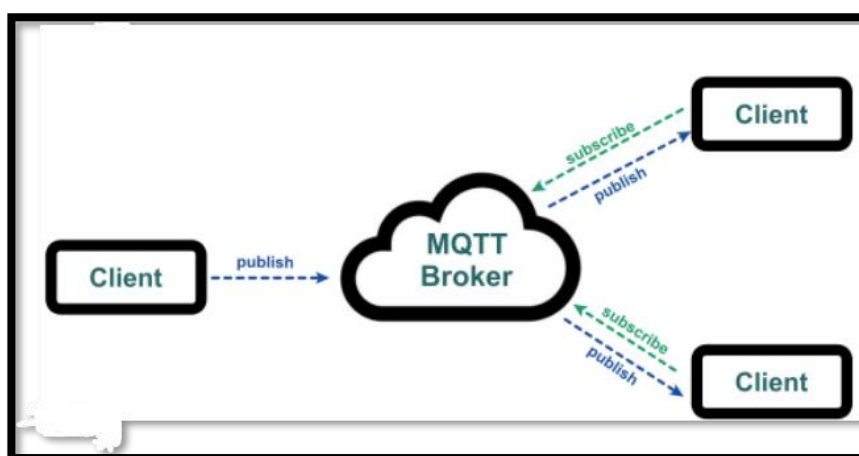
ภาพที่ 2-9 Internet of Things (IoT)

ที่มา: <https://www.ops.go.th/main/images/2561/ICTC/Untitled-1.png>

2.1.10 MQTT Protocol

MQTT สำหรับระบบ IoT นั้น การติดต่อสู่ Internet นั้นเป็นสิ่งสำคัญอย่างยิ่ง เพราะอินเทอร์เน็ตทำให้อุปกรณ์ IoT ต่างๆ สามารถติดต่อสื่อสารและแลกเปลี่ยนข้อมูลกันได้ MQTT (Message Queue Telemetry Transport) ซึ่งพัฒนาต่อมาจาก TCP/IP อีกทีนั้นได้กลายเป็น protocol มาตรฐานสำหรับระบบ IoT และเนื่องจากมันสร้างมาจาก TCP/IP นั้นทำให้ MQTT ประกันว่าข้อมูลที่ส่งกันระหว่างอุปกรณ์ IoT นั้นจะไม่มีสูญหายระหว่างการรับส่งข้อมูล

MQTT ใช้โมเดล publish/subscribe และออกแบบมาเพื่ออุปกรณ์ที่มีความเร็วในการรับและส่งข้อมูลต่ำ (low bandwidth) ซึ่งส่วนมากแล้วอุปกรณ์ของระบบ IoT จะเป็นแบบนั้น จุดประสงค์ของ MQTT ก็คือเพื่อให้ระบบของเรานั้นมีการส่งหรือรับข้อมูลที่มีประสิทธิภาพมากขึ้น รวมทั้งทำให้อุปกรณ์นั้นใช้พลังงานน้อยลง ซึ่งในระบบ IoT ต้องการส่งข้อมูลแบบ real-time และไม่ต้องการให้อุปกรณ์ใช้พลังงานเยอะเกินไปโดยไม่จำเป็น ดังนั้น MQTT จึงเหมาะสมกับระบบเหล่านี้ [12]

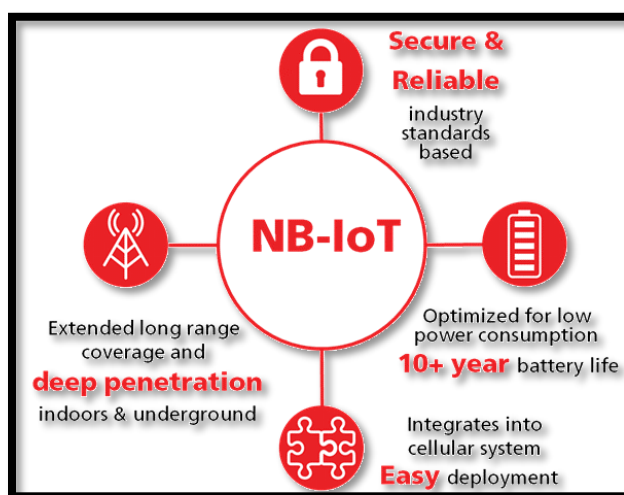


ภาพที่ 2-10 การส่งผ่านข้อมูลผ่านระบบ MQTT

ที่มา: <https://1sheeld.com/mqtt-protocol/pure-javascript-mqtt-broker/>

2.4.3.11 NB-IoT

NB-IoT หรือ Narrow Band Internet of Things เป็นเทคโนโลยี IoT ถูกพัฒนามาเพื่อให้อุปกรณ์ต่าง ๆ สามารถเชื่อมต่อเข้าหากันได้โดยผ่านโครงข่ายของสัญญาณโทรศัพท์เคลื่อนที่ที่ทำให้อุปกรณ์ต่าง ๆ เชื่อมต่อเข้าหากันได้โดยใช้พลังงานที่ต่ำและข้อมูลที่ส่งหากันก็ไม่ต้องเยอะ แม้จะอยู่ในจุดที่ไกลกันก็ยังคุยกันได้โดยไม่ต้องเชื่อมต่อผ่านสายให้ยุ่งยาก เพียงแค่มีซิม (SIM) ติดตั้งในอุปกรณ์นั้น ๆ ก็ทำให้เชื่อมต่อและคุยกันได้ โดย NB-IoTถูกพัฒนาในมาตรฐาน 3GPP Rel.13 ทั้งนี้ 3GPP ที่ออกแบบให้รับส่งข้อมูลได้ถึง 100 kbps ทั้ง Uplink และ Downlink โดยใช้ช่องสัญญาณแคบขนาดแบนด์วิธ 200 kHz จุดเด่นของเทคโนโลยี IoT นี้คือใช้พลังงานต่ำมาก แบตเตอรี่สามารถอยู่ได้นานถึง 10 ปี เหมาะกับการใช้งานกับอุปกรณ์ IoT ที่ส่งข้อมูลขนาดเล็ก ไม่บ่อยมากนัก และไม่มีการเคลื่อนที่ (stationary sensor) ตัวอย่างการนำไปใช้งานมีมากมาย เช่น อุปกรณ์วัดสภาพอากาศและสิ่งแวดล้อมในพื้นที่การเกษตรกับพื้นที่เมือง อุปกรณ์ Smart Parking ที่ตรวจสอบว่าช่องจอดรถแต่ละช่องว่างอยู่หรือไม่ รวมทั้งอุปกรณ์ sensor ในพื้นที่อุตสาหกรรม (ที่เรียกว่า IIoT – Industrial IoT) เพื่อตรวจวัดการใช้งานของอุปกรณ์เครื่องจักรในโรงงานอุตสาหกรรม และมีเตอร์วัดการใช้น้ำและไฟฟ้า ก็เหมาะสมกับการใช้ NB-IoT ได้เช่นกัน [13]



ภาพที่ 2-11 การส่งผ่านข้อมูลผ่านระบบ NB-IoT

ที่มา: <https://www.iphonemod.net/gsma-truemove-h-nb-iot-thailand.html>.

2.4.3.12 MongoDB

MongoDB หมายถึง open-source document database ประเภทหนึ่ง โดยเป็น database แบบ NoSQL Database จะไม่มีการใช้คำสั่ง SQL ไม่นั่นในการสร้างความสัมพันธ์ของข้อมูล แต่จะเป็นรูปแบบโครงสร้างที่เจ้าของ NoSQL สร้างขึ้นมาเองและจัดเก็บข้อมูลเป็นแบบ JSON (JavaScript Object Notation) ซึ่งจะเก็บค่าเป็น key และ value โดยจุดเด่นอยู่ที่ความเร็วในการทำงานเป็นหลัก query ข้อมูลได้เร็วขึ้น การทำงานในส่วนของ database จะลดลง แต่จะไปเน้นการทำงานในส่วนของโปรแกรมที่พัฒนาขึ้นมาแทน โดย database ประเภทนี้ จะเหมาะกับข้อมูลขนาดใหญ่ที่ไม่ซับซ้อน การทำงานที่ไม่หนักมาก สามารถทำงานกับระบบที่เป็นการทำงานแบบเรียลไทม์ (Real Time) ได้ดี[14]

ภาพที่ 2-12 การเก็บข้อมูลผ่านระบบ NB-IoT

ที่มา: <https://sysadmin.psu.ac.th/2017/01/11/what-is-mongodb/>.

2.4.3.13 Solar cell

เซลล์แสงอาทิตย์ (Solar Cell) เป็นสิ่งประดิษฐ์กรรมทางอิเล็กทรอนิกส์ ที่สร้างขึ้นเพื่อเป็นอุปกรณ์สำหรับเปลี่ยนพลังงานแสงอาทิตย์ให้เป็นพลังงานไฟฟ้า โดยการนำสารกึ่งตัวนำ เช่น ซิลิกอน ซึ่งมีราคาถูกที่สุดและมีมากที่สุดบนพื้นโลกมาผ่านกระบวนการทางวิทยาศาสตร์เพื่อผลิตให้เป็นแผ่นบางบริสุทธิ์ และทันทีที่แสงตกกระทบบนแผ่นเซลล์ รังสีของแสงที่มีอนุภาคของพลังงานประกอบที่เรียกว่า โฟตอน (Photon) จะถ่ายเทพลังงานให้กับอิเล็กตรอน (Electron) ในสารกึ่งตัวนำจนมีพลังงานมากพอที่จะกระโดดออกมาจากแรงดึงดูดของอะตอม (atom) และเคลื่อนที่ได้อย่างอิสระ ดังนั้นเมื่ออิเล็กตรอนเคลื่อนที่ครบวงจรจะทำให้เกิดไฟฟ้ากระแสตรงขึ้น เมื่อพิจารณาลักษณะการผลิตไฟฟ้าจากเซลล์แสงอาทิตย์พบว่า เซลล์แสงอาทิตย์จะมีประสิทธิภาพการผลิตไฟฟ้าสูงสุดในช่วงเวลากลางวัน ซึ่งสอดคล้องและเหมาะสมในการนำเซลล์แสงอาทิตย์มาผลิตไฟฟ้า เพื่อแก้ไขปัญหาการขาดแคลนพลังงานไฟฟ้าในช่วงเวลากลางวัน[15]



ภาพที่ 2-13 Solar cell

ที่มา: <http://www.ccsolar-thai.com/>

2.4.3.14 Battery

แบตเตอรี่ (Battery) คือ อุปกรณ์ที่ทำหน้าที่จัดเก็บพลังงานเพื่อไว้ใช้ต่อไป ถือเป็นอุปกรณ์ที่สามารถแปลงพลังงานเคมีให้เป็นไฟฟ้าได้โดยตรงด้วยการใช้เซลล์กัลวานิก (galvanic cell) ที่ประกอบด้วยขั้วบวกและขั้วลบ พร้อมกับสารละลายอิเล็กโทรไลต์ (electrolyte solution) แบตเตอรี่อาจประกอบด้วยเซลล์กัลวานิกเพียง 1 เซลล์ หรือมากกว่าก็ได้ แบตเตอรี่เป็นอุปกรณ์สำหรับจัดเก็บไฟฟ้าเท่านั้น ไม่ได้ผลิตไฟฟ้า สามารถประจุไฟฟ้าเข้าไปใหม่ (recharge) ได้หลายครั้งและประสิทธิภาพจะไม่เต็ม

100% จะอยู่ที่ประมาณ 80% เพราะมีการสูญเสียพลังงานบางส่วนไปในรูปความร้อนและปฏิกิริยาเคมีจากการประจุ/จ่ายประจุนั่นเอง แบตเตอรี่จัดเป็นอุปกรณ์ที่มีราคาแพง และเสียหายได้ง่าย หากดูแลรักษาไม่ดีเพียงพอหรือใช้งานผิดวิธี รวมถึง อายุการใช้งานของแบตเตอรี่แต่ละชนิดจะแตกต่างกันไป เนื่องด้วยวิธีการใช้, การบำรุงรักษา, การประจุและอุณหภูมิ โดย ความจุของแบตเตอรี่ในการบรรจุพลังงานมีหน่วยเป็น แอมแปร์-ชั่วโมง (Ampere-Hour; Ah) พลังงานในแบตเตอรี่ 12 V 100 Ah เท่ากับ $12V \times 100Ah$ หรือ $12V \times 100A \times 3600s$ จะได้เท่ากับ 4.32 MJ ถ้าแบตเตอรี่ 100 Ah เท่ากับว่าแบตเตอรี่จะจ่ายกระแส 1 แอมแปร์อย่างต่อเนื่องเป็นเวลา 100 ชั่วโมง หรือ แบตเตอรี่จ่ายกระแส 10 แอมแปร์อย่างต่อเนื่องเป็นเวลา 10 ชั่วโมง เช่นเดียวกับแบตเตอรี่จ่ายกระแส 5 แอมแปร์อย่างต่อเนื่องเป็นเวลา 20 ชั่วโมง ซึ่งทั้งหมดนี้จ่าย กระแสเท่ากับ 100 Ah ทั้งสิ้น จะเห็นได้ว่า แบตเตอรี่ที่มีความจุเท่ากันอาจมีความเร็วในการจ่ายกระแสต่างกันได้ ดังนั้น การจะทราบความจุของแบตเตอรี่ต้องทราบถึง อัตราการจ่ายกระแสด้วย จะกำหนดเป็นจำนวนชั่วโมงของการจ่ายกระแสเต็มที่การกำหนดขนาดของแบตเตอรี่สำหรับระบบเซลล์แสงอาทิตย์นั้น ขึ้นอยู่กับความจุของ แบตเตอรี่ในการจัดเก็บพลังงาน, อัตราการจ่ายประจุสูงสุด, อัตราการประจุสูงสุดและอุณหภูมิต่ำสุดที่จะนำแบตเตอรี่ไปใช้งาน (อุณหภูมิที่ได้ผลดีที่สุดของแบตเตอรี่ตะกั่ว-กรดคือ 77 F หรือประมาณ 60-80 F) [16]



ภาพที่ 2-14 Battery

ที่มา: <http://www.diy-solarcell.com/>

2.4.3.14 เซ็นเซอร์ GPS Module GY-NEO6MV2

เป็นอุปกรณ์ที่ทำหน้ารับสัญญาณ พิกัด ที่ส่งจากดาวเทียม GPS แล้วคำนวณข้อมูลพิกัดเหล่านั้นออกมา ใ้รับสัญญาณของฮาร์ดแวร์ ที่สามารถเพิ่มเข้ากับชิ้นส่วนอื่น ๆ ของฮาร์ดแวร์ต่าง ๆ เพื่อให้สามารถรับข้อมูลจากดาวเทียม GPS ได้[17]



ภาพที่ 2-15 GPS Module GY-NEO6MV2

ที่มา: <https://www.cartrack.co.th/gps-module>

2.2 งานวิจัยที่เกี่ยวข้อง

Chawis Boonme, Siwach Chamchanya (2563) งานวิจัยนี้มีวัตถุประสงค์เพื่อนำเสนอการเลือกพื้นที่พักพิงชั่วคราวในช่วงฝุ่นละอองขนาดเล็กที่มีขนาดเส้นผ่านศูนย์กลาง 2.5 ไมครอน (PM 2.5) ภัยพิบัติมลพิษทางอากาศผ่านกระบวนการวิเคราะห์ลำดับชั้น (AHP) สำหรับมหาวิทยาลัยเชียงใหม่ กรณีศึกษา ประการแรกเกณฑ์หลักและเกณฑ์ย่อยสำหรับการประเมินทางเลือกถูกกำหนดโดยการสัมภาษณ์ผู้เชี่ยวชาญที่มีประสบการณ์เกี่ยวกับการก่อสร้างที่พักพิง PM 2.5 จากนั้นพื้นที่ที่เหมาะสมสำหรับสถานที่พักพิงชั่วคราวคือ ระบุโดยการแจกจ่ายแบบสอบถาม 617 ชุดเพื่อแสวงหาพื้นที่ที่เหมาะสมตามความต้องการใช้บริการ หลังจากการสำรวจพื้นที่ที่เหมาะสมผู้สมัครที่พักพิงชั่วคราวสามคนได้รับการ

พิจารณาตามเกณฑ์จากการสัมภาษณ์ของผู้เชี่ยวชาญและรวบรวมข้อมูลเบื้องต้นของผู้สมัครสามคน ถัดไป AHP ถูกนำไปใช้ในการเลือกสถานที่พักพิงชั่วคราว ในที่สุดผลการวิจัยได้วิเคราะห์และเสนอแนวทางเบื้องต้นของศูนย์พักพิงชั่วคราวที่ได้คะแนนสูงสุดในการสร้างศูนย์พักพิงชั่วคราว ผลการวิจัยพบว่าเกณฑ์หลักประกอบด้วยฟังก์ชันหาค่าใช้จ่ายการเข้าถึงลักษณะห้องและสาธารณูปโภคตามลำดับ สถานที่พักพิงชั่วคราวที่เหมาะสมที่สุดโดยอิงจากผู้สมัคร 3 คนคือห้องบนชั้น 4 ห้องสมุดหลักของมข. เนื่องจากการประยุกต์ใช้ AHP เราพบว่าเทคนิคสามารถช่วยผมอำนาจตัดสินใจในการเลือกที่พักพิงชั่วคราวที่เหมาะสมภายใต้เกณฑ์ที่กำหนดและวิธีการวิจัยสามารถนำไปใช้กับกรณีอื่น ๆ สำหรับการเลือกสถานที่ตั้ง

Petcharat Limsupreeyarat (2563) กิจกรรมการก่อสร้างรวมถึงการซ่อมแซมและปรับปรุงเป็นงานที่ก่อให้เกิดฝุ่นซึ่งฝุ่นละอองเป็นมลพิษทางอากาศที่ส่งผลกระทบต่อสุขภาพของมนุษย์เมื่อฝุ่นละอองเข้าสู่ระบบทางเดินหายใจทำให้เกิดการระคายเคืองและทำลายเนื้อเยื่อของอวัยวะนั้น ๆ เช่น เนื้อเยื่อปอด งานวิจัยนี้จึงพัฒนาเครื่องตรวจจับฝุ่นส่วนบุคคลสำหรับคนงานก่อสร้างหรือปรับปรุงอาคารซึ่งอุปกรณ์ที่พัฒนาต้องเหมาะสมกับการใช้งานราคาถูกกว่าอุปกรณ์ที่ขายทั่วไปและอุปกรณ์สามารถแจ้งเตือนผู้ปฏิบัติงานให้ออกจากบริเวณที่มีฝุ่นละอองเกินมาตรฐานได้ มูลค่า. งานวิจัยเริ่มต้นด้วยการศึกษาอุปกรณ์ตรวจจับฝุ่นและฝุ่นในปัจจุบันจากนั้นออกแบบและพัฒนาเครื่องตรวจจับฝุ่นส่วนบุคคลสำหรับกิจกรรมการก่อสร้างโดยพิจารณาจากความแข็งแรงและราคาของอุปกรณ์เพื่อทดสอบอุปกรณ์ที่พัฒนาขึ้นเมื่อเทียบกบอุปกรณ์สำรวจฝุ่นมาตรฐาน ตามที่กรมควบคุมมลพิษมีค่าฝุ่นละอองเฉลี่ย 24 ชั่วโมงต้องไม่เกิน 50 ไมโครกรัมต่อลูกบาศก์เมตร ผลการวิจัยชี้ให้เห็นว่าอุปกรณ์ต้นแบบที่พัฒนาขึ้นเพื่อวัดค่าฝุ่นละอองเมื่อเทียบกับอุปกรณ์วัดฝุ่นมาตรฐานและสามารถแจ้งเตือนผู้ปฏิบัติงานซึ่งสามารถป้องกันคนงานก่อสร้างหรือปรับปรุงการก่อสร้างจากฝุ่นละอองที่เกินค่ามาตรฐานในไซต์งานก่อสร้างและมีราคาถูกกว่าอุปกรณ์ที่ซื้อที่ขายทั่วไป

Nuanrath Wattana (2563) งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาฝุ่นในระบบวัดอากาศบนอุปกรณ์สมาร์ทโฟน เนื่องจากความกังวลของเราเกี่ยวกับปัญหามลพิษทางอากาศจากควันมาจากประเทศอินโดนีเซียซึ่งส่งผลกระทบต่อสุขภาพของประชาชนในจังหวัดตรังในเดือนมีนาคม 2562 ดังนั้นเราจึงจำเป็นต้องพัฒนาระบบนี้โดยใช้แนวคิดของวงจรชีวิตการพัฒนาระบบฐานข้อมูล การออกแบบและอุปกรณ์ไมโครคอนโทรลเลอร์สำหรับการพัฒนาระบบ สำหรับเครื่องมือในการวิจัยคืออุปกรณ์เซ็นเซอร์ฝุ่นที่ติดตั้งระบบที่มหาวิทยาลัยสวนดุสิตศูนย์ตรังและนำเสนอค่าฝุ่นละอองบนแอปพลิเคชันมือถือ Dust @

SDU บนอุปกรณ์สมาร์ทโฟน ผลการศึกษาพบว่าค่า PM2.5 จากการตรวจวัดระบบฝุ่นใน 24 ชั่วโมงมีค่าใกล้เคียงกับกรมควบคุมมลพิษโดยตั้งแต่วันที่ 20-22 กันยายน 2562 ที่ตำบลบ้านวนอำเภอตรังจังหวัดตรังว่าค่าจากการตรวจวัดระบบฝุ่น มีมูลค่าสูงกว่า

2.3 บทสรุป

จากการศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง เป็นการค้นคว้าหาข้อมูลทฤษฎีที่เกี่ยวข้องและเอกสารที่เกี่ยวข้อง เพื่อนำมาศึกษาความรู้ที่จะเป็นไปได้ของโครงการ และเป็นการศึกษาโปรแกรมต่าง ๆ ที่สามารถนำมาประยุกต์ใช้ให้เกิดระบบติดตามและประเมินผลการนิเทศนักศึกษาขึ้น เช่นภาษาที่ใช้ในการจัดทำโครงการ รวมถึงเครื่องมือต่าง ๆ มาประยุกต์ใช้ในการพัฒนาโครงการพิเศษในครั้งนี้ เพื่อให้มีประสิทธิภาพสูงสุด

บทที่ 3

วิธีการดำเนินการ

ในบทนี้จะเป็นการนำความรู้จากการศึกษาค้นคว้าในบทก่อนหน้ามาประยุกต์ใช้ในการตัดสินใจเลือกใช้อุปกรณ์ต่าง ๆ และแพลตฟอร์มที่ใช้ในการเก็บข้อมูล รวมถึงการออกแบบหน้าเว็บแสดงผล ให้เหมาะกับการใช้งาน มีกระบวนการและขั้นตอนการดำเนินงานดังนี้

- 3.1 กำหนดปัญหา
- 3.2 การรวบรวมข้อมูล
- 3.3 การวิเคราะห์
- 3.4 การออกแบบระบบ
- 3.5 การพัฒนา
- 3.6 การทดสอบระบบ
- 3.7 บทสรุป

3.1 กำหนดปัญหา

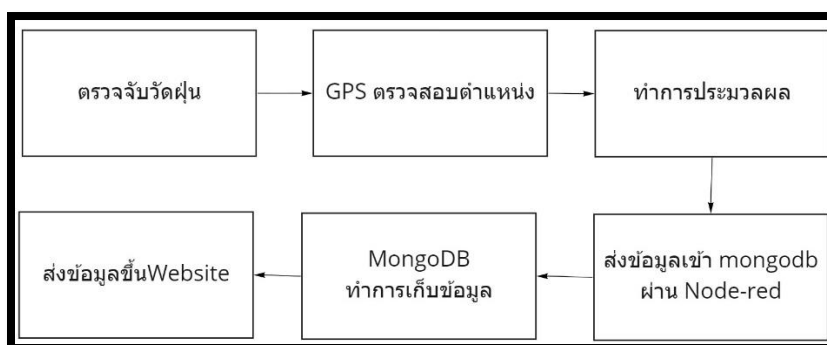
อุปกรณ์วัดฝุ่นเคลื่อนที่ ที่มีในปัจจุบันยังทำงานไม่เป็นที่น่าพอใจเมื่อเทียบกับเทคโนโลยีในปัจจุบัน จากปัญหาดังที่กล่าวมาข้างต้น ซึ่งหน้าเว็บยังแสดงข้อมูลได้ยังไม่ครบ และยังไม่สมบูรณ์นัก ในส่วนของระบบจัดเก็บข้อมูล ของเดิมเป็นการใช้ MySQL ซึ่งเป็นระบบที่ไม่เหมาะสมกับการทำงานในด้านนี้ และในส่วน of ตัวอุปกรณ์ เซ็นเซอร์ที่ใช้อยู่ในปัจจุบันเป็นรุ่นเก่าซึ่งอาจเกิดค่าผิดพลาด และโครงสร้างไม่มีการกันลม หรือปัจจัยรบกวนอื่นๆ เช่น ฝน จึงได้มีแนวคิดที่จะนำอุปกรณ์วัดฝุ่นเคลื่อนที่มาพัฒนาต่อ โดยนำอุปกรณ์วัดฝุ่นเคลื่อนที่มาพัฒนาด้านโครงสร้าง โดยเปลี่ยนตำแหน่งเซ็นเซอร์จากด้านหน้าเป็นด้านข้างเพื่อเพิ่มความถูกต้องของข้อมูล สิ่งต่อมาที่พัฒนาคือด้านการเก็บข้อมูล ซึ่งการเก็บข้อมูลในปัจจุบันใช้เป็นฐานข้อมูล MySQL ซึ่งผู้จัดทำโครงการพิเศษเห็นว่าเป็นการเก็บข้อมูลที่ไม่เหมาะสมกับการทำงานในด้าน IoT จึงได้มีการเปลี่ยนการเก็บข้อมูลจาก MySQL มาเป็น การเก็บข้อมูลแบบ NOSQL โดยใช้ MongoDB ในการเก็บข้อมูล และใช้ Node-Red ให้ส่งข้อมูลมาเพื่อแสดงผลใน Website ได้แบบ real time

3.2 การรวบรวมข้อมูล

จากการสำรวจในปัจจุบันนั้นมีการใช้ สมาร์ทโฟน และ คอมพิวเตอร์เป็นส่วนใหญ่ เพราะมีความสะดวกในการรับข้อมูลต่าง ๆ และการพัฒนาระบบนี้ขึ้นมาให้สามารถรองรับความต้องการในการใช้งานได้ ปัจจุบัน อุปกรณ์ IoT ถูกพัฒนาให้รองรับและตอบโต้ภัยในการทำงานในชีวิตประจำวันของมนุษย์ ทำให้อุปกรณ์มีราคาถูกลงมาก เมื่อเทียบกับผลลัพธ์ที่ได้จึงเป็นสิ่งที่คุ้มค่าในการใช้งาน

3.3 การวิเคราะห์

อุปกรณ์วัดฝุ่นเคลื่อนที่จะทำการวัดค่าปริมาณฝุ่นและส่งข้อมูลโดยผ่าน node-red ขึ้นไปเก็บไว้บน Mongo DB จากนั้นหน้าเว็บแสดงผลจะดึงข้อมูลข้อมูล มาแสดงหน้าเว็บ โดยภายในตัวอุปกรณ์จะมี DEVIO NB-DEVKIT I เป็นตัวควบคุมการทำงาน ของตัวอุปกรณ์ทั้งหมด และมีเซนเซอร์ต่าง ๆ ตามที่ได้ศึกษาค้นคว้ามาแล้วว่ามีความเหมาะสม ที่จะนำมาใช้งานในการวัดฝุ่น แจ้งตำแหน่ง ทำตามเงื่อนไขการทำงาน ของตัวอุปกรณ์ให้ตรงตามวัตถุประสงค์ที่กำหนดไว้

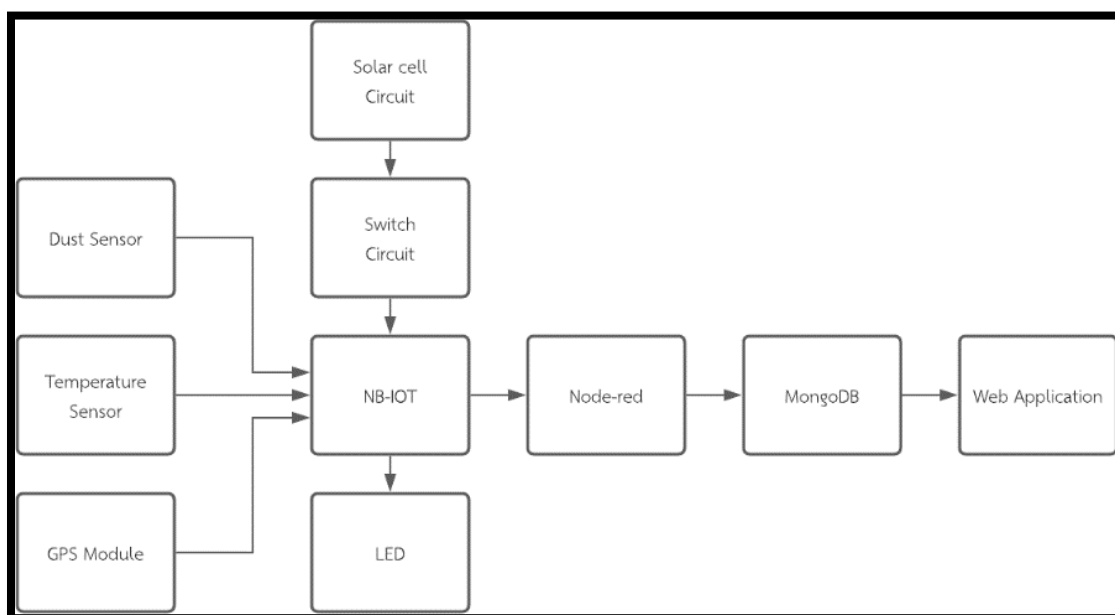


ภาพที่ 3-1 แสดงการทำงานของอุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IoT

จากภาพที่ 3-1 อุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IoT ประกอบไปด้วยส่วน หลัก ๆ ดังนี้

- NB-IoT
- เซนเซอร์ตรวจจับฝุ่น
- DHT22
- GPS
- Battery 5V DC

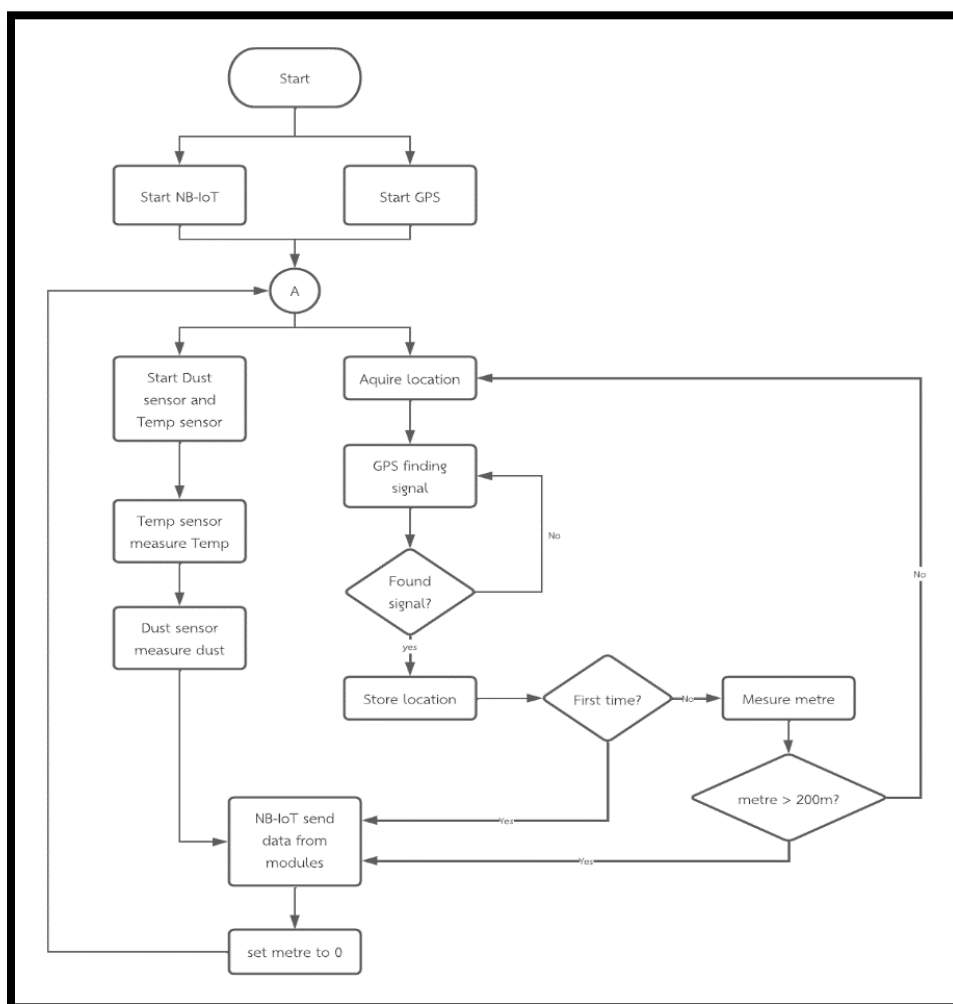
- Solar cell
- วงจรขับแรงดันไฟขึ้น จาก 3.7V ไป 5V
- หลอดไฟ (LED)
- รีเลย์ (Relay)



ภาพที่ 3-2 Block Diagram อุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IoT

จากภาพที่ 3-2 เซนเซอร์วัดฝุ่นและเซนเซอร์วัดอุณหภูมิและ GPS จะส่งข้อมูลไปประมวลผลในส่วนของ Process ESP32 ทำหน้าที่รับข้อมูล และส่งข้อมูลไปยัง MongoDB โดยผ่าน Node-red

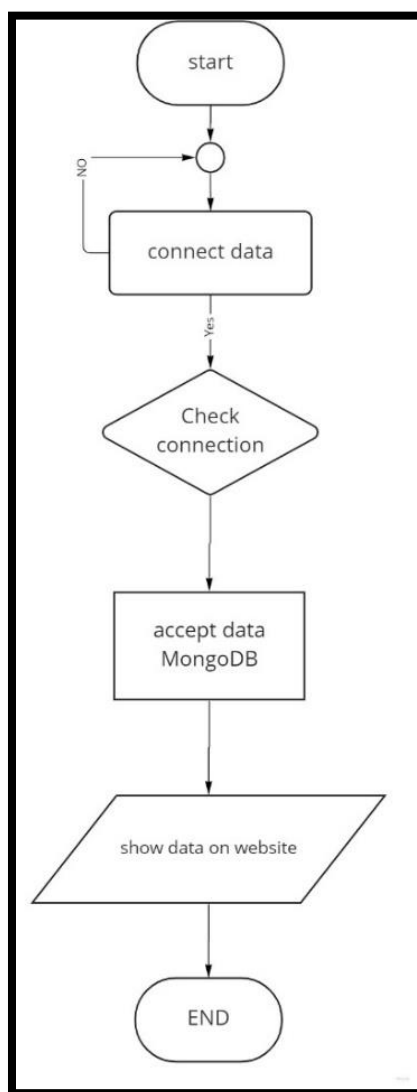
3.2.1 ขั้นตอนการทำงานของ NB-IoT



ภาพที่ 3-3 FlowChart อุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IoT

จากภาพที่ 3-3 ตั้งแต่เปิดเครื่อง MCU จะทำการเชื่อมกับ server และเริ่มใช้งาน GPS จุด A คือ จุด checkpoint สำหรับการวนกลับมาทำซ้ำ จากนั้นเริ่มการทำงานของ sensor วัดฝุ่นและ วัดอุณหภูมิ โดยจะวัดอุณหภูมิก่อนจากนั้นจึงวัดค่าฝุ่นในฝั่งของ GPS จะทำการหาตำแหน่งโดยมีเงื่อนไขคือเมื่อส่งข้อมูล ครั้งแรกจะให้ส่งข้อมูลได้เลยเมื่อข้อมูลพร้อมแต่ถ้าเป็นการส่งข้อมูลครั้งถัดๆ มาจะมีเงื่อนไขคือหาก ตำแหน่งที่ส่งข้อมูลไปครั้งแรก วัดระยะทางกับตำแหน่งในปัจจุบันที่เก็บเป็นตัวแปรได้เกิน 200 เมตร ให้สามารถส่งข้อมูลได้เมื่อข้อมูลพร้อมเมื่อข้อมูลพร้อม MCU จะส่งข้อมูลขึ้นไปยัง server เมื่อข้อมูลสำเร็จให้ กำหนดค่าระยะทางเป็น 0

3.2.2 แผนผังการทำงานของการทำงานการตรวจจับค่าฝุ่นและส่งข้อมูลขึ้น MongoDB และนำมาแสดงที่เว็บไซต์

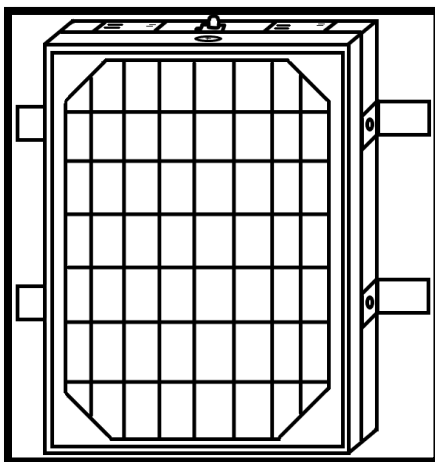


ภาพที่ 3-4 Flowchart การส่งข้อมูลอุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IoT ไปยังเว็บไซต์แสดงผล

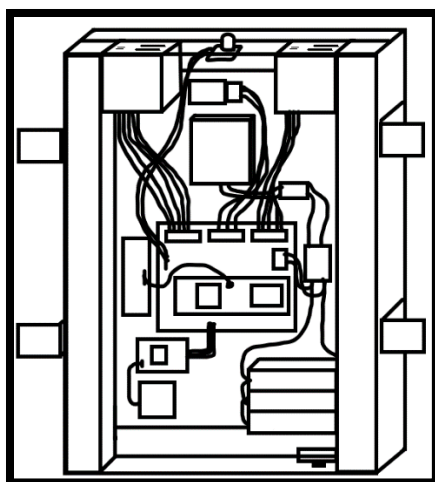
จากภาพที่ 3-4 จะเป็นการส่งข้อมูลจากตัวอุปกรณ์จาก node-red ไปยัง MongoDB และดึงข้อมูลมาแสดงผลบนเว็บไซต์

3.4 การออกแบบระบบ

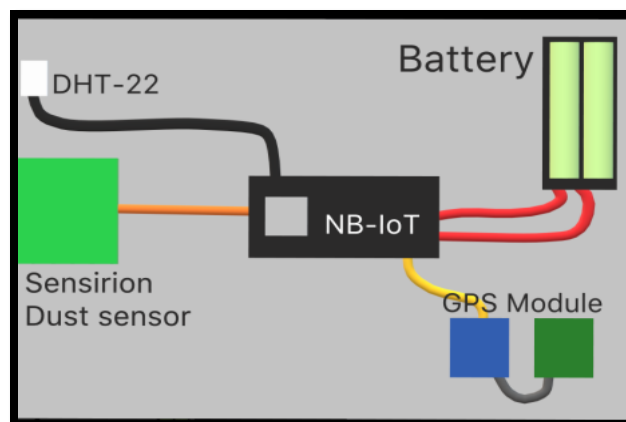
3.4.1 การออกแบบ hardware อุปกรณ์วัดฝุ่นด้วยเทคโนโลยี IoT



ภาพที่ 3-5 แบบจำลองการติดตั้งอุปกรณ์ด้านหน้า

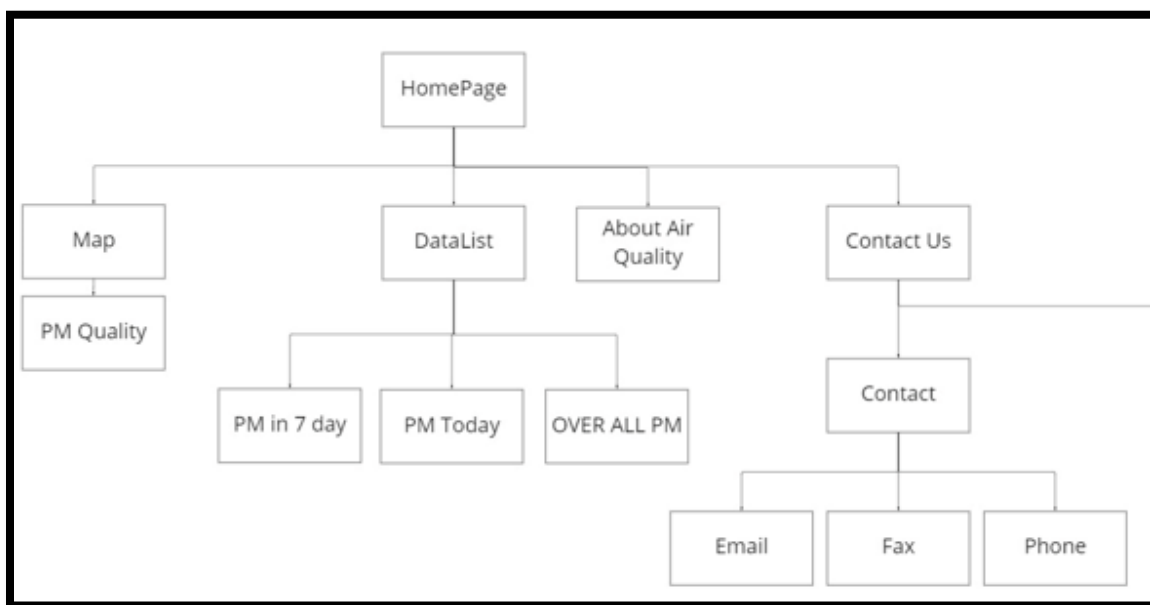


ภาพที่ 3-6 แบบจำลองการติดตั้งอุปกรณ์ด้านหลัง



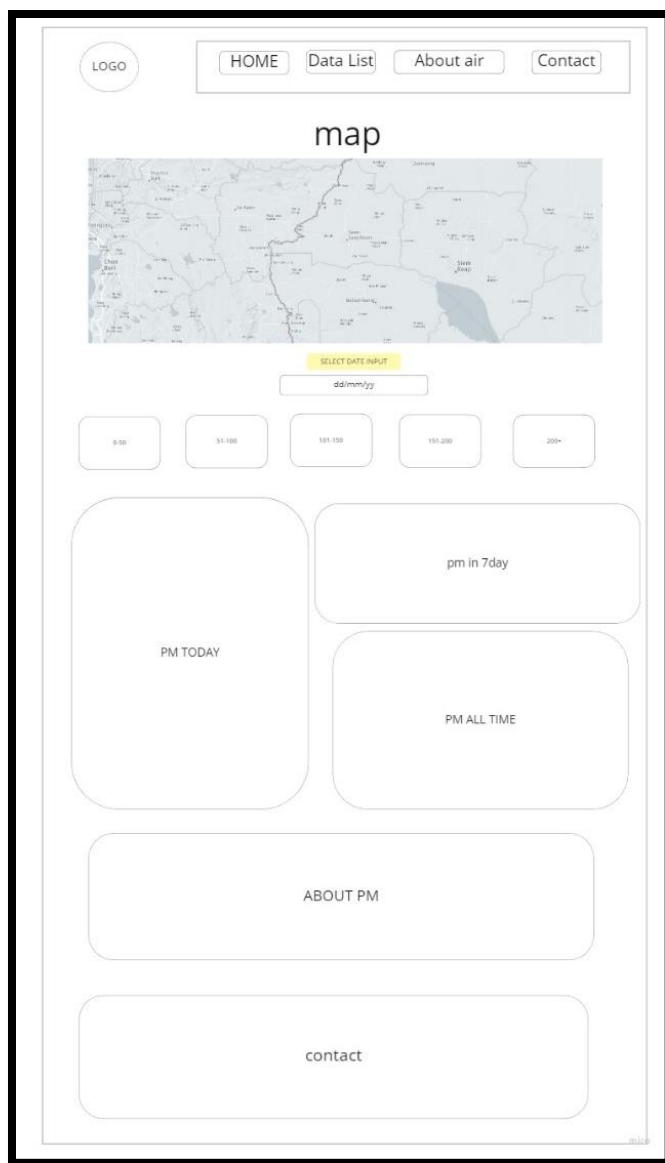
ภาพที่ 3-7 แผนภาพจำลองการติดตั้งอุปกรณ์ภายใน

3.4.2 การออกแบบ website



ภาพที่ 3-8 แผนผังเว็บไซต์

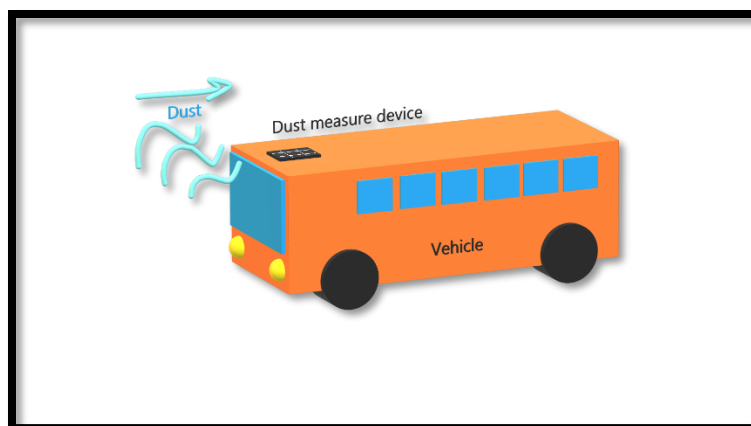
จากภาพที่ 3-8 เป็นการแสดงผลหน้าเว็บไซต์โดยจะมี แผนที่ ในการแสดงปริมาณฝุ่นที่วัดได้บนแผนที่ และมี Data List ในการแสดงฝุ่นประจำวัน ย้อนหลัง 7 วัน และ มีหน้า Contact สำหรับการติดต่อทาง email Fax และ โทรศัพท์



ภาพที่ 3-9 แบบจำลองการออกแบบเว็บไซต์

3.4.3 จำลองการติดตั้งอุปกรณ์

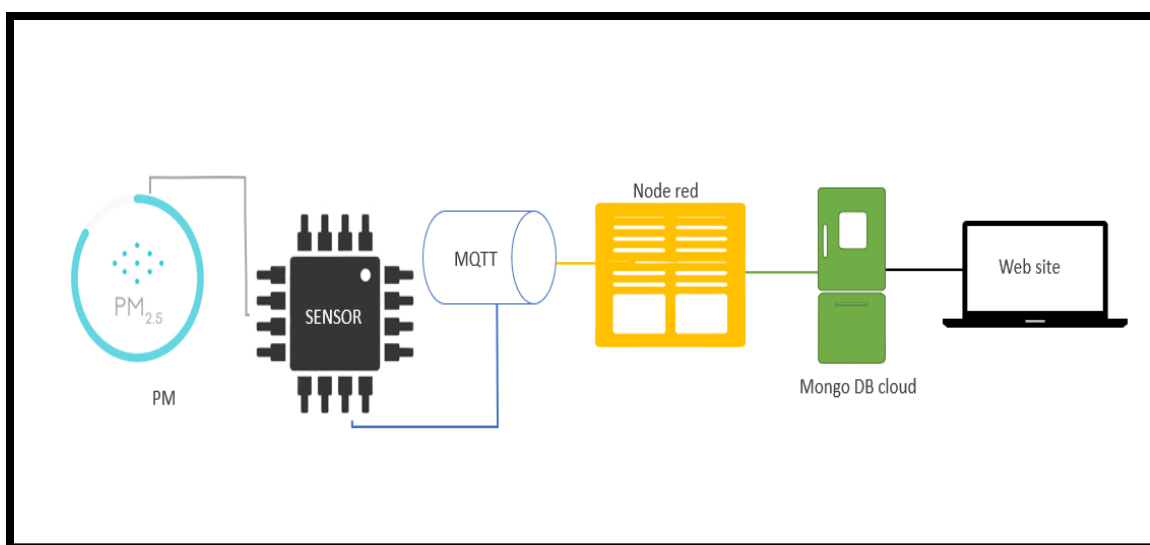
แบบจำลองการติดตั้งของอุปกรณ์ โดยที่จะติดตั้งบนรถโดยสารประจำทางหรือยานพาหนะการ ออกแบบให้เหมาะกับการเคลื่อนที่โดยที่อุปกรณ์ไม่ดำนกับลม อุปกรณ์ติดตั้งแผงโซลาเซลล์ เพื่อชาร์จพลัง ให้กับอุปกรณ์โดยที่ไม่จำเป็นต้องถอดอุปกรณ์มาชาร์จแล้วนำไปติดตั้งตามเดิมเพื่อให้เกิดความเสีเวลา



ภาพที่ 3-10 แผนภาพจำลองการติดตั้งอุปกรณ์

3.4.4 จำลองการทำงานของระบบ

การทำงานของระบบ เริ่มจากเมื่ออุปกรณ์ตรวจจับค่าฝุ่น จะส่งค่าฝุ่นพร้อมทั้งตำแหน่งของอุปกรณ์ที่ไป MongoDB ผ่าน Node-red และจะเก็บค่าทั้งหมดไว้แล้วส่งไปที่เว็บไซต์เพื่อแสดงผล

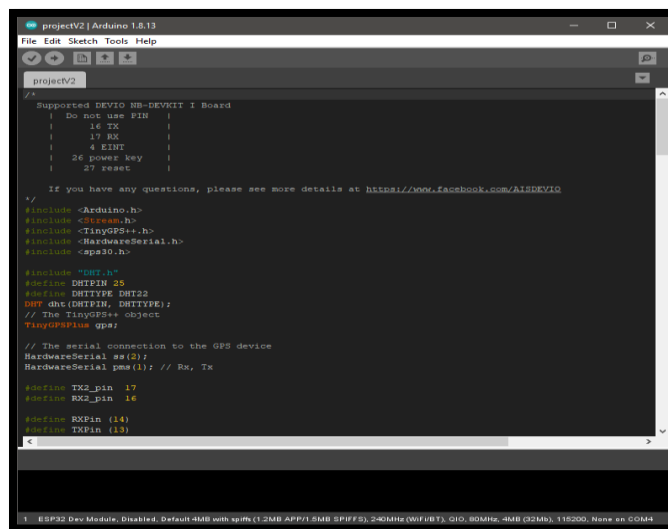


ภาพที่ 3-11 แสดงการทำงานของระบบอุปกรณ์วัดฝุ่นเคลื่อนที่

3.5 การพัฒนา

3.5.1 การพัฒนาระบบ

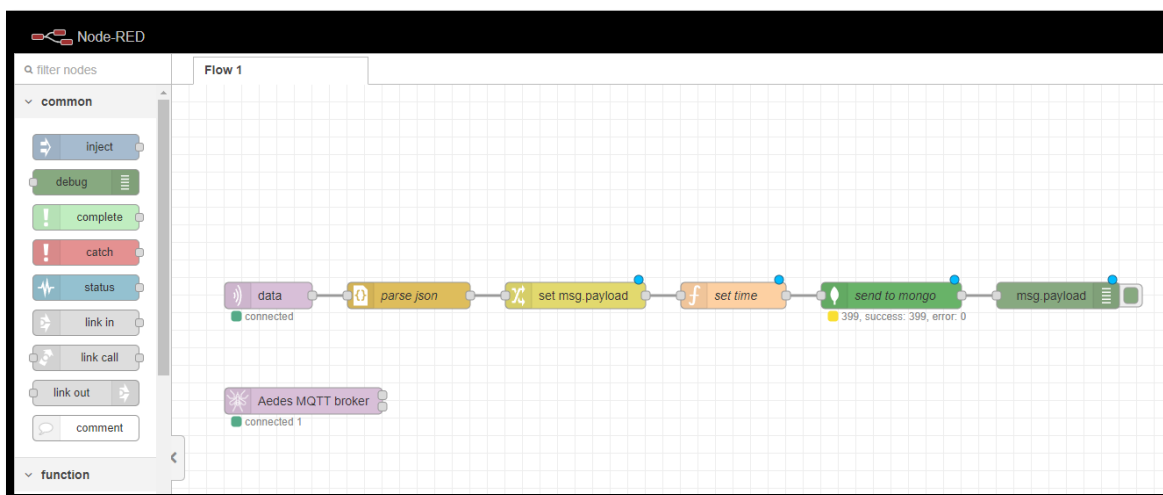
พัฒนาโดยใช้ภาษา Arduino เป็นหลักในการควบคุมอุปกรณ์ให้มีประสิทธิภาพมากที่สุด



ภาพที่ 3-12 โปรแกรม Arduino IDE

3.5.2 การพัฒนาการส่งข้อมูล

พัฒนาโดยใช้ภาษา Node-red เป็นหลักในการส่งข้อมูลไปยัง database

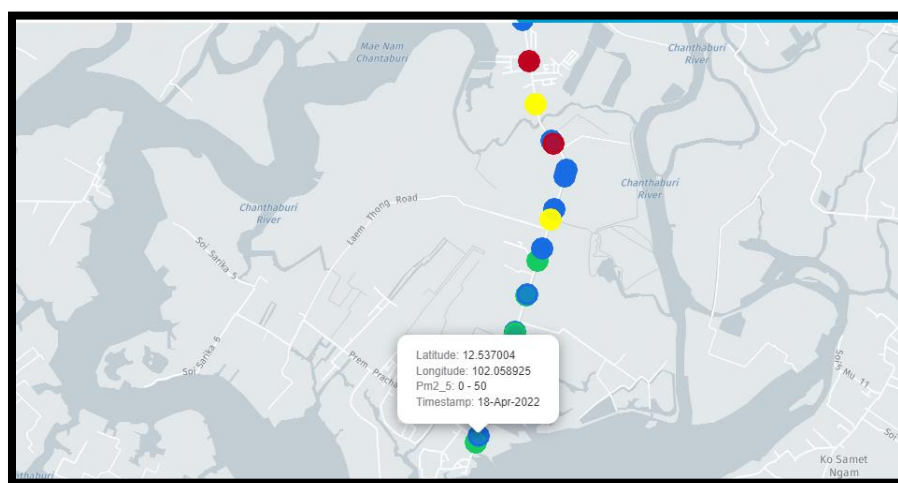


ภาพที่ 3-13 แสดง dataFlow การทำงานของ Node-red

ภาพที่ 3-15 ตัวอย่าง code HTMLและ Java Script ในการพัฒนาเว็บไซต์

3.6 การทดสอบระบบ

ในส่วนนี้จะเป็นการทดสอบการส่งข้อมูลที่เซนเซอร์ตรวจจับได้โดยจะแสดงค่าตัวเลขเฉลี่ยที่วัดได้บนแผนที่ตามตำแหน่งที่วัดได้



ภาพที่ 3-16 ตัวอย่าง การส่งข้อมูลจากอุปกรณ์ไปยังเว็บไซต์

จากภาพที่ 4-1 เป็นการทดสอบการส่งข้อมูลจากอุปกรณ์ไปแสดงผลบนเว็บไซต์ โดยจะแสดงปริมาณฝุ่นที่วัดได้

3.7 การนำไปใช้และการบำรุงรักษา

เมื่อทำการติดตั้ง คณะผู้จัดทำโครงการจะทำการปรับปรุงตามข้อเสนอแนะ ของผู้เชี่ยวชาญและ ผู้ใช้งานต่อไป

3.8 บทสรุป

จากผลการดำเนินงานที่ผ่านมา มีการพัฒนาในส่วนต่าง ๆ ของอุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IoTจะดำเนินการสรุปผล ในบทถัดไป

บทที่ 4

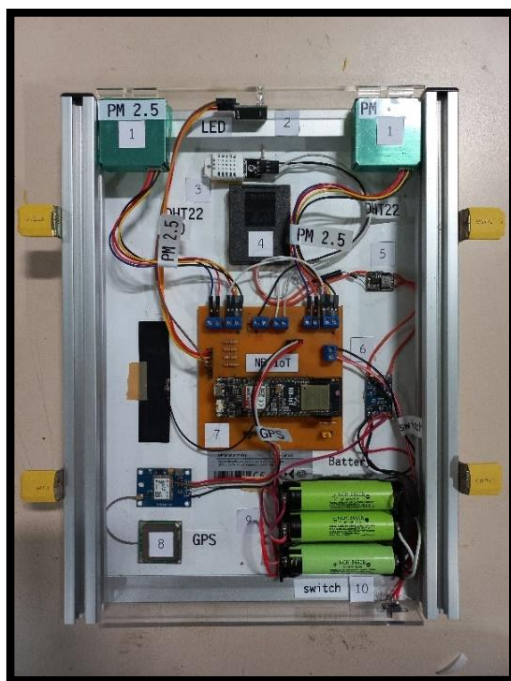
ผลการดำเนินงาน

ผลการดำเนินงานเป็นผลจากการออกแบบ แสดงผลจากการสืบค้นหาข้อมูล พัฒนา จากการวิจัย และทดลองโปรแกรมและเลือกนำมาทดลองใช้งานจริงว่าได้ผลลัพธ์ออกมาในลักษณะเป็นอย่างไรและรวมไปจนถึงโครงสร้างต่าง ๆ ที่ได้นำมาประกอบกันเสร็จสมบูรณ์แล้วมีหน้าตาเป็นอย่างไรการแสดงผลออกมาเป็นส่วนที่ใช้งานจริงเพื่อทดสอบอุปกรณ์ต่าง ๆ ดังนี้

- 4.1 ผลลัพธ์ของอุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IoT
- 4.2 ผลการทำงานของอุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IoT และเว็บไซต์
- 4.3 สรุปผลดำเนินงาน

4.1 ผลลัพธ์ของอุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IoT

4.1.1 อุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IoT



ภาพที่ 4-1 อุปกรณ์วัดฝุ่น

จากภาพที่ 4-1 เป็นการออกแบบอุปกรณ์ ที่นำมาประกอบกันเสร็จสมบูรณ์ และนำมาทดลองใช้งาน

ตารางที่ 4-1 ตำแหน่งอุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IoT

หมายเลข	อุปกรณ์
1	Particulate Matter Sensor SPS30
2	LED
3	DHT22
4	Solar cell
5	Step down
6	battery charger
7	NB-IoT

ตารางที่ 4-1 ตำแหน่งอุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IoT (ต่อ)

8	GPS
9	Battery
10	Switch

4.1.2 อุปกรณ์วัดฝุ่นเคลื่อนที่



ภาพที่ 4-2 อุปกรณ์วัดฝุ่นเคลื่อนที่ ด้านหน้า



ภาพที่ 4-3 อุปกรณ์วัดฝุ่นเคลื่อนที่ ด้านหลัง



ภาพที่ 4-4 อุปกรณ์วัดฝุ่นเคลื่อนที่ ด้านซ้าย



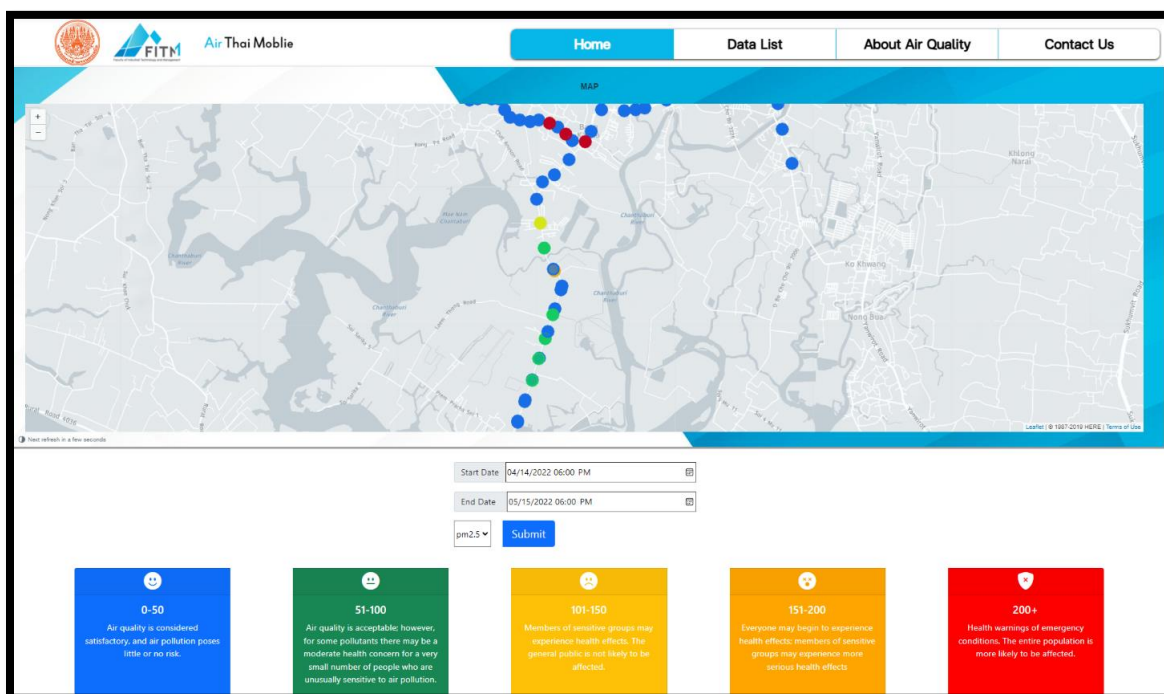
ภาพที่ 4-5 อุปกรณ์วัดฝุ่นเคลื่อนที่ ด้านขวา



ภาพที่ 4-6 อุปกรณ์วัดฝุ่นเคลื่อนที่ ด้านบน

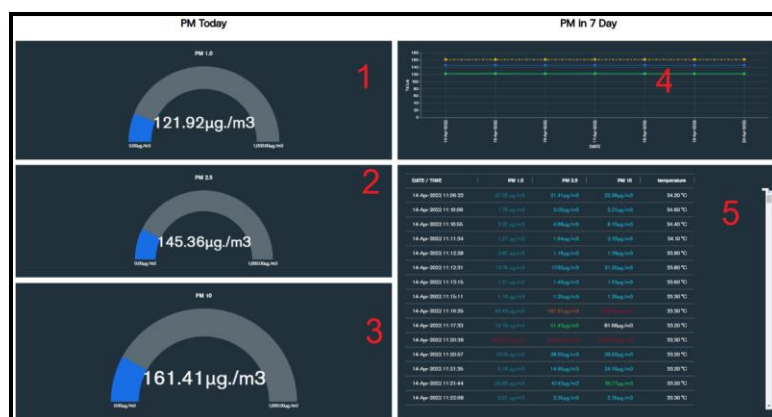
4.1.3 เว็บไซต์แสดงผล

เว็บไซต์แสดงผล จะแบ่งออกเป็น4ส่วน ได้แก่ แผนที่ ปริมาณฝุ่นประจำวันและปริมาณ
ย้อนหลัง ความเป็นมาของฝุ่น pm และ หน้าติดต่อเรา



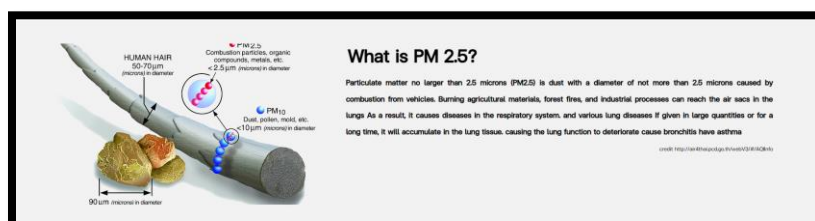
ภาพที่ 4-7 เว็บไซต์แสดงผลหน้า map

จากภาพที่ 4-7 จะเป็นการแสดงแผนที่และสามารถเลือกช่วงวันและเวลาและค่าฝุ่นที่ต้องการดูได้ โดยบอกสีและค่าฝุ่นPMตามลำดับ โดยค่าฝุ่น 0-50 หมายถึง คุณภาพอากาศถือว่าน่าพอใจ และมลภาวะในอากาศมีความเสี่ยงเพียงเล็กน้อยหรือไม่มีเลย 51-100 หมายถึง คุณภาพอากาศเป็นที่ยอมรับ อาจมีมลพิษบางชนิด 101-150 หมายถึง สมาชิกของกลุ่มที่มีความอ่อนไหวอาจประสบปัญหาด้านสุขภาพ ประชาชนทั่วไปไม่น่าจะได้รับผลกระทบ 151-200 หมายถึง สมาชิกของกลุ่มที่มีความอ่อนไหวอาจได้รับผลกระทบด้านสุขภาพที่รุนแรงมากขึ้น และ 200+ หมายถึง ค่าเตือนด้านสุขภาพของภาวะฉุกเฉิน ประชากรทั้งหมดมีแนวโน้มที่จะได้รับผลกระทบมากขึ้น



ภาพที่ 4-8 เว็บไซต์แสดงผลหน้า Data List

จากภาพที่ 4-8 จะเป็นการปริมาณฝุ่นเฉลี่ยต่างๆโดย หมายเลข1 คือปริมาณฝุ่นPM1.0 ประจำวัน หมายเลข2 คือปริมาณฝุ่นPM2.5 ประจำวัน หมายเลข3 คือปริมาณฝุ่นPM10 ประจำวัน หมายเลข4 คือกราฟแสดงปริมาณฝุ่นPM ในช่วง 7 วัน และ หมายเลข5 คือตารางแสดงข้อมูลฝุ่นPM ทั้งหมดที่วัดได้



ภาพที่ 4-9 เว็บไซต์แสดงผลหน้า About Air Quality

จากภาพที่ 4-9 จะเป็นการบอกถึงความเป็นมาของฝุ่น PM



ภาพที่ 4-10 เว็บไซต์แสดงผลหน้า Contact Us

จากภาพที่ 4-10 จะเป็นการบอกช่องทางการติดต่อเรา

4.2 ผลการของการทดลอง

4.2.1 ผลของการทดลอง อุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IoT



ภาพที่ 4-11 ทดสอบการวัดฝุ่นโดยการติดตั้งบนรถยนต์

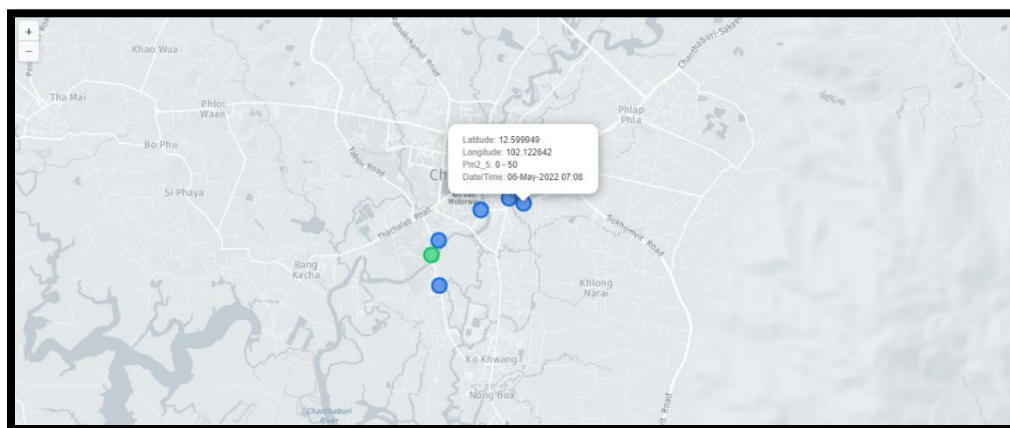
จากภาพที่ 4-11 จะเป็นการทดสอบการวัดฝุ่นโดยการติดตั้งบนรถยนต์



ภาพที่ 4-12 ทดสอบการวัดฝุ่นขณะขับรถยนต์

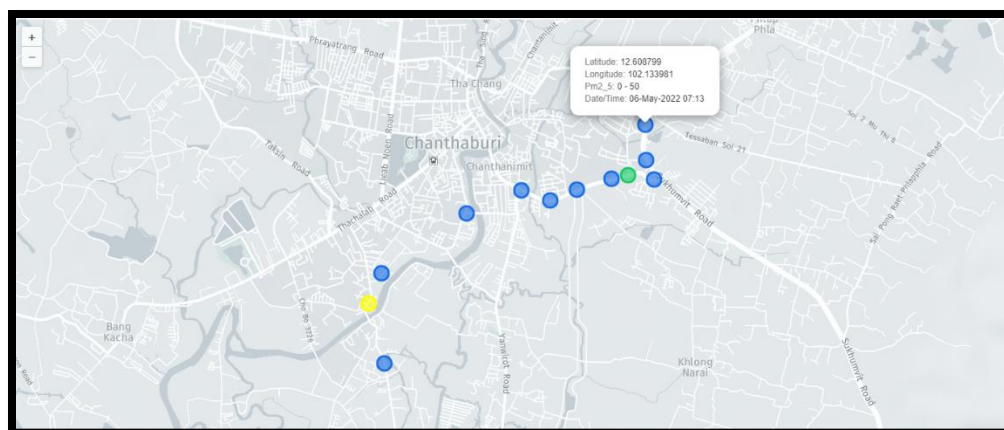
จากภาพที่ 4-12 จะเป็นการทดสอบการวัดฝุ่นขณะขับรถยนต์

4.2.2 ผลของการทดลองเว็บไซต์แสดงปริมาณฝุ่นPM



ภาพที่ 4-13 ทดสอบการแสดงผลข้อมูลปริมาณฝุ่นที่วัดได้ขณะขับรถ

จากภาพที่ 4-13 ทดสอบการแสดงผลข้อมูลปริมาณฝุ่นที่วัดได้ขณะขับรถ



ภาพที่ 4-14 ทดสอบการแสดงผลข้อมูลปริมาณฝุ่นที่วัดได้ขณะขับรถ

จากภาพที่ 4-14 จะเป็นการทดสอบการแสดงผลข้อมูลปริมาณฝุ่นที่วัดได้ขณะขับรถ

4.3 สรุปผลดำเนินงาน

จากผลการดำเนินงาน สามารถทำได้ตามวัตถุประสงค์ที่ต้องการ ซึ่งในบทถัดไปจะเป็นบทที่ 5 จะกล่าวถึงผลการดำเนินงานโดยสรุปอีกครั้ง ปัญหาที่เกิดขึ้นระหว่างการพัฒนา และแนวทางการแก้ปัญหา และข้อเสนอแนะ จากคณะผู้จัดทำ

บทที่ 5

สรุปผลการดำเนินงานและข้อเสนอแนะ

ในบทนี้จะกล่าวถึงบทสรุปผลการดำเนินงานและแนวทางการพัฒนาแบบประเมินความสามารถระบบวัดฝุ่นแบบเคลื่อนที่ เพื่อให้ผู้พัฒนาได้มีแนวทางการดำเนินงานและปรับปรุงแก้ไขจุดต่าง ๆ ของระบบแบบประเมินความสามารถอุปกรณ์วัดฝุ่นเคลื่อนที่ เพื่อให้ระบบมีความสมบูรณ์และมีประสิทธิภาพและประสิทธิผลในการใช้งานมากยิ่งขึ้น

- 5.1 สรุปผลดำเนินงาน
- 5.2 ข้อจำกัดในการทำงาน
- 5.3 ปัญหาและอุปสรรค
- 5.4 ข้อเสนอแนะ

5.1 สรุปผลดำเนินงาน

จากการพัฒนาอุปกรณ์วัดฝุ่นเคลื่อนที่ มีวัตถุประสงค์เพื่อ พัฒนาโครงสร้างของอุปกรณ์วัดฝุ่นเคลื่อนที่และแสดงผลบนเว็บไซต์ได้โดยแสดงตำแหน่งฝุ่นบนแผนที่ การพัฒนาครั้งนี้ คณะผู้จัดทำโครงการระบบอุปกรณ์วัดฝุ่นเคลื่อนที่ เพื่ออำนวยความสะดวกให้แก่ผู้ใช้ทั่วไป โดยอุปกรณ์ที่ติดอยู่บนรถโดยสารหรือยานพาหนะต่าง ๆ เมื่อตรวจจับฝุ่นได้จะส่งค่าไปยัง Mongo DB โดยฐานข้อมูล MongoDB สามารถรองรับข้อมูลจากอุปกรณ์หลายตัวได้พร้อมกัน และแสดงผลผ่านบน เว็บไซต์ได้ และอุปกรณ์วัดฝุ่นมี LED แสดง เป็นสีตามระดับฝุ่นในบริเวณที่วัดได้

5.2 ข้อจำกัดในการทำงาน

- 5.2.1 ด้านงบประมาณ อุปกรณ์มีราคาค่อนข้างสูง
- 5.2.2 ด้านสภาพแวดล้อม ถ้าฝนตกอาจทำให้ค่าฝุ่นคลาดเคลื่อนได้

5.3 ปัญหาและอุปสรรค

- 5.3.1 หากไม่มีสัญญาณอินเทอร์เน็ตอาจทำให้จุดที่วัดมีความคลาดเคลื่อนได้

5.4 ข้อเสนอแนะ

5.4.1 เพิ่มโครงสร้างให้สามารถกันฝนได้

5.4.2 เพิ่มระบบประหยัดพลังงาน

บรรณานุกรม

[1] ฐานข้อมูล, 2564. [Online].

Available: <https://sites.google.com/site/bb28003a/home/khwam-hmay-khxng-thankxmul>

[Accessed: : 6 มิถุนายน 2563]

[2] NB-IoT คืออะไร?, 2563. [Online].

Available: <https://www.iphonemod.net/gsma-truemove-h-nb-iot-thailand.html>.

[Accessed: 6 มิถุนายน 2563]

[3] MongoDB, 2563. [Online].

Available: <https://sysadmin.psu.ac.th/2017/01/11/what-is-mongodb/>.

[Accessed: 6 มิถุนายน 2563]

[4] เซ็นเซอร์วัดฝุ่น, 2563. [Online].

Available: <https://www.sensirion.com/en/environmental-sensors/particulate-matter>.

[Accessed: 6 มิถุนายน 2563]

[5] Node-RED, 2563. [Online].

Available: <http://nrc-intelligentsystems.com>.

[Accessed: 6 มิถุนายน 2563]

[6] ความรู้เกี่ยวกับฝุ่นละออง, 2563. [Online].

Available: <https://www.inmindclean.com/รายละเอียด/ความรู้เกี่ยวกับฝุ่นละออง#:~:text=ฝุ่นละอองคือ%20อนุภาคของแข็ง,สเบสต่อส>.

[Accessed: 6 มิถุนายน 2563]

[6] PM2.5 คืออะไร? อันตรายและการป้องกันฝุ่นละอองขนาดเล็ก, 2563. [Online].

Available: <https://www.daikin.co.th/service-knowledge/pm-2-5/>.

[Accessed: 6 มิถุนายน 2563]

[7] ESP32 เบื้องต้น :: บทที่ 1 แนะนำ ESP32, 2563. [Online].

Available: <https://www.ioxhop.com/article/62/esp32-เบื้องต้น-บทที่-1-แนะนำ-esp32>.

[Accessed: 6 มิถุนายน 2563]

บรรณานุกรม (ต่อ)

[8] HTML, 2563. [Online].

Available: https://www.bu.ac.th/knowledgecenter/executive_journal/july_sep_11/pdf/aw32.pdf. [Accessed: 6 มิถุนายน 2563]

[9] Javascript คืออะไร, 2563. [Online].

Available: <https://www.mindphp.com/คู่มือ/73-คืออะไร/2187-java-javascript-คืออะไร.html>. [Accessed: 6 มิถุนายน 2563]

[10] โครงสร้างโปรแกรมของ ARDUINO, 2533. [Online]

Available: http://www.sbt.ac.th/new/sites/default/files/TNP_Unit_3.pdf. [Accessed: 6 มิถุนายน 2563]

[11] ทำความรู้จักกับ Internet of Things, 2563. [Online]

Available: <https://www.aware.co.th/ IoT-คืออะไร/>. [Accessed: 6 มิถุนายน 2563]

[12] MQTT กับระบบ IoT, 2563. [Online].

Available: https://www.mostori.com/blog_detail.php?b_id=9320. [Accessed: 6 มิถุนายน 2563]

[13] NB-IoT, 2563. [Online].

Available: <https://www.iphonemod.net/gsma-truemove-h-nb-iot-thailand.html>. [Accessed: 6 มิถุนายน 2563]

[14] MongoDB, 2563. [Online].

Available: <https://sysadmin.psu.ac.th/2017/01/11/what-is-mongodb/>. [Accessed: 6 มิถุนายน 2563]

[15] Solar Cell, 2563. [Online].

Available: <http://www.ccsolar-thai.com/>. [Accessed: 6 มิถุนายน 2563]

[16] battery, 2563. [Online].

Available: <http://www.diy-solarcell.com/>. [Accessed: 6 มิถุนายน 2563]

บรรณานุกรม (ต่อ)

[17] เซ็นเซอร์ GPS Module GY-NEO6MV2, 2563. [Online].

Available: <https://www.cartrack.co.th/gps-module>

[Accessed: 6 มิถุนายน 2563]

[18] Visual Studio Code คืออะไร, 2563. [Online].

Available: <https://eleceasy.com/t/visual-studio-code-vs-code/2042>.

[Accessed: 6 มิถุนายน 2563]

[19] วิธีใช้งานโปรแกรม Arduino IDE เบื้องต้น, 2563. [Online].

Available: <https://poundxi.com/วิธีใช้งานโปรแกรม-arduino-ide-เบื้องต้น/>.

[Accessed: 6 มิถุนายน 2563]

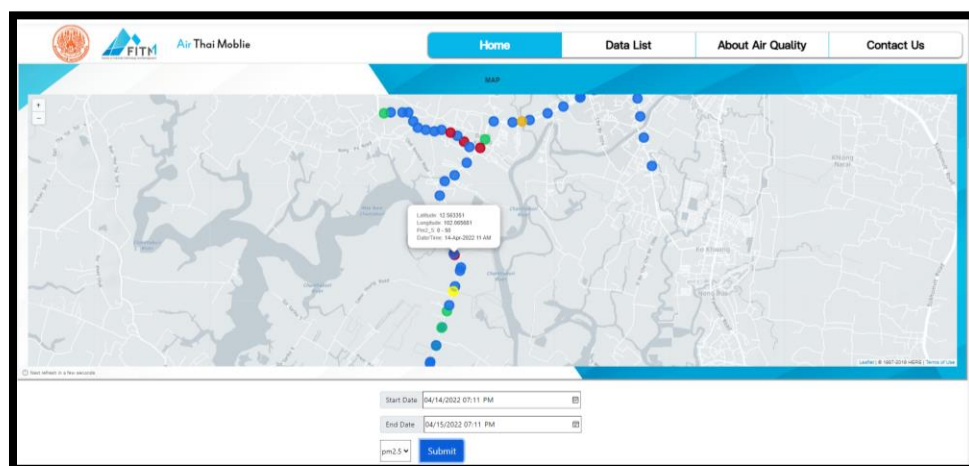
ภาคผนวก ก.

คู่มือการใช้งานอุปกรณ์วัดฝุ่นเคลื่อนที่ด้วยเทคโนโลยี IoT



ภาพที่ ก.1 อุปกรณ์วัดฝุ่นเคลื่อนที่

จากภาพที่ ก.1 ติดตั้งอุปกรณ์ ที่ และเปิดเครื่องใช้งาน



ภาพที่ ก.2 เว็บไซต์แสดงผลหน้า map

จากภาพที่ ก.2 จะเป็นการแสดงแผนที่และสามารถเลือกช่วงวันและเวลาและค่าฝุ่นที่ต้องการดูได้

ภาคผนวก ข.

โค้ดการทำงาน

เขียนโค้ดการทำงานของระบบ

```
#if CONFIG_FREERTOS_UNICORE
#define ARDUINO_RUNNING_CORE 0
#else
#define ARDUINO_RUNNING_CORE 1
#endif

#include "AIS_SIM7020E_API.h"
#include <Stream.h>
#include <TinyGPS++.h>
#include <HardwareSerial.h>
#include <sps30.h>
#include <math.h>

#include "DHT.h"
#define DHTPIN 26
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
TinyGPSPlus gps;
HardwareSerial SerialGPS(2);

double latitude, longitude;
String lat_str, lng_str;
double metre;
double latd1, latd2, lngd1, lngd2;

double pm1, pm25, pm10;
double t = 37.0;
```



```

unsigned long timer;
int dataready = 0;
bool tempR = true, dustR = false, posR = false;

String address  = "34.83.166.107";          //Your IPAddress or mqtt server url
String serverPort = "1883";                //Your server port
String clientID  = "";                     //Your client id < 120 characters
String topic     = "data";                 //Your topic    < 128 characters
String payload;
String username  = "";                     //username for mqtt server, username <= 100
characters
String password  = "";                     //password for mqtt server, password <= 100
characters
unsigned int subQoS    = 0;
unsigned int pubQoS    = 0;
unsigned int pubRetained = 0;
unsigned int pubDuplicate = 0;

const long interval = 2000;                //time in millisecond
unsigned long previousMillis = 0;
int cnt = 0;
int gpsinit = 2;

bool flag = false;
int stage = 10;

AIS_SIM7020E_API nb;

//LED

```

```
int led = 0;
uint8_t ledR = 14;
uint8_t ledG = 15;
uint8_t ledB = 25;

uint8_t ledArray[3] = {1, 2, 3};
const boolean invert = true;

uint8_t color = 0;      // a value from 0 to 255 representing the hue
uint32_t R, G, B;       // the Red Green and Blue color components
uint8_t brightness = 255; // 255 is maximum brightness, but can be changed. Might
                           // need 256 for common anode to fully turn off.

int dc = 0;
int sgps = 0;

void TaskGPS( void *pvParameters );

void setup() {
  Serial.begin(115200);
  pinMode(2, OUTPUT);
  nb_init();
  sps30_init();
  //hdc1080.begin(0x40);
  dht.begin();
  ledcAttachPin(ledR, 1);
  ledcAttachPin(ledG, 2);
  ledcAttachPin(ledB, 3);
}
```

```

ledcSetup(1, 12000, 8);
ledcSetup(2, 12000, 8);
ledcSetup(3, 12000, 8);

SerialGPS.begin(9600, SERIAL_8N1, 32, 33); //tx, rx
sgps = 1;

/*xTaskCreatePinnedToCore(
    TaskNB
    , "TaskNB" // A name just for humans
    , 2048 // This stack size can be checked & adjusted by reading the Stack
    Highwater
    , NULL
    , 3 // Priority, with 3 (configMAX_PRIORITIES - 1) being the highest, and 0 being the
    lowest.
    , NULL
    , ARDUINO_RUNNING_CORE);*/

xTaskCreatePinnedToCore(
    TaskGPS
    , "TaskGPS" // A name just for humans
    , 1024 // This stack size can be checked & adjusted by reading the Stack
    Highwater
    , NULL
    , 1 // Priority, with 3 (configMAX_PRIORITIES - 1) being the highest, and 0 being the
    lowest.
    , NULL
    , ARDUINO_RUNNING_CORE);
}

```

```
void loop() {
  double te = dht.readTemperature();
  if (isnan(te)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    //tempR = false;
    //return;
  }
  else {
    t = te;
    //tempR = true;
  }
  Serial.println("Temp : " + String(t));
  //t = String(hdc1080.readTemperature());
  //Serial.println("Temp : " + t);
  struct sps30_measurement m;
  char serial[SPS30_MAX_SERIAL_LEN];
  uint16_t data_ready;
  int16_t ret;

  do {
    ret = sps30_read_data_ready(&data_ready);
    if (ret < 0) {
      Serial.print("error reading data-ready flag: ");
      Serial.println(ret);
    } else if (!data_ready)
      Serial.print("data not ready, no new measurement available\n");
    else
      break;
  }
```

```

    delay(100); /* retry in 100ms */
} while (1);

ret = sps30_read_measurement(&m);
if (ret < 0) {
    Serial.print("error reading measurement\n");
} else {

#ifndef PLOTTER_FORMAT
    Serial.print("PM 1.0: ");
    Serial.println(m.mc_1p0);
    Serial.print("PM 2.5: ");
    Serial.println(m.mc_2p5);
    Serial.print("PM 10.0: ");
    Serial.println(m.mc_10p0);

    pm1 = (double)m.mc_1p0;
    pm25 = (double)m.mc_2p5;
    pm10 = (double)m.mc_10p0;

    payload = "{\"data\":{\"temp\":" + String(t) +
        "\",\"pm1_0\":" + String(pm1) +
        "\",\"pm2_5\":" + String(pm25) +
        "\",\"pm10_0\":" + String(pm10) +
        "\",\"lat\":" + lat_str +
        "\",\"lng\":" + lng_str + "}}";

    if (pm25 <= 50) led = 3;
    if (pm25 >= 51) led = 2;
    if (pm25 >= 101) led = 50;

```

```
if (pm25 >= 151) led = 101;
if (pm25 > 200) led = 1;

ledcWrite(1, 0);
ledcWrite(2, 0);
ledcWrite(3, 0);

if (led == 101) {
    ledcWrite(1, 255);
    ledcWrite(2, 10);
}
if (led == 50) {
    ledcWrite(1, 255);
    ledcWrite(2, 50);
}

else ledcWrite(led, 255);
dustR = true;
#endif /* PLOTTER_FORMAT */
}
if (tempR && dustR && posR) {
    if (pm25 < 300) {
        pm25 = 300;
    }
    digitalWrite(2, 1);
    nb_send();
    digitalWrite(2, 0);
    gpsinit = 0;
    posR = false;
```

```
    //vTaskDelay(1000);
}
else {
    Serial.println("Data not ready");
}
vTaskDelay(1000);
}

void nb_init() {
    nb.begin();
    setupMQTT();
    nb.setCallback(callback);
    previousMillis = millis();
    cnt = 0;
}

void nb_send() {
    nb.MQTTresponse();
    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis >= interval) {
        connectStatus();
        nb.publish(topic, payload);
        vTaskDelay(500);
        previousMillis = currentMillis;
    }
    vTaskDelay(10000);
}

void sps30_init() {
```

```
int16_t ret;
uint8_t auto_clean_days = 4;
uint32_t auto_clean;

sensirion_i2c_init();

while (sps30_probe() != 0) {
    Serial.print("SPS sensor probing failed\n");
    vTaskDelay(500);
}

#ifndef PLOTTER_FORMAT
    Serial.print("SPS sensor probing successful\n");
    flag = true;
#endif // PLOTTER_FORMAT

ret = sps30_set_fan_auto_cleaning_interval_days(auto_clean_days);
if (ret) {
    Serial.print("error setting the auto-clean interval: ");
    Serial.println(ret);
}

ret = sps30_start_measurement();
if (ret < 0) {
    Serial.print("error starting measurement\n");
}

#ifndef PLOTTER_FORMAT
    Serial.print("measurements started\n");
#endif // PLOTTER_FORMAT*/
```



```

}

void DeepSleep(int TIME_TO_SLEEP) {
    uint64_t sleeptime = UINT64_C(TIME_TO_SLEEP * 1000000);
    esp_sleep_enable_timer_wakeup(sleeptime);
    esp_deep_sleep_start();
}

//===== MQTT Function =====

void setupMQTT() {
    if (!nb.connectMQTT(address, serverPort, clientId, username, password)) {
        Serial.println("\nconnectMQTT");
    }
    nb.subscribe(topic, subQoS);
    // nb.unsubscribe(topic);
}

void connectStatus() {
    if (!nb.MQTTstatus()) {
        if (!nb.NBstatus()) {
            Serial.println("reconnectNB ");
            nb.begin();
        }
        Serial.println("reconnectMQ ");
        setupMQTT();
    }
}

void callback(String & topic, String & payload, String & QoS, String & retained) {
    Serial.println("-----");
    Serial.println("# Message from Topic \"" + topic + "\" : " + nb.toString(payload));
}

```

```

Serial.println("# QoS = " + QoS);
if (retained.indexOf(F("1")) != -1) {
    Serial.println("# Retained = " + retained);
}
}

void TaskNB(void *pvParameters) // This is a task.
{
    (void) pvParameters;

    for (;;) // A Task shall never return or exit.
    {
        if (tempR && dustR && posR) {
            if (pm25 < 300) {
                digitalWrite(2, 1);
                nb_send();
                digitalWrite(2, 0);
                gpsinit = 0;
                posR = false;
                vTaskDelay(1000);
            }
        }

        Serial.println("Data not ready");
        vTaskDelay(1000);
    }
}

void TaskGPS(void *pvParameters) // This is a task.

```

```

{
  (void) pvParameters;

  for (;;) // A Task shall never return or exit.
  {
    if (gps.encode(SerialGPS.read()))
    {
      if (gps.location.isValid())
      {
        latitude = gps.location.lat();
        lat_str = String(latitude , 6);
        longitude = gps.location.lng();
        lng_str = String(longitude , 6);
        latd2 = lat_str.toDouble();
        lngd2 = lng_str.toDouble();
        Serial.print("Latitude = ");
        Serial.println(lat_str);
        Serial.print("Longitude = ");
        Serial.println(lng_str);
        //posR = true;
        if (gpsinit == 2) {
          posR = true;
        }

        if (gpsinit == 0) {
          latd1 = lat_str.toDouble();
          lngd1 = lng_str.toDouble();
          gpsinit = 1;
        }
      }
    }
  }
}

```

```

    metre = toMetre(latd1, lngd1, latd2, lngd2);
    Serial.println(String(metre) + " m.");

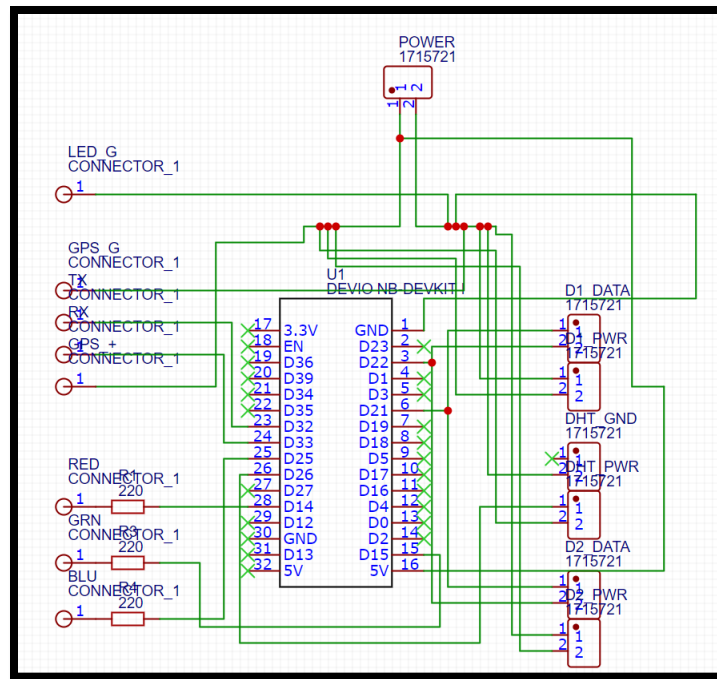
    if (metre >= 200) {
        //gpsinit = 0;
        posR = true;
    }
}
vTaskDelay(1000);
}
}

double toMetre(double lat1, double lon1, double lat2, double lon2) { // generally
used geo measurement function
    double R = 6378.137; // Radius of earth in KM
    double dLat = lat2 * PI / 180 - lat1 * PI / 180;
    double dLon = lon2 * PI / 180 - lon1 * PI / 180;
    double a = sin(dLat / 2) * sin(dLat / 2) + cos(lat1 * PI / 180) * cos(lat2 * PI / 180) *
sin(dLon / 2) * sin(dLon / 2);
    double c = 2 * atan2(sqrt(a), sqrt(1 - a));
    double d = R * c;
    return d * 1000; // meters
}

```

ภาคผนวก ค.

การต่อวงจร



ภาพที่ ค.1 การต่อวงจร

ตารางที่ ค-1 การเชื่อมต่อของอุปกรณ์

ช่องเชื่อมต่อ	จุดเชื่อมต่อ
MCU	
vcc	power
gnd	gnd
DHT22	
data	MCU PIN 26
power	power
gnd	gnd
GPS	
RX	MCU PIN 32
TX	MCU PIN 33
power	power

ตารางที่ ค-1 การเชื่อมต่อของอุปกรณ์ (ต่อ)

gnd	gnd
LED	
red	MCU PIN 14
green	MCU PIN 15
blue	MCU PIN 25
gnd	gnd
Dust sensors	
Power	power
SDA	MCU PIN 21
SCL	MCU PIN 22
SEL	gnd
gnd	gnd