



โครงการ

การเรียนการสอนเพื่อเสริมประสบการณ์

ชื่อโครงการ การออกแบบระบบตรวจวัด PM2.5 ด้วยระบบสมองกลฝังตัว
Designing of PM2.5 detector with embedded system

ชื่อนิสิต นายศุภณัฐ ฤทัยเข้มชื่น **เลขประจำตัว** 6033440623

ภาควิชา ฟิสิกส์

ปีการศึกษา 2563

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

| | |
|------------------|---|
| โครงการ | การออกแบบระบบตรวจวัด PM2.5 ด้วยระบบสมองกลฝังตัว |
| ผู้จัดทำโครงการ | นายศุภณัฐ ฤทัยเข้มชื่น |
| อาจารย์ที่ปรึกษา | รองศาสตราจารย์ ดร.ต้นพงศ์ แก้วคงคา |
| ภาควิชา | ฟิสิกส์ |
| ปีการศึกษา | 2563 |

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต ภาควิชาฟิสิกส์

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ภาควิชาฟิสิกส์ ปีการศึกษา 2563

คณะกรรมการได้ตรวจรับรายงานฉบับนี้แล้ว

.....*ชูศรี วงศ์มณีโรจน์*..... ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ชูศรี วงศ์มณีโรจน์)

.....*Bault*..... กรรมการสอบ

(รองศาสตราจารย์ ดร.บุญโชติ เผ่าสวัสดิ์ยรรยง)

.....*ต้นพงศ์ แก้วคงคา*..... อาจารย์ที่ปรึกษา

(รองศาสตราจารย์ ดร.ต้นพงศ์ แก้วคงคา)

| | |
|------------------|---|
| หัวข้อโครงการ | การออกแบบระบบตรวจวัด PM2.5 ด้วยระบบสมองกลฝังตัว |
| ผู้จัดทำโครงการ | นายศุภณัฐ ฤทัยเข้มชื่น |
| อาจารย์ที่ปรึกษา | รองศาสตราจารย์ ดร.ต้นพงศ์ แก้วคงคา |
| ภาควิชา | ฟิสิกส์ |
| ปีการศึกษา | 2563 |

บทคัดย่อ

โครงการนี้มีจุดประสงค์เพื่อออกแบบระบบตรวจวัด PM2.5 ด้วยระบบสมองกลฝังตัว โดยโครงการนี้ใช้ไมโครคอนโทรลเลอร์ ESP32 ซึ่งมีคุณสมบัติที่สามารถเชื่อมต่อระบบเครือข่ายคอมพิวเตอร์ไร้สาย (Wireless LAN หรือ Wi-Fi) มาเชื่อมต่อกับตัวรับรู้ปริมาณฝุ่นละออง (PMS7003) ตัวรับรู้ความชื้นสัมพัทธ์และอุณหภูมิ (DHT22) และตัวรับรู้ปริมาณปริมาณก๊าซคาร์บอนไดออกไซด์ (MQ-135) ภายในระบบตรวจวัด PM2.5 ด้วยระบบสมองกลฝังตัวมีการจัดเก็บข้อมูลไว้ใน MySQL ซึ่งเป็นระบบจัดการฐานข้อมูล โดยสถานที่ที่ได้ทำการเก็บข้อมูลมีทั้งหมด 4 สถานที่ ได้แก่ คณะวิทยาศาสตร์ คณะวิศวกรรมศาสตร์ คณะอักษรศาสตร์ และสระน้ำของจุฬาลงกรณ์มหาวิทยาลัย และนำข้อมูลที่ได้ไปใช้ในการทำเว็บไซต์ เพื่อใช้ในการเฝ้าระวังปริมาณ PM 2.5 ได้หลาย ๆ สถานที่ ในหน้าเว็บไซต์เดียวกันและยังช่วยให้รู้ค่าปริมาณสูงสุดและต่ำสุดของฝุ่นละอองได้อย่างรวดเร็ว โดยใช้คำสั่งของภาษา JavaScript ทำให้เมื่อลากเคอร์เซอร์เมาส์ไปยังบริเวณที่ต้องการให้ปรากฏค่าสูงสุดและต่ำสุดของปริมาณ PM 2.5 ณ บริเวณที่มีเครื่องวัดติดตั้งไว้ ทำให้สะดวกและง่ายต่อการติดตามค่าปริมาณฝุ่นละอองในหน้าเว็บไซต์เดียว ในส่วนของหน้าเว็บยังใช้ภาษา HTML เพื่อที่จะสามารถคลิกเข้าไปดูข้อมูลค่าฝุ่นละออง ณ เวลาต่าง ๆ ได้อย่างละเอียดในบริเวณที่สนใจได้อีกด้วย อีกทั้งยังส่งข้อมูลของปริมาณฝุ่นละอองในอากาศ อุณหภูมิ ความชื้นสัมพัทธ์ และปริมาณก๊าซคาร์บอนไดออกไซด์ ไปแสดงค่าได้แบบ Real time ผ่านทางหน้าเว็บไซต์ของ NETPIE นอกจากนี้ยังมีการกำหนดเงื่อนไขเมื่อปริมาณฝุ่นละออง PM2.5 เกินกว่า 50 ug/m^3 ให้ส่งข้อความแจ้งเตือนพร้อมทั้งระบุปริมาณฝุ่นละอองที่เกินเกณฑ์ที่กำหนดไว้ไปยัง LINE Notify เพื่อป้องกันอันตรายจากมลพิษทางอากาศได้ทันที่

| | |
|---------------|--|
| Title | Designing of PM2.5 detector with embedded system |
| Name | Supanut Ruethaichaemchuen |
| Adviser | Assoc.Prof.Dr.Tonphong Kaewkongka |
| Major | Physics |
| Academic year | 2563 |

Abstract

This project proposes the design of PM2.5 detector with embedded system. In this project, using ESP32 microcontroller that can connect wirelessly to dust sensor (PMS7003), gas sensor (MQ-135) and relative humidity and temperature sensor (DHT22). The PM2.5 monitoring system consists of embedded system and data storage in MySQL which is a database management system. Measurements of collected data are performed in 4 different areas where are faculty of Science, faculty of Engineering, faculty of Arts and the pond of Chulalongkorn University. The collected data are used for generating website in order to monitor the amount of PM 2.5 in many places on the same website and show the maximum and minimum amount of dust. For easy tracking the amount of dust on single web page, using the command of the JavaScript language to appear the maximum and minimum values of PM2.5 by moving mouse cursor to the desired area and also using HTML in order to view the dust values at different times in the area of interest. The system is also capable of sending the collected data in order to display information about the amount of dust in the air, temperature, relative humidity and the amount of carbon dioxide in real time via the NETPIE website. In addition, the proposed system is programmed to indicate warning condition when the values of the PM2.5 exceed 50 ug/m^3 to send notification messages to LINE Notify for prevent dangers from air pollution in time.

กิตติกรรมประกาศ

ขอขอบพระคุณอาจารย์ที่ปรึกษาโครงการ รองศาสตราจารย์ ดร.ต้นพงศ์ แก้วคงคา ที่ช่วยดูแลให้คำปรึกษา ให้ความรู้และคำแนะนำสำหรับแนวทางในการดำเนินงานที่ดีมาโดยตลอด

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ชูศรี วงศ์มณีโรจน์ และรองศาสตราจารย์ ดร.บุญโชติ เผ่าสวัสดิ์ยรียง ที่ได้กรุณาสละเวลามาเป็นกรรมการตรวจสอบ รวมทั้งให้คำแนะนำ พร้อมทั้งให้ความสนใจในการตรวจสอบและแก้ไขโครงการ

ขอบพระคุณคณาจารย์ภาควิชาฟิสิกส์ทุกท่านที่ได้อบรมสั่งสอน ให้ความรู้ ให้คำแนะนำ ฝึกให้คิดและแก้ปัญหาเป็น

ขอบพระคุณนายเฉลิมวุฒิ ชำนาญฉา พนักงานประจำห้องปฏิบัติการอิเล็กทรอนิกส์ที่สละเวลาให้คำปรึกษา ให้ความช่วยเหลือ ให้คำแนะนำ และจัดหาอุปกรณ์เครื่องมือในการทำโครงการให้สำเร็จไปได้ด้วยดี

และสุดท้ายขอขอบพระคุณอย่างยิ่ง สำหรับบิดา มารดา ครอบครัว และเพื่อน ๆ รุ่นพี่ ภาควิชาฟิสิกส์ ที่คอยให้การสนับสนุน ให้กำลังใจ และยังร่วมช่วยแก้ปัญหา ให้คำปรึกษาตลอดเวลาเสมอมา

สารบัญ

| | |
|---|----|
| บทคัดย่อ (ภาษาไทย)..... | ก |
| บทคัดย่อ (ภาษาอังกฤษ)..... | ข |
| กิตติกรรมประกาศ..... | ค |
| บทที่ 1 บทนำ..... | 1 |
| 1.1 ที่มาและความสำคัญ..... | 1 |
| 1.2 วัตถุประสงค์ของโครงการ..... | 1 |
| 1.3 ระเบียบวิธีวิจัย..... | 2 |
| 1.4 ประโยชน์ที่คาดว่าจะได้รับ..... | 3 |
| บทที่ 2 ทฤษฎีที่เกี่ยวข้องกับอุปกรณ์ที่ใช้..... | 4 |
| 2.1 ทฤษฎีที่เกี่ยวข้อง..... | 4 |
| 2.1.1 ฝุ่นละออง (Dust)..... | 4 |
| 2.1.2 ผลกระทบของฝุ่นละออง..... | 4 |
| 2.1.3 อุณหภูมิ (Temperature)..... | 5 |
| 2.1.4 ความชื้น (Humidity)..... | 5 |
| 2.1.5 ความชื้นสัมพัทธ์ (Relative humidity)..... | 6 |
| 2.1.6 ตัวรับรู้ (Receiver)..... | 6 |
| 2.2 อุปกรณ์ที่ใช้..... | 7 |
| 2.2.1 ESP32..... | 8 |
| 2.2.2 ตัวรับรู้ปริมาณฝุ่นละออง (PMS7003)..... | 8 |
| 2.2.3 ตัวรับรู้ความชื้นสัมพัทธ์และอุณหภูมิในอากาศ (DHT 22)..... | 9 |
| 2.2.4 ตัวรับรู้ปริมาณก๊าซในอากาศ (MQ-135 – Air quality sensor)..... | 10 |
| 2.2.5 NETPIE..... | 11 |
| 2.2.6 Dashboard..... | 11 |
| 2.2.7 MySQL..... | 12 |
| 2.2.8 LINE Notify..... | 13 |

| | |
|--|-----------|
| 2.2.9 ภาษา HTML..... | 13 |
| 2.2.10 ภาษาจาวาสคริปต์ (JavaScript)..... | 13 |
| บทที่ 3 การออกแบบการทำงานและขั้นตอนการทำงาน..... | 15 |
| บทที่ 4 ผลการทดลอง..... | 22 |
| 4.1 ทำเว็บไซต์เพื่อดูค่าปริมาณฝุ่นละอองของบริเวณต่าง ๆ..... | 22 |
| 4.2 ทดสอบการอ่านค่าข้อมูลจากตัวรับสัญญาณในระยะไกลผ่านทางหน้าเว็บไซต์ NETPIE..... | 29 |
| 4.3 ทดสอบการส่งข้อความแจ้งเตือนปริมาณฝุ่นละออง PM2.5 ที่เกินเกณฑ์กำหนดไปยัง LINE Notify..... | 34 |
| บทที่ 5 สรุปผลการทดลอง ปัญหาและข้อเสนอแนะ..... | 36 |
| 5.1 สรุปผลการทดลอง..... | 36 |
| 5.2 ปัญหาที่พบ..... | 38 |
| 5.3 ข้อเสนอแนะ..... | 39 |
| บรรณานุกรม..... | 40 |
| ภาคผนวก..... | 42 |

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

เนื่องจากประเทศไทยเคยเผชิญปัญหาหมอกควันเข้าขั้นรุนแรง โดยเฉพาะฝุ่นควันขนาดเล็กที่เล็กกว่า 2.5 ไมโครเมตร หรือที่เรียกว่า PM2.5 โดยฝุ่นละอองขนาดเล็กนี้สามารถส่งผลกระทบต่อร่างกายได้ทั้งในระยะยาว และในระยะสั้น ซึ่งเราควรจะต้องหาหนทางที่ป้องกันฝุ่นละอองขนาดเล็กเหล่านี้ได้ หรือพยายามหลีกเลี่ยงสถานที่ ที่มีฝุ่นละอองเยอะ โดยยังมีปัจจัยอื่นที่ส่งผลกระทบต่อปริมาณของฝุ่นละออง เช่น ก๊าซไนโตรเจนไดออกไซด์ (NO₂) ก๊าซซัลเฟต (SO₄) ก๊าซแอมโมเนีย (NH₃) อนุภาคคาร์บอน ความชื้นในอากาศ และอุณหภูมิ เป็นต้น [1,2]

จากปัจจัยที่ได้กล่าวมา จึงทำให้เกิดแนวคิดที่จะจัดทำเครื่องวัดสภาพอากาศ โดยใช้ตัวรับรู้ปริมาณ ฝุ่นละออง ตัวรับรู้ความชื้นและอุณหภูมิ และตัวรับรู้ก๊าซ ซึ่งตัวรับรู้เหล่านี้เป็นตัววัดค่าปริมาณต่าง ๆ โดยใช้ ไมโครคอนโทรลเลอร์ในการรับส่งข้อมูลจากระยะไกล และนำข้อมูลมาบันทึกไว้ในฐานข้อมูล เพื่อนำไปสร้าง เว็บไซต์ด้วยภาษา HTML และภาษา JavaScript นอกจากนี้ยังสามารถส่งข้อมูลไปแสดงผลผ่าน Dashboard โดย เทคโนโลยี ที่นำมาใช้ในการส่งข้อมูล คือ ไมโครคอนโทรลเลอร์ ซึ่งสามารถเขียนโปรแกรมควบคุมการทำงานของ อุปกรณ์ต่างๆ ที่เชื่อมต่อกับไมโครคอนโทรลเลอร์ได้ ในโครงการนี้เลือกใช้ไมโครคอนโทรลเลอร์ที่สามารถเชื่อมต่อกับอินเทอร์เน็ตได้ ส่งผลให้สามารถอ่านค่าข้อมูลได้แบบ Real time แม้อยู่ไกลจากเครื่องมือ และยังสามารถส่งค่า ปริมาณฝุ่นละอองที่เกินเกณฑ์ที่กำหนดไว้ ไปยัง LINE Notify เพื่อป้องกันอันตรายจากมลพิษทางอากาศได้ ทันที

1.2 วัตถุประสงค์ของโครงการ

1. ศึกษาการทำงานของตัวรับรู้ปริมาณฝุ่นละออง ตัวรับรู้ความชื้นสัมพัทธ์และอุณหภูมิ และตัวรับรู้ ปริมาณก๊าซคาร์บอนไดออกไซด์
2. สร้างเครื่องมือตรวจวัดสภาพอากาศด้วยไมโครคอนโทรลเลอร์ที่สามารถรับส่งข้อมูลจากตัวรับรู้ ได้ในระยะไกล
3. สามารถเก็บข้อมูลที่ได้จากตัวรับรู้ไว้ในฐานข้อมูลและแสดงผลได้แบบ Real time
4. สามารถสร้างเว็บไซต์ที่จะทำการเฝ้าระวังปริมาณ PM 2.5 ได้หลาย ๆ สถานที่ในเวลาเดียวกัน

1.3 ระเบียบวิจัย

1.3.1 แผนการศึกษา

ขั้นตอนการดำเนินงานแบ่งออกเป็น 11 ขั้นตอน ดังนี้

- 1) ศึกษาการทำงานของอุปกรณ์
 - 1.1 ศึกษาการทำงานของไมโครคอนโทรลเลอร์
 - 1.2 ศึกษาการทำงานของตัวรับรู้วัดปริมาณฝุ่นละออง อุณหภูมิ ความชื้น และก๊าซคาร์บอนไดออกไซด์
 - 1.3 เขียนโปรแกรมออกแบบอัลกอริทึม
 - 1.4 ศึกษาการเขียนโปรแกรมเพื่อฝังตัวบนไมโครคอนโทรลเลอร์
 - 1.5 ศึกษาการเขียนเว็บไซต์ด้วยภาษา HTML
- 2) ออกแบบวงจรควบคุมการทำงาน
- 3) เขียนโปรแกรมที่ทำให้อุปกรณ์ต่าง ๆ ทำงานได้ ในวงจรที่ออกแบบไว้
- 4) สร้างส่วนแสดงผลข้อมูลบน NETPIE
- 5) สร้างส่วนเก็บข้อมูลบน MySQL
- 6) สร้างส่วนแสดงผลข้อมูลบน LINE Notify
- 7) สร้างส่วนแสดงผลบนเว็บไซต์
- 8) ทดสอบการส่งข้อมูลด้วยระบบสมองกลฝังตัว
- 9) วิเคราะห์ข้อมูลและสรุปผล
- 10) จัดทำรายงาน
- 11) นำเสนอผลงาน

1.3.2 ระยะเวลาที่ศึกษา

| | รายการ | พ.ศ. 2563 | | | | พ.ศ. 2564 | | | | | | |
|----|---|-----------|------|------|------|-----------|------|-------|-------|------|-------|------|
| | | ก.ย. | ต.ค. | พ.ย. | ธ.ค. | ม.ค. | ก.พ. | มี.ค. | เม.ย. | พ.ค. | มิ.ย. | ก.ค. |
| 1 | ศึกษาการทำงานของอุปกรณ์ | | | | | | | | | | | |
| 2 | ออกแบบวงจรควบคุมการทำงาน | | | | | | | | | | | |
| 3 | เขียนโปรแกรมที่ทำให้อุปกรณ์ต่าง ๆ ทำงานได้ ในวงจรที่ออกแบบไว้ | | | | | | | | | | | |
| 4 | สร้างส่วนแสดงผลข้อมูลบน NETPIE | | | | | | | | | | | |
| 5 | สร้างส่วนเก็บข้อมูลบน MySQL | | | | | | | | | | | |
| 6 | สร้างส่วนแสดงผลข้อมูลบน LINE Notify | | | | | | | | | | | |
| 7 | สร้างส่วนแสดงผลบนเว็บไซต์ | | | | | | | | | | | |
| 8 | ทดสอบการส่งข้อมูลด้วยระบบสมองกลฝังตัว | | | | | | | | | | | |
| 9 | วิเคราะห์ข้อมูลและสรุปผล | | | | | | | | | | | |
| 10 | จัดทำรายงาน | | | | | | | | | | | |
| 11 | นำเสนอผลงาน | | | | | | | | | | | |

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับความรู้ทางด้านการทำงานของตัวรับรู้ที่ใช้ตรวจวัดสภาพอากาศและสามารถนำมาใช้ในการสร้างเครื่องมือที่สามารถตรวจวัดสภาพอากาศแบบ Real time เพื่อป้องกันอันตรายจากมลพิษทางอากาศได้ โดยการเช็คข้อมูลผ่านทางแอปพลิเคชัน Line
2. ได้ข้อมูลเพื่อนำมาใช้ในการสร้างเว็บไซต์เพื่อใช้ในการติดตามและศึกษาปริมาณต่าง ๆ โดยปริมาณต่าง ๆ ที่แสดงในเว็บไซค์มีดังนี้ปริมาณฝุ่นละออง อุณหภูมิ ความชื้นในอากาศ และปริมาณก๊าซคาร์บอนไดออกไซด์

บทที่ 2

ทฤษฎีที่เกี่ยวข้องและอุปกรณ์ที่ใช้

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 ฝุ่นละออง (Dust)

ฝุ่นละออง หมายถึง อนุภาคของแข็งและหยดละอองของเหลวที่แขวนลอยกระจายในอากาศ อนุภาคที่แขวนลอยอยู่ในอากาศ บางชนิดมีขนาดใหญ่และสีดำจนมองเห็นเป็นเขม่าควัน แต่บางชนิดมีขนาดเล็กมากจนมองไม่เห็นด้วยตาเปล่า ฝุ่นละอองที่แขวนลอยในอากาศ โดยทั่วไปมีขนาดตั้งแต่ 100 ไมครอนลงมา ก่อให้เกิดผลกระทบต่อสุขภาพของคน สัตว์ พืช และเกิดความเสียหายต่ออาคารบ้านเรือน บดบังทัศนวิสัย จนทำให้เกิดอุปสรรคในการคมนาคมขนส่ง

ฝุ่นละอองในอากาศแบ่งออกเป็น 3 ช่วงขนาด ได้แก่ ฝุ่นรวม (Total suspended particulate) มีขนาดเส้นผ่านศูนย์กลางเล็กกว่า 100 ไมครอน ฝุ่นละอองขนาดไม่เกิน 10 ไมครอน (PM10) มีขนาดเล็กกว่าเส้นผม และฝุ่นละอองขนาดไม่เกิน 2.5 ไมครอน (PM2.5) มีขนาดเท่ากับเชื้อโรคจนไปถึงระดับโมเลกุล ซึ่งฝุ่นละอองที่มีขนาดเล็กจะตกสะสมได้ช้ากว่าขนาดใหญ่ คงอยู่ในอากาศได้นานกว่า และยังเข้าสู่ระบบทางเดินหายใจจนไปถึงชั้นถุงลมปอดได้ แต่เนื่องจากการศึกษาวิจัยพบว่าฝุ่นขนาดเล็กนั้นจะเป็นอันตรายต่อสุขภาพมากกว่าฝุ่นรวม ดังนั้น US.EPA จึงได้มีการยกเลิกค่ามาตรฐานฝุ่นรวมและกำหนดค่าฝุ่นขนาดเล็กเป็น 2 ชนิด คือ ฝุ่นละอองขนาดเล็กกว่า 10 ไมครอน (PM10) และฝุ่นละอองที่มีขนาดเล็กกว่า 2.5 ไมครอน (PM2.5)

PM10 ตามคำจำกัดความของ US.EPA หมายถึง ฝุ่นหยาบ (Course particle) เป็นอนุภาคที่มีเส้นผ่านศูนย์กลาง 2.5 - 10 ไมครอน มีแหล่งกำเนิดจากการจราจรบนถนนที่ไม่ได้ลาดยาง

PM2.5 ตามคำจำกัดความของ US.EPA หมายถึง ฝุ่นละเอียด (Fine particle) เป็นอนุภาคที่มีเส้นผ่านศูนย์กลางเล็กกว่า 2.5 ไมครอน ฝุ่นละเอียดมีแหล่งกำเนิดจากควันเสียของรถยนต์ โรงไฟฟ้า โรงงานอุตสาหกรรม และควันที่เกิดจากหุงต้มอาหารโดยใช้ฟืน [3,4]

2.1.2 ผลกระทบของฝุ่นละออง

1. ผลกระทบต่อความสามารถในการมองเห็น

ฝุ่นละอองทุกขนาดในบรรยากาศนั้นมีผลโดยตรงต่อการปิดกั้นทางเดินของแสง ถ้ามีปริมาณฝุ่นละอองสูงจะทำให้ระยะในการมองเห็นลดลง ดังรูปที่ 2.1 อันตรกิริยาของแสงกับฝุ่นละอองมีด้วยการ 2 แบบคือแบบการกระเจิง และการสะท้อนกลับหมด เราสามารถมองเห็นตึกได้จากที่ระยะไกลในวันที่แสงแดดจัด แสงแดด

จะสะท้อนมายังตาของเราทำให้เห็นตึกชัดเจน แต่ในวันที่มีปริมาณฝุ่นละอองมาก แสงแดดจะถูกฝุ่นละอองสะท้อนกลับ หรือเกิดการกระเจิง ทำให้เรามองไม่เห็นอาคาร

การสะท้อนกลับหรือเกิดการกระเจิงขึ้นอยู่กับอัตราส่วนระหว่างเส้นผ่านศูนย์กลางของอนุภาคกับค่าความยาวช่วงคลื่นของแสง ถ้าขนาดอนุภาคใหญ่กว่าความยาวคลื่น แสงจะถูกสะท้อนกลับหมด และถ้าขนาดของอนุภาคเล็กกว่าความยาวคลื่น แสงก็จะส่องผ่านไปได้ แต่อนุภาคเล็กขนาดใกล้เคียงกับความยาวคลื่น จะเกิดการกระเจิงของแสง

2. ผลกระทบต่อสุขภาพของมนุษย์

ฝุ่นละอองขนาดเล็กจะมีผลกระทบต่อสุขภาพเป็นอย่างมาก เมื่อหายใจเข้าไปในปอดจะเข้าไปอยู่ในระบบทางเดินหายใจส่วนล่าง ในสหรัฐอเมริกาพบว่าผู้ที่ได้รับฝุ่น PM10 ในระดับหนึ่งจะทำให้เกิดโรคหอบหืด และฝุ่น PM2.5 ในบรรยากาศจะมีความสัมพันธ์กับอัตราการเพิ่มของผู้ป่วยที่เป็นโรคหัวใจและโรคปอดที่เข้ามารักษาตัวในห้องฉุกเฉิน ลดประสิทธิภาพการทำงานของปอด และเกี่ยวข้องกับความเสี่ยงชีวิตก่อนวัยอันควร โดยเฉพาะผู้ป่วยสูงอายุ ผู้ป่วยโรคหัวใจ ผู้ป่วยโรคหอบหืด และเด็กจะมีอัตราเสี่ยงสูงกว่าคนปกติ



รูปที่ 2.1 ฝุ่นละอองจำนวนมากบดบังการมองเห็นอาคารต่าง ๆ

<https://thairesidents.com/local/see-not-fog-pollution-bangkok/>

2.1.3 อุณหภูมิ (Temperature)

อุณหภูมิ คือ การวัดค่าเฉลี่ยของพลังงานจลน์ของอนุภาคในสสารใด ๆ ซึ่งสอดคล้องกับความร้อนหรือเย็นของสสารนั้น

2.1.4 ความชื้น (Humidity)

ความชื้น คือ ปริมาณไอน้ำที่มีอยู่ในอากาศบริเวณใดบริเวณหนึ่ง โดยความชื้นของอากาศมีการเปลี่ยนแปลงอยู่ตลอดเวลา จะมากหรือน้อยขึ้นอยู่กับความดันและอุณหภูมิ [5]

2.1.5 ความชื้นสัมพัทธ์ (Relative humidity)

“อัตราส่วนของ ปริมาณไอน้ำที่มีอยู่จริงในอากาศ ต่อ ปริมาณไอน้ำที่จะทำให้อากาศอิ่มตัว ณ อุณหภูมิเดียวกัน” ค่าความชื้นสัมพัทธ์แสดงในรูปของร้อยละ (%) เขียนเป็นสูตรได้ดังนี้

$$\text{ความชื้นสัมพัทธ์} = \frac{\text{ปริมาณไอน้ำที่อยู่ในอากาศ}}{\text{ปริมาณไอน้ำที่ทำให้อากาศอิ่มตัว}} \times 100\% \quad (1)$$

2.1.6 ตัวรับรู้ (Receiver)

ตัวรับรู้ คือ อุปกรณ์ตรวจจับสัญญาณหรือปริมาณทางด้านต่าง ๆ เช่น อุณหภูมิ แสง เสียง เป็นต้น จากนั้น จะทำหน้าที่เปลี่ยนเป็นสัญญาณออกหรือปริมาณเอาต์พุตที่ได้จากการวัด เพื่อที่จะสามารถนำไปคำนวณหรือวิเคราะห์ผลต่อได้

โดยการวัดปริมาณฝุ่นละออง เป็นการวัดการกระเจิงของแสงที่กระทบกับอนุภาคในอากาศ แล้วไปโดนกับฉาก รับแสง จึงเป็นที่มาของการศึกษาการกระเจิงของแสง

ลักษณะของการกระเจิงแบ่งออกได้เป็น 3 แบบ โดยแบ่งแยกตามอัตราการกระเจิง

1. การกระเจิงแบบเรลห์ (Rayleigh scattering) เป็นกรณีที่ขนาดของอนุภาคในอากาศเล็กกว่าขนาดของความยาวคลื่นมาก โดยเส้นผ่านศูนย์กลางของอนุภาคจะประมาณ 0.0001-0.001 μm และอัตราการกระเจิง (α) $\ll 1$ สามารถหาอัตราการกระเจิงได้จากสูตรดังนี้

$$\alpha = \frac{2\pi r}{\lambda} \quad (2)$$

โดยที่ α คือ อัตราการกระเจิง

r คือ รัศมีของอนุภาค

λ คือ ความยาวคลื่น

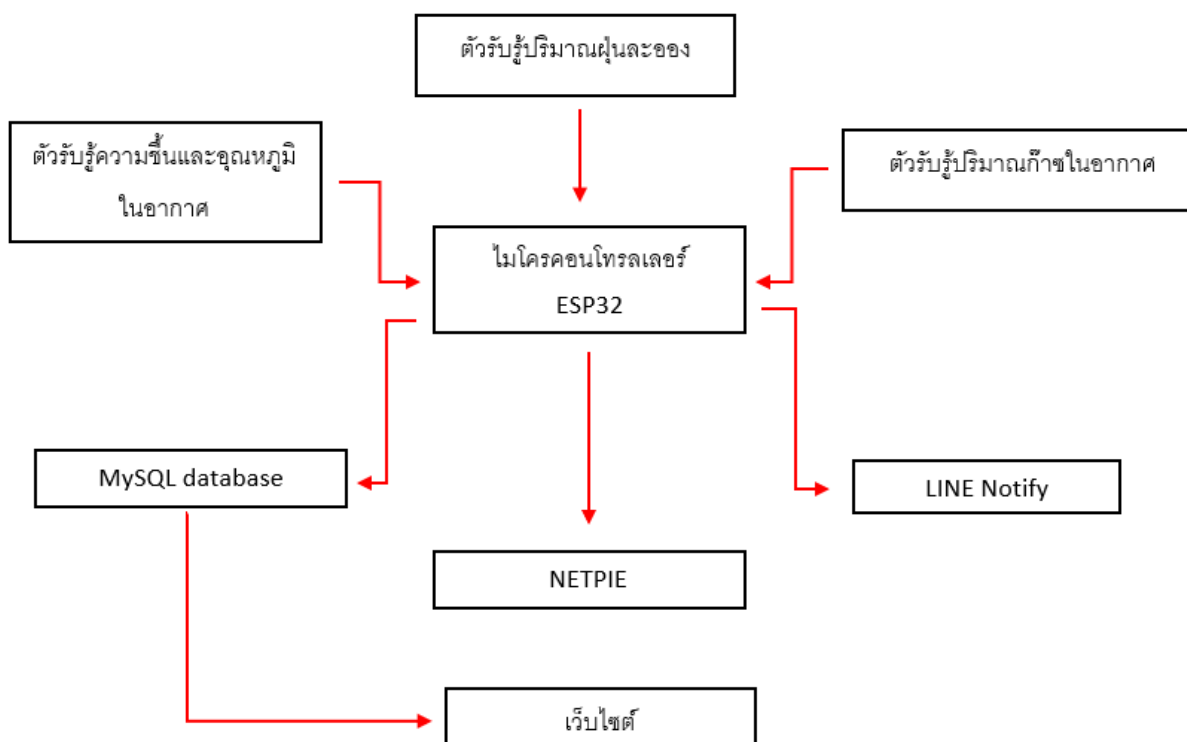
2. การกระเจิงแบบเมีย (Mie scattering) เป็นกรณีที่ขนาดของอนุภาคในอากาศใกล้เคียงหรือใหญ่กว่าขนาดของความยาวคลื่นไม่มากนัก ($\sim 0.1-10$ เท่า) โดยเส้นผ่านศูนย์กลางของอนุภาคจะประมาณ 0.01-10 μm และอัตราการกระเจิง (α) ประมาณ 1 อนุภาคที่สำคัญในกลุ่มนี้ คือ ละอองน้ำกับฝุ่นละออง

3. การกระเจิงแบบเรซาคณิต (Geometric scattering) เป็นกรณีที่ขนาดของอนุภาคใหญ่กว่าขนาดของความยาวคลื่นมาก (ประมาณ >10 เท่า) โดยเส้นผ่านศูนย์กลางของอนุภาคจะประมาณ 10-100 μm ในกรณีนี้อัตราการกระเจิงจะไม่ขึ้นกับขนาดความยาวคลื่นอีกต่อไป แต่จะมีค่าคงที่สำหรับทุกความยาวคลื่นเรียกว่าเป็น Non-selective scattering และอัตราการกระเจิง (α) $\gg 1$ [6]

2.2 อุปกรณ์ที่ใช้

อุปกรณ์ที่นำมาใช้ในระบบตรวจวัด PM2.5 ด้วยระบบสมองกลฝังตัวได้แก่

- ไมโครคอนโทรลเลอร์ ESP32 ใช้ในการประมวลผลและควบคุมการทำงานของอุปกรณ์ที่นำมาเชื่อมต่อ
- ตัวรับรู้ความชื้นสัมพัทธ์และอุณหภูมิ (DHT22) ใช้ในการวัดความชื้นสัมพัทธ์ และอุณหภูมิของอากาศ ณ บริเวณนั้น
- ตัวรับรู้ปริมาณฝุ่นละออง (PMS7003) ใช้ในการวัดปริมาณฝุ่นละออง ณ บริเวณนั้น
- ตัวรับรู้ปริมาณก๊าซในอากาศ (MQ-135) ใช้ในการวัดปริมาณก๊าซคาร์บอนไดออกไซด์ ณ บริเวณนั้น
- NETPIE ใช้สำหรับแสดงค่าความชื้นสัมพัทธ์ของอากาศ อุณหภูมิ ปริมาณฝุ่นละออง ปริมาณก๊าซคาร์บอนไดออกไซด์ ที่วัดได้ในบริเวณที่ได้ติดตั้งเครื่องวัด
- MySQL database ใช้สำหรับเป็นที่เก็บข้อมูลปริมาณต่าง ๆ ที่วัดได้ ณ วันและเวลาที่ได้เปิดเครื่องวัด
- LINE Notify ใช้สำหรับแจ้งเตือนค่าปริมาณฝุ่นละออง PM2.5 ที่เกินเกณฑ์ที่ได้กำหนดไว้
- เว็บไซต์ ใช้สำหรับทำการเฝ้าระวังปริมาณ PM 2.5 ได้หลาย ๆ สถานที่ในเวลาเดียวกัน

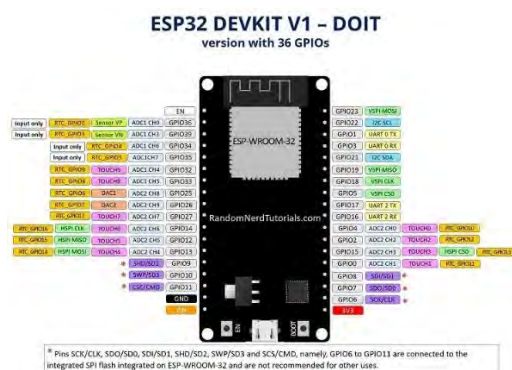


รูปที่ 2.2 แผนภาพแสดงการทำงานและองค์ประกอบของระบบสมองกลฝังตัว

2.2.1 ESP32

ESP32 เป็นชิปไอซีไมโครคอนโทรลเลอร์ 32 บิต ที่สามารถเชื่อมต่อ Wi-Fi ได้ ซึ่งเป็นรุ่นต่อของชิปไอซี ESP8266 รองรับการเขียนโปรแกรมโดยใช้โปรแกรม Arduino IDE และรองรับชุดคำสั่งส่วนใหญ่ของ Arduino IDE ทำให้สามารถใช้งานได้ง่าย

ESP32 ทำงานที่แรงดันไฟฟ้า 3.3 V – 5 V ดังนั้นการนำไปใช้งานร่วมกับตัวรับรู้อื่น ๆ ที่ใช้แรงดันเกิน 5 V ต้องใช้วงจรแบ่งแรงดันมาช่วย เพื่อไม่ให้โมดูลพังเสียหาย
ขาของ Node MCU แบ่งได้ดังนี้



รูปที่ 2.3 การเชื่อมต่อขาของ Node MCU

<https://randomnerdtutorials.com/esp32-pinout-reference-gpios/>

- Vin เป็นขาสำหรับจ่ายไฟเข้าไป เพื่อให้โมดูลทำงานได้ ซึ่งแรงดันที่ใช้งานได้คือ 3.3 V – 5 V
- GND เป็นขา ground หรือขาที่ต่อสายดิน
- GPIO เป็นขาดิจิตอลอินพุตหรือเอาต์พุต ทำงานที่แรงดัน 3.3 V
- ADC หรือ analog to digital converter เป็นตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล [7-9]

2.2.2 ตัวรับรู้ปริมาณฝุ่นละออง (PMS7003)

PMS7003 เป็นตัวรับรู้ปริมาณฝุ่นละอองในอากาศ และให้ค่าแบบดิจิตอลที่ใช้กันอย่างแพร่หลาย ลักษณะสำคัญของตัวรับรู้ PMS7003 คือ ให้ข้อมูลแบบ Real time และเส้นผ่านศูนย์กลางของอนุภาคขั้นต่ำที่แยกแยะได้ คือ 0.3 ไมโครเมตร

หลักการทำงาน

หลักการทำงานของตัวรับรู้ปริมาณฝุ่นละออง ประการแรกคือพัดลมดูดอากาศทำหน้าที่ดูดอากาศโดยรอบเข้ามาภายในตัวรับรู้ ประการที่สองคือห้องตรวจจับ ภายในนั้นจะมี Laser source ทำหน้าที่ปล่อยแสงออกมา

ให้ไปกระทบกับอนุภาคแล้วเกิดการกระเจิงของแสง ประการที่สามแสงที่กระเจิงจะโดนตรวจจับโดย Photodiode ที่มุมที่เฉพาะเจาะจง ซึ่งจะสร้างพัลส์ตามสัดส่วนกับความเข้มข้นของแสงที่กระเจิงออกมา แล้ววาดกราฟแสดงความสัมพันธ์ระหว่างความเข้มของแสงที่กระเจิงกับเวลาขึ้นมา แอมพลิจูดของกราฟจะขึ้นกับขนาดของอนุภาค [10]

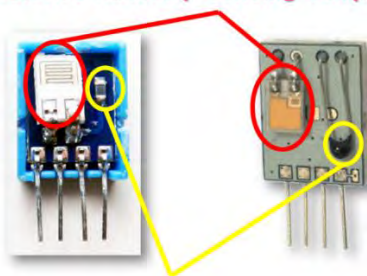
2.2.3 ตัวรับรู้ความชื้นสัมพัทธ์และอุณหภูมิในอากาศ (DHT 22)

หลักการทำงาน

ตัวรับรู้ความชื้นสัมพัทธ์และอุณหภูมิในอากาศ ใช้วัดความชื้นสัมพัทธ์และอุณหภูมิ และให้สัญญาณเอาต์พุตแบบ Digital output โดยประกอบด้วยการทำงาน 2 ส่วน ได้แก่

1. Resistive humidity sensing component: ตัวรับรู้ความชื้นสัมพัทธ์ที่จะวัดการเปลี่ยนแปลงอิมพีแดนซ์ไฟฟ้าของตัวกลางดูดความชื้น การทำงานของตัวรับรู้ คือดูดซับไอน้ำและไอออนที่แตกตัว เป็นผลให้ค่าความนำไฟฟ้าของตัวกลางเพิ่มขึ้น ความถี่ในการวัด 1 Hz อ่านค่าได้วินาทีละครั้ง ความชื้น ที่สามารถวัดได้อยู่ในช่วง 0 ถึง 100 % มีความคลาดเคลื่อนอยู่ที่ $\pm 5\%$
2. NTC temperature sensor thermistor: เป็นตัวรับรู้ความต้านทานลดลงเมื่ออุณหภูมิเพิ่มขึ้น แต่มีการเปลี่ยนแปลงความต้านทานสูงมาก ตัวอย่างเช่น ที่อุณหภูมิ 0 °C NTC จะมีความต้านทาน 10 k Ω แต่ที่อุณหภูมิ 100 °C NTC จะมีความต้านทานลดลงเหลือเพียง 200 Ω เท่านั้น อุณหภูมิที่สามารถวัดได้อยู่ในช่วง - 40 ถึง 125 °C มีความคลาดเคลื่อน ± 0.5 °C

Resistive Humidity Sensing Component



NTC Temperature Sensor Thermistor

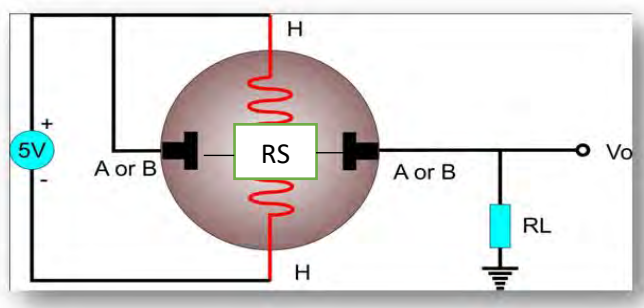
รูปที่ 2.4 โครงสร้างภายในของ DHT11/22

<https://embed58.learninginventions.org/g1-4/>

2.2.4 ตัวรับรู้ปริมาณก๊าซในอากาศ (MQ-135 – Air quality sensor)

MQ-135 เป็นตัวรับรู้ไว้ตรวจสอบคุณภาพของอากาศ โดยจะตรวจจับปริมาณก๊าซ NH_3 , NO_x , alcohol, Benzene, smoke และ CO_2 ในอากาศ

หลักการทำงาน



รูปที่ 2.5 โครงสร้างภายในของ MQ-135

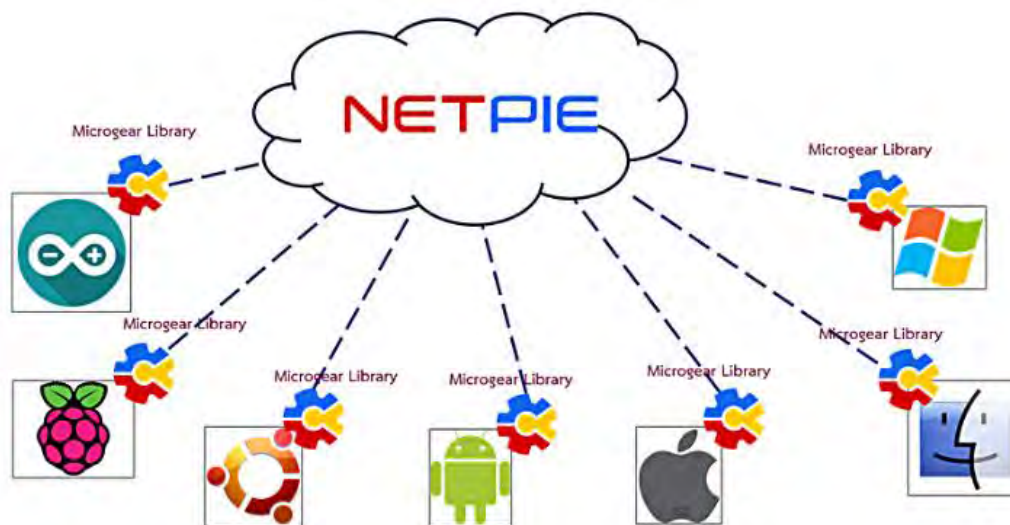
<https://blog.thaieasyelec.com/getting-started-gas-sensor/>

จากรูปที่ 2.5 เมื่อจ่ายไฟให้กับขา H จะทำให้เกิดความร้อนที่ขดลวดเพื่อให้สารเคมีภายในตัวรับรู้สามารถทำปฏิกิริยากับก๊าซพิษต่าง ๆ ได้ และเมื่อ MQ-135 ตรวจจับก๊าซพิษต่างได้ จะทำให้ค่าความต้านทานระหว่างขา A และ ขา B เปลี่ยนแปลง กล่าวคือเมื่อ MQ-135 ตรวจจับปริมาณก๊าซพิษต่าง ๆ ได้ จะทำให้ค่าความต้านทาน RS ลดลง ดังนั้นค่าความต้านทานแปรผกผันกับปริมาณของก๊าซพิษชนิดต่าง ๆ [11]

2.2.5 NETPIE

1. ข้อมูลทั่วไปเกี่ยวกับ NETPIE

NETPIE เป็น IoT (Internet of things) Cloud platform ที่พัฒนาขึ้นโดยทีมงานวิจัยและเปิดให้บุคคลทั่วไปใช้งาน โดยมี Web portal ที่ให้สามารถลงทะเบียนและจัดการตัวตนและสิทธิ์ของแอปพลิเคชันและอุปกรณ์ได้ที่เว็บไซต์ <https://netpie.io> ตั้งแต่เดือนกันยายน 2558 เป็นต้นมา NETPIE เป็น Middleware ที่มีหัวใจหลัก (นอกเหนือจากส่วนอื่นๆ) เป็น Distributed MQTT Brokers ซึ่งเป็นเสมือนจุดนัดพบให้สิ่งต่าง ๆ (Things) มาติดต่อสื่อสารและทำงานร่วมกันผ่านวิธีการส่งข้อความแบบ Publish/Subscribe NETPIE มีโครงสร้างเป็นคลาวด์ ทำให้สามารถขยายตัวได้อย่างอัตโนมัติ (Auto-scale) สามารถดูแลและซ่อมแซมตัวเองได้อัตโนมัติเมื่อส่วนหนึ่ง ส่วนใดในระบบมีปัญหา (Self-healing) โดยไม่ต้องพึ่งผู้ดูแลระบบ การบริหารจัดการระบบ เป็นแบบ Plug-and-Play ไม่ต้อง Configure หรือปรับแต่ง ในฝั่งอุปกรณ์ NETPIE มี Client library ซึ่งทำหน้าที่สร้างและดูแลช่องทางสื่อสารระหว่างอุปกรณ์กับ NETPIE รวมไปถึงรักษาความปลอดภัยในการส่งข้อมูล โมเดลการสื่อสารของ NETPIE แสดงไว้ในรูปที่ 2.6



รูปที่ 2.6 วิธีการสื่อสารของสิ่งต่างๆ ผ่าน NETPIE

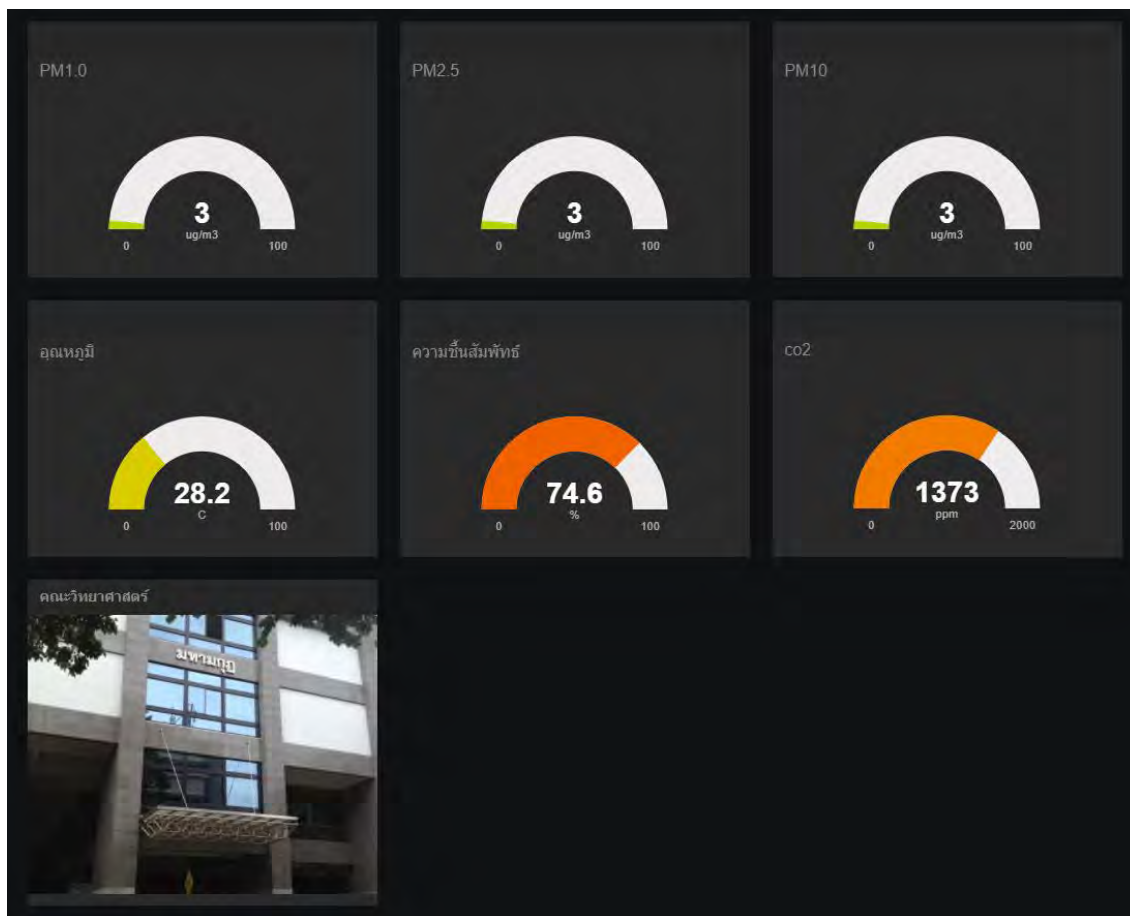
<https://netpie.gitbooks.io/doc/content/chapter1.html>

2. ประโยชน์ของ NETPIE

ช่วยลดการใช้ทรัพยากรของการเชื่อมต่อ NETPIE ช่วยให้อุปกรณ์สามารถสื่อสารกันได้โดยผู้ใช้งานไม่ต้องกังวลว่าอุปกรณ์นั้นจะอยู่ที่ใด เพียงแค่นำ Library ไปติดตั้งในอุปกรณ์ NETPIE จะรับหน้าที่ดูแลเชื่อมต่อให้ไม่ว่าอุปกรณ์นั้นจะอยู่ในเครือข่ายชนิดใด ลักษณะใด หรือแม้กระทั่งเคลื่อนย้ายไปอยู่ที่ใด ไม่เพียงเท่านั้น NETPIE ยังช่วยให้การเริ่มต้นใช้งานเป็นไปได้โดยง่าย โดยออกแบบให้อุปกรณ์ถูกค้นพบและเข้าสู่บริการโดยอัตโนมัติ (Automatic discovery) [12]

2.2.6 Dashboard

Dashboard เปรียบเสมือนหน้ากระดานที่ใช้ในการสรุปข้อมูลในมุมมองต่าง ๆ เพื่อให้สามารถดูข้อมูลได้ง่าย และใช้เวลาในการตีความสั้น Dashboard ใช้ในการติดตามข้อมูลที่เราสนใจ เพื่อมองการเปลี่ยนแปลงของข้อมูลได้ตลอดเวลา โดยโปรแกรมที่ใช้ในการทำ Dashboard ในโครงการนี้คือ NETPIE Freeboard [13]



รูปที่ 2.7 หน้าแสดงผลใน NETPIE Freeboard

2.2.7 MySQL

MySQL คือ โปรแกรมระบบจัดการฐานข้อมูล ที่พัฒนาโดยบริษัท MySQL AB มีหน้าที่เก็บข้อมูล อย่างเป็นระบบ MySQL เป็นเครื่องมือสำหรับเก็บข้อมูล ที่ต้องใช้ร่วมกับเครื่องมือหรือโปรแกรมอื่น เพื่อให้ได้ระบบงานที่รองรับความต้องการของผู้ใช้ เช่นทำงานร่วมกับเครื่องบริการเว็บ (Web Server) เป็นต้น โปรแกรมถูกออกแบบให้สามารถทำงานได้บนระบบปฏิบัติการที่หลากหลาย และเป็นระบบฐานข้อมูลแบบโอเพนซอร์ส (Open source) ที่ถูกนำไปใช้งานมากที่สุด [14]



รูปที่ 2.8 ฐานข้อมูล MySQL

http://www.networks365.net/indexac6b.html?route=product/product&product_id=110

2.2.8 LINE Notify

LINE Notify คือ บริการทาง Line ที่ได้เตรียมไว้ให้ในรูปแบบของ API สามารถนำไปใช้ต่อยอดพัฒนาโครงการที่มีความต้องการที่จะส่งข้อความไปยังกลุ่มหรือบัญชีส่วนตัวของเราได้ [15]

2.2.9 ภาษา HTML

HTML ย่อมาจาก hypertext markup language เป็นภาษาคอมพิวเตอร์รูปแบบหนึ่ง ที่มีโครงสร้างการเขียนโดยอาศัยตัวกำกับ (TAG) ควบคุมการแสดงผลข้อความ รูปภาพ ผ่านโปรแกรมเบราว์เซอร์ แต่ละ Tag อาจจะมีส่วนขยาย เรียกว่า Attribute สำหรับระบุ หรือควบคุมการแสดงผลของเว็บไซต์ได้

คำสั่งหรือ Tag ที่ใช้ในภาษา HTML ประกอบไปด้วยเครื่องหมายน้อยกว่า (<) ตามด้วย ชื่อคำสั่งและต่อท้ายด้วยเครื่องหมายมากกว่า (>) เป็นส่วนที่ทำหน้าที่ตกแต่ง ข้อความ เพื่อการแสดงผลข้อมูล โดยทั่วไปคำสั่งของ HTML ส่วนใหญ่จะอยู่เป็นคู่ มีเพียงบางคำสั่งเท่านั้นที่มีรูปแบบคำสั่งอยู่เพียงตัวเดียว ในแต่ละคำสั่งจะมีคำสั่งเปิดและปิด คำสั่งปิดของแต่ละคำสั่งจะมีรูปแบบเหมือนคำสั่งเปิด เพียงแต่จะเพิ่มเครื่องหมาย / (Slash) นำหน้าคำสั่งปิด ยกตัวอย่างคำสั่งที่ใช้ได้การสร้างตารางบนเว็บไซต์ [16,17]

ในการสร้างตารางนั้นจะมีคำสั่งหรือป้ายระบุ (Tag) ต่างๆ ดังต่อไปนี้

| Tag | คำอธิบาย |
|--------------------------|--|
| <table> ... </table> | คำสั่งในการสร้างตาราง |
| <caption> ... </caption> | คำสั่งในการกำหนดข้อความกำกับตาราง |
| <tr> ... </tr> | คำสั่งในการกำหนดแถวของตาราง 1 แถว (แนวนอน) |
| <th> ... </th> | คำสั่งในการกำหนดส่วนหัวของตารางแถวแรก |
| <td> ... </td> | คำสั่งในการกำหนดส่วนของคอลัมน์ (แนวตั้ง) |

2.2.10 ภาษาจาวาสคริปต์ (JavaScript)

จาวาสคริปต์ คือ ภาษาคอมพิวเตอร์ภาษาหนึ่งที่มีไว้เพื่อจัดทำเอฟเฟคหรือพฤติกรรมของหน้าเว็บไซต์ นอกเหนือจาก HTML ที่ใช้จัดการเรื่องของเนื้อหาของเว็บไซต์ ทำให้เว็บไซต์มีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น JavaScript สามารถช่วยจัดการงานต่าง ๆ โดยไม่จำเป็นต้องทำการโหลดหน้าเว็บไซต์ใหม่ซ้ำ การกดปุ่มหรือการนำเคอร์เซอร์เมาส์ไปยังบริเวณที่ต้องการ ทำให้ปรากฏข้อความโดยไม่จำเป็นต้องโหลดหน้าใหม่อีกครั้ง มีการเปลี่ยนสีของตัวอักษร หรือเปลี่ยนแปลงเนื้อหา นั่นคือหน้าเว็บไซต์นั้นได้ถูกเรียกใช้ด้วยภาษา JavaScript [18]

ในโครงการนี้ได้ใช้ภาษา JavaScript แสดงส่วนข้อความที่ต้องการได้ กล่าวคือเมื่อนำเคอร์เซอร์เมาส์ไปยังบริเวณที่ต้องการให้ปรากฏค่าสูงสุดต่ำสุดของปริมาณ PM 2.5 ดังแสดงในรูปที่ 2.9 และเมื่อนำเคอร์เซอร์เมาส์ออกยังบริเวณที่ต้องการ ทำให้ข้อความค่าสูงสุดต่ำสุดปริมาณ PM 2.5 หายไป ดังแสดงในรูปที่ 2.10 การเพิ่ม JavaScript เข้าไปในโครงการนี้เพื่อให้สะดวกและง่ายต่อการติดตามค่าปริมาณฝุ่นละอองในหน้าเว็บไซต์เดียวกัน

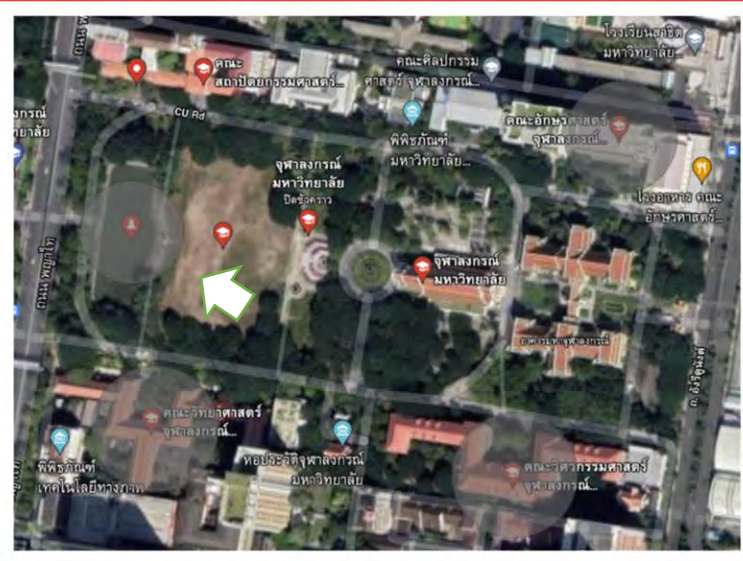
คลิกที่คณะวิทยาศาสตร์ คณะวิศวกรรมศาสตร์ คณะอักษรศาสตร์ หรือบริเวณสระน้ำ เพื่อเช็คค่าปริมาณฝุ่นละออง :

บริเวณสระน้ำฟ้าๆ มีปริมาณ PM2.5 สูงสุดอยู่ที่ 9 ug/m³ ต่ำสุดอยู่ที่ 1 ug/m³



รูปที่ 2.9 ภาพตัวอย่างการปรากฏของค่าปริมาณฝุ่นละออง เมื่อนำเคอร์เซอร์เมาส์ไปยังบริเวณที่ต้องการ

คลิกที่คณะวิทยาศาสตร์ คณะวิศวกรรมศาสตร์ คณะอักษรศาสตร์ หรือบริเวณสระน้ำ เพื่อเช็คค่าปริมาณฝุ่นละออง :



รูปที่ 2.10 ภาพตัวอย่างการหายไปของค่าปริมาณฝุ่นละออง เมื่อนำเคอร์เซอร์เมาส์ออกจากบริเวณที่ต้องการ

บทที่ 3

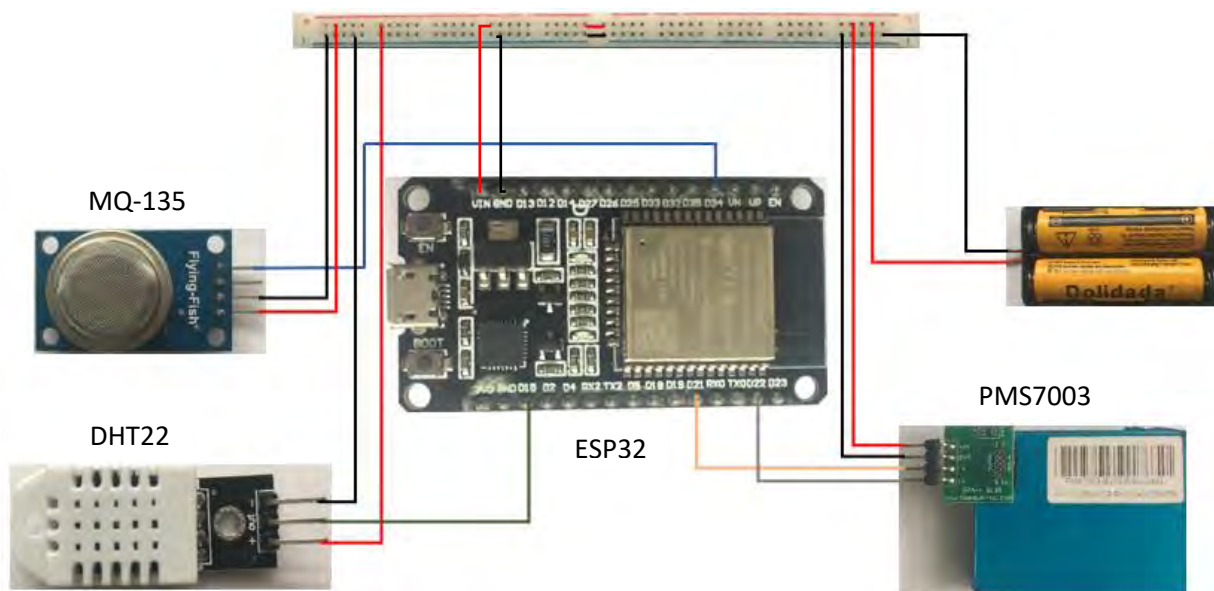
การออกแบบการทำงานและขั้นตอนการทำงาน

ขั้นตอนการดำเนินงานแบ่งออกเป็น 9 ขั้นตอน ดังนี้

1) ศึกษาการทำงานของอุปกรณ์

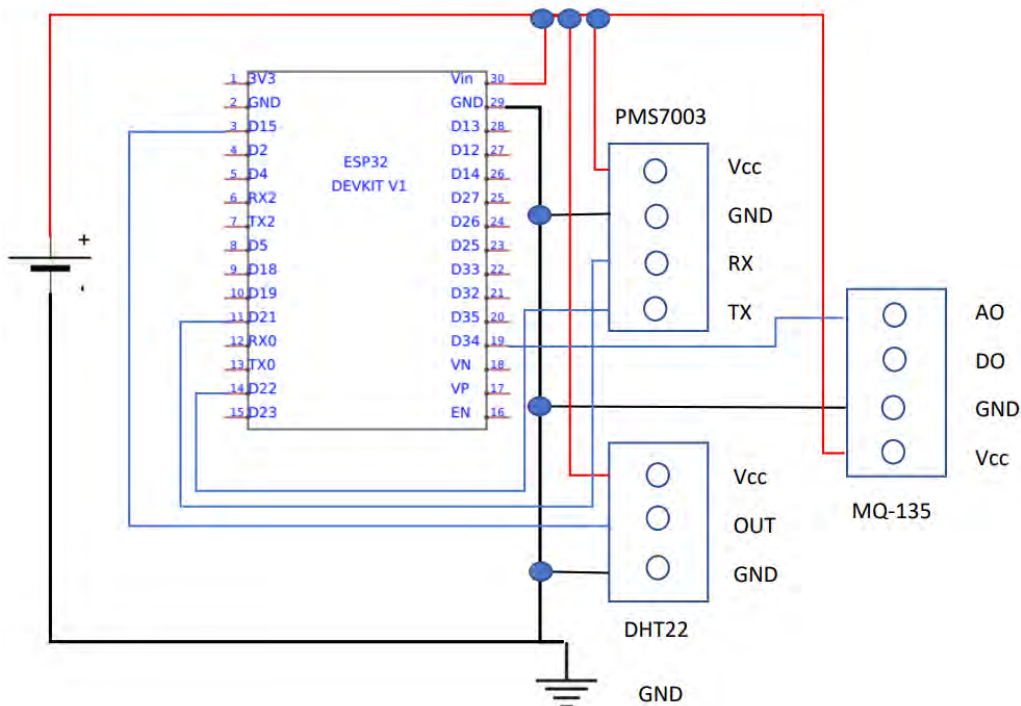
- 1.1 ศึกษาการทำงานของไมโครคอนโทรลเลอร์
- 1.2 ศึกษาการทำงานของตัวรับรู้ปริมาณต่าง ๆ
- 1.3 เขียนโปรแกรมออกแบบอัลกอริทึม
- 1.4 ศึกษาการเขียนโปรแกรมเพื่อฝังตัวบนไมโครคอนโทรลเลอร์
- 1.5 ศึกษาการเขียนเว็บไซต์ด้วยภาษา HTML

2) ออกแบบวงจรควบคุมการทำงาน



รูปที่ 3.1 วงจรของระบบตรวจวัด PM2.5 ด้วยระบบสมองกลฝังตัว

รูปที่ 3.1 แสดงถึงวงจรของระบบตรวจวัด PM2.5 ด้วยระบบสมองกลฝังตัว โดยมีไมโครคอนโทรลเลอร์ (ESP32) เป็นตัวรับข้อมูลจากตัวรับรู้ปริมาณฝุ่นละออง (PMS7003) ตัวรับรู้ความชื้นสัมพัทธ์และอุณหภูมิในอากาศ (DHT 22) และตัวรับรู้ปริมาณก๊าซในอากาศ (MQ-135) แล้วนำมาเชื่อมต่อกับถ่าน เพื่อเป็นแหล่งจ่ายไฟให้กับระบบนี้



รูปที่ 3.2 การเชื่อมต่ออุปกรณ์ของระบบตรวจวัด PM2.5 ด้วยระบบสมองกลฝังตัว

รูปที่ 3.2 แสดงถึงการเชื่อมต่ออุปกรณ์ของระบบตรวจวัด PM2.5 ด้วยระบบสมองกลฝังตัว โดยมี ESP32 เชื่อมต่อกับตัวรับรู้ 3 ตัว ดังนี้ 1) ตัวรับรู้ปริมาณฝุ่นละออง (PMS7003) ขา RX และ TX ต่อเข้ากับ ESP32 ที่ตำแหน่ง D21 และ D22 ตามลำดับ 2) นำขาส่งข้อมูล (OUT) ของตัวรับรู้ความชื้นสัมพัทธ์และอุณหภูมิในอากาศ (DHT22) เชื่อมเข้ากับ ESP32 ที่ตำแหน่ง D15 3) นำขาส่งข้อมูล (AO) ของตัวรับรู้ปริมาณก๊าซในอากาศ (MQ-135) ต่อเข้ามากับ ESP32 ที่ตำแหน่ง D34 และลำดับสุดท้ายนำไมโครคอนโทรลเลอร์ (ESP32) ตัวรับรู้ปริมาณฝุ่นละออง ตัวรับรู้ความชื้นสัมพัทธ์และอุณหภูมิในอากาศ และตัวรับรู้ปริมาณก๊าซในอากาศเชื่อมเข้ากับแหล่งจ่ายไฟ (Vcc) และกราวด์ (GND)

3) เขียนโปรแกรมที่ทำให้อุปกรณ์ต่าง ๆ ทำงานได้ ในวงจรที่ออกแบบไว้ แสดงได้ดังรูปที่ 3.2

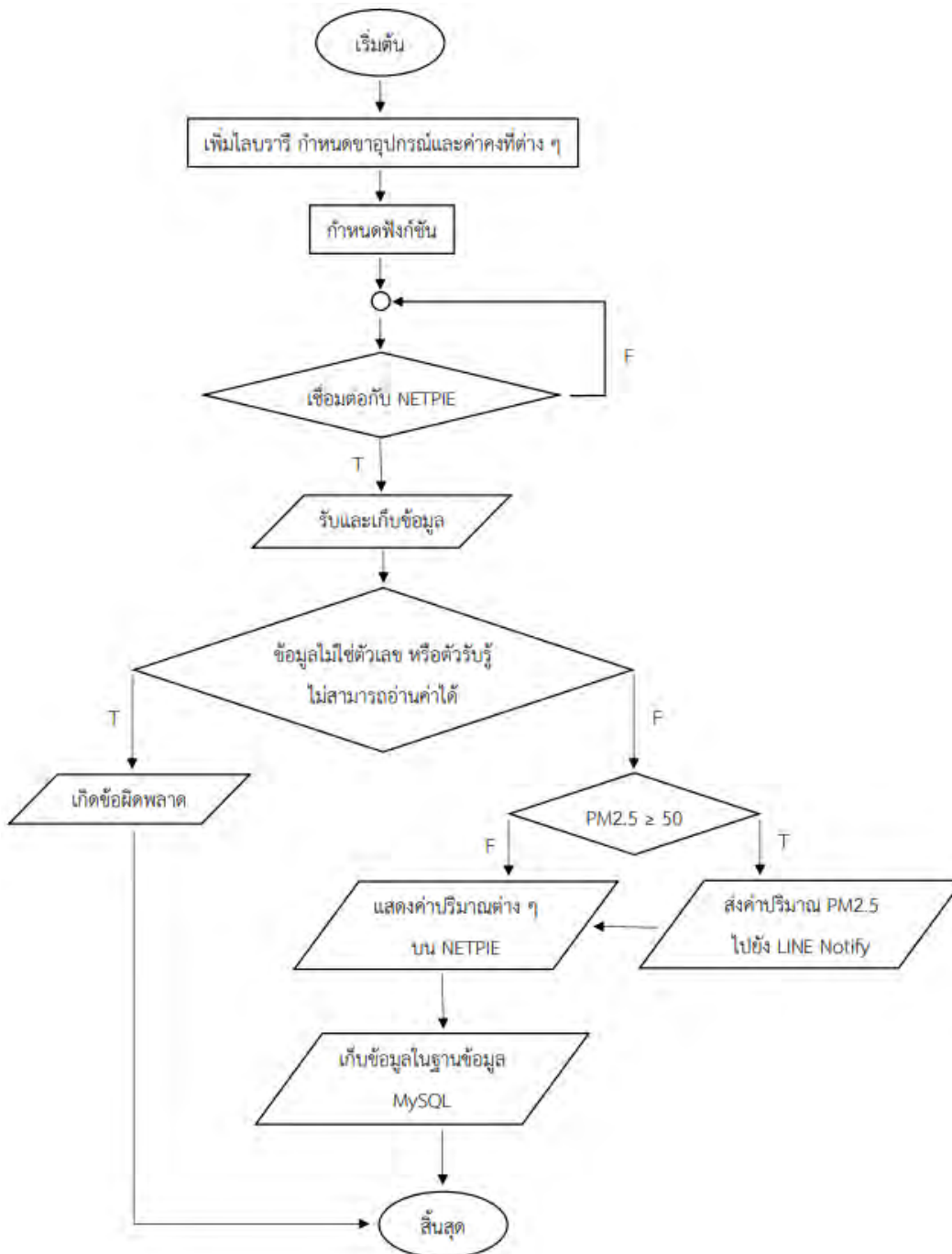
3.1 เพิ่มไลบรารีสำหรับ PMS, DHT, ESP32, NETPIE พร้อมทั้งกำหนดค่าคงที่ต่าง ๆ ชื่อและรหัสผ่านของเครือข่าย WiFi ที่ใช้และกำหนดขาที่ใช้ในการเชื่อมต่ออุปกรณ์

3.2 กำหนดฟังก์ชัน ที่ใช้ในการทำงานของโปรแกรม ได้แก่ ฟังก์ชันตรวจสอบการปรากฏของข้อความ และฟังก์ชันตรวจพบการเชื่อมต่ออุปกรณ์กับ NETPIE

3.3 ทำการเชื่อมต่อทำ NETPIE หากไม่สามารถเชื่อมต่อกับ NETPIE ได้ จะต้องทำการเชื่อมต่ออีกครั้ง เมื่อสามารถเชื่อมต่อกับ NETPIE ได้ ให้ตัวรับรู้อุณหภูมิความชื้น ตัวรับรู้ปริมาณฝุ่นละออง และตัวรับรู้ปริมาณ

ก๊าซคาร์บอนไดออกไซด์ ทำการส่งข้อมูลไปยังไมโครคอนโทรลเลอร์จากนั้นทำการเก็บค่าใส่ในตัวแปรหนึ่งเพื่อส่งไปยัง NETPIE และ MySQL

3.4 นำข้อมูลที่ได้มาตรวจสอบว่ารูปแบบของข้อมูลนั้นถูกต้องหรือไม่ (ข้อมูลที่ถูกต้อง คือ ข้อมูลที่เป็นตัวเลข และสำหรับกรณีข้อมูลอุณหภูมิและความชื้นสัมพัทธ์ในอากาศ ต้องมีค่าไม่เกิน 100 % และไม่ใช่อ) หากข้อมูลนั้นถูกต้องให้ทำการอ่านค่าปริมาณก๊าซคาร์บอนไดออกไซด์ และปริมาณฝุ่นละออง เป็นลำดับต่อมา



รูปที่ 3.3 ผังงานแสดงการทำงานของโปรแกรม

4) สร้างส่วนแสดงผลข้อมูลบน NETPIE

4.1 เข้าไปที่ <http://netpie.io/> ทำการสมัครสมาชิก จากนั้นเข้าสู่ระบบโดยเลือก Login NETPIE 2020 จากนั้นสร้างโครงการขึ้นใหม่โดยการกดเครื่องหมาย + -> ใส่ชื่อโครงการที่ต้องการในช่อง Name -> Create จากนั้นให้กดไปที่โครงการที่สร้าง

4.2 กดที่ Device List เพื่อสร้าง Device จากนั้นให้คัดลอก Key เพื่อนำไปใส่ในโปรแกรม

4.3 สร้างส่วนแสดงผลโดยการกดไปที่ Freeboard แล้วกดที่เครื่องหมาย + -> ใส่ชื่อ Freeboard -> Create จากนั้นให้กดไปที่ Freeboard ที่สร้าง

4.4 กดที่ Add เพิ่ม Data source ใหม่ เพื่อรับข้อมูลจาก Device ที่สร้างไว้ ตั้งชื่อ Data sources ใส่ Client ID, Token และ Secret (จากหน้า Device) -> ตั้งค่าเวลาของข้อมูลย้อนหลังที่จะแสดงใน Feed และความละเอียดของข้อมูล เช่น แสดงผลทั้งหมด 6 ชั่วโมง และรับค่าทุก ๆ 1 วินาที -> Save

4.5 คลิกไปที่ ADD Plane จะมีหน้าต่างตั้งขึ้นมาด้านล่าง คลิกที่เครื่องหมาย + ตรงขอบหน้าต่าง -> Select type ->

4.6 สำหรับข้อมูลอุณหภูมิความชื้นสัมพัทธ์ในอากาศ ปริมาณฝุ่นละอองต่าง ๆ ให้เลือก
- Gauge (แสดงผลแบบหน้าปัด) -> ตั้งชื่อ Title, Value (ค่าที่ต้องการแสดง), Units (หน่วยของสิ่งที่แสดง), ค่าต่ำสุดและค่าสูงสุด -> Save

หมายเหตุ สำหรับค่า Value ของ Gauge ใส่โค้ดดังนี้

อุณหภูมิ -> `datasources["PM"]["shadow"]["temperature"]`

ความชื้นสัมพัทธ์ -> `datasources["PM"]["shadow"]["humidity"]`

PM1 -> `datasources["PM"]["shadow"]["pm1"]`

PM2.5 -> `datasources["PM"]["shadow"]["pm2.5"]`

PM10 -> `datasources["PM"]["shadow"]["pm10"]`

4.7 ดูความสัมพันธ์ของปริมาณทั้งหมด ให้เลือก

- Feed View (แสดงผลแบบกราฟ) -> Title, Data source (ค่าที่ต้องการแสดง), ชื่อแกน X และแกน Y -> Save

หมายเหตุ สำหรับ Data source ใส่โค้ดดังนี้

Data source -> `datasources["PM"]["feed"]`

5) สร้างส่วนเก็บข้อมูลบน MySQL

5.1 เข้าไปที่ <https://www.apachefriends.org/index.html> เพื่อทำการดาวน์โหลดโปรแกรม XAMPP จากนั้นทำการติดตั้งโปรแกรม XAMPP ลงในเครื่อง

5.2 เปิดโปรแกรม XAMPP Control Panel จากนั้นให้ทำการ start Apache และ MySQL

5.3 ทดสอบการเป็น host server โดยการพิมพ์ในช่องค้นหาว่า localhost และคลิกไปที่ phpMyAdmin

5.4 เข้าไปที่โฟลเดอร์ที่ชื่อ xampp ที่ติดตั้งไว้ในเครื่อง จากนั้นเปิดโฟลเดอร์ที่ชื่อ htdocs แล้วทำการสร้างโฟลเดอร์ที่จะเป็นที่อยู่ของโปรแกรมที่เป็นตัวกลางในการเชื่อมต่อ MySQL กับคอมพิวเตอร์ของเรา และเป็นที่อยู่ของการดึงข้อมูลจาก ESP32 มาไปยัง host server (คอมพิวเตอร์ของผู้ทดลอง) แล้วส่งข้อมูลมายัง MySQL

5.5 เข้าไปที่ <https://code-writer.en.softonic.com/> เพื่อทำการดาวน์โหลดโปรแกรม Code Writer เพื่อที่จะทำการเขียนโปรแกรมในข้อ 5.4

5.6 เข้าไปที่ phpMyAdmin เพื่อทำการสร้าง Database คลิกไปที่ New -> สร้างชื่อ Database ของเรา ในช่อง Database name -> Create

5.7 คลิกที่ชื่อ Database ที่ตั้งไว้ในข้อ 5.6 -> ตั้งชื่อตารางในช่อง Name และ จำนวนข้อมูลที่ต้องการ ในช่อง Number of columns -> Go

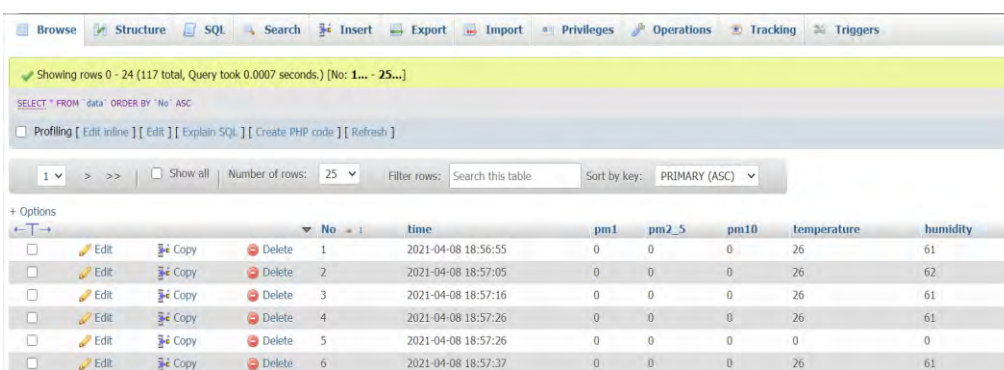
5.8 ตั้งชื่อแต่ละคอลัมน์ (Name) -> กำหนดประเภทของข้อมูล (Type) -> กำหนดค่าที่รับมาสามารถเป็น 0 ได้ไหม ถ้าได้ให้เลือก NULL ถ้าไม่ได้ให้เลือก None (Default)

หมายเหตุ - ถ้าเป็นจำนวนครั้งของข้อมูลให้คลิกที่ช่อง A_I Comments เพื่อให้ AI ช่วยเพิ่มจำนวนครั้งของข้อมูลแบบอัตโนมัติ

- ถ้าต้องการแสดงเวลาในการได้รับข้อมูล ให้กำหนดประเภทเป็น timestamp -> กำหนดค่าที่รับมาเป็น current_timestamp()

เมื่อตั้งคอลัมน์ที่ต้องการทั้งหมดเสร็จเรียบร้อยแล้ว ให้กด save

5.9 กดที่ Browse ก็จะได้เห็นตารางที่ตั้งชื่อไว้ในส่วนนี้จะเป็นที่แสดงของข้อมูล



| No | time | pm1 | pm2_5 | pm10 | temperature | humidity |
|----|---------------------|-----|-------|------|-------------|----------|
| 1 | 2021-04-08 18:56:55 | 0 | 0 | 0 | 26 | 61 |
| 2 | 2021-04-08 18:57:05 | 0 | 0 | 0 | 26 | 62 |
| 3 | 2021-04-08 18:57:16 | 0 | 0 | 0 | 26 | 61 |
| 4 | 2021-04-08 18:57:26 | 0 | 0 | 0 | 26 | 61 |
| 5 | 2021-04-08 18:57:26 | 0 | 0 | 0 | 0 | 0 |
| 6 | 2021-04-08 18:57:37 | 0 | 0 | 0 | 26 | 61 |

รูปที่ 3.4 ภาพตารางและคอลัมน์ต่าง ๆ ในโปรแกรม MySQL

6) สร้างส่วนแสดงผลข้อมูลบน LINE Notify

ขั้นตอนสมัครและขอ Token ของ LINE Notify มี 4 ขั้นตอนดังนี้

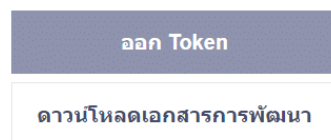
6.1 เพิ่ม LINE Notify เป็นเพื่อน โดยการสแกน QR Code นี้ หรือจาก <https://notify-bot.line.me/my/>



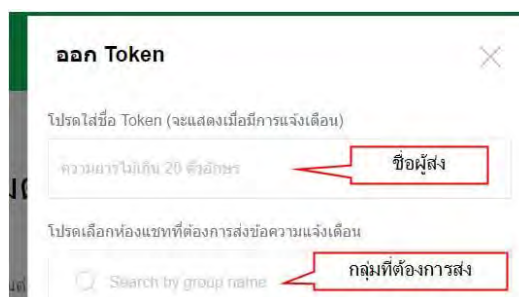
6.2 ล็อกอิน LINE เพื่อขอ Token ที่หน้า <https://notify-bot.line.me/my/> แล้วกดปุ่ม ออก Token

ออก Access Token (สำหรับผู้พัฒนา)

เมื่อใช้ Access Token แบบบุคคล จะสามารถตั้งค่าการแจ้งเตือนได้โดยไม่ต้องลงทะเบียนกับเว็บเซิร์ฟเวอร์



6.3 ตั้งค่าการใช้งาน LINE Notify กำหนดชื่อผู้ส่ง และเลือกกลุ่มที่ต้องการส่งข้อความไปหา



6.4 ขั้นตอนสุดท้าย รับ Token ให้กรอกชื่อที่ใช้ส่งข้อมูลเข้า Line แล้วเลือกกลุ่มที่จะส่งข้อมูล กดปุ่ม “ออก Token” จะได้ค่า Token ให้คัดลอกค่านี้ไว้ เพื่อนำไปใช้ในโปรแกรมของเรา



7) สร้างส่วนแสดงผลบนเว็บไซต์ ในการสร้างเว็บไซต์ จะแบ่งออกเป็น 2 ส่วน

- ส่วนที่ 1 ส่วนแสดงผล

7.1) สร้าง New Text Document

7.2) กำหนดคำสั่งต่าง ๆ โดยใช้คำสั่งดังต่อไปนี้

1. คำสั่งกำหนดลักษณะหัวเรื่อง: <Hn>.....</Hn> โดย n คือตัวเลขแสดงขนาดของตัวอักษร ค่าของ n นั้นจะมีค่าอยู่ที่ 1 – 6 ถ้า n=1 ขนาดตัวอักษรใหญ่สุด และ ถ้า n = 6 จะมีขนาดตัวอักษรเล็กสุด

2. คำสั่งแทรกรูปภาพในเอกสาร:

- ส่วนที่ 2 ส่วนเก็บข้อมูล

7.3) สร้าง New Text Document

7.4) กำหนดคำสั่งต่าง ๆ โดยใช้คำสั่งดังต่อไปนี้

1. คำสั่งกำหนดลักษณะหัวเรื่อง (Heading) : <Hn>.....</Hn>

2. คำสั่งตีเส้นบรรทัดหรือเส้นคั่นแนวนอน: <HR>

3. คำสั่งสำหรับจัดแบบเอกสารตามที่กำหนด: <PRE>.....</PRE>

4. คำสั่งกำหนดเส้นขอบของตาราง: <table border="n" bordercolor="ชื่อสี หรือ รหัสสี">

โดยที่ n หมายถึง ค่าตัวเลขที่ใช้กำหนดขนาดของเส้นขอบของตาราง ค่าตัวเลขยิ่งมากเส้นขอบตารางจะมีขนาดใหญ่เพิ่มขึ้น [19-22]

8) ทดสอบการส่งข้อมูลด้วยระบบสมองกลฝังตัว

9) วิเคราะห์ข้อมูลและสรุปผล

บทที่ 4

ผลการทดลอง

การออกแบบระบบตรวจวัด PM2.5 ด้วยระบบสมองกลฝังตัวเพื่อใช้ในการเก็บข้อมูลและแจ้งเตือนค่าปริมาณฝุ่นละออง PM2.5 ที่เกินเกณฑ์ที่ได้กำหนดไว้ ในโครงการนี้ใช้ชุดการวัดที่ประกอบด้วยไมโครโทรลเลอร์ (ESP32) เชื่อมต่อกับตัวรับรู้สัญญาณที่แปลงปริมาณทางฟิสิกส์ให้กลายเป็นสัญญาณทางไฟฟ้า ซึ่งตัวรับรู้สัญญาณนั้นได้แก่ PMS7003 DHT22 และ MQ-135 สำหรับโครงการนี้ได้ทำการวัดค่าปริมาณฝุ่นละอองที่บริเวณต่าง ๆ เพื่อใช้ในการทำเว็บไซต์เพื่อใช้ในการเฝ้าระวังปริมาณ PM 2.5 ได้หลาย ๆ สถานที่ในเวลาเดียวกัน และสามารถเก็บข้อมูลที่ได้จากตัวรับรู้ไว้ในฐานข้อมูลและแสดงผลได้แบบ Real time โดยบริเวณที่ได้ทำการเลือกมีดังนี้ คณะวิทยาศาสตร์ คณะวิศวกรรมศาสตร์ คณะอักษรศาสตร์ และบริเวณสระน้ำจุฬาฯ

4.1 ทำเว็บไซต์เพื่อดูค่าปริมาณฝุ่นละอองของบริเวณต่าง ๆ ภายในจุฬาลงกรณ์มหาวิทยาลัย

โดยทำการวัดค่าปริมาณฝุ่นละอองในบริเวณคณะวิทยาศาสตร์ คณะวิศวกรรมศาสตร์ คณะอักษรศาสตร์ และบริเวณสระน้ำจุฬาฯ ในวันที่ 18-19 พฤษภาคม พ.ศ. 2564 และใช้ภาษา HTML ในการทำเว็บไซต์ที่จะใช้ในการเฝ้าระวังปริมาณฝุ่นละอองได้ในหลาย ๆ จุดพร้อมกัน

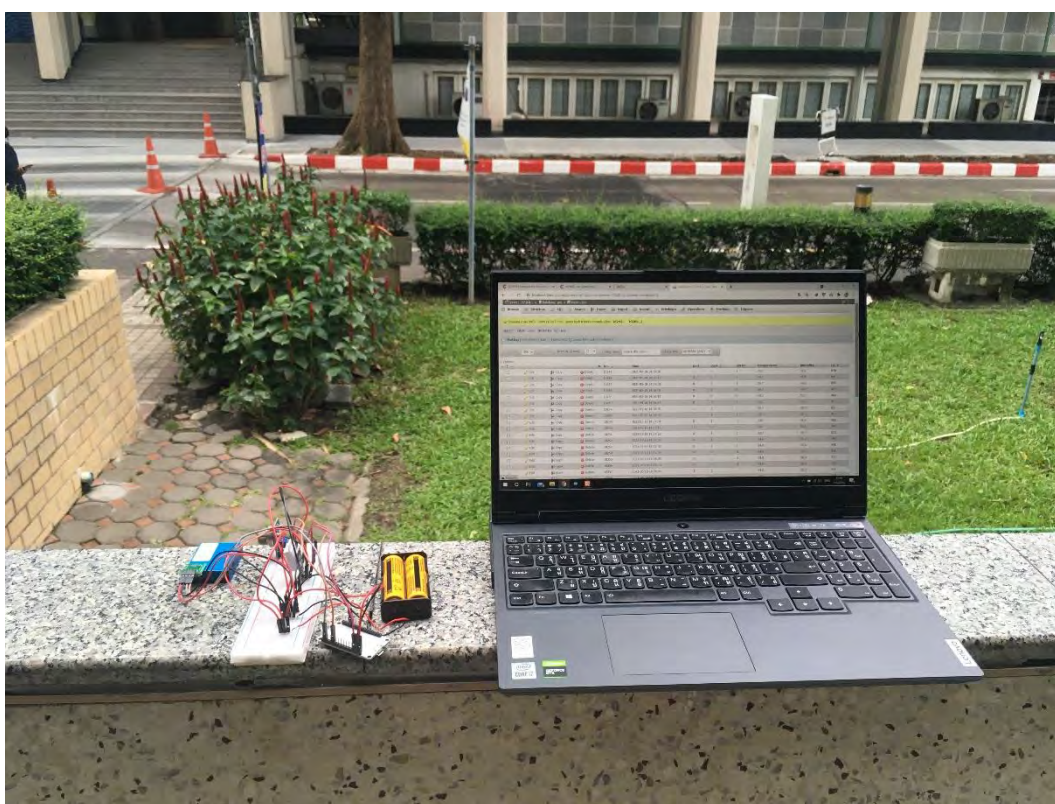
คลิกที่คณะวิทยาศาสตร์ คณะวิศวกรรมศาสตร์ คณะอักษรศาสตร์ หรือบริเวณสระน้ำ เพื่อเช็คค่าปริมาณฝุ่นละออง :



รูปที่ 4.1 หน้าเว็บไซต์แสดงถึงแผนที่ของจุฬาลงกรณ์มหาวิทยาลัย และบริเวณทั้ง 4 บริเวณที่สนใจ

จากรูปที่ 4.1 แสดงถึงแผนที่ของจุฬาลงกรณ์มหาวิทยาลัย และวงกลมสี่เทาทั้ง 4 จุดแสดงถึงบริเวณที่ได้ทำการเก็บข้อมูล เพื่อจะนำข้อมูลที่ได้มาใช้ในแสดงผล แต่เว็บไซต์ข้างต้นยังไม่สามารถปรากฏข้อความใด ๆ ได้เนื่องจากแต่ละตำแหน่งยังไม่มีฐานข้อมูลของตัวเอง จึงต้องทำการสร้างฐานข้อมูลของแต่ละบริเวณก่อน แล้วจึงนำมาเชื่อมโยงกับเว็บไซต์ข้างต้น

หลังจากที่ได้ทำการเขียนโปรแกรมสร้างเว็บไซต์เพื่อแสดงภาพของจุฬาลงกรณ์มหาวิทยาลัยจาก Google Maps ได้แล้ว ในขั้นตอนต่อมาเป็นการสร้างฐานข้อมูลของแต่ละบริเวณ โดยเริ่มจาก คณะวิทยาศาสตร์ ได้ทำการเก็บข้อมูลในวันอังคารที่ 18 พฤษภาคม พ.ศ. 2564 เวลา 14.00-15.00 หลังจากที่ได้ทำการเขียนหน้าเว็บไซต์แสดงฐานข้อมูลของบริเวณคณะวิทยาศาสตร์ จะได้หน้าเว็บไซต์แสดงดังรูปที่ 4.3



รูปที่ 4.2 บริเวณที่ติดตั้งเครื่องวัดปริมาณต่าง ๆ ณ คณะวิทยาศาสตร์

ปริมาณฝุ่นละอองในช่วงเวลา 14.00-15.00 ของวันที่ 18 พฤษภาคม พ.ศ. 2564 ณ คณะวิทยาศาสตร์



ค่า PM2.5

สูงสุดอยู่ที่ 4 ug/m³

ต่ำสุดอยู่ที่ 0 ug/m³

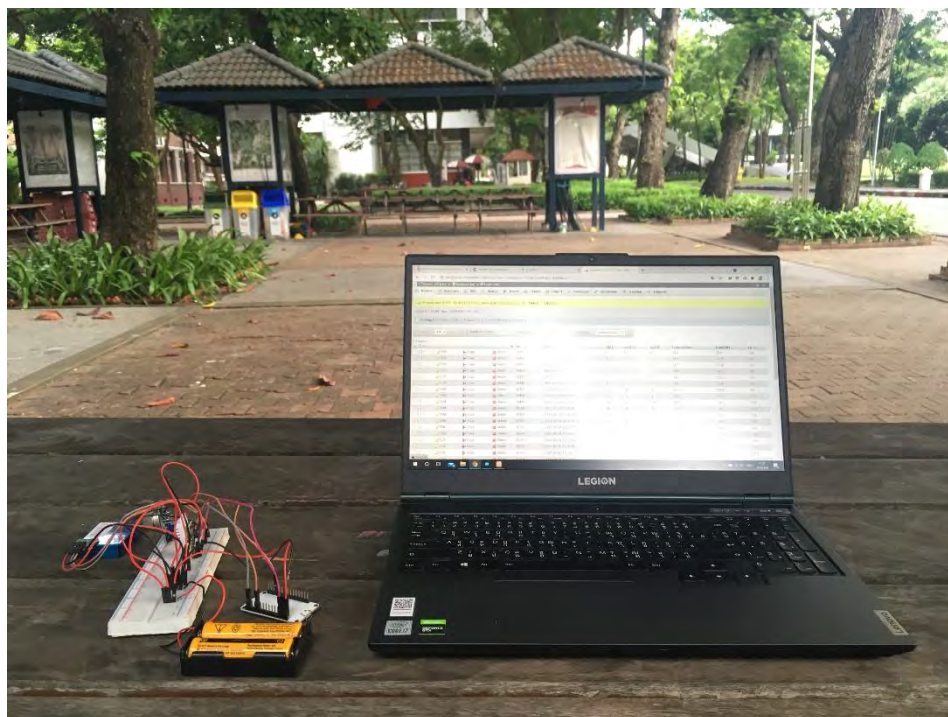
ตัวอย่างข้อมูล

| time | pm1 | pm2.5 | pm10 | อุณหภูมิ | ความชื้น | CO 2 |
|---------------------|-----|-------|------|----------|----------|------|
| 2021-05-18 14:30:08 | 0 | 0 | 0 | 34.4 | 56.3 | 832 |
| 2021-05-18 14:30:19 | 0 | 0 | 0 | 34.4 | 56.4 | 809 |
| 2021-05-18 14:30:29 | 0 | 0 | 0 | 34.4 | 56.5 | 848 |
| 2021-05-18 14:30:40 | 0 | 0 | 0 | 34.4 | 56.7 | 863 |
| 2021-05-18 14:30:51 | 0 | 1 | 1 | 34.3 | 57 | 770 |
| 2021-05-18 14:31:01 | 0 | 1 | 1 | 34.3 | 57.2 | 873 |
| 2021-05-18 14:31:12 | 1 | 1 | 1 | 34.3 | 57 | 790 |
| 2021-05-18 14:31:22 | 1 | 1 | 1 | 34.3 | 57.4 | 739 |
| 2021-05-18 14:31:33 | 0 | 1 | 1 | 34.3 | 57.5 | 768 |
| 2021-05-18 14:31:43 | 0 | 1 | 1 | 34.3 | 57.6 | 806 |
| 2021-05-18 14:31:54 | 0 | 1 | 1 | 34.3 | 57.6 | 793 |
| 2021-05-18 14:32:05 | 0 | 1 | 1 | 34.3 | 57.6 | 818 |
| 2021-05-18 14:32:15 | 0 | 1 | 1 | 34.3 | 57.7 | 826 |
| 2021-05-18 14:32:26 | 0 | 0 | 0 | 34.3 | 57.6 | 867 |
| 2021-05-18 14:32:36 | 0 | 1 | 1 | 34.4 | 57.6 | 880 |
| 2021-05-18 14:32:47 | 0 | 1 | 1 | 34.3 | 57.6 | 880 |
| 2021-05-18 14:32:58 | 0 | 1 | 1 | 34.4 | 57.5 | 846 |
| 2021-05-18 14:33:08 | 0 | 1 | 1 | 34.4 | 57.4 | 889 |
| 2021-05-18 14:33:19 | 0 | 1 | 1 | 34.4 | 57.2 | 801 |
| 2021-05-18 14:33:29 | 0 | 1 | 1 | 34.5 | 57.2 | 870 |
| 2021-05-18 14:33:40 | 0 | 0 | 3 | 34.5 | 57.4 | 845 |

รูปที่ 4.3 หน้าเว็บที่แสดงค่าสูงสุดต่ำสุดของค่า PM2.5 และตัวอย่างตารางข้อมูลที่ได้จากการเก็บข้อมูลบริเวณคณะวิทยาศาสตร์

จากรูปที่ 4.3 เป็นหน้าเว็บไซต์ฐานข้อมูลที่แสดงถึงข้อมูลที่ได้จากการวัดปริมาณฝุ่นละอองบริเวณคณะวิทยาศาสตร์ และยังสามารถแสดงค่าสูงสุดต่ำสุดของปริมาณ PM2.5 โดยที่ค่าปริมาณ PM2.5 สูงสุดอยู่ที่ 4 ug/m³ และต่ำสุดที่ 0 ug/m³ ของช่วงเวลาที่ได้ทำการเก็บข้อมูล นอกจากนี้ยังทำการแสดงภาพของบริเวณที่ได้ทำการเก็บข้อมูล เพื่อให้ผู้ที่เข้ามาดูค่าปริมาณฝุ่นละอองทราบถึงบริเวณที่ได้ทำการเก็บข้อมูลค่าปริมาณฝุ่นละออง

ขั้นตอนต่อมาเป็นการสร้างฐานข้อมูลบริเวณ คณะวิศวกรรมศาสตร์ ทำการเก็บข้อมูลในวันอังคารที่ 18 พฤษภาคม พ.ศ. 2564 เวลา 15.12-16.00 หลังจากที่ได้ทำการเขียนหน้าเว็บไซต์แสดงฐานข้อมูลของบริเวณ คณะวิศวกรรมศาสตร์ จะได้หน้าเว็บไซต์ดังรูปที่ 4.5



รูปที่ 4.4 บริเวณที่ติดตั้งเครื่องวัดปริมาณต่าง ๆ ณ คณะวิศวกรรมศาสตร์

ปริมาณฝุ่นละอองในช่วงเวลา 15.12-16.00 ของวันที่ 18 พฤษภาคม พ.ศ. 2564 ณ คณะวิศวกรรมศาสตร์



ค่า PM2.5

สูงสุดอยู่ที่ 3 ug/m³

ต่ำสุดอยู่ที่ 0 ug/m³

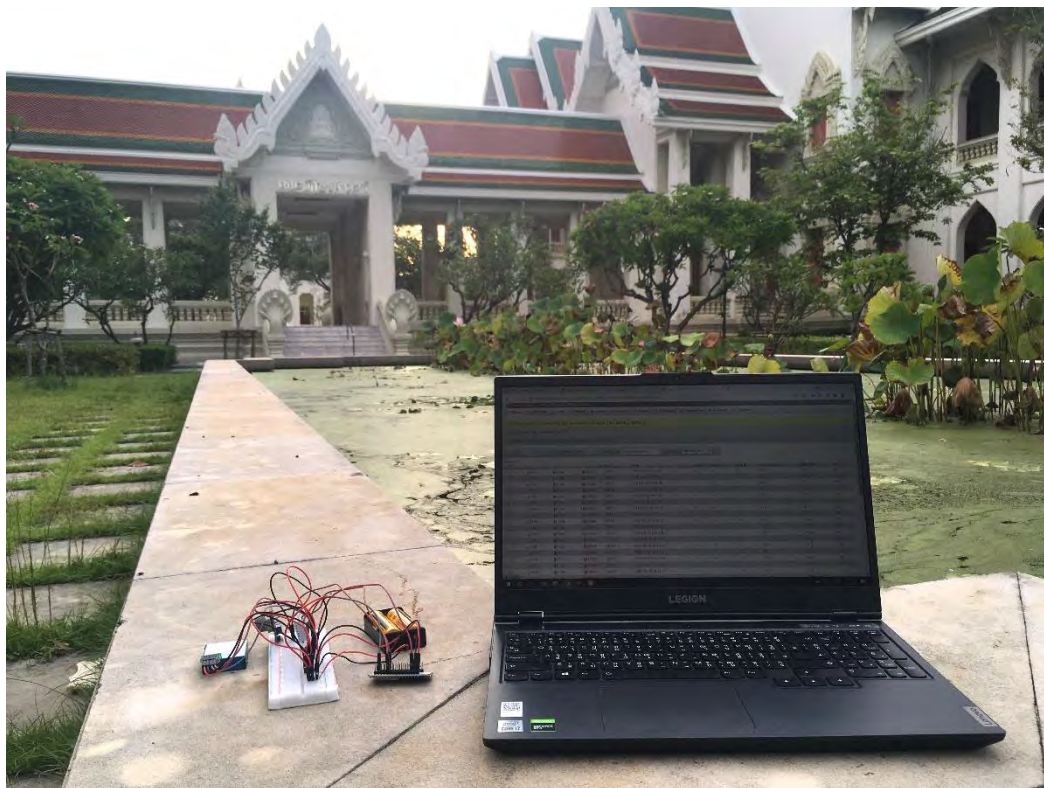
ตัวอย่างข้อมูล

| time | pm1 | pm2.5 | pm10 | อุณหภูมิ | ความชื้น | CO_2 |
|---------------------|-----|-------|------|----------|----------|------|
| 2021-05-18 15:30:10 | 1 | 3 | 3 | 34.9 | 55.8 | 772 |
| 2021-05-18 15:30:20 | 0 | 1 | 1 | 34.8 | 55.7 | 735 |
| 2021-05-18 15:30:31 | 0 | 1 | 1 | 34.8 | 55.8 | 751 |
| 2021-05-18 15:30:42 | 1 | 1 | 1 | 34.8 | 55.9 | 769 |
| 2021-05-18 15:30:52 | 1 | 1 | 3 | 34.8 | 55.9 | 749 |
| 2021-05-18 15:31:03 | 0 | 1 | 3 | 34.8 | 55.9 | 749 |
| 2021-05-18 15:31:13 | 0 | 1 | 1 | 34.8 | 56 | 753 |
| 2021-05-18 15:31:24 | 0 | 1 | 1 | 34.8 | 56.2 | 779 |
| 2021-05-18 15:31:35 | 0 | 1 | 1 | 34.8 | 56.2 | 705 |
| 2021-05-18 15:31:45 | 0 | 0 | 1 | 34.8 | 56.4 | 759 |
| 2021-05-18 15:31:56 | 0 | 0 | 1 | 34.7 | 56.4 | 751 |
| 2021-05-18 15:32:07 | 0 | 0 | 3 | 34.7 | 56.1 | 752 |
| 2021-05-18 15:32:17 | 0 | 1 | 3 | 34.7 | 56.1 | 758 |
| 2021-05-18 15:32:28 | 1 | 3 | 3 | 34.6 | 56.3 | 759 |
| 2021-05-18 15:32:38 | 1 | 3 | 3 | 34.5 | 56.4 | 765 |
| 2021-05-18 15:32:49 | 1 | 3 | 3 | 34.5 | 56.7 | 752 |
| 2021-05-18 15:32:59 | 0 | 1 | 1 | 34.6 | 56.9 | 758 |
| 2021-05-18 15:33:10 | 0 | 1 | 1 | 34.6 | 57 | 751 |
| 2021-05-18 15:33:20 | 0 | 1 | 1 | 34.6 | 57.2 | 758 |
| 2021-05-18 15:33:31 | 0 | 1 | 1 | 34.7 | 57.3 | 742 |
| 2021-05-18 15:33:42 | 0 | 1 | 1 | 34.7 | 57 | 775 |

รูปที่ 4.5 หน้าเว็บที่แสดงค่าสูงสุดต่ำสุดของค่า PM2.5 และตัวอย่างตารางข้อมูลที่ได้จากการเก็บข้อมูลบริเวณคณะวิศวกรรมศาสตร์

จากรูปที่ 4.5 เป็นหน้าเว็บไซต์ฐานข้อมูลที่แสดงถึงข้อมูลที่ได้จากการวัดปริมาณฝุ่นละอองบริเวณคณะวิศวกรรมศาสตร์ และยังสามารถแสดงค่าสูงสุดต่ำสุดของปริมาณ PM2.5 ได้เช่นเดียวกับฐานข้อมูลของบริเวณคณะวิทยาศาสตร์ โดยที่ค่าปริมาณ PM2.5 สูงสุดอยู่ที่ 3 ug/m^3 และต่ำสุดที่ 0 ug/m^3 ของช่วงเวลาที่ได้ทำการเก็บข้อมูล

บริเวณต่อมาที่จะทำการสร้างหน้าเว็บไซต์แสดงฐานข้อมูลคือ คณะอักษรศาสตร์ ทำการเก็บข้อมูลในวันพุธที่ 19 พฤษภาคม พ.ศ. 2564 เวลา 16.00-17.00 จะได้หน้าเว็บไซต์ดังรูปที่ 4.7



รูปที่ 4.6 บริเวณที่ติดตั้งเครื่องวัดปริมาณต่าง ๆ ณ คณะอักษรศาสตร์

ปริมาณฝุ่นละอองในช่วงเวลา 16.00-17.00 ของวันที่ 19 พฤษภาคม พ.ศ. 2564 ณ คณะอักษรศาสตร์



ค่า PM2.5

สูงสุดอยู่ที่ 6 ug/m³

ต่ำสุดอยู่ที่ 1 ug/m³

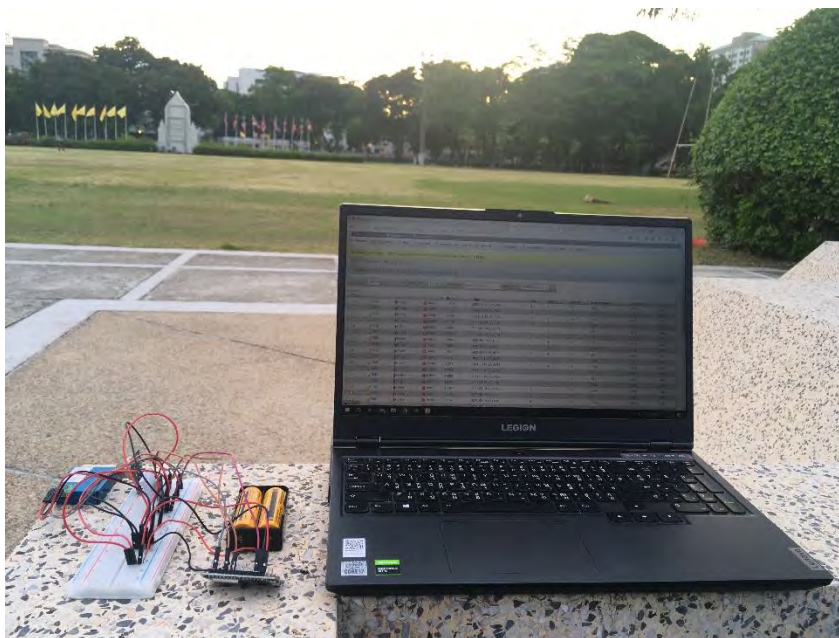
ตัวอย่างข้อมูล

| time | pm1 | pm2.5 | pm10 | อุณหภูมิ | ความชื้น | CO_2 |
|---------------------|-----|-------|------|----------|----------|------|
| 2021-05-19 16:10:15 | 1 | 1 | 1 | 32.6 | 59.7 | 861 |
| 2021-05-19 16:10:26 | 0 | 1 | 1 | 32.6 | 59.7 | 895 |
| 2021-05-19 16:10:37 | 1 | 1 | 1 | 32.6 | 59.6 | 911 |
| 2021-05-19 16:10:48 | 1 | 1 | 1 | 32.7 | 59.6 | 911 |
| 2021-05-19 16:10:59 | 1 | 2 | 2 | 32.6 | 59.5 | 881 |
| 2021-05-19 16:11:09 | 1 | 3 | 3 | 32.7 | 59.6 | 883 |
| 2021-05-19 16:11:20 | 1 | 1 | 1 | 32.7 | 59.6 | 876 |
| 2021-05-19 16:11:31 | 0 | 3 | 3 | 32.7 | 59.5 | 923 |
| 2021-05-19 16:11:42 | 1 | 6 | 6 | 32.7 | 59.4 | 918 |
| 2021-05-19 16:11:53 | 1 | 6 | 6 | 32.7 | 59.4 | 877 |
| 2021-05-19 16:12:04 | 1 | 3 | 3 | 32.7 | 59.3 | 895 |
| 2021-05-19 16:12:15 | 1 | 3 | 3 | 32.7 | 59.3 | 843 |
| 2021-05-19 16:12:25 | 1 | 3 | 3 | 32.7 | 59.4 | 891 |
| 2021-05-19 16:12:36 | 1 | 3 | 3 | 32.7 | 59.3 | 869 |
| 2021-05-19 16:12:46 | 1 | 3 | 3 | 32.7 | 59.3 | 912 |
| 2021-05-19 16:12:57 | 1 | 1 | 1 | 32.7 | 59.2 | 911 |
| 2021-05-19 16:13:08 | 1 | 3 | 3 | 32.7 | 59.2 | 861 |
| 2021-05-19 16:13:18 | 1 | 3 | 3 | 32.7 | 59.3 | 850 |
| 2021-05-19 16:13:29 | 2 | 4 | 4 | 32.7 | 59.4 | 865 |
| 2021-05-19 16:13:39 | 3 | 4 | 6 | 32.7 | 59.6 | 865 |
| 2021-05-19 16:13:50 | 1 | 2 | 2 | 32.7 | 59.5 | 907 |

รูปที่ 4.7 หน้าเว็บที่แสดงค่าสูงสุดต่ำสุดของค่า PM2.5 และตัวอย่างตารางข้อมูลที่ได้จากการเก็บข้อมูลบริเวณคณะอักษรศาสตร์

จากรูปที่ 4.7 แสดงถึงหน้าเว็บไซต์ฐานข้อมูลที่ได้จากการวัดปริมาณฝุ่นละอองบริเวณคณะอักษรศาสตร์ และยังสามารถแสดงค่าสูงสุดต่ำสุดปริมาณ PM2.5 ได้เช่นเดียวกับฐานข้อมูลก่อนหน้า โดยที่ค่าปริมาณ PM2.5 สูงสุดอยู่ที่ 6 ug/m³ และต่ำสุดที่ 1 ug/m³ ของช่วงเวลาที่ได้ทำการเก็บข้อมูล

บริเวณสุดท้ายที่ทำการเก็บข้อมูลและสร้างฐานข้อมูลคือ บริเวณสระน้ำจุฬาฯ ทำการเก็บข้อมูล ในวันพุธ ที่ 19 พฤษภาคม พ.ศ. 2564 เวลา 17.10-17.35 จะได้หน้าเว็บไซต์แสดงฐานข้อมูลดังรูปที่ 4.9



รูปที่ 4.8 บริเวณที่ติดตั้งเครื่องวัดปริมาณต่าง ๆ ณ สระน้ำจุฬาฯ

ปริมาณฝุ่นละอองในช่วงเวลา 17.10-17.35 ของวันที่ 19 พฤษภาคม พ.ศ. 2564 ณ บริเวณสระน้ำ



ค่า PM2.5

สูงสุดอยู่ที่ 9 ug/m³

ต่ำสุดอยู่ที่ 1 ug/m³

ตัวอย่างข้อมูล

| time | pm1 | pm2.5 | pm10 | อุณหภูมิ | ความชื้น | CO_2 |
|---------------------|-----|-------|------|----------|----------|------|
| 2021-05-19 17:11:02 | 0 | 1 | 1 | 32.9 | 59.3 | 771 |
| 2021-05-19 17:11:13 | 1 | 3 | 3 | 32.9 | 59.5 | 765 |
| 2021-05-19 17:11:23 | 1 | 3 | 3 | 32.9 | 59.4 | 805 |
| 2021-05-19 17:11:34 | 3 | 6 | 6 | 32.9 | 59.4 | 795 |
| 2021-05-19 17:11:45 | 3 | 6 | 6 | 32.9 | 60 | 794 |
| 2021-05-19 17:11:56 | 3 | 4 | 4 | 32.9 | 60.6 | 814 |
| 2021-05-19 17:12:07 | 1 | 4 | 4 | 32.9 | 60.1 | 805 |
| 2021-05-19 17:12:17 | 1 | 6 | 7 | 32.9 | 59.9 | 751 |
| 2021-05-19 17:12:28 | 3 | 6 | 6 | 32.9 | 60.1 | 783 |
| 2021-05-19 17:12:38 | 3 | 4 | 4 | 32.9 | 60.1 | 799 |
| 2021-05-19 17:12:49 | 3 | 6 | 6 | 32.9 | 59.7 | 814 |
| 2021-05-19 17:12:59 | 3 | 4 | 4 | 32.9 | 59.7 | 791 |
| 2021-05-19 17:13:10 | 3 | 4 | 4 | 32.9 | 60.2 | 768 |
| 2021-05-19 17:13:21 | 3 | 7 | 7 | 32.9 | 60.1 | 761 |
| 2021-05-19 17:13:31 | 1 | 6 | 6 | 32.9 | 60.2 | 785 |
| 2021-05-19 17:13:42 | 1 | 6 | 9 | 32.9 | 60.2 | 806 |
| 2021-05-19 17:13:52 | 3 | 6 | 6 | 32.9 | 59.8 | 828 |
| 2021-05-19 17:14:03 | 3 | 3 | 4 | 32.9 | 59.8 | 756 |
| 2021-05-19 17:14:14 | 3 | 6 | 6 | 32.9 | 59.8 | 752 |
| 2021-05-19 17:14:24 | 1 | 3 | 3 | 32.9 | 59.8 | 720 |
| 2021-05-19 17:14:35 | 3 | 3 | 3 | 32.9 | 59.9 | 752 |

รูปที่ 4.9 หน้าเว็บที่แสดงค่าสูงสุดต่ำสุดของค่า PM2.5 และตัวอย่างตารางข้อมูลที่ได้จากการเก็บข้อมูลบริเวณสระน้ำจุฬาฯ

จากรูปที่ 4.9 แสดงถึงหน้าเว็บไซต์ฐานข้อมูลที่ได้จากการวัดปริมาณฝุ่นละอองบริเวณสระน้ำจุฬาฯ และยังสามารถแสดงค่าปริมาณ PM2.5 สูงสุดต่ำสุดได้ โดยที่ค่าปริมาณ PM2.5 สูงสุดอยู่ที่ 9 ug/m^3 และต่ำสุดที่ 1 ug/m^3 ของช่วงเวลาที่ได้ทำการเก็บข้อมูล

หลังจากที่ได้หน้าเว็บไซต์แสดงฐานข้อมูลทั้ง 4 บริเวณแล้ว ต่อไปคือการนำฐานข้อมูลไปแสดงบนเว็บไซต์ที่ได้ทำในตอนแรก โดยใช้ภาษา HTML ในการเชื่อมโยงแต่ละบริเวณเข้ากับฐานข้อมูลที่ได้ทำไว้ หลังจากเชื่อมโยงฐานข้อมูลเข้ากับแผนที่จุฬาลงกรณ์มหาวิทยาลัย ก็จะสามารถคลิกบริเวณที่สนใจแล้วทำให้ปรากฏฐานข้อมูลของแต่ละบริเวณได้ และยังทำการแสดงค่าสูงสุดต่ำสุดของปริมาณฝุ่นละอองได้โดยการนำเคอร์เซอร์เมาส์ไปไว้ในบริเวณที่เราอยากจะทำค่าปริมาณฝุ่นละออง ยกตัวอย่างเช่น



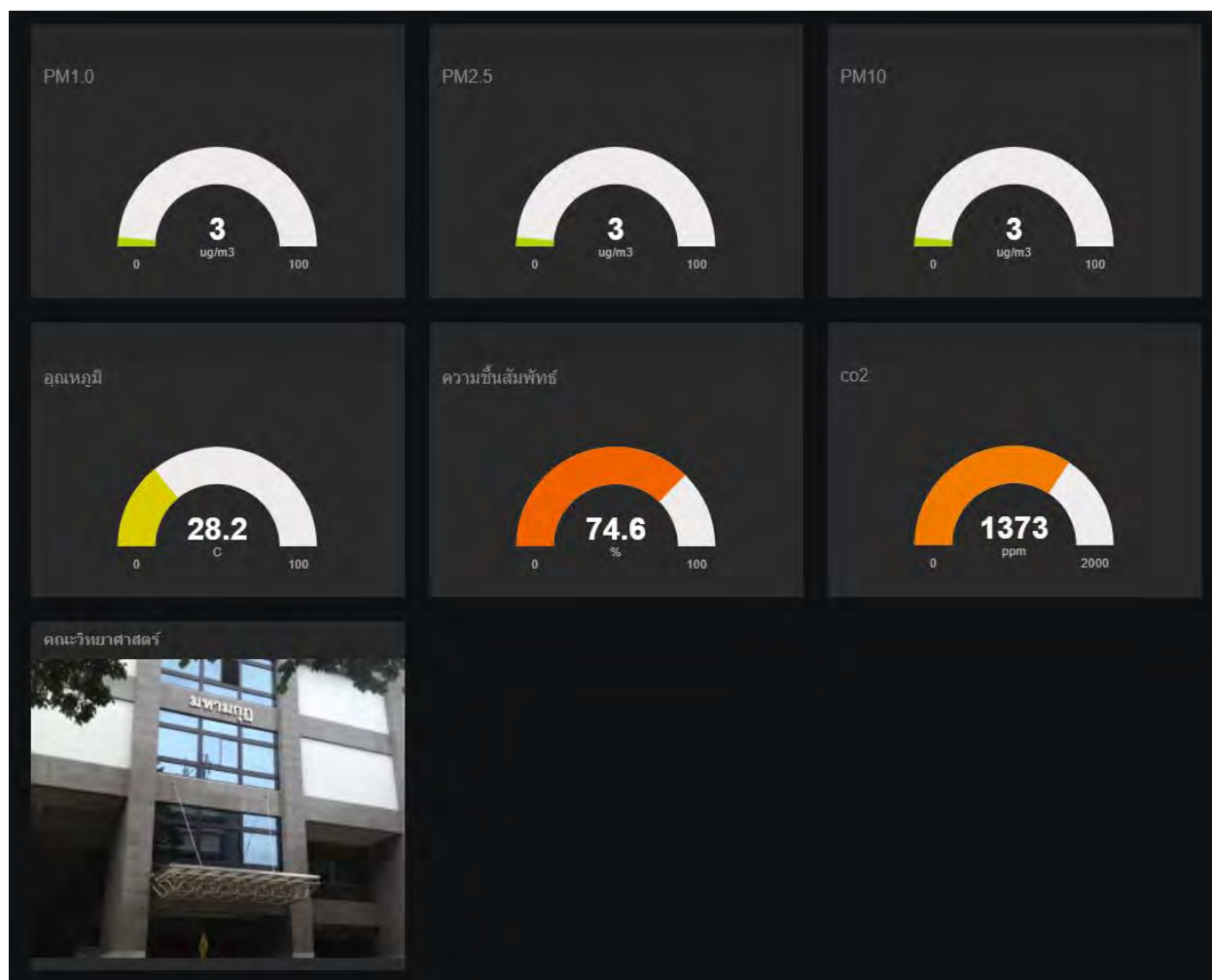
รูปที่ 4.10 ภาพตัวอย่างการปรากฏของข้อความแสดงข้อความค่าสูงสุดต่ำสุดปริมาณ PM 2.5

จากรูปที่ 4.10 แสดงถึงการปรากฏของค่าปริมาณฝุ่นละอองสูงสุดต่ำสุด ณ บริเวณสระน้ำจุฬาฯ เมื่อนำเคอร์เซอร์เมาส์ไปไว้ในบริเวณสระน้ำจุฬาฯ โดยมีปริมาณ PM2.5 สูงสุดอยู่ที่ 9 ug/m^3 และต่ำสุดที่ 1 ug/m^3 เช่นเดียวกับที่แสดงในฐานข้อมูลของบริเวณสระน้ำจุฬาฯ

4.2 ทดสอบการอ่านค่าข้อมูลจากตัวรับสัญญาณในระยะไกลผ่านทางหน้าเว็บไซต์ NETPIE

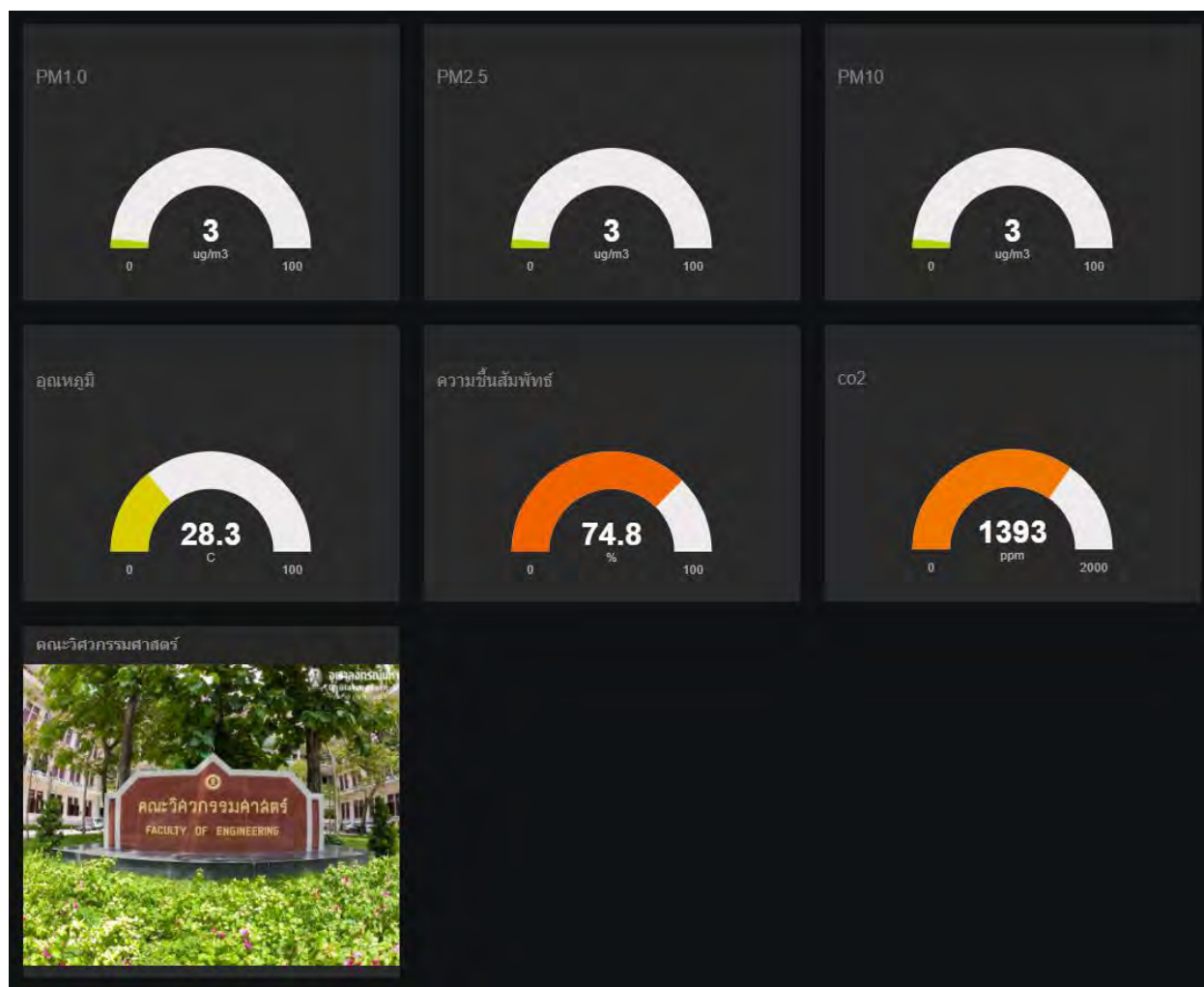
สำหรับการทดลองตอนที่ 2 มีจุดประสงค์คือ เพื่อทดสอบว่าระบบตรวจวัดปริมาณฝุ่นละออง อุณหภูมิ ความชื้นสัมพัทธ์ และปริมาณก๊าซคาร์บอนไดออกไซด์ สามารถตรวจวัดปริมาณค่าดังกล่าวได้และแสดงผลผ่านทางหน้าเว็บไซต์ NETPIE แบบ Real time ได้หรือไม่

ระบบตรวจวัดปริมาณต่าง ๆ ด้วยระบบสมองกลฝังตัว ในโครงการนี้ใช้ ESP32 เป็นตัวควบคุมหลัก โดย ESP32 จะต้องมีการเชื่อมต่อกับเครือข่าย WiFi โดยต่อสัญญาณ Hotspot ของโทรศัพท์มือถือ หลังจากทำการเชื่อมต่อกับเครือข่ายอินเทอร์เน็ตได้แล้ว ESP32 จะทำการส่งค่าปริมาณฝุ่นละออง อุณหภูมิ ความชื้นสัมพัทธ์ และปริมาณก๊าซคาร์บอนไดออกไซด์ไปยังส่วนแสดงผลที่เรียกว่า แดชบอร์ด (Dashboard) ส่วนแสดงผลที่ 1 บริเวณ คณะวิทยาศาสตร์ ข้อมูล ณ วันอังคารที่ 18 พฤษภาคม พ.ศ. 2564



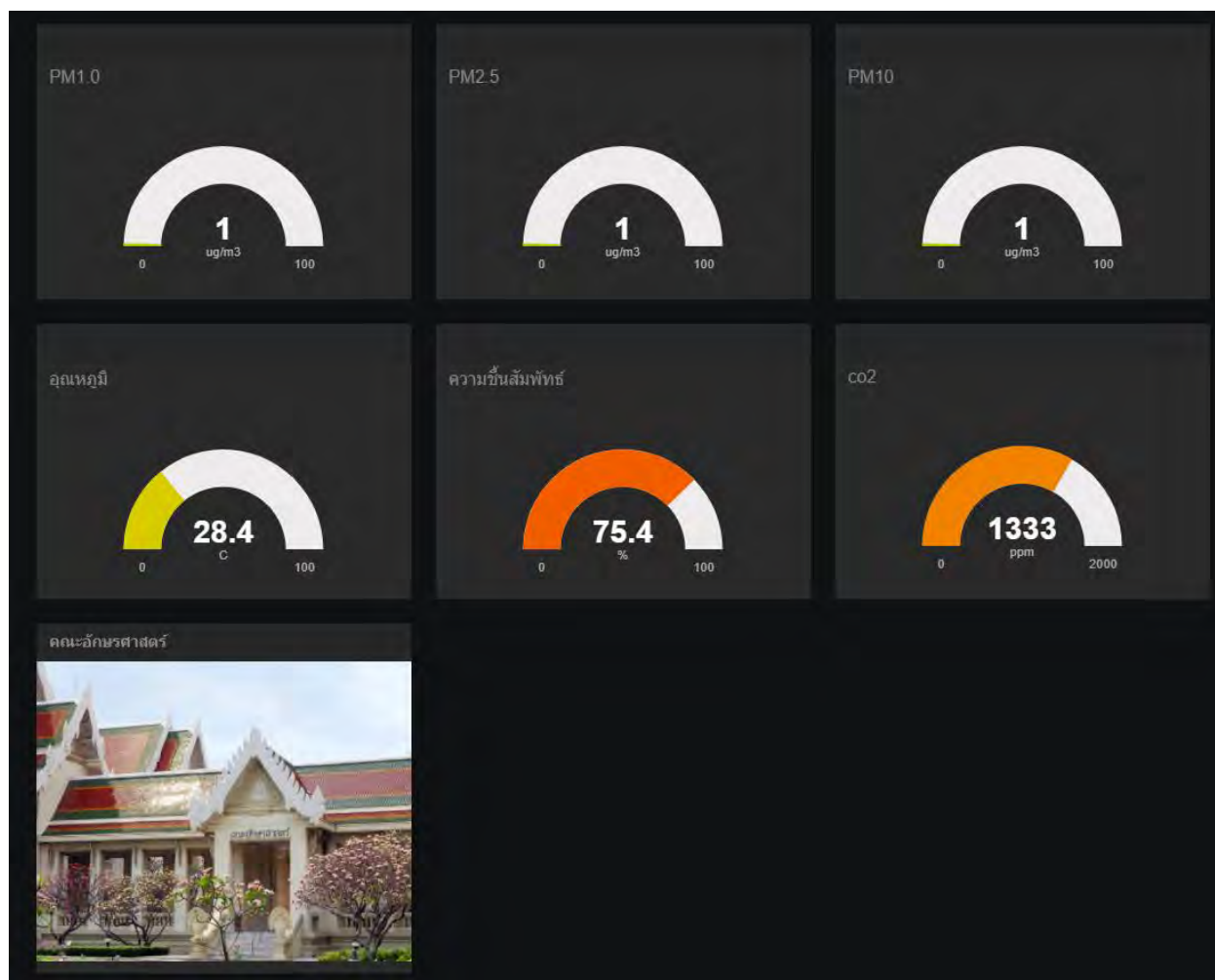
รูปที่ 4.11 ภาพการแสดงผลการทำงานของระบบตรวจวัดปริมาณต่าง ๆ ด้วยระบบสมองกลฝังตัว ณ คณะวิทยาศาสตร์

ส่วนแสดงผลที่ 2 บริเวณ คณะวิศวกรรมศาสตร์ ข้อมูล ณ วันอังคารที่ 18 พฤษภาคม พ.ศ. 2564



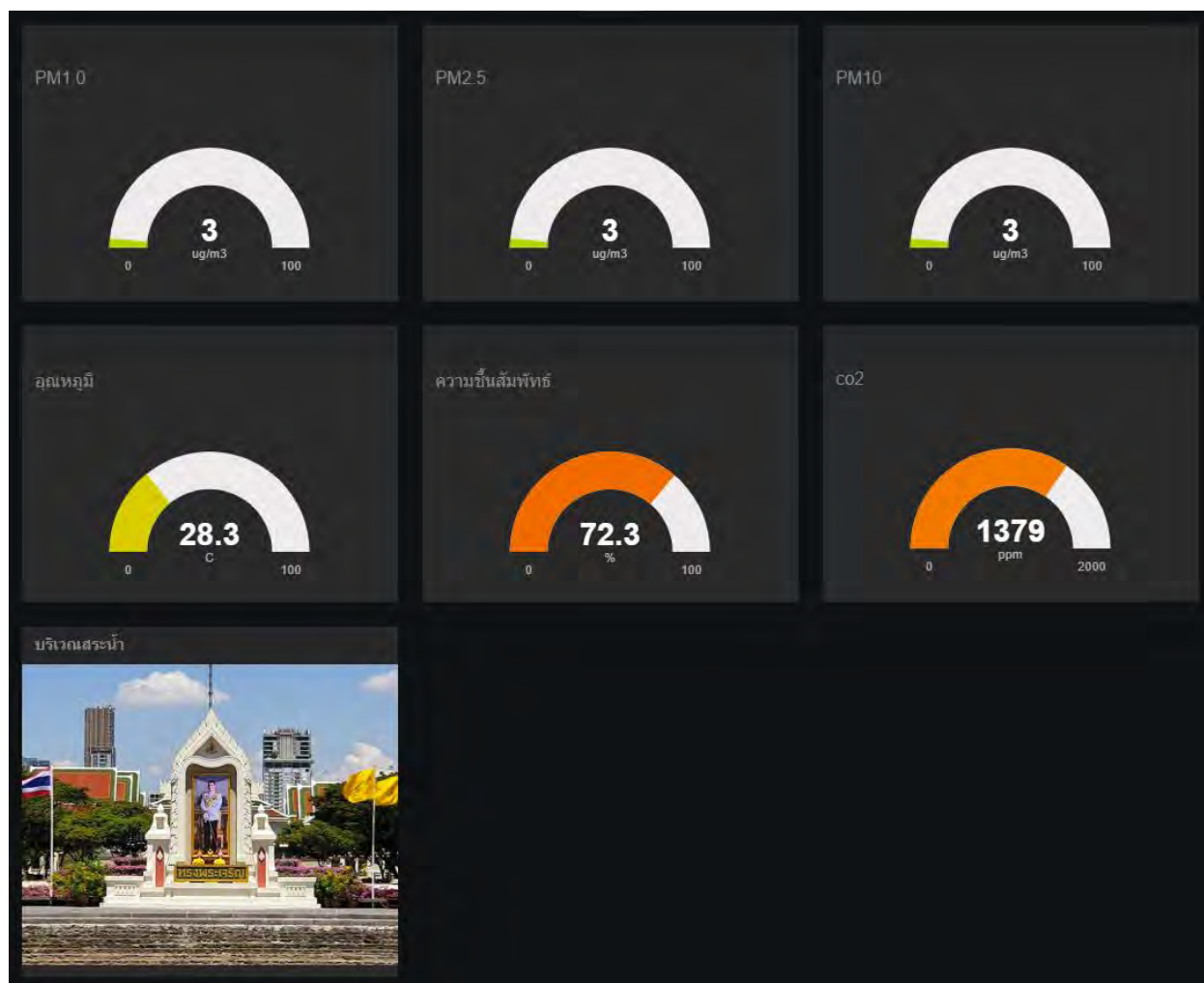
รูปที่ 4.12 ภาพการแสดงผลการทำงานของระบบตรวจวัดปริมาณต่าง ๆ ด้วยระบบสมองกลฝังตัว ณ คณะวิศวกรรมศาสตร์

ส่วนแสดงผลที่ 3 บริเวณ คณะอักษรศาสตร์ ข้อมูล ณ วันพุธที่ 19 พฤษภาคม พ.ศ. 2564



รูปที่ 4.13 ภาพการแสดงผลการทำงานของระบบตรวจวัดปริมาณต่าง ๆ ด้วยระบบสมองกลฝังตัว ณ คณะอักษรศาสตร์

ส่วนแสดงผลที่ 4 บริเวณ สระน้ำจุฬาฯ ข้อมูล ณ วันพุธที่ 19 พฤษภาคม พ.ศ. 2564



รูปที่ 4.14 ภาพการแสดงผลการทำงานของระบบตรวจวัดปริมาณต่าง ๆ ด้วยระบบสมองกลฝังตัว ณ บริเวณสระน้ำจุฬาฯ

4.3 ทดสอบการส่งข้อความแจ้งเตือนปริมาณฝุ่นละออง PM2.5 ที่เกินเกณฑ์กำหนดไปยัง LINE Notify

สำหรับการทดลองตอนที่ 3 มีจุดประสงค์ เพื่อทดสอบว่าระบบตรวจวัดปริมาณฝุ่นละออง สามารถแจ้งเตือนค่าปริมาณฝุ่นละออง PM2.5 ที่เกินเกณฑ์ที่กำหนดไว้ได้หรือไม่ โดยเกณฑ์ที่ตั้งไว้คือ PM2.5 ต้องมากกว่าหรือเท่ากับ $50 \mu\text{g}/\text{m}^3$ ซึ่งเป็นค่าที่คณะกรรมการสิ่งแวดล้อมแห่งชาติของประเทศไทย เป็นผู้กำหนดปริมาณฝุ่นละออง PM2.5 ที่เริ่มส่งผลกระทบต่อสุขภาพของมนุษย์ ทำการวัดในวันพุธที่ 14 มกราคม พ.ศ. 2564 โดยในวันนั้นพื้นที่กรุงเทพมหานครและปริมณฑลมีปริมาณ¹ PM2.5 เฉลี่ยอยู่ที่ $71 \mu\text{g}/\text{m}^3$ [23]

```

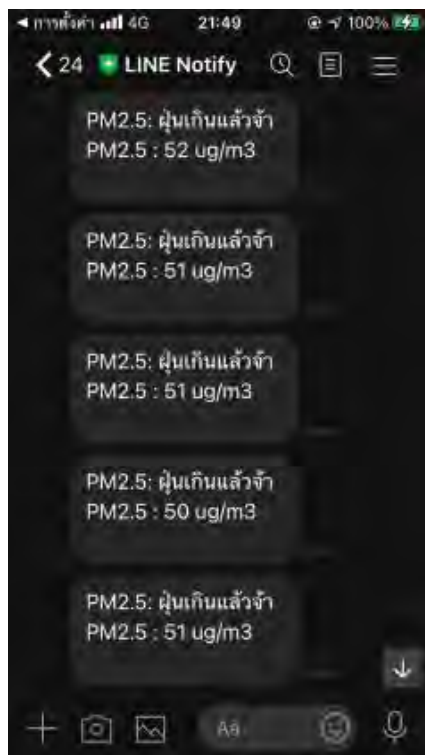
COM6
[
  { "pm1": 37 ug/m3, "pm2_5": 53 ug/m3, "pm10": 53 ug/m3 }
  Sending Line : ฝุ่นกึ่งกลาง PM2.5 : 53 ug/m3
  { "pm1": 39 ug/m3, "pm2_5": 54 ug/m3, "pm10": 54 ug/m3 }
  Sending Line : ฝุ่นกึ่งกลาง PM2.5 : 54 ug/m3
  { "pm1": 39 ug/m3, "pm2_5": 54 ug/m3, "pm10": 54 ug/m3 }
  Sending Line : ฝุ่นกึ่งกลาง PM2.5 : 54 ug/m3
  { "pm1": 38 ug/m3, "pm2_5": 52 ug/m3, "pm10": 52 ug/m3 }
  Sending Line : ฝุ่นกึ่งกลาง PM2.5 : 52 ug/m3
  { "pm1": 39 ug/m3, "pm2_5": 51 ug/m3, "pm10": 52 ug/m3 }
  Sending Line : ฝุ่นกึ่งกลาง PM2.5 : 51 ug/m3
  { "pm1": 38 ug/m3, "pm2_5": 51 ug/m3, "pm10": 55 ug/m3 }
  Sending Line : ฝุ่นกึ่งกลาง PM2.5 : 51 ug/m3
  { "pm1": 39 ug/m3, "pm2_5": 50 ug/m3, "pm10": 54 ug/m3 }
  Sending Line : ฝุ่นกึ่งกลาง PM2.5 : 50 ug/m3
  { "pm1": 38 ug/m3, "pm2_5": 51 ug/m3, "pm10": 55 ug/m3 }
  Sending Line : ฝุ่นกึ่งกลาง PM2.5 : 51 ug/m3
  { "pm1": 39 ug/m3, "pm2_5": 53 ug/m3, "pm10": 57 ug/m3 }
  Sending Line : ฝุ่นกึ่งกลาง PM2.5 : 53 ug/m3
  { "pm1": 39 ug/m3, "pm2_5": 52 ug/m3, "pm10": 57 ug/m3 }
  Sending Line : ฝุ่นกึ่งกลาง PM2.5 : 52 ug/m3
  { "pm1": 39 ug/m3, "pm2_5": 52 ug/m3, "pm10": 54 ug/m3 }
  Sending Line : ฝุ่นกึ่งกลาง PM2.5 : 52 ug/m3
  { "pm1": 38 ug/m3, "pm2_5": 53 ug/m3, "pm10": 55 ug/m3 }
  Sending Line : ฝุ่นกึ่งกลาง PM2.5 : 53 ug/m3
  { "pm1": 38 ug/m3, "pm2_5": 53 ug/m3, "pm10": 56 ug/m3 }
  Sending Line : ฝุ่นกึ่งกลาง PM2.5 : 53 ug/m3
  { "pm1": 39 ug/m3, "pm2_5": 54 ug/m3, "pm10": 58 ug/m3 }
  Sending Line : ฝุ่นกึ่งกลาง PM2.5 : 54 ug/m3
  { "pm1": 39 ug/m3, "pm2_5": 53 ug/m3, "pm10": 58 ug/m3 }
  Sending Line : ฝุ่นกึ่งกลาง PM2.5 : 53 ug/m3

```

รูปที่ 4.15 ส่วนแสดงผลของค่าปริมาณ PM2.5 พร้อมทั้งส่งค่าปริมาณฝุ่นละออง PM2.5 ที่เกินเกณฑ์กำหนดไปยัง Line

จากรูปที่ 4.15 เป็นการทดลองวัดปริมาณฝุ่นละอองภายในห้องปิด เพื่อทดสอบว่าเมื่อค่าปริมาณฝุ่นละอองที่วัดได้ภายในห้องเกินเกณฑ์ที่กำหนดไว้จะสามารถส่งข้อความไปยัง LINE Notify ได้หรือไม่ โดยที่ตั้งเกณฑ์ไว้ที่ $50 \mu\text{g}/\text{m}^3$ และในขณะที่ทำการทดลองอยู่ค่าปริมาณ PM2.5 อยู่ที่ประมาณ $52 \pm 2 \mu\text{g}/\text{m}^3$ ซึ่งเกินเกณฑ์ที่ได้กำหนดไว้ จึงทำการส่งข้อความค่าปริมาณฝุ่นละอองที่เกินเกณฑ์ที่กำหนดไว้ไปยัง LINE Notify แสดงได้ดังรูปที่ 4.16

¹ อ้างอิงจกตารางแสดงค่าเฉลี่ย 24 ชั่วโมงของ PM-2.5 ณ เวลา 00.00 น. (มคก./ลบ.ม.) ในพื้นที่กรุงเทพมหานครและปริมณฑลในเดือนมกราคม 2564 ของกรมควบคุมมลพิษ



รูปที่ 4.16 รูปแสดงการส่งข้อความไปยัง Line พร้อมทั้งระบุค่าปริมาณ PM2.5 ที่เกินเกณฑ์ที่กำหนดในขณะนั้น

บทที่ 5

สรุปผลการทดลอง ปัญหาและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

การออกแบบระบบตรวจวัด PM2.5 ด้วยระบบสมองกลฝังตัว ออกแบบโดยใช้ไมโครคอนโทรลเลอร์ (ESP32) ซึ่งเชื่อมต่อกับตัวรับรู้ปริมาณฝุ่นละออง อุณหภูมิ ความชื้นสัมพัทธ์ และปริมาณก๊าซคาร์บอนไดออกไซด์ โดยสถานที่ที่ได้ทำการเก็บข้อมูลข้างต้นมีทั้งหมด 4 สถานที่ ได้แก่ คณะวิทยาศาสตร์ คณะวิศวกรรมศาสตร์ คณะอักษรศาสตร์ และบริเวณสระน้ำของจุฬาลงกรณ์มหาวิทยาลัย แล้วส่งค่าข้อมูลไปเก็บในฐานข้อมูล MySQL เพื่อนำข้อมูลที่ได้มาแสดงผลผ่านหน้าเว็บไซต์ที่ได้ออกแบบเอง โดยหน้าเว็บไซต์ใช้ภาษา JavaScript ซึ่งมีฟังก์ชันเมื่อนำเคอร์เซอร์เมาส์ไปยังบริเวณที่ต้องการ ทำให้ปรากฏข้อความโดยไม่จำเป็นต้องโหลดหน้านั้นใหม่อีกครั้ง และใช้ภาษา HTML เพื่อให้สามารถคลิกในบริเวณที่สนใจให้เข้าไปดูข้อมูล ณ บริเวณนั้นอย่างละเอียดได้ อีกทั้งยังนำข้อมูลปริมาณฝุ่นละออง อุณหภูมิ ความชื้นสัมพัทธ์ และปริมาณก๊าซคาร์บอนไดออกไซด์ของทั้ง 4 สถานที่ไปแสดงผลผ่าน NETPIE เพื่อแสดงค่าข้อมูลได้แบบ Real time นอกจากนี้เมื่อระบบตรวจวัด PM2.5 ด้วยระบบสมองกลฝังตัวตรวจวัดค่าฝุ่นละออง PM2.5 เกินกว่า 50 ug/m^3 ซึ่งเป็นค่าที่เริ่มส่งผลเสียต่อร่างกายของมนุษย์ ให้ทำการส่งค่าปริมาณฝุ่นละออง PM2.5 ไปยัง LINE Notify เพื่อป้องกันอันตรายจากมลพิษทางอากาศได้ทันทั่วทั้งที่ โดยสามารถสรุปผลโครงการได้ดังนี้

การทดลองระบบตรวจวัด PM2.5 ด้วยระบบสมองกลฝังตัว พบว่าค่าที่วัดได้มีปริมาณที่สอดคล้องกับสถานการณ์ PM2.5 ในพื้นที่กรุงเทพมหานครและปริมณฑล จากกรมควบคุมมลพิษ ตัวอย่างเช่น ในวันที่ 14 มกราคม พ.ศ. 2564 ที่กรมควบคุมมลพิษระบุว่าค่าฝุ่นละอองเป็นจำนวนมาก เครื่องวัดนี้ก็สามารถวัดปริมาณฝุ่นละออง PM2.5 ได้ $52 \pm 2 \text{ ug/m}^3$ และในวันที่ 19 พฤษภาคม พ.ศ. 2564 ที่กรมควบคุมมลพิษระบุว่าปริมาณฝุ่นละอองต่ำ เครื่องวัดนี้ก็สามารถวัดปริมาณฝุ่นละออง PM 2.5 ได้ $4 \pm 2 \text{ ug/m}^3$ ซึ่งสอดคล้องกับกรมควบคุมมลพิษ

การศึกษาการทำเว็บไซต์และเก็บรวบรวมข้อมูลปริมาณฝุ่นละออง อุณหภูมิ ความชื้นสัมพัทธ์ และปริมาณก๊าซคาร์บอนไดออกไซด์ เพื่อใช้ในเฝ้าระวังและติดตามปริมาณต่าง ๆ ได้ในหลาย ๆ สถานที่ในหน้าเว็บไซต์เดียวกันพบว่าเว็บไซต์ที่ได้สร้างขึ้นมานั้นแสดงผลได้ตามเป้าหมายที่ตั้งไว้ กล่าวคือสามารถแสดงค่าปริมาณฝุ่นละออง อุณหภูมิ ความชื้นสัมพัทธ์ และปริมาณก๊าซคาร์บอนไดออกไซด์ ของคณะวิทยาศาสตร์ คณะวิศวกรรมศาสตร์ คณะอักษรศาสตร์ และบริเวณสระน้ำของจุฬาลงกรณ์มหาวิทยาลัยได้ในหน้าเว็บไซต์เดียวกันและยังช่วยให้รู้ค่าปริมาณสูงสุดและต่ำสุดของฝุ่นละอองได้อย่างรวดเร็ว โดยใช้คำสั่งของภาษา JavaScript ทำให้เมื่อลากเคอร์เซอร์

เมาส์ไปยังบริเวณที่ต้องการให้ปรากฏค่าสูงสุดและต่ำสุดของปริมาณ PM 2.5 ณ บริเวณที่มีเครื่องวัดติดตั้งไว้ ทำให้สะดวกและง่ายต่อการติดตามค่าปริมาณฝุ่นละอองในหน้าเว็บไซต์เดียวกัน ดังแสดงในรูปที่ 5.1 นอกจากนี้ยังใช้ภาษา HTML เพื่อที่จะสามารถคลิกเข้าไปดูข้อมูลค่าฝุ่นละออง ณ เวลาต่าง ๆ ได้อย่างละเอียดในบริเวณที่สนใจได้อีกด้วย ดังแสดงในรูปที่ 5.2


คลิกที่คณะวิทยาศาสตร์ คณะวิศวกรรมศาสตร์ คณะอักษรศาสตร์ หรือบริเวณสระน้ำ เพื่อเช็คค่าปริมาณฝุ่นละออง :

บริเวณสระน้ำจุฬาฯ มีปริมาณ PM2.5 สูงสุดอยู่ที่ 9 ug/m³ ต่ำสุดอยู่ที่ 1 ug/m³



รูปที่ 5.1 ภาพตัวอย่างการปรากฏของค่าปริมาณฝุ่นละออง เมื่อนำเคอร์เซอร์เมาส์ไปยังบริเวณที่ต้องการ

ปริมาณฝุ่นละอองในช่วงเวลา 17.10-17.35 ของวันที่ 19 พฤษภาคม พ.ศ. 2564 ณ บริเวณสระน้ำ



ค่า PM2.5

สูงสุดอยู่ที่ 9 ug/m³

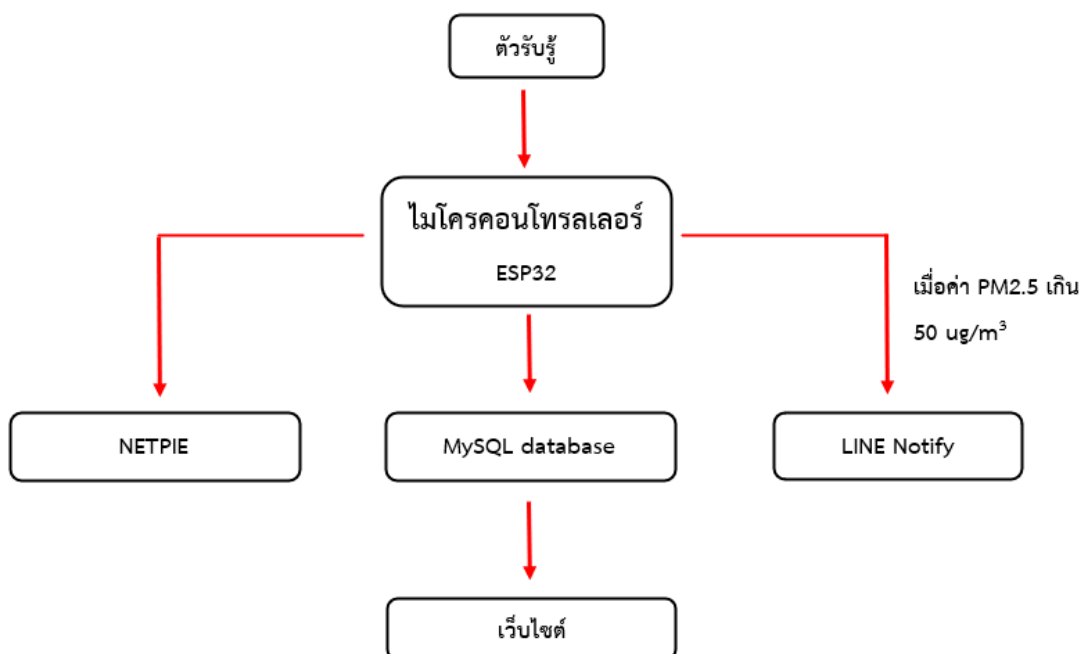
ต่ำสุดอยู่ที่ 1 ug/m³

ตัวอย่างข้อมูล

| time | pm1 | pm2.5 | pm10 | อุณหภูมิ | ความชื้น | CO ₂ |
|---------------------|-----|-------|------|----------|----------|-----------------|
| 2021-05-19 17:11:02 | 0 | 1 | 1 | 32.9 | 59.3 | 771 |
| 2021-05-19 17:11:13 | 1 | 3 | 3 | 32.9 | 59.5 | 765 |
| 2021-05-19 17:11:23 | 1 | 3 | 3 | 32.9 | 59.4 | 805 |
| 2021-05-19 17:11:34 | 3 | 6 | 6 | 32.9 | 59.4 | 795 |
| 2021-05-19 17:11:45 | 3 | 6 | 6 | 32.9 | 60 | 794 |
| 2021-05-19 17:11:56 | 3 | 4 | 4 | 32.9 | 60.6 | 814 |
| 2021-05-19 17:12:07 | 1 | 4 | 4 | 32.9 | 60.1 | 805 |
| 2021-05-19 17:12:17 | 1 | 6 | 7 | 32.9 | 59.9 | 751 |
| 2021-05-19 17:12:28 | 3 | 6 | 6 | 32.9 | 60.1 | 783 |
| 2021-05-19 17:12:38 | 3 | 4 | 4 | 32.9 | 60.1 | 799 |
| 2021-05-19 17:12:49 | 3 | 6 | 6 | 32.9 | 59.7 | 814 |
| 2021-05-19 17:12:59 | 3 | 4 | 4 | 32.9 | 59.7 | 791 |
| 2021-05-19 17:13:10 | 3 | 4 | 4 | 32.9 | 60.2 | 768 |
| 2021-05-19 17:13:21 | 3 | 7 | 7 | 32.9 | 60.1 | 761 |
| 2021-05-19 17:13:31 | 1 | 6 | 6 | 32.9 | 60.2 | 785 |
| 2021-05-19 17:13:42 | 1 | 6 | 9 | 32.9 | 60.2 | 806 |
| 2021-05-19 17:13:52 | 3 | 6 | 6 | 32.9 | 59.8 | 828 |
| 2021-05-19 17:14:03 | 3 | 3 | 4 | 32.9 | 59.8 | 756 |
| 2021-05-19 17:14:14 | 3 | 6 | 6 | 32.9 | 59.8 | 752 |
| 2021-05-19 17:14:24 | 1 | 3 | 3 | 32.9 | 59.8 | 720 |
| 2021-05-19 17:14:35 | 3 | 3 | 3 | 32.9 | 59.9 | 752 |

รูปที่ 5.2 ภาพตัวอย่างเมื่อคลิกเพื่อเข้าไปดูข้อมูลค่าฝุ่นละออง ณ เวลาต่าง ๆ ของบริเวณสระน้ำจุฬาฯ

ในส่วนที่นำข้อมูลไปแสดงผลผ่าน Dashboard ได้เลือกใช้ NETPIE Freeboard ในการแสดงผลข้อมูลแบบ Real time ของทั้ง 4 สถานี นอกจากนี้ภายในระบบตรวจวัด PM2.5 ด้วยระบบสมองกลฝังตัว ยังมีการกำหนดเงื่อนไขเมื่อปริมาณฝุ่นละออง PM2.5 เกินเกณฑ์ที่กำหนดไว้ ให้ส่งข้อความแจ้งเตือน พร้อมทั้งระบุปริมาณฝุ่นละอองที่เกินเกณฑ์ที่กำหนดไว้ไปยัง LINE Notify เพื่อป้องกันอันตรายจากมลพิษทางอากาศได้ทันที โดยการใช้ข้อมูลผ่านทางแอปพลิเคชัน Line และมีการเก็บข้อมูลปริมาณฝุ่นละออง อุณหภูมิ ความชื้นสัมพัทธ์ และปริมาณก๊าซคาร์บอนไดออกไซด์ของแต่ละวันไว้ในฐานข้อมูล MySQL เพื่อที่จะสามารถนำข้อมูลเหล่านั้นมาทำเป็นเว็บไซต์เพื่อนำไปแสดงผลได้ดังที่แสดงในบทที่ 4



รูปที่ 5.3 แผนภาพแสดงการทำงานของระบบสมองกลฝังตัว

5.2 ปัญหาที่พบ

1. Dashboard ของ NETPIE ไม่สามารถให้บุคคลอื่นเข้ามาดูข้อมูลได้ นอกจากจะเข้าระบบด้วยบัญชีของผู้ทำโครงการ
2. เว็บไซต์เพื่อดูค่าปริมาณฝุ่นละอองของบริเวณต่าง ๆ ภายในจุฬาลงกรณ์มหาวิทยาลัย ยังไม่สามารถทำเป็นแบบ Real time ได้
3. ระบบตรวจวัด PM2.5 ด้วยระบบสมองกลฝังตัวในโครงการนี้ยังไม่สามารถเก็บข้อมูลได้แบบ 24 ชั่วโมง เนื่องจากแหล่งพลังงานที่ใช้ไม่สามารถให้พลังงานได้ตลอด 24 ชั่วโมง

5.3 ข้อเสนอแนะ

จากการศึกษาของโครงการนี้ พบว่าเพื่อให้ระบบสามารถแสดงผลข้อมูลได้แบบ Real time จะต้องใช้ Dashboard และ Cloud server ของผู้พัฒนาคนอื่นและอาจจะมีค่าใช้จ่ายเพิ่มเติมจากการซื้อหรือเช่าพื้นที่ในการจัดเก็บและแสดงผลของข้อมูล นอกจากนี้การที่จะให้ระบบระบบตรวจวัด PM2.5 ด้วยระบบสมองกลฝังตัวนี้ทำงานได้ตลอด จำเป็นต้องต่อระบบเข้ากับแหล่งกำเนิดไฟฟ้า หรืออาจเพิ่มด้วยแหล่งจ่ายพลังงานแสงอาทิตย์ (Solar cell) และมีตัวปล่อยสัญญาณ Wi-Fi ตลอด 24 ชม. เพื่อที่จะสามารถเก็บข้อมูลปริมาณต่าง ๆ และแจ้งเตือนค่าปริมาณ PM2.5 ที่เกินเกณฑ์ที่กำหนดได้ตลอดเวลา

บรรณานุกรม

- [1] ตระวรรณ หาญกิจรุ่ง. (2556). อิทธิพลของปัจจัยทางอุตุนิยมวิทยาและการจราจรต่อปริมาณฝุ่นละอองขนาดเล็กไม่เกิน 10 ไมครอน ของพื้นที่ริมถนนในเขตกรุงเทพมหานคร. (วิทยานิพนธ์มหาบัณฑิต), จุฬาลงกรณ์มหาวิทยาลัย.
- [2] รพีพัฒน์ เกิร์กไกล. (2543). องค์ประกอบธาตุในฝุ่นละอองขนาดเล็กกว่า 2.5 ไมครอน ใน กรุงเทพมหานคร. (วิทยานิพนธ์มหาบัณฑิต), จุฬาลงกรณ์มหาวิทยาลัย.
- [3] US.EPA. (1996). Air Quality Criteria for Particulate Matter. Washington DC. United States.
- [4] US.EPA. (2010). Integrated Science Assessment for Particulate Matter. United States of America.
- [5] ศุภนิตา สรญาณธนาวุธ. (2558). DHT 11/22 Relative Humidity and Temperature Sensor. สืบค้นเมื่อ 25 พฤศจิกายน 2563. จาก <https://embed58.learninginventions.org/g1-4/>
- [6] ศูนย์วิทยบริการ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี. ลักษณะของการกระเจิง. สืบค้นเมื่อ 25 พฤศจิกายน 2563. จาก <https://etsrc.lib.kmutt.ac.th/interactive/cve424/chapter2/theory18.html>
- [7] ESP32 เบื้องต้น :: บทที่ 1 แนะนำ ESP32. (2560). สืบค้นเมื่อ 12 พฤศจิกายน 2563 จาก <http://www.ioxhop.com/article/62/esp32->
- [8] Building an ESP32 based IoT Weather Station with Ubidots. (2020) Retrieved 15 November 2020. from <https://www.hackster.io/shahizat005/building-an-esp32-based-iot-weather-station-with-ubidots-05adcc>
- [9] Sara Santos. (2561). ESP32 Pinout Reference: Which GPIO pins should you use?. Retrieved 15 November 2020. from <https://randomnerdtutorials.com/esp32-pinout-reference-gpios/>
- [10] SEN0177 PM2.5 laser dust sensor. Retrieved 20 November 2020. from https://wiki.dfrobot.com/PM2.5_laser_dust_sensor_SKU_SEN0177
- [11] Gas Sensor Getting Started. (2017). สืบค้นเมื่อ 20 พฤศจิกายน 2563. จาก <https://blog.thaieasyelec.com/getting-started-gas-sensor/>
- [12] NETPIE Introduction. สืบค้นเมื่อ 15 มกราคม 2564. จาก <https://netpie.gitbooks.io/doc/content/chapter1.html>
- [13] การใช้งาน FREEBOARD. สืบค้นเมื่อ 15 มกราคม 2564. จาก <https://netpie.gitbooks.io/freeboard-reference/content/>

- [14] MySQL คืออะไร?. สืบค้นเมื่อ 15 มกราคม 2564. จาก <https://saixiii.com/what-is-mysql/>
- [15] Suwat Nakchukaew. (2561). สร้างการแจ้งเตือนด้วย Line Notify. สืบค้นเมื่อ 20 พฤษภาคม 2564. จาก <https://engineering.thinknet.co.th/สร้างการแจ้งเตือนด้วย-line-notify-670f9b20ac27>
- [16] ความรู้เบื้องต้นเกี่ยวกับ ภาษา HTML. สืบค้นเมื่อ 20 พฤษภาคม 2564. จาก <https://sites.google.com/site/karchiporkaermdwyphasahhtml/khwam-ru-beuxng-tn-keiyw-kab-phasahhtml>
- [17] การสร้างตาราง. สืบค้นเมื่อ 22 พฤษภาคม 2564. จาก <https://sites.google.com/site/html2558/table>
- [18] JavaScript คืออะไร ใช้งานอย่างไร. (2020). สืบค้นเมื่อ 30 พฤษภาคม 2564. จาก <https://www.webdodee.com/what-is-javascript/#JavaScript-2>
- [19] การจัดรูปแบบเอกสาร. สืบค้นเมื่อ 20 พฤษภาคม 2564. จาก <https://sites.google.com/site/cirayus/sux-kar-reiyn-ru/4-kar-cad-rup-baeb-xeksar>
- [20] การแทรกรูปภาพและมัลติมีเดีย. สืบค้นเมื่อ 25 พฤษภาคม 2564. จาก <https://sites.google.com/site/html2558/images-and-multimedia>
- [21] html วิธีใส่เลขยกกำลังและตัวห้อย. สืบค้นเมื่อ 25 พฤษภาคม 2564. จาก <https://www.atdesigncm.com/blog/html->
- [22] Yogesh Singh. (2020). HTML – How to Show Text Above Image on Hover. Retrieved 27 May 2021. from <https://makitweb.com/html-how-to-show-text-above-image-on-hover>
- [23] กรมควบคุมมลพิษ. (2564). รายงานสถานการณ์และคุณภาพอากาศประเทศไทย. สืบค้นเมื่อ 2 มิถุนายน 2564. จาก <http://air4thai.pcd.go.th/webV2/download.php>

ภาคผนวก

```

#include "PMS.h"
#include "DHT.h"
#include <WiFi.h>
#include <HTTPClient.h>
#include <PubSubClient.h>
#include <SoftwareSerial.h>
#include <TridentTD_LineNotify.h>
#define DHTPIN 15
#define DHTTYPE DHT22
#define MQ135_PIN 34
DHT dht(DHTPIN, DHTTYPE);
PMS pms(Serial1);
PMS::DATA data;
const char* ssid = "iPhone";
const char* password = "1230802937359";
const char* mqtt_server = "broker.netpie.io";
const int mqtt_port = 1883;
const char* mqtt_Client = "04c1cdf-e78e-499c-95cd-c5955ddaaa1";
const char* mqtt_username =
"E7kmos1icg1FZmpiyzJjbZuEhfat63MM";
const char* mqtt_password =
"GVQu(1#rWo73-4lW4KAPON7qnnKwTsht";
#define LINE_TOKEN
"VC8pNChGrfBd180O9XxhqLIIVE1cz4aIJgqxE
SK1AI"
String pm1;
String pm2_5;
String pm10;
String temperature;
String humidity;
String co_2;
WiFiClient espClient;
PubSubClient client(espClient);
unsigned long previousMillis = 0;
unsigned long currentMillis = 0;
float PM_1_0, PM_2_5, PM_10_0;
unsigned int x=10;
void reconnect() {
while (!client.connected()) {
Serial.print("Attempting MQTT
connection...");
if (client.connect(mqtt_Client,
mqtt_username, mqtt_password)) {
Serial.println("connected");
client.subscribe("@msg/pm");
} else {
Serial.print("failed, rc=");
Serial.print(client.state());
Serial.println("try again in 5 seconds");
delay(5000);
}
}
}

```

```

void callback(char* topic, byte* payload,
unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  String message;
  for (int i = 0; i < length; i++) {
    message = message + (char)payload[i];
  }
  Serial.println(message);
  if (String(topic) == "@msg/pm") {
  }
}

void setup() {
  Serial.begin(9600);
  // Pre heat 30 วินาที
  Serial.println("Pre-heat sensor 30 seconds");
  delay(30000);
  Serial.println("Sensor ready start reading");
  Serial1.begin(PMS::BAUD_RATE,
SERIAL_8N1, 21, 22);
  dht.begin();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  client.setServer(mqtt_server, mqtt_port);
  client.setCallback(callback);
  // กำหนด Line Token
  LINE.setToken(LINE_TOKEN);
}

void loop() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT
sensor!");
    return;
  }
  int MQ135_data =
analogRead(MQ135_PIN);
  if (isnan(MQ135_data)){
    Serial.println("Failed to read form
MQ_135");
    return;
  }
  while (Serial1.available()) { Serial1.read(); }
  pms.requestRead();
}

```

```

        if (pms.readUntil(data))
    {
        Serial.print("PM 1.0 (ug/m3): ");
        Serial.println(data.PM_AE_UG_1_0);
        Serial.print("PM 2.5 (ug/m3): ");
        Serial.println(data.PM_AE_UG_2_5);
        Serial.print("PM 10.0 (ug/m3): ");
        Serial.println(data.PM_AE_UG_10_0);
        Serial.print("Humidity: ");
        Serial.print(h);
        Serial.print(" %\t");
        Serial.print("Temperature: ");
        Serial.print(t);
        Serial.println(" *C ");
        Serial.print("CO2 : ");
        Serial.print(MQ135_data); // analog data
        Serial.println(" PPM"); // Unit = part per
million
        Serial.println();
    }
    if (!client.connected()) {
        reconnect();
    }
    client.loop();
    //-----receive from PMS7003-----
    if (pms.read(data)) {
        //-----Sum PM value-----
        x++;
        PM_1_0 += data.PM_AE_UG_1_0;
        PM_2_5 += data.PM_AE_UG_2_5;
        PM_10_0 += data.PM_AE_UG_10_0;
    }
    //-----Update to NETPIE (Update to
shadow device)-----
    currentMillis = millis();
    if (currentMillis - previousMillis >= 10000) {
//ตั้งเวลาในการอัปเดตขึ้น NETPIE [มิลลิวินาที]
        previousMillis = currentMillis;

        //-----Average PM value-----
        PM_1_0 = PM_1_0 / x;
        PM_2_5 = PM_2_5 / x;
        PM_10_0 = PM_10_0 / x;
        //-----Send PM value to NETPIE-----
        String payload = "{\"data\": {\"pm1\": " +
String(data.PM_AE_UG_1_0)
        + ", \"pm2.5\": " +
String(data.PM_AE_UG_2_5)
        + ", \"pm10\": " +
String(data.PM_AE_UG_10_0)
        + ", \"temperature\": " + String(t,1)
        + ", \"humidity\": " + String(h,1)
        + ", \"co2\": " + String(co_2)
        + "}}";
        char msg[payload.length()+1];

```

```

    payload.toCharArray(msg,
(payload.length() + 1));
    Serial.println(msg);
    client.publish("@shadow/data/update",
msg);
pm1 = data.PM_AE_UG_1_0;
pm2_5 = data.PM_AE_UG_2_5;
pm10 = data.PM_AE_UG_10_0;
temperature = t;
humidity = h;
co_2 = MQ135_data;
//Connect database
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
http.begin("http://172.20.10.3/PM2_5/databa
se.php");
        http.addHeader("Content-Type",
"application/x-www-form-urlencoded");
        int httpCode =
http.POST("pm1="+pm1+"&pm2_5="+pm2_5
+"&pm10="+pm10+"&temperature="+tempe
rature+"&humidity="+humidity+"&co_2="+co
_2);
        if(httpCode == HTTP_CODE_OK)
        {
            Serial.println("Insert to database
success !!!");
        }
        else
        {
            Serial.println("Error in HTTP request");
        }
http.end();
    }
    else {
        Serial.println("WiFi Disconnected");
    }
// LINE Notify
int PM_Alert = 50; //ค่า PM2.5 ที่ต้องการให้
แจ้งเตือน
    if (data.PM_AE_UG_2_5 >= PM_Alert) {
        String msg = "ฝุ่นเกินแล้วจ้า PM2.5 : " +
String(data.PM_AE_UG_2_5) + " ug/m3"; //
ข้อความที่จะส่ง LINE
        Serial.println("Sending Line : " + msg);
        LINE.notify(msg);
    }
//-----Clear PM value-----
PM_1_0 = 0;
PM_2_5 = 0;
PM_10_0 = 0;
x = 0;
}
delay(10000);
}

```



```

        <td>6</td>    <td>32.9</td>
        <td>59.4</td><td>795</td>
</tr>
<tr align="center">
        <td>2021-05-19 17:11:45</td>
        <td>3</td>    <td>6</td>
        <td>6</td>    <td>32.9</td>
        <td>60</td>  <td>794</td>
</tr>
<tr align="center">
        <td>2021-05-19 17:11:56</td>
        <td>3</td>    <td>4</td>
        <td>4</td>    <td>32.9</td>
        <td>60.6</td><td>814</td>
</tr>
<tr align="center">
        <td>2021-05-19 17:12:07</td>
        <td>1</td>    <td>4</td>
        <td>4</td>    <td>32.9</td>
        <td>60.1</td><td>805</td>
</tr>
<tr align="center">
        <td>2021-05-19 17:12:17</td>
        <td>1</td>    <td>6</td>
        <td>7</td>    <td>32.9</td>
        <td>59.9</td><td>751</td>
</tr>
<tr align="center">

```

```

        <td>2021-05-19 17:12:28</td>
        <td>3</td>    <td>6</td>
        <td>6</td>    <td>32.9</td>
        <td>60.1</td><td>783</td>
</tr>
<tr align="center">
        <td>2021-05-19 17:12:38</td>
        <td>3</td>    <td>4</td>
        <td>4</td>    <td>32.9</td>
        <td>60.1</td><td>799</td>
</tr>
<tr align="center">
        <td>2021-05-19 17:12:49</td>
        <td>3</td>    <td>6</td>
        <td>6</td>    <td>32.9</td>
        <td>59.7</td><td>814</td>
</tr>
<tr align="center">
        <td>2021-05-19 17:12:59</td>
        <td>3</td>    <td>4</td>
        <td>4</td>    <td>32.9</td>
        <td>59.7</td><td>791</td>
</tr>
<tr align="center">
        <td>2021-05-19 17:13:10</td>
        <td>3</td>    <td>4</td>
        <td>4</td>    <td>32.9</td>
        <td>60.2</td><td>768</td>

```

```

</tr>
<tr align="center">
  <td>2021-05-19 17:13:21</td>
  <td>3</td> <td>7</td>
  <td>7</td> <td>32.9</td>
  <td>60.1</td><td>761</td>
</tr>
<tr align="center">
  <td>2021-05-19 17:13:31</td>
  <td>1</td> <td>6</td>
  <td>6</td> <td>32.9</td>
  <td>60.2</td><td>785</td>
</tr>
<tr align="center">
  <td>2021-05-19 17:13:42</td>
  <td>1</td> <td>6</td>
  <td>9</td> <td>32.9</td>
  <td>60.2</td><td>806</td>
</tr>
<tr align="center">
  <td>2021-05-19 17:13:52</td>
  <td>3</td> <td>6</td>
  <td>6</td> <td>32.9</td>
  <td>59.8</td><td>828</td>
</tr>
<tr align="center">
  <td>2021-05-19 17:14:03</td>
  <td>3</td> <td>3</td>
  <td>4</td> <td>32.9</td>
  <td>59.8</td><td>756</td>
</tr>
<tr align="center">
  <td>2021-05-19 17:14:14</td>
  <td>3</td> <td>6</td>
  <td>6</td> <td>32.9</td>
  <td>59.8</td><td>752</td>
</tr>
<tr align="center">
  <td>2021-05-19 17:14:24</td>
  <td>1</td> <td>3</td>
  <td>3</td> <td>32.9</td>
  <td>59.8</td><td>720</td>
</tr>
<tr align="center">
  <td>2021-05-19 17:14:35</td>
  <td>3</td> <td>3</td>
  <td>3</td> <td>32.9</td>
  <td>59.9</td><td>752</td>
</tr>

```