

# Final Project Report

Group สวัสดิ์วันอังคาร

6431343021 Werapat Wangrungrroj

6432022721 Jirayut Phothawilkiat

6432179621 Sorawit Kamphoi

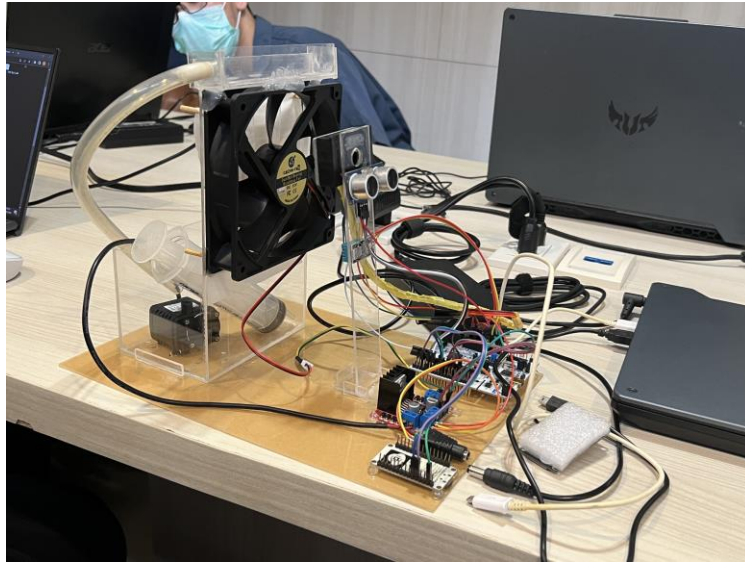
6431317721 Nattapong Anansomsin

2110366 Embedded System Laboratory

Semester 2/2022

Chulalongkorn University

## ภาพรวมของโครงการ (Project Overview)



โครงการนี้มีจุดประสงค์เพื่อลดปริมาณฝุ่นที่สะสมอยู่ในอากาศและอำนวยความสะดวกสบายให้แก่ผู้ใช้โดยลดอุณหภูมิของอากาศด้วยน้ำ โดยนำ sensor มาใช้เพื่อวัดปริมาณฝุ่น อุณหภูมิของอากาศ และความชื้นในอากาศ เพื่อให้พัดลมกรองฝุ่นทำงานตามสภาพอากาศ และใช้น้ำเพื่อลดอุณหภูมิของอากาศ พร้อมทั้งแสดงปริมาณฝุ่น อุณหภูมิ และความชื้นด้วยกราฟ ทั้งนี้ผู้ใช้สามารถตั้งค่าให้อุปกรณ์ทำงานด้วยตัวเองแบบอัตโนมัติหรือควบคุมอุปกรณ์ด้วยตัวเองได้ตามต้องการผ่านทางเว็บไซต์

สามารถดู source code และเว็บไซต์ได้ตามลิงก์ดังนี้

Source Code : <https://github.com/creampiney/final-embedded-lab>

Website : <https://final-project-embedded.netlify.app/>

## อุปกรณ์ (Components)

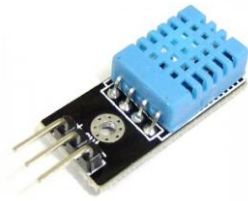
1. STM32 NUCLEO-F411RE



2. ESP8266 NodeMCU



3. DHT11 (Humidity & Temperature Sensor)



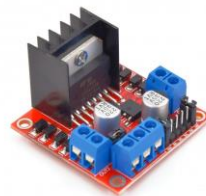
4. SHARP GP2Y1010AU0F (Dust Sensor)



5. HC-SR04 (Ultrasonic Sensor)



6. L298N Dual H Bridge



7. Water Pump DC 12V



8. PWM Fan DC 12V



9. Adapter DC 12V



10. สายยาง

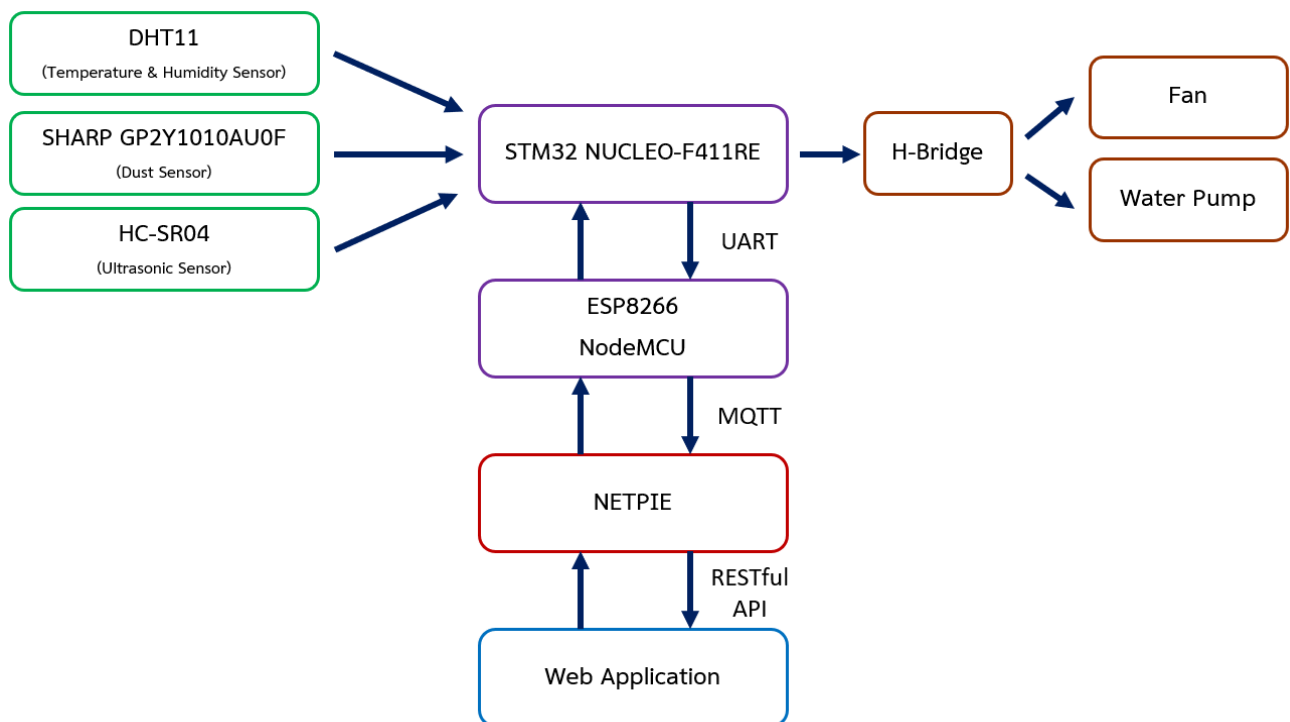


## สถาปัตยกรรมระบบ (System Architecture)

โครงงานนี้ใช้ STM32 NUCLEO-F411RE เป็น microcontroller ในการควบคุมพัดลมและปั้มน้ำผ่าน H-Bridge และต่อ sensor 3 ตัวได้แก่ temperature & humidity sensor (DHT11), dust sensor (SHARP GP2Y1010AU0F), และ ultrasonic sensor (HC-SR04) ข้อมูลจากเซนเซอร์ต่างๆจะส่งมายัง NodeMCU ผ่านทาง UART

เราใช้ NodeMCU เพื่อเชื่อมต่อกับ wifi เพื่อทำให้สามารถส่งข้อมูลและรับข้อมูลจาก NETPIE ได้ผ่านทาง MQTT protocol

ทั้งนี้เราสามารถดูข้อมูลจากอุปกรณ์รวมถึงควบคุมอุปกรณ์ได้ผ่านทาง web application ซึ่งเชื่อมต่อ NETPIE ด้วย RESTful API



## หลักการทำงานของอุปกรณ์ (Device Principle)

อุปกรณ์จะอ่านค่าอุณหภูมิ, ความชื้นและระยะทางทุกๆ 200 มิลลิวินาที และหาค่าปริมาณฝุ่นเฉลี่ยโดยการอ่านค่าปริมาณฝุ่นทุกๆ 100 มิลลิวินาที เป็นจำนวน 50 ครั้งแล้วนำมาเฉลี่ยทุกๆ 10 วินาที

อุปกรณ์มีสองรูปแบบการทำงาน ได้แก่รูปแบบอัตโนมัติและรูปแบบควบคุมด้วยตัวเอง

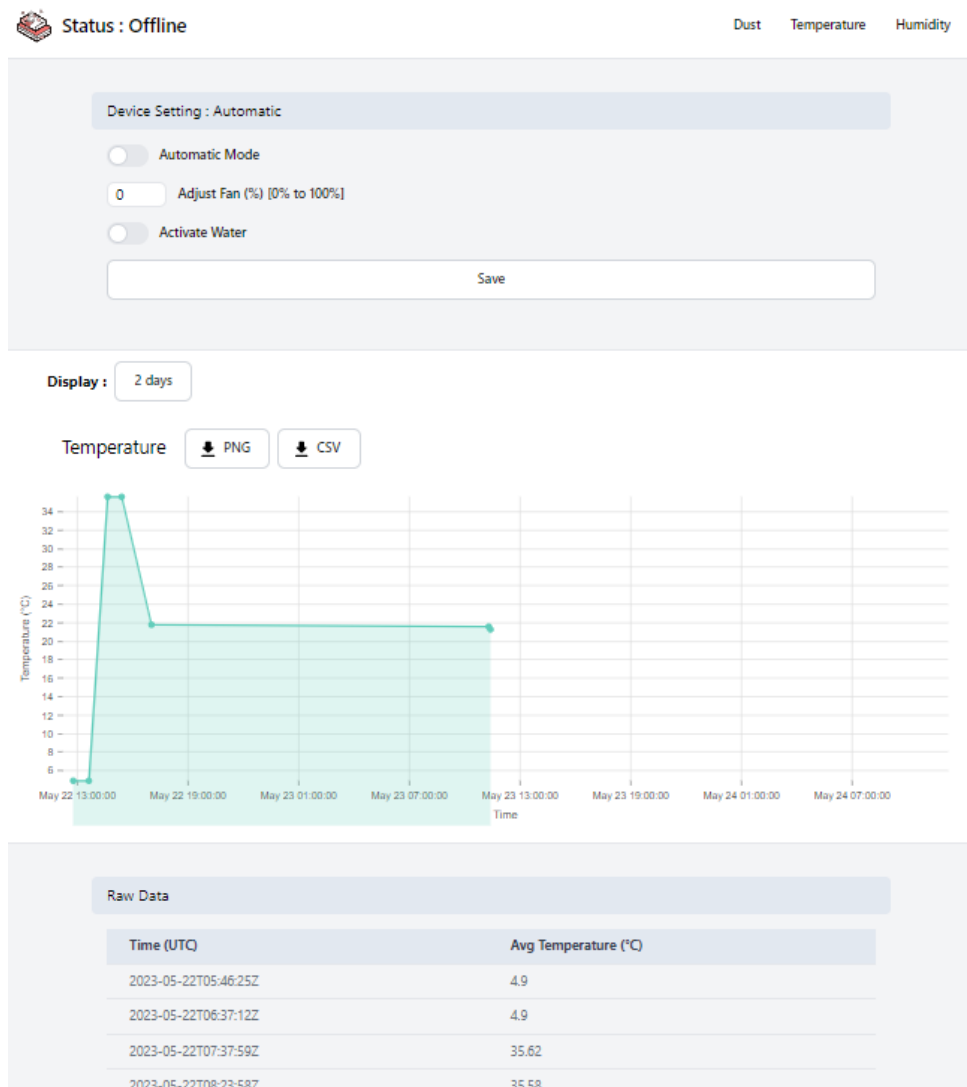
ในรูปแบบอัตโนมัติอุปกรณ์จะมีลักษณะการทำงานดังนี้

1. เมื่อมีการทำงานความเร็วพัดลมขั้นต่ำจะอยู่ที่ 50%
2. ความเร็วพัดลมจะปรับอัตโนมัติตามอุณหภูมิและค่าปริมาณฝุ่นในขณะนั้น โดยเมื่ออุณหภูมิมากกว่า 25 องศาเซลเซียส หรือ ปริมาณฝุ่นมากกว่า 5 ไมโครกรัมต่อลูกบาศก์เมตร อุปกรณ์จะปรับความเร็วพัดลมขึ้นทีละ 1% จนถึง 100% หรือจนกว่าอุณหภูมิจะน้อยกว่าเท่ากับ 25 องศาเซลเซียส และ ปริมาณฝุ่นน้อยกว่าเท่ากับ 5 ไมโครกรัมต่อลูกบาศก์เมตร
3. เมื่ออุณหภูมిన้อยกว่าเท่ากับ 25 องศาเซลเซียส และ ปริมาณฝุ่นน้อยกว่าเท่ากับ 5 ไมโครกรัมต่อลูกบาศก์เมตร ความเร็วพัดลมจะถูกปรับเป็น 50%
4. บิ๊มน้ำจะทำงานเมื่อความชื้นน้อยกว่าเท่ากับ 75% และหยุดทำงานเมื่อความชื้นมากกว่าเท่ากับ 80% โดยบิ๊มน้ำจะมีความแรงคงที่อยู่ที่ 5%
5. ถ้าไม่มีการเคลื่อนไหวมากกว่า 1 นาทีจะหยุดการทำงานของพัดลมและบิ๊มน้ำ โดยจะถือว่าไม่มีการเคลื่อนไหวเมื่อค่าระยะทางมากกว่า 50 เซนติเมตร

ในรูปแบบควบคุมด้วยตัวเองผู้ใช้จะสามารถปรับความเร็วพัดลมได้ตั้งแต่ 0% ถึง 100% และสามารถควบคุมการเปิดปิดของบิ๊มน้ำได้

## Website and Features

ข้อมูลของปริมาณฝุ่น อุณหภูมิ และความชื้น รวมทั้งสถานะของอุปกรณ์และการตั้งค่าต่างๆสามารถดูได้ในเว็บไซต์ <https://final-project-embedded.netlify.app> โดยเว็บไซต์ถูกเขียนด้วย react framework และเชื่อมต่อกับ NETPIE 2020 ผ่าน RESTful API



Features ต่างๆที่ผู้ใช้งานสามารถดูแลควบคุมได้มีดังนี้

1. ผู้ใช้สามารถเลือกที่จะควบคุมการทำงานของพัดลมและปั้มน้ำตามความต้องการผู้ใช้ หรือจะให้ปรับค่าดังกล่าวตามความเหมาะสมโดยอัตโนมัติก็ได้ อีกทั้งยังแสดงสถานะปัจจุบันของอุปกรณ์ว่าเป็นแบบอัตโนมัติหรือตั้งค่าเอง

Device Setting : Automatic

☒ Automatic Mode

Save

Device Setting : Automatic

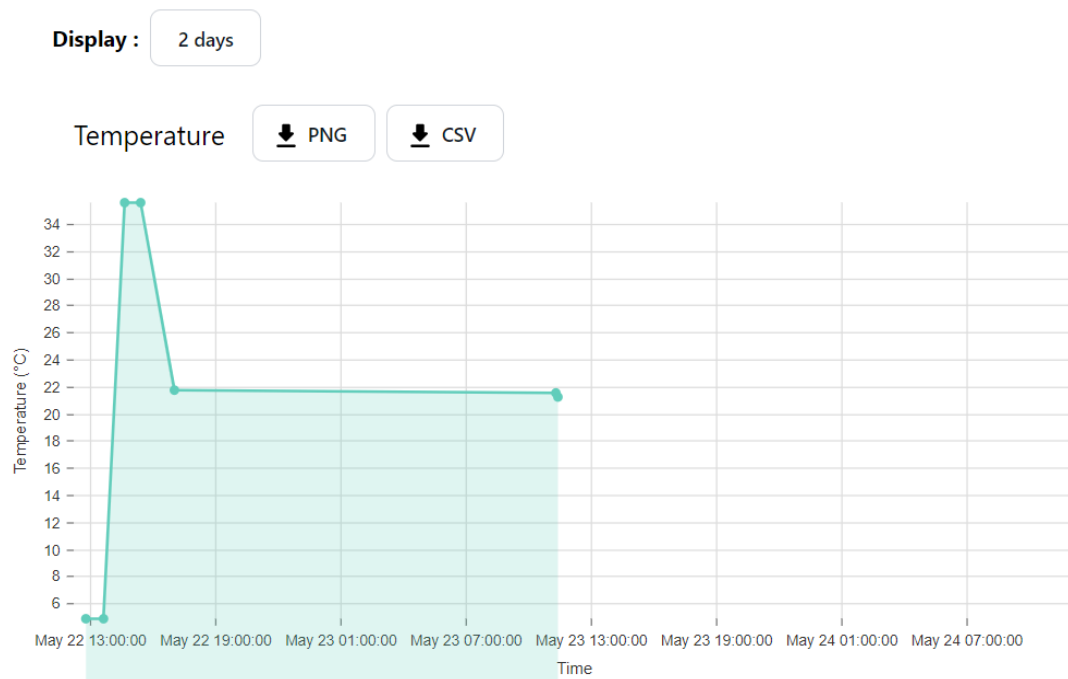
☐ Automatic Mode

Adjust Fan (%) [0% to 100%]

☐ Activate Water

Save

2. ผู้ใช้สามารถดูของค่าความหนาแน่นฝุ่น อุณหภูมิ และความชื้นของอากาศได้ผ่านทางกราฟที่แสดงบนเว็บไซต์



- ผู้ใช้สามารถเปลี่ยนค่าที่แสดงบนกราฟให้เป็นความหนาแน่นฝุ่น อุณหภูมิ หรือความชื้นได้ทางมุมขวาบนของเว็บไซต์

Dust Temperature Humidity

- ผู้ใช้สามารถปรับระยะเวลาที่แสดงบนกราฟได้ตามต้องการ ซึ่งมี 2 แบบ
    - a) แสดงกราฟแบบ realtime ซึ่งจะดึงข้อมูลจากฐานข้อมูลทุกๆ 30 วินาที
    - b) แสดงกราฟย้อนหลังโดยเฉลี่ย ซึ่งสามารถปรับระยะเวลาได้ใน dropdown
- ผู้ใช้สามารถปรับได้ที่บริเวณ Display ดังภาพ

Display : Realtime

3. ผู้ใช้สามารถดูสถานะของเครื่องว่าเปิดหรือปิดได้ที่มุมซ้ายบนของเว็บไซต์



Status : Offline

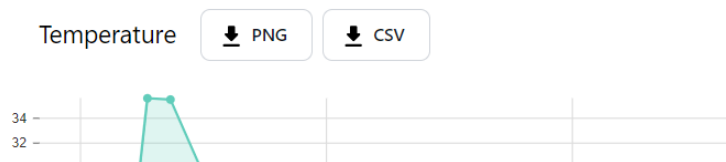


Status : Online

4. ผู้ใช้สามารถดูข้อมูลดิบ (raw data) ของข้อมูลที่แสดงบนกราฟได้ที่ด้านล่างของกราฟดังภาพ

Raw Data	
Time (UTC)	Avg Temperature (°C)
2023-05-22T05:46:25Z	4.9
2023-05-22T06:37:12Z	4.9
2023-05-22T07:37:59Z	35.62
2023-05-22T08:23:58Z	35.58
2023-05-22T10:01:00Z	21.77
2023-05-23T04:17:41Z	21.6
2023-05-23T04:23:37Z	21.31

5. ผู้ใช้สามารถ download รูปของกราฟเป็นไฟล์ png หรือเลือก download ข้อมูลดิบซึ่ง export เป็นไฟล์ csv ได้ที่ปุ่มดาวน์โหลดที่หัวข้อของกราฟดังภาพ





## หน้าที่และความรับผิดชอบ (Roles and Responsibility)

- Werapat Wangrungrroj (6431343021)

### Frontend Developing

- เชื่อมเว็บไซต์ให้สามารถส่ง POST หรือ PUT requests ถึง NETPIE ได้ โดยการใช้ RESTful api ของ NETPIE
- แสดงข้อมูลดิบ (raw data) ด้านล่างของกราฟโดยสร้างเป็นตาราง และให้แสดงตามที่ใช้เลือกไว้ (เช่นหากกราฟแสดงข้อมูลอุณหภูมิ ข้อมูลดิบที่แสดงจะเป็นข้อมูลอุณหภูมิเหมือนกัน)
- เขียนฟังก์ชันและระบบที่สามารถ export ข้อมูลดิบ (raw data) ออกมาเป็นไฟล์ CSV ได้ ด้วยการไล่แต่ละ tr element และ td หรือ th element แล้วนำ data มาต่อเป็น string ยาวๆเพื่อให้สามารถ export เป็นไฟล์ text csv ได้
- เขียนฟังก์ชันที่สามารถดึงข้อมูลจาก NETPIE ด้วยการ fetch method ของ RESTful api และจัดเรียงข้อมูลให้ตรงตามที่สามารถแสดงบนกราฟได้
- สร้าง prototype react component เพื่อนำมาทดสอบการทำงานของ function การส่งข้อมูลหรือการรับข้อมูลของ frontend

### NodeMCU Implementation

- เขียนฟังก์ชันรับข้อมูลปริมาณฝุ่น อุณหภูมิ และความชื้นจาก STM32 ด้วย UART
- เขียนฟังก์ชันส่งข้อมูลใหม่คของอุปกรณ์ที่ต้องการเปลี่ยนซึ่งรับมาจากการ subscribe topic `@msg/states` และ `@private/shadow/data/get/response` แล้วส่งไปยัง STM32 เพื่อปรับเปลี่ยน parameter ของความเร็วพัดลม การเปิดปิดของปั้มน้ำ และการปรับเปลี่ยนระหว่างระบบ Automatic กับแบบ Manual

### STM32 Implementation

- ส่งข้อมูลปริมาณฝุ่น อุณหภูมิ และความชื้นไปยัง NodeMCU ผ่านทาง UART transmit
- เขียนฟังก์ชันรอรับข้อมูลใหม่คของอุปกรณ์จาก NodeMCU ด้วย receive interrupt

- Jirayut Phothawilkiat (6432022721)

## Team Management

- กำหนดวัตถุประสงค์ของงาน
- แบ่งงานให้สมาชิกในกลุ่มตามที่สมาชิกต้องการและตามความถนัด
- นัดหมายการทำงานที่มหาวิทยาลัย

## STM32 Implementation

- อ่านค่าอุณหภูมิและความชื้นจาก sensor DHT11 โดยต่อผ่าน GPIO input และเรียกอ่านค่าด้วยฟังก์ชัน DHT11\_Read() โดยในฟังก์ชันจะมีการ set pin เป็น input และ output เอง ใช้ delay 200 เพื่ออ่านค่าทุกๆ 200 ms
- อ่านค่าความหนาแน่นฝุ่นจาก sensor GP2Y1010AU0F ผ่าน ADC ใช้ delay 100 เพื่ออ่านค่าทุกๆ 100 ms
- อ่านค่าระยะห่างจาก sensor Ultrasonic โดยต่อขา trig กับ GPIO output และขา echo กับ GPIO input ใช้ delay 200 เพื่ออ่านค่าทุกๆ 200 ms
- ควบคุมความเร็วพัดลมและปั๊มน้ำผ่าน L298N Dual H Bridge โดยต่อขา in2และin3 เข้ากับ GPIO output ที่set output เป็น 1 เพื่อใช้ขาout2และout3 เป็นขาground 12V ของพัดลมและปั๊มน้ำ และต่อขา inAและinB เข้ากับ TIM ที่ตั้งค่าเป็น PWM generation เพื่อควบคุมความเร็ว
- ต่อรวมsensorทั้งหมดโดยแยกการทำงานเป็น Thread Thermal\_and\_Humidity, Dust และ Ultrasonic เพื่ออ่านค่าจากsensor และควบคุมความเร็วพัดลมและปั๊มน้ำในThread Control ตามหลักการทำงานของอุปกรณ์ข้างต้น
- เพิ่ม Thread Average ที่จะเอาค่าจากsensor Dust ทุกๆ200ms 50ครั้ง และนำมาหาค่าเฉลี่ยออกทุกๆ10s และจะใช้ osDealy ในThreadนี้นับเวลาถอยหลัง1นาที่เมื่อไม่มีการเคลื่อนไหว
- เพิ่มThread MCUsend เพื่อเรียกฟังก์ชัน sendData() ส่งค่าอุณหภูมิ ความชื้น และฝุ่นให้ node MCU ทุกๆ 30s

## เชื่อมต่ออุปกรณ์ต่างๆ สร้างโมเดล

- Sorawit Kamphoi (6432179621)

## Frontend Developing

- สร้างเว็บไซต์ด้วย html, css, typescript ด้วย react framework โดยใช้ tailwind css เป็น css framework เพื่ออำนวยความสะดวกในการเขียน style ของเว็บไซต์ โดยออกแบบตามที่ UX/UI designer ออกแบบไว้และเพิ่มเติมบางส่วน
- เชื่อมต่อเว็บไซต์กับ NETPIE 2020 ด้วย RESTful API เพื่อ fetch ข้อมูลดังนี้
  - a) สถานะของอุปกรณ์ว่าอุปกรณ์กำลังเปิดอยู่หรือไม่
  - b) ข้อมูลปริมาณฝุ่น อุณหภูมิ และความชื้นของอากาศ
  - c) การตั้งค่าของอุปกรณ์ว่ากำลังทำงานแบบอัตโนมัติหรือควบคุมด้วยตัวเอง หากควบคุมด้วยตนเองจะสามารถควบคุมความแรงของพัดลมและการปล่อยน้ำได้
- ออกแบบและสร้างกราฟที่แสดงข้อมูลฝุ่น อุณหภูมิ และความชื้นของอากาศ
- เขียนฟังก์ชันที่ปรับช่วงเวลาในการแสดงข้อมูลของกราฟ
- สร้างระบบการ export กราฟออกมาเป็นไฟล์ PNG
- Host เว็บไซต์บน Netlify : <https://final-project-embedded.netlify.app/>
- Upload source code บน Github : <https://github.com/creampiney/final-embedded-lab>

## NodeMCU Implementation

- เชื่อมต่อกับ wifi และเชื่อมต่อกับ NETPIE ด้วย MQTT protocol หากเชื่อมต่อไม่สำเร็จจะทำการพยายามเชื่อมต่อ NETPIE ไปเรื่อยๆจนกว่าจะสำเร็จ
- ทำการส่งข้อมูล (publish) ด้วย topic **@shadow/data/update** ได้แก่ปริมาณฝุ่น อุณหภูมิ และความชื้น และ topic **@shadow/data/get** ซึ่งมาจาก STM32 เพื่อขอข้อมูล setting ของอุปกรณ์ตอนเริ่มต้น
- ทำการ subscribe message เพื่อรับ message ด้วย topic **@msg/state** และ **@private/shadow/data/get/response** ซึ่งมาจากการเปลี่ยนโหมดของอุปกรณ์ (อัตโนมัติ/ควบคุมเอง) จากเว็บไซต์ และ STM32 ตามลำดับ

## STM32 Implementation

- ทดลองและทดสอบ dust sensor

- Nattapong Anansomsin (6431317721)

### UX/UI Designing

- ออกแบบเว็บไซต์โดยแบ่งเว็บไซต์ออกเป็น 4 ส่วนได้แก่
  - a) Header ซึ่งจะบอกสถานะของอุปกรณ์ว่าเปิดหรือปิด
  - b) Panel สำหรับเปลี่ยนโหมดของอุปกรณ์ว่าให้ทำงานอัตโนมัติหรือควบคุมเอง
  - c) Graph สำหรับแสดงข้อมูลต่างๆ
  - d) Raw data ให้แสดงข้อมูลดิบที่ใช้แสดงบนกราฟ
- ออกแบบกราฟว่าควรปรับช่วงเวลาที่จะแสดงแบบใดบ้าง ได้แก่ realtime และดูข้อมูลย้อนหลัง

### Backend Developing

- สร้าง account สำหรับ NETPIE 2020 เพื่อใช้เป็น backend ของ project
- เขียน schema ของข้อมูลที่จะเก็บใน feed หรือ database

### STM32 Implementation

- ทดลองและทดสอบ temperature & humidity sensor และ ultrasonic sensor

## Project Timeline

8/14/2566 - 14/04/2566	<ul style="list-style-type: none"><li>- ประชุมเพื่อคิด project</li></ul>
15/04/2566 - 21/04/2566	<ul style="list-style-type: none"><li>- สรุปลำดับ project</li><li>- Brief project กับอาจารย์และเริ่มทำงาน</li></ul>
22/04/2566 - 28/04/2566	<ul style="list-style-type: none"><li>- แบ่งงานให้สมาชิกในกลุ่ม</li><li>- สร้าง account NETPIE 2020</li><li>- เริ่มออกแบบ web application</li></ul>
16/05/2566	<ul style="list-style-type: none"><li>- รับอุปกรณ์</li><li>- ทดลองต่อ sensor ต่างๆ</li><li>- เริ่มเขียน web application</li></ul>
17/05/2566	<ul style="list-style-type: none"><li>- เริ่มทำในส่วน STM32 implementation</li><li>- เริ่มต่อ sensor, พัดลม, และปั๊มน้ำ บน microcontroller</li></ul>
18/05/2566	<ul style="list-style-type: none"><li>- เริ่มทำในส่วน NodeMCU implementation</li><li>- ทดลองใช้ RESTful api กับ MQTT ของ NETPIE</li></ul>
19/05/2566	<ul style="list-style-type: none"><li>- ทดลองการส่งข้อมูลจาก NodeMCU กับ NETPIE และ STM32</li></ul>
20/05/2566	<ul style="list-style-type: none"><li>- implementation เกี่ยวกับการ fetch data ระหว่าง frontend กับ backend</li></ul>
21/05/2566	<ul style="list-style-type: none"><li>- ทำ UI เพื่อแสดงผลข้อมูลที่ได้จากการ fetch data จาก backend</li><li>- ทดสอบ project</li></ul>
22/05/2566	<ul style="list-style-type: none"><li>- ทดลองการควบคุมอุปกรณ์จาก web application</li><li>- เขียน feature บางส่วนใน web application เพิ่มเติมเช่นการ export ไฟล์</li></ul>
23/05/2566	<ul style="list-style-type: none"><li>- ทดสอบ project ครั้งสุดท้าย</li></ul>
24/05/2566	<ul style="list-style-type: none"><li>- ทำรายงาน</li></ul>