ระบบตรวจจับแก๊ส CO รั่วภายในรถยนต์พร้อมแจ้งเตือน

In-car CO Gas Detection and Alert System

ธนนท์ ดอกรักกลาง

Thanon Dokrakklang

ธิติพนธ์ เทียนแย้ม

Thitipon Tienyam

ชัชนันท์ เปรมศรี

Chatchanan Premsri

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยศรีนครินทรวิโรฒ ปีการศึกษา 2562

ระบบตรวจจับแก๊ส CO รั่วภายในรถยนต์พร้อมแจ้งเตือน

In-car CO Gas Detection and Alert System

ธนนท์ ดอกรักกลาง

Thanon Dokrakklang

ธิติพนธ์ เทียนแย้ม

Thitipon Tienyam

ชัชนันท์ เปรมศรี

Chatchanan Premsri

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยศรีนครินทรวิโรฒ ปีการศึกษา 2562

คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

ชื่อหัวข้อโครงงาน	ระบบตรวจจับแก๊ส CO รั่วภายในรถยนต์พร้อมแจ้งเตือน		
	In-car CO Gas Detection and Alert System		
นิสิต	นายธนนท์	ดอกรักกลาง	59102010254
	นายธิติพนธ์	เทียนแย้ม	59102010257
	นายชัชนันท์	เปรมศรี	59102010901
ปริญญา	วิทยาศาสตร์บัณฑิต (วท.บ.)		
ภาควิชา	วิทยาการคอมพิวเตอร์		
อาจารย์ที่ปรึกษาโครงงาน	ผู้ช่วยศาสตราจารย์ ศศิวิมล สุขพัฒน์		

ลงชื่อ.....

(ผู้ช่วยศาสตราจารย์ ศศิวิมล สุขพัฒน์)

อาจารย์ที่ปรึกษาโครงงาน

บทคัดย่อ

งานวิจัยเรื่องนี้มีวัตถุประสงค์เพื่อพัฒนาระบบตรวจจับแก๊สคาร์บอนมอนออกไซด์รั่วภายในรถยนต์โดยใช้ไมโครคอนโทรลเลอร์ NodeMCU และ MQ7 sensor ซึ่งมีไว้สำหรับตรวจจับแก๊สคาร์บอนมอนออกไซด์ โดยนำปริมาณแก๊ซคาร์บอนมอนออกไซด์ที่อ่านได้จากเซ็นเซอร์มาเข้าสู่กระบวนการวิเคราะห์เชิงลอจิก หากปริมาณแก๊ซคาร์บอนมอนออกไซด์ที่อ่านได้เกินกว่าเกณฑ์ที่กำหนดไว้ ระบบจะมีการส่งค่าที่อ่านได้และแจ้งเตือนไปยังผู้ใช้ผ่านแอปพลิเคชัน Line Notify นอกจากนี้ปริมาณค่าแก๊ซที่ได้จะส่งต่อไปที่ระบบ ThingSpeak เพื่อแสดงผลในรูปแบบกราฟ และสังเกตแนวโน้มการรั่วไหลของแก๊สคาร์บอนมอนออกไซด์โดยใช้โปรแกรม MatLab

จากผลการทดลองของงานวิจัยนี้พบว่า ระบบตรวจจับแก๊สคาร์บอนมอนออกไซด์รั่วภายในรถยนต์ที่พัฒนาสามารถตรวจวัดค่าแก๊สคาร์บอนมอนออกไซด์ได้ โดยมาจากการทดลองทั้ง 3 รูปแบบ คือลดกระจกรถครึ่งนึง, เปิดระบบระบายอากาศแบบหมุนเวียน และเปิดระบบดึงอากาศภายนอกเข้ามาในรถ โดยแต่ละรูปแบบทดลองที่ความเร็ว 0, 40 และ 80 กิโลเมตรต่อชั่วโมง รูปแบบละ 10 ครั้ง ซึ่งระบบสามารถแจ้งเตือนมาที่ผู้ใช้ได้ผ่าน Line Notify เมื่อระบบตรวจจับแก๊สได้เกินมาตรฐาน ทำให้ผู้ใช้สามารถรับรู้และรับมือกับสถานการณ์ที่มีการรั่วไหลของแก๊ส CO ภายในรถยนต์ได้

Abstract

This research aims to develop a carbon monoxide leakage detection system using the NodeMCU microcontroller with MQ7 sensor. The amount of carbon monoxide which detected from sensor will be analyzed through a logic operation. If results from sensor exceed than our thresholds, the CO detection system will send the result and send notifications to the user using Line Notify. In addition, the results will show in a form of graph on ThingSpeak system and predict the trend of CO leakage on MATLAB.

The result of our experiment, CO detection system was able to detect CO gas correctly. From the preliminary experiment of 3 types: half windows opened, air-condition on recirculation mode and air-condition on fresh air mode. Each mode tested at car speed 0, 40 and 80 km/hr., 10 times testing per each mode, the system was able to alert, our system were able to inform the user through the Line notify application when the system found any CO exceeds than the threshold. Therefore, user will acknowledge and can prepare to respond quickly if there is a CO leakage.

กิตติกรรมประกาศ

งานวิจัยเรื่อง ระบบตรวจจับแก๊ส CO รั่วภายในรถยนต์พร้อมแจ้งเตือน (In-car CO Gas Detection and Alert System) สามารถสำเร็จลุล่วงได้ด้วยความกรุณาของอาจารย์ที่ปรึกษาโครงงาน ผศ.ศศิวิมล สุขพัฒน์ ที่ได้กรุณาเสียสละเวลาให้คำปรึกษา คำแนะนำ และข้อเสนอแนะต่างๆ ตลอดจนการแก้ไขปัญหาหรือข้อบกพร่องที่เกิดขึ้นระหว่างการดำเนินโครงงาน คณะผู้จัดทำขอขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบคุณภาควิชาวิทยาการคอมพิวเตอร์ สำหรับเครื่องมือ เครื่องใช้ และอุปกรณ์ต่างๆ ที่ใช้ในการดำเนินงานในครั้งนี้ อีกทั้งขอขอบพระคุณอาจารย์ประจำภาควิชาวิทยาการคอมพิวเตอร์ทุกท่านที่ได้ให้ความรู้ ให้คำปรึกษา และคำแนะนำที่เป็นประโยชน์ทำให้คณะผู้จัดทำ สามารถปรับปรุงแก้ไขการดำเนินโครงงานในครั้งนี้ให้ดียิ่งขึ้นได้

ขอขอบคุณมหาวิทยาลัยศรีนครินทรวิโรฒ ซึ่งเป็นสถานที่ศึกษาหาความรู้ที่เป็นประโยชน์ในการดำเนินโครงงาน และอำนวยความสะดวกสำหรับสถานที่ในการดำเนินโครงงานในครั้งนี้ รวมถึงให้ประสบการณ์จากการทำกิจกรรมต่างๆ เพื่อที่จะนำไปประยุกต์ใช้ในการทำงานและการดำเนินชีวิตต่อไป

ขอขอบพระคุณบิดา มารดา รวมถึงครอบครัวของคณะผู้จัดทำ ที่มีส่วนช่วยเหลือในการดำเนินการทดลองต่างๆ เพื่อให้ได้ผลการทดลองที่คณะผู้จัดทำคาดหวังไว้ อีกทั้งเป็นผู้ให้โอกาสการศึกษาอันมีค่ายิ่งและเป็นกำลังใจที่ดีในการทำโครงงานตลอดมา

สุดท้ายขอขอบพระคุณเพื่อนๆ ร่วมชั้นปีทุกคนที่ให้ความช่วยเหลือและคำแนะนำที่เป็นประโยชน์ในการดำเนินโครงงาน รวมถึงผู้มีส่วนเกี่ยวข้องทุกท่านที่ทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี คณะผู้จัดทำหวังเป็นอย่างยิ่งว่าโครงงานเรื่อง ระบบตรวจจับแก๊ส CO รั่วภายในรถยนต์พร้อมแจ้งเตือน (In-car CO Gas Detection and Alert System) จะเป็นประโยชน์ส่วนหนึ่งต่อผู้สนใจศึกษาเพื่อสามารถนำไปพัฒนาต่อยอดในอนาคตได้ให้ดียิ่งๆ ขึ้นไป

คณะผู้จัดทำ

สารบัญ

บทคัดย่อ ..... **ก**

Abstract ..... **ข**

กิตติกรรมประกาศ ..... **ค**

สารบัญ ..... **ง**

สารบัญรูปภาพ ..... **ช**

สารบัญตาราง ..... **ญ**

บทที่ 1 บทนำ **1**

1.1 ที่มาและความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบเขตของโครงงาน 1

1.4 ประโยชน์ที่คาดว่าจะได้รับ 2

บทที่ 2 องค์ความรู้ที่เกี่ยวข้อง **3**

2.1 ทฤษฎีและแนวคิดที่ใช้ในการทำวิจัย 3

2.1.1 NodeMCU Devkit 1.0 (V2) 3

2.1.2 MQ7 Gas Sensor 4

2.1.3 ทฤษฎี IoT 5

2.1.4 ทฤษฎี Cloud Computing 6

2.1.5 ทฤษฎีแก๊สคาร์บอนมอนอกไซด์ 7

2.2 งานวิจัยที่เกี่ยวข้อง 8

2.2.1 งานวิจัยเรื่อง “Vehicle Gas Leakage Detector” 8

2.2.2 งานวิจัยเรื่อง “Is in cabin exposure to carbon monoxide and fine particulate matter amplified by the vehicle s self pollution potential? Quantifying the rate of exhaust intrusion” 9

2.2.3 งานวิจัยเรื่อง “IoT Based Car Pollution Detection Using AWS” 9

บทที่ 3 วิธีดำเนินโครงงาน 11

3.1 แผนการดำเนินงาน 11

3.1.1 เริ่มต้นและวางแผนโครงการ 11

3.1.2 การวิเคราะห์ระบบ 11

3.1.3 การออกแบบระบบ 11

3.1.4 การพัฒนาและติดตั้งระบบ 11

3.2 แผนการดำเนินงานตลอดโครงงาน 12

3.3 อุปกรณ์และเครื่องมือที่ใช้ 13

3.3.1 ฮาร์ดแวร์ (Hardware) 13

3.3.2 ซอฟต์แวร์ (Software) 13

3.3.3 ภาษาที่ใช้ (Language) 13

3.3.4 ขั้นตอนการออกแบบระบบ 13

3.4 ขั้นตอนการดำเนินงาน 13

3.4.1 การติดตั้งอุปกรณ์ 13

3.4.2 ภาพรวมการทำงานของระบบ 16

3.4.3 วิธีการทดสอบระบบ 18

บทที่ 4 ผลการดำเนินงาน 20

4.1 ผลการทดสอบที่ความเร็ว 0 กิโลเมตรต่อชั่วโมง (จอดรถหยุดนิ่ง) 20

4.2 ผลการทดสอบที่ความเร็ว 40 กิโลเมตรต่อชั่วโมง 22

4.3 ผลการทดสอบที่ความเร็ว 80 กิโลเมตรต่อชั่วโมง 25

บทที่ 5 สรุปผล อภิปรายผล และข้อเสนอแนะ 28

5.1 สรุปผลการดำเนินโครงงาน 28

5.2 อภิปรายผล 29

5.3 ปัญหาและอุปสรรค 29

5.4 ข้อเสนอแนะ 30

บรรณานุกรม 31

ภาคผนวก ก 34

ภาคผนวก ข 38

ภาคผนวก ค 42

สารบัญรูปภาพ

รูปที่ 2-1 NodeMCU 1.0 (V2)....4
รูปที่ 2-2 MQ7 Gas Sensor....5
รูปที่ 2-3 MQ7 Gas Sensor Pinout....5
รูปที่ 2-4 Internet of Things (IoT)....6
รูปที่ 2-5 Cloud Computing....7
รูปที่ 2-6 ความเข้นข้นของแก๊สที่เกิดขึ้นจากปัจจัยต่างๆ....8
รูปที่ 2-7 การแย่งจับฮีโมโกลบินของคาร์บอนมอนอกไซด์....8

รูปที่ 3-1 วงจรการเชื่อมต่อเซนเซอร์ MQ7.... 14
รูปที่ 3-2 วงจรการเชื่อมต่อจอ LCD-I2c (16x2).... 14
รูปที่ 3-3 อุปกรณ์ตรวจจับแก๊ส CO ชุดที่ 1.... 15
รูปที่ 3-4 อุปกรณ์ตรวจจับแก๊ส CO ชุดที่ 2.... 15
รูปที่ 3-5 อุปกรณ์แสดงผลค่าเฉลี่ยของปริมาณแก๊ส CO.... 16
รูปที่ 3-6 แนวคิดการทำงานของระบบ.... 17
รูปที่ 3-7 การแจ้งเตือนผ่าน Line Notify เมื่อเกิดแก๊สรั่วมากกว่า 50 ppm.... 18
รูปที่ 3-8 การแสดงผลข้อมูลค่าปริมาณแก๊ส CO ใน ThingSpeak.... 18
รูปที่ 3-9 กราฟแสดงผลข้อมูลค่าปริมาณแก๊ส CO 3 วันย้อนหลังใน MatLab.... 18
รูปที่ 3-10 ตัวอย่างส่วนหนึ่งของการทดสอบระบบ.... 19

รูปที่ 4-1 กราฟแสดงการเปรียบเทียบค่าที่อ่านได้จากเครื่องตรวจจับแก๊สมาตรฐานกับชุดอุปกรณ์ที่มีการติดตั้งจอ LCD-I2c ที่ความเร็ว 0 km/h (จอดรถหยุดนิ่ง) จำนวน 10 ครั้ง.... 21
รูปที่ 4-2 กราฟแสดงการเปรียบเทียบค่าที่อ่านได้จากเครื่องตรวจจับแก๊สมาตรฐานกับชุดอุปกรณ์ที่มีการติดตั้งจอ LCD-I2c ในขณะที่ดูดอากาศจากข้างนอกเข้ามาภายในรถ จำนวน 10 ครั้ง.... 21
รูปที่ 4-3 กราฟแสดงการเปรียบเทียบค่าที่อ่านได้จากเครื่องตรวจจับแก๊สมาตรฐานกับชุดอุปกรณ์ที่มีการติดตั้งจอ LCD-I2c ในขณะที่เปิดหน้าต่างรถลงครึ่งนึง จำนวน 10 ครั้ง.... 22
รูปที่ 4-4 กราฟแสดงการเปรียบเทียบค่าที่อ่านได้จากเครื่องตรวจจับแก๊สมาตรฐานกับชุดอุปกรณ์ที่มีการติดตั้งจอ LCD-I2c ที่ความเร็ว 40 km/h ในขณะที่เปิดระบบอากาศหมุนเวียนภายในรถ จำนวน 10 ครั้ง.... 23

รูปที่ 4-5 กราฟแสดงการเปรียบเทียบค่าที่อ่านได้จากเครื่องตรวจจับแก๊สมาตรฐานกับชุดอุปกรณ์ที่มีการติดตั้งจอ LCD-I2c ที่ความเร็ว 40 km/h ในขณะที่ดูดอากาศจากข้างนอกเข้ามาภายในรถ จำนวน 10 ครั้ง..... 24
รูปที่ 4-6 กราฟแสดงการเปรียบเทียบค่าที่อ่านได้จากเครื่องตรวจจับแก๊สมาตรฐานกับชุดอุปกรณ์ที่มีการติดตั้งจอ LCD-I2c ที่ความเร็ว 40 km/h ในขณะที่เปิดหน้าต่างรถลงครึ่งนึง จำนวน 10 ครั้ง..... 24
รูปที่ 4-7 กราฟแสดงการเปรียบเทียบค่าที่อ่านได้จากเครื่องตรวจจับแก๊สมาตรฐานกับชุดอุปกรณ์ที่มีการติดตั้งจอ LCD-I2c ที่ความเร็ว 80 km/h ในขณะที่เปิดระบบอากาศหมุนเวียนภายในรถ จำนวน 10 ครั้ง..... 26
รูปที่ 4-8 กราฟแสดงการเปรียบเทียบค่าที่อ่านได้จากเครื่องตรวจจับแก๊สมาตรฐานกับชุดอุปกรณ์ที่มีการติดตั้งจอ LCD-I2c ที่ความเร็ว 80 km/h ในขณะที่ดูดอากาศจากข้างนอกเข้ามาภายในรถ จำนวน 10 ครั้ง..... 26
รูปที่ 4-9 กราฟแสดงการเปรียบเทียบค่าที่อ่านได้จากเครื่องตรวจจับแก๊สมาตรฐานกับชุดอุปกรณ์ที่มีการติดตั้งจอ LCD-I2c ที่ความเร็ว 80 km/h ในขณะที่เปิดหน้าต่างรถลงครึ่งนึง จำนวน 10 ครั้ง..... 27

รูปที่ ก-1 เว็บไซต์ Arduino IDE 34
รูปที่ ก-2 หน้าเว็บไซต์สำหรับดาวน์โหลดโปรแกรม..... 34
รูปที่ ก-3 หน้าตาโปรแกรม Arduino IDE..... 35
รูปที่ ก-4 การตั้งค่าบอร์ดใน Arduino IDE [I]..... 35
รูปที่ ก-5 การตั้งค่าบอร์ดใน Arduino IDE [II]..... 36
รูปที่ ก-6 การตั้งค่าบอร์ดใน Arduino IDE [III]..... 36
รูปที่ ก-7 การตั้งค่าบอร์ดใน Arduino IDE [IV] 37

รูปที่ ข-1 การสมัครเพื่อใช้บริการ LINE Notify [I]..... 38
รูปที่ ข-2 การสมัครเพื่อใช้บริการ LINE Notify [II]..... 38
รูปที่ ข-3 การสมัครเพื่อใช้บริการ LINE Notify [III]..... 39
รูปที่ ข-4 การสมัครเพื่อใช้บริการ LINE Notify [IV]..... 39
รูปที่ ข-5 การสมัครเพื่อใช้บริการ LINE Notify [V] 40
รูปที่ ข-6 การสมัครเพื่อใช้บริการ LINE Notify [VI]..... 40
รูปที่ ข-7 การสมัครเพื่อใช้บริการ LINE Notify [VII]..... 41
รูปที่ ข-8 การสมัครเพื่อใช้บริการ LINE Notify [VII]..... 41

รูปที่ ค-1 หน้าเว็บไซต์ ThingSpeak..... 42
รูปที่ ค-2 หน้าสร้างบัญชีผู้ใช้ ThingSpeak [I]..... 43
รูปที่ ค-3 หน้าสร้างบัญชีผู้ใช้ ThingSpeak [II]..... 44
รูปที่ ค-4 หน้าสร้างบัญชีผู้ใช้ ThingSpeak [III]..... 44
รูปที่ ค-5 อีเมล์ยืนยันการสร้างบัญชีผู้ใช้ใหม่จาก ThingSpeak..... 45
รูปที่ ค-6 หน้ากรอกรหัสผ่าน..... 45
รูปที่ ค-7 หน้า My Channel..... 45
รูปที่ ค-8 หน้ากรอกรายละเอียด Channel [I]..... 46
รูปที่ ค-9 หน้ากรอกรายละเอียด Channel [II]..... 46
รูปที่ ค-10 หน้าแสดงผลข้อมูล..... 47
รูปที่ ค-11 หน้า API Keys..... 47
รูปที่ ค-12 หน้าแสดงผล เมื่อติดต่อกับ NodeMCU แล้ว..... 48
รูปที่ ค-13 การนำข้อมูลไปใช้กับ MatLab [I]..... 48
รูปที่ ค-14 การนำข้อมูลไปใช้กับ MatLab [II]..... 48
รูปที่ ค-15 การนำข้อมูลไปใช้กับ MatLab [III]..... 49
รูปที่ ค-16 การนำข้อมูลไปใช้กับ MatLab [IV]..... 49
รูปที่ ค-17 การนำข้อมูลไปใช้กับ MatLab [V]..... 50

สารบัญตาราง

ตารางที่ 3-1 แผนการดำเนินงานตลอดโครงการ 12

ตารางที่ 4-1 แสดงผลการทดสอบอุปกรณ์ที่ความเร็ว 0 km/h 20
ตารางที่ 4-2 แสดงผลการทดสอบอุปกรณ์ที่ความเร็ว 40 km/h 23
ตารางที่ 4-3 แสดงผลการทดสอบอุปกรณ์ที่ความเร็ว 80 km/h 25

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

เนื่องจากในปัจจุบันมีการเสียชีวิตจากการนอนหลับในรถยนต์ ซึ่งสาเหตุหลักที่ทำให้เสียชีวิตก็คือการสูดดมแก๊ส CO ซึ่งถ้าหากผู้โดยสารสูดดมเข้าไปจำนวนมากจะทำให้เกิดการอาการง่วงนอน โดยแก๊ส CO นั้นเป็นสารที่ไม่มีสี ไม่มีกลิ่น ทำให้ผู้โดยสารไม่ทราบได้ว่า กำลังสูดดมแก๊สนี้เข้าไป

ดังนั้นทางพวกเราจึงเล็งเห็นถึงความสำคัญนี้ จึงได้พัฒนาอุปกรณ์ตรวจจับแก๊ส CO รั่วภายในรถยนต์ ซึ่งอุปกรณ์ตัวนี้สามารถส่งข้อความแจ้งเตือนไปยังไลน์กลุ่มที่ผู้ใช้ต้องการผ่านทาง Line Notify ทำให้ผู้ใช้งานสามารถรับรู้และรับมือกับสถานการณ์ในขณะนั้นได้ พร้อมทั้งสามารถรายงานผลข้อมูลการรั่วไหลของแก๊ส CO ผ่านระบบ ThingSpeak และสังเกตแนวโน้มผ่าน MatLab ได้ เพื่อให้ผู้ใช้ได้ทราบว่ามีการรั่วไหล และสามารถป้องกันปัญหาที่เกิดขึ้นอย่างทันท่วงที

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 พัฒนาอุปกรณ์ที่สามารถตรวจจับแก๊ส CO ในรถยนต์ได้

1.2.2 พัฒนาระบบการแจ้งเตือนผู้ใช้ เมื่อมีการรั่วไหลของแก๊ส CO เกิน 50 ppm

1.2.3 ศึกษาการเก็บข้อมูลการรั่วไหลของแก๊ส CO ในระบบ Cloud เพื่อนำไปวิเคราะห์ และทำนายแนวโน้มของการรั่วไหลของแก๊ส CO หรือรายงานผลมาที่ผู้ใช้ได้ เพื่อให้ผู้ใช้สามารถป้องกันปัญหาที่เกิดขึ้นอย่างทันท่วงที

1.3 ขอบเขตของโครงงาน

1.3.1 ใช้บอร์ดไมโครคอนโทรลเลอร์ (Node MCU) ในการตรวจจับแก๊ส CO ในรถยนต์

1.3.2 สามารถแจ้งเตือนไปยังไลน์กลุ่มของผู้ใช้ เมื่อมีแก๊ส CO รั่วไหลเกินกำหนดผ่าน Line Notify

1.3.3 สามารถรายงานผลการรั่วไหลของแก๊ส CO ในรูปแบบกราฟผ่าน ThingSpeak ได้

1.3.4 สามารถนำข้อมูลมาสังเกตแนวโน้มการรั่วไหลของแก๊ส CO ผ่าน MatLab ได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

สามารถสร้างอุปกรณ์ตรวจจับแก๊ส CO ภายในห้องโดยสารรถยนต์ และแจ้งเตือนผ่าน Line Notify เพื่อป้องกันอุบัติเหตุที่เกิดขึ้นจากการรั่วไหลของแก๊ส CO โดยมีการรายงานผลข้อมูลการรั่วไหลของแก๊ส CO ผ่านระบบ ThingSpeak และสังเกตแนวโน้มผ่าน MatLab ได้ เพื่อนำข้อมูลที่ได้ไปใช้ประโยชน์ในด้านอื่นๆ ต่อไป

บทที่ 2
องค์ความรู้ที่เกี่ยวข้อง

2.1 ทฤษฎีและแนวคิดที่ใช้ในการทำวิจัย

2.1.1 NodeMCU Devkit 1.0 (V2)

NodeMCU คือ แพลตฟอร์มหนึ่งที่ใช้ช่วยในการสร้างโปรเจค Internet of Things (IoT) ที่ประกอบไปด้วย Development Kit (ตัวบอร์ด) และ Firmware (Software บนบอร์ด) ที่เป็น open source สามารถเขียนโปรแกรมด้วยภาษา Lau ได้ ทำให้ใช้งานได้ง่ายขึ้น มาพร้อมกับโมดูล WiFi (ESP8266) ซึ่งเป็นหัวใจสำคัญในการใช้เชื่อมต่อกับอินเตอร์เน็ต ตัวโมดูล ESP8266 นั้นมีอยู่ด้วยกันหลายรุ่น ตั้งแต่เวอร์ชั่นแรกที่เป็น ESP-01 จนปัจจุบันมีถึง ESP-12 โดยในเวอร์ชั่นที่ 2 จะใช้เป็น ESP-12E ซึ่งการใช้งานโดยรวมไม่แตกต่างกันมากนัก โดยที่ NodeMCU นั้นมีลักษณะคล้ายกับ Arduino ตรงที่มีพอร์ต Input Output ให้มาในตัว สามารถเขียนโปรแกรมคอนโทรลอุปกรณ์ I/O ได้โดยไม่ต้องผ่านอุปกรณ์อื่นๆ ซึ่งมีนักพัฒนาที่สามารถทำให้ Arduino IDE ใช้งานร่วมกับ NodeMCU ได้ จึงทำให้ใช้ภาษา C/C++ ในการเขียนโปรแกรมได้ ทำให้สามารถใช้งานได้หลากหลายมากยิ่งขึ้น และ NodeMCU นี้สามารถทำอะไรได้หลายอย่าง โดยเฉพาะเรื่องที่เกี่ยวข้องกับ IoT ไม่ว่าจะเป็นการทำ Web Server ขนาดเล็ก การควบคุมการเปิดปิดไฟผ่าน WiFi และอื่นๆอีกมากมาย

ชุดพัฒนาบอร์ด NodeMCU หรือเรียกสั้นๆว่า NodeMCU DevKit ปัจจุบันมีอยู่ 2 เวอร์ชั่นด้วยกัน คือ NodeMCU Devkit 0.9 (ESP-12) Version 1 และ NodeMCU Devkit 1.0 (ESP-12E) Version 2

คุณสมบัติ :

- ชุดพัฒนานี้ based on โมดูล WiFi ที่ชื่อว่า ESP8266
- มี GPIO PWM, I2C, 1-Wire และ ADC รวมมาอยู่บนบอร์ดเดียว
- มี USB-TTL มาในตัว ไม่ต้องซื้อแยกเหมือนกับการใช้ ESP8266 ปกติ ทำให้ใช้งานได้สะดวกขึ้น
- มีขา GPIO 10 ขา โดยทุกๆ ขาสามารถเป็น PWM, I2C และ 1-wire ได้
- มี PCB antenna สำหรับรับส่งสัญญาณไร้สาย

- ใช้คอนเนกเตอร์แบบ micro-usb สำหรับจ่ายแรงดันไฟเลี้ยงและสำหรับดาวน์โหลดเฟิร์มแวร์ (Sirima Inson. 2016)

รูปที่ 2-1 NodeMCU 1.0 (V2)

ที่มา : [ออนไลน์] https://www.elektor.com/nodemcu-microcontroller-board-with-esp8266-and-lua

2.1.2 MQ7 Gas Sensor

MQ7 Gas Sensor เป็นตัวเซนเซอร์ตรวจวัดความหนาแน่นของแก๊สคาร์บอนมอนอกไซด์ (CO) ในอากาศ โดยสามารถวัดได้ความละเอียดอยู่ในช่วง 0 ถึง 2000 ppm ซึ่งเซนเซอร์นี้ทำงานที่ 5V-DC มีพิน output ทั้งหมด 4 พิน ได้แก่ A0, D0, GND and VCC และสามารถวัดค่าความหนาแน่นได้ทั้งแบบ Digital และ Analog (Arduitronics. 2019)

โดย MQ7 Gas Sensor มีความไวต่อแก๊สคาร์บอนมอนอกไซด์ที่สูง ทำให้สามารถนำมาใช้ตรวจจับแก๊ส CO ได้ รวมถึงแก๊สที่มีแก๊ส CO เป็นส่วนประกอบอยู่ ซึ่งเซนเซอร์นี้มีราคาที่ถูกและสามารถนำไปใช้งานได้หลากหลาย (osoyoo. 2018)

คุณสมบัติ :

- ขนาด 33mm x 20mm x 16mm
- มีกำลังไฟที่ 3 ถึง 5V
- ใช้ชิป LM393 และ MQ7 Gas Sensor
- มีตัวต้านทานปรับค่าได้สาหรับปรับความไวในการตรวจจับ
- มีความเร็วในการรับค่าและส่งค่าน้อยกว่า 150 วินาที

- ค่าความต้านทานของการ Heating อยู่ที่ 31Ω ± 3Ω

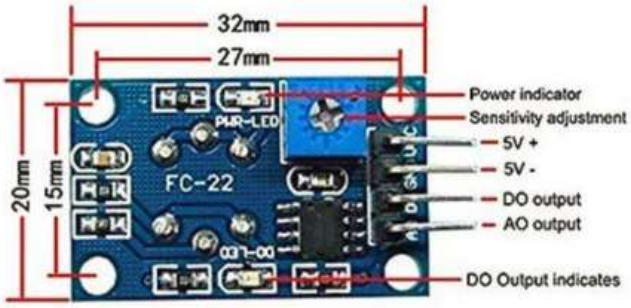
- แรงดันไฟฟ้าของการ Heating คือ 5.0V±0.2V/1.5V±0.1V

- พลังงานไฟฟ้าที่ใช้ในการ Heating อยู่ที่ประมาณ 350 mW

- สามารถวัดค่าได้ดีที่อุณหภูมิ -20 ℃ ~ +50℃ และที่ความชื้น ≤95%RH (myarduino. 2015)

รูปที่ 2-2 MQ7 Gas Sensor

ที่มา : [ออนไลน์] https://www.arduitronics.com/product/660/mq7-gas-sensor-module-carbon-monoxide-gas



รูปที่ 2-3 MQ7 Gas Sensor Pinout

ที่มา : [ออนไลน์] https://osoyoo.com/2018/11/15/arduino-lesson-mq-7-gas-sensor/

2.1.3 ทฤษฎี IoT

Internet of Things (IoT) คือ การที่อุปกรณ์อิเล็กทรอนิกส์ต่างๆ สามารถเชื่อมโยงหรือส่งข้อมูลถึงกันได้ด้วยอินเทอร์เน็ต โดยไม่ต้องป้อนข้อมูล การเชื่อมโยงนี้ง่ายจนทำให้เราสามารถสั่งการควบคุมการใช้งานอุปกรณ์อิเล็กทรอนิกส์ต่างๆ ผ่านทางเครือข่ายอินเตอร์เน็ตได้ ไปจนถึงการเชื่อมโยงการใช้งานอุปกรณ์อิเล็กทรอนิกส์ต่างๆ ผ่านทางเครือข่ายอินเตอร์เน็ตเข้ากับการใช้งานอื่นๆ จนเกิดเป็นบรรดา Smart ต่างๆ ได้แก่

Smart Device, Smart Grid, Smart Home, Smart Network, Smart Intelligent Transportation ทั้งหลายที่เราเคยได้ยินนั่นเอง ซึ่งแตกต่างจากในอดีตที่อุปกรณ์อิเล็กทรอนิกส์เป็นเพียงสื่อกลางในการส่งและแสดงข้อมูลเท่านั้น

กล่าวได้ว่า Internet of Things นี้ได้แก่การเชื่อมโยงของอุปกรณ์อัจฉริยะทั้งหลายผ่านอินเทอร์เน็ต เช่น แอปพลิเคชัน แว่นตากูเกิลกลาส รองเท้าวิ่งที่สามารถเชื่อมต่อข้อมูลการวิ่ง ทั้งความเร็ว ระยะทาง สถานที่ และสถิติได้ (Matana Wiboonyasake. 2018)

รูปที่ 2-4 Internet of Things (IoT)

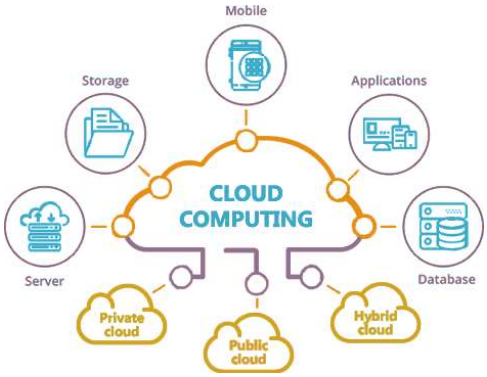
ที่มา : [ออนไลน์] https://mitechnews.com/internet-of-things/how-big-is-iot-20-6-billion-connected-device-by-2020

2.1.4 ทฤษฎี Cloud Computing

Cloud Computing เป็นระบบคอมพิวเตอร์ที่พร้อมรองรับการทำงานของผู้ใช้งานในทุกๆ ด้านไม่ว่าจะเป็นระบบเครือข่าย การจัดเก็บข้อมูล การทดสอบระบบหรือติดตั้งฐานข้อมูล หรือการใช้งานซอฟต์แวร์เฉพาะด้านในธุรกิจต่างๆ โดยที่ผู้ใช้งานไม่ต้องติดตั้งระบบฮาร์ดแวร์และซอฟต์แวร์ไว้ที่สำนักงานให้ยุ่งยาก แต่สามารถใช้งานในสิ่งที่ต้องการได้ด้วยการเชื่อมต่อกับระบบ Cloud Computing ผ่านอินเทอร์เน็ต

ในการเชื่อมต่อระบบหรือการใช้งานผ่าน Cloud เป็นการเชื่อมต่อกับศูนย์ข้อมูลคอมพิวเตอร์ หรือ Data Center ของผู้ให้บริการ ซึ่งศูนย์ข้อมูลคอมพิวเตอร์อยู่ภายใต้ระบบที่มีมาตรฐานและความปลอดภัยอย่างมาก

ไม่เพียงเท่านั้น การที่สามารถเชื่อมต่อและใช้งานระบบได้ผ่านอินเทอร์เน็ตทำให้การทำงานไม่ถูกจำกัดอยู่ในสถานที่หนึ่งสถานที่ใดอีกต่อไป แต่สามารถทำงานได้ทุกที่ทุกเวลาผ่านอุปกรณ์ต่างๆ ทั้งโน้ตบุ๊ก สมาร์ทโฟน หรือแท็บเล็ต (Csloxinfo. 2017)



รูปที่ 2-5 Cloud Computing

ที่มา : [ออนไลน์] https://arnondora.in.th/cloud-computing

2.1.5 ทฤษฎีแก๊สคาร์บอนมอนอกไซด์

เป็นแก๊สที่ไม่มีสี ไม่มีกลิ่น ไม่มีรส สูตรทางเคมีคือ CO แก๊สคาร์บอนมอนอกไซด์พบได้ในหลายแหล่ง โดยแก๊สเกิดจากการเผาไหม้ที่ไม่สมบูรณ์ของเชื้อเพลิงที่มีคาร์บอนเป็นส่วนประกอบ ซึ่งสามารถเกิดขึ้นได้จากกิจกรรมทั่วไปในชีวิตประจำวัน เช่น การเผาไหม้ของแก๊สหุงต้ม ไม้ ถ่านหิน ควันจากท่อไอเสียรถยนต์ หรือแม้กระทั่งการสูบบุหรี่ ซึ่งกิจกรรมเหล่านี้ก่อให้เกิดแก๊สคาร์บอนมอนอกไซด์ในสภาวะที่เหมาะสม เช่น อยู่ในพื้นที่ที่มีอากาศถ่ายเทได้สะดวก แต่ในทางตรงกันข้าม หากเกิดการเผาไหม้ในพื้นที่ที่ไม่มีการระบายอากาศที่ดี แก๊สคาร์บอนมอนอกไซด์สามารถก่อให้เกิดอันตรายต่อร่างกายได้ (พรรณพร กะตะจิตต์. 2018)

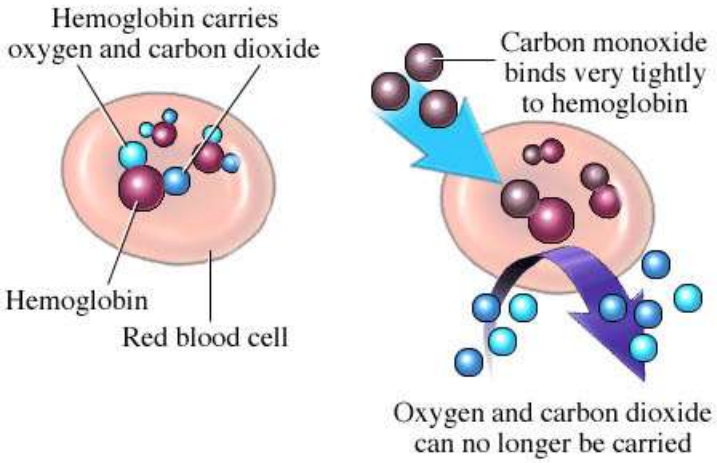
โดยเมื่อเรารับแก๊สคาร์บอนมอนอกไซด์เข้าไปในร่างกายมากกว่า 50 ppm จะทำให้เม็ดเลือดแดงไม่สามารถรวมตัวกับฮีโมโกลบิน เป็นผลทำให้ร่างกายเกิดอาการอ่อนเพลีย วิงเวียนศีรษะ เพราะว่าสมองได้รับออกซิเจน (Oxygen) เข้าไปน้อยกว่าสภาวะปกติ (Quarantiello; et al. 1998)

Table 1. Carbon Monoxide source concentration

Level of Carbon Monoxide	**Source**
0.1 ppm	Natural atmosphere level
0.5 to 5 ppm	Average level in homes
5 to 15 ppm	Near properly adjusted gas stove in homes
100 to 200 ppm	Exhaust from automobiles in the city
5000 ppm	Exhaust from a home wood fire
7000 ppm	Undiluted warm car exhaust without a catalytic converter

รูปที่ 2-6 ความเข้นข้นของแก๊สที่เกิดขึ้นจากปัจจัยต่างๆ

ที่มา : [ออนไลน์] https://iopscience.iop.org/article/10.1088/1757-899X/99/1/012009/pdf



รูปที่ 2-7 การแย่งจับฮีโมโกลบินของคาร์บอนมอนอกไซด์

ที่มา : [ออนไลน์] sites.google.com/site/igcsechemistry2017/home/year-11-topics/11-01-crude-oil/11-01-04-oil-the-environment

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 งานวิจัยเรื่อง “Vehicle Gas Leakage Detector”

โดย Azura che soh และคณะ

ใช้ Microcontroller (PIC 16 F 84 มาใช้งานร่วมกับเซ็นเซอร์ (NAP 11A) เพื่อตรวจจับแก๊ส CO ที่รั่วภายในห้องโดยสารรถยนต์ แล้วทำการส่งข้อมูลไปที่ Logic Detector Detector ซึ่งจะเป็นตัวที่ส่งข้อมูล output ไปที่ระบบแจ้งเตือน เพื่อส่งสัญญาณเตือนไปยังผู้โดยสารว่ามีแก๊ส CO อยู่ภายในรถยนต์ (Azura che soh; et al. 2010)

2.2.2 งานวิจัยเรื่อง "Is in cabin exposure to carbon monoxide and fine particulate matter amplified by the vehicle s self pollution potential? Quantifying the rate of exhaust intrusion"

โดย G. HarikM. และคณะ

ในงานวิจัยมีการใช้อุปกรณ์ DustTrak analyzers และ CO analyzers ในการตรวจวัดค่า CO และค่า PM 2.5 ภายในและภายนอกรถยนต์ เพื่อหาปัจจัยสาคัญที่ทาให้เกิดแก๊ส CO และ PM 2.5 (G. HarikM.; et al. 2017)

2.2.3 งานวิจัยเรื่อง "IoT Based Car Pollution Detection Using AWS"

โดย Aditya Bhatnagar และคณะ

นำ Raspberry Pi 3 มาใช้งานร่วมกับเซ็นเซอร์ MQ 7 เพื่อทดลองใช้อุปกรณ์ตรวจจับแก๊ส CO แล้วนำข้อมูลที่ตรวจจับได้ทั้งปริมาณแก๊ส CO กับตำแหน่งของอุปกรณ์ไปเก็บไว้ที่ Amazon DynamoDB พร้อมกับแจ้งเตือนผู้ใช้ผ่านฟีเจอร์ Push Notification ซึ่งเป็นฟีเจอร์ที่อยู่ในAWS IoT และมีการแสดงผลข้อมูลผ่าน Kibana (Aditya Bhatnagar; et al. 2018)

บทที่ 3

วิธีดำเนินโครงงาน

3.1 แผนการดำเนินงาน

3.1.1 เริ่มต้นและวางแผนโครงการ

3.1.1.1 วางแผนจัดทำโครงการ

3.1.1.2 หาข้อมูลศึกษาแนวคิดและทฤษฎี

3.1.1.3 เลือกเครื่องมือ

3.1.1.4 ศึกษาระบบงานที่เกี่ยวข้อง

3.1.2 การวิเคราะห์ระบบ

3.1.2.1 ทดลองดึงข้อมูล โดยเขียนโปรแกรมในการอ่านค่าจาก MQ7 sensor

3.1.2.2 เก็บรวบรวมข้อมูลจากค่าต่างๆ ที่ได้จาก sensor

3.1.3 การออกแบบระบบ

3.1.3.1 เชื่อมต่อและส่งข้อมูลการแจ้งเตือนไปยัง LINE Notify

3.1.3.2 กำหนดรายละเอียดของการแจ้งเตือน

3.1.3.3 เชื่อมต่อและส่งข้อมูลไปยัง ThingSpeak

3.1.4 การพัฒนาและติดตั้งระบบ

3.1.4.1 ทดสอบการแจ้งเตือนจาก LINE Notify สู่ LINE ของผู้ใช้งาน

3.1.4.2 พัฒนาการนำข้อมูลจาก ThingSpeak ไปแสดงเป็นกราฟเพื่อสังเกตแนวโน้มการรั่วไหลของแก๊ส CO ผ่าน MatLab

3.1.4.3 หาข้อผิดพลาดและแก้ไขปรับปรุง

3.1.4.4 จัดทำเอกสารการติดตั้งระบบและคู่มือการใช้งาน

3.1.5 สรุปและเผยแพร่งานวิจัย

3.2 แผนการดำเนินงานตลอดโครงงาน

แผนดำเนินงานโครงการ								
ขั้นตอนการทำงาน	ต.ค. 62	พ.ย. 62	ธ.ค. 62	ม.ค. 63	ก.พ. 63	มี.ค. 63	เม.ย. 63	พ.ค. 63
1. เริ่มต้นและวางแผนโครงการ 1.1 วางแผนจัดทำโครงการ 1.2 หาข้อมูลศึกษาแนวคิดและทฤษฎี 1.3 เลือกเครื่องมือ 1.4 ศึกษาระบบงานที่เกี่ยวข้อง	↔							
2. การวิเคราะห์ระบบ 2.1 ทดลองดึงข้อมูล โดยเขียนโปรแกรมในการอ่านค่าจาก MQ7 sensor 2.2 เก็บรวบรวมข้อมูลจากค่าต่างๆ ที่ได้จาก sensor		↔	↔					
3. การออกแบบระบบ 3.1 เชื่อมต่อและส่งข้อมูลการแจ้งเตือนไปยัง LINE Notify 3.2 กำหนดรายละเอียดของการแจ้งเตือน 3.3 เชื่อมต่อและส่งข้อมูลไปยัง ThingSpeak			↔	↔	↔			
4. การพัฒนาและติดตั้งระบบ 4.1 ทดสอบการแจ้งเตือนจาก LINE Notify สู่ LINE ของผู้ใช้งาน 4.2 พัฒนาการนำข้อมูลจาก ThingSpeak ไปทำนายแนวโน้มการรั่วของแก๊ส CO ผ่าน MatLab				↔	↔			
4.3 หาข้อผิดพลาดและแก้ไขปรับปรุง 4.4 จัดทำเอกสารการติดตั้งระบบและคู่มือการใช้งาน						↔	↔	↔
5. สรุปและเผยแพร่งานวิจัย							↔	↔

ตารางที่ 3-1 แผนการดำเนินงานตลอดโครงการ

3.3 อุปกรณ์และเครื่องมือที่ใช้

3.3.1 ฮาร์ดแวร์ (Hardware)

3.3.1.1 NodeMCU Devkit 1.0 (V2) จำนวน 3 ตัว

3.3.1.2 MQ7 Gas sensor จำนวน 2 ตัว

3.3.1.3 LCD-I2c (16x2) จำนวน 1 ตัว

3.3.2 ซอฟต์แวร์ (Software)

3.3.2.1 Arduino IDE

3.3.2.2 Line Notify

3.3.2.3 ThingSpeak

3.3.2.4 MatLab

3.3.3 ภาษาที่ใช้ (Language)

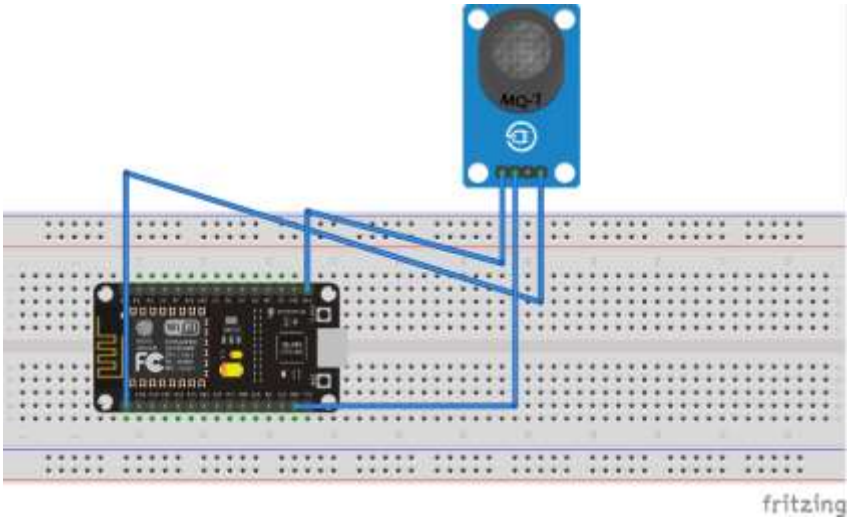
3.3.3.1 ภาษา C/C++

3.3.4 ขั้นตอนการออกแบบระบบ

3.4 ขั้นตอนการดำเนินงาน

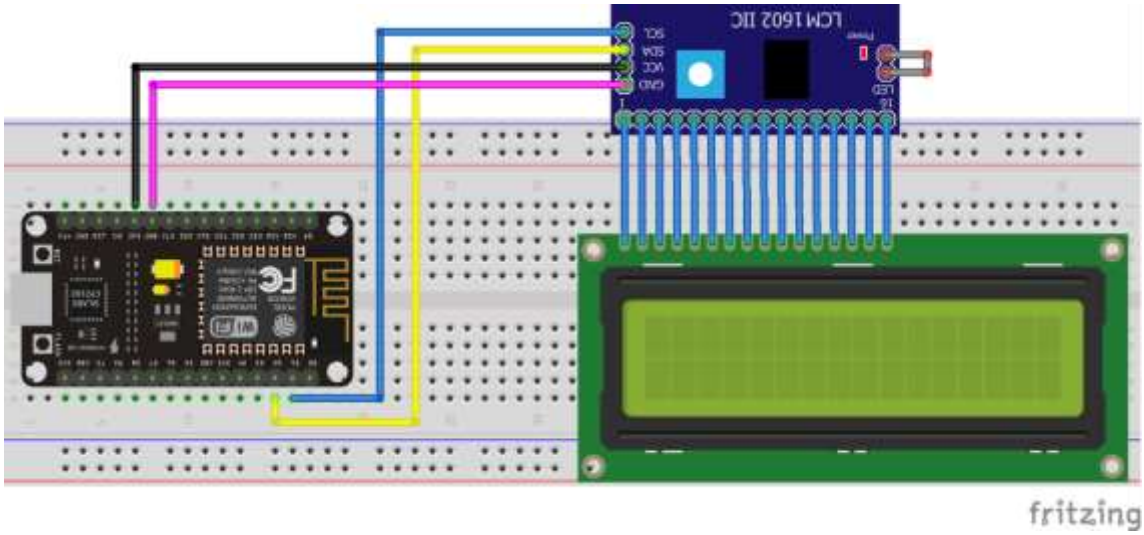
3.4.1 การติดตั้งอุปกรณ์

ทำการต่อเซนเซอร์ MQ7 โดยขา A0 ของเซนเซอร์ MQ7 ต่อกับขา A0 ของ NodeMCU, ขา GND ของเซนเซอร์ MQ7 ต่อกับขา GND ของ NodeMCU และขา VCC ของเซนเซอร์ MQ7 ต่อกับขา 3.3V ของ NodeMCU โดยทำแบบนี้ทั้ง 2 ชุด

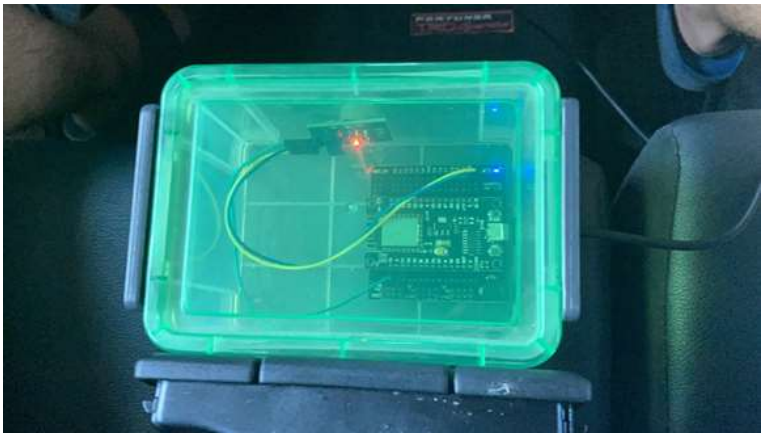


รูปที่ 3-1 วงจรการเชื่อมต่อเซนเซอร์ MQ7

ทำการต่อจอ LCD-I2c (16x2) โดยขา SCL ของ LCD-I2c เชื่อมกับขา D1 ของ NodeMCU, ขา SDA ของ LCD-I2c เชื่อมกับขา D2 ของ NodeMCU, ขา VCC ของ LCD-I2c เชื่อมกับขา 3.3V ของ NodeMCU และขา GND ของ LCD-I2c เชื่อมกับขา GND ของ NodeMCU



รูปที่ 3-2 วงจรการเชื่อมต่อจอ LCD-I2c (16x2)



รูปที่ 3-3 อุปกรณ์ตรวจจับแก๊ส CO ชุดที่ 1

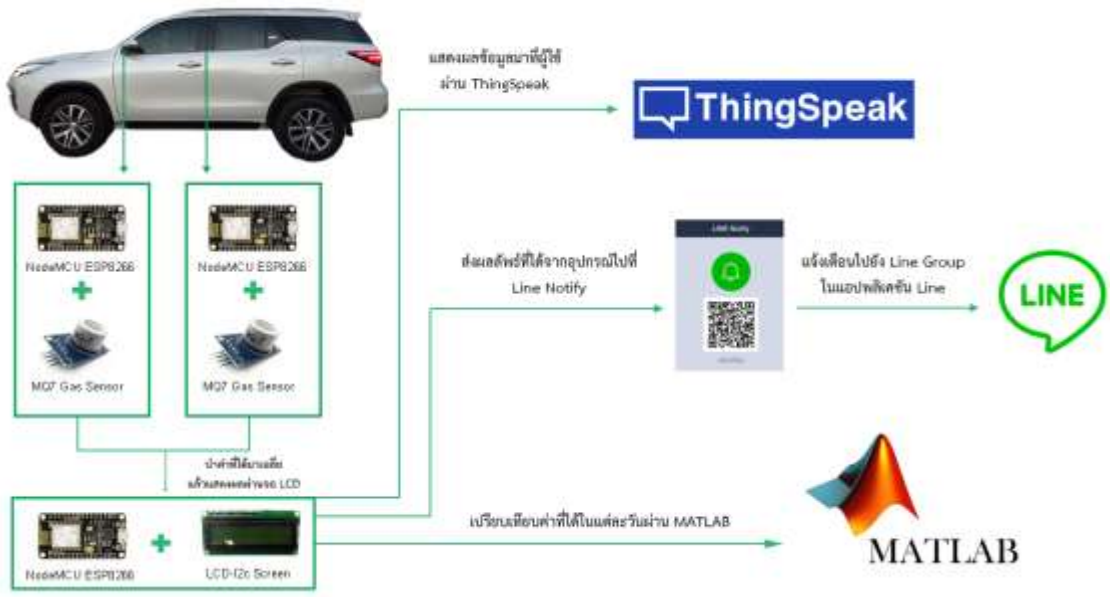
รูปที่ 3-4 อุปกรณ์ตรวจจับแก๊ส CO ชุดที่ 2

รูปที่ 3-5 อุปกรณ์แสดงผลค่าเฉลี่ยของปริมาณแก๊ส CO

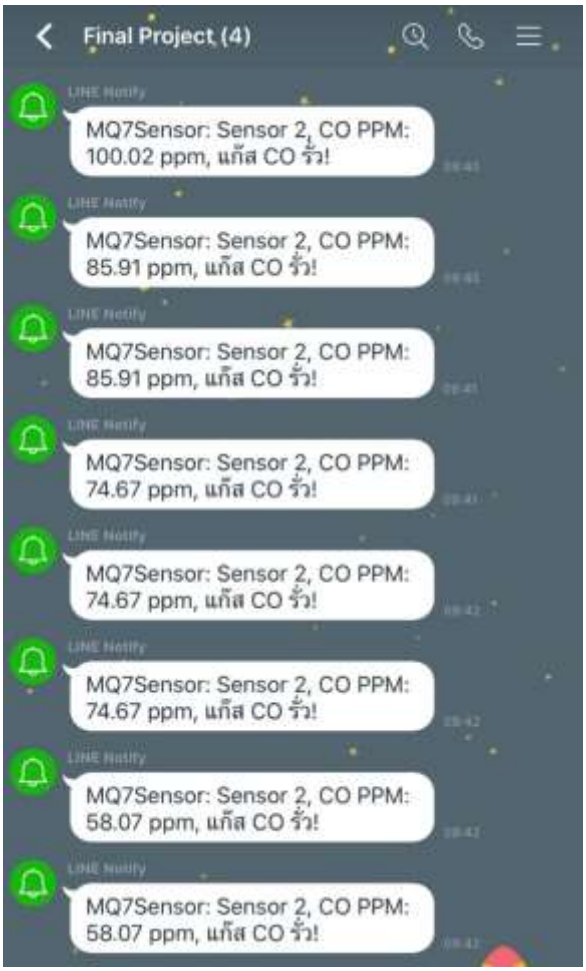
3.4.2 ภาพรวมการทำงานของระบบ

อุปกรณ์ชุดที่ 1 จะติดตั้งอยู่ที่ด้านหน้าของช่องปรับอากาศภายในห้องโดยสาร ส่วนอุปกรณ์ชุดที่ 2 จะติดตั้งที่ตรงกลางของห้องโดยสาร โดยแต่ละชุดมีเซ็นเซอร์ MQ7 ตรวจจับแก๊สคาร์บอนมอนออกไซด์ เมื่อเซ็นเซอร์สามารถตรวจจับแก๊สได้ จะทำการส่งข้อมูลจาก NodeMCU ไปยัง ThingSpeak ซึ่งเป็นระบบ Cloud ที่ออกแบบมาสำหรับการส่งข้อมูล และรายงานผลข้อมูลมาในลักษณะของตัวเลขและกราฟ ในขณะเดียวกัน อุปกรณ์ชุดที่ 3 ที่มีส่วนของหน้าจอ LCD-I2c จะทำการดึงข้อมูลของอุปกรณ์ชุดที่ 1 และ 2 มาหาค่าเฉลี่ย เพื่อให้ได้ค่าที่ครอบคลุมทั้งห้องโดยสาร พร้อมกับแสดงผลมาที่หน้าจอ LCD-I2c อีกทั้งส่งข้อมูลนั้นไปยัง ThingSpeak อีกครั้ง และนำข้อมูลไปทำนายแนวโน้มการรั่วไหลของแก๊ส CO ผ่าน MatLab เพื่อสังเกตว่า ในแต่ละวันมีการรั่วไหลของแก๊ส CO เพิ่มมากขึ้นหรือลดลงจากวันที่ผ่านมา

และในส่วนของการแจ้งเตือน ถ้าห้องโดยสารมีค่าปริมาณแก๊สเกิน 50 ppm จะมีการแจ้งเตือนมาที่หน้าจอ LCD-I2c พร้อมกับแจ้งเตือนไปที่ Line Notify เพื่อส่งข้อมูลการตรวจวัดแก๊สมาที่ไลน์กลุ่มที่ได้ตั้งค่าไว้ให้แจ้งเตือน เพื่อให้ทั้งผู้โดยสารและคนในไลน์กลุ่มสามารถรับรู้ได้ว่า รถยนต์คันที่ติดตั้งอุปกรณ์นี้ได้มีการรั่วไหลของแก๊สคาร์บอนมอนออกไซด์ และทำการรับมือกับสถานการณ์ดังกล่าวต่อไป

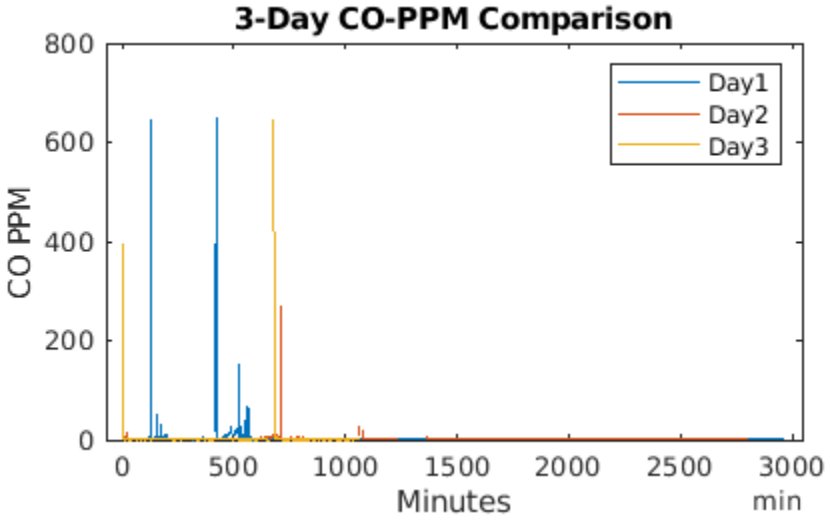


รูปที่ 3-6 แนวคิดการทำงานของระบบ



รูปที่ 3-7 การแจ้งเตือนผ่าน Line Notify เมื่อเกิดแก๊สรั่วมากกว่า 50 ppm

รูปที่ 3-8 การแสดงผลข้อมูลค่าปริมาณแก๊ส CO ใน ThingSpeak



รูปที่ 3-9 กราฟแสดงผลข้อมูลค่าปริมาณแก๊ส CO 3 วันย้อนหลังใน MatLab

3.4.3 วิธีการทดสอบระบบ

ทดสอบในห้องโดยสารรถยนต์ Toyota Fortuner โดยขับรถใช้เส้นทางแถวมอเตอร์เวย์ ถนนลำลูกกา และเมืองเอก ซึ่งมีการทดสอบทั้งหมด 3 รูปแบบ คือ ลดกระจกรถครึ่งนึง, เปิดระบบระบายอากาศแบบหมุนเวียน และเปิดระบบดึงอากาศภายนอกเข้ามาในรถ แต่ละรูปแบบทดลองที่ความเร็ว 0, 40 และ 80 กิโลเมตรต่อชั่วโมง รูปแบบละ 10 ครั้ง และมีการติดตั้งอุปกรณ์ 3 ชุดตามที่กล่าวไว้ใน 3.4.2 ซึ่งก่อนติดตั้งจะทำการเปิดอุปกรณ์ทิ้งไว้เป็นเวลา 5-10 นาที หรือจนกว่าจะได้ค่าที่เหมาะสม แล้วทำการทดสอบเปรียบเทียบกับอุปกรณ์มาตรฐาน โดยผู้

ทดสอบถืออุปกรณ์มาตรฐานไว้ในตำแหน่งใกล้เคียงกับอุปกรณ์ตรวจจับแก๊สของเรา แล้วทำการบันทึกผลการทดสอบ

รูปที่ 3-10 ตัวอย่างส่วนหนึ่งของการทดสอบระบบ

บทที่ 4

ผลการดำเนินงาน

จากการดำเนินโครงงาน คณะผู้จัดทำได้เสนอผลการดำเนินโครงงาน โดยแสดงผลลัพธ์ดังต่อไปนี้

4.1 ผลการทดสอบที่ความเร็ว 0 กิโลเมตรต่อชั่วโมง (จอดรถหยุดนิ่ง)

ตารางแสดงค่าที่อ่านได้จากชุดอุปกรณ์ที่มีการติดตั้งจอ LCD-I2c ซึ่งได้ค่ามาจากการหาค่าเฉลี่ยของอุปกรณ์ 2 ชุด ที่มีการติดตั้ง MQ7 ชุดละ 1 ตัว จำนวนทั้งหมด 10 ครั้ง ในขณะที่เปิดระบบอากาศหมุนเวียนภายในรถ ดูดอากาศจากข้างนอกเข้ามาภายในรถ และเปิดหน้าต่างรถลงครึ่งนึง ที่ความเร็ว 0 กิโลเมตรต่อชั่วโมงหรือจอดรถหยุดนิ่ง โดยนำมาเปรียบเทียบค่ากับเครื่องตรวจจับแก๊สมาตรฐาน

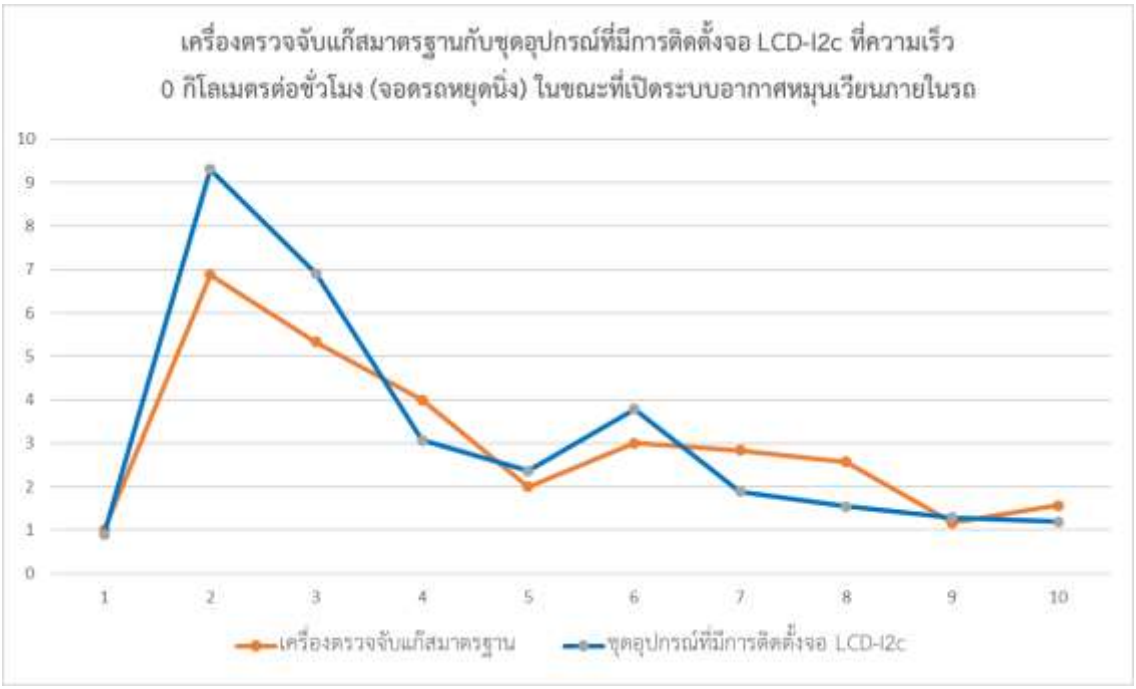
เปิดระบบอากาศหมุนเวียนภายในรถ			
ครั้งที่	เครื่องตรวจจับแก๊สมาตรฐาน (หน่วย ppm)	อุปกรณ์ที่มีการติดตั้งจอ LCD-I2c (หน่วย ppm)	ความคลาดเคลื่อนสัมพัทธ์ (%)
1	1	0.91	9
2	6.883	9.305	36.17
3	5.333	6.906	29.49
4	4	3.07	23.25
5	2	2.36	18
6	3	3.784	26.13
7	2.833	1.89	33.28
8	2.571	1.545	39.82
9	1.166	1.296	11.14
10	1.571	1.193	24.06
ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย (%)			25.03
ค่าความแม่นยำเฉลี่ย (%)			74.97

ดูดอากาศจากข้างนอกเข้ามาภายในรถ			
ครั้งที่	เครื่องตรวจจับแก๊สมาตรฐาน (หน่วย ppm)	อุปกรณ์ที่มีการติดตั้งจอ LCD-I2c (หน่วย ppm)	ความคลาดเคลื่อนสัมพัทธ์ (%)
1	1.8	1.11	38.33
2	2.333	0.506	78.31
3	1.248	2.142	71.63
4	2	0.508	74.6
5	0.428	0.572	33.64
6	0.8	0.745	6.87
7	0.571	0.671	17.51
8	0.5	0.77	54
9	0.571	0.677	18.56
10	0.857	0.778	9.21
ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย (%)			40.26
ค่าความแม่นยำเฉลี่ย (%)			59.74

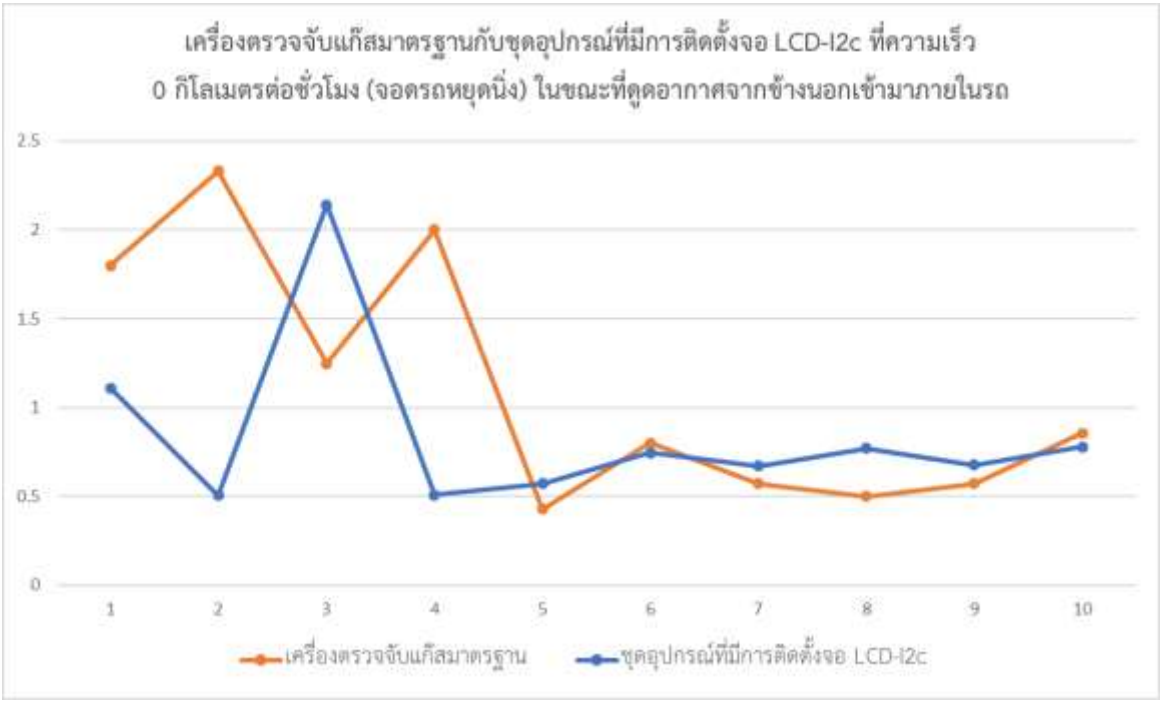
เปิดหน้าต่างรถลงครึ่งนึง			
ครั้งที่	เครื่องตรวจจับแก๊สมาตรฐาน (หน่วย ppm)	อุปกรณ์ที่มีการติดตั้งจอ LCD-I2c (หน่วย ppm)	ความคลาดเคลื่อนสัมพัทธ์ (%)
1	1	0.692	30.8
2	0.5	0.68	36
3	1	0.696	30.4
4	1	0.755	24.5
5	0.5	0.723	44.6
6	0.727	0.778	7.01
7	0.833	0.886	6.36
8	0.857	0.764	10.85
9	1	0.757	24.3
10	1	0.726	27.4
ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย (%)			24.22
ค่าความแม่นยำเฉลี่ย (%)			75.78

ตารางที่ 4-1 แสดงผลการทดสอบอุปกรณ์ที่ความเร็ว 0 km/h

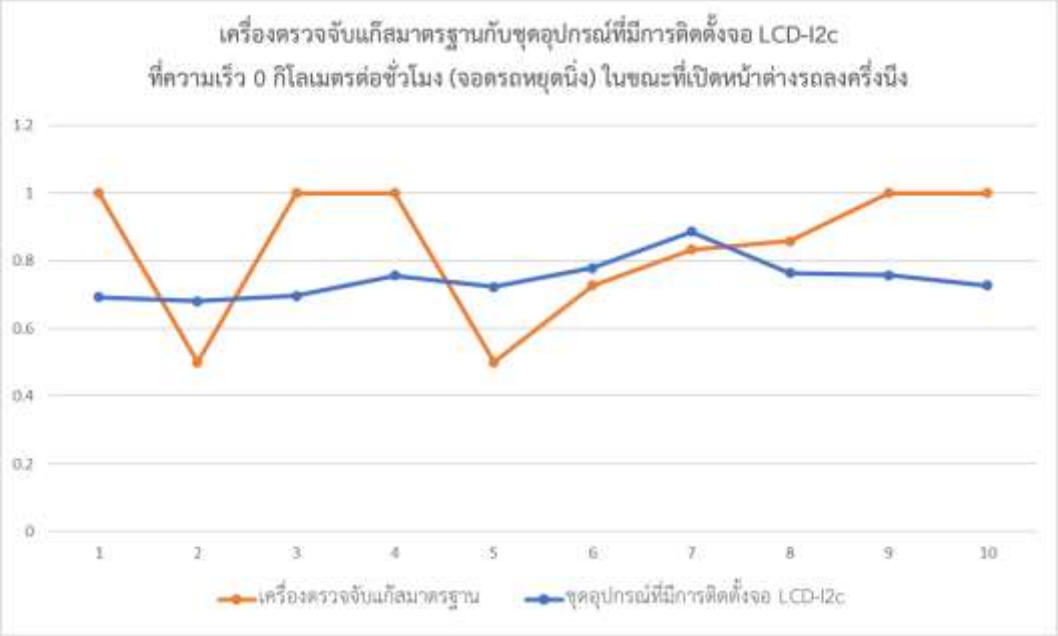
จากการทดสอบทั้ง 3 รูปแบบ ได้ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยร้อยละ 29.84 และค่าความแม่นยำเฉลี่ยร้อยละ 70.16



รูปที่ 4-1 กราฟแสดงการเปรียบเทียบค่าที่อ่านได้จากเครื่องตรวจจับแก๊สมาตรฐานกับชุดอุปกรณ์ที่มีการติดตั้งจอ LCD-I2c ที่ความเร็ว 0 km/h (จอดรถหยุดนิ่ง) จำนวน 10 ครั้ง



รูปที่ 4-2 กราฟแสดงการเปรียบเทียบค่าที่อ่านได้จากเครื่องตรวจจับแก๊สมาตรฐานกับชุดอุปกรณ์ที่มีการติดตั้งจอ LCD-I2c ในขณะที่ดูดอากาศจากข้างนอกเข้ามาภายในรถ จำนวน 10 ครั้ง



รูปที่ 4-3 กราฟแสดงการเปรียบเทียบค่าที่อ่านได้จากเครื่องตรวจจับแก๊สมาตรฐานกับชุดอุปกรณ์ที่มีการติดตั้งจอ LCD-I2c ในขณะที่เปิดหน้าต่างรถลงครึ่งนึง จำนวน 10 ครั้ง

4.2 ผลการทดสอบที่ความเร็ว 40 กิโลเมตรต่อชั่วโมง

ตารางแสดงค่าที่อ่านได้จากชุดอุปกรณ์ที่มีการติดตั้งจอ LCD-I2c ซึ่งได้ค่ามาจากการหาค่าเฉลี่ยของอุปกรณ์ 2 ชุด ที่มีการติดตั้ง MQ7 ชุดละ 1 ตัว จำนวนทั้งหมด 10 ครั้ง ในขณะที่เปิดระบบอากาศหมุนเวียนภายในรถ ดูดอากาศจากข้างนอกเข้ามาภายในรถ และเปิดหน้าต่างรถลงครึ่งนึง ที่ความเร็ว 40 กิโลเมตรต่อชั่วโมง โดยนำมาเปรียบเทียบค่ากับเครื่องตรวจจับแก๊สมาตรฐาน

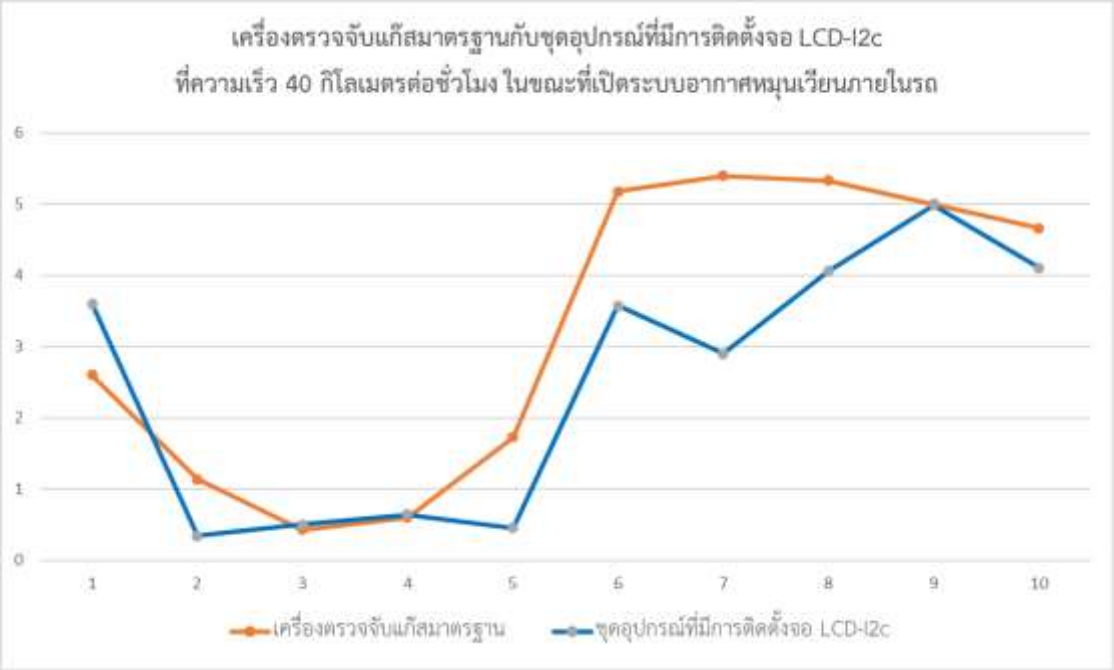
เปิดระบบอากาศหมุนเวียนภายในรถ			
ครั้งที่	เครื่องตรวจจับแก๊สมาตรฐาน (หน่วย ppm)	อุปกรณ์ที่มีการติดตั้งจอ LCD-I2c (หน่วย ppm)	ความคลาดเคลื่อนสัมพัทธ์ (%)
1	2.6	3.606	38.69
2	1.142	0.345	69.78
3	0.428	0.501	17.05
4	0.6	0.642	7
5	1.727	0.454	73.71
6	5.181	3.577	30.95
7	5.4	2.904	46.22
8	5.333	4.06	23.87
9	5	4.985	0.3
10	4.666	4.105	12.02
ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย (%)			31.95
ค่าความแม่นยำเฉลี่ย (%)			68.05

ดูดอากาศจากข้างนอกเข้ามาภายในรถ			
ครั้งที่	เครื่องตรวจจับแก๊สมาตรฐาน (หน่วย ppm)	อุปกรณ์ที่มีการติดตั้งจอ LCD-I2c (หน่วย ppm)	ความคลาดเคลื่อนสัมพัทธ์ (%)
1	3	1.21	59.66
2	2.5	1.93	22.80
3	2.5	1.873	25.08
4	2.75	1.677	39.02
5	3.125	1.942	37.85
6	2.428	0.56	76.93
7	2.428	1.715	29.36
8	1	0.654	34.6
9	0.625	0.522	16.48
10	0.714	0.592	17.08
ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย (%)			35.88
ค่าความแม่นยำเฉลี่ย (%)			64.12

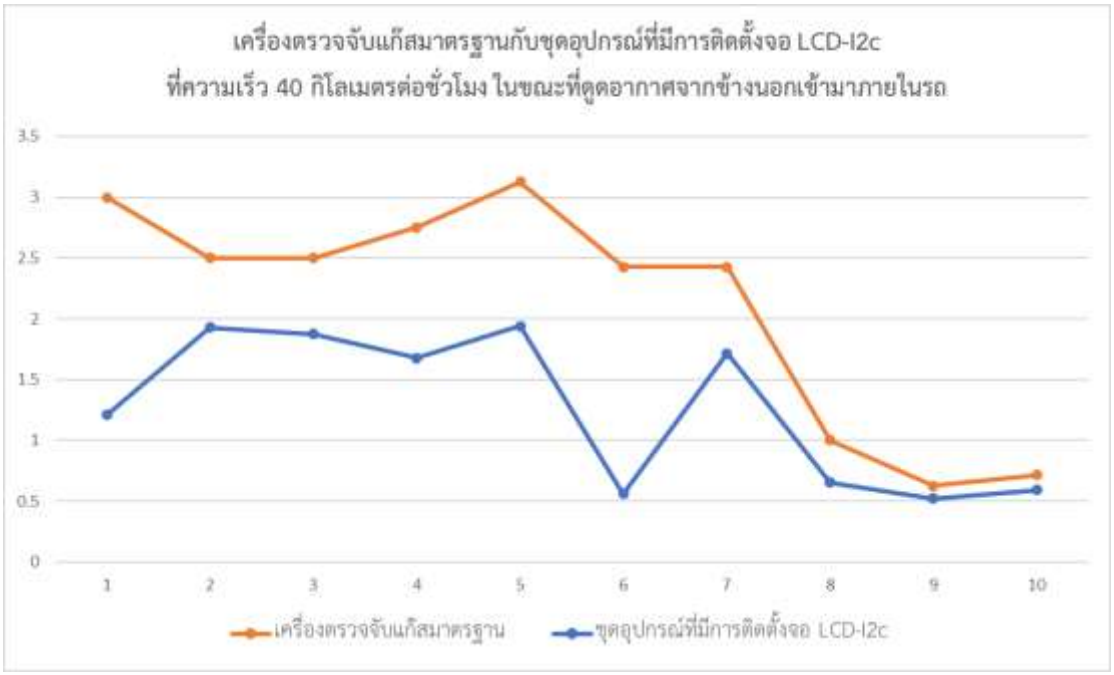
เปิดหน้าต่างรถลงครึ่งนึง			
ครั้งที่	เครื่องตรวจจับแก๊สมาตรฐาน (หน่วย ppm)	อุปกรณ์ที่มีการติดตั้งจอ LCD-I2c (หน่วย ppm)	ความคลาดเคลื่อนสัมพัทธ์ (%)
1	0.5	0.337	32.6
2	5.25	2.475	52.85
3	1.5	0.41	72.67
4	0.625	0.32	48.8
5	4.4	1.53	65.22
6	3.88	1.97	49.22
7	1.75	1.556	11.08
8	1.5	1.267	15.53
9	10.6	8.146	23.15
10	12.166	10.978	9.76
ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย (%)			38.08
ค่าความแม่นยำเฉลี่ย (%)			61.92

ตารางที่ 4-2 แสดงผลการทดสอบอุปกรณ์ที่ความเร็ว 40 km/h

จากการทดสอบทั้ง 3 รูปแบบ ได้ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยร้อยละ 35.30 และค่าความแม่นยำเฉลี่ยร้อยละ 64.70



รูปที่ 4-4 กราฟแสดงการเปรียบเทียบค่าที่อ่านได้จากเครื่องตรวจจับแก๊สมาตรฐานกับชุดอุปกรณ์ที่มีการติดตั้งจอ LCD-I2c ที่ความเร็ว 40 km/h ในขณะที่เปิดระบบอากาศหมุนเวียนภายในรถ จำนวน 10 ครั้ง



รูปที่ 4-5 กราฟแสดงการเปรียบเทียบค่าที่อ่านได้จากเครื่องตรวจจับแก๊สมาตรฐานกับชุดอุปกรณ์ที่มีการติดตั้งจอ LCD-I2c ที่ความเร็ว 40 km/h ในขณะที่ดูดอากาศจากข้างนอกเข้ามาภายในรถ จำนวน 10 ครั้ง

รูปที่ 4-6 กราฟแสดงการเปรียบเทียบค่าที่อ่านได้จากเครื่องตรวจจับแก๊สมาตรฐานกับชุดอุปกรณ์ที่มีการติดตั้งจอ LCD-I2c ที่ความเร็ว 40 km/h ในขณะที่เปิดหน้าต่างรถลงครึ่งนึง จำนวน 10 ครั้ง

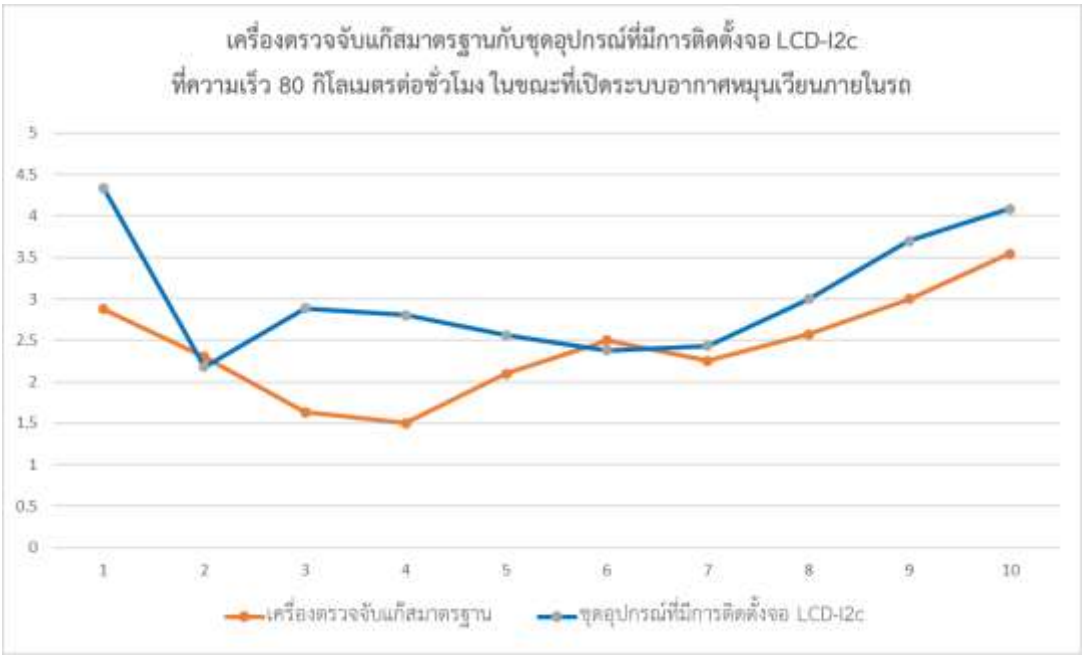
4.3 ผลการทดสอบที่ความเร็ว 80 กิโลเมตรต่อชั่วโมง

ตารางแสดงค่าที่อ่านได้จากชุดอุปกรณ์ที่มีการติดตั้งจอ LCD-I2c ซึ่งได้ค่ามาจากการหาค่าเฉลี่ยของอุปกรณ์ 2 ชุด ที่มีการติดตั้ง MQ7 ชุดละ 1 ตัว จำนวนทั้งหมด 10 ครั้ง ในขณะที่เปิดระบบอากาศหมุนเวียนภายในรถ ดูดอากาศจากข้างนอกเข้ามาภายในรถ และเปิดหน้าต่างรถลงครึ่งนึง ที่ความเร็ว 80 กิโลเมตรต่อชั่วโมง โดยนำมาเปรียบเทียบค่ากับเครื่องตรวจจับแก๊สมาตรฐาน

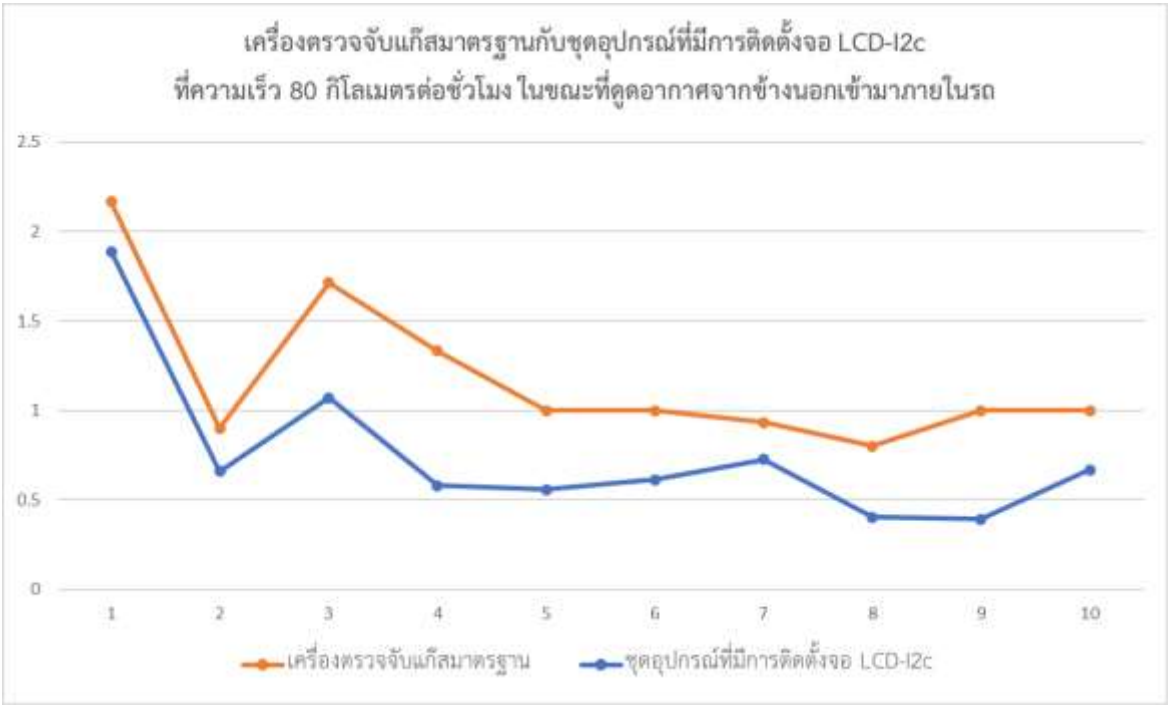
เปิดระบบอากาศหมุนเวียนภายในรถ				ดูดอากาศจากข้างนอกเข้ามาภายในรถ				เปิดหน้าต่างรถลงครึ่งนึง			
ครั้งที่	เครื่องตรวจจับแก๊สมาตรฐาน (หน่วย ppm)	อุปกรณ์ที่มีการติดตั้งจอ LCD-I2c (หน่วย ppm)	ความคลาดเคลื่อนสัมพัทธ์ (%)	ครั้งที่	เครื่องตรวจจับแก๊สมาตรฐาน (หน่วย ppm)	อุปกรณ์ที่มีการติดตั้งจอ LCD-I2c (หน่วย ppm)	ความคลาดเคลื่อนสัมพัทธ์ (%)	ครั้งที่	เครื่องตรวจจับแก๊สมาตรฐาน (หน่วย ppm)	อุปกรณ์ที่มีการติดตั้งจอ LCD-I2c (หน่วย ppm)	ความคลาดเคลื่อนสัมพัทธ์ (%)
1	2.875	4.335	50.78	1	2.168	1.888	12.91	1	0.562	0.311	44.66
2	2.3	2.182	5.13	2	0.9	0.66	26.66	2	0.833	0.415	50.18
3	1.636	2.888	76.52	3	1.714	1.07	37.57	3	1.04	0.73	29.08
4	1.5	2.802	86.8	4	1.333	0.58	56.49	4	2.666	0.93	65.12
5	2.1	2.56	21.9	5	1	0.557	44.3	5	3.333	1.403	57.91
6	2.5	2.377	4.92	6	1	0.613	38.7	6	5.2	3.816	26.62
7	2.25	2.437	8.31	7	0.933	0.726	22.18	7	6.67	3.27	50.97
8	2.571	3	16.68	8	0.8	0.404	49.5	8	9.75	3.24	66.77
9	3	3.696	23.2	9	1	0.392	60.8	9	7.4	4.236	42.76
10	3.545	4.083	15.17	10	1	0.667	33.3	10	5.833	3.598	38.32
ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย (%)			30.94	ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย (%)			38.24	ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ย (%)			47.24
ค่าความแม่นยำเฉลี่ย (%)			69.06	ค่าความแม่นยำเฉลี่ย (%)			61.76	ค่าความแม่นยำเฉลี่ย (%)			52.76

ตารางที่ 4-3 แสดงผลการทดสอบอุปกรณ์ที่ความเร็ว 80 km/h

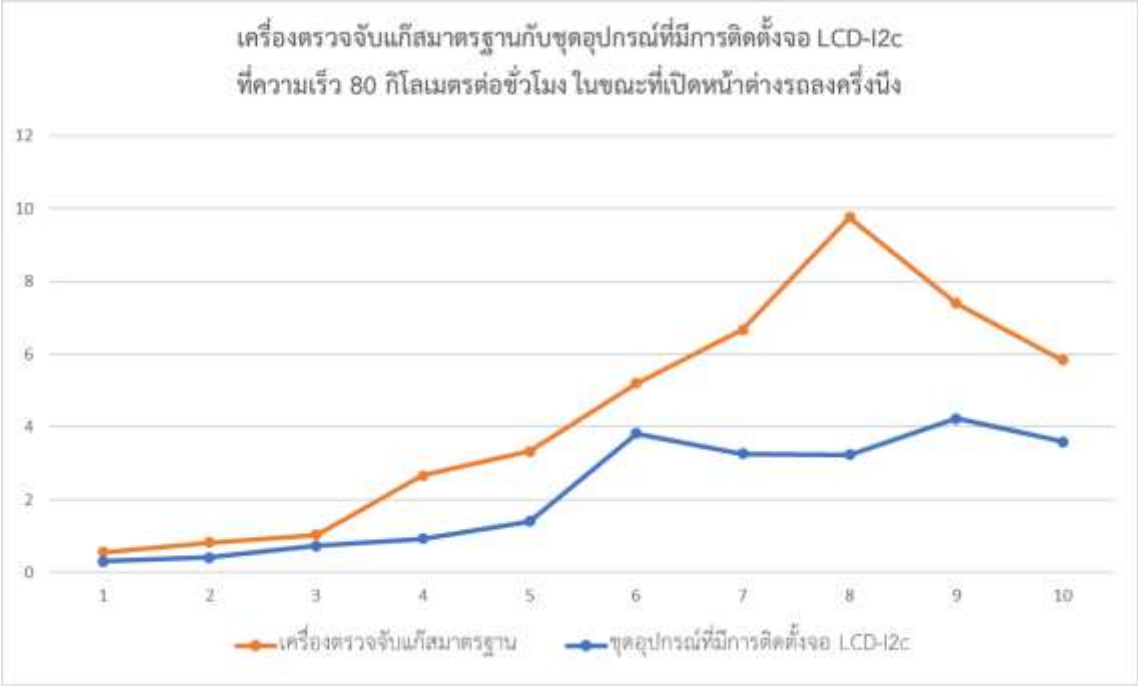
จากการทดสอบทั้ง 3 รูปแบบ ได้ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยร้อยละ 38.81 และค่าความแม่นยำเฉลี่ยร้อยละ 61.19



รูปที่ 4-7 กราฟแสดงการเปรียบเทียบค่าที่อ่านได้จากเครื่องตรวจจับแก๊สมาตรฐานกับชุดอุปกรณ์ที่มีการติดตั้งจอ LCD-I2c ที่ความเร็ว 80 km/h ในขณะที่เปิดระบบอากาศหมุนเวียนภายในรถ จำนวน 10 ครั้ง



รูปที่ 4-8 กราฟแสดงการเปรียบเทียบค่าที่อ่านได้จากเครื่องตรวจจับแก๊สมาตรฐานกับชุดอุปกรณ์ที่มีการติดตั้งจอ LCD-I2c ที่ความเร็ว 80 km/h ในขณะที่ดูดอากาศจากข้างนอกเข้ามาภายในรถ จำนวน 10 ครั้ง



รูปที่ 4-9 กราฟแสดงการเปรียบเทียบค่าที่อ่านได้จากเครื่องตรวจจับแก๊สมาตรฐานกับชุดอุปกรณ์ที่มีการติดตั้งจอ LCD-I2c ที่ความเร็ว 80 km/h ในขณะที่เปิดหน้าต่างรถลงครึ่งนึง จำนวน 10 ครั้ง

บทที่ 5

สรุปผล อภิปรายผล และข้อเสนอแนะ

5.1 สรุปผลการดำเนินโครงงาน

จากการศึกษาค้นคว้า และดำเนินโครงงาน ระบบตรวจจับแก๊ส CO รั่วภายในรถยนต์พร้อมแจ้งเตือน (In-car CO Gas Detection and Alert System) โดยอาศัยอุปกรณ์หลัก 3 อย่าง ได้แก่ ไมโครคอนโทรลเลอร์ NodeMCU, เซนเซอร์ MQ7 สำหรับตรวจจับแก๊ส CO และจอ LCD-I2c สำหรับแสดงค่าแก๊สที่ตรวจจับได้ โดยมีการทำอุปกรณ์ตรวจจับแก๊ส CO แยกเป็น 2 ชุด ได้แก่ ชุดที่หนึ่งวางไว้บริเวณคอนโซลแอร์หน้ารถ ส่วนชุดที่สองวางไว้บริเวณกลางห้องโดยสารรถ เพื่อให้ได้ค่าแก๊ส CO ครอบคลุมทั้งห้องโดยสาร โดยนำค่านั้นมาแสดงที่อุปกรณ์อีกชุด ซึ่งมีการติดตั้งจอ LCD-I2c พร้อมกับส่งข้อมูลไปแสดงที่ ThingSpeak อีกทั้งมีการใช้ MatLab ในการแสดงกราฟข้อมูล 3 วันย้อนหลัง นอกจากนั้นจะมีการส่งข้อความแจ้งเตือนไปยัง Line ของผู้ใช้ผ่านทาง Line Notify ถ้ามีค่าแก๊สเกิน 50 ppm

วิธีการทดสอบอุปกรณ์คือ ทดสอบในห้องโดยสารรถยนต์ Toyota Fortuner โดยขับรถใช้เส้นทางแถวมอเตอร์เวย์ ถนนลำลูกกา และเมืองเอก ซึ่งมีการทดสอบทั้งหมด 3 รูปแบบ คือ ลดกระจกรถครึ่งนึง, เปิดระบบระบายอากาศแบบหมุนเวียน และเปิดระบบดึงอากาศภายนอกเข้ามาในรถ แต่ละรูปแบบทดลองที่ความเร็ว 0, 40 และ 80 กิโลเมตรต่อชั่วโมง รูปแบบละ 10 ครั้ง พร้อมกับทดสอบเปรียบเทียบกับอุปกรณ์มาตรฐาน โดยวางในตำแหน่งที่ใกล้เคียงกัน

จากการทดสอบอุปกรณ์พบว่า อุปกรณ์สามารถตรวจจับปริมาณแก๊ส CO ที่มีอยู่ในห้องโดยสารได้ มีการแสดงผลข้อมูลผ่านจอ LCD-I2c ได้ รวมถึงมีการแจ้งเตือนไปยังผู้ใช้ผ่านทาง Line Notify ได้อย่างถูกต้อง นอกจากนี้ยังพบว่า ค่าความคลาดเคลื่อนเฉลี่ยสัมพัทธ์ และค่าความแม่นยำเฉลี่ยที่ทำการทดสอบในขณะที่รถมีความเร็ว 0 กิโลเมตรต่อชั่วโมง หรือจอดรถอยู่นิ่ง มีค่าค่อนข้างดีกว่าการทดสอบในขณะที่รถมีความเร็ว 40 และ 80 กิโลเมตรต่อชั่วโมง เนื่องจากมีปัจจัยที่ทำให้ความสามารถในการตรวจจับของอุปกรณ์คลาดเคลื่อนได้ เช่น ความเร็วของรถและคุณภาพของเซนเซอร์ เป็นต้น นอกจากนั้นยังมีข้อจำกัดในเรื่องของค่าปริมาณแก๊ส ที่จะแสดงค่าใหม่ทุกๆ 15 วินาที ซึ่งเป็นผลมาจาก ThingSpeak ที่ให้มีการอัพเดตค่าใหม่ได้ต่ำสุดที่ทุกๆ 15 วินาที ทำให้ผู้ใช้ได้ค่าปริมาณแก๊สที่ไม่รวดเร็วมากนัก

5.2 อภิปรายผล

จากการดำเนินโครงงาน อุปกรณ์สามารถตรวจจับปริมาณแก๊ส CO ที่มีอยู่ในห้องโดยสารได้ มีการแสดงผลข้อมูลผ่านจอ LCD-I2c ได้ โดยจากการทดลองในห้องโดยสารรถยนต์ Toyota Fortuner โดยขับรถใช้เส้นทางแถวมอเตอร์เวย์ ถนนลำลูกกา และเมืองเอก ซึ่งมีการทดสอบทั้งหมด 3 รูปแบบ คือ ลดกระจกรถครึ่งนึง, เปิดระบบระบายอากาศแบบหมุนเวียน และเปิดระบบดึงอากาศภายนอกเข้ามาในรถ แต่ละรูปแบบทดลองที่ความเร็ว 0, 40 และ 80 กิโลเมตรต่อชั่วโมง รูปแบบละ 10 ครั้ง มาทดสอบเปรียบเทียบกับค่าที่ได้จากอุปกรณ์มาตรฐาน

- ทดสอบที่ความเร็ว 0 กิโลเมตรต่อชั่วโมง (จอดรถหยุดนิ่ง)

  จากการทดสอบทั้งหมด 3 รูปแบบ ได้ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยร้อยละ 29.84 และค่าความแม่นยำเฉลี่ยร้อยละ 70.16

- ทดสอบที่ความเร็ว 40 กิโลเมตรต่อชั่วโมง

  จากการทดสอบทั้งหมด 3 รูปแบบ ได้ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยร้อยละ 35.30 และค่าความแม่นยำเฉลี่ยร้อยละ 64.70

- ทดสอบที่ความเร็ว 80 กิโลเมตรต่อชั่วโมง

  จากการทดสอบทั้งหมด 3 รูปแบบ ได้ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยร้อยละ 38.81 และค่าความแม่นยำเฉลี่ยร้อยละ 61.19

  สรุปได้ว่าอุปกรณ์ตรวจจับปริมาณแก๊ส CO นี้ สามารถใช้ตรวจจับปริมาณของแก๊สได้แม่นยำในระดับหนึ่ง โดยที่อุปกรณ์จะตรวจจับแก๊สได้ดีและมีความแม่นยำมากขึ้นในขณะที่รถจอดหยุดนิ่ง หรือมีความเร็วของรถที่ไม่มากนัก

5.3 ปัญหาและอุปสรรค

- การเปิดใช้งานเซนเซอร์ MQ7 ครั้งแรก ต้องมีการเปิดให้ทำงานไปสักระยะหนึ่งก่อน ถึงจะได้ค่าที่เหมาะสมในการนำมาใช้งาน จึงอาจทำให้เสียเวลา
- ThingSpeak มีข้อจำกัดในการส่งข้อมูล คือมีดีเลย์ในการส่งข้อมูลต่ำสุดที่ 15 วินาที ต่อการส่งหนึ่งครั้ง จึงทำให้ข้อมูลที่อ่านได้อาจไม่ใช่ข้อมูลที่ถูกส่งมาจากเซนเซอร์ในทันที

- ความเร็วของรถที่ใช้ในการทดสอบและตำแหน่งอากาศที่ผ่านเซนเซอร์อาจมีผลทำให้ไม่สามารถตรวจจับแก๊สได้อย่างแม่นยำ

5.4 ข้อเสนอแนะ

- ในอนาคตสามารถพัฒนาให้มีการแจ้งเตือนไปยังผู้เชี่ยวชาญที่เกี่ยวข้องได้ เพื่อที่จะสามารถป้องกันอันตรายจากการเกิดแก๊ส CO รั่วได้อย่างทันท่วงที
- อุปกรณ์ที่ใช้ในการทดสอบนี้อาจเหมาะสำหรับการตรวจจับแก๊สในบริเวณที่ไม่มีความเร็วมากระทบกับตัวอุปกรณ์ ซึ่งความเร็วอาจมีผลทำให้อุปกรณ์มีความคลาดเคลื่อนในการตรวจจับ
- อุปกรณ์ยังมีค่าความคลาดเคลื่อนเฉลี่ยที่มากกว่า 10% ซึ่งอยู่ในเกณฑ์ต้องปรับปรุงให้มีประสิทธิภาพที่มากกว่านี้

บรรณานุกรม

[1] Sirima Inson. (2016). **มารู้จัก NodeMCU**. สืบค้นเมื่อ 2 ตุลาคม 2562, จาก https://embedded523372.wordpress.com/2016/04/15/nodemcu/.

[2] Arduitronics. (2019). **MQ7 Gas Sensor Module (Carbon Monoxide Gas)**. สืบค้นเมื่อ 2 ตุลาคม 2562, จาก https://www.arduitronics.com/product/660/mq7-gas-sensor-module-carbon-monoxide-gas.

[3] Osoyoo. (2018). **Arduino lesson – MQ-7 Gas Sensor**. สืบค้นเมื่อ 2 ตุลาคม 2562, จาก https://osoyoo.com/2018/11/15/arduino-lesson-mq-7-gas-sensor/.

[4] Myarduino. (2015). **MQ-7 CO Carbon Monoxide Gas Sensor Module เซ็นเซอร์ตรวจจับก๊าซคาร์บอนมอนอกไซด์**. สืบค้นเมื่อ 2 ตุลาคม 2562, จาก https://www.myarduino.net/product/541/mq-7-co-carbon-monoxide-gas-sensor-module-เซ็นเซอร์ตรวจจับก๊าซคาร์บอนมอนอกไซด์.

[5] Matana Wiboonyasake. (2018). **ทำความรู้จักกับ Internet of Things**. สืบค้นเมื่อ 3 ตุลาคม 2562, จาก https://www.aware.co.th/iot-คืออะไร/.

[6] Csloxinfo. (2017). **รู้จักเทคโนโลยี Cloud Computing**. สืบค้นเมื่อ 5 ตุลาคม 2562, จาก http://dccloud.csloxinfo.com/th/wecloud01/.

[7] พรรณพร กะตะจิตต์. (2018). **ความเป็นพิษของคาร์บอนมอนอกไซด์ (Carbon monoxide)**. สืบค้นเมื่อ 5 ตุลาคม 2562, จาก https://www.scimath.org/article-chemistry/item/7856-2018-02-22-02-37-59.

[8] Quarantiello, Flavio, Gallicola, Federica, Timpone, Laura. (1998). **Carbon monoxide poisoning**. สืบค้นเมื่อ 5 ตุลาคม 2562, จาก https://doi.org/10.1056/NEJM199811263392206.

[9] A. Che Soh, M.K. Hassan, A.J. Ishak. (2010). **Vehichle Gas Leagkage Detector**. สืบค้นเมื่อ 25 สิงหาคม 2562, จาก https://www.researchgate.net/publication/235335659_Vehicle_Gas_Leakage_Detector.

[10] Harik, G, El-Fadel, M, Shihadeh, A, Alameddine, I, Hatzopoulou, M. (2017). **Is in-cabin exposure to carbon monoxide and fine particulate matter amplified by the vehicle's self-pollution potential? Quantifying the rate of exhaust intrusion**. สืบค้นเมื่อ 25 สิงหาคม 2562, จาก http://www.sciencedirect.com/science/article/pii/S1361920917300238.

[11] Bhatnagar, A, Sharma, V, Raj, G. (2018). **IoT based Car Pollution Detection Using AWS**. สืบค้นเมื่อ 26 สิงหาคม 2562, จาก https://ieeexplore.ieee.org/document/8441730.

ภาคผนวก

ภาคผนวก ก

วิธีการติดตั้งโปรแกรม Arduino IDE และ Library

1. การติดตั้ง Arduino IDE

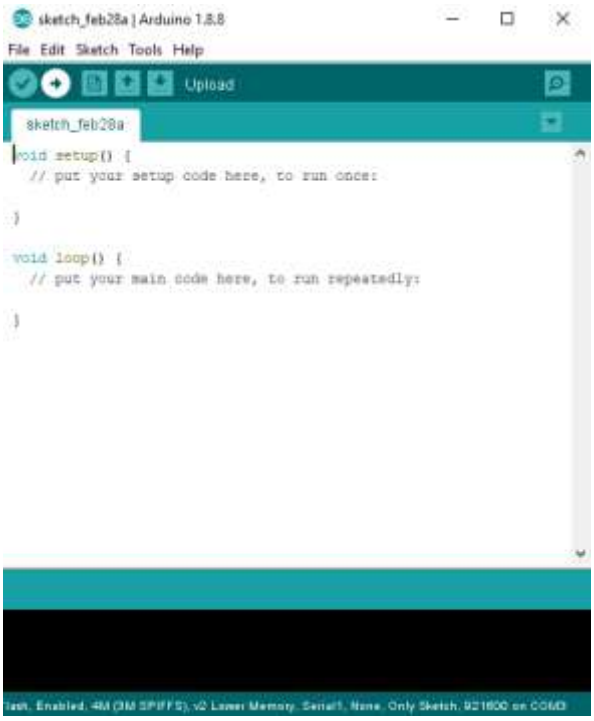
1.1 ดาวน์โหลด Arduino IDE จากเว็บไซต์ *https://www.arduino.cc/en/main/software* แล้วเลือกระบบปฏิบัติการของเครื่องที่ต้องการติดตั้ง

รูปที่ ก-1 เว็บไซต์ Arduino IDE

1.2 เลือกที่ Just Download เพื่อทำการดาวน์โหลดโปรแกรม

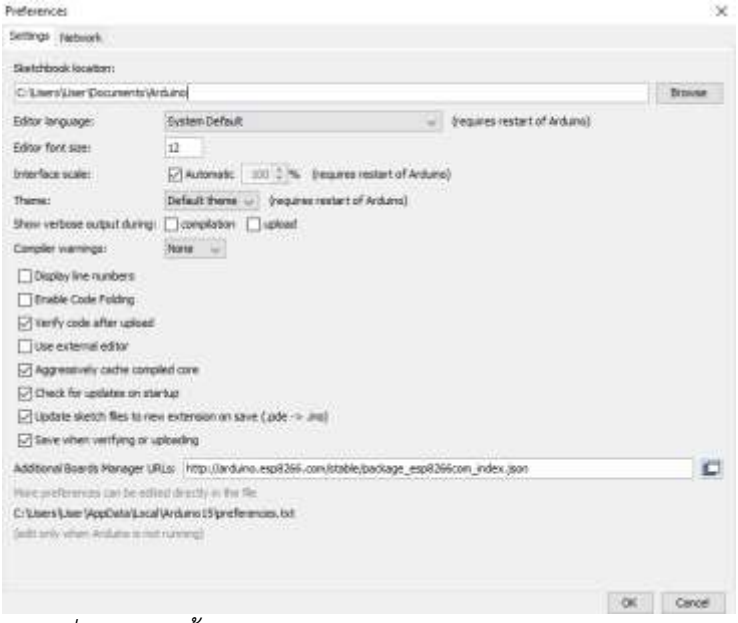
รูปที่ ก-2 หน้าเว็บไซต์สำหรับดาวน์โหลดโปรแกรม

1.3 เมื่อดาวน์โหลดเสร็จ ให้ทำการ Extract file และเริ่มการใช้งานโปรแกรมโดยคลิกที่ arduino.exe



รูปที่ ก-3 หน้าตาโปรแกรม Arduino IDE

1.4 เลือกที่ File -> Preferences เพื่อติดตั้งบอร์ด NodeMCU ESP8266 แบบออนไลน์ โดยทำการเพิ่มลิงก์ *http://arduino.esp8266.com/stable/package_esp8266com_index.json* ลงในช่อง Additional Boards Manager URLs ดังภาพ



รูปที่ ก-4 การตั้งค่าบอร์ดใน Arduino IDE [I]

1.5 คลิกเลือกที่เมนู Tools -> Board -> Board Manager และพิมพ์ค้นหาด้วยคำว่า ESP8266 ในช่องการค้นหา เพื่อเริ่มต้นติดตั้งบอร์ด ESP8266 ลงในโปรแกรม

รูปที่ ก-5 การตั้งค่าบอร์ดใน Arduino IDE [II]

1.6 ไปที่เมนู Tools เพื่อตั้งค่าบอร์ดและหมายเลขพอร์ต

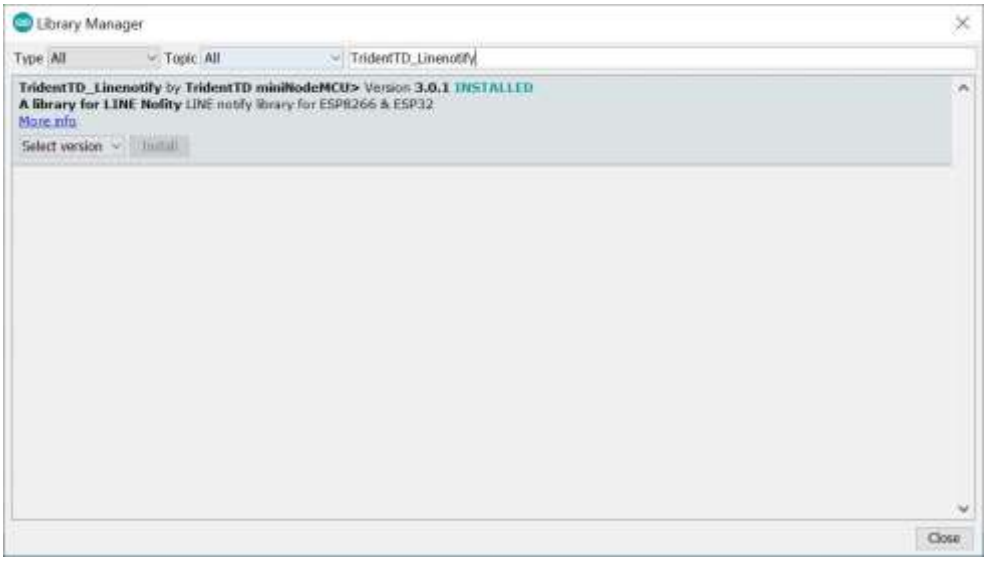
รูปที่ ก-6 การตั้งค่าบอร์ดใน Arduino IDE [III]

2. การติดตั้ง Library

2.1 ทำการติดตั้ง Library ดังนี้ (หากค้นหาแล้วไม่มี ให้ทำการค้นหาจากเว็บไซต์)

- ESP8266WiFi สำหรับการเชื่อมต่อ WiFi
- TridentTD_LineNotify สำหรับการเชื่อมต่อ LINE Notify
- ThingSpeak.h สำหรับการเชื่อมต่อ ThingSpeak
- LiquidCrystal_I2C สำหรับจอ LCD

โดยคลิกเลือกที่เมนู Tools -> Board -> Board Manager พิมพ์ Library ที่ต้องการติดตั้งลงในช่องค้นหา เพื่อเริ่มต้นการติดตั้ง

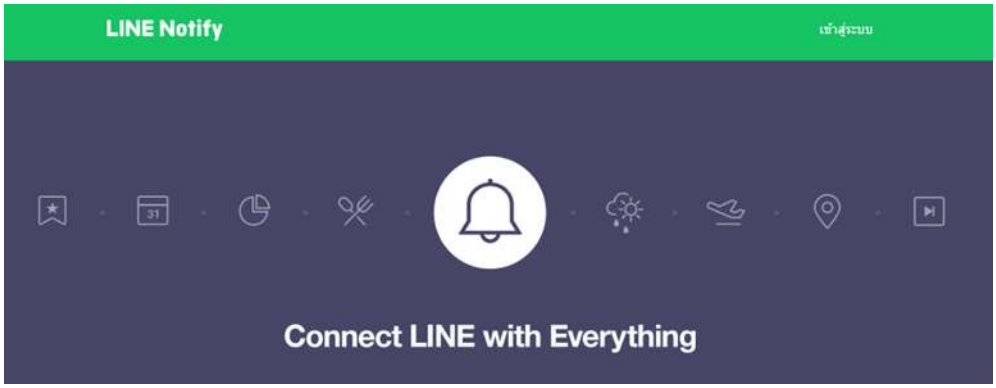


รูปที่ ก-7 การตั้งค่าบอร์ดใน Arduino IDE [IV]

ภาคผนวก ข

วิธีการใช้งาน Line Notify

1. เข้าไปที่เว็บไซต์ *https://notify-bot.line.me/th/*



รูปที่ ข-1 การสมัครเพื่อใช้บริการ LINE Notify [I]

2. ทำการเข้าสู่ระบบ เพื่อเข้าใช้งาน

รูปที่ ข-2 การสมัครเพื่อใช้บริการ LINE Notify [II]

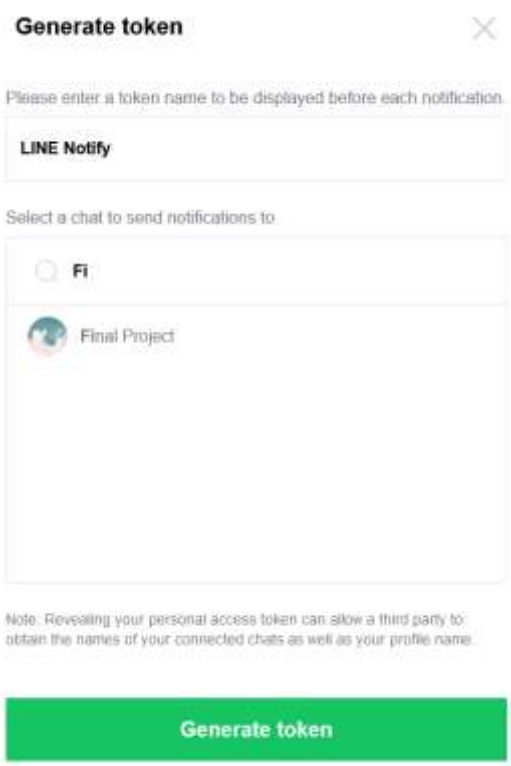
3. เมื่อเข้าสู่ระบบเสร็จ ให้คลิกที่ชื่อของบัญชีผู้ใช้ที่มุมขวาดังรูป ข-3 แล้วเลือก “หน้าของฉัน”

รูปที่ ข-3 การสมัครเพื่อใช้บริการ LINE Notify [III]

4. เลือก “Generate token” เพื่อสร้าง Token

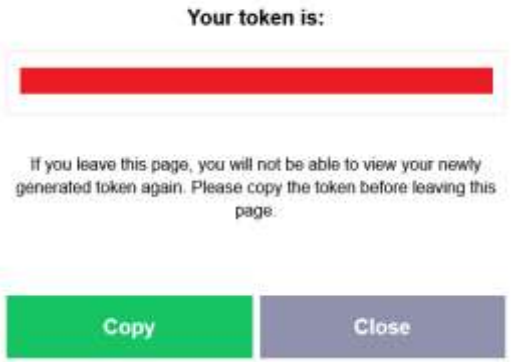
รูปที่ ข-4 การสมัครเพื่อใช้บริการ LINE Notify [IV]

5. ทำการใส่ชื่อของ Token เพื่อให้แสดงเมื่อมีการแจ้งเตือนในช่องแรก และเลือกบัญชีหรือกลุ่มแชทที่จะให้แสดงการแจ้งเตือน



รูปที่ ข-5 การสมัครเพื่อใช้บริการ LINE Notify [V]

6. คลิก “Generate token” หลังจากนั้นจะปรากฏ Token ดังภาพ เพื่อนำไปใช้โปรแกรม Arduino IDE

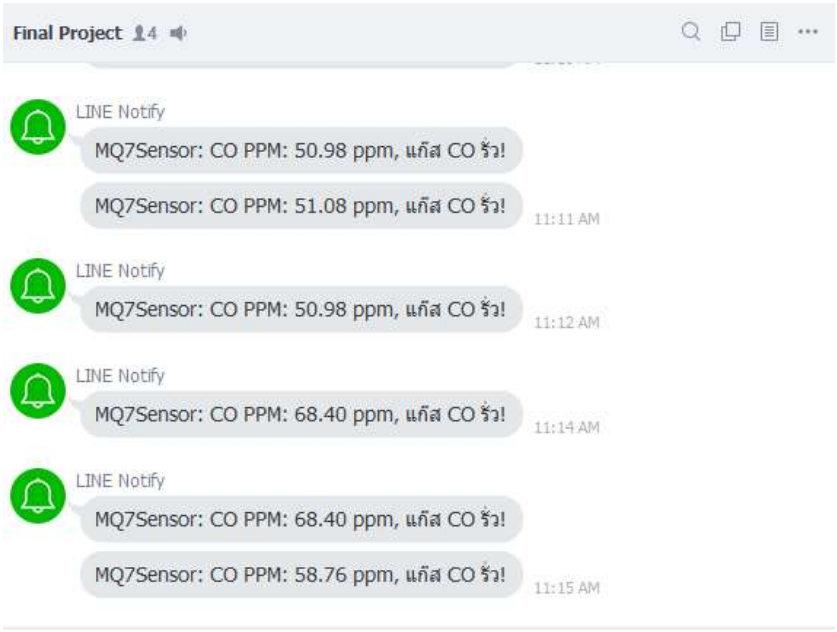


รูปที่ ข-6 การสมัครเพื่อใช้บริการ LINE Notify [VI]

7. เมื่อกดปิดหน้า “Generate token” จะแสดงบริการที่เชื่อมต่อ

รูปที่ ข-7 การสมัครเพื่อใช้บริการ LINE Notify [VII]

8. จากนั้นก็จะมีการแจ้งเตือนไปที่ Line ดังรูป ข-8 โดยหากมีการตรวจจับได้ว่ามีแก๊ส CO รั่ว อุปกรณ์ที่มีการติดตั้งจอ LCD I2c จะทำการประมวลผลและระบบจะทำการส่งการแจ้งเตือนอัตโนมัติผ่านทาง LINE Notify มาที่บัญชีหรือกลุ่มแชทที่ได้เชื่อมต่อไว้



รูปที่ ข-8 การสมัครเพื่อใช้บริการ LINE Notify [VII]

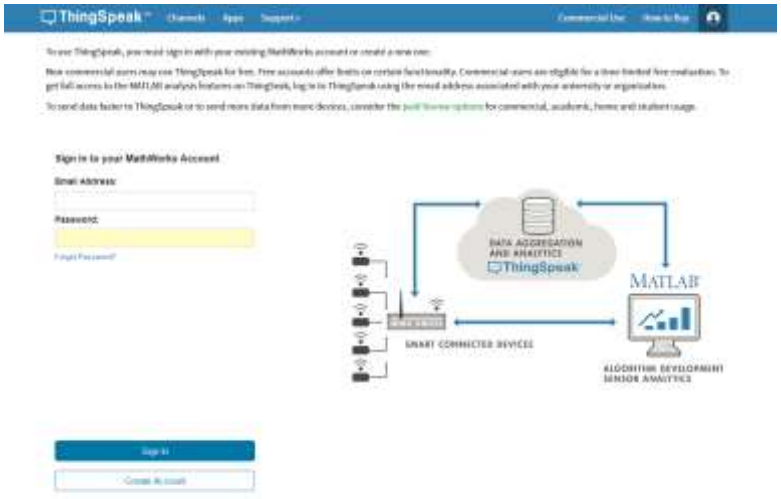
ภาคผนวก ค

วิธีสมัครเพื่อใช้บริการ ThingSpeak และนำข้อมูลไปใช้กับ MatLab

1. เข้าไปที่เว็บไซต์ *https://thingspeak.com/* โดยจะเห็นสัญลักษณ์วงกลมรูปคนที่ด้านมุมขวามือบน (ลูกศรสีแดง) ให้ตลิกที่ปุ่มนี้

รูปที่ ค-1 หน้าเว็บไซต์ ThingSpeak

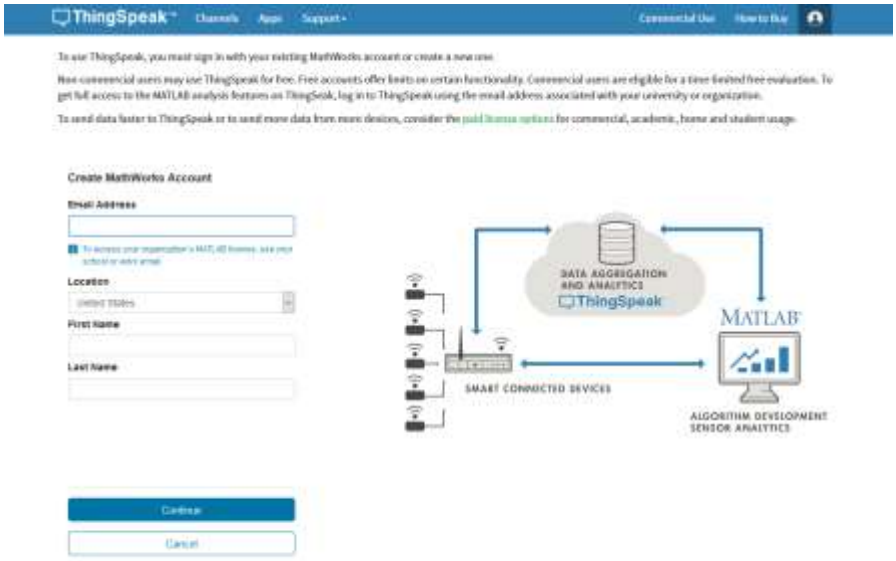
2. จะเข้าสู่หน้า Sign In ให้คลิกที่ปุ่ม Create Account เพื่อสร้างบัญชีใหม่



รูปที่ ค-2 หน้าสร้างบัญชีผู้ใช้ ThingSpeak [I]

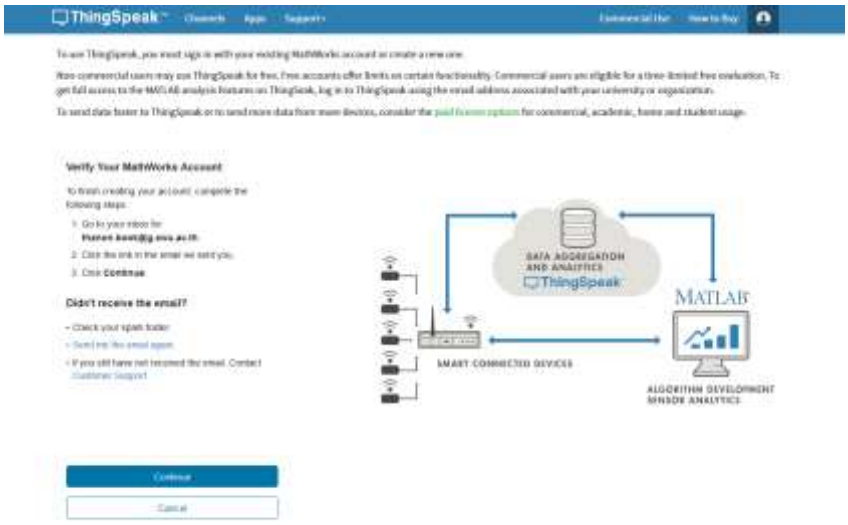
3. ในหน้าสร้างบัญชีผู้ใช้ใหม่ จะให้เรากรอกข้อมูลต่างๆ ดังนี้ แล้วกด Continue

    - Email Address (สำหรับการเข้าสู่ระบบ ให้ใช้อีเมล์ของมหาวิทยาลัย เพื่อได้สิทธิพิเศษที่มากกว่าอีเมล์ปกติ)
    - Location (ให้เลือก Thailand)
    - First Name (ชื่อจริง)
    - Last Name (นามสกุล)



รูปที่ ค-3 หน้าสร้างบัญชีผู้ใช้ ThingSpeak [II]

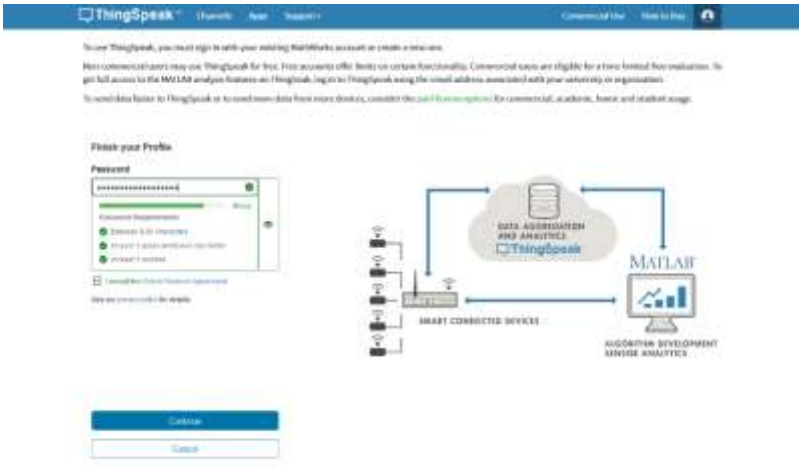
4. จะเข้าสู่หน้า Verify Your MathWorks Account ให้เข้าไปเช็คอีเมล์ที่ได้กรอกไว้ เพื่อยืนยันการสร้างบัญชีผู้ใช้ใหม่ จากนั้นทำการกด Continue



รูปที่ ค-4 หน้าสร้างบัญชีผู้ใช้ ThingSpeak [III]

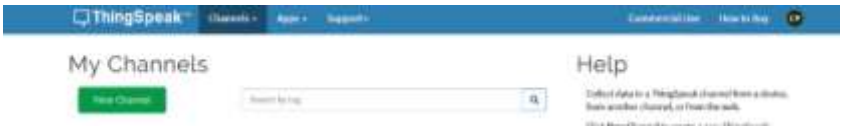
รูปที่ ค-5 อีเมล์ยืนยันการสร้างบัญชีผู้ใช้ใหม่จาก ThingSpeak

5. จะเข้าสู่หน้าให้กรอกรหัสผ่านที่ต้องการใช้ เมื่อกรอกเสร็จให้ติ้กถูกที่ช่อง “I accept the Online Services Agreement Online Services Agreement” และกด Continue



รูปที่ ค-6 หน้ากรอกรหัสผ่าน

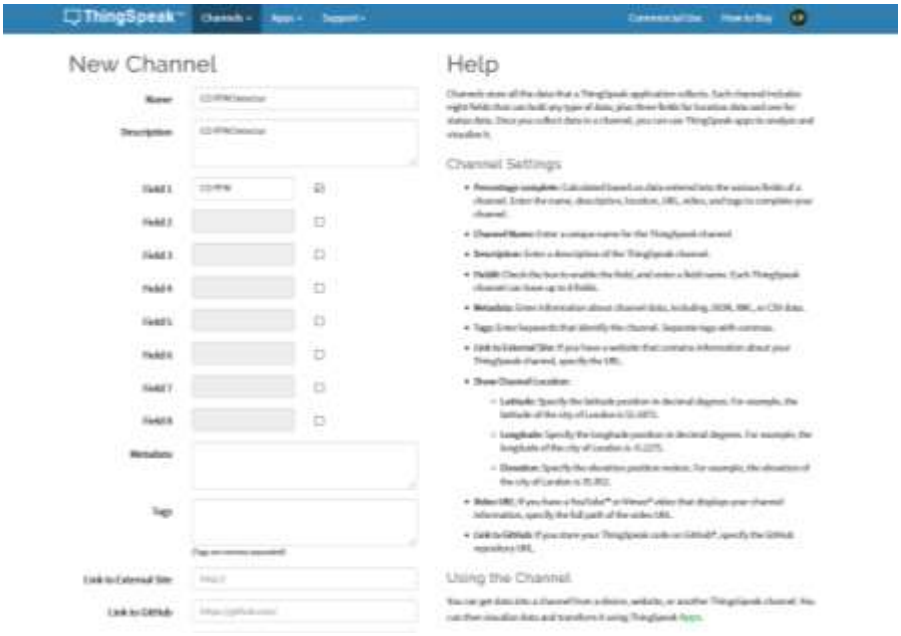
6. เมื่อกรอกรหัสผ่านเสร็จ จะเข้าสู่หน้า My Channel ของบัญชีผู้ใช้นั้นโดยอัตโนมัติ ให้คลิกที่ New Channel เพื่อทำการสร้าง Channel ใหม่



รูปที่ ค-7 หน้า My Channel

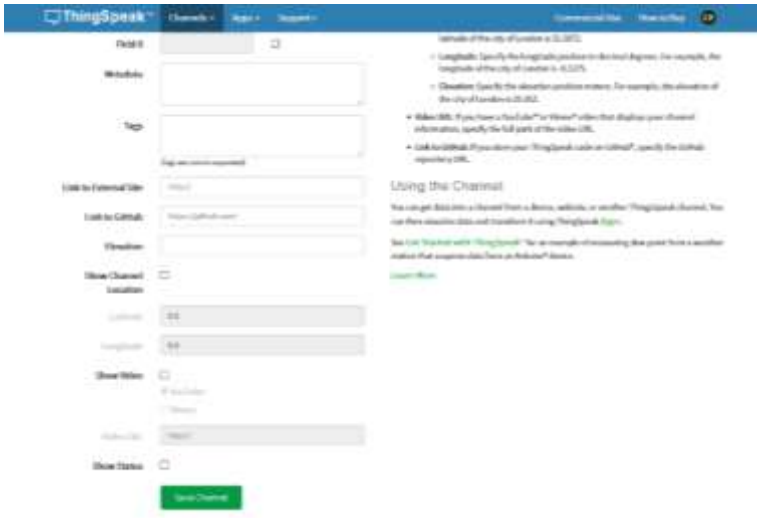
7. หน้า New Channel คือหน้าสำหรับการสร้างหน้าแสดงผลข้อมูลที่ต้องการนำมาแสดง โดยต้องกรอกรายละเอียดดังนี้

- Name (ชื่อหน้าแสดงผลข้อมูล)
- Field 1-8 (Parameter หรือข้อมูลที่ต้องการนำมาแสดง)



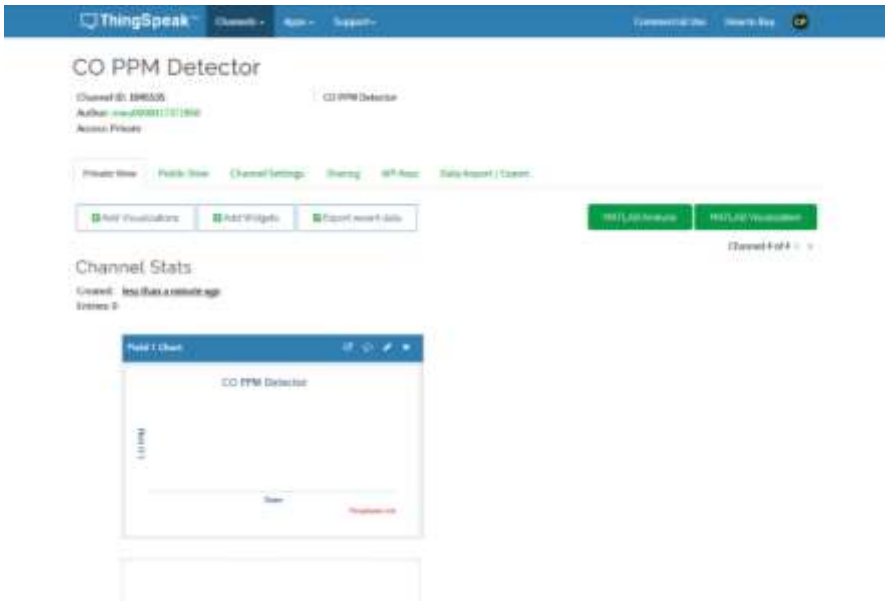
รูปที่ ค-8 หน้ากรอกรายละเอียด Channel [I]

8. เมื่อกรอกข้อมูลเรียบร้อยแล้ว ให้กด Save Channel เพื่อบันทึกการสร้างหน้าแสดงผลข้อมูล



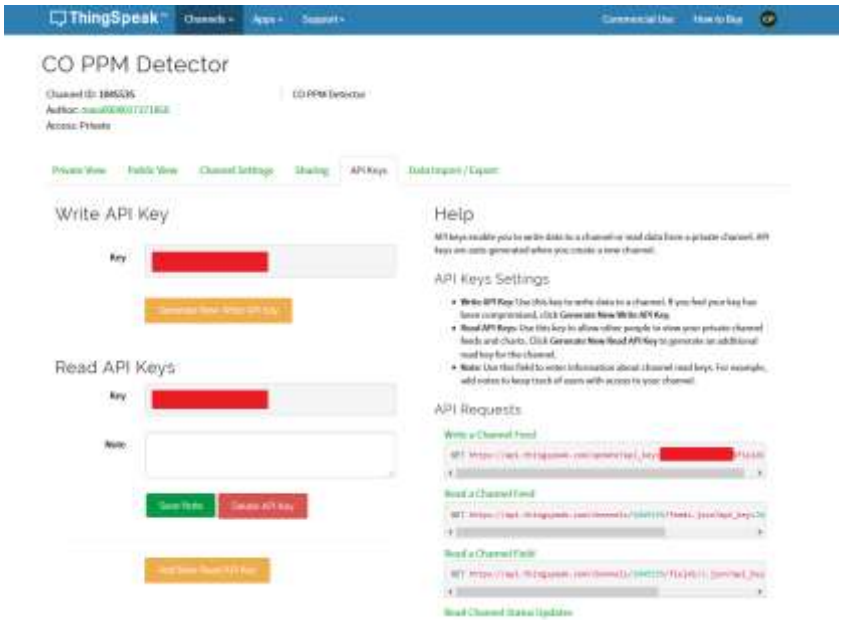
รูปที่ ค-9 หน้ากรอกรายละเอียด Channel [II]

9. หน้าแสดงผลข้อมูลหลังจากที่สร้างเสร็จเรียบร้อย โดยจะแสดงชื่อของ Channel และกราฟของข้อมูล



รูปที่ ค-10 หน้าแสดงผลข้อมูล

10. หน้า API Keys สำหรับการนำไปใช้ในโปรแกรม Arduino IDE เพื่อติดต่อกับ NodeMCU



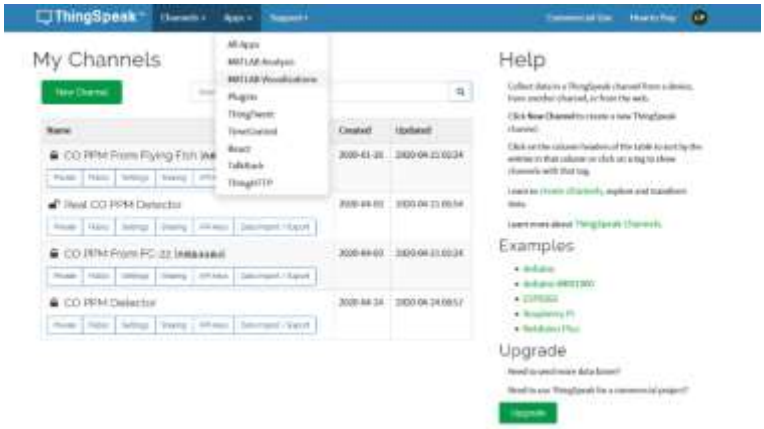
รูปที่ ค-11 หน้า API Keys

11. เมื่อทำการติดต่อกับ NodeMCU แล้ว จะมีข้อมูลขึ้นมาที่หน้าแสดงผลข้อมูลดังรูป ค-12

รูปที่ ค-12 หน้าแสดงผล เมื่อติดต่อกับ NodeMCU แล้ว

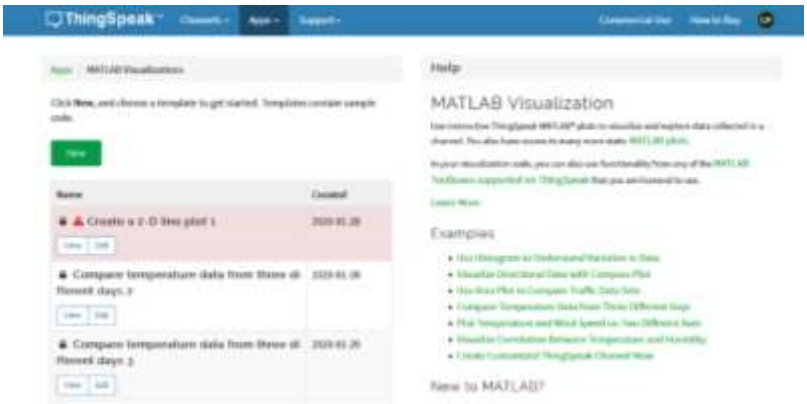
การนำข้อมูลไปใช้กับ MatLab

1. ให้กดที่แท็บ Apps และเลือก MATLAB Visualizaions



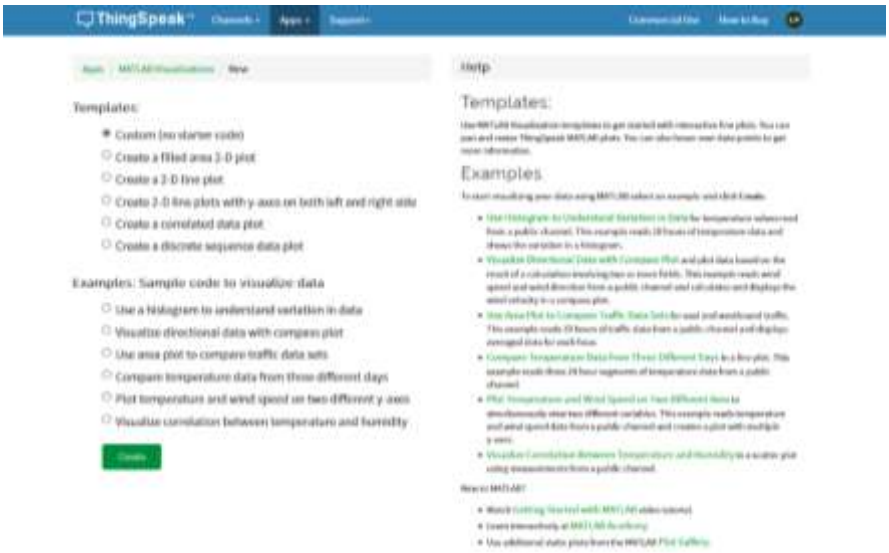
รูปที่ ค-13 การนำข้อมูลไปใช้กับ MatLab [I]

2. จากนั้นให้กด New เพื่อสร้างการทำงาน MatLab ใหม่



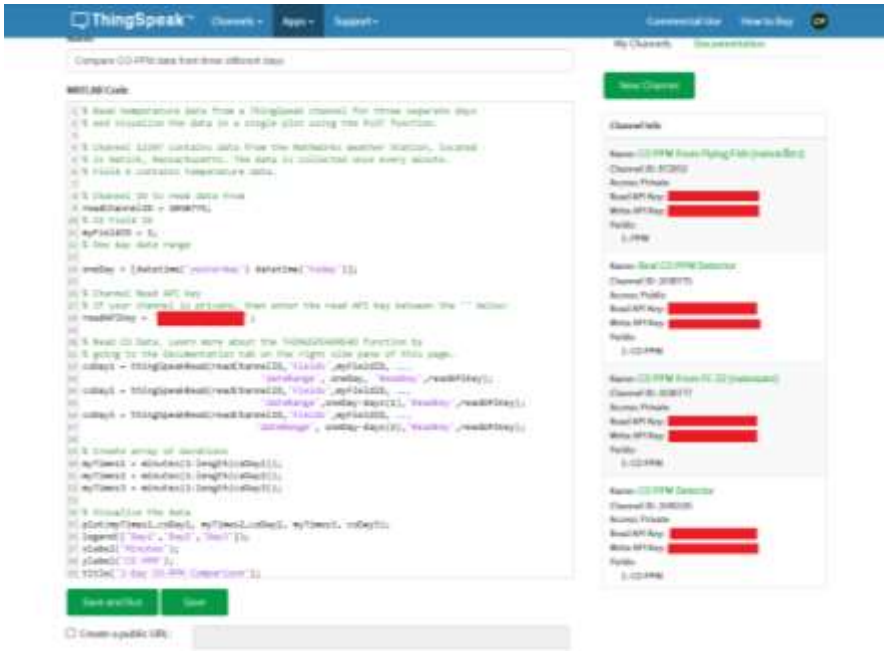
รูปที่ ค-14 การนำข้อมูลไปใช้กับ MatLab [II]

3. ให้เลือกที่ Custom (no starter code) แล้วกด Create



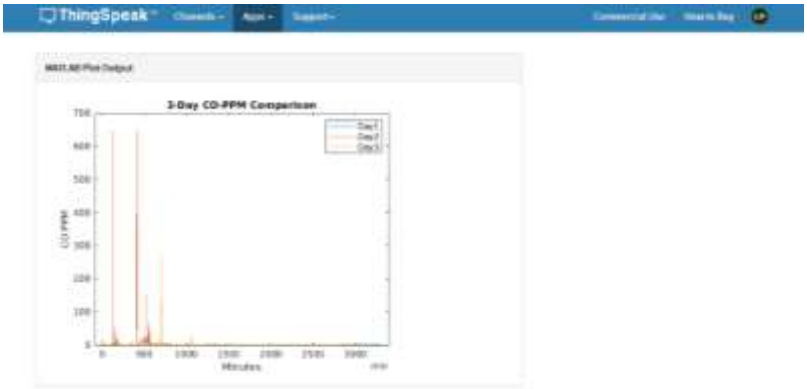
รูปที่ ค-15 การนำข้อมูลไปใช้กับ MatLab [III]

4. ให้เปลี่ยน Name เป็นชื่อที่ต้องการ และ MATLAB Code ให้ใช้โค้ดตามรูป ค-16 จากนั้นกด Save and Run



รูปที่ ค-16 การนำข้อมูลไปใช้กับ MatLab [IV]

5. จะได้กราฟแสดงข้อมูล 3 วันย้อนหลัง ซึ่งเป็นกราฟที่ให้ผู้ใช้สังเกตว่า แต่ละวันมีค่าปริมาณแก๊สเท่าใด มีแนวโน้มเพิ่มขึ้นหรือลดลงจากวันที่ผ่านมาหรือไม่



รูปที่ ค-17 การนำข้อมูลไปใช้กับ MatLab [V]