



โครงการ

การเรียนการสอนเพื่อเสริมประสบการณ์

ชื่อโครงการ การพัฒนาโดรนและอุปกรณ์ตรวจวัดแก๊สสำหรับการวิเคราะห์คุณภาพอากาศ
Development of drone and gas sensing device for air quality monitoring

ชื่อนิสิต นายสุทธิเกียรติ กรมจรรยา **เลขประจำตัว** 5833101323
ภาควิชา เคมี
ปีการศึกษา 2561

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของโครงการทางวิชาการที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)

เป็นแฟ้มข้อมูลของนิสิตเจ้าของโครงการทางวิชาการที่ส่งผ่านทางคณะที่สังกัด

The abstract and full text of senior projects in Chulalongkorn University Intellectual Repository(CUIR)
are the senior project authors' files submitted through the faculty.

การพัฒนาโดรนและอุปกรณ์ตรวจวัดแก๊สสำหรับการวิเคราะห์คุณภาพ
อากาศ

Development of drone and gas sensing device for air
quality monitoring

โดย

นายสุทธิเกียรติ กรมจรรยา

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรบัณฑิต

ภาควิชาเคมีคณะวิทยาศาสตร์

จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2561

โครงการ การพัฒนาโดรนและอุปกรณ์ตรวจวัดแก๊สสำหรับการวิเคราะห์คุณภาพอากาศ

โดย นายสุทธิเกียรติ กรมจรรยา

ได้รับอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาเคมี

ภาควิชาเคมี คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย


คณะกรรมการสอบโครงการ


..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.คณศ วงษ์ระวี)


..... อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ ดร.ภัทสรพล งามอุโฆษ)


..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.อนวัช อาชวาคม)

รายงานฉบับนี้ได้รับความเห็นชอบและอนุมัติโดยหัวหน้าภาควิชาเคมี


..... หัวหน้าภาควิชาเคมี
(รองศาสตราจารย์ ดร.วุฒิชัย พาราสุข)

วันที่ เดือน มิถุนายน พ.ศ. 2562

ชื่อโครงการ การพัฒนาโดรนและอุปกรณ์ตรวจวัดแก๊สสำหรับการวิเคราะห์คุณภาพอากาศ
ชื่อนิสิตในโครงการ นายสุทธิเกียรติ กรมจรรยา เลขประจำตัว 5833101323
ชื่ออาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร.ภัสสรพล งามอุโฆษ
ภาควิชาเคมี คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัยปีการศึกษา 2561

บทคัดย่อ

ปัจจุบันการตรวจวัดคุณภาพอากาศกลายเป็นประเด็นสำคัญที่เกี่ยวข้องกับคุณภาพชีวิตของคนในสังคม การใช้โดรนที่ติดตั้งหัวตรวจวัดเพื่อใช้ในการตรวจติดตามคุณภาพของอากาศจึงกลายเป็นแนวคิดที่น่าสนใจ อุปกรณ์ดังกล่าวนี้ไม่เพียงแต่สามารถส่งสัญญาณตอบสนองจากแก๊สชนิดต่าง ๆ ได้แต่ยังสามารถส่งข้อมูลอื่น เช่น อุณหภูมิ พิกัด GPS หรือแม้กระทั่งสัญญาณภาพแบบไร้สายตามเวลาจริงได้อีกด้วย ในงานวิจัยนี้ได้ทำการออกแบบอุปกรณ์โดยใช้ไมโครคอนโทรลเลอร์และทำการติดตั้งหัวตรวจวัดแก๊สซึ่งค่าความต้านทานไฟฟ้าจะเกิดการเปลี่ยนแปลงแตกต่างกันไปเมื่อตรวจพบแก๊สชนิดต่าง ๆ สัญญาณตอบสนองต่อแก๊ส LPG คาร์บอนมอนนอกไซด์ และอุณหภูมิสามารถแสดงผลผ่านสมาร์ทโฟนในลักษณะไร้สายตามเวลาจริงโดยอาศัยการต่อเชื่อมสัญญาณแบบ Wi-Fi จากการทดสอบเบื้องต้นพบว่าอุปกรณ์ที่พัฒนาขึ้นสามารถตรวจวัดแก๊สและอุณหภูมิแบบไร้สายได้โดยโดรนที่ประกอบขึ้นสามารถบินได้ตามปกติ แต่ด้วยข้อจำกัดในเรื่องการขออนุญาตบินโดรนในเขตชุมชนรวมถึงข้อจำกัดในด้านเวลาทำให้ไม่สามารถทดสอบการทำงานเต็มรูปแบบได้ตามที่วางแผนไว้ ดังนั้นการศึกษาประสิทธิภาพของอุปกรณ์ดังกล่าวจึงเป็นงานที่จะทำต่อไปในอนาคต

คำสำคัญ: ตรวจวัดคุณภาพอากาศ, โดรน, ไมโครคอนโทรลเลอร์, หัวตรวจวัด

Project Title Development of drone and gas sensing device for air quality monitoring
Student name Mr. Suttikiat Kromjanya Student ID 5833101323
Advisor name Assistant Professor Passapol Ngamukot, Ph.D.
Department of Chemistry, Faculty of Science, Chulalongkorn University, Academic Year 2018

Abstract

Recently, air-quality monitoring has become an issue of our society. An observation drone integrated with gas sensing probes has become an interesting concept. A whole device not only transmits all the signals response to the target gas species but also an ambient temperature, GPS data including video transmission in real-time. In this work, a whole device was fabricated by using microcontroller connected to temperature and gases sensors, in which their electrical resistance changes in the present of the target gas species. All signal responses to LPG, carbon monoxide and temperature can be wirelessly monitored in real-time with smartphone via a Wi-Fi connection. The fabricated device can be used to monitor all the target gas species and ambient temperature. The custom-made drone can also be operated without any problem. Unfortunately, due to a drone-flying regulation in public area and a limited time scale of this project, the full study of the fabricated device could not be completed as planned. All the analytical performance of the fabricated device should be investigated in the future.

Keywords: Air-quality monitoring, Drone, Microcontroller, Sensors

กิตติกรรมประกาศ

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.ภัสสรพล งามอุโฆษ อาจารย์ที่ปรึกษาโครงการเป็นอย่างสูงที่
สละเวลามาให้ความรู้คำแนะนำและความช่วยเหลือตลอดการทำงานวิจัยงานวิจัยนี้จึงสำเร็จได้รวมถึงการ
ตรวจทานและแก้ไขรายงานวิจัยฉบับนี้จนเสร็จสมบูรณ์

ขอขอบพระคุณ รองศาสตราจารย์ ดร.คณิศ วังษ์ระวี และ ผู้ช่วยศาสตราจารย์ ดร.อนวัช อาชวาคม
ที่สละเวลาให้เกียรติมาเป็นคณะกรรมการสอบโครงการในครั้งนี้

ขอขอบพระคุณอาจารย์ทุกท่านที่ให้ความรู้และข้อคิดตลอด 4 ปีที่ผ่านมา

ขอขอบคุณ นายเฉลิมวุฒิ ชำนาญญา ที่ช่วยให้คำแนะนำเกี่ยวกับเรื่องวงจรรีเลย์ทรานซิสเตอร์และชุดคำสั่ง
ของอุปกรณ์ตรวจวัด

ขอขอบคุณ นางสาวสุธิตา เพ็งโสภา ที่ช่วยให้คำแนะนำเกี่ยวกับงานวิจัยที่ผ่านมา

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ค
บทคัดย่อภาษาอังกฤษ	ง
กิตติกรรมประกาศ	จ
สารบัญ	ฉ
สารบัญรูปภาพ	ช
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและมูลเหตุจูงใจในการเสนอโครงการ	1
1.2 หลักการทำงานของเซนเซอร์แก๊ส	2
1.3 งานวิจัยที่เกี่ยวข้อง	3
1.4 แนวคิด เหตุผล และขอบเขต	4
1.5 วัตถุประสงค์ของโครงการ	4
1.6 ประโยชน์ที่คาดว่าจะได้รับ	5
บทที่ 2 การทดลอง	6
2.1 เครื่องมือและอุปกรณ์	6
2.2 วิธีการออกแบบอุปกรณ์	10
2.3 วิธีการทดลอง	12
2.4 ทดสอบการทำงานของโมดูลทั้ง 2 ชนิด	13
2.5 การทดสอบการส่งข้อมูลแบบไร้สาย	15
2.6 ทดสอบระยะทางของ Wi-Fi Hotspot	16
2.7 การนำอุปกรณ์ตรวจวัดไปติดตั้งที่โดรน	16

บทที่ 3 ผลการทดลองและอภิปรายผลการทดลอง	17
3.1. ทดสอบการทำงานของอุปกรณ์ต่างๆ	17
3.2. การทดสอบการส่งข้อมูลแบบไร้สาย	18
3.3. ทดสอบระยะทางของ Wi-Fi Hotspot	20
3.4. การนำอุปกรณ์ตรวจวัดไปติดตั้งที่โดรน	22
บทที่ 4 สรุปผลการทดลอง	24
4.1. ทดสอบการทำงานของอุปกรณ์ต่างๆ และ การส่งข้อมูลแบบไร้สาย	24
4.2. ทดสอบระยะทางของ Wi-Fi Hotspot	24
4.3. การนำอุปกรณ์ตรวจวัดไปติดตั้งที่โดรน	24
4.4. แนวทางแก้ไขสำหรับงานในอนาคต	24
เอกสารอ้างอิง	25
ภาคผนวก	27
ประวัติผู้วิจัย	49

สารบัญรูป

	หน้า
รูปที่ 1.1 ภาพแสดงอัตราส่วนของ Rs/Ro กับ ppm	2
รูปที่ 2.1 ไมโครคอนโทรลเลอร์ ESP8266 NodeMCU	7
รูปที่ 2.2 โมดูลเซนเซอร์แก๊สรุ่น MQ-2	7
รูปที่ 2.3 โมดูลวัดอุณหภูมิรุ่น DS18B20	8
รูปที่ 2.4 โดรน	9
รูปที่ 2.5 คอมพิวเตอร์	9
รูปที่ 2.6 แผนภาพลำดับการทำงานของอุปกรณ์ตรวจวัดแก๊สและอุณหภูมิ	11
รูปที่ 2.7 โปรแกรม Arduino IDE	12
รูปที่ 2.8 ผลการตรวจวัดใน Serial Monitor	13
รูปที่ 2.9 การติดตั้งอุปกรณ์ทดสอบเซนเซอร์ MQ-2	13
รูปที่ 2.10 ผลแสดงจาก serial monitor	14
รูปที่ 2.11 การติดตั้งอุปกรณ์เพื่อทดสอบเซนเซอร์ DS18B20	14
รูปที่ 2.12 การแสดงผลจาก Serial Monitor ที่ทดสอบ DS18B20	15
รูปที่ 2.13 หน้าเว็บ ThingSpeak	15
รูปที่ 2.14 รูปการแสดงผลจากเว็บของThingSpeak	16
รูปที่ 3.1 ภาพแสดง serial monitor ของ DS18B20	17
รูปที่ 3.2 ภาพแสดง Serial monitor ของ MQ-2 ที่วัดค่า CO,LPG และ ควัน	18
รูปที่ 3.3 ภาพแสดงการส่งข้อมูลที่ตรวจวัดแบบไร้สาย	18
รูปที่ 3.4 ผลการตรวจวัดอุณหภูมิที่ดูจากสมาร์ทโฟน	19
รูปที่ 3.5 ผลการตรวจวัด LPG ที่ดูจากสมาร์ทโฟน	19
รูปที่ 3.6 ผลการตรวจวัดควันที่ดูจากสมาร์ทโฟน	20
รูปที่ 3.7 ผลการตรวจวัด CO ที่ดูจากสมาร์ทโฟน	20
รูปที่ 3.8 แสดงตอนที่มีการเชื่อมต่อสัญญาณ (a) และเมื่อสัญญาณขาดการเชื่อมต่อ (b)	21
รูปที่ 3.9 การติดตั้งอุปกรณ์ตรวจวัดบนโดรน	22

รูปที่ 3.10 แผนภาพแสดงการดัดแปลงวงจร

บทที่ 1

บทนำ

1.1 ความเป็นมาและมูลเหตุจูงใจในการเสนอโครงการ

ในปัจจุบันมีการเกิดอุบัติเหตุบ่อยครั้งเช่นการระเบิดของโรงงานอุตสาหกรรม การเกิดเพลิงไหม้ เป็นต้น ซึ่งก่อให้เกิดมลพิษทางอากาศในปริมาณมากส่งผลให้เป็นอันตรายต่อสุขภาพและอาจเป็นอันตรายถึงชีวิตรวมไปถึงบุคคลที่อยู่ในบริเวณโดยรอบ ดังนั้นการพัฒนาโดรน (Drone) และอุปกรณ์ตรวจวัดแก๊สสำหรับการวิเคราะห์คุณภาพอากาศเพื่อใช้ในการสำรวจพื้นที่เกิดเหตุเพื่อตรวจวัดปริมาณแก๊สในอากาศเพื่อให้สามารถเตรียมพร้อมรับมือและป้องกันอย่างถูกวิธีในกรณีที่ต้องเข้าไปยังสถานที่เกิดเหตุและแจ้งเตือนบุคคลที่อยู่ในบริเวณโดยรอบให้ออกจากพื้นที่ในระยะทางที่พ้นเขตอันตรายได้นอกจากนี้ โดรนยังเหมาะสำหรับการทำงานในการสำรวจและเฝ้าระวังมลพิษทางอากาศ

ในปัจจุบันมีการนำโดรนมาใช้หลากหลายรูปแบบ เช่น นำกล้องมาติดเพื่อถ่ายรูปรูปจากมุมสูง ใช้ตรวจสอบสภาพจราจรและเก็บข้อมูลภูมิศาสตร์ เป็นต้น โดยโดรนที่ถูกนำไปดัดแปลงเพื่อใช้งานในรูปแบบต่าง ๆ คือ มัลติโรเตอร์ (Multicopter) ซึ่งสามารถเคลื่อนที่และเปลี่ยนทิศทางได้โดยใบพัด ทำให้สามารถบินขึ้น-ลงในแนวตั้งได้ โดยอุปกรณ์ตรวจวัดปริมาณแก๊สและอุณหภูมิที่นำมาติดบนโดรนสามารถประดิษฐ์ได้โดยใช้ไมโครคอนโทรลเลอร์ (Microcontroller)² ซึ่งเป็นอุปกรณ์ควบคุมขนาดเล็ก โดยบรรจุความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์นำไปต่อเชื่อมกับโมดูล (Module) และเชื่อมต่อกับระบบเครือข่ายไร้สาย (Wireless Network) เพื่อติดตามข้อมูลผ่านเว็บที่สามารถดูได้ทั้งในคอมพิวเตอร์และสมาร์ทโฟน (Smartphone)³ ทำให้ติดตามผลการตรวจวัดได้อย่างต่อเนื่องและรวดเร็ว

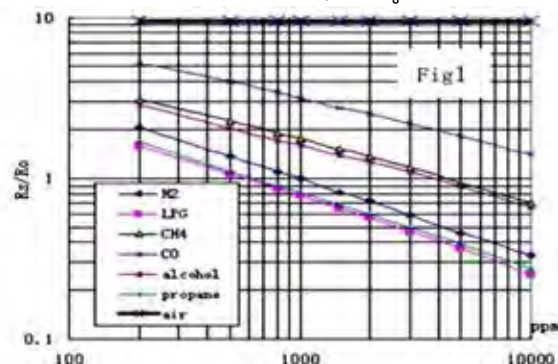
ดังนั้นในงานวิจัยนี้จึงสนใจที่จะพัฒนาโดรนและอุปกรณ์ตรวจวัดแก๊สสำหรับการวิเคราะห์คุณภาพอากาศที่ตรวจวัดปริมาณแก๊สคาร์บอนมอนอกไซด์และแก๊สที่ติดไฟได้ เช่น ไฮโดรเจน โพรเพน บิวเทน และแก๊สธรรมชาติต่างๆรวมทั้งอุณหภูมิของพื้นที่โดยจะแสดงผลในรูปแบบความเข้มข้นและองศาเซลเซียสตามลำดับซึ่งสามารถติดตามผลการตรวจวัดได้อย่างต่อเนื่องและสามารถดูพื้นที่โดยรอบผ่านกล้องที่ติดในโดรนได้โดยมีแผนที่แสดงตำแหน่งผ่านจีพีเอส (GPS)⁴ และให้อุปกรณ์เชื่อมต่อกับระบบเครือข่ายไร้สาย ซึ่งใช้เซ็นเซอร์สำหรับตรวจวัดแก๊สคาร์บอนมอนอกไซด์ คือ โมดูลเซนเซอร์แก๊สรุ่น MQ-2 เมื่อเราเริ่มจ่ายพลังงานให้ MQ-2 ที่ขาของเซนเซอร์ทำให้เกิดความร้อนขึ้นที่ขดลวด เมื่อก๊าซไวไฟต่างๆ เข้ามาทำปฏิกิริยาจะทำให้ค่าความต้านทานที่เกิด (Rs) ลดลงหรือค่าความต้านทานแปรผกผันกับปริมาณของ ก๊าซไวไฟต่างๆ และเซ็นเซอร์สำหรับ

ตรวจวัดอุณหภูมิมาต่อเชื่อมกับไมโครคอนโทรลเลอร์ คือโมดูลรุ่น DS18B20 เป็นไอซี (Integrated Circuit, IC) ตรวจวัดอุณหภูมิแบบดิจิตอลของ Dallas Semiconductor สามารถตรวจจับอุณหภูมิที่ช่วง -55°C ถึง 125°C และเขียนชุดคำสั่งเพื่อให้เห็นผลและเก็บข้อมูลซึ่งวิธีนี้ทำให้ไมโครคอนโทรลเลอร์สามารถตรวจวัดคุณภาพอากาศในพื้นที่อันตรายและมีความเสี่ยงหากใช้บุคคลเข้าไปตรวจวัดนอกจากนี้ข้อมูลเซ็นเซอร์และภาพจากกล้องที่ส่งผ่านระบบเครือข่ายไร้สายยังช่วยให้สามารถประเมินสถานการณ์และความเสี่ยงที่เกิดขึ้นได้

1.2 หลักการทำงานของเซนเซอร์แก๊ส

หลักการทำงานของเซนเซอร์ MQ - 2 คือเมื่อไมโครคอนโทรลเลอร์ได้รับแรงดันไฟฟ้าจากเซนเซอร์แก๊สพลังงานไฟฟ้าจะทำให้ขดลวดร้อนเมื่อขดลวดร้อนแล้วทินออกไซด์ (SnO_2) ที่อยู่ภายในตัวเซนเซอร์ซึ่งมีการนำไฟฟ้าต่ำหรือมีค่าต้านทานไฟฟ้าสูงในสภาวะอากาศปกติกลายเป็นสารกึ่งตัวนำส่งผลให้อิเล็กตรอนมีการไหลเวียนจากชั้น Valence band ขึ้นไปยังชั้น Conduction band ที่ผิวของทินออกไซด์จึงอิเล็กตรอนหนาแน่นและแก๊สที่วัดจะปรับอิเล็กตรอนทำให้ประจุลบบริเวณผิวมีค่าน้อยลงค่าการนำไฟฟ้าของเซนเซอร์จึงสูงขึ้นหรือค่าความต้านทานไฟฟ้าต่ำลงแปรผันตามความเข้มข้นของแก๊สที่เพิ่มมากขึ้นเนื่องจากค่าการนำไฟฟ้าหรือค่าความต้านทานไฟฟ้าเกิดการเปลี่ยนแปลงจึงสามารถหาปริมาณของแก๊สได้จากค่าความแตกต่างของค่าการนำไฟฟ้าหรือค่าความต้านทานไฟฟ้าที่เปลี่ยนแปลงไปของเซนเซอร์แก๊ส

นอกจากนี้แล้วเรายังสามารถแยแยะประเภทของแก๊สออกได้ โดยการสร้างกราฟ (R_s/R_o) กับความเข้มข้น (ppm) ซึ่ง R_o คือค่าความต้านทานในอากาศ และ R_s คือค่าความต้านทานในความเข้มข้นของแก๊ส โดยขั้นแรกทำการเปรียบเทียบค่าของเซนเซอร์ (Calibration) ของความต้านทานในอากาศหลังจากนั้นคำนวณหาความต้านทานในความเข้มข้นของแก๊ส เมื่อเราคำนวณ R_s และ R_o แล้วเราสามารถหาอัตราส่วนแล้วสามารถคำนวณค่าเทียบกับความเข้มข้นสำหรับก๊าซนั้นๆได้ดังรูปที่ 1.1



รูปที่ 1.1 ภาพแสดงอัตราส่วนของ R_s/R_o กับ ppm

1.3 งานวิจัยที่เกี่ยวข้อง

1.3.1 การรักษาความปลอดภัยของมนุษย์จากแก๊สอันตรายโดยใช้ระบบเซ็นเซอร์อัจฉริยะ

จากงานวิจัยของ Ambeth¹ ศึกษาเกี่ยวกับแก๊สที่เป็นอันตรายต่อมนุษย์ เช่น แก๊สไฮโดรเจนซัลไฟด์, แก๊สคาร์บอนมอนอกไซด์, แก๊สมีเทน, แก๊สซัลเฟอร์ไดออกไซด์, แก๊สไนโตรเจนออกไซด์ เป็นต้น จึงพัฒนาระบบเซ็นเซอร์ตรวจวัดปริมาณแก๊สขึ้น ซึ่งสามารถตรวจวัดปริมาณความเข้มข้นของแก๊สคาร์บอนมอนอกไซด์และแก๊สที่ติดไฟได้ และระบุขีดจำกัดความปลอดภัยของแก๊สเหล่านี้ได้ โดยใช้ไมโครคอนโทรลเลอร์ ยี่ห้อ Arduino รุ่น Atmega328 สำหรับประมวลผลและใช้โมดูล 3 ชนิด คือ โมดูลตรวจวัดแก๊สคาร์บอนมอนอกไซด์ (MQ-7), โมดูลตรวจวัดแก๊สที่ติดไฟได้ (MQ-2) และโมดูลสำหรับเชื่อมต่อผ่านเครือข่ายไร้สาย (GSM shield) นอกจากนี้ยังติดตั้งสัญญาณเตือนภัยและไดโอดเปล่งแสง (LED) สำหรับแจ้งเตือนเมื่อแก๊สที่มีอยู่ในอากาศเกินเกณฑ์ที่กำหนด โดยมีหลักการทำงาน คือ เมื่อมีแก๊สเกิดมากขึ้น เซ็นเซอร์สามารถตรวจจับและส่งไปประมวลผลที่ไมโครคอนโทรลเลอร์ โดยจะแสดงผลผ่านจอภาพผลึกเหลว (LCD) ในรูปความเข้มข้น ถ้าแก๊สมีปริมาณมากเกินกว่าที่กำหนด ระบบจะส่งข้อความแจ้งเตือนโดยใช้ระบบเครือข่ายไร้สาย

1.3.2 ระบบเตือนภัยฉุกเฉินอัตโนมัติผ่านคลาวด์เซิร์ฟเวอร์และบริการเครือข่ายสังคม

งานวิจัยของ Buribayeva และคณะ³ พบว่าเมืองอัลมาตี ประเทศคาซัคสถาน ไม่มีระบบเตือนภัยสำหรับภัยพิบัติแผ่นดินไหวอัคคีภัยและแก๊สรั่วไหล จึงออกแบบสร้างระบบสำหรับเตือนภัยฉุกเฉินอัตโนมัติผ่านทางคลาวด์เซิร์ฟเวอร์และบริการเครือข่ายสังคม โดยตรวจจับคลื่นแผ่นดินไหว คิวไฟ และแก๊สที่เป็นอันตราย ซึ่งระบบจะส่งข้อความเตือนโดยอัตโนมัติผ่านทางทวีตเตอร์และเว็บไซต์ของระบบ โดยระบบนี้สามารถระบุระดับความเป็นอันตรายและแผนที่แสดงบริเวณที่ปลอดภัยสำหรับพลเมืองที่อาศัยอยู่ใกล้บริเวณที่เป็นอันตราย ระบบเตือนภัยฉุกเฉินนี้ประกอบด้วยเซิร์ฟเวอร์ฐานข้อมูลภัยพิบัติ (CloRob), เซิร์ฟเวอร์ Google-map, เว็บไซต์เซิร์ฟเวอร์ฉุกเฉิน (Twitter), บอร์ดยี่ห้อ Arduino รุ่น Mega ADK, ไมโครคอนโทรลเลอร์สำหรับเชื่อมต่ออินเทอร์เน็ตยี่ห้อ Arduino รุ่น ATmega328 และโมดูล 4 ชนิด ได้แก่ โมดูลตรวจจับความสั่นไหว, โมดูลตรวจวัดแก๊สคาร์บอนมอนอกไซด์ (MQ-7), โมดูลตรวจวัดแก๊สที่ติดไฟได้ (MQ-2) และโมดูลตรวจจับเปลวไฟ

1.3.3 การออกแบบและติดตั้งไอโอทีเกตเวย์เพื่อสร้างระบบสิ่งแวดล้อมอัจฉริยะ

งานวิจัยของ Gl'oria และคณะ⁴ ได้ออกแบบและสร้างระบบที่สามารถตรวจสอบแบบเรียลไทม์และควบคุมสภาพแวดล้อมของสระว่ายน้ำจากระยะไกลได้ โดยใช้รหัสเบอร์รี่พายเป็นโหนดรวม เกตเวย์ช่วยให้สามารถสื่อสารแบบสองทิศทางและแลกเปลี่ยนข้อมูลระหว่างผู้ใช้กับเครือข่ายเซ็นเซอร์ที่ใช้วัดและควบคุมสภาพแวดล้อมโดยใช้ไมโครคอนโทรลเลอร์ ยี่ห้อ Arduino ซึ่งสามารถตรวจสอบอุณหภูมิและระดับของน้ำ, อุณหภูมิแวดล้อม, ความชื้นสัมพัทธ์, มลพิษในอากาศ และความสว่าง นอกจากนี้สามารถควบคุมเครื่องสูบน้ำและไฟสระว่ายน้ำได้จากระยะไกลได้

1.4 แนวคิดเหตุผลและขอบเขต

จากงานวิจัยที่เกี่ยวข้องจะเห็นได้ว่าทั้งสามงานวิจัยนี้มีข้อจำกัดคืออุปกรณ์ที่สร้างขึ้นไม่สะดวกต่อการเคลื่อนย้ายแต่งงานวิจัยที่ 1.3.2 และ 1.3.3 มีข้อดีคือสามารถส่งข้อมูลผ่านระบบเครือข่ายไร้สายได้ผู้วิจัยจึงนำข้อดีและข้อด้อยนี้มาปรับปรุงให้งานวิจัยนี้สามารถนำอุปกรณ์ที่สร้างขึ้นไปติดตั้งบนโดรนและสามารถส่งข้อมูลจากระยะไกลผ่านระบบเครือข่ายไร้สายและติดตามผลแบบต่อเนื่องได้โดยมีขอบเขตงานวิจัยดังนี้

1. อุปกรณ์ที่สร้างขึ้นสามารถใช้กับไมโครคอนโทรลเลอร์รุ่น ESP8266 NodeMCU เท่านั้นโดยเชื่อมต่อกับโมดูลตรวจวัดปริมาณแก๊สที่ติดไฟได้ (LPG), โมดูลตรวจวัดปริมาณแก๊สคาร์บอนมอนอกไซด์และโมดูลตรวจวัดอุณหภูมิและเป็นตัวส่งข้อมูลผ่านระบบเครือข่ายไร้สาย

2. โปรแกรมสำหรับควบคุมอุปกรณ์ (Arduino IDE) เป็นโปรแกรมที่ใช้ภาษาซีในการเขียนโปรแกรมและใช้โค้ด (Code) สำหรับไมโครคอนโทรลเลอร์รุ่น ESP8266 NodeMCU เท่านั้น

1.5 วัตถุประสงค์ของโครงการ

1. ออกแบบและสร้างอุปกรณ์ตรวจวัดปริมาณแก๊สและอุณหภูมิสำหรับการวิเคราะห์ในพื้นที่ต่าง ๆ และสามารถนำไปติดตั้งที่โดรนได้

2. พัฒนาอุปกรณ์ตรวจวิเคราะห์คุณภาพอากาศให้สามารถส่งข้อมูลและติดตามผลแบบต่อเนื่องผ่านระบบเครือข่ายไร้สายเพื่อดูข้อมูลผ่านเว็บในคอมพิวเตอร์และสมาร์ตโฟนได้

1.6 ประโยชน์ที่คาดว่าจะได้รับ

ได้อุปกรณ์ตรวจวัดปริมาณแก๊สและอุณหภูมิที่ติดตั้งบนโดรนและติดตามผลแบบต่อเนื่องผ่านระบบเครือข่ายไร้สายคอมพิวเตอร์และสมาร์ทโฟน

บทที่ 2

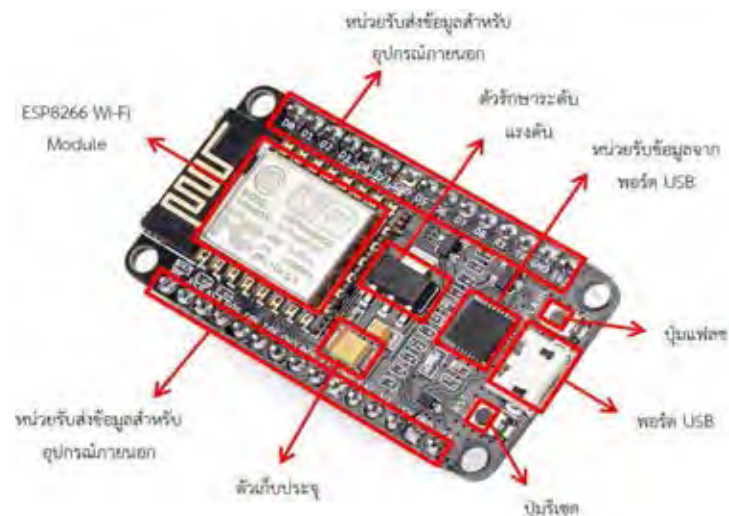
การทดลอง

2.1 เครื่องมือและอุปกรณ์

2.1.1 ไมโครคอนโทรลเลอร์รุ่น ESP8266 NodeMCU

ไมโครคอนโทรลเลอร์รุ่น ESP8266 NodeMCU รูปที่ 2.1 ประกอบด้วย

- 2.1.1.1 พอร์ต USB ใช้สำหรับรับชุดคำสั่งจากคอมพิวเตอร์สู่ไมโครคอนโทรลเลอร์และเป็นตัวรับไฟเข้าไปในบอร์ดได้อีกด้วย
- 2.1.1.2 ESP8266 Wi-Fi Module ใช้สำหรับเชื่อมต่อระบบเครือข่ายไร้สายและโหลดโปรแกรมลงไป
- 2.1.1.3 ตัวเก็บประจุ
- 2.1.1.4 ตัวรักษาระดับแรงดัน ใช้สำหรับรับแรงดันจากอุปกรณ์ภายนอกโดยต่อจากพอร์ต USB
- 2.1.1.5 หน่วยรับข้อมูลจากพอร์ต USB ใช้สำหรับรับข้อมูลจากพอร์ต USB
- 2.1.1.6 ปุ่มรีเซ็ต ใช้สำหรับเริ่มต้นการทำงานของบอร์ดใหม่
- 2.1.1.7 ปุ่มแฟลช ใช้สำหรับโปรแกรมเฟิร์มแวร์ใหม่
- 2.1.1.8 หน่วยรับส่งข้อมูลสำหรับอุปกรณ์ภายนอก ใช้สำหรับรับและส่งข้อมูลระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก



รูปที่ 2.1 ไมโครคอนโทรลเลอร์ ESP8266 NodeMCU

2.1.2 โมดูล

เป็นอุปกรณ์ที่นำไปเชื่อมต่อกับไมโครคอนโทรลเลอร์โดยในงานวิจัยนี้ใช้โมดูล 2 ชนิดได้แก่โมดูล เซนเซอร์แก๊สรุ่น MQ-2 และโมดูลวัดอุณหภูมิรุ่น DS18B20

2.1.2.1 โมดูลเซนเซอร์แก๊สรุ่น MQ-2

เป็นโมดูลที่ตรวจวัดปริมาณแก๊สคาร์บอนมอนอกไซด์, แอลพีจี (LPG) (โพรเพน, ไฮโดรเจน, มีเทน, บิวเทน, แอลกอฮอล์) และควันซึ่งสามารถตรวจจับระดับความเข้มข้นของแก๊สในช่วง 300 - 10,000 ppm รูปที่ 2.2



รูปที่ 2.2 โมดูลเซนเซอร์แก๊สรุ่น MQ-2

2.1.2.2 โมดูลวัดอุณหภูมิรุ่น DS18B20

เป็นไอซี (Integrated Circuit, IC) ตรวจวัดอุณหภูมิแบบดิจิทัลของ DallasSemiconductor สามารถตรวจจับอุณหภูมิที่ช่วง -55°C ถึง 125°C (รูปที่ 2.3)



รูปที่ 2.3 โมดูลวัดอุณหภูมิรุ่น DS18B20

2.1.3 โดรน

เป็นอุปกรณ์สำหรับสำรวจพื้นที่โดยสามารถถ่ายภาพ บันทึกวีดีโอและระบุพิกัดตำแหน่งได้นอกจากนี้สามารถนำอุปกรณ์ที่พัฒนาในงานวิจัยนี้ไปติดกับตัวโดรนเพื่อตรวจวัดได้ รูปที่ 2.4

โดรนประกอบสำเร็จมีคุณสมบัติดังนี้

1. สามารถรับน้ำหนักได้สูงสุด 2 กิโลกรัม
2. สามารถบินได้ 20 นาที/รอบ
3. มีพื้นที่สำหรับวางอุปกรณ์ตรวจวัดแก๊สและอุณหภูมิ
4. สามารถถ่ายและบันทึกภาพซึ่งสามารถดูผ่านมอนิเตอร์ได้
5. มีแผนที่แสดงตำแหน่งผ่านจีพีเอส



รูปที่ 2.4 โดรน

2.1.4 คอมพิวเตอร์

ต้องมีระบบปฏิบัติการตั้งแต่ Window 7 ขึ้นไปใช้สำหรับเขียนโปรแกรมเพื่อควบคุมไมโครคอนโทรลเลอร์และใช้ติดตามผลซึ่งส่งผ่านทางระบบเครือข่ายไร้สาย

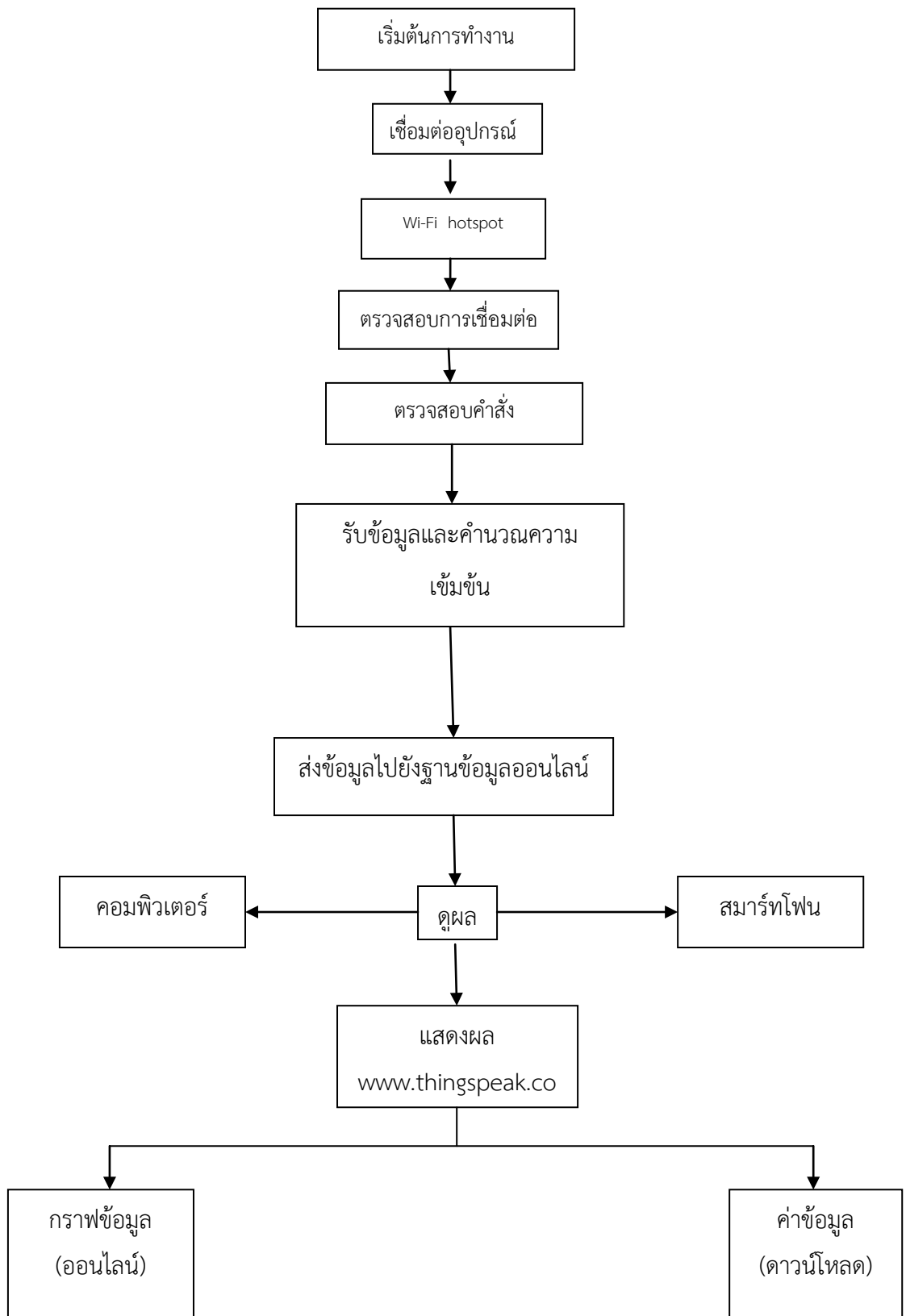


รูปที่ 2.5 คอมพิวเตอร์

2.2 วิธีการออกแบบอุปกรณ์

อุปกรณ์ที่สร้างขึ้นต้องมีการรับ-ส่งข้อมูลจากโมดูลทั้ง 2 ชนิดผ่านระบบเครือข่ายไร้สายได้โดยมีลำดับการทำงานดังรูปที่ 2.6 ดังนี้

- 2.2.1 **เริ่มต้นการทำงาน** แหล่งจ่ายกระแสไฟฟ้าจ่ายกระแสไฟฟ้าให้ไหลไปเลี้ยงวงจรของอุปกรณ์ เพื่อเริ่มต้นการทำงาน
- 2.2.2 **เชื่อมต่อกับอุปกรณ์** อุปกรณ์จะเชื่อมต่อกับอุปกรณ์ภายนอกทันทีหลังจากเริ่มต้นการทำงาน โดยสามารถเชื่อมต่อกับอุปกรณ์สำหรับกระจายสัญญาณเครือข่ายไร้สาย (Wi-Fi Hotspot) เป็นการเชื่อมต่อผ่านระบบเครือข่ายไร้สายเพื่อรับ-ส่งข้อมูลไปยังฐานข้อมูลออนไลน์โดยสามารถติดตามผลการตรวจวัดได้ทันที
- 2.2.3 **ตรวจสอบการเชื่อมต่อ** อุปกรณ์สามารถตรวจสอบการเชื่อมต่อกับ Wi-Fi Hotspot ได้หากไม่มีการเชื่อมต่อจะทำการเชื่อมต่อใหม่จนกว่าการเชื่อมต่อจะสำเร็จ
- 2.2.4 **ตรวจสอบคำสั่ง** อุปกรณ์จะตรวจสอบคำสั่งโค้ดที่อัปเดตลงไป
- 2.2.5 **รับข้อมูลวัดอุณหภูมิและคำนวณความเข้มข้นของแก๊ส** อุปกรณ์จะรับข้อมูลจากเซ็นเซอร์แก๊สและเซ็นเซอร์อุณหภูมิจากนั้นคำนวณความเข้มข้นของแก๊สก่อนจะส่งข้อมูลออกไป
- 2.2.6 **ส่งข้อมูลไปยังฐานข้อมูลออนไลน์** อุปกรณ์จะส่งข้อมูลอุณหภูมิและความเข้มข้นของแก๊สไปที่ฐานข้อมูลออนไลน์ที่กำหนดไว้เพื่อรายงานผลให้ผู้ใช้งานทราบ
- 2.2.7 **ดูผล** ใช้คอมพิวเตอร์หรือ สมาร์ทโฟนเพื่อดูผลที่ฐานข้อมูลออนไลน์ที่ผู้ใช้งานกำหนดไว้
- 2.2.8 **แสดงผล** การแสดงผลข้อมูลสามารถแสดงผลได้ 2 แบบได้แก่กราฟข้อมูลและค่าข้อมูลโดยกราฟข้อมูลสามารถดูผลการตรวจวัดได้ทันทีส่วนค่าข้อมูลต้องดูแบบออฟไลน์โดยดาวน์โหลดไฟล์ซึ่งจะแสดงเป็นตารางโปรแกรม Microsoft Excel



รูปที่ 2.6 แผนภาพลำดับการทำงานของอุปกรณ์ตรวจวัดแก๊สและอุณหภูมิ

2.3 วิธีการทดลอง

2.3.1 ชุดคำสั่ง

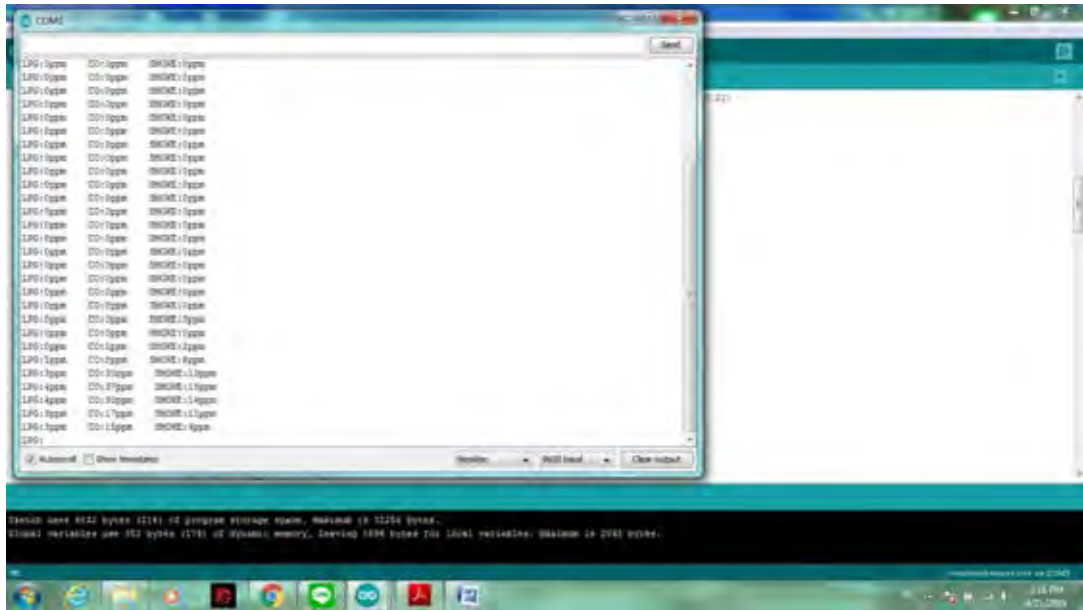
เขียนชุดคำสั่งเพื่อควบคุมการทำงานของอุปกรณ์ตรวจจับแก๊สและอุณหภูมิด้วยโปรแกรม Arduino IDE ซึ่งวิธีการใช้งานและชุดคำสั่งแสดงในภาคผนวก



รูปที่ 2.7 โปรแกรม Arduino IDE

2.3.2 ทดสอบการทำงานของอุปกรณ์ต่าง ๆ ทดสอบการทำงานของไมโครคอนโทรลเลอร์

ใช้สาย USB เชื่อมต่อไมโครคอนโทรลเลอร์กับคอมพิวเตอร์และอัปโหลดชุดคำสั่งขณะที่อัปโหลดชุดคำสั่งที่ไมโครคอนโทรลเลอร์จะมีไฟสีน้ำเงินกระพริบเมื่ออัปโหลดชุดคำสั่งเสร็จติดตามผลการตรวจวัดผ่าน Serial Monitor รูปที่ 2.8

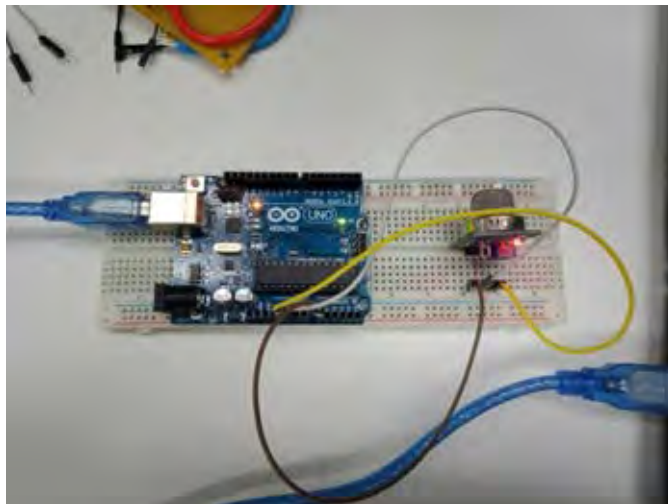


รูปที่ 2.8 ผลการตรวจวัดใน Serial Monitor

2.4 ทดสอบการทำงานของโมดูลทั้ง 2 ชนิด

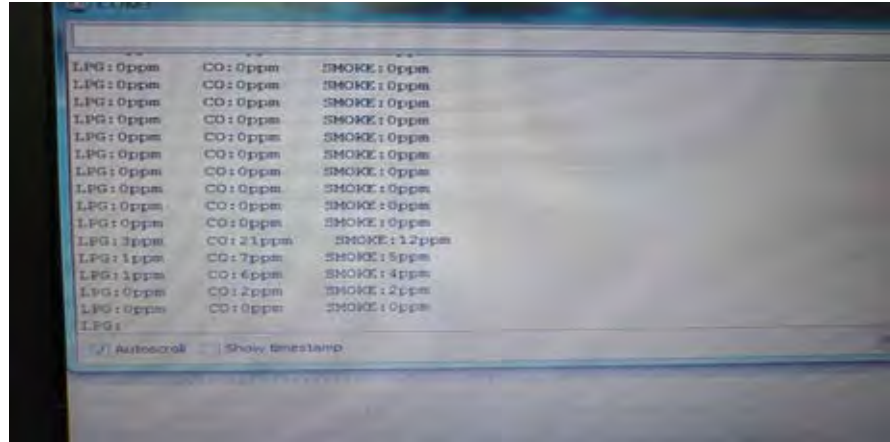
2.4.1 การทดสอบโมดูล MQ-2

การทดสอบโมดูล MQ-2 โดยใช้โปรแกรม Arduino โดยมีการจัดตั้งอุปกรณ์ดังรูปที่ 2.9 โดยทำการอัปโหลดข้อมูลลงบนบอร์ด Arduino



รูปที่ 2.9 การติดตั้งอุปกรณ์ทดสอบเซนเซอร์ MQ-2

และเมื่อทำการทดสอบอัดแก๊สที่เป็น LPG ในที่นี้คือแก๊สจากอุปกรณ์ติดไฟทำการปล่อยใส่ตัวเซนเซอร์ และควันทันรูปที่แสดงถึงการเกิดควันจะให้ผลทาง Serial Monitor ดังรูปที่ 2.10



รูปที่ 2.10 ผลแสดงจาก Serial Monitor

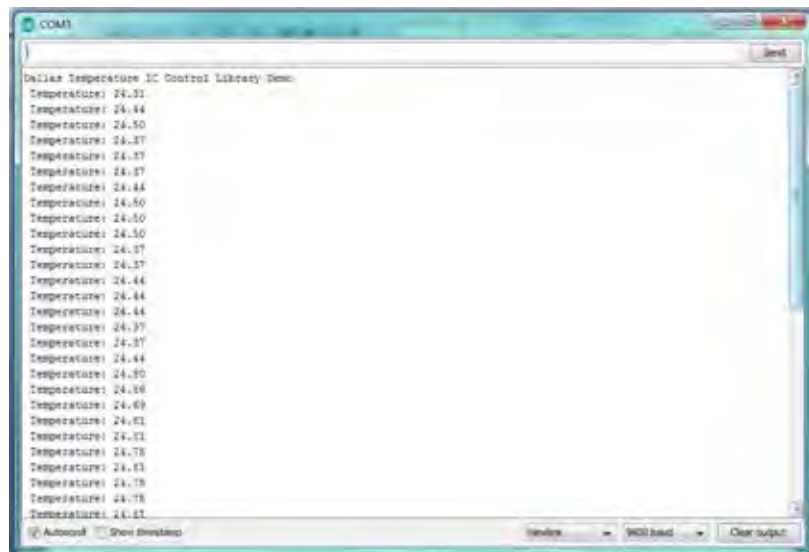
2.4.2 การทดสอบโมดูล DS18B20

การทดสอบโมดูล DS18B20 โดยใช้โปรแกรม Arduino โดยมีการจัดตั้งอุปกรณ์ดังรูปที่ 2.11 โดยทำการอัปโหลดข้อมูลลงบนบอร์ด Arduino



รูปที่ 2.11 การจัดตั้งอุปกรณ์เพื่อทดสอบเซนเซอร์ DS18B20

และเมื่อปล่อยไว้เป็นเวลา 30 นาทีเพื่อเช็คความเสถียรของการวัดอุณหภูมิจะได้ค่าจากจอ Serial Monitor ในโปรแกรม Arduino ดังรูปที่ 2.12



รูปที่ 2.12 การแสดงผลจาก Serial Monitor ที่ทดสอบ DS18B20

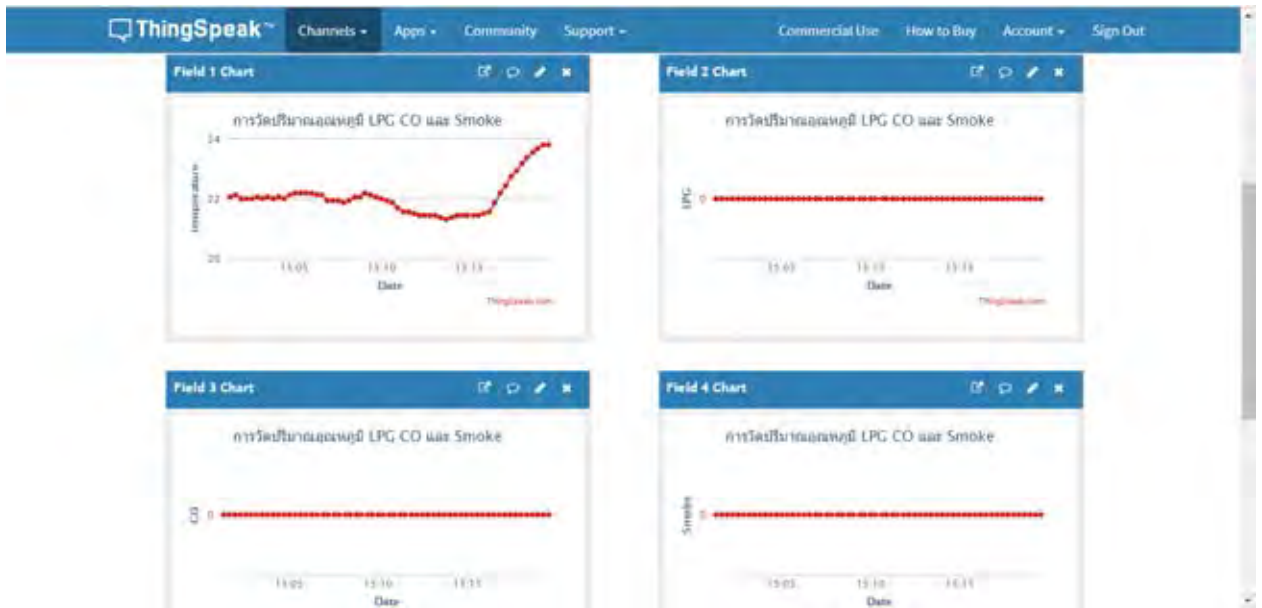
2.5 การทดสอบการส่งข้อมูลแบบไร้สาย

เริ่มด้วยโดยการลงทะเบียนที่ www.thingspeak.com ดังรูปที่ 2.13 โดยที่ตัวเซนเซอร์และคอมพิวเตอร์ต้องมีการเชื่อมต่อ Wi-Fi Hotspot จากโทรศัพท์มือถือที่ปล่อยสัญญาณอินเทอร์เน็ตได้



รูปที่ 2.13 หน้าเว็บ ThingSpeak

และเมื่อทำการวัดอุณหภูมิจากตัวเซนเซอร์เพื่อจะให้ข้อมูลมาแสดงที่หน้าเว็บจะได้ดังรูปที่ 2.14



รูปที่ 2.14 รูปการแสดงผลจากเว็บของ ThingSpeak

2.6 ทดสอบระยะทางของ Wi-Fi Hotspot

- 2.6.1 วางอุปกรณ์กระจายสัญญาณเครือข่ายไร้สายไว้หนึ่งๆแล้วเลื่อนอุปกรณ์ตรวจวัดปริมาณแก๊สและอุณหภูมิห่างออกไปเรื่อย ๆ
- 2.6.2 ขณะนั้นก็ดูการเชื่อมต่อของอุปกรณ์กระจายสัญญาณเครือข่ายไร้สายกับอุปกรณ์ตรวจวัดปริมาณแก๊สและอุณหภูมิและดูการส่งข้อมูลไปยังฐานข้อมูลออนไลน์
- 2.6.3 ทำการวัดระยะทางที่ไกลที่สุดที่อุปกรณ์ตรวจวัดปริมาณแก๊สและอุณหภูมิสามารถเชื่อมต่อ Wi-Fi Hotspot ได้

2.7 การนำอุปกรณ์ตรวจวัดไปติดตั้งที่โดรน

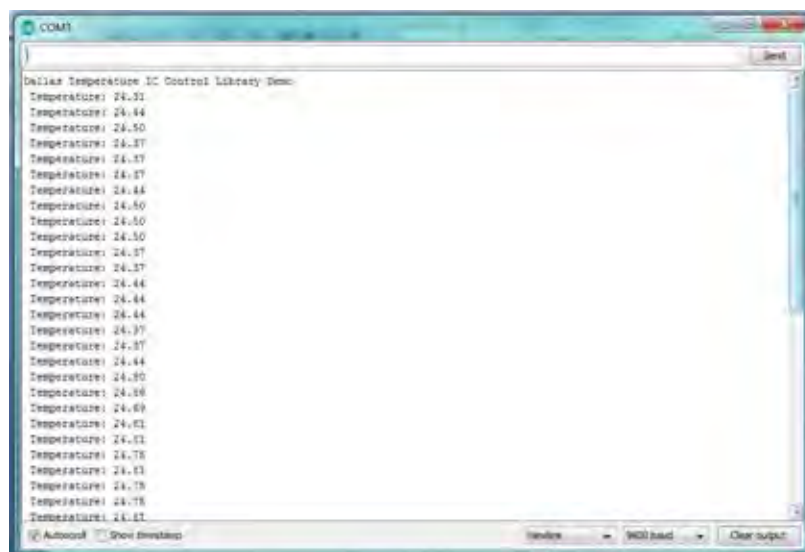
- 2.7.1 เมื่ออุปกรณ์ตรวจวัดแก๊สและอุณหภูมิมีความเสถียรและสามารถตรวจวัดได้จริงแล้วนำอุปกรณ์ตรวจวัดแก๊สและอุณหภูมิไปติดตั้งบนโดรน
- 2.7.2 นำโดรนพร้อมอุปกรณ์ตรวจวัดแก๊สและอุณหภูมิไปบินเพื่อตรวจวัดแก๊สและอุณหภูมิในพื้นที่ทำการตรวจวัด

บทที่ 3

ผลการทดลองและอภิปรายผลการทดลอง

3.1 ทดสอบการทำงานของอุปกรณ์ต่าง ๆ

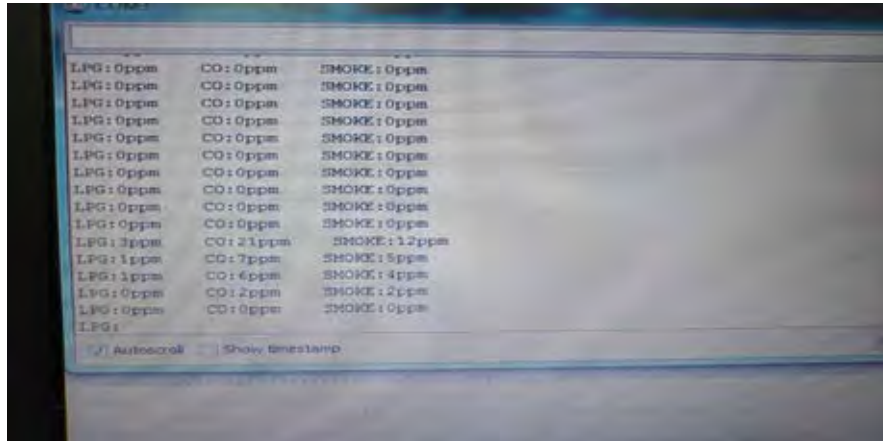
จากการทดลองวัดอุณหภูมิโดยใช้ตัวเซนเซอร์ DS18B20 เพื่อทำการวัดจะได้ผลออกมาในหน่วยของ เซลเซียส ซึ่งแสดงออกมาจากจอ Serial monitor ได้ดังรูปที่ 3.1



รูปที่ 3.1 ภาพแสดง serial monitor ของ DS18B20

จากรูปที่ 3.1 จะเห็นได้ว่าค่าที่ได้มีการส่งผลแบบเรียลไทม์และต่อเนื่อง ซึ่งวัดจากในห้องทดลองที่มีการตั้งอุณหภูมิไว้ที่ 25 องศา จะเห็นได้ว่าค่าที่แสดงออกมาอย่างต่อเนื่องนั้นมีความคาดเคลื่อนของอุปกรณ์ที่ทำการวัดอยู่ที่ไม่เกิน +1 หรือ -1 องศา ซึ่งเป็นผลที่พอเชื่อถือได้

ต่อมาได้ทำการวัดโดยใช้โมดูล MQ-2 ซึ่งสามารถวัดความเข้มข้นของคาร์บอนมอนอกไซด์ แอลพีจีหรือแก๊สที่สามารถติดไฟได้ และควันได้ในหน่วยของ ppm โดยใช้ควันจากรูปเป็นตัวทดสอบควันและคาร์บอนมอนอกไซด์ และใช้แก๊สในไฟแช็กเป็นตัวทดสอบแก๊สแอลพีจีที่ได้ทำการพ่นเข้าสู่ตัวเซนเซอร์ซึ่งแสดงออกมาจากจอ serial monitor ดังรูปที่ 3.2

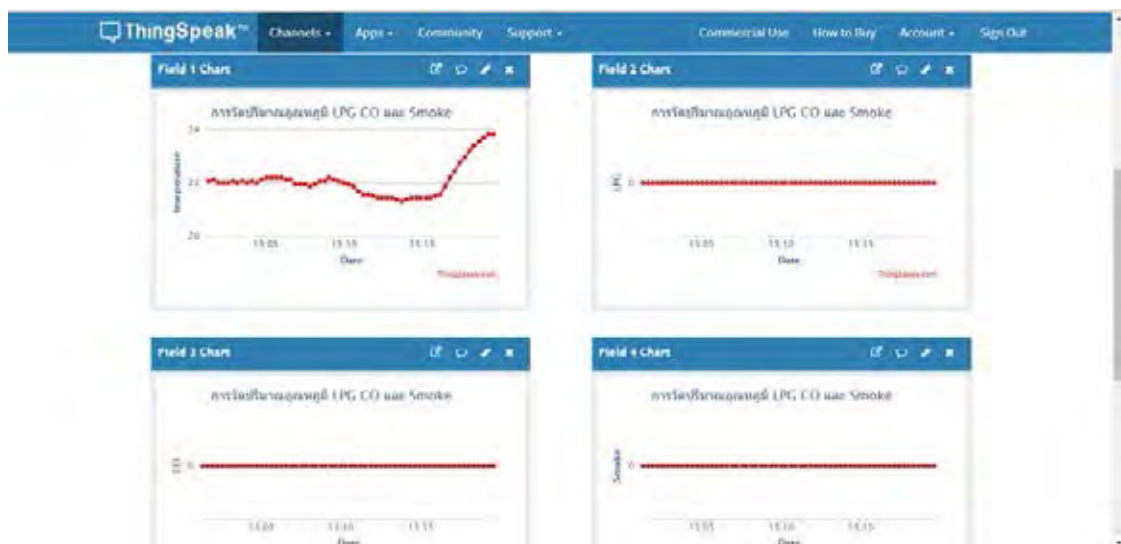


รูปที่ 3.2 ภาพแสดง Serial monitor ของ MQ-2 ที่วัดค่า CO,LPG และ คิว

จากรูปที่ 3.2 จะเห็นได้ว่าเมื่อตอนที่ยังไม่มีควันของรูปหรือแก๊สจากที่ติดไฟค่าที่ได้นั้นจะมีค่าเท่ากับ 0 และเมื่อมีควันจากรูปหรือแก๊สจากไฟแช็กก็จะมีค่าออกมาแบบเรียลไทม์และต่อเนื่องในหน่วยของ ppm

3.2 การทดสอบการส่งข้อมูลแบบไร้สาย

การส่งข้อมูลที่วัดได้นั้นจะส่งข้อมูลแบบเรียลไทม์และต่อเนื่องไปที่ www.thingspeak.com ซึ่งได้ผลดังรูปที่ 3.3



รูปที่ 3.3 ภาพแสดงการส่งข้อมูลที่ตรวจวัดแบบไร้สาย

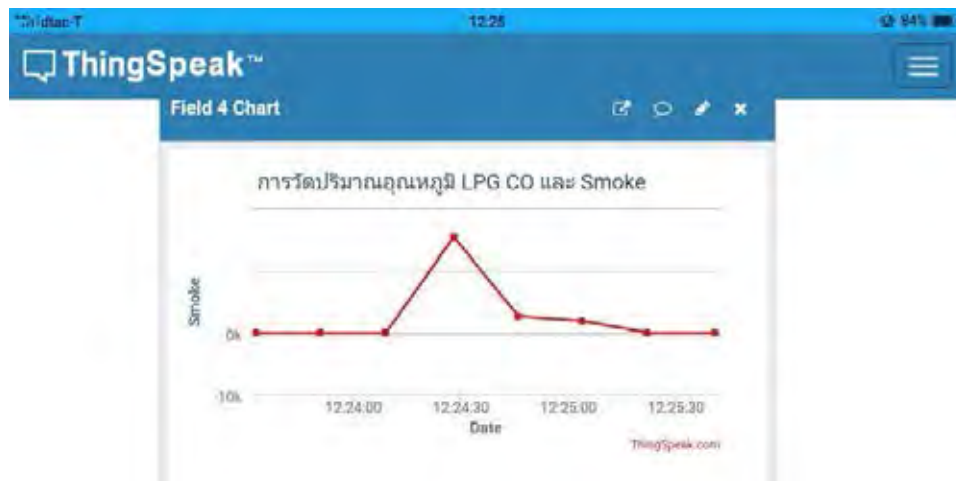
จากนี้ไม่เพียงแต่สามารถส่งข้อมูลไปยังเว็บไซต์ที่เป็นคอมพิวเตอร์เพียงอย่างเดียวแต่ยังสามารถดูผลได้จากโทรศัพท์ที่เป็นสมาร์ทโฟนอีกด้วยซึ่งแสดงดังรูปที่ 3.4-3.7



รูปที่ 3.4 ผลการตรวจวัดอุณหภูมิที่ดูจากสมาร์ทโฟน



รูปที่ 3.5 ผลการตรวจวัด LPG ที่ดูจากสมาร์ทโฟน



รูปที่ 3.6 ผลการตรวจควันที่ดูจากสมาร์ทโฟน



รูปที่ 3.7 ผลการตรวจวัด CO ที่ดูจากสมาร์ทโฟน

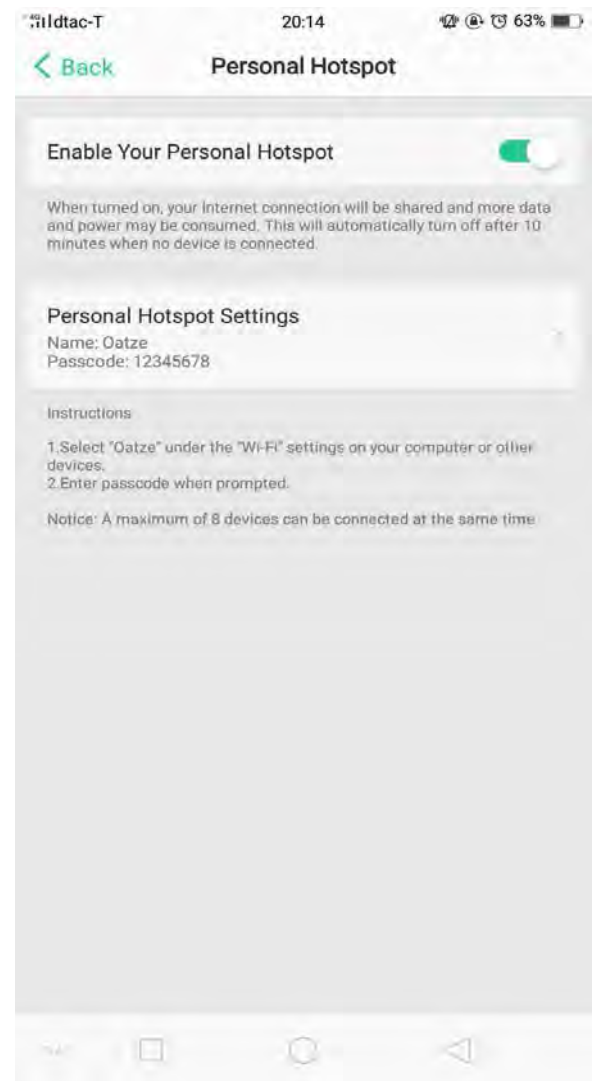
3.3 ทดสอบระยะทางของ Wi-Fi Hotspot

การทดสอบระยะของสัญญาณที่สามารถรับได้โดยมีการปล่อยสัญญาณอินเทอร์เน็ตหรือ hotspot Wi-Fi จากมือถือนึงถึงตัวเซิร์ฟเวอร์ที่ทำการเชื่อมต่อกันได้โดยการที่ตั้งตัวเซิร์ฟเวอร์ไว้แล้วทำการถือโทรศัพท์มือถือที่ปล่อยสัญญาณแล้วเดินออกจากตัวของเซิร์ฟเวอร์ไปเรื่อย ๆ จนสัญญาณที่เชื่อมต่อของอุปกรณ์หายไปแล้วทำ

การวัดระยะทางที่สัญญาณที่สามารถเชื่อมต่อกันได้จนถึงระยะทางที่สัญญาณขาดหายไปอยู่ที่ 24 เมตร โดยปกติแล้ว Wi-Fi hotspot ที่ปล่อยจากมือถือจะอยู่ประมาณ 15-30 เมตรขึ้นอยู่กับความสามารถในการปล่อยสัญญาณของโทรศัพท์ด้วย



(a)



(b)

รูปที่ 3.8 แสดงตอนที่มีการเชื่อมต่อสัญญาณ (a) และเมื่อสัญญาณขาดการเชื่อมต่อ (b)

3.4 การนำอุปกรณ์ตรวจวัดไปติดตั้งที่โดรน

การนำอุปกรณ์มาติดบนโดรนนั้นสามารถทำได้โดยการนำอุปกรณ์มาติดกับตัวโดรนด้วยเทปหนามเตย ดังรูปที่ 3.9



รูปที่ 3.9 การติดตั้งอุปกรณ์ตรวจวัดบนโดรน

จากรูปที่ 3.9 การติดตั้งอุปกรณ์บนตัวโดรนนั้นมีประเด็นที่ต้องพิจารณาอยู่ 3 ประเด็นคือ

1. อิทธิพลของแรงลมจากใบพัดของตัวโดรนทั้ง 4 ข้างในขณะทำการบินจะส่งผลต่อค่าความไวสัญญาณที่วัดได้ ทำให้ค่าสัญญาณที่วัดได้ขณะที่โดรนจอดนิ่งกับขณะที่บินอยู่มีความแตกต่างกันมาก ด้วยเหตุนี้จึงต้องทำการทดสอบสัญญาณในพื้นที่จริงขณะทำการบินอยู่ที่ความสูงที่เหมาะสมและเปรียบเทียบกับข้อมูลจากแหล่งข้อมูลของหน่วยงานราชการเท่านั้นแต่ด้วยข้อจำกัดด้านเวลา และระเบียบการขออนุญาตบินโดรนในพื้นที่สาธารณะส่งผลให้ไม่สามารถทำการแก้ไขปรับปรุงและทดสอบการทำงานของอุปกรณ์ขณะที่โดรนบินอยู่ได้ทันตามที่วางแผนไว้

2. ระยะเวลาในการบินโดรนโดยปกติที่ไม่อุปกรณ์มาติดบนโดรนนั้นสามารถบินอยู่ในอากาศได้ประมาณ 15-20 นาที ขึ้นกับค่าความจุของแบตเตอรี่ที่ใช้เมื่อนำอุปกรณ์ตรวจวัดมาติดตั้งบนโดรน จะส่งผลให้น้ำหนักของโดรนเมื่อรวมกับอุปกรณ์ก็มากขึ้น จะจำเป็นต้องใช้พลังงานในการยกตัวขณะบินมากขึ้นด้วย จึงทำให้ระยะเวลาในการบินโดรนนั้นจะน้อยลงหากใช้แบตเตอรี่ที่มีค่าความจุเท่าเดิม แนวทางในการปรับปรุงที่จะทำให้ระยะเวลาในการบินโดรนบินอยู่ในอากาศได้นานยิ่งขึ้นทำได้โดยการปรับขนาดของอุปกรณ์ให้มีขนาดเล็กกลงเพื่อลดน้ำหนักหรือการเปลี่ยนแหล่งจ่ายพลังงานของอุปกรณ์ตรวจวัดมาใช้แหล่งจ่ายพลังงานเดียวกับตัวโดรน

และเพิ่มจำนวนปริมาณความจุของแบตเตอรี่ให้มากขึ้นโดยที่ขนาดไม่เพิ่มขึ้นตามไปด้วยโดยการดัดแปลงได้ ดังรูปที่ 3.8



รูปที่ 3.10 แผนภาพแสดงการดัดแปลงวงจร

จากรูปที่ 3.10 จะเห็นได้ว่าความต่างศักย์ (V) ของตัวอุปกรณ์นั้นมีค่าที่แตกต่างกัน จึงจำเป็นต้องมีตัว IC regulator 7805 ที่แปลงความต่างศักย์ของแบตเตอรี่ของโดรนที่มีค่าเท่ากับ 8 V ให้เปลี่ยนความต่างศักย์เป็น 5 V เพื่อให้มีความเหมาะสมกับตัวเซนเซอร์

3. ระยะทางของสัญญาณ Wi-Fi ของโทรศัพท์มือถือที่ติดกับอุปกรณ์ตรวจวัดที่ติดตั้งบนโดรน จากข้อมูลที่ทำการศึกษาพบว่าระยะทางที่โทรศัพท์มือถือสามารถปล่อยสัญญาณ Wi-Fi hotspot ได้นั้นอยู่ที่รัศมี 24 เมตรเท่านั้น แตกต่างจากค่าที่เคยทดสอบจากโครงการเมื่อปีการศึกษา 2560 ที่วัดระยะได้ 95 เมตร เมื่อเวลาบินโดรนขึ้นไปในอากาศเพื่อทำการตรวจวัดนั้นไม่สามารถตรวจวัดบริเวณที่ไกลออกไปได้ เบื้องต้นสันนิษฐานว่า รัศมีการเชื่อมต่อนั้นขึ้นกับ ยี่ห้อ และรุ่นของโทรศัพท์มือถือที่ใช้ในการทดสอบด้วย ในอนาคตอาจแก้ปัญหาด้วยการเปลี่ยนวิธีเชื่อมต่อจากเดิม คือ Wi-Fi hotspot เป็นการเชื่อมต่อโดยใช้ SIM card ติดตั้งบนตัวอุปกรณ์ และส่งผ่านข้อมูลผ่านทาง Cellular Network แทน

บทที่ 4

สรุปผลการทดลอง

4.1 ทดสอบการทำงานของอุปกรณ์ต่างๆและ การส่งข้อมูลแบบไร้สาย

อุปกรณ์ตรวจวัดที่สร้างขึ้นสามารถใช้งานได้และมีความเสถียรที่ดีสามารถใช้ในการติดตาม gas ชนิดต่าง ๆ รวมทั้งค่าอุณหภูมิ และข้อมูลแบบไร้สายเพื่อแสดงผลบนโทรศัพท์มือถือได้แบบเรียลไทม์

4.2 ทดสอบระยะทางของ Wi-Fi Hotspot

จากการทดสอบระยะทางของ Wi-Fi Hotspot พบว่าในพื้นที่แนวราบและไม่มีสิ่งกีดขวางอุปกรณ์ตรวจวัดปริมาณแก๊สและอุณหภูมิสามารถเชื่อมต่อกับ Wi-Fi Hotspot ได้ระยะทางที่มากที่สุดคือ 24 เมตร

4.3 การนำอุปกรณ์ตรวจวัดไปติดตั้งที่โดรน

เมื่อพยายามติดตั้งอุปกรณ์เข้ากับตัวโดรนพบว่าทำได้ค่อนข้างลำบากเนื่องจากอุปกรณ์มีขนาดใหญ่ เกะกะเกินไปที่จะประกอบกับแบตเตอรี่ของอุปกรณ์ที่มีน้ำหนักค่อนข้างมากทำให้เสียสมดุล จนส่งผลให้ไม่สามารถบินขึ้นได้อย่างมีประสิทธิภาพนอกจากนี้พบว่าระยะเชื่อมต่อแบบไร้สายของ อุปกรณ์มีระยะทำการเพียง 24 เมตรซึ่งน้อยกว่าที่คาดการณ์เอาไว้อย่างมาก

ด้วยข้อจำกัดด้านเวลา และการขออนุญาตบินโดรนส่งผลให้ไม่สามารถทำการแก้ไขปรับปรุงและทดสอบการทำงานของอุปกรณ์ขณะที่โดรนบินอยู่ได้ทันตามที่วางแผนไว้

4.4 แนวทางแก้ไขสำหรับงานในอนาคต

ออกแบบวงจรให้มีขนาดเล็กลง เพื่อให้ติดตั้งบนโดรนได้อย่างเหมาะสม และออกแบบแบตเตอรี่ใหม่ เพื่อให้แบตเตอรี่นี้ทำหน้าที่จ่ายพลังงานให้กับทั้งโดรนและอุปกรณ์พร้อมๆ กัน จะได้เป็นการลดขนาดและน้ำหนัก เนื่องจากโดรนใช้ไฟฟ้า 8 V แต่ตัวเซนเซอร์ใช้ไฟฟ้าเท่ากับ 5 V จึงต้องทำขั้วแยกต่อกับ IC regulator เพื่อแปลงความต่างศักย์ของแบตเตอรี่จาก 8 V ให้เหลือ 5 V จึงจะสามารถจ่ายพลังงานให้กับอุปกรณ์ได้อย่างเหมาะสม และเปลี่ยนวิธีเชื่อมต่อจากเดิม คือ Wi-Fi hotspot เป็นการเชื่อมต่อโดยใช้ SIM card ติดตั้งบนตัวอุปกรณ์ และส่งผ่านข้อมูลผ่านทาง Cellular Network แทน

เอกสารอ้างอิง

1. Ambeth, K. V. D. Human security from death defying gases using an intelligent sensor system. *Sens. Biosensing. Res* **2016**, *7*, 107-114.
2. Ibrahim, D. A new approach for teaching microcontroller courses toundergraduate students. *Procedia Soc. Behav. Sci.* **2014**, *131*, 411 – 414.
3. Buribayeva, G.; Miyachi, T.; Yeshmukhametov, A.; Mikami, Y. An autonomous emergency warning system based on cloud servers and SNS. *Procedia Comput. Sci.* **2015**, *60*, 722-729.
4. Gloria, A.; Cercas, F.; Souto, N. Design and implementation of an IoT gateway to create smart environments. *Procedia Comput. Sci.* **2017**, *109*, 568-575.
5. Suarez, J. I.; Arroyo, P.; Lozano, J.; Herrero, J. L.; Padilla, M. Bluetooth gas sensing module combined with smartphones for air quality monitoring. *Chemosphere* **2018**, *205*, 618-626.
6. [.http://www.votejoel.com/drone](http://www.votejoel.com/drone)
7. [.http://thaimulticopter.blogspot.com/2013/07/blog-post.html](http://thaimulticopter.blogspot.com/2013/07/blog-post.html)
8. [.http://www.sbt.ac.th/new/sites/default/files/TNP_Unit_1.pdf](http://www.sbt.ac.th/new/sites/default/files/TNP_Unit_1.pdf).https://mcu56.learninginventions.org/?page_id=264
9. [.http://www.ayarafun.com/2015/08/introduction-arduino-esp8266-nodemcu](http://www.ayarafun.com/2015/08/introduction-arduino-esp8266-nodemcu)
10. <https://www.thaieasyelec.com/products/retired-product/nodemcu-development-kitdetail.html>
11. <http://sat2you.com/web/2017/01/31/nodemcu-กับ-iot-ตอนที่-1-nodemcu-คืออะไร>
12. [.https://inex.co.th/shop/nodemcu-v2-0.html](https://inex.co.th/shop/nodemcu-v2-0.html)
13. <https://www.amazon.in/VEEROBOT-Butane-Hydrogen-Sensor-Detector/dp/B01307SNHG>
14. <https://www.makerlab-electronics.com/product/mq7-carbon-monoxide-gas-sensor-module/>

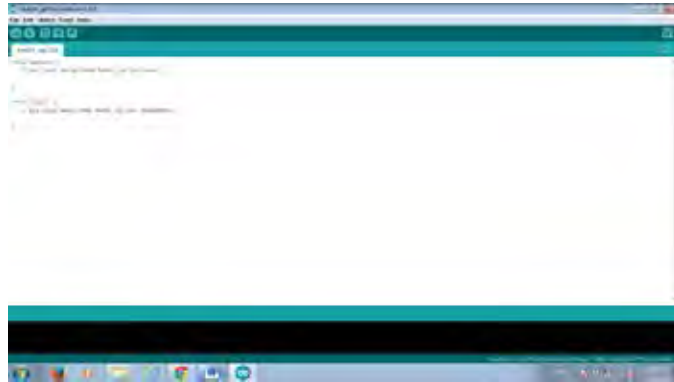
15. <http://www.adaline.co.th/product/369/temperature-sensor-ds18b20>
16. <https://www.arduinoall.com/product/1741/micro-usb-cable-50cm-สาย-micro-usb-ยาว-50-เซนติเมตร>
17. https://www.makefab.com/index.php?route=product/product&product_id=116
18. https://www.makefab.com/index.php?route=product/product&product_id=115
19. https://www.makefab.com/index.php?route=product/product&product_id=118
20. <https://www.kaidee.com/product-121011963/>
21. <https://www.ecig-vapo.com/ultrafire-battery-18650-3000mah-37v-li-ion-with-button-top-p-1202.html>
22. <http://i2celectronica.com/protoboards-placas-pcb/98-protoboard-400-puntos.html>
23. <http://rees52.com/288-multi-purpose-pcb-board-30-x-15-cm.html>
24. https://www.banggood.com/One-Side-Copper-Clad-70x100x1_5mm-Single-PCB-Board-Glass-Fiber-p-950900.html?cur_warehouse=CN
25. <http://commandronestore.com/products/br6222.php>
26. <https://www.sparkfun.com/products/8375>
27. https://www.banggood.com/50Pcs-100uF-50V-105C-Radial-Electrolytic-Capacitor-8x12mm-p-1129091.html?cur_warehouse=CN
28. <https://www.amazon.in/Versatile-7805-5v-voltage-regulator/dp/B00KHK1C6Y>
29. <https://www.taydaelectronics.com/led-5mm-red.html>
30. <http://www.naturalhealthsupply.com/assets/images/on-off-switch.jpg>
31. <https://www.ebay.com/itm/ABS-Plastic-Project-Box-3-L-x-1-96-W-x-1-06-H-inch-Black-/272034899736>
32. <https://www.ioxhop.com/article/13/esp8266-ตอนที่-1-รู้จักกับ-esp-และรุ่นที่นิยมใช้งาน>
33. <https://www.ba-na-na.net/tag/firmware-คืออะไร>
34. <http://itnews4u.com/What-is-Firmware.html>
35. <https://www.pololu.com/file/0J309/MQ2.pdf>

ภาคผนวก

ก. วิธีการใช้งานโปรแกรม Arduino IDE

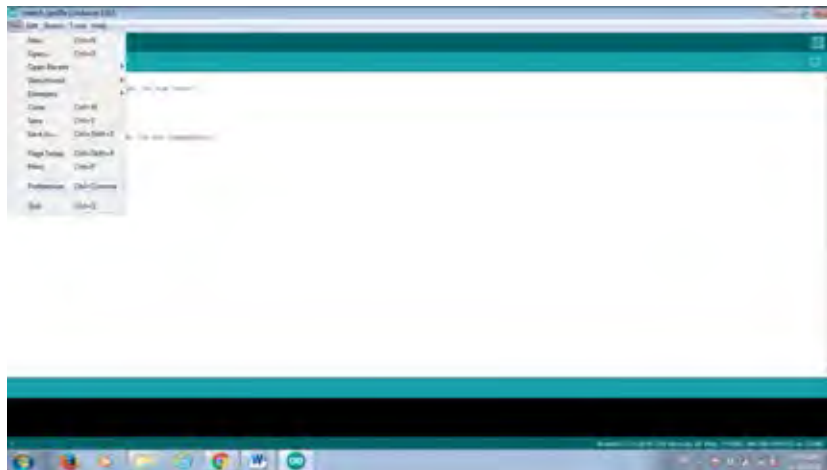
I. ดาวน์โหลดและติดตั้งโปรแกรม Arduino IDE จาก <https://www.arduino.cc/en/Main/Software>

II. เมื่อเข้าโปรแกรม Arduino IDE หน้าต่างโปรแกรม Arduino IDE จะปรากฏขึ้น (รูปที่ 1 ก.)



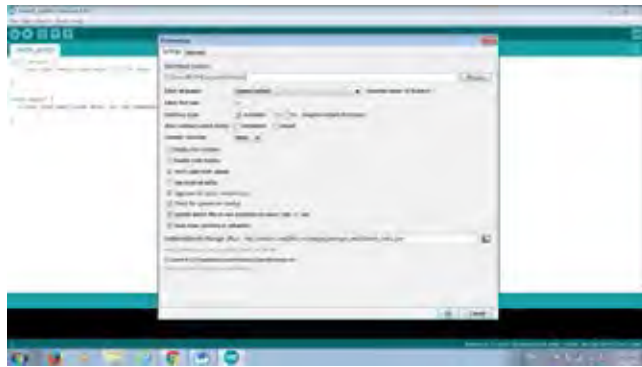
รูปที่ 1 ก. Arduino IDE

III. เลือกเมนู File ⇒ Preferences เพื่อติดตั้งไมโครคอนโทรลเลอร์ NodeMCU ESP8266 (รูปที่ 2 ก.)



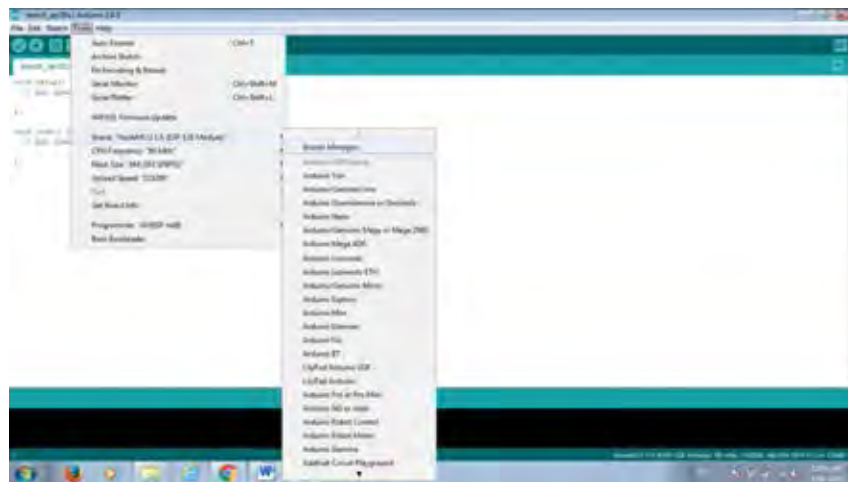
รูปที่ 2 ก. การติดตั้งไมโครคอนโทรลเลอร์

IV. เพิ่ม http://arduino.esp8266.com/stable/package_esp8266com_index.json ลงในช่อง Additional Boards Manager URLs แล้วกด OK (รูปที่ 3 ก.)



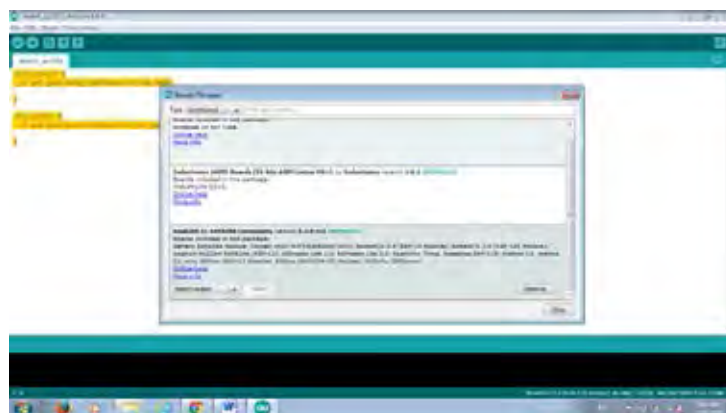
รูปที่ 3 ก. การดาวน์โหลดข้อมูลไมโครคอนโทรลเลอร์

V. เลือกเมนู Tools ⇒ Board ⇒ Board Manager (รูปที่ 4 ก.)



รูปที่ 4 ก. การเข้าเมนูเพื่อดาวน์โหลดข้อมูลไมโครคอนโทรลเลอร์

VI. เลือก Type เป็น Contributed จากนั้นเลือก ESP8266 by ESP8266 Community (รูปที่ 5 ก.)



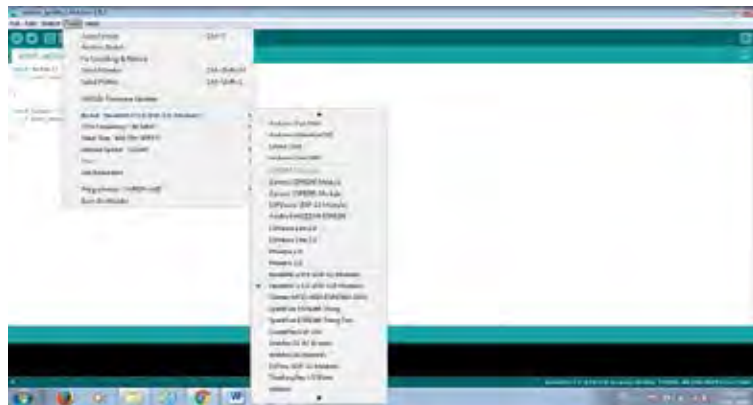
รูปที่ 5 ก. การเลือกชนิดเพื่อดาวน์โหลดข้อมูลไมโครคอนโทรลเลอร์

VII. ปิดโปรแกรม Arduino IDE เพื่อให้โปรแกรมอัปเดตข้อมูลบอร์ดแล้วเปิดโปรแกรม Arduino IDE ใหม่

VIII. ใช้สาย USB เชื่อมต่อไมโครคอนโทรลเลอร์ NodeMCU ESP8266 กับคอมพิวเตอร์

IX. เลือกเมนู Tools ⇒ Board ⇒ ... (รูปที่ 6 ก.)

- ไมโครคอนโทรลเลอร์ ESP8266 NodeMCU Devkit V0.9 เลือก NodeMCU 0.9 (ESP-12E Module)
- ไมโครคอนโทรลเลอร์ ESP8266 NodeMCU Devkit V2.0 เลือก NodeMCU 1.0 (ESP-12E Module)



รูปที่ 6 ก. การเลือกชนิดไมโครคอนโทรลเลอร์

X. เขียนชุดคำสั่ง (ภาคผนวก) แล้วกดปุ่ม Compile เพื่อตรวจสอบความถูกต้องของโค้ดหากไม่ผิดพลาดจะปรากฏ Done compiling (รูปที่ 7 ก.)



รูปที่ 7 ก. การตรวจสอบความถูกต้องของชุดคำสั่ง

XI. อัปโหลดชุดคำสั่ง (ภาคผนวก) แล้วกดปุ่ม Upload เพื่อนำชุดคำสั่งเข้าสู่ไมโครคอนโทรลเลอร์ (รูปที่ 8 ก.)



รูปที่ 8 ก. การอัปโหลดชุดคำสั่ง

XII. ทดสอบการทำงานของอุปกรณ์โดยเปิดสวิตช์ของอุปกรณ์ตรวจวัดแก๊สและอุณหภูมิโดยดูผลการตรวจวัดผ่านฐานข้อมูลออนไลน์ www.thingspeak.com

ข. ชุดคำสั่งที่ใช้ในอุปกรณ์ตรวจวัดปริมาณแก๊ส

1 ชุดคำสั่งสำหรับทดสอบ DS18B20

```
#include <OneWire.h>
```

```
#include <DallasTemperature.h>
```

```
#define ONE_WIRE_BUS 5
```

```
OneWire oneWire(ONE_WIRE_BUS);
```

```
DallasTemperature sensors(&oneWire);
```

```
float Celcius=0;
```

```
float Fahrenheit=0;
```

```
void setup(void)

{

Serial.begin(9600);

sensors.begin();

}

void loop(void)

{

sensors.requestTemperatures();

Celcius=sensors.getTempCByIndex(0);

    Fahrenheit=sensors.toFahrenheit(Celcius);

Serial.print(" Temp ");

Serial.print(Celcius);

Serial.print(" F ");

Serial.println(Fahrenheit);

delay(1000);

}
```

2. ชุดคำสั่งทดสอบ MQ-2 ตรวจวัด CO, LPG, คว้น

```
#define MQ_PIN (0)
```

```
#define RL_VALUE (5)
```

```
#define RO_CLEAN_AIR_FACTOR (9.83)
```

```
/******Software Related Macros*****/
```

```
#define CALIBRATION_SAMPLE_TIMES (50)
```

```
#define CALIBRATION_SAMPLE_INTERVAL (500)
```

```
#define READ_SAMPLE_INTERVAL (50)
```

```
#define READ_SAMPLE_TIMES (5)
```

```
#include <LiquidCrystal.h>
```

```
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
```

```
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

```
/******Application Related Macros*****/
```

```
#define GAS_LPG (0)
```

```
#define GAS_CO (1)
```

```
#define GAS_SMOKE (2)
```

```
/******Globals*****/
```

```
floatLPGCurve[3] = {2.3,0.21,-0.47};

floatCOCurve[3] = {2.3,0.72,-0.34};

floatSmokeCurve[3] = {2.3,0.53,-0.44};

float      Ro      = 10;

void setup()

{

Serial.begin(9600);

Serial.print("Calibrating...\n");

    Ro = MQCalibration(MQ_PIN);

lcd.begin(16, 2);

Serial.print("Calibration is done...\n");

Serial.print("Ro=");

Serial.print(Ro);

Serial.print("kohm");

Serial.print("\n");

lcd.print("Calibration is done...\n");

lcd.print("Ro=");

lcd.print(Ro);

lcd.print("kohm");
```

```
lcd.print("\n");

}

void loop()

{

Serial.print("LPG:");

Serial.print(MQGetGasPercentage(MQRead(MQ_PIN)/Ro,GAS_LPG) );

Serial.print( "ppm" );

Serial.print("  ");

Serial.print("CO:");

Serial.print(MQGetGasPercentage(MQRead(MQ_PIN)/Ro,GAS_CO) );

Serial.print( "ppm" );

Serial.print("  ");

Serial.print("SMOKE:");

Serial.print(MQGetGasPercentage(MQRead(MQ_PIN)/Ro,GAS_SMOKE) );

Serial.print( "ppm" );

Serial.print("\n");

lcd.setCursor(0, 0);

lcd.print("LPG:");

lcd.print(MQGetGasPercentage(MQRead(MQ_PIN)/Ro,GAS_LPG) );
```

```
//lcd.print( "ppm" );

lcd.print("  ");

lcd.setCursor(9, 0);

lcd.print("CO:");

lcd.print(MQGetGasPercentage(MQRead(MQ_PIN)/Ro,GAS_CO) );

//lcd.print( "ppm" );

lcd.print("  ");

lcd.setCursor(0, 1);

lcd.print("SMOKE:");

lcd.print(MQGetGasPercentage(MQRead(MQ_PIN)/Ro,GAS_SMOKE) );

//lcd.print( "ppm" );

lcd.print("  ");

delay(200);

}

floatMQResistanceCalculation(intraw_adc)

{

return ( ((float)RL_VALUE*(1023-raw_adc)/raw_adc));

}

floatMQCalibration(intmq_pin)
```

```
{  
  
  inti;  
  
  floatval=0;  
  
  for (i=0;i<CALIBARAION_SAMPLE_TIMES;i++) {  
  
    val += MQResistanceCalculation(analogRead(mq_pin));  
  
    delay(CALIBRATION_SAMPLE_INTERVAL);  
  
  }  
  
  val = val/CALIBARAION_SAMPLE_TIMES;  
  
  
  val = val/RO_CLEAN_AIR_FACTOR;  
  
  returnval;  
  
}  
  
floatMQRead(intmq_pin)  
  
{  
  
  inti;  
  
  floatrs=0;  
  
  for (i=0;i<READ_SAMPLE_TIMES;i++) {  
  
    rs += MQResistanceCalculation(analogRead(mq_pin));  
  
    delay(READ_SAMPLE_INTERVAL);  
  
  }  
  
}
```

```

    }

    rs = rs/READ_SAMPLE_TIMES;

    returnrs;

}

intMQGetGasPercentage(float rs_ro_ratio, intgas_id)

{

if ( gas_id == GAS_LPG ) {

returnMQGetPercentage(rs_ro_ratio,LPGCurve);

} else if ( gas_id == GAS_CO ) {

returnMQGetPercentage(rs_ro_ratio,COCurve);

} else if ( gas_id == GAS_SMOKE ) {

returnMQGetPercentage(rs_ro_ratio,SmokeCurve);

}

return 0;

}

intMQGetPercentage(float rs_ro_ratio, float *pcurve)

{

return (pow(10,(((log(rs_ro_ratio)-pcurve[1])/pcurve[2]) + pcurve[0])));

}

```


สำหรับไมโครคอนโทรลเลอร์ ESP8266 NodeMCU Devkit V2.0 (ตรวจวัดอุณหภูมิ, ปริมาณแก๊สแอลพีจี, คาร์บอนมอนอกไซด์ และ ควีน)

```
#include <ESP8266WiFi.h>
```

```
#include <OneWire.h>
```

```
#include <DallasTemperature.h>
```

```
#define myPeriodic 15
```

```
#define ONE_WIRE_BUS D1
```

```
#define MQ_PIN (A0)
```

```
#define RL_VALUE (5)
```

```
#define RO_CLEAN_AIR_FACTOR (9.83)
```

```
#define CALIBRATION_SAMPLE_TIMES (50)
```

```
#define CALIBRATION_SAMPLE_INTERVAL (500)
```

```
#define READ_SAMPLE_INTERVAL (50)
```

```
#define READ_SAMPLE_TIMES (5)
```

```
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
```

```
#define    GAS_LPG            (0)

#define    GAS_CO            (1)

#define    GAS_SMOKE        (2)

floatLPGCurve[3] = {2.3,0.21,-0.47};

floatCOCurve[3] = {2.3,0.72,-0.34};

floatSmokeCurve[3] = {2.3,0.53,-0.44};

float    Ro    = 10;

OneWireoneWire(ONE_WIRE_BUS);

DallasTemperatureDS18B20(&oneWire);

floatprevTemp = 0;

const char* server = "api.thingspeak.com";

String apiKey ="0M93X021NVQ79EPQ";

const char* MY_SSID = "Oatze";

const char* MY_PWD = "12345678";

int sent = 0;
```

```
void setup()

{

Serial.begin(9600);

connectWifi();

Serial.print("Calibrating...\n");

    Ro = MQCalibration(MQ_PIN);

Serial.print("Calibration is done...\n");

Serial.print("Ro=");

Serial.print(Ro);

Serial.print("kohm");

Serial.print("\n");

}

void loop()

{

float temp;

DS18B20.requestTemperatures();

temp = DS18B20.getTempCByIndex(0);
```

```
Serial.print(String(sent)+" Temperature: ");
```

```
Serial.println(temp);
```

```
float LPG = MQGetGasPercentage(MQRead(MQ_PIN)/Ro,GAS_LPG);
```

```
float CO = MQGetGasPercentage(MQRead(MQ_PIN)/Ro,GAS_CO);
```

```
float SMOKE = MQGetGasPercentage(MQRead(MQ_PIN)/Ro,GAS_SMOKE);
```

```
Serial.print("LPG:");
```

```
Serial.print(LPG);
```

```
Serial.print( "ppm" );
```

```
Serial.print("  ");
```

```
Serial.print("CO:");
```

```
Serial.print(CO);
```

```
Serial.print( "ppm" );
```

```
Serial.print("  ");
```

```
Serial.print("SMOKE:");
```

```
Serial.print(SMOKE);
```

```
Serial.print( "ppm" );
```

```
Serial.print("\n");
```

```
delay(1000);
```

```
A_sendDataToTS(temp,LPG,CO,SMOKE);
```

```
int count = myPeriodic;
```

```
while(count--)
```

```
delay(1000);
```

```
}
```

```
voidconnectWifi()
```

```
{
```

```
Serial.print("Connecting to "+MY_SSID);
```

```
WiFi.begin(MY_SSID, MY_PWD);
```

```
while (WiFi.status() != WL_CONNECTED) {
```

```
delay(1000);
```

```
Serial.print(".");
```

```
}
```

```
Serial.println("WiFi connected");
```

```
Serial.println("IP address: ");
```

```
Serial.println(WiFi.localIP());
```

```
}
```

```
voidA_sendDataToTS(float temp,floatLPG,floatCO,float SMOKE)
```

```
{
```

```
WiFiClient client;
```

```
if (client.connect(server, 80))
```

```
Serial.println("WiFi Client connected ");
```

```
String postStr = apiKey;
```

```
postStr += "&field1=";
```

```
postStr += String(temp);
```

```
postStr += "&field2=";
```

```
postStr += String(LPG);

postStr += "&field3=";

postStr += String(CO);

postStr += "&field4=";

postStr += String(SMOKE);

postStr += "\r\n\r\n";

client.print("POST /update HTTP/1.1\n");

client.print("Host: api.thingspeak.com\n");

client.print("Connection: close\n");

client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");

client.print("Content-Type: application/x-www-form-urlencoded\n");

client.print("Content-Length: ");

client.print(postStr.length());

client.print("\n\n");

client.print(postStr);

delay(1000);

}
```

```
floatMQResistanceCalculation(intraw_adc)

{

return ( ((float)RL_VALUE*(1023-raw_adc)/raw_adc));

}

floatMQCalibration(intmq_pin)

{

inti;

floatval=0;

for (i=0;i<CALIBARAION_SAMPLE_TIMES;i++) {

val += MQResistanceCalculation(analogRead(mq_pin));

delay(CALIBRATION_SAMPLE_INTERVAL);

}

val = val/CALIBARAION_SAMPLE_TIMES;

val = val/RO_CLEAN_AIR_FACTOR;
```



```
returnval;
```

```
}
```

```
floatMQRead(intmq_pin)
```

```
{
```

```
inti;
```

```
floatrs=0;
```

```
for (i=0;i<READ_SAMPLE_TIMES;i++) {
```

```
rs += MQResistanceCalculation(analogRead(mq_pin));
```

```
delay(READ_SAMPLE_INTERVAL);
```

```
}
```

```
rs = rs/READ_SAMPLE_TIMES;
```

```
returnrs;
```

```
}
```

```
intMQGetGasPercentage(float rs_ro_ratio, intgas_id)
```

```
{  
  
if ( gas_id == GAS_LPG ) {  
  
returnMQGetPercentage(rs_ro_ratio,LPGCurve);  
  
} else if ( gas_id == GAS_CO ) {  
  
returnMQGetPercentage(rs_ro_ratio,COCurve);  
  
} else if ( gas_id == GAS_SMOKE ) {  
  
returnMQGetPercentage(rs_ro_ratio,SmokeCurve);  
  
}  
  
return 0;  
  
}  
  
intMQGetPercentage(float rs_ro_ratio, float *pcurve)  
  
{  
  
return (pow(10,(((log(rs_ro_ratio)-pcurve[1])/pcurve[2]) + pcurve[0])));  
  
}
```

ประวัติผู้วิจัย

นายสุทธิเกียรติ กรมจรรยา เกิดเมื่อวันที่ 5 เดือนเมษายน พ.ศ. 2539 ที่จังหวัดพระนครศรีอยุธยา สำเร็จการศึกษาชั้นมัธยมศึกษาตอนปลายจากโรงเรียนสารคามพิทยาคม จังหวัดมหาสารคาม เมื่อปีการศึกษา 2557 เข้าศึกษาต่อในหลักสูตรวิทยาศาสตรบัณฑิตภาควิชาเคมี คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2558 ที่อยู่ที่สามารถติดต่อได้บ้านเลขที่ 170 หมู่ที่ 5 ตำบล โนนราศี อำเภอบรบือ จังหวัดมหาสารคาม รหัสไปรษณีย์ 11430 อีเมล oatss_dreamzone@hotmail.com