โครงการ

การเรียนการสอนเพื่อเสริมประสบการณ์

ชื่อโครงการ อุปกรณ์ติดตามคุณภาพอากาศระยะไกลแบบเรียลไทม์

Real-time remote sensing device for air-quality monitoring

ชื่อนิสิต นายวศิษฐ์พล สินธุผล

ภาควิชา เคมี

ปีการศึกษา 2558

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

เรื่อง อุปกรณ์ติดตามคุณภาพอากาศระยะไกลแบบเรียลไทม์

โดย นายวศิษฐ์พล สินธุผล

ได้รับอนุมัติให้เป็นส่วนหนึ่งของการศึกษา

ตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต ภาควิชาเคมี

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

คณะกรรมการสอบโครงการ

..... ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ปกรณ์ วรานุศาสกล)

..... อาจารย์ที่ปรึกษา

(อาจารย์ ดร.ภัสสร์พล งามอโฆษ)

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.อนวัช อาชวาคม)

รายงานฉบับนี้ได้รับความเห็นชอบและอนุมัติโดยหัวหน้าภาควิชาเคมี

.....

(รองศาสตราจารย์ ดร.วุฒิชัย พาราสุข)

หัวหน้าภาควิชาเคมี

วัน..... เดือน .....พ.ศ. .....

คุณภาพของการเขียนรายงานเล่มนี้อยู่ในระดับ ☐ดีมาก ☑ดี ☐พอใช้

ชื่อโครงการ อุปกรณ์ติดตามคุณภาพอากาศระยะไกลแบบเรียลไทม์

ชื่อนิสิตในโครงการ นายวศิษฐ์พล สินธุผล เลขประจำตัว 553 31486 23

ชื่ออาจารย์ที่ปรึกษา อาจารย์ ดร.ภัสสร์พล งามอุโฆษ

ภาควิชาเคมี คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2558

บทคัดย่อ

ปัจจุบันเทคโนโลยีพัฒนาไปอย่างรวดเร็ว ซึ่งไมโครคอนโทรลเลอร์เป็นอีกหนึ่งเทคโนโลยีที่ได้รับความนิยม เนื่องจากสามารถเพิ่มฮาร์ดแวร์แบบต่าง ๆ มารองรับการทำงานที่หลายมากขึ้นของไมโครคอนโทรลเลอร์ ซึ่ง Arduino เป็นหนึ่งในไมโครคอนโทรลเลอร์ที่มีความง่ายและหลากหลายของการใช้งาน อีกทั้งยังมีราคาถูก น้ำหนักเบา และสามารถประยุกต์ใช้ได้หลากหลาย ผู้พัฒนาจึงมีแนวคิดที่จะนำไมโครคอนโทรลเลอร์ Arduino มาใช้ในงานด้านวิทยาศาสตร์โดยใช้ติดตามคุณภาพอากาศ เนื่องจากอุปกรณ์ติดตามคุณภาพอากาศมีราคาแพงและหาซื้อได้ยาก ผู้พัฒนาจึงต้องการพัฒนาอุปกรณ์ติดตามคุณภาพอากาศให้สามารถใช้งานได้เทียบเท่าอุปกรณ์ที่ใช้อยู่ในปัจจุบัน และต้องการให้มีการใช้งานที่ง่ายต่อผู้ใช้งานอีกด้วย และด้วยปริมาณแก๊สคาร์บอนไดออกไซด์ที่มีปริมาณเพิ่มขึ้นเรื่อย ๆ ในปัจจุบัน เนื่องจากแก๊สคาร์บอนไดออกไซด์เป็นหนึ่งในปัญหาคุณภาพอากาศ ที่ส่งผลต่อสุขภาพของมนุษย์และยังส่งผลต่อปัญหาโลกร้อนอีกด้วย ดังนั้นทางผู้พัฒนาสนใจที่จะพัฒนาอุปกรณ์ที่ใช้สำหรับตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์ โดยสามารถทำการสอบเทียบอุปกรณ์ให้สามารถตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์ที่มีความเข้มข้นตั้งแต่ 10-2000 ppm

คำสำคัญ : ไมโครคอนโทรลเลอร์, ตรวจวัดแก๊ส, คุณภาพอากาศ

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

Title Real-Time Remote-Sensing Device for Air-Quality Monitoring

Student Name Mr.Wasitpon Sinthuphon ID 553 31486 23

Advisor Name Dr.Passapol Ngamukot

Department of Chemistry, Faculty of science, Chulalongkorn University

Abstract

Nowadays, the technology has grown rapidly, and the microcontroller is one of the most popular technology, because there are many types of hardware that can support various uses of the microcontroller. Arduino is I brand of microcontroller that is easy to use and has various applications. Moreover, it is inexpensive with light-weight. From the above reasons, we intended to create device for air-quality monitoring based on the Arduino microcontroller concept. Since the device for air-quality monitoring is expensive and very rare item, the device for air-quality monitoring with working effectiveness close to that of the conventional device, and friendly to use has become our development target. Carbon dioxide (CO_2) gas has become a serious problem that can affect human-health and also cause global-warming. Therefore, the device that can be used to measure CO_2 gas by calibration method in the concentration range of 10-2000 ppm.

Keywords : Microcontroller, Gas Sensing, Air quality

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	**1**
1.1 ความเป็นมาและมูลเหตุจูงใจ	1
1.2 ผลงานวิจัยที่เกี่ยวข้อง	2
1.3 แนวคิดเหตุผลและขอบเขต	6
1.4 วัตถุประสงค์ของโครงการ	6
1.5 ประโยชน์ที่คาดว่าจะได้รับ	7
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	**8**
2.1 อุปกรณ์พื้นฐาน	8
2.1.1 ไมโครคอนโทรลเลอร์ Arduino	8
2.1.2 โมดูล (Module)	11
2.2 ขั้นตอนในการเขียนชุดคำสั่งเพื่อให้ไมโครคอนโทรลเลอร์ Arduino ทำงาน	15
2.3 รูปแบบการเขียนชุดคำสั่งลงบนไมโครคอนโทรลเลอร์ยี่ห้อ Arduino	17
2.4 โครงสร้างการเขียนชุดคำสั่งภาษาซีสำหรับไมโครคอนโทรลเลอร์ Arduino	19
2.4.1 โครงสร้าง (Structure)	19
2.4.2 ตัวแปรและค่าคงที่	21
2.4.3 ฟังก์ชัน	22
2.5 จุดกระจายสัญญาณอินเตอร์เน็ตไร้สาย (Wi-Fi Hotspot)	24
บทที่ 3 อุปกรณ์ การออกแบบระบบ และวิธีการทดลอง	**26**
3.1 อุปกรณ์ที่ใช้ในการทดลอง	26
3.2 สารเคมีที่ใช้ในการทดลอง	26
3.3 การออกแบบระบบ	29
3.3.1 การรวบรวมข้อมูล	29
3.3.2 แผนภาพโครงสร้าง	30
3.3.3 แผนภาพยูสเคส	31
3.3.4 แผนภาพสถานะ	34
3.3.5 แผนภาพลำดับการทำงาน	36
3.4 วิธีการทดลอง	38
3.4.1 ทำเครื่องตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์ในอากาศ	38

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

	หน้า
3.4.2 ทดสอบความเสถียรของเครื่องตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซค์ในอากาศ	40
3.4.3 สอบเทียบเครื่องตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซค์ในอากาศ	41
3.4.4 ใช้เครื่องตรวจวัดปริมาณแก๊สตัววัดปริมาณแก๊สคาร์บอนไดออกไซค์ในอากาศ	43
บทที่ 4 ผลการทดลองและอภิปรายผลการทดลอง	**45**
4.1 ขั้นตอนการทดสอบความเสถียรของเครื่องตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซค์ในอากาศ	45
4.2 ขั้นตอนการสอบเทียบเครื่องตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซค์ในอากาศ	48
4.3 ขั้นตอนการใช้เครื่องตรวจวัดปริมาณแก๊สตัววัดปริมาณแก๊สคาร์บอนไดออกไซค์ในอากาศ	51
บทที่ 5 สรุปผลการทดลอง	**56**
5.1 สรุปผล	56
5.2 ปัญหาและอุปสรรค	56
5.3 การแก้ไขปัญหา	56
5.4 ข้อเสนอแนะ 55	
อ้างอิง	**58**
ภาคผนวก	**60**
ประวัติผู้วิจัย	**75**

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญรูป

รูปที่	หน้า
รูปที่ 1.1 ไมโครคอนโทรเลอร์ยี่ห้อ Arduino	2
รูปที่ 1.2 โมดูลสำหรับวัดแก๊ส	2
รูปที่ 1.3 แผนภาพของชุดตรวจวัดการดูดกลืนของแสง	2
รูปที่ 1.4 กราฟแสดงค่าการดูดกลืนแสงระหว่างชุดการตรวจวัดการดูดกลืนแสงจากไมโครคอน-โทรลเลอร์เทียบกับเครื่องมือวิทยาศาสตร์ที่ใช้สำหรับการดูดกลืนแสง	3
รูปที่ 1.5 ผังวงจรสำหรับอุปกรณ์ติดตามแรงดันที่บริเวณเท้า	3
รูปที่ 1.6 อุปกรณ์ติดตามแรงดันที่บริเวณเท้า	4
รูปที่ 1.7 ผังการทำงานของอุปกรณ์ตรวจวัดอุณหภูมิและอัตราการเต้นของหัวใจ	5
รูปที่ 1.8 โมดูลเซนเซอร์สำหรับวัดค่าความเป็นกรด-เบส	5
รูปที่ 1.9 ไมโครคอนโทรลเลอร์ที่ใช้สำหรับรับข้อมูลจากไมโครคอนโทรลเลอร์ที่วัดค่าความเป็นกรด	6
รูปที่ 2.1 ส่วนประกอบของไมโครคอนโทรลเลอร์ยี่ห้อ Arduino รุ่น UNO R3	9
รูปที่ 2.2 เปรียบเทียบความแตกต่างระหว่างไมโครคอนโทรเลอร์ Arduino ในแต่ละรุ่น	10
รูปที่ 2.3 วรจรของโมดูลแก๊สเซนเซอร์	11
รูปที่ 2.4 ใช้ Ohm meter วัดหาค่าความต้านทานที่ขา A และ B	12
รูปที่ 2.5 วงจรของโมดูลแก๊สเซนเซอร์	12
รูปที่ 2.6 การคำนวณหาความต้านทานของโมดูลแก๊สเซนเซอร์	12
รูปที่ 2.7 ทินออกไซด์ในแก๊สเซนเซอร์ MQ-135	13
รูปที่ 2.8 การเชื่อมต่อระหว่างโมดูลแก๊สเซนเซอร์กับไมโครคอนโทรลเลอร์	14
รูปที่ 2.9 ตัวอย่างโมดูลไวร์เลส ขนาดประมาณ 14.3 × 24.8 มิลลิติเมตรและพอร์ตที่เชื่อมต่อ	14
รูปที่ 2.10 การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์ Arduino กับคอมพิวเตอร์ผ่านสาย USB	15
รูปที่ 2.11 เลือกรุ่นบอร์ด Arduino ที่ต้องการ upload	15
รูปที่ 2.12 เลือกหมายเลขพอร์ตที่จะใช้เชื่อมต่อกับบอร์ด Arduino	16
รูปที่ 2.13 การกด verify และการ upload ลงบนบอร์ด Arduino	16
รูปที่ 2.14 แถบเครื่องมือของโปรแกรม Arduino IDE	17
รูปที่ 2.15 หน้าต่างโปรแกรม Arduino IDE	17
รูปที่ 2.16 โครงสร้างของฟังก์ชันในภาษา C สำหรับบอร์ด Arduino	23
รูปที่ 2.17 ตัวอย่างชุดคำสั่งที่มีส่วนของโครงสร้าง ตัวแปร และฟังก์ชัน	23
รูปที่ 2.18 การใช้งานของ Wi-Fi Hotspot	24

หน้า

รูปที่ 3.1 ไมโครคอนโทรลเลอร์ Arduino 26
รูปที่ 3.2 โมดูลแก๊สเซนเซอร์ รุ่น MQ-135 27
รูปที่ 3.3 โมดูลไวเรสรุ่น ESP8266 27
รูปที่ 3.4 สายไฟจัมเปอร์สำหรับเชื่อมต่อ 27
รูปที่ 3.5 สาย USB สำหรับ Arduino 27
รูปที่ 3.6 ถ่านไฟฉายแบบชาร์จได้ขนาด 9 โวลต์ 27
รูปที่ 3.7 รางถ่านที่มีหัวเชื่อมต่อกับไมโครคอนโทรลเลอร์ 28
รูปที่ 3.8 เครื่องวัดปริมาณแก๊สคาร์บอนไดออกไซค์จาก www.co2meter.com 29
รูปที่ 3.9 แผนภาพโครงสร้างของระบบ 30
รูปที่ 3.10 แผนภาพยูสเคส ของระบบ 31
รูปที่ 3.11 แผนภาพสถานะ 34
รูปที่ 3.12 แผนภาพลำดับของอุปกรณ์ควบคุมที่เป็น คอมพิวเตอร์ และ สมาร์ทโฟน 36
รูปที่ 3.13 แผนภาพลำดับของอุปกรณ์ควบคุมที่เป็น คอมพิวเตอร์ และ สมาร์ทโฟน 37
รูปที่ 3.14 การเชื่อมต่อโมดูลไวเลสรุ่น ESP8266 กับไมโครคอนโทรลเลอร์ Arduino 38
รูปที่ 3.15 การเชื่อมต่อโมดูลแก๊สเซนเซอร์รุ่น MQ-135 กับไมโครคอนโทรลเลอร์ Arduino 39
รูปที่ 3.16 การเชื่อมต่อระหว่างคอมพิวเตอร์และไมโครคอนโทรลเลอร์ Arduino 39
รูปที่ 3.17 การต่อแหล่งจ่ายไฟฟ้าให้กับไมโครคอนโทรลเลอร์ Arduino 39
รูปที่ 3.18 การนำเครื่องตรวจวัดปริมาณแก๊สใส่ในตู้ควบคุมสภาพอากาศ 41
รูปที่ 3.19 ตำแหน่งกระจกนาฬิกาที่มีโซเดียมไบคาร์บอเนตในตู้ควบคุมสภาพอากาศ 42
รูปที่ 3.20 แผนที่ในการเก็บข้อมูลปริมาณแก๊สคาร์บอนไดออกไซค์ 43
รูปที่ 4.1 กราฟความสัมพันธ์ระหว่างค่าที่อ่านได้จากเครื่องมือกับเวลาในหน่วยวินาที 45
รูปที่ 4.2 แสดงส่วนขยายจากรูปที่ 4.1 ที่ช่วงเวลา 70000 – 90000 วินาที 46
รูปที่ 4.3 กราฟแสดงการเก็บข้อมูลทุก ๆ 10 วินาทีเป็นเวลา 6 ชั่วโมง 3 ครั้ง 46
รูปที่ 4.4 กราฟแสดงส่วนขยายจากรูปที่ 4.3 ที่ช่วงเวลา 0-3600 วินาที 47
รูปที่ 4.5 กราฟส่วนขยายจากรูปที่ 4.3 ที่ช่วงเวลา 0-1800 วินาที 48
รูปที่ 4.6 กราฟความเข้มข้นมาตรฐานระหว่าง ความเข้มข้นของแก๊สคาร์บอนไดออกไซค์ และค่าที่อ่านได้จากเครื่องมือ 49
รูปที่ 4.7 กราฟความเข้มข้นมาตรฐานระหว่าง ความเข้มข้นของแก๊สคาร์บอนไดออกไซด์ และค่าที่อ่านได้จากเครื่องมือ ที่ช่วงความเข้มข้นของแก๊สคาร์บอนไดออกไซด์ 100-1000 ppm 49

หน้า

รูปที่ 4.8 การเตรียมสารจากขวดรูปชมพู่และตำแหน่งในการวัดปริมาณแก๊สคาร์บอนไดออกไซด์ 49
รูปที่ 4.9 กราฟแสดงข้อมูลปริมาณแก๊สคาร์บอนไดออกไซด์บริเวณริมถนนพญาไท 51
รูปที่ 4.10 กราฟแสดงข้อมูลปริมาณแก๊สคาร์บอนไดออกไซด์บริเวณริมถนนอังรีดูนังต์ 52
รูปที่ 4.11 กราฟแสดงข้อมูลปริมาณแก๊สคาร์บอนไดออกไซด์บริเวณริมถนนพญาไท 52
รูปที่ 4.12 กราฟแสดงข้อมูลปริมาณแก๊สคาร์บอนไดออกไซด์บริเวณริมถนนพระราม 4 53
รูปที่ 4.13 ระดับแก๊สคาร์บอนไดออกไซด์ที่ส่งผลต่อสุขภาพ 55

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

ตารางที่ **หน้า**

ตารางที่ 2.1 คำสงวนสำหรับตัวแปร ที่เป็นค่าคงที่ 20

ตารางที่ 2.2 คำสงวนสำหรับตัวแปร ที่เป็นค่าคงที่ของพอร์ตในบอร์ด Arduino 20

ตารางที่ 2.3 คำสงวนสำหรับตัวแปร ที่เป็นชนิดข้อมูล 20

ตารางที่ 2.4 คุณสมบัติของตัวแปรในภาษาซี 21

ตารางที่ 3.1 ยูสเคส เชื่อมต่ออุปกรณ์ควบคุม 31

ตารางที่ 3.2 ยูสเคส รับข้อมูลการควบคุม 31

ตารางที่ 3.3 ยูสเคส รับข้อมูลจากโมดูลแก๊สเซนเซอร์ 31

ตารางที่ 3.4 ยูสเคส คำนวณปริมาณแก๊ส 32

ตารางที่ 3.5 ยูสเคส ส่งข้อมูลปริมาณแก๊สไปยังอุปกรณ์ควบคุม 32

ตารางที่ 3.6 ยูสเคส แสดงผล ปริมาณแก๊ส 33

ตารางที่ 4.1 ปริมาณโซเดียมไบคาร์บอเนตที่ต้องใช้ในสำหรับความเข้มข้นต่าง ๆ ของแก๊สคาร์บอนไดออกไซค์ 49

ตารางที่ 4.2 ปริมาณแก๊สคาร์บอนไดออกไซค์เฉลี่ย ณ บริเวณริมถนนต่าง ๆ 52

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1
บทนำ

1.1 ความเป็นมาและมูลเหตุจูงใจ

สิ่งประดิษฐ์ในปัจจุบันเริ่มเข้ามามีบทบาทในชีวิตมากขึ้นเรื่อย ๆ ในยุคที่ทุกอย่างขับเคลื่อนไปด้วยเทคโนโลยีงานวิจัยทางด้านวิทยาศาสตร์ถูกหยิบมาประยุกต์ใช้ในชีวิตประจำวันมากขึ้น สิ่งประดิษฐ์นอกจากจะต้องสามารถใช้งานได้จริงแล้วยังต้องมีความทันสมัยและสามารถตอบโจทย์ต่อผู้ใช้งานได้อีกด้วย ความรวดเร็ว ขนาดที่สามารถพกพา และราคาคงยังเป็นตัวแปรสำคัญสำหรับสิ่งประดิษฐ์ในยุคปัจจุบัน หนึ่งในสิ่งประดิษฐ์นั้นใช้สำหรับตรวจสอบคุณภาพอากาศ เป็นสิ่งประดิษฐ์ที่ถูกนำมาใช้เป็นเครื่องมือในการตรวจวัดคุณภาพอากาศ โดยใช้การตรวจวัดปริมาณแก๊สในอากาศแล้วนำมาหาค่าคุณภาพอากาศ โดยพื้นฐานมีการใช้งานที่ง่ายและสามารถหาซื้อได้ทั่วไปอยู่แล้ว อย่างไรก็ตามเครื่องมือดังกล่าวยังมีข้อจำกัดอีกมาก เนื่องจากไม่สามารถติดตามข้อมูลในระยะไกลได้ หากพื้นที่ ที่ทำการตรวจวัดคุณภาพอากาศที่คุณภาพอากาศที่ไม่ดีอาจส่งผลให้ผู้ที่ทำการตรวจวัดได้รับอันตรายได้ด้วย นอกจากนี้ยังมีข้อจำกัดในด้านการเก็บบันทึกข้อมูลที่จำเป็นต้องเอาไปเชื่อมต่อกับคอมพิวเตอร์ก่อนเพื่อทำการวิเคราะห์ ไม่สามารถวิเคราะห์ข้อมูลที่ได้มาในทันที เป็นต้น

เนื่องจากเทคโนโลยีในปัจจุบันที่พัฒนาขึ้นอย่างมากเมื่อเทียบกับหลายสิบปีที่ผ่านมาทำให้คอมพิวเตอร์สมาร์ทโฟนถูกใช้งานกันอย่างแพร่หลาย สิ่งประดิษฐ์หลาย ๆ อย่างจึงเริ่มมีการประดิษฐ์ให้สามารถใช้ร่วมกับเทคโนโลยีสมัยใหม่มากขึ้นเรื่อย ๆ เพราะนอกจากจะทำให้ใช้งานง่ายขึ้นแล้ว ยังสามารถใช้งานได้หลากหลายมากยิ่งขึ้น ไม่ว่าจะเป็นการใช้คอมพิวเตอร์หรือสมาร์ทโฟนในการรับ-ส่งข้อมูล นำข้อมูลมาวิเคราะห์ผล ทำให้สามารถได้ผลการวิเคราะห์รวดเร็วขึ้นมาก ประกอบกับสิ่งประดิษฐ์ที่เริ่มมีการติดต่อกับคอมพิวเตอร์และสมาร์ทโฟนมากยิ่งขึ้น เช่นไมโครคอนโทรลเลอร์ (Microcontroller) ดังรูปที่ 1.1 เป็นเสมือนคอมพิวเตอร์ขนาดจิ๋ว โดยอุปกรณ์ตัววัดต่าง ๆ เช่นตัววัดสี แสง ระยะทาง อุณหภูมิ ความชื้น ปริมาณแก๊ส (รูปที่ 1.2) โดยใช้โมดูลต่อเข้ากับไมโครคอนโทรลเลอร์ เป็นต้น นอกจากนี้ยังสามารถติดตั้งระบบส่งผ่านข้อมูลให้กับไมโครคอนโทรลเลอร์ได้ โดยส่งผ่านข้อมูลผ่านเครือข่ายไร้สาย (Wireless) ซึ่งยังสามารถนำไปเชื่อมต่อกับระบบอินเตอร์เน็ต (Wi-Fi) ผ่านเพื่อสามารถติดตามข้อมูลที่ส่งผ่านจากไมโครคอนโทรลเลอร์ได้

ดังนั้นผู้พัฒนาจึงมีความสนใจที่จะพัฒนาไมโครคอนโทรลเลอร์ที่ใช้วัดปริมาณแก๊ส ที่สามารถส่งผ่านข้อมูลผ่านระบบเครือข่ายไร้สายกับคอมพิวเตอร์หรืออินเตอร์เน็ตด้วยวิธีนี้จะทำให้สามารถติดตามคุณภาพอากาศได้ตลอดเพราะสามารถส่งข้อมูลผ่านระบบอินเตอร์เน็ตได้ทันทีและตลอดเวลา นอกจากนี้ยังสามารถนำไมโครคอนโทรลเลอร์ไปวางไว้ในบริเวณที่มีคุณภาพอากาศไม่ดี แล้วผู้ใช้ก็สามารถตรวจวัดคุณภาพอากาศจากระยะไกล ๆ ได้ผ่านสมาร์ทโฟน เป็นต้น นอกจากนี้แล้วทางผู้พัฒนายังทำไมโครคอนโทรเลอร์ไว้เป็นระบบที่สามารถนำไปใช้งานได้ง่าย ทั้งนี้เพื่อรองรับสิ่งประดิษฐ์อื่นให้สามารถมาใช้งานร่วมกับไมโครคอนโทรเลอร์นี้ได้อีกด้วย ตัวอย่างสิ่งประดิษฐ์ที่สามารถนำมาใช้นั้นมีตัวอย่างเพียงแค่เป็นสิ่งประดิษฐ์ที่สามารถรับและส่งแรงดันไฟฟ้าได้ เช่น อิเล็กโทรด

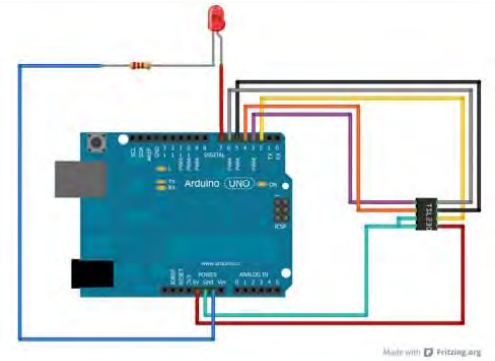
(Electrode) เป็นต้น ทำให้เกิดการนำงานวิจัยจากทางวิทยาศาสตร์มาเป็นสิ่งประดิษฐ์มากขึ้นซึ่งเป็นการเปิดโลกผลงานวิจัยที่มาจากทางด้านวิทยาศาสตร์ ส่งผลให้เยาวชนไทยเกิดความคิดสร้างสรรค์จนทำให้เกิดความรักในวิทยาศาสตร์มากขึ้นอีกด้วย

รูปที่ 1.1 ไมโครคอนโทรเลอร์ยี่ห้อ Arduino

รูปที่ 1.2 โมดูลสำหรับวัดแก๊ส

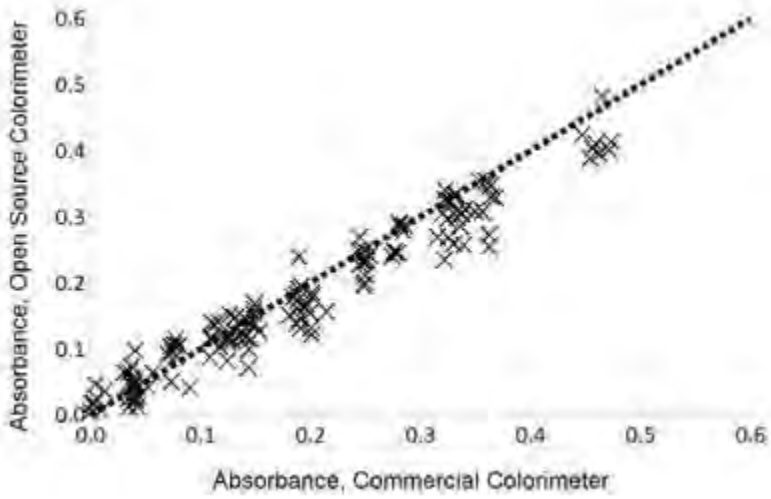
1.2 ผลงานวิจัยที่เกี่ยวข้อง

เริ่มต้นจากการเริ่มพัฒนาไมโครคอนโทรเลอร์ของยี่ห้อ Arduino ในช่วงปี 2005 จากนั้นจึงค่อย ๆ เริ่มเป็นที่รู้จักและมีการเริ่มนำมาใช้งานกันอย่างแพร่หลาย และในช่วงปี 2012 เริ่มมีงานวิจัยที่นำไมโครคอนโทรลเลอร์มาใช้ในงานวิจัย ดังในงานวิจัย Open-Source Colorimeter โดย Gerald C. Anzalone และคณะ [(1)] ได้ศึกษาเกี่ยวกับ เซนเซอร์โมดูล ที่สามารถตรวจวัดความแตกต่างของการดูดกลืนของแสง (Absorbance) ทางคณะผู้จัดจึงได้สร้างชุดตรวจวัดการดูดกลืนของแสงโดยอาศัยการควบคุมจากไมโครคอนโทรลเลอร์ยี่ห้อ Arduino ดังรูปที่ 1.3



รูปที่ 1.3 แผนภาพของชุดตรวจวัดการดูดกลืนของแสง[(1)]

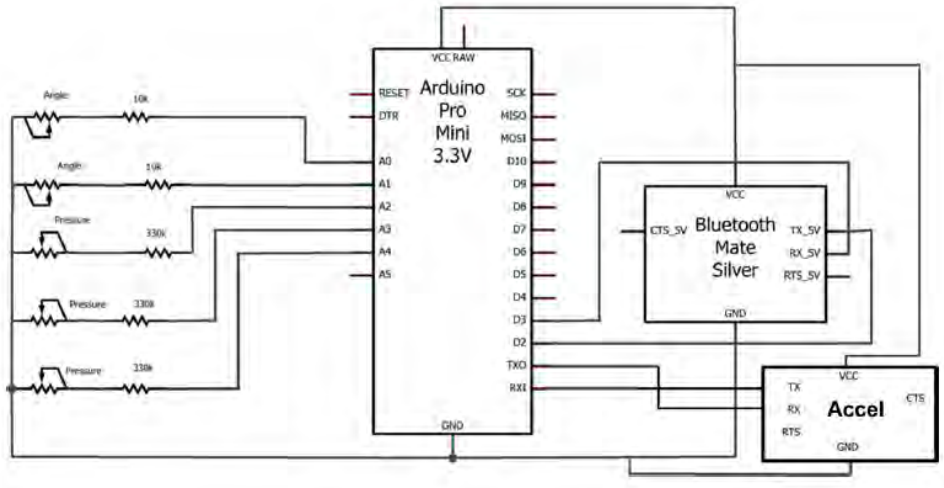
จากนั้นนำไดคลอโรมีเทน (Dichloromethane) มาทำการวัดค่าการดูดกลืนแสงที่ความยาวคลื่น 440 นาโนเมตร พบว่าสามารถนำมาใช้เปรียบเทียบกับเครื่องมือทางวิทยาศาสตร์แล้วได้ผลการทดลองได้รูปที่ 1.4



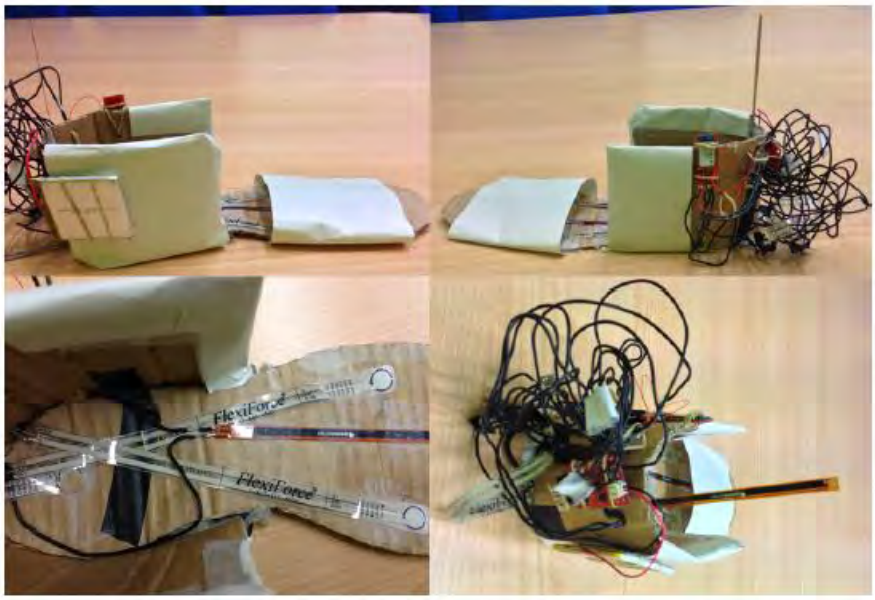
รูปที่ 1.4 กราฟแสดงค่าการดูดกลืนแสงระหว่างชุดการตรวจวัดการดูดกลืนแสงจากไมโครคอนโทรลเลอร์เทียบกับเครื่องมือวิทยาศาสตร์ที่ใช้สำหรับการดูดกลืนแสง[2]

จากข้อมูลจะเห็นว่าเครื่องมือที่พัฒนาโดยใช้ไมโครคอนโทรลเลอร์นั้นสามารถตรวจวัดค่าการดูดกลืนแสงได้เทียบเท่ากับเครื่องมือทางวิทยาศาสตร์ ที่ความยาวคลื่น 440 นาโนเมตร จากงานวิจัยนี้ทำให้เป็นจุดเริ่มต้นของงานวิจัยอีกมากมายที่ใช้ความสามารถของไมโครคอนโทรลเลอร์ในการวิเคราะห์ค่าต่าง ๆ ในทางวิทยาศาสตร์

ต่อมาในปีเดียวกันนี้ มีงานวิจัยเรื่อง Wireless Prototype Based on Pressure and Bending Sensors for Measuring Gate Quality ซึ่งงานวิจัยนี้ใช้ไมโครคอนโทรลเลอร์เพื่อติดตามแรงดันบริเวณเท้าด้วยโมดูลเซนเซอร์ที่เป็นแบบโค้งงอได้ โดยใช้อุปกรณ์ดังรูปที่ 1.5 และ 1.6



รูปที่ 1.5 ผังวงจรสำหรับอุปกรณ์ติดตามแรงดันที่บริเวณเท้า[3]



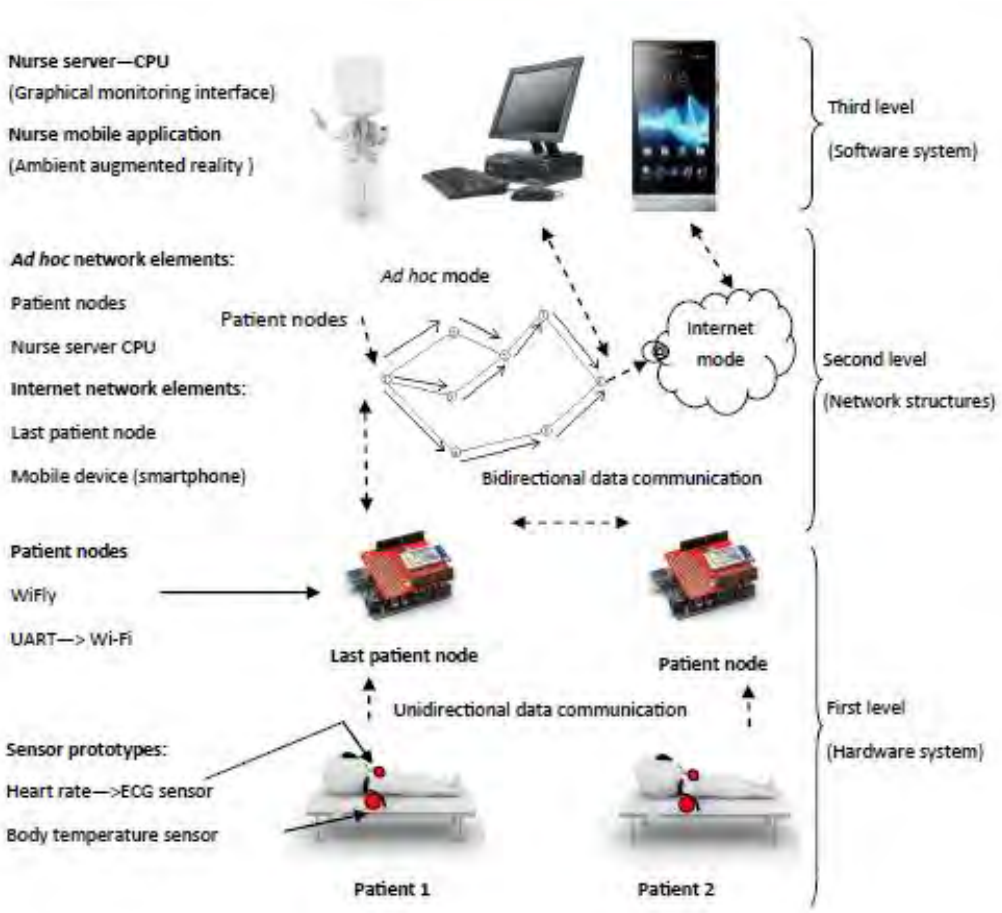
รูปที่ 1.6 อุปกรณ์ติดตามแรงดันที่บริเวณเท้า[4]

จากนั้นนำอุปกรณ์ดังกล่าวมาสวมเข้ากับเท้าของผู้ทดลองซึ่งเราสามารถส่งผ่านข้อมูลระยะใกล้ได้โดยการใช้ บลูธูทโมดูล ทำให้เริ่มมีการส่งผ่านข้อมูลในระยะใกล้ ๆ โดยไม่ต้องใช้สายเพื่อส่งผ่านข้อมูลระหว่างไมโครคอนโทรลเลอร์กับเครื่องคอมพิวเตอร์ได้

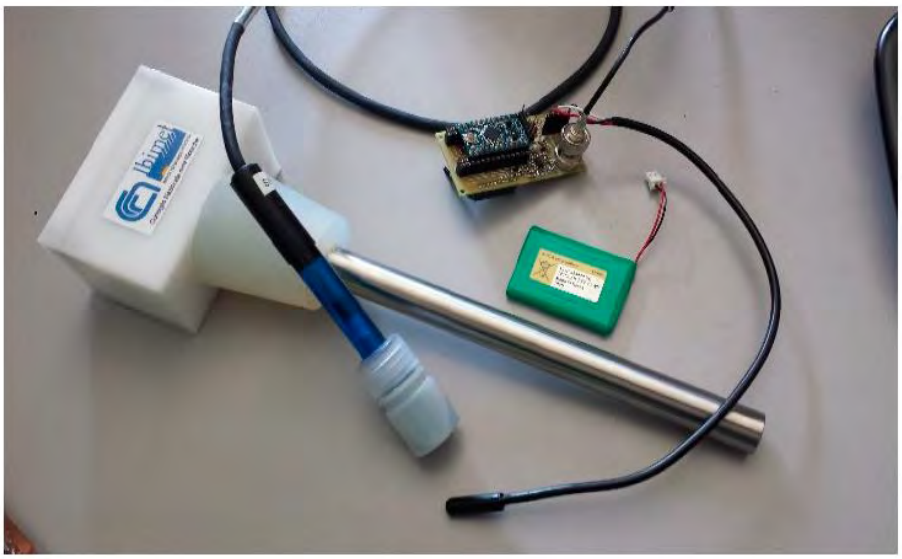
ต่อมาในปี 2014 ไมโครคอนโทรเลอร์ถูกนำไปใช้ในทางการแพทย์มากขึ้น โดยถูกนำไปใช้วัดอุณหภูมิและอัตราการเต้นหัวใจของคนไข้ โดยใช้ไมโครคอนโทรเลอร์อ่านค่าอุณหภูมิและอัตราการเต้นของหัวใจ จากนั้นจึงส่งผ่านข้อมูลสู่เครือข่ายอินเตอร์เน็ตผ่านระบบ Wi-Fi จากนั้นก็ใช้คอมพิวเตอร์ หรือโทรศัพท์สมาร์ทโฟนดังรูปที่ 1.7 จากงานวิจัยชิ้นนี้ทำให้เห็นว่าเราสามารถนำไมโครคอนโทรลเลอร์มาใช้เป็นระบบเสมือนกับเป็นเครื่องมือติดตามข้อมูลที่มีราคาแพงได้ นอกจากนี้ยังสามารถนำไปใช้ร่วมกับอินเตอร์เน็ต คอมพิวเตอร์และสมาร์ทโฟนได้อีกด้วย

และในปีเดียวกันนั้นมีงานวิจัยเรื่อง An Open-Source and Low-Cost Monitoring System for Precision Enology เป็นงานวิจัยที่นำเอาไมโครคอนโทรลเลอร์มาใช้ตรวจวัดค่าความเป็นกรด-เบส แล้วนำมาทำกราฟความเข้มข้นมาตรฐาน (Calibration Curve) ซึ่งได้ค่าความเป็นเส้นตรงสูงถึง 99.99% นอกจากนี้ยังมีการนำเอาไมโครคอนโทรลเลอร์สองชนิดที่แตกต่างกันซึ่งสามารถส่งผ่านข้อมูลให้กันได้ผ่านระบบ Wi-Fi ซึ่งในงานวิจัยทำให้เห็นว่า นอกจากเราสามารถส่งข้อมูลผ่านอินเตอร์เน็ตได้แล้ว ยังสามารถติดต่อหากันได้อีกด้วยจากรูปที่ 1.8 วัดค่าความเป็นกรด-เบส จากนั้นก็ส่งผ่านข้อมูลไปยัง ไมโครคอนโทรลเลอร์ตามรูปที่ 1.9

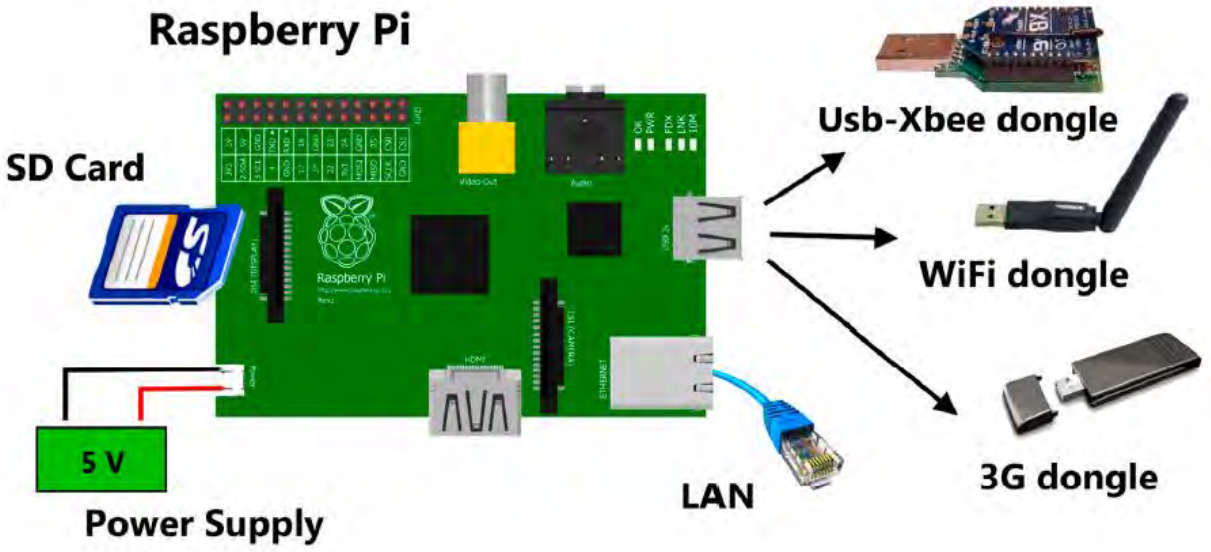
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 1.7 ผังการทำงานของอุปกรณ์ตรวจวัดอุณหภูมิและอัตราการเต้นของหัวใจ [5]



รูปที่ 1.8 โมดูลเซนเซอร์สำหรับวัดค่าความเป็นกรด-เบส[6]



รูปที่ 1.9 ไมโครคอนโทรลเลอร์ที่ใช้สำหรับรับข้อมูลจากไมโครคอนโทรลเลอร์ที่วัดค่าความเป็นกรด-เบส [6]

1.3 แนวคิดเหตุผลและขอบเขต

ทางผู้พัฒนาได้นำข้อดีของไมโครคอนโทรลเลอร์มาพัฒนาต่อ เนื่องจากมีความสามารถเข้ากันได้กับเทคโนโลยีในปัจจุบันหลาย ๆ อย่างซึ่งเป็นจุดแข็งของสิ่งประดิษฐ์ที่จำเป็นจะต้องมีในยุคปัจจุบัน โดยสามารถส่งผ่านข้อมูลเข้าสู่ระบบอินเตอร์เน็ต โทรศัพท์สมาร์ทโฟน คอมพิวเตอร์ และสามารถส่งผ่านข้อมูลให้กันได้อีกด้วย โดยในรายงานนี้มีขอบเขตในการทำงานดังนี้

1. อุปกรณ์ที่พัฒนาขึ้นจะใช้กับไมโครคอนโทรลเลอร์ยี่ห้อ Arduino กับรุ่นใดก็ได้ โดยเชื่อมต่อกับโมดูลแก๊สเซนเซอร์เพื่อตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์
2. โปรแกรมสำหรับควบคุมอุปกรณ์ที่พัฒนาขึ้นจะใช้ภาษาซี เฉพาะของไมโครคอนโทรลเลอร์ยี่ห้อ Arduino เท่านั้น

1.4 วัตถุประสงค์ของโครงการ

1. เพื่อพัฒนาโปรแกรมวัดปริมาณแก๊สคาร์บอนไดออกไซด์สำหรับเครื่องมือวัดปริมาณแก๊สโดยใช้ไมโครคอนโทรลเลอร์ Arduino
2. พัฒนาเครื่องมือที่ใช้ในการหาปริมาณของสารมลพิษทางอากาศให้สามารถพกพาได้ง่ายและส่งข้อมูลผ่านคอมพิวเตอร์แบบเรียลไทม์ได้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ผู้พัฒนาได้ประสบการณ์การเรียนรู้การพัฒนาโปรแกรจากไมโครคอนโทรลเลอร์ Arduino
2. สามารถพัฒนาการเขียนโปรแกรมรวมไปถึงเทคนิคต่าง ๆ ของภาษาซี
3. สามารถนำผลงานไปใช้งานได้จริงอีกทั้งสามารถใช้เป็นแนวทางให้กับนักพัฒนาที่ต้องการเครื่องมือที่เป็นระบบไปใช้เชื่อมต่อกับโมดูลที่ตนต้องการได้
4. ได้เครื่องมือที่สามารถใช้วัดปริมาณแก๊สในราคาถูก และสามารถเชื่อมต่อกับอุปกรณ์เทคโนโลยีในปัจจุบันได้คือ คอมพิวเตอร์ และสมาร์ทโฟน

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

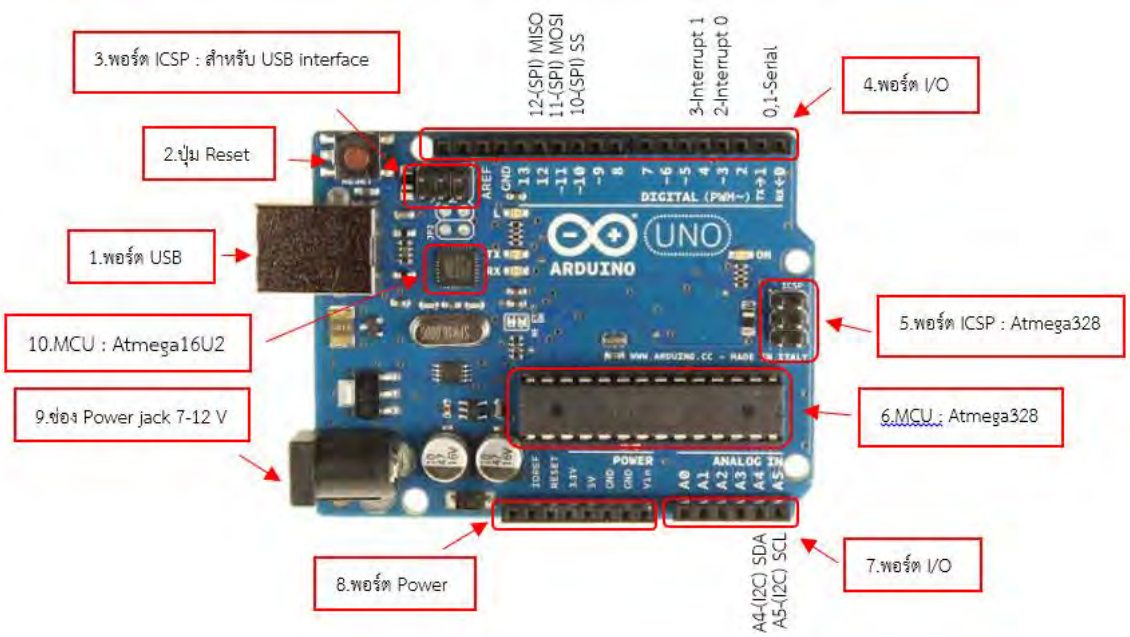
2.1 อุปกรณ์พื้นฐาน

2.1.1 ไมโครคอนโทรลเลอร์ Arduino

ไมโครคอนโทรลเลอร์คืออุปกรณ์ชนิดหนึ่งที่มีชิปไอซีพิเศษ ซึ่งเราสามารถเขียนชุดคำสั่งลงบนโปรแกรมเพื่อควบคุมการทำงานตามงานที่เราต้องการได้โดย ตามรูปที่ 2.1 ซึ่งภายในไมโครคอนโทรลเลอร์นั้นจะประกอบไปด้วย 5 ส่วนหลัก ๆ คือ

- หน่วยประมวลผล (Processing Unit) เป็นหน่วยหลักของไมโครคอนโทรลเลอร์ซึ่งชุดคำสั่งที่เราใส่เข้าไปในไมโครคอนโทรลเลอร์นั้นจะถูกส่งมายังหน่วยประมวลผลนี้จากนั้นหน่วยประมวลผลจะแบ่งการทำงานออกเป็นสองส่วนคือ 1. หน่วยควบคุม (Control Unit) เป็นส่วนที่ควบคุมการที่อยู่ของชุดคำสั่งว่าควรไปอยู่ส่วนใดของหน่วยความจำ 2. หน่วยคำนวณและตรรกะ (Arithmetic and Logical Unit) เป็นส่วนที่ใช้คำนวณเกี่ยวกับชุดคำสั่งที่เราให้เข้ามาแล้วจึงออกคำสั่งต่อไปที่ส่วนอื่นโดยสรุปในส่วนประมวลผลนี้จะเปรียบเสมือนกับสมองของมนุษย์ ใช้เพื่อสั่งการนั่นเอง
- หน่วยความจำชั่วคราว (Random Access Memory) เป็นส่วนความจำสำหรับไมโครคอนโทรลเลอร์ชุดคำสั่งที่ถูกเขียนขึ้นนั้นจะถูกส่งมาเก็บไว้ที่ส่วนนี้ชั่วคราว / พักรอการประมวลผล แล้วจึงถูกส่งต่อไปยังส่วนประมวลผลเพื่อทำกาประมวลผลต่อไป
- หน่วยความจำถาวร (Read Only Memory) ชุดคำสั่งที่ถูกเขียนนั้นหากมีบางส่วนที่เราต้องเรียกใช้อย่างสม่ำเสมอ ในหน่วยความจำถาวรนี้ก็จะจดจำข้อมูลเอาไว้อย่างถาวร แม้จะไม่มีการให้ไฟฟ้ากับไมโครคอนโทรลเลอร์ก็ตาม
- พอร์ตอินพุต , พอร์ตเอาท์พุต (Input/Output Port) หรือเรียกอีกอย่างหนึ่งว่า (I/O) เป็นจุดที่ใช้ในการเชื่อมต่อกับอุปกรณ์หรือโมดูลต่าง ๆ ที่จะนำมาเชื่อมต่อกับไมโครคอนโทรลเลอร์ โดยจุดเชื่อมต่อนี้แบ่งออกเป็นสองส่วนคือส่วนของ Input คือส่วนที่รับข้อมูล (แรงดันไฟฟ้า) จากอุปกรณ์หรือโมดูลนั้น ๆ และอีกส่วนคือส่วนของ Output เป็นส่วนที่ให้ข้อมูล (แรงดันไฟฟ้า) กับอุปกรณ์นั้น ๆ ยกตัวอย่าง เช่น หลอดไฟ LEDเป็นประเภท Output ไมโครคอนโทรลเลอร์ต้องให้แรงดันไฟฟ้ากับหลอดไฟ LED จึงจะสว่าง ถ้าหากเป็นเซนเซอร์ตรวจวัดปริมาณแก๊ส เราจะต้องรับแรงดันไฟฟ้าจากเซนเซอร์นั้นมาที่ไมโครคอนโทรลเลอร์เพื่อให้เราสามารถอ่านข้อมูลจากเซนเซอร์นั้นได้
- แหล่งรับแรงดันไฟฟ้าจากภายนอก ไมโครคอนโทรลเลอร์นั้นมีส่วนที่รับแรงดันไฟฟ้าเพื่อใช้ในการทำงาน ได้สองแหล่งเมื่อพิจารณาจากรูปที่ 2.1 ในส่วนที่ 1 คือจากพอร์ต USB และส่วนที่ 9 จากที่ชาร์จ Arduino

นอกจากนี้แล้วไมโครคอนโทรลเลอร์ยี่ห้อ Arduino นั้นยังมีรุ่นที่พัฒนามาเพื่อทำงานที่แตกต่างกันซึ่งเราสามารถดูภาพรวมได้จากรูปที่ 2.2 เราจะเห็นว่าแต่ละรุ่นนั้นมีส่วนของ 4 ส่วนข้างต้นที่ไม่เหมือนกัน



รูปที่ 2.1 แสดงส่วนประกอบของไมโครคอนโทรลเลอร์ยี่ห้อ Arduino รุ่น UNO R3 [6]

จากรูปที่ 2.1 สามารถอธิบายแต่ละส่วนได้ดังนี้

1. พอร์ต USB ใช้สำหรับเชื่อมต่อกับคอมพิวเตอร์เพื่อรับชุดคำสั่งจากคอมพิวเตอร์เข้าสู่ไมโครคอนโทรล และสามารถรับแรงดันไฟฟ้าจากคอมพิวเตอร์เข้าสู่บอร์ดได้อีกด้วย
2. ปุ่ม Reset เป็นปุ่มกดเมื่อต้องการให้บอร์ดทำการเริ่มทำงานใหม่อีกครั้ง
3. พอร์ต ICSP เป็นพอร์ตที่ใช้สำหรับโปรแกรม Visual Com port บนชิพ MCU Atmega16U2
4. และ 7. พอร์ต I/O ใช้สำหรับติดต่อกับโมดูลโดยมีลักษณะเป็นสัญญาณ Analog และ Digital นอกจากนี้บางพอร์ตยังทำหน้าที่ในการรับและส่งข้อมูลอนุกรม (ใน Pin 0,1 เป็น Tx สำหรับส่งข้อมูล และ Rx สำหรับอ่านข้อมูล)
5. พอร์ต ICSP: Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader
6. MCU: Atmega328 เป็นส่วนที่เป็นหัวใจหลักของไมโครคอนโทรลเลอร์ ซึ่งเป็นส่วนที่ทำหน้าที่ในการเป็นหน่วยประมวลผล หน่วยความจำหลัก และหน่วยความจำสำรอง ในส่วนเดียวกัน
8. พอร์ต Power ใช้สำหรับจ่ายแรงดันไฟฟ้าให้กับโมดูลที่มาเชื่อมต่อเพื่อให้โมดูลนั้นสามารถทำงานได้
9. ช่อง Power jack 7-12 V เป็นช่องสำหรับจ่ายแรงดันไฟฟ้าให้กับตัวไมโครคอนโทรลเลอร์
10. MCU: Atmega16U2 เป็นส่วนที่ทำหน้าที่แปลงข้อมูลจากพอร์ต USB สู่ MCU: Atmega328 ในข้อ 6

	Processor					Input / Output							Power				Connectivity				
	Family	SRAM	FLASH	EEPROM	Clock	Digital I/O	Analog In	ADC Bits	PWM	UART	Analog Out	DAC Bits	VCC	Vin Range	5V	3V3	USB-Serial	I2C	Ethernet	USB-Host	SD Card
Arduino UNO R3	ATmega328	2k	32k	1k	16MHz	14	6	10	6	1	N/A	N/A	5V	7-12V	Yes	Yes	ATmega16U2	1	No	No	No
Arduino UNO SMD	ATmega328	2k	32k	1k	16MHz	14	6	10	6	1	N/A	N/A	5V	7-12V	Yes	Yes	ATmega16U2	1	No	No	No
Arduino Mega 2560 R3	ATmega2560	8k	256k	4k	16MHz	54	16	10	14	4	N/A	N/A	5V	7-18V	Yes	Yes	ATmega16U2	1	No	No	No
Arduino Mega ADK	ATmega2560	8k	256k	4k	16MHz	54	16	10	14	4	N/A	N/A	5V	7-18V	Yes	Yes	ATmega16U2	1	MAX3421E	No	No
Arduino Leonardo	ATmega32U4	2.5k	32k	1k	16MHz	25	12	10	7	1	N/A	N/A	5V	7-12V	Yes	Yes	Built-In	1	No	No	No
Arduino Mini 05	ATmega328	2k	32k	1k	16MHz	14	6	10	6	1	N/A	N/A	5V	7V-9V	Yes	No	N/A	1	No	No	No
Arduino Pro Mini 328 - 3.3V	ATmega328	2k	32k	1k	8MHz	14	6	10	6	1	N/A	N/A	3.3V	5V-12V	No	Yes	N/A	1	No	No	No
Arduino Pro Mini 328 – 5V	ATmega328	2k	32k	1k	16MHz	14	6	10	6	1	N/A	N/A	5V	7V-12V	Yes	No	N/A	1	No	No	No
Arduino Ethernet with PoE module	ATmega328	2k	32k	1k	16MHz	9	6	10	4	1	N/A	N/A	5V	6-18V	Yes	Yes	N/A	1	No	No	No
Arduino Ethernet without PoE module	ATmega328	2k	32k	1k	16MHz	9	6	10	4	1	N/A	N/A	5V	6-18V	Yes	Yes	N/A	1	No	No	No
Arduino DUE	SAM3X8E	96kb	512k	N/A	84MHz	70	12	12	12	4	2	12	3.3V	7-12V	No	VCC	Built-In	2	No	Yes	No

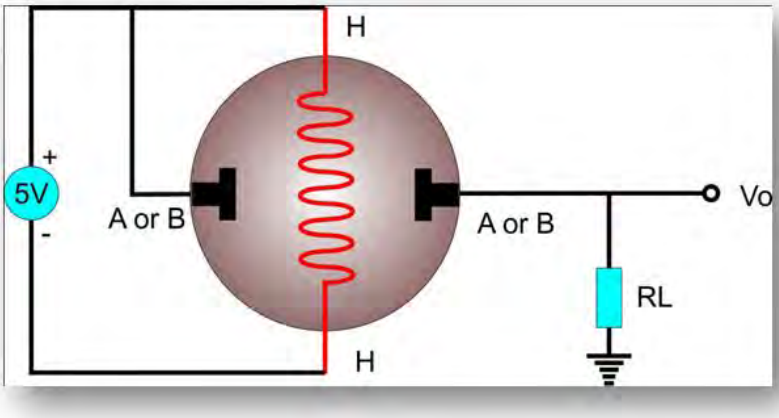
รูปที่ 2.2 เปรียบเทียบความแตกต่างระหว่างไมโครคอนโทรเลอร์ Arduino ในแต่ละรุ่น [6]

2.1.2 โมดูล (Module)

โมดูลนั้นเป็นชื่อที่เรียกอุปกรณ์ทุกอย่างที่จะนำมาเชื่อมต่อกับไมโครคอนโทรลเลอร์ผ่านพอร์ตอินพุต หรือพอร์ตเอาท์พุต โมดูลนั้นจะต้องมีวงจรไฟฟ้าเฉพาะ ของตนเองและสามารถรับหรือส่งข้อมูลให้กับไมโครคอนโทร-เลอร์ได้ ซึ่งในงานวิจัยนี้จะมีการใช้โมดูลสองชนิดคือ

2.1.2.1 โมดูลเซนเซอร์แก๊ส รุ่น MQ-135

[7]MQ-135 เป็น Sensor ตรวจสอบคุณภาพของอากาศโดยจะตรวจจับปริมาณ ก๊าซ NH_3, NO_x, alcohol, Benzene, smoke, CO_2, และ CO ในอากาศ จากรูปที่ 2.3 ซึ่งเมื่อเราเริ่มจ่ายพลังงานให้ MQ-135 ที่ขา H ทำให้เกิดความร้อนขึ้นที่ขดลวด เมื่อก๊าซพิษต่างๆ เข้ามาทำปฏิกิริยาจะทำให้ ค่าความต้านทานที่เกิดขึ้นระหว่าง ขา A และ B (RS) ลดลง



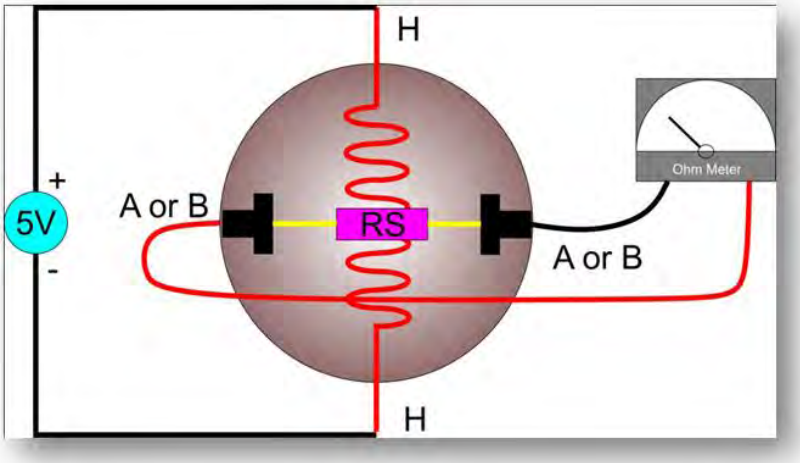
รูปที่ 2.3 วงจรของโมดูลแก๊สเซนเซอร์

จากรูปที่ 2.3 เมื่อจ่ายพลังงานให้กับ ขา H จะทำให้เกิดพลังงานความร้อนเพื่อให้สารเคมีภายในตัว Sensor สามารถทำปฏิกิริยากับก๊าซพิษต่างๆได้ และเมื่อ MQ-135 ตรวจจับก๊าซพิษต่างๆ ได้จะทำให้ค่าความต้านทานระหว่าง ขา A และ ขา B เปลี่ยนแปลง (ขา A และ B เป็นขาที่ไม่ตายตัวเราสามารถกำหนดเองได้โดยเลือกขาใดเป็นขา A ขาที่อยู่ฝั่งตรงข้ามก็จะเป็นขา B) โดยเมื่อ MQ-135 ตรวจจับปริมาณก๊าซพิษต่างๆ ได้มากจะทำให้ค่าความต้านทาน RS ลดลง หรือ ค่าความต้านทานแปรผกผันกับปริมาณของก๊าซพิษต่างๆ

- การหาค่าความต้านทานของ RS สามารถทำได้ 2 วิธีคือ

วิธีที่ 1 สามารถใช้ Ohm meter วัดหาค่าความต้านทานที่ขา A และ B ได้โดยตรงดังรูปที่ 2.4

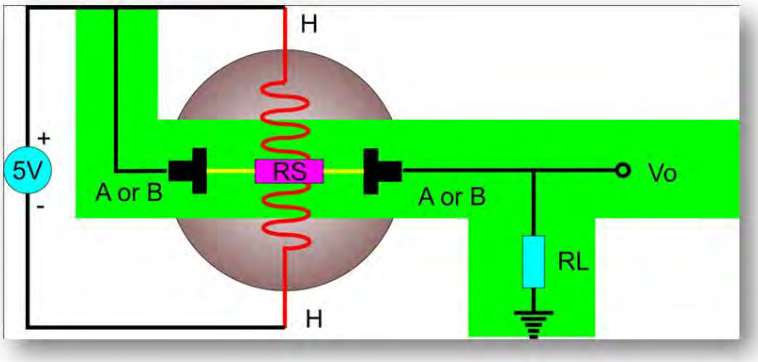
จุฬาลงกรณ์มหาวิทยาลัย



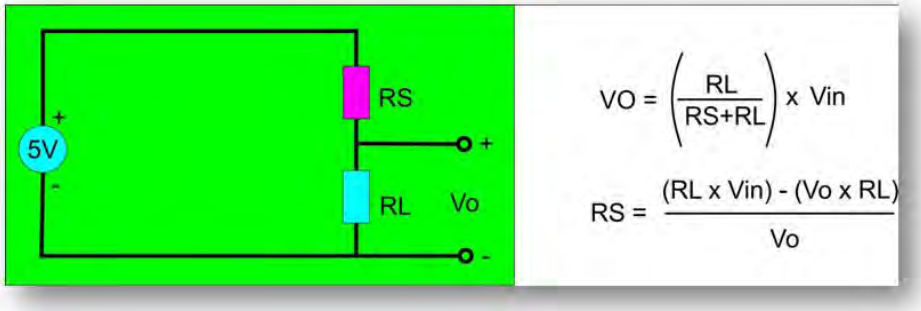
รูปที่ 2.4 ใช้ Ohm meter วัดหาค่าความต้านทานที่ขา A และ B

วิธีที่ 2 เป็นวิธีการวัดโดยอ้อม โดยใช้ กฎ แบ่งแรงดัน Voltage Divider

และจากรูปที่ 2.5 ตาม แถบสีเขียว จะเห็นว่าเราจ่ายกระแสไฟฟ้าเขามาทาง ขา A or B ทางด้านซ้าย ซึ่งเมื่อนำมาจัดเรียงใหม่ให้เป็นรูปที่เราเข้าใจกับ กฎแบ่งแรงดันจะเป็นรูปภาพดังรูปที่ 2.5 และ 2.6



รูปที่ 2.5 วงจรของโมดูลแก๊สเซนเซอร์



รูปที่ 2.6 การคำนวณหาความต้านทานของโมดูลแก๊สเซนเซอร์

จากรูปที่ 2.6 VO แรงดันไฟฟ้าที่ส่งไปยัง Microcontroller RL คือค่าความต้านทานภายใน (20 kΩ) RS คือค่าความต้านทานบริเวณ semiconductor ที่เกิดการดึงดูดบริเวณพื้นผิว Vin คือ แรงดันไฟฟ้าที่ให้กับระบบ

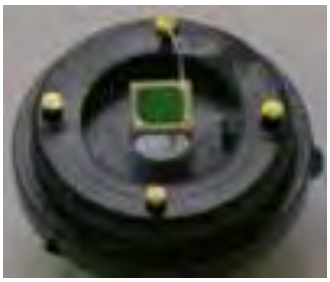
การหาค่า RS โดยใช้กฎแบ่งแรงดันนี้สามารถนำไปใช้กับ Microcontroller ได้โดยนำค่า Output ที่ได้ป้อนเข้าขา ADC ของ Microcontroller (การป้อนแรงดันเข้าขา ADC ของ MCU ต้องแน่ใจว่าระดับแรงดันที่ป้อนเข้าไปไม่เกินกว่าที่ Port ADC ของ MCU จะรับได้) และจะเห็นว่า แรงดันไฟฟ้าที่วัดได้จะมีค่ามากขึ้นเมื่อ RS ลดลง

ปฏิกิริยาเคมีที่เกิดขึ้นใน MQ-135

ภายในแก๊สเซนเซอร์นั้นจะมีสารเคมีที่ใช้สำหรับจับกับแก๊สมลพิษรวมไปถึงแก๊สคาร์บอนไดออกไซด์ด้วย โดยสารเคมีดังกล่าวคือ ผลึกของทินออกไซด์ (SnO_2) ดังรูปที่ 2.8 โดยมีหลักการสำหรับจับแก๊สต่าง ๆ ดังนี้

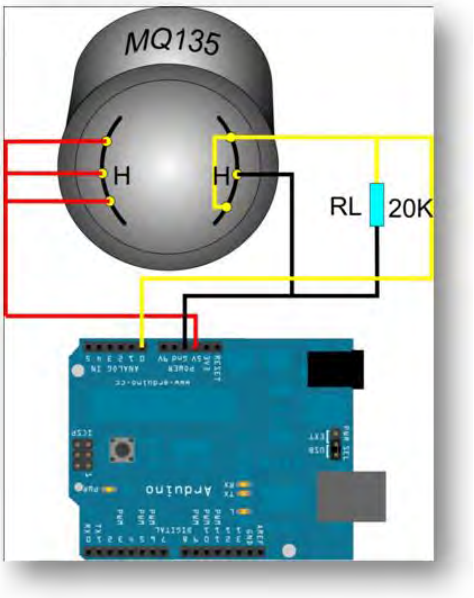
เริ่มต้นจากแก๊สเซนเซอร์ได้รับแรงดันไฟฟ้าจากไมโครคอนโทรลเลอร์ แล้วทำให้มีความร้อนเกิดขึ้นภายในอุปกรณ์จากตัวทำความร้อน ซึ่งความร้อนดังกล่าวจะทำให้ อิเล็กตรอนในผลึกของทินออกไซด์จากชั้น Valene band ขึ้นไปยังชั้น Conduction band จากเหตุการณ์นี้ส่งผลให้ผลึกของทินออกไซด์มีอิเล็กตรอนหนาแน่นที่บริเวณผิว และแก๊สคาร์บอนไดออกไซด์ที่ประกอบด้วยแก๊สออกซิเจนนั้นจะไปรับอิเล็กตรอน (ด้วยผลของออกซิเจนที่มีค่า Electro negativity สูง) บริเวณผิวซึ่งมีประจุลบอยู่หนาแน่น เมื่อมาเกาะในปริมาณที่มากขึ้นเรื่อย ๆ จะทำประจุลบบริเวณผิวมีน้อยลง ทำให้เซตปลอดพาหะซึ่งอยู่ระหว่าง Valence band และ Conduction band มีค่าลดลงแล้วทำให้ค่าความต้านหรือ RS จากรูปที่ 2.6 มีค่าต่ำลงด้วย ประกอบกับการเกิด Holes หรือประจุบวก (บริเวณผิวที่ขาดอิเล็กตรอน) จะได้รับอิเล็กตรอนจากออกซิเจน และหลุดออกไปเป็นออกซิเจนประจุบวก

โดยสรุปจากการเข้ามาเกาะบริเวณผิวของออกซิเจนจากคาร์บอนไดออกไซด์นั้น ส่งผลให้ค่าความต้านทานมีค่าลดต่ำลง หรือก็คือมีแรงดันไฟฟ้าสูงขึ้น เมื่อทำการวัดความเข้มข้นของแก๊สที่มีปริมาณน้อยจึงมีการเกาะบริเวณผิวของทินออกไซด์น้อย ทำให้ค่าความต้านทานลดลงน้อย จึงได้ค่าแรงดันไฟฟ้าที่น้อยนั่นเอง



รูปที่ 2.7 ทินออกไซด์ในแก๊สเซนเซอร์ MQ-135

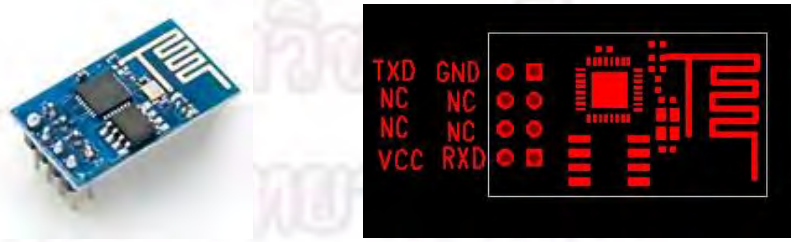
ตัวอย่างการใช้งาน MQ-135 กับ Arduino



รูปที่ 2.8 การเชื่อมต่อระหว่างโมดูลแก๊สเซนเซอร์กับไมโครคอนโทรลเลอร์

2.1.2.2 โมดูลไวร์เลส ยี่ห้อ ESP8266 รุ่น 01

โมดูลนี้ใช้สำหรับการส่งผ่านข้อมูลผ่านระบบการเชื่อมต่อไร้สายโดยมีองค์กร Institute of Electrical and Electronics Engineers หรือเรียกกันว่า IEEE เป็นผู้กำกับมาตรฐานของอิเล็กทรอนิกส์ต่าง ๆ ซึ่งโมดูลนี้ถูกควบคุมมาตรฐานด้วย IEEE 802.11 b/g/n กล่าวคือเป็นมาตรฐานที่ใช้สำหรับการส่งสัญญาณที่ความถี่ 2.4 GHz หน้าที่ของโมดูลนี้จึงเป็นอุปกรณ์ที่ใช้สำหรับการส่งผ่านข้อมูลที่ความถี่ 2.4 GHz เป็นความถี่มาตรฐานที่ใช้ในปัจจุบันสำหรับการปล่อยคลื่นความถี่ไวร์เลส (Wireless) หรือ Wi-Fi โดยสมาร์ทโฟน และคอมพิวเตอร์นั้นใช้มาตรฐาน IEEE 802.11 b/g/n ในการรับส่งข้อมูลเช่นกัน ทำให้โมดูลนี้สามารถติดต่อกับโมดูลชนิดเดียวกันสมาร์ทโฟน คอมพิวเตอร์ และอุปกรณ์อื่น ๆ ที่มีการใช้สัญญาณไวร์เลสได้นั่นเอง

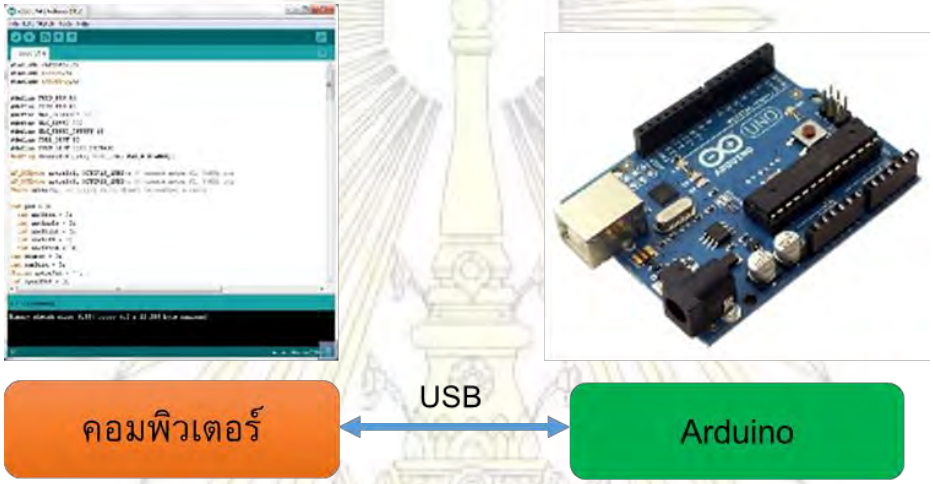


รูปที่ 2.9 ตัวอย่างโมดูลไวร์เลส ขนาดประมาณ 14.3 × 24.8 มิลลิเมตรและพอร์ตที่เชื่อมต่อ

2.2 ขั้นตอนในการเขียนชุดคำสั่งเพื่อให้ไมโครคอนโทรลเลอร์ Arduino ทำงาน

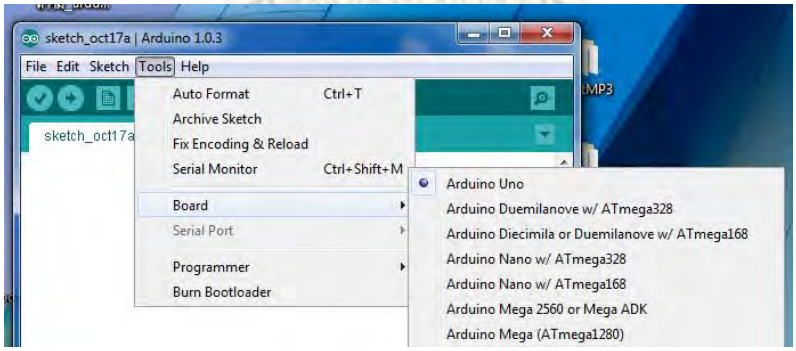
1. ขั้นตอนแรกนั้นจะต้องดาวน์โหลดโปรแกรม Arduino IDE ผ่านคอมพิวเตอร์ จาก http://www.arduino.cc

2. ทำการติดตั้งโปรแกรม Arduino IDE หลังจากนั้นให้ทำการเชื่อมต่อสาย USB ระหว่างช่อง USB ของคอมพิวเตอร์และพอร์ต USB ของไมโครคอนโทรลเลอร์ ซึ่งแสดงความสัมพันธ์ได้ดังรูปที่ 2.9



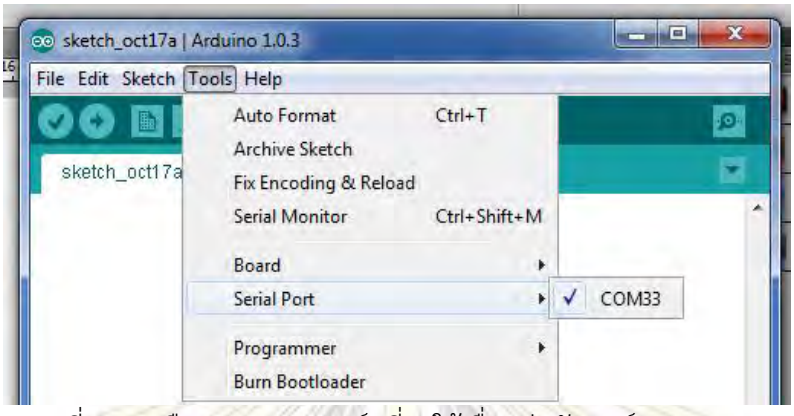
รูปที่ 2.10 การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์ Arduino กับคอมพิวเตอร์ผ่านสาย USB

3. เมื่อเข้าสู่โปรแกรม Arduino IDE แล้วให้ผู้ใช้ทำการเลือกรุ่นของบอร์ด Arduino ที่ใช้ และหมายเลขของพอร์ตของพิวเตอร์ที่จะใช้เชื่อมต่อกับ Arduino ดังรูปที่ 2.10 และ 2.11



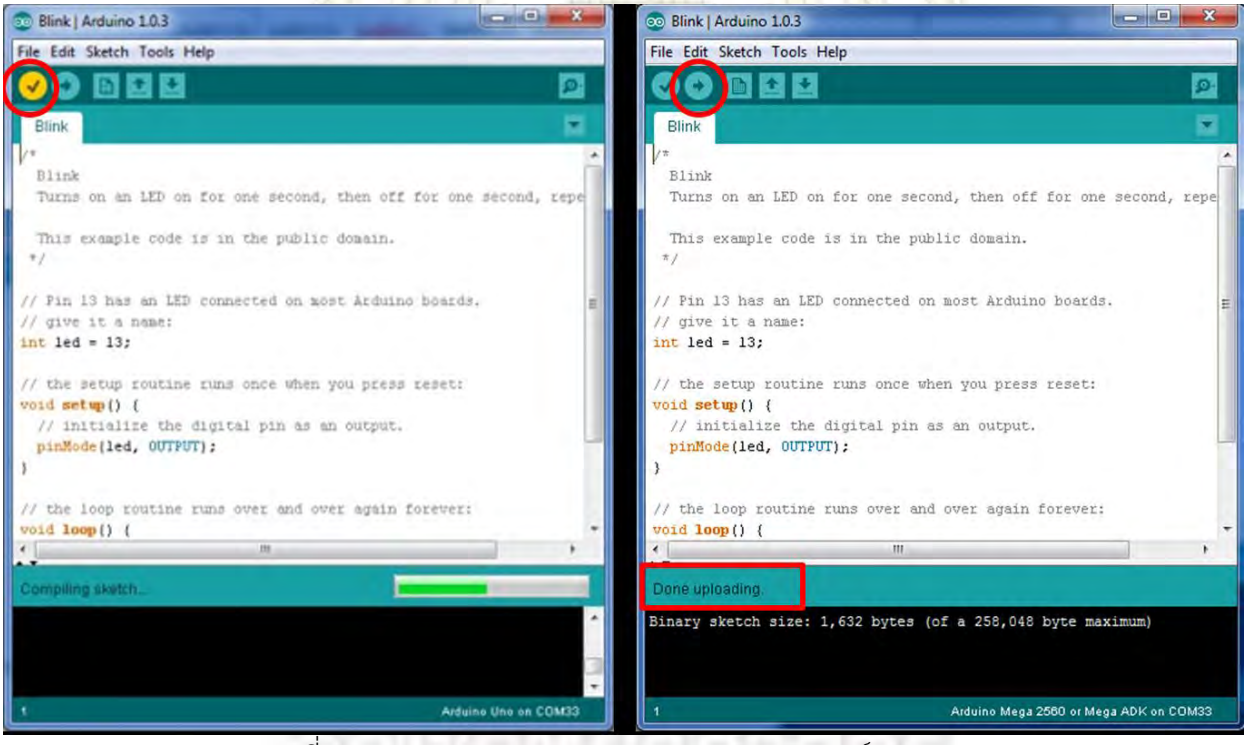
รูปที่ 2.11 เลือกรุ่นบอร์ด Arduino ที่ต้องการ upload

คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 2.12 เลือกหมายเลขพอร์ตที่จะใช้เชื่อมต่อกับบอร์ด Arduino

4. เมื่อเขียนชุดคำสั่งเสร็จเรียบร้อยให้ทำการตรวจเช็คความถูกต้องของไวยากรณ์ที่เขียนโดยการกดปุ่ม verify จากนั้นกดปุ่ม upload ชุดคำสั่งไปยังบอร์ด Arduino ผ่านสาย USB เมื่อ upload สำเร็จจะปรากฏข้อความแถบข้างล่าง “Done Uploading” หลังจากนั้นบอร์ดก็จะเริ่มทำงานตามชุดคำสั่งที่เขียนขึ้นมา ดังรูปที่ 2.12



รูปที่ 2.13 การกด verify และการ upload ลงบนบอร์ด Arduino

จุฬาลงกรณ์มหาวิทยาลัย

2.3 รูปแบบการเขียนชุดคำสั่งลงบนไมโครคอนโทรลเลอร์ยี่ห้อ Arduino

เราจะใช้การเขียนชุดคำสั่งผ่านคอมพิวเตอร์โดยใช้โปรแกรม Arduino IDE ดังรูปที่ 2.11 จากรูปนั้นเราจะเห็นส่วนแถบเครื่องมือในรูปที่ 2.13

ตรวจเช็คไวยากรณ์ที่เขียน (Verify)

ส่งข้อมูลเข้าสู่ไมโครคอนโทรลเลอร์ (Upload)

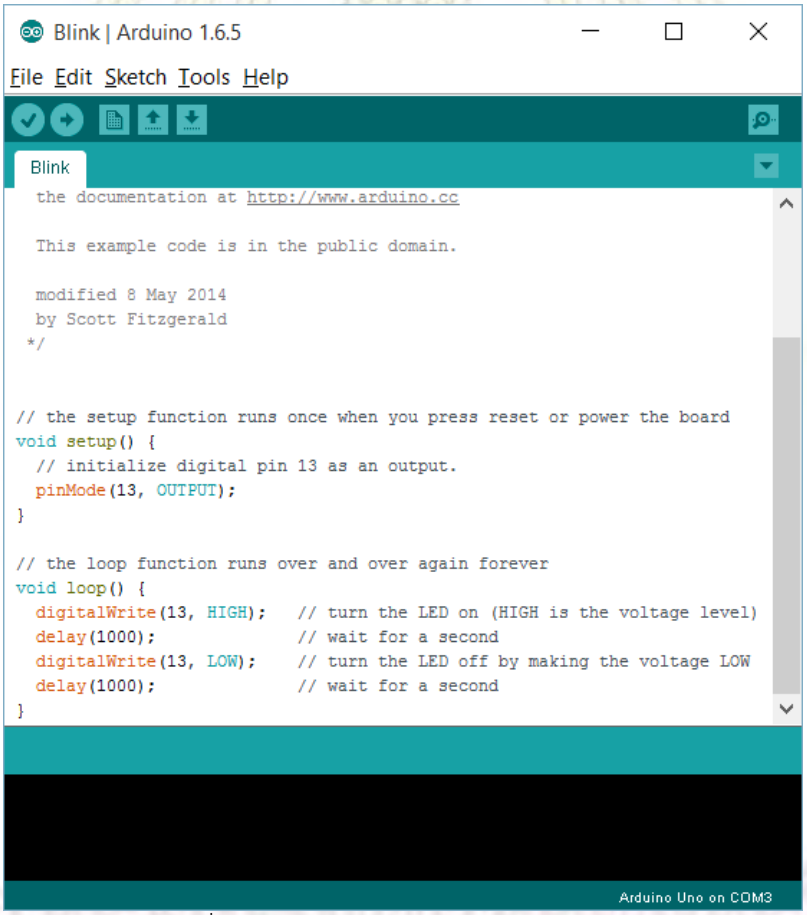
เปิดหน้าต่างใหม่ (New)

เปิดไฟล์ที่บันทึกไว้ (Open)

บันทึกไฟล์ (Save)

ส่วนแสดงผลทางหน้าจอ (Serial Monitor)

รูปที่ 2.14 แถบเครื่องมือของโปรแกรม Arduino IDE



รูปที่ 2.15 หน้าต่างโปรแกรม Arduino IDE

จากรูปที่ 2.14 จะเห็นว่าส่วนของโปรแกรมนั้นมีพื้นที่ให้พิมพ์ข้อความ โดยพื้นที่สำหรับพิมพ์ข้อความลงไปนี้คือส่วนของชุดคำสั่งที่ผู้ใช้จะทำการเขียนเพื่อสั่งลำดับการทำงานให้กับไมโครคอนโทรลเลอร์ ในส่วนนี้จะต้องใช้ภาษาซี เพื่อใช้สำหรับสั่งงาน การทำงานของโปรแกรมนี้จะทำการจัดแบ่งรูปแบบโครงสร้างของการเขียนชุดคำสั่งออกเป็นส่วนย่อย ๆ โดยแต่ละส่วนจะเรียกว่า “ฟังก์ชัน” และเมื่อฟังก์ชันมารวมกันโดยมีฟังก์ชันหลัก จะเรียกว่าชุดคำสั่ง โครงสร้างของการเขียนชุดคำสั่งออกมานั้นสามารถเขียนจำนวนฟังก์ชันเท่าไหร่ก็ได้ โดยจะต้องมีฟังก์ชันหลัก 2 ฟังก์ชัน คือ setup() และ loop()

การเขียนชุดคำสั่งแบ่งออกเป็นสามส่วนหลัก ๆ คือ

1. ส่วนของ Header เป็นส่วนที่อยู่ด้านบนสุดของการเขียนโปรแกรม จากภาพคือในส่วนที่เป็นข้อความตั้งแต่ “the documentation at http://www.arduino.cc” จนถึงด้านบนของข้อความ “void setup() {” ส่วนนี้ใช้สำหรับประกาศตัวแปร ค่าคงที่ต่าง ๆ ที่จะใช้ในโปรแกรม รวมไปถึงสามารถสร้างฟังก์ชันอื่นได้ในบริเวณนี้และบริเวณด้านล่างสุดได้อีกด้วย

2. ส่วนของฟังก์ชัน setup() ในส่วนนี้มีอาณาบริเวณภายในปีกกา { } ที่ติดกับส่วนนี้เท่านั้น ส่วนนี้ใช้สำหรับคำสั่งที่ต้องการให้ทำงานเพียงรอบเดียวเท่านั้น ส่วนใหญ่จะเป็นคำสั่งที่ใช้ในการตั้งค่าอุปกรณ์ว่าจะมีการเชื่อมต่อกับโมดูลใดและที่พอร์ตใด เป็นต้น

3. ส่วนของฟังก์ชัน loop() ส่วนนี้จะมีอาณาเขตบริเวณการทำงานภายในปีกกา { } เช่นเดียวกับฟังก์ชัน setup() เป็นส่วนหลักของที่จะให้ไมโครคอนโทรลเลอร์ทำงาน โดยจะมีการทำงานภายในฟังก์ชันนี้ซ้ำไปเรื่อย ๆ จนกว่าจะมีการปิดเครื่องหรือกดปุ่มเริ่มทำงานใหม่อีกครั้ง

2.4 โครงสร้างการเขียนชุดคำสั่งภาษาซีสำหรับไมโครคอนโทรลเลอร์ Arduino

ในการเขียนชุดคำสั่งภาษาซีลงบนไมโครคอนโทรลเลอร์ Arduino จะแบ่งออกเป็นสามส่วนหลัก ๆ คือ โครงสร้าง ตัวแปรและค่าคงที่ ฟังก์ชัน สามารถอธิบายได้ดังนี้

2.4.1 โครงสร้าง (Structure)

ในการเขียนภาษาซีจะใช้หลักการ และแนวคิดเหมือนกันไม่ว่าจะเขียนบนระบบใด ๆ และส่วนสำคัญในการตัดสินใจในการทำคำสั่งนั้น ๆ ซึ่งเราจะใช้คำสั่งเพื่อใช้ตรวจสอบเงื่อนไข จากนั้นจึงเริ่มทำคำสั่งที่เราเขียนขึ้นต่อไป โดยคำสั่งเพื่อใช้ในการตรวจสอบเงื่อนไข ดังนี้

- if

การใช้งาน : if(เงื่อนไขที่ต้องการ) { ขอบเขตการทำงานเมื่อเงื่อนไขถูกต้อง }

ตัวอย่างการใช้งาน : ถ้า x =50 จะได้ว่า

if(x>=50) { print “Hello” }

คำอธิบาย : เนื่องจากมีการตรวจสอบเงื่อนไขว่า x มากกว่าหรือเท่ากับ 50 ซึ่งเป็นจริงเนื่องมาจาก x = 50 ดังนั้นจะพิมพ์คำว่า Hello ออกมา

- if...else

การใช้งาน : if(เงื่อนไขที่ต้องการ) { ขอบเขตการทำงานเมื่อเงื่อนไขถูกต้อง }
else { ขอบเขตการทำงานเมื่อเงื่อนไขไม่ถูกต้อง }

ตัวอย่างการใช้งาน : ถ้า x =50 จะได้ว่า

if(x>=100) { print “Hello” }
else { print “World” }

คำอธิบาย : เนื่องจากมีการตรวจสอบเงื่อนไขว่า x มากกว่าหรือเท่ากับ 100 ซึ่งเป็นเท็จเนื่องมาจาก x = 50 ดังนั้นจะพิมพ์คำว่า World ออกมา

- switch case

การใช้งาน : switch(ตัวแปร) { case “ค่าของตัวแปร” : { ขอบเขตการทำงานเมื่อตัวแปรมีค่าตามที่กำหนดไว้ break(); } }

ตัวอย่างการใช้งาน : ถ้า x = 50

switch(x) { case 25 : { print “Hello” break(); }
case 50 : { print “World” break(); } }

คำอธิบาย : เริ่มจาก x ถูกคำสั่ง switch มาตรวจสอบว่ามีค่าตรงกับ case ใดบ้าง ในที่นี้ x = 50 ดังนั้นโปรแกรมจึงตัดสินใจทำตามคำสั่งใน case 50 โดยพิมพ์คำว่า World ออกมา และ break เพื่อออกจากการตรวจสอบค่า x (เพราะว่าถูกตรวจสอบแล้ว)

และสำหรับคำสั่งที่ใช้เพื่อให้ไมโครคอนโทรลเลอร์ทำงานแบบไม่รู้จบ (Loop) เราก็จะใช้คำสั่งต่อไปนี้ร่วมกับการตรวจสอบเงื่อนไข ได้แก่คำสั่ง while, for, do...while คำสั่งสำหรับตรวจสอบเงื่อนไขเหล่านี้จะใช้ควบคู่กับ เครื่องหมายเปรียบเทียบ (Comparative operator) และเครื่องหมายทางตรรกะ (Logical operator) สำหรับกำหนดเงื่อนไขการทำงานให้กับคำสั่ง ได้แก่

== มีความหมายว่า เท่ากับ

!= มีความหมายว่า ไม่เท่ากับ

<	มีความหมายว่า	น้อยกว่า
>	มีความหมายว่า	มากกว่า
<=	มีความหมายว่า	น้อยกว่าหรือเท่ากับ
>=	มีความหมายว่า	มากกว่าหรือเท่ากับ
&&	มีความหมายว่า	และ
||	มีความหมายว่า	หรือ
!	มีความหมายว่า	ไม่

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

2.4.2 ตัวแปรและค่าคงที่

ในภาษาซีตัวแปรหมายถึง กลุ่มของอักขระที่รวมกันเป็นชื่อเพื่อใช้แทนค่าของข้อมูลที่ต้องการจะอ้างอิง เนื่องจากการทำงานเมื่อเราทำการเขียนชุดคำสั่งนั้น ชุดคำสั่งจะถูกแปลงให้เป็นเลขฐาน 2 แล้วนำไปเก็บตามที่อยู่ของหน่วยความจำ การสร้างตัวแปรขึ้นมาทำให้เราสามารถใช้ข้อมูลนั้น ๆ ได้โดยใช้ตัวแปรเก็บที่อยู่ของข้อมูลเหล่านั้นไว้ ตัวแปรจึงมีความสำคัญในการเขียนชุดคำสั่ง โดยลักษณะของข้อมูลนั้นมีข้อมูลที่สามรถเปลี่ยนแปลงค่าได้ (Variable) และเป็นค่าคงที่ซึ่งไม่สามารถเปลี่ยนแปลงข้อมูลได้ (Constant) ดังนั้นตัวแปรจึงสามารถแบ่งออกเป็นสองส่วนหลัก ๆ คือตัวแปรที่เก็บข้อมูลที่สามารถเปลี่ยนแปลงได้ และตัวแปรที่เก็บข้อมูลที่เป็นค่าคงที่ โดยตัวแปรบางส่วนจะถูกระบบเป็นผู้สร้างไว้อยู่แล้วส่วนหนึ่ง ซึ่งผู้ใช้ไม่สามารถตั้งชื่อตัวแปรซ้ำกับที่ระบบสร้างขึ้นมาไว้แล้ว เราเรียกตัวแปรที่ระบบสร้างไว้แล้วว่า “คำสงวน” (Reserve Word) ตัวแปรที่ระบบสร้างไว้แล้วนั้นได้แก่ ชื่อคำสั่ง และชื่อฟังก์ชันภายในต่าง ๆ ดังตารางที่ 2.1 – 2.4

ตารางที่ 2.1 คำสงวนสำหรับตัวแปร ที่เป็นค่าคงที่

HIGH	LOW	INPUT	OUTPUT	SERIAL	DISPLAY	PI	HALF_PI	TWO_PI
LSBFIRST	MSBFIRST	CHANGE	FALLING	RISING	"false"	"true"	"null"	

ตารางที่ 2.2 คำสงวนสำหรับตัวแปร ที่เป็นค่าคงที่ของพอร์ตในบอร์ด Arduino

DDRB	PINB	PORTB	PB0	PB1	PB2	PB3	PB4	PB5
PB6	PB7	DDRC	PINC	PORTC	PC0	PC1	PC2	PC3
PC4	PC5	PC6	PC7	DDRD	PIND	PORTD	PD0	PD1
PD2	PD3	PD4	PD5	PD6	PD7			

ตารางที่ 2.3 คำสงวนสำหรับตัวแปร ที่เป็นชนิดข้อมูล

boolean	byte	char	class	default	do	double	int	long
private	protected	public	return	short	signed	static	switch	throw
try	unsigned	void	boolean	case	ceil	char	class	constrain
cos	delay	microseconds		else	exp	float	floor	for
int	log	loop	max	mills	min	new	return	Setup
sin	sq	sqrt	tan	this	void	while	begin	read
write	print	println	available	digitalWrite		digitalRead		pinMode
analogWrite		attachInterrupts		detachInterrupts		beginSerial		printHex
serialWrite		serialRead		serialAvailable		printString		shiftOut
printInteger		printBinary		printNewline		printOctal		pulseIn

นอกจากนี้ตัวแปรสามารถจำแนกชนิดของตัวแปรได้ 5 ชนิดด้วยกัน ซึ่งแต่ละชนิดของตัวแปรมีคุณสมบัติในการเก็บข้อมูลที่มีรูปแบบที่แตกต่างกันดังและใช้พื้นที่ในการใช้ในการเก็บข้อมูลในหน่วยความจำ ตารางที่ 2.4

ตารางที่ 2.4 คุณสมบัติของตัวแปรในภาษาซี

ชนิดตัวแปร	จำนวนบิต	ช่วงของข้อมูลที่เก็บได้
char	8	-128 ถึง +127
int	16	-32,768 ถึง +32,767
float	32	$3.4x10^{-38}$ ถึง $3.4x10^{38}$
double	64	$1.7x10^{-308}$ ถึง $1.7x10^{308}$
void	0	ไม่มีค่า

2.4.3 ฟังก์ชัน

นอกจากส่วนของโครงสร้าง ตัวแปรและค่าคงที่แล้ว ส่วนของฟังก์ชันนั้นนับเป็นส่วนสำคัญสำหรับผู้ใช้งาน เพราะฟังก์ชันนั้นเป็นการนำคำสั่งซึ่งมาจากตัวแปร รวมเอาไว้เป็นหมวดหมู่เพื่อให้ผู้ใชงานสามารถเรียกใช้งานคำสั่งเหล่านี้โดยใช้ฟังก์ชันแล้วนำมาใช้ในชุดคำสั่งได้ทันที ชุดคำสั่งหลักๆ ในบอร์ด Arduino ได้แก่

1. กลุ่มคำสั่งสำหรับใช้งาน Digital I/O หรือ Analog I/O

- pinMode() ใช้สำหรับกำหนดการทำงานของพอร์ต I/O เป็นรูปแบบ input หรือ output
- digitalWrite() ใช้สำหรับกำหนดสภาวะทางตรรกะของพอร์ต I/O เป็น 0(LOW) หรือ 1(HIGH)
- digitalRead() ใช้สำหรับอ่านสภาวะทางตรรกะจากพอร์ตเป้าหมายซึ่งมีค่าเป็นไปได้คือ 0,1
- analogRead() ใช้สำหรับอ่านข้อมูลจากพอร์ตเป้าหมายโดยมีค่าตั้งแต่ -1023 ถึง 1024
- analogWrite() ใช้สำหรับการส่งข้อมูลจากพอร์ตเป้าหมายโดยมีค่าตั้งแต่ -1023 ถึง 1024

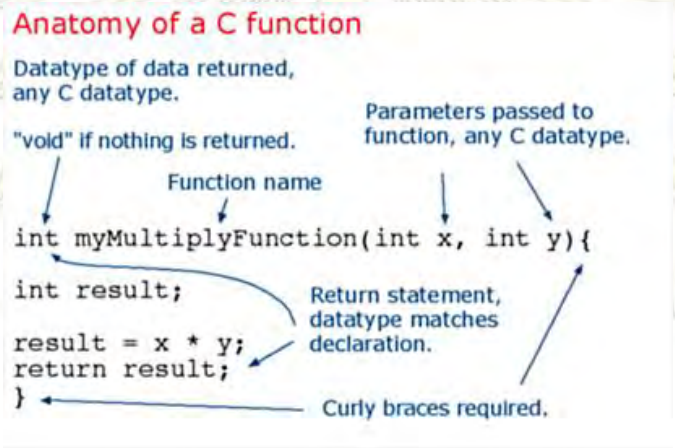
2. กลุ่มคำสั่งสำหรับหน่วงเวลา (Time)

- miils() ใช้สำหรับการนับเวลามีหน่วยเป็นไมโครวินาที
- delay() ใช้สำหรับหน่วงเวลาในหน่วยไมโครวินาที

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

3. กลุ่มคำสั่งการสื่อสารอนุกรม (Serial Communication)

- Serial.begin() ใช้สำหรับกำหนดความเร็วในการเชื่อมต่อระหว่างบอร์ดและคอมพิวเตอร์
- Serial.available() ใช้ตรวจสอบว่ามีข้อมูลถูกส่งมายังบอร์ด Arduino หรือไม่
- Serial.read() ใช้สำหรับอ่านข้อมูลที่ถูกส่งมายังบอร์ด Arduino
- Serial.flush() ใช้สำหรับล้างข้อมูลที่ถูกส่งเข้ามาในบอร์ด Arduino
- Serial.print ใช้สำหรับการพิมพ์ข้อมูลของ Arduino สู่พอร์ตเป้าหมาย
- Serial.println ใช้สำหรับการพิมพ์ข้อมูลและเพิ่มส่วนของการขึ้นบรรทัดใหม่ของ Arduino สู่พอร์ตเป้าหมาย



รูปที่ 2.16 โครงสร้างของฟังก์ชันในภาษา C สำหรับบอร์ด Arduino

```
void setup(){
  Serial.begin(9600);
}

void loop() {
  int i = 2;
  int j = 3;
  int k;

  k = myMultiplyFunction(i, j); // k now contains 6
  Serial.println(k);
  delay(500);
}

int myMultiplyFunction(int x, int y){
  int result;
  result = x * y;
  return result;
}
```

รูปที่ 2.17 แสดงตัวอย่างชุดคำสั่งที่มีส่วนของโครงสร้าง ตัวแปร และฟังก์ชัน

2.5 จุดกระจายสัญญาณอินเตอร์เน็ตไร้สาย (Wi-Fi Hotspot)

Wi-Fi Hotspot คือ จุดกระจายสัญญาณอินเตอร์เน็ตไร้สาย (Wi-Fi) ซึ่งอุปกรณ์ที่ใช้ในการกระจายสัญญาณ Wi-Fi นั้นเราจะเรียกว่าเราเตอร์โมเด็ม เป็นอุปกรณ์ที่ต่อกับสายแลนแบบ Ethernet เป็นการเชื่อมต่อโดยตรงกับสายสัญญาณอินเตอร์เน็ต ความแรงของจุดกระจายสัญญาณอินเตอร์เน็ตไร้สาย (Wi-Fi Hotspot) จะขึ้นอยู่กับกำลังส่งของอุปกรณ์ในการส่งสัญญาณ สภาพแวดล้อมภายนอก เช่น สิ่งที่บดบังสัญญาณ สภาพอากาศ เป็นต้น ด้วยสาเหตุนี้เราถึงได้เห็น Wi-Fi Hotspot อยู่บนตึกสูงหรืออยู่บนเสาที่มีความสูงที่ไม่มีอาคารบดบัง เพื่อให้กระจายสัญญาณอินเตอร์เน็ตไร้สายๆได้ไกลมากยิ่งขึ้น

รูปที่ 2.18 การใช้งานของ Wi-Fi Hotspot [8]

Wi-Fi Hotspot ทำหน้าที่อะไร

Wi-Fi Hotspot มีหน้าที่ที่จะต้องคอยส่งสัญญาณอินเตอร์เน็ตไร้สาย ไปยังอุปกรณ์อิเล็กทรอนิกส์ที่รองรับอาทิ โทรศัพท์มือถือ คอมพิวเตอร์ และแท็บเล็ต หรือจะเป็นอุปกรณ์อื่นๆที่รองรับระบบ Wi-Fi ก็ได้เช่นกัน ซึ่งจุดกระจายสัญญาณอินเตอร์เน็ตไร้สาย หรือ Wi-Fi Hotspot จะคอยส่งสัญญาณอินเตอร์เน็ตไร้สายให้กับผู้ที่ใช้งานที่อยู่ในรัศมีที่สามารถกระจายสัญญาณได้ ระดับความแรงก็ขึ้นอยู่กับ ตัวอุปกรณ์อิเล็กทรอนิกส์ที่เป็นเครื่องรับสัญญาณ Wi-Fi อยู่ใกล้กับจุดกระจายสัญญาณมากน้อยแค่ไหน แน่นอนว่าอุปกรณ์อิเล็กทรอนิกส์ที่อยู่ใกล้ Wi-Fi Hotspot ก็จะสามารถส่งถ่ายข้อมูลได้เร็วกว่า อุปกรณ์อิเล็กทรอนิกส์ที่อยู่ไกลกว่า

ประโยชน์ของ Wi-Fi Hotspot

ประโยชน์ของ Wi-Fi Hotspot นั้นมีมากมาย สามารถแจกแจงออกมาเป็นข้อๆ ดังนี้

1. Wi-Fi Hotspot สามารถทำให้การเข้าถึงอินเตอร์เน็ตเป็นเรื่องที่สะดวกสบายขึ้น และมีค่าบริการในการใช้งานถูกลง ซึ่งเราสามารถเลือกค่าบริการการใช้งานสัญญาณอินเตอร์เน็ตไร้สาย (Wi-Fi) ได้ทั้งแบบชั่วโมง หรือจะเป็นตามจำนวนข้อมูลหรือตามโปรโมชั่นของผู้ให้บริการก็ได้เช่นกัน

2. Wi-Fi Hotspot ยังช่วยเพิ่มศักยภาพในการแข่งขันธุรกิจได้เป็นอย่างดี ตัวอย่างที่เห็นได้ชัดเจนก็คือ ร้านกาแฟและร้านอาหาร ที่ติดตั้ง Wi-Fi Hotspot ในร้านจะช่วยดึงดูดลูกค้าเข้าร้านมากยิ่งขึ้น เปรียบได้เหมือนกับเป็นบริการเสริมให้กับลูกค้าที่มาใช้บริการ

3. Wi-Fi Hotspot ทำให้การรับส่งอีเมล์ ดาวน์โหลดข้อมูลได้อย่างรวดเร็ว ทำให้การทำงานเป็นไปอย่างต่อเนื่อง ขอเพียงเราอยู่ในจุดที่มี Wi-Fi Hotspot เราก็สามารถที่จะทำงานได้ทุกที่ไม่จำเป็นต้องอยู่ที่ Office อย่างเดียวอีกต่อไปแล้ว

4. Wi-Fi Hotspot ทำให้การทำงานในสถานที่ต่าง ๆในสะดวกสบายมากยิ่งขึ้นเพราะว่าไม่ต้องต่อสายสัญญาณกับอุปกรณ์อิเล็กทรอนิกส์แบบในสมัยก่อนอีกแล้ว

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3
อุปกรณ์ การออกแบบระบบ และวิธีการทดลอง

3.1 อุปกรณ์ที่ใช้ในการทดลอง

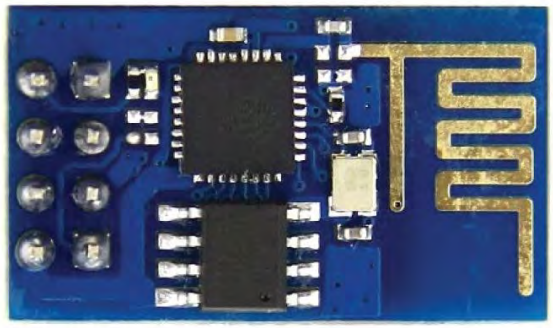
1. ไมโครคอนโทรลเลอร์ยี่ห้อ Arduino รุ่น UNO R3 (รูปที่ 3.1)
2. โมดูลแก๊สเซนเซอร์ รุ่น MQ-135 (รูปที่ 3.2)
3. โมดูลไวร์เลส รุ่น ESP8266 (รูปที่ 3.3)
4. สายไฟจัมเปอร์สำหรับเชื่อมต่อ ขั้ว ผู้-ผู้ เมีย-เมีย ระหว่างบอร์ด (รูปที่ 3.4)
5. สาย USB สำหรับเชื่อมต่อระหว่างคอมพิวเตอร์และไมโครคอนโทรลเลอร์ (รูปที่ 3.5)
6. ถ่านไฟฉายแบบชาร์จได้ ขนาด 9 โวลต์ (รูปที่ 3.6)
7. รางถ่านที่มีหัวเชื่อมต่อเข้ากับไมโครคอนโทรลเลอร์ (รูปที่ 3.7)
8. คอมพิวเตอร์ที่มีระบบปฏิบัติการตั้งแต่ Window XP ขึ้นไป
9. เครื่องชั่งสาร
10. กระจกนาฬิกา
11. ขวดรูปชมพู่ปริมาตร 125 mL
11. ไมโครปิเปต

3.2 สารเคมีที่ใช้ในการทดลอง

1. โซเดียมไบคาร์บอเนต
2. กรดไฮโดรคลอริก

รูปที่ 3.1 ไมโครคอนโทรลเลอร์ Arduino

รูปที่ 3.2 โมดูลแก๊สเซนเซอร์ รุ่น MQ-135



รูปที่ 3.3 โมดูลไวเรสรุ่น ESP8266

รูปที่ 3.4 สายไฟจัมเปอร์สำหรับเชื่อมต่อ

รูปที่ 3.5 สาย USB สำหรับ Arduino

รูปที่ 3.6 ถ่านไฟฉายแบบชาร์จได้ขนาด 9 โวลต์

รูปที่ 3.7 รางถ่านที่มีหัวเชื่อมต่อกับไมโครคอนโทรลเลอร์

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

3.3 การออกแบบระบบ

3.3.1 การรวบรวมข้อมูล

การเก็บข้อมูลสำหรับนำมาวิเคราะห์และออกแบบโปรแกรมสำหรับเครื่องมือวัดปริมาณแก๊สโดยใช้ไมโครคอนโทรลเลอร์นั้นผู้พัฒนาได้เก็บรวบรวมข้อมูลต่าง ๆ ดังนี้

1. การวิเคราะห์จากระบบเดิม โดยใช้การทำงานของเครื่องมือต้นแบบจาก เครื่องตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์ จาก www.co2meter.com ดังรูปที่ 3.8 จะเห็นว่า เครื่องมือมีราคาแพง (129$ หรือประมาณ 4500 บาท) และยังไม่สามารถควบคุมตัวเครื่องจากระยะไกลได้อีกด้วย

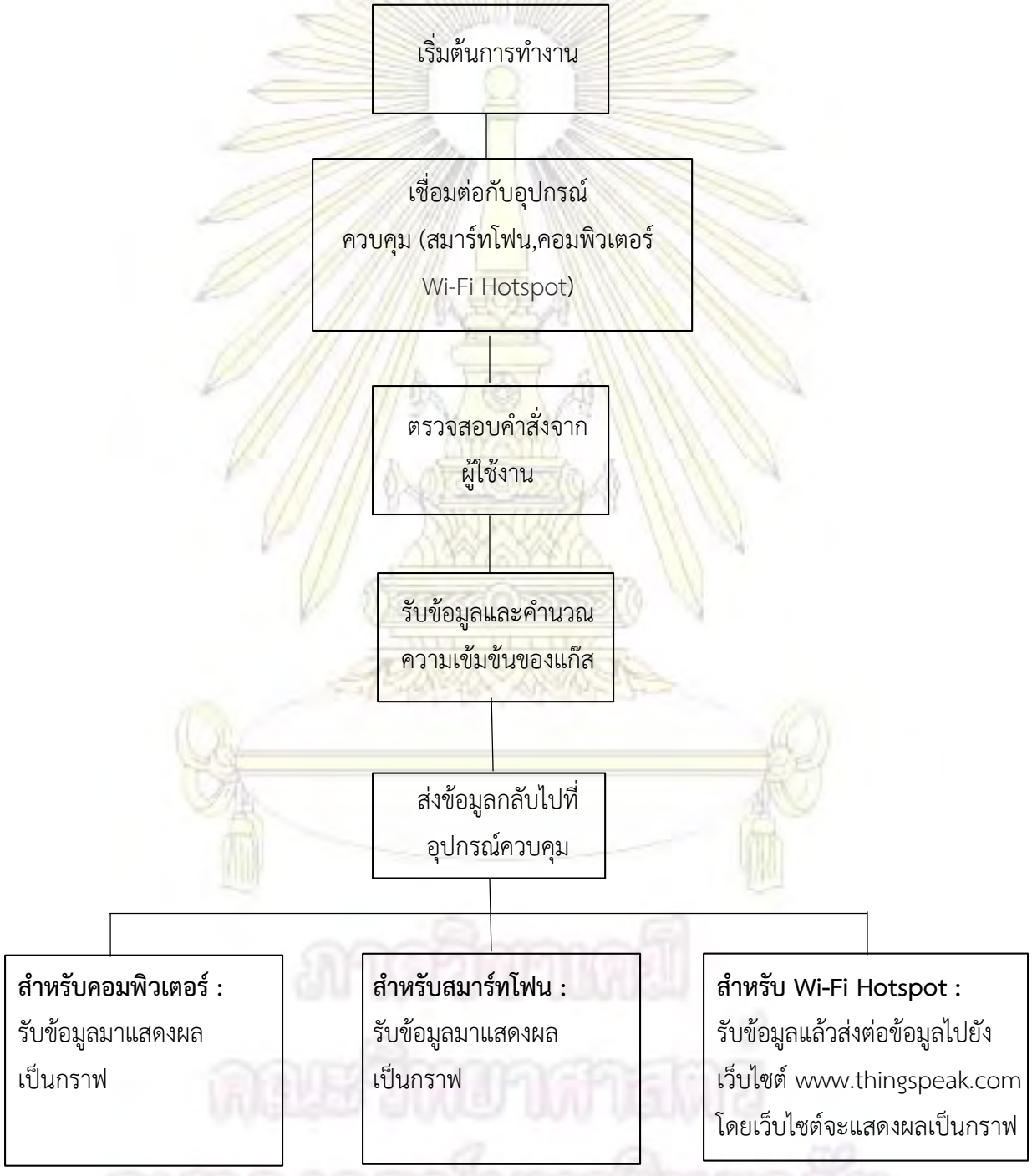
2. ความสามารถของระบบ เนื่องจากเครื่องต้นแบบจากรูปที่ 3.8 นั้นมีข้อจำกัดหลายอย่างในการใช้งานไม่ว่าจะเป็นในเรื่องของการไม่สามารถควบคุมจากระยะไกลได้ หรือมีราคาแพงทางผู้พัฒนาจึงได้กำหนดขีดความสามารถของระบบที่จะพัฒนาขึ้นใหม่ให้มีการใช้งานได้มากขึ้นดังนี้

1. โปรแกรมสำหรับเครื่องมือวัดปริมาณแก๊สโดยใช้ไมโครคอนโทรลเลอร์สามารถรายงานข้อมูลแบบทันที
2. ปรับปรุงให้สามารถใช้เครื่องมือวัดปริมาณแก๊สในแบบพกพาได้และลดต้นทุนในการสร้างเครื่องมือ
3. สามารถออกคำสั่งควบคุมเครื่องมือวัดปริมาณแก๊สจากระยะไกลได้

รูปที่ 3.8 เครื่องวัดปริมาณแก๊สคาร์บอนไดออกไซด์จาก www.co2meter.com

3.3.2 แผนภาพโครงสร้าง

การออกแบบโปรแกรมวัดปริมาณแก๊สสำหรับไมโครคอนโทรลเลอร์ Arduino ผู้พัฒนาได้กำหนดความสัมพันธ์ของฟังก์ชันและรายละเอียดของโปรแกรมที่จะทำงานทั้งหมดดังรูปที่ 3.9



รูปที่ 3.9 แผนภาพโครงสร้างของระบบสำหรับควบคุมอุปกรณ์ติดตามคุณภาพอากาศระยะไกลแบบเรียลไทม์

3.3.3 แผนภาพยูสเคส

ความสามารถของผู้ใช้งานที่สามารถกระทำได้บนระบบนั้น เป็นความสามารถที่ผู้ใช้สามารถอ่านข้อมูลหรือสามารถควบคุมอุปกรณ์สำหรับตัววัดปริมาณแก๊สได้ โดยผู้พัฒนาได้ทำการออกแบบโปรแกรมวัดปริมาณแก๊สสำหรับไมโครคอนโทรลเลอร์ Arduino โดยมีทางเลือกการใช้ระบบได้ดังรูปที่ 3.10

รูปที่ 3.10 แผนภาพยูสเคส ของระบบสำหรับควบคุมอุปกรณ์ติดตามคุณภาพอากาศระยะไกลแบบเรียลไทม์

Use Case Template

Name:	โปรแกรมควบคุมอุปกรณ์สำหรับตัววัดปริมาณแก๊ส
Participant Actor:	ผู้ใช้งาน
Entry Condition:	ผู้ใช้งานเปิดเครื่องมือโดยต่ออุปกรณ์เข้ากับแหล่งจ่ายไฟ
Flow of event:	1. ผู้ใช้งานกดปุ่มเพื่อตั้งค่าหรือส่งคำสั่งไปยังเครื่องมือ
	2. ผู้ใช้งานกดปุ่มระบบแสดงปริมาณแก๊สในรูปกราฟ
Exit Condition:	ผู้ใช้งานปิดสวิทซ์

Use Case Scenario

ผู้ใช้งานจะเริ่มใช้งานโดยทำการต่อแหล่งจ่ายไฟฟ้าเข้ากับเครื่องมือ จากนั้นโปรแกรมวัดปริมาณแก๊สจะทำงานอย่างอัตโนมัติ ซึ่งผู้ใช้งานสามารถออกคำสั่งกับอุปกรณ์ได้ว่าต้องการให้วัดทุก ๆ กี่นาที ซึ่งโปรแกรมจะแสดงผลในรูปของกราฟความสัมพันธ์ระหว่างความเข้มข้นของปริมาณแก๊สกับจำนวนครั้งที่ทำการวัดโดยสามารถอธิบายการทำงานของแต่ละระบบได้ดังตารางที่ 3.1-3.6

Use Case Description

ตารางที่ 3.1 ยูสเคส เชื่อมต่ออุปกรณ์ควบคุม

Use case name:	เชื่อมต่ออุปกรณ์ควบคุม
Participating actor:	ผู้ใช้งาน
Entry condition:	ผู้ใช้งานต่อแหล่งจ่ายไฟฟ้าเข้ากับเครื่อง
Flow of events:	1. ผู้ใช้งานเปิดเครื่องเพื่อให้อุปกรณ์เริ่มต้นทำงาน 2. เครื่องจะทำการเชื่อมต่อกับอุปกรณ์ควบคุม
Exit condition:	ผู้ใช้งานปิดเครื่อง

ตารางที่ 3.2 ยูสเคส รับข้อมูลการควบคุม

Use case name:	รับข้อมูลการควบคุม
Participating actor:	ผู้ใช้งาน
Entry condition:	ผู้ใช้งานต่อแหล่งจ่ายไฟฟ้าเข้ากับเครื่อง
Flow of events:	1. ผู้ใช้งานตั้งค่าเครื่องมือวัดปริมาณแก๊ส
Exit condition:	ผู้ใช้งานปิดเครื่อง

ตารางที่ 3.3 ยูสเคส รับข้อมูลจากโมดูลแก๊สเซนเซอร์

Use case name:	รับข้อมูลจากโมดูลแก๊สเซนเซอร์
Participating actor:	ระบบ
Entry condition:	ผู้ใช้งานต่อแหล่งจ่ายไฟฟ้าเข้ากับเครื่อง
Flow of events:	1. ระบบทำการอ่านข้อมูลจากโมดูลแก๊สเซนเซอร์
Exit condition:	ผู้ใช้งานปิดเครื่อง

ตารางที่ 3.4 ยูสเคส คำนวณปริมาณแก๊ส

Use case name:	คำนวณปริมาณแก๊ส
Participating actor:	ระบบ
Entry condition:	ผู้ใช้งานต่อแหล่งจ่ายไฟฟ้าเข้ากับเครื่อง
Flow of events:	1. เครื่องจะทำการคำนวณปริมาณแก๊สจากข้อมูลที่รับมาจากโมดูลแก๊สเซนเซอร์
Exit condition:	ผู้ใช้งานปิดเครื่อง

ตารางที่ 3.5 ยูสเคส ส่งข้อมูลปริมาณแก๊สไปยังอุปกรณ์ควบคุม

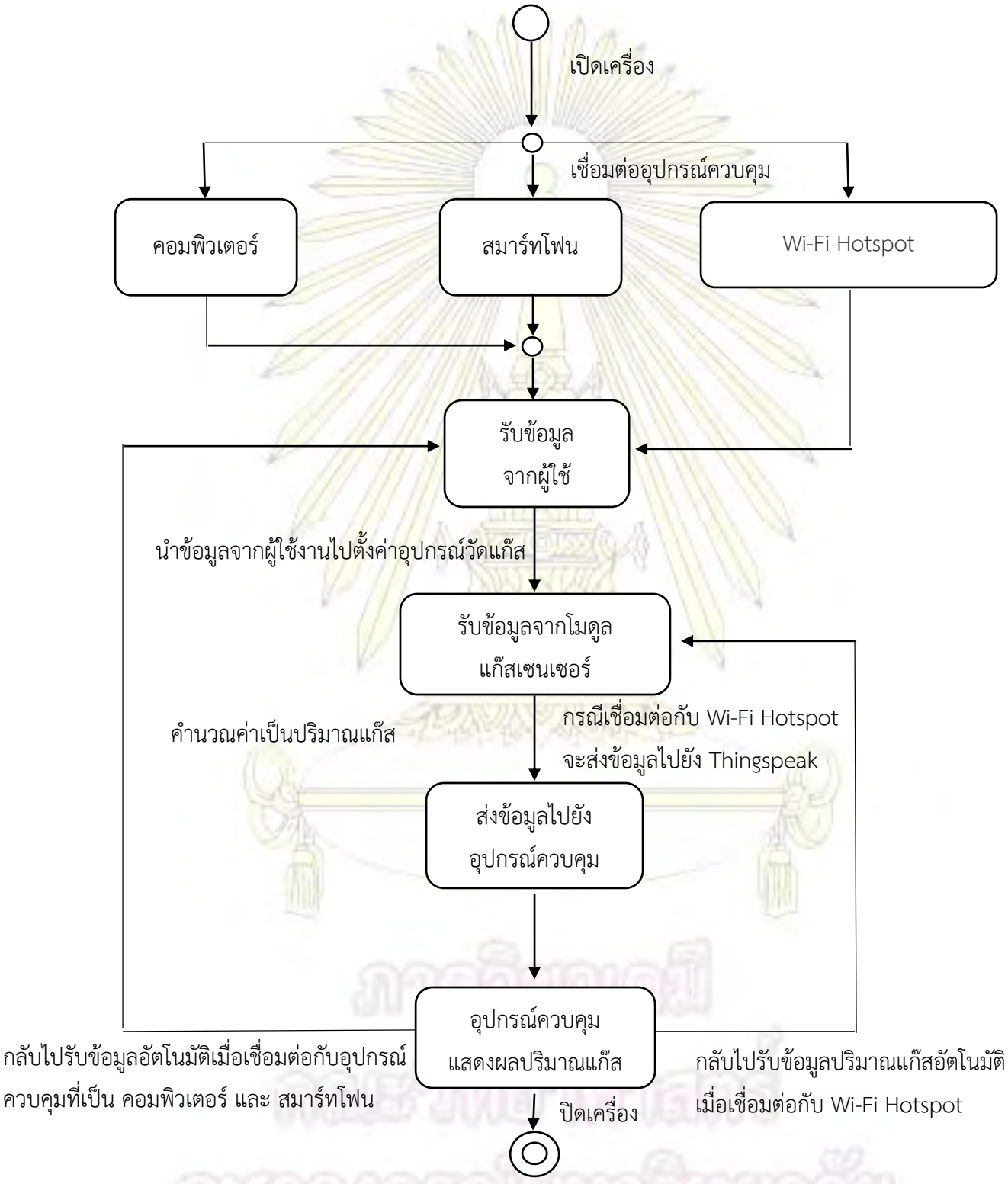
Use case name:	ส่งข้อมูลปริมาณแก๊สไปยังอุปกรณ์ควบคุม
Participating actor:	ระบบ
Entry condition:	ผู้ใช้งานต่อแหล่งจ่ายไฟฟ้าเข้ากับเครื่อง
Flow of events:	1. ระบบจะทำการส่งข้อมูลกลับไปยังอุปกรณ์ควบคุม
Exit condition:	ผู้ใช้งานปิดเครื่อง

ตารางที่ 3.6 ยูสเคส แสดงผล ปริมาณแก๊ส

Use case name:	แสดงผล ปริมาณแก๊ส
Participating actor:	ผู้ใช้งาน
Entry condition:	ผู้ใช้งานต่อแหล่งจ่ายไฟฟ้าเข้ากับเครื่อง
Flow of events:	1. ผู้ใช้งานดูค่าปริมาณแก๊สที่วัดได้ จากกราฟความสัมพันธ์ระหว่างปริมาณแก๊สและจำนวนครั้งที่วัด
Exit condition:	ผู้ใช้งานปิดเครื่อง

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

3.3.4 แผนภาพสถานะ



รูปที่ 3.11 แผนภาพสถานะของระบบควบคุมอุปกรณ์ติดตามคุณภาพอากาศระยะไกลแบบเรียลไทม์

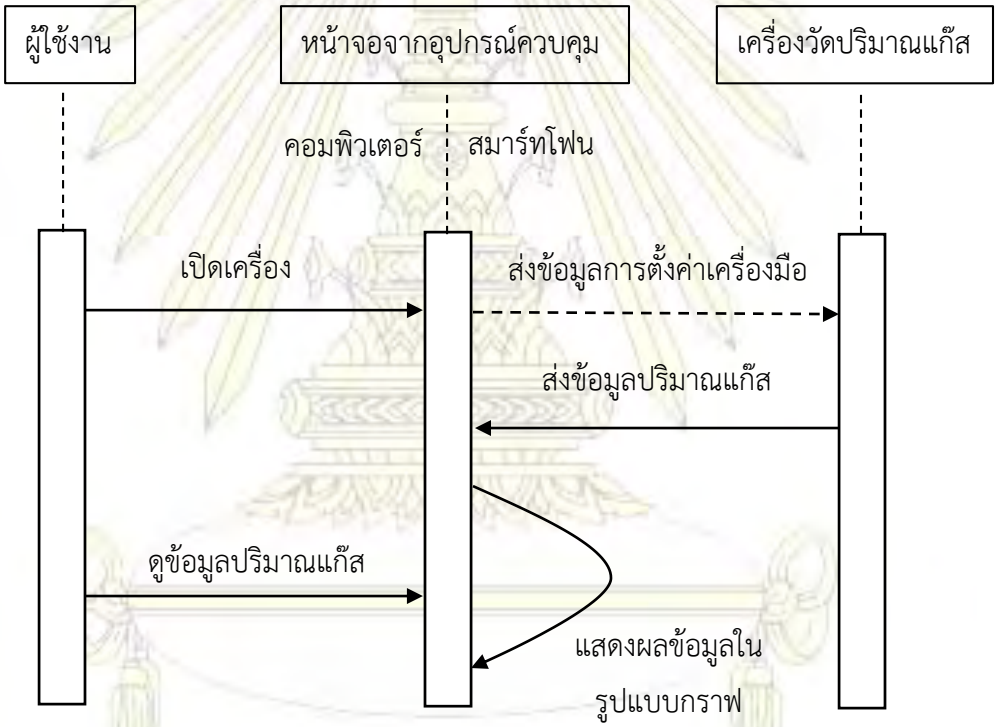
จากรูปที่ 3.11 แสดงแผนภาพสถานะ ใช้สำหรับบอกสถานะของเครื่องวัดแก๊สที่สภาวะต่าง ๆ เริ่มจากผู้ใช้ทำการต่อแหล่งจ่ายไฟฟ้าเข้ากับไมโครคอนโทรลเลอร์เพื่อเปิดเครื่อง เครื่องจะเริ่มค้นหาอุปกรณ์ควบคุมอัตโนมัติ เมื่อค้นหาสำเร็จจะเข้าสู่การรับคำสั่งโดยตรงจากอุปกรณ์ควบคุม (สำหรับคอมพิวเตอร์ และสมาร์ทโฟน) เครื่องวัดแก๊สจะอ่านค่าที่ได้จากโมดูลแก๊สเซนเซอร์แล้วคำนวณออกมาเป็นปริมาณความเข้มข้นของแก๊ส เมื่อคำนวณเสร็จก็จะทำการส่งข้อมูลกลับมาที่อุปกรณ์ควบคุมข้อมูลจะถูกแสดงผลเป็นกราฟความสัมพันธ์ระหว่างปริมาณแก๊สที่วัดได้กับจำนวนครั้งที่ทำการวัด หลังจากนั้นก็จะกลับไปเริ่มต้นรับข้อมูลจากผู้ใช้งานอีกครั้งแล้ววนไปเรื่อย ๆ จนกว่าจะได้รับคำสั่งจากผู้ใช้หรือปิดเครื่อง กรณีที่ไม่มีอุปกรณ์ควบคุม เครื่องตรวจวัดแก๊สจะทำการเชื่อมต่อกับ Wi-Fi Hotspot เพื่อเริ่มต้นการส่งมูลแบบทางเดียวในโหมดนี้จะไม่สามารถควบคุมจากระยะไกลได้ เครื่องจะทำการรับและคำนวณปริมาณแก๊สแล้วส่งไปยังอินเตอร์เน็ตที่เว็บไซต์ www.thingspeak.com เพื่อให้เว็บไซต์รับค่าแล้วนำไปแสดงผลเป็นกราฟความสัมพันธ์ในรูปแบบเดียวกันกับอุปกรณ์ควบคุมอื่น ๆ และปิดเครื่องเพื่อหยุดการทำงานของระบบ ซึ่งในกรณีนี้ใช้สำหรับตั้งเป็นสถานีตรวจวัดคุณภาพที่ไม่จำเป็นต้องเคลื่อนที่และต้องตรวจวัดปริมาณแก๊สอยู่เป็นประจำ

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

3.3.5 แผนภาพลำดับการทำงาน

จากรูป 3.12 แสดงรูปแผนลำดับโดยใช้อุปกรณ์ควบคุมเป็น คอมพิวเตอร์ และ สมาร์ทโฟน มีลำดับการใช้งานดังนี้

1. ผู้ใช้เปิดเครื่องเพื่อเริ่มต้นการใช้งาน
2. อุปกรณ์ควบคุมส่งข้อมูลการตั้งค่าไปยังเครื่องวัดปริมาณแก๊ส
3. เครื่องวัดปริมาณแก๊สรับการตั้งค่า (กรณีไม่ตั้งค่าใดๆ จะทำการส่งข้อมูลตลอดเวลา)
4. ส่งข้อมูลปริมาณแก๊สสู่อุปกรณ์ควบคุม
5. อุปกรณ์ควบคุมแสดงผลข้อมูลในรูปแบบกราฟความสัมพันธ์ระหว่างปริมาณแก๊สและจำนวนครั้งที่วัด
6. ผู้ใช้ดูข้อมูลที่แสดง

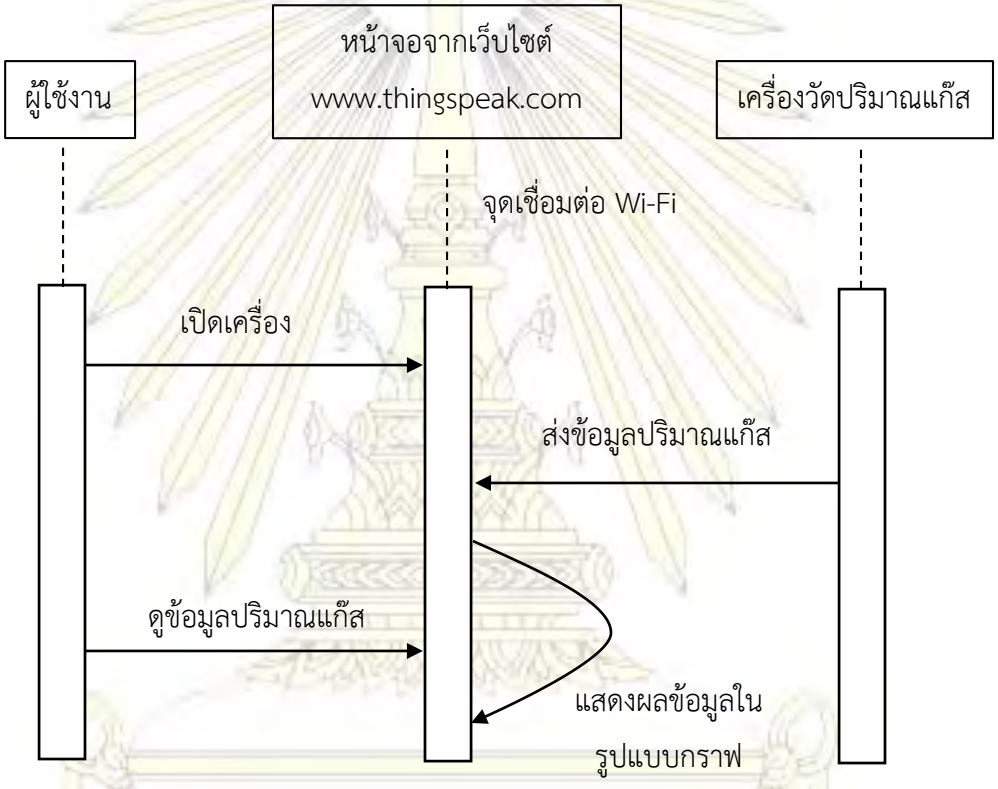


รูปที่ 3.12 แผนภาพลำดับของอุปกรณ์ควบคุมที่เป็น คอมพิวเตอร์ และ สมาร์ทโฟน

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

จากรูป 3.13 แสดงรูปแผนลำดับโดยใช้อุปกรณ์ควบคุมเป็น จุดเชื่อมต่อ Wi-Fi มีลำดับการใช้งานดังนี้

1. ผู้ใช้เปิดเครื่องเพื่อเริ่มต้นการใช้งาน
2. ส่งข้อมูลปริมาณแก๊สสู่เว็บไซต์ www.thingspeak.com ผ่านจุดเชื่อมต่อ Wi-Fi
3. เว็บไซต์จากข้อ 2) แสดงผลข้อมูลในรูปแบบกราฟความสัมพันธ์ระหว่างปริมาณแก๊สและเวลาที่วัด
4. ผู้ใช้ดูข้อมูลที่แสดง



รูปที่ 3.13 แผนภาพลำดับของอุปกรณ์ควบคุมที่เป็น Wi-Fi Hotspot

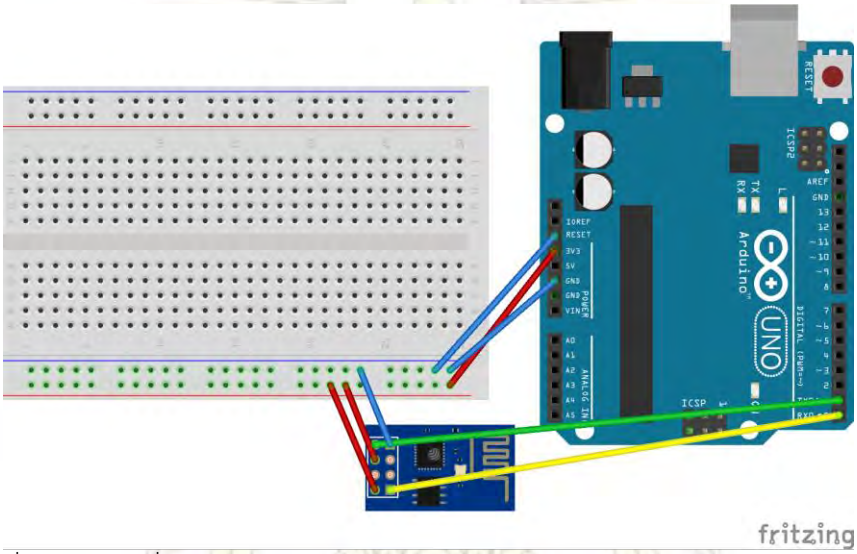
ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

3.4 วิธีการทดลอง

3.4.1 ทำเครื่องตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์ในอากาศ

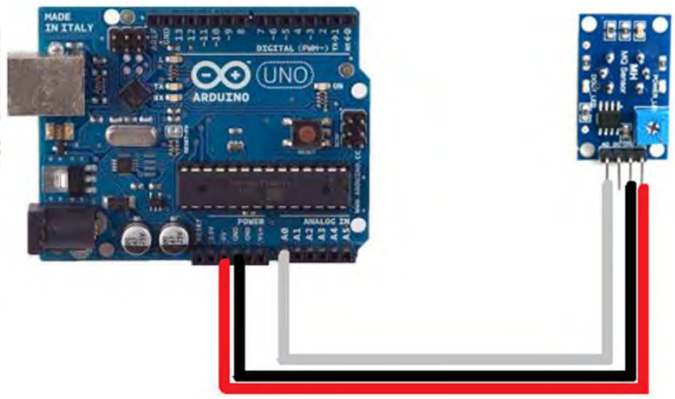
ในขั้นตอนแรกในการทำการทดลองนั้นจำเป็นจะต้องต่ออุปกรณ์เพื่อให้เป็นเครื่องตรวจวัดปริมาณแก๊สเสียก่อนซึ่งมีวิธีการต่ออุปกรณ์ดังนี้

1. เริ่มจากนำไมโครคอนโทรลเลอร์ Arduino มาเชื่อมต่อกับโมดูลไวเลสรุ่น ESP8266 ดังรูปที่ 3.14



รูปที่ 3.14 การเชื่อมต่อโมดูลไวเลสรุ่น ESP8266 กับไมโครคอนโทรลเลอร์ Arduino

2. ทำให้การเชื่อมต่อโมดูลแก๊สเซนเซอร์รุ่น MQ-135 เข้ากับไมโครคอนโทรลเลอร์ Arduino ดังรูปที่ 3.15



รูปที่ 3.15 การเชื่อมต่อโมดูลแก๊สเซนเซอร์รุ่น MQ-135 กับไมโครคอนโทรลเลอร์ Arduino

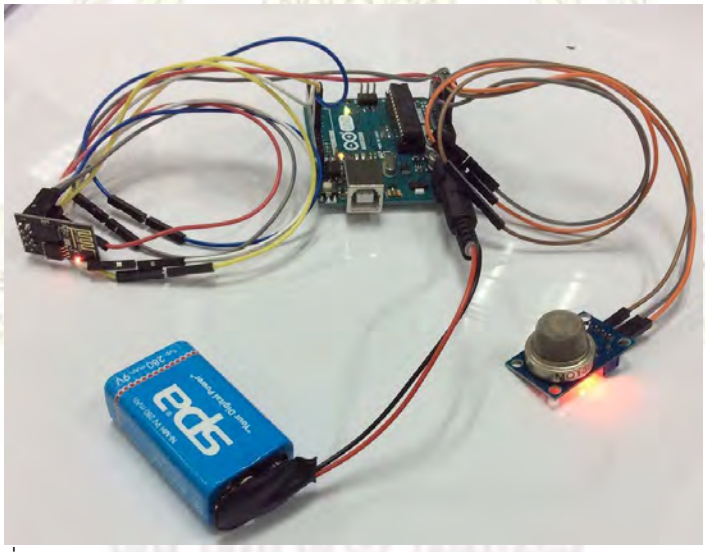
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

3. เสียบสาย USB ระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ Arduino เพื่อทำการส่งชุดคำสั่งเข้าสู่ไมโครคอนโทรลเลอร์ Arduino ดังรูปที่ 3.16

รูปที่ 3.16 การเชื่อมต่อระหว่างคอมพิวเตอร์และไมโครคอนโทรลเลอร์ Arduino

4. ทำการอัพโหลดชุดคำสั่ง (ภาคผนวก) โดยใช้เข้าโปรแกรม Arduino IDE จากคอมพิวเตอร์แล้วนำชุดคำสั่งจากภาคผนวก มาใส่ในโปรแกรมจากนั้นทำตามรูปที่ 2.7 ถึง 2.9

5. ต่อแหล่งจ่ายไฟฟ้าให้กับไมโครคอนโทรลเลอร์ดังรูปที่ 3.17 ซึ่งเครื่องตรวจวัดปริมาณแก๊สคาร์บอน-ไดออกไซด์ก็พร้อมสำหรับใช้งานเรียบร้อยแล้ว



รูปที่ 3.17 การต่อแหล่งจ่ายไฟฟ้าให้กับไมโครคอนโทรลเลอร์ Arduino

คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

3.4.2 ทดสอบความเสถียรของเครื่องตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์ในอากาศ

เมื่อเราได้เครื่องตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์ในอากาศเรียบร้อยแล้ว เราจำเป็นต้องนำเครื่องมาทำทดสอบความเสถียรกับเครื่องตรวจวัดเพื่อหาช่วงเวลาที่ทำให้เครื่องตรวจวัดมีความเสถียร

1. เปิดเครื่องตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์ในอากาศ ทิ้งไว้ 24 ชั่วโมงเพื่อให้เครื่องตรวจวัดสามารถทำงานได้อย่างเต็มประสิทธิภาพ แล้วติดตามช่วงที่เครื่องมือมีความเสถียร

2. ทำการวัดค่าที่อ่านได้จากเครื่องตรวจวัด เพื่อดูเวลาที่เครื่องตรวจวัดเสถียรโดย เปิดเครื่องตรวจวัดทิ้งไว้โดยใช้เวลาที่เครื่องมือมีความเสถียรจากข้อ 1) แล้วเพิ่มเวลาอีกเป็นสองเท่า เพื่อทดสอบว่าช่วงเวลาที่มีความเสถียรของเครื่องมือนั้นเป็นจริงตามข้อ 1) หรือไม่

3. เมื่อได้ช่วงเวลาที่เครื่องมือมีความเสถียรแล้ว นำทดสอบซ้ำด้วยการทำซ้ำ 7 วัน โดยใช้ช่วงเวลาที่เครื่องมือมีความเสถียรจากข้อ 2) จากนั้นติดตามดูความเสถียรของเครื่องมือ

4. เมื่อได้ช่วงเวลาที่เครื่องมือมีความเสถียรแน่นอนแล้วจากข้อ 3) เราจะใช้ช่วงเวลาดังกล่าวในการเปิดเครื่องมือทิ้งไว้เพื่อทำการรอให้เครื่องมือเสถียรก่อนจะนำไปใช้วัดปริมาณแก๊สคาร์บอนไดออกไซด์ต่อไป

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

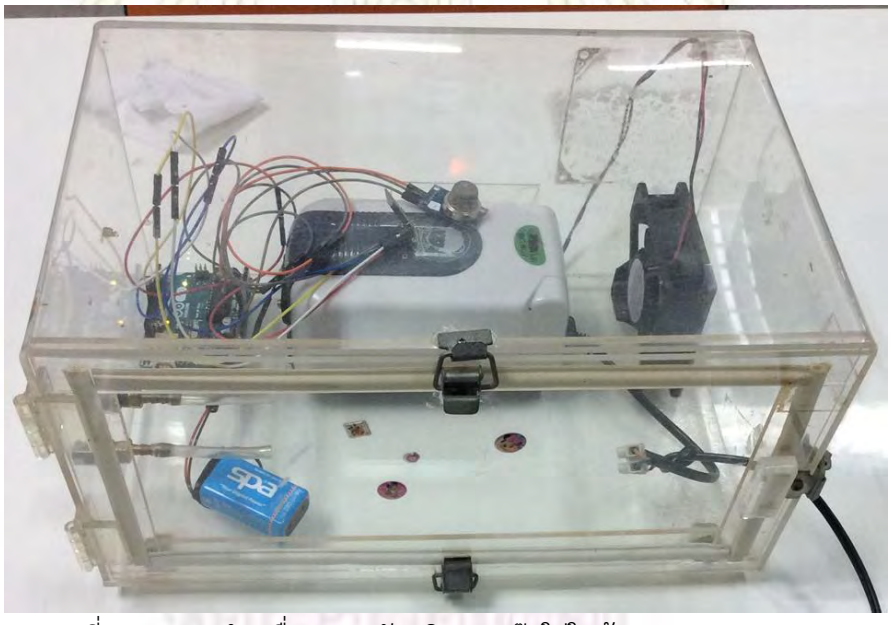
3.4.3 สอบเทียบเครื่องตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์ในอากาศ

จากข้อ 3.4.2 เมื่อได้ช่วงเวลาที่เครื่องมือที่มีความเสถียรแล้ว ต่อมาเราจะนำเครื่องมือมาทำการสอบเทียบเพื่อดูว่าปริมาณแก๊สคาร์บอนไดออกไซด์ที่อ่านได้ตรงกับความเป็นจริงมากน้อยเพียงใด หลังจากผ่านขั้นตอนนี้จะทำให้เราสามารถนำเครื่องตรวจวัดไปใช้งานได้จริง โดยมีขั้นตอนดังนี้

1. คำนวณความเข้มข้นของแก๊สคาร์บอนไดออกไซด์ที่ต้องการวัด ใช้ความเข้มข้น 100-1000 ppm โดยใช้ กรดไฮโดรคลอริกกับโซเดียมไบคาร์บอเนต

2. เตรียมกรดไฮโดรคลอริกเข้มข้น 1 โมลาร์และชั่งโซเดียมไบคาร์บอเนตตามที่คำนวณไว้ในข้อ 1)

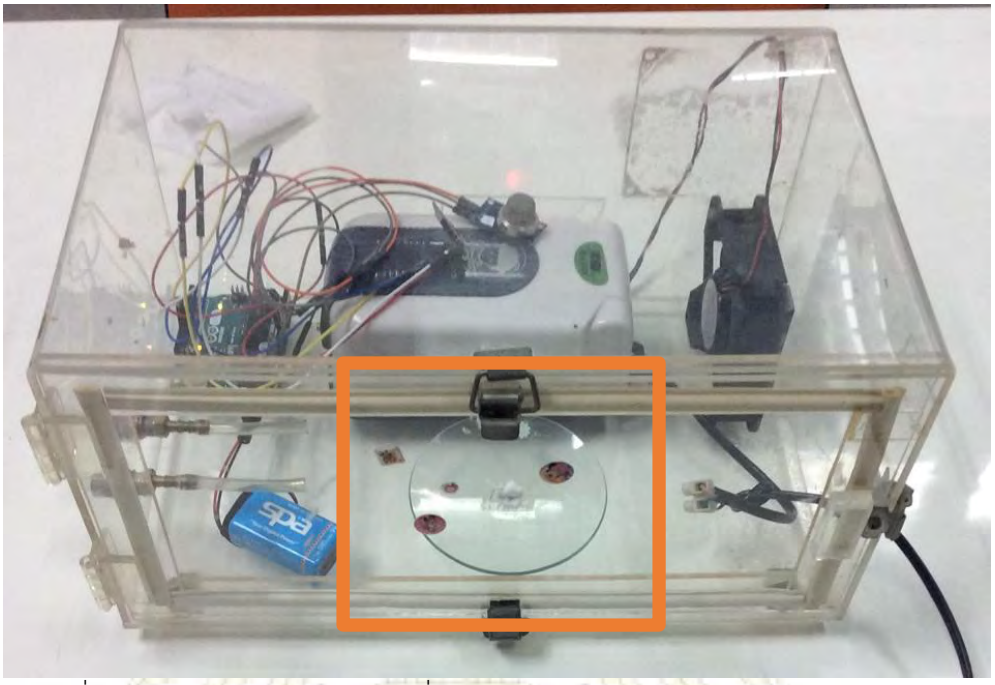
3. หลังจากเตรียมสารเรียบร้อยแล้วให้นำเครื่องตรวจวัดปริมาณแก๊สเข้าสู่ตู้ควบคุมสภาพอากาศ ซึ่งเป็นระบบปิดจากนั้นทำการดูดอากาศออกโดยใช้เครื่องปั๊มอากาศ ดังรูปที่ 3.18



รูปที่ 3.18 การนำเครื่องตรวจวัดปริมาณแก๊สใส่ในตู้ควบคุมสภาพอากาศ

4. อ่านค่าที่วัดได้จากเครื่องมือตรวจวัดปริมาณแก๊สโดยให้ค่าดังกล่าวเป็นค่าพื้นหลัง (Background)

5. นำโซเดียมไบคาร์บอเนตที่ชั่งซึ่งอยู่บนกระจกแก้วมาใส่ลงในตู้ควบคุมสภาพอากาศดังรูปที่ 3.19



รูปที่ 3.19 ตำแหน่งกระจกนาฬิกาที่มีโซเดียมไบคาร์บอเนตในตู้ควบคุมสภาพอากาศ

6. ทำการปิเปตด้วยไมโครปิเปตจากนั้นฉีดสารกรดไฮโดรคลอริกลงบนปลายกระจกนาฬิกาเพื่อให้สารค่อย ๆ ไหลไปทำปฏิกิริยากับโซเดียมไบคาร์บอเนต

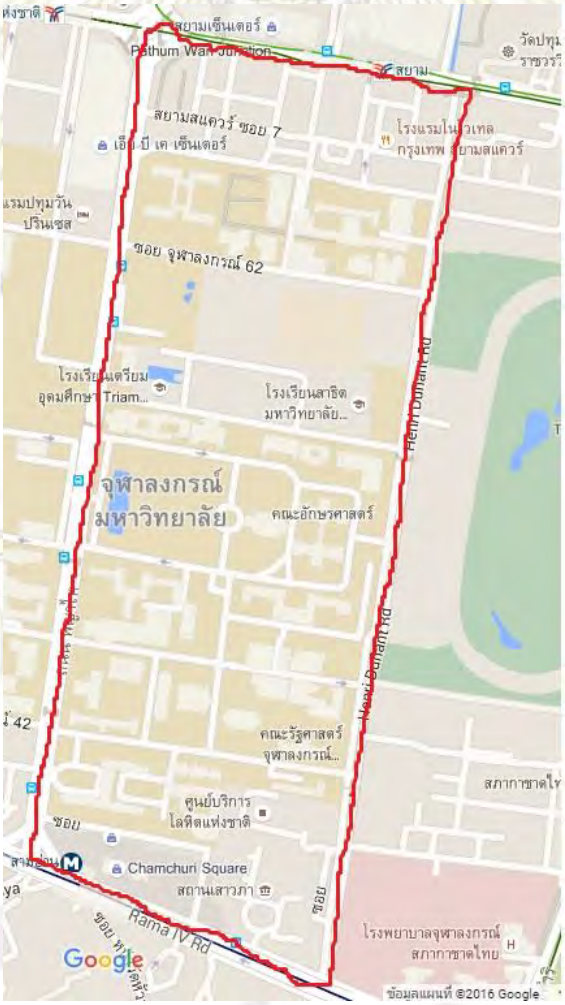
7. อ่านค่าที่วัดได้จากเครื่องวัดปริมาณแก๊สแล้วนำข้อมูลที่ได้มาสร้างกราฟความเข้มข้นมาตรฐาน

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

3.4.4 ใช้เครื่องตรวจวัดปริมาณแก๊สตัววัดปริมาณแก๊สคาร์บอนไดออกไซด์ในอากาศ

หลังจากสอบเทียบเครื่องมือแล้ว ทำให้เครื่องมือสามารถตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์ได้อย่างแม่นยำ จากนั้นทำการเก็บข้อมูลปริมาณแก๊สคาร์บอนไดออกไซด์ในอากาศเพื่อดูคุณภาพอากาศในบริเวณที่ทำการตรวจวัด ซึ่งมีวิธีเก็บข้อมูลดังนี้

1. เปิดเครื่องทิ้งไว้จนเครื่องมือมีความเสถียรพร้อมใช้งาน (จาก 3.4.2) ให้เครื่องตรวจวัดปริมาณแก๊สเชื่อมต่อกับ Wi-Fi Hotspot (สามารถใช้โหมดนี้ได้จากการเปิด Wi-Fi Hotspot จากสมาร์ทโฟนของตนเอง) เพื่อให้เครื่องมือสามารถส่งข้อมูลไปยังอินเตอร์เน็ตจากนั้นทำการสำรวจบริเวณที่เราสนใจ ดังรูปที่ 3.20 ซึ่งเป็นบริเวณถนนพญาไทไล่มาจนถึงถนนอังรีดูนังค์



รูปที่ 3.20 แผนที่ในการเก็บข้อมูลปริมาณแก๊สคาร์บอนไดออกไซด์

2. เมื่อระหว่างทำการเดินสำรวจให้ทำการบันทึกเวลาและตำแหน่งที่อยู่ ณ ปัจจุบันเพื่อใช้ในการอ้างอิงกับข้อมูลที่ได้จากเครื่องตรวจวัดปริมาณแก๊ส

3. นำข้อมูลมาแผนภาพเทียบกับแผนที่ว่าแต่ละบริเวณมีความเข้มข้นของปริมาณแก๊สคาร์บอนไดออกไซด์เท่าใด โดยเทียบกับข้อมูลจากองค์กรอนามัยว่าด้วยการความสามารถในการรับปริมาณแก๊สคาร์บอนไดออกไซด์ แล้วทำการระบุว่าบริเวณใดบ้างมีความเสี่ยงต่อสุขภาพของประชาชน

4. ทำการเก็บข้อมูลใน 1 สัปดาห์เปรียบเทียบว่าวันใดในหนึ่งสัปดาห์จะมีปริมาณแก๊สคาร์บอนไดออกไซด์ในปริมาณสูง

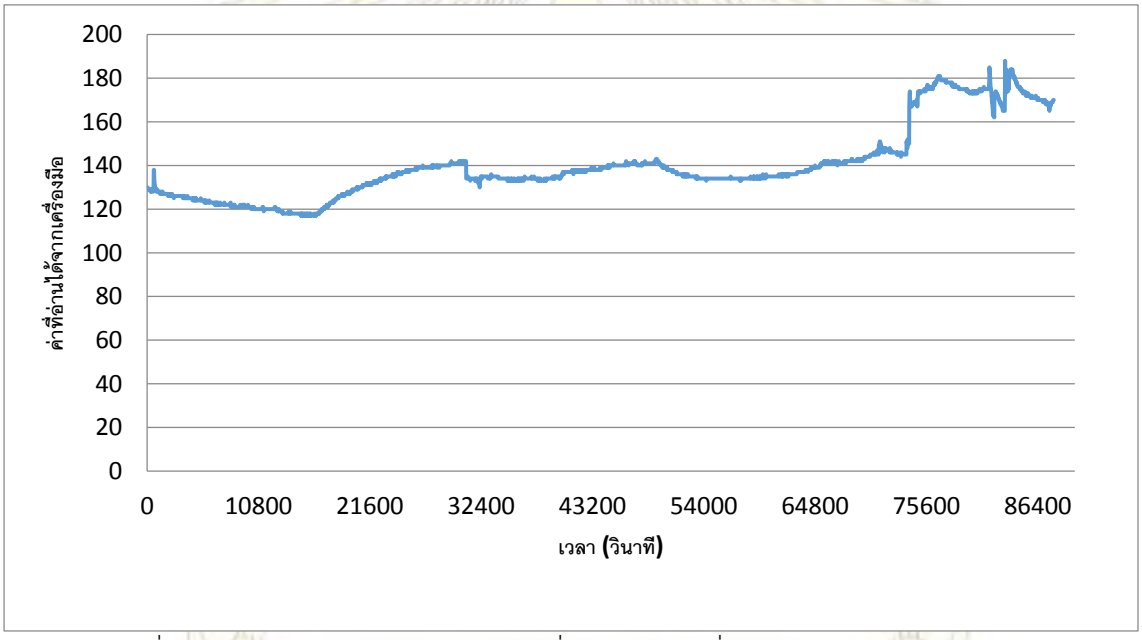
ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4
ผลการทดลองและอภิปรายผลการทดลอง

4.1 ขั้นตอนการทดสอบความเสถียรของเครื่องตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์ในอากาศ

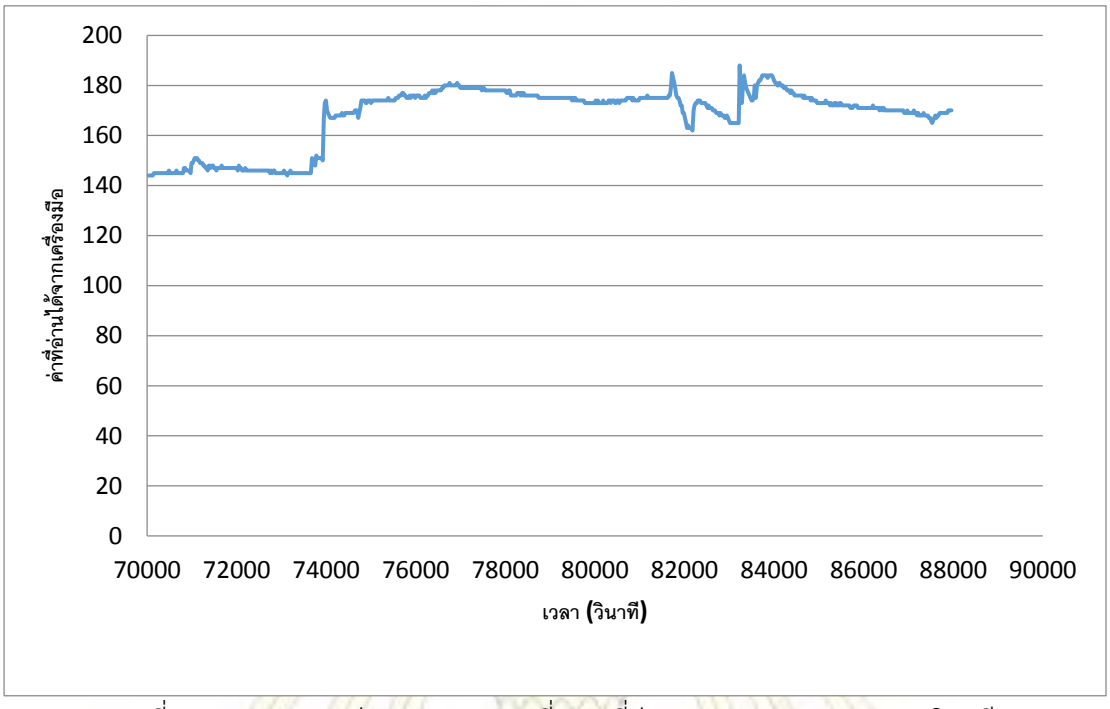
ผลการทดลอง

จากการทดสอบความเสถียรของเครื่องตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์ในอากาศ เมื่อทำการเปิดทิ้งไว้ 24 ชั่วโมง และวัดปริมาณแก๊สทุก ๆ 10 วินาที พบว่าได้ข้อมูลได้รูปที่ 4.1



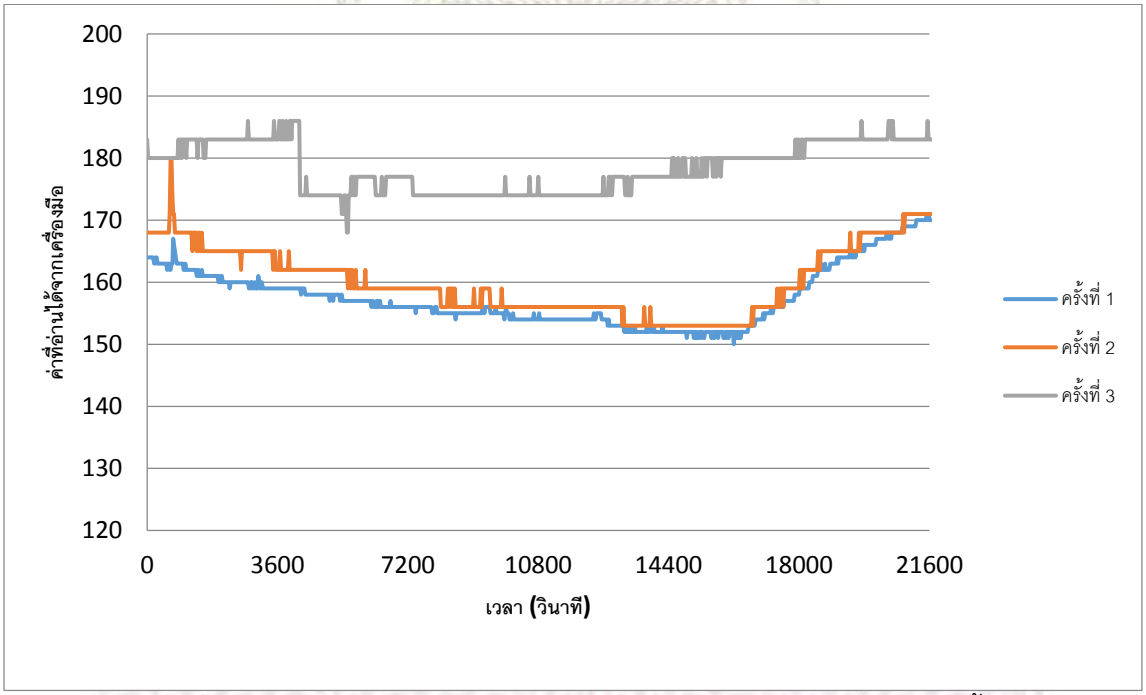
รูปที่ 4.1 กราฟความสัมพันธ์ระหว่างค่าที่อ่านได้จากเครื่องมือกับเวลาในหน่วยวินาที

จากรูปจะเห็นว่าในช่วง 3 ชั่วโมงแรกหรือ 10800 วินาทีแรกนั้น เครื่องมือให้ค่าที่มีแนวโน้มลดลงเรื่อย ๆ อย่างเป็นเส้นตรงหากดูโดยภาพรวม และเมื่อพิจารณาจนถึงช่วงวินาทีที่ 75600 วินาที ข้อมูลที่ได้เกิดการขาดช่วงแล้วขึ้นไปจากค่าเดิมไม่มากนัก และหลังจากช่วงวินาทีที่ 75600 วินาทีไปแล้ว ข้อมูลมีการขึ้นลงอย่างไม่เป็นแนวโน้มเดียวกันอย่างเห็นได้ชัด (จากรูปที่ 4.2)



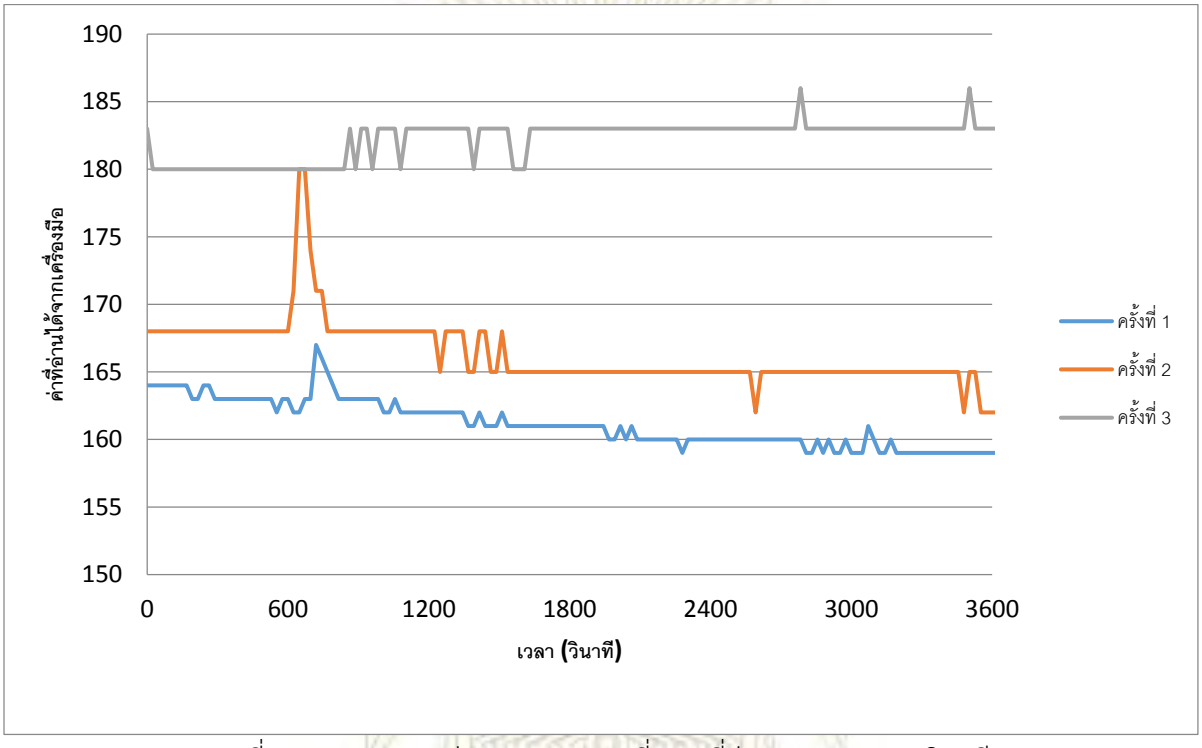
รูปที่ 4.2 กราฟแสดงส่วนขยายจากรูปที่ 4.1 ที่ช่วงเวลา 70000 – 90000 วินาที

ดังนั้นผู้ทำการพัฒนาจึงทำการวัดซ้ำโดยเก็บข้อมูลทุก ๆ 6 ชั่วโมง 3 ครั้ง เพื่อติดตามดูค่าที่อ่านได้จากเครื่องมือดังรูปที่ 4.3



รูปที่ 4.3 กราฟแสดงการเก็บข้อมูลทุก ๆ 10 วินาทีเป็นเวลา 6 ชั่วโมง 3 ครั้ง

จากรูปที่ 4.3 จะเห็นว่าช่วง 0-3600 วินาที นั้นเครื่องมือตรวจวัดปริมาณแก๊สสามารถให้ค่าที่มีแนวโน้มไปในทิศทางเดียวกันเมื่อพิจารณาที่ช่วง 0-3600 วินาที ดังรูปที่ 4.4

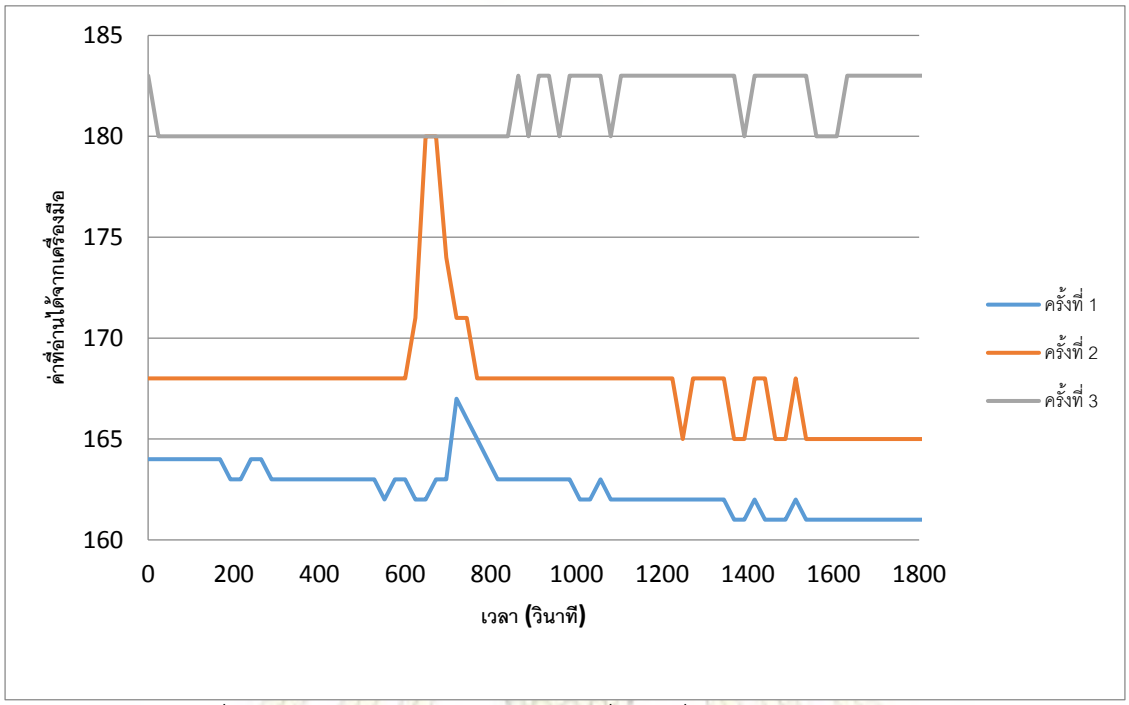


รูปที่ 4.4 กราฟแสดงส่วนขยายจากรูปที่ 4.3 ที่ช่วงเวลา 0-3600 วินาที

จากรูปที่ 4.4 จะเห็นว่าค่าที่อ่านได้จากข้อมูลเริ่มเสถียรตั้งแต่ช่วงเวลาวินาทีที่ 1800 เป็นต้นไป หรือประมาณ 30 นาที ดังนั้นในขั้นตอนนี้เราสามารถได้ช่วงเวลาที่เครื่องมือเสถียรตั้งแต่ช่วงเวลาที่ 30 นาทีเป็นต้นไป

วิจารณ์ผลการทดลอง

เนื่องจากเครื่องมือที่จะใช้ปริมาณแก๊สคาร์บอนไดออกไซด์นั้น ใช้หลัก Thermal Conductivity ดังที่อธิบายในหัวข้อที่ 2.1.2 โมดูล ซึ่งจำเป็นจะต้องให้ความร้อนกับโมดูลเซนเซอร์แก๊ส เพื่อใช้ในการตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์ ในช่วงเวลา 30 นาทีแรกจึงเป็นการให้ความร้อนกับโมดูลเซนเซอร์แก๊สให้พร้อมที่ใช้ในการตรวจวัดปริมาณแก๊สได้ ทั้งนี้ถ้าพิจารณาในช่วง 0-1800 วินาที ดังรูปที่ 4.5 จะเห็นว่าช่วงวินาทีที่ 0-600 หรือ 10 นาทีแรกนั้นเครื่องมืออ่านค่าได้ไม่ดีเพราะเป็นช่วงเริ่มต้นในการให้ความร้อนกับโมดูลแก๊สเซนเซอร์ และอีก 10 นาทีถัดมา ค่าที่อ่านได้เริ่มมีแนวโน้มไปในทางเดียวกันแล้ว แต่ก็ยังไม่เสถียรพอ จนกระทั่งช่วงที่ 1600 วินาที หรือประมาณ 30 นาทีนั้นเครื่องมือเริ่มอ่านค่าได้ดีขึ้นเรื่อย ๆ จนมีความเสถียรที่ช่วงเวลานี้เป็นต้นไป ดังรูปที่ 4.3 ดังนั้นช่วงเวลา 30 นาทีแรกนั้นจึงเป็นช่วงเวลาที่ใช้เพื่อให้เครื่องมือเสถียรนั่นเอง ดังนั้นหลังจากขั้นตอนนี้แล้วจะต้องทำการเปิดเครื่องทิ้งไว้ก่อน 30 นาทีเพื่อรอให้เครื่องมือเสถียรแล้วจึงเริ่มใช้งาน



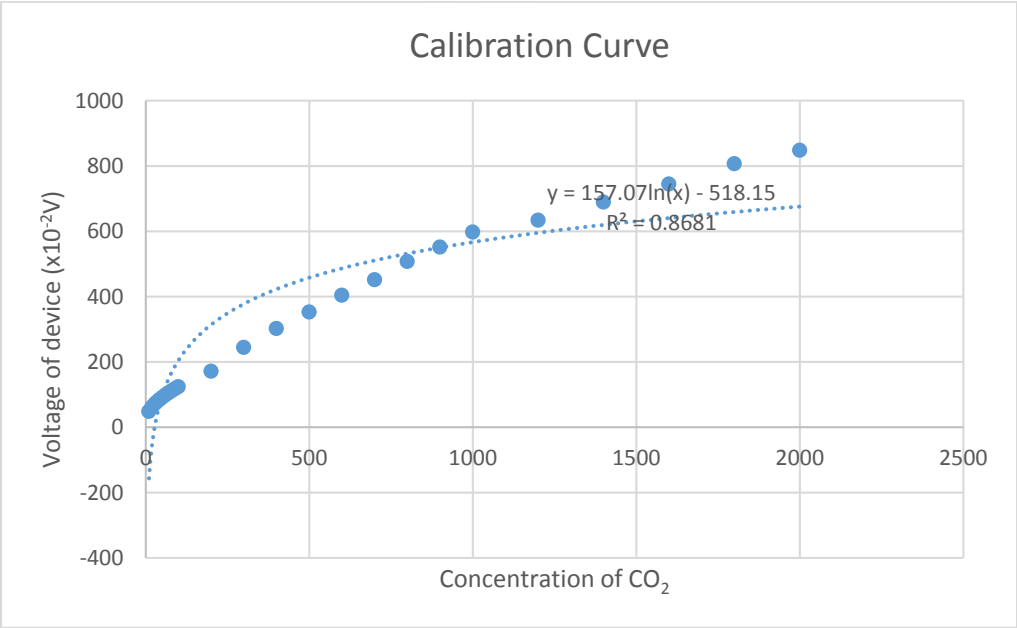
รูปที่ 4.5 แสดงกราฟส่วนขยายจากรูปที่ 4.3 ที่ช่วงเวลา 0-1800 วินาที

4.2 ขั้นตอนการสอบเทียบเครื่องตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซค์ในอากาศ

จากหัวข้อที่ 4.1 เราได้ช่วงเวลาที่ทำให้เครื่องมือเสถียรแล้วคือ 30 นาที เมื่อเริ่มทำการทดลองจึงใช้เวลา 30 นาทีเพื่อทำให้เครื่องมือเสถียร จากนั้นจึงทำการสอบเทียบโดยให้เครื่องมือตรวจวัดปริมาณแก๊สอยู่ในตู้ควบคุมสภาพอากาศซึ่งเป็นระบบปิด แล้วทำการดูดอากาศออกแล้วทำการอ่านค่า เพื่อวัดค่าพื้นหลัง (Background) จากนั้นจึงใส่โซเดียมไบคาร์บอเนตและหยดกรดไฮโดรคลอริกเพื่อทำให้เกิดแก๊สคาร์บอนไดออกไซค์ ซึ่งส่วนการคำนวณและอธิบายสมการในส่วนวิจารณ์ผลการทดลอง

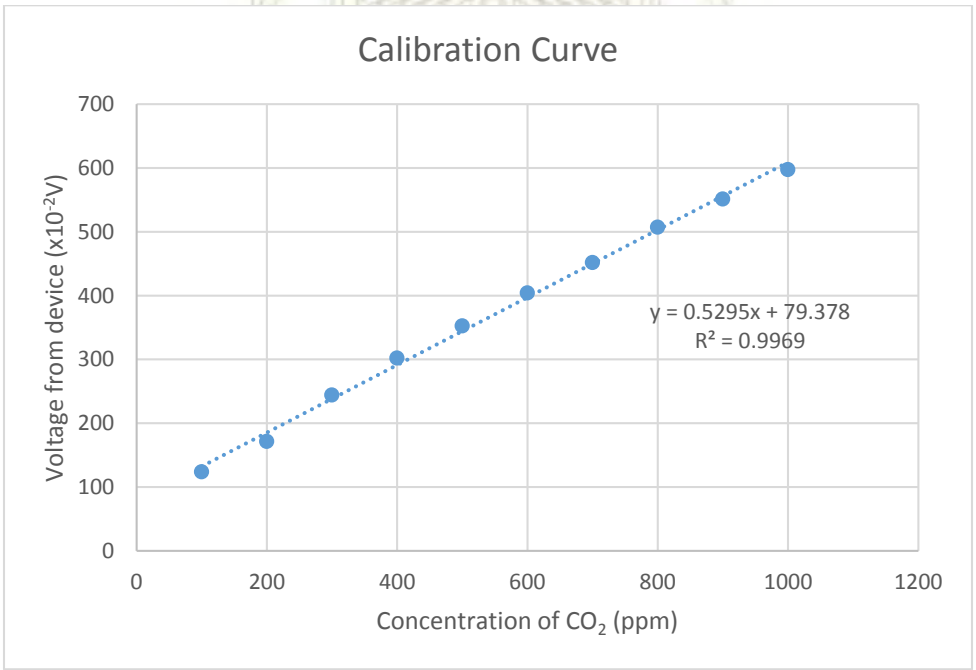
ผลการทดลอง

จากการทดลองโดยทำการตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซค์ตั้งแต่ช่วงความเข้มข้น 0-2000 ppm ดังรูปที่ 4.6 โดยทำการวัดซ้ำ 3 ครั้งแล้วหาค่าเฉลี่ย จะเห็นว่าข้อมูลมีแนวโน้มเป็นความสัมพันธ์เอกซ์โพเนนเชียล (Exponential) โดยมีความสัมพันธ์ $y = 0.0238x^{1.6874}$ เมื่อ y คือความเข้มข้นของแก๊สคาร์บอนไดออกไซค์และ x คือค่าที่อ่านได้จากเครื่องมือที่ความเชื่อมั่นของความสัมพันธ์เอกซ์โพเนนเชียล 0.9884 หรือ 98.84%



รูปที่ 4.6 กราฟความเข้มข้นมาตรฐานระหว่าง ความเข้มข้นของแก๊สคาร์บอนไดออกไซด์ และค่าที่อ่านได้จากเครื่องมือ

เนื่องจากความสัมพันธ์ของข้อมูลนั้นมีค่าความน่าเชื่อถือน้อย (R^2 ต่ำกว่า 0.95) ดังนั้นทางผู้ทำการทดลองจึงเลือกพิจารณาความเข้มข้นของแก๊ส CO_2 ที่ 100-1000 ppm ดังรูปที่ 4.7



รูปที่ 4.7 กราฟความเข้มข้นมาตรฐานระหว่าง ความเข้มข้นของแก๊สคาร์บอนไดออกไซด์ และค่าที่อ่านได้จากเครื่องมือ ที่ช่วงความเข้มข้นของแก๊สคาร์บอนไดออกไซด์ 100-1000 ppm

วิจารณ์ผลการทดลอง

ในขั้นการเตรียมแก๊สคาร์บอนไดออกไซด์ที่ใช้สำหรับการสอบเทียบเครื่องมือนั้นเตรียมจากการทำปฏิกิริยาของโซเดียมไบคาร์บอเนตและกรดไฮโดรคลอริกดังสมการ

$$NaHCO_3(s) + HCl \rightarrow NaCl(s) + H_2O(l) + CO_2(g)$$

ตัวอย่างการสามารถคำนวณความเข้มข้นของแก๊สคาร์บอนไดออกไซด์ที่ 100 ppm ดังสมการ

$$g\ NaHCO_3 = \frac{100\ mg\ CO_2}{1000\ mL\ CO_2}\ x\ 10000\ mL \text{ ตู้ควบคุมสภาพอากาศ } x\ \frac{1\ mol\ CO_2}{44\ g\ CO_2}\ x\ \frac{1\ mol\ NaHCO_3}{1\ mol\ CO_2}\ x\ \frac{84\ g\ NaHCO_3}{1\ mol\ NaHCO_3}$$
$$= 1.909\ g\ NaHCO_3$$

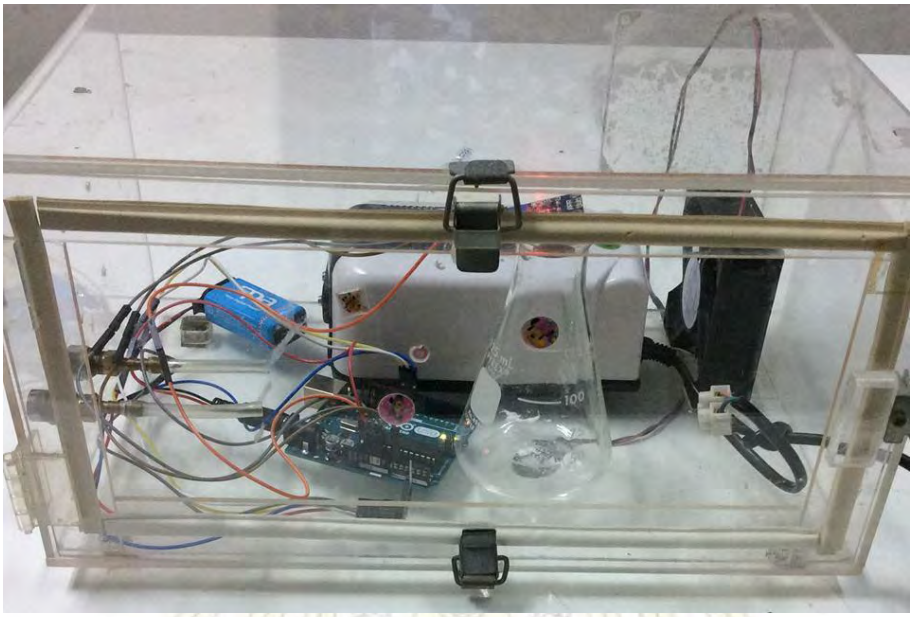
จากสมการ เราสามารถเตรียม $NaHCO_3$ ได้ดังตารางที่ 4.1 และเนื่องจากที่ความเข้มข้นสูง ๆ จะต้องใช้โซเดียมไบคาร์บอเนตจำนวนมากจึงเปลี่ยนการทดลองโดยนำขวดรูปชมพู่ปริมาตร 125 mL ใส่ลงไปในตู้ควบคุมสภาพอากาศ แล้วทำการดูดอากาศ เมื่อเตรียมสารเสร็จ (ที่ 250 mL ขวดรูปชมพู่) ให้ทำการหยดกรดไฮโดรคลอริก เมื่อหยดเสร็จให้นำเครื่องวัดปริมาณแก๊สมาวัดที่บริเวณปากขวดทันทีดังรูปที่ 4.7

ตารางที่ 4.1 ปริมาณโซเดียมไบคาร์บอเนตที่ต้องใช้ในสำหรับความเข้มข้นต่าง ๆ ของแก๊สคาร์บอนไดออกไซด์

ppm CO_2	10	20	30	40	50	60	70	80
g $NaHCO_3$	1.909091	3.818182	5.727273	7.636364	9.545455	11.45455	13.36364	15.27273

ppm CO_2	90	100	200	300	400	500	600	700
g $NaHCO_3$	17.18182	0.238636	0.477273	0.715909	0.954545	1.193182	1.431818	1.670455

ppm CO_2	800	900	1000	1200	1400	1600	1800	2000
g $NaHCO_3$	1.909091	2.147727	2.386364	2.863636	3.340909	3.818182	4.295455	4.772727



รูปที่ 4.8 การเตรียมสารจากขวดรูปชมพู่และตำแหน่งในการวัดปริมาณแก๊สคาร์บอนไดออกไซค์

ดังนั้นในขั้นตอนนี้เราสามารถนำสมการที่ได้จากกราฟความเข้มข้นมาตรฐานมาใช้ในการตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์ และสามารถนำไปตรวจวัดคุณภาพอากาศได้

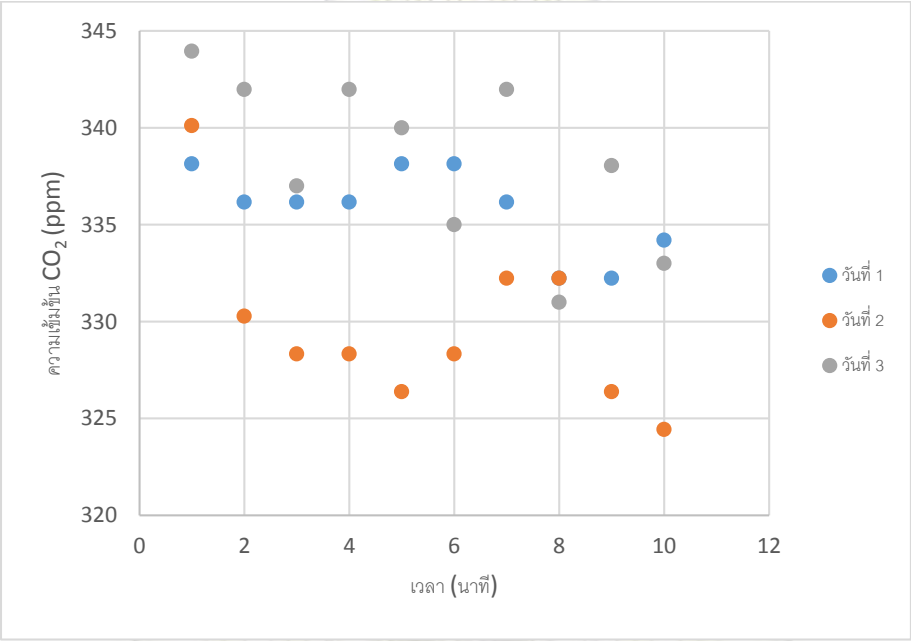
4.3 ขั้นตอนการใช้เครื่องตรวจวัดปริมาณแก๊สตัววัดปริมาณแก๊สคาร์บอนไดออกไซด์ในอากาศ

หลังจากที่เราได้ทำการสอบเทียบเครื่องมือแล้ว จากนั้นเรานำเครื่องมือไปทำการตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์ดังรูปที่ 4.9-4.12 แสดงแผนที่ในการเก็บข้อมูลปริมาณแก๊สคาร์บอนไดออกไซด์ โดยการวัดซ้ำทุก ๆ 1 นาที ซึ่งได้ข้อมูลดังนี้

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

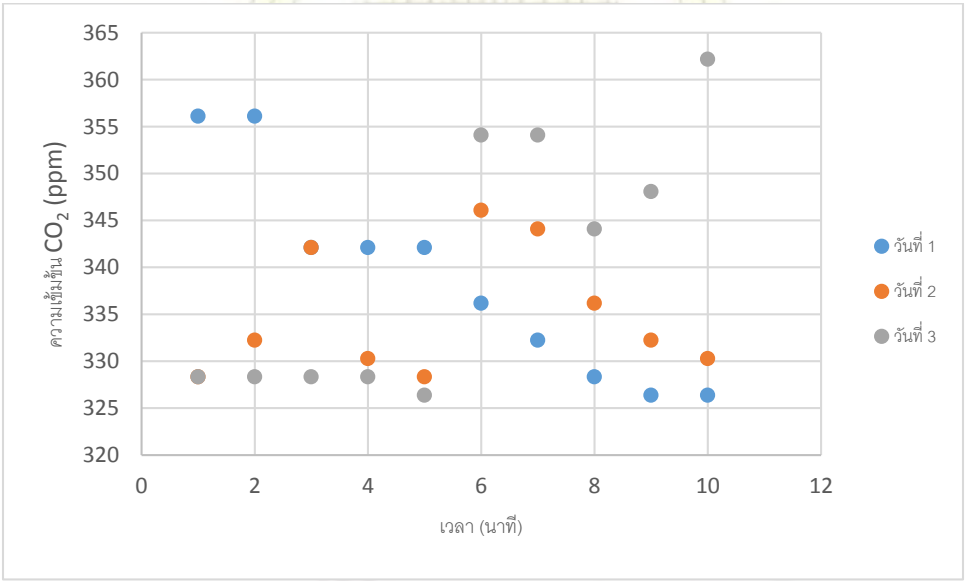
ผลการทดลอง

บริเวณริมถนนพญาไท



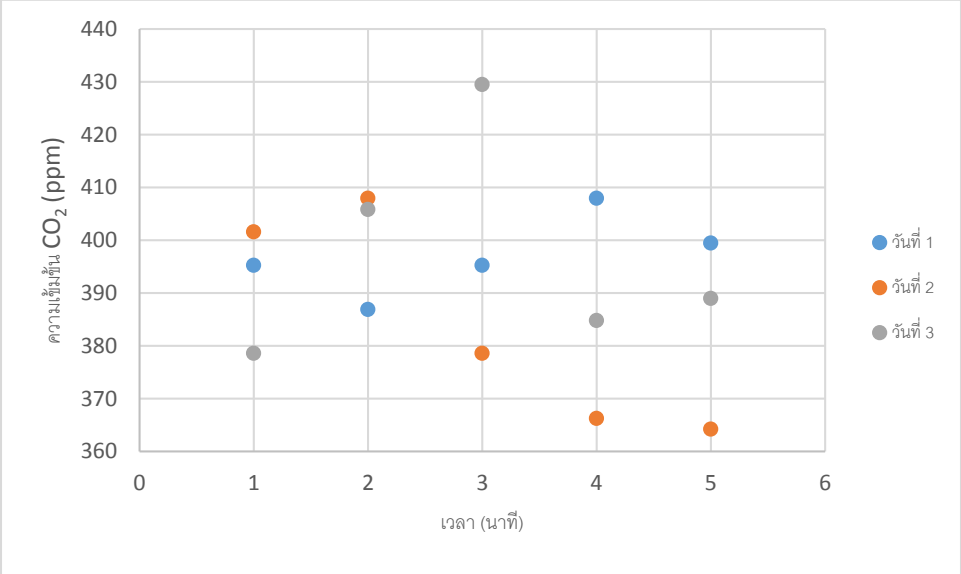
รูปที่ 4.9 กราฟแสดงข้อมูลปริมาณแก๊สคาร์บอนไดออกไซด์บริเวณริมถนนพญาไท

บริเวณริมถนนอังรีดูนังค์



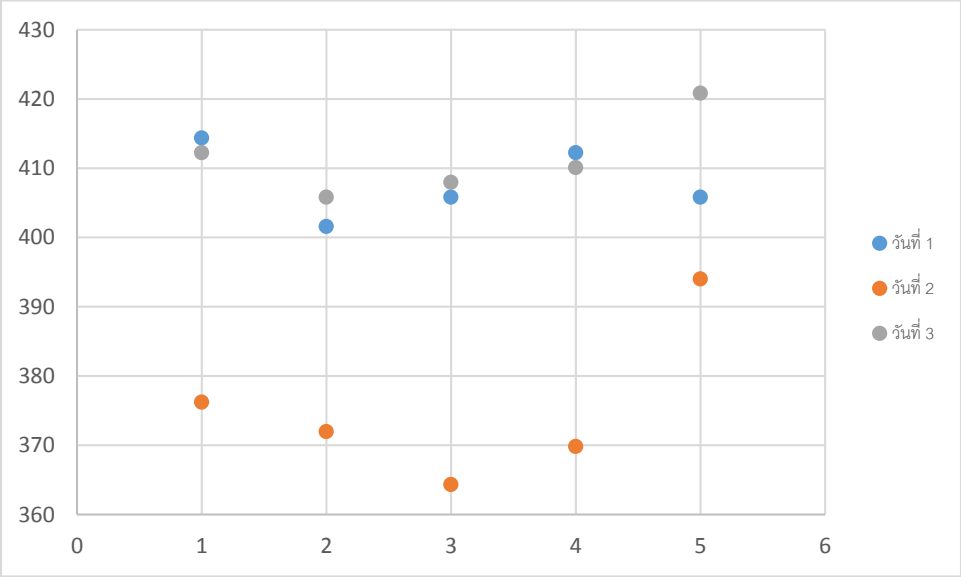
รูปที่ 4.10 กราฟแสดงข้อมูลปริมาณแก๊สคาร์บอนไดออกไซด์บริเวณริมถนนอังรีดูนังค์

บริเวณถนนพระราม 1



รูปที่ 4.11 กราฟแสดงข้อมูลปริมาณแก๊สคาร์บอนไดออกไซค์บริเวณริมถนนพญาไท

บริเวณถนนพระราม 4



รูปที่ 4.12 กราฟแสดงข้อมูลปริมาณแก๊สคาร์บอนไดออกไซค์บริเวณริมถนนพระราม 4

เนื่องจากการเก็บข้อมูลต้องใช้ความเร็วในการตรวจวัดแต่ละบริเวณให้ใกล้เคียงกัน ส่งผลให้บริเวณถนนพระราม 4 และ พระราม 1 ที่มีบริเวณถนนที่สั้นกว่านั้นสามารถเก็บข้อมูลเก็บได้เพียงแค่ 5 นาทีเท่านั้น และบริเวณต่าง ๆ มีปริมาณแก๊สคาร์บอนไดออกไซค์เฉลี่ยตามตารางที่ 4.2

	ปริมาณแก๊สคาร์บอนไดออกไซด์ (ppm)		
บริเวณที่เก็บข้อมูล	**วันที่ 1**	**วันที่ 2**	**วันที่ 3**
ริมถนนพญาไท	336	330	338
ริมถนนอังรีดูนังค์	339	335	340
ริมถนนพระราม 1	397	384	398
ริมถนนพระราม 4	408	375	411

ตารางที่ 4.2 ปริมาณแก๊สคาร์บอนไดออกไซด์เฉลี่ย ณ บริเวณริมถนนต่าง ๆ

วิจารณ์ผลการทดลอง

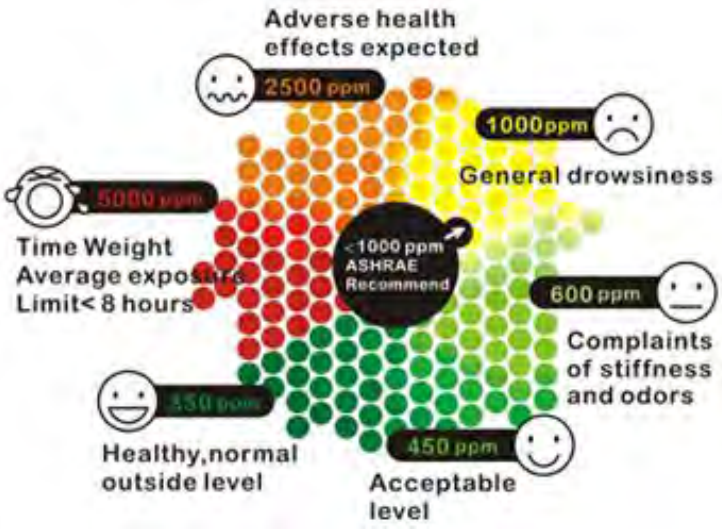
จากข้อมูลปริมาณแก๊สคาร์บอนไดออกไซด์ในอากาศที่ทำการสำรวจพบว่าค่าเฉลี่ยของบริเวณริมถนนพระราม 4 มีค่าเฉลี่ยของปริมาณแก๊สคาร์บอนไดออกไซด์ในอากาศมากที่สุด คือ 411 ppm และบริเวณริมถนนพญาไทมีค่าเฉลี่ยของปริมาณแก๊สคาร์บอนไดออกไซด์ในอากาศน้อยที่สุด คือ 330 ppm เนื่องด้วยสาเหตุจากบริเวณริมถนนพระราม 4 นั้นเป็นถนนสองเลนซึ่งทำให้มีการจราจรติดขัดอยู่เป็นประจำ ทำให้อากาศในบริเวณนั้นมีค่าเฉลี่ยของปริมาณแก๊สคาร์บอนไดออกไซด์มากกว่าบริเวณอื่น ๆ และในบริเวณริมถนนพญาไทนั้นมีปริมาณต้นไม้ตามริมถนนอยู่เยอะอีกทั้งมีถนน 6 เลน (ขาเข้า 3 เลน ขาออก 3 เลน) ทำให้การตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์ได้น้อยที่สุดเมื่อเทียบกับบริเวณอื่น ๆ

ต่อมาบริเวณริมถนนพระราม 1 บริเวณนี้มักจะมีการจราจรติดขัดอยู่เสมอ เพราะใกล้กับบริเวณห้างสรรพสินค้า และมีจำนวนคนเยอะมากเมื่อเทียบกับบริเวณอื่น ๆ ส่งผลให้ค่าเฉลี่ยแก๊สคาร์บอนไดออกไซด์ในแต่ละวันมีปริมาณสูงอยู่ตลอดเวลาเมื่อเทียบกับบริเวณอื่น ๆ ในขณะที่บริเวณริมถนนอังรีดูนังค์มีค่าเฉลี่ยปริมาณแก๊สคาร์บอนไดออกไซด์มากกว่าบริเวณริมถนนพญาไทเล็กน้อยซึ่งบริเวณถนนอังรีดูนังค์เนื่องมาจากมีสภาพการจราจรที่ติดขัดมากกว่า ถนนมี 6 เลนเหมือนถนนพญาไท แต่มักถูกเป็นที่จอดรถไว้ 1 เลนทั้งถนนขาเข้าและออก ในสภาพเช่นนี้ทำให้มีการจราจรติดขัดและทำให้บริเวณนี้มีปริมาณแก๊สคาร์บอนไดออกไซด์เยอะกว่าบริเวณริมถนนพญาไทแม้ว่าจะมีต้นไม้อยู่ริมถนนเยอะก็ตาม

นอกจากนี้หากพิจารณาแก๊สมลพิษต่าง ๆ มีอยู่บริเวณท้องถนนแล้วพบว่ามีแก๊สหลายชนิดมาก ตัวอย่างเช่น CO,NO_2,O_3,SO_2 เป็นต้น ซึ่งมีปริมาณแก๊สเหล่านี้อยู่ในปริมาณน้อยมาก [9] น้อยกว่า 5 ppm เนื่องจากแก๊สเซนเซอร์สามารถตรวจจับแก๊สได้หลายชนิดจริง แต่แก๊สพิษเหล่านี้นั้นมีปริมาณที่น้อยมาก ๆ เมื่อเทียบกับ

แก๊สคาร์บอนไดออกไซด์ จึงส่งผลน้อยมากสำหรับการวัดแก๊สคาร์บอนไดออกไซด์ทำให้สรุปได้ว่าแก๊สพิษชนิดอื่น ๆ จะไม่ส่งผลใด ๆ ที่มีนัยสำคัญสำหรับการตรวจวัดแก๊สคาร์บอนไดออกไซด์

จากข้อมูลในตารางที่ 4.2 นั้นพบว่าปริมาณของแก๊สคาร์บอนไดออกไซด์ที่อยู่แต่ละบริเวณไม่ทำให้ประชาชนที่ใช้รถ ใช้ถนน หรือผู้ที่สัญจรไปมาได้รับอันตราย เนื่องจากยังมีระดับแก๊สคาร์บอนไดออกไซด์ในระดับที่ยังปลอดภัย ดังรูปที่ 4.13



รูปที่ 4.13 ระดับแก๊สคาร์บอนไดออกไซด์ที่ส่งผลต่อสุขภาพ

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5
สรุปผลการทดลอง

จากการพัฒนาเครื่องมือตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์ในอากาศโดยใช้ไมโครคอนโทรลเลอร์ Arduino สามารถสรุปการดำเนินโครงงาน รวมถึงปัญหาและอุปสรรคที่พบในระหว่างการดำเนินโครงการ รวมถึงวิธีการแก้ไขปัญหาและข้อเสนอแนะเพื่อพัฒนาเครื่องมือตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์ในอากาศต่อไปในอนาคต

5.1 สรุปผล

1. สามารถสร้างเครื่องมือที่ใช้ในการตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์ในอากาศใช้ไมโครคอน-โทรลเลอร์ Arduino ได้สำเร็จ โดยมีราคาถูกกว่าเครื่องมืออื่น ๆ มาก (สามารถดูราคาได้จากภาคผนวก)

2. สามารถติดตามคุณภาพอากาศได้ตลอดเวลาสำเร็จผ่านเว็บไซต์ www.Thingspeak.com

3. ช่วงเวลาที่รอให้เครื่องมือมีความเสถียรในการตรวจวัดที่เวลา 30 นาที

4. ค่าที่อ่านได้จากเครื่องมือมีความสัมพันธ์เป็นเส้นตรง = 0.5295x + 79.378 ที่ความเชื่อมั่นของความสัมพันธ์เอกซ์โพเนนเชียล 0.9969 หรือ 99.69%

5.2 ปัญหาและอุปสรรค

1. ปัญหาแหล่งจ่ายไฟฟ้าโดยใช้ถ่านชาร์จ 9 โวลต์ ที่ใช้ในการจ่ายไฟฟ้าให้กับเครื่องตรวจวัดปริมาณคุณภาพอากาศหากใช้เป็นเวลานาน ๆ มากกว่าสองชั่วโมงจะทำให้ค่าที่อ่านได้ผิดปกติไปจากเดิม เนื่องมาจากปริมาณแรงดันไฟฟ้าที่จ่ายให้กับเครื่องมือต่ำลง ส่งผลให้โมดูลแก๊สเซนเซอร์ให้ความร้อนได้น้อยลง จนทำให้ค่าที่อ่านออกมาได้ผิดปกติ อาจจะมากหรือน้อยกว่าค่าปกติก็ได้

2. การตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์สามารถถูกรบกวนด้วยแก๊สพิษอื่น ๆ ที่ระดับความเข้มข้นใกล้เคียงกับปริมาณแก๊สคาร์บอนไดออกไซด์หากนำไปวัดในบริเวณอื่นที่ไม่ใช่ริมถนน ตัวอย่างเช่นหากใช้ในวัดที่บริเวณโรงงานผลิตแก๊สแอมโมเนีย จะถูกแก๊สแอมโมเนียจากบริเวณที่ผลิตรบกวนได้เป็นต้น

3. การเรียงสายไฟที่ใช้สำหรับเชื่อมต่ออุปกรณ์ระหว่างไมโครคอนโทรลเลอร์และโมดูล หากใช้สายไฟสีเดียวกันจะส่งผลให้เกิดความสับสนในการเขียนชุดคำสั่งได้

4. การใช้งานหากใช้งานโดยไม่ระมัดระวังที่ทำตกจะส่งผลโดยตรงกับอุปกรณ์ไฟฟ้าและสายไฟทันที

5.3 การแก้ไขปัญหา

1. หากจำเป็นต้องใช้เครื่องตรวจวัดในระยะเวลานาน ๆ ควรมีแหล่งจ่ายไฟถาวรหรืออาจจะใช้ถ่านชาร์จมากกว่า 2 ก้อนขึ้นไปเพื่อใช้งานผลัดกันชาร์จไฟ ทั้งนี้เพราะโมดูลแก๊สเซนเซอร์จะเป็นตัวกินไฟฟ้ามาก ๆ เนื่องจากต้องใช้ไฟฟ้าผลิตความร้อนอยู่ตลอดเวลา

2. การใช้งานที่บริเวณต่าง ๆ ต้องทราบอย่างคร่าว ๆ ว่ามีแก๊สพิษหรือสารระเหยอื่น ๆ อยู่ในบริเวณนั้นด้วยหรือไม่ หากมีจะส่งผลโดยตรงกับค่าที่อ่านได้จากเครื่อง ดังนั้นควรหลีกเลี่ยงที่จะอยู่ในบริเวณที่มีสารระเหยอื่น ๆ ในสภาพเข้มข้น

3. ควรแยกสีสายไฟอย่างละเอียดและใช้ในแบบเดียวกันตลอดการทดลอง จะทำให้ไม่เกิดปัญหาดังกล่าว

4. หากเครื่องมือนำไปใส่กล่องพลาสติกและเจาะรูสำหรับวัดปริมาณแก๊ส โดยในกล่องดังกล่าวมีโฟมหรือฟองน้ำเพื่อดูดซับแรงกระแทกจะทำให้เครื่องมือสามารถใช้ได้อย่างยาวนานขึ้น

5.4 ข้อเสนอแนะ

จากการพัฒนาเครื่องมือในการวัดปริมาณแก๊สคาร์บอนไดออกไซด์ในอากาศนั้น เครื่องมือนี่สามารถเปลี่ยนการวัดแก๊สหรือวัดสารละลายได้ โดยการเปลี่ยนโมดูลแก๊สเซนเซอร์ออก แล้วใส่โมดูลอื่นเข้าไปได้ทันที จากนั้นให้ทำการหาช่วงเวลาที่ทำให้เครื่องมือเสถียรและทำการสอบเทียบเครื่องมือ ซึ่งจะทำให้สามารถใช้ในการวัดปริมาณสารที่เราต้องการได้ เป็นการลดต้นทุนและเพิ่มโอกาสในการใช้เครื่องมือทางวิทยาศาสตร์ได้มากขึ้น

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

เอกสารอ้างอิง

1. Gerald, C.; Anzalone,1.; Alexandra, G.; Glover, M.; Joshua, M. Open-Source Colorimeter. *Sensors*, **2013**, 13, 5338-5346.

2. Florent,G.; Mar´ıa Viqueira,V.; Bego˜na Garc´ıa,Z.; Amaia M´endez,Z. Wireless Prototype Based on Pressure and Bending Sensors for Measuring Gate Quality. Sensors. 2013, 13, 9679-9703.

3. Fernando,C.; Osslan,O.; Dulce,E.; Vianey,G.; Humberto,O. Smart Multi-Level Tool for Remote Patient Monitoring Based on a Wireless Sensor Network and Mobile Augmented Reality. Sensors. 2014, 14, 17212-17234.

4. Salvatore,G.; Alessandro,M.; Mirko,M.; Jacopo,P.; Alberto,P. An Open-Source and Low-Cost Monitoring System for Precision Enology. Sensors 2014, 14, 23388-23397.

5. http://www.thaieasyelec.com/article-wiki/basic-electronics/%E0%B8%9A%E0%B8%97%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1-arduino-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-%E0%B9%80%E0%B8%A3%E0%B8%B4%E0%B9%88%E0%B8%A1%E0%B8%95%E0%B9%89%E0%B8%99%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99-arduino.html (Accessed March 2016)

6. http://www.thaieasyelec.com/article-wiki/review-product-article/gas-sensor-getting-started.html (Accessed March 2016)

7. http://davidegironi.blogspot.com/2014/01/cheap-co2-meter-using-mq135-sensor-with.html (Accessed April 2016)

8. http://www.ifix.com.tw/files/%E6%9E%B6%E6%A7%8B.jpg (Accessed March 2016)

9. http://aqmthai.com (Accessed May 2016)

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

ชุดคำสั่งที่อยู่ในเครื่องมือตรววัดปริมาณแก๊ส

อธิบายชุดฟังก์ชั่นต่าง ๆ

void sendData()	ใช้สำหรับส่งข้อมูลเมื่อเชื่อมต่อกับอุปกรณ์ควบคุมที่เป็นคอมพิวเตอร์หรือสมาร์ทโฟน
void sendDataTS()	ส่งข้อมูลสู่เว็บไซต์ Thingspaek เมื่อเชื่อมต่อกับ Wi-Fi Hotspot
void reads()	อ่านค่าจากโมดูลแก๊สเซนเซอร์
boolean response()	ตรวจสอบว่ามีการตอบสนองจากโมดูลไวเลส ESP8266 หรือไม่
void sendAndWait(String command, int delays)	ส่งข้อมูลไปยังโมดูลไวเลส ESP8266 แล้วรอตอบกลับ
void receiveData()	รับข้อมูลจากโมดูลไวเลส ESP8266
void doCommand()	รับคำสั่งจากผู้ใช้งาน
void stopSending()	สั่งให้เครื่องตรวจวัดหยุดการส่งข้อมูล
void softwareReset()	เริ่มการทำงานของเครื่องมืออีกครั้ง
void createHost()	เมื่อเชื่อมต่อกับอุปกรณ์ควบคุมจะทำการค้นหาที่อยู่ของ อุปกรณ์ควบคุม
void setup()	ฟังก์ชันหลักใช้ในการตั้งค่าเริ่มต้นของเครื่องมือ
void loop()	ฟังก์ชันหลักใช้โดยเครื่องมือจะงานตามฟังก์ชันนี้เป็นหลักตลอดเวลา
void serverMode()	การทำงานของเครื่องมือเมื่อเชื่อมต่อกับ Wi-Fi Hotspot
void preparing()	ให้แรงดันไฟฟ้าเพื่อใช้งานโมดูลไวเลส ESP8266
void setupWifi()	สำหรับเชื่อมต่อกับอุปกรณ์ควบคุม
boolean standCon()	ตรวจสอบว่าเชื่อมต่อกับอุปกรณ์ควบคุมที่เป็น Wi-Fi Hotspot หรือไม่
boolean normalCon()	ตรวจสอบว่าเชื่อมต่อกับอุปกรณ์ควบคุมที่เป็น คอมพิวเตอร์และสมาร์ทโฟนหรือไม่
boolean serverCon()	ตรวจสอบว่าเชื่อมต่อกับอุปกรณ์ควบคุมที่เป็น คอมพิวเตอร์และสมาร์ทโฟนหรือไม่สำหรับตั้งค่าเครื่องมือโดยตรง
void normalMode()	การทำงานของเครื่องมือเมื่อเชื่อมต่อกับคอมพิวเตอร์และสมาร์ทโฟน

คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

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

SoftwareSerial wifi(10,11);   //Connected from ESP8266 to arduino
                              //(rx,tx) Serial port

// Arduino
String data = "";
int delays = 1000;
int mode = 0;
String res = "";
float value;
float sec;

// VOC Sensor
int gnd = 9;
int VOC1 = A0;

// ESP8266 Config
String id = "1";
String protocol = "UDP";
String host = "";
String portNumber = "8888";
String hostTS = "pubsub.pubnub.com";
String portNumberTS = "80";
String api_key = "U95LZX08AYKBV29Y";
```

ศูนย์วิทยทรัพยากร
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

```
void sendData()
{
    data = id + "," + data;
    String cmd = "AT+CIPSEND=" + id + ","  + (String)data.length();
    sendAndWait(cmd,2000);
    sendAndWait(data,delays);
    Serial.println(data);
    data = "";
}
void sendDataTS()
{
    String message = "GET /subscribe/demo/VOC/0/0";
    String cmd = "AT+CIPSEND=" + id + ","  + (String)(message.length()+2);
     sendAndWait(cmd,2000);
    sendAndWait(message,delays);
    data = "";
}
void reads()
{
    float start = millis();
    float resTime = millis() - start;

  while(true)
    {
      if(wifi.available())
      {
          while(wifi.available())
          {
              if(response())  doCommand();
              char c = wifi.read();    Serial.write(c);  } break; }
```

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

```
        if(resTime > 10000)    {    break;  }
        resTime = millis() - start;
        Serial.println(resTime);
    }
{     if(response())  doCommand();   while (wifi.available())  {char c = wifi.read();    Serial.write(c);}} //
Clear Buffer
}

boolean response()
{
    if(wifi.find("+IPD,"))     {    char c;          while(wifi.read()!=58);
// to remove <id>,<len> of data
                              c = wifi.read();      do{ res += c;   c = wifi.read();   }while(c!=13);    return
true;}       // to get data
    return false;
}

void sendAndWait(String command, int delays)
{     Serial.println(command);  wifi.println(command);    reads();  }

void receiveData()  {    data = String(analogRead(VOC1),DEC);  }

void doCommand()
{
      int choice = res.toInt();
      Serial.print("do Command");
      Serial.print(choice);
      // 0 = Stop , 1 =
      switch(choice)
```

```
    {
        case 0 :{
            stopSending();
            break;
        }
        default :{
            break;
        }
    }
}

void stopSending()
{
    boolean check = true;
    res = "";

    while(check)
    {
        if(wifi.available())
        {
            if(wifi.find("+IPD,"))     {   char c;          while(wifi.read()!=58);
// to remove <id>,<len> of data
                                        c = wifi.read();      do{ res += c;   c = wifi.read();   }while(c!=13);   }
// to get data

            if(!(res == ""))  {   Serial.print("StopSending"); int cmd = res.toInt();   switch(cmd) { case 0:{
sendAndWait("AT+CIPCLOSE",500);  res = "";  softwareReset(); break;  } } }

        }
    }
```

```
}

void softwareReset() // Restarts program from beginning but does not reset the peripherals and registers
{
    asm volatile ("  jmp 0");
}

void createHost()                                   // for normal mode only if you use stand mode you must use "hostTS" variable
{
    int count=0;
    wifi.println("AT+CIFSR");
    delay(1000);
    if(wifi.find("192.168.4.1"))                    // default ip of ESP8266
    {
        wifi.read();                                // to remove new line character
        if (wifi.available()) {
            char cut = wifi.read();
            while (wifi.available() && count < 3)           // check host xxx.xxx.xxx.
            {
                // The esp has data so display its output to the serial window
              char c = wifi.read(); // read the next character.
              host += c;
              if((int)c == 46)     count++;
            }
        }
    }

    host += "1";
```

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

```
    reads();
}

void setup() {

  // Set up port for VOC-Sensor and ESP8266
  pinMode(gnd,OUTPUT);
  pinMode(VOC1,INPUT);
  preparing();

  Serial.begin(9600);
  wifi.begin(9600);

  setupWifi();                    //  Setting ESP8266 to be Server and Multiconnection
  float value = millis();
}

void loop() {

    if(serverCon())                    // if device don't have any connection it will be transform to
server for set-up
    {
        serverMode();
    }

    else if(normalCon())               // Check connection from Master Arduino into Standalone
Mode
    {
        normalMode();
    }
```

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

```
    else if(standCon())          // Check connection from other device such as computer,android,ios
into Normal Mode
    {
        //standMode();
    }

}

void serverMode()                          // สำหรับต่อกับ Wi-Fi Hotspot
{
    Serial.println("Server Mode");
    protocol = "TCP";

    while(true)
    {
        String message = "AT+CIPSTART=" + id + "," + "\"" + protocol + "\"" + "," + "\"" + hostTS + "\""
+ "," + portNumberTS;
        sendAndWait(message,delays);
        receiveData();
        sendDataTS();
        //sendAndWait("AT+CIPCLOSE",delays);
        sec = millis()-value;
        value = millis();

        Serial.print("Time:");    Serial.println(sec/1000.0);
    }

}
```

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

```
void preparing()
{
  digitalWrite(gnd, LOW);
  delay(500);
  digitalWrite(gnd, HIGH);
}

void setupWifi()
{
    sendAndWait("AT+RST",delays);
    sendAndWait("AT+CIPMUX=1",delays);
    //sendAndWait("AT+CIPSERVER=1,8888",delays);
    wifi.flush();
}

boolean standCon()
{
    sendAndWait("AT+CWJAP=\"ESP_F8C491\",\"\"",6000);
    //sendAndWait("AT+CIPSTART=1,\"TCP\",\"192.168.4.100\",8888,0",delays);

    if(wifi.find("\n\nOK")) {   return true;  }

     return false;

}

boolean normalCon()
{
    float start = millis();
    float resTime = millis() - start;
```

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

```
    wifi.println("AT+CWJAP=\"Wasitpon Sinthuphon\",\"5533148623\"");
    wifi.println("AT+CWJAP=\"Wasitpon Sinthuphon\",\"5533148623\"");
    Serial.println("AT+CWJAP=\"Wasitpon Sinthuphon\",\"5533148623\"");

    while(true)
    {
      if(wifi.available())
      {
          while(wifi.available() > 0)
          {
              if(wifi.find("OK")) {   return true;  }
          }
      }

      if(resTime > 6000)    {   break;  }
      resTime = millis() - start;

    }
     return false;

}
```

boolean serverCon()

```
{
    float start = millis();
    float resTime = millis() - start;

    wifi.println("AT+CWJAP=\"Wasitpon Sinthuphon\",\"5533148623\"");
    Serial.println("AT+CWJAP=\"Wasitpon Sinthuphon\",\"5533148623\"");
```

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

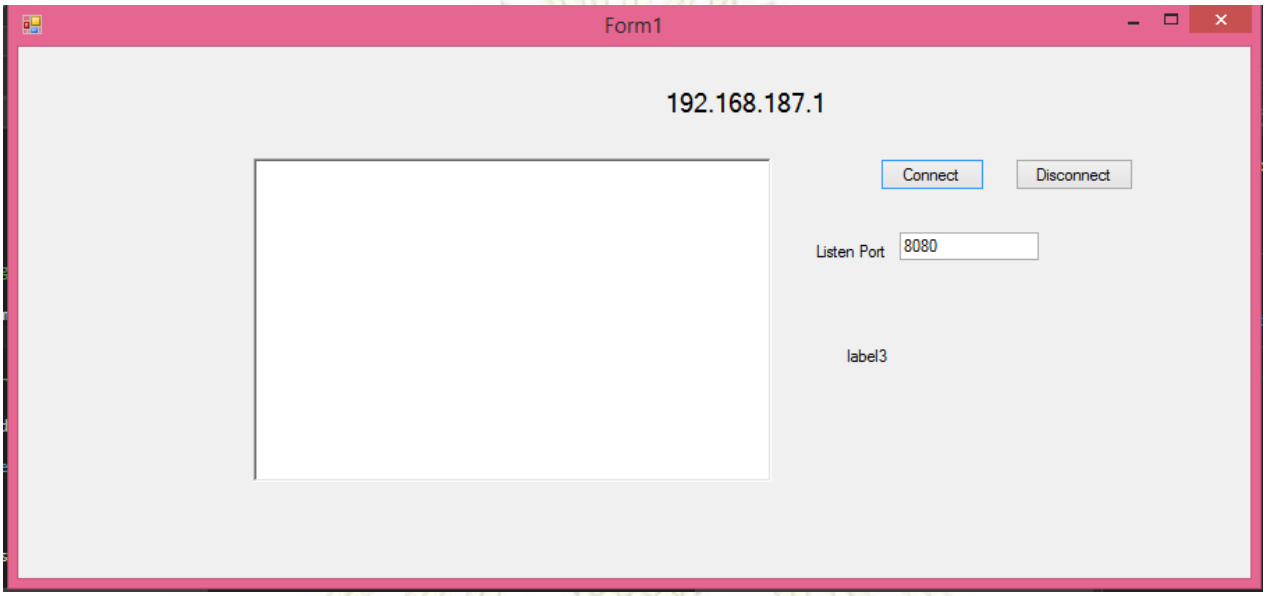
```
    while(true)
    {
      if(wifi.available())
      {
          while(wifi.available() > 0)
          {
              if(wifi.find("OK")) {   return true;  }
          }
      }

      if(resTime > 6000)    {   break;  }
      resTime = millis() - start;
    }
     return false;
}

void normalMode()
{
    Serial.println("Normal Mode");
    String cmd = "";

    sendAndWait("AT+CIFSR",delays);
    createHost();
    String message = "AT+CIPSTART=" + id + "," + "\"" + protocol + "\"" + "," + "\"" + host + "\"" + ","
+ portNumber;
    sendAndWait(message,3000);

    sec = millis()-value;
    Serial.println(sec/1000.0);
```

```
    while(true)
    {
        receiveData();
        sendData();

        sec = millis()-value;
        value = millis();

        Serial.print("Time:");    Serial.println(sec/1000.0);
    }
}
```

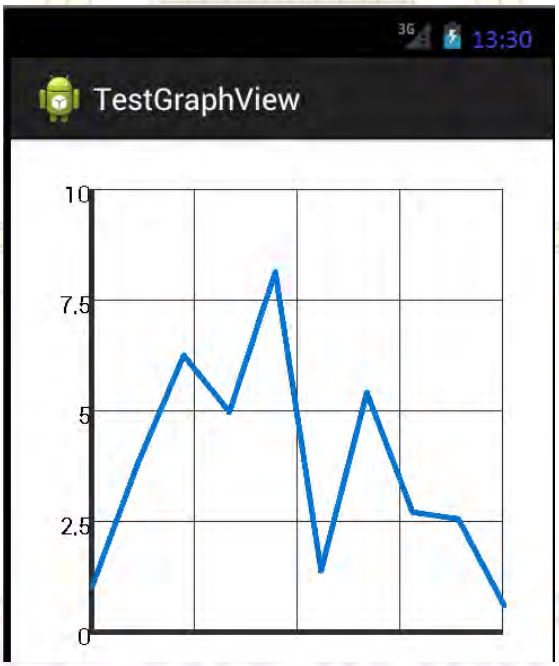
ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

โปรแกรมสำหรับติดต่อกับเครื่องมือตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซด์ในอากาศ

คอมพิวเตอร์



สมาร์ทโฟน



จุฬาลงกรณ์มหาวิทยาลัย

ราคาที่ใช้ทำเครื่องมือตรวจวัดปริมาณแก๊สคาร์บอนไดออกไซค์ในอากาศ

1.	**ไมโครคอนโทรลเลอร์ Arduino**	300.0	**บาท**
2.	**โมดูลไวเลสรุ่น ESP8266**	180.0	**บาท**
3.	**โมดูลแก๊สเซนเซอร์รุ่น MQ-135**	220.0	**บาท**
4.	**ถ่านชาร์จขนาด 9 โวลต์**	280.0	**บาท**
5.	**รางถ่านไฟฉายขนาด 9 โวลต์**	5.0	**บาท**
	รวม	**985.0**	**บาท**

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้วิจัย

นายวศิษฐ์พล สินธุผล เกิดเมื่อวันที่ 25 พฤศจิกายน พ.ศ. 2536 ภูมิลำเนาจังหวัดปทุมธานี สำเร็จการศึกษาระดับมัธยมศึกษาปีที่ 6 จากโรงเรียนนวมินทราชินูทิศ หอวัง นนทบุรี และได้เข้าศึกษาต่อในระดับอุดมศึกษา ในหลักสูตรวิทยาศาสตรบัณฑิต ภาควิชาเคมี คณะวิทยาศาสตร์ และวิชาโท ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ ที่อยู่ที่สามารถติดต่อได้หลังจากจบการศึกษาระดับปริญญาตรี บ้านเลขที่ 115/156 หมู่ 10 ตำบลบางคูวัด อำเภอเมือง จังหวัดปทุมธานี รหัสไปรษณีย์ 17130 อีเมล์ Wasitpon.s@hotmail.com

ภาควิชาเคมี
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย