ระบบแจ้งเตือนอัคคีภัย

FIRE ALARM SYSTEM

โดย

นางสาวลลิตา	ขำครุธ	62010784
นางสาววิภาวี	เลิกพยัพ	62010828
นางสาวสุพิชญาณ์	ณัฏฐ์เมธี	62010972

อาจารย์ที่ปรึกษา

ผศ.มนต์ชัย แช่มช้อย

ผศ.ดร.พิชญ สุพรรณกูล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2565

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบแจ้งเตือนอัคคีภัย

FIRE ALARM SYSTEM

ผู้จัดทำ

1. นางสาวลลิตา ขำครุธ 62010784
2. นางสาววิภาวี เลิกพยัพ 62010828
3. นางสาวสุพิชญาณ์ ณัฏฐ์เมธี 62010972

..... **อาจารย์ที่ปรึกษา**

(ผศ.มนต์ชัย แช่มช้อย)

..... **อาจารย์ที่ปรึกษาร่วม**

(ผศ.ดร.พิชญ สุพรรณกูล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การดำเนินโครงงานเรื่อง “ระบบแจ้งเตือนอัคคีภัย” จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และความอนุเคราะห์อย่างดียิ่งจากอาจารย์ที่ปรึกษาโครงงานคือ ผศ. มนต์ชัย แช่มช้อย และอาจารย์ที่ปรึกษาร่วมคือ ผศ. ดร. พิชญ สุพรรณกูล รวมทั้งรุ่นพี่ทุกท่านในห้องปฏิบัติการทดลองที่กรุณาให้คำแนะนำ คำปรึกษา และแนวทางการแก้ไขปัญหาที่เป็นประโยชน์ต่อการศึกษา ค้นคว้าวิจัยให้โครงงานนี้ สำเร็จสมบูรณ์ยิ่งขึ้น รวมถึงสนับสนุนสถานที่ เครื่องมือ และอุปกรณ์ต่าง ๆ ที่ใช้ระหว่างการจัดทำโครงงาน ขอขอบพระคุณในความปรารถนาดีและความห่วงใยที่มีให้แก่คณะผู้จัดทำเป็นอย่างยิ่ง

ขอขอบคุณคณาจารย์และเจ้าหน้าที่ประจำภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้อบรมสั่งสอนประสิทธิ์ประสาทวิชา ความรู้ และประสบการณ์ให้แก่ผู้จัดทำ

ขอกราบขอบพระคุณ บิดา มารดา และครอบครัว ที่ให้ความรัก ความห่วงใย และเป็นกำลังใจที่สำคัญเสมอมาและที่สำคัญคือสนับสนุนให้โอกาสทางด้านการศึกษาอันมีค่ายิ่งแก่ผู้จัดทำ

นางสาวลลิตา ขำครุธ
นางสาววิภาวี เลิกพยัพ
นางสาวสุพิชญาณ์ ณัฏฐ์เมธี
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบแจ้งเตือนอัคคีภัย

FIRE ALARM SYSTEM

โดย นางสาวลลิตา ขำครุธ 62010784
นางสาววิภาวี เลิกพยัพ 62010828
นางสาวสุพิชญาณ์ ณัฏฐ์เมธี 62010972

อาจารย์ที่ปรึกษา ผศ. มนต์ชัย แช่มช้อย
อาจารย์ที่ปรึกษาร่วม ผศ. ดร.พิชญ สุพรรณกูล

บทคัดย่อ

ในปัจจุบันการเกิดอัคคีภัยนั้นเป็นเรื่องที่สามารถเกิดขึ้นได้ตลอดเวลาและมักจะเกิดในบริเวณที่ไม่มีคนสังเกตเห็นซึ่งกว่าผู้ประสบเหตุจะรู้ตัวเพลิงก็ลุกไหม้จนไม่สามารถควบคุมได้ ดังนั้นเพื่อช่วยเพิ่มความปลอดภัยให้แก่ชีวิตและทรัพย์สินภายในอาคารหรือภายในห้องต่างๆ จึงได้ทำการศึกษาระบบแจ้งเตือนอัคคีภัย โดยจะมีการแสดงค่าปริมาณเปลวไฟ ควัน และอุณหภูมิผ่านการบริการคลาวด์ของ Adafruit IO ที่แสดงผลแบบ Dashboard และจะมีการแจ้งเตือนข้อความไปยังสมาร์ทโฟน เมื่อปริมาณเปลวไฟ ควัน และอุณหภูมิ มีค่ามากกว่าหรือเท่ากับปริมาณที่กำหนด ระบบสปริงเกอร์ดับเพลิงจะทำงานเพื่อดับไฟเบื้องต้น โดยในระบบจะประกอบด้วยเซนเซอร์ตรวจจับเปลวไฟ เซนเซอร์ตรวจจับควัน และเซนเซอร์อุณหภูมิ โดยจะมีการประมวลผลข้อมูลผ่านบอร์ดไมโครคอนโทรลเลอร์เพื่อควบคุมอุปกรณ์ของระบบป้องกัน และมีการนำบอร์ดไมโครคอนโทรลเลอร์มาเป็นตัวรับส่งข้อมูลเพื่อแสดงค่าปริมาณเปลวไฟ ควัน รวมถึงอุณหภูมิผ่านการบริการคลาวด์ของ Adafruit IO ที่แสดงผลแบบ Dashboard มีการใช้ Buzzer เพื่อส่งสัญญาณเสียงเตือน และมีการนำ GSM Module มาใช้ในการแจ้งเตือนข้อความไปยังสมาร์ทโฟนเมื่อมีค่าปริมาณที่เกินจากที่กำหนด และจะมีการสั่งให้สปริงเกอร์ดับเพลิงทำงานเพื่อดับไฟเบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSTRACT

Nowadays, fires can happen all the time and often occur in unnoticed areas where before the victims are aware the fire is uncontrollable. Therefore, to help increase the safety of life and property within the building or in different rooms. Therefore, a study of the fire alarm system has been conducted. Flame, smoke and temperature values will be displayed via the Adafruit IO cloud service dashboard display and will send text notifications to a smartphone. When the amount of flame, smoke, and temperature is greater than or equal to the specified amount The fire sprinkler system will work to preliminarily extinguish the fire. The system consists of a flame detection sensor. smoke sensor and temperature sensor. The data will be processed through the microcontroller board to control the equipment of the protection system. and a microcontroller board is used as a data receiver to display the amount of flame, smoke, and temperature through the Adafruit IO cloud service that displays the Dashboard display. A buzzer is used to send warning alarm, and the GSM Module is used to alert the message to the smartphone when the quantity value exceeds the specified and fire sprinklers will be ordered to work to extinguish the fire.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		IV
สารบัญรูป		VII
สารบัญตาราง		XIV
บทที่ 1	**บทนำ**	**1**
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	1
	1.3 ขอบเขตของโครงงาน	2
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**3**
	2.1 บอร์ดไมโครคอนโทรลเลอร์Arduino Uno R3 (ATmega328P)	3
	2.2 NodeMCU ESP8266 V2	4
	2.3 SIM800L GSM/GPRS Module	5
	2.4 หลักการทำงานของเซนเซอร์ตรวจจับควัน Gas Sensor (MQ-2)	6
	2.5 เซนเซอร์วัดความชื้นและอุณหภูมิ DHT22	10
	2.6 เซนเซอร์ตรวจจับเปลวไฟด้วย Infrared	11
	2.7 Active Buzzer Module	12
	2.8 จอแสดงผล Liquid Crystal Display (LCD)	13
	2.9 Lithium battery 18650	14
	2.10 18650 Battery Shield V3	15
	2.11 Arduino Relay Module แบบ 2 ช่อง	16
	2.12 ปั๊ม DC 12V แรงดัน 8 บาร์	17
	2.13 Solenoid Valve DC 12V	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
	2.14 มินิสปริงเกอร์ (Mini Sprinkler)	19
	2.15 Adafruit IO	20
	2.16 Dashboard	21
	2.17 MQTT (Message Queue Telemetry Transport)	21
บทที่ 3	**การออกแบบและการจัดทำปริญญานิพนธ์**	**22**
	3.1 การออกแบบ	22
	3.2 เครื่องมือที่ใช้ในการทดลอง	35
	3.3 การจัดเก็บผลการทดลอง	44
บทที่ 4	**ผลการทดลอง**	**51**
	4.1 ผลการทดสอบชุดคำสั่งการทำงานของเซนเซอร์ตรวจจับควัน (MQ-2)	51
	4.2 ผลการทดสอบชุดคำสั่งการทำงานของเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared	57
	4.3 ผลการทดสอบชุดคำสั่งการทำงานของโมดูล Active Buzzer	66
	4.4 ผลการทดสอบการควบคุมการทำงานของเซนเซอร์วัดอุณหภูมิและความชื้น (DHT22)	67
	4.5 ผลการทดสอบการควบคุมการแสดงค่าอุณหภูมิและความชื้นบนหน้าจอ LCD	68
	4.6 ผลการทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าความชื้นสัมพัทธ์และค่าอุณหภูมิโดยเฉลี่ยในหน่วยองศาเซลเซียสแสดงผลบน Dashboard	69
	4.7 ผลการทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าเซนเซอร์ตรวจจับควัน (MQ-2) เซนเซอร์ตรวจจับเปลวไฟ รวมถึงความชื้นสัมพัทธ์และอุณหภูมิโดยเฉลี่ยแสดงผลพร้อมกันบนหน้า Dashboard เดียวกัน	70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
	4.8 ผลการทดสอบชุดคำสั่งการทำงานของ SIM800L GSM Module	72
	4.9 ผลการทดสอบชุดคำสั่งการทำงานของโมดูล Relay 2 Channel	79
	4.10 ผลการทดสอบการส่งค่าของเซนเซอร์จากบอร์ด Arduino Uno R3 ไปยังบอร์ด NodeMCU ESP8266 V2 แบบ Serial	82
	4.11 ผลการทดสอบประสิทธิภาพของโมดูลชาร์จแบตเตอรี่ 18650	83
	4.12 ผลการทดสอบระบบแจ้งเตือนอัคคีภัยในการสั่งงานให้มีการดับไฟเบื้องต้นและส่งข้อความ SMS แจ้งเตือนไฟไหม้	84
	4.13 การทดสอบระบบแจ้งเตือนอัคคีภัยที่มีการใช้บริการคลาวด์ของ Adafruit IO ในการสั่งงานให้มีการส่งสัญญาณเสียงเตือนและส่งข้อความ SMS แจ้งเตือนไฟไหม้ พร้อมกับมีการดับไฟเบื้องต้น	85
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**90**
	5.1 สรุปผล	90
	5.2 ข้อเสนอแนะ	91
บรรณานุกรม		**92**
ภาคผนวก ก	คำสั่งควบคุมการทำงานของเซนเซอร์ตรวจจับควัน	95
ภาคผนวก ข	คำสั่งควบคุมการทำงานของเซนเซอร์ตรวจจับเปลวไฟและการส่งสัญญาณเสียงเตือนไฟไหม้ของโมดูล Active Buzzer	99
ภาคผนวก ค	คำสั่งควบคุมการทำงานของเซนเซอร์วัดอุณหภูมิและความชื้น (DHT22)	104
ภาคผนวก ง	คำสั่งควบคุมการทำงานของการดับไฟเบื้องต้น	109
ภาคผนวก จ	คำสั่งควบคุมการทำงานของการส่งข้อความ SMS แจ้งเตือนไฟไหม	116

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	Arduino Uno R3 (ATmega328P)	4
2.2	NodeMCU ESP8266 V2	5
2.3	SIM800L GSM/GPRS Module	6
2.4	เซนเซอร์ตรวจจับควัน Gas Sensor (MQ-2)	7
2.5	การจ่ายพลังงานให้MQ-2 ที่ขา H	7
2.6	วิธีการหาค่าความต้านทาน RS แบบที่ 1	8
2.7	วิธีการหาค่าความต้านทาน RS แบบที่ 2	8
2.8	จัดรูปให้อยู่ในรูปของกฎแบ่งแรงดัน	8
2.9	เซนเซอร์สำหรับวัดอุณหภูมิและความชื้นสัมพัทธ์DHT22	10
2.10	ลำดับของข้อมูลบิตทั้งหมดในการอ่านค่าจากโมดูล DHT22	11
2.11	ลำดับข้อมูลบิตอ่านค่าจากโมดูล DHT22 และความกว้างของช่วง LOW และ HIGH	11
2.12	เซนเซอร์ตรวจจับเปลวไฟด้วย Infrared	12
2.13	Active Buzzer Module	13
2.14	จอแสดงผล Liquid Crystal Display (LCD)	14
2.15	ถ่านชาร์จ Lithium battery 18650	15
2.16	18650 Battery Shield V3	16
2.17	Arduino Relay 2 Channel	17
2.18	ปั๊ม DC 12V Green-03 แรงดัน 8 บาร์	18
2.19	หลักการทำงานของโซลินอยด์วาล์ว	18
2.20	โซลินอยด์วาล์ว	19
2.21	มินิสปริงเกอร์ 360 องศา	20
3.1	บล็อกไดอะแกรมของระบบแจ้งเตือนอัคคีภัย	23
3.2	การออกแบบการเชื่อมต่อวงจรรวมของระบบ	24
3.3	การเชื่อมต่อวงจรรวมของระบบที่ใช้ในการทดสอบ	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.4	แผนผังการทำงานของเซนเซอร์ตรวจจับควัน	25
3.5	แผนผังการทำงานของเซนเซอร์ตรวจจับเปลวไฟและการส่งสัญญาณเสียงเตือนไฟไหม้ของโมดูล Active Buzzer	27
3.6	แผนผังการทำงานของเซนเซอร์วัดอุณหภูมิและความชื้น (DHT22)	29
3.7	แผนผังการทำงานของการดับไฟเบื้องต้น	31
3.8	แผนผังการทำงานของการส่งข้อความ SMS แจ้งเตือนไฟไหม้	32
3.9	ออกแบบชุดจำลองห้องพัก	33
3.10	ชุดจำลองห้องพัก	33
3.11	การออกแบบการติดตั้งอุปกรณ์ลงกล่องพลาสติกกล่องที่ 1	34
3.12	การออกแบบการติดตั้งอุปกรณ์ลงกล่องพลาสติกกล่องที่ 2	35
3.13	การติดตั้งอุปกรณ์ลงกล่องพลาสติก	35
3.14	ไมโครคอนโทรลเลอร์ Arduino Uno R3 (ATmega328P)	36
3.15	ไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2	37
3.16	เซนเซอร์ตรวจจับควัน Gas Sensor (MQ-2)	37
3.17	เซนเซอร์ตรวจจับเปลวไฟด้วย Infrared	38
3.18	เซนเซอร์วัดความชื้นและอุณหภูมิ DHT22	38
3.19	จอแสดงผล Liquid Crystal Display (LCD)	39
3.20	SIM800L GSM/GPRS Module	39
3.21	ปั๊ม DC 12V แรงดัน 8 บาร์	40
3.22	มินิสปริงเกอร์ (Mini Sprinkler)	41
3.23	Solenoid Valve DC 12V	41
3.24	Arduino Relay Module แบบ 2 ช่อง	42
3.25	Active Buzzer Module	42
3.26	Lithium battery 18650	43
3.27	18650 Battery Shield V3	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.1	หน้าต่าง Serial Monitor ที่แสดงค่า 3 ค่า ได้แก่ 1) ค่าที่เซนเซอร์ตรวจจับควันอ่านค่าได้ 2) ค่าปริมาณควัน หน่วย ppm 3) ค่าความต้านทานภายในของเซนเซอร์	51
4.2	ทำการทดสอบเซนเซอร์ตรวจจับควันขณะที่ยังไม่มีควัน	52
4.3	หน้าต่าง Serial Monitor ที่แสดงค่าสัญญาณแอนะล็อกจากเซนเซอร์ตรวจจับควันขณะที่ยังไม่มีควัน	52
4.4	ทำการทดสอบเซนเซอร์ตรวจจับควันขณะที่มีควัน	53
4.5	หน้าต่าง Serial Monitor ที่แสดงค่าสัญญาณแอนะล็อกจากเซนเซอร์ตรวจจับควันขณะที่มีควัน	53
4.6	การทดสอบการตรวจจับควันไฟด้วยเซนเซอร์ตรวจจับควัน (MQ-2)	54
4.7	ความสัมพันธ์ระหว่างค่าปริมาณควันกับระยะเวลาในการทดสอบ	55
4.8	หน้าต่าง Serial Monitor ที่แสดงค่าที่เซนเซอร์ตรวจจับควันอ่านค่าได้และค่าปริมาณควัน หน่วย ppm	56
4.9	หน้า Dashboards แสดงค่าและกราฟที่เซนเซอร์ตรวจจับควันอ่านค่าได้และค่าปริมาณควัน หน่วย ppm	56
4.10	หน้าต่าง Serial Monitor ที่แสดงการทำงานของเซนเซอร์ตรวจจับเปลวไฟ เมื่อทำการตรวจพบเปลวไฟในพื้นที่ได้	57
4.11	หน้าต่าง Serial Monitor ที่แสดงการทำงานของเซนเซอร์ตรวจจับเปลวไฟ เมื่อทำการตรวจไม่พบเปลวไฟในพื้นที่	57
4.12	การทดสอบการทำงานของเซนเซอร์ตรวจจับเปลวไฟ	58
4.13	การทดสอบการตรวจจับเปลวไฟของเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared	58
4.14	มุมบนแสดงตำแหน่งที่จุดเทียนในการทดสอบ	59
4.15	ความสัมพันธ์ระหว่างค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อกกับตำแหน่งการทดสอบ	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.16	เปอร์เซ็นต์ความแตกต่างระหว่างสภาวะปกติและเมื่อจุดไฟ ที่ระยะทางต่างๆ	64
4.17	หน้าต่าง Serial Monitor ที่แสดงค่าเอาต์พุตสัญญาณแอนะล็อกที่เซนเซอร์ตรวจจับเปลวไฟอ่านค่าได้	65
4.18	หน้า Dashboard ที่แสดงค่าเอาต์พุตสัญญาณแอนะล็อกของเซนเซอร์ตรวจจับเปลวไฟ	65
4.19	หน้าต่าง Serial Monitor ที่แสดงการทำงานของ Active Buzzer	66
4.20	หน้าต่าง Serial Monitor ที่แสดงการทำงานของ Active Buzzer เมื่อทำการตรวจจับพบควันไฟหรือเปลวไฟในพื้นที่	67
4.21	หน้า Dashboard ที่แสดงสถานะการเปิดการทำงานของโมดูล Active Buzzer	67
4.22	หน้าต่าง Serial Monitor ที่แสดงค่าความชื้นสัมพัทธ์และค่าอุณหภูมิโดยเฉลี่ย	68
4.23	การแสดงค่าอุณหภูมิและความชื้นบนหน้าจอ LCD	68
4.24	หน้าจอแสดงผล Liquid Crystal Display (LCD) จะแสดงค่าอุณหภูมิและความชื้น	58
4.25	หน้า Dashboard ที่แสดงค่าความชื้นสัมพัทธ์และค่าอุณหภูมิโดยเฉลี่ย	69
4.26	หน้า Dashboard ที่แสดงค่าเซนเซอร์ตรวจจับควัน เซนเซอร์ตรวจจับเปลวไฟ รวมถึงค่าความชื้นสัมพัทธ์และอุณหภูมิโดยเฉลี่ย	71
4.27	หน้าจอแสดงผล Liquid Crystal Display (LCD) จะแสดงค่าอุณหภูมิและความชื้น	72
4.28	ภาพการทดสอบการส่งค่าเซนเซอร์ตรวจจับควัน เซนเซอร์ตรวจจับเปลวไฟ รวมถึงค่าความชื้นสัมพัทธ์และอุณหภูมิโดยเฉลี่ยแสดงผลพร้อมกันบนหน้า Dashboard เดียวกัน	72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.29	หน้าต่าง Serial Monitor ที่แสดงว่า SIM800L GSM Module สามารถเชื่อมต่อสัญญาณเข้ากับซิมได้แล้ว	73
4.30	หน้าต่าง Serial Monitor ที่แสดงว่าสามารถส่ง SMS ไปหาเบอร์โทรศัพท์ที่ต้องการได้	74
4.31	ข้อความที่ถูกส่ง SMS ไปหาเบอร์โทรศัพท์ที่กำหนด	74
4.32	หน้าต่าง Serial Monitor ที่แสดงการทำงานของเซนเซอร์ตรวจจับควันและเซนเซอร์ตรวจจับเปลวไฟ เมื่อทำการตรวจจับพบควันไฟหรือเปลวไฟในพื้นที่ และสามารถส่ง SMS ไปหาเบอร์โทรศัพท์ที่ต้องการได้	75
4.33	ข้อความแจ้งเตือนไฟไหม้ที่ถูกส่ง SMS ไปยังสมาร์ทโฟนของผู้ดูแล	75
4.34	หน้าต่าง Serial Monitor ที่แสดงการทำงานของเซนเซอร์ตรวจจับควันและเซนเซอร์ตรวจจับเปลวไฟ เมื่อทำการตรวจจับไม่พบควันไฟและเปลวไฟในพื้นที่	76
4.35	หน้า Dashboard ที่แสดงค่าเซนเซอร์ตรวจจับควัน เซนเซอร์ตรวจจับเปลวไฟ รวมถึงค่าความชื้นสัมพัทธ์และอุณหภูมิโดยเฉลี่ย	77
4.36	หน้าจอแสดงผล Liquid Crystal Display (LCD) จะแสดงค่าอุณหภูมิและความชื้น	78
4.37	หน้าต่าง Serial Monitor ที่แสดงการทำงานของเซนเซอร์ตรวจจับควันและเซนเซอร์ตรวจจับเปลวไฟ เมื่อทำการตรวจจับพบควันไฟหรือเปลวไฟในพื้นที่และสามารถส่ง SMS ไปหาเบอร์โทรศัพท์ที่ต้องการได้	78
4.38	ข้อความแจ้งเตือนไฟไหม้ที่ถูกส่ง SMS ไปยังสมาร์ทโฟนของผู้ดูแล	78
4.39	หน้าต่าง Serial Monitor ที่แสดงการทำงานของเซนเซอร์ตรวจจับควันและเซนเซอร์ตรวจจับเปลวไฟ เมื่อทำการตรวจจับไม่พบควันไฟและเปลวไฟในพื้นที	79
4.40	หน้าต่าง Serial Monitor ที่แสดงว่าโซลินอยด์วาล์ว และปั๊ม DC 12V เริ่มทำงาน	79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.41	ทำการทดสอบโมดูล Relay 2 Channel เพื่อทำการสั่งงานให้มีการดับไฟเบื้องต้น	80
4.42	หน้าต่าง Serial Monitor ที่แสดงการทำงานของเซนเซอร์ตรวจจับควันและเซนเซอร์ตรวจจับเปลวไฟ เมื่อทำการตรวจจับพบควันไฟหรือเปลวไฟในพื้นที่รวมถึงโซลินอยด์วาล์ว และปั๊ม DC 12V เริ่มทำงาน	81
4.43	มินิสปริงเกอร์ (Mini Sprinkler) ทำงานในการดับไฟเบื้องต้น	81
4.44	หน้าต่าง Serial monitor ที่แสดงค่าที่ใช้บอร์ด Arduino Uno R3 เป็นตัวส่ง	82
4.45	หน้าต่าง Serial monitor ทำการแสดงค่าที่ใช้ NodeMCU ESP8266 V2 เป็นตัวรับค่า	83
4.46	ความสัมพันธ์ระหว่างค่าความจุแบตเตอรี่ทั้ง 3 ก้อนกับระยะเวลาในการใช้งาน	84
4.47	มินิสปริงเกอร์ (Mini Sprinkler) ทำงานในการดับไฟเบื้องต้น	85
4.48	ข้อความแจ้งเตือนไฟไหม้ที่ถูกส่ง SMS ไปยังสมาร์ทโฟนของผู้ดูแล	85
4.49	การทดสอบระบบแจ้งเตือนอัคคีภัย	86
4.50	หน้า Dashboard ที่แสดงค่าเซนเซอร์ตรวจจับควัน เซนเซอร์ตรวจจับเปลวไฟ ค่าความชื้นสัมพัทธ์และค่าอุณหภูมิโดยเฉลี่ย รวมถึงสถานะการทำงานของโมดูล Active Buzzer และโมดูล Relay 2 Channel ที่ทำงานพร้อมกันบนหน้า Dashboard เดียวกัน	87
4.51	หน้าจอแสดงผล Liquid Crystal Display (LCD) จะแสดงค่าอุณหภูมิและความชื้น	88
4.52	หน้าต่าง Serial Monitor แสดงการทำงานของเซนเซอร์ตรวจจับควันและเปลวไฟ เมื่อทำการตรวจจับพบควันไฟหรือเปลวไฟในพื้นที่ และการทำงานของโมดูล Active Buzzer ที่สร้างเสียงเตือน รวมถึงโมดูล Relay 2 Channel เริ่มทำงาน และสามารถส่ง SMS ไปหาเบอร์โทรศัพท์ที่ต้องการ	88

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่

4.53	ข้อความแจ้งเตือนไฟไหม้ที่ถูกส่ง SMS ไปยังสมาร์ทโฟนของผู้ดูแล	88
4.54	มินิสปริงเกอร์ (Mini Sprinkler) ทำงานในการดับไฟเบื้องต้น	89
4.55	หน้า Dashboard ที่แสดงสถานะการเปิดการทำงานของโมดูล Active Buzzer และโมดูล Relay 2 Channel	89
4.56	หน้าต่าง Serial Monitor ที่แสดงการทำงานของเซนเซอร์ตรวจจับควันและเซนเซอร์ตรวจจับเปลวไฟ เมื่อทำการตรวจจับไม่พบควันไฟและเปลวไฟในพื้นที่	89

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่

4.1 การทดสอบการตรวจจับควันไฟของเซนเซอร์ตรวจจับควัน (MQ-2) ในการเผาวัสดุต่างๆ 54

4.2 การทดสอบการตรวจจับเปลวไฟของเซนเซอร์ตรวจเปลวไฟ Infrared ครั้งที่ 1 59

4.3 การทดสอบการตรวจจับเปลวไฟของเซนเซอร์ตรวจเปลวไฟ Infrared ครั้งที่ 2 60

4.4 การทดสอบการตรวจจับเปลวไฟของเซนเซอร์ตรวจเปลวไฟ Infrared ครั้งที่ 3 62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ที่อยู่อาศัยเป็นปัจจัยพื้นฐานของมนุษย์ ซึ่งเป็นทรัพย์สินที่มีค่า ดังนั้นความปลอดภัยภายในที่พักถือเป็นสิ่งสำคัญในการรักษาป้องกันชีวิตและทรัพย์สินที่อยู่ภายในที่พักอาศัย อัคคีภัยเป็นหนึ่งในสาเหตุที่สร้างความเสียหายแก่ทรัพย์สิน ทำให้บาดเจ็บและสูญเสียชีวิต อัคคีภัยจึงเป็นภัยอันตราย สภาวะของไฟจะรุนแรงมากขึ้นถ้าการไหม้ที่มีเชื้อเพลิงหนุนเนื่องหรือมีไอของเชื้อเพลิงถูกขับออกมามากความร้อนก็จะมากยิ่งขึ้น สร้างความสูญเสียให้ทั้งชีวิตและทรัพย์สิน สร้างความเสียหายลุกลามต่อไปเป็นวงกว้าง ปริญญานิพนธ์นี้มีวัตถุประสงค์เพื่อที่จะป้องกันอันตรายหรือความเสียหายที่จะเกิดขึ้นกับบุคคลและทรัพย์สิน จึงได้ทำการศึกษาและจัดทำระบบแจ้งเตือนอัคคีภัย เพื่อเป็นทางเลือกหนึ่งในการแจ้งเตือนให้ผู้ใช้รับรู้ได้ทันทีถึง ความผิดปกติภายในบ้าน แม้ผู้ใช้จะอยู่นอกบริเวณบ้านก็ตาม

ดังนั้นทางผู้จัดทำจึงได้ศึกษาและมีแนวคิดที่จะนำเทคโนโลยีต่าง ๆ มาประยุกต์ใช้กับระบบแจ้งเตือนอัคคีภัยเพื่อช่วยเพิ่มความปลอดภัยภายในที่พักอาศัย โดยอาศัยการทำงานของอุปกรณ์อิเล็กทรอนิกส์ และไมโครคอนโทรลเลอร์มาใช้ภายในระบบ ที่มีการใช้เซนเซอร์ตรวจจับเปลวไฟ เซนเซอร์ตรวจจับควัน รวมถึงเซนเซอร์วัดความชื้นและอุณหภูมิทำงานร่วมกับการประมวณผลด้วยบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 และ NodeMCU ESP8266 V2 โดยสั่งการด้วยชุดคำสั่งผ่านโปรแกรม Arduino IDE เพื่อให้สามารถแสดงค่าสถานะเปลวไฟ ควัน และอุณหภูมิ ผ่านการบริการคลาวด์ของ Adafruit IO ที่แสดงผลแบบ Dashboard และเมื่อมีปริมาณค่าเปลวไฟ ควัน และอุณหภูมิที่เกินจากที่กำหนดจะมีการส่งสัญญาณเสียงเตือน และแจ้งเตือนข้อความไปยังสมาร์ทโฟนในรูปแบบอัตโนมัติ รวมทั้งสปริงเกอร์ดับเพลิงจะทำงานเพื่อดับไฟเบื้องต้น

1.2 วัตถุประสงค์

1) เพื่อจัดทำระบบแจ้งเตือนอัคคีภัยภายในบ้านหรือที่อยู่อาศัย

2) เพื่อออกแบบและสร้างอุปกรณ์ระงับการเกิดอัคคีภัยเบื้องต้นได้

3) เพื่อศึกษาหลักการทำงานของเซนเซอร์ทั้งหมดที่ใช้ในระบบ

4) เพื่อใช้ประโยชน์ในการแจ้งเตือนผ่านระบบ GSM ส่งไปยังผู้ใช้งาน เมื่อเซนเซอร์ตรวจจับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) เพื่อใช้งานบริการคลาวด์ของ Adafruit IO ที่แสดงผลแบบ Dashboard ตรวจดูข้อมูลสถานะของระบบ

1.3 ขอบเขตของปริญญานิพนธ์

ผู้จัดทำได้ทำการออกแบบและสร้างระบบเฝ้าระวังการเกิดอัคคีภัย ซึ่งระบบประกอบไปด้วยเซนเซอร์ตรวจจับเปลวไฟ เซนเซอร์ตรวจจับควัน รวมถึงเซนเซอร์วัดความชื้นและอุณหภูมิ ที่ทำงานร่วมกับบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 เพื่อทำหน้าที่เป็นตัวรับและส่งข้อมูลไปยัง NodeMCU ESP8266 V2 หน้าที่เป็นตัวรับส่งข้อมูลจะมีการประมวลผลข้อมูลเพื่อแสดงค่าสถานะเปลวไฟ ควัน รวมถึงอุณหภูมิผ่านการบริการคลาวด์ของ Adafruit IO ที่แสดงผลแบบ Dashboard โดยมีการรับค่าอุณหภูมิจากเซนเซอร์อุณหภูมิและส่งค่าอุณหภูมิเพื่อแสดงผลที่จอ LCD โดยเมื่อมีค่าปริมาณที่ตรวจจับเกินจากที่กำหนดบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 จะมีการส่งข้อมูลเพื่อสั่งงาน GSM Module ให้ทำการแจ้งเตือนข้อความไปยังสมาร์ทโฟนของผู้ดูแลและมีการสั่งงานให้รีเลย์ทำงานเพื่อให้โซลินอยด์วาล์วและปั๊มน้ำทำงานทำให้สปริงเกอร์ทำการดับไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

ปริญญานิพนธ์ "ระบบแจ้งเตือนอัคคีภัย" ได้จัดทำเพื่อออกแบบและสร้างอุปกรณ์ที่ป้องกันการเกิดอัคคีภัยและสามารถดับไฟเบื้องต้นได้ ซึ่งมีทฤษฎีและหลักการที่เกี่ยวข้องดังต่อไปนี้

2.1 บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 (ATmega328P)

Arduino Uno R3 ดังในรูปที่ 2.1 คือบอร์ดไมโครคอนโทรลเลอร์ ATmega328P ขนาด 8 บิต เป็นบอร์ดที่ได้รับความนิยมอย่างมากเนื่องจากใช้งานง่าย มีขาสัญญาณแอนะล็อก 6 อินพุต ขาสัญญาณดิจิทัล 13 ขา ไลบรารีและอุปกรณ์เชื่อมต่อส่วนใหญ่สามารถใช้ร่วมกับ Arduino Uno ได้ ซึ่ง Arduino Uno R3 มีคุณลักษณะดังนี้ [1]

1) ไมโครคอนโทรลเลอร์ ATmega328P
2) แรงดันไฟฟ้า 5 โวลต์
3) แรงดันไฟฟ้าอินพุต (แนะนำ) 7-12 โวลต์
4) แรงดันไฟฟ้าอินพุต (จำกัด) 6-20 โวลต์
5) ดิจิทัลอินพุต และเอาต์พุต 14 บิต (6 เอาต์พุต PWM)
6) แอนะล็อกอินพุต 6 บิต
7) แรงดันและกระแสไฟที่จ่ายได้ในแต่ละบิต 5 โวลต์ 40 มิลลิแอมป์
8) แรงดันและกระแสไฟที่จ่ายได้ในแต่ละบิต 3.3 โวลต์ 50 มิลลิแอมป์
9) หน่วยความจำแบบแฟลช (Flash) 32 กิโลไบต์ (500 ไบต์ Boot Loader)
10) หน่วยความจำแรม (RAM) 2 กิโลไบต์
11) หน่วยความจำรอม (ROM) 1 กิโลไบต์
12) ความถี่ 16 เมกะเฮิรตซ์
13) ขนาด 68.6 x 53.4 มิลลิเมตร
14) น้ำหนัก 25 กรัม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.1 Arduino Uno R3 (ATmega328P) [1]

2.2 NodeMCU ESP8266 V2

NodeMCU คือ บอร์ดไมโครคอนโทรลเลอร์ที่มีลักษณะการทำงานคล้าย Arduino แต่มีลักษณะพิเศษกว่าตรงที่สามารถเชื่อมต่อกับ Wi-Fi ได้ การควบคุมการทำงานสามารถใช้โปรแกรม Arduino IDE ได้เช่นเดียวกันกับบอร์ด Arduino บอร์ด NodeMCU ประกอบไปด้วย ESP8266 (ไมโครคอนโทรลเลอร์ที่สามารถเชื่อมต่อ Wi-Fi ได้) พร้อมอุปกรณ์อำนวยความสะดวกต่างๆ เช่น พอร์ต micro-USB สำหรับจ่ายไฟ/อัปโหลดโปรแกรม, ชิพสำหรับอัปโหลดโปรแกรมผ่านสาย USB, ชิพแปลงแรงดันไฟฟ้า และขาสำหรับเชื่อมต่ออุปกรณ์ภายนอก เป็นต้น [2]

NodeMCU ESP8266 V2 ดังในรูปที่ 2.2 นี้ ถูกพัฒนาจาก NodeMCU ESP8266 V1 มีการปรับปรุงให้เล็กลงกว่าเดิมสามารถเสียบบอร์ดทดลองได้เหลือช่องให้เสียบสายไฟเพิ่มเติมได้ประสิทธิภาพดีขึ้น [3] ซึ่งเป็นการนำโมดูล ESP8266-12E มาต่อร่วมกับชิปแปลงสัญญาณ USB เป็น UART เบอร์ CP2102 ของ Slilcon Lab มีสวิตช์เพื่อเข้าสู่โหมดโปรแกรมเฟิร์มแวร์ โดยคุณสมบัติของ NodeMCU ESP8266 มีดังนี้ [4]

1) ใช้โมดูล ESP8266-12E ที่ภายในมีไมโครคอนโทรลเลอร์ 32 บิต

2) หน่วยความจำแบบแฟลช ความจุ 4 เมกะไบต์และโมดูล WiFi ในตัว

3) มีชิป CP2102 สำหรับแปลงสัญญาณพอร์ต USB เป็น UART เพื่อเชื่อมต่อคอมพิวเตอร์สำหรับโปรแกรมเฟิร์มแวร์

4) ใช้ไฟเลี้ยงภายนอก +5V มีวงจรควบคุมแรงดันไฟเลี้ยงสำหรับอุปกรณ์ 3.3V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) กระแสไฟฟ้าสูงสุด 800 mA

6) มีขาพอร์ต SPI สำหรับติดต่อกับ SD การ์ด

7) มีสวิตช์ RESET และ Flash สำหรับโปรแกรมเฟิร์มแวร์

8) ตัวรับส่งสัญญาณ Wi-Fi 802.11b / g / n

9) โปรโตคอล TCP / IP

10) มีอินพุตเอาต์พุตดิจิตอล (ลอจิก 3.3V) รวม 16 ขา

11) มีอินพุตแอนะล็อก 1 ช่อง รับแรงดันไฟตรง 0 ถึง +1Vdc เข้าสู่วงจรแปลงสัญญาณแอนะล็อกเป็นดิจิตอล ความละเอียด 10 บิต

รูปที่ 2.2 NodeMCU ESP8266 V2 [5]

2.3 SIM800L GSM/GPRS Module

โมดูล Sim800L ดังในรูปที่ 2.3 เป็นรุ่นที่มีขนาดเล็กที่สุด จากตระกูล GSM Module ทั้งหมด สามารถเชื่อมต่อ Internet Gprs รองรับการส่ง SMS และสามารถโทรออกไปยังเบอร์ต่างๆ แต่ถูกตัดความสามารถในเรื่องลำโพงและเสียงออกไป ไม่สามารถพูดคุยได้ แต่สามารถโทรเพื่อเป็นการแจ้งเตือนได้ จึงเหมาะกับงานเฉพาะ เช่นระบบแจ้งเตือนต่างๆ ระบบส่ง SMS เป็นต้น

สำหรับการต่อร่วมกับ Arduino หากใช้แหล่งจ่ายไฟจาก USB กระแสอาจจะไม่พอทำให้ SIM800L GSM/GPRS Module ไม่ทำงาน แนะนำให้แยกแหล่งจ่ายไฟ เน้นที่กระแสสูงๆ เมื่อจ่ายไฟให้กับ SIM800L GSM/GPRS Module ก็จะเริ่มทำงาน จะมี LED แสดงผล หากจับสัญญาณโทรศัพท์ ได้จะกระพริบช้าๆ แต่ถ้าหากไม่สามารถจับสัญญาณโทรศัพท์ได้ ก็จะกระพริบถี่ ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมดูลนี้รองรับแรงดันไฟ 5V ดังนั้นสามารถใช้คู่กับ TTL to USB เพื่อ debug จาก terminal ในคอมพิวเตอร์ได้ทันที แต่ต้องใช้ไฟเลี้ยงที่จ่ายกระแสได้มากกว่า 800mA (ใช้ไฟ USB จากคอมพิวเตอร์ไม่ได้) คุณสมบัติเพิ่มเติม SIM800L GSM/GPRS Module มีดังนี้ [6]

1) รองรับแหล่งจ่ายไฟ 5V

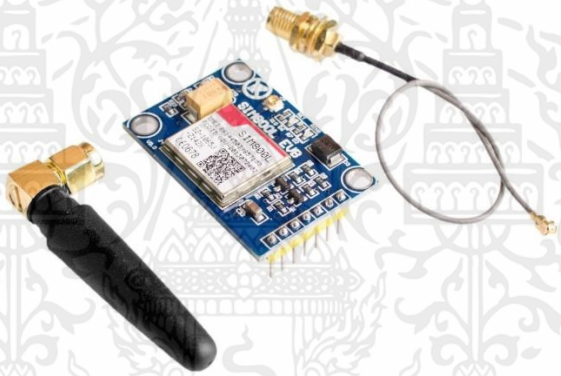
2) กระแสที่ต้องการ 800mA

3) สามารถรองรับ Serial Port ได้ทั้ง 3.3V และ 5V เหมาะกับไมโครคอนโทรลเลอร์หลากหลายรุ่น

4) ใช้ Micro sim

5) มีขนาดเล็ก เหมาะกับงาน ที่ต้องการออกแบบให้เล็ก

6) สามารถต่อเสาอากาศออกมาได้



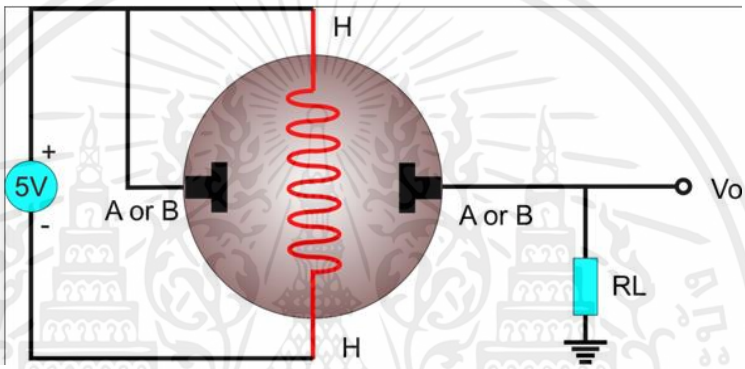
รูปที่ 2.3 SIM800L GSM/GPRS Module [7]

2.4 หลักการทำงานของเซนเซอร์ตรวจจับควัน Gas Sensor (MQ-2)

Gas Sensor (MQ-2) ดังในรูปที่ 2.4 เป็นเซนเซอร์ตรวจสอบปริมาณก๊าซไวไฟ และควัน เช่น LPG, i-butane, propane, methane, alcohol, Hydrogen, smoke ในอากาศ ซึ่งเมื่อเราเริ่มจ่ายพลังงานให้ MQ-2 ที่ขา H ทำให้เกิดความร้อนขึ้นที่ขดลวด เมื่อก๊าซไวไฟต่างๆเข้ามาทำปฏิกิริยาจะทำให้ค่าความต้านทานที่เกิดขึ้นระหว่าง ขา A และ B (RS) ลดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.4 เซนเซอร์ตรวจจับควัน Gas Sensor (MQ-2) [8]



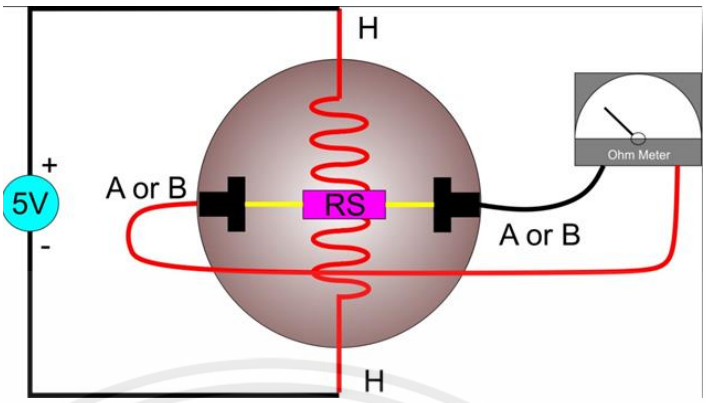
รูปที่ 2.5 การจ่ายพลังงานให้ MQ-2 ที่ขา H [9]

เมื่อจ่ายพลังงานให้กับ MQ-2 ที่ขา H ดังในรูปที่ 2.5 จะทำให้เกิดพลังงานความร้อนเพื่อให้สารเคมีภายในตัวเซนเซอร์ สามารถทำปฏิกิริยากับก๊าซไวไฟต่างๆได้ และเมื่อ MQ-2 ตรวจจับก๊าซไวไฟต่างๆ ได้จะทำให้ค่าความต้านทานระหว่างขา A และขา B เปลี่ยนแปลง (ขา A และ B เป็นขาที่ไม่ตายตัวเราสามารถกำหนดเองได้โดยเลือกขาใดเป็นขา A ขาที่อยู่ฝั่งตรงข้ามก็จะเป็นขา B) โดยเมื่อ MQ-2 ตรวจจับปริมาณก๊าซพิษต่างๆได้มากจะทำให้ค่าความต้านทาน RS ลดลง หรือค่าความต้านทานแปรผกผันกับปริมาณของก๊าซไวไฟต่างๆ

การหาค่าความต้านทาน RS สามารถทำได้ 2 วิธีคือ

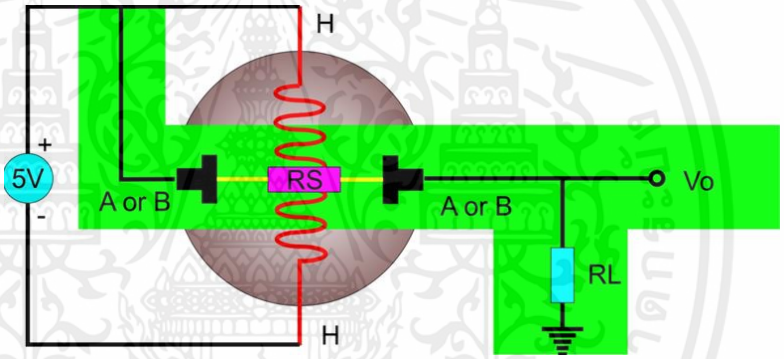
1. วิธีที่ 1 สามารถใช้ Ohm meter วัดหาค่าความต้านทานที่ขา A และ B ได้โดยตรงดังในรูปที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



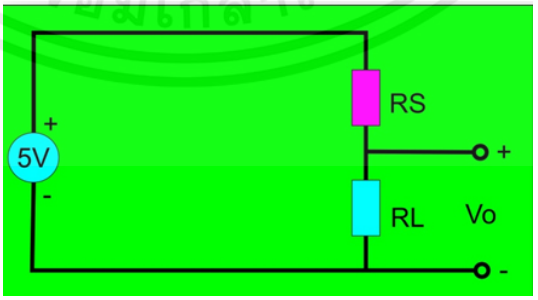
รูปที่ 2.6 วิธีการหาค่าความต้านทาน RS แบบที่ 1 [9]

2. วิธีที่ 2 เป็นวิธีการวัดโดยอ้อม โดยใช้กฎแบ่งแรงดัน Voltage Divider



รูปที่ 2.7 วิธีการหาค่าความต้านทาน RS แบบที่ 2 [9]

จากแถบสีเขียว ดังในรูปที่ 2.7 จะเห็นว่าเราจ่ายกระแสไฟฟ้าเขามาทางขา A หรือ B ทางด้านซ้าย ซึ่งเมื่อนำมาจัดเรียงใหม่ให้เป็นรูปที่เราคุ้นตากับกฎแบ่งแรงดัน ดังในรูปที่ 2.8



รูปที่ 2.8 จัดรูปให้อยู่ในรูปของกฎแบ่งแรงดัน [9]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$Vo = \left(\frac{RL}{RS + RL}\right) x\ Vin \tag{2.1}$$

$$RS = \left(\frac{RL\ x\ Vin}{Vo}\right) - RL \tag{2.2}$$

เมื่อ Vo คือ แรงดันไฟฟ้าที่ตกคร่อมความต้านทาน RS มีหน่วยเป็น โวลต์

Vin คือ แรงดันไฟฟ้าทั้งหมดของวงจรอนุกรม มีหน่วยเป็น โวลต์

RS คือ ความต้านทาน RS มีหน่วยเป็น โอห์ม

RL คือ ความต้านทาน RL มีหน่วยเป็น โอห์ม

การหาค่าความต้านทาน RS จากสมการที่ 2.2 โดยใช้กฎแบ่งแรงดัน จากสมการที่ 2.1 นี้สามารถนำไปใช้กับไมโครคอนโทรลเลอร์ได้ โดยนำค่าเอาต์พุตที่ได้ป้อนเข้าขา ADC ของไมโครคอนโทรลเลอร์ (การป้อนแรงดันเข้าขา ADC ของ MCU ต้องแน่ใจว่าระดับแรงดันที่ป้อนเข้าไปไม่เกินกว่าที่ Port ADC ของ MCU จะรับได้) แล้วแก้สมการหาค่าของ RS ออกมา และนำไปเปรียบเทียบกับปริมาณ LPG [9]

MQ-2 เซนเซอร์ตรวจจับควัน อุปกรณ์สำคัญที่ใช้ในการตรวจจับอากาศสะอาดที่เป็นก๊าซดีบุกต่ำ (SnO2) เมื่อเซนเซอร์อยู่ในสภาพแวดล้อมที่อากาศหรือก๊าซที่สามารถติดไฟได้ ตัวเซนเซอร์ตรวจจับความเข้มข้นของก๊าซในอากาศเพิ่มขึ้น วงจรไฟฟ้าจะเริ่มทำงานและสามารถนำไปต่อกับอุปกรณ์อื่นๆเพื่อทำงานต่อไป ตามความหลากหลายของโปรแกรม เซนเซอร์ตรวจจับควัน หรือเซนเซอร์ตรวจจับก๊าซเหมาะสำหรับตรวจจับก๊าซโพรเพน, ไฮโดรเจนความไวสูงหรือก๊าซธรรมชาติและอื่นๆ โดยโมดูล MQ-2 มีคุณสมบัติทางไฟฟ้า ดังนี้ [10]

1) ใช้แรงดันไฟ 5V การใช้พลังงาน (กระแสไฟ) 150 mA

2) ให้เอาต์พุตทั้งสัญญาณแอนะล็อกซึ่งเป็นค่าที่วัดได้จริง และสัญญาณดิจิตอลสามารถปรับตั้งระดับแจ้งเตือนได้

3) เอาต์พุต DO: TTL ปริมาณดิจิตอล 1 และ 0 (0.1 และ 5 V)

4) สามารถนำค่ามาคำนวณเป็น PPM ได้

5) เมื่อป้อนแรงดันให้แก่เซนเซอร์ ต้องรอการอุ่นชิพอย่างน้อย 20 วินาทีก่อนทำการวัดค่า

6) ปรับความไวของเซนเซอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 เซนเซอร์วัดความชื้นและอุณหภูมิ DHT22

อุปกรณ์เซนเซอร์สำหรับวัดอุณหภูมิและความชื้นสัมพัทธ์ DHT22 (Temperature & Relative Humidity Sensor) ดังในรูปที่ 2.9 เป็นอุปกรณ์ที่สามารถนำมาประยุกต์ใช้งานทางด้านระบบสมองกลฝังตัวได้หลากหลาย เช่น การวัด และควบคุมอุณหภูมิและความชื้น ระบบบันทึกข้อมูลเกี่ยวกับอุณหภูมิและความชื้นในห้อง เป็นต้น อุปกรณ์ประเภทนี้แตกต่างกันตามผู้ผลิต ราคา ความแม่นยำ ความละเอียดในการวัด การให้ค่าแบบดิจิทัลหรือแบบแอนะล็อกเป็นต้น การทดลองใช้งานโมดูล DHT22 ซึ่งมีราคาถูก ให้ค่าเป็นแบบดิจิทัลใช้ขาสัญญาณดิจิทัลเพียงเส้นเดียวในการเชื่อมต่อแบบบิตอนุกรมสองทิศทาง (Serial Data, Bi-Directional) โดยนำมาเชื่อมต่อกับ Arduino เพื่ออ่านค่าจากเซนเซอร์ โดยโมดูล DHT22 มีคุณสมบัติทางไฟฟ้า ดังนี้

1) ใช้แรงดันไฟเลี้ยงได้ในช่วง 3.3V ถึง 5.5V DC (ดังนั้นจึงใช้ได้กับ 3.3V และ 5V)

2) วัดอุณหภูมิได้ในช่วง -40 to 80 °C (±0.5 °C accuracy)

3) วัดความชื้นสัมพัทธ์ได้ในช่วง 0 - 100 RH% (2 - 5% accuracy)

4) อัตราการวัดสูงสุด 0.5Hz

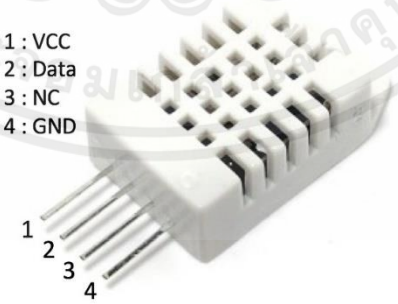
5) คอนเนคเตอร์แบบ 4 ขา (0.1" / 2.54 mm spacing)

Pin 1 = VCC

Pin 2 = SDA (Serial data, bidirectional)

Pin 3 = N.C. (Not Connect)

Pin 4 = GND



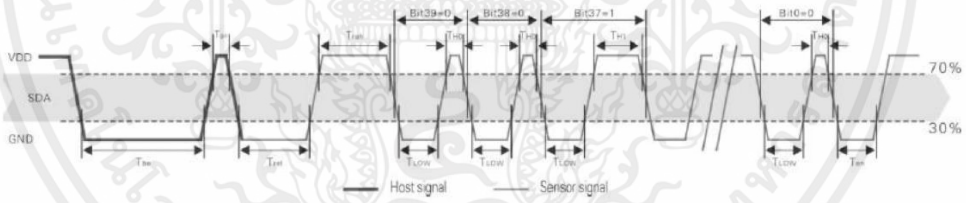
รูปที่ 2.9 เซนเซอร์สำหรับวัดอุณหภูมิและความชื้นสัมพัทธ์ DHT22 [11]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการอ่านข้อมูลจากโมดูล DHT22 นั้น ดังในรูปที่ 2.10 จะใช้ขาสัญญาณเพียงเส้นเดียวคือ DATA (หรือ SDA) แบบสองทิศทาง และ ในสถานะปกติสัญญาณ DATA จะเป็น HIGH ในการอ่านข้อมูลแต่ละครั้ง ไมโครคอนโทรลเลอร์จะต้องกำหนดให้ ขา DATA เป็นเอาต์พุต และสร้างบิต START ซึ่งจะต้องเป็น LOW อย่างน้อย 800 usec จากนั้นจึงให้เป็น HIGH อย่างน้อย 20 μsec หลังจากนั้นเป็นการรอการตอบกลับ (Response) และไอซีขา DATA จะถูกเปลี่ยนเป็นอินพุต

การเริ่มต้นของการตอบกลับโมดูล DHT22 ดังในรูปที่ 2.11 สัญญาณจะมีค่าเป็น LOW และปล่อยให้เป็น HIGH ช่วงละ 80 μsec โดยประมาณ (เรียกว่า Response Bit) จากนั้นจึงจะเป็นการส่งข้อมูลทีละบิตรวม 40 บิต (ช่วง LOW ตามด้วยช่วง HIGH) ช่วง LOW ของแต่ละบิตจะกว้างเท่ากัน แต่จะต่างกันในช่วง HIGH สำหรับบิตที่มีค่าเป็น 0 หรือ 1 (ใช้ความ กว้างช่วง HIGH ในการจำแนกค่าของบิต) [12]

รูปที่ 2.10 ลำดับของข้อมูลบิตทั้งหมดในการอ่านค่าจากโมดูล DHT22 [4]



รูปที่ 2.11 ลำดับข้อมูลบิตอ่านค่าจากโมดูล DHT22 และความกว้างของช่วง LOW และHIGH [12]

2.6 เซนเซอร์ตรวจจับเปลวไฟด้วย Infrared

เซนเซอร์ตรวจจับเปลวไฟด้วย Infrared ดังในรูปที่ 2.12 ใช้ตรวจจับเปลวไฟ โดยใช้เซนเซอร์อินฟาเรด ให้สัญญาณเอาต์พุตออกมาทั้งแบบดิจิตอลเมื่อตรวจจับได้ให้สัญญาณเป็น 1 เมื่อตรวจจับไม่พบให้สัญญาณเป็น 0 และแบบสัญญาณแอนะล็อก ให้ค่า 0-5V สามารถปรับความไวได้ที่บนโมดูล [13]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.12 เซนเซอร์ตรวจจับเปลวไฟด้วย Infrared [13]

โดยเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared มีคุณสมบัติทางไฟฟ้า ดังนี้ [14]

1) สามารถตรวจจับเปลวไฟหรือความยาวคลื่นในช่วง 760 nm ถึง 1100 nm ช่วงแหล่งกำเนิดแสง ระยะการทดสอบเปลวที่ 80 cm สำหรับเปลวไฟขนาดใหญ่มากขึ้นระยะการทดสอบจะไกลขึ้น

2) มุมตรวจจับประมาณ 60 องศา โดยเฉพาะอย่างยิ่งความไวต่อแสงของเปลวไฟ

3) ปรับความไวได้ (การปรับโพเทนชิโอมิเตอร์แบบดิจิตอลสีน้ำเงิน)

4) แรงดันไฟฟ้า 3.3V-5V

5) รูปแบบเอาต์พุตขา DO เป็นเอาต์พุตสวิตชิ่งดิจิตอล (0 และ 1) และเอาต์พุตขา AO เป็นเอาต์พุตแรงดันไฟฟ้าแอนะล็อก

2.7 Active Buzzer Module

Active Buzzer Module 3.3 – 5V ดังในรูปที่ 2.13 เป็นโมดูลลำโพงเสียงบัสเซอร์ สามารถสร้างเสียงเตือนได้อย่างง่าย ๆ เพียงแค่จ่ายไฟ เข้าไปที่ ขา I/O โมดูลนี้มีวงจรขยายช่วยขยายสัญญาณจึงมีความดังเป็นพิเศษ [15]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.13 Active Buzzer Module [15]

2.8 จอแสดงผล Liquid Crystal Display (LCD)

เป็นจอแสดงผลรูปแบบหนึ่งที่นิยมนำมาใช้งานกับระบบสมองกลฝังตัวอย่างแพร่หลาย จอแสดงผล LCD มีทั้งแบบแสดงผลเป็นตัวอักขระเรียกว่า Character LCD ซึ่งมีการกำหนดตัวอักษรหรืออักขระที่สามารถแสดงผลไว้ได้อยู่แล้ว และแบบที่สามารถแสดงผลเป็น รูปภาพหรือสัญลักษณ์ได้ตามความต้องการของผู้ใช้งานเรียกว่า Graphic LCD [16] จอแสดงผล LCD จะแบ่งเป็น 2 แบบตามลักษณะการแสดงผลดังนี้

1) Character LCD เป็นจอที่แสดงผลเป็นตัวอักษรตามช่องแบบตายตัว เช่น จอ LCD ขนาด 16×2 ตัวอักษร หมายถึงใน 1 แถว มีตัวอักษรใส่ได้ 16 ตัว และมีทั้งหมด 2 บรรทัดให้ใช้งาน ส่วน 20x4 ตัวอักษร จะหมายถึงใน 1 แถว มีตัวอักษรใส่ได้ 20 ตัว และมีทั้งหมด 2 บรรทัด การเชื่อมต่อกับจอ Character LCD จะมีด้วยกัน 2 แบบ คือ

(ก) การเชื่อมต่อแบบขนาน เป็นการเชื่อมต่อจอ LCD เข้ากับบอร์ด Arduino โดยตรง โดยจะแบ่งเป็นการเชื่อมต่อแบบ 4 บิต และการเชื่อมต่อแบบ 8 บิต ในการต่อกับ Arduino นั้นจึงนิยมต่อเพียง 4 ขาหรือ 4 บิตเท่านั้น เนื่องจากใช้สายในการเชื่อมต่อน้อยกว่า เพื่อเป็นการประหยัดขาในการต่อใช้งานไปไว้ต่อกับอุปกรณ์อื่น

(ข) การเชื่อมต่อแบบอนุกรม เป็นการเชื่อมต่อกับจอ LCD ผ่านโมดูลแปลงรูปแบบ การเชื่อมต่อกับจอ LCD จากแบบขนาน มาเป็นการเชื่อมต่อแบบอื่นที่ใช้สายน้อยกว่า เช่น การใช้โมดูล I2C Serial Interface จะเป็นการนำโมดูลเชื่อมเข้ากับตัวจอ LCD แล้วใช้บอร์ด Arduino เชื่อมต่อกับบอร์ดโมดูล ผ่านโปรโตคอล I2C ทำให้ใช้สายเพียง 2 เส้น ทำให้หน้าจอแสดงผลข้อความต่าง ๆ ออกมาได้ จอ LCD ที่มีการเชื่อมต่อแบบอนุกรม แสดงดังในรูปที่ 2.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.14 จอแสดงผล Liquid Crystal Display (LCD) [17]

2) Graphic LCD เป็นจอแสดงผลที่สามารถกำหนดได้ว่าจะให้แต่ละจุดบนหน้าจอกั้นแสง หรือปล่อยแสงออกไป ทำให้จอนี้สามารถสร้างรูปขึ้นมาบนหน้าจอได้ การระบุขนาดจะระบุในลักษณะของ จำนวนจุด (Pixels) ในแต่ละแนว เช่น 128x64 Pixels หมายถึงจอที่มีจำนวนจุดตามแนวนอน 128 Pixels และมีจุดตามแนวตั้ง 64 Pixels [16]

2.9 Lithium battery 18650

ถ่านชาร์จ 18650 ดังในรูปที่ 2.15 เหมาะสำหรับการใช้งานอิเล็กทรอนิกส์ เช่น ไฟฉาย เลเซอร์ เป็นต้น มีความจุ 1200 mAh ใช้งานได้นาน สามารถชาร์จไฟได้ถึง 500 ครั้ง โดยมีข้อมูลคือ [18]

1) หนัก 30 กรัม
2) ขนาด 1.8 x 6.5 เซนติเมตร
3) ถ่านชาร์จ 18650 ความจุ 1200 mAh
4) แบตเตอรี่ แบบ 3.7 โวลต์
5) สามารถนำกลับมาชาร์จใหม่ได้ 500 ครั้ง
6) ใช้สำหรับไฟฉาย เลเซอร์ และอุปกรณ์อิเล็กทรอนิกส์อื่นๆ
7) ปลอดภัยต่อสิ่งแวดล้อม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.15 ถ่านชาร์จ Lithium battery 18650 [18]

2.10 18650 Battery Shield V3

โมดูล Power Bank แปลงไฟจากถ่าน 18650 3.7V เป็นไฟ 5V 2A และ 3V 1A สำหรับจ่ายไฟให้กับบอร์ด Arduino , ESP8266 , Raspberry Pi โดย 18650 battery shield V3 ดังในรูปที่ 2.16 ใส่แบตเตอรี่ได้ 1 ก้อน เหมาะสำหรับอุปกรณ์กินกระแสน้อย ถึงปานกลาง มีคุณสมบัติทั่วไปคือ [19]

1) สามารถชาร์จไฟและใช้ไฟพร้อมกันได้
2) ขนาดโมดูล 9.8 x 2.9 เซนติเมตร
3) ชาร์จไฟเข้าผ่าน Micro USB กระแสชาร์จที่ 0.6 - 0.8 A
4) มีวงจรป้องกันแบตเตอรี่ในตัว
5) เอาต์พุตพอร์ตเป็น USB type A
6) เอาต์พุตพอร์ต 3V จำนวน 3 ตำแหน่ง
7) เอาต์พุตพอร์ต 5V จำนวน 3 ตำแหน่ง และควรปฏิบัติตามทิศทางบวกและลบบน PCB
8) ไฟแสดงสถานะ (แดง กำลังชาร์จ เขียว ชาร์จเต็ม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.16 18650 Battery Shield V3 [20]

2.11 Arduino Relay Module แบบ 2 ช่อง

บอร์ด Arduino relay 2 channel ดังรูปที่ 2.17 ใช้สำหรับควบคุมการเปิด/ปิด Relay Module ได้ 2 ช่อง ใช้ไฟเข้า 5 โวลต์ เหมาะสำหรับ Arduino การควบคุม ส่งสัญญาณควบคุมแบบ Active Low คือถ้าต้องการให้รีเลย์ติดจะส่งสัญญาณ 0 ไป และถ้าต้องการให้รีเลย์ดับจะส่งสัญญาณ 1 ไป วงจร Arduino relay module เป็นแบบ แยกกราวด์ Opto isolated Relay ปลอดภัยต่อวงจรไมโครคอนโทรเลอร์ และ Arduino[21]

รายละเอียด Arduino Relay module บอร์ดรีเลย์ 2 ช่อง สำหรับ Arduino

1) การเชื่อมต่อ Arduino Relay module มาตรฐานที่สามารถใช้ควบคุมได้โดยตรงจากไมโครคอนโทรนเลอร์ เช่น Arduino , 8051, AVR, PIC, DSP, ARM, ARM, MSP430, TTL logic)

2) ใช้ไฟฟ้าที่ 5 โวลต์

3) ใช้ควบคุมไฟฟ้าแรงสูงได้ที่ DC30V 10A, AC250V 10A

4) มีไฟบอกสถานะการทำงานของ Arduino Relay รีเลย์ทุกช่อง

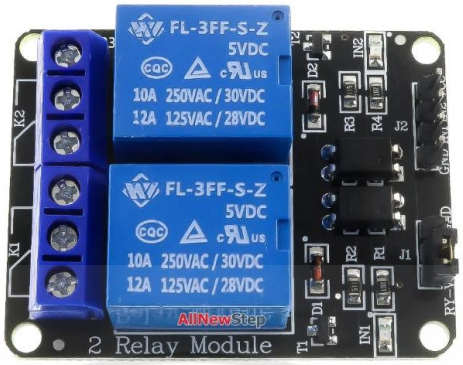
5) เชื่อมต่อด้วยขั้วสกรู ทำให้ติดตั้งได้ง่ายและสะดวก

6) ใช้กระแสขับ relay แต่ละตัวที่ 15-20 mA

7) การส่งสัญญาณควบคุมรีเลย์เป็นแบบ Active low

8) วงจรขับรีเลย์เป็นแบบแยกกราวด์ Opto isolated Relay ปลอดภัยต่อวงจรไมโครคอนโทรเลอร์ Arduino Relay

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 Arduino Relay 2 Channel [22]

2.12 ปั๊ม DC 12V แรงดัน 8 บาร์

ปั๊มพ่นหมอก DC 12V Green-03 แรงดัน 8 บาร์ 60W แบบไม่มี Pressure Switch ดังในรูปที่ 2.18 มีคุณสมบัติ ดังนี้ [23]

1) แรงดันไฟฟ้า 12VDC

2) กำลังไฟฟ้า 60W

3) กระแสไฟ 5A

4) แรงดันน้ำ 8 บาร์ (120 PSI)

5) อัตราการไหลสูงสุด 300 ลิตร/ชั่วโมง

6) อัตราการไหลสูงสุด 5 ลิตร/นาที

7) ใช้ได้กับแบตเตอรี่ 12V

8) ใช้ได้กับ Adapter 12V 5A ขึ้นไป

9) ระยะดูดไม่เกิน 1.5 m (ควรให้สั้นที่สุด)

10) ไม่มี Pressure switch สามารถคลายนอตเพื่อปรับแรงดันไม่ให้ปั๊มทำงานกระตุกกรณีเกิดการอั้น

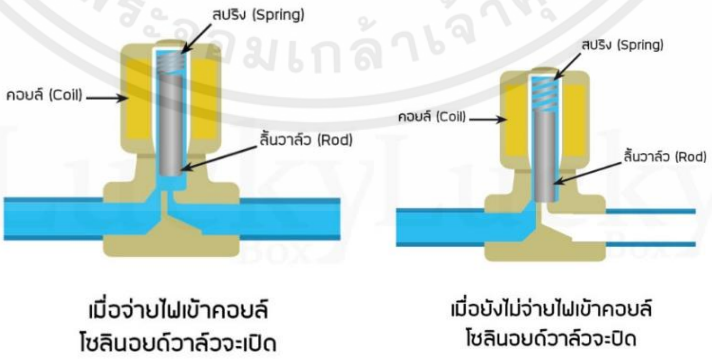
11) การทำงานต่อเนื่อง 2 ชั่วโมง (ควรพัก 30 นาที)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.18 ปั๊ม DC 12V Green-03 แรงดัน 8 บาร์ [24]

2.13 Solenoid Valve DC 12V

โซลินอยด์ (Solenoid) เป็นอุปกรณ์แม่เหล็กไฟฟ้าชนิดหนึ่ง ที่มีหลักการทำงานคล้ายกับรีเลย์ (Relay) ภายในโครงสร้างของโซลินอยด์จะประกอบด้วยขดลวดที่พันอยู่รอบแท่งเหล็กที่ภายในประกอบด้วยแม่เหล็กชุดบนกับชุดล่าง เมื่อมีกระแสไฟฟ้าไหลผ่านขดลวดที่พันรอบแท่งเหล็กทำให้แท่งเหล็กชุดล่างมีอำนาจแม่เหล็กดึงแท่งเหล็กชุดบนลงมาสัมผัสกันทำให้ครบวงจรทำงาน เมื่อวงจรถูกตัดกระแสไฟฟ้าทำให้แท่งเหล็กส่วนล่างหมดอำนาจแม่เหล็ก สปริงก็จะดันแท่งเหล็กส่วนบนกลับสู่ตำแหน่งปกติ ดังในรูปที่ 2.19 จากหลักการดังกล่าวของโซลินอยด์ก็จะนำมาใช้ในการเลื่อนลิ้นวาล์วของระบบนิวเมติกส์ การปิด-เปิดการจ่ายน้ำหรือของเหลวอื่นๆ โครงสร้างของโซลินอยด์โดยทั่วไปแบ่งออกเป็น 2 ชนิด คือเลื่อนวาล์วด้วยโซลินอยด์วาล์วกลับด้วยสปริง (Single Solenoid Valve) และเลื่อนวาล์วด้วยโซลินอยด์วาล์วกลับด้วยโซลินอยด์วาล์ว (Double Solenoid Valve) ในที่นี้ใช้แบบเลื่อนวาล์วด้วยโซลินอยด์วาล์วกลับด้วยสปริง (Single Solenoid Valve)



รูปที่ 2.19 หลักการทำงานของโซลินอยด์วาล์ว [26]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โซลินอยด์วาล์ว DC 12V ดังในรูปที่ 2.20 มีคุณสมบัติดังนี้ [25]

1) รับแรงดันได้สูงสุด : 7-10 บาร์

2) ขนาดทางเข้า-ออก : เกลียวใน 1 นิ้ว ทั้งสองด้าน

3) ระบบการทำงานที่ความร้อนสูงสุด : ทนอุณหภูมิได้สูงสุด 60-100 องศาเซลเซียส

4) ระบบการทำงานที่แรงดันสูงสุด : ทนแรงดัน 0 - 10 บาร์ (ไม่มีแรงดันก็สามารถทำงานได้)

5) สถานะการใช้งาน : น้ำ, น้ำยาแอร์, ลม, อากาศ, น้ำมัน, แก๊ส

6) แรงดันไฟฟ้า 12VDC : สายไฟสีแดง คือขั้วบวก (+), สายไฟสีดำคือขั้วลบ (-) กินกระแสไฟประมาณ 1-2 A

รูปที่ 2.20 โซลินอยด์วาล์ว [27]

2.14 มินิสปริงเกอร์ (Mini Sprinkler)

มินิสปริงเกอร์นั้นจะมีลักษณะการกระจายน้ำที่หลากหลายตั้งแต่ ละอองขนาดเล็กจนถึงขนาดใหญ่ รัศมีการให้น้ำอยู่ที่ประมาณ 1-4 เมตร แรงดันน้ำประมาณ 1.5-2 บาร์ ในที่นี้จะขอแบ่งมินิสปริงเกอร์เป็นทั้งหมด 2 กลุ่มตามอัตราการจ่ายน้ำดังนี้ [28]

2.14.1 มินิสปริงเกอร์ชนิดกระจายน้ำน้อย

อัตราการจ่ายน้ำจะอยู่ที่ 35-300 ลิตร/ชม. ลักษณะการกระจายน้ำจะมีทั้งแบบแคบและแบบไกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.14.2 มินิสปริงเกอร์ชนิดกระจายน้ำมาก

อัตราการจ่ายน้ำจะอยู่ที่ 400 -1,000 ลิตร/ชม. ลักษณะการให้น้ำมีหลากหลายรูปแบบ ไม่ว่าจะเป็นแบบพุ่งกระจาย หมุนเป็นเกลียวออก 2 ด้าน หรือแบบเป็นเม็ดฝอยขนาดใหญ่รอบทิศทาง

สปริงเกอร์ที่เลือกใช้เป็นมินิสปริงเกอร์ 360 องศา ดังในรูปที่ 2.21 ลักษณะของน้ำที่ถูกปล่อยออกมาจากหัวจ่ายน้ำแบบนี้จะมีลักษณะกระจายตัว อัตราการจ่ายน้ำประมาณ 150-400 ลิตร/ชั่วโมง โดยขึ้นอยู่กับแรงดันน้ำ ซึ่งต้องการแรงดันน้ำในการใช้งาน 1.5-3 บาร์ขึ้นไปเพื่อทำให้น้ำที่ถูกพ่นออกมากระจายตัวได้เต็มที่ตามรัศมีที่มีความกว้าง 3-5 เมตร [29]

รูปที่ 2.21 มินิสปริงเกอร์ 360 องศา [30]

2.15 Adafruit IO

Adafruit IO เป็นแพลตฟอร์มที่ออกแบบเพื่อให้บริการคลาวด์ โดยทำงานเพื่อแสดงตอบสนอง และโต้ตอบกับข้อมูลระบบของผู้ใช้บริการ ใช้สำหรับการจัดเก็บและดึงข้อมูลเป็นหลัก พร้อมทั้งรักษาข้อมูลของผู้ใช้บริการให้เป็นส่วนตัวและปลอดภัย ผ่านการเชื่อมต่ออินเทอร์เน็ต โดย Adafruit IO สามารถได้ดังนี้ [31]

1) แสดงข้อมูลของระบบผู้ใช้บริการแบบเรียลไทม์ออนไลน์

2) เชื่อมต่อระบบของผู้ใช้บริการกับอินเทอร์เน็ต เพื่อควบคุมมอเตอร์ อ่านข้อมูลเซนเซอร์ เป็นต้น

3) เชื่อมต่อระบบบริการบนเว็บไซต์ เช่น Twitter, RSS feed, บริการสภาพอากาศ เป็นต้น

4) เชื่อมต่อระบบของผู้ใช้บริการกับอุปกรณ์อื่น ๆ ที่ใช้งานอินเทอร์เน็ตได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.16 Dashboard

Dashboard คือส่วนแสดงผลเพื่อตรวจสอบและควบคุมการทำงานของระบบโดยผู้ใช้งาน ซึ่งจะอยู่ในรูปแบบของหน้าจอแสดงผลสรุปของข้อมูลภายในระบบ IoT โดยการรวบรวมสถานะทำงานของการเชื่อมต่ออุปกรณ์ และแอปพลิเคชันของระบบ IoT ทั้งหมดมาแสดงร่วมกัน [32]

2.17 MQTT (Message Queue Telemetry Transport)

MQTT เป็นโปรโตคอลการส่งข้อความที่อิงตามมาตรฐาน หรือชุดของกฎที่ใช้สำหรับการสื่อสารระหว่างเครื่องต่อเครื่อง เซนเซอร์ และอุปกรณ์ IoT อื่น ๆ มักจะต้องส่งและรับข้อมูลผ่านเครือข่ายที่มีข้อจำกัดด้านทรัพยากร ซึ่งมีแบนด์วิดท์จำกัด โดยอุปกรณ์ IoT จะใช้ MQTT ในการรับส่งข้อมูล เนื่องจากการใช้ที่งานง่ายและสามารถสื่อสารข้อมูล IoT ได้อย่างมีประสิทธิภาพ รวดเร็ว ใช้พลังงานต่ำ และรองรับการส่งข้อมูลจากอุปกรณ์ไปยังคลาวด์ และจากคลาวด์ไปยังอุปกรณ์ ซึ่งพัฒนาต่อมาจาก TCP/IP เพื่อป้องกันการหล่นหายของข้อมูลระหว่างการส่ง [33]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

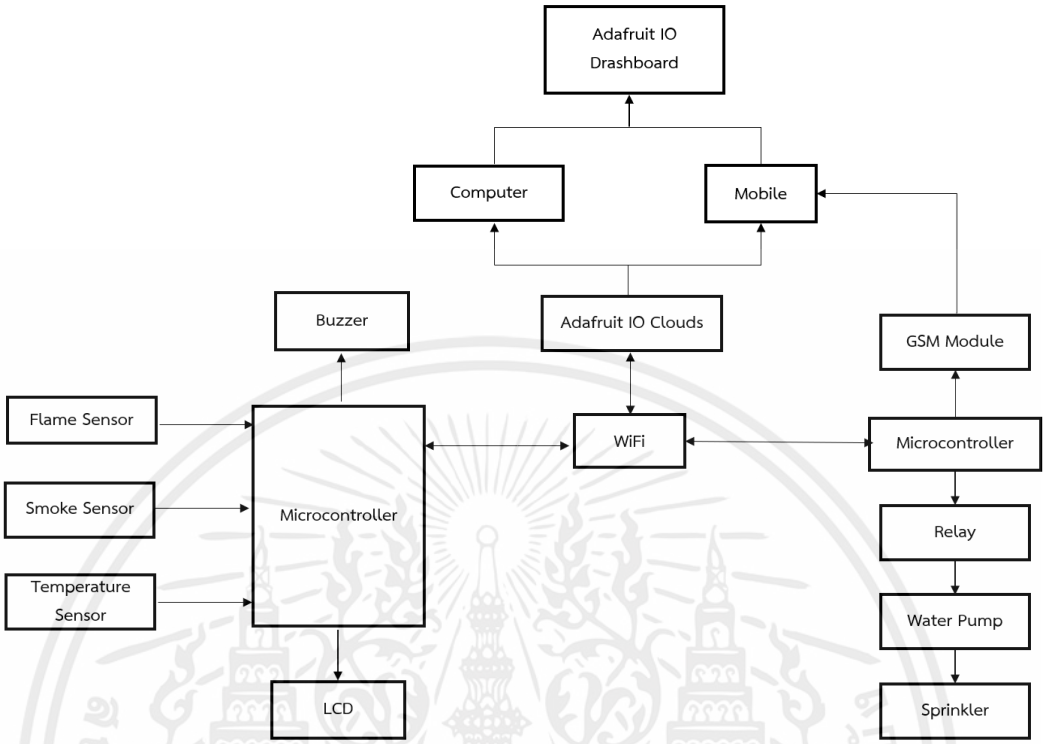
บทที่ 3
การออกแบบและการจัดทำปริญญานิพนธ์

3.1 การออกแบบ

3.1.1 การออกแบบการทำงานของระบบ

ระบบแจ้งเตือนอัคคีภัยประกอบไปด้วยเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared เซนเซอร์วัดความชื้นและอุณหภูมิ DHT22 เซนเซอร์ตรวจจับควัน MQ-2 โดยจะติดตั้งเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared และเซนเซอร์ตรวจจับควัน MQ-2 เพื่อทำหน้าที่ตรวจจับปริมาณเปลวไฟ ควัน รวมถึงอุณหภูมิโดยเฉลี่ยและความชื้นสัมพัทธ์ ซึ่งจะมีการประมวลผลข้อมูลผ่านบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 และ NodeMCU ESP8266 V2 เพื่อทำหน้าที่เป็นตัวรับส่งข้อมูลเพื่อแสดงค่าสถานะเปลวไฟ ควัน รวมถึงอุณหภูมิโดยเฉลี่ยและความชื้นสัมพัทธ์ และค่าสถานะการเปิดการทำงานของโมดูล Active Buzzer และโมดูล Relay 2 Channel ผ่านการบริการคลาวด์ของ Adafruit IO ที่แสดงผลแบบ Dashboard มีการใช้ Buzzer เพื่อส่งสัญญาณเสียงเตือน และมีการนำ GSM Module มาใช้ในการแจ้งเตือนข้อความไปยังสมาร์ทโฟนของผู้ดูแล เมื่อมีค่าปริมาณที่ตรวจจับเกินจากที่กำหนด และมีการสั่งให้สปริงเกอร์ดับเพลิงทำการดับไฟเบื้องต้น โดยข้อมูลและคำสั่งต่าง ๆ ที่ได้ออกแบบจะมาจากการเขียนโปรแกรม Arduino IDE หลักการทำงานของระบบเริ่มจากส่วนของเซนเซอร์ตรวจจับเปลวไฟ เซนเซอร์ตรวจจับควัน และเซนเซอร์วัดอุณหภูมิและความชื้นทำหน้าที่ในการตรวจจับควันและเปลวไฟ รวมถึงอุณหภูมิโดยเฉลี่ยและความชื้นสัมพัทธ์ แล้วทำการส่งค่าสถานะไปยังบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 เพื่อทำหน้าที่เป็นตัวรับส่งข้อมูลจะมีการประมวลผลข้อมูลเพื่อแสดงค่าสถานะเปลวไฟ ควัน รวมถึงค่าอุณหภูมิโดยเฉลี่ยและความชื้นสัมพัทธ์ผ่านการบริการคลาวด์ของ Adafruit IO ที่แสดงผลแบบ Dashboard และมีการส่งค่าอุณหภูมิโดยเฉลี่ยและความชื้นสัมพัทธ์เพื่อแสดงผลที่จอ LCD โดยเมื่อมีค่าปริมาณที่ตรวจจับเกินจากที่กำหนดบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 จะมีการสั่งงานไปยัง Buzzer เพื่อส่งสัญญาณเสียงเตือน และสั่งงานให้รีเลย์ทำงานเพื่อให้ปั๊มน้ำทำงานและสปริงเกอร์ดับเพลิงจะทำการดับไฟเบื้องต้น และจะมีการส่งข้อมูลเชื่อมต่อกับบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 แบบ Serial เพื่อทำหน้าที่เป็นตัวรับข้อมูลเพื่อสั่งงาน GSM Module ให้ทำการแจ้งเตือนข้อความไปยังสมาร์ทโฟนของผู้ดูแล สามารถแสดงเป็นบล็อกไดอะแกรมของระบบได้ดังในรูปที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

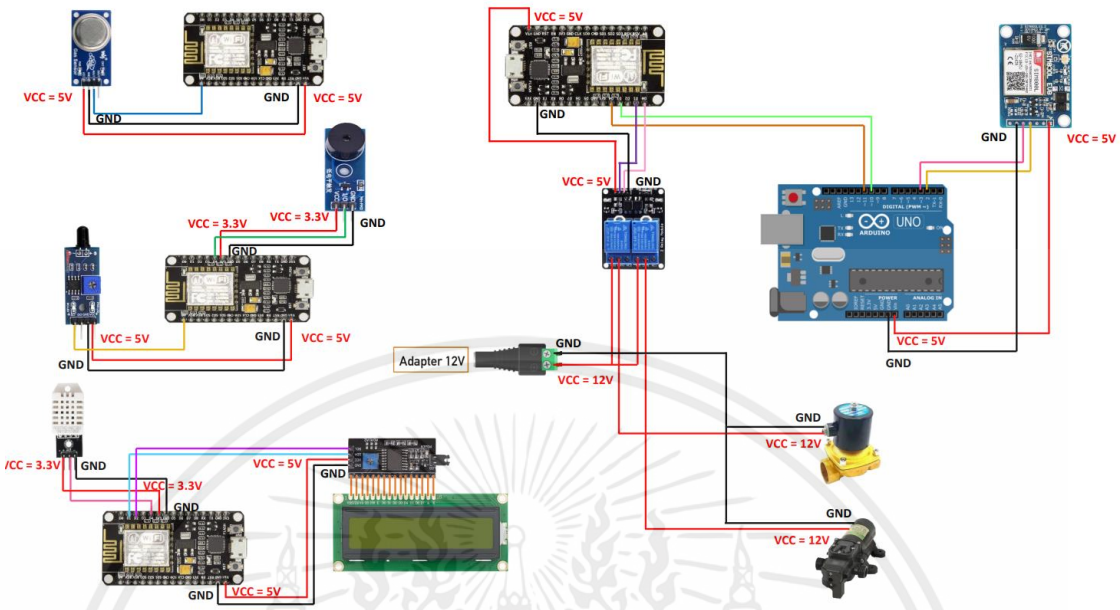


รูปที่ 3.1 บล็อกไดอะแกรมของระบบแจ้งเตือนอัคคีภัย

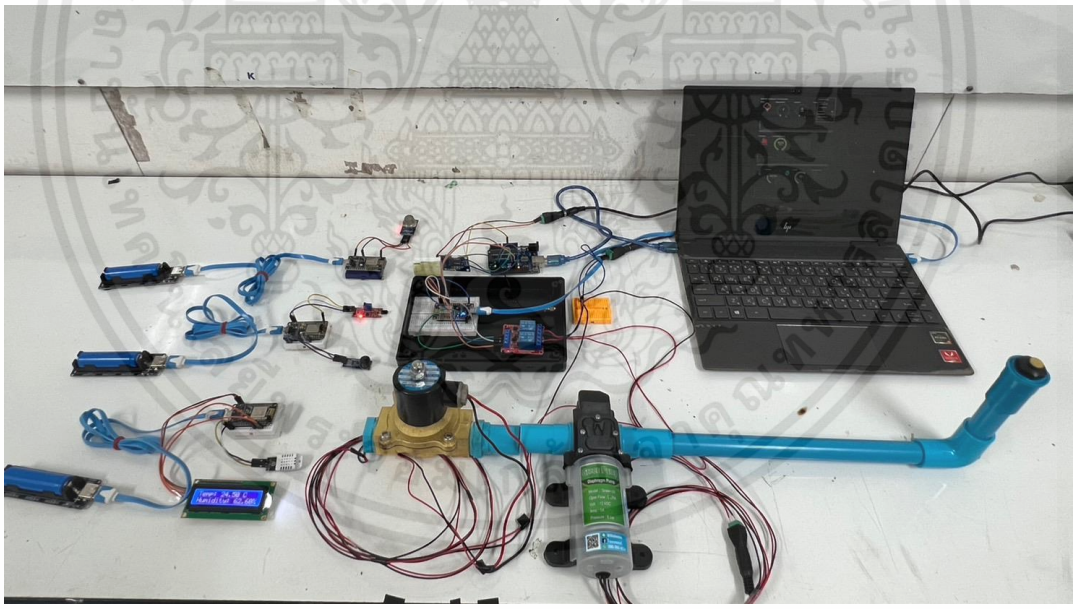
3.1.2 การออกแบบการทำงานของระบบ

ทำการเชื่อมต่ออุปกรณ์โมดูล Arduino Uno R3 กับโมดูล NodeMCU ESP8266 V2 เซนเซอร์ตรวจจับควัน (MQ-2) เซนเซอร์ตรวจจับเปลวไฟด้วย Infrared โมดูล Active Buzzer เซนเซอร์วัดอุณหภูมิและความชื้น (DHT22) จอแสดงผล Liquid Crystal Display (LCD) SIM800L GSM Module Infrared รวมถึงโมดูล Relay 2 Channel ที่เชื่อมต่ออยู่กับโซลินอยด์วาล์ว (Solenoid Valve) และปั๊ม DC 12V เพื่อสั่งการมินิสปริงเกอร์ (Mini Sprinkler) ดังในรูปที่ 3.2 และการเชื่อมต่ออุปกรณ์ที่ใช้ในการทดสอบ ดังในรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 การออกแบบการเชื่อมต่อวงจรรวมของระบบ



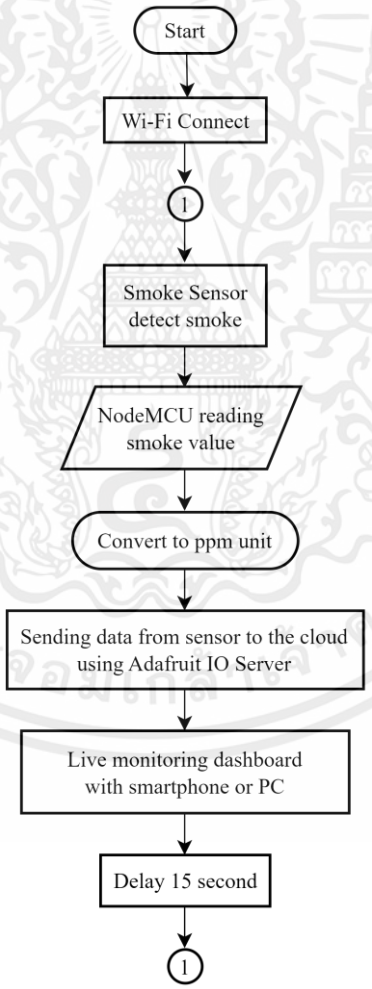
รูปที่ 3.3 การเชื่อมต่อวงจรรวมของระบบที่ใช้ในการทดสอบ

3.1.3 การออกแบบชุดคำสั่งการควบคุมการทำงานของระบบแจ้งเตือนอัคคีภัย

3.1.3.1 การออกแบบชุดคำสั่งควบคุมการทำงานของเซนเซอร์ตรวจจับควัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยเริ่มระบบบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 จะทำการเชื่อมต่อกับสัญญาณ Wi-Fi ในลำดับแรกเพื่อเข้าถึงเครือข่ายอินเทอร์เน็ต จากนั้นเซนเซอร์ตรวจจับควันจะทำการตรวจสอบควันไฟในพื้นที่โดยที่บอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 จะทำการเลือกอ่านค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อกจากเซนเซอร์ตรวจจับควันและทำการประมวลผลค่าโดยการแปลงเป็นค่าปริมาณควันที่อยู่ในอากาศในหน่วย ppm จากนั้นจะใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าสัญญาณแอนะล็อกที่เซนเซอร์ตรวจจับควันอ่านค่าได้ และค่าปริมาณควันในหน่วย ppm ไปแสดงผลบน Dashboard ที่สามารถดูบนคอมพิวเตอร์และสมาร์ทโฟนได้ โดยจะมีการดีเลย์ทุกๆ 15 วินาที แล้วกลับไปรับค่าใหม่ ดังในรูปที่ 3.4



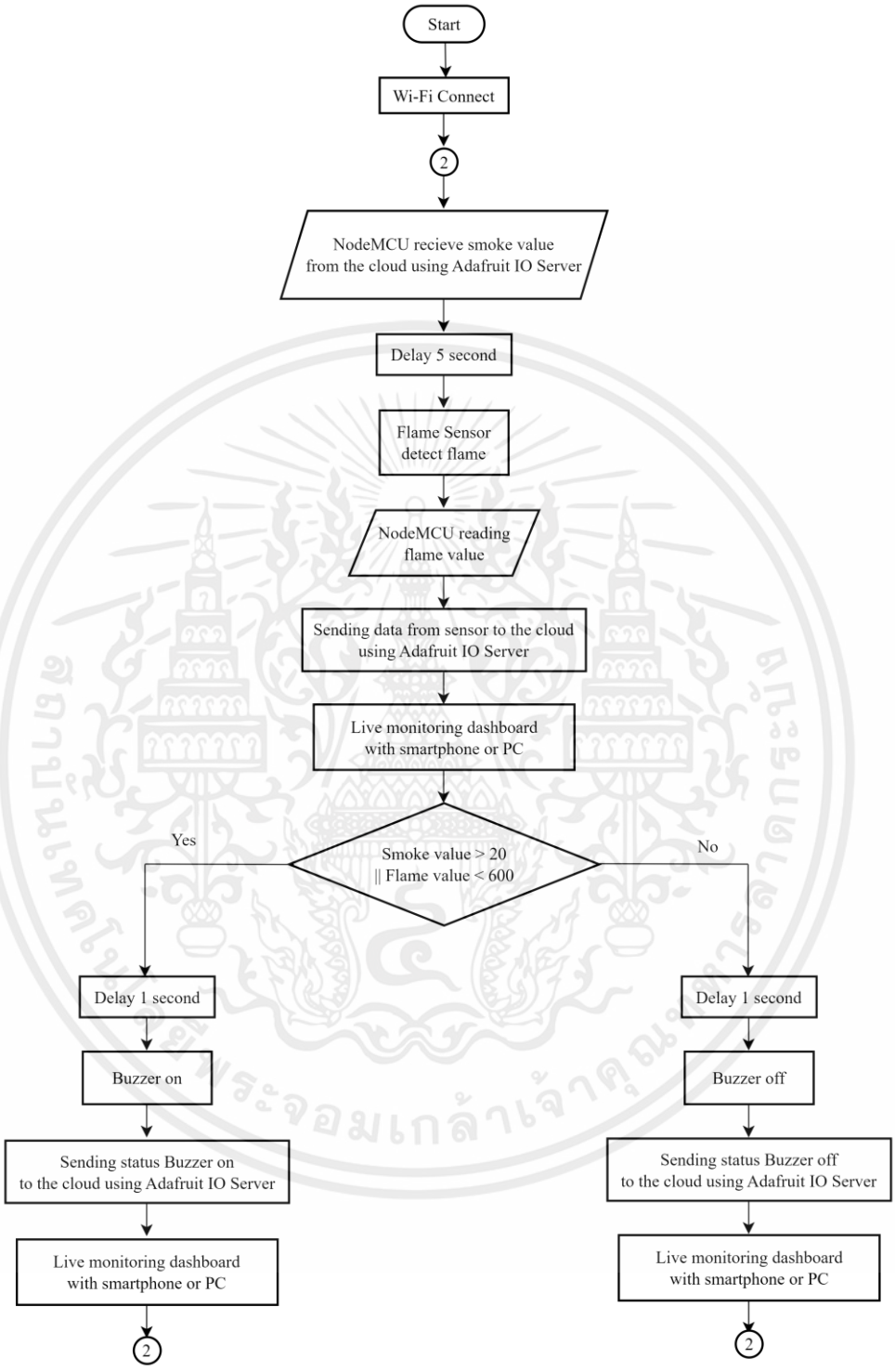
รูปที่ 3.4 แผนผังการทำงานของเซนเซอร์ตรวจจับควัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3.2 การออกแบบชุดคำสั่งควบคุมการทำงานของเซนเซอร์ตรวจจับเปลวไฟและการส่งสัญญาณเสียงเตือนไฟไหม้ของโมดูล Active Buzzer

โดยเริ่มระบบบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 จะทำการเชื่อมต่อกับสัญญาณ Wi-Fi ในลำดับแรกเพื่อเข้าถึงเครือข่ายอินเทอร์เน็ต จากนั้นจะทำการรับค่าของเซนเซอร์ตรวจจับควันที่ส่งมาจากการให้บริการคลาวด์ของ Adafruit IO ผ่านเครือข่ายอินเทอร์เน็ตโดยจะมีการดีเลย์ 5 วินาที จากนั้นเซนเซอร์ตรวจจับเปลวไฟจะทำการตรวจสอบเปลวไฟในพื้นที่ โดยที่บอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 จะทำการเลือกอ่านค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อกจากเซนเซอร์ตรวจเปลวไฟในพื้นที่ จากนั้นจะใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าสัญญาณแอนะล็อกที่เซนเซอร์ตรวจจับเปลวไฟอ่านค่าได้ไปแสดงผลบน Dashboard จากนั้นทำการตรวจสอบค่าเซนเซอร์ตรวจจับควันและค่าของเซนเซอร์ตรวจจับเปลวไฟที่รับได้ โดยแบ่งเป็น 2 เงื่อนไข เงื่อนไขแรกหากค่าเซองเซนเซอร์ตรวจจับควันมีค่ามากกว่า 20 หรือค่าของเซนเซอร์ตรวจจับเปลวไฟมีค่าน้อยกว่า 600 จะทำการสั่งงานไปยัง Active Buzzer เพื่อส่งสัญญาณเสียงเตือน จากนั้นจะใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าสถานะการเปิดการทำงานของโมดูล Active Buzzer ไปแสดงผลบน Dashboard แล้วกลับไปรับค่าใหม่ เงื่อนไขที่สองหากค่าของเซนเซอร์ตรวจจับควันมีค่าน้อยกว่า 20 และค่าของเซนเซอร์ตรวจจับเปลวไฟมีค่ามากกว่า 600 จะทำการสั่งงานไปยัง Active Buzzer เพื่อหยุดสร้างสัญญาณเสียงเตือน จากนั้นจะใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าสถานะการปิดการทำงานของโมดูล Active Buzzer ไปแสดงผลบน Dashboard ที่สามารถดูบนคอมพิวเตอร์และสมาร์ทโฟนได้ แล้วกลับไปรับค่าใหม่ ดังในรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



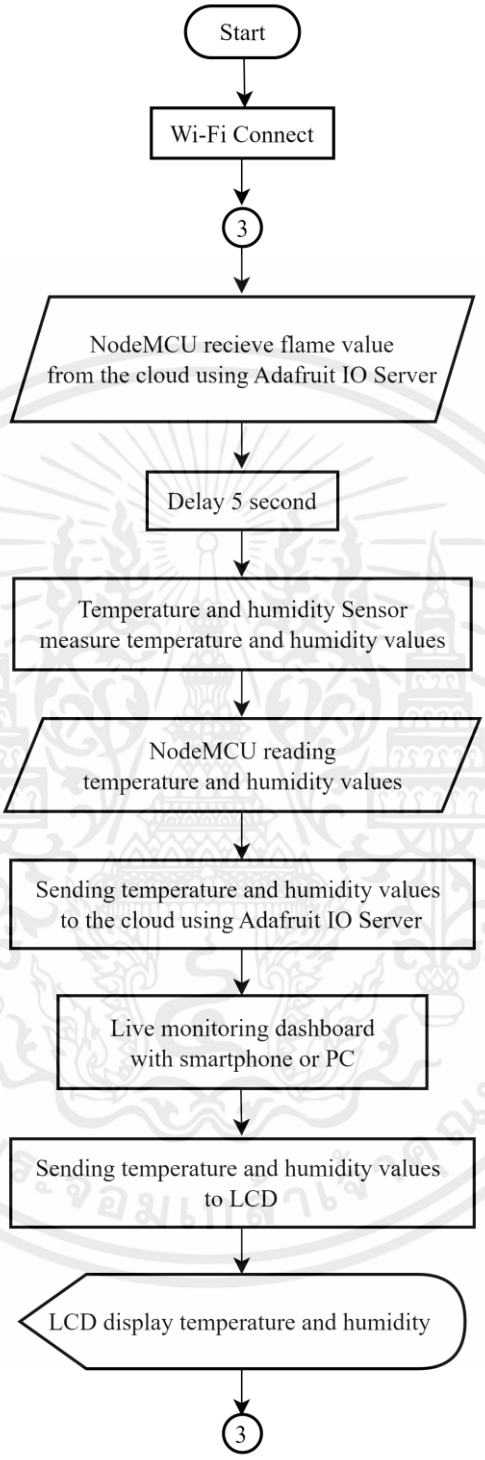
รูปที่ 3.5 แผนผังการทำงานของเซนเซอร์ตรวจจับเปลวไฟและการส่งสัญญาณเสียงเตือนไฟไหม้ของโมดูล Active Buzzer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3.3 การออกแบบชุดคำสั่งควบคุมการทำงานของเซนเซอร์วัดอุณหภูมิและความชื้น (DHT22)

โดยเริ่มระบบบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 จะทำการเชื่อมต่อกับสัญญาณ Wi-Fi ในลำดับแรกเพื่อเข้าถึงเครือข่ายอินเทอร์เน็ต จากนั้นจะทำการรับค่าของเซนเซอร์ตรวจจับเปลวไฟที่ส่งมาจากการให้บริการคลาวด์ของ Adafruit IO ผ่านเครือข่ายอินเทอร์เน็ตโดยจะมีการดีเลย์ 5 วินาที จากนั้นเซนเซอร์วัดอุณหภูมิและความชื้น จะทำการตรวจสอบค่าอุณหภูมิในพื้นที่ โดยที่บอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 จะทำการอ่านค่าความชื้นสัมพัทธ์และค่าอุณหภูมิโดยเฉลี่ยจากเซนเซอร์วัดอุณหภูมิและความชื้น จากนั้นจะใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าความชื้นสัมพัทธ์และค่าอุณหภูมิโดยเฉลี่ยในหน่วยองศาเซลเซียสที่เซนเซอร์อ่านค่าได้ไปแสดงผลบน Dashboard ที่สามารถดูบนคอมพิวเตอร์และสมาร์ทโฟนได้ หลังจากนั้นทำการส่งค่าความชื้นสัมพัทธ์และค่าอุณหภูมิโดยเฉลี่ยไปยังจอแสดงผล Liquid Crystal Display (LCD) เพื่อแสดงค่าความชื้นสัมพัทธ์และค่าอุณหภูมิโดยเฉลี่ยในหน่วยองศาเซลเซียส แล้วกลับไปรับค่าใหม่ ดังในรูปที่ 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



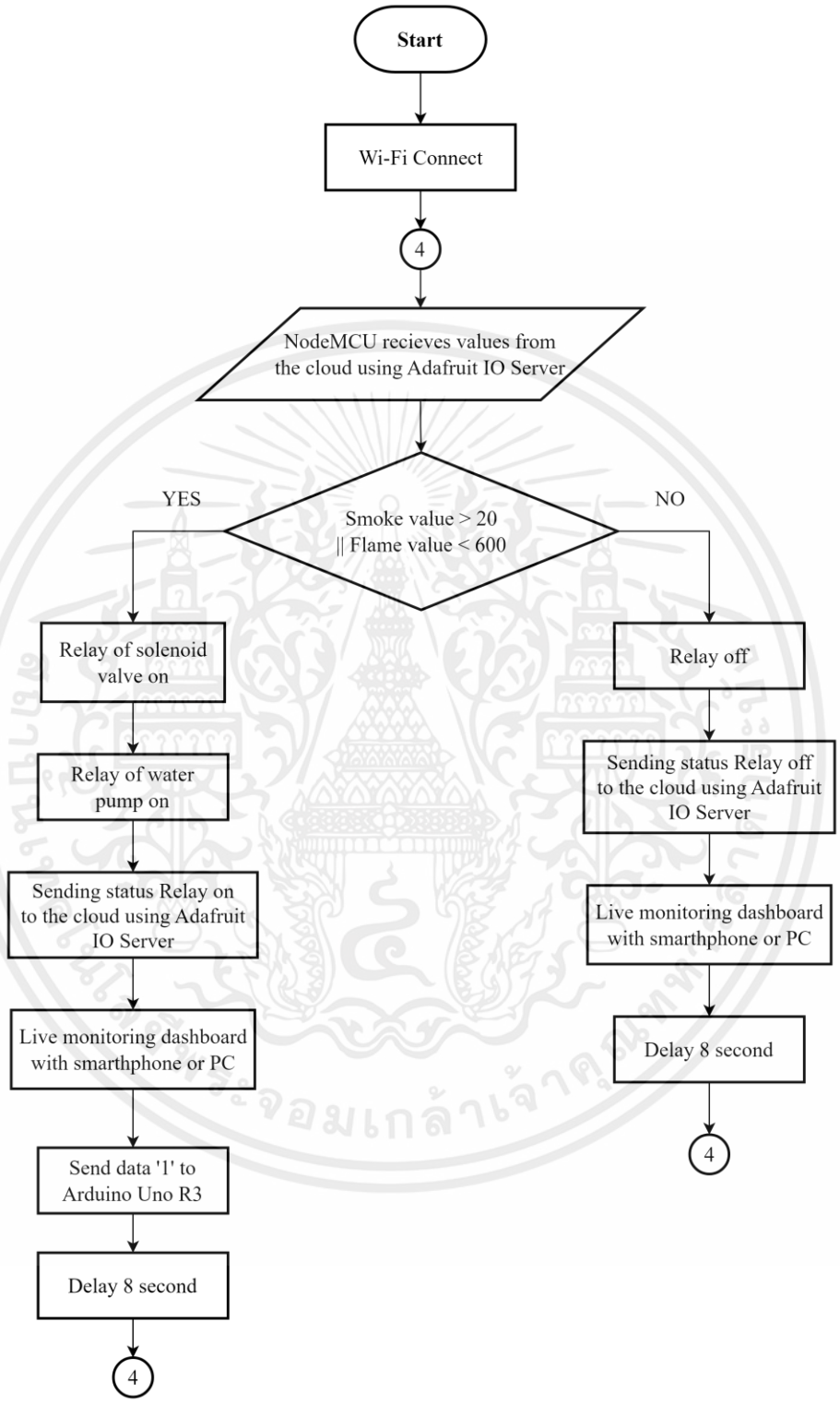
รูปที่ 3.6 แผนผังการทำงานของเซนเซอร์วัดอุณหภูมิและความชื้น (DHT22)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3.4 การออกแบบชุดคำสั่งควบคุมการทำงานของการดับไฟเบื้องต้น

โดยเริ่มระบบบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 จะทำการเชื่อมต่อกับสัญญาณ Wi-Fi ในลำดับแรกเพื่อเข้าถึงเครือข่ายอินเทอร์เน็ต จากนั้นทำการตรวจสอบค่าเซนเซอร์ตรวจจับควันและค่าของเซนเซอร์ตรวจจับเปลวไฟที่รับได้ โดยแบ่งเป็น 2 เงื่อนไข เงื่อนไขแรกหากค่าเของเซนเซอร์ตรวจจับควันมีค่ามากกว่า 20 หรือค่าของเซนเซอร์ตรวจจับเปลวไฟมีค่าน้อยกว่า 600 จะทำการสั่งงานไปยังรีเลย์เพื่อจ่ายไฟให้โซลินอยด์วาล์วและปั๊มน้ำทำงานตามลำดับเพื่อทำการดับไฟเบื้องต้น จากนั้นจะใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าสถานะการเปิดการทำงานของโมดูล Relay 2 Channel ไปแสดงผลบน Dashboard แล้วกลับไปรับค่าใหม่ เงื่อนไขที่สองหากค่าของเซนเซอร์ตรวจจับควันมีค่าน้อยกว่า 20 และค่าของเซนเซอร์ตรวจจับเปลวไฟมีค่ามากกว่า 600 จะไม่มีการดับไฟเบื้องต้น แต่จะมีการสั่งให้รีเลย์หยุดการจ่ายไฟให้โซลินอยด์วาล์วและปั๊มน้ำ จากนั้นจะใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าสถานะการปิดการทำงานของโมดูล Relay 2 Channel ไปแสดงผลบน Dashboard ที่สามารถดูบนคอมพิวเตอร์และสมาร์ทโฟนได้ จากนั้นจะมีการส่งค่า 1 ไปยังบอร์ด Arduino Uno R3 แบบ Serial แล้วกลับไปรับค่าใหม่ ดังในรูปที่ 3.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

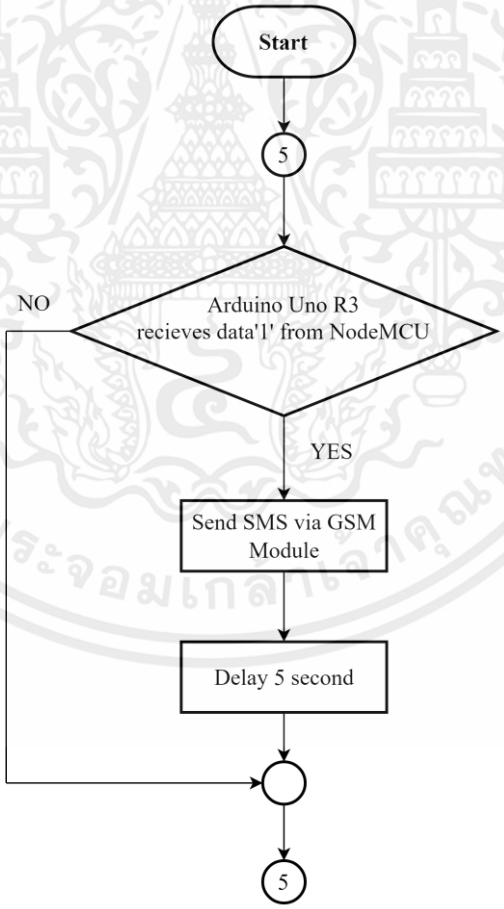


รูปที่ 3.7 แผนผังการทำงานของการดับไฟเบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3.4 การออกแบบชุดคำสั่งควบคุมการทำงานของการส่งข้อความ SMS แจ้งเตือนไฟไหม้

โดยเริ่มระบบบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 จะทำตรวจสอบการส่งค่ามาจากบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 แบบ Serial โดยแบ่งเป็น 2 เงื่อนไข เงื่อนไขแรกหากบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 ได้รับค่า 1 ที่มีการส่งมาจากบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 จะทำการสั่งงานให้ GSM Module ส่งข้อความ SMS แจ้งเตือนไฟไหม้ไปยังสมาร์ทโฟนของผู้ดูแลโดยจะมีการดีเลย์ 5 วินาที แล้วกลับไปรับค่าใหม่ เงื่อนไขที่สองหากบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 ไม่ได้รับค่าที่มีการส่งมาจากบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 จะไม่มีการสั่งงานแจ้งเตือนไฟไหม้เกิดขึ้น แล้วกลับไปรับค่าใหม่ ดังในรูปที่ 3.8

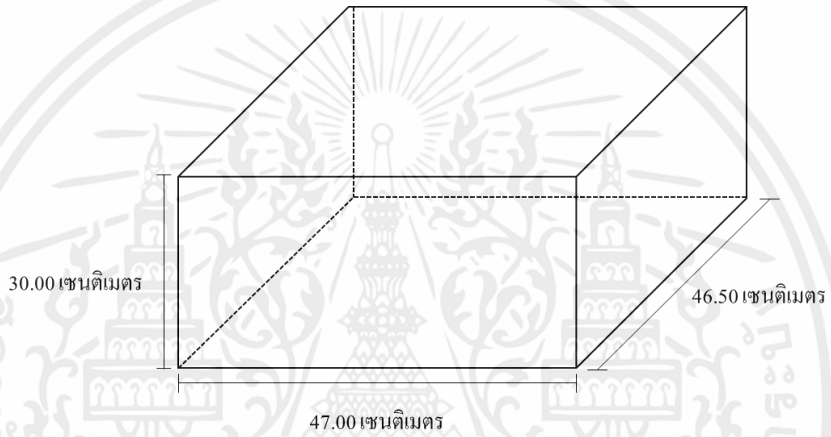


รูปที่ 3.8 แผนผังการทำงานของการส่งข้อความ SMS แจ้งเตือนไฟไหม้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 การออกแบบชุดจำลองห้องพัก

การออกแบบชุดจำลองห้องพักโดยถอดขนาดจากห้องพักขนาดเล็กขนาด 46.5 เซนติเมตร x 47.0 เซนติเมตร x 30.0 เซนติเมตร จำลองในอัตราส่วน 1:10 เซนติเมตรจากขนาดจริง คือมีขนาด 4.65 เมตร x 4.7 เมตร x 3.0 เมตร การออกแบบชุดจำลองห้องพัก ดังในรูปที่ 3.9 โดยใช้วัสดุเป็นแผ่นอะคริลิคใส ซึ่งเป็นเทอร์โมพลาสติกที่มีความทนทาน เพื่อรองรับน้ำหนักของโครงสร้างชุดจำลอง และอุปกรณ์ต่างๆ ของระบบ โดยชุดจำลองห้องพักที่จำลองขึ้น ดังในรูปที่ 3.10



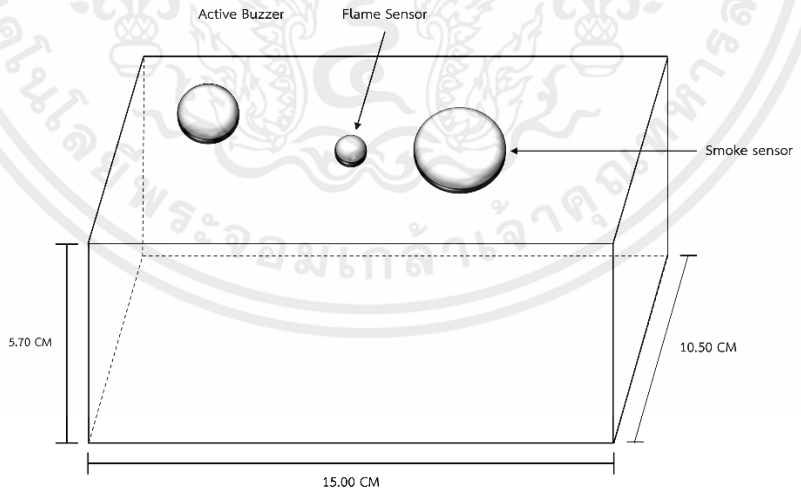
รูปที่ 3.9 ออกแบบชุดจำลองห้องพัก

รูปที่ 3.10 ชุดจำลองห้องพัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

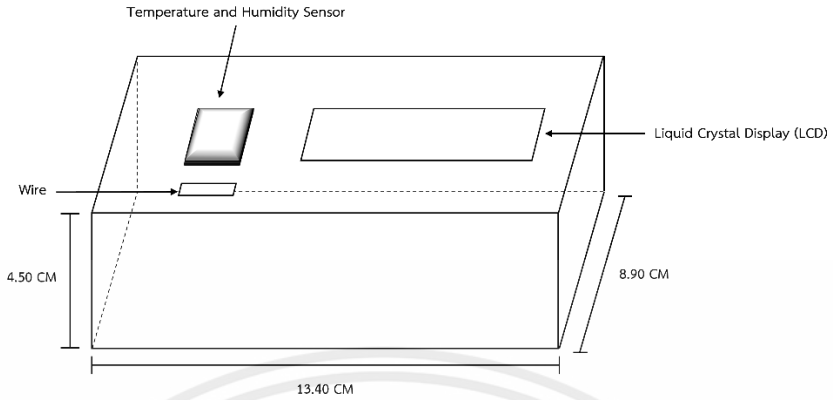
3.1.5 การออกแบบการติดตั้งอุปกรณ์ลงกล่องพลาสติก

กล่องใส่อุปกรณ์ต้นแบบจะออกแบบแบ่งเป็น 2 ขนาด คือ กล่องแรกมีขนาด 10.5 เซนติเมตร x 15.0 เซนติเมตร x 5.7 เซนติเมตร โดยอุปกรณ์ภายในกล่องจะประกอบไปด้วย เซนเซอร์ตรวจจับควันและเปลวไฟ โดยเซนเซอร์แต่ละตัวจะเชื่อมกับบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266, Lithium battery 18650 และ18650 Battery Shield V3 และในส่วนตัวกล่องที่ออกแบบทำการเจาะช่องด้านหน้าสำหรับจัดเก็บเซนเซอร์ตรวจจับควันและเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared เพื่อให้สามารถทำการตรวจสอบควันและเปลวไฟในพื้นที่ได้ รวมถึงเจาะช่องเพื่อติดตั้ง Active Buzzer เพื่อให้สัญญาณเสียงเตือนดังชัดเจน ดังรูปที่ 3.11 ในส่วนของกล่องขนาดที่สองคือ 8.9 เซนติเมตร x 13.4 เซนติเมตร x 4.5 เซนติเมตร ซึ่งออกแบบไว้จัดเก็บเซนเซอร์วัดความชื้นและอุณหภูมิ, จอแสดงผล Liquid Crystal Display (LCD), NodeMCU ESP8266, Lithium battery 18650 และ18650 Battery Shield V3 ในส่วนตัวกล่องออกแบบแล้วทำการเจาะช่องด้านหน้าสำหรับจัดเก็บเซนเซอร์วัดความชื้นและอุณหภูมิ เพื่อให้สามารถทำการตรวจสอบค่าความชื้นสัมพัทธ์และอุณหภูมิโดยเฉลี่ยในพื้นที่ได้ รวมถึงเจาะช่องข้างหน้าเพื่อให้มีช่องสำหรับจอแสดงผล Liquid Crystal Display (LCD) เพื่อแสดงค่าอุณหภูมิโดยเฉลี่ยในหน่วยองศาเซลเซียสและค่าความชื้น ดังรูปที่ 3.12 โดยทำการติดตั้งอุปกรณ์ลงกล่องพลาสติก ดังรูปที่ 3.13



รูปที่ 3.11 การออกแบบการติดตั้งอุปกรณ์ลงกล่องพลาสติกกล่องที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 การออกแบบการติดตั้งอุปกรณ์ลงกล่องพลาสติกกล่องที่ 2

รูปที่ 3.13 การติดตั้งอุปกรณ์ลงกล่องพลาสติก

3.2 เครื่องมือที่ใช้ในการทดลอง

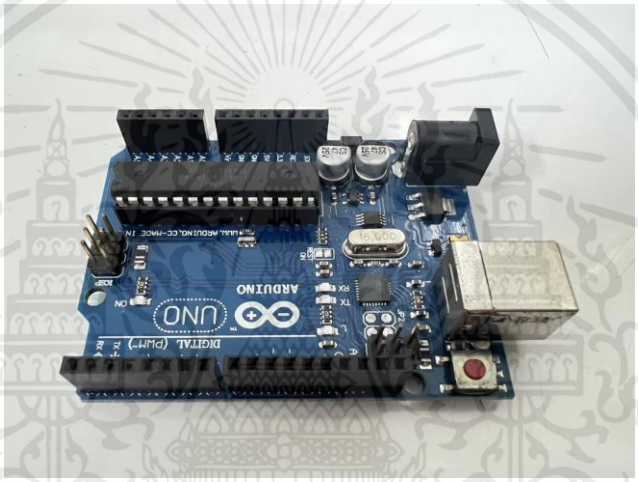
ในโครงงานนี้ มีอุปกรณ์และเครื่องมือที่ใช้ในการทดลอง ดังนี้

3.2.1 ไมโครคอนโทรลเลอร์ Arduino Uno R3 (ATmega328P)

เป็นบอร์ดไมโครคอนโทรลเลอร์ ATmega328P ขนาด 8 บิต เป็นบอร์ดที่ได้รับความนิยมอย่างมากเนื่องจากใช้งานง่าย มีขาสัญญาณแอนะล็อก 6 อินพุต ขาสัญญาณดิจิทัล 13 ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไลบรารีและอุปกรณ์เชื่อมต่อส่วนใหญ่สามารถใช้ร่วมกับ Arduino Uno ได้ โดยในโครงงานบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 ใช้ในการประมวลผลที่ทำงานร่วมกับเซนเซอร์ตรวจจับเปลวไฟและเซนเซอร์ตรวจจับควันและทำการตรวจจับเปลวไฟและควันและทำหน้าที่เป็นตัวรับและส่งข้อมูลไปยังบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 แบบ Serial และเมื่อมีค่าปริมาณที่ตรวจจับเกินจากที่กำหนดบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 จะมีการส่งข้อมูลเพื่อสั่งงาน GSM Module ให้ทำการแจ้งเตือนข้อความไปยังสมาร์ทโฟนของผู้ดูแลและมีการสั่งงานให้รีเลย์ทำงานเพื่อให้โซลินอยด์วาล์วและปั๊มน้ำทำงานทำให้สปริงเกอร์ทำการดับไฟเบื้องต้น ดังในรูปที่ 3.14

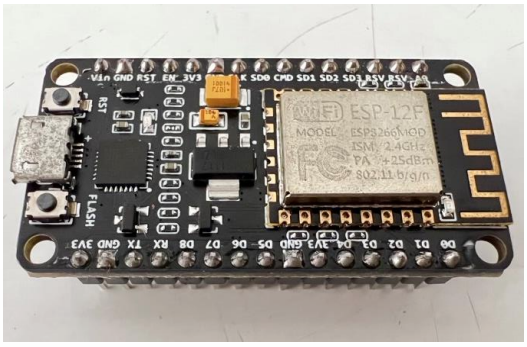


รูปที่ 3.14 ไมโครคอนโทรลเลอร์ Arduino Uno R3 (ATmega328P)

3.2.2 ไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2

เป็นบอร์ดไมโครคอนโทรลเลอร์ที่มีลักษณะการทำงานคล้าย Arduino แต่มีลักษณะพิเศษกว่าตรงที่สามารถเชื่อมต่อกับ Wi-Fi ได้ การควบคุมการทำงานสามารถใช้โปรแกรม Arduino IDE ได้ ถูกพัฒนาจาก NodeMCU ESP8266 V1 มีการปรับปรุงให้เล็กลงกว่าเดิมสามารถเสียบบอร์ดทดลองได้เหลือช่องให้เสียบสายไฟเพิ่มเติมได้ประสิทธิภาพดีขึ้น โดยในโครงงานบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 ใช้ในการรับค่าอุณหภูมิจากเซนเซอร์อุณหภูมิและส่งค่าอุณหภูมิเพื่อแสดงผลที่จอ LCD และเป็นตัวรับส่งข้อมูลเพื่อแสดงค่าสถานะเปลวไฟ ควัน รวมถึงอุณหภูมิผ่านแอพพลิเคชั่นบนระบบปฏิบัติการแอนดรอยด์ ดังในรูปที่ 3.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 ไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2

3.2.3 เซนเซอร์ตรวจจับควัน Gas Sensor (MQ-2)

ผู้จัดทำได้เลือกใช้โมดูล MQ-2 เซนเซอร์ตรวจจับควัน ซึ่งใช้เป็นอุปกรณ์ที่ตรวจสอบปริมาณก๊าซไวไฟ และควันในพื้นที่ที่ต้องการ ซึ่งเมื่อเราเริ่มจ่ายพลังงานให้ MQ-2 ที่ขา H ทำให้เกิดความร้อนขึ้นที่ขดลวด เมื่อก๊าซไวไฟต่างๆเข้ามาทำปฏิกิริยาจะทำให้ค่าความต้านทานที่เกิดขึ้นระหว่าง ขา A และ B (RS) ลดลง หรือค่าความต้านทานแปรผกผันกับปริมาณของก๊าซไวไฟต่างๆจึงทำให้สามารถตรวจสอบปริมาณก๊าซไวไฟและควันได้ ดังในรูปที่ 3.16

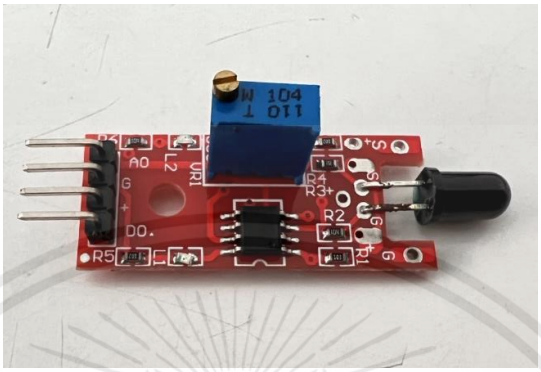
รูปที่ 3.16 เซนเซอร์ตรวจจับควัน Gas Sensor (MQ-2)

3.2.4 เซนเซอร์ตรวจจับเปลวไฟด้วย Infrared

ผู้จัดทำได้เลือกใช้โมดูล เซนเซอร์ตรวจจับเปลวไฟด้วย Infrared ซึ่งใช้เป็นอุปกรณ์ใช้ตรวจจับเปลวไฟ โดยใช้เซนเซอร์อินฟาเรด ให้สัญญาณเอาต์พุตออกมาทั้งแบบดิจิตอลเมื่อตรวจจับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

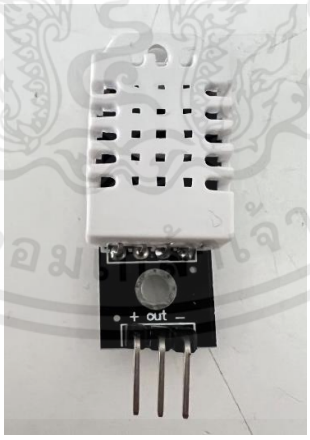
ได้ให้สัญญาณเป็น 1 เมื่อตรวจจับไม่พบให้สัญญาณเป็น 0 สามารถปรับความไวได้ที่บนโมดูล และแบบสัญญาณแอนะล็อก ดังในรูปที่ 3.17



รูปที่ 3.17 เซนเซอร์ตรวจจับเปลวไฟด้วย Infrared

3.2.5 เซนเซอร์วัดความชื้นและอุณหภูมิ DHT22

ผู้จัดทำได้เลือกใช้โมดูล DHT22 เซนเซอร์สำหรับวัดอุณหภูมิและความชื้น ซึ่งใช้เป็นอุปกรณ์ในการวัดอุณหภูมิ โมดูล DHT22 ให้ค่าเป็นแบบดิจิทัลใช้ขาสัญญาณดิจิทัลเพียงเส้นเดียวในการเชื่อมต่อแบบบิตอนุกรมสองทิศทาง (Serial Data, Bi-Directional) โดยนำมาเชื่อมต่อกับบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 เพื่ออ่านค่าจากเซนเซอร์ ดังในรูปที่ 3.18

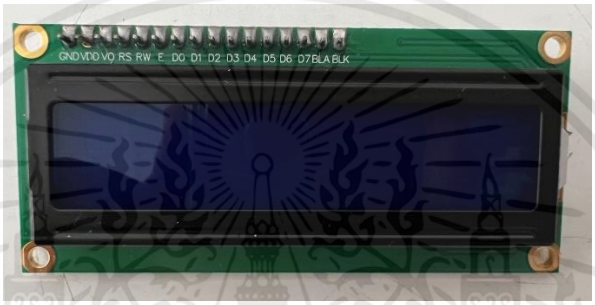


รูปที่ 3.18 เซนเซอร์วัดความชื้นและอุณหภูมิ DHT22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 จอแสดงผล Liquid Crystal Display (LCD)

ผู้จัดทำได้เลือกใช้โมดูล จอแสดงผล Liquid Crystal Display (LCD) เป็นจอแสดงผลรูปแบบหนึ่งมีการเชื่อมต่อแบบอนุกรม โดยการนำโมดูลเชื่อมเข้ากับตัวจอ LCD แล้วใช้บอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 เชื่อมต่อกับบอร์ดโมดูล ผ่านโปรโตคอล I2C ทำให้ใช้สายเพียง 2 เส้น ก็ทำให้หน้าจอแสดงผลข้อความต่าง ๆ ออกมาได้ โดยในโครงงานใช้ในการแสดงผลค่าอุณหภูมิที่จอ LCD ดังในรูปที่ 3.19



รูปที่ 3.19 จอแสดงผล Liquid Crystal Display (LCD)

3.2.7 SIM800L GSM/GPRS Module

ผู้จัดทำได้เลือกใช้โมดูล Sim800L เป็นรุ่นที่มีขนาดเล็กที่สุดจากตระกูล GSM Module ทั้งหมด สามารถเชื่อมต่อ Internet Gprs รองรับการส่ง SMS และสามารถโทรออกไปยังเบอร์ต่างๆ ซึ่งใช้เป็นอุปกรณ์ในการแจ้งเตือนข้อความไฟไหม้ไปยังสมาร์ทโฟนของผู้ดูแล ดังในรูปที่ 3.20

รูปที่ 3.20 SIM800L GSM/GPRS Module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.8 ปั๊ม DC 12V แรงดัน 8 บาร์

ผู้จัดทำได้เลือกใช้ ปั๊ม DC 12V แรงดัน 8 บาร์ 60W แบบไม่มี Pressure Switch ทำหน้าที่เพิ่มพลังงานให้แก่น้ำ เพื่อทำให้น้ำเคลื่อนที่จากตำแหน่งหนึ่ง ไปยังอีกตำแหน่งที่อยู่สูงกว่าหรือในระยะทางที่ไกลออกไป ซึ่งใช้เป็นอุปกรณ์ในปั๊มน้ำเพื่อทำการดับไฟเบื้องต้น ดังในรูปที่ 3.21

รูปที่ 3.21 ปั๊ม DC 12V แรงดัน 8 บาร์

3.2.9 มินิสปริงเกอร์ (Mini Sprinkler)

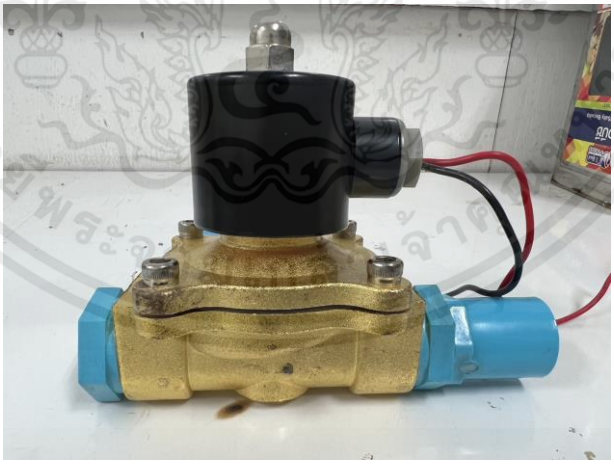
ผู้จัดทำได้เลือกใช้มินิสปริงเกอร์ เป็นสปริงเกอร์หัวจ่ายน้ำแบบ 360 องศา ลักษณะของน้ำที่ถูกปล่อยออกมาจากหัวจ่ายน้ำแบบนี้จะมีอัตราการจ่ายน้ำประมาณ 150-400 ลิตร/ชั่วโมง โดยขึ้นอยู่กับแรงดันน้ำ ซึ่งต้องการแรงดันน้ำในการใช้งาน 1.5-3 บาร์ขึ้นไปเพื่อทำให้น้ำที่ถูกพ่นออกมากระจายตัวได้เต็มที่ตามรัศมีที่มีความกว้าง 3-5 เมตร ซึ่งใช้เป็นอุปกรณ์เพื่อทำการดับไฟเบื้องต้น ดังในรูปที่ 3.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.22 มินิสปริงเกอร์ (Mini Sprinkler)

3.2.10 Solenoid Valve DC 12V

ผู้จัดทำได้เลือกใช้ Solenoid Valve DC 12V เป็นอุปกรณ์แม่เหล็กไฟฟ้าชนิดหนึ่ง นำมาใช้ในการเลื่อนลิ้นวาล์วของระบบนิวเมติกส์ การปิดเปิดการจ่ายน้ำหรือของเหลวอื่นๆ ซึ่งใช้เป็นอุปกรณ์ในการควบคุมการไหลของน้ำก่อนการไหลเข้าปั๊มน้ำ ดังในรูปที่ 3.23

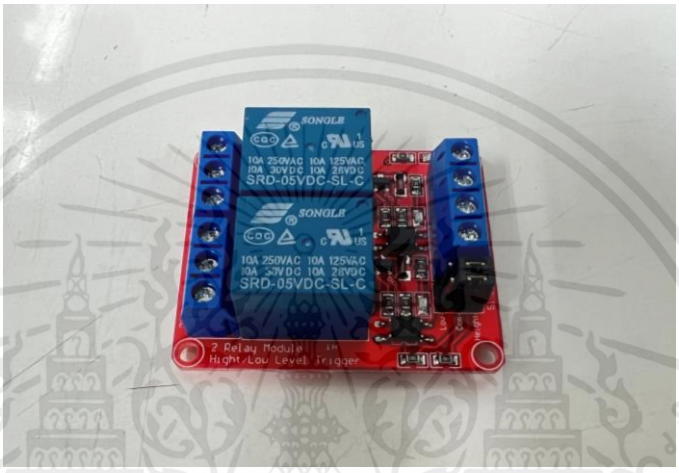


รูปที่ 3.23 Solenoid Valve DC 12V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.11 Arduino Relay Module แบบ 2 ช่อง

ผู้จัดทำได้เลือกใช้ Arduino Relay Module แบบ 2 ช่อง เป็นอุปกรณ์ที่ใช้สำหรับควบคุมการเปิด/ปิด Relay Module ได้ 2 ช่อง ใช้ไฟเข้า 5 โวลต์ ส่งสัญญาณควบคุมแบบ Active Low คือถ้าต้องการให้รีเลย์ติดจะส่งสัญญาณ 0 ไป และถ้าต้องการให้รีเลย์ดับจะส่งสัญญาณ 1 ซึ่งใช้เป็นอุปกรณ์สั่งงานให้โซลินอยด์วาล์วและปั๊มน้ำทำงาน ดังในรูปที่ 3.24



รูปที่ 3.24 Arduino Relay Module แบบ 2 ช่อง

3.2.12 Active Buzzer Module

ผู้จัดทำได้เลือกใช้โมดูล Active Buzzer Module 3.3 - 5V ดังในรูปที่ สำหรับใช้ในการสร้างเสียงแจ้งเตือนอย่างง่าย เพียงแค่จ่ายไฟเข้าไปที่ ขา I/O โมดูลนี้มีวงจรขยายช่วยขยายสัญญาณจึงมีความดังเป็นพิเศษ โดยในโครงงานใช้ในการสร้างสัญญาณเสียงเตือนไฟไหม้ ดังในรูปที่ 3.25

รูปที่ 3.25 Active Buzzer Module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

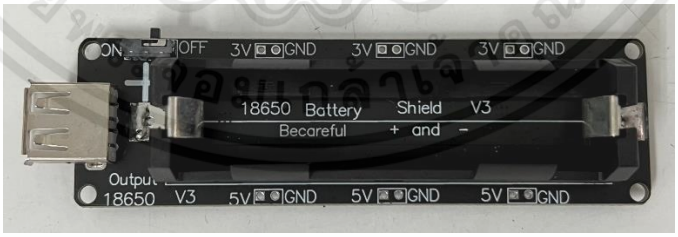
3.2.13 Lithium battery 18650

ผู้จัดทำได้เลือกใช้โมดูลชาร์จแบตเตอรี่ 18650 ที่มีความจุ 1200 mAh สามารถชาร์จไฟได้ถึง 500 ครั้ง โดยในโครงงานใช้ในใช้เป็นแหล่งจ่ายไฟให้กับบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 ดังในรูปที่ 3.26

รูปที่ 3.26 Lithium battery 18650

3.2.14 18650 Battery Shield V3

ผู้จัดทำได้เลือกใช้โมดูล Power Bank แปลงไฟจากถ่าน 18650 3.7V เป็นไฟ 5V 2A และ 3V 1A สำหรับจ่ายไฟให้กับบอร์ด Arduino , ESP8266 , Raspberry Pi โดบใส่แบตเตอรี่ 18650 ได้ 1 ก้อน เหมาะสำหรับอุปกรณ์กินกระแสน้อย ถึงปานกลาง โดยในโครงงานใช้ในการแปลงไฟจากถ่าน 18650 3.7V เป็นไฟ 5V 2A เพื่อจ่ายไฟให้กับบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 ดังในรูปที่ 3.27



รูปที่ 3.27 18650 Battery Shield V3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การจัดเก็บผลการทดลอง

3.3.1 การทดสอบชุดคำสั่งการทำงานของเซนเซอร์ตรวจจับควัน (MQ-2)

ทำการตรวจจับควันด้วยด้วยโมดูล Gas Sensor (MQ-2) โดยการตรวจสอบปริมาณก๊าซไวไฟ และควันในพื้นที่ โดยใช้โปรแกรม Arduino (IDE) ในการเขียนคำสั่งและระบุเงื่อนไขการทำงานของ Gas Sensor (MQ-2) ลงบนบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 เพื่อนำข้อมูลที่ได้มาวิเคราะห์และประมวลผล

3.3.1.1 การทดสอบการทำงานของเซนเซอร์ตรวจจับควัน (MQ-2)

ทำการอัปโหลดคำสั่งและระบุเงื่อนไขลงบนบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 โดยเซนเซอร์ตรวจจับควันจะมีการเลือกอ่านค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อก โดยมีการตรวจจับปริมาณแก๊สแอลพีจีและควันได้แบบต่อเนื่องโดยจะแสดงผลผ่าน Serial Monitor

3.3.1.2 การทดสอบความสามารถตรวจจับควันด้วยเซนเซอร์ตรวจจับควัน (MQ-2) ในชุดจำลองห้องพัก

ทำการทดสอบความสามารถตรวจจับควันด้วยเซนเซอร์ตรวจจับควัน (MQ-2) ในชุดจำลองห้องพัก โดยการเขียนคำสั่งและระบุเงื่อนไขการทำงานเพื่อควบคุมการทำงานของบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 โดยทำการบันทึกค่าปริมาณควันที่อยู่ในอากาศในหน่วย ppm ทุก ๆ 1 นาที ของการเผาวัสดุ 3 ชนิด คือ กาบมะพร้าว, เศษผ้า และธูป ในปริมาณที่เท่ากัน ซึ่งระยะห่างในการทดสอบระหว่างเซนเซอร์ตรวจจับและวัสดุมีระยะ 30 เซนติเมตร โดยติดเซนเซอร์ไว้ที่กึ่งกลางภายในด้านบนของชุดจำลองห้องพัก นำผลมาวิเคราะห์ความสามารถในการตรวจจับควันของเซนเซอร์ตรวจจับควัน (MQ-2) ของการเผาไหม้วัสดุทั้ง 3 ชนิด และทำการวิเคราะห์ความสามารถตรวจจับค่าควันของวัสดุ 3 ชนิด ที่มีผลทำให้เกิดอัคคีภัย

3.3.1.3 การทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าเซนเซอร์ตรวจจับควัน (MQ-2) ในหน่วย PPM แสดงผลบน Dashboard

ทำการอัปโหลดคำสั่งและระบุเงื่อนไขการทำงานของเซนเซอร์ตรวจจับควัน (MQ-2) และการส่งค่าเซนเซอร์ไปแสดงผลบน Dashboard ลงบนบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 โดยทำการส่งค่าสัญญาณแอนะล็อกที่เซนเซอร์ตรวจจับควันที่อ่านค่าได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และค่าปริมาณควันในหน่วย ppm ไปแสดงผลบน Dashboard ผ่านบริการคลาวด์ของ Adafruit IO

3.3.2 การทดสอบชุดคำสั่งการทำงานของเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared

ทำการตรวจจับเปลวไฟด้วยด้วยโมดูลเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared โดยใช้เซนเซอร์อินฟาเรดตรวจจับเปลวไฟในพื้นที่ โดยใช้โปรแกรม Arduino (IDE) ในการเขียนคำสั่งและระบุเงื่อนไขการทำงานของ เซนเซอร์ตรวจจับเปลวไฟด้วย Infrared ลงบนบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 เพื่อนำข้อมูลที่ได้มาวิเคราะห์และประมวลผล

3.3.2.1 การทดสอบการทำงานของเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared

ทำการอัปโหลดคำสั่งและระบุเงื่อนไขลงบนบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 เพื่อรับค่าสถานะจากตรวจจับเปลวไฟในพื้นที่โดยจะแสดงผลผ่าน Serial Monitor

3.3.2.2 การทดสอบความสามารถเซนเซอร์ตรวจจับเปลวไฟในชุดจำลองห้องพัก

ทำการทดสอบความสามารถตรวจจับเปลวไฟด้วยเซนเซอร์ตรวจจับเปลวไฟในชุดจำลองห้องพัก โดยการเขียนคำสั่งและระบุเงื่อนไขการทำงานเพื่อควบคุมการทำงานของบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 โดยทำการบันทึกค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อกและสถานะการตรวจจับเปลวไฟในพื้นที่จำลอง ทั้งหมด 25 จุด ที่มีระยะห่างจากเซนเซอร์ที่ความสูง 30 เซนติเมตร และนำมาวิเคราะห์หาความสามารถที่เซนเซอร์ตรวจจับได้ในชุดจำลองห้องพัก

3.3.2.3 การทดสอบหาระยะที่เซนเซอร์สามารถตรวจจับเปลวไฟด้วย Infrared

ทำการทดสอบหาระยะที่เซนเซอร์สามารถตรวจจับเปลวไฟได้ด้วยเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared โดยเขียนคำสั่งและระบุเงื่อนไขการทำงานของเซนเซอร์ลงบนบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 โดยทำการบันทึกค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อกที่สภาวะปกติและบันทึกค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อกเมื่อทำการจุดไฟโดยเริ่มต้นระยะ 0 เมตรจนถึงระยะที่ค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อกสามารถตรวจจับเปลวไฟได้ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สภาวะปกติและนำมาวิเคราะห์หาเปอร์เซ็นต์ความแตกต่างระหว่างสภาวะปกติและขณะที่ทำการจุดไฟในระยะทางต่างๆและระยะที่เซนเซอร์ตรวจจับได้

3.3.2.4 การทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าเซนเซอร์ตรวจจับเปลวไฟแสดงผลบน Dashboard

ทำการอัปโหลดคำสั่งและระบุเงื่อนไขการทำงานของเซนเซอร์ตรวจจับเปลวไฟ และการส่งค่าเซนเซอร์ไปแสดงผลบน Dashboard ที่มีการประมวลผลด้วยบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 โดยทำการส่งค่าเอาต์พุตสัญญาณแอนะล็อกที่เซนเซอร์ตรวจจับเปลวไฟอ่านค่าได้ไปแสดงผลบน Dashboard ผ่านบริการคลาวด์ของ Adafruit IO

3.3.3 การทดสอบชุดคำสั่งการทำงานของโมดูล Active Buzzer

ทำการทดสอบโดยการเขียนคำสั่งและระบุเงื่อนไขการทำงานเพื่อควบคุมการทำงานของโมดูล Active Buzzer โดยรับค่ามาจากค่าเซนเซอร์ตรวจจับควัน (MQ-2) และเซนเซอร์ตรวจจับเปลวไฟ ที่มีการประมวลผลด้วยบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2

3.3.3.1 การทดสอบการทำงานของโมดูล Active Buzzer

ทำการอัปโหลดคำสั่งและระบุเงื่อนไขลงบนบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 เพื่อควบคุมการทำงานของโมดูล Active Buzzer ในการสร้างสัญญาณเสียงแจ้งเตือน โดยจะแสดงผลผ่าน Serial Monitor

3.3.3.2 การทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการแสดงสถานะการทำงานของโมดูล Active Buzzer ในการส่งสัญญาณเสียงเตือนแสดงผลบน Dashboard

ทำการอัปโหลดคำสั่งและระบุเงื่อนไขการทำงานของโมดูล Active Buzzer ในการประมวลผลค่าเซนเซอร์ตรวจจับควัน (MQ-2) และเซนเซอร์ตรวจจับเปลวไฟ และการส่งสถานะการทำงานของโมดูล Active Buzzer ไปแสดงผลบน Dashboard ที่มีการประมวลผลด้วยบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 โดยทำการประมวลผลค่าเซนเซอร์ตรวจจับควัน (MQ-2) และเซนเซอร์ตรวจจับเปลวไฟ เพื่อควบคุมการแจ้งเตือนเสียงเมื่อตรวจพบควันไฟหรือเปลวไฟ และส่งค่าสถานะการทำงานของโมดูล Active Buzzer ไปแสดงผลบน Dashboard ผ่านบริการคลาวด์ของ Adafruit IO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.4 การทดสอบการควบคุมการทำงานของเซนเซอร์วัดอุณหภูมิและความชื้น (DHT22)

ทำการทดสอบโดยการเขียนคำสั่งและระบุเงื่อนไขการทำงานเพื่อควบคุมการทำงานของโมดูล DHT22 โดยรับค่ามาจากเซนเซอร์วัดอุณหภูมิและความชื้นที่มีการประมวลผลด้วยบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 โดยทำการแสดงค่าอุณหภูมิโดยเฉลี่ยในหน่วยองศาเซลเซียส และองศาฟาเรนไฮต์และค่าความชื้นสัมพัทธ์โดยจะแสดงผลผ่าน Serial Monitor

3.3.5 การทดสอบการควบคุมการแสดงค่าอุณหภูมิและความชื้นบนหน้าจอ LCD

ทำการทดสอบโดยการเขียนคำสั่งและระบุเงื่อนไขการทำงานเพื่อควบคุมการทำงานของโมดูล DHT22 และจอแสดงผล Liquid Crystal Display (LCD) โดยรับค่ามาจากเซนเซอร์วัดอุณหภูมิและความชื้นที่มีการประมวลผลด้วยบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 โดยหน้าจอ Liquid Crystal Display (LCD) จะแสดงค่าอุณหภูมิในหน่วยองศาเซลเซียส และค่าความชื้นที่อ่านได้จากเซนเซอร์วัดอุณหภูมิและความชื้น (DHT22)

3.3.6 การทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าความชื้นสัมพัทธ์และค่าอุณหภูมิโดยเฉลี่ยในหน่วยองศาเซลเซียสแสดงผลบน Dashboard

ทำการทดสอบโดยการเขียนคำสั่งและระบุเงื่อนไขการทำงานเพื่อควบคุมการทำงานของโมดูล DHT22 และจอแสดงผล Liquid Crystal Display (LCD) และการส่งค่าเซนเซอร์ไปแสดงผลบน Dashboard ที่มีการประมวลผลด้วยบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 โดยทำการส่งค่าความชื้นสัมพัทธ์และค่าอุณหภูมิโดยเฉลี่ยในหน่วยองศาเซลเซียสไปแสดงผลบน Dashboard ผ่านบริการคลาวด์ของ Adafruit IO และหน้าจอ Liquid Crystal Display (LCD) จะแสดงค่าอุณหภูมิในหน่วยองศาเซลเซียส และค่าความชื้นที่อ่านได้จากเซนเซอร์วัดอุณหภูมิและความชื้น (DHT22)

3.3.7 การทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าเซนเซอร์ตรวจจับควัน (MQ-2) เซนเซอร์ตรวจจับเปลวไฟ และความชื้นสัมพัทธ์และอุณหภูมิโดยเฉลี่ยแสดงผลพร้อมกันบนหน้า Dashboard เดียวกัน

ทำการทดสอบโดยการเขียนคำสั่งและระบุเงื่อนไขการทำงานของเซนเซอร์ตรวจจับควัน (MQ-2) เซนเซอร์ตรวจจับเปลวไฟ เซนเซอร์วัดอุณหภูมิและความชื้น (DHT22) และจอแสดงผล Liquid Crystal Display (LCD) และการส่งค่าเซนเซอร์ไปแสดงผลบน Dashboard ที่มีการประมวลผลด้วยบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 โดยทำการส่งค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซนเซอร์ตรวจจับควัน (MQ-2) เซนเซอร์ตรวจจับเปลวไฟ และความชื้นสัมพัทธ์และค่าอุณหภูมิโดยเฉลี่ยในหน่วยองศาที่ไปแสดงผลพร้อมกันบนหน้า Dashboard เดียวกัน ผ่านบริการคลาวด์ของ Adafruit IO และหน้าจอ Liquid Crystal Display (LCD) จะแสดงค่าอุณหภูมิในหน่วยองศาเซลเซียส และค่าความชื้นที่อ่านได้จากเซนเซอร์วัดอุณหภูมิและความชื้น (DHT22)

3.3.8 การทดสอบชุดคำสั่งการทำงานของ SIM800L GSM Module

ทำการทดสอบโดยการเขียนคำสั่งและระบุเงื่อนไขการทำงานเพื่อควบคุมการทำงานของ SIM800L GSM Module โดยรับค่ามาจากเซนเซอร์วัดอุณหภูมิและความชื้นที่มีการประมวลผลด้วยบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 โดยทำการส่งข้อความ SMS แจ้งเตือนไปยังสมาร์ทโฟนของผู้ดูแล

3.3.8.1 การทดสอบการใช้งาน SIM800L GSM Module สำหรับใช้ในการส่ง SMS

ทำการอัปโหลดคำสั่งและระบุเงื่อนไขลงบนบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 เพื่อทำการส่งข้อความ SMS แจ้งเตือนไปยังสมาร์ทโฟนของผู้ดูแล

3.3.8.2 การทดสอบผลการทำงานร่วมกันของเซนเซอร์ตรวจจับควันและเซนเซอร์ตรวจจับเปลวไฟกับ SIM800L GSM Module ในการส่งข้อความ SMS แจ้งเตือนไฟไหม้

ทำการการทดสอบผลการทำงานร่วมกันของเซนเซอร์ตรวจจับควันและเซนเซอร์ตรวจจับเปลวไฟกับ SIM800L GSM Module ในการส่งข้อความ SMS แจ้งเตือนไฟไหม้ โดยเขียนคำสั่งและระบุเงื่อนไขการทำงานของเซนเซอร์ลงบนบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 โดยมีการตรวจจับปริมาณควันและเปลวไฟได้แบบต่อเนื่อง และเมื่อมีค่าปริมาณที่ตรวจจับเกินจากที่กำหนด SIM800L GSM Module จะส่งข้อความ SMS แจ้งเตือนไฟไหม้ไปยังสมาร์ทโฟนของผู้ดูแล

3.3.8.3 การทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการทำงานร่วมกันของเซนเซอร์ตรวจจับควัน (MQ-2) เซนเซอร์ตรวจจับเปลวไฟด้วย Infrared และเซนเซอร์วัดอุณหภูมิและความชื้น (DHT22) กับ SIM800L GSM Module ในการส่งข้อความ SMS แจ้งเตือนไฟไหม้

ทำการทดสอบโดยการเขียนคำสั่งและระบุเงื่อนไขการทำงานร่วมกันของเซนเซอร์ตรวจจับควัน (MQ-2) และเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared และเซนเซอร์วัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุณหภูมิและความชื้น (DHT22) และจอแสดงผล Liquid Crystal Display (LCD) และการส่งค่าเซนเซอร์ไปแสดงผลบน Dashboard ลงบนบอร์ด NodeMCU ESP8266 V2 และเขียนโปรแกรมคำสั่งและระบุเงื่อนไขการทำงานของ SIM800L GSM Module ลงบนบอร์ Arduino Uno R3 โดยเซนเซอร์ตรวจจับควันจะทำการอ่านค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อก และเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared จะทำการอ่านค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อก และเซนเซอร์วัดอุณหภูมิและความชื้น (DHT22) จะทำการอ่านค่าค่าความชื้นสัมพัทธ์และค่าอุณหภูมิโดยเฉลี่ยในหน่วยองศาเซลเซียส ไปแสดงผลพร้อมกันบนหน้า Dashboard เดียวกัน ผ่านบริการคลาวด์ของ Adafruit IO โดยจะมีการตรวจจับปริมาณควันและเปลวไฟได้แบบต่อเนื่อง และเมื่อมีค่าปริมาณที่ตรวจจับเกินจากที่กำหนดจะส่งข้อความ SMS แจ้งเตือนไฟไหม้ไปยังสมาร์ทโฟนของผู้ดูแล

3.3.9 การทดสอบชุดคำสั่งการทำงานของโมดูล Relay 2 Channel

ทำการตรวจจับเปลวไฟด้วยด้วยโมดูลเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared โดยใช้เซนเซอร์อินฟาเรดตรวจจับเปลวไฟในพื้นที่ โดยใช้โปรแกรม Arduino (IDE) ในการเขียนคำสั่งและระบุเงื่อนไขการทำงานของเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared ลงบนบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 เพื่อนำข้อมูลที่ได้มาวิเคราะห์และประมวลผล

3.3.9.1 การทดสอบการใช้งานโมดูล Relay 2 Channel

ทำการอัปโหลดคำสั่งและระบุเงื่อนไขลงบนบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 เพื่อควบคุมการทำงานของโซลินอยด์วาล์ว (Solenoid Valve) และปั๊ม DC 12V

3.3.9.2 การทดสอบการทำงานร่วมกันของเซนเซอร์ตรวจจับควันและเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared และการสั่งงานให้มีการดับไฟเบื้องต้น

ทำการทดสอบการทำงานร่วมกันของเซนเซอร์ตรวจจับควันและเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared และการสั่งงานให้มีการดับไฟเบื้องต้น โดยเขียนคำสั่งและระบุเงื่อนไขการทำงานของเซนเซอร์ลงบนบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 โดยมีการตรวจจับปริมาณควันและเปลวไฟได้แบบต่อเนื่อง และเมื่อมีค่าปริมาณที่ตรวจจับเกินจากที่กำหนดจะมีการควบคุมการทำงานแรงดันไฟฟ้า DC 12V ผ่านโมดูล Relay ที่สามารถสั่งให้โซลินอยด์วาล์ว (Solenoid Valve) และปั๊ม DC 12V เปิดและปิดอุปกรณ์ได้

3.3.10 การทดสอบการส่งค่าของเซนเซอร์จากบอร์ด Arduino Uno R3 ไปยังบอร์ด NodeMCU ESP8266 V2 แบบ Serial

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการทดสอบการส่งค่าของเซนเซอร์จากบอร์ด Arduino Uno R3 ไปยังบอร์ด NodeMCU ESP8266 V2 แบบ Serial โดยเขียนคำสั่งและระบุเงื่อนไขการทำงานของเซนเซอร์ลงบนบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 และบอร์ด NodeMCU ESP8266 V2 โดยจะทำการแสดงค่าเซนเซอร์ตรวจจับควันและค่าเซนเซอร์ตรวจจับเปลวไฟที่ใช้บอร์ด Arduino Uno R3 เป็นตัวส่งค่าและแสดงค่าเซนเซอร์ตรวจจับควันและค่าเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared ที่ใช้ NodeMCU ESP8266 V2 เป็นตัวรับค่าที่ส่งมาโดยจะแสดงผลผ่าน Serial Monitor

3.3.11 การทดสอบประสิทธิภาพของโมดูลชาร์จแบตเตอรี่ 18650

ทำการทดสอบประสิทธิภาพของโมดูลชาร์จแบตเตอรี่ 18650 จำนวน 3 ก้อน โดยทำการชาร์จแบตเตอรี่ทั้ง 3 ก้อนให้เต็ม ทำการบันทึกค่าความจุแบตเตอรี่ของทั้ง 3 ก้อน มีหน่วยเป็น mAh คือ มิลลิแอมป์ชั่วโมง ทำการวัดทุก 30 นาที จนกระทั่งแบตเตอรี่หมด

3.3.12 การทดสอบระบบแจ้งเตือนอัคคีภัยในการสั่งงานให้มีการดับไฟเบื้องต้นและส่งข้อความ SMS แจ้งเตือนไฟไหม้

ทำการทดสอบระบบแจ้งเตือนอัคคีภัยในการสั่งงานให้มีการดับไฟเบื้องต้นและส่งข้อความ SMS แจ้งเตือนไฟไหม้ โดยเขียนคำสั่งและระบุเงื่อนไขการทำงานของเซนเซอร์ลงบนบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 และบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 โดยจะทำการจำลองสถานการณ์เพื่อเป็นการทดสอบระบบ โดยจะมีการจุดไฟทำการบันทึกผลการสั่งงานให้มีการดับไฟเบื้องต้นและส่งข้อความ SMS แจ้งเตือนไฟไหม้

3.3.13 การทดสอบระบบแจ้งเตือนอัคคีภัยที่มีการใช้บริการคลาวด์ของ Adafruit IO ในการสั่งงานให้มีการส่งสัญญาณเสียงเตือนและส่งข้อความ SMS แจ้งเตือนไฟไหม้พร้อมกับมีการดับไฟเบื้องต้น

ทำการทดสอบระบบแจ้งเตือนอัคคีภัยที่มีการใช้บริการคลาวด์ของ Adafruit IO ในการสั่งงานให้มีการส่งสัญญาณเสียงเตือนและส่งข้อความ SMS แจ้งเตือนไฟไหม้พร้อมกับมีการดับไฟเบื้องต้น โดยเขียนคำสั่งและระบุเงื่อนไขการทำงานของเซนเซอร์ลงบนบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 และบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 โดยจะทำการจำลองสถานการณ์เพื่อเป็นการทดสอบระบบ โดยจะมีการจุดไฟทำการบันทึกผลการสั่งงานให้สร้างสัญญาณเสียงเตือนและส่งข้อความ SMS แจ้งเตือนไฟไหม้ไปยังสมาร์ทโฟนของผู้ดูแลพร้อมกับมีการดับไฟเบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

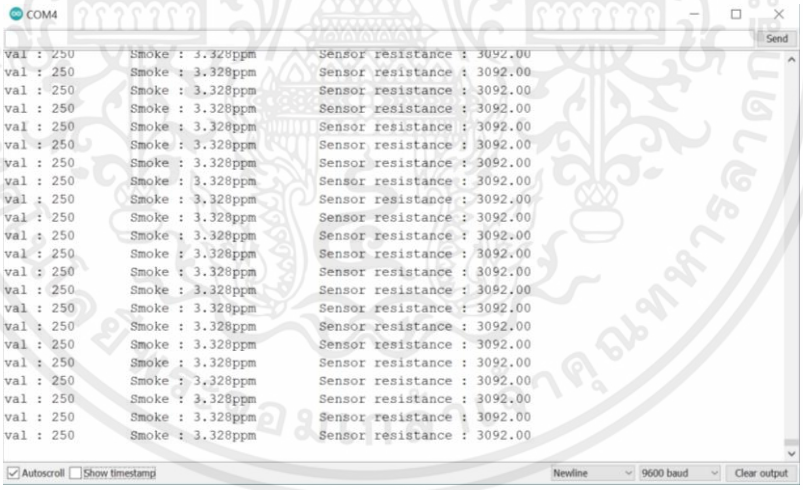
บทที่ 4
ผลการทดลอง

ผู้จัดทำได้ทำการเก็บผลการทำงานของระบบ โดยแบ่งการทดลองและจัดเก็บผลการทดลองเป็นส่วนๆ ดังต่อไปนี้

4.1 ผลการทดสอบชุดคำสั่งการทำงานของเซนเซอร์ตรวจจับควัน (MQ-2)

4.1.1 การทดสอบการทำงานของเซนเซอร์ตรวจจับควัน (MQ-2)

ในการทดสอบการทำงานของเซนเซอร์ตรวจจับควันโดยใช้โปรแกรม Arduino (IDE) ในการเขียนคำสั่งและระบุเงื่อนไขการทำงานของเซนเซอร์ตรวจจับควัน (MQ-2) โดยจากผลการทดลองหน้าต่าง Serial monitor จะทำการแสดงค่า 3 ค่าได้แก่ 1) ค่าสัญญาณแอนะล็อกที่เซนเซอร์ตรวจจับควันอ่านค่าได้ อยู่ในช่วงระหว่าง 0-1023 2) ค่าปริมาณควันที่อยู่ในอากาศในหน่วย ppm 3) ค่าความต้านทานภายในของเซนเซอร์เมื่อดูดซับควันที่อยู่ในอากาศ โดยค่าความต้านทานเซนเซอร์ที่แสดงจะแปรผกผันกับปริมาณควันที่อยู่ในอากาศ ดังในรูปที่ 4.1

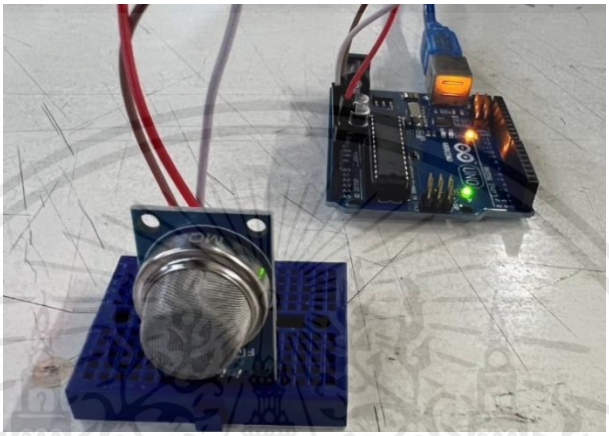


รูปที่ 4.1 หน้าต่าง Serial Monitor ที่แสดงค่า 3 ค่า ได้แก่ 1) ค่าที่เซนเซอร์ตรวจจับควันอ่านค่าได้ 2) ค่าปริมาณควัน หน่วย ppm 3) ค่าความต้านทานภายในของเซนเซอร์

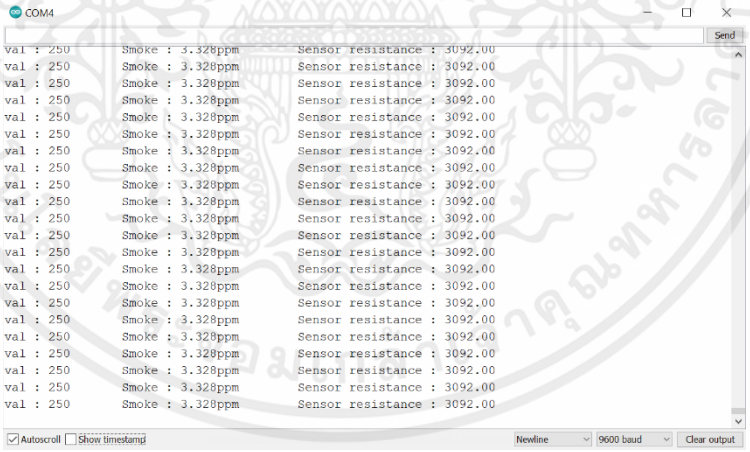
ผลการทดสอบการทำงานของเซนเซอร์ตรวจจับควัน เมื่อทำการทดสอบเซนเซอร์ขณะที่ยังไม่มีควัน ดังในรูปที่ 4.2 จะเห็นว่าค่าสัญญาณแอนะล็อกจากเซนเซอร์ตรวจจับควันจะมีค่าคงที่ 250 และค่าปริมาณควันที่อยู่ในอากาศมีค่า 3.328 ppm ดังในรูปที่ 4.3 และเมื่อทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซนเซอร์ตรวจจับควันขณะที่มีควัน โดยทำการทดสอบเบื้องต้นจากควันธูปจำนวน 7 ดอก ดังในรูปที่ 4.4 จะเห็นว่าค่าสัญญาณแอนะล็อกที่วัดได้จากเซนเซอร์ตรวจจับควันมีการเปลี่ยนแปลง โดยมีค่าเพิ่มขึ้นอยู่ในช่วงระหว่าง 300-350 และค่าปริมาณควันที่อยู่ในอากาศมีค่าเพิ่มขึ้นอยู่ในช่วงระหว่าง 6-9 ppm ดังในรูปที่ 4.5

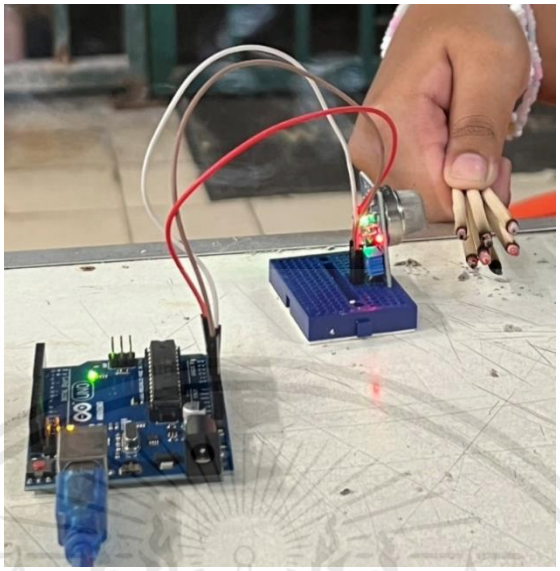


รูปที่ 4.2 ทำการทดสอบเซนเซอร์ตรวจจับควันขณะที่ยังไม่มีควัน

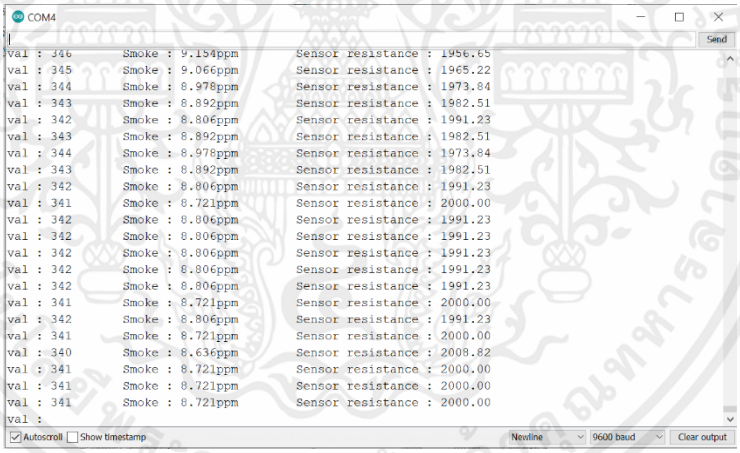


รูปที่ 4.3 หน้าต่าง Serial Monitor ที่แสดงค่าสัญญาณแอนะล็อกจากเซนเซอร์ตรวจจับควันขณะที่ยังไม่มีควัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 ทำการทดสอบเซนเซอร์ตรวจจับควันขณะที่มีควัน



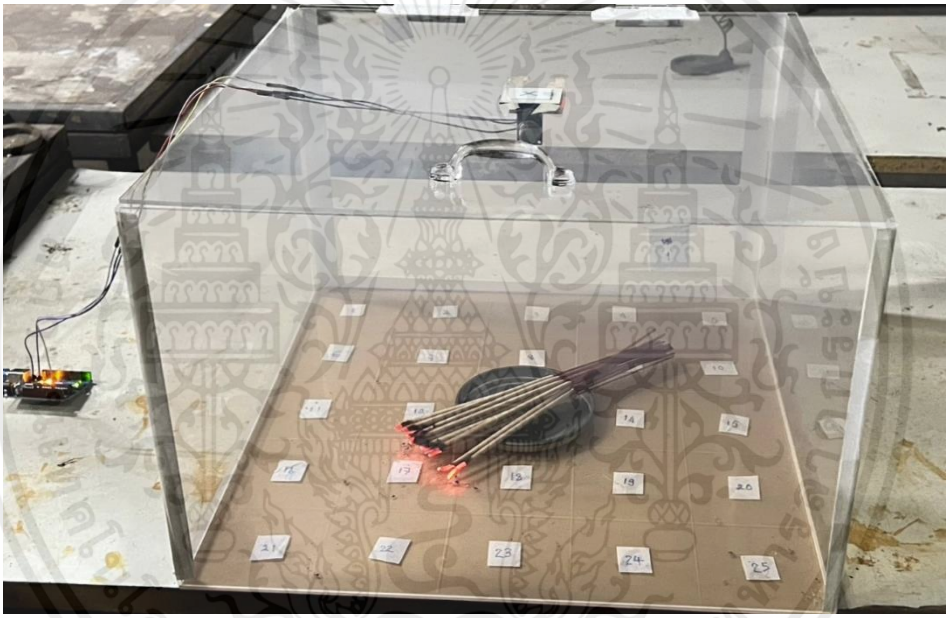
รูปที่ 4.5 หน้าต่าง Serial Monitor ที่แสดงค่าสัญญาณแอนะล็อกจากเซนเซอร์ตรวจจับควันขณะที่มีควัน

4.1.2 การทดสอบความสามารถตรวจจับควันด้วยเซนเซอร์ตรวจจับควัน (MQ-2) ในชุดจำลองห้องพัก

ในการทดสอบความสามารถตรวจจับควันด้วยเซนเซอร์ตรวจจับควัน (MQ-2) ในชุดจำลองห้องพัก โดยใช้โปรแกรม Arduino (IDE) ในการเขียนโดยเขียนคำสั่งและระบุเงื่อนไขการทำงานของเซนเซอร์ตรวจจับควันลงบนบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 โดยทำการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บันทึกค่าปริมาณควันที่อยู่ในอากาศในหน่วย ppm ของการเผาวัสดุ 3 ชนิด คือ กาบมะพร้าว, เศษผ้า และธูป ในปริมาณที่เท่ากัน ซึ่งระยะห่างในการทดสอบระหว่างเซนเซอร์ตรวจจับและวัสดุมีระยะ 30 เซนติเมตร โดยติดเซนเซอร์ไว้ที่กึ่งกลางภายในด้านบนของชุดจำลองห้องพักขนาด 47.00 เซนติเมตร x 46.50 เซนติเมตร x 30.00 เซนติเมตร พร้อมทั้งจับเวลาในการทดสอบ 5 นาที โดยจะทำการบันทึกค่าปริมาณควัน (ppm) ทุก ๆ 1 นาที ดังในตารางที่ 3.4 โดยภาพการทดสอบ ดังในรูปที่ 20 และนำผลมาวิเคราะห์ความสามารถในการตรวจจับควันของเซนเซอร์ตรวจจับควัน (MQ-2) ของการเผาไหม้วัสดุทั้ง 3 ชนิด และทำการวิเคราะห์ความสามารถตรวจจับค่าควันของวัสดุ 3 ชนิด ที่มีผลทำให้เกิดอัคคีภัยด้วย Infrared ดังในรูปที่ 4.6



รูปที่ 4.6 การทดสอบการตรวจจับควันไฟด้วยเซนเซอร์ตรวจจับควัน (MQ-2)

ตารางที่ 4.1 การทดสอบการตรวจจับควันไฟของเซนเซอร์ตรวจจับควัน (MQ-2) ในการเผาวัสดุต่างๆ

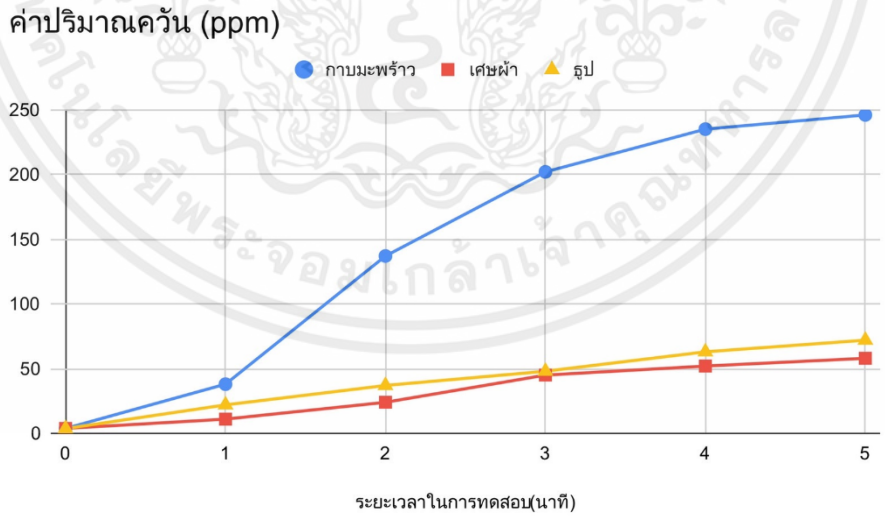
ระยะเวลาในการทดสอบ (นาที)	ค่าปริมาณควัน (ppm)		
	กาบมะพร้าว	เศษผ้า	ธูป
0	3.78	3.82	3.73
1	38.00	11.00	22.00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 การทดสอบการตรวจจับควันไฟของเซนเซอร์ตรวจจับควัน (MQ-2) ในการเผาวัสดุต่างๆ

ระยะเวลาในการทดสอบ (นาที)	ค่าปริมาณควัน (ppm)		
	กาบมะพร้าว	เศษผ้า	ธูป
2	137.00	24.00	37.00
3	202.00	45.00	48.00
4	235.00	52.00	63.00
5	246.00	58.00	72.00

จากตารางที่ 4.1 พบว่าเซนเซอร์ตรวจจับควัน (MQ-2) สามารถตรวจจับค่าควันของวัสดุชนิดกาบมะพร้าวได้มากที่สุด โดยในระยะเวลา 5 นาที มีค่าปริมาณควันสูงสุด 246.00 ppm ต่อมาคือธูปโดยในระยะเวลา 5 นาที มีค่าปริมาณควันสูงสุด 72.00 ppm และวัสดุที่ตรวจจับค่าควันได้น้อยที่สุดคือเศษผ้าโดยในระยะเวลา 5 นาทีมีค่าปริมาณควันสูงสุด 58.00 ppm เมื่อทำการบันทึกข้อมูลการทดสอบการตรวจจับควันไฟของเซนเซอร์ตรวจจับควัน (MQ-2) ในการเผาวัสดุต่างๆ สามารถวิเคราะห์ความสัมพันธ์ระหว่างค่าปริมาณควัน (ppm) กับระยะเวลาในการทดสอบ (นาที) ดังแสดงในรูปที่ 4.7

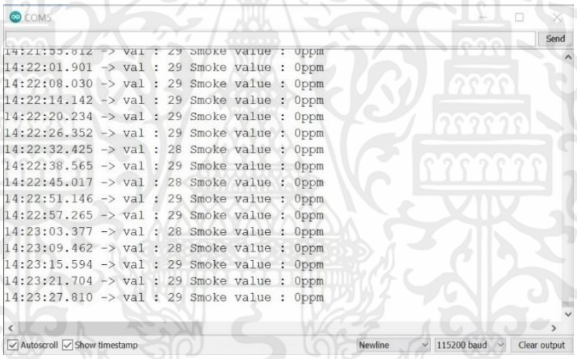


รูปที่ 4.7 ความสัมพันธ์ระหว่างค่าปริมาณควันกับระยะเวลาในการทดสอบ

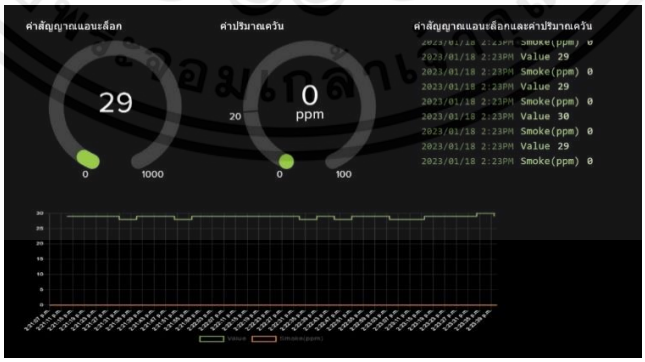
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 การทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าเซนเซอร์ตรวจจับควัน (MQ-2) ในหน่วย PPM แสดงผลบน Dashboard

ในการทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าเซนเซอร์ตรวจจับควัน (MQ-2) ในหน่วย PPM แสดงผลบน Dashboard โดยใช้โปรแกรม Arduino (IDE) ในการเขียนคำสั่งและระบุเงื่อนไขการทำงานของเซนเซอร์ตรวจจับเปลวไฟเพื่อส่งค่าสัญญาณแอนะล็อกที่เซนเซอร์ตรวจจับควันที่อ่านค่าได้ และค่าปริมาณควันในหน่วย ppm ไปแสดงผลบน Dashboard โดยจากผลการทดลองหน้าต่าง Serial monitor จะทำการแสดงค่าสัญญาณแอนะล็อกที่เซนเซอร์ตรวจจับควันอ่านค่าได้ และค่าปริมาณควันในหน่วย ppm ดังในรูปที่ 4.8 โดยในหน้า Dashboard จะแสดงค่าสัญญาณแอนะล็อกที่เซนเซอร์ตรวจจับควันที่อ่านค่าได้ และค่าปริมาณควันในหน่วย ppm แสดงว่า ข้อมูลจากบอร์ด NodeMCU ESP8266 V2 สามารถส่งเข้า Adafruit IO บนอินเตอร์เน็ตได้แล้ว ดังในรูปที่ 4.9



รูปที่ 4.8 หน้าต่าง Serial Monitor ที่แสดงค่าที่เซนเซอร์ตรวจจับควันอ่านค่าได้ และค่าปริมาณควัน หน่วย ppm



รูปที่ 4.9 หน้า Dashboards แสดงค่าและกราฟที่เซนเซอร์ตรวจจับควันอ่านค่าได้ และค่าปริมาณควัน หน่วย ppm

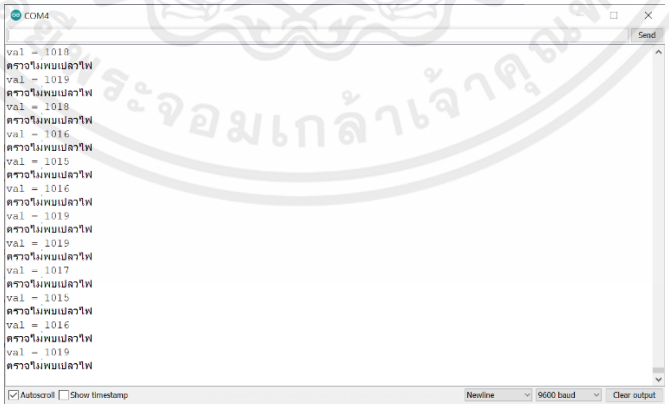
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดสอบชุดคำสั่งการทำงานของเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared

4.2.1 การทดสอบการทำงานของเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared

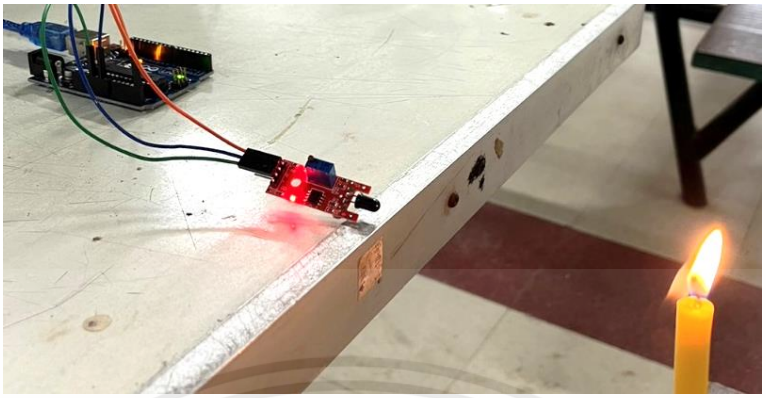
ในการทดสอบการทำงานของเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared โดยใช้โปรแกรม Arduino (IDE) ในการเขียนคำสั่งและระบุเงื่อนไขการทำงานของเซนเซอร์ตรวจจับเปลวไฟ โดยจากผลการทดลองเมื่อ Serial Monitor แสดงคำว่า “ตรวจพบเปลวไฟ” แสดงว่าเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared ทำการตรวจจับเปลวไฟในพื้นที่ได้ ดังในรูปที่ 4.10 และเมื่อ Serial Monitor แสดงคำว่า “ตรวจไม่พบเปลวไฟ” แสดงว่า เซนเซอร์ตรวจจับเปลวไฟด้วย Infrared ตรวจจับไม่พบเปลวไฟในพื้นที่ ดังในรูปที่ 4.11 โดยภาพการทดสอบเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared แสดงดังในรูปที่ 4.12

รูปที่ 4.10 หน้าต่าง Serial Monitor ที่แสดงการทำงานของเซนเซอร์ตรวจจับเปลวไฟ เมื่อทำการตรวจพบเปลวไฟในพื้นที่ได้



รูปที่ 4.11 หน้าต่าง Serial Monitor ที่แสดงการทำงานของเซนเซอร์ตรวจจับเปลวไฟ เมื่อทำการตรวจไม่พบเปลวไฟในพื้นที่

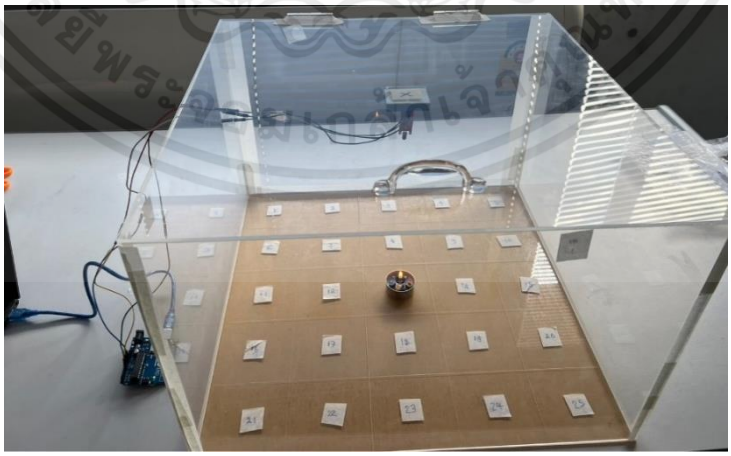
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 การทดสอบการทำงานของเซนเซอร์ตรวจจับเปลวไฟ

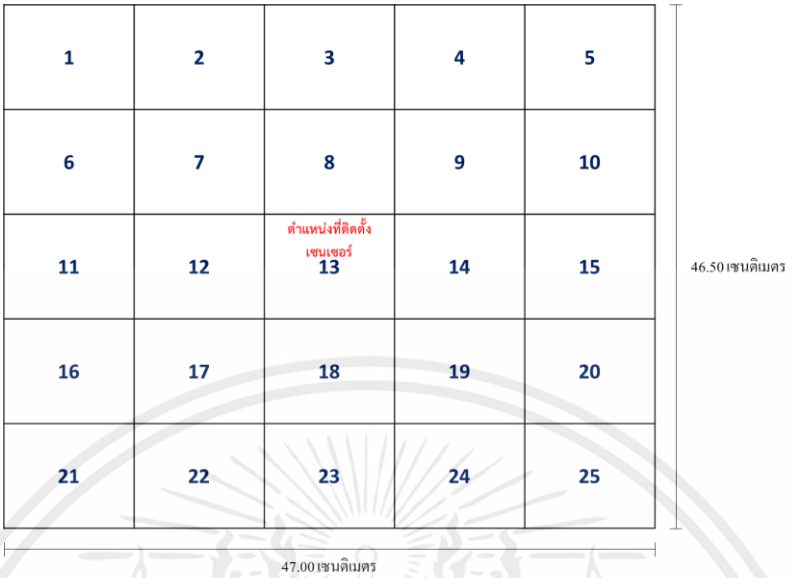
4.2.2 การทดสอบความสามารถเซนเซอร์ตรวจจับเปลวไฟในชุดจำลองห้องพัก

ในการทดสอบความสามารถตรวจจับเปลวไฟด้วยเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared ดังในรูปที่ 4.13 โดยเขียนคำสั่งและระบุเงื่อนไขการทำงานของเซนเซอร์ลงบนบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 โดยทำการบันทึกค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อกและสถานะการตรวจจับเปลวไฟในพื้นที่ โดยทำการจุดเทียนทดสอบการตรวจจับเปลวไฟของเซนเซอร์ตรวจจับเปลวไฟและกำหนดจุดจากพื้นที่จำลองขนาด 47.00 เซนติเมตร x 46.50 เซนติเมตร ออกเป็นทั้งหมด 25 จุด ดังในรูปที่ 4.14 ที่มีระยะห่างจากเซนเซอร์ที่ความสูง 30 เซนติเมตรโดยติดเซนเซอร์ไว้ที่กึ่งกลางภายในด้านบนของชุดจำลอง ทำการทดสอบทั้งหมด 3 ครั้ง โดยผลการทดสอบครั้งที่ 1 แสดงดังตารางที่ 4.2 โดยผลการทดสอบครั้งที่ 2 แสดงดังตารางที่ 4.3 โดยผลการทดสอบครั้งที่ 3 แสดงดังตารางที่ 4.4



รูปที่ 4.13 การทดสอบการตรวจจับเปลวไฟของเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 มุมบนแสดงตำแหน่งที่จุดเทียนในการทดสอบ

ตารางที่ 4.2 การทดสอบการตรวจจับเปลวไฟของเซนเซอร์ตรวจเปลวไฟ Infrared ครั้งที่ 1

ตำแหน่งการทดสอบ	ค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อก	สถานะการตรวจจับเปลวไฟ (ตรวจจับได้ /ตรวจจับไม่ได้)
1	665	ตรวจจับได้
2	597	ตรวจจับได้
3	482	ตรวจจับได้
4	548	ตรวจจับได้
5	620	ตรวจจับได้
6	442	ตรวจจับได้
7	351	ตรวจจับได้
8	49	ตรวจจับได้
9	335	ตรวจจับได้
10	652	ตรวจจับได้
11	530	ตรวจจับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 การทดสอบการตรวจจับเปลวไฟของเซนเซอร์ตรวจเปลวไฟ Infrared ครั้งที่ 1

ตำแหน่งการทดสอบ	ค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อก	สถานะการตรวจจับเปลวไฟ (ตรวจจับได้ /ตรวจจับไม่ได้)
12	173	ตรวจจับได้
13	30	ตรวจจับได้
14	57	ตรวจจับได้
15	502	ตรวจจับได้
16	530	ตรวจจับได้
17	186	ตรวจจับได้
18	49	ตรวจจับได้
19	554	ตรวจจับได้
20	592	ตรวจจับได้
21	539	ตรวจจับได้
22	521	ตรวจจับได้
23	607	ตรวจจับได้
24	630	ตรวจจับได้
25	596	ตรวจจับได้

ตารางที่ 4.3 การทดสอบการตรวจจับเปลวไฟของเซนเซอร์ตรวจเปลวไฟ Infrared ครั้งที่ 2

ตำแหน่งการทดสอบ	ค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อก	สถานะการตรวจจับเปลวไฟ (ตรวจจับได้ /ตรวจจับไม่ได้)
1	648	ตรวจจับได้
2	593	ตรวจจับได้
3	473	ตรวจจับได้
4	557	ตรวจจับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 การทดสอบการตรวจจับเปลวไฟของเซนเซอร์ตรวจเปลวไฟ Infrared ครั้งที่ 2

ตำแหน่งการทดสอบ	ค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อก	สถานะการตรวจจับเปลวไฟ (ตรวจจับได้ /ตรวจจับไม่ได้)
5	614	ตรวจจับได้
6	453	ตรวจจับได้
7	367	ตรวจจับได้
8	47	ตรวจจับได้
9	329	ตรวจจับได้
10	659	ตรวจจับได้
11	523	ตรวจจับได้
12	168	ตรวจจับได้
13	30	ตรวจจับได้
14	69	ตรวจจับได้
15	507	ตรวจจับได้
16	525	ตรวจจับได้
17	192	ตรวจจับได้
18	48	ตรวจจับได้
19	523	ตรวจจับได้
20	604	ตรวจจับได้
21	528	ตรวจจับได้
22	517	ตรวจจับได้
23	609	ตรวจจับได้
24	612	ตรวจจับได้
25	588	ตรวจจับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 การทดสอบการตรวจจับเปลวไฟของเซนเซอร์ตรวจเปลวไฟ Infrared ครั้งที่ 3

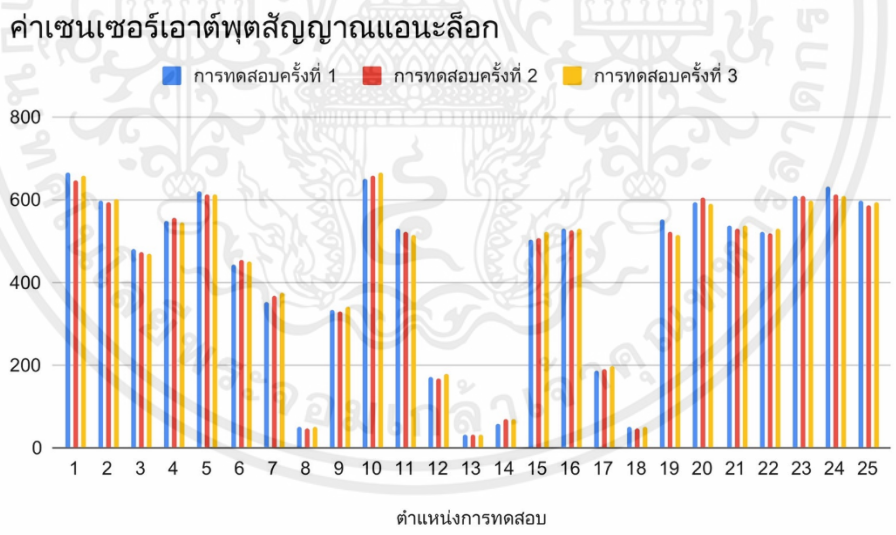
ตำแหน่งการทดสอบ	ค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อก	สถานะการตรวจจับเปลวไฟ (ตรวจจับได้ /ตรวจจับไม่ได้)
1	658	ตรวจจับได้
2	603	ตรวจจับได้
3	468	ตรวจจับได้
4	545	ตรวจจับได้
5	612	ตรวจจับได้
6	449	ตรวจจับได้
7	375	ตรวจจับได้
8	49	ตรวจจับได้
9	340	ตรวจจับได้
10	665	ตรวจจับได้
11	516	ตรวจจับได้
12	178	ตรวจจับได้
13	30	ตรวจจับได้
14	68	ตรวจจับได้
15	521	ตรวจจับได้
16	528	ตรวจจับได้
17	197	ตรวจจับได้
18	49	ตรวจจับได้
19	514	ตรวจจับได้
20	589	ตรวจจับได้
21	538	ตรวจจับได้
22	529	ตรวจจับได้
23	598	ตรวจจับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 การทดสอบการตรวจจับเปลวไฟของเซนเซอร์ตรวจเปลวไฟ Infrared ครั้งที่ 3

ตำแหน่งการทดสอบ	ค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อก	สถานะการตรวจจับเปลวไฟ (ตรวจจับได้ /ตรวจจับไม่ได้)
24	608	ตรวจจับได้
25	592	ตรวจจับได้

จากตารางที่ 4.2-4.4 แสดงผลการทดสอบความสามารถตรวจจับเปลวไฟทั้งหมด 3 ครั้ง พบว่าความสามารถตรวจจับเปลวไฟด้วยเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared ในพื้นที่จำลองขนาด 47.00 เซนติเมตร x 46.50 เซนติเมตร มีความสามารถในการตรวจจับเปลวไฟในพื้นที่จำลองได้ในทุกจุด โดยมีค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อกอยู่ในช่วงระหว่าง 30–665 เมื่อทำการบันทึกข้อมูลสถานการณ์ตรวจจับเปลวไฟของเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared สามารถวิเคราะห์ความสัมพันธ์ระหว่างค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อกกับตำแหน่งการทดสอบที่ 1 ถึง 25 ดังแสดงในรูปที่ 4.15



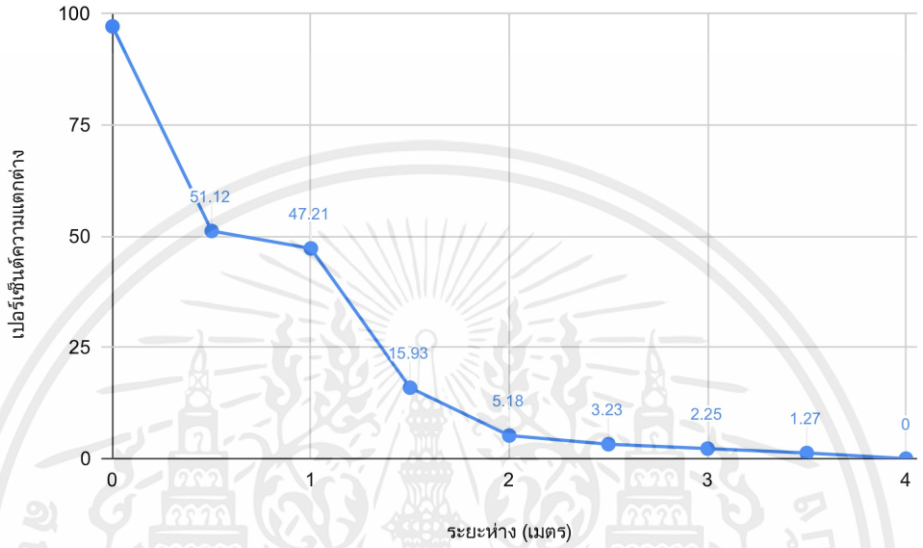
รูปที่ 4.15 ความสัมพันธ์ระหว่างค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อกกับตำแหน่งการทดสอบ

4.2.3 การทดสอบหาระยะที่เซนเซอร์สามารถตรวจจับเปลวไฟด้วย Infrared

ในการทดสอบหาระยะที่เซนเซอร์สามารถตรวจจับเปลวไฟได้ด้วยเซนเซอร์ตรวจจับเปลวไฟ โดยทำการบันทึกค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อกที่สภาวะปกติและบันทึกค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซนเซอร์เอาต์พุตสัญญาณแอนะล็อกเมื่อทำการจุดไฟโดยเริ่มต้นระยะ 0 เมตรจนถึงระยะที่ค่าเซนเซอร์เอาต์พุตสัญญาณแอนะล็อกสามารถตรวจจับเปลวไฟได้ในสภาวะปกติแล้วนำมาวิเคราะห์หาเปอร์เซ็นต์ความแตกต่างระหว่างสภาวะปกติและเมื่อจุดไฟ ที่ระยะทางต่างๆ



รูปที่ 4.16 เปอร์เซ็นต์ความแตกต่างระหว่างสภาวะปกติและเมื่อจุดไฟ ที่ระยะทางต่างๆ

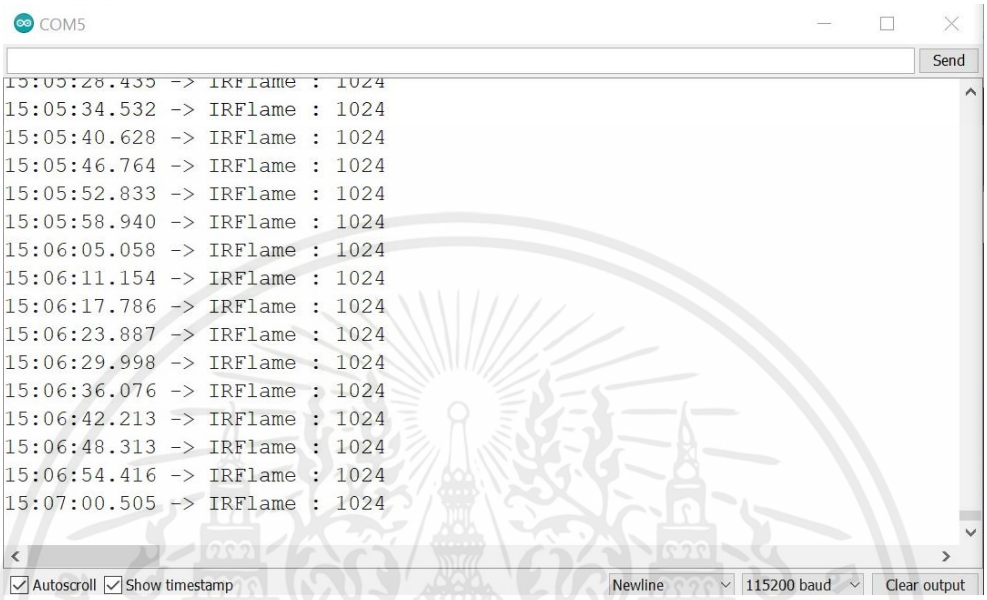
จากการทดสอบหาระยะที่เซนเซอร์สามารถตรวจจับเปลวไฟ ดังในรูปที่ 4.16 พบว่าเปอร์เซ็นต์ความแตกต่างระหว่างสภาวะปกติและเมื่อจุดไฟที่มีค่าน้อยที่สุดคือระยะ 3.5 เมตร มีค่า 1.27 เปอร์เซ็นต์ แสดงว่าระยะที่เซนเซอร์สามารถตรวจจับเปลวไฟมีระยะการตรวจจับที่ 3.5 เมตร ถ้าหากมีเปลวไฟนอกระยะเซนเซอร์จะไม่สามารถตรวจจับเปลวไฟได้

4.2.4 การทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าเซนเซอร์ตรวจจับเปลวไฟแสดงผลบน Dashboard

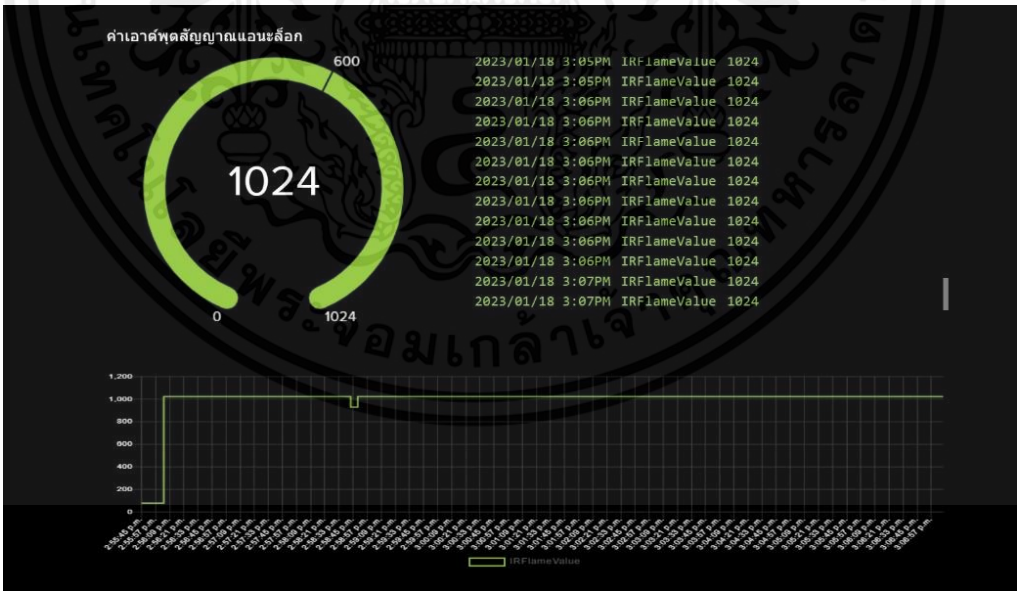
ในการทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าเซนเซอร์ตรวจจับเปลวไฟแสดงผลบน Dashboard โดยใช้โปรแกรม Arduino (IDE) ในการเขียนคำสั่งและระบุเงื่อนไขการทำงานของเซนเซอร์ตรวจจับเปลวไฟเพื่อส่งค่าเอาต์พุตสัญญาณแอนะล็อกที่เซนเซอร์ตรวจจับเปลวไฟอ่านค่าได้ไปแสดงผลบน Dashboard โดยจากผลการทดลองหน้าต่าง Serial monitor จะทำการแสดงค่าเอาต์พุตสัญญาณแอนะล็อกที่เซนเซอร์ตรวจจับเปลวไฟอ่านค่าได้ ดังในรูปที่ 4.17 โดยในหน้า Dashboard ดังในรูปที่ 4.18 จะแสดงค่าเอาต์พุตสัญญาณแอนะล็อกที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซนเซอร์ตรวจจับเปลวไฟอ่านค่าได้ แสดงว่า ข้อมูลจากบอร์ด NodeMCU ESP8266 V2 สามารถส่งเข้า Adafruit IO บนอินเตอร์เน็ตได้แล้ว



รูปที่ 4.17 หน้าต่าง Serial Monitor ที่แสดงค่าเอาต์พุตสัญญาณแอนะล็อกที่เซนเซอร์ตรวจจับเปลวไฟอ่านค่าได้



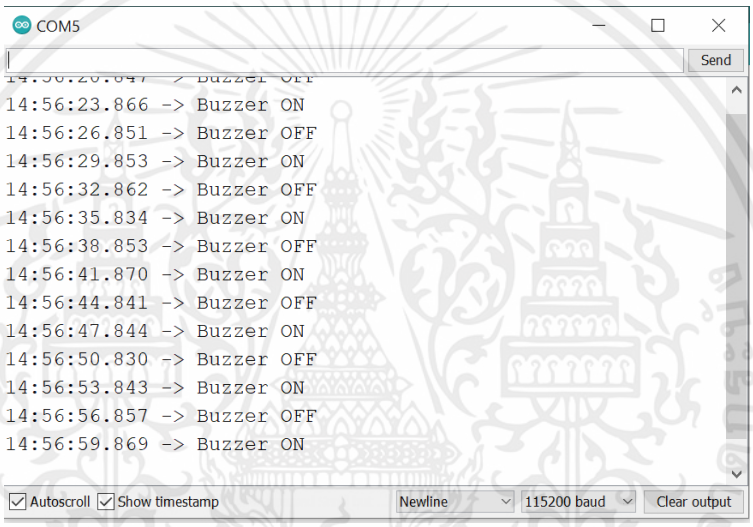
รูปที่ 4.18 หน้า Dashboard ที่แสดงค่าเอาต์พุตสัญญาณแอนะล็อกของเซนเซอร์ตรวจจับเปลวไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการทดสอบชุดคำสั่งการทำงานของโมดูล Active Buzzer

4.3.1 การทดสอบการทำงานของโมดูล Active Buzzer

ในการทดสอบการทำงานของโมดูล Active Buzzer โดยใช้โปรแกรม Arduino (IDE) ในการเขียนคำสั่งและระบุเงื่อนไขการทำงานของโมดูล Active Buzzer ในการสร้างสัญญาณเสียงแจ้งเตือน โดยจากผลการทดลองเมื่อ Serial Monitor แสดงคำว่า “Buzzer on!” แสดงว่า โมดูล Active Buzzer ได้สร้างสัญญาณเสียงเตือน และเมื่อ Serial Monitor แสดงคำว่า “Buzzer off!” แสดงว่า โมดูล Active Buzzer ไม่ได้สร้างสัญญาณเสียงเตือนขึ้น ดังในรูปที่ 4.19



รูปที่ 4.19 หน้าต่าง Serial Monitor ที่แสดงการทำงานของ Active Buzzer

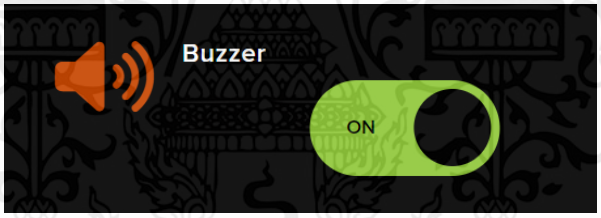
4.3.2 การทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการแสดงสถานะการทำงานของโมดูล Active Buzzer ในการส่งสัญญาณเสียงเตือนแสดงผลบน Dashboard

ในการทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการแสดงสถานะการทำงานของโมดูล Active Buzzer ในการส่งสัญญาณเสียงเตือนแสดงผลบน Dashboard โดยใช้โปรแกรม Arduino (IDE) ในการเขียนคำสั่งและระบุเงื่อนไขการทำงานของเซนเซอร์ตรวจจับควัน (MQ-2) และเซนเซอร์ตรวจจับเปลวไฟ และการส่งสถานะการทำงานของโมดูล Active Buzzer ไปแสดงผลบน Dashboard โดยจากผลการทดลองหน้าต่าง Serial Monitor แสดงคำว่า “ตรวจพบไฟไหม้” และ “Buzzer on!” ดังในรูปที่ 4.20 แสดงว่าเซนเซอร์ตรวจจับควัน และเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared ทำการตรวจพบควันไฟหรือเปลวไฟในพื้นที่ โดยโมดูล Active Buzzer ได้สร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณเสียงเตือนไฟไหม้ขึ้น โดยในหน้า Dashboard จะแสดงค่าสถานะการเปิดการทำงานของโมดูล Active Buzzer แสดงว่าข้อมูลจากบอร์ด NodeMCU ESP8266 V2 สามารถรับและส่งค่าเข้า Adafruit IO บนอินเตอร์เน็ตได้แล้ว ดังในรูปที่ 4.21

รูปที่ 4.20 หน้าต่าง Serial Monitor ที่แสดงการทำงานของ Active Buzzer เมื่อทำการตรวจจับพบควันไฟหรือเปลวไฟในพื้นที่

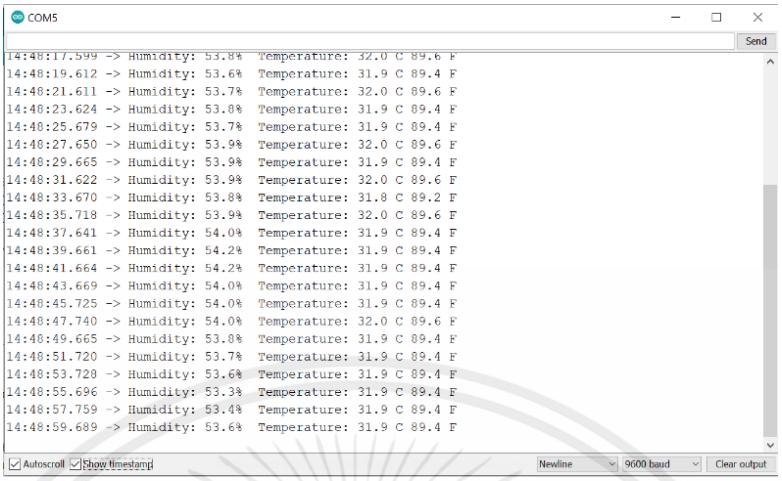


รูปที่ 4.21 หน้า Dashboard ที่แสดงสถานะการเปิดการทำงานของโมดูล Active Buzzer

4.4 ผลการทดสอบการควบคุมการทำงานของเซนเซอร์วัดอุณหภูมิและความชื้น (DHT22)

ในการทดสอบการควบคุมการทำงานของเซนเซอร์วัดอุณหภูมิและความชื้น (DHT22) โดยในการเขียนนั้นจะใช้ไลบรารี่อ้างอิงของ DHT-sensor-library-master และ Adafruit_Sensor-master ที่ได้มาจาก Github ที่ใช้ในการดึงค่าอุณหภูมิจากโมดูล DHT22 โดยโปรแกรมที่ใช้ควบคุมการทำงาน โดยจากผลการทดลองเมื่อ Serial Monitor แสดงคำว่า "DHT22 test!" แสดงว่าเซนเซอร์วัดอุณหภูมิและความชื้นพร้อมทำงานแล้ว โดยหน้าต่าง Serial monitor จะทำการแสดงค่าความชื้นสัมพัทธ์และค่าอุณหภูมิโดยเฉลี่ยในหน่วยองศาเซลเซียส และองศาฟาเรนไฮต์ ดังในรูปที่ 4.22

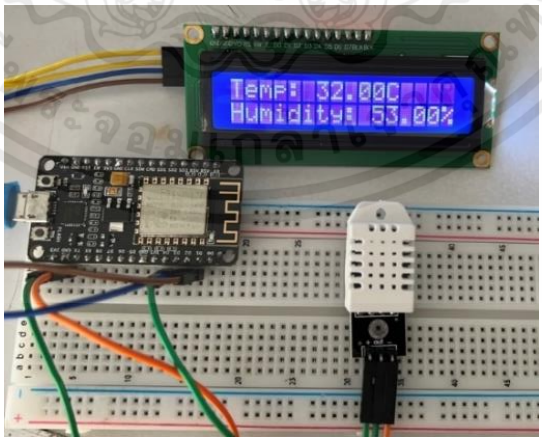
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 หน้าต่าง Serial Monitor ที่แสดงค่าความชื้นสัมพัทธ์และค่าอุณหภูมิโดยเฉลี่ย

4.5 ผลการทดสอบการควบคุมการแสดงค่าอุณหภูมิและความชื้นบนหน้าจอ LCD

ในการทดสอบการทำงานของเซนเซอร์วัดอุณหภูมิและความชื้น (DHT22) โดยในการเขียนนั้นจะใช้ไลบรารี่อ้างอิงของ Arduino-LiquidCrystal-I2C-library-master ที่ได้มาจาก Github ที่ใช้ในการควบคุมจอแสดงผล LCD แบบ I2C โดยโปรแกรมที่ใช้ควบคุมการทำงาน โดยจากผลการทดลองเมื่อ Serial Monitor แสดงคำว่า "DHT22 test!" แสดงว่าเซนเซอร์วัดอุณหภูมิและความชื้น และจอแสดงผล LCD พร้อมทำงานแล้ว โดยหน้าจอ Liquid Crystal Display (LCD) จะแสดงค่าอุณหภูมิในหน่วยองศาเซลเซียส และค่าความชื้นที่อ่านได้จากเซนเซอร์วัดอุณหภูมิและความชื้น (DHT22) ดังในรูปที่ 4.23



รูปที่ 4.23 การแสดงค่าอุณหภูมิและความชื้นบนหน้าจอ LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 ผลการทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าความชื้นสัมพัทธ์และค่าอุณหภูมิโดยเฉลี่ยในหน่วยองศาเซลเซียสแสดงผลบน Dashboard

ในผลการทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าความชื้นสัมพัทธ์และค่าอุณหภูมิโดยเฉลี่ยในหน่วยองศาเซลเซียสแสดงผลบน Dashboard โดยใช้โปรแกรม Arduino (IDE) ในการเขียนคำสั่งและระบุเงื่อนไขการทำงานของโมดูล DHT22 และจอแสดงผล Liquid Crystal Display (LCD) และการส่งค่าเซนเซอร์ไปแสดงผลบน Dashboard โดยจากผลการทดลองหน้าจอ Liquid Crystal Display (LCD) จะแสดงค่าอุณหภูมิในหน่วยองศาเซลเซียส และค่าความชื้นที่อ่านได้จากเซนเซอร์วัดอุณหภูมิและความชื้น (DHT22) ดังในรูปที่ 4.24 โดยในหน้า Dashboard จะแสดงค่าความชื้นสัมพัทธ์และค่าอุณหภูมิโดยเฉลี่ยในหน่วยองศาเซลเซียส แสดงว่าข้อมูลจากบอร์ด NodeMCU ESP8266 V2 สามารถส่งเข้า Adafruit IO บนอินเตอร์เน็ตได้แล้ว ดังในรูปที่ 4.25

รูปที่ 4.24 หน้าจอแสดงผล Liquid Crystal Display (LCD) จะแสดงค่าอุณหภูมิและความชื้น

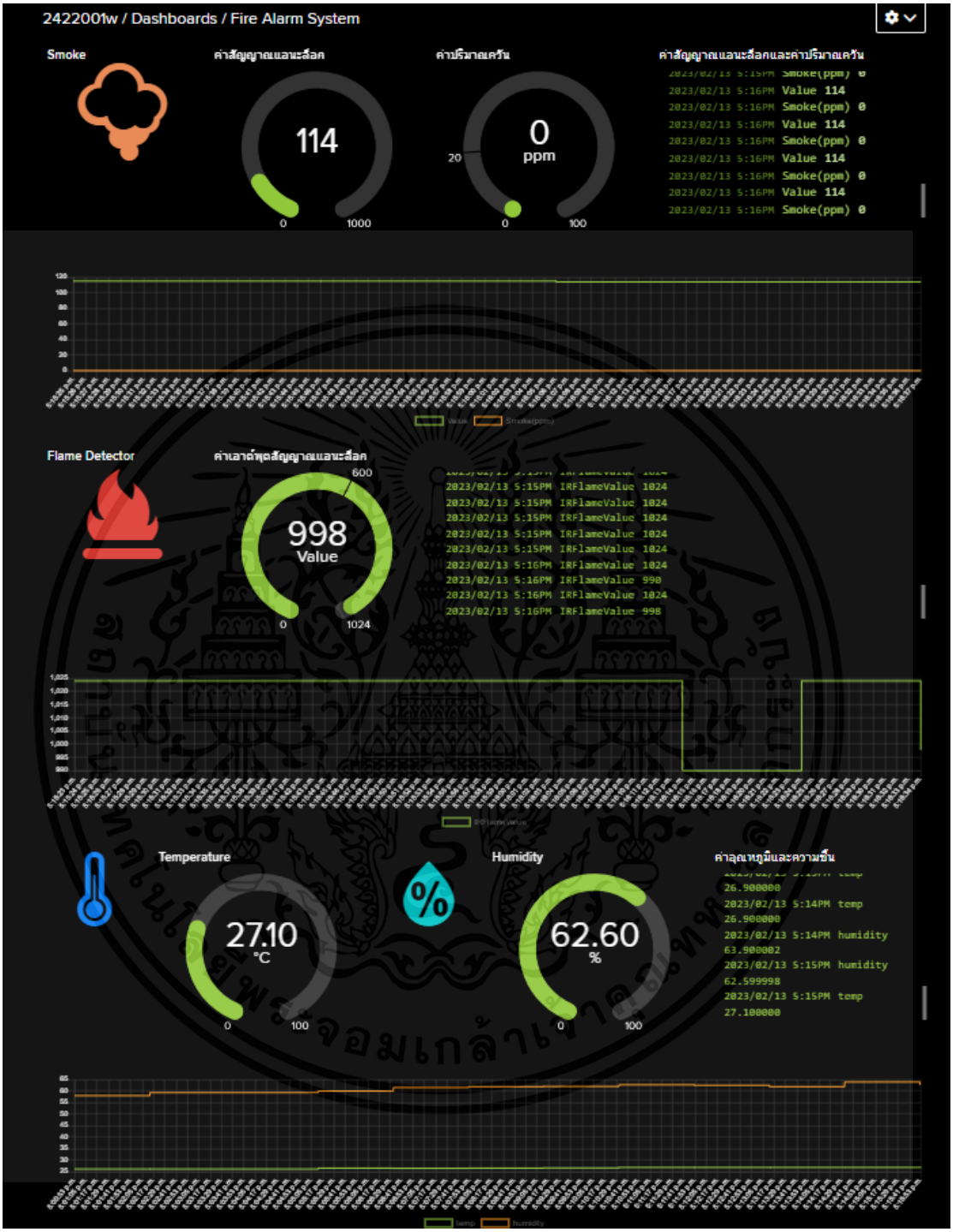
รูปที่ 4.25 หน้า Dashboard ที่แสดงค่าความชื้นสัมพัทธ์และค่าอุณหภูมิโดยเฉลี่ย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 ผลการทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าเซนเซอร์ตรวจจับควัน (MQ-2) เซนเซอร์ตรวจจับเปลวไฟ รวมถึงความชื้นสัมพัทธ์และอุณหภูมิโดยเฉลี่ยแสดงผลพร้อมกันบนหน้า Dashboard เดียวกัน

ผลการทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าเซนเซอร์ตรวจจับควัน (MQ-2) เซนเซอร์ตรวจจับเปลวไฟ รวมถึงค่าความชื้นสัมพัทธ์และอุณหภูมิโดยเฉลี่ยแสดงผลพร้อมกันบนหน้า Dashboard เดียวกัน โดยใช้โปรแกรม Arduino (IDE) ในการเขียนคำสั่งและระบุเงื่อนไขการทำงานของเซนเซอร์ตรวจจับควัน (MQ-2) เซนเซอร์ตรวจจับเปลวไฟ เซนเซอร์วัดอุณหภูมิและความชื้น (DHT22) และจอแสดงผล Liquid Crystal Display (LCD) ทำการส่งค่าเซนเซอร์ไปแสดงผลบน Dashboard โดยจากผลการทดลองในหน้า Dashboard จะแสดงค่าเซนเซอร์ตรวจจับควัน (MQ-2) เซนเซอร์ตรวจจับเปลวไฟ รวมถึงค่าความชื้นสัมพัทธ์และอุณหภูมิโดยเฉลี่ยที่แสดงผลพร้อมกันบนหน้า Dashboard เดียวกัน ดังในรูปที่ 4.26 โดยหน้าจอ Liquid Crystal Display (LCD) จะแสดงค่าอุณหภูมิในหน่วยองศาเซลเซียส และค่าความชื้นที่อ่านได้ ดังในรูปที่ 4.27 โดยภาพการทดสอบการแสดงค่าเซนเซอร์ตรวจจับควัน (MQ-2) เซนเซอร์ตรวจจับเปลวไฟ รวมถึงเซนเซอร์ความชื้นสัมพัทธ์และค่าอุณหภูมิโดยเฉลี่ยแสดงผลพร้อมกันบนหน้า Dashboard เดียวกัน ดังในรูปที่ 4.28

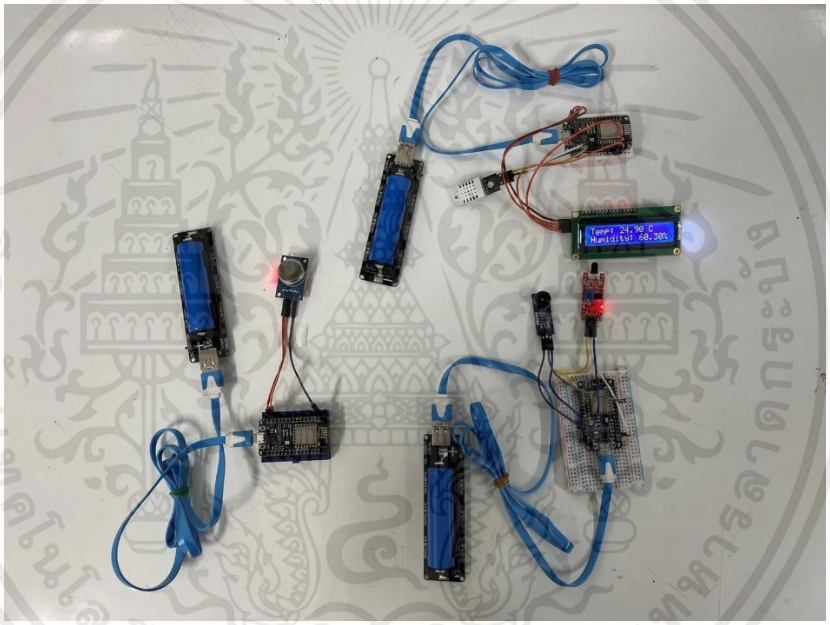
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.26 หน้า Dashboard ที่แสดงค่าเซนเซอร์ตรวจจับควัน เซนเซอร์ตรวจจับเปลวไฟ รวมถึงค่าความชื้นสัมพัทธ์และอุณหภูมิโดยเฉลี่ย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.27 หน้าจอแสดงผล Liquid Crystal Display (LCD) จะแสดงค่าอุณหภูมิและความชื้น



รูปที่ 4.28 ภาพการทดสอบการส่งค่าเซนเซอร์ตรวจจับควัน เซนเซอร์ตรวจจับเปลวไฟ รวมถึงค่าความชื้นสัมพัทธ์และอุณหภูมิโดยเฉลี่ยแสดงผลพร้อมกันบนหน้า Dashboard เดียวกัน

4.8 ผลการทดสอบชุดคำสั่งการทำงานของ SIM800L GSM Module

4.8.1 การทดสอบการใช้งาน SIM800L GSM Module สำหรับใช้ในการส่ง SMS

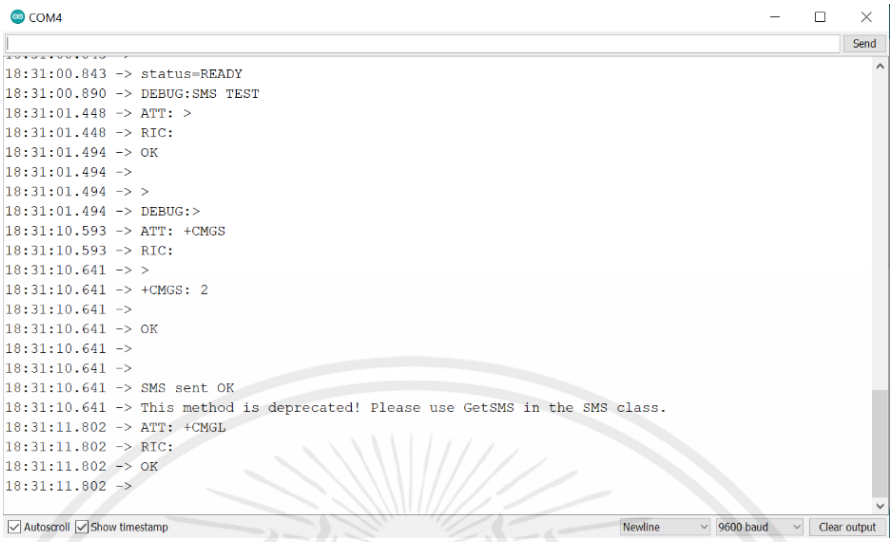
ในการทดสอบการใช้งาน SIM800L GSM Module สำหรับใช้ในการส่ง SMS โดยใช้โปรแกรม Arduino (IDE) ในการเขียนคำสั่งและระบุเงื่อนไขการทำงานของ SIM800L GSM Module โดยในการเขียนนั้นจะใช้ไลบรารี่อ้างอิงของ GSM900 ที่ได้มาจาก Github โดย SIM800L GSM Module สามารถรองรับในการรับส่ง SMS หาเบอร์โทรศัพท์ที่ต้องการ โดยในการตรวจสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

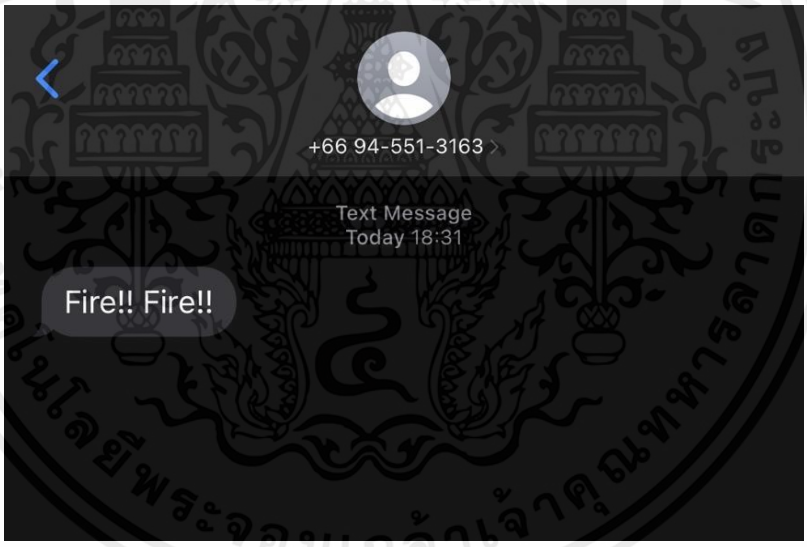
การเชื่อมต่อสัญญาณระหว่าง SIM800L GSM Module และซิม โดยทำการพิมพ์ข้อความ “AT+COPS?” ลงในช่องส่งข้อมูลแล้วกดปุ่ม Send ซึ่งเมื่อกดส่งข้อมูลไปแล้วจะเห็นได้ว่าที่หน้าต่าง Serial Monitor นั้นจะปรากฏชื่อสัญญาณเครือข่ายของซิมที่ใช้งานขึ้นมา ดังในรูปที่ 4.29 ซึ่งเป็นการยืนยันว่าตอนนี้ SIM800L GSM Module สามารถเชื่อมต่อสัญญาณเข้ากับซิมที่ใช้งานได้แล้ว โดยจากผลการทดลองเมื่อ Serial Monitor แสดงคำว่า“This method is deprecated! Please use GetSMS in the SMS class” ดังในรูปที่ 4.30 แสดงว่าข้อความ “Fire!! Fire!!” ที่กำหนดสามารถส่ง SMS ไปหาเบอร์โทรศัพท์ที่ต้องการได้แล้ว ดังในรูปที่ 4.31

รูปที่ 4.29 หน้าต่าง Serial Monitor ที่แสดงว่า SIM800L GSM Module สามารถเชื่อมต่อสัญญาณเข้ากับซิมได้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.30 หน้าต่าง Serial Monitor ที่แสดงว่าสามารถส่ง SMS ไปหาเบอร์โทรศัพท์ที่ต้องการได้



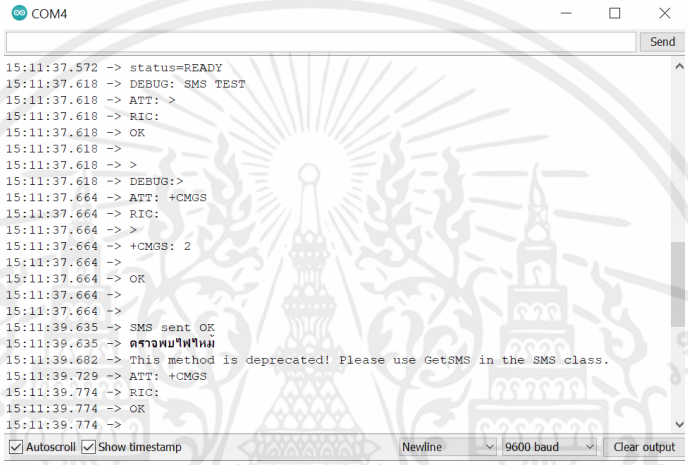
รูปที่ 4.31 ข้อความที่ถูกส่ง SMS ไปหาเบอร์โทรศัพท์ที่กำหนด

4.8.2 การทดสอบผลการทำงานร่วมกันของเซนเซอร์ตรวจจับควันและเซนเซอร์ตรวจจับเปลวไฟกับ SIM800L GSM Module ในการส่งข้อความ SMS แจ้งเตือนไฟไหม้

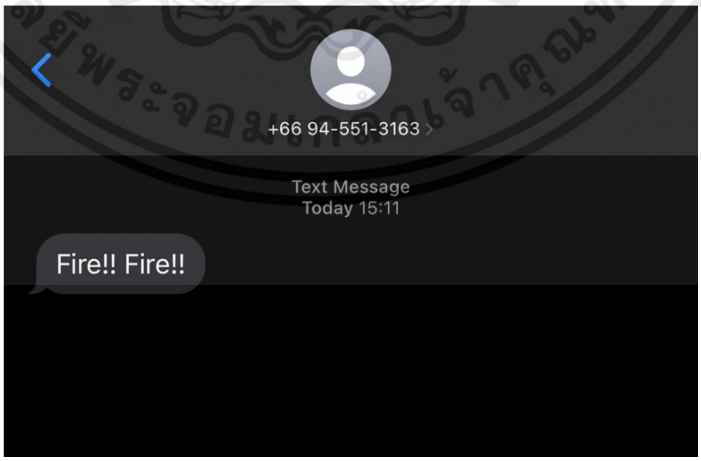
ในการทดสอบผลการทำงานร่วมกันของเซนเซอร์ตรวจจับควันและเซนเซอร์ตรวจจับเปลวไฟกับ SIM800L GSM Module ในการส่งข้อความ SMS แจ้งเตือนไฟไหม้ โดยใช้โปรแกรม Arduino (IDE) โดยจากผลการทดลองเมื่อ Serial Monitor แสดงคำว่า “ตรวจพบไฟไหม้” และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

“This method is deprecated! Please use GetSMS in the SMS class” ดังรูปที่ 4.32 แสดงว่าเซนเซอร์ตรวจจับควันทั้ง 3 ตัว และเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared ทั้ง 3 ตัว ทำการตรวจจับพบควันไฟหรือเปลวไฟในพื้นที่ และแสดงว่าข้อความ “Fire!! Fire!!” ที่แจ้งเตือนไฟไหม้สามารถส่ง SMS ไปยังสมาร์ทโฟนของผู้ดูแลต้องการได้แล้ว ดังในรูปที่ 4.33 เมื่อ Serial Monitor แสดงคำว่า “ตรวจไม่พบไฟไหม้” แสดงว่าเซนเซอร์ตรวจจับควันทั้ง 3 ตัว และเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared ทั้ง 3 ตัว ตรวจจับไม่พบควันไฟและเปลวไฟในพื้นที่ ดังในรูปที่ 4.34

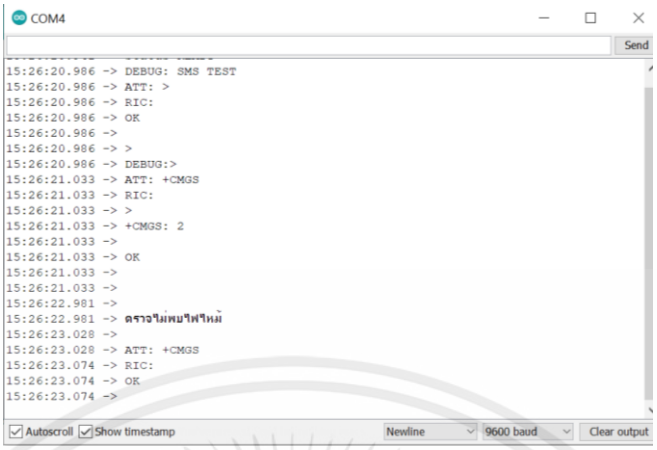


รูปที่ 4.32 หน้าต่าง Serial Monitor ที่แสดงการทำงานของเซนเซอร์ตรวจจับควันและเซนเซอร์ตรวจจับเปลวไฟ เมื่อทำการตรวจจับพบควันไฟหรือเปลวไฟในพื้นที่และสามารถส่ง SMS ไปหาเบอร์โทรศัพท์ที่ต้องการได้



รูปที่ 4.33 ข้อความแจ้งเตือนไฟไหม้ที่ถูกส่ง SMS ไปยังสมาร์ทโฟนของผู้ดูแล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



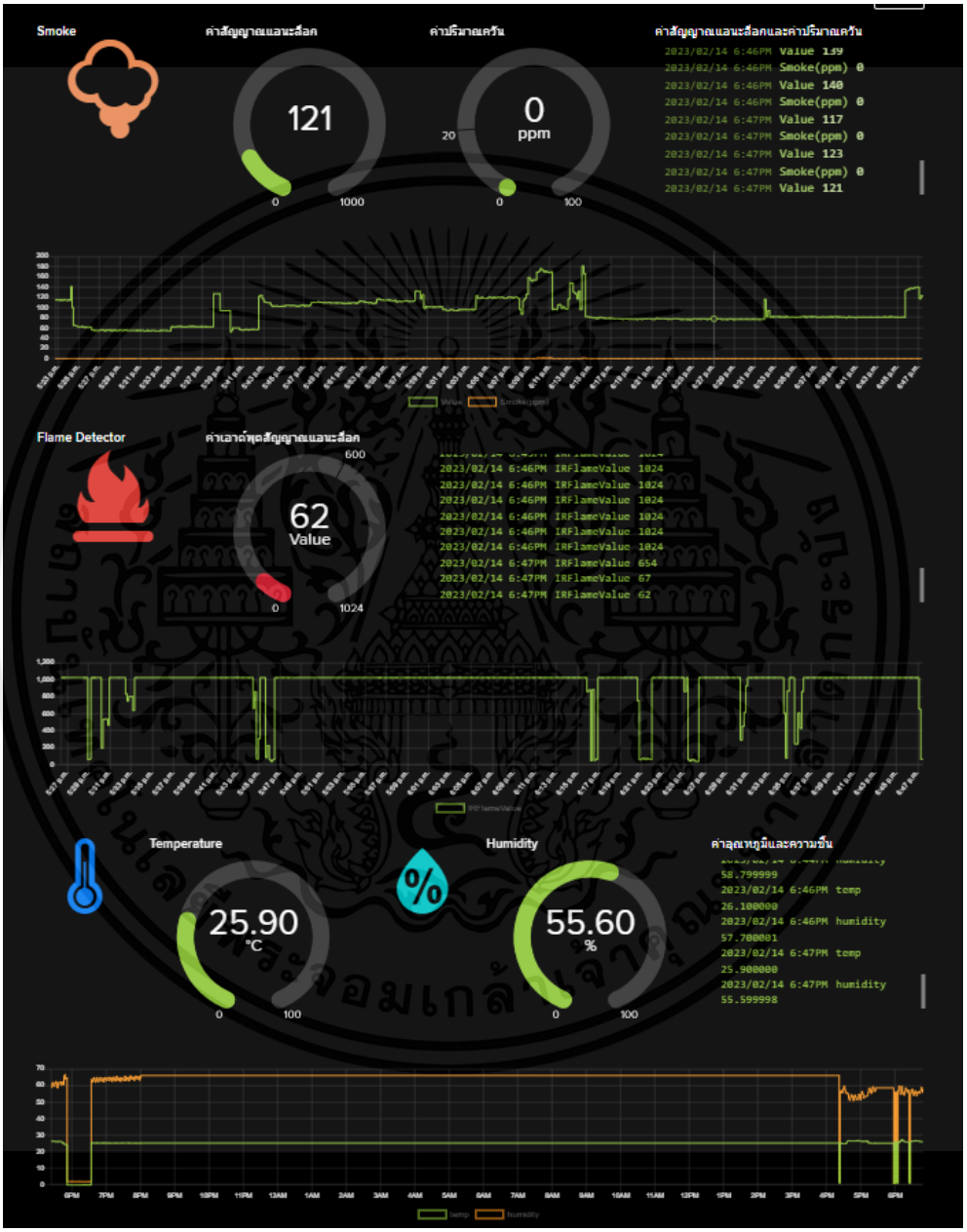
รูปที่ 4.34 หน้าต่าง Serial Monitor ที่แสดงการทำงานของเซนเซอร์ตรวจจับควันและเซนเซอร์ตรวจจับเปลวไฟ เมื่อทำการตรวจจับไม่พบควันไฟและเปลวไฟในพื้นที่

4.8.3 การทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการทำงานร่วมกันของเซนเซอร์ตรวจจับควัน (MQ-2) เซนเซอร์ตรวจจับเปลวไฟด้วย Infrared และเซนเซอร์วัดอุณหภูมิและความชื้น (DHT22) กับ SIM800L GSM Module ในการส่งข้อความ SMS แจ้งเตือนไฟไหม้

ในการทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการทำงานร่วมกันของเซนเซอร์ตรวจจับควัน (MQ-2) เซนเซอร์ตรวจจับเปลวไฟด้วย Infrared และเซนเซอร์วัดอุณหภูมิและความชื้น (DHT22) กับ SIM800L GSM Module ในการส่งข้อความ SMS แจ้งเตือนไฟไหม้ โดยใช้โปรแกรม Arduino (IDE) ในการเขียนคำสั่งและระบุเงื่อนไขการทำงานของเซนเซอร์ตรวจจับควัน (MQ-2) เซนเซอร์ตรวจจับเปลวไฟ เซนเซอร์วัดอุณหภูมิและความชื้น (DHT22) และจอแสดงผล Liquid Crystal Display (LCD) และการส่งค่าเซนเซอร์ไปแสดงผลบน Dashboard โดยจากผลการทดลองในหน้า Dashboard จะแสดงค่าเซนเซอร์ตรวจจับควัน (MQ-2) เซนเซอร์ตรวจจับเปลวไฟ รวมถึงค่าความชื้นสัมพัทธ์และอุณหภูมิโดยเฉลี่ยในหน่วยองศาที่ทำงานพร้อมกันบนหน้า Dashboard เดียวกัน ดังในรูปที่ 4.35 โดยหน้าจอ Liquid Crystal Display (LCD) จะแสดงค่าอุณหภูมิในหน่วยองศาเซลเซียส และค่าความชื้นที่อ่านได้ ดังในรูปที่ 4.36 เมื่อ Serial Monitor แสดงคำว่า “ตรวจพบไฟไหม้” และ “This method is deprecated! Please use GetSMS in the SMS class” ดังในรูปที่ 4.37 แสดงว่าเซนเซอร์ตรวจจับควัน และเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared ทำการตรวจจับพบควันไฟหรือเปลวไฟในพื้นที่ และแสดงว่าข้อความ “Fire!! Fire!!” ที่แจ้งเตือนไฟไหม้สามารถส่ง SMS ไปยังสมาร์ทโฟนของผู้ดูแลต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

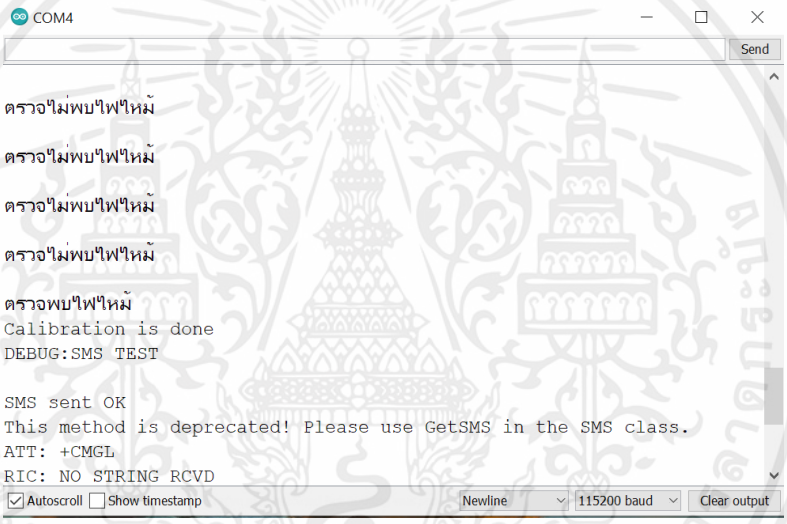
แล้ว ดังในรูปที่ 4.38 เมื่อ Serial Monitor แสดงคำว่า “ตรวจไม่พบไฟไหม้” แสดงว่าเซนเซอร์ตรวจจับควัน และเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared ตรวจจับไม่พบควันไฟและเปลวไฟในพื้นที่ ดังในรูปที่ 4.39



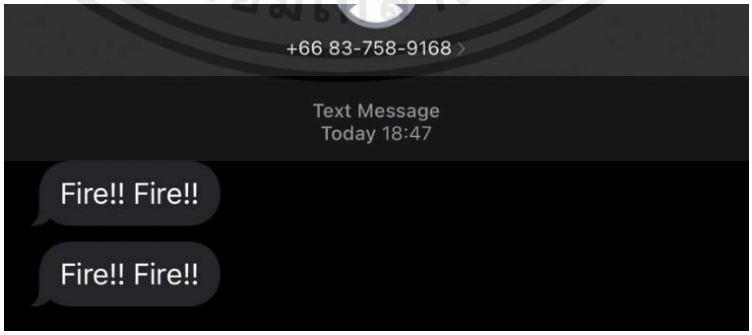
รูปที่ 4.35 หน้า Dashboard ที่แสดงค่าเซนเซอร์ตรวจจับควัน เซนเซอร์ตรวจจับเปลวไฟ รวมถึงค่าความชื้นสัมพัทธ์และอุณหภูมิโดยเฉลี่ย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.36 หน้าจอแสดงผล Liquid Crystal Display (LCD) จะแสดงค่าอุณหภูมิและความชื้น

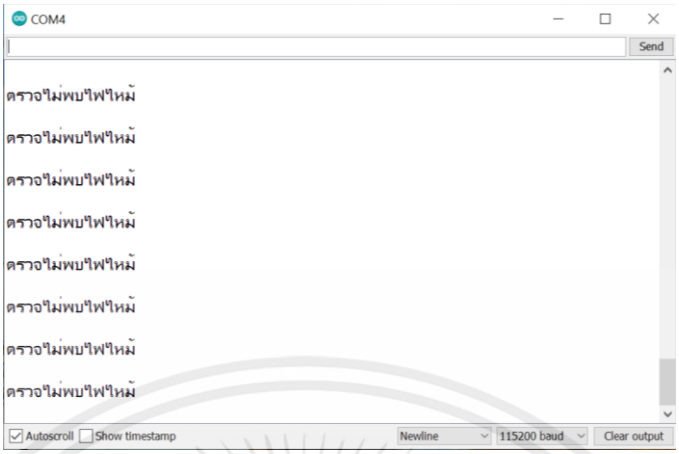


รูปที่ 4.37 หน้าต่าง Serial Monitor ที่แสดงการทำงานของเซนเซอร์ตรวจจับควัน และเซนเซอร์ตรวจจับเปลวไฟ เมื่อทำการตรวจจับพบควันไฟหรือเปลวไฟในพื้นที่ และสามารถส่ง SMS ไปหาเบอร์โทรศัพท์ที่ต้องการได้



รูปที่ 4.38 ข้อความแจ้งเตือนไฟไหม้ที่ถูกส่ง SMS ไปยังสมาร์ทโฟนของผู้ดูแล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

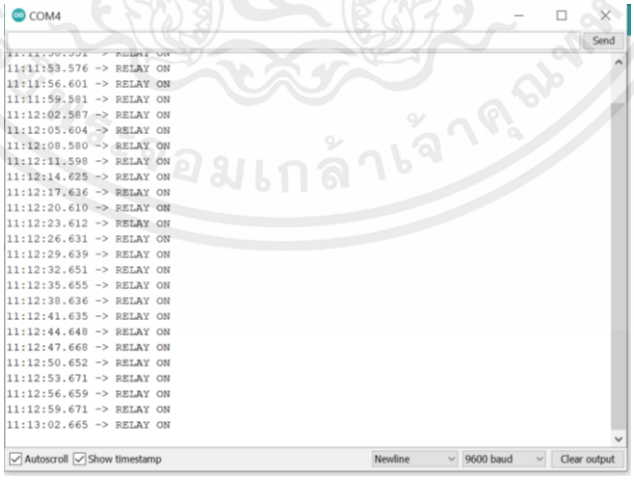


รูปที่ 4.39 หน้าต่าง Serial Monitor ที่แสดงการทำงานของเซนเซอร์ตรวจจับควัน และเซนเซอร์ตรวจจับเปลวไฟ เมื่อทำการตรวจจับไม่พบควันไฟและเปลวไฟในพื้นที่

4.9 ผลการทดสอบชุดคำสั่งการทำงานของโมดูล Relay 2 Channel

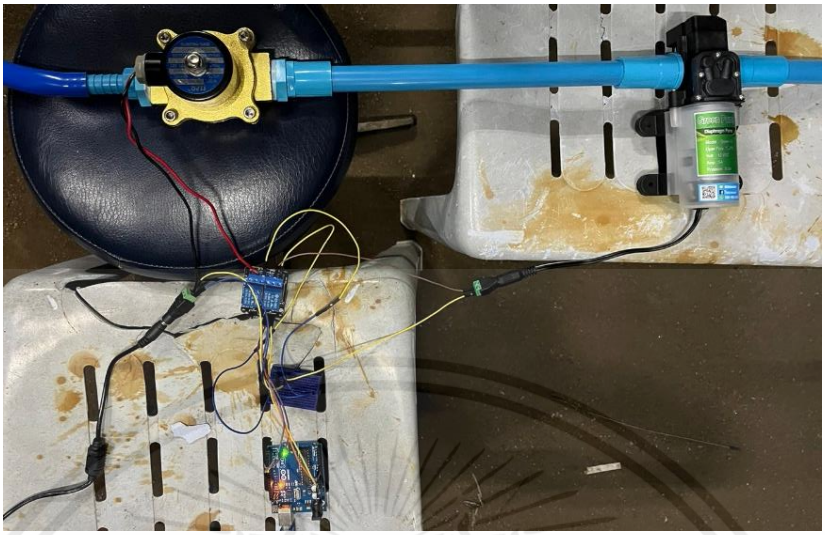
4.9.1 การทดสอบการใช้งานโมดูล Relay 2 Channel

ในการทดสอบการใช้งานโมดูล Relay 2 Channel โดยใช้โปรแกรม Arduino (IDE) ในการเขียนคำสั่งและระบุเงื่อนไขการทำงานของโมดูล Relay 2 Channel โดยจากผลการทดลอง เมื่อ Serial Monitor แสดงคำว่า " RELAY ON " ดังในรูปที่ 4.40 แสดงว่าโซลินอยด์วาล์ว (Solenoid Valve) และปั๊ม DC 12V เริ่มทำงาน และมินิสปริงเกอร์ (Mini Sprinkler) จะทำงานในการดับไฟเบื้องต้น ภาพการทดสอบโมดูล Relay 2 Channel แสดงดังในรูปที่ 4.41



รูปที่ 4.40 หน้าต่าง Serial Monitor ที่แสดงว่าโซลินอยด์วาล์ว และปั๊ม DC 12V เริ่มทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

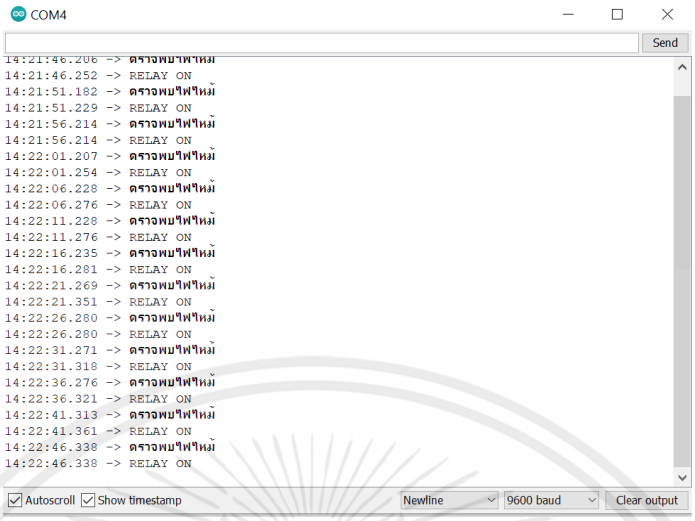


รูปที่ 4.41 ทำการทดสอบโมดูล Relay 2 Channel เพื่อทำการสั่งงานให้มีการดับไฟเบื้องต้น

4.9.2 การทดสอบการทำงานร่วมกันของเซนเซอร์ตรวจจับควัน (MQ-2) และเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared และการสั่งงานให้มีการดับไฟเบื้องต้น

ในการทดสอบการทำงานร่วมกันของเซนเซอร์ตรวจจับควัน (MQ-2) และเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared และการสั่งงานให้มีการดับไฟเบื้องต้น โดยใช้โปรแกรม Arduino (IDE) ในการเขียนคำสั่งและระบุเงื่อนไข โดยจากผลการทดลองเมื่อ Serial Monitor แสดงคำว่า “ตรวจพบไฟไหม้” และ " RELAY ON " ดังรูปที่ 4.42 แสดงว่าเซนเซอร์ตรวจจับควันทั้ง 3 ตัว และเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared ทั้ง 3 ตัว ทำการตรวจจับพบควันไฟหรือเปลวไฟในพื้นที่ทำให้โซลินอยด์วาล์ว (Solenoid Valve) และปั๊ม DC 12V เริ่มทำงาน และมินิสปริงเกอร์ (Mini Sprinkler) จะทำงานในการดับไฟเบื้องต้น ดังในรูปที่ 4.43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



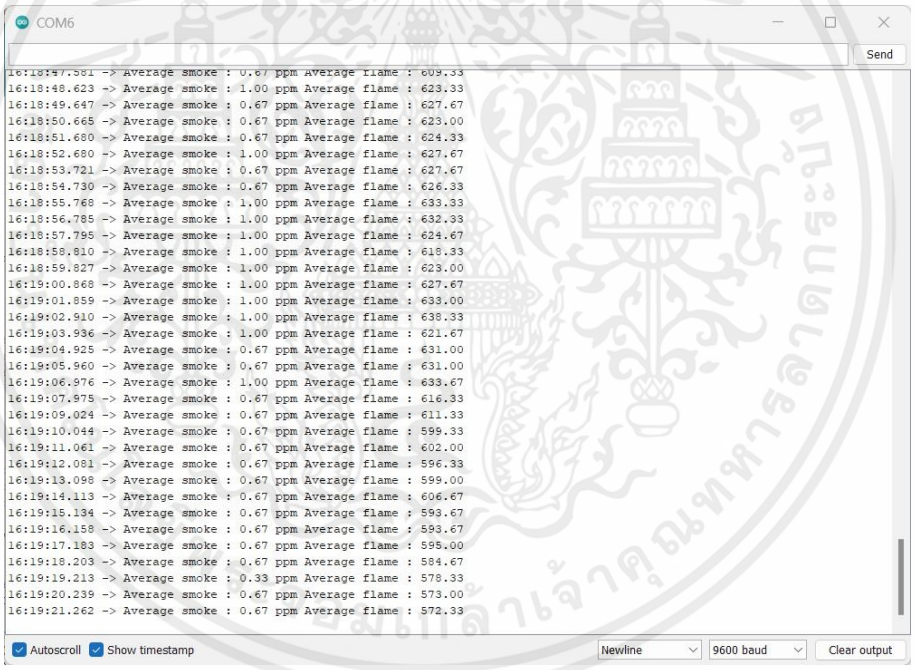
รูปที่ 4.42 หน้าต่าง Serial Monitor ที่แสดงการทำงานของเซนเซอร์ตรวจจับควันและเซนเซอร์ตรวจจับเปลวไฟ เมื่อทำการตรวจจับพบควันไฟหรือเปลวไฟในพื้นที่รวมถึงโซลินอยด์วาล์ว และปั๊ม DC 12V เริ่มทำงาน

รูปที่ 4.43 มินิสปริงเกอร์ (Mini Sprinkler) ทำงานในการดับไฟเบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

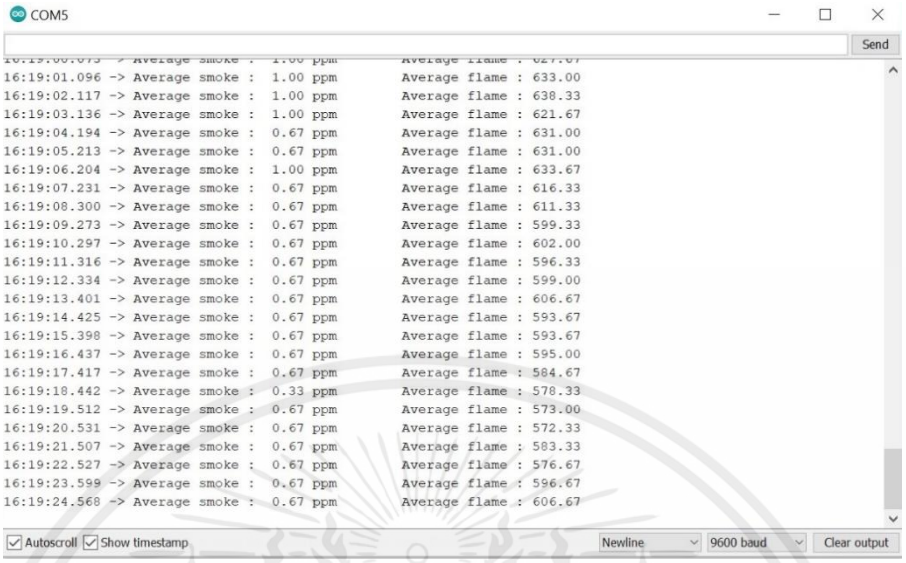
4.10 ผลการทดสอบการส่งค่าของเซนเซอร์จากบอร์ด Arduino Uno R3 ไปยังบอร์ด NodeMCU ESP8266 V2 แบบ Serial

ในการทดสอบการส่งค่าของเซนเซอร์จากบอร์ด Arduino Uno R3 ไปยังบอร์ด NodeMCU ESP8266 V2 แบบ Serial โดยใช้โปรแกรม Arduino (IDE) ในการเขียนคำสั่งและระบุเงื่อนไข โดยหน้าต่าง Serial monitor จะทำการแสดงค่าเซนเซอร์ตรวจจับควัน (MQ-2) และค่าเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared ที่ใช้บอร์ด Arduino Uno R3 เป็นตัวส่งค่า ดังในรูปที่ 4.44 และหน้าต่าง Serial monitor จะทำการแสดงค่าเซนเซอร์ตรวจจับควัน (MQ-2) และค่าเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared ที่ใช้ NodeMCU ESP8266 V2 เป็นตัวรับค่าที่ส่งมา ดังในรูปที่ 4.45ซึ่งแสดงให้เห็นว่าสามารถใช้บอร์ด Arduino Uno R3 สื่อสารกับบอร์ด NodeMCU ESP8266 V2 แบบ Serial ได้



รูปที่ 4.44 หน้าต่าง Serial monitor ที่แสดงค่าที่ใช้บอร์ด Arduino Uno R3 เป็นตัวส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

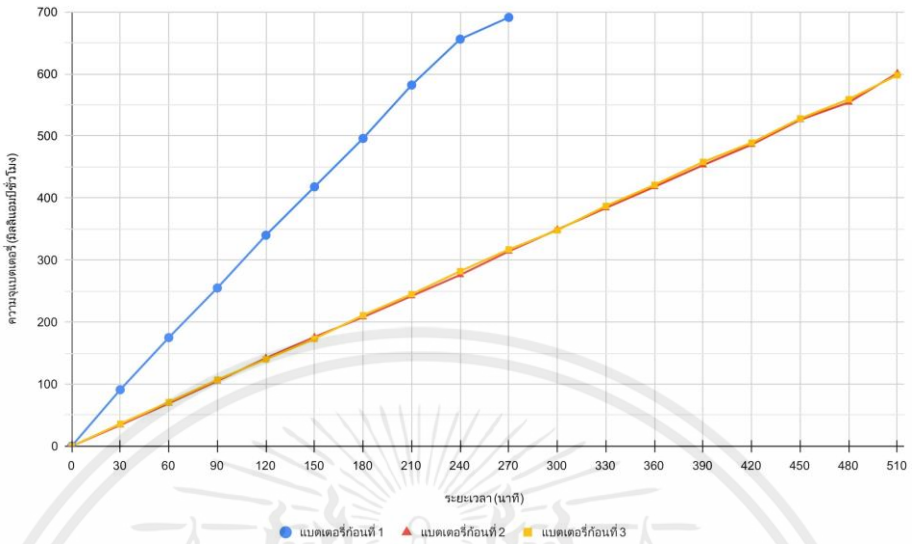


รูปที่ 4.45 หน้าต่าง Serial monitor ทำการแสดงค่าที่ใช้ NodeMCU ESP8266 V2 เป็นตัวรับค่า

4.11 ผลการทดสอบประสิทธิภาพของโมดูลชาร์จแบตเตอรี่ 18650

ในการทดสอบประสิทธิภาพของโมดูลชาร์จแบตเตอรี่ 18650 จำนวน 3 ก้อน โดยทำการชาร์จแบตเตอรี่ทั้ง 3 ก้อนให้เต็ม แล้วทำการต่อแบตเตอรี่กับมิเตอร์ดิจิทัล USB วัดค่าความจุแบตเตอรี่ที่ต่อเข้ากับบอร์ดไมโครคอนโทรลเลอร์ที่ใช้สำหรับประมวลผลข้อมูล ทำการบันทึกค่าความจุแบตเตอรี่ของทั้ง 3 ก้อน มีหน่วยเป็น mAh คือ มิลลิแอมป์ชั่วโมง ทำการวัดทุก 30 นาที จนกระทั่งแบตเตอรี่หมด สามารถวิเคราะห์ความสัมพันธ์ระหว่างค่าความจุแบตเตอรี่ทั้ง 3 ก้อนกับระยะเวลาในการใช้งาน ดังในรูปที่ 4.46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.46 ความสัมพันธ์ระหว่างค่าความจุแบตเตอรี่ทั้ง 3 ก้อนกับระยะเวลาในการใช้งาน

จากการทดสอบพบว่าแบตเตอรี่ก้อนที่ 1 มีการจ่ายกระแสไฟฟ้าเท่ากับ 0.15 แอมแปร์ เมื่อใช้งานแบตเตอรี่จนหมดจะมีค่าความจุแบตเตอรี่ที่สามารถจ่ายไฟได้เท่ากับ 691 มิลลิแอมป์ชั่วโมง โดยมีระยะเวลาการใช้งานประมาณ 270 นาที หรือ 4.5 ชั่วโมง แบตเตอรี่ก้อนที่ 2 มีค่ากระแสไฟฟ้าเท่ากับ 0.07 แอมแปร์ เมื่อใช้งานแบตเตอรี่จนหมดจะมีค่าความจุแบตเตอรี่ที่สามารถจ่ายไฟได้เท่ากับ 601 มิลลิแอมป์ชั่วโมง โดยมีระยะเวลาการใช้งานประมาณ 510 นาที หรือ 8.5 ชั่วโมง แบตเตอรี่ก้อนที่ 3 มีค่ากระแสไฟฟ้าเท่ากับ 0.07 แอมแปร์ เมื่อใช้งานแบตเตอรี่จนหมดจะมีค่าความจุแบตเตอรี่ที่สามารถจ่ายไฟได้เท่ากับ 598 มิลลิแอมป์ชั่วโมง โดยมีระยะเวลาการใช้งานประมาณ 510 นาที หรือ 8.5 ชั่วโมง

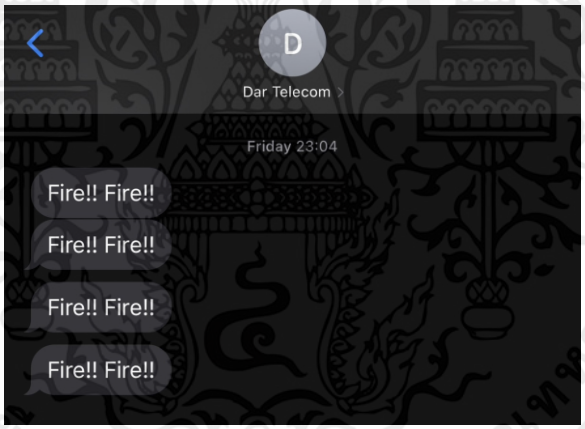
4.12 ผลการทดสอบระบบแจ้งเตือนอัคคีภัยในการสั่งงานให้มีการดับไฟเบื้องต้นและส่งข้อความ SMS แจ้งเตือนไฟไหม้

ในการทดสอบระบบแจ้งเตือนอัคคีภัยในการสั่งงานให้มีการดับไฟเบื้องต้นและส่งข้อความ SMS แจ้งเตือนไฟไหม้ โดยใช้โปรแกรม Arduino (IDE) ในการเขียนคำสั่งและระบุเงื่อนไข จากผลการทดลองเมื่อทำการทดสอบระบบจริง เมื่อเซนเซอร์ทำการตรวจจับพบควันและเปลวไฟในพื้นที่พบว่าสามารถส่งข้อความ SMS แจ้งเตือนไฟไหม้ไปหาเบอร์โทรศัพท์ที่ต้องการได้ โดยแสดงว่าข้อความ “Fire!! Fire!!” ดังในรูปที่ 4.47 และมีการสั่งงานให้โซลินอยด์วาล์ว (Solenoid Valve)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และปั๊ม DC 12V เริ่มทำงาน รวมถึงมินิสปริงเกอร์ (Mini Sprinkler) จะทำงานในการดับไฟเบื้องต้น ดังในรูปที่ 4.48

รูปที่ 4.47 มินิสปริงเกอร์ (Mini Sprinkler) ทำงานในการดับไฟเบื้องต้น



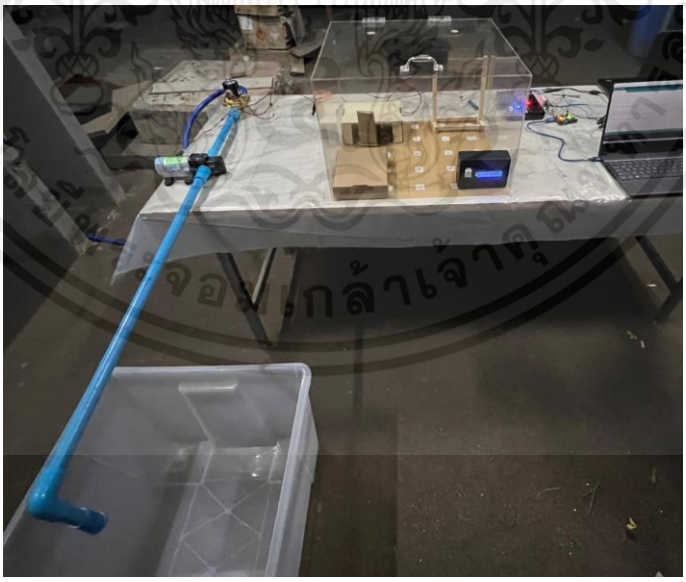
รูปที่ 4.48 ข้อความแจ้งเตือนไฟไหม้ที่ถูกส่ง SMS ไปยังสมาร์ทโฟนของผู้ดูแล

4.13 การทดสอบระบบแจ้งเตือนอัคคีภัยที่มีการใช้บริการคลาวด์ของ Adafruit IO ในการสั่งงานให้มีการส่งสัญญาณเสียงเตือนและส่งข้อความ SMS แจ้งเตือนไฟไหม้พร้อมกับมีการดับไฟเบื้องต้น

ในการทดสอบระบบแจ้งเตือนอัคคีภัยที่มีการใช้บริการคลาวด์ของ Adafruit IO ในการสั่งงานให้มีการส่งสัญญาณเสียงเตือนและส่งข้อความ SMS แจ้งเตือนไฟไหม้พร้อมกับมีการดับไฟเบื้องต้น โดยใช้โปรแกรม Arduino (IDE) ในการเขียนคำสั่งและระบุเงื่อนไข จากผลการทดลอง

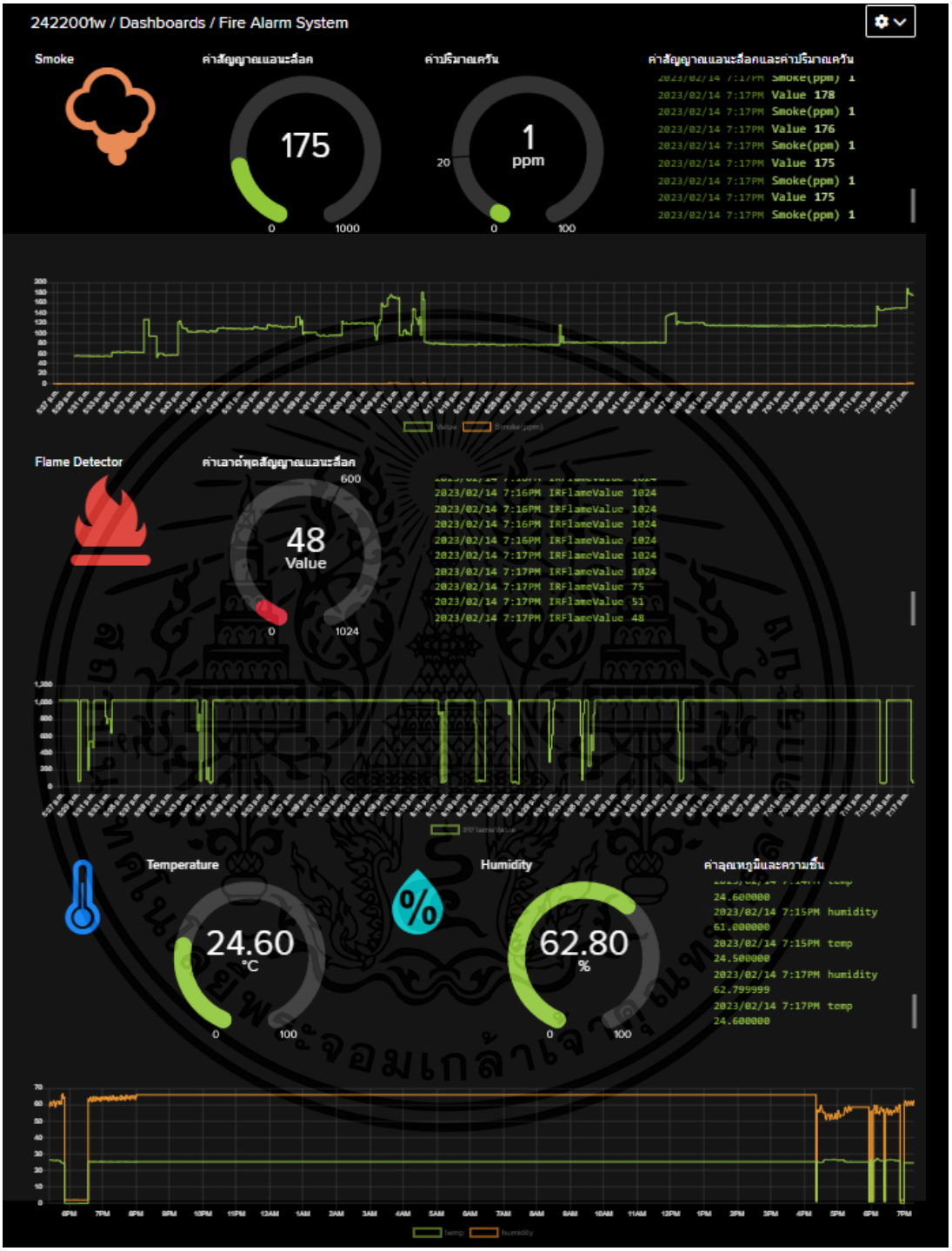
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการทดสอบระบบจริง ดังในรูปที่ 4.49 โดยในหน้า Dashboard จะแสดงค่าเซนเซอร์ตรวจจับควัน (MQ-2) เซนเซอร์ตรวจจับเปลวไฟ ค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ โดยเฉลี่ยในหน่วยองศาเซลเซียส รวมถึงสถานะการทำงานของโมดูล Active Buzzer และโมดูล Relay 2 Channel ที่ทำงานพร้อมกันบนหน้า Dashboard เดียวกัน ดังในรูปที่ 4.50 โดยหน้าจอ Liquid Crystal Display (LCD) จะแสดงค่าอุณหภูมิในหน่วยองศาเซลเซียสและค่าความชื้นที่อ่านได้ ดังในรูปที่ 4.51 เมื่อ Serial Monitor แสดงคำว่า “ตรวจพบไฟไหม้” และ “Buzzer on! Relay on!” และ“This method is deprecated! Please use GetSMS in the SMS class” ดังในรูปที่ 4.52 แสดงว่าเซนเซอร์ตรวจจับควัน และเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared ทำการตรวจจับพบควันไฟหรือเปลวไฟในพื้นที่ แสดงว่าโมดูล Active Buzzer ได้สร้างสัญญาณเสียงเตือน และสามารถส่งข้อความ SMS แจ้งเตือนไฟไหม้ไปหาเบอร์โทรศัพท์ที่ต้องการได้ โดยแสดงว่าข้อความ “ Fire!! Fire!!” ดังในรูปที่ 4.53 และมีการสั่งงานให้โซลินอยด์วาล์ว (Solenoid Valve) และปั๊ม DC 12V เริ่มทำงาน รวมถึงมินิสปริงเกอร์ (Mini Sprinkler) จะทำงานในการดับไฟเบื้องต้น ดังในรูปที่ 4.54 และบนหน้า Dashboard แสดงค่าสถานะการเปิดการทำงานของโมดูล Active Buzzer และโมดูล Relay 2 Channel ดังในรูปที่ 4.55 เมื่อ Serial Monitor แสดงคำว่า “ตรวจไม่พบไฟไหม้” แสดงว่าเซนเซอร์ตรวจจับควัน และเซนเซอร์ตรวจจับเปลวไฟด้วย Infrared ตรวจจับไม่พบควันไฟและเปลวไฟในพื้นที่ ดังในรูปที่ 4.56



รูปที่ 4.49 การทดสอบระบบแจ้งเตือนอัคคีภัย

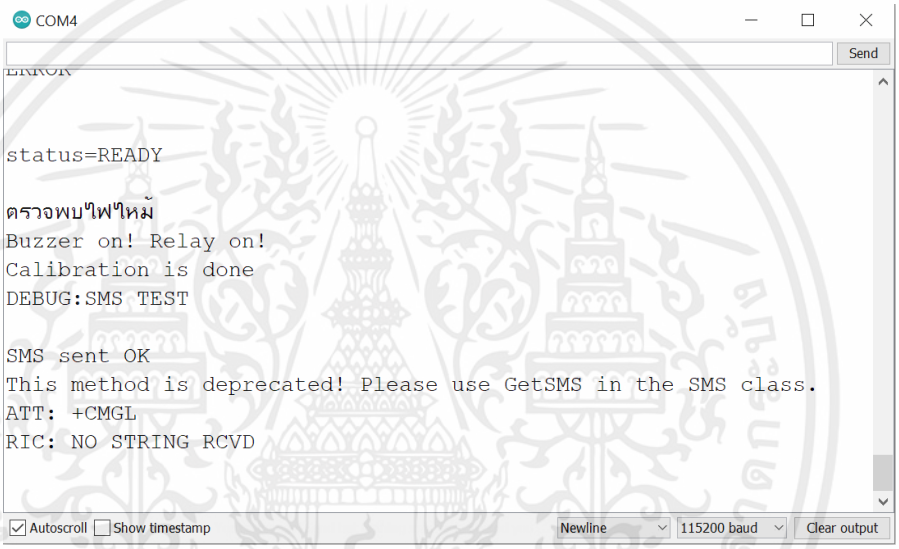
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



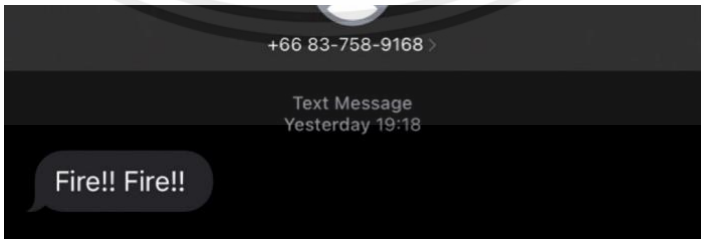
รูปที่ 4.50 หน้า Dashboard ที่แสดงค่าเซนเซอร์ตรวจจับควัน เซนเซอร์ตรวจจับเปลวไฟ ค่าความชื้นสัมพัทธ์และค่าอุณหภูมิโดยเฉลี่ย รวมถึงสถานะการทำงานของโมดูล Active Buzzer และโมดูล Relay 2 Channel ที่ทำงานพร้อมกันบนหน้า Dashboard เดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.51 หน้าจอแสดงผล Liquid Crystal Display (LCD) จะแสดงค่าอุณหภูมิและความชื้น



รูปที่ 4.52 หน้าต่าง Serial Monitor แสดงการทำงานของเซนเซอร์ตรวจจับควันและเปลวไฟ เมื่อทำการตรวจจับพบควันไฟหรือเปลวไฟในพื้นที่ และการทำงานของโมดูล Active Buzzer ที่สร้างเสียงเตือน รวมถึงโมดูล Relay 2 Channel เริ่มทำงาน และสามารถส่ง SMS ไปหาเบอร์โทรศัพท์ที่ต้องการ

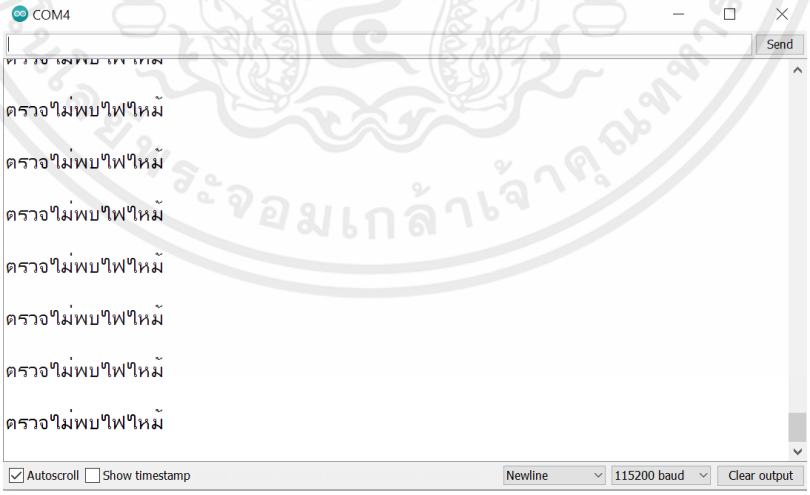


รูปที่ 4.53 ข้อความแจ้งเตือนไฟไหม้ที่ถูกส่ง SMS ไปยังสมาร์ทโฟนของผู้ดูแล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.54 มินิสปริงเกอร์ (Mini Sprinkler) ทำงานในการดับไฟเบื้องต้น

รูปที่ 4.55 หน้า Dashboard ที่แสดงสถานะการเปิดการทำงานของโมดูล Active Buzzer และโมดูล Relay 2 Channel



รูปที่ 4.56 หน้าต่าง Serial Monitor ที่แสดงการทำงานของเซนเซอร์ตรวจจับควันและเซนเซอร์ตรวจจับเปลวไฟ เมื่อทำการตรวจจับไม่พบควันไฟและเปลวไฟในพื้นที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปริญญานิพนธ์นี้มีเป้าหมายเพื่อนำเทคโนโลยีต่าง ๆ มาประยุกต์ใช้กับระบบแจ้งเตือนอัคคีภัย เพื่อที่จะมีการป้องกันอันตรายหรือความเสียหายที่จะเกิดขึ้นกับบุคคลและทรัพย์สิน โดยอาศัยการทำงานของอุปกรณ์อิเล็กทรอนิกส์ และไมโครคอนโทรลเลอร์มาใช้ภายในระบบ โดยมีการใช้เซนเซอร์ตรวจจับเปลวไฟ เซนเซอร์ตรวจจับควัน และเซนเซอร์อุณหภูมิทำงานร่วมกับการประมวณผลด้วยบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 และบอร์ดไมโครคอนโทรลเลอร์ NodeMCU ESP8266 V2 เพื่อให้สามารถแสดงค่าสถานะควัน เปลวไฟ และค่าความชื้นสัมพัทธ์และอุณหภูมิโดยเฉลี่ย และค่าสถานะการทำงานของสัญญาณเสียงเตือนไฟไหม้และการดับไฟเบื้องต้น ผ่านการบริการคลาวด์ของ Adafruit IO ที่แสดงผลแบบ Dashboard และเมื่อมีปริมาณค่าเปลวไฟ ควัน และอุณหภูมิที่เกินจากที่กำหนดจะมีการส่งสัญญาณเสียงเตือน และแจ้งเตือนข้อความไปยังสมาร์ทโฟนในรูปแบบอัตโนมัติ รวมทั้งสปริงเกอร์ดับเพลิงจะทำงานเพื่อดับไฟเบื้องต้นทรัพย์สินภายในอาคารหรือภายในห้องต่างๆ โดยอาศัยเทคโนโลยีเข้ามาช่วยให้การใช้งานง่ายขึ้น

ในภาคการศึกษาที่ 1 ผู้จัดทำได้ทำการออกแบบและจัดทำระบบแจ้งเตือนอัคคีภัยที่ประกอบไปด้วยการออกแบบและจัดทำชิ้นงาน ทดสอบการทำงานของอุปกรณ์ต่างๆ พร้อมบันทึกผลเบื้องต้นได้แก่ ทดสอบการทำงานของเซนเซอร์ตรวจจับควัน ทดสอบการทำงานของเซนเซอร์ตรวจจับเปลวไฟ ทดสอบการทำงานของเซนเซอร์วัดอุณหภูมิ (DHT22) ทดสอบการแสดงค่าอุณหภูมิและความชื้นบนหน้าจอ LCD ทดสอบการทำงานของ SIM800L GSM Module ทดสอบการทำงานของโมดูล Relay 2 Channel ทดสอบการส่งค่าของเซนเซอร์จากบอร์ด Arduino Uno R3 ไปยังบอร์ด NodeMCU ESP8266 V2 แบบ Serial และทดสอบระบบแจ้งเตือนอัคคีภัยในการสั่งงานให้มีการดับไฟเบื้องต้นและส่งข้อความ SMS แจ้งเตือนไฟไหม้ พบว่าอุปกรณ์ทั้งหมดสามารถทำงานร่วมกันได้อย่างสมบูรณ์ส่งผลให้ระบบแจ้งเตือนอัคคีภัยทำงานได้ตามเงื่อนไขที่กำหนด

โดยในภาคการศึกษาที่ 2 ผู้จัดทำได้ทำการออกแบบและจัดทำระบบแจ้งเตือนอัคคีภัยที่ประกอบไปด้วยการออกแบบและจัดทำชิ้นงาน ทดสอบการทำงานของอุปกรณ์ต่างๆ และการใช้บริการคลาวด์ของ Adafruit IO ในการแสดงผลแบบ Dashboard พร้อมบันทึกผลเบื้องต้นได้แก่ ทดสอบการทำงานของโมดูล Active Buzzer ทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงสถานะการทำงานของโมดูล Active Buzzer ในการส่งสัญญาณเสียงเตือนบน Dashboard ทดสอบการใช้บริการคลาวด์ของ Adafruit IO ในการส่งค่าเซนเซอร์ตรวจจับควัน (MQ-2) ส่งค่าเซนเซอร์ตรวจจับเปลวไฟ และส่งค่าความชื้นสัมพัทธ์และอุณหภูมิโดยเฉลี่ยแสดงผลพร้อมกันบนหน้า Dashboard เดียวกัน ทดสอบระบบแจ้งเตือนอัคคีภัยที่มีการใช้บริการคลาวด์ของ Adafruit IO ในการสั่งงานให้มีการส่งสัญญาณเสียงเตือนและส่งข้อความ SMS แจ้งเตือนไฟไหม้พร้อมกับมีการดับไฟเบื้องต้น โดยเมื่อนำส่วนที่ได้ออกแบบทั้งหมดในภาคการศึกษาที่ 2 ไปใช้งานร่วมกันกับที่ออกแบบใน ภาคการศึกษาที่ 1 พบว่าการออกแบบและทดสอบทั้งหมดสามารถทำงานร่วมกันได้อย่างสมบูรณ์ ส่งผลให้ระบบแจ้งเตือนอัคคีภัยทำงานได้ตามวัตถุประสงค์ที่กำหนด

5.2 ข้อเสนอแนะ

ระบบแจ้งเตือนอัคคีภัยมีข้อจำกัดในเรื่องของระยะการตรวจจับของเซนเซอร์ตรวจจับควันและเปลวไฟที่ตรวจจับในระยะที่จำกัด เพราะถ้ามีการตรวจจับในห้องที่มีขนาดใหญ่จะมีการระยะเวลาการตรวจจับที่นานกว่าห้องพักขนาดปกติ โดยการรับข้อมูลจากเซนเซอร์ตรวจจับควันและเซนเซอร์ตรวจจับเปลวไฟต้องมีความแม่นยำที่สูง เพราะจะส่งผลให้การแจ้งเตือนข้อความและการดับไฟเบื้องต้นมีการผิดพลาดขึ้นได้ วิธีแก้คือควรใช้เซนเซอร์ที่มีคุณภาพที่สูงขึ้นเพื่อให้สามารถตรวจจับควันและเปลวไฟได้แม่นยำมากขึ้น รวมถึงในเรื่องของแบตเตอรี่ที่มีข้อจำกัดเรื่องของความจุที่น้อย จึงมีระยะเวลาการใช้งานได้ไม่ต่อเนื่อง วิธีแก้คือควรเลือกแบตเตอรี่ที่มีความจุสูงขึ้นเพื่อการใช้งานได้ต่อเนื่องและยาวนานขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] ผศ. ดอนสัน ปงผาบ. “ไมโครคอนโทรลเลอร์ Arduino”. (2560). สืบค้นเมื่อ 4 กันยายน 2565. สืบค้นจาก : https://images-se-ed.com/ws/Storage/PDF/552284/014/552284-0140449PDF.pdf

[2] AllNewStep. NodeMCU ESP8266/ESP8285 Arduino #1 ESP8266 คือ. (2559). สืบค้นเมื่อ 9 กันยายน 2565. สืบค้นจาก: https://www.allnewstep.com/article/30-nodemcu-esp8266-esp8285-arduino -1-/.

[3] Ab.in.th. ESP8266 Nodemcu V1V2V3. (2562). สืบค้นเมื่อ 9 กันยายน 2565. สืบค้นจาก: https:// www.ab.in.th/article/28/esp8266-nodemcu-v1-v2-v3/.

[4] inex. NodeMCU V2.0. (2558). สืบค้นเมื่อ 9 กันยายน 2565. สืบค้นจาก: https://inexco.th home/product /nodemcu-v2/.

[5] LnwShop. NodeMCU V2 LUA based ESP8266-Development USB CP2102. (2558). สืบค้นเมื่อ 9 กันยายน 2565. สืบค้นจาก: https://www.arduino4.com/product/313/nodemcu-v2-lua-based-esp8266-12e-development-kit-usb-cp2102/.

[6] sim800l gsm/gprs-module. (2563). สืบค้นเมื่อ 12 กันยายน 2565. สืบค้นจาก : https:// www.ab.in.th/product/738/sim800l-gsm-gprs-module

[7] aei.th. SIM800L SMS data GSM GPRS 4 frequency. (2558). สืบค้น เมื่อ 12 กันยายน 2565. สืบค้นจาก: https://www.lazada.co.th/tag/arduino-gsm-gprs/.

[8] Fromfactory. เครื่องตรวจจับควัน เซนเซอร์ตรวจจับก๊าซ เมื่อมีไฟไหม้ มีเอาต์พุต MQ-2. (2561). สืบค้นเมื่อ 1 กันยายน 2565. สืบค้นจาก: https://blog.thaieasyelec.com/getting-started-gas-sensor/.

[9] SUPPORT THAIEASYELEC. Gas Sensor Getting Started. (2560). สืบค้นเมื่อ 1 กันยายน 2565.สืบค้นจาก: https://blog.thaieasyelec.com/getting-started-gas-sensor/.

[10] modela.rayong. เซนเซอร์ตรวจจับควันและแก๊ส MQ-2 Smoke Gas LPG Butane Hydrogen Sensor Module. (2552). สืบค้นเมื่อ 1 กันยายน 2565. สืบค้นจาก: https://shopee.co.th/AA-MQ-2-Smoke-Gas-LPG-Butane-Hydrogen-Sensor-Module-/.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[11] Pure Ltd. DHT22/AM230x/RHT0x Temperature and RH Sensor. (2563). สืบค้นเมื่อ 3 กันยายน 2565. สืบค้นจาก: https://www.espruino.com/DHT22.

[12] ทันพงษ์ ภู่รักษ์. การใช้งาน ARDUINO กับไอซีวัดอุณภูมิและโมดูลตรวจจับสัญญาณอินพุต. (2563). สืบค้นเมื่อ 3 กันยายน 2565. สืบค้นจาก: http://www.sbt.ac.th/new/sites/default/files/TNP_Unit_10.pdf

[13] Allnewstep. Infrared IR Flame Detector Sensor Module เซนเซอร์ตรวจจับเปลวไฟ. (2558). สืบค้นเมื่อ 4 กันยายน 2565. สืบค้นจาก : https://www.allnewstep.com/product/835/infrared-ir-flame-detector-sensor-module/.

[14] Aliexpress. ตรวจจับเปลวไฟ Sensor IR อินฟราเรด Receiver. (2556). สืบค้นเมื่อ 4 กันยายน 2565. สืบค้นจาก: https://th.aliexpress.com/item/32821922264.html

[15] Cybertice. Active Buzzer Module (2563). สืบค้นเมื่อ 1 กุมภาพันธ์ 2566. สืบค้นจาก: https://www.cybertice.com/product/405/active-buzzer-module-arduino-3-3-5v

[16] ทันพงษ์ ภู่รักษ์.“LCD Display” (2560). สืบค้นเมื่อ 15 กันยายน 2565. สืบค้นจาก: http://www.sbt.ac.th/new/sites/default/files/TNP_Unit_6.pdf

[17] LCD Display. (2563). สืบค้นเมื่อ 16 กันยายน 2565. สืบค้นจาก: https://ardush-op.ro/en/home/2 19-lcd-display-1602-i2c-adapter.htm

[18] Cybertice. Lithium battery 18650 (2564). สืบค้นเมื่อ 1 กุมภาพันธ์ 2566. สืบค้นจาก: https://www.cybertice.com/product/16305/ถ่านชาร์จ-lithium-battery-แบตเตอรี่ลิเธียม -18650-3-7v-1200mah-หัวแบน-1-ก้อน

[19] As99shop. 18650 Battery Shield V3 (2565). สืบค้นเมื่อ 1 กุมภาพันธ์ 2566. สืบค้นจาก: https://www.as99shop.com/product/2128/18650-battery-shield-v3-v8-1-2-ก้อน-จ่ายไฟ-arduino-esp8266-ไมโครคอนโทรเลอร์--5v-สายusb

[20] Arduino4com. Battery Shield V3 (2564). สืบค้นเมื่อ 1 กุมภาพันธ์ 2566. สืบค้นจาก: https://www.arduinocom/product/74 3 / 1 8 6 5 0 -battery-shield-v3 -แหล่งจ่ายไฟ-esp8266-arduino-raspberry-พร้อมสาย-usb

[21] Cybertice. Arduino Relay 2 ช่อง. (2564). สืบค้นเมื่อ 1 พฤศจิกายน 2565. สืบค้นจาก: https://www.cybertice.com/product/5/arduino-relay-2-channel-5v-relay-module-active-low-5v-2-channel-isolation-control-250v-10a

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[22] Allnewstep. Relay 5V 2 ช่อง Relay module. (2565). สืบค้นเมื่อ 1 พฤศจิกายน 2565. สืบค้นจาก: https://www.allnewstep.com/product/3/relay-5v-2-ช่อง-relay-module-10a-250v-active-low-สำหรับ-arduino

[23] Kasetfun. DC Pump 12V without Pressure Switch. (2564). สืบค้นเมื่อ 1 ตุลาคม 2565. สืบค้นจาก: https://www.kasetfun.com/product/ปั๊มพ่นหมอก-dc-12v-แรงดัน-8-บาร์-แบบไม่มี-pressure-switch-ปรับแรงดัน/

[24] Thaiwatersystem. ปั๊มน้ำ12โวลต์ ปั๊มพ่นยา DC12V GREEN-03 แรงดัน 8 บาร์. (2565). สืบค้นเมื่อ 1 ตุลาคม2565. สืบค้นจาก: https://www.thaiwatersystem.com/product/418

[25] Thaiwatersystem. หลักการทำงานของโซลินอยด์วาล์ว. (2557). สืบค้นเมื่อ 2 ตุลาคม 2565. สืบค้นจาก: https://www.thaiwatersystem.com/article/45/หลักการทำงานของโซลินอยด์วาล์ว-solinoidvalve

[26] Knsciindustry. solenoid-valve DC 12 V. (2562). สืบค้นเมื่อ 2 ตุลาคม 2565. สืบค้นจาก: https://knsciindustry.com/blog-solenoid-valve/

[27] Tapumishop. โซลินอยด์วาล์วทองเหลือง. (2560). สืบค้นเมื่อ 2 ตุลาคม 2565. สืบค้นจาก: http://www.tapumishop.com/product/115

[28] Kasetnovate. สปริงเกอร์แบบต่างๆ. (2560). สืบค้นเมื่อ 3 ตุลาคม 2565. สืบค้นจาก: https://www.kasetnovate.com/what-are-differences-between-sprinklers

[29] Watertimerreview. หัวพ่นมินิสปริงเกอร์. (2564). สืบค้นเมื่อ 3 ตุลาคม 2565. สืบค้นจาก: https://www.watertimerreview.com/product/160/หัวพ่นมินิสปริงเกอร์-360-องศา-เกลียวนอก-1-2

[30] Hiachet. หัวพ่นมินิสปริงเกอร์ 360 องศา. (2564). สืบค้นเมื่อ 3 ตุลาคม 2565. สืบค้นจาก: https://www.hiachet.com/springer/

[31] Learn Adafruit. Adafruit IO (2562). สืบค้นเมื่อ 5 มกราคม 2565. สืบค้นจาก: https://learn.adafruit.com/welcome-to-adafruit-io/what-is-adafruit-io

[32] Techtalkthai.Dashboard. (2565). สืบค้นเมื่อ 5 มกราคม 2565. สืบค้นจาก: https://www.techtalkthai.com/aruba-iot-operations-introduction-by-it-green/

[33] Mostori. Message Queue Telemetry Transport. (2565). สืบค้นเมื่อ 5 มกราคม 2565. สืบค้นจาก: https://www.mostori.com/blog93.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

คำสั่งควบคุมการทำงานของเซนเซอร์ตรวจจับควัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include "AdafruitIO_WiFi.h"

int analogPin = A0; //ประกาศตัวแปร ให้ analogPin
int val = 0;

float analysis(int);
float Rs;

// ตั้งค่า key ของ Adafruit IO
#define IO_USERNAME "2422001w"
#define IO_KEY "aio_sfyn045IcdKYRiWiQWPltlKCryLW"

// ตั้งค่าชื่อ wifi
#define WIFI_SSID "JayR"
#define WIFI_PASS "sxpitchy"

AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS); // สร้างออปเจกชื่อ io สำหรับติดต่อ Adafruit IO
AdafruitIO_Feed *Value = io.feed("Value"); // สร้างออปเจกสำหรับติดต่อ feed ชื่อ Value
AdafruitIO_Feed *ppm = io.feed("Smoke(ppm)"); // สร้างออปเจกสำหรับติดต่อ feed ชื่อ Smoke(ppm)

void setup() {
  Serial.begin(115200);
  while (! Serial);
  Serial.print("Connecting to Adafruit IO");
  io.connect(); // เริ่มเชื่อมต่อ Adafruit IO

  while (io.status() < AIO_CONNECTED) { //เช็คว่าเชื่อมต่อสำเร็จ
    Serial.print(".");
    delay(500);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  }
  Serial.println();
  Serial.println(io.statusText()); //แสดงข้อความสถานะการเชื่อมต่อกับ Adafruit IO
}

void loop() {
  io.run(); // เริ่มทำงานร่วมกับ Adafruit IO
  int sensorValue = analogRead(analogPin);
  int val = analysis(sensorValue);

  Serial.print("val : ");
  Serial.print(sensorValue);
  Serial.print(" Smoke value : ");
  Serial.print(val);
  Serial.print("ppm\n");

   Value->save(sensorValue); // ส่งค่าไปที่ feed ชื่อ Value
  delay(2000);
   ppm->save(val); // ส่งค่าไปที่ feed ชื่อ Smoke(ppm)
  delay(15000);
}

  float analysis(int adc){
  float slope = -0.4523051706;
  float A = 41.12563414;
  float Rseries = 1000;
  float V_Rseries = ((float)adc*5)/1023;
  Rs = ((5-V_Rseries)/V_Rseries)*Rseries;
  float R0 = 129.52;
  /*
   * R0 คือ ค่าความต้านทานเซ็นเซอร์ขณะที่แก๊ส H2 เท่ากับ 1000 ppm
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   * ในอากาศที่สะอาด
   * ควรปรับจูนด้วยการสอบเทียบกับเซ็นเซอร์มาตรฐาน
   */
  float Y = Rs/R0;
  float Smoke = pow(10,(log10(Y/A)/slope));
  return Smoke;
}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

คำสั่งควบคุมการทำงานของเซนเซอร์ตรวจจับเปลวไฟและการส่งสัญญาณเสียงเตือนไฟไหม้ของโมดูล Active Buzzer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include "AdafruitIO_WiFi.h"

int analogPin = A0; //ประกาศตัวแปร ให้ analogPin

const unsigned char BUZZER_PIN = 2;


// ตั้งค่า key ของ Adafruit IO

#define IO_USERNAME "2422001w"

#define IO_KEY "aio_sfyn045IcdKYRiWiQWPltlKCryLW"


// ตั้งค่าชื่อ wifi

#define WIFI_SSID "JayR"

#define WIFI_PASS "sxpitchy"


AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS); // สร้างออปเจกชื่อ io สำหรับติดต่อ Adafruit IO

AdafruitIO_Feed *ppm = io.feed("Smoke(ppm)"); // สร้างออปเจกสำหรับติดต่อ feed ชื่อ Smoke(ppm)

AdafruitIO_Feed *IRFlame = io.feed("IRFlameValue"); // สร้างออปเจกสำหรับติดต่อ feed ชื่อ IRFlameValue

AdafruitIO_Feed *buzzer = io.feed("Buzzer"); // สร้างออปเจกสำหรับติดต่อ feed ชื่อ Buzzer


bool buzzer1 = false;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void setup() {
  Serial.begin(115200);
  while (! Serial);
  Serial.print("Connecting to Adafruit IO");
  io.connect(); // เริ่มเชื่อมต่อ Adafruit IO

  ppm->onMessage(handleMessage);//เมื่อมีข้อความ ให้ทำงานที่ฟังก์ชั่น handleMessage
  buzzer->onMessage(handleMessage1);//เมื่อมีข้อความ ให้ทำงานที่ฟังก์ชั่น handleMessage

  while (io.status() < AIO_CONNECTED) { //เช็คว่าเชื่อมต่อสำเร็จ
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.println(io.statusText()); //แสดงข้อความสถานะการเชื่อมต่อกับ Adafruit IO
  ppm->get(); // อ่านค่าจาก Feed ชื่อ Smoke(ppm)
  buzzer->get(); // อ่านค่าจาก Feed ชื่อ Buzzer
  pinMode(BUZZER_PIN, OUTPUT);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void loop() {
  io.run(); // เริ่มทำงานร่วมกับ Adafruit IO
}

void handleMessage(AdafruitIO_Data *data) {
  int reading = data->toInt(); // แปลงข้อความให้เป็นตัวเลข
  Serial.print(" Smoke val : ");
  Serial.println(reading);
  delay(5000);
  int IRFlameValue = analogRead(analogPin);
  IRFlame->save(IRFlameValue); // ส่งค่าไปที่ feed ชื่อ IRFlameValue

  if ((reading > 10) || (IRFlameValue < 600)) {
    Serial.print(" ตรวจพบเปลวไฟ\n");
    delay(1000);
    buzzer1 = true;;
    buzzer->save(buzzer1); // ส่งค่าไปที่ feed ชื่อ button
  }
  else{
    Serial.print(" ตรวจไม่พบเปลวไฟ\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    delay(1000);
    buzzer1 = false;
    buzzer->save(buzzer1); // ส่งค่าไปที่ feed ชื่อ button
  }
}

void handleMessage1(AdafruitIO_Data *data1) {//เมื่อมีข้อมูลจาก feed ชื่อ button จะทำในฟังก์ชั่นนี้
  if (data1->toPinLevel() == HIGH) { // ถ้าค่าเป็น 1
    digitalWrite(BUZZER_PIN, LOW);
    Serial.println("Buzzer ON");
  }
  else {
    Serial.println("Buzzer OFF");
    digitalWrite(BUZZER_PIN, HIGH);
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

คำสั่งควบคุมการทำงานของเซนเซอร์วัดอุณหภูมิและความชื้น (DHT22)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include "AdafruitIO_WiFi.h"

#include "DHT.h"

DHT dht;


#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);   //Module IIC/I2C Interface


// ตั้งค่า key ของ Adafruit IO

#define IO_USERNAME "2422001w"

#define IO_KEY "aio_sfyn045IcdKYRiWiQWPltlKCryLW"


// ตั้งค่าชื่อ wifi

#define WIFI_SSID "JayR"

#define WIFI_PASS "sxpitchy"


AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS); // สร้างออปเจกชื่อ io สำหรับติดต่อ Adafruit IO

AdafruitIO_Feed *temp = io.feed("temp"); // สร้างออปเจกสำหรับติดต่อ feed ชื่อ temp

AdafruitIO_Feed *hum = io.feed("humidity"); // สร้างออปเจกสำหรับติดต่อ feed ชื่อ humidity
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
AdafruitIO_Feed *IRFlame = io.feed("IRFlameValue"); // สร้างออปเจกสำหรับติดต่อ feed ชื่อ IRFlameValue


void setup() {
  Serial.begin(115200);
  dht.setup(2); // data pin 2
  lcd.begin();
  lcd.backlight();

  while (! Serial);
  Serial.print("Connecting to Adafruit IO");
  io.connect(); // เริ่มเชื่อมต่อ Adafruit IO

  IRFlame->onMessage(handleMessage);//เมื่อมีข้อความ ให้ทำงานที่ฟังก์ชั่น handleMessage

  while (io.status() < AIO_CONNECTED) { //เช็คว่าเชื่อมต่อสำเร็จ
    Serial.print(".");
    delay(500);
  }
  Serial.println();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Serial.println(io.statusText()); //แสดงข้อความสถานะการเชื่อมต่อกับ Adafruit IO
  IRFlame->get(); // อ่านค่าจาก Feed
}

void loop() {
  io.run(); // เริ่มทำงานร่วมกับ Adafruit IO
  float humidity = dht.getHumidity();
  float temperature = dht.getTemperature();

  lcd.home();
  lcd.setCursor(0, 0);
  lcd.print("Temp: ");
  lcd.print(temperature);
  lcd.print(" C");
  lcd.setCursor(0, 1);
  lcd.print("Humidity: ");
  lcd.print(humidity);
  lcd.print("%");
  delay(3000);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void handleMessage(AdafruitIO_Data *data) {

  int reading = data->toInt(); // แปลงข้อความให้เป็นตัวเลข

  delay(5000);

  float humidity = dht.getHumidity();

  float temperature = dht.getTemperature();



  hum->save(humidity); // ส่งค่าไปที่ feed ชื่อ humidity

  delay(2000);

  temp->save(temperature);// ส่งค่าไปที่ feed ชื่อ temp

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

คำสั่งควบคุมการทำงานของการดับไฟเบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include "AdafruitIO_WiFi.h"

#include <SoftwareSerial.h>


SoftwareSerial Com(D4,D3);


// ตั้งค่า key ของ Adafruit IO

#define IO_USERNAME "2422001w"

#define IO_KEY "aio_sfyn045IcdKYRiWiQWPltlKCryLW"


// ตั้งค่าชื่อ wifi

#define WIFI_SSID "JayR"

#define WIFI_PASS "sxpitchy"


AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS); // สร้างออปเจกชื่อ io สำหรับติดต่อ Adafruit IO

AdafruitIO_Feed *digital = io.feed("button"); // สร้างออปเจกสำหรับติดต่อ feed ชื่อ button

AdafruitIO_Feed *ppm = io.feed("Smoke(ppm)"); // สร้างออปเจกสำหรับติดต่อ feed ชื่อ Smoke(ppm)

AdafruitIO_Feed *IRFlame = io.feed("IRFlameValue"); // สร้างออปเจกสำหรับติดต่อ feed ชื่อ IRFlameValue
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int Relay1 = D0;

int Relay2 = D1;


bool current1 = false;

bool current2 = false;

bool current3 = false;


void setup() {
  Com.begin(115200);

  Serial.begin(115200);

  while (! Serial);

  Serial.print("Connecting to Adafruit IO");

  io.connect(); // เริ่มเชื่อมต่อ Adafruit IO


  ppm->onMessage(handleMessage);//เมื่อมีข้อความ ให้ทำงานที่ฟังก์ชั่น handleMessage

  IRFlame->onMessage(handleMessage1 );//เมื่อมีข้อความ ให้ทำงานที่ฟังก์ชั่น
handleMessage1


  while (io.status() < AIO_CONNECTED) { //เช็คว่าเชื่อมต่อสำเร็จ
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.println(io.statusText()); //แสดงข้อความสถานะการเชื่อมต่อกับ Adafruit IO

  ppm->get(); // อ่านค่าจาก Feed ชื่อ Smoke(ppm)
  IRFlame->get(); // อ่านค่าจาก Feed ชื่อ IRFlameValue

  pinMode(Relay1, OUTPUT);
  pinMode(Relay2, OUTPUT);

}

void loop() {
  io.run(); // เริ่มทำงานร่วมกับ Adafruit IO
  delay(8000);
    Serial.print("button1 : ");
    Serial.println(current1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Serial.print("button2 : ");

Serial.println(current2);


if ((current1 == 1) || (current2 == 1)){

  Serial.println("Relay ON");

  current3 = 1;

  digital->save(current3); // ส่งค่าไปที่ feed ชื่อ button

  Serial.print("button3 : ");

  Serial.println(current3);


   Com.write('#');

   Com.print(current3);

   Com.write(';');

}


if((current1 == 0) && (current2 == 0)){

digitalWrite(Relay1,LOW); // ส่งให้โซลินอยด์วาล์ว

digitalWrite(Relay2,LOW);

current3 = 0;

digital->save(current3); // ส่งค่าไปที่ feed ชื่อ button
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Serial.print("button3 : ");
    Serial.println(current3);
    }
}

void handleMessage(AdafruitIO_Data *data) {
  int reading = data->toInt(); // แปลงข้อความให้เป็นตัวเลข
  Serial.print(" Smoke val : ");
  Serial.println(reading);

   if(reading > 20){
   Serial.print(" ตรวจพบควันไฟไหม้\n");
   current1 = true;
  }
  else{
   Serial.print(" ตรวจไม่พบควันไฟไหม้\n");
   current1 = false;
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void handleMessage1(AdafruitIO_Data *data1) {
  int reading1 = data1->toInt(); // แปลงข้อความให้เป็นตัวเลข
  Serial.print("IRFlame : ");
  Serial.println(reading1);

  if (reading1 < 600 ) {
    Serial.print(" ตรวจพบเปลวไฟ\n");
    current2 = true;;
  }
  else{
    Serial.print(" ตรวจไม่พบเปลวไฟ\n");
    current2 = false;
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ

คำสั่งควบคุมการทำงานของการส่งข้อความ SMS แจ้งเตือนไฟไหม้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include "SIM900.h"
#include <SoftwareSerial.h>
SoftwareSerial Com(10,11); // 11 คือ RX, 10 คือ TX
#include "sms.h"
SMSGSM sms;


int numdata;
boolean started=false;
char smsbuffer[160];
char n[20];



void setup() {
  Serial.begin(115200);
       //Serial connection.
     Serial.println("GSM Shield testing.");
     //Start configuration of shield with baudrate.
     //For http uses is raccomanded to use 4800 or slower.
     if (gsm.begin(2400)) {
          Serial.println("\nstatus=READY");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            started=true;
        } else Serial.println("\nstatus=IDLE");
    Com.begin(115200);
}

void loop() {

    static int value1 = 0;
    static char SerialBuffer[4];
    static char SerialData;
    static byte SerialState = 0;
    static byte SerialIndex = 0;
    static double Timeout = millis();

    if(Com.available()){
      SerialData = Com.read();

      switch(SerialState){

        case 0: if(SerialData=='#'){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SerialState = 1;
        SerialIndex = 0;
      }
      break;

case 1: if(SerialData==';'){
        SerialBuffer[SerialIndex] = '\0';
        value1 = atoi(SerialBuffer);

        Serial.print("\n");
        Serial.print("value1 = ");
        Serial.print(value1);
        Serial.print("\n");

        Timeout = millis();
        SerialState = 0;
      }else{
        SerialBuffer[SerialIndex] = SerialData;
        if(++SerialIndex==4){
          SerialState = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
          }
          break;
    }
}
        if(value1 == 1){
          Serial.print("\n");
          Serial.print("ตรวจพบไฟไหม้");
          Serial.print("\n");
          Serial.print("Buzzer on! Relay on!");
          Serial.print("\n");
        Serial.print("Calibration is done\n");
        if(started) {
       //Enable this two lines if you want to send an SMS.
       if (sms.SendSMS("0970253741", "Fire!! Fire!! "))
       Serial.println("\nSMS sent OK");
    }
      if(started) {
       //Read if there are messages on SIM card and print them.
       if(gsm.readSMS(smsbuffer, 160, n, 20)) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Serial.println(n);
            Serial.println(smsbuffer);
        }
        }
}
        value1 = 0;
        Serial.print("\n");
        delay(5000);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้