

เครื่องอบฆ่าเชื้อด้วยรังสียูวีซีพร้อมด้วยระบบอบลมร้อนขนาดเล็ก

Mini UVC with hot air Sterilizer

โดย

ณัชชา แก้วโสภานิมิต 63010290

นันทน์ภัส อ่อนสนิท 63010514

อาจารย์ที่ปรึกษา

ผศ.ดร.แสงระวี บัวแก้ว

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2566

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2566

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องอบฆ่าเชื้อด้วยรังสียูวีซีพร้อมด้วยระบบอบลมร้อนขนาดเล็ก

Mini UVC with hot air Sterilizer

ผู้จัดทำ นางสาวณัชชา แก้วโสภานิมิต รหัสนักศึกษา 63010290

นางสาวนันทน์ภัส อ่อนสนิท รหัสนักศึกษา 63010514

ปริญญานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว



(ผศ.ดร. แสงระวี บัวแก้ว)

อาจารย์ที่ปรึกษา

หัวข้อโครงการ	เครื่องอบฆ่าเชื้อด้วยรังสียูวีซีพร้อมด้วยระบบอบลมร้อนขนาดเล็ก
นักศึกษา	นางสาวณัชชา แก้วโสภานิมิต 63010290 นางสาวนันทน์ภัส อ่อนสนิท 63010514
ปริญญา	วิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา	2566
อาจารย์ที่ปรึกษาโครงการ	ผศ.ดร. แสงระวี บัวแก้ว

บทคัดย่อ

โครงการฉบับนี้จัดทำขึ้นเพื่อศึกษาประสิทธิภาพในการฆ่าเชื้อของรังสี UVC และระบบอบลมร้อน สำหรับใช้กับอุปกรณ์ทางครัวเรือนและอุปกรณ์ที่ใช้ทั่วไปในชีวิตประจำวัน และศึกษาการทำงานของวงจรอิเล็กทรอนิกส์ โดยใช้ Arduino Nano ในการควบคุมการทำงานของวงจร ซึ่งโครงการนี้จะแบ่งออกเป็น 2 ส่วน คือ การเขียนโปรแกรม เพื่อให้วงจรสามารถใช้งานและควบคุมการทำงานของหลอดไฟ UVC และระบบอบลมร้อนได้ โดยการใช้โปรแกรม ArduinoIDE ในการเขียนคำสั่ง และการออกแบบวงจรทางด้าน Hardware ได้แก่ การออกแบบบอร์ดควบคุมการทำงานของเครื่องอบฆ่าเชื้อและระบบอบลมร้อน

Project Title Mini UVC with hot air Sterilizer

Student Miss Nutchka Kaewsophonimit Student ID 63010290
Miss Nannapat Onsanit Student ID 63010514

Degree Bachelor of Engineering

Program Electronics Engineering

Year 2023

Project Advisor Assist. Prof Sangrawee Buakaew , D.Eng.

ABSTRACT

This project was designed to study the sterilizing efficiency of UVC rays and hot air sterilizer for use with household equipment and commonly used in daily life. This project will study the operation of electronic circuits by using Arduino Nano to control the operation of the circuit. This project has 2 parts: First part is programming that able to control the operation of the sterilizer by using Coding Program's Arduino IDE. The Other part is hardware circuit design, make circuit to control the operation of the UVC light and hot air sterilizer.

กิตติกรรมประกาศ

โครงการ เครื่องอบฆ่าเชื้อด้วยรังสียูวีซีพร้อมด้วยระบบอบลมร้อนขนาดเล็ก สำเร็จได้ด้วยดี ด้วยความช่วยเหลือของ ผศ.ดร.แสงระวี บัวแก้ว อาจารย์ที่ปรึกษาโครงการ ที่คอยให้คำแนะนำและคำปรึกษาในการทำโครงการ คอยบอกข้อผิดพลาดและช่วยแก้ไขโครงการ รวมถึงอำนวยความสะดวกในการทำโครงการ ขอขอบคุณเพื่อนๆที่คอยให้ความช่วยเหลือและเป็นกำลังใจให้ผู้จัดทำสามารถทำโครงการนี้สำเร็จไปได้ด้วยดี ผู้จัดทำหวังว่าโครงการเล่มนี้จะเป็นประโยชน์แก่ผู้อ่านไม่มากนักน้อย หากผิดพลาดประการใด ต้องขออภัยไว้ ณ ที่นี้ด้วย

นางสาวณัชชา แก้วโสภานิมิต

นางสาวนันทน์ภัส อ่อนสนิท

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ	
1.1 ความเป็นมาและความสำคัญของโครงการ.....	1
1.2 จุดมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 ขอบเขตในการจัดทำโครงการ.....	1
1.4 ระยะเวลาในการทำโครงการ.....	2
บทที่ 2 หลักการและทฤษฎีอุปกรณ์	
2.1 ริงสี่ยูวีซี (UVC).....	4
2.2 พัดลมมอเตอร์.....	5
2.3 Arduino IDE.....	5
2.4 Wokwi.....	7
2.5 Arduino Nano.....	8
2.6 Relay.....	11
2.7 จอ LCD.....	12
2.8 Rotary encoder.....	14

สารบัญ (ต่อ)

เรื่อง	หน้า
2.9 XH-W1209 (Temperature controller).....	16
2.10 Heater.....	18
2.11 Switching power supply.....	19
บทที่ 3 หลักการทำงานและการออกแบบ	
3.1 ลักษณะการทำงานของวงจร.....	21
3.2 การออกแบบวงจร.....	21
3.2.1 ส่วนของโปรแกรม.....	21
1) การเขียนโปรแกรม.....	21
2) Flowchart การเขียนโปรแกรม.....	43
3.2.2 การออกแบบวงจร และอุปกรณ์ที่ใช้.....	49
3.2.3 ขั้นตอนการทำงาน.....	54
3.2.4 การออกแบบตู้ต่อขั้ว.....	54
บทที่ 4 ผลการทดลอง	
4.1 การทำงานของโปรแกรม.....	56
4.2 การทดสอบความสามารถในการฆ่าเชื้อของยูวีซี.....	59
4.3 การทดสอบความสามารถของตัวควบคุมอุณหภูมิ.....	61
บทที่ 5 สรุปผลการทดลอง.....	63
อ้างอิง.....	65
ภาคผนวก.....	ก

สารบัญตาราง

เรื่อง

หน้า

ตารางที่ 1.1 ระยะเวลาในการทำโครงการ..... 2 - 3



สารบัญรูป

เรื่อง	หน้า
รูปที่ 2.1 แสดงแถบสีของรังสียูวีซี (UVC).....	4
รูปที่ 2.2 แสดงภาพหลอดไฟรังสียูวีซีที่ใช้ในชีวิตประจำวัน (UVC).....	4
รูปที่ 2.3 พัดลมมอเตอร์.....	5
รูปที่ 2.4 หน้าเว็บออนไลน์ของ Wokwi.....	7
รูปที่ 2.5 ตัวอย่างโปรเจ็คของ Wokwi.....	8
รูปที่ 2.6 ขาการทำงานของ Arduino Nano.....	9
รูปที่ 2.7 Arduino Nano.....	10
รูปที่ 2.8 แสดงส่วนประกอบของรีเลย์.....	11
รูปที่ 2.9 จอ LCD.....	12
รูปที่ 2.10 LCD(I2C).....	13
รูปที่ 2.11 rotary encoder.....	15
รูปที่ 2.12 การอ่านค่า rotary encoder.....	15
รูปที่ 2.13 XH-W1209 (temperature controller).....	16
รูปที่ 2.14 ตัวอย่างการทำงาน temperature controller.....	17
รูปที่ 2.15 heater.....	19
รูปที่ 2.16 โครงสร้างการทำงานของสวิตซ์เพาเวอร์ซัพพลาย (Switching Power Supply).....	18
รูปที่ 2.17 สวิตซ์เพาเวอร์ซัพพลาย (Switching Power Supply).....	20
รูปที่ 3.1 การจ่ายไฟด้วย Switching power supply ให้อุปกรณ์ต่างๆ.....	52
รูปที่ 3.2 การต่อสายไฟภายในตัวกล่อง.....	53
รูปที่ 3.3 แบบตู้อบฆ่าเชื้อ.....	54

สารบัญรูป (ต่อ)

เรื่อง	หน้า
รูปที่ 4.1 รูปทดลองต่อวงจรด้วย LED.....	56
รูปที่ 4.2 ตู้บข่าเชื้อที่ใช้ในการทดลองจริง.....	56
รูปที่ 4.3 – 4.12 การแสดงผลผ่านจอ lcd.....	57 - 59
รูปที่ 4.13 - 4.15 การแสดงผลของตัวควบคุมอุณหภูมิ.....	61 - 62



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของโครงการ

ในปัจจุบันเกิดการแพร่ระบาดของเชื้อโรคมามากมาย เช่น โควิด-19 เป็นต้น ซึ่งปัจจัยสำคัญส่วนหนึ่งที่ทำให้เราติดเชื้อโรค มาจากการสัมผัสสิ่งของเครื่องใช้ต่างๆ ที่มีเชื้อโรค และของใช้ที่มักจะเกิดเชื้อโรคได้บ่อยที่สุด คือ ของใช้ในชีวิตประจำวัน เช่น ภาชนะใส่อาหารและเครื่องดื่มนั้นๆ ดังนั้นเพื่อให้มั่นใจว่าภาชนะเหล่านั้นสะอาดปราศจากเชื้อโรค ทางผู้จัดทำจึงสร้างตู้ฆ่าเชื้อด้วยรังสียูวีซีขึ้น เนื่องจากรังสียูวีซีมีคุณสมบัติในการกำจัดเชื้อโรค แต่รังสียูวีซีก็มีข้อจำกัด คือ มีความอันตรายต่อผิวหนัง และดวงตาของมนุษย์ ทำให้ต้องใช้ในพื้นที่ปิดเท่านั้น เพื่อควบคุมข้อจำกัดเหล่านี้ ทางผู้จัดทำจึงนำความรู้ทางด้านอิเล็กทรอนิกส์เข้ามาช่วย อีกทั้งยังพัฒนาตู้อบฆ่าเชื้อนี้ด้วยการติดตั้งระบบอบลมร้อน เพื่อให้เหมาะแก่การใช้งานกับภาชนะใส่อาหารและเครื่องดื่มมากยิ่งขึ้น

1.2 จุดมุ่งหมายและวัตถุประสงค์ของการศึกษา

- 1.2.2 เพื่อสร้างเครื่องอบฆ่าเชื้อด้วยรังสียูวีซีพร้อมด้วยระบบอบลมร้อนขนาดเล็กให้สามารถใช้งานได้จริง
- 1.2.2 เพื่อฝึกการใช้งาน Arduino IDE ในการเขียนโปรแกรม
- 1.2.3 เพื่อฝึกการออกแบบอุปกรณ์ให้สามารถใช้งานได้จริง
- 1.2.4 เพื่อฝึกการคิดวิเคราะห์และสามารถแก้ไขปัญหาต่างๆที่เกิดขึ้นได้

1.3 ขอบเขตในการจัดทำโครงการ

- 1.3.1 อุปกรณ์ต่างๆที่ออกแบบต้องสามารถทำงานร่วมกับโปรแกรมที่เขียนขึ้นได้
- 1.3.2 ตู้อบฆ่าเชื้อสามารถด้วยรังสียูวีซีได้
- 1.3.3 เมื่อกดสวิตช์ จะต้องสามารถเปิด/ปิดการทำงานของหลอดไฟและระบบอบลมร้อนได้
- 1.3.4 เมื่อกดสวิตช์ หน้าจอ LCD จะต้องแสดงสถานการณ์ทำงานของตู้อบฆ่าเชื้อ
- 1.3.5 วงจรสามารถควบคุมอุณหภูมิไม่ให้อุณหภูมิสูงหรือต่ำกว่าอุณหภูมิที่ตั้งไว้ได้

1.4 ระยะเวลาในการทำโครงการ

ตั้งแต่วันที่ 1 ธันวาคม 2566 ถึงวันที่ 5 มีนาคม 2567

รายละเอียด	ธันวาคม				มกราคม			
	สัปดาห์ที่ 1	สัปดาห์ที่ 2	สัปดาห์ที่ 3	สัปดาห์ที่ 4	สัปดาห์ที่ 1	สัปดาห์ที่ 2	สัปดาห์ที่ 3	สัปดาห์ที่ 4
1. ได้รับ โครงสร้างของ โครงการ								
2. ศึกษาข้อมูล เกี่ยวกับหัวข้อ โครงการ								
3. เขียนโค้ดบน เว็บ Wokwi								
4. ทดสอบโค้ด บนเว็บ Wokwi								
รายละเอียด	กุมภาพันธ์				มีนาคม			
	สัปดาห์ที่ 1	สัปดาห์ที่ 2	สัปดาห์ที่ 3	สัปดาห์ที่ 4	สัปดาห์ที่ 1	สัปดาห์ที่ 2	สัปดาห์ที่ 3	สัปดาห์ที่ 4
5. ซื่อและ จัดเตรียม อุปกรณ์								
6. ใส่อุปกรณ์ เพิ่มในวงจร								

รายละเอียด	กุมภาพันธ์				มีนาคม			
	สัปดาห์ที่ 1	สัปดาห์ที่ 2	สัปดาห์ที่ 3	สัปดาห์ที่ 4	สัปดาห์ที่ 1	สัปดาห์ที่ 2	สัปดาห์ที่ 3	สัปดาห์ที่ 4
7. ทดสอบวงจร			←→					
8. วิเคราะห์ และแก้ไข ปัญหา				←→				
9 สรุปผล พร้อมจัดทำ รูปเล่มโครงการ		←→						

ตารางที่ 1.1 ระยะเวลาในการทำโครงการ

บทที่ 2

หลักการและทฤษฎีอุปกรณ์

2.1 รังสียูวีซี (UVC)

เป็นรังสีอัลตราไวโอเล็ตชนิดหนึ่ง มีความยาวคลื่น 200-280 nm ในธรรมชาติรังสี UVC ถูกกรองไว้โดยชั้นโอโซน จึงไม่สามารถผ่านชั้นบรรยากาศของโลกลงมาได้ เนื่องจากรังสี UVC ถูกดูดซับโดยโมเลกุลอินทรีย์ รวมถึง DNA ทำให้เกิดการตายของจุลินทรีย์ จึงได้กลายเป็นที่นิยมสำหรับการใช้งานฆ่าเชื้อโรค โดยในปัจจุบันมีการนำหลอด UVC ที่มีความยาวคลื่น 254 nm มาใช้ในการฆ่าเชื้อโรค



รูปที่ 2.1 แสดงแถบสีของรังสียูวีซี (UVC)

ที่มา <https://www.shinelongled.com/uvc-led-light/>



รูปที่ 2.2 แสดงภาพหลอดไฟรังสียูวีซีที่ใช้ในชีวิตประจำวัน (UVC)

ที่มา <https://www.ebay.co.uk/itm/38W-UVC-Ultraviolet-UV-Ozone-Disinfection-Light-Germicidal-Sterilization-Lamp-US/>

2.2 พัดลมมอเตอร์

พัดลมมอเตอร์ (Fan Motor) เป็นอุปกรณ์ที่ทำหน้าที่ระบายความร้อนจากภายในของอุปกรณ์ โดยการหมุนของใบพัดจะทำให้เกิดความร้อนภายในตัวอุปกรณ์และมอเตอร์ต่างๆ รวมไปถึงเครื่องจักร โดยมีการควบคุมและแลกเปลี่ยนความร้อนจากภายนอก พัดลมมอเตอร์ยังสามารถป้องกันฝุ่นได้ดี ทำให้อายุการใช้งานของมอเตอร์ยาวนานมากขึ้น โดยพัดลมมอเตอร์เหมาะสำหรับงานที่มีพื้นที่ระบายความร้อนไม่มาก เช่น เครื่องเซิร์ฟเวอร์ เครื่องคอมพิวเตอร์ กล้องวงจรปิด ตู้ควบคุมสำหรับเครื่องจักร ตู้ควบคุมระบบสื่อสาร เป็นต้น



รูปที่ 2.3 พัดลมมอเตอร์

ที่มา AFB1212ME - delta fans - Authorized Distributor (onlinecomponents.com)

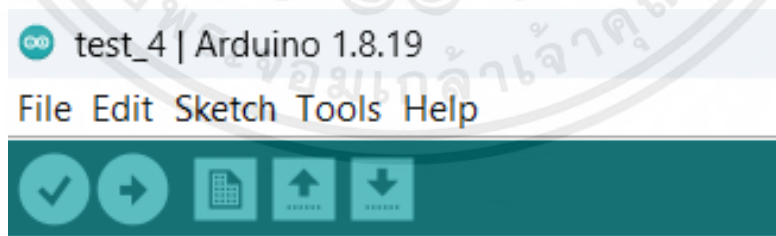
2.3 Arduino IDE

เป็นโปรแกรมที่ “แจกฟรี” ในการใช้งานลักษณะ Open source ซึ่ง Arduino IDE จะทำหน้าที่ ติดต่อระหว่างคอมพิวเตอร์ ไม่ว่าจะเป็นระบบ Windows, Mac OS X หรือ Linux กับ บอร์ด Arduino ซึ่งโปรแกรมนี้ ออกแบบให้ง่ายต่อการเขียนโค้ดและอัปโหลดโปรแกรมที่เราเขียนเข้าสู่บอร์ด Arduino ส่วน IDE ย่อมาจาก (Integrated Development Environment) คือ ส่วนเสริมของระบบการพัฒนา หรือตัวช่วยต่าง ๆ ที่จะคอยช่วยเหลือ Developer หรือช่วยเหลือคนที่พัฒนา Application เพื่อเสริมให้เกิดความรวดเร็ว ถูกต้อง แม่นยำ ตรวจสอบระบบที่จัดทำได้ ทำให้การพัฒนางานต่าง ๆ เร็วมากขึ้น ส่วนในการเขียนโปรแกรมและคอมไพล์ลงบอร์ด โดยขนาดของโปรแกรม Arduino โดยปกติแล้วจะใหญ่กว่าโค้ด AVR ปกติเนื่องจากโค้ด AVR เป็นการเข้าถึงจากรีจิสเตอร์โดยตรง แต่โค้ด Arduino เข้าถึงผ่านฟังก์ชัน เพื่อให้สามารถเขียนโค้ดได้ง่ายมากกว่าการเขียนโค้ดแบบ AVR หรือเวอร์ชันอื่นๆ ของ Arduino วิธีการการใช้งานโปรแกรม

1. เรียกใช้โปรแกรมด้วยการดับเบิลคลิกที่ไอคอน Arduino.exe ในตำแหน่งที่ติดตั้งโปรแกรมไว้
2. ตั้งค่าบอร์ดให้ตรงกับที่ใช้งานซึ่งมีการตั้งค่าอยู่ด้วยกัน 4 รายการดังนี้
 - 1) ชนิดของบอร์ด Arduino ที่ใช้งาน (กำลังเชื่อมต่ออยู่)
 - 2) ชนิดของโปรเซสเซอร์ เลือกให้ตรงกับโปรเซสเซอร์ของบอร์ด (บอร์ดบางชนิดไม่ต้องเลือกเนื่องจากมีใช้งานเพียงเบอร์เดียว)
 - 3) คอมพอร์ตที่กำลังเชื่อมต่อ เลือกให้ตรงกับที่บอร์ด Arduino กำลังเชื่อมต่อ (ดูจาก Device manager)
 - 4) ชนิดของเครื่องโปรแกรม เลือก AVRISP mkII



3. เขียนโปรแกรมที่ต้องการติดตั้งลงในบอร์ด จากนั้นกดปุ่ม verify ที่เป็นเครื่องหมายถูกด้านซ้ายของโปรแกรมเพื่อตรวจสอบความถูกต้อง จากนั้นเลือก run ที่เป็นลูกศรชี้ไปทางขวาเพื่อนำข้อมูลไปใส่บอร์ดที่เราต้องการ

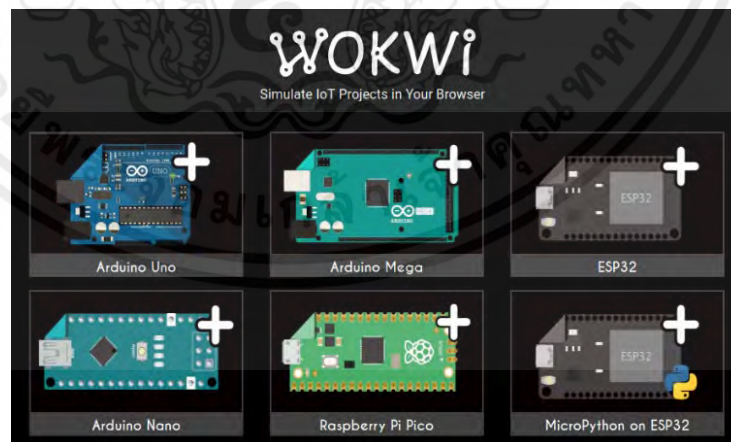


2.4 Wokwi

เป็นซอฟต์แวร์ที่ทำงานบนหน้าเว็บออนไลน์ สามารถเขียนโปรแกรมและจำลอง (Simulator) การทำงานของบอร์ด Arduino, Raspberry Pi Pico และ ESP32 หรือบอร์ดไมโครคอนโทรลเลอร์ ที่ออกแบบมาเพื่อเรียนรู้การเขียนโปรแกรมโดยไม่ต้องใช้ฮาร์ดแวร์จริง สามารถใช้งานได้บนหน้าเว็บเบราว์เซอร์ จึงไม่จำเป็นต้องติดตั้งโปรแกรมใด ๆ

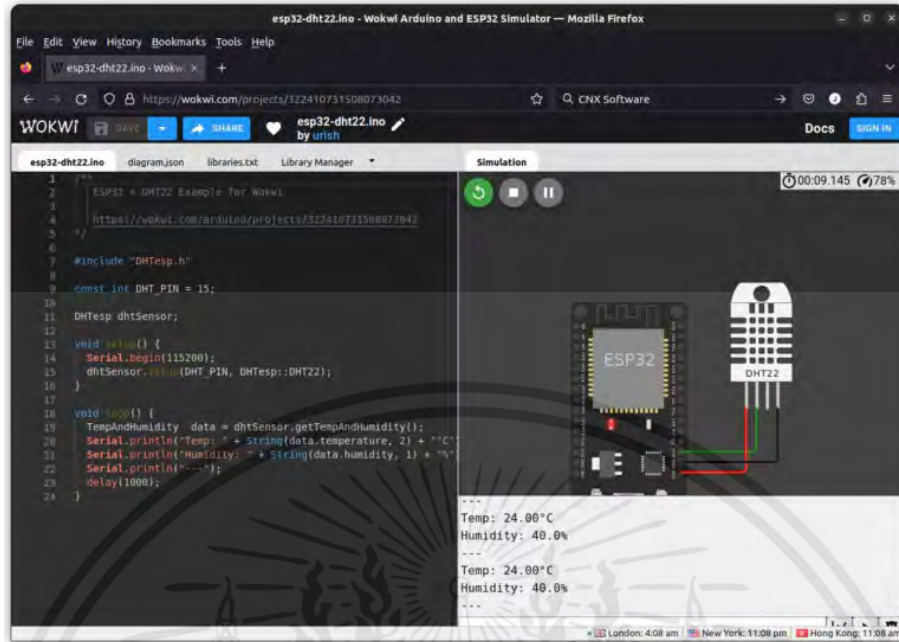
ซอฟต์แวร์ที่เป็นองค์ประกอบสำคัญของโปรเจกต์ Wokwi ได้แก่

- AVR8js JavaScript library ใช้ภาษา TypeScript / JavaScript สำหรับการพัฒนาตัวจำลองการทำงานของชิปไมโครคอนโทรลเลอร์ AVR หรือ เรียกว่า JavaScript-based AVR Simulator ไลบรารีดังกล่าวซึ่งเป็น Open Source ถือว่าเป็นส่วนที่สำคัญสำหรับการทำงานของซอฟต์แวร์ Wokwi Arduino Simulator
- Wokwi Elements เป็น Graphical Web Components สำหรับอุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ ที่สามารถนำมาสร้างวงจรเสมือนจริง (Virtual Circuit Prototyping) และจำลองการทำงานร่วมกันได้ (Interactive Simulation) เช่น บอร์ด Arduino หลอดแอลอีดี (LED) ปุ่มกด (Button) เป็นต้น
 - ดูรายการอุปกรณ์ที่ใช้ได้กับ Wokwi Simulator ได้จากเว็บ <https://elements.wokwi.com/>
- นอกจากซอฟต์แวร์ที่ทำหน้าที่จำลองการทำงานของชิพตามชุดคำสั่ง AVR หรือ AVR Simulator ยังมีตัวจำลองการทำงานสำหรับตัวประมวลผลอื่นอีก ได้แก่
 - RP2040 Simulator ที่มีชื่อว่า rp2040js เป็นซอฟต์แวร์ Open Source และรองรับการรันคำสั่งโดยใช้เฟิร์มแวร์ของ MicroPython หรือ Circuit Python ได้
 - ESP32 Simulator สำหรับชุดคำสั่งตามสถาปัตยกรรมของชิพที่มีชื่อว่า Xtensa และ RISC-V



รูปที่ 2.4 หน้าเว็บออนไลน์ของ Wokwi

ที่มา Wokwi : จำลองการทำงานออนไลน์ของ ESP32, Arduino, และ Raspberry Pi Pico - CNX Software (cnx-software.com)



รูปที่ 2.5 ตัวอย่างโปรเจ็คของ Wokwi

ที่มา Wokwi : จำลองการทำงานออนไลน์ของ ESP32, Arduino, และ Raspberry Pi Pico - CNX Software (cnx-software.com)

2.5 Arduino NANO

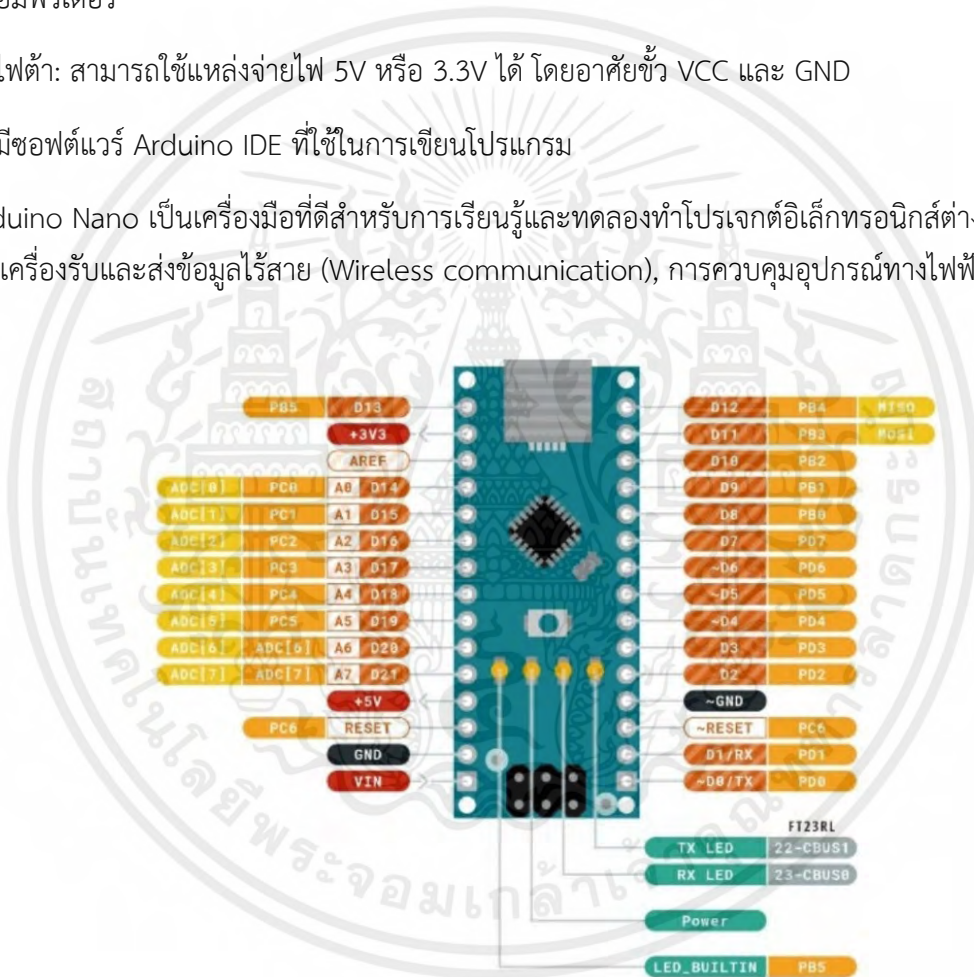
Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open source คือ มีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย จึงเหมาะสำหรับผู้เริ่มต้นศึกษา และผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดตัวบอร์ด หรือโปรแกรมต่อได้อีก ด้วย ความง่ายของบอร์ด Arduino ในการต่อกับอุปกรณ์เสริมต่างๆ คือ ผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอก แล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด เพื่อความสะดวก สามารถเลือกต่อบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Sheild, Arduino GPRS Shield เป็นต้น แล้วมาเสียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อ

Arduino Nano เป็นบอร์ดควบคุมไมโครคอนโทรลเลอร์ที่พัฒนาโดย Arduino ซึ่งเป็นแพลตฟอร์มโอเพนซอร์สสำหรับการพัฒนาโปรเจกต์อิเล็กทรอนิกส์และโปรแกรมมิ่งต่าง ๆ โดยเฉพาะอย่างยิ่งสำหรับผู้เริ่มต้นในการพัฒนาตัวเอง ซึ่งมีความสามารถที่จะควบคุมอุปกรณ์และระบบต่าง ๆ ได้อย่างยืดหยุ่น

Arduino Nano มีคุณสมบัติต่าง ๆ ดังนี้ :

1. ขนาดเล็ก: มีขนาดเล็กและเหมาะสำหรับโปรเจกต์ที่ต้องการพื้นที่น้อย
2. ใช้ไมโครคอนโทรลเลอร์ ATmega328P (ATmega168 เวอร์ชันเก่า) ที่มีความเร็ว 16 MHz
3. ขาเข้าและขาออก: มีขาดีจิตอลและขานาล็อกในการเชื่อมต่ออุปกรณ์ภายนอก.
4. USB ในการโปรแกรมและการเชื่อมต่อ: มีพอร์ต USB ที่ใช้ในการโปรแกรม Arduino Nano และสื่อสารกับคอมพิวเตอร์
5. ไฟต้า: สามารถใช้แหล่งจ่ายไฟ 5V หรือ 3.3V ได้ โดยอาศัยขั้ว VCC และ GND
6. มีซอฟต์แวร์ Arduino IDE ที่ใช้ในการเขียนโปรแกรม

Arduino Nano เป็นเครื่องมือที่ดีสำหรับการเรียนรู้และทดลองทำโปรเจกต์อิเล็กทรอนิกส์ต่าง ๆ เช่นตัวรถหุ่นยนต์, เครื่องรับและส่งข้อมูลไร้สาย (Wireless communication), การควบคุมอุปกรณ์ทางไฟฟ้า เป็นต้น



รูปที่ 2.6 ขาการทำงานของ Arduino Nano

ที่มา <https://th.mouser.com/new/seeed-studio/seeed-arduino-nano-rp2040-connect/>

อุปกรณ์ที่อยู่ด้านบนของบอร์ด Arduino Nano เรียงจากบนลงล่าง

- USB (micro) Port เชื่อมต่อกับ computer หรือแหล่งจ่ายไฟให้กับ board
- ATmega328

- Crystal ทำหน้าที่กำเนิดสัญญาณให้ microcontroller ทำงานได้ตามจังหวะของสัญญาณ
- Switch ทำหน้าที่ reset ตัว microcontroller
- LED แสดงสถานะการทำงาน เรียงจากซ้ายไปขวา
- L: เมื่อมีการใช้งานขา D13 ของ Board
- ON: เมื่อมีไฟจ่ายให้กับ microcontroller
- Rx: เมื่อมีการใช้งานขา Rx (D0) ของ Board ซึ่งเป็นขารับข้อมูลของ UART
- Tx: เมื่อมีการใช้งานขา Tx (D1) ของ Board ซึ่งเป็นขาส่งข้อมูลของ UART
- ICSP (In Circuit Serial Programmer) ทำหน้าที่เป็นจุดเชื่อมต่อกับเครื่อง Programmer สำหรับการโปรแกรม microcontroller (ในกรณีที่ไม่มีใช้ หรือไม่มี Bootloader)

อุปกรณ์ที่อยู่ด้านล่างของบอร์ด Arduino Nano เรียงจากบนลงล่าง

- USB to UART ทำหน้าที่เปลี่ยนสัญญาณกลับไปมาระหว่าง USB (ซึ่งเป็นสัญญาณมาจาก computer) และ UART (ซึ่งเป็นสัญญาณจาก microcontroller) ทำให้ computer และ microcontroller สามารถสื่อสารกันได้
- Voltage Regulator ทำหน้าที่แปลงไฟที่ส่งเข้ามาที่ขา Vin ของ board ให้เป็น 5V ก่อนที่จะส่งให้ microcontroller และอุปกรณ์อื่น ๆ

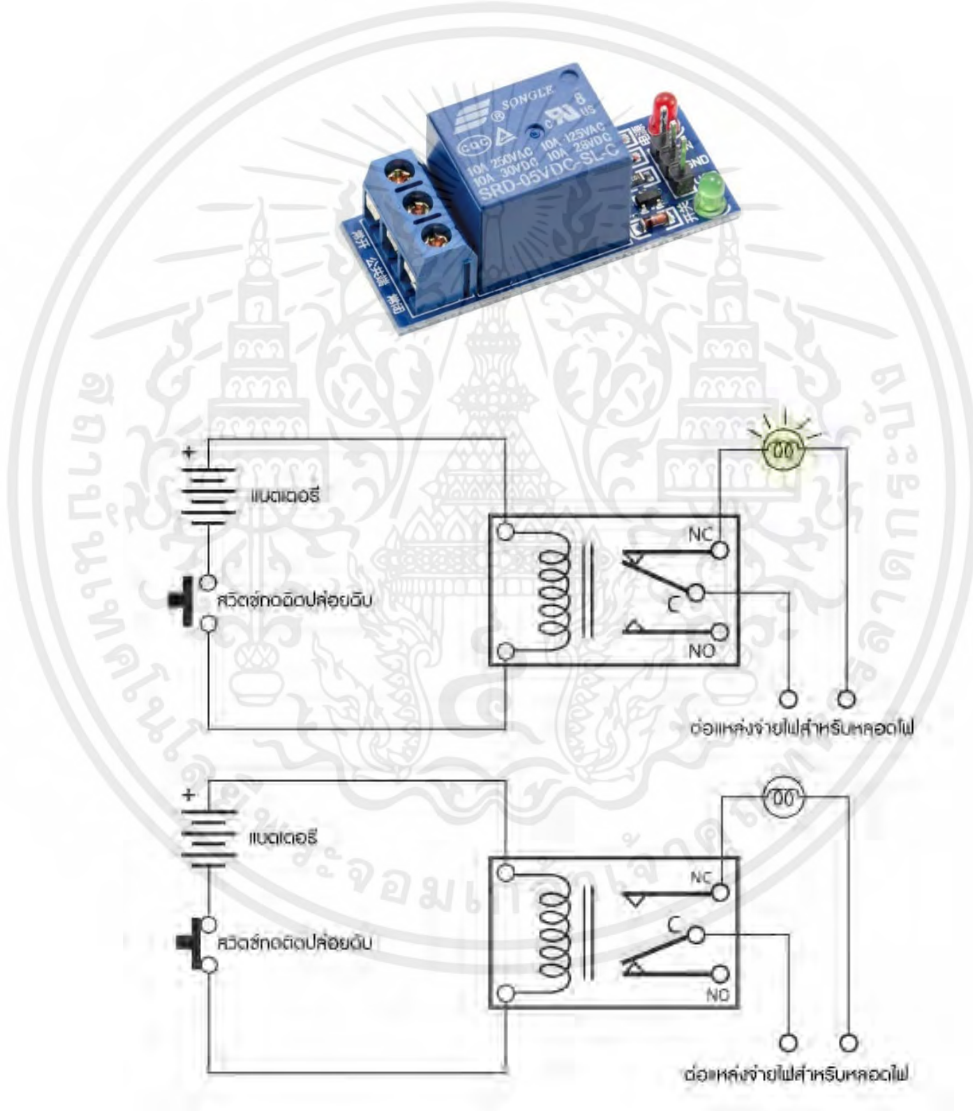


รูปที่ 2.7 Arduino Nano

ที่มา https://jota.uniq.com.au/electronics/arduino/arduino_nano.html

2.6 รีเลย์ (Relay)

คืออุปกรณ์ที่ทำหน้าที่ตัด-ต่อวงจร โดยการจ่ายไฟเข้าไปเพื่อให้มันทำงาน เปรียบเสมือนสวิตช์ไฟ ที่ใช้แรงดันไฟฟ้าในการเปิดและปิดอุปกรณ์ไฟฟ้าเพื่อควบคุมวงจรต่างๆ โดยภายในรีเลย์จะมีส่วนประกอบอยู่สองส่วน คือส่วนคอยล์ (Coil) และส่วนหน้าสัมผัส (Contact) ส่วนคอยล์มีหน้าที่รับไฟฟ้าที่ใช้ควบคุมและสร้างสนามแม่เหล็กตามสัญญาณที่เข้ามา ส่วนหน้าสัมผัสจะทำหน้าที่เป็นสวิตช์ตัด-ต่อวงจร จุดต่อ NC ย่อมาจาก normal close หมายความว่า ปกติปิด คือ หากยังไม่มีกระแสไฟไหลให้ขดลวด(coil) หน้าสัมผัสนี้จะเชื่อมต่อกับจุดต่อ C โดยทั่วไปแล้วเรามักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการให้ทำงานตลอดเวลา



รูปที่ 2.8 แสดงส่วนประกอบของรีเลย์

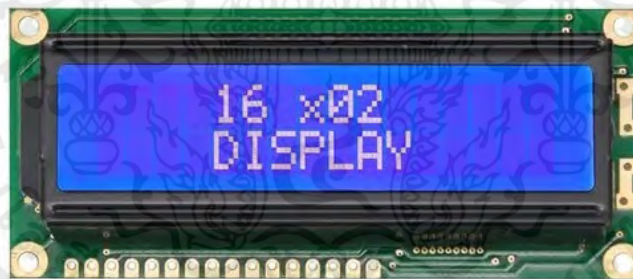
ที่มา <http://tortarakit.com/wp/blog/article/>

- จุดต่อใช้งานมาตรฐานในวงจร

- จุดต่อ NC ย่อมาจาก normal close หมายความว่า ปกติปิด คือ หากยังไม่มีกระแสไฟให้ขดลวด (coil) หน้าสัมผัสนี้จะเชื่อมต่อกับจุดต่อ C โดยทั่วไปแล้วเรามักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการให้ทำงานตลอดเวลา
- จุดต่อ NO ย่อมาจาก normal open หมายความว่า ปกติเปิด คือ หากยังไม่มีกระแสไฟให้ขดลวด (coil) หน้าสัมผัสจะยังไม่เชื่อมต่อกับจุดต่อ C โดยทั่วไปแล้วเรามักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการให้ทำงานในช่วงเวลาจำกัดเท่านั้น
- จุดต่อ C ย่อมาจาก common หมายถึง จุดร่วมที่ต่อมาจากแหล่งจ่ายไฟ

2.7 จอ LCD (LCD Display)

LCD ย่อมาจาก Liquid Crystal Display ซึ่งเป็นจอแสดงผลแบบ (Digital) โดยภาพที่ปรากฏเกิดจากแสงที่ถูกปล่อยออกมาจากหลอดไฟด้านหลังของจอภาพ (Black Light) ผ่านชั้นกรองแสง (Polarized filter) แล้ววิ่งไปยัง คริสตัลเหลวที่เรียงตัวกัน 3 เซลล์คือ แสงสีแดง เขียวและน้ำเงิน กลายเป็นพิกเซล (Pixel) ที่สว่างสดใสเกิดขึ้น



รูปที่ 2.9 จอ LCD

ที่มา <https://www.indiamart.com/proddetail/jhd-lcd-displays-yellowish-green-and-blue-color-19934678048.html>

การควบคุมการแสดงผลของจอ LCD (I2C)

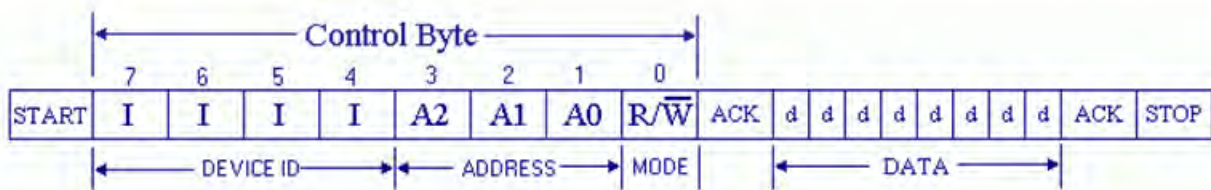


รูปที่ 2.10 LCD(I2C)

ในการควบคุมหรือสั่งงาน โดยทั่วไปจอ LCD จะมีส่วนควบคุม (Controller) อยู่ในตัวแล้ว ผู้ใช้สามารถส่งรหัสคำสั่งสำหรับควบคุมการทำงานของจอ LCD (I2C) เช่นเดียวกันกับจอ LCD แบบธรรมดา รหัสคำสั่งที่ใช้ในการควบคุมนั้นเหมือนกัน แต่ต่างกันตรงที่รูปแบบในการรับส่งข้อมูล ในบทความนี้เราจะมาพูดถึงจอ LCD 16x2 ที่มีการส่งข้อมูลรูปแบบ I2C ที่ใช้ขาเพียง 4 ขาที่ใช้ในการเชื่อมต่อเท่านั้น

1. GND เป็น Ground ใช้ต่อระหว่าง Ground ของระบบ Microcontroller กับ LCD
2. VCC เป็นไฟเลี้ยงวงจรที่ป้อนให้กับ LCD มีขนาด +5VDC
3. SDA (Serial Data) เป็นขาที่ใช้ในการรับส่งข้อมูล
4. SCL (Serial Clock) เป็นขาสัญญาณนาฬิกาในการรับส่งข้อมูล

การรับ-ส่งข้อมูลแบบ I2C BUS



MCU จะทำการส่งสถานะเริ่มต้น (START Conditions) เพื่อแสดงการขอใช้บัส แล้วตามด้วย รหัสควบคุม (Control Byte) ซึ่งประกอบ ด้วยรหัส ประจำตัวอุปกรณ์ Device ID, Device Address และ Mode ในการเขียน หรืออ่านข้อมูล เมื่ออุปกรณ์รับทราบว่า MCU ต้องการจะติดต่อด้วย ก็ต้องส่งสถานะรับรู้ (Acknowledge) หรือแจ้งให้ MCU รับรู้ว่าข้อมูลที่ได้ส่งมามีความถูกต้อง และเมื่อสิ้นสุดการส่งข้อมูล MCU จะต้องส่งสถานะสิ้นสุด (STOP Conditions) เพื่อบอกกับอุปกรณ์ว่า สิ้นสุดการใช้บัสSDA (Serial Data) และ SCL (Serial Clock) เป็นสองขาในการสื่อสารผ่าน I2C (Inter-Integrated Circuit) และมีหน้าที่แตกต่างกันในการสื่อสารข้อมูลดิจิทัลระหว่างอุปกรณ์

SDA (Serial Data):

- SDA เป็นขาที่ใช้สำหรับส่งข้อมูลระหว่างอุปกรณ์ที่เชื่อมต่อผ่าน I2C
- ข้อมูลถูกส่งผ่าน SDA แบบเซรียล แสดงว่าข้อมูลถูกส่งเป็นบิตต่อบิต และอุปกรณ์ต้นทางและปลายทางต้องแชร์ช่องเดียวกันสำหรับ SDA

SCL (Serial Clock):

- SCL เป็นขาที่ใช้สำหรับส่งสัญญาณนาฬิกา (Clock) ที่ช่วยในการควบคุมการสื่อสาร
- สัญญาณนาฬิกาที่ถูกส่งผ่าน SCL ใช้ในการซิงเวลาข้อมูลและช่วยให้อุปกรณ์ต้นทางและปลายทางรวมถึงควบคุมความเร็วในการสื่อสารข้อมูล

2.8 Rotary Encoder

Rotary encoder (โรตารี เอนโคเดอร์) เป็นอุปกรณ์ที่ใช้ในการตรวจจับความเคลื่อนไหวของวัตถุหมุนหรืออุปกรณ์ที่มีการหมุน เช่น ตลับลิคหมุนหรือตำแหน่งของลวดขดลวด อุปกรณ์นี้สามารถระบุทิศทางและจำนวนความหมุนได้ โดยที่ไม่ต้องมีการติดตามตำแหน่งเริ่มต้น (home position) หรือ limit switches ทำให้เหมาะสำหรับการนำมาใช้ในแอปพลิเคชันที่ต้องการควบคุมหรือตรวจวัดการหมุนหรือตำแหน่งของวัตถุหมุนแบบละเอียด ๆ Rotary encoder สามารถใช้ได้หลายแอปพลิเคชัน เช่น การควบคุมความหมุนของลวดขดลวดในเครื่อง CNC, การปรับค่าในระบบควบคุมอุณหภูมิ, ระบบหมุนเพื่อเลื่อนหน้าจอ (rotary encoder บนหน้าจอคอมพิวเตอร์), หรือการใช้ในการวัดตำแหน่งในหลายแอปพลิเคชันอื่น ๆ ที่ต้องการความแม่นยำในการวัดตำแหน่งหรือความหมุนของอุปกรณ์.

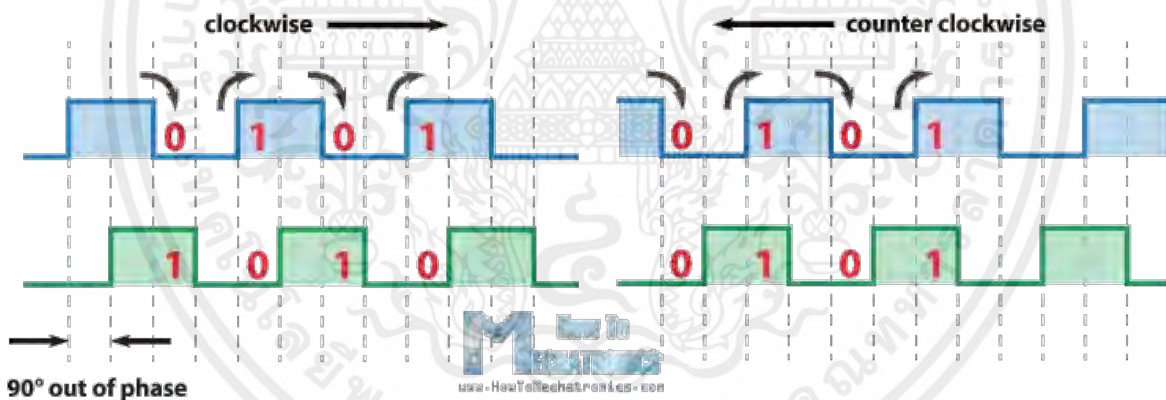


รูปที่ 2.11 rotary encoder

ที่มา ESP32 Pulse Counter PCNT with ESP-IDF and Rotary Encoder Example (esp32tutorials.com)

Rotary encoder module KY-040 เป็นโมดูลสำหรับอ่านความเร็วและทิศทางของการหมุน พร้อมสวิตช์กด สามารถใช้วัดความเร็วรอบ หรือใช้เป็นวาร์วเพื่อหมุนปรับค่าต่างๆ ต่างจาก potentiometer ตัวนี้จะหมุนได้ 360 องศาและไม่จำกัดรอบการหมุน

วิธีการหาทิศทางของการหมุน เกิดจากการสร้างสัญญาณ pulse : เมื่องานเริ่มหมุน ขา A และ B จะเริ่มทำการติดต่อกับขาคอมมอน และสร้างสัญญาณพัลส์เหลี่ยมสองสัญญาณตามที่แสดงในภาพด้านล่างนี้



รูปที่ 2.12 การอ่านค่า rotary encoder

ที่มา <https://howtomechatronics.com/tutorials/arduino/rotary-encoder-works-use-arduino/>

สามารถใช้สัญญาณที่ออกมา เพื่อกำหนดตำแหน่งที่หมุนไปแล้วได้โดยเพียงแค่นับจำนวนพัลส์ของสัญญาณ อย่างไรก็ตามหากเราต้องการกำหนดทิศทางการหมุนด้วย เราต้องพิจารณาสองสัญญาณพร้อมกัน

เราสามารถสังเกตได้ว่าสองสัญญาณออกมานั้นมีการเลื่อนที่ 90 องศาออกจากกันในเฟส หากเซ็นเซอร์หมุนตามเข็มนาฬิกา สัญญาณ A จะนำสัญญาณ B

ดังนั้นหากเรานับสเปคการทำงานทุกครั้งที่สัญญาณเปลี่ยนไปจาก High เป็น Low หรือจาก Low เป็น High เราสามารถสังเกตได้ว่าในเวลานั้นสองสัญญาณจะมีค่าตรงข้ามกัน ในทางกลับกัน หากเซ็นเซอร์หมุนตามทวนเข็มนาฬิกา สัญญาณออกมาจะมีค่าเท่ากัน ดังนั้นเราจึงสามารถเขียนโปรแกรมควบคุมเพื่ออ่านตำแหน่งของเซ็นเซอร์และทิศทางการหมุนได้อย่างง่ายดาย

2.9 XH-W1209 (ตัวควบคุมอุณหภูมิ)

ตัวควบคุมอุณหภูมิ หรือ Temperature Controller มีหลักการทำงานที่เรียกว่า การควบคุมแบบ On-Off Control ซึ่งตัวควบคุมอุณหภูมิ จะรับค่า SV (Setting Value) หรือ การตั้งค่าอุณหภูมิที่เราต้องการ และทำการเปรียบเทียบค่า PV กับค่า SV ซึ่งค่า PV มาจากอุปกรณ์ Sensor input เช่น ตัว Thermocouple หากค่า PV น้อยกว่าค่า SV ก็จะสั่งให้ Control Signal Output จ่ายสัญญาณไปควบคุมอุปกรณ์ขยายสัญญาณ Output เช่น SCR Power Regulator เพื่อให้แรงตัว Heater ให้ทำงานร้อนขึ้นเพื่อให้ได้อุณหภูมิตามที่ต้องการ แต่ถ้าค่า PV มากกว่าค่า SV ก็จะทำการลดสัญญาณ Control Signal Output เพื่อให้ Heater ทำงานลดลง

XH-W1209 ใช้เซ็นเซอร์ NTC แบบกันน้ำ ผู้ใช้สามารถนำไปประยุกต์ใช้งานได้หลากหลาย เช่น ตู้ฟักไข่, เปิด-ปิดพัดลมตามอุณหภูมิ, ตู้ปลา, ตู้แช่ เป็นต้น และยังสามารนำไปต่อ เพื่อควบคุมไหลตอย่างอื่นได้อีกด้วย

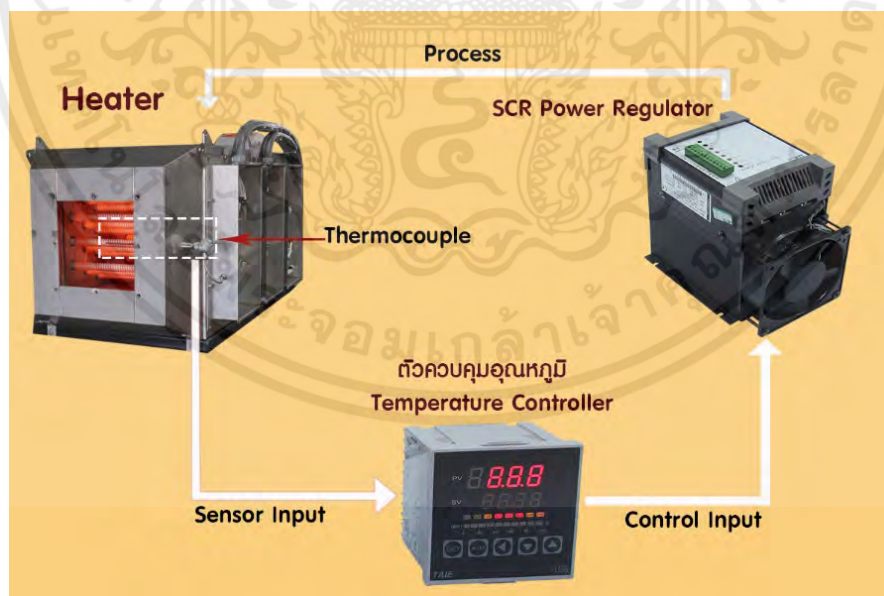


รูปที่ 2.13 XH-W1209

ที่มา https://hallroadlahore.pk/images/watermarked/1/detailed/1/w1209-temerature-controller-w1209-thermostat_2.jpg

โดยค่าต่างๆมีความหมาย ดังนี้

- **SV (Set Value)** เป็นค่าอุณหภูมิที่เราต้องการควบคุม
- **PV (Process Value)** เป็นค่าที่เกิดจากการอ่านค่าอุณหภูมิที่กำลังควบคุมอยู่ในกระบวนการผลิต ผ่านทาง Sensor Input
- **Control Signal Output** เป็นสัญญาณOutput เพื่อไปควบคุม Actuator ซึ่งได้จากการคำนวณของคอนโทรลเลอร์ PID, ON-OFF, Fuzzy เพื่อส่งสัญญาณ เช่น 4..20mA, 0-10VDC, Relay, SSR
- **Actuator** เป็นอุปกรณ์ทำหน้าที่ขยายสัญญาณ Control Signal Output เช่น Solid State Relay, SCR Power Regulator, Solinoid Valve, Magnetics Contactor เพื่อให้สามารถควบคุมโหลดได้
- **Heating Device** คือ ฮีตเตอร์ หรืออุปกรณ์ที่ใช้ทำความร้อนในระบบ
- **Thermocouple** เป็นอุปกรณ์ที่ใช้บอกค่าอุณหภูมิในกระบวนการ หรือ Sensor Input เพื่อส่งค่ากลับมาประมวลผล นอกจากThermocouple อุปกรณ์ Sensor Input มีหลายชนิดเช่น RTDs, Analog Signal



รูปที่ 2.14 temperature controller

ที่มา <http://www.hi-den.co.th/images/Knowledge/Temp/Temperature-Controller.png>

2.10 Heater

ฮีตเตอร์ หรือ เครื่องทำความร้อน คือ อุปกรณ์ชนิดหนึ่งใช้เพื่อสร้างความร้อนโดยเปลี่ยนพลังงานจากกระแสไฟฟ้าให้เป็นพลังงานความร้อน ฮีตเตอร์สามารถพบได้แทบทุกอุตสาหกรรมซึ่งอาจจะใช้ในกระบวนการบางอย่างที่ต้องใช้ความร้อน เช่น การต้มน้ำหรือให้ความร้อนตามระดับที่ต้องการแก่สารเคมี การอบแห้ง การอบสี การหลอม การเผา การซีลหรือปิดผนึก เป็นต้น นอกจากนี้ตามบ้านเรือนก็มีการใช้งานอุปกรณ์ประเภทฮีตเตอร์ เตารีด เครื่องทำน้ำอุ่น กระทะหรือหม้อไฟฟ้า ไดร์เป่าผม เป็นต้นเช่น หม้อหุงข้าว ฮีตเตอร์มีหลักการทำงาน คือ การสร้างความร้อนโดยการจ่ายกระแสไฟฟ้าแรงดัน 220VAC หรือ 380VAC ให้ไหลผ่านลวดตัวนำ (ตัวความต้านทาน R) ทำให้ลวดตัวนำมีความร้อนเกิดขึ้น โดยฮีตเตอร์มีหลายประเภท แต่ในโปรเจกต์นี้ทางผู้จัดทำจะใช้ฮีตเตอร์แบบหลอดแก้ว ซึ่งใช้งานกับตู้เย็น



รูปที่ 2.15 heater

ที่มา

<https://th.bing.com/th/id/R.906aca7f06d2d530476a089c25f0903a?rik=NINjwoq5Je4wdQ&riu=http%3a%2f%2fwww.isethailand.com>

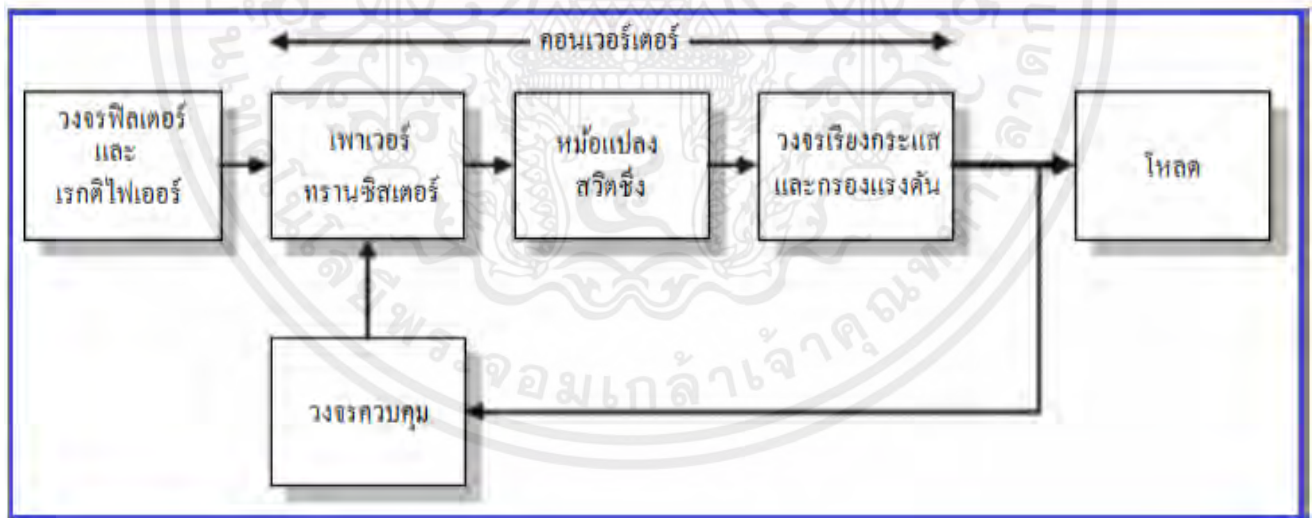
2.11 Switching power supply

สวิตชิงเพาเวอร์ซัพพลาย (Switching Power Supply) คือ อุปกรณ์แปลงแรงดันไฟฟ้ากระแสสลับ AC (Alternating Current) ที่มีแรงดันสูง (High Voltage) เป็นแรงดันไฟฟ้ากระแสตรง DC (Direct Current) ที่มีแรงดันต่ำ (Low Voltage) เช่น แปลงแรงดันไฟฟ้าทางด้านอินพุต (Input) 220Vac เป็นแรงดันไฟฟ้าทางด้านเอาต์พุต (Output) ที่มีแรงดันต่ำ 5Vdc, 12Vdc, 24Vdc เป็นต้น

สวิตชิงเพาเวอร์ซัพพลาย (Switching Power Supply) จะทำงานในลักษณะเดียวกันกับหม้อแปลงแรงดัน (Transformer) ทั่วไป แต่สวิตชิงเพาเวอร์ซัพพลาย (Switching Power Supply) จะมีประสิทธิภาพที่ดีกว่าและมีขนาดเล็กกว่า โดยหลักการทั่วไปของสวิตชิงเพาเวอร์ซัพพลาย (Switching Power Supply) จะประกอบด้วยดังนี้

1. เรกติไฟเออร์ (Rectifier) ทำหน้าที่แปลงแรงดันไฟฟ้ากระแสสลับ AC (Alternating Current) ให้เป็นแรงดันไฟฟ้ากระแสตรง DC (Direct Current)

2. คอนเวอร์เตอร์ (Converter) ทำหน้าที่แปลงความถี่แรงดันไฟฟ้ากระแสตรง (Frequency Converter DC Voltage) ให้เป็นแรงดันไฟฟ้ากระแสสลับที่มีความถี่สูง (High Frequency AC Voltage) และแปลงแรงดันไฟฟ้ากระแสสลับ (AC) ให้เป็นแรงดันไฟฟ้ากระแสตรง (DC) โดยมีความต้านทานทางด้านเอาต์พุตของแรงดันไฟฟ้ากระแสตรงให้ได้ตามความต้องการอีกครั้ง ดังรูป



รูปที่ 2.16 โครงสร้างการทำงานของ Switching Power Supply

ที่มา <https://www.primusthai.com/primusthai/1.SwitchingPowerSupply-Know90.png>



รูปที่ 2.17 สวิตชิงเพาเวอร์ซัพพลาย (Switching Power Supply)

ที่มา [https://s2ins.com/wp-](https://s2ins.com/wp-content/uploads/2021/10/%E0%B8%AA%E0%B8%A7%E0%B8%B4%E0%B8%95%E0%B8%8A%E0%B8%B4%E0%B9%88%E0%B8%87%E0%B8%9E%E0%B8%B2%E0%B8%A7%E0%B9%80%E0%B8%A7%E0%B8%AD%E0%B8%A3%E0%B9%8C%E0%B8%8B%E0%B8%B1%E0%B8%9E%E0%B8%9E%E0%B8%A5%E0%B8%B2%E0%B8%A2-Switching-Power-Supply-5V-4A-12V-3A-600x450.jpg)

[content/uploads/2021/10/%E0%B8%AA%E0%B8%A7%E0%B8%B4%E0%B8%95%E0%B8%8A%E0%B8%B4%E0%B9%88%E0%B8%87%E0%B8%9E%E0%B8%B2%E0%B8%A7%E0%B9%80%E0%B8%A7%E0%B8%AD%E0%B8%A3%E0%B9%8C%E0%B8%8B%E0%B8%B1%E0%B8%9E%E0%B8%9E%E0%B8%A5%E0%B8%B2%E0%B8%A2-Switching-Power-Supply-5V-4A-12V-3A-600x450.jpg](https://s2ins.com/wp-content/uploads/2021/10/%E0%B8%AA%E0%B8%A7%E0%B8%B4%E0%B8%95%E0%B8%8A%E0%B8%B4%E0%B9%88%E0%B8%87%E0%B8%9E%E0%B8%B2%E0%B8%A7%E0%B9%80%E0%B8%A7%E0%B8%AD%E0%B8%A3%E0%B9%8C%E0%B8%8B%E0%B8%B1%E0%B8%9E%E0%B8%9E%E0%B8%A5%E0%B8%B2%E0%B8%A2-Switching-Power-Supply-5V-4A-12V-3A-600x450.jpg)

บทที่ 3

หลักการทํางาน และการออกแบบ

3.1 ลักษณะการทํางานของวงจร

เริ่มจากการต่อไฟบ้าน 220 โวลต์เข้าหม้อแปลงแรงดันเป็นไฟตรง จากนั้นส่งต่อสัญญาณเข้าวงจรปรับแรงดันให้เหลือแรงดันไฟฟ้า 5 โวลต์ เพื่อนำเข้าไปเลี้ยงบอร์ด Arduino NANO เมื่อมีแรงดันเข้าไปในบอร์ด บอร์ดจะเริ่มทํางานตามคำสั่งที่เขียนไว้ในโปรแกรม Arduino IDE ทำให้ผู้ใช้งานสามารถควบคุมการเปิดปิดของหลอดไฟยูวีซีและระบบอบลมร้อนได้ โดยผู้ใช้งานสามารถควบคุมการทํางานของตู้อบได้ 2 โหมด ได้แก่ แบบ manual mode ในโหมดนี้สามารถเปิด-ปิดการทํางานของหลอดไฟยูวีซีและพัดลมมอเตอร์ได้ด้วยการกดปุ่ม และ แบบ timer mode จะสามารถตั้งเวลาให้หลอดไฟยูวีซีและพัดลมมอเตอร์ทํางานกี่ชั่วโมง กี่นาที หรือกี่วินาที ก็ได้ เมื่อตั้งเวลาเสร็จแล้วและกดปุ่ม start เวลา ก็จะเริ่มนับถอยหลังและหลอดไฟหรือพัดลมมอเตอร์ก็จะเริ่มทํางาน

3.2 การออกแบบวงจร

3.2.1 ส่วนของโปรแกรม

1) การเขียนโปรแกรม

เขียนโปรแกรมโดยใช้โปรแกรม Wokwi เพื่อเขียนโปรแกรม ทดสอบโค้ด และจำลองการทํางานของวงจร และใช้โปรแกรม Arduino IDE ในการ upload โค้ดใส่ Arduino Nano ซึ่ง code ที่ใช้ในวงจร คือ

```
#include <LiquidCrystal_I2C.h>  
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

ติดตั้ง library

```
#define ENCODER_CLK 2  
#define ENCODER_DT 3  
#define ENCODER_SW 4
```

```
long counter = 0;
```

```
bool updateDisplay = false;
```

```
uint8_t Menu = 0;
```

```
uint8_t MainMenu = 0;
```

```
bool SelectMode = 0;
```

```
uint8_t cursor = 0;
```

```
bool lock_data = false;
```

```
int Hset = 0;
```

```
int Mset = 0;
```

```
int Sset = 0;
```

```
bool startorstop = false;
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  pinMode(5,OUTPUT);
```

```
  digitalWrite(5,LOW);
```

```
  pinMode(6,OUTPUT);
```

กำหนดขาอุปกรณ์ และตัวแปรต่างๆในการเก็บค่า
ตั้งค่า output ขา 5 และ 6 ของ Arduino เป็น
low

```
digitalWrite(6,LOW);  
  
pinMode(ENCODER_CLK, INPUT);  
  
pinMode(ENCODER_DT, INPUT);  
  
pinMode(ENCODER_SW, INPUT_PULLUP);  
  
attachInterrupt(digitalPinToInterrupt(ENCODER_CLK), readEncoder, FALLING);
```

```
lcd.init();  
  
lcd.backlight();  
  
lcd.setCursor(0, 0);  
lcd.print("UVC with Hot air ");  
  
lcd.setCursor(0, 1);  
lcd.print(" Sterilizer ");  
  
delay(2000);  
  
LCD_Display();  
  
}
```

หน้าจอ lcd แสดงตัวอักษร
"UVC with Hot air Sterilizer"

```
void readEncoder() {  
  
int dtValue = digitalRead(ENCODER_DT);  
  
if (dtValue == HIGH) {  
  
counter++; // Clockwise
```

กำหนดให้ค่า dt เพิ่มขึ้นเมื่อหมุน
rotary encoder ตามเข็มนาฬิกา

```
if((Menu == 3) || (Menu == 7))
```

```
{
```

```
if(lock_data == false)
```

```
{
```

```
cursor++;
```

```
if(cursor > 4) cursor = 0;
```

```
}
```

```
else
```

```
{
```

```
if(cursor == 0) if(Hset < 100) Hset++;
```

```
if(cursor == 1) if(Mset < 59) Mset++;
```

```
if(cursor == 2) if(Sset < 59) Sset++;
```

```
}
```

```
}
```

```
}
```

```
if(dtValue == LOW) {
```

```
counter--; // Counterclockwise
```

```
if((Menu == 3) || (Menu == 7))
```

```
{
```

```
if(lock_data == false)
```

```
{
```

```
cursor--;
```

```
if(cursor < 0) cursor = 4;
```

และที่ Menu ที่ 3 และ 7 กำหนดให้ cursor มีค่าเพิ่มขึ้นจากการหมุน rotary encoder ตามเข็มนาฬิกา และเมื่อค่า cursor มากกว่า 4 ให้รีเซ็ตค่ากลับมาที่ 0

โดยสามารถปรับเวลา
ในหน่วยชั่วโมงได้สูงสุด 100 ชั่วโมง
ในหน่วยนาทีได้สูงสุด 59 นาที
ในหน่วยวินาทีได้สูงสุด 59 วินาที

กำหนดให้ค่า dt ลดลงเมื่อหมุน
rotary encoder ทวนเข็มนาฬิกา

และที่ Menu ที่ 3 และ 7 กำหนดให้ cursor มีค่าลดลงจากการหมุน rotary encoder ทวนเข็มนาฬิกา และเมื่อค่า cursor มากกว่า 4 ให้รีเซ็ตค่ากลับมาที่ 0

```

}
else
{
    if(cursor == 0) if(Hset > 0) Hset--;
    if(cursor == 1) if(Mset > 0) Mset--;
    if(cursor == 2) if(Sset > 0) Sset--;
}
}
}
if(Menu == 0) SelectMode = !SelectMode;
if((Menu == 1) || (Menu == 5))
{
    MainMenu++;
    if(MainMenu == 3) MainMenu = 0;
}
updateDisplay = true;
}

long task1s;

void loop()
{
    if(digitalRead(ENCODER_SW) == LOW)

```

โดยสามารถปรับเวลา
 ในหน่วยชั่วโมงได้ต่ำสุด 0 ชั่วโมง
 ในหน่วยนาทีได้ต่ำสุด 0 นาที
 ในหน่วยวินาทีได้ต่ำสุด 0 วินาที

```

{
  if(Menu == 0)
  {
    if(SelectMode == 0) Menu = 1;
    else Menu = 5;
  }
  else if(Menu == 1)
  {
    if(MainMenu == 0) Menu = 2;
    {
      Menu = 3;
      Hset = 0;
      Mset = 0;
      Sset = 0;
    }
    else
    {
      Menu = 0;
      MainMenu = 0;
    }
  }
}

```

เมื่อกดสวิตช์เพื่อเลือกการทำงานของ UVC หรือ Hot air (Menu 0) จอ lcd จะแสดงข้อความ manual mode หรือ timer mode หรือ exit จะตรงกับ Menu 1 หากเลือก UVC mode Menu 5 หากเลือก Hot air mode

เมื่อกดสวิตช์เพื่อเลือกการทำงานของ timer mode ของ UVC (Menu 1) จอ lcd จะแสดงตัวเลข 0:0:0

เมื่อกดสวิตช์ที่ Exit ก็จะไปหน้าหลักเช่นเดิม

```

else if(Menu == 2)
{
  if((abs(counter % 2)) == 0)
  {
    Menu = 4;
    digitalWrite(5,HIGH);
  }
else
{
  Menu = 1;
  MainMenu = 0;
}
}
else if(Menu == 3)
{
  if(cursor < 3) lock_data = !lock_data;
  else if(cursor == 3)
  {
    if((startorstop == false) && ((Hset + Mset + Sset) > 0)) startorstop = true;
    else if((Hset + Mset + Sset) > 0) startorstop = !startorstop;
    digitalWrite(5,startorstop);
  }
}

```

เมื่อกดสวิตช์เพื่อเลือกการทำงาน manual mode ของ UVC จอ lcd จะแสดงข้อความ Start หรือ Exit (Menu 2) ถ้ากดสวิตช์ที่ Start การทำงานของหลอดไฟยูวีซีจะเริ่มขึ้นพร้อมกับข้อความ UV Light ON (Menu 4)

ถ้ากดสวิตช์ที่ Exit ก็จะไปหน้าหลักเช่นเดิม

เป็นการเซตค่าเวลา timer mode ของ UVC ถ้าจำนวนเวลาชั่วโมง นาที วินาทีรวมกันมากกว่า 0 หลอดไฟยูวีซีจะทำงาน

```

else if(cursor == 4)
{
    Menu = 1;
    MainMenu = 1;
    cursor = 0;
}
}
else if(Menu == 4)
{
    digitalWrite(5,LOW);
}
else if(Menu == 5)
{
    if(MainMenu == 0) Menu = 6;
    else if(MainMenu == 1)
    {
        Menu = 7;
        Hset = 0;
        Mset = 0;
        Sset = 0;
    }
}

```

ถ้า cursor อยู่ที่ Exit แล้วผู้ใช้งานกด หน้าจอก็จะกลับไปหน้าจอเดิม และเซตค่า cursor เป็น 0

ถ้ากดสวิตช์อีกครั้งในขณะที่หลอดยูวีซีกำลังทำงานอยู่ หลอดไฟก็จะดับลงพร้อมกับที่หน้าจอ lcd แสดงข้อความ UV Light Off และกลับสู่หน้าหลัก (Menu 4)

เมื่อกดสวิตช์เพื่อเลือกการทำงาน timer mode ของระบบอบลมร้อน (Menu 7) จอ lcd จะแสดงตัวเลข 0:0:0

```

else
{
    Menu = 0;
    MainMenu = 0;
}

```

เมื่อกดสวิตช์ที่ Exit ก็จะกลับไปหน้าหลักเช่นเดิม

```

else if(Menu == 6)

```

```

{
    if((abs(counter % 2)) == 0)
    {
        Menu = 8;
        digitalWrite(6,HIGH);
    }
}

```

เมื่อกดสวิตช์เพื่อเลือกการทำงาน manual mode ของ Hot air จอ lcd จะแสดงข้อความ Start หรือ Exit (Menu 6) ถ้ากดสวิตช์ที่ Start การทำงานของฮีสเตอร์และพัดลมจะเริ่มขึ้นพร้อมกับข้อความ Heater ON (Menu 8)

```

else

```

```

{
    Menu = 1;
    MainMenu = 0;
}
}

```

ถ้ากดสวิตช์ที่ Exit ก็จะกลับไปหน้าหลักเช่นเดิม

```

else if(Menu == 7)

```

```

{
    if(cursor < 3) lock_data = !lock_data;
}

```

```

else if(cursor == 3)
{
    if((startorstop == false) && ((Hset + Mset + Sset) > 0)) startorstop = true;
    else if((Hset + Mset + Sset) > 0) startorstop = !startorstop;
    digitalWrite(6,startorstop);
}
else if(cursor == 4)
{
    Menu = 1;
    MainMenu = 1;
    cursor = 0;
}
}
else if(Menu == 8)
{
    digitalWrite(6,LOW);
}
while(digitalRead(ENCODER_SW) == LOW);
counter = 0;
updateDisplay = true;
}

```

เป็นการเซตค่าเวลา timer mode ของ Hot air ถ้าจำนวนเวลาชั่วโมง นาที วินาทีรวมกันมากกว่า 0 ฮีสเตอร์และพัดลมจะทำงาน

ถ้า cursor อยู่ที่ Exit แล้วผู้ใช้งานกด หน้าจอก็จะกลับไปหน้าเดิม และเซตค่า cursor เป็น 0

ถ้ากดสวิตช์อีกครั้งในขณะที่ฮีสเตอร์และพัดลม กำลังทำงานอยู่ ฮีสเตอร์และพัดลมก็จะดับลง พร้อมกับที่หน้าจอ lcd แสดงข้อความ Heater Off และกลับสู่หน้าหลัก (Menu 8)

```
if(((millis() - task1s) >= 1000) && (startorstop == true))
```

```
{
```

```
task1s = millis();
```

```
Sset--;
```

```
if((Sset < 0) && ((Hset > 0) || (Mset > 0)))
```

```
{
```

```
Sset = 59;
```

```
Mset--;
```

```
}
```

```
if((Mset < 0) && (Hset > 0))
```

```
{
```

```
Mset = 59;
```

```
Hset--;
```

```
}
```

```
if((Hset == 0) && (Mset == 0) && (Sset == 0))
```

```
{
```

```
digitalWrite(5,LOW);
```

```
digitalWrite(6,LOW);
```

```
startorstop = false;
```

```
}
```

```
updateDisplay = true;
```

```
}
```

เป็นการเซตค่าหลักวินาที คือ เมื่อหมุน rotary encoder จนถึง 59 แล้ว ซึ่งเป็นเลขสูงสุด จะไม่นับเป็น 1 ที่หลักวินาที

เป็นการเซตค่าหลักนาที คือ เมื่อหมุน rotary encoder จนถึง 59 แล้ว ซึ่งเป็นเลขสูงสุด จะไม่นับเป็น 1 ที่หลักชั่วโมง

ถ้าจำนวนชั่วโมง นาที วินาทีที่มีค่าเท่ากับ 0 ทั้งหลอดไฟยูวีซี ฮีสเตอร์ และพัดลมระบายความร้อน จะไม่ทำงาน

```

if(updateDisplay == true)
{
    LCD_Display();
    updateDisplay = false;
}
}

```

```

void LCD_Display()
{
    if(Menu == 0)
    {
        if(SelectMode == 0)
        {
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print(" Select Mode ");
            lcd.setCursor(0, 1);
            lcd.print(" < UVC Mode > ");
        }
    }
}

```

Menu 0 หน้าจอ lcd แสดงตัวอักษร
 “Select Mode และ UVC Mode”

```

else
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(" Select Mode ");
  lcd.setCursor(0, 1);
  lcd.print("< Hot Air Mode >");
}
}
else if(Menu == 1)
{
  if(MainMenu == 0)
  {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("[UV] Main Menu ");
    lcd.setCursor(0, 1);
    lcd.print(" < ManualMode > ");
  }
}

```

หรือ แสดงตัวอักษร
 “Select Mode และ Hot air Mode”

Menu 1 หน้าจอ lcd แสดงตัวอักษร
 “Main Menu และ Manual Mode”
 โดยใช้ใน UVC Mode

```

else if(MainMenu == 1)
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("[UV] Main Menu ");
    lcd.setCursor(0, 1);
    lcd.print(" < Timer Mode > ");
}
else
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("[UV] Main Menu ");
    lcd.setCursor(0, 1);
    lcd.print(" < EXIT > ");
}
} else if(Menu == 2)
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("  START!  ");
    lcd.setCursor(0, 1);
    lcd.print("  EXIT>>  ");

```

หรือ แสดงตัวอักษร
 “Main Menu และ Timer Mode”
 โดยใช้ใน UVC Mode

หรือ แสดงตัวอักษร
 “Main Menu และ Exit”
 โดยใช้ใน UVC Mode

Menu 2 หน้าจอ lcd แสดงตัวอักษร
 “Start และ Exit”
 โดยใช้ใน UVC Mode

```

lcd.setCursor(0, abs(counter % 2));

lcd.print(">");

lcd.setCursor(15, abs(counter % 2));

lcd.print("<");
}
else if(Menu == 3)
{
if(startorstop == false) lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Time:");
if(startorstop == true)
{
lcd.print(" ");
lcd.setCursor(5, 0);
}
if(cursor == 0) lcd.print(">");
lcd.print(Hset);
if((cursor == 0) && (lock_data == true)) lcd.print("<");
lcd.print(":");
if(cursor == 1) lcd.print(">");
lcd.print(Mset);
if((cursor == 1) && (lock_data == true)) lcd.print("<");
lcd.print(":");

```

Menu 3 หน้าจอ lcd แสดงตัวอักษร

“Time”

โดยใช้ใน UVC Mode

และกำหนดตำแหน่งการเลื่อนของ
rotary encoder

หมุน 0 ครั้ง จะไปที่ตำแหน่งชั่วโมง

หมุน 1 ครั้ง จะไปที่ตำแหน่ง นาที

หมุน 2 ครั้ง จะไปที่ตำแหน่งวินาที

หมุน 3 ครั้ง cursor จะไปที่ Start และเมื่อกด
rotary encoder เพื่อหยุดการทำงานของ
หลอดไฟแล้ว หน้าจอ lcd จะแสดงตัวอักษร
Pause แทน

และเมื่อหมุน 4 ครั้ง cursor จะไปที่ Exit

```

if(cursor == 2) lcd.print(">");

lcd.print(Sset);

if((cursor == 2) && (lock_data == true)) lcd.print("<");

lcd.setCursor(0, 1);

if(cursor == 3) lcd.print(">");

else lcd.print(" ");

if(startorstop == false) lcd.print("START");

else lcd.print("PAUSE");

if(cursor == 3) lcd.print("<");

lcd.setCursor(10,1);

if(cursor == 4) lcd.print(">");

else lcd.print(" ");

lcd.print("EXIT");

if(cursor == 4) lcd.print("<");

else lcd.print(" ");

}

else if(Menu == 4)

{

lcd.clear();

lcd.setCursor(0, 0);

if(digitalRead(5) == HIGH) lcd.print(" UV Light ON! ");

else

```

Menu 4 จะแสดงการทำงานของหลอด UVC และ
หน้าจอ lcd จะแสดงข้อความ UV Light ON

```

{
  lcd.print(" UV Light OFF ");
  delay(1000);
  Menu = 1;
  MainMenu = 0;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("[UV] Main Menu ");
  lcd.setCursor(0, 1);
  lcd.print(" < ManualMode > ");
}
} else if(Menu == 5)
{
  if(MainMenu == 0)
  {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("[HF] Main Menu ");
    lcd.setCursor(0, 1);
    lcd.print(" < ManualMode > ");
  }
}

```

และจะหยุดการทำงานของหลอด UVC โดย หน้าจอ lcd จะแสดงข้อความ UV Light Off จากนั้นก็จะกลับไป Menu 1

Menu 5 หน้าจอ lcd แสดงตัวอักษร “Main Menu และ Manual Mode” โดยใช้ใน Hot air Mode

```

else if(MainMenu == 1)
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("[HF] Main Menu ");
    lcd.setCursor(0, 1);
    lcd.print(" < Timer Mode > ");
}
else
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("[HF] Main Menu ");
    lcd.setCursor(0, 1);
    lcd.print(" < EXIT > ");
}
}

```

หรือ แสดงตัวอักษร
 “Main Menu และ Timer Mode”
 โดยใช้ใน Hot air Mode

หรือ แสดงตัวอักษร
 “Main Menu และ Exit”
 โดยใช้ใน Hot air Mode

```

else if(Menu == 6)
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("  START!  ");
    lcd.setCursor(0, 1);
    lcd.print("  EXIT>>  ");
    lcd.setCursor(0, abs(counter % 2));
    lcd.print(">");
    lcd.setCursor(15, abs(counter % 2));
    lcd.print("<");
}

else if(Menu == 7)
{
    if(startorstop == false) lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Time:");

```

Menu 6 หน้าจอ lcd แสดงตัวอักษร

“Start และ Exit”

โดยใช้ใน Hot air Mode

Menu 7 หน้าจอ lcd แสดงตัวอักษร

“Time”

โดยใช้ใน Hot air Mode

```

if(startorstop == true)
{
    lcd.print(" ");
    lcd.setCursor(5, 0);
}

if(cursor == 0) lcd.print(">");

lcd.print(Hset);

if((cursor == 0) && (lock_data == true)) lcd.print("<");

lcd.print(":");

if(cursor == 1) lcd.print(">");

lcd.print(Mset);

if((cursor == 1) && (lock_data == true)) lcd.print("<");

lcd.print(":");

if(cursor == 2) lcd.print(">");

lcd.print(Sset);

if((cursor == 2) && (lock_data == true)) lcd.print("<");

lcd.setCursor(0, 1);

if(cursor == 3) lcd.print(">");

else lcd.print(" ");

if(startorstop == false) lcd.print("START");

else lcd.print("PAUSE");

if(cursor == 3) lcd.print("<");

```

และกำหนดตำแหน่งการเลื่อนของ

rotary encoder

จะเขียนการทำงานให้เหมือนกับเมนูที่ 3

```

lcd.setCursor(10,1);

if(cursor == 4) lcd.print(">");

else lcd.print(" ");

lcd.print("EXIT");

if(cursor == 4) lcd.print("<");

else lcd.print(" ");

}

else if(Menu == 8)

{

lcd.clear();

lcd.setCursor(0, 0);

if(digitalRead(6) == HIGH) lcd.print(" Heater ON! ");

else

{

lcd.print(" Heater OFF ");

delay(1000);

Menu = 5;

MainMenu = 0;

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("[HF] Main Menu ");

lcd.setCursor(0, 1);

```

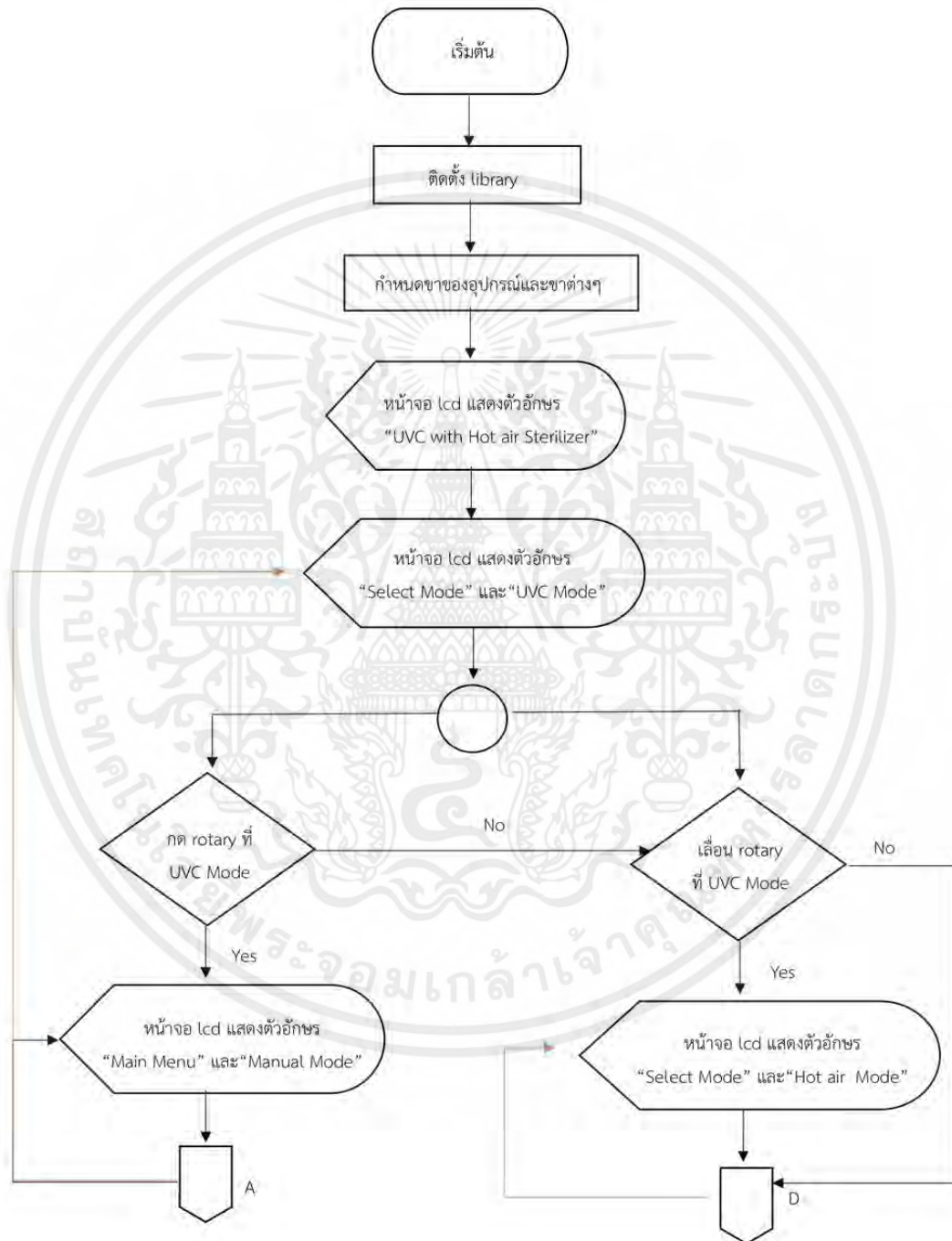
Menu 8 จะแสดงการทำงานของ
ฮีสเตอร์และพัดลม พร้อมกับที่หน้าจอ lcd
แสดงข้อความ Heater ON

และจะหยุดการทำงานของฮีสเตอร์และพัดลม
โดยหน้าจอ lcd จะแสดงข้อความ Heater OFF
จากนั้นก็กลับไป Menu 5

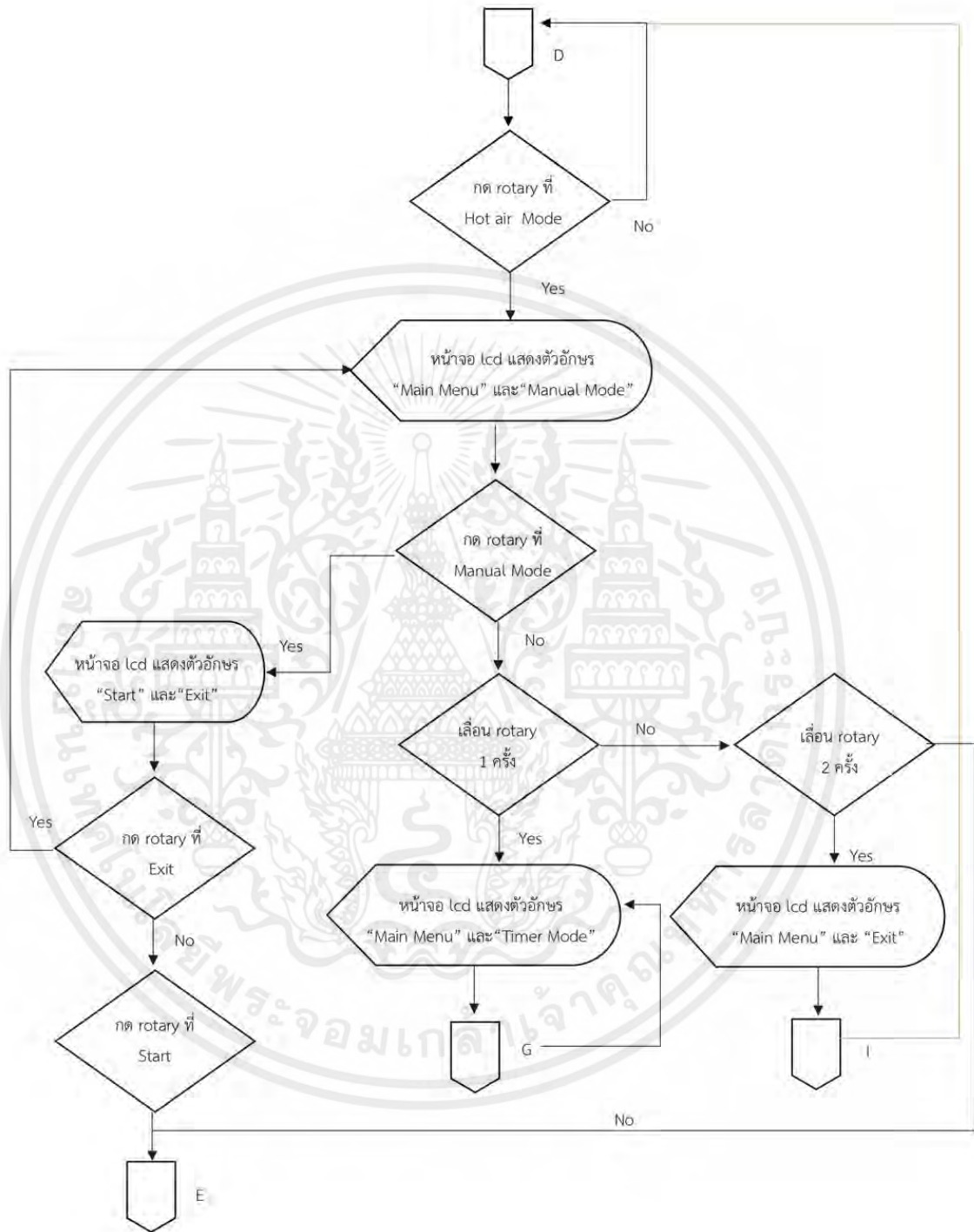
```
lcd.print(" < ManualMode > ");  
  
}  
  
}  
  
}
```

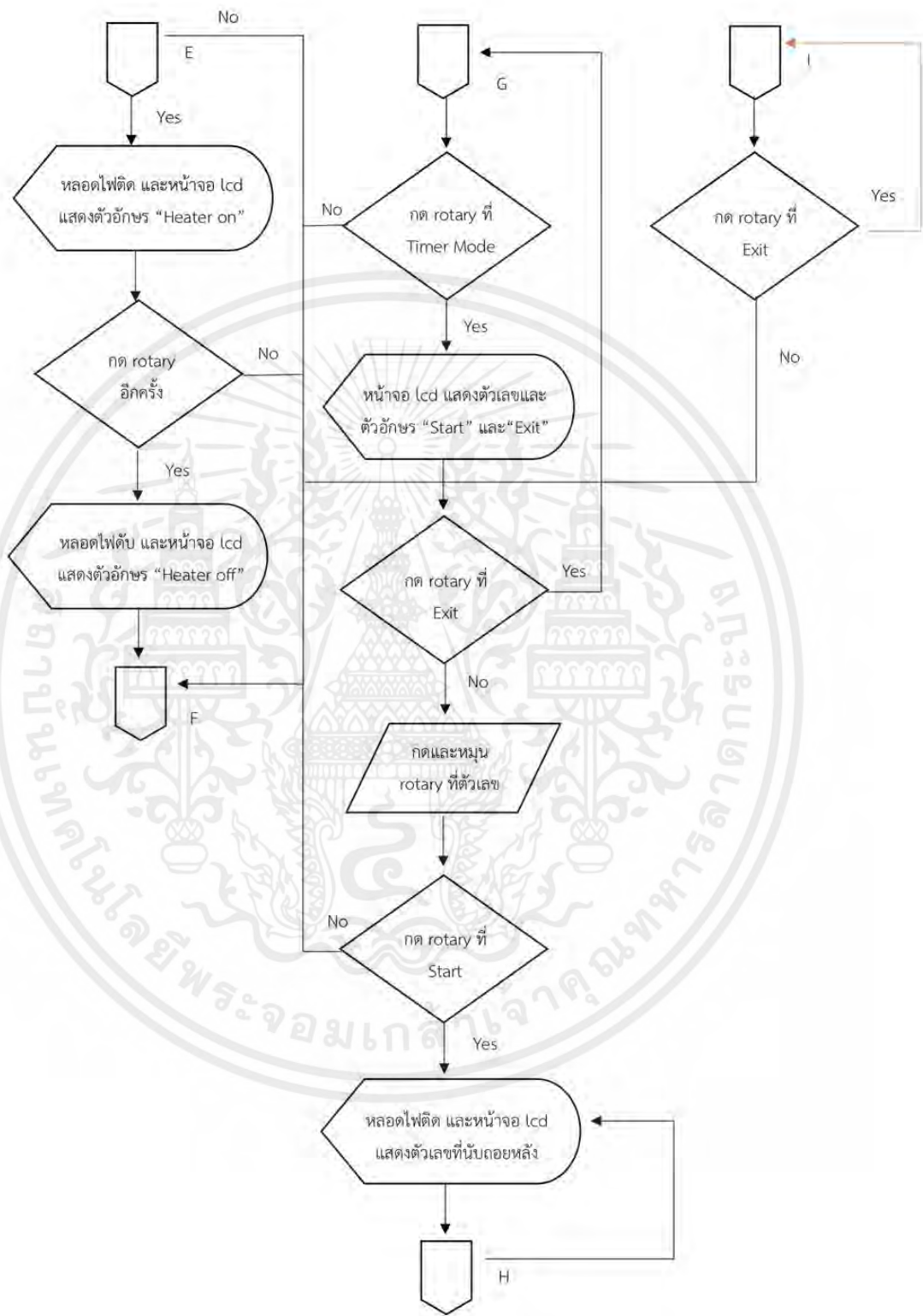


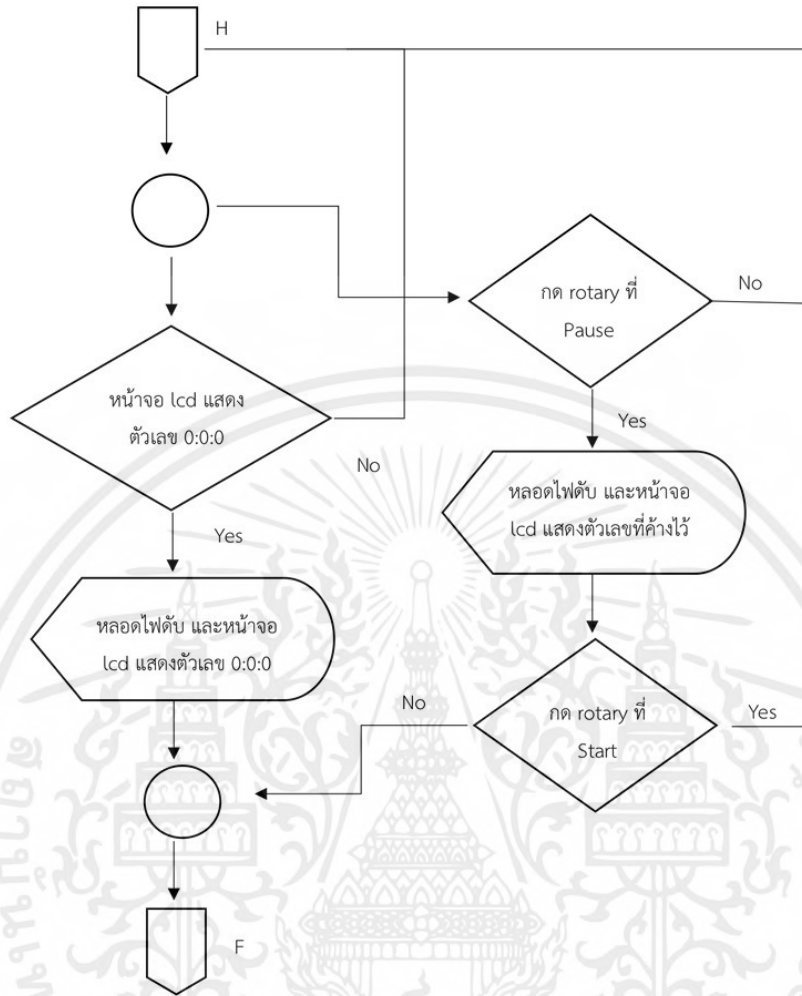
2) Flowchart จากการเขียนโปรแกรม











3.2.2 การออกแบบวงจร และอุปกรณ์ที่ใช้

อุปกรณ์ที่ใช้ไฟกระแสตรง 5 โวลต์ เพื่อเป็นไฟเลี้ยงในอุปกรณ์ของวงจรของตู้อบฆ่าเชื้อด้วยรังสียูวีซีพร้อมด้วยระบบบอบลมร้อน ได้แก่ Arduino NANO ,Relay ,Rotary encoder และ จอLCD แต่ไฟที่จ่ายจากครัวเรือนนั้นเป็นไฟฟ้ากระแสสลับ 220 โวลต์ ทำให้ไม่สามารถจ่ายไฟเข้าอุปกรณ์ได้โดยตรง จึงต้องใช้หม้อแปลงแปลงจากไฟฟ้ากระแสสลับเป็นไฟฟ้ากระแสตรง จากนั้นสร้างวงจรลดแรงดันไฟปรับแรงดันให้เหลือ 5 และ 12 โวลต์ก่อนจึงจะนำมาใช้ได้ โดยต้องต่อฟิวส์และสวิตช์สำหรับป้องกันอันตราย และการลัดวงจรด้วย

หลังจากทำวงจรปรับแรงดันเสร็จแล้ว ก็จะสร้างวงจรสำหรับใช้ในการทำตู้อบ เพื่อติดตั้งลงกล่อง โดยอุปกรณ์ที่ใช้ในวงจรมีดังนี้

- UVC light (6 Watt)

แรงดันที่ต้องใช้ : 220 V AC

เมื่อทำงานกินกระแส : ~ 30 mA

เนื่องจากหลอดไฟยูวีซีใช้ไฟฟ้ากระแสสลับ สายไฟเส้นหนึ่งของหลอดยูวีซีจึงต้องกับไฟบ้าน 220 โวลต์ และสายอีกเส้นจะทำการ Break เข้ากับขา common ของรีเลย์ตัวที่ 1 เพื่อใช้ในการควบคุมการเปิดปิดหลอดไฟ

- Heater (112 Watt)

แรงดันที่ต้องใช้ : 220 V AC

เมื่อทำงานกินกระแส : ~ 500 mA

เช่นเดียวกับหลอดยูวีซีใช้ไฟฟ้ากระแสสลับ สายไฟเส้นหนึ่งของหลอดยูวีซีจึงต้องกับไฟบ้าน 220 โวลต์ และสายอีกเส้นจะทำการ Break เข้ากับขา common ของรีเลย์ตัวที่ 2 เพื่อใช้ในการควบคุมการเปิดปิดหลอดไฟ

- AC Fan

แรงดันที่ต้องใช้ : 220 V AC

เมื่อทำงานกินกระแส : ~ 80 mA

จะต่อขนานกับฮีสเตอร์เพื่อให้ทำงานพร้อมกันระหว่างที่มีกระแสไฟฟ้าไหลผ่าน เพื่อพัดไอความร้อนให้กระจายไปในพื้นที่ว่างภายในกล่อง

- W1209 (ตัวตัดวงจร ควบคุมอุณหภูมิ)

แรงดันที่ต้องใช้ : 12 V DC

เมื่อทำงานกินกระแส : ~ 80 mA

ตัวควบคุมนี้มีหน้าที่ตัดวงจร เมื่อเซนเซอร์อุณหภูมิวัดอุณหภูมิได้ค่ามากกว่าอุณหภูมิที่ตั้งไว้ และจะทำงานอีกครั้งเมื่ออุณหภูมิต่ำกว่าค่าที่ตั้งไว้ อุปกรณ์ตัวนี้จึงต้องต่อกับรีเลย์ตัวที่ 2 ก่อนจะต่อดำเนินการ และพัลลัม เพื่อรอคำสั่งจาก Arduino ให้ทำงาน และสามารถตัดวงจรได้เมื่อเราต้องการ

- Relay 5 V 2

แรงดันที่ต้องใช้ : 5 V DC

เมื่อทำงานกินกระแส : ~70 mA

ใช้รีเลย์ในการควบคุมเวลาการทำงานของหลอดยูวีซี ฮีสเตอร์ และพัลลัม ทำให้สามารถปรับเปลี่ยนระยะเวลาการใช้งานตามความต้องการ และป้องกันไม่ให้หลอด UVC เปิดทำงานตลอดเวลาที่ไม่จำเป็น โดยใช้วิธีการต่อเราจะต่อขา COM ของรีเลย์เข้ากับหลอดไฟยูวีซี และขา NO ของรีเลย์ต่อเข้ากับแหล่งจ่ายไฟ 220 โวลต์ เพื่อให้เมื่อเริ่มเสียบปลั๊กไฟ หลอดไฟจะยังคงดับอยู่ ในสภาวะปกติที่ไม่มีการกดสวิทช์(หรือไม่มีคำสั่งให้จ่ายไฟ) จะไม่มีการจ่ายไฟเข้าไปที่ขดลวดของรีเลย์ ทำให้ไม่เกิดการเหนี่ยวนำที่หน้าสัมผัส หลอดไฟจึงไม่ติด แต่เมื่อมีการกดสวิทช์(หรือมีคำสั่งให้จ่ายไฟ) จะทำให้เกิดการเหนี่ยวนำส่งผลให้หน้าสัมผัสของรีเลย์เปลี่ยนไปเป็น NC หรือ normal close ทำให้หลอดไฟติด

- Rotary Encoder

แรงดันที่ต้องใช้ : 5 V DC

เมื่อทำงานกินกระแส : ~ 40 mA

ใช้ในการปรับค่า โดยวัดจากการเทียบสัญญาณที่ได้จากการหมุน และยังเป็นปุ่มที่เอาไว้เลือกโหมดการทำงานของหลอดยูวีซี และฮีสเตอร์

- LCD Display 16x2

แรงดันที่ต้องใช้ : 5 V DC

เมื่อทำงานกินกระแส : ~ 20 mA

ใช้ควบคู่กับ Rotary encoder เพื่อแสดงโหมดหรือการทำงานระหว่างที่กำลังเสียบปลั๊กดูบ โดยขึ้นตัวอักษรที่หน้าจอตามโปรแกรมที่เขียนไว้ใน Arduino IDE

- Arduino NANO

แรงดันที่ต้องใช้ : 5 V DC

เมื่อทำงานกินกระแส :

- กระแส ต่อ I/P pin 40 mA
- กระแส VCC และ GND 200mA

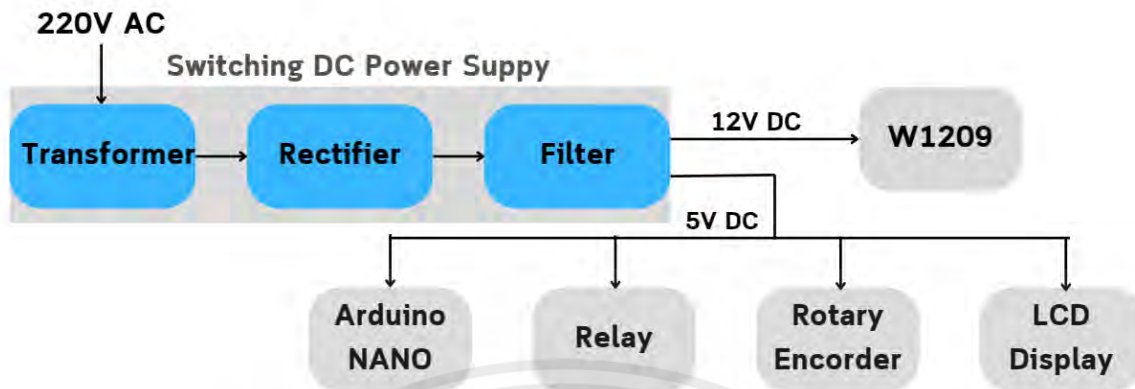
ซึ่ง Arduino มีการใช้ I/P pin ทั้งหมด 7 pin กระแสสูงสุดที่ใช้ทั้งหมดคือ $(40 \times 7) + 200 = 480$ mA

อุปกรณ์ที่สำคัญที่สุดสำหรับโปรเจกต์นี้ ใช้เพื่อควบคุมการทำงานของอุปกรณ์ทั้งหมดในวงจร โดยจะต้องต่อแต่ละขาให้ถูกต้องตามการทำงาน ซึ่งมีการต่อดังนี้

- Relay : ต่อขา IN ของรีเลย์เข้ากับขาดิจิทัลของ Arduino Nano ซึ่งในโครงการนี้เราใช้ขา D5 และ D6 เพื่อใช้ในการการส่งสัญญาณไฟฟ้าไปยังรีเลย์เพื่อเปิดหรือปิดการทำงานของอุปกรณ์
- Rotary encoder : ต่อขา CLK (Clock) กับขา D2 / ต่อขา DT (Data) กับขา D3 / ต่อขา SW (Switch) กับขา D4 เพื่ออ่านค่าการหมุน และค่านี้สามารถนำไปใช้ในการควบคุมการทำงานของระบบหรืออุปกรณ์อื่น ๆ ได้
- LCD (มีตัวแปลง I2C) : ต่อขา SDA และ SCL ของตัวแปลง I2C กับขา A4 และ A5 ของ Arduino ตามลำดับ

- Switching Power Supply

สำหรับการเลือกใช้ตัว Switching จะเลือกตัวที่สามารถจ่ายกระแสได้มากกว่าหรือเท่ากับกระแสที่ใช้ในวงจรสามารถจ่ายแรงดันได้ทั้ง 5V DC และ 12V DC จึงเลือกใช้ Switching DC Power Supply รุ่น D-60A สามารถจ่ายแรงดัน 5V 4A และ 12 V 3A



รูปที่ 3.1 การจ่ายไฟด้วย Switching power supply ให้อุปกรณ์ต่างๆ

- ฟิวส์

เป็นอุปกรณ์ที่มีความสำคัญในการช่วยป้องกันระบบไฟฟ้าจากการเกิดสัญญาณไฟฟ้าเกินขนาด กระแสที่ระบบต้องการป้องกันจะมีผลต่อการเลือกใช้ฟิวส์ เนื่องจากต้องให้ฟิวส์มีค่าต้านทานในการส่งผ่านกระแสไฟฟ้าที่เหมาะสมกับกระแสที่ใช้ในวงจร โดยต้องมองหาฟิวส์ที่ค่าใกล้เคียงกับค่าที่คำนวณหากระแสของวงจร

เราสามารถคำนวณการคำนวณกระแสรวมทั้งหมดที่จำเป็นในวงจรได้ดังนี้:

1. ส่วนของ DC ที่ต่อกับ Arduino Nano:

- Arduino Nano : 480 mA
- LCD : 25 mA
- Rotary encoder : 40 mA
- Relay : 70 mA (2 ตัว)

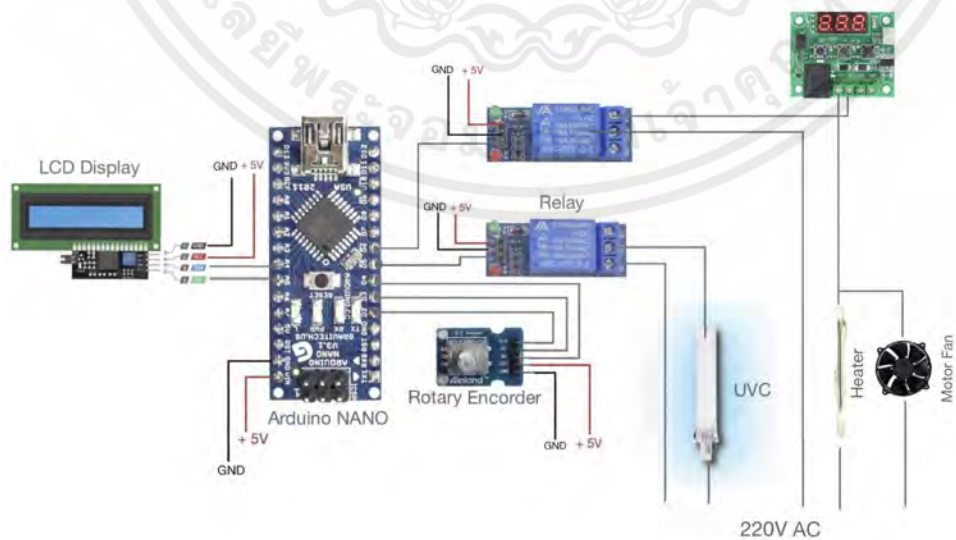
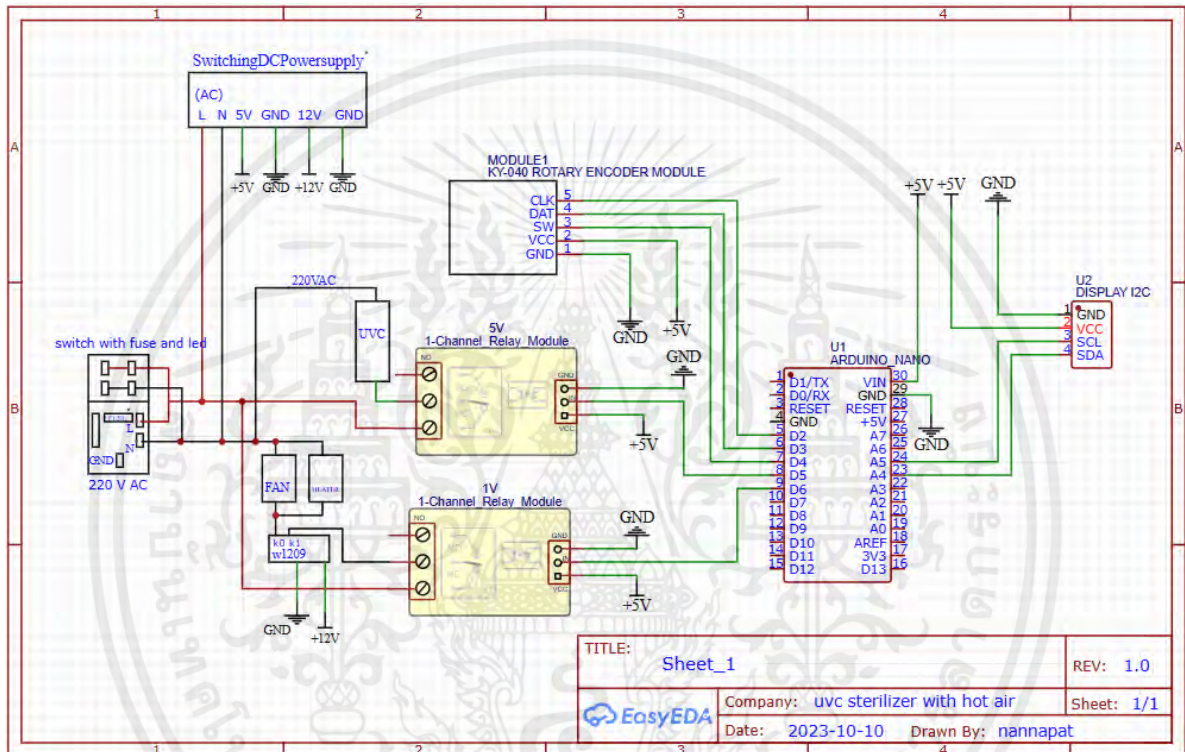
ดังนั้น กระแสรวมที่จะไปที่ Arduino Nano คือ $480 \text{ mA} + 25 \text{ mA} + 40 \text{ mA} + 70 \times 2 \text{ mA} = 685 \text{ mA}$

2. ส่วนของ AC ที่จะไปที่รีเลย์:

- UVC : 30 mA
- W1209 : 80 mA
- ฮีสเตอร์: 500 mA
- พัดลม: 100 mA

ดังนั้น กระแสรวมที่จะไปที่ฮีเตอร์และยูวีซี คือ $30\text{ mA} + 80\text{ mA} + 500\text{ mA} + 100\text{ mA} = 710\text{ mA}$
 คำนวณกระแสรวมทั้งหมดที่จำเป็นในวงจรได้เป็น : $685 + 710 = 1395\text{ mA}$ หรือ 1.39 A เราจึงเลือกใช้ฟิวส์
 เซรามิกขนาด 2 A ใส่ไว้ในสวิตช์ก่อนจ่ายไฟเข้าในวงจร

เมื่อคำนวณหาอุปกรณ์ทั้งหมดที่เราต้องการแล้วจึงนำมาต่อในวงจร ดังรูป



รูปที่ 3.2 การต่อสายไฟภายในตัวกล่อง

3.2.3 ขั้นตอนการทำงาน

ในด้านของซอฟต์แวร์จะใช้โปรแกรม Arduino IDE ในการเขียนคำสั่งลงบอร์ด Arduino โดยจะเขียนให้สามารถทำงานได้ดังนี้

1. เมื่อเริ่มการทำงานจอ lcd จะแสดงผลว่า “ UVC with Hot air Sterilizer ”
2. สามารถเลือกได้ว่าจะเปิดการทำงานของหลอดไฟยูวีซี หรือ ฮีสเตอร์ จากการหมุน Rotary encoder
3. สามารถเลือกโหมดจากการหมุน Rotary encoder โดยมีสองโหมดคือ Manual Mode และ Timer Mode
4. Manual Mode จะต้องกดปุ่มแล้วไฟเปิด-ปิดได้
5. Timer Mode จะต้องกดปุ่มแล้วตั้งเวลาให้ไฟได้ โดยหลังจากกำหนดเวลาเสร็จแล้ว จอ lcd จะนับถอยหลังตามเวลาที่ตั้ง และหลอดไฟต้องดับเมื่อเวลาหมด

3.2.4 การออกแบบตู้อบฆ่าเชื้อ



รูปที่ 3.3 แบบตู้อบฆ่าเชื้อ

ในส่วนของการออกแบบจะใช้ Tinker cad ในการออกแบบ โดยในการประกอบจริงจะใช้วัสดุเป็นไม้กระดานความหนา 4 mm มีขนาดของตู้กว้าง 25 cm ยาว 30 cm และสูง 25 cm สามารถเปิดได้ทั้งด้านหน้า และด้านหลัง โดยด้านหน้าจะมีจอ lcd ,Rotary encoder และกระดกไฟที่สามารถเปิด-ปิด สำหรับเช็คการทำงานของ

หลอด UVC ส่วนด้านหลังจะมีไว้เพื่อใส่หม้อแปลง และสวิตช์ พื้นที่ด้านบนภายในตัวกล่องใช้สำหรับการวางวงจร และพัดลมมอเตอร์ ซึ่งสามารถเชื่อมต่อกับส่วนด้านหลังของกล่องได้ และส่วนที่กว้างที่สุดภายในกล่องจะเป็นส่วนของหลอดUVC อีสี่เตอร์ โดยจะทำการคุมพื้นผิวทั้งหมดด้วยแผ่นกันรังสี

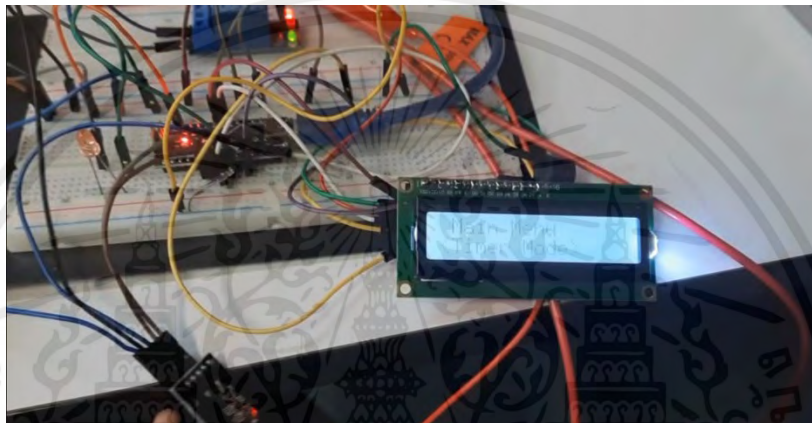


บทที่ 4

ผลการทดลอง

4.1 การทำงานของโปรแกรม

เริ่มต้นทดลองโดยการทดลองต่อวงจรกับ led ขนาดเล็กซึ่งใช้แรงดันแค่ 5 V เมื่อทดลองต่อแล้วผลออกมาตามที่กำหนดไว้ จึงเอามาต่อหลอด UVC และหลอดฮีสเตอร์



รูปที่ 4.1 รูปทดลองต่อวงจรด้วย LED

ผลการทดลองหลังต่อลงกล่องจริง

วงจรสามารถทำงานให้หลอดไฟติดตามคำสั่งที่ลงไว้ได้ โดยจะมีการทำงานดังนี้



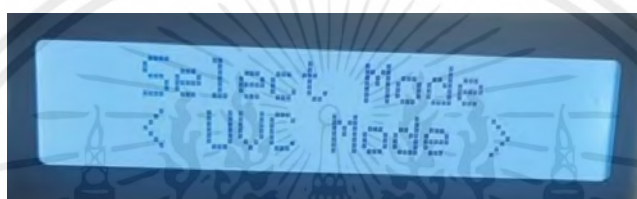
รูปที่ 4.2 ตู้อบฆ่าเชื้อที่ใช้ในการทดลองจริง

เมื่อทำการเสียบปลั๊ก และเปิดสวิตช์ไฟ หน้าจอของ lcd จะปรากฏข้อความดังรูป



รูปที่ 4.3 lcd เมื่อเริ่มการทำงาน

หน้าจอ lcd จะเปลี่ยนไปเป็นจอ Main Menu โดยให้หน้านี้จะสามารถใช้ Rotary Encoder ในการเลือกโหมด เป็นการทำงานระหว่างโหมด UVC และโหมดลมร้อน



รูปที่ 4.4 Select Mode โหมด UVC



รูปที่ 4.5 Select Mode โหมด Hot Air

เมื่อกดเลือกการทำงานได้แล้วหน้าจอ lcd จะเปลี่ยนไปเป็นจอ Main Menu จากนั้นเลือกโหมดเป็น Manual Mode หรือ Timer Mode ได้



รูปที่ 4.6 Main Menu โหมด manual



รูปที่ 4.7 Main Menu โหมด timer

หากต้องการเปลี่ยนโหมดสามารถกดที่ปุ่ม Exit หน้าจอจะกลับไป Select Menu อีกครั้ง



รูปที่ 4.8 Exit จาก Main Menu

- ถ้าเราเลือกเป็น Manual Mode หน้าจอจะสับไปให้เลือกระหว่าง START และ EXIT



รูปที่ 4.9 การทำงานแบบ manual 1

เมื่อกดปุ่ม start หน้าจอจะเปลี่ยนไปเป็น UVC light ON และหลอดไฟก็จะติด และถ้ากดปุ่มอีกครั้ง หน้าจอก็จะเปลี่ยนไปเป็น UV light Off ไฟก็จะดับ และย้อนกลับไปหน้า Main Menu อีกครั้ง



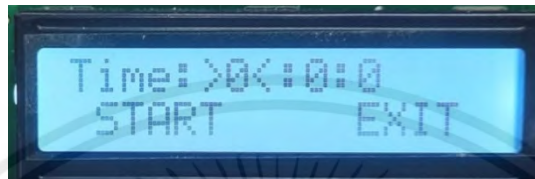
รูปที่ 4.10 การทำงานแบบ manual 2



รูปที่ 4.11 รูป การทำงานแบบ manual 3

เมื่อกด exit หน้าจอจะกลับเป็น Main Menu อีกครั้ง

- ถ้าเราเลือกเป็น Timer Mode จะขึ้นหน้าจอให้สามารถตั้งเวลาได้ สามารถตั้งได้ทั้งชั่วโมง นาที และวินาที เรียงตามลำดับเลข 0 โดยการหมุนรอบแรกในการเลือกตำแหน่งที่ต้องการตั้ง แล้วกดปุ่ม จากนั้นหมุนอีกครั้งเพื่อตั้งเวลาที่ต้องการ



รูปที่ 4.12 การทำงานแบบ timer 1

เมื่อกด exit หน้าจอจะกลับเป็น Main Menu อีกครั้ง

4.2 การทดสอบความสามารถในการฆ่าเชื้อของยูวีซี

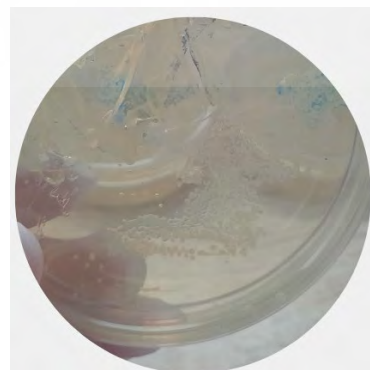
อุปกรณ์ในการทดสอบนั้นมีด้วยกันทั้งหมด 3 อย่าง คือ สำลীগ้าวน จานเพาะเชื้อ Agar และโทรศัพท์มือถือ โดยขั้นแรกนำสำลীগ้าวนมาถูกับโทรศัพท์ จากนั้นนำสำลীগ้าวนไปถูกับจานเพาะเชื้อให้ทั่ว แล้วนำโทรศัพท์เข้าเครื่องอบฆ่าเชื้อ จากนั้นตั้งเวลาและกดปุ่ม Start เพื่อให้เครื่องอบฆ่าเชื้อทำงาน โดยเวลาที่ทดสอบได้แก่ 15 นาที, 30 นาที และ 45 นาที เมื่อเครื่องอบฆ่าเชื้อทำงานถึงเวลาที่ต้องการทดสอบ ให้ทำการกดปุ่ม Pause เพื่อนำโทรศัพท์ออกมา จากนั้นนำสำลীগ้าวนมาถูกับโทรศัพท์ และจานเพาะเชื้อเหมือนกับก่อนที่จะทำการทดสอบ เมื่อถูเสร็จแล้ว ก็นำโทรศัพท์เข้าไปในตู้อบฆ่าเชื้อเช่นเดิม โดยทำอย่างนี้ซ้ำไปเรื่อยๆจนกว่าจะครบเวลาที่ต้องการทดสอบ

- ก่อนใช้เครื่องอบฆ่าเชื้อ

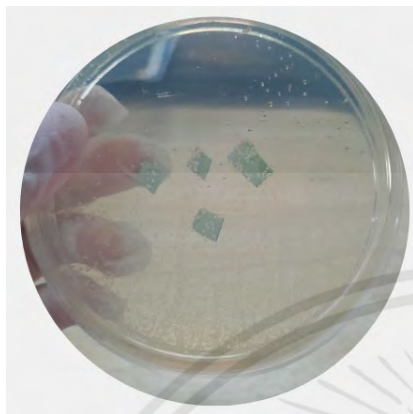
โทรศัพท์เครื่องที่ 1



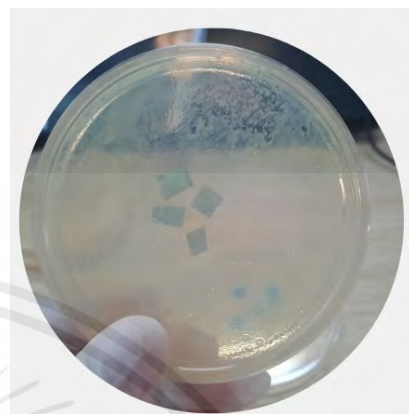
โทรศัพท์เครื่องที่ 2



- ใช้เครื่องอบฆ่าเชื้อเป็นเวลา 15 นาที
โทรศัพท์เครื่องที่ 1



โทรศัพท์เครื่องที่ 2



- ใช้เครื่องอบฆ่าเชื้อเป็นเวลา 30 นาที
โทรศัพท์เครื่องที่ 1



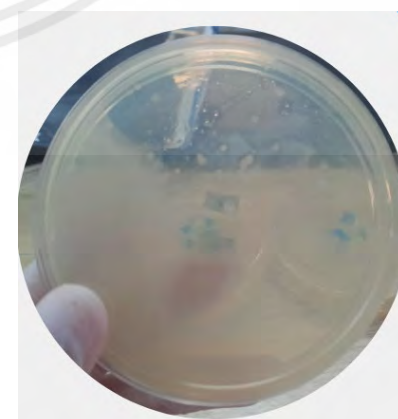
โทรศัพท์เครื่องที่ 2



- ใช้เครื่องอบฆ่าเชื้อเป็นเวลา 45 นาที
โทรศัพท์เครื่องที่ 1



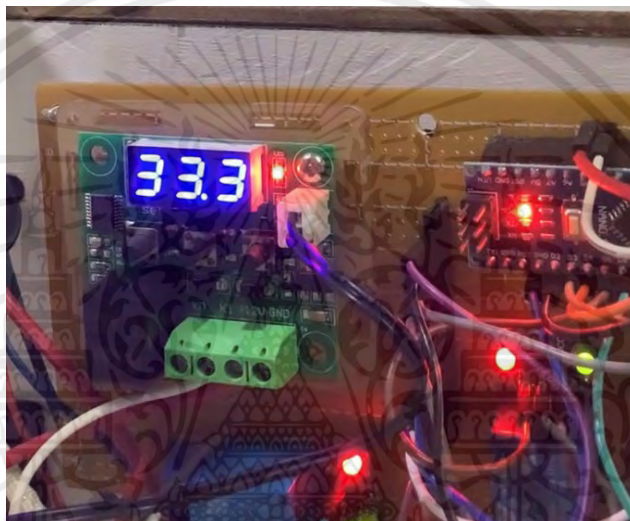
โทรศัพท์เครื่องที่ 2



4.3 การทดสอบความสามารถของตัวควบคุมอุณหภูมิ

ทดสอบอุณหภูมิภายในตู้อบฆ่าเชื้อโดยใช้ตัวควบคุมอุณหภูมิ ซึ่งหลักการของตัวควบคุมอุณหภูมิ คือ เมื่อตัวควบคุมอุณหภูมิตัดได้ว่าภายในตู้อบฆ่าเชื้อมีอุณหภูมิมากกว่าที่ตั้งไว้ 2 องศา ตัวควบคุมจะสั่งให้ฮีสเตอร์หยุดทำงานทันที และเมื่ออุณหภูมิลดลงกว่าที่ตั้งเอาไว้แล้ว 2 องศา ฮีสเตอร์จะกลับมาทำงานอีกครั้ง โดยผู้จัดทำได้ตั้งอุณหภูมิไว้ที่ 34 องศาเซลเซียส

- ก่อนจะถึงค่าอุณหภูมิที่กำหนดไว้ ไฟ led ของตัวควบคุมอุณหภูมิติด ซึ่งแสดงให้เห็นว่าวงจรปล่อยให้กระแสไหลผ่าน หรือก็คือ ฮีสเตอร์และพัดลมระบายความร้อนกำลังทำงานอยู่



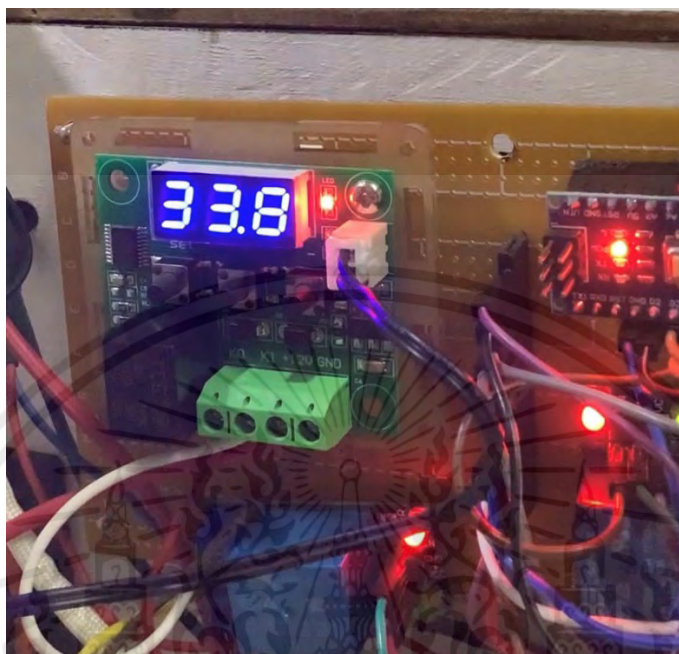
รูปที่ 4.13 การทำงานของตัวควบคุมอุณหภูมิตอนที่ยังไม่เกินอุณหภูมิที่ตั้งไว้

- เมื่อค่าอุณหภูมิเพิ่มขึ้นไปจนถึง 34.2 องศาเซลเซียส ไฟ led จะดับลง ซึ่งแสดงให้เห็นว่าค่าอุณหภูมิเกินกว่าที่กำหนดแล้ว ส่งผลให้รีเลย์ในตัวควบคุมอุณหภูมิต่างงาน เพื่อหยุดกระแสที่ไหลผ่านวงจร



รูปที่ 4.14 รูปตัวควบคุมอุณหภูมิตอนที่อุณหภูมิเกินที่ตั้งไว้

- เมื่ออุณหภูมิลดลงถึง 33.8 องศาเซลเซียสแล้ว ไฟ led จะกลับมาติดอีกครั้ง เพื่อแสดงให้เห็นว่าวงจรปล่อยให้กระแสไหลผ่านแล้ว หรือก็คือ ฮีสเตอร์และพัดลมกลับมาทำงานอีกครั้ง



รูปที่ 4.15 รูปตัวควบคุมอุณหภูมิตอนที่วงจรกลับมาทำงานอีกครั้ง

บทที่ 5

สรุปผลการทดลอง

สรุปการทำปริญญานิพนธ์

1. ส่วนของการเขียนคำสั่งให้วงจร สามารถต่อวงจร และติดตั้งโปรแกรมผ่าน Arduino IDE โดยเครื่องอบยูวีซี และลมร้อนนี้สามารถทำงานได้ตามขอบเขตที่เรากำหนดไว้
 - สามารถใช้โหมด UVC และ โหมดอบลมร้อนได้
 - สามารถตั้งค่าให้เป็นโหมด Manual และโหมด Timer ซึ่งโหมด Timer สามารถตั้งเวลาได้ตั้งแต่หน่วยวินาที ไปจนถึงชั่วโมง และสามารถนับถอยหลังเวลา เปิด-ปิดของหลอดไฟได้อีกด้วย และเพื่อความปลอดภัยที่มากขึ้นของวงจร จึงได้มีการต่อสวิตช์ไฟ และฟิวส์ ก่อนที่จะจ่ายไฟเข้าไปในตู้อบ เพื่อเป็นการตัดกระแส และป้องกันการลัดวงจร
 - สามารถกดหยุดเวลา กดย้อนกลับ และยกเลิกการทำงานระหว่างที่กำลังทำงานได้
2. การทำงานของระบบอบลมร้อน
 - วงจรสามารถทำงาน และหยุดการทำงานได้ด้วยอุณหภูมิที่ได้ตั้งค่าไว้ใน W1209 ทำให้สามารถควบคุมอุณหภูมิภายในกล่องตามที่ต้องการตั้งได้ และยังสามารถเปลี่ยนอุณหภูมิได้ตามที่ต้องการ
3. การทำงานของระบบอบยูวีซี
 - วงจรสามารถฆ่าเชื้อโรคที่เกาะอยู่ตามร่างกายเราได้บางส่วน โดยอ้างอิงจากผลการทดลองที่ใช้วุ้นเลี้ยงเชื้อ เพาะเชื้อที่ได้จากโทรศัพท์มือถือ ก่อนและหลังการเข้าตู้อบฆ่าเชื้อ
4. ความปลอดภัยของวงจร
 - หลังจากการทดลองเปิดการทำงานของฮีสเตอร์ และหลอดยูวีซีทิ้งไว้ 1 ชั่วโมง ไม่พบปัญหาการลัดวงจร หรือกระแสรั่วไหลออกจากวงจร วงจรสามารถทำงานได้เป็นปกติ
 - ก่อนที่ไฟจะเริ่มเข้าในวงจร จะมีสวิตช์ LED พร้อมทั้งใส่ฟิวส์ หลังจากที่ได้คำนวณกระแสในวงจรแล้ว ได้ค่าประมาณ 1.39 A จึงได้กำหนดฟิวส์ที่ทนกระแส 2 A เพื่อป้องกันการเกิดกระแสเวอร์โหลด หรือกระแสลัดวงจร
 - เมื่อหลอดยูวีซี หรือฮีสเตอร์กำลังทำงาน หากผู้ใช้เปิดประตูด้านหน้าของตู้ สวิตช์ที่อยู่ด้านหน้าประตูสามารถตัดวงจรให้หยุดทำงาน เพื่อป้องกันรังสียูวี และความร้อนที่ออกมาจากกล่องได้

ข้อเสนอแนะ

จากการทำปริญญานิพนธ์ในครั้งนี้พบว่า การทดลองในด้านของประสิทธิภาพของการทำงานนั้นยังไม่สามารถวัดประสิทธิภาพได้แน่นอน ความสามารถในการฆ่าเชื้อ อาจจะต้องใช้การทดลองซ้ำหลายๆครั้ง และใช้ตัวทดลองที่หลากหลายขึ้น เพื่อทดลองประสิทธิภาพสูงสุดในการฆ่าเชื้อที่ทำได้ และในส่วนของฮีสเตอร์นั้นใช้เวลาคลายความร้อนภายในกล่องค่อนข้างนาน อาจจะต้องเพิ่มอุปกรณ์ที่ช่วยในการระบายความร้อนให้เร็วขึ้น เพื่อให้สะดวกต่อการนำไปใช้งานต่อ และเพื่อความปลอดภัย



เอกสารอ้างอิง [1]

1. ผศ.พญ.พลอยทราย รัตนเขมากร. (2563). แสงยูวี และข้อควรระวังก่อนใช้หลอด UVC ฆ่าเชื้อโรค. สืบค้นจาก : <https://www.prachachat.net/rama-health/news-46960126> [6 ธันวาคม 2566]
2. บริษัท ทรัพย์วรา เทคโนโลยี จำกัด. (ไม่ทราบปีที่เขียน). หน้าที่ของพัดลมระบายความร้อน (subwara.com). สืบค้นจาก : <https://www.subwara.com/Blog/detail/5> [6 ธันวาคม 2566]
3. SUTHINEE KERDKAEW. (2566). Wokwi : จำลองการทำงานออนไลน์ของ ESP32, Arduino, และ Raspberry Pi Pico. สืบค้นจาก : <https://th.cnx-software.com/2023/04/11/wokwi-simulator-esp32-arduino-raspberry-pi-pico/> [16 ธันวาคม 2566]
4. Weidmüller. (2565). Relay Module D-SERIES อุปกรณ์เปิด-ปิด ภายในวงจรควบคุมอัตโนมัติ. สืบค้นจาก : <https://planet.co.th/en/relay-module-d-series-001/> [23 ธันวาคม 2566]
5. บริษัท ไฮ-เด็น ฮีตเทค จำกัด. (ไม่ทราบปีที่เขียน). Temperature Controller มีหลักการทำงานอย่างไร?. สืบค้นจาก : <http://www.hi-den.co.th/heater-knowledge/145-temperature-controller.html> []
6. Sumipol Agile. (ไม่ทราบปีที่เขียน). Rotary Encoder โรตารี เอ็นโค้ดเดอร์ คือ?. สืบค้นจาก : <https://www.techsourcemarketplace.com/product-category/all-products/rotary-encoder/> [11 มกราคม 2567]
7. Cybertice. (2560). การใช้งานจอ Character LCD กับ Arduino แบบละเอียด. สืบค้นจาก : <https://www.cybertice.com/article/47> [11 มกราคม 2567]
8. SUPPORT THAIEASYELEC. (2560). บทความ Arduino คืออะไร. สืบค้นจาก : <https://blog.thaieasyelec.com/what-is-arduino-ch1/> [11 มกราคม 2567]
9. Misumi Technology Center. (2566). ฮีตเตอร์ คือ อะไร (Heater). สืบค้นจาก : <https://misumitechnical.com/technical/tools/industrial-heating-solutions-engineering/> [11 มกราคม 2567]
10. SUPPORT THAIEASYELEC. (2561). การใช้งาน Character LCD Display กับ Arduino (ตอนที่2 – รูปแบบการเชื่อมต่อแบบ I2C). สืบค้นจาก : <https://blog.thaieasyelec.com/how-to-use-character-lcd-display-arduino-ch2-i2c/> [14 มกราคม 2567]

เอกสารอ้างอิง [2]

11. fitroxelec. (2564). Display - ตอนที่ 3 : Liquid Crystal I2C & Custom Character. สืบค้นจาก : <http://fitrox.lnwshop.com/article/127> [14 มกราคม 2567]
12. primus. (2563). สิ่งที่ต้องรู้ก่อนเลือกซื้อสวิตซิ่งเพาเวอร์ซัพพลาย (Switching Power Supply). สืบค้นจาก : <https://www.primusthai.com/primus/Knowledge/info?ID=253> [14 มกราคม 2567]



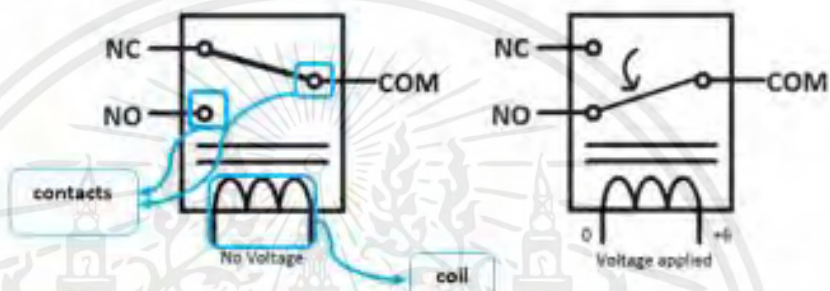


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RELAY MODULES

RELAY WORKING IDEA

Relays consist of three pins normally open pin , normally closed pin, common pin and coil. When coil power on magnetic field is generated the contacts connected to each other.

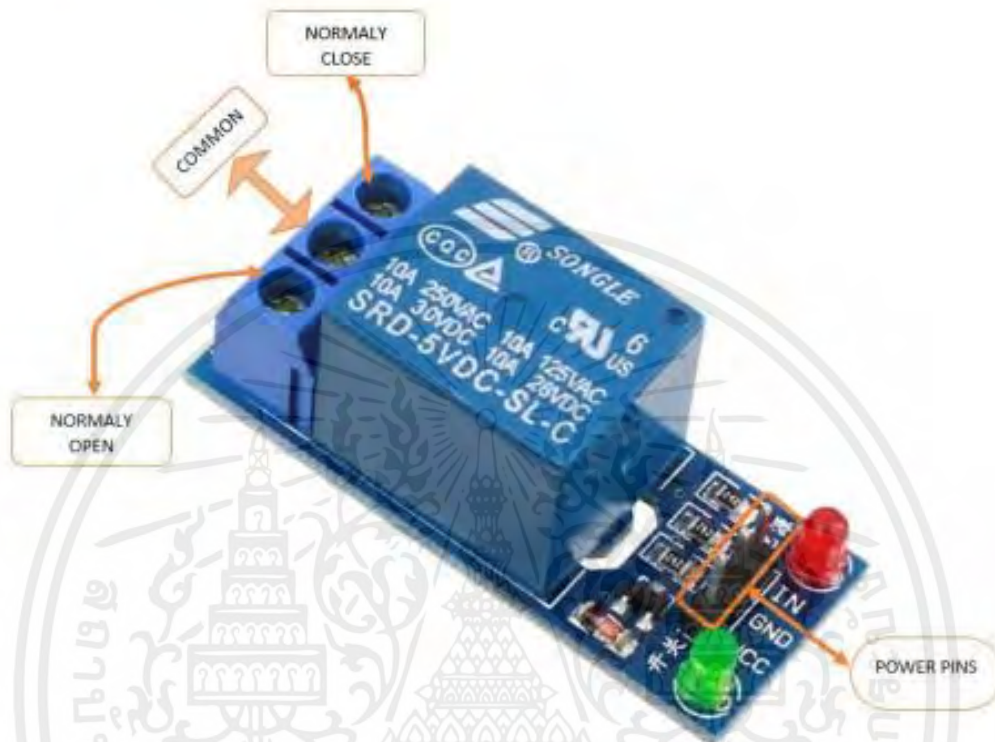


Relay modules 1-channel features

- Contact current 10A and 250V AC or 30V DC.
- Each channel has indication LED.
- Coil voltage 12V per channel.
- Kit operating voltage 5-12 V
- Input signal 3-5 V for each channel.
- Three pins for normally open and closed for each channel.

How to connect relay module with Arduino

As shown in relay working idea it depends on magnetic field generated from the coil so there is power isolation between the coil and the switching pins so coils can be easily powered from Arduino by connecting VCC and GND pins from Arduino kit to the relay module kit after that we choose Arduino output pins depending on the number of relays needed in project designed and set these pins to output and make it out high (5 V) to control the coil that allow controlling of switching process.



NOTE : whatever was the relay channels number the pinconfiguration is the same for every channel except the power pins (VCC and GND) are for the board itself. The input signal (IN) pin for every relay.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

KY-040 Rotary encoder

Encoder

1) Rotation: _____

2) Distance Rot.: _____

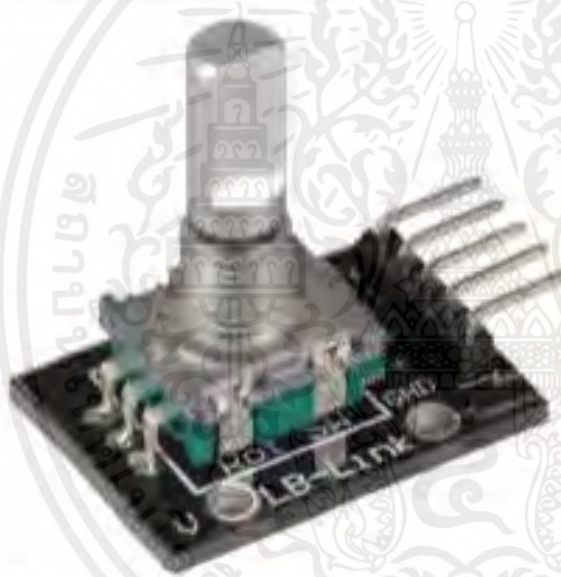
3) Frequency: _____

4) Phase: _____

5) Data example: _____

6) Data example frequency: _____

Picture



Technical data / Short description

The current position of the rotary switch will be sent encoded in the output.

Encoding

The idea of the rotary switch is that with every "step" only one of the conditions will change. In order of which status have changed first, you can see the rotational direction if you look at the following encoding.

Clockwise [A will change first] → **Pin_CLK**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

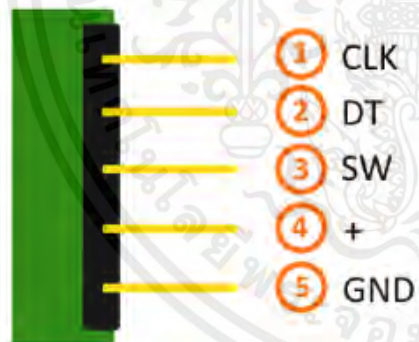
KY-040 Rotary encoder

A	B
0	0
1	0
1	1
0	1
0	0

Counterclockwise [B will change first] → Pin_DT

A	B
0	0
0	1
1	1
1	0
0	0

Pinout



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



I2C Serial Interface 1602 LCD Module

This is I2C interface 16x2 LCD display module, a high-quality 2 line 16 character LCD module with on-board contrast control adjustment, backlight and I2C communication interface. For Arduino beginners, no more cumbersome and complex LCD driver circuit connection. The real significance advantages of this I2C Serial LCD module will simplify the circuit connection, save some I/O pins on Arduino board, simplified firmware development with widely available Arduino library.

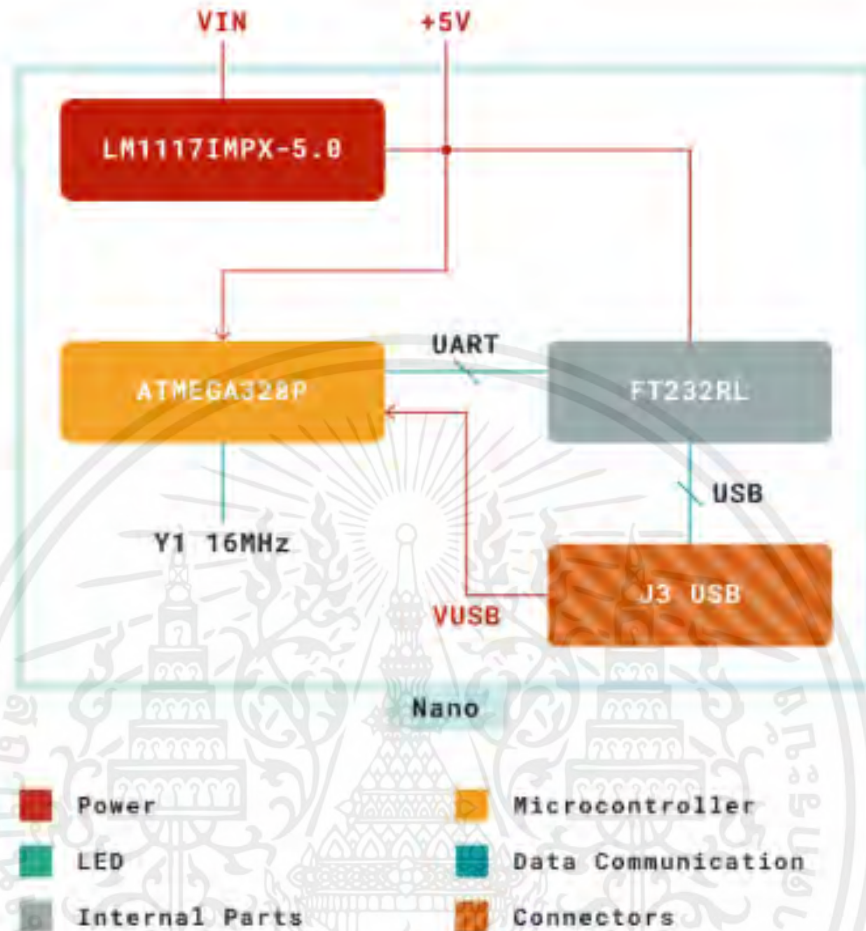


SKU: [DSP-1182](#)

Brief Data:

- Compatible with Arduino Board or other controller board with I2C bus.
- Display Type: Negative white on Blue backlight.
- I2C Address: 0x38-0x3F (0x3F default)
- Supply voltage: 5V
- Interface: I2C to 4bits LCD data and control lines.
- Contrast Adjustment: built-in Potentiometer.
- Backlight Control: Firmware or jumper wire.
- Board Size: 80x36 mm.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Block Diagram of Arduino Nano

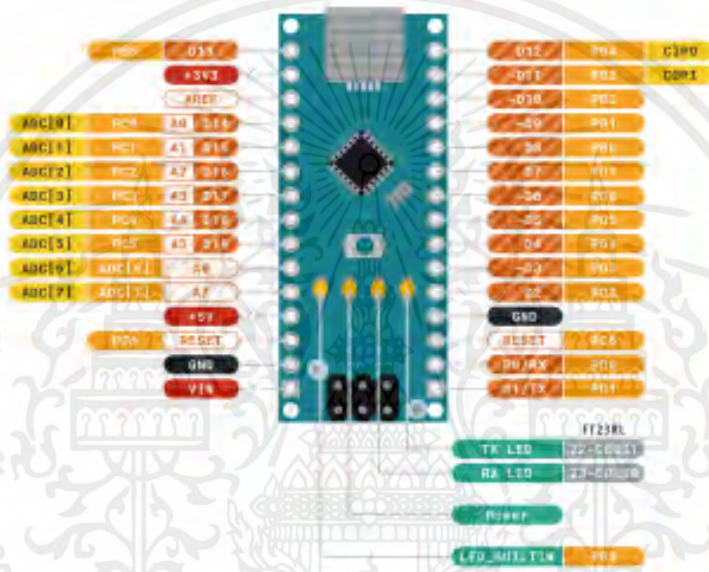
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



5 Connector Pinouts



ARDUINO
NANO



- Ground
- Internal Pin
- Digital Pin
- Microcontroller's Port
- Power
- SND Pin
- Analog Pin
- LED
- Other Pin
- Default

Power Tree of Arduino Nano

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



5.1 Analog

Pin	Function	Type	Description
1	+3V3	Power	5V USB Power
2	A0	Analog	Analog input 0 /GPIO
3	A1	Analog	Analog input 1 /GPIO
4	A2	Analog	Analog input 2 /GPIO
5	A3	Analog	Analog input 3 /GPIO
6	A4	Analog	Analog input 4 /GPIO
7	A5	Analog	Analog input 5 /GPIO
8	A6	Analog	Analog input 6 /GPIO
9	A7	Analog	Analog input 7 /GPIO
10	+5V	Power	+5V Power Rail
11	Reset	Reset	Reset
12	GND	Power	Ground
12	VIN	Power	Voltage Input

5.2 Digital

Pin	Function	Type	Description
1	D1/TX1	Digital	Digital Input 1 /GPIO
2	D0/RX0	Digital	Digital Input 0 /GPIO
3	D2	Digital	Digital Input 2 /GPIO
4	D3	Digital	Digital Input 3 /GPIO
5	D4	Digital	Digital Input 4 /GPIO
6	D5	Digital	Digital Input 5 /GPIO
7	D6	Digital	Digital Input 6 /GPIO
8	D7	Digital	Digital Input 7 /GPIO
9	D8	Digital	Digital Input 8 /GPIO
10	D9	Digital	Digital Input 9 /GPIO
11	D10	Digital	Digital Input 10 /GPIO
12	D11	Digital	Digital Input 11 /GPIO
13	D12	Digital	Digital Input 12 /GPIO
14	D13	Digital	Digital Input 13 /GPIO
15	Reset	Reset	Reset
16	GND	Power	Ground

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

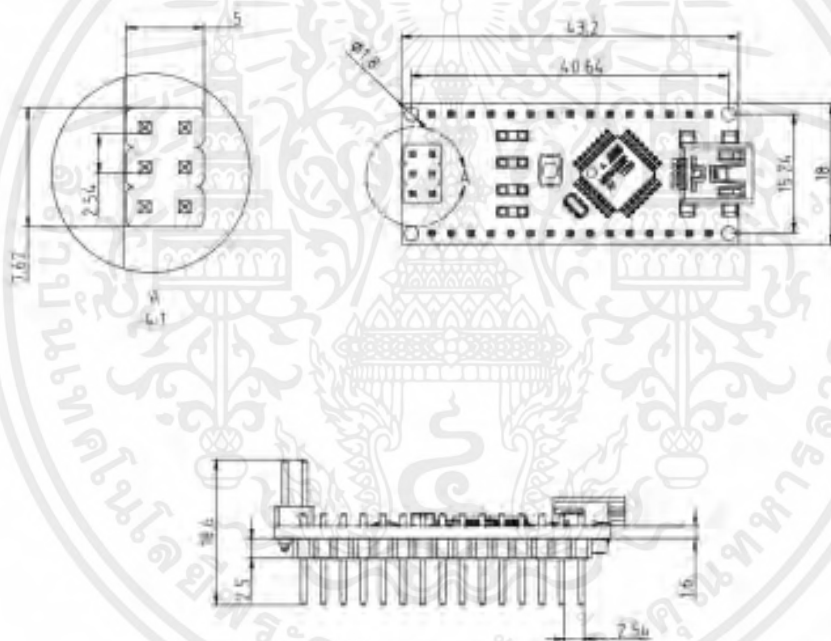


5.3 ATmega328

Pin	Function	Type	Description
1	PB0	Internal	Serial Wire Debug
2	PB1	Internal	Serial Wire Debug
3	PB2	Internal	Serial Wire Debug
4	PB3	Internal	Serial Wire Debug
5	PB4	Internal	Serial Wire Debug
6	PB5	Internal	Serial Wire Debug

6 Mechanical Information

ARDUINO
NANO
Size



2020/11/19

Mechanical dimensions of Arduino Nano

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้