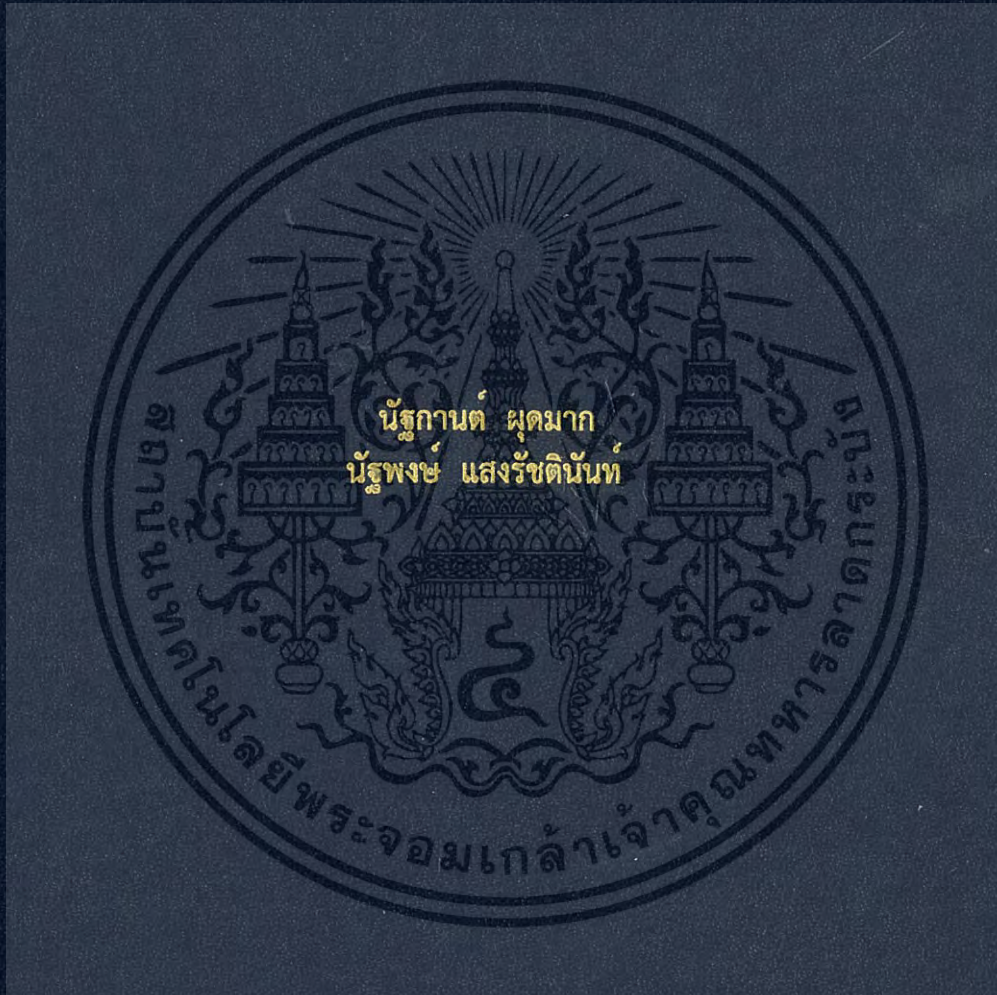


การปรับปรุงประสิทธิภาพของเครื่องเขย่าแบบควบคุมอุณหภูมิและแสงส่องสว่าง
EFFICIENCY IMPROVEMENT FOR ILLUMINATING INCUBATOR SHAKER



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

การปรับปรุงประสิทธิภาพของเครื่องเขย่าแบบควบคุมอุณหภูมิและแสงส่องสว่าง
EFFICIENCY IMPROVEMENT FOR ILLUMINATING INCUBATOR SHAKER



T143872



นัฐกานต์ ผุดมาก
นัฐพงษ์ แสงรัชตินันท์

ร.พ.

๒๖๓๘๗๗

๒๐๖๘

เลขหมู่.....

เลขทะเบียน 143872

วันเดือนปี 04 ต.ค. 2559

๖๐๐๒๖๖๗๘



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2558

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EFFICIENCY IMPROVEMENT FOR ILLUMINATING INCUBATOR SHAKER

Nattakan Pudmark
Nattapong Sangrattinan



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2015

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2558
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาานิพนธ์

หัวข้อปริญญาานิพนธ์ การปรับปรุงประสิทธิภาพของเครื่องเขย่าแบบควบคุมอุณหภูมิและแสงส่องสว่าง
EFFICIENCY IMPROVEMENT FOR ILLUMINATING INCUBATOR SHAKER
นักศึกษาผู้จัดทำ นางสาวนัฐกานต์ ผุดมาก รหัสนักศึกษา 55010641
นายรัฐพงษ์ แสงรัชตินันท์ รหัสนักศึกษา 55010644
ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2558

อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
ผศ.สุธรรม สัทธรรมสกุล	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การปรับปรุงประสิทธิภาพของเครื่องเขย่าแบบควบคุมอุณหภูมิและแสงส่องสว่าง EFFICIENCY IMPROVEMENT FOR ILLUMINATING INCUBATOR SHAKER	
นักศึกษาผู้จัดทำ	นางสาวนัฐกานต์ ผุดมาก	รหัสนักศึกษา 55010641
	นายรัฐพงษ์ แสงรัชตินันท์	รหัสนักศึกษา 55010644
อาจารย์ที่ปรึกษา	ผศ.สุธรรม สัทธรรมสกุล	
ปีการศึกษา	2558	

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการปรับปรุงประสิทธิภาพของเครื่องเขย่าแบบควบคุมอุณหภูมิและแสงส่องสว่างที่ใช้บอร์ดอาร์ดูโนเมกา 2560 เป็นตัวควบคุม ซึ่งมีการปรับปรุงประสิทธิภาพโดยการเพิ่มส่วนของวงจรควบคุมแรงดันไฟฟ้าที่ป้อนให้กับบัลลาสต์อิเล็กทรอนิกส์ในรูปแบบของแรงดันต่อเนื่องในช่วง 0 ถึง 10 โวลต์ จากเดิมที่ใช้เป็นสัญญาณแบบ PWM (Pulse Width Modulation) ซึ่งส่งผลทำให้แสงที่ได้มีการกระพริบ และได้เพิ่มโปรแกรมในส่วนของการสอบเทียบความสว่างของแสงเข้าไปเนื่องจากเมื่อใช้งานไปซักระยะหนึ่ง อาจเกิดความคลาดเคลื่อนของแสงที่วัดได้ รวมถึงได้ทำการแก้ไขโปรแกรมควบคุมเดิมเพื่อแก้ปัญหาเครื่องหยุดทำงานในบางครั้ง และสุดท้ายจะมีการเพิ่มเติมในส่วนของระบบทำความเย็นเพื่อรักษาอุณหภูมิให้คงที่ในช่วงที่อากาศร้อนเกินกว่าค่าที่ตั้งไว้

Thesis Title	EFFICIENCY IMPROVEMENT FOR ILLUMINATING INCUBATOR SHAKER
Authors	Miss Nattakan Pudmark Mr. Nattapong Sangrattinan
Thesis Advisor	Asst. Prof. Sutham Satthamsakul
Year	2015

ABSTRACT

This project was to improve efficiency of thermostatic shaker and lighting with Arduino MEGA 2560 controlled board. Efficiency improving by addition of voltage controlling circuits, which supplied 0 to 10 volts to electronic ballast instead of PWM signals, which results in rippling light. In addition, we added in program for lighting calibration, when it is used for a long time, it could have an error in lighting measurement. So, we added lighting calibration in program, which sometime stopped working. Finally, we added cooling system for maintaining temperature, when it was hotter than setting.

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้จะไม่ประสบความสำเร็จล่วงไปได้ด้วยดี ถ้าขาดคำแนะนำ คำปรึกษา และความเอาใจใส่เป็นอย่างดีจาก ผศ.สุธรรม สัทธรรมสกุล ซึ่งเป็นอาจารย์ที่ปรึกษาปริญญาบัตรของคณะผู้จัดทำ ซึ่งคณะผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณท่านเป็นอย่างสูง

ขอขอบคุณเพื่อน ๆ ทุกคนในภาควิชาวิศวกรรมการวัดคุมที่ร่วมกันฝ่าฟันอุปสรรคและให้กำลังใจในการทำงานสำเร็จล่วงไปได้ด้วยดี

สุดท้ายนี้ขอกราบขอบพระคุณบิดา มารดา ของคณะผู้จัดทำทุกท่าน ซึ่งเป็นผู้ที่มีชีวิตมอบการศึกษา และอนาคตที่ดี คอยสนับสนุน ตลอดจนให้คำปรึกษาและความช่วยเหลือในด้านต่าง ๆ ซึ่งทำให้ปริญญาบัตรฉบับนี้สำเร็จล่วงไปได้ด้วยดี

คุณค่าและประโยชน์อันพึงมีจากปริญญาบัตรฉบับนี้ทั้งหมด ทางคณะผู้จัดทำขอมอบให้แก่ผู้มีพระคุณทุก ๆ ท่าน

คณะผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญภาพ.....	IX
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญของปริญญาโท.....	1
1.2 วัตถุประสงค์ของปริญญาโท.....	1
1.3 ขอบเขตของปริญญาโท.....	2
1.4 ขั้นตอนการดำเนินการ.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง.....	4
2.1 บทนำ.....	4
2.2 บอร์ดอาร์ดูโนเมกา 2560 (Arduino Mega 2560).....	4
2.2.1 การจ่ายพลังงาน.....	5
2.2.2 หน่วยความจำ.....	5
2.2.3 อินพุตและเอาต์พุต.....	5
2.2.4 การสื่อสาร.....	6
2.3 การสื่อสารด้วยพอร์ต UART.....	6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ IV ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.13 ไอซี PCF8591.....	20
2.13.1 การอ้างถึงตำแหน่งของ PCF8591.....	22
บทที่ 3 การออกแบบ.....	24
3.1 การออกแบบและการทำงานโดยรวมของระบบ.....	24
3.2 การออกแบบแผนผังการทำงานของระบบ.....	24
3.2.1 โครงสร้างการทำงานหลักของระบบ.....	26
3.2.2 โครงสร้างการทำงานของฟังก์ชันย่อยระบบการสอบเทียบแสงส่องสว่าง.....	26
3.2.3 โครงสร้างการทำงานของฟังก์ชันการทำงานตามสถานะ.....	30
3.2.4 โครงสร้างการทำงานของฟังก์ชันย่อยระบบควบคุมแสงส่องสว่าง.....	31
3.2.5 โครงสร้างการทำงานของฟังก์ชันย่อยระบบควบคุมอุณหภูมิ.....	32
3.2.5.1 โครงสร้างการทำงานของฟังก์ชันย่อยระบบควบคุมอุณหภูมิในส่วน การจำกัดการอ่านค่าเซนเซอร์ตามที่ต้องการ.....	33
3.3 การออกแบบหน้าจอสำหรับใช้งาน.....	34
3.3.1 โครงสร้างการเชื่อมโยงของหน้าจอระบบสัมผัส.....	35
3.3.2 หน้าจอเริ่มต้น.....	35
3.3.3 หน้าจอแสดงผล.....	36
3.3.4 หน้าจอตั้งค่าแสงส่องสว่าง.....	36
3.3.5 หน้าจอตั้งค่าการสอบเทียบแสงส่องสว่าง.....	37
3.3.6 หน้าจอตั้งค่าอุณหภูมิ.....	37
บทที่ 4 การดำเนินงาน.....	38
4.1 บทนำ.....	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ VI ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
4.2 ระบบควบคุมแสงส่องสว่าง	38
4.2.1 โครงสร้างระบบควบคุมแสงส่องสว่าง.....	38
4.2.2 การรับค่าความสว่างจากเซนเซอร์วัดแสง (BH1750FVI)	39
4.2.3 การใช้งานบัลลาสต์อิเล็กทรอนิกส์กับหลอดฟลูออเรสเซนต์.....	40
4.2.4 วงจรแปลงแรงดัน.....	40
4.2.5 วงจรแปลงสัญญาณทางดิจิทัลเป็นแอนะล็อก.....	41
4.2.6 ฟังก์ชันการสอบเทียบแสงส่องสว่าง	41
4.3 ระบบควบคุมอุณหภูมิ.....	41
4.3.1 โครงสร้างของระบบควบคุมอุณหภูมิ.....	42
4.3.2 การรับค่าอุณหภูมิจากเซนเซอร์อุณหภูมิ (DS18B20).....	42
4.3.3 การเพิ่มหรือลดอุณหภูมิภายในระบบด้วยฮีตเตอร์ คูลเลอร์และพัดลมระบาย อากาศ.....	43
บทที่ 5 ผลการทดลอง	44
5.1 บทนำ.....	44
5.2 การทดลองและผลการทดลองการทำงานของระบบควบคุมอุณหภูมิ.....	44
5.2.1 การทดลองการทำงานของระบบควบคุมอุณหภูมิ.....	44
5.2.2 ผลการทดลองการทำงานของระบบควบคุมอุณหภูมิ.....	44
5.3 การทดลองและผลการทดลองการทำงานของระบบควบคุมแสงส่องสว่าง	46
5.3.1 การทดลองการทำงานของระบบควบคุมแสงส่องสว่าง	46
5.3.2 ผลการทดลองการทำงานของระบบควบคุมแสงส่องสว่าง	47
5.4 การทดลองและผลการทดลองการทำงานของระบบควบคุมแสงส่องสว่างร่วมกับ อุณหภูมิ.....	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา VII อ่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
5.4.1 การทดลองการทำงานของระบบควบคุมแสงส่องสว่างร่วมกับอุณหภูมิ	47
5.4.2 ผลการทดลองการทำงานของระบบควบคุมแสงส่องสว่างร่วมกับอุณหภูมิ.....	48
บทที่ 6 สรุปผล ปัญหาและข้อเสนอแนะ	49
6.1 สรุปผล.....	49
6.2 ปัญหาและแนวทางแก้ไข.....	49
6.3 ข้อเสนอแนะ	50
บรรณานุกรม.....	51
ภาคผนวก.....	52

สารบัญรูป

รูปที่	หน้า
2.1 อุปกรณ์อาร์ดูโนเมกา 2560.....	4
2.2 รูปแบบการสื่อสารของ TTL และ RS232.....	7
2.3 รูปแบบการสื่อสารของ UART.....	7
2.4 การใช้บัสสายสัญญาณ 1 เส้นต่อร่วมกับอุปกรณ์ DS18B20 หลายตัว.....	11
2.5 การเชื่อมต่อระหว่างจอสัมผัสกับโฮสต์.....	12
2.6 โครงสร้างและขาของ DS18B20.....	13
2.7 การติดต่อสื่อสารแบบสายสัญญาณ 1 เส้นกับ DS18B20.....	14
2.8 การเขียนข้อมูลลงเซนเซอร์อุณหภูมิ DS18B20.....	14
2.9 การอ่านข้อมูลจากบัสสัญญาณ 1 เส้น.....	15
2.10 บล็อกไดอะแกรมของ BH1750FVI.....	16
2.11 โครงสร้างภายในของหลอดฟลูออเรสเซนต์.....	16
2.12 การต่อหลอดฟลูออเรสเซนต์เข้ากับบัลลาสต์ธรรมดา.....	17
2.13 การต่อหลอดฟลูออเรสเซนต์เข้ากับบัลลาสต์อิเล็กทรอนิกส์แบบ 2 หลอด.....	17
2.14 ออปป์แอมป์เบอร์ LM741.....	18
2.15 สัญลักษณ์ของออปป์แอมป์.....	19
2.16 วงจรภายในออปป์แอมป์.....	19
2.17 วงจรสมมูลอย่างง่ายของออปป์แอมป์..... และวงจรสมมูลอย่างง่ายของออปป์แอมป์ในอุดมคติ	19
2.18 ไอซี 555.....	20
2.19 บล็อกไดอะแกรมของไอซี A/D..... และ D/A PCF8591 (8-bit A/D and D/A Converter)	21
2.20 การต่ออุปกรณ์มาสเตอร์และอุปกรณ์สเลฟบนระบบบัสแบบ I ² C	22
2.21 สัญญาณความถี่ SCL และ SDA..... ในสภาวะต่างๆของการสื่อสารข้อมูลบนบัสแบบ I ² C	22
2.22 การจัดขาของ PCF8591.....	22
2.23 รูปแบบการอ้างถึงไอซี PCF8591.....	22
2.24 โหมดการทำงานต่างๆ ของ PCF8591.....	23
3.1 บล็อกไดอะแกรมโครงสร้างการทำงานหลักของระบบเดิม.....	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์อื่นใด

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ IX ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.2 บล็อกไดอะแกรมโครงสร้างการทำงานหลักของระบบใหม่.....	25
3.3 โครงสร้างการทำงานหลักของระบบ.....	26
3.4 โครงสร้างการทำงานของฟังก์ชันย่อยระบบการสอบเทียบแสงส่องสว่าง.....	28
3.4 โครงสร้างการทำงานของฟังก์ชันย่อยระบบการสอบเทียบแสงส่องสว่าง (ต่อ).....	29
3.5 โครงสร้างการทำงานของฟังก์ชันการทำงานตามสถานะ.....	30
3.6 โครงสร้างการทำงานของฟังก์ชันย่อยระบบควบคุมแสงส่องสว่าง.....	31
3.7 โครงสร้างการทำงานของฟังก์ชันย่อยระบบควบคุมอุณหภูมิ.....	32
3.8 โครงสร้างการทำงานของฟังก์ชันย่อยระบบควบคุมอุณหภูมิ.....	34
ในส่วนการจำกัดการอ่านค่าเซนเซอร์ตามที่ต้องการ	
3.9 การเชื่อมโยงของหน้าจอระบบสัมผัส.....	35
3.10 หน้าจอเริ่มต้น	35
3.11 หน้าจอแสดงผล.....	36
3.12 หน้าจอตั้งค่าแสงส่องสว่าง.....	36
3.13 หน้าจอตั้งค่าการสอบเทียบแสงส่องสว่าง.....	37
3.14 หน้าจอตั้งค่าอุณหภูมิ.....	37
4.1 โครงสร้างของระบบควบคุมแสงส่องสว่าง.....	39
4.2 การเชื่อมต่อ BH1750FVI เข้ากับบอร์ดอาร์ดูโน.....	39
4.3 การเชื่อมต่อบัลลาสต์อิเล็กทรอนิกส์กับหลอดฟลูออเรสเซนต์.....	40
4.4 การเชื่อมต่อวงจรแปลงแรงดัน.....	40
4.5 การเชื่อมต่อวงจรแปลงสัญญาณทางดิจิทัลเป็นแอนะล็อก.....	41
4.6 โครงสร้างของระบบควบคุมอุณหภูมิ.....	42
4.7 การเชื่อมต่อเซนเซอร์อุณหภูมิเข้ากับบอร์ดอาร์ดูโน.....	42
4.8 ทดลองใช้งานคูลเลอร์.....	43
4.9 ทดลองใช้งานพัดลมระบายอากาศแทน.....	43
5.1 ผลการทดลองการทำงานของระบบควบคุมอุณหภูมิที่ 33 องศาเซลเซียส.....	45
5.2 ผลการทดลองการทำงานของระบบควบคุมอุณหภูมิที่ 36 องศาเซลเซียส.....	45
5.3 ผลการทดลองการทำงานของระบบควบคุมอุณหภูมิที่ 39 องศาเซลเซียส.....	46
5.4 ผลการทดลองการทำงานของระบบควบคุมแสงส่องสว่าง.....	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปเผยแพร่ให้ผู้อื่นเห็นโดยไม่ได้รับอนุญาตให้ถือว่าผิดกฎหมาย

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ X ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
5.5 ผลการทดลองการทำงานของระบบควบคุมการสอบเทียบแสงส่องสว่าง.....	47
5.6 ผลการทดลองการทำงานของระบบควบคุมแสงส่องสว่าง.....	48

ร่วมกับอนุภูมิที่ 36 องศาเซลเซียส



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา **XI** ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

เนื่องจากโครงการเครื่องเขย่าแบบควบคุมอุณหภูมิและแสงส่องสว่าง เป็นโครงการที่ถูกสร้างขึ้นเพื่อนำไปใช้กับงานวิจัยเพาะเลี้ยงสาหร่ายในเขตร้อนเพื่อผลิตไฮโดรเจน ของทางคณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งหลังจากการใช้งานจริงแล้วทำให้ทราบว่า มีข้อบกพร่องหลายอย่างเกิดขึ้น ไม่ว่าจะเป็นเรื่องของความคลาดเคลื่อนในการวัดค่าความสว่างของแสง ซึ่งส่วนหนึ่งน่าจะมีผลกระทบมาจากแรงดันไฟฟ้าที่ใช้ในการควบคุมบัลลาสต์อิเล็กทรอนิกส์ที่ใช้เป็นแบบ PWM (Pulse Width Modulation) เนื่องจากตัวบัลลาสต์เองนั้นต้องการแรงดันควบคุมในช่วง 0 ถึง 10 โวลต์ อีกทั้งในบางครั้งเมื่อทำงานไปได้ชั่วระยะเวลาหนึ่ง เครื่องจะเกิดอาการค้าง ซึ่งน่าจะเกิดจากโปรแกรมควบคุมที่เขียนอาจไม่สมบูรณ์ รวมถึงตัวเครื่องนั้นนำไปใช้ในห้องปฏิบัติการที่เปิดโล่งไม่มีเครื่องปรับอากาศ ดังนั้นหากใช้งานในวันที่อากาศร้อนก็จะพบกับปัญหาเนื่องจากไม่สามารถควบคุมอุณหภูมิให้อยู่ในช่วงที่ต้องการใช้งานได้ จึงมีแนวทางในการแก้ไขคือติดตั้งอุปกรณ์ทำความเย็นเพิ่มเติม เพื่อควบคุมอุณหภูมิภายในตัวเครื่องเขย่าให้อยู่ในช่วงที่ตั้งค่าอุณหภูมิไว้

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. ศึกษาการใช้งาน และเขียนโปรแกรมควบคุมอุปกรณ์ต่อพ่วงต่าง ๆ กับบอร์ดอาร์ดูโน
2. ศึกษาการใช้งาน และโปรแกรมควบคุมการทำงานของหน้าจอแบบสัมผัสร่วมกับบอร์ดอาร์ดูโน
3. ศึกษาและแก้ไขการส่งสัญญาณควบคุมบัลลาสต์อิเล็กทรอนิกส์เพื่อให้เอาต์พุต ที่ได้ออกมาในรูปแบบของสัญญาณแอนะล็อก 0 ถึง 10 โวลต์
4. ศึกษาการทำงานของเซนเซอร์ตรวจวัดความเข้มของแสง รวมถึงการเชื่อมต่อและเขียนโปรแกรมควบคุมความเข้มแสง และเพิ่มฟังก์ชันการสอบเทียบแสงส่องสว่าง (Calibration) เพื่อหาค่าแก้ (Correction) มาชดเชยกับค่าความผิดพลาดในกระบวนการ
5. ศึกษาการทำงานของเซนเซอร์ตรวจวัดอุณหภูมิ รวมถึงการเชื่อมต่อและเขียนโปรแกรมเพื่อรักษาอุณหภูมิให้คงที่โดยการเพิ่มระบบทำความเย็น (cooling)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของปริญญาโท

1. ตัวเครื่องสามารถแสดงความเข้มแสงได้เสถียรกว่าเดิมเนื่องจากแก้ไขโดยการทำวงจรแปลงสัญญาณจากดิจิทัลเป็นแอนะล็อก (DAC) เพื่อควบคุมบัลลาสต์อิเล็กทรอนิกส์แทนสัญญาณ PWM (Pulse Width Modulation) เดิม
2. ตัวเครื่องสามารถตรวจวัดอุณหภูมิภายในเครื่อง รวมถึงสามารถควบคุม และรักษาระดับของอุณหภูมิให้อยู่ช่วงที่ใกล้เคียงกับอุณหภูมิที่ตั้งไว้ได้
3. ตัวเครื่องสามารถตรวจวัดอุณหภูมิภายในตัวเครื่อง รวมถึงสามารถควบคุม และรักษาระดับของอุณหภูมิให้อยู่ช่วงที่ใกล้เคียงกับอุณหภูมิที่ตั้งไว้ได้โดยการเพิ่มระบบทำความเย็น (cooling)
4. ตัวเครื่องมีโหมดการทำงานที่สามารถทำการสอบเทียบตัวอุปกรณ์ที่ทำการวัดความเข้มแสง และแก้ไขค่าที่อ่านได้ให้มีความถูกต้องและแม่นยำมากขึ้น
5. ตัวเครื่องเข่าสามารถทำงานได้อย่างเป็นปกติ และไม่หยุดทำงานโดยไม่มีคำสั่งงานจากผู้ใช้งาน

1.4 ขั้นตอนการดำเนินการ

1. ทำการศึกษาทฤษฎีและหลักการที่เกี่ยวข้องกับโครงการในเบื้องต้น
2. ศึกษาและค้นคว้าหาข้อมูลเกี่ยวกับอุปกรณ์ต่าง ๆ ที่จะนำมาใช้ในการออกแบบและปรับปรุงประสิทธิภาพของเครื่องเข่าแบบควบคุมอุณหภูมิและแสงส่องสว่าง
3. วางแผนการ และออกแบบรายละเอียดในส่วนของการปรับปรุงประสิทธิภาพของเครื่องเข่าแบบควบคุมอุณหภูมิและแสงส่องสว่าง
4. ศึกษาโปรแกรมสำหรับควบคุมบอร์ดอาร์ดูโน
5. ศึกษาการใช้งานและการเขียนโปรแกรมแปลงสัญญาณจากดิจิทัลเป็นแอนะล็อก (DAC) เพื่อทดแทนการใช้สัญญาณ PWM (Pulse Width Modulation) เดิม
6. ศึกษาการใช้งานและเขียนโปรแกรมควบคุมแสงส่องสว่าง พร้อมทั้งเขียนโปรแกรมสำหรับสอบเทียบแสงส่องสว่างเพื่อทำการแก้ไขให้อ่านค่าได้อย่างถูกต้อง
7. ศึกษาการใช้งาน และเขียนโปรแกรมเพื่อรักษาอุณหภูมิให้คงที่โดยการจำกัดการรับค่าจากเซนเซอร์อุณหภูมิให้อยู่ในช่วงที่ต้องการ
8. ศึกษาการใช้งานของระบบทำความเย็น (cooling) และเขียนโปรแกรมเพื่อรักษาอุณหภูมิให้คงที่
9. สรุปผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ความรู้และประสบการณ์จากการใช้งาน และการเขียนโปรแกรมควบคุม รวมถึงฟังก์ชันต่าง ๆ ของอาร์ดูโน
2. ได้ความรู้และประสบการณ์จากการใช้งาน และเขียนโปรแกรมควบคุมการทำงานหน้าจอแบบสัมผัสร่วมกับอาร์ดูโน
3. ได้ความรู้และประสบการณ์จากการเขียนโปรแกรม เพื่อเชื่อมต่อกับอุปกรณ์ตรวจวัดอุณหภูมิ และควบคุมเพื่อรักษาอุณหภูมิให้คงที่
4. ได้ความรู้และประสบการณ์จากการเขียนโปรแกรม ในการสั่งงานอุปกรณ์เพื่อควบคุมอุณหภูมิและค่าความเข้มของแสง และให้สามารถสอบเทียบตัวอุปกรณ์ได้ภายในตัวเอง
5. ได้ความรู้จากการศึกษา และแก้ไขการส่งสัญญาณควบคุมบัลลาสต์อิเล็กทรอนิกส์ เพื่อให้เอาต์พุตที่ได้ออกมาในรูปแบบของสัญญาณแอนะล็อก 0 ถึง 10 โวลต์
6. ได้ความรู้และประสบการณ์จากการศึกษา และใช้งานระบบทำความเย็น เพื่อลดอุณหภูมิภายในตัวเครื่องให้ได้ตามต้องการ

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 บทนำ

ในบทนี้จะกล่าวถึงทฤษฎีและหลักการที่เกี่ยวข้องในการออกแบบ และดำเนินการโครงการปรับปรุงประสิทธิภาพของเครื่องเขย่าแบบควบคุมอุณหภูมิและแสงส่องสว่าง ซึ่งประกอบไปด้วยบอร์ดอาร์ดูโน มาตรฐานการสื่อสารแบบต่าง ๆ หน้าจอร์บบสัมผัส เซนเซอร์วัดอุณหภูมิ เซนเซอร์วัดแสงสว่าง บัลลาสต์อิเล็กทรอนิกส์และวงจรแปลงแรงดัน

2.2 บอร์ดอาร์ดูโนเมกา 2560 (Arduino Mega 2560)

อาร์ดูโนเมกา 2560 มีลักษณะดังรูปที่ 2.1 อาร์ดูโนเมกา 2560 นั้นเป็นบอร์ดที่มีการใช้ไมโครคอนโทรลเลอร์ ATmega 2560 เป็นตัวควบคุม โดยจะมีขาอินพุตและเอาต์พุตทั้งหมด 54 ขา ซึ่ง 15 ขานั้นสามารถที่จะใช้เป็นขา PWM เอาต์พุตได้, ขาแอนะล็อกอินพุตมีทั้งหมด 16 ขา, 4 ขาเป็น UARTs (serial port), ใช้คริสตัลความถี่ 16 เมกกะเฮิร์ตซ์ (MHz), มี USB คอนเน็คเตอร์แบบ B, มีที่เสียบแหล่งจ่ายไฟแบบแจ๊ค เพื่อรับไฟกระแสตรงมาจากอแดปเตอร์ โดยที่แรงดันป้อนเข้าอยู่ระหว่าง 7 ถึง 12 โวลต์, มีพอร์ตสำหรับโปรแกรมแบบ ICSP (In-Circuit Serial Programming) และสวิตช์สำหรับรีเซ็ต เริ่มทำงานได้เพียงเชื่อมต่อกับคอมพิวเตอร์ด้วยสาย USB หรือจ่ายพลังงานด้วยไฟฟ้ากระแสตรง รวมถึงการใช้แบตเตอรี่ บอร์ดอาร์ดูโนเมกา 2560 นั้นจะไม่ได้ใช้ชิพไดรเวอร์ของ FTDI เหมือนบอร์ดชนิดอื่น แต่จะใช้ชิพ Atmega16U2 เป็นตัวกลางในการเชื่อมต่อกับคอมพิวเตอร์ และติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ ATmega 2560 ในรูปแบบของพอร์ตอนุกรมเสมือน (Visual Com Port) โดยโปรแกรมอาร์ดูโนจะมองเห็นพอร์ต USB ที่เชื่อมต่อกับบอร์ดอาร์ดูโนเมกา 2560 เป็นพอร์ตสื่อสารอนุกรม (Com Port) ช่องหนึ่งเท่านั้น และเนื่องจากมีบูตโหลดเดอร์ในตัว จึงสามารถโปรแกรมหน่วยความจำผ่านพอร์ต USB ได้โดยไม่ต้องใช้เครื่องโปรแกรมภายนอก และทำให้การดาวน์โหลดโปรแกรมจากคอมพิวเตอร์ผ่านพอร์ต USB กระทำได้สะดวกขึ้น การออกแบบฮาร์ดแวร์ที่ง่ายขึ้น ทำให้ใช้จำนวนอุปกรณ์ต่อรวมลดลง



รูปที่ 2.1 บอร์ดอาร์ดูโนเมกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 การจ่ายพลังงาน

บอร์ดอาร์ดูโนเมกานั้นสามารถรับพลังงานได้โดยการต่อสาย USB หรือการจ่ายพลังงานโดยตรงจากแหล่งจ่ายภายนอก ทั้งจากอแดปเตอร์แปลงไฟกระแสสลับเป็นกระแสตรงหรือแบตเตอรี่ โดยจะเชื่อมต่อผ่านขั้วเสียบแบบแจ๊คขนาด 2.1 มิลลิเมตร บอร์ดอาร์ดูโนเมกานั้นสามารถทำงานได้ในช่วงแรงดัน 6 ถึง 20 โวลต์ หากแหล่งจ่ายมีแรงดันต่ำกว่า 7 โวลต์ จะทำให้ขาจ่ายไฟ 5 โวลต์ จ่ายไฟออกได้ไม่ถึง 5 โวลต์ และจะทำให้บอร์ดไม่เสถียร แต่หากแรงดันมากกว่า 12 โวลต์ จะทำให้โวลต์เตจเรกูเลเตอร์ร้อนมากเกินไป และทำให้เกิดความเสียหายขึ้นกับบอร์ดได้ โดยช่วงแรงดันที่แนะนำคือ 7 ถึง 12 โวลต์ ขาที่เกี่ยวข้องกับภาคจ่ายไฟมีดังต่อไปนี้

- 1) ขา V_{IN} เป็นขาแรงดันขาเข้าไปยังบอร์ดอาร์ดูโนเมกา เมื่อใช้แหล่งจ่ายภายนอก จะสามารถป้องกันแรงดันเข้าที่ขา
- 2) ขา 5 โวลต์ เป็นขาแรงดันขาออก 5 โวลต์จากบอร์ด และสามารถจ่ายกระแสได้สูงที่สุด 40 มิลลิแอมป์ (mA)
- 3) ขา 3.3 โวลต์นั้นจ่ายมาจากเรกูเลเตอร์ของบอร์ด สามารถจ่ายกระแสได้สูงที่สุด 50 มิลลิแอมป์ (mA)
- 4) ขากราวด์ (GND)
- 5) ขา IOREF ขานี้จะให้แรงดันอ้างอิงเท่ากับแรงดันที่ใช้งานของไมโครคอนโทรลเลอร์ เพื่อเลือกค่าแรงดันให้กับชิลด์ที่มาเชื่อมต่อกับบอร์ด

2.2.2 หน่วยความจำ

ATmega2560 มีหน่วยความจำ 256 กิโลไบต์ (KB) ซึ่ง 8 กิโลไบต์ (KB) นั้นจะใช้ไปกับบูตโหลดเดอร์ (bootloader) และมีสแตติกแรม (Static Random Access Memory : SRAM) ขนาด 8 กิโลไบต์ (KB) รวมถึงอีอีพรอม (EEPROM) ขนาด 4 กิโลไบต์ (KB) เป็นไมโครคอนโทรลเลอร์ซึ่งใช้ตัวถังแบบ DIP ทำให้ง่ายต่อการพัฒนา เช่น เวลาที่ต้องการนำชิพที่ถูกลงโปรแกรมไปใช้ในบอร์ดอื่น หรือเวลาที่ชิพเสียหายก็สามารถที่จะหาชิพใหม่มาเปลี่ยนทดแทนได้

2.2.3 อินพุตและเอาต์พุต

แต่ละขาของขาดีจิทัล 54 ขานั้น จะใช้เพื่อเป็นอินพุตหรือเอาต์พุต โดยทำงานได้ที่แรงดัน 5 โวลต์ ในแต่ละขาสามารถจ่ายหรือรับกระแสได้สูงที่สุด 40 มิลลิแอมป์ (mA) และจะมีการความต้านทานพูลอัพภายใน 20 ถึง 50 กิโลโอห์ม โดยขาบางขาจะมีหน้าที่พิเศษ คือ

- 1) Serial: ขา 0 (RX) และ ขา 1 (TX), Serial1: ขา 19 (RX) และ ขา 18 (TX), Serial2: ขา 17 (RX) และ ขา 16 (TX), Serial3: ขา 15 (RX) และ ขา 14 (TX)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งขา RX นั้นเป็นขาสำหรับรับข้อมูล และขา TX เป็นขาสำหรับส่งข้อมูล ใช้การสื่อสารข้อมูลแบบ TTL (Transistor-Transistor Logic)

- 2) ขา PWM ขา 2 ถึง 13 และ ขา 44 ถึง 46 ให้ค่าเอาต์พุต PWM แบบ 8 บิต
- 3) ขา LED ขา 13 เป็นขาที่ไว้ใช้สำหรับต่อ LED จะติดเมื่อค่าที่ขานี้มีสถานะเป็นค่าศักย์ไฟสูง และจะดับเมื่อมีค่าศักย์ไฟต่ำ
- 4) ขา TWI (Two Wire Interface) ขา 20 เป็นขาสัญญาณ SDA (Serial Line Data) และขา 21 เป็นขาสัญญาณ SCL (Serial Clock) โดยขาเหล่านี้จะมีการใช้คำสั่งจากไลบรารีที่ชื่อว่า Wire ของอาร์ดูโน
- 5) ขา AREF เป็นสัญญาณอ้างอิงแบบแอนะล็อกจากภายนอกที่ต้องการป้อนให้กับไมโครคอนโทรลเลอร์ ซึ่งตามปกติแล้ว ATmega 2560 สามารถโปรแกรมให้เลือกใช้แรงดันอ้างอิงจากภายในได้อยู่แล้วโดยสามารถเลือกเป็น 1.1 โวลต์ หรือ 2.56 โวลต์ หรือ AVCC (+5 โวลต์) โดยไม่จำเป็นต้องป้อนแรงดันอ้างอิงจากภายนอกให้กับบอร์ด แต่ถ้าต้องการแรงดันอ้างอิงที่มีความแตกต่างจากที่กล่าวมาแล้ว ก็สามารถป้อนแรงดันอ้างอิงจากภายนอกผ่านทางขา AREF นี้ได้ โดยแรงดันที่ป้อนต้องมีค่าอยู่ช่วง 0 ถึง 5 โวลต์

2.2.4 การสื่อสาร

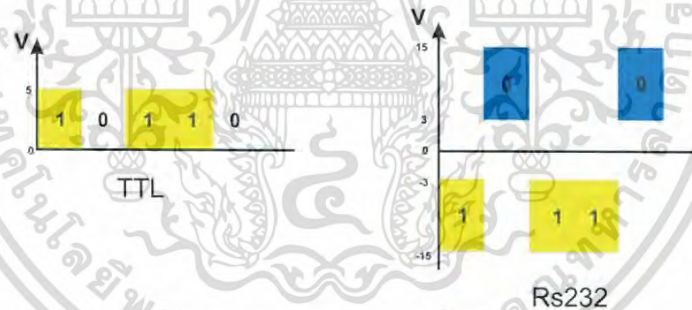
บอร์ดอาร์ดูโนเมกานั้นสามารถที่จะสื่อสารกับคอมพิวเตอร์ บอร์ดอาร์ดูโนอื่น และไมโครคอนโทรลเลอร์อื่น ๆ ได้โดยที่ไมโครคอนโทรลเลอร์บนบอร์ด คือ ATmega 2560 จะใช้การสื่อสารแบบ UART TTL (5 โวลต์) ซึ่งมีอยู่ในขา 0 (RX) และ ขา 1 (TX) นอกจากนี้ ATmega 2560 ยังรองรับการสื่อสารแบบ I²C และการสื่อสารแบบ SPI ส่วนชิพ ATmega16U2 จะใช้การสื่อสารแบบอนุกรมผ่านพอร์ต USB และปรากฏเป็นพอร์ตสื่อสารอนุกรมเสมือนไปยังซอฟต์แวร์บนคอมพิวเตอร์โดยที่ 16U2 จะใช้เฟิร์มแวร์ไดรเวอร์ที่เป็นมาตรฐาน โดยไม่ต้องติดตั้งไดรเวอร์จากภายนอก

2.3 การสื่อสารด้วยพอร์ต UART

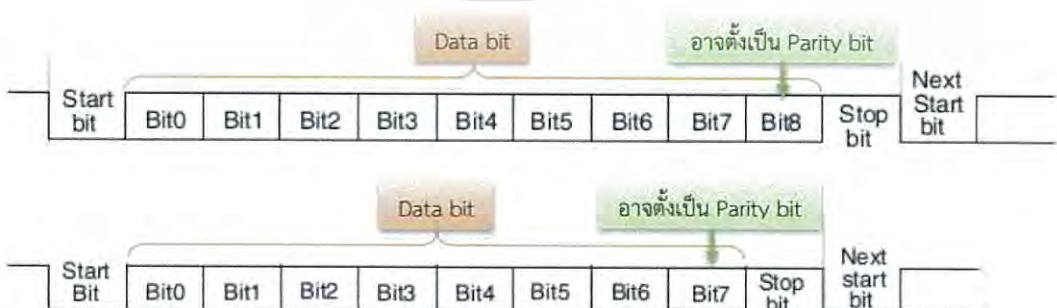
UART ย่อมาจากคำว่า Universal Asynchronous Receiver Transmitter เป็นการเชื่อมต่อและสื่อสารข้อมูลอนุกรมกับอุปกรณ์ต่าง ๆ เช่น คอมพิวเตอร์, RFID, GPS, GSM Module, Wifi Module เป็นต้น โดยข้อดีของการใช้การสื่อสารแบบอะซิงโครนัส (Asynchronous) คือสามารถสื่อสารแบบสองทิศทาง (Full duplex) กล่าวคือสามารถรับ และส่งข้อมูลระหว่างรีซีฟเวอร์ และทรานสมิตเตอร์ได้ในเวลาเดียวกัน นอกจากนี้ยังไม่จำเป็นต้องใช้สายสัญญาณนาฬิกา เพื่อกำหนดเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จังหวะการรับส่งข้อมูล แต่มีการกำหนดรูปแบบ หรือโปรโตคอลการรับส่งข้อมูลขึ้นมาแทน และอาศัยการกำหนดความเร็วของการรับส่งข้อมูลให้เท่ากัน การสื่อสารแบบ UART สามารถแบ่งตามระดับแรงดันไฟฟ้าของสัญญาณได้ 2 แบบคือ การสื่อสารแบบ TTL (Transistor-Transistor Logic) และการสื่อสารแบบ RS232 (Recommended Standard 232) ซึ่งการสื่อสารแบบ TTL เป็นการสื่อสารแบบดิจิทัลทั่วไปที่ใช้ภายในวงจรเดียวกัน โดยสถานะลอจิก “0” จะมีแรงดันเท่ากับ 0 โวลต์ สถานะลอจิก “1” จะมีแรงดันเท่ากับ 3.3 ถึง 5 โวลต์ แต่การสื่อสารแบบ RS232 เป็นการสื่อสารที่ใช้สำหรับการเชื่อมต่อกับอุปกรณ์ภายนอกโดยสถานะลอจิก “1” จะมีแรงดันตั้งแต่ -15 ถึง -5 โวลต์ สถานะลอจิก “0” จะมีแรงดันตั้งแต่ +5 ถึง +15 โวลต์ รูปที่ 2.2 แสดงการเปรียบเทียบความแตกต่างของสัญญาณทั้งสองรูปแบบ โดยรูปแบบการสื่อสารของ UART แสดงดังรูปที่ 2.3 โดยมีความจำเป็นที่จะต้องกำหนดคุณสมบัติเหล่านี้ให้เหมือนกันทั้งรีซีฟเวอร์ และ ทรานสมิตเตอร์ประกอบด้วย

- 1) เริ่มต้นบิตเป็นสถานะต่ำ
- 2) สามารถกำหนดจำนวนข้อมูลของบิตข้อมูลให้เป็น 8 และ 9 บิต
- 3) สามารถกำหนดชนิดของบิตแสดงสถานะเลขคู่หรือเลขคี่ (Parity Bit)
- 4) บิตหยุดมีจำนวน 0.5, 1, 1.5 และ 2 บิต



รูปที่ 2.2 รูปแบบการสื่อสารของ TTL และ RS232



รูปที่ 2.3 รูปแบบการสื่อสารของ UART

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การสื่อสารข้อมูลด้วย I²C

I²C ย่อมาจาก Inter-IC Communication หมายถึงการติดต่อสื่อสารระหว่างไอซี โดยบัส I²C ได้รับการพัฒนาโดยบริษัทฟิลิปส์ (Philips) ด้วยจุดมุ่งหมายหลักคือ ต้องการให้ไอซีหรือโมดูลสามารถติดต่อสั่งงาน และควบคุมภายใต้สายสัญญาณเพียง 2 เส้น เส้นหนึ่งคือ สายสัญญาณนาฬิกาที่ใช้กำหนดจังหวะการทำงาน การต่อร่วมกันของอุปกรณ์บนบัส I²C ทำได้ง่ายมาก เพียงต่อสายข้อมูลและสายสัญญาณนาฬิกาของอุปกรณ์แต่ละตัวขนานหรือพ่วงกันไป ส่วนการกำหนดแอดเดรสหรือตำแหน่งสำหรับติดต่ออุปกรณ์แต่ละตัว จะใช้รหัสข้อมูลและการกำหนดสภาวะลอจิกที่ขาแอดเดรสของอุปกรณ์แต่ละตัวสายข้อมูลบนบัส I²C มีชื่อเรียกอย่างเป็นทางการว่า สายข้อมูลอนุกรม หรือ SDA (Serial Data line) และส่วนสายสัญญาณนาฬิกามีชื่อเรียกว่า สายสัญญาณนาฬิกาอนุกรม หรือ SCL (Serial Clock line)

2.4.1 คุณสมบัติโดยทั่วไปของบัส I²C

ใช้สายสัญญาณ SDA และ SCL เป็นสายสัญญาณ 2 ทิศทาง (bi-directional line) ต้องมีการต่อตัวต้านทานพูลอัพ (Rp) กับแรงดัน +5 โวลต์ไว้ตลอดเวลา เพื่อให้สายมีสถานะลอจิกสูง ในขณะที่ไม่มีการติดต่อใช้งาน ทั้งยังช่วยป้องกันสัญญาณรบกวนที่อาจมีเข้ามาในสายสัญญาณทั้งสอง วงจรเอาต์พุตของอุปกรณ์ที่ต่ออยู่บนบัส I²C ต้องมีลักษณะเป็นวงจรเดรนเปิด (Open-drain) หรือคอลเลกเตอร์เปิด (Open-collector) อัตราการถ่ายทอดข้อมูลบนบัส I²C สูงถึง 100 กิโลบิตต่อวินาที ในโหมดปรกติ และสูงถึง 400 กิโลบิตต่อวินาทีในโหมดความเร็วสูง อุปกรณ์ที่ต่ออยู่บนบัส I²C จะต้องมีความจุไฟฟ้ารวมที่เกิดขึ้นระหว่างสายสัญญาณ SDA และ SCL ไม่เกิน 400 พิโกฟารัด (pF) การเข้าถึงอุปกรณ์บนบัส I²C ใช้ข้อมูลสำหรับการเข้าถึงสองค่าคือ 7 บิต (7-bit addressing) และ 10 บิต (10-bit addressing) ข้อดีอีกประการของบัส I²C คือ สามารถเชื่อมต่ออุปกรณ์ที่ใช้ไฟเลี้ยง +5 โวลต์ ในขณะที่อีกตัวหนึ่งใช้ไฟเลี้ยง +12 โวลต์ การต่อร่วมกันบนบัส I²C สามารถกระทำได้ในลักษณะเดียวกันกับกรณีอุปกรณ์ทั้งสองใช้ไฟเลี้ยงเท่ากัน กล่าวคือให้ต่อสายสัญญาณ SDA และ SCL ของอุปกรณ์แต่ละตัวเข้าด้วยกัน และต้องต่อตัวต้านทานพูลอัพ เข้ากับแรงดัน +5 โวลต์ด้วยเสมอ

2.4.2 หลักการของบัส I²C

บัส I²C ประกอบด้วยสายสัญญาณ 2 เส้นคือ SDA และ SCL อุปกรณ์ที่ต่อพ่วงบนบัสสามารถมีได้มากมาย ดังนั้นจึงต้องมีการกำหนดรูปแบบของการติดต่อบนบัส เพื่อให้ผู้ใช้งานทราบว่าขณะนี้อุปกรณ์ใดติดต่อกันอยู่ และอุปกรณ์ใดเป็นตัวรับข้อมูลและตัวส่งข้อมูล ซึ่งต่อไปจะขออธิบายลักษณะ หน้าที่ และนิยามของอุปกรณ์บนบัส I²C เพื่อเป็นข้อตกลงก่อนอธิบายการทำงานของบัส I²C ต่อไป อุปกรณ์ที่เป็นผู้สร้างข้อมูล หรือส่งข้อมูล เรียกว่า ตัวส่ง (transmitter) อุปกรณ์ที่เป็นผู้รับเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล เรียกว่า ตัวรับ (receiver) อุปกรณ์บนบัส I²C สามารถเป็นได้ทั้งตัวรับและตัวส่ง บางอุปกรณ์ทำหน้าที่เป็นตัวรับอย่างเดียว จะไม่มีอุปกรณ์ใดบนบัส I²C ที่ทำหน้าที่เป็นตัวส่งอย่างเดียว อุปกรณ์ที่ทำหน้าที่ควบคุมจังหวะการติดต่อบนบัส I²C เรียกว่า มาสเตอร์ (master) อุปกรณ์ที่ถูกควบคุมหรืออุปกรณ์ที่ต่อพ่วงเข้าไปบนบัส I²C เรียกว่า สเลฟ (slave) ข้อกำหนด 2 ประการสำคัญของการติดต่อบนบัส I²C คือ

- 1) การรับส่งข้อมูลจะเริ่มขึ้นได้เมื่อบัสสถานะว่างเท่านั้น
- 2) ในช่วงที่ทำการรับส่งข้อมูลอยู่ สายสัญญาณ SDA ต้องไม่เปลี่ยนสถานะในช่วงที่ SCL มีสถานะเป็นลอจิก “1” ถ้า SDA มีการเปลี่ยนสถานะในช่วงที่ SCL เป็นลอจิก “1” จะถือว่าเป็นสัญญาณควบคุมการรับส่งข้อมูล

2.4.3 สภาวะที่เกิดขึ้นบนบัส I²C

- 1) สถานะว่าง (Bus not busy) คือสายสัญญาณ SDA และ SCL มีระดับสัญญาณเป็นค่าสูง
- 2) เริ่มส่งข้อมูล (Start data transfer) มีการเปลี่ยนระดับสัญญาณของ SDA จากค่าสูงเป็นค่าต่ำ ในขณะที่ SCL มีระดับสัญญาณเป็นค่าสูงค้างไว้
- 3) หยุดส่งข้อมูล (Stop data transfer) มีการเปลี่ยนระดับสัญญาณของ SDA จากค่าต่ำเป็นค่าสูง ในขณะที่ SCL มีระดับสัญญาณเป็นค่าสูงค้างไว้
- 4) รับส่งข้อมูล (Data valid) มีการรับส่งข้อมูลผ่านสายสัญญาณ SDA โดยข้อมูลแต่ละบิตจะถูกส่งในช่วงที่ SCL มีระดับเป็นค่าสูง โดยในช่วงที่ SCL มีสถานะเป็นค่าสูงอยู่นั้น SDA จะต้องไม่เกิดการเปลี่ยนระดับสัญญาณ
- 5) ตอบรับ (Acknowledge) เกิดขึ้นหลังจากที่มีการรับส่งข้อมูลครบแล้ว โดยอุปกรณ์มาสเตอร์ต้องสร้างสัญญาณนาฬิกาบน SCL เพิ่มอีกลูก อุปกรณ์ที่เป็นตัวรับข้อมูลจะดึงระดับสัญญาณบน SDA ให้เป็นค่าต่ำ เพื่อให้ตัวส่งรับรู้ว่าตัวรับได้รับข้อมูลครบแล้ว

2.4.4 การทำงานบนบัส I²C

ก่อนจะเริ่มต้นการถ่ายทอดข้อมูลระหว่างอุปกรณ์ต่าง ๆ ที่ต่ออยู่บนบัสด้วยการเข้าถึงอุปกรณ์เสียก่อน โดยการเข้าถึงอุปกรณ์บนบัส I²C นั้น จะใช้การอ้างอิงถึงแบบ 7 และ 10 บิต ในกรณีที่มีอุปกรณ์ต่ออยู่บนบัสไม่มาก ใช้การอ้างอิงถึงแบบ 7 บิตก็เพียงพอ แต่ในบางอุปกรณ์ต้องใช้การอ้างอิงถึงแบบ 10 บิต หลังจากที่ได้ติดต่ออุปกรณ์แต่ละตัวเรียบร้อยแล้ว ก็จะเริ่มต้นการถ่ายทอด

ข้อมูลต่อไป ดังนั้นหัวใจสำคัญอันดับแรกของการทำงานบนบัส I²C คือการอ้างอิงถึงอุปกรณ์แต่ละตัว ซึ่งจะได้อธิบายรายละเอียดของการอ้างอิงถึงทั้ง 2 รูปแบบในลำดับต่อไป

2.4.4.1 การอ้างอิงถึงแบบ 7 บิต (7-bit addressing)

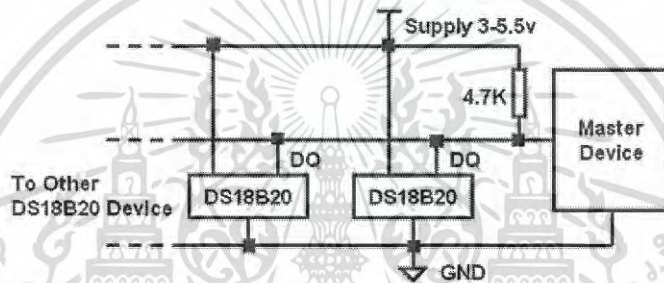
ข้อมูลไบต์แรกที่เกิดขึ้นหลังจากสถานะเริ่มต้นคือ ข้อมูลที่ใช้อ้างอิงอุปกรณ์ที่ต้องการติดต่อใน 7 บิต บนรวมทั้งบิต LSB ด้วย จะเป็นข้อมูลแอดเดรสของอุปกรณ์สเลฟที่ต้องการติดต่อ โดยแบ่งเป็นบิตกำหนดแอดเดรสคงที่ (fix address bit) จำนวน 4 บิต ซึ่งข้อมูลนี้ อุปกรณ์แต่ละตัวจะถูกกำหนดมาจากผู้ผลิต ไม่สามารถเปลี่ยนแปลงแก้ไขได้ ถัดมาอีก 3 บิตเป็นบิตกำหนดแอดเดรสที่สามารถโปรแกรมได้ (programmable address bit) โดยผู้ใช้งานต้องกำหนดสถานะลอจิกให้แก่ขา A0 ถึง A2 ของอุปกรณ์ที่มีการเชื่อมต่อแบบบัส I²C ส่วนในบิต LSB ที่ใช้กำหนดการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวนั้น ๆ หากบิต LSB เป็น "0" หมายถึงต้องการเขียนข้อมูลไปยังอุปกรณ์นั้น ถ้าเป็น "1" จะเป็นการอ่านข้อมูลจากอุปกรณ์สเลฟ ข้อมูลในไบต์ต่อมาคือ ข้อมูลควบคุม (control byte) ในอุปกรณ์แต่ละตัวจะมีการกำหนดข้อมูลควบคุมที่แตกต่างกันไป ยกตัวอย่างเช่น ไอซีขยายพอร์ตมีข้อมูลควบคุมที่ใช้กำหนดว่า บิตใดเป็นอินพุต บิตใดเป็นเอาต์พุต ในขณะที่ไอซีแปลงสัญญาณแอนะล็อกให้เป็นสัญญาณดิจิทัล (ADC) และแปลงสัญญาณดิจิทัลให้เป็นสัญญาณแอนะล็อก (DAC) ต้องการข้อมูลควบคุมเพื่อกำหนดให้ทำงานเป็นวงจรแปลงสัญญาณแอนะล็อกให้เป็นสัญญาณดิจิทัล หรือ แปลงสัญญาณดิจิทัลให้เป็นสัญญาณแอนะล็อก เป็นต้น ข้อมูลในไบต์ต่อมา คือข้อมูลที่ทำการถ่ายทอดจริง (data) หลังจากการถ่ายทอดข้อมูลในแต่ละไบต์ อุปกรณ์สเลฟที่ได้รับการติดต่อต้องส่งสัญญาณรับรู้ออกกลับมาด้วยทุกครั้ง เพื่อให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้

2.4.4.2 การเข้าถึงแบบ 10 บิต (10-bit addressing)

ในการอ้างอิงแบบนี้ยังคงใช้รูปแบบข้อมูลเหมือนกับแบบ 7 บิตโดยจะมีข้อมูลเพิ่มเติมขึ้นมาเล็กน้อย โดยในไบต์แรกหลังจากสถานะเริ่มต้น ต้องกำหนดให้ 5 บิตบนมีข้อมูลเป็น 11110 ส่วนอีก 2 บิตถัดมาเป็นบิตแอดเดรสของอุปกรณ์ที่ต้องการติดต่อในบิต LSB ของข้อมูลไบต์แรกยังคงเป็นการกำหนดว่า ต้องการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวที่ต้องการติดต่อกับข้อมูลไบต์ต่อมาเป็นข้อมูลแอดเดรสในไบต์ที่ 2 ของอุปกรณ์ที่ต้องการติดต่อกับข้อมูลไบต์ถัดไปจึงเป็นข้อมูลควบคุม ข้อมูลหลังจากนี้ก็จะข้อมูลจริงที่ใช้ในการติดต่อเช่นเดียวกันกับการเข้าถึงแบบ 7 บิตหลังจากถ่ายทอดข้อมูลครบทุกไบต์ ต้องมีสถานะรับรู้เกิดขึ้น เพื่อให้ขบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้

2.5 การสื่อสารด้วยสายสัญญาณ 1 เส้น (1-wire Communication)

การสื่อสารแบบสายสัญญาณ 1 เส้น เป็นระบบบัสข้อมูลแบบฮาร์ตวูฟเฟลกซ์นั้นคือสามารถสื่อสารได้ 2 ทิศทาง แต่ไม่สามารถรับและส่งข้อมูลในเวลาเดียวกันได้ ระบบบัสมีการทำงานเป็นแบบอุปกรณ์มาสเตอร์ หรืออุปกรณ์สเลฟ โดยอุปกรณ์มาสเตอร์จะเป็นตัวควบคุมสถานะ และจังหวะการรับส่งของบัสข้อมูล ในขณะที่อุปกรณ์สเลฟจะทำงานตามการควบคุมของอุปกรณ์มาสเตอร์เท่านั้น ในการใช้งานบัสแบบสายสัญญาณ 1 เส้นนี้ สายสัญญาณข้อมูล DQ จะต้องมีความต้านทานที่ล่อจิกสูงสามารถทำได้โดยการต่อตัวต้านทานพูลอัพประมาณ 5 กิโลโอห์มไว้กับไฟเลี้ยง หรือในกรณีที่ใช้บัสสายสัญญาณ 1 เส้นต่อร่วมกับอุปกรณ์ DS18B20 หลายตัว ก็สามารถทำได้ดังแสดงในรูปที่ 2.4



รูปที่ 2.4 การใช้บัสสายสัญญาณ 1 เส้นต่อร่วมกับอุปกรณ์ DS18B20 หลายตัว

2.6 หน้าจอบระบบสัมผัส

หน้าจอบระบบสัมผัสที่ใช้นั้นเป็นของบริษัท 4Dsystem ประเทศออสเตรเลีย ซึ่งเป็นผู้นำในการวิจัยและพัฒนาการผลิตเกี่ยวกับการใช้เทคโนโลยีกราฟิกของหน้าจอบระบบสัมผัส โดยหน้าจอบระบบสัมผัสนั้นมีความเป็นเอกลักษณ์ที่โดดเด่น และมีประสิทธิภาพสูงในการใช้งานที่หลากหลาย และยังมีพัฒนาเทคโนโลยีล่าสุดของจอภาพ คือจอภาพแบบ OLED (organic light-emitting diode) และจอภาพแบบ LCD (liquid-crystal display) รวมไปถึงการพัฒนาการประมวลผลต่าง ๆ ของหน้าจอให้มีประสิทธิภาพ จึงได้รับมาตรฐาน ISO 9001 และบริษัท 4Dsystem ได้มีวางยุทธศาสตร์ทางการค้า โดยเป็นพันธมิตรร่วมกับบริษัทหลากหลายบริษัททั่วโลก เพื่อให้เทคโนโลยีที่เป็นเอกลักษณ์เฉพาะ ได้เปลี่ยนแปลงแนวทางของตลาด และทำการขยายตลาดให้เติบโตขึ้นไป

2.6.1 โปรแกรมสำหรับออกแบบหน้าจอบระบบสัมผัส

Workshop 4 นั้นเป็นซอฟต์แวร์ IDE ที่ครอบคลุมสำหรับ Microsoft Windows ซึ่งสามารถที่จะเลือกใช้รูปแบบของการพัฒนาหลากหลายรูปแบบ สำหรับโมดูล และ ส่วนประมวลผลตระกูล 4D ทั้งหมด ซึ่งรวมกันมาไว้ทีเดียว โดยจะประกอบไปด้วย Editor, Compiler, linker และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Downloader เพื่อสร้าง 4DGL Application code ที่สมบูรณ์ โดยโปรแกรมทั้งหมดนั้นจะถูกพัฒนาขึ้นจาก Workshop 4 IDE ซึ่งจะประกอบไปด้วย Designer, Visi, Visi – Genie, Serial โดยตัวที่ใช้ในปริญญานิพนธ์นั้นคือ Visi – Genie ซึ่งจะเป็นการทำงานแบบขั้นสูง ซึ่งไม่จำเป็นต้องเขียนโปรแกรมใด ๆ การทำงานนี้จะไปแบบอัตโนมัติ โดยการจัดวางวัตถุบนจออย่างที่ต้องการ ซึ่งจะต้องมีการกำหนดเหตุการณ์เพื่อให้วัตถุทำงาน และการเขียนโปรแกรมนั้นจะเป็นไปอย่างอัตโนมัติ โดยจะมีการพัฒนาจาก 4D System โดยตรง Visi-Genie เป็นโปรแกรมที่แยกออกมาจากส่วนของกราฟิก 4D system โดยทำโปรแกรมให้ใช้งานได้ง่ายขึ้นสำหรับการออกแบบ GUI (Graphical User Interface) ที่มีความซับซ้อนโดยปราศจากการเขียนโปรแกรม ซึ่งผู้ใช้สามารถเห็นหน้า GUI ได้โดยตรง และสามารถสร้าง GUI พื้นฐานได้จากแบบร่างได้ในไม่กี่วินาที Visi-Genie นั้นมีการเขียนโปรแกรมในเบื้องหลัง ซึ่งทำให้ไม่ต้องมีการเรียนรู้การเขียนโปรแกรม 4DGL โดยต้องเลือกวัตถุที่ต้องการมาวางลงบนจอเหมือนกับ Visi โดยที่ไม่ต้องเขียนโปรแกรม และการทำงานแบบอนิเมชันก็สามารถสร้างเสร็จได้อย่างรวดเร็ว โดยวัตถุแต่ละตัวสามารถกำหนดตัวแปรต่าง ๆ ได้ และกำหนดเหตุการณ์เพื่อให้เคลื่อนไหวได้ รวมถึงการไปกระตุ้นวัตถุตัวอื่นให้ทำงาน และสื่อสารกับโฮสต์ภายนอกผ่านพอร์ตอนุกรม ดังแสดงตามรูปที่ 2.5 เช่น อาร์ดูโน เพียงแค่วางวัตถุบนจอกำหนดตำแหน่ง ขนาด และกำหนดตัวแปรต่าง ๆ เช่น สี ช่วงข้อมูล ตัวอักษร และเหตุการณ์ที่ต้องการให้วัตถุนี้เกี่ยวข้อง โดยวัตถุนั้นสามารถแบ่งออกเป็น 3 กลุ่มดังนี้

- 1) วัตถุรับค่า (Input object) คือ บุ่มหรือคีย์บอร์ด
- 2) วัตถุแสดงค่า (output object) เช่น เกจวัด หรือ มิเตอร์ต่าง ๆ
- 3) วัตถุทั้งรับและแสดงค่า (combined object, input/output object) นั้นเป็นวัตถุที่จะเป็นทั้ง ส่วนอินพุตและเอาต์พุต



รูปที่ 2.5 การเชื่อมต่อระหว่างจอสัมผัสกับโฮสต์

2.7 เซนเซอร์วัดอุณหภูมิ

DS18B20 เป็นไอซีวัดอุณหภูมิแบบดิจิทัลแบบ 3 ขาของ Dallas Semiconductor ดังรูปที่ 2.6 สามารถวัดอุณหภูมิเป็นหน่วยองศาเซลเซียส ได้ในช่วง -55 องศาเซลเซียส ถึง 125 องศาเซลเซียส มีความละเอียด 9 ถึง 10 บิต และมีความแม่นยำอยู่ที่ 0.5 องศาเซลเซียส ในช่วง -10 องศาเซลเซียส เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

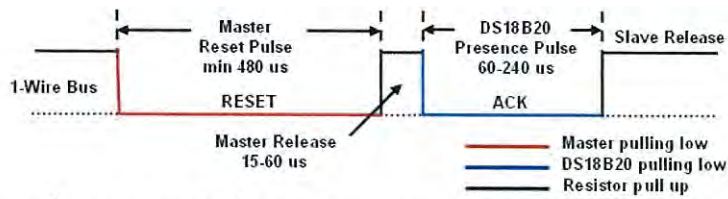
เซลเซียส ถึง 85 องศาเซลเซียส การสื่อสารและควบคุม DS18B20 สามารถทำได้โดยใช้บัสข้อมูลแบบสัญญาณ 1 เส้นของ Dallas Semiconductor เท่านั้น ภายใน DS18B20 แต่ละตัว จะมีการเขียนโปรแกรมประจำตัวขนาด 64 บิต ทำให้สามารถใช้งาน DS18B20 ได้หลายตัวทำงานบนบัสแบบ 1 เส้นพร้อมกันได้ นอกจากนี้ DS18B20 ยังสามารถทำงานในโหมดพาราสิติก (Parasite Power Mode) ซึ่งเป็นการทำงานแบบไม่ต้องใช้ไฟเลี้ยงแต่ใช้พลังงานจากสายสัญญาณ 1 เส้น ซึ่งมีประโยชน์มากสำหรับการวัดอุณหภูมิระยะไกล หรือในการใช้ในที่ ๆ มีเนื้อที่จำกัด ข้อมูลอุณหภูมิที่วัดได้จะถูกเก็บอยู่ในรีจิสเตอร์อุณหภูมิ ซึ่งมีขนาด 16 บิต ในกรณีที่ DS18B20 ทำงานในโหมดความละเอียด 12 บิต บิตทุกบิตในรีจิสเตอร์อุณหภูมิจะถูกใช้หมด แต่ในกรณี 9 ถึง 11 บิต บิตล่าง (บิต 0 ถึง บิต 2) จะไม่ถูกใช้งาน สามารถกำหนดความละเอียดของบิตโดยกำหนดที่การตั้งค่ารีจิสเตอร์



รูปที่ 2.6 โครงสร้างและขาของ DS18B20

2.7.1 การต่อใช้งานเซนเซอร์วัดอุณหภูมิ (DS18B20)

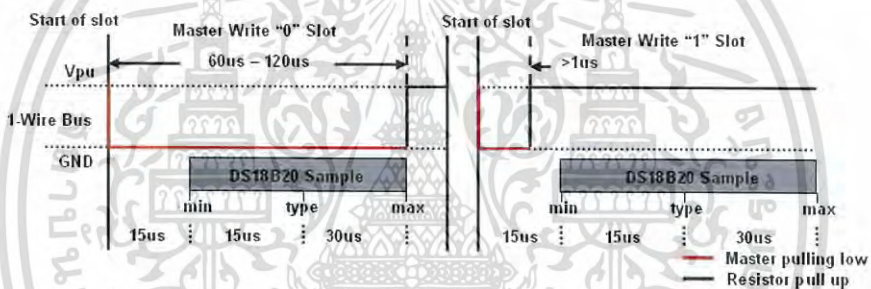
รูปแบบของสัญญาณบนบัสสายสัญญาณ 1 เส้น (1-wire Communication) สามารถแบ่งออกได้เป็น 6 รูปแบบ คือ Reset pulse, Presence pulse, write 0, write 1, read 0 และ read 1 ซึ่งในกระบวนการเริ่มต้นการสื่อสารสายสัญญาณ 1 เส้นทั้งหมดนั้น อุปกรณ์มาสเตอร์ (Master) ต้องขอเริ่มการสื่อสารด้วยการสร้าง Reset pulse ก่อน เมื่ออุปกรณ์สเลฟ (Slave) ได้รับ Reset pulse ก็จะสร้าง Presence pulse เพื่อตอบรับการขอเริ่มการสื่อสารนั้น ซึ่งมีรายละเอียดของช่วงเวลาต่าง ๆ ดังแสดงในรูปที่ 2.7



รูปที่ 2.7 การติดต่อสื่อสารแบบสายสัญญาณ 1 เส้นกับ DS18B20

2.7.1.1 การเริ่มการติดต่อสื่อสารแบบสายสัญญาณ 1 เส้น ด้วย Reset Pulse และ Presence Pulse

ในการเขียนข้อมูลแบ่งออกเป็น 2 ชนิด คือการเขียนข้อมูล “1” และการเขียนข้อมูล “0” ดังแสดงในรูปที่ 2.8 การเขียนข้อมูลลง DS18B20 ต้องใช้ช่วงเวลาอย่างต่ำ 60 ไมโครวินาที

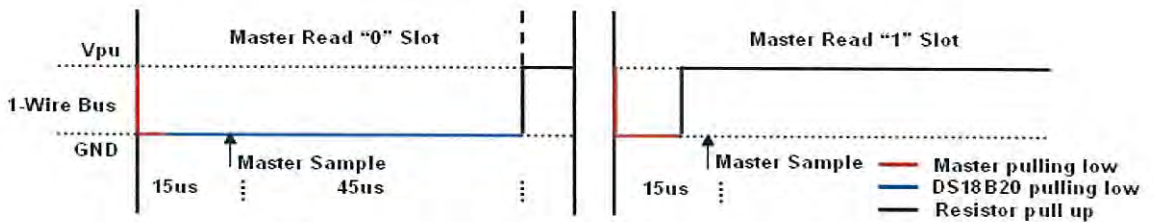


รูปที่ 2.8 การเขียนข้อมูลลงเซนเซอร์อุณหภูมิ DS18B20

2.7.1.2 การเขียนข้อมูลลงเซนเซอร์อุณหภูมิ (DS18B20)

การเขียนข้อมูลทั้ง 2 ชนิด เริ่มแรกอุปกรณ์มาสเตอร์ต้องดึงสัญญาณบนบัสสายสัญญาณ 1 เส้นลงมาอยู่ในสถานะลอจิกต่ำก่อน ในกรณีที่ต้องการเขียนข้อมูล “0” ลงใน DS18B20 อุปกรณ์มาสเตอร์ต้องดึงสัญญาณบนบัสให้เป็นลอจิกต่ำต่อไป จนกว่าจะครบช่วงเวลาที่ไหม้สล็อต (อย่างต่ำ 60 ไมโครวินาที) ส่วนในกรณีที่ต้องการเขียนข้อมูล “1” ลง DS18B20 อุปกรณ์มาสเตอร์ต้องปล่อยบัส เพื่อให้บัสกลับไปอยู่ในสถานะลอจิกสูงก่อนการสุ่มของ DS18B20 ซึ่งจะอยู่ในช่วง 15 ไมโครวินาทีถึง 60 ไมโครวินาที หลังจากที่อุปกรณ์มาสเตอร์ดึงสัญญาณบัส 1-wire ลงมาในการอ่านค่าภายในสแตติกแรมของ DS18B20 สามารถทำได้ก็ต่อเมื่ออุปกรณ์มาสเตอร์ได้เขียนข้อมูลเพื่อขอทำการอ่านค่าในสแตติกแรม (Read Scratchpad) ซึ่งมีการรับข้อมูลแบบเลขฐานสิบหก (hexadecimal) ลงไปที่ DS18B20 ก่อน จากนั้นจึงเริ่มอ่านข้อมูลจากบัส 1-wire โดยไหม้สล็อตของการอ่านต้องมีช่วงเวลาอย่างต่ำ 60 ไมโครวินาที และต้องมีช่วงเวลาระหว่างไหม้สล็อตอย่างต่ำ 1 ไมโครวินาทีดังแสดงในรูปที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 แสดงการอ่านข้อมูลจากบัสสัญญาณ 1 เส้น

2.7.1.3 การอ่านข้อมูลจาก DS18B20

การอ่านข้อมูลจากบัสสัญญาณ 1 เส้น เริ่มแรกอุปกรณ์มาสเตอร์จะต้องดึงบัสสัญญาณ 1 เส้นลงมาอยู่ในสถานะลอจิกต่ำเป็นช่วงเวลาอย่างน้อย 1 ไมโครวินาที จากนั้นจึงค่อยปล่อยบัส ในกรณีที่ DS18B20 ส่งข้อมูล "0" DS18B20 จะดึงบัสให้เป็นลอจิกต่ำจนกว่าจะสิ้นสุดไทม์สล็อตจึงจะปล่อยให้บัสกลับไปอยู่ในสถานะลอจิกสูง ส่วนในกรณีที่ DS18B20 ส่งข้อมูล "1" DS18B20 จะปล่อยบัสให้อยู่ในสถานะลอจิกสูงตลอด ในการสุ่มเพื่อรับข้อมูลจาก DS18B20 ควรทำภายใน 15 ไมโครวินาที หลังจากจุดเริ่มของไทม์สล็อตแสดงในรูปที่ 2.9

2.8 เซนเซอร์วัดความสว่าง (BH1750FVI)

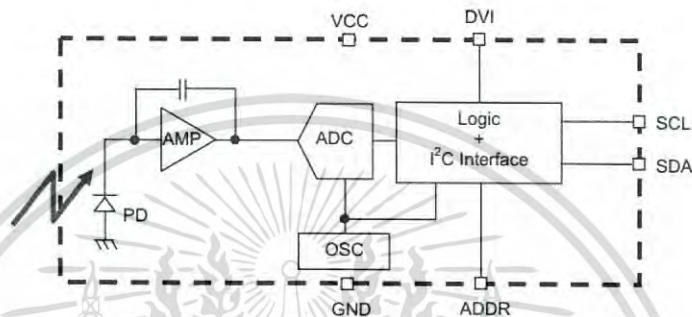
BH1750FVI เป็นโมดูลวัดความเข้มแสงโดยรอบแบบดิจิทัลซึ่งใช้การเชื่อมต่อแบบ I²C bus interface ดังรูปที่ 2.10 ในการเชื่อมต่อกับอุปกรณ์อื่น ๆ โมดูลเซนเซอร์นี้เป็นตัวที่เหมาะสมสำหรับการวัดแสงโดยรอบเพื่อปรับความสว่างของหน้าจอ และแสงที่แบ่นพิมพ์ของโทรศัพท์มือถือซึ่ง BH1750 มีรัศมีการวัดกว้างและมีความละเอียดที่สูงถึง 65,532 ลักซ์

2.8.1 คุณสมบัติของ BH1750FVI

- 1) ใช้การเชื่อมต่อแบบ I²C
- 2) เสถียรภาพการตอบสนองประมาณสายตามนุษย์
- 3) แปลงค่าความสว่างเป็นดิจิทัลเอาต์พุต
- 4) รัศมีการวัดกว้าง และมีความละเอียดสูง (1 ถึง 65,535 ลักซ์)
- 5) ใช้พลังงานน้อย โดยใช้โหมดประหยัดพลังงาน
- 6) มีฟังก์ชันการกรองแสงรบกวนความถี่ 50/60 เฮิร์ตซ์ (Hz)
- 7) ไม่จำเป็นต้องมีอุปกรณ์เพิ่มเติม
- 8) พึ่งพาแหล่งกำเนิดแสงเพียงเล็กน้อย (เช่น หลอดฟลูออเรสเซนต์หลอดฮาโลเจน แสง LED แสงอาทิตย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 9) สามารถเลือกทั้ง 2 ชนิดของ I²C slave-address
- 10) สามารถปรับค่าผลของการวัดสำหรับนำไปแสดงผล (สามารถวัดได้น้อยสุด 0.11 ลักซ์, มากสุด 100,000 ลักซ์โดยปรับที่ฟังก์ชันการทำงาน)
- 11) ค่าการวัดเปลี่ยนแปลงน้อย (+/- 20 เปอร์เซ็นต์)
- 12) ได้รับอิทธิพลจากอินฟราเรดเพียงเล็กน้อย



รูปที่ 2.10 บล็อกไดอะแกรมของ BH1750FVI

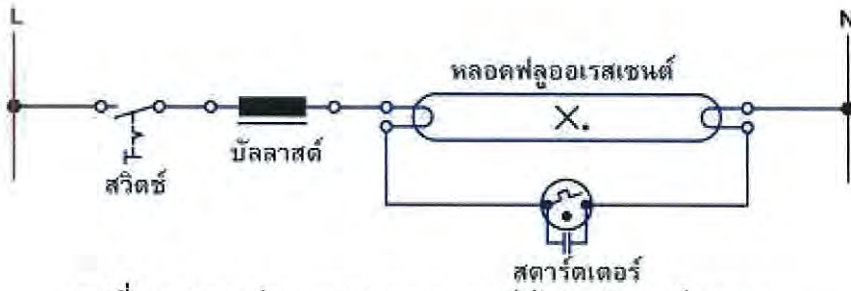
2.9 หลอดฟลูออเรสเซนต์ (Fluorescent Lamp)

หลอดฟลูออเรสเซนต์ทำด้วยหลอดแก้วที่สุบอากาศออกจนหมด แล้วบรรจุไอปรอทไว้เล็กน้อย มีไส้ที่ปลายหลอดทั้งสองข้าง หลักการทำงานของหลอดฟลูออเรสเซนต์ คือเมื่อกระแสไฟฟ้าผ่านไอปรอทจะคายพลังงานไฟฟ้าให้อะตอมไอปรอท ทำให้อะตอมของไอปรอทอยู่ในสภาวะถูกกระตุ้น (excited state) และอะตอมของปรอทจะคายพลังงานออกมา เพื่อลดระดับพลังงาน ในรูปของรังสีอัลตราไวโอเล็ต ซึ่งอยู่ในช่วงของแสงที่มองไม่เห็น เมื่อรังสีนี้กระทบสารเรืองแสงที่ฉาบไว้ที่ผิวหลอดสารเรืองแสงจะเปล่งแสงสีต่าง ๆ ตามชนิดของสารเรืองแสงที่ฉาบไว้ในหลอดนั้น ดังที่แสดงในรูปที่ 2.11 หลอดฟลูออเรสเซนต์จะใช้งานร่วมกับบัลลาสต์ และสตาร์ทเตอร์ (เฉพาะหลอดฟลูออเรสเซนต์ที่ใช้งานร่วมกับบัลลาสต์อิเล็กทรอนิกส์ ไม่ต้องทำงานร่วมกับสตาร์ทเตอร์) ดังที่แสดงในรูปที่ 2.12 และรูปที่ 2.13

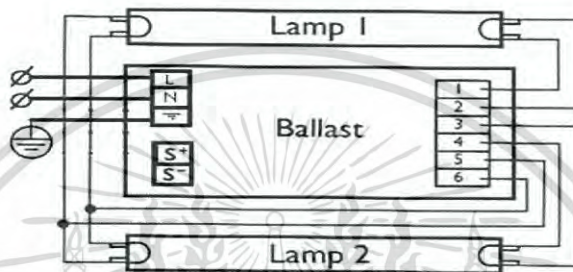


รูปที่ 2.11 โครงสร้างภายในของหลอดฟลูออเรสเซนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 การต่อหลอดฟลูออเรสเซนต์เข้ากับบัลลาสต์ธรรมดา



รูปที่ 2.13 การต่อหลอดฟลูออเรสเซนต์เข้ากับบัลลาสต์อิเล็กทรอนิกส์แบบ 2 หลอด

2.10 บัลลาสต์อิเล็กทรอนิกส์ (Electronic Ballast)

บัลลาสต์อิเล็กทรอนิกส์มีหน้าที่ควบคุมกระแสไฟฟ้าที่ผ่านเข้าไปในขั้วหลอด ให้มีค่าเหมาะสมตามแต่ละประเภทหลอด แต่ละชนิด แต่ละรุ่น แต่ละขนาด บัลลาสต์ประหยัดพลังงานที่นิยมใช้กันมาก คือ บัลลาสต์โลว์ลอส (Low Loss Ballast) และบัลลาสต์อิเล็กทรอนิกส์ (Electronic Ballast) ในที่นี้ขออธิบายแค่บัลลาสต์อิเล็กทรอนิกส์ ซึ่งเป็นอุปกรณ์อีกชนิดหนึ่งทำหน้าที่เหมือนกับอุปกรณ์เสริมทั้ง 3 อย่างของการต่อหลอดฟลูออเรสเซนต์แบบใช้บัลลาสต์ชนิดขดลวด คือ บัลลาสต์ชนิดขดลวด สตาร์ทเตอร์ และตัวเก็บประจุ โดยที่บัลลาสต์อิเล็กทรอนิกส์จะสร้างความถี่สูงถึง 50 กิโลเฮิร์ตซ์ (kHz) จ่ายให้กับหลอดฟลูออเรสเซนต์ ซึ่งหลอดฟลูออเรสเซนต์จะตอบสนองความถี่สูงได้ดีกว่าความถี่ 50 เฮิร์ตซ์ (Hz) ที่ใช้ในบ้าน ส่งผลให้สามารถประหยัดพลังงานไฟฟ้าได้ไม่น้อยกว่า 30 เปอร์เซ็นต์ และยังยืดอายุการใช้งานของหลอดได้นานกว่า 25 เปอร์เซ็นต์ เมื่อเทียบกับการต่อวงจรโดยใช้บัลลาสต์ชนิดขดลวด การใช้บัลลาสต์อิเล็กทรอนิกส์ยังมีข้อดีอื่น ๆ อีกเช่นหลอดเปิดติดทันทีแสงสว่างที่ออกมาไม่กระพริบให้รำคาญสายตา อีกทั้งยังไม่สะสมความร้อนเหมือนบัลลาสต์ชนิดขดลวด และหลอดยังคงติดแม้แรงดันไฟฟ้าจะตก ในเรื่องของการทำงานบัลลาสต์อิเล็กทรอนิกส์จะบอกลักษณะจำเพาะที่จะนำไปใช้งานอย่างเช่น จำนวนหลอด ขนาดกำลังงานไฟฟ้า (Watt) เป็นต้น ซึ่งผู้ใช้งานควรศึกษาลักษณะของงานที่จะนำไปใช้กับบัลลาสต์ที่เลือกใช้ เพราะหากเลือกขนาดไม่ตรงกันหลอดฟลูออเรสเซนต์ก็จะไม่สามารถใช้งานได้ โดยจะเลือกบัลลาสต์อิเล็กทรอนิกส์แบบปรับค่าได้ (Dimmable Electronic Ballast) ซึ่งเป็นบัลลาสต์อิเล็กทรอนิกส์อีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดหนึ่งที่มีความสามารถในการปรับแสงสว่างของหลอดไฟ ด้วยวิธีการต่าง ๆ เช่น การใช้แรงดันในการควบคุมความสว่างเป็นต้น นิยมใช้กันเพื่อลดความสว่างของแสงให้พอดีกับสายตา หรือช่วยลดค่าแสงสว่างเมื่อมีแสงจากภายนอกเข้ามา

2.11 ออปแอมป์

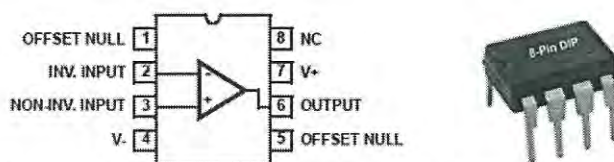
ออปแอมป์ หรือวงจรรขยายสัญญาณเชิงดำเนินการ (Operational Amplifier : Op-Amp) ดังรูปที่ 2.14 เป็นอุปกรณ์อิเล็กทรอนิกส์ชนิดหนึ่งที่ภายในประกอบด้วยทรานซิสเตอร์จำนวนหนึ่ง ทำหน้าที่เป็นวงจรรขยายสัญญาณแรงดันด้านอินพุตโดยมีเอาต์พุตเป็นแรงดัน โดยสัญลักษณ์ของออปแอมป์จะเป็นรูปสามเหลี่ยมโดยมีขั้วอินพุต 2 ขั้ว และขั้วเอาต์พุต 1 ขั้ว ดังรูปที่ 2.15 ขั้วอินพุต 2 ขั้ว ได้แก่ ขั้วอินพุตแบบไม่กลับเฟส (non-inverting input) และขั้วอินพุตแบบกลับเฟส (inverting input) และวงจรรภายในออปแอมป์ประกอบไปด้วยทรานซิสเตอร์จำนวนหนึ่งตามตัวอย่างในรูปที่ 2.16 นั้น สามารถเขียนแทนได้ด้วยวงจรมูลอย่างง่ายดังรูปที่ 2.17 โดย R_{in} คือความต้านทานทางด้านอินพุต (input resistance) เป็นความต้านทานที่ต่ออยู่ที่ขั้วอินพุตทั้งสอง R_{out} คือความต้านทานทางด้านเอาต์พุต (output resistance) และ A คืออัตราขยายแรงดันไฟฟ้าลูปเปิด (open loop voltage gain) เมื่อกำหนดให้ออปแอมป์เป็นอุดมคติจะได้คุณสมบัติต่าง ๆ ของออปแอมป์ดังนี้

$R_{in} = \infty$ ความต้านทานทางด้านอินพุตมีค่าเป็นอนันต์

$R_{out} = 0$ ความต้านทานทางด้านอินพุตมีค่าเป็นศูนย์

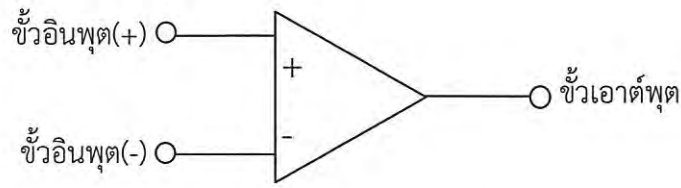
$A = \infty$ อัตราขยายแรงดันไฟฟ้าวงรอบเปิดมีค่าเป็นอนันต์

ดังนั้นจะได้วงจรมูลของออปแอมป์อุดมคติดังรูปที่ 2.17 ซึ่งในการวิเคราะห์วงจรที่ประกอบด้วยออปแอมป์นี้ จะใช้วงจรมูลของออปแอมป์อุดมคติในการวิเคราะห์ โดยการแทนที่ออปแอมป์ในวงจรด้วยวงจรมูลดังรูปที่ 2.17 ในส่วนของค่าอัตราขยายแรงดันไฟฟ้าลูปเปิดนั้น ถ้าเป็นออปแอมป์ในอุดมคติจะมีค่าเป็นอนันต์ตามที่กล่าวถึงข้างต้น แต่เนื่องจากในโปรแกรม Schematics ไม่สามารถใส่ค่าอัตราขยายที่เป็นอนันต์ได้จึงกำหนดให้อัตราขยายมีค่าที่สูงมาก ๆ เช่น มากกว่า 100,000 ขึ้นไป เป็นต้น

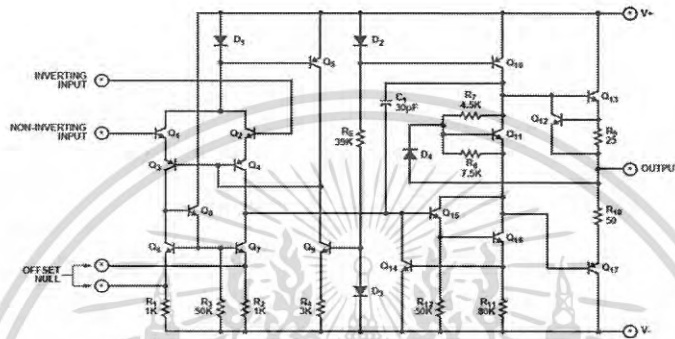


รูปที่ 2.14 ออปแอมป์เบอร์ LM741

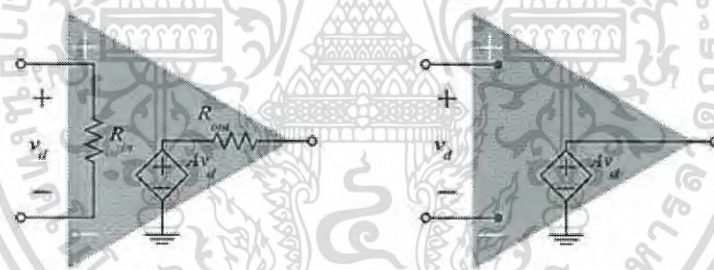
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 สัญลักษณ์ของออปแอมป์



รูปที่ 2.16 วงจรภายในออปแอมป์



รูปที่ 2.17 วงจรสมมูลอย่างง่ายของออปแอมป์และวงจรสมมูลอย่างง่ายของออปแอมป์ในอุดมคติ

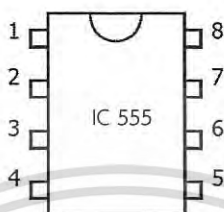
2.12 ไอซี 555

ไอซี 555 ดังรูปที่ 2.18 จัดเป็นไอซีประเภทไทเมอร์ นิยมใช้กันมากในการนำไปสร้างสัญญาณรูปคลื่นแบบต่าง ๆ และวงจรตั้งเวลา เนื่องจากไอซี 555 ใช้อุปกรณ์ต่อร่วมน้อย มีความเสถียร สามารถควบคุมความถี่ได้ด้วยค่าความต้านทาน และตัวเก็บประจุ มีขาใช้งานทั้งหมด 8 ขา สามารถทำงานได้ที่แรงดัน 4.5 ถึง 18 โวลต์ แบ่งหลักการทำงานของไอซี 555 เป็นวงจรที่สำคัญได้ 2 ชนิดคือ

- 1) วงจรโมโนสเตเบิลมัลติไวเบเรเตอร์ (Monostable Multivibrator) คือ วงจรที่จะสร้างความถี่ขึ้นมา 1 ลูก เมื่อมีการกระตุ้นทริกเกอร์ (Trigger) เกิดขึ้นโดยความกว้างของพัลส์สามารถกำหนดได้ด้วย R และ C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2) วงจรออสเตเบิลมัลติไวเบรเตอร์ (Astable Multivibrator) คือ วงจรสร้างความถี่ขึ้นมาอย่างต่อเนื่อง สามารถกำหนดได้ด้วยตัวต้านทาน (R) จำนวน 2 ตัว และตัวเก็บประจุ (C) 1 ตัว



1. GND
2. Trigger
3. Output
4. Reset
5. Control Voltage
6. Threshold
7. Discharge
8. Vcc

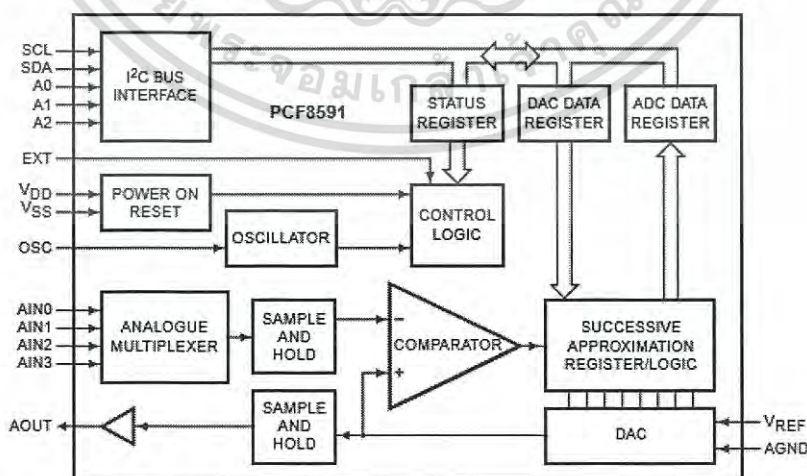
รูปที่ 2.18 ไอซี 555

2.13 ไอซี PCF8591

PCF8591 คือชิพที่สามารถแปลงสัญญาณได้ทั้งสัญญาณแอนะล็อกให้เป็นสัญญาณดิจิทัล (ADC) และ สัญญาณดิจิทัลให้เป็นสัญญาณแอนะล็อก (DAC) ดังรูปที่ 2.19 ซึ่งมีขนาด 8 บิต โดยรูปแบบการสื่อสารข้อมูลกับไมโครคอนโทรลเลอร์ I²C บัส ดังรูปที่ 2.20 และ 2.21 เช่นเดียวกับไอซีขยายพอร์ต PCF8574A กล่าวคือใช้สายสัญญาณในการสื่อสารข้อมูลเพียง 2 เส้นเท่านั้น คือสายสัญญาณ SDA และ SCL PCF8591 สามารถรับสัญญาณอินพุตที่เป็นสัญญาณทางแอนะล็อกได้ 4 ช่อง (AIN0 ถึง AIN3) สัญญาณแอนะล็อกทั้ง 4 ช่องนี้สามารถแยกใช้งานได้อย่างอิสระ และยังสามารถนำสัญญาณจากอินพุตช่องต่าง ๆ มาทำการเปรียบเทียบหรือหาผลต่างได้อีกด้วย นอกจากนี้แล้วยังมีช่องสัญญาณสำหรับส่งสัญญาณแอนะล็อก (AOUT) ออกไปให้อุปกรณ์ภายนอกอีก 1 ช่องสัญญาณ มีขาต่าง ๆ ทั้งหมด 16 ขา ดังรูปที่ 2.22 แต่ละขามีหน้าที่การทำงานดังต่อไปนี้

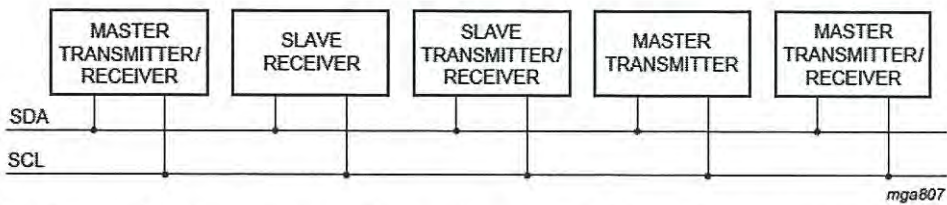
- 1) ขา 1 ถึง 3 หรือขา AIN0 ถึง AIN3 เป็นขาอินพุตใช้ในการรับสัญญาณแอนะล็อก (0 ถึง 5 โวลต์) สัญญาณแอนะล็อกที่รับเข้ามาทางขาเหล่านี้จะถูกส่งผ่านไปยังวงจรแปลงสัญญาณแอนะล็อก เป็นสัญญาณดิจิทัลภายใน
- 2) ขา 5 ถึง 7 หรือขา A ถึง A2 เป็นขาอินพุตใช้ในการรับสัญญาณลอจิก (“0” หรือ “1”) เพื่อทำการกำหนดตำแหน่ง (Address) ของ PCF8591 จึงทำให้สามารถเลือกตำแหน่งที่แตกต่างกันได้ถึง 8 ตำแหน่ง นั่นหมายความว่าสามารถต่อ PCF8591 บนบัสเดียวกันได้สูงสุดถึง 8 ตัว
- 3) ขา 8 หรือขา VSS เป็นขาอินพุตทำหน้าที่รับแรงดัน 0 โวลต์ (GND)

- 4) ขา 9 หรือขา SDA เป็นขาอินพุตและเอาต์พุต ทำหน้าที่รับและส่งข้อมูลที่ส่งกัน ไมโครคอนโทรลเลอร์ การรับเมื่อไมโครคอนโทรลเลอร์ต้องการติดต่อเพื่ออ่านข้อมูลขา นี้จะเป็นขาเอาต์พุต และในทางกลับกันเมื่อไมโครคอนโทรลเลอร์ต้องการติดต่อเพื่อ เขียนข้อมูล ขานี้จะเป็นขาอินพุต
- 5) ขา 10 หรือ SCL เป็นขาอินพุต ทำหน้าที่รับสัญญาณที่ส่งมาจากไมโครคอนโทรลเลอร์ สัญญาณที่ป้อนเข้ามาที่ขานี้จึงเป็นตัวกำหนดจังหวะการอ่านและเขียนข้อมูล
- 6) ขา 11 หรือขา OSC เป็นขาอินพุตและเอาต์พุต จะรับสัญญาณนาฬิกาจากภายนอกเข้ามาเมื่อขา EXT ต่อกับไฟ +5 โวลต์ และจะส่งสัญญาณนาฬิกาออกไปหากขา EXT ต่อ กับกราวด์
- 7) ขา 12 หรือขา EXT เป็นขาอินพุตใช้ในการเลือกแหล่งกำเนิดสัญญาณนาฬิกา คือถ้าขา นี้ต่อกับไฟ +5 โวลต์ จะเป็นการเลือกแหล่งกำเนิดสัญญาณนาฬิกาจากภายนอก ถ้าต่อกับกราวด์จะเป็นการเลือกแหล่งกำเนิดสัญญาณนาฬิกาจากภายใน
- 8) ขา 13 หรือขา AGND เป็นขาอินพุตใช้เป็นขากราวด์ของสัญญาณอ้างอิง ในการใช้งาน ปกติจะต่อลงกราวด์ของสัญญาณอ้างอิงเป็น 0 โวลต์นั่นเอง
- 9) ขา 14 หรือขา VREF เป็นขาอินพุตใช้ในการรับสัญญาณอ้างอิง ในการใช้งานปกติจะ ต่อกับไฟ +5 โวลต์
- 10) ขา 15 หรือขา AUOT เป็นขาเอาต์พุตใช้ในการส่งสัญญาณแอนะล็อก ที่ถูกแปลงมา จากวงจรแปลงสัญญาณดิจิทัลเป็นแอนะล็อกภายในออกไปใช้งาน
- 11) ขา 16 หรือขา VDD เป็นขาอินพุตใช้ในการรับไฟเลี้ยง +5 โวลต์

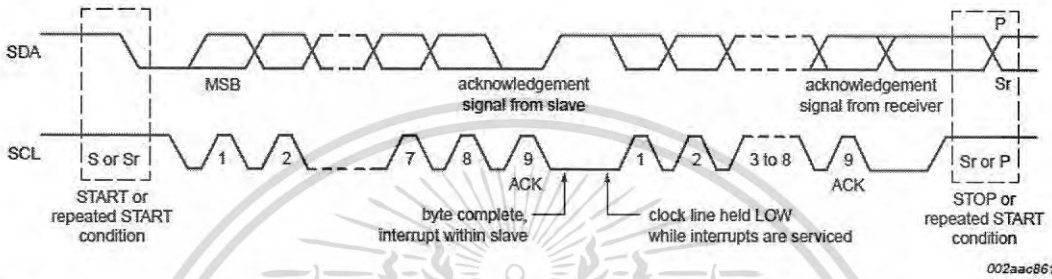


รูปที่ 2.19 บล็อกไดอะแกรมของไอซี A/D และ D/A PCF8591 (8-bit A/D and D/A Converter)

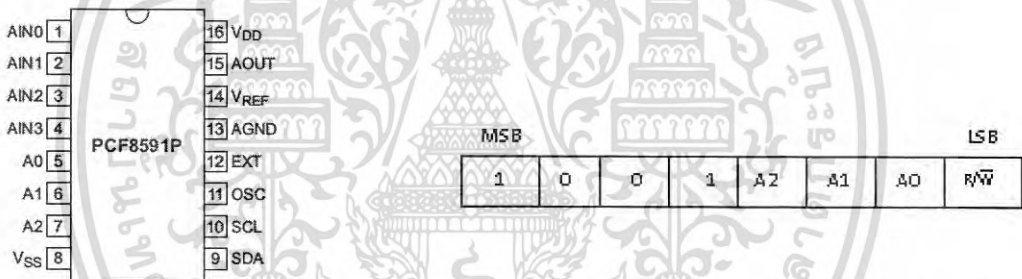
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น. อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 การต่ออุปกรณ์มาสเตอร์ (Master) และอุปกรณ์สเลฟ (Slave) บนระบบบัสแบบ I²C



รูปที่ 2.21 สัญญาณความถี่ขา SCL และ SDA ในสภาวะต่าง ๆ ของการสื่อสารข้อมูลบนบัสแบบ I²C



รูปที่ 2.22 การจัดขาของ PCF8591

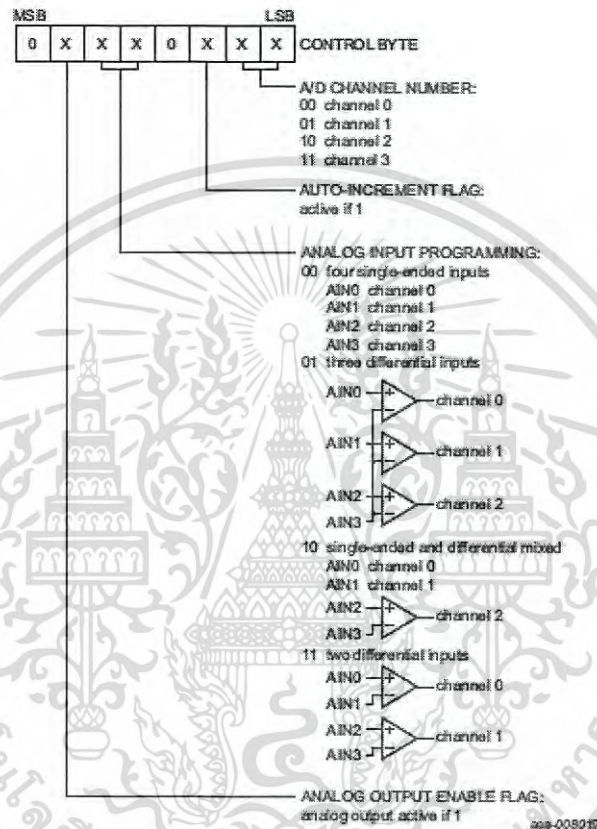
รูปที่ 2.23 รูปแบบการอ้างถึงไอซี PCF8591

2.13.1 การอ้างถึงตำแหน่งของ PCF8591

PCF8591 สามารถเลือกแอดเดรสได้ 8 ตำแหน่ง โดยการกำหนดระดับลอจิก (“0” หรือ “1”) ให้กับขา A0, A1 และ A2 ภายใต้ค่าแอดเดรสที่กำหนดร่วมกับค่าคงที่ประจำตัวชิพ ดังรูปที่ 2.23 จากรูปจะเห็นได้ว่า 4 บิตบนจะถูกกำหนดเป็นค่าคงที่เอาไว้ (1001) ส่วน 3 บิตถัดมาเป็น A2, A1 และ A0 ซึ่งสามารถเปลี่ยนแปลงค่าได้ตามความต้องการ ส่วนบิตสุดท้าย (บิต 0) ใช้สำหรับการกำหนดรูปแบบติดต่อว่าจะอ่าน (“1”) หรือเขียน (“0”) และการเลือกโหมดของ PCF8591 สามารถเลือกได้ 4 รูปแบบด้วยกัน คือ

- 1) บิต 0 และบิต 1 : ใช้ในการเลือกช่องสัญญาณแอนะล็อกอินพุตที่ต้องการ
- 2) บิต 2 : ใช้ในการเลือกรูปแบบการอ่านข้อมูลจากช่องสัญญาณแอนะล็อกอินพุต

- 3) บิต 4 และบิต 5 : ใช้ในการเลือกรูปแบบการทำงานของ PCF8591 สามารถเลือกการทำงานได้ 4 รูปแบบ ดังรูปที่ 2.24
- 4) บิต 6 : ใช้ในการปิดและเปิดส่งสัญญาณแอนะล็อกที่ส่งออกไปยังขา AOUT ถ้าต้องการส่งสัญญาณแอนะล็อกออกไป ต้องกำหนดให้บิตนี้เป็น “1”



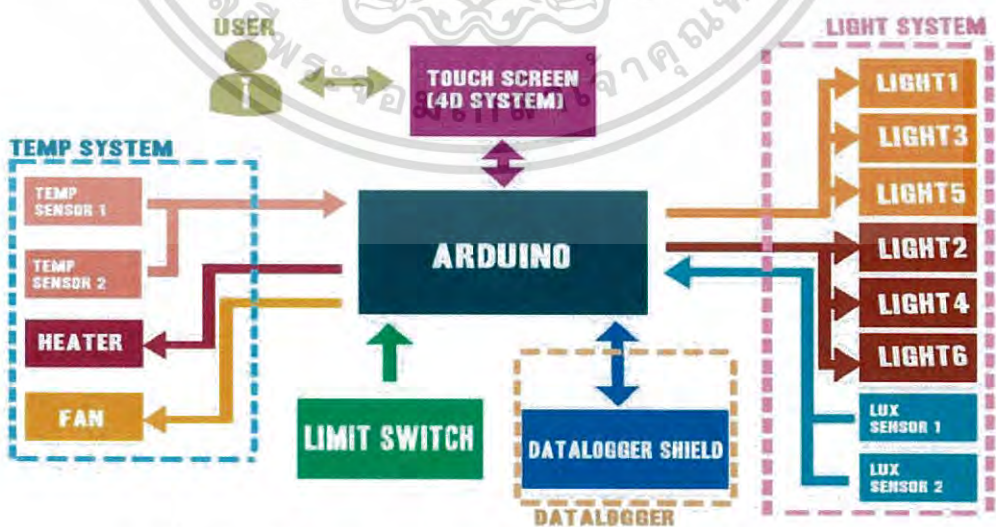
รูปที่ 2.24 โหมดการทำงานต่าง ๆ ของ PCF8591

บทที่ 3

การออกแบบ

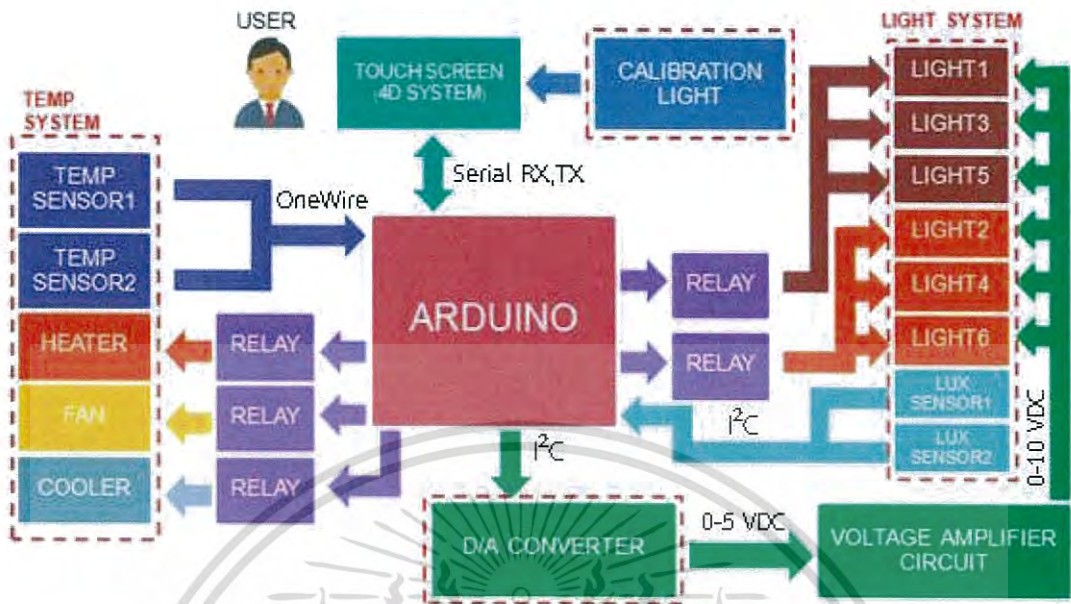
3.1 การออกแบบและการทำงานโดยรวมของระบบ

ในบทนี้จะกล่าวถึงองค์ประกอบของระบบ และการทำงานโดยรวม ซึ่งองค์ประกอบของระบบการทำงานแบบเดิม ดังแสดงในรูปที่ 3.1 และรายละเอียดในการออกแบบซึ่งเป็นการปรับปรุงประสิทธิภาพของเครื่องเขย่าให้มีประสิทธิภาพดีขึ้น โดยการออกแบบระบบนั้นมียังค์ประกอบโดยรวมแสดงได้ดังรูปที่ 3.2 ซึ่งในการควบคุมส่วนต่าง ๆ ภายในระบบนั้น จะใช้บอร์ดอาร์ดูโนเป็นส่วนประมวลผลหลักเพื่อไปควบคุมยังส่วนต่าง ๆ โดยภายในระบบจะประกอบไปด้วย ระบบควบคุมอุณหภูมิ ระบบควบคุมแสงส่องสว่าง ระบบสอบเทียบแสงส่องสว่าง และส่วนแสดงผล โดยระบบควบคุมอุณหภูมิแบบเดิมนั้นประกอบไปด้วย เซนเซอร์วัดอุณหภูมิ ฮีทเตอร์ และพัดลม และในส่วน of ระบบควบคุมอุณหภูมิที่ได้ทำการปรับปรุงประสิทธิภาพนั้นมีการเพิ่มคูลเลอร์เข้ามา ส่วนระบบควบคุมแสงสว่างเดิมประกอบด้วย เซนเซอร์วัดความสว่าง วงจรแปลงแรงดัน และหลอดฟลูออเรสเซนต์ และในส่วน of ระบบควบคุมแสงสว่างที่ได้ทำการปรับปรุงประสิทธิภาพนั้นมีการเพิ่มวงจรแปลงสัญญาณดิจิทัลเป็นแอนะล็อกบัลลัสต์ต่ออิเล็กทรอนิกส์เข้ามา และส่วนระบบสอบเทียบแสงส่องสว่างที่ได้ทำการเพิ่มเข้ามาโดยใช้การเขียนโปรแกรมเพื่อรับค่าแสง และนำไปใช้ในการคำนวณหาค่าแก้ (Correetion) ส่วนสุดท้ายคือส่วนการแสดงผล และติดต่อกับผู้ใช้งานจะเป็นส่วนของหน้าจอสัมผัส



รูปที่ 3.1 บล็อกไดอะแกรมโครงสร้างการทำงานหลักของระบบเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

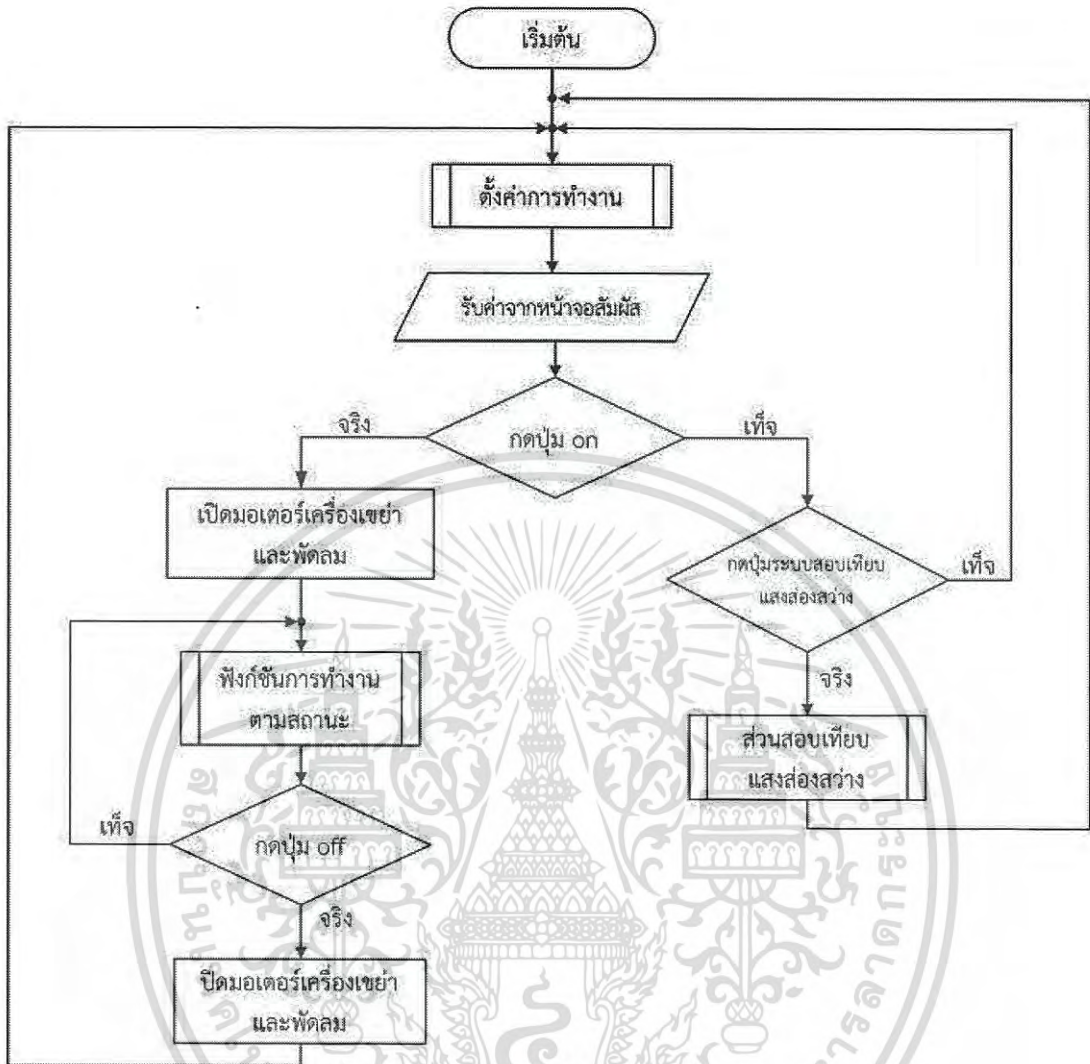


รูปที่ 3.2 บล็อกไดอะแกรมโครงสร้างการทำงานหลักของระบบแบบใหม่

3.2 การออกแบบแผนผังการทำงานของระบบ

3.2.1 โครงสร้างการทำงานหลักของระบบ

จากรูปที่ 3.3 แสดงโครงสร้างการทำงานหลักของระบบที่ได้ปรับปรุงประสิทธิภาพใหม่สามารถอธิบายได้ดังนี้ เริ่มต้นด้วยการตั้งค่าการทำงานต่าง ๆ และรับค่าจากหน้าจอระบบสัมผัส ต่อมาทำการตรวจสอบการกดปุ่ม on ถ้าหากไม่ได้กดปุ่ม on และกดปุ่มเข้าสู่ฟังก์ชันการสอบเทียบแสงส่องสว่าง ก็จะไปยังส่วนสอบเทียบแสงส่องสว่าง ต่อมาจะวนลูปกลับไปรับค่าจากหน้าจอระบบสัมผัสไปเรื่อย ๆ แต่ถ้าหากกดปุ่ม on จะทำการเปิดมอเตอร์เครื่องเขย่าและพัดลม หลังจากนั้นจะเข้าสู่ฟังก์ชันการคำนวณสถานะของการทำงานตามค่าที่ได้รับจากหน้าจอ และทำงานตามสถานะที่คำนวณได้ต่อไป เครื่องเขย่าจะทำงานตามสถานะนี้ต่อไปเรื่อย ๆ จนกว่าจะมีการกดปุ่ม off จึงจะทำการปิดมอเตอร์เครื่องเขย่าและพัดลม แต่หากไม่ได้กดปุ่ม on และไม่ได้กดปุ่มเข้าสู่ฟังก์ชันการสอบเทียบแสงส่องสว่างก็จะวนลูปกลับไปรับค่าจากหน้าจอระบบสัมผัสอีกต่อไปเรื่อย ๆ เสมือนกับว่าทำการกดปุ่ม off ค้างไว้



รูปที่ 3.3 โครงสร้างการทำงานหลักของระบบ

3.2.2 โครงสร้างการทำงานของฟังก์ชันย่อยระบบการสอบเทียบแสงส่องสว่าง

จากรูปที่ 3.4 แสดงโครงสร้างการทำงานของฟังก์ชันย่อยระบบสอบเทียบแสงส่องสว่างสามารถอธิบายได้ดังนี้ เมื่อเข้าสู่ระบบสอบเทียบแสงส่องสว่าง ในส่วนของการกำหนดค่าความสว่างสูงสุด (Max Lux) นั้น จะเริ่มต้นด้วยการตรวจสอบการกดปุ่ม on หากไม่ได้กดปุ่ม on จะทำการปิดมอเตอร์เครื่องเขย่าและพัคลมตามเดิม พร้อมกับรอรับค่าจากหน้าจออีกครั้ง แต่หากกดปุ่ม on ตั้งค่าให้เปิดหลอดไฟให้สว่างสูงสุด และวัดค่าแสงสว่างจากเครื่องวัดความเข้มแสง (Lux meter) จากนั้นตั้งค่าแสงที่สูงสุด ตามที่วัดได้จากเครื่องวัดความเข้มแสงมาตรฐาน และตรวจสอบการกดปุ่ม off หากกดปุ่ม off จะทำการปิดมอเตอร์เครื่องเขย่าและพัคลม พร้อมทั้งเก็บค่าแสงที่สูงสุด และค่าความสว่างจากเซนเซอร์ไปเก็บไว้ ในส่วนของการกำหนดค่าความสว่างกึ่งหนึ่ง (Half Lux) นั้นทำกระบวนการเช่นเดียวกับการกำหนดค่าความสว่างสูงสุด (Max Lux) หลังจากนั้นทำการตรวจสอบการกดปุ่ม on

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกดปุ่ม on system ของระบบสอบเทียบแสงส่องสว่าง หากมีการกดปุ่ม on system จริง จะทำการนำค่าที่เก็บไว้ในส่วนของการกำหนดค่าความสว่างสูงสุด (Max Lux) และในส่วนของการกำหนดค่าความสว่างกึ่งหนึ่ง (Half Lux) ไปคำนวณหาอัตราส่วนขยายใหม่และค่าคงที่ใหม่ แต่หากไม่ได้กดปุ่ม on system จะทำการรอการกดปุ่ม on system สุดท้ายตรวจสอบการกดปุ่ม off system ของระบบสอบเทียบแสงส่องสว่าง หากมีการกดปุ่ม off system จริง จะทำการเก็บค่าอัตราส่วนขยายใหม่และค่าคงที่ใหม่ไปแทนค่าอัตราส่วนขยายเดิมและค่าคงที่เดิม แต่หากไม่ได้กดปุ่ม off system จะทำการรอการกดปุ่ม off system หลังจากนั้นจึงออกจากระบบการสอบเทียบแสงส่องสว่าง ดังนั้นในการคำนวณหาอัตราส่วนขยายใหม่และค่าคงที่ใหม่ ส่วนเซนเซอร์ตัวที่ 1 สามารถหาค่าได้ตามสมการที่ 3.1 และ 3.2 ส่วนเซนเซอร์ตัวที่ 2 จะใช้สมการที่ 3.3 และ 3.4

$$y_1 = m_1 x_1 + c \quad (3.1)$$

$$y_3 = m_1 x_3 + c \quad (3.2)$$

$$y_2 = m_2 x_2 + d \quad (3.3)$$

$$y_4 = m_2 x_4 + d \quad (3.4)$$

เมื่อ

y_1, y_2 = ค่าแสงส่องสว่างที่วัดได้จากเครื่องวัดความเข้มแสงเมื่อเปิดแสงส่องสว่างสูงสุด (Max Lux)

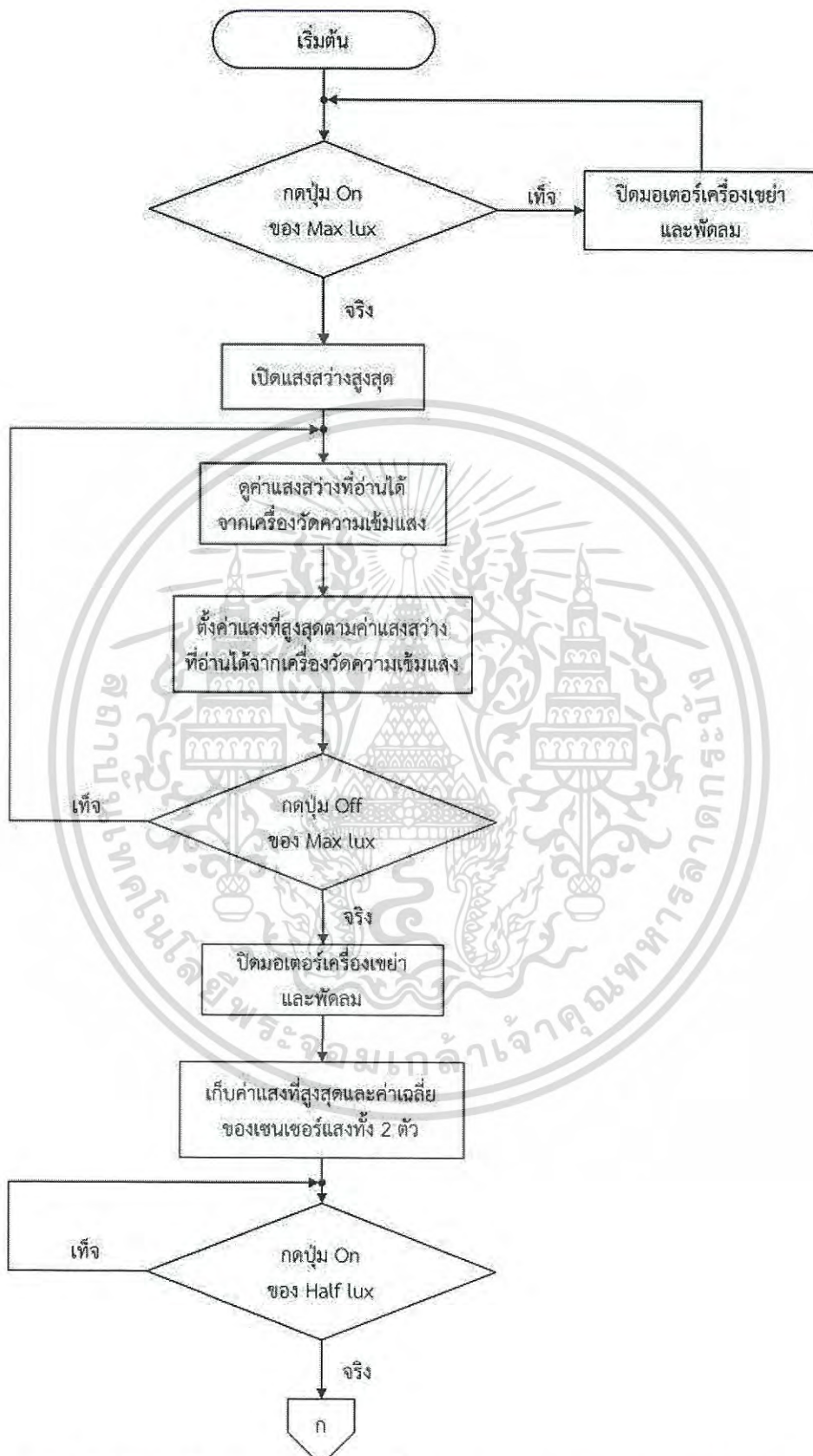
y_3, y_4 = ค่าแสงส่องสว่างที่วัดได้จากเครื่องวัดความเข้มแสงเมื่อเปิดแสงส่องสว่างกึ่งหนึ่ง (Half Lux)

m_1, m_2 = ค่าอัตราส่วนขยายใหม่ (new ratio)

x_1, x_2 = ค่าแสงส่องสว่างที่วัดได้จากเซนเซอร์เมื่อเปิดแสงส่องสว่างสูงสุด (Max Lux)

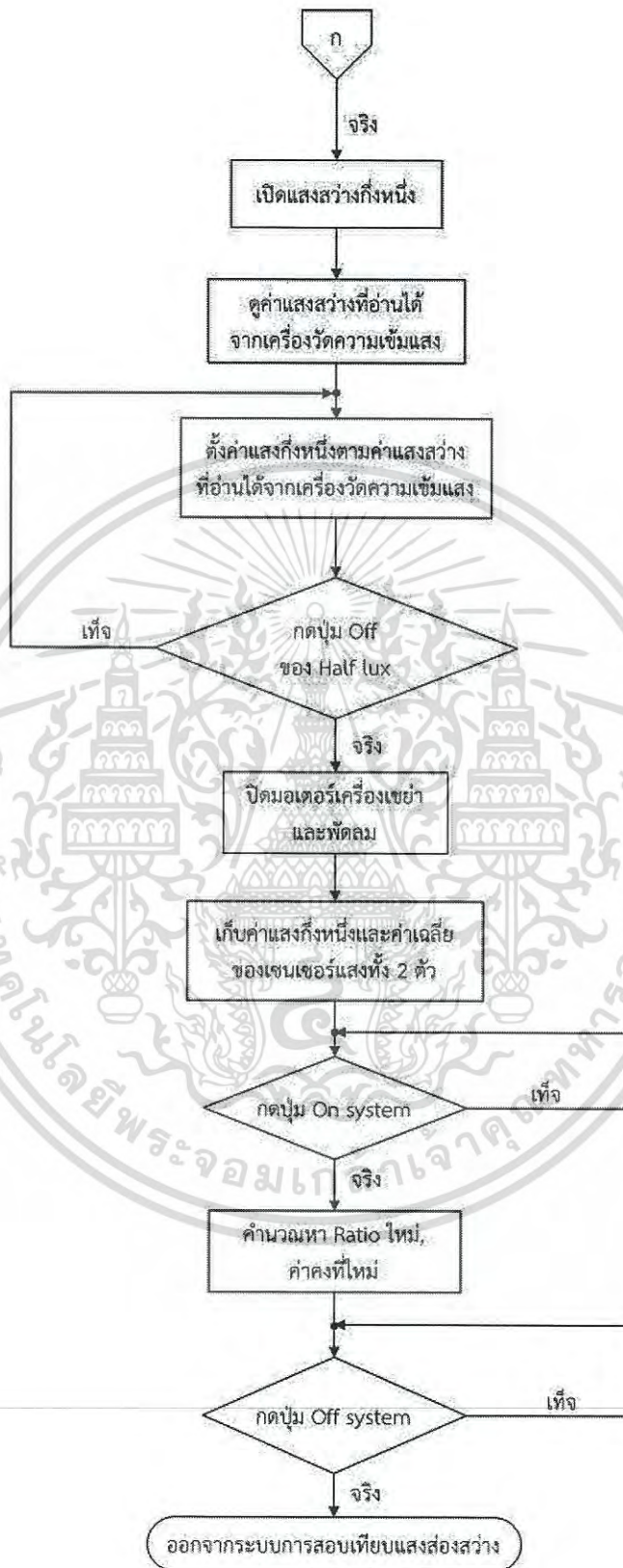
x_3, x_4 = ค่าแสงส่องสว่างที่วัดได้จากเซนเซอร์เมื่อเปิดแสงส่องสว่างกึ่งหนึ่ง (Half Lux)

c, d = ค่าคงที่ใหม่ (new constant)



รูปที่ 3.4 โครงสร้างการทำงานของฟังก์ชันย่อยระบบการสอบเทียบแสงส่องสว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

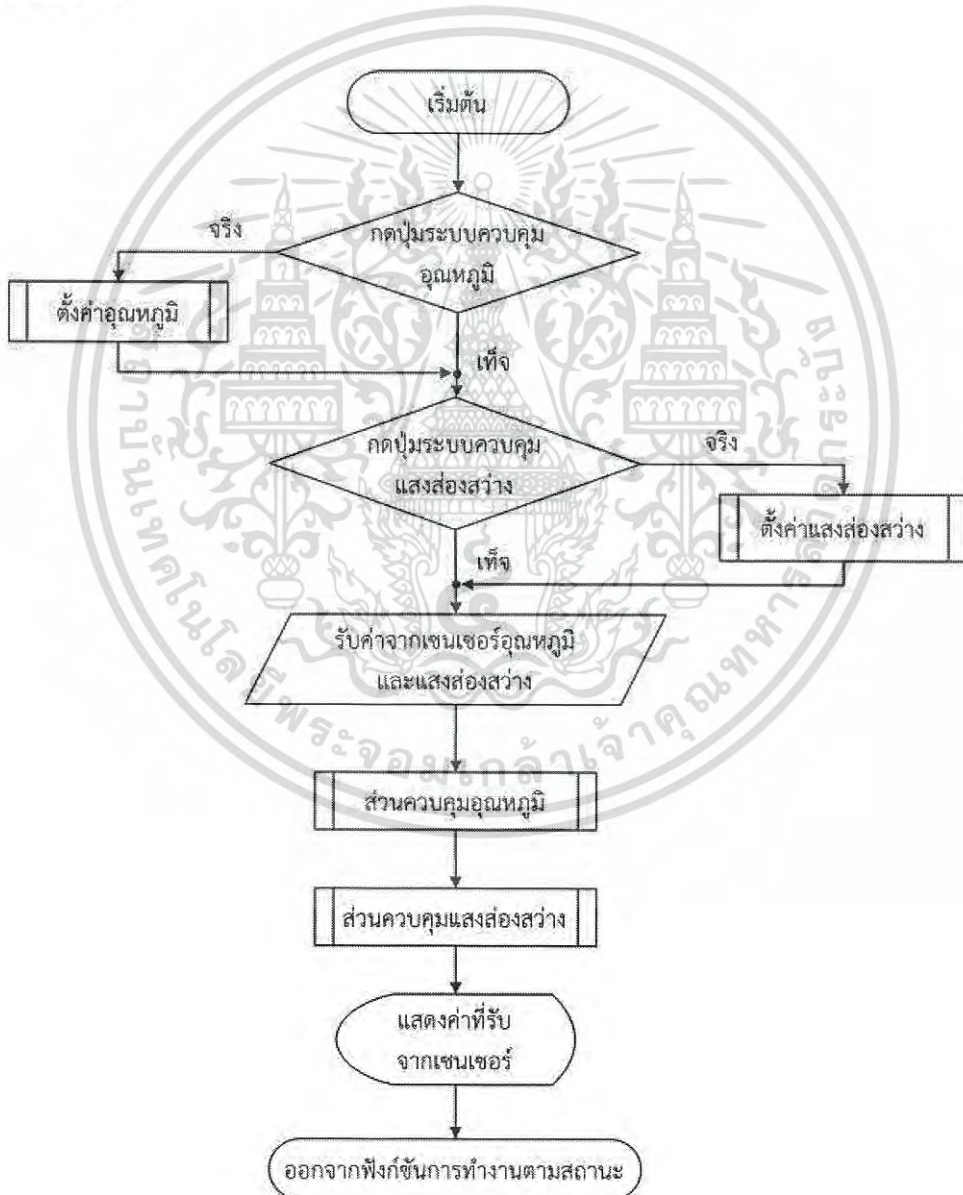


รูปที่ 3.4 โครงสร้างการทำงานของฟังก์ชันย่อยระบบการสอบเทียบแสงส่องสว่าง (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 โครงสร้างการทำงานของฟังก์ชันการทำงานตามสถานะ

จากรูปที่ 3.5 แสดงโครงสร้างการทำงานของฟังก์ชันการทำงานตามสถานะสามารถอธิบายได้ดังนี้ จะเริ่มต้นด้วยการตรวจสอบการกดปุ่มเข้าสู่ฟังก์ชันต่าง ๆ กล่าวคือ ตรวจสอบการกดปุ่มควบคุมระบบควบคุมอุณหภูมิ และตรวจสอบการกดปุ่มระบบควบคุมแสงส่องสว่าง หากมีการกดปุ่มใดก็จะเข้าไปตั้งค่าการใช้งานในส่วนของระบบนั้น ๆ หลังจากนั้นจะเข้าสู่การรับค่าจากเซนเซอร์อุณหภูมิและแสงส่องสว่างมาคำนวณ โดยเครื่องจะทำงานในส่วนควบคุมอุณหภูมิและส่วนควบคุมแสงส่องสว่างตามลำดับ พร้อมกับนำค่าต่าง ๆ ไปแสดงยังหน้าจอและออกจากฟังก์ชันการทำงานตามสถานะ

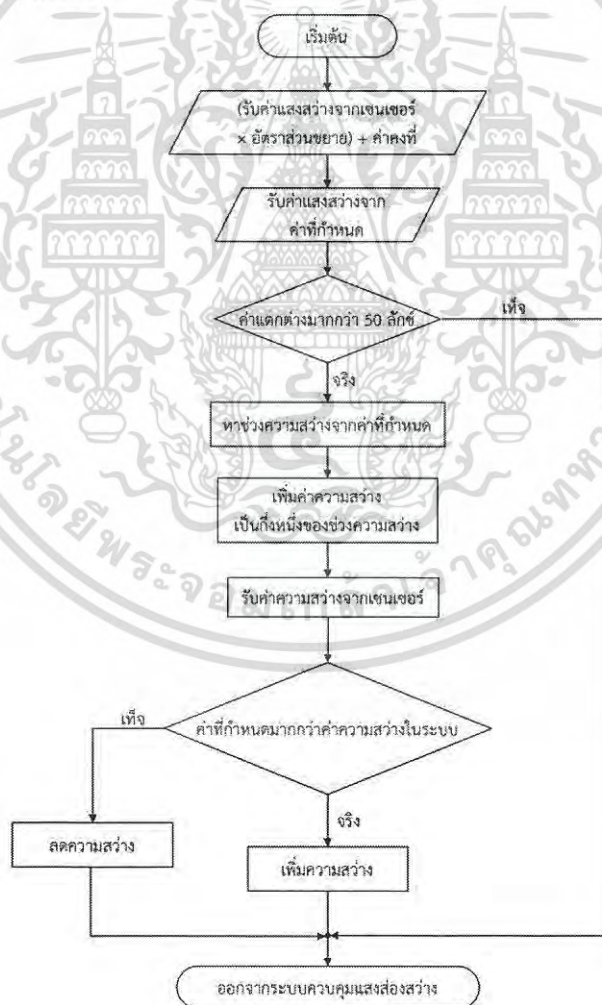


รูปที่ 3.5 โครงสร้างการทำงานของฟังก์ชันการทำงานตามสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 โครงสร้างการทำงานของฟังก์ชันย่อยระบบควบคุมแสงส่องสว่าง

จากรูปที่ 3.6 แสดงโครงสร้างการทำงานของฟังก์ชันย่อยระบบควบคุมแสงส่องสว่างสามารถอธิบายได้ดังนี้ เมื่อเข้าสู่ระบบควบคุมแสงส่องสว่าง จะเริ่มต้นด้วยการรับค่าความสว่างจากเซนเซอร์มาคูณกับอัตราส่วนขยาย และบวกกับค่าคงที่แล้วนำไปเปรียบเทียบกับที่กำหนดไว้ จากนั้นจึงนำค่าที่ได้มาทำการตรวจสอบ หากมีความแตกต่างไม่มากกว่า 50 ลักซ์ ก็จะออกจากระบบควบคุมแสงส่องสว่าง แต่หากมีความแตกต่างกันมากกว่า 50 ลักซ์ ก็จะเข้าสู่การหาค่าช่วงของความสว่างจากค่าที่กำหนดแล้วจึงเพิ่มความสว่างให้มีค่าเท่ากับกึ่งหนึ่งของค่าช่วงที่หาได้ เช่น ค่าที่กำหนดอยู่ในช่วง 4,000 ถึง 5,000 ลักซ์ ก็จะเพิ่มความสว่างให้เป็น 4,500 ลักซ์ เป็นต้น หลังจากนั้นจะรับค่าจากเซนเซอร์แสงอีกครั้ง หากค่าที่กำหนดมากกว่าค่าความสว่างในระบบจริงก็จะทำการเพิ่มค่าความสว่างขึ้น แต่หากค่าที่กำหนดน้อยกว่าค่าความสว่างในระบบก็จะลดค่าความสว่างลง หลังจากนั้นจึงออกจากระบบควบคุมแสงส่องสว่าง

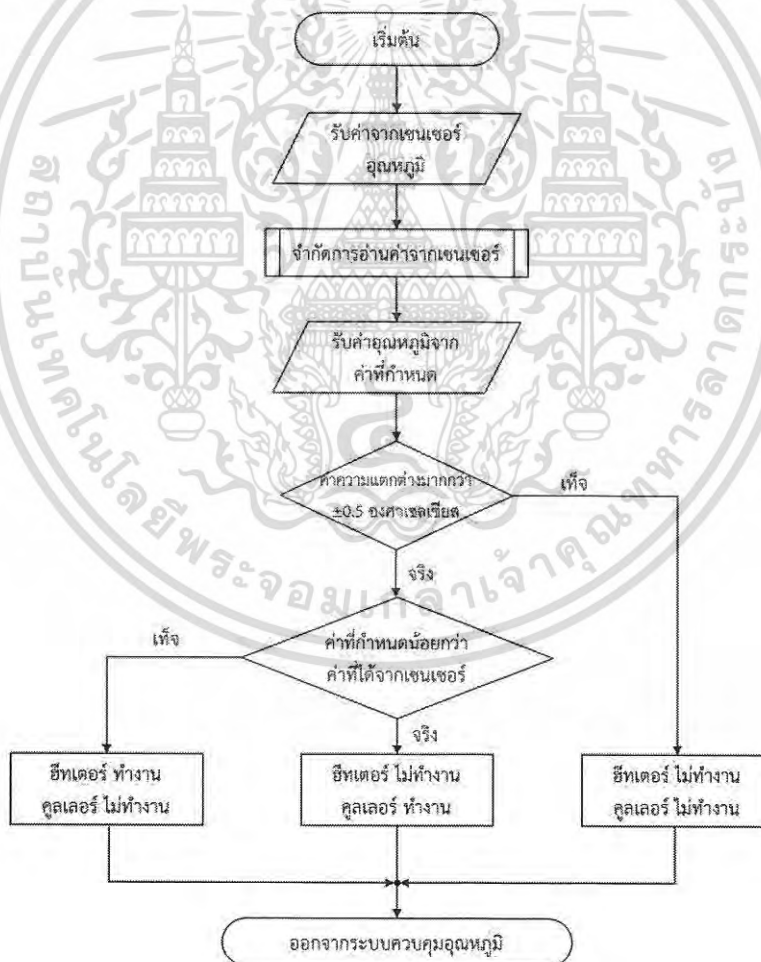


รูปที่ 3.6 โครงสร้างการทำงานของฟังก์ชันย่อยระบบควบคุมแสงส่องสว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 โครงสร้างการทำงานของฟังก์ชันย่อยระบบควบคุมอุณหภูมิ

จากรูปที่ 3.7 แสดงโครงสร้างการทำงานของฟังก์ชันย่อยระบบควบคุมอุณหภูมิสามารถอธิบายได้ดังนี้ เมื่อเข้าสู่ระบบควบคุมอุณหภูมิจะทำการรับค่าจากเซนเซอร์อุณหภูมิและเข้าสู่ฟังก์ชันจำกัดการอ่านค่าจากเซนเซอร์ ต่อมาคำนวณหาค่าความแตกต่างระหว่างค่าที่กำหนด (Set point) กับค่าที่ได้รับจากเซนเซอร์อุณหภูมิ หากตรวจสอบแล้วค่าที่คำนวณได้แตกต่างกันมากกว่า 0.5 องศาเซลเซียสจริง ก็จะตรวจสอบต่อว่าค่าที่กำหนดนั้นน้อยกว่าอุณหภูมิภายในระบบหรือไม่ หากน้อยกว่าจริงก็จะทำการปิดฮีตเตอร์และเปิดพัดลมระบายอากาศ แต่หากมากกว่าก็จะทำการเปิดฮีตเตอร์และปิดพัดลมระบายอากาศ หลังจากนั้นจึงออกจากระบบควบคุมอุณหภูมิ โดยหากตรวจสอบแล้วค่าที่คำนวณได้ไม่แตกต่างกันมากกว่า 0.5 องศาเซลเซียสก็จะทำการปิดฮีตเตอร์และปิดพัดลมระบายอากาศ หลังจากนั้นจึงออกจากระบบควบคุมอุณหภูมิ

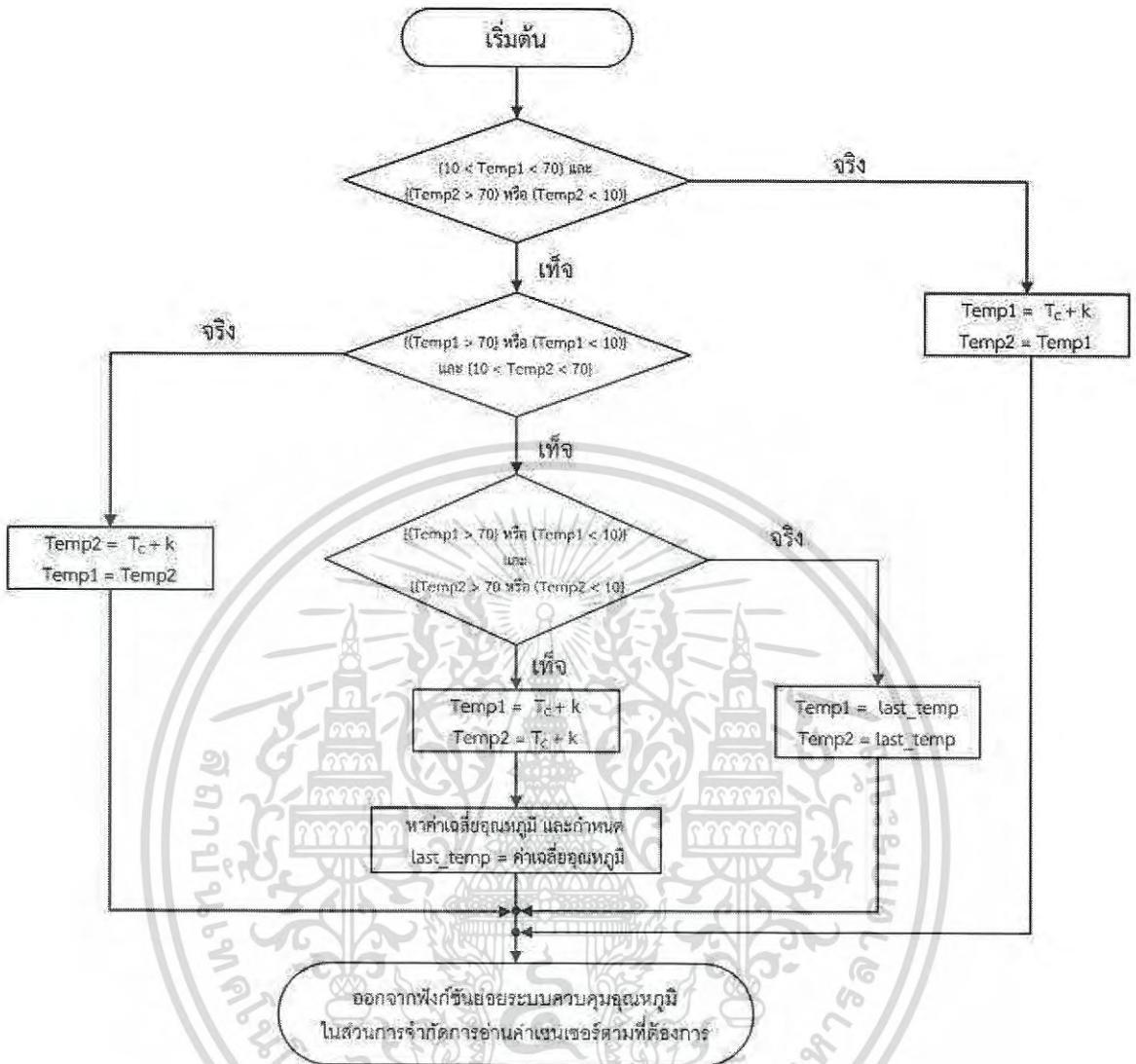


รูปที่ 3.7 โครงสร้างการทำงานของฟังก์ชันย่อยระบบควบคุมอุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5.1 โครงสร้างการทำงานของฟังก์ชันย่อยระบบควบคุมอุณหภูมิในส่วนการ จำกัดการอ่านค่าเซนเซอร์ตามที่ต้องการ

จากรูปที่ 3.8 โครงสร้างการทำงานของฟังก์ชันย่อยระบบควบคุมอุณหภูมิในส่วนการจำกัดการอ่านค่าเซนเซอร์ตามที่ต้องการสามารถอธิบายได้ดังนี้ จากเดิมเมื่อเข้าสู่ระบบควบคุมอุณหภูมิจะทำการรับค่าจากเซนเซอร์อุณหภูมิ แล้วมาคำนวณหาค่าความแตกต่างระหว่างค่าที่กำหนด (Set point) กับค่าที่ได้รับจากเซนเซอร์อุณหภูมิ แต่ในบางครั้งค่าที่ได้รับมาจากเซนเซอร์อุณหภูมิเกิดผิดพลาดติด ๆ กันอย่างต่อเนื่อง เช่น อ่านค่าได้ -127 องศาเซลเซียสทำให้เกิดอาการเครื่องเขย่าเปิดฮีตเตอร์ค้าง ส่งผลให้ไม่สามารถควบคุมอุณหภูมิได้ตามที่ต้องการ ดังนั้นจึงทำการเขียนโปรแกรมจำกัดการรับค่าเซนเซอร์อุณหภูมิ ดังโครงสร้างการทำงานของฟังก์ชันย่อยระบบควบคุมอุณหภูมิในส่วนการจำกัดการอ่านค่าเซนเซอร์ตามที่ต้องการดังรูปที่ 3.7 สามารถอธิบายได้ดังนี้ เมื่อเข้าสู่ระบบควบคุมอุณหภูมิจะทำการรับค่าจากเซนเซอร์อุณหภูมิทั้ง 2 ตัว ในขั้นแรกทำการตรวจสอบว่าค่าที่รับมาจากเซนเซอร์อุณหภูมิทั้ง 2 ตัว ถ้าค่าที่รับมาในตัวใดตัวหนึ่งมีค่าอยู่ในช่วง 10 ถึง 70 องศาเซลเซียสและค่าที่รับมาอีกตัว มีค่านอกเหนือช่วง 10 ถึง 70 องศาเซลเซียส กำหนดให้เซนเซอร์ตัวที่อ่านค่าได้ในช่วง 10 ถึง 70 องศาเซลเซียสเป็นค่าที่จะนำไปคำนวณ ต่อมาบวกกับค่าคงที่ ซึ่งเป็นค่าแก้ของเซนเซอร์วัดอุณหภูมิ ซึ่งมีค่าความผิดพลาดในการอ่านค่า โดยในโครงงานนี้ได้ทดสอบเพื่อหาค่าแก้เฉลี่ยที่ให้ค่าที่ใกล้เคียงกับค่าจริงอยู่ที่ประมาณ +0.7 ส่วนเซนเซอร์ตัวที่อ่านค่าได้นอกเหนือช่วง 10 ถึง 70 องศาเซลเซียสนั้น ให้กำหนดค่าเท่ากับเซนเซอร์ตัวที่อ่านค่าได้ในช่วง 10 ถึง 70 องศาเซลเซียส แต่ถ้าเกิดกรณีที่รับค่าจากเซนเซอร์อุณหภูมิทั้ง 2 ตัวแล้วพบว่ามีความนอกเหนือช่วง 10 ถึง 70 องศาเซลเซียสทั้งคู่ ให้ค่าที่จะนำไปคำนวณต่อเป็นค่าอุณหภูมิที่อ่านได้จากก่อนหน้านี้ จากนั้นจึงเข้าสู่ขั้นตอนการคำนวณการหาค่าความแตกต่างระหว่างค่าที่กำหนด (Set point) กับค่าที่ได้รับจากเซนเซอร์อุณหภูมิตามปกติ



รูปที่ 3.8 โครงสร้างการทำงานของฟังก์ชันย่อยระบบควบคุมอุณหภูมิในส่วนการจำกัดการอ่านค่าเซนเซอร์ตามที่ต้องการ

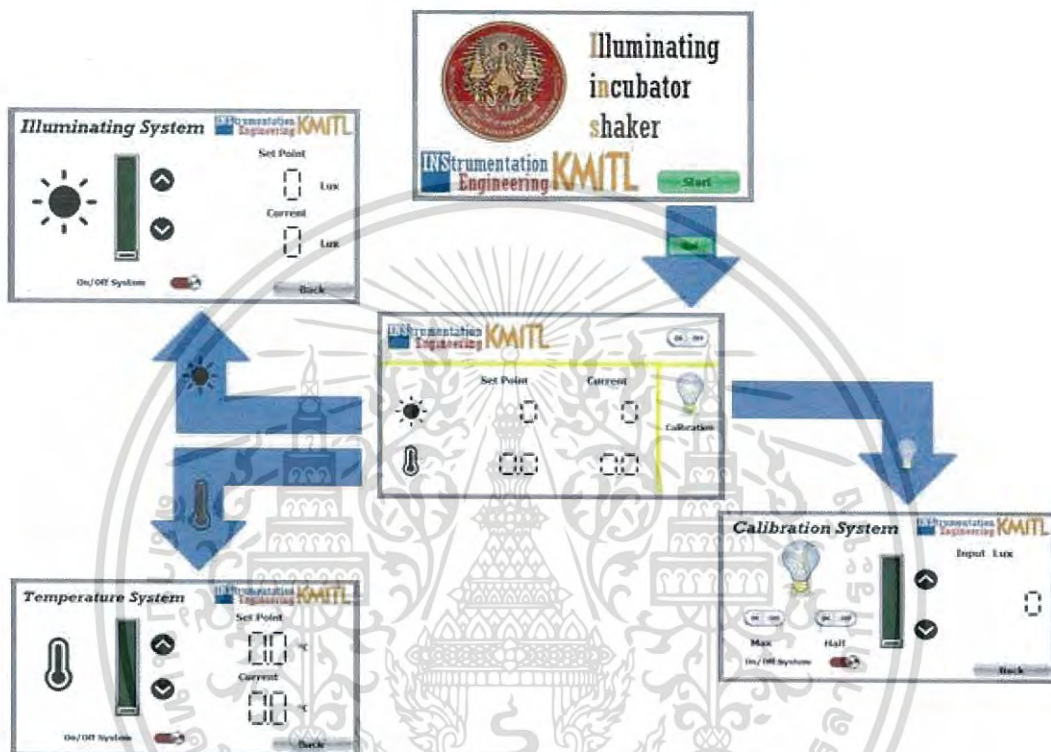
3.3 การออกแบบหน้าจอสำหรับใช้งาน

การออกแบบหน้าจอสำหรับใช้งานนั้น เป็นการออกแบบส่วนแสดงผลบนหน้าจอระบบสัมผัสไปแสดงยังผู้ใช้งาน ผ่านโปรแกรม Workshop4 โดยการออกแบบจะประกอบไปด้วย หน้าจอเริ่มต้น หน้าจอแสดงผลหลัก หน้าจอตั้งค่าอุณหภูมิ หน้าจอแสงส่องสว่าง และหน้าจอในการสอบเทียบแสงส่องสว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 โครงสร้างการเชื่อมโยงของหน้าจอระบบสัมผัส

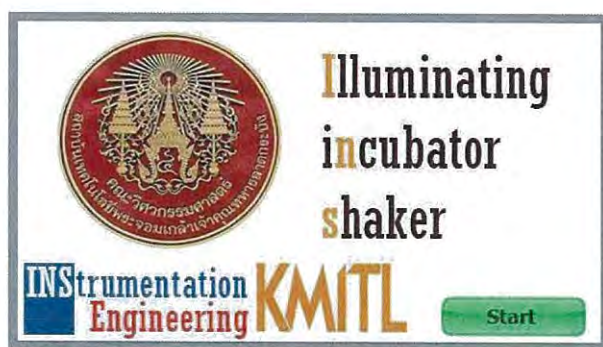
เมื่อกดปุ่ม Start จากหน้าเริ่มต้นจะเข้าสู่หน้าจอแสดงผลหลัก ซึ่งเมื่อกดปุ่มต่าง ๆ บนหน้าจอแสดงผลหลัก จะนำไปสู่หน้าจออื่น ๆ เช่น กดปุ่มสัญลักษณ์ดวงอาทิตย์ ปรอท และหลอดไฟ เพื่อไปยังหน้าการตั้งค่าแสงส่องสว่าง อุณหภูมิและสอบเทียบแสงส่องสว่างตามลำดับ ดังแสดงในรูปที่ 3.9



รูปที่ 3.9 การเชื่อมโยงของหน้าจอระบบสัมผัส

3.3.2 หน้าจอเริ่มต้น

สำหรับในหน้าแรกนั้น สามารถแสดงดังรูปที่ 3.10 จะเป็นหน้าที่แสดงเมื่อเครื่องถูกเปิดการใช้งานในครั้งแรก และเมื่อทำการกดปุ่ม Start จะนำไปสู่หน้าแสดงผลหลักได้

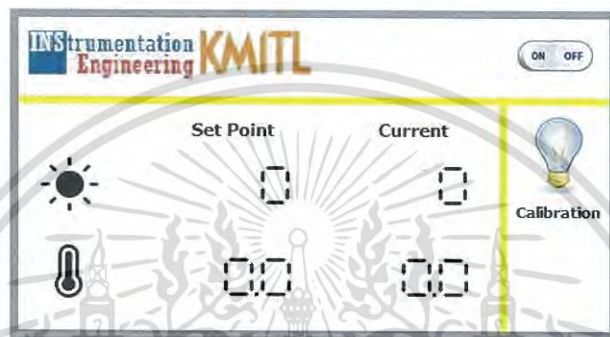


รูปที่ 3.10 หน้าจอเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 หน้าจอแสดงผล

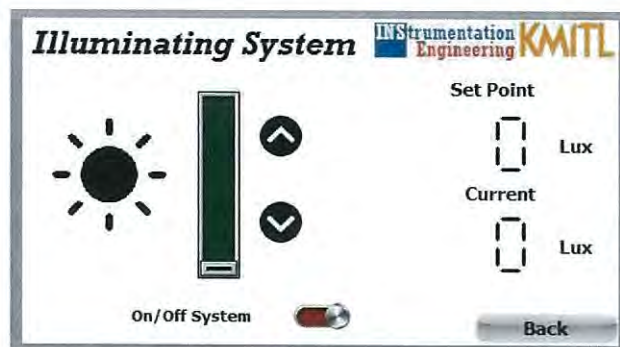
เมื่อกดปุ่ม Start จากหน้าจอเริ่มต้น จะเข้ามาในหน้าแสดงผลการทำงานของเครื่อง เขย่าดังรูปที่ 3.11 โดยจะแสดงค่าที่กำหนด (Set point) และค่าที่วัดได้ภายในระบบแสงส่องสว่าง และอุณหภูมิ (Current) การตั้งค่าแสงส่องสว่างสามารถทำได้โดยการกดปุ่มสัญลักษณ์รูปดวงอาทิตย์ และการตั้งค่าอุณหภูมิสามารถทำได้โดยการกดปุ่มสัญลักษณ์รูปปรอท สุดท้ายสามารถกดปุ่มสัญลักษณ์รูปหลอดไฟเพื่อไปยังหน้าตั้งค่าการสอบเทียบแสงส่องสว่างภายในระบบ



รูปที่ 3.11 หน้าจอแสดงผล

3.3.4 หน้าจอตั้งค่าแสงส่องสว่าง

เมื่อมีการกดปุ่มสัญลักษณ์ดวงอาทิตย์จะนำเข้าสู่หน้าจอตั้งค่าแสงส่องสว่าง ดังรูปที่ 3.12 เป็นหน้าจอที่ใช้ตั้งค่าแสงส่องสว่าง โดยเมื่อทำการเลื่อนแถบเลื่อนจะเป็นการตั้งค่าแสงส่องสว่างที่ต้องการ ค่าตัวเลขที่เลื่อนจากแถบเลื่อนแสดงใน (Set Point) และสามารถปรับค่าเพิ่มขึ้นหรือลดลงแบบเป็นสเตปได้ โดยมีความละเอียดของการตั้งค่าในระดับ 100 ลักซ์ โดยการกดปุ่มเพิ่มหรือลดข้างแถบเลื่อน ส่วนค่าแสงส่องสว่างที่วัดได้ภายในระบบแสดงใน (Current) สามารถเลือกเปิดหรือปิดการควบคุมในส่วนของแสงส่องสว่างได้โดยการกดปุ่ม On/Off System และสามารถออกจากหน้าการตั้งค่าแสงส่องสว่างกลับไปยังหน้าแสดงผลหลักด้วยการกดปุ่ม Back

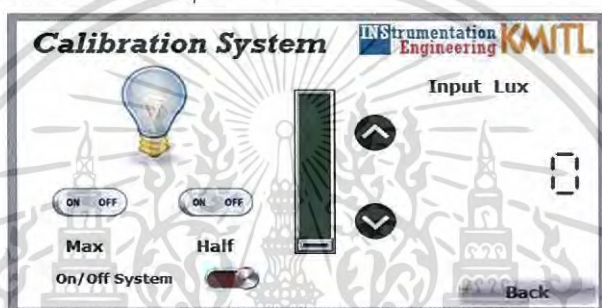


รูปที่ 3.12 หน้าจอตั้งค่าแสงส่องสว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.5 หน้าจอตั้งค่าการสอบเทียบแสงส่องสว่าง

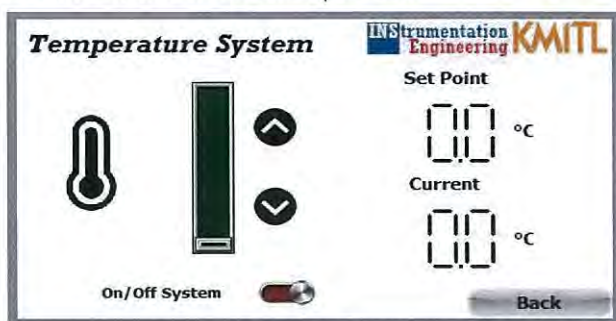
เมื่อมีการกดปุ่มสัญลักษณ์หลอดไฟ จะนำมาสู่หน้าจอตั้งค่าการสอบเทียบแสงส่องสว่างดังรูปที่ 3.13 เป็นหน้าที่ใช้ตั้งค่าการสอบเทียบแสงส่องสว่าง แบ่งเป็น 2 ส่วนคือ เปิดแสงส่องสว่างสูงสุด (Max Lux) และเปิดแสงส่องสว่างกึ่งหนึ่ง (Half Lux) สามารถทำได้โดยการกดปุ่ม On ที่ Max และกดปุ่ม On ที่ Half ตามลำดับ โดยเมื่อทำการเลื่อนแถบเลื่อนจะเป็นการตั้งค่าแสงส่องสว่างที่วัดได้จากเครื่องวัดความเข้มแสง ค่าตัวเลขที่เลื่อนจากแถบเลื่อนแสดงใน (Input Lux) และสามารถปรับค่าเพิ่มขึ้นหรือลดลงแบบเป็นสเตปได้ โดยมีความละเอียดของการตั้งค่าในระดับ 100 ลักซ์ โดยการกดปุ่มเพิ่มหรือลดข้างแถบเลื่อน และสามารถออกจากหน้าการตั้งค่าการสอบเทียบแสงส่องสว่างกลับไปยังหน้าแสดงผลหลักด้วยการกดปุ่ม Back



รูปที่ 3.13 หน้าจอตั้งค่าการสอบเทียบแสงส่องสว่าง

3.3.6 หน้าจอตั้งค่าอุณหภูมิ

เมื่อกดสัญลักษณ์รูปปรอทจะเข้ามาถึงหน้าตั้งค่าอุณหภูมิ ดังแสดงในรูปที่ 3.14 เป็นหน้าสำหรับการตั้งค่าอุณหภูมิ โดยเมื่อทำการเลื่อนแถบเลื่อนจะเป็นการตั้งค่าอุณหภูมิที่ต้องการ ค่าตัวเลขที่เลื่อนจากแถบเลื่อนแสดงใน (Set Point) และสามารถปรับค่าเพิ่มขึ้นหรือลดลงแบบเป็นสเตปได้ โดยมีความละเอียดของการตั้งค่าได้ในระดับ 0.5 องศาเซลเซียส โดยการกดปุ่มเพิ่มหรือลดข้างแถบเลื่อน ส่วนค่าอุณหภูมิที่วัดได้ภายในระบบแสดงใน (Current) สามารถเลือกเปิดหรือปิดการควบคุมในส่วนของอุณหภูมิได้โดยการกดปุ่ม On/Off System และสามารถออกจากหน้าตั้งค่าอุณหภูมิกลับไปยังหน้าแสดงผลหลักด้วยการกดปุ่ม Back



รูปที่ 3.14 หน้าจอตั้งค่าอุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การดำเนินงาน

4.1 บทนำ

บทนี้จะกล่าวถึงการดำเนินงานและเชื่อมต่ออุปกรณ์หลังจากที่ได้ทำการออกแบบ โดยการดำเนินงานจะแบ่งออกเป็นส่วนตัว่าง ๆ ประกอบด้วย ระบบควบคุมแสงส่องสว่าง ระบบควบคุมอุณหภูมิ และระบบสอบเทียบแสงส่องสว่าง

4.2 ระบบควบคุมแสงส่องสว่าง

ระบบควบคุมแสงส่องสว่าง เป็นระบบที่ใช้ควบคุมแสงส่องสว่างภายในเครื่อง ดังรูปที่ 4.1 ซึ่งประกอบไปด้วย

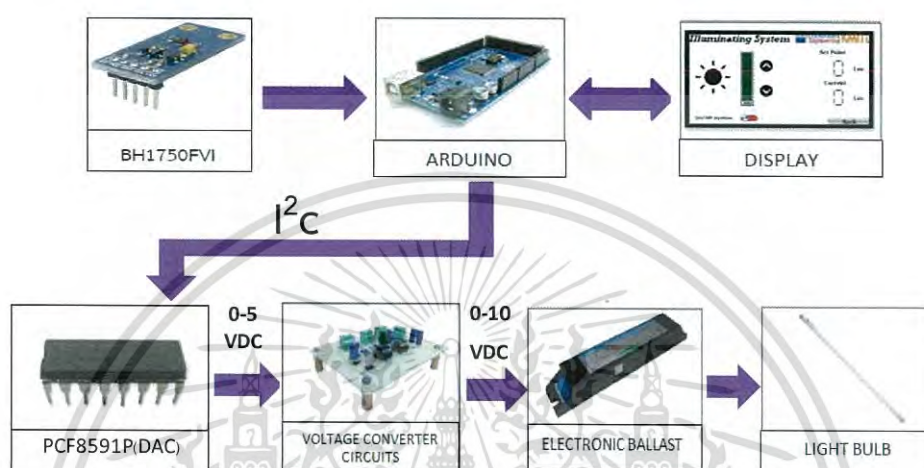
- 1) บอร์ดอาร์ดูโน ทำหน้าที่เป็นส่วนควบคุม
- 2) เซนเซอร์แสง (BH1750FVI) ทำหน้าที่ตรวจวัดแสงสว่างภายในเครื่อง
- 3) วงจรแปลงสัญญาณดิจิทัลเป็นแอนะล็อก ทำหน้าที่แปลงสัญญาณดิจิทัลจากบอร์ดอาดูโนเป็นสัญญาณแอนะล็อกก่อนส่งไปวงจรขยายสัญญาณ
- 4) วงจรขยายสัญญาณ ทำหน้าที่ขยายสัญญาณ 0 ถึง 5 Vdc ซึ่งรับมาจากวงจรแปลงสัญญาณดิจิทัลเป็นแอนะล็อกเพื่อขยายแรงดันเป็น 0 ถึง 10 Vdc เพื่อไปควบคุมความสว่างผ่านบัลลาสต์อิเล็กทรอนิกส์
- 5) บัลลาสต์อิเล็กทรอนิกส์ ทำหน้าที่ปรับความสว่างของหลอดฟลูออเรสเซนต์
- 6) หลอดฟลูออเรสเซนต์ ทำหน้าที่ให้แสงสว่างภายในเครื่อง
- 7) หน้าจอร์บบสัมผัส ทำหน้าที่เป็นส่วนแสดงผลและกำหนดค่า

4.2.1 โครงสร้างระบบควบคุมแสงส่องสว่าง

จากรูปที่ 4.1 แสดงโครงสร้างระบบควบคุมแสงส่องสว่างซึ่งมีบอร์ดอาร์ดูโนทำหน้าที่เป็นส่วนควบคุม โดยบอร์ดอาร์ดูโนจะรับค่าความสว่างจากเซนเซอร์แสงและหน้าจอสัมผัส และนำมาประมวลผลและส่งสัญญาณเป็นแบบ I^2C เข้าสู่วงจรแปลงสัญญาณทางดิจิทัลเป็นแอนะล็อก (Digital to Analog Converter) ซึ่งใช้ไอซีเบอร์ PCF8591P เป็นตัวหลักของวงจร และเมื่อแรงดันไฟฟ้าจากบอร์ดอาดูโนซึ่งเป็นสัญญาณดิจิทัล 0 ถึง 5 โวลต์ ผ่านวงจรแปลงสัญญาณทางดิจิทัลเป็นแอนะล็อก (Digital to Analog Converter) นี้จะได้แรงดันไฟฟ้าเอาต์พุตเป็น 0 ถึง 5 Vdc และผ่านวงจรขยายสัญญาณให้เป็น 0 ถึง 10 Vdc เพราะว่าบัลลาสต์อิเล็กทรอนิกส์ทำงานที่ 0 ถึง 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

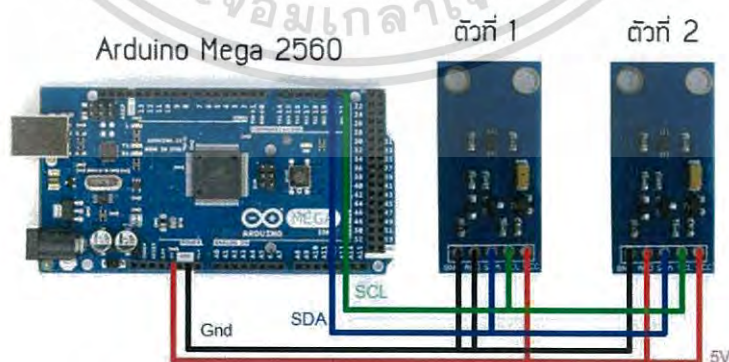
โวลต์ และมีอินพุตเป็นแบบแอนะล็อก ทำให้เมื่อรับค่าแรงดัน 0 ถึง 10 โวลต์ แบบแอนะล็อกและส่งต่อไปควบคุมความสว่างของหลอดฟลูออเรสเซนต์ ส่งผลให้หลอดฟลูออเรสเซนต์ทำงานได้อย่างมีประสิทธิภาพไม่เกิดแสงที่เป็นลักษณะของระลอกคลื่นเหมือนกับการใช้ PWM (Pulse Width Modulation)



รูปที่ 4.1 โครงสร้างของระบบควบคุมแสงส่องสว่าง

4.2.2 การรับค่าความสว่างจากเซนเซอร์วัดแสง (BH1750FVI)

ในส่วนการตรวจวัดค่าความสว่างนั้นจะใช้เซนเซอร์ BH1750FVI โมดูล GY-30 จำนวน 2 ตัว ในการวัดค่าความสว่าง ซึ่งการเชื่อมต่อเป็นไปตามรูปที่ 4.2 โดยสัญญาณที่ได้จากเซนเซอร์นั้นเป็นสัญญาณดิจิทัล และสื่อสารกับบอร์ดไมโครคอนโทรลเลอร์แบบ I²C ผ่านขา SCL และ SDA

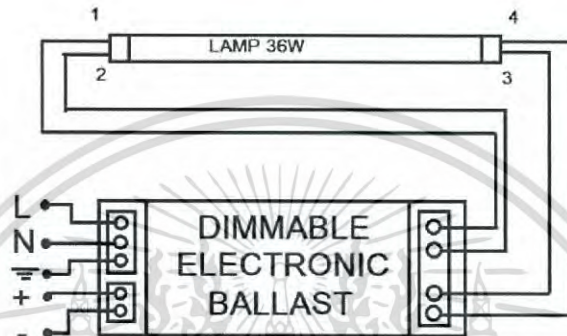


รูปที่ 4.2 การเชื่อมต่อ BH1750FVI เข้ากับบอร์ดอาร์ดูโน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 การใช้งานบัลลาสต์อิเล็กทรอนิกส์กับหลอดฟลูออเรสเซนต์

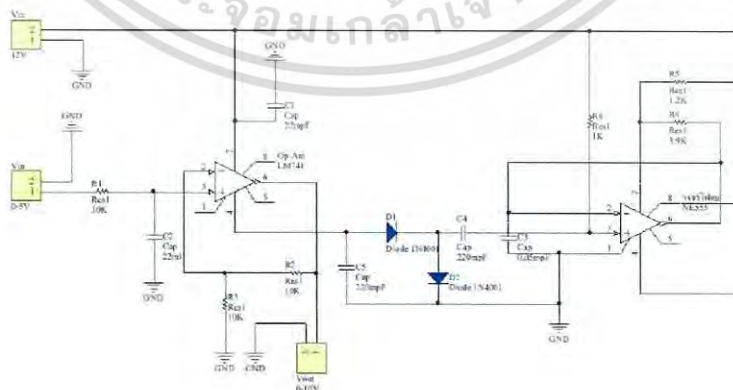
บัลลาสต์อิเล็กทรอนิกส์ชนิดปรับค่าได้ นั้นสามารถควบคุมแสงสว่างของหลอดไฟฟลูออเรสเซนต์ โดยการควบคุมผ่านแรงดันไฟฟ้ากระแสตรง 1 ถึง 10 โวลต์ เข้ามาควบคุมความสว่าง การเชื่อมต่อกับหลอดฟลูออเรสเซนต์เป็นไปตามรูปที่ 4.3 ซึ่งความสว่างจะแปรผันตรงกับแรงดันที่ป้อนให้กับบัลลาสต์อิเล็กทรอนิกส์ชนิดปรับค่าได้



รูปที่ 4.3 การเชื่อมต่อบัลลาสต์อิเล็กทรอนิกส์กับหลอดฟลูออเรสเซนต์

4.2.4 วงจรแปลงแรงดัน

ค่าแรงดันควบคุมที่ถูกส่งมาจากอาร์ดูโนอยู่ในช่วง 0 ถึง 5 โวลต์ ดังนั้นเพื่อให้แรงดันที่ได้ส่งไปควบคุมแสงสว่างมีความเหมาะสม จึงต้องใช้วงจรขยายแรงดันเข้ามาช่วยเพิ่มแรงดันจาก 0 ถึง 5 โวลต์ ให้เป็น 0 ถึง 10 โวลต์ เพื่อส่งไปควบคุมบัลลาสต์อิเล็กทรอนิกส์ โดยมีวงจรขยายแรงดัน ดังรูปที่ 4.4

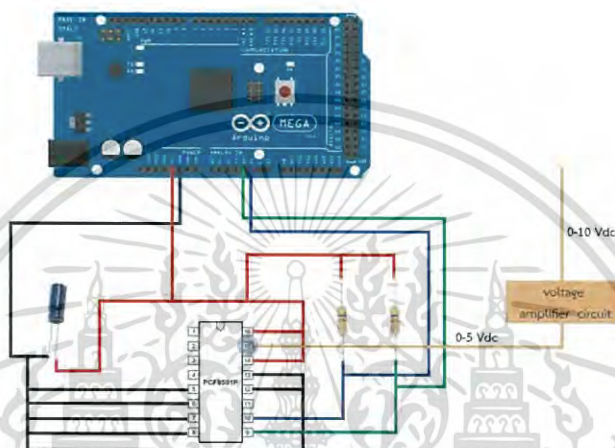


รูปที่ 4.4 การเชื่อมต่อวงจรแปลงแรงดัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.5 วงจรแปลงสัญญาณทางดิจิทัลเป็นแอนะล็อก (Digital to Analog Converter)

การแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก หมายถึงการแปลงค่าถ่วงน้ำหนักของรหัสตัวเลขฐานสองผ่านวงจรแปลงให้เป็นแรงดันแอนะล็อก โดยใช้ PCF8591P เป็นตัวแปลงสัญญาณดิจิทัลจากบอร์ดอาduinoเป็นสัญญาณแอนะล็อกทางด้านเอาต์พุต ก่อนส่งสัญญาณเข้าสู่วงจรขยายแรงดัน โดยวงจรแปลงสัญญาณทางดิจิทัลเป็นแอนะล็อก ดังรูปที่ 4.5



รูปที่ 4.5 การเชื่อมต่อวงจรแปลงสัญญาณทางดิจิทัลเป็นแอนะล็อก

4.2.6 ฟังก์ชันการสอบเทียบแสงส่องสว่าง

จากการใช้งานจริง ค่าที่กำหนดในช่วง 0 ถึง 7,000 ลักซ์ สามารถควบคุมให้อยู่ในค่าที่กำหนดได้ แต่ค่าที่กำหนดในช่วง 7,000 ถึง 9,000 ลักซ์ ไม่สามารถควบคุมให้อยู่ในค่าที่กำหนดได้ จึงได้ทำการเขียนโปรแกรมสอบเทียบแสงส่องสว่างเพิ่มขึ้นมา เพื่อทำการสอบเทียบเซนเซอร์แสงสว่างทั้ง 2 ตัว เพื่อให้ทำการควบคุมแสงสว่างที่มีค่ามากกว่า 7,000 ลักซ์ ขึ้นไปได้ ดังผลการทดลองที่จะกล่าวถึงต่อไปในบทที่ 5

4.3 ระบบควบคุมอุณหภูมิ

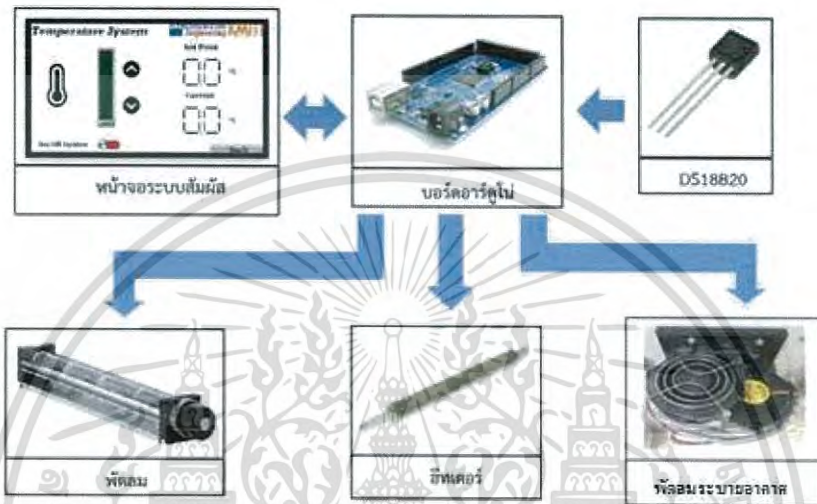
ระบบควบคุมอุณหภูมิเป็นระบบที่ใช้ควบคุมอุณหภูมิภายในเครื่อง ซึ่งประกอบไปด้วย

- 1) บอร์ดอาร์ดูโน ทำหน้าที่เป็นส่วนควบคุม
- 2) เซนเซอร์อุณหภูมิ (DS18B20) ทำหน้าที่ตรวจวัดอุณหภูมิภายในเครื่อง
- 3) พัดลมระบายอากาศ ทำหน้าที่ระบายความร้อนจากภายในเครื่องเขย้าออกสู่ภายนอก
- 4) ฮีทเตอร์ ทำหน้าที่สร้างความร้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1 โครงสร้างของระบบควบคุมอุณหภูมิ

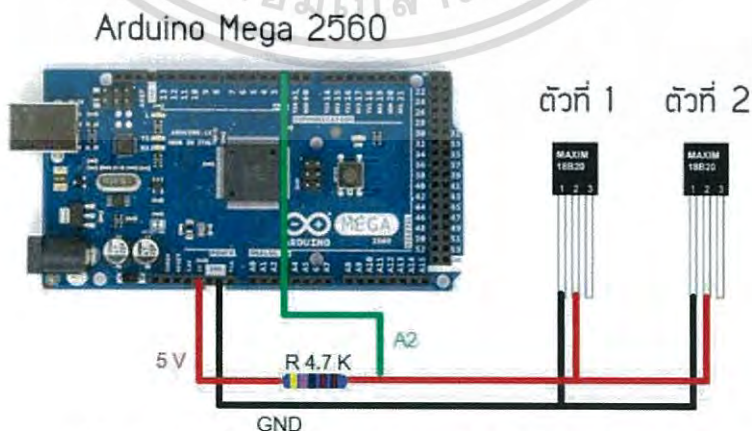
จากรูปที่ 4.6 แสดงโครงสร้างของระบบควบคุมอุณหภูมิ ซึ่งมีบอร์ดอาร์ดูโนเป็นส่วนควบคุมบอร์ดอาร์ดูโนจะรับค่าอุณหภูมิจากภายในเครื่อง และมาประมวลผลเพื่อส่งไปควบคุมการเปิดปิดของพัดลมระบายอากาศและฮีทเตอร์ โดยบอร์ดอาร์ดูโนจะสามารถรับและส่งค่าผ่านทางหน้าจอสัมผัส



รูปที่ 4.6 โครงสร้างของระบบควบคุมอุณหภูมิ

4.3.2 การรับค่าอุณหภูมิจากเซนเซอร์อุณหภูมิ (DS18B20)

ในส่วนการตรวจวัดค่าอุณหภูมินั้นจะใช้เซนเซอร์ DS18B20 จำนวน 2 ตัว ทำการตรวจวัดอุณหภูมิ โดยสัญญาณที่ได้จากเซนเซอร์นั้นเป็นสัญญาณดิจิทัล และสื่อสารกับบอร์ดอาร์ดูโนด้วยการสื่อสารแบบสัญญาณ 1 เส้น (1-wire-Communication) ซึ่งการเชื่อมต่อเป็นไปตามรูปที่ 4.7



รูปที่ 4.7 การเชื่อมต่อเซนเซอร์อุณหภูมิเข้ากับบอร์ดอาร์ดูโน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 การเพิ่มหรือลดอุณหภูมิภายในระบบด้วยฮีทเตอร์ कुलเลอร์และพัดลมระบายอากาศ

จากการทดลองใช้งานकुलเลอร์ดังรูปที่ 4.8 ในการลดอุณหภูมิภายในระบบแล้วพบว่าความสามารถของकुलเลอร์ไม่เพียงพอสำหรับการลดอุณหภูมิภายในระบบของเครื่องเขย่า ที่มีความจุประมาณ 720 ลิตร ซึ่งกำลังทำความเย็นของकुलเลอร์สามารถลดอุณหภูมิภายในตู้ที่มีความจุประมาณ 50 ลิตรเท่านั้น ดังนั้นจึงเปลี่ยนมาใช้พัดลมระบายอากาศแทนดังรูปที่ 4.9 เมื่อต้องการที่จะให้อุณหภูมิในระบบมีอุณหภูมิสูงขึ้นกว่าอุณหภูมิที่วัดได้จากระบบ อาร์ดูโนจะส่งสัญญาณให้เปิดฮีทเตอร์เพื่อทำให้อุณหภูมิในระบบมีค่าสูงขึ้นจนถึงค่าที่กำหนด จึงจะสั่งปิดฮีทเตอร์ แต่ถ้าอุณหภูมียังสูงกว่าค่าที่กำหนดไว้ก็จะทำการสั่งปิดฮีทเตอร์ และสั่งเปิดพัดลมระบายอากาศเพื่อลดอุณหภูมิภายในระบบลงมาให้ได้จนถึงค่าที่กำหนด แต่ถ้าหากว่าอุณหภูมิภายในระบบมีค่าใกล้เคียงกับค่าที่กำหนดในช่วง ± 0.5 องศาเซลเซียส ก็จะสั่งปิดทั้งฮีทเตอร์ และพัดลมระบายอากาศ



รูปที่ 4.8 ทดลองใช้งานकुलเลอร์



รูปที่ 4.9 ทดลองใช้งานพัดลมระบายอากาศแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ผลการทดลอง

5.1 บทนำ

ในบทนี้จะกล่าวถึงการทดลอง และผลการทดลองของโครงการปรับปรุงประสิทธิภาพของเครื่องเย็บแบบควบคุมอุณหภูมิและแสงส่องสว่าง โดยจะแบ่งการทดลองออกเป็น 4 ลักษณะคือ การทดลองการทำงานของระบบควบคุมอุณหภูมิ การทดลองการทำงานของระบบควบคุมแสงส่องสว่าง การทดลองการทำงานของระบบการสอบเทียบแสงส่องสว่าง และการทดลองการทำงานของระบบควบคุมแสงส่องสว่างร่วมกับอุณหภูมิ

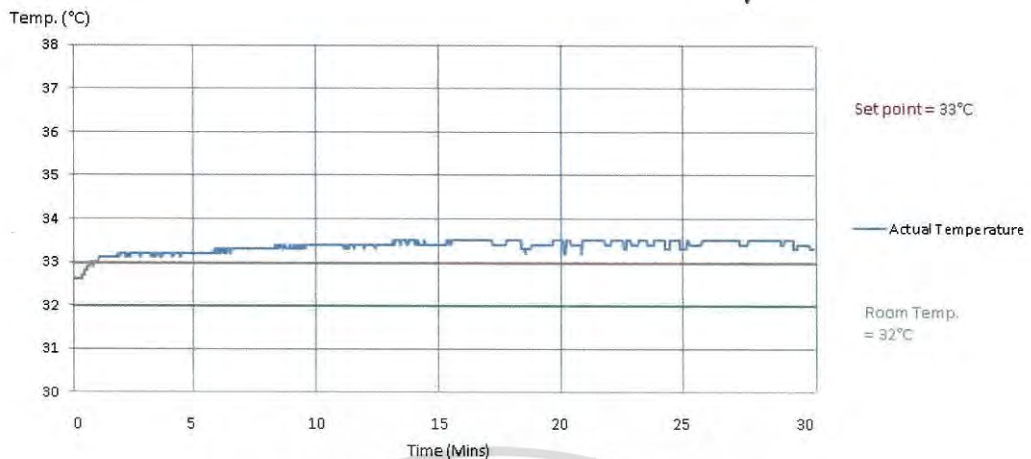
5.2 การทดลองและผลการทดลองการทำงานของระบบควบคุมอุณหภูมิ

5.2.1 การทดลองการทำงานของระบบควบคุมอุณหภูมิ

การทดลองการทำงานของระบบควบคุมอุณหภูมิ เป็นการทดลองเพื่อทดสอบประสิทธิภาพการทำงานของเครื่องเย็บ ว่ามีความสามารถในการควบคุมอุณหภูมิให้กับระบบมากน้อยเพียงใด ภายใต้สภาวะแวดล้อมที่ไม่สามารถควบคุมได้ คือการทดลองภายในห้องที่มีการเปิดโล่งอยู่ตลอดเวลา ทั้งนี้อุณหภูมิเริ่มต้นจะขึ้นอยู่กับอุณหภูมิภายนอก สำหรับค่าอุณหภูมิที่กำหนดในทดสอบคือ 33 36 และ 39 องศาเซลเซียส

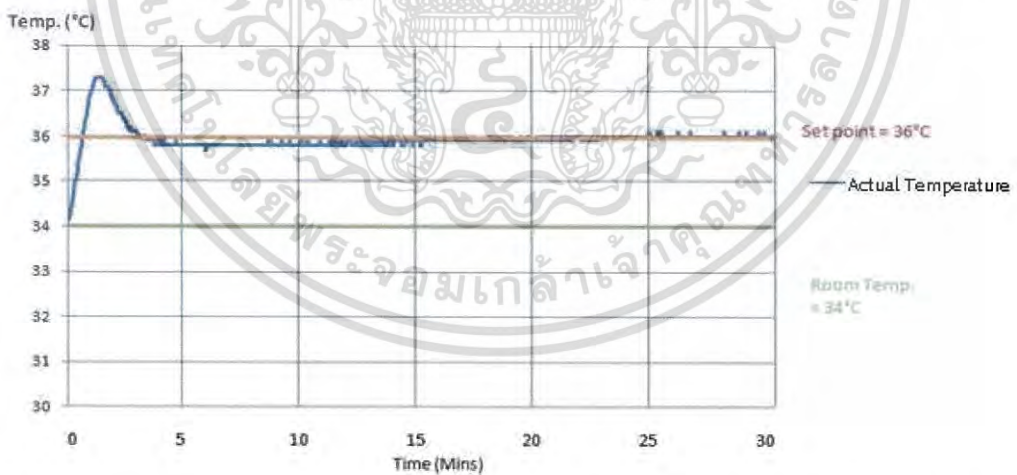
5.2.2 ผลการทดลองการทำงานของระบบควบคุมอุณหภูมิ

จากการทดลองพบว่า ตัวเครื่องสามารถควบคุมอุณหภูมิภายในตัวเครื่องให้มีอุณหภูมิเท่ากับค่าที่กำหนดไว้ได้ โดยอุณหภูมิที่มีค่าสูงจะต้องใช้เวลาในการปรับค่าอุณหภูมิให้เท่ากับค่าที่ตั้งไว้มากกว่าอุณหภูมิที่มีค่าต่ำกว่า และหลังจากอุณหภูมิภายในตัวเครื่องมีค่าเท่ากับค่าที่กำหนดไว้ ระบบควบคุมสามารถรักษาระดับอุณหภูมิให้มีค่าใกล้เคียงกับค่าที่กำหนดไว้ได้ แต่จะมีการเปลี่ยนแปลงอยู่ในช่วง 0.5 องศาเซลเซียส ซึ่งถือว่ามีความเสถียรภาพอยู่พอสมควร



รูปที่ 5.1 ผลการทดลองการทำงานของระบบควบคุมอุณหภูมิที่ 33 องศาเซลเซียส

จากรูปที่ 5.1 พบว่าอุณหภูมิภายนอกเท่ากับ 32 องศาเซลเซียส เมื่อเริ่มทำการทดลอง อุณหภูมิภายในเครื่องเขย่ามีค่าเริ่มต้น 32.6 องศาเซลเซียส ซึ่งมีค่าใกล้เคียงกับอุณหภูมิที่กำหนด 33 องศาเซลเซียส โดยมีการเปลี่ยนแปลงอยู่ในช่วง 0.5 องศาเซลเซียส ถือว่ามีเสถียรภาพอยู่พอสมควร ทำให้ฮีตเตอร์และพัดลมระบายอากาศไม่ทำงาน แต่อุณหภูมิภายในเครื่องเขย่ามีการเพิ่มขึ้นที่เล็กน้อย เนื่องจากความร้อนสะสมจากมอเตอร์ของเครื่องเขย่า และสามารถควบคุมอุณหภูมิให้อยู่ในช่วงค่าที่กำหนดได้

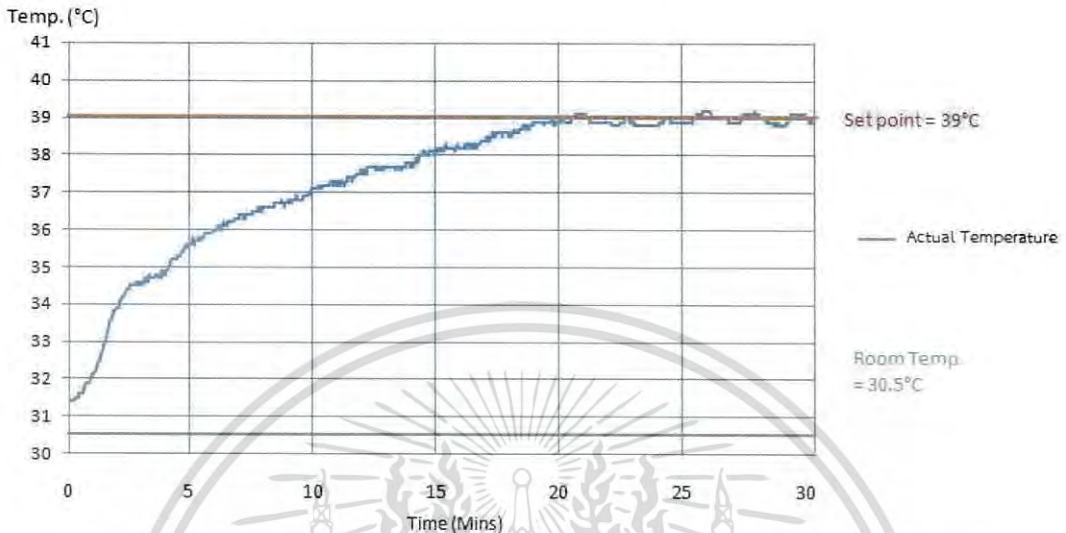


รูปที่ 5.2 ผลการทดลองการทำงานของระบบควบคุมอุณหภูมิที่ 36 องศาเซลเซียส

จากรูปที่ 5.2 พบว่าอุณหภูมิภายนอกเท่ากับ 34 องศาเซลเซียส เมื่อเริ่มทำการทดลอง อุณหภูมิภายในเครื่องเขย่ามีค่าเริ่มต้น 34.1 องศาเซลเซียส และอุณหภูมิที่กำหนดเท่ากับ 36 องศาเซลเซียส ซึ่งอุณหภูมิที่กำหนดสูงกว่าอุณหภูมิภายในเครื่องเขย่า ทำให้ฮีตเตอร์ทำงานเพื่อเพิ่มอุณหภูมิให้สูงขึ้นใกล้เคียงกับค่าที่กำหนด เมื่ออุณหภูมิภายในสูงเกินกว่า 36.5 องศาเซลเซียส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฮีทเตอร์ก็จะหยุดทำงาน แต่อุณหภูมิจะยังสูงขึ้นอีกระยะเวลาหนึ่ง เนื่องจากมีความร้อนสะสมจากฮีทเตอร์ และพัดลมระบายอากาศก็จะทำงาน เพื่อลดอุณหภูมิให้ใกล้เคียงกับค่าที่กำหนด



รูปที่ 5.3 ผลการทดลองการทำงานของระบบควบคุมอุณหภูมิที่ 39 องศาเซลเซียส

จากรูปที่ 5.3 พบว่าอุณหภูมิภายนอกเท่ากับ 30.5 องศาเซลเซียส เมื่อเริ่มทำการทดลองอุณหภูมิภายในเครื่องเขยามีค่าเริ่มต้น 31.4 องศาเซลเซียส และอุณหภูมิที่กำหนดเท่ากับ 39 องศาเซลเซียส ซึ่งอุณหภูมิที่กำหนดสูงกว่าอุณหภูมิภายในเครื่องเขยา ทำให้ฮีทเตอร์ทำงานเพื่อเพิ่มอุณหภูมิให้สูงขึ้นใกล้เคียงกับค่าที่กำหนด และสามารถควบคุมอุณหภูมิภายในให้อยู่ในช่วงที่ต้องการได้

5.3 การทดลองและผลการทดลองการทำงานของระบบควบคุมแสงส่องสว่าง

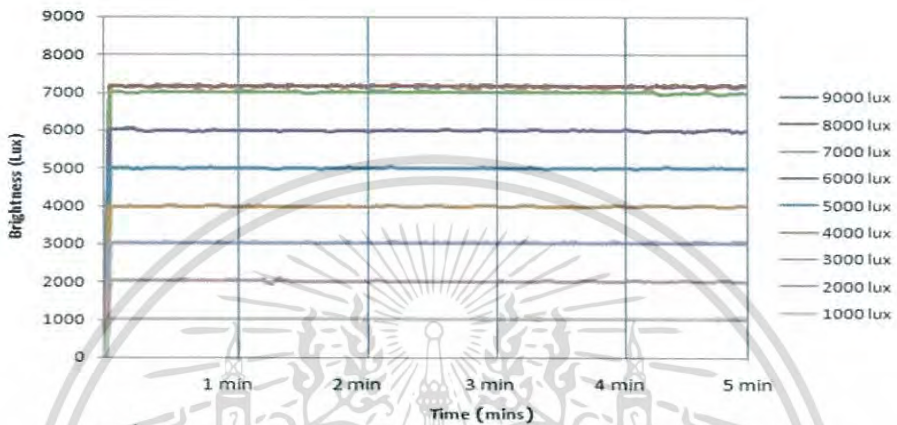
5.3.1 การทดลองการทำงานของระบบควบคุมแสงส่องสว่าง

การทดลองการทำงานของระบบควบคุมแสงส่องสว่าง เป็นการทดลองเพื่อทดสอบประสิทธิภาพในการให้แสงสว่างในตัวเครื่องมีค่าเท่ากับค่าแสงสว่างที่กำหนดไว้ และรักษาระดับความสว่างของแสงให้มีค่าใกล้เคียงกับค่าที่กำหนดไว้ การทดลองเริ่มต้นตั้งแต่เริ่มการสั่งเปิดไฟจนกระทั่งตัวเครื่องสามารถปรับความสว่างภายในให้มีค่าเท่ากับแสงสว่างที่กำหนดไว้ และบันทึกผลการทดลองต่อไปอีกระยะหนึ่งเพื่อดูเสถียรภาพของการรักษาระดับความสว่าง โดยจะทำการทดลองตั้งค่าความสว่างในหลายระดับตั้งแต่ 1,000 ถึง 9,000 ลักซ์ โดยมีความห่างในแต่ละระดับเท่ากับ 1,000 ลักซ์

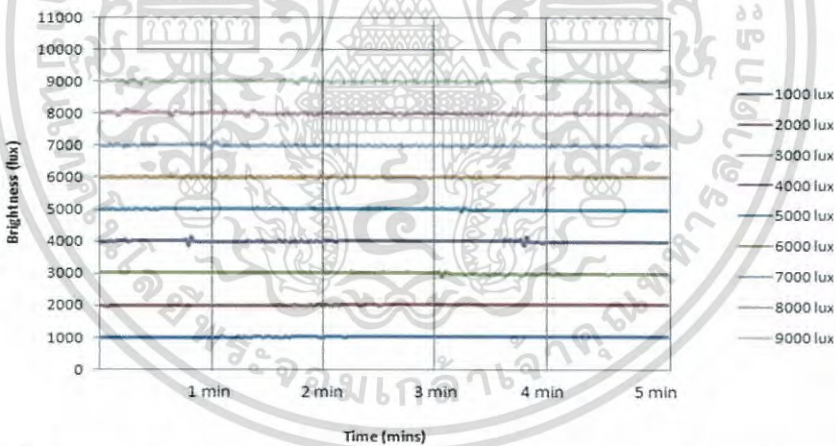
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.2 ผลการทดลองการทำงานของระบบควบคุมแสงส่องสว่าง

จากรูปที่ 5.4 และ 5.5 พบว่าก่อนทำการสอบเทียบความสว่างของแสงนั้น ในช่วง 8,000 ถึง 9,000 ลักซ์ ไม่เป็นไปตามค่าที่กำหนดไว้ แต่หลังจากทำการสอบเทียบแล้ว ทุกช่วงของค่าความสว่างของแสงมีค่าเป็นไปตามที่กำหนด



รูปที่ 5.4 ผลการทดลองการทำงานของระบบควบคุมแสงส่องสว่าง



รูปที่ 5.5 ผลการทดลองการทำงานของระบบควบคุมการสอบเทียบแสงส่องสว่าง

5.4 การทดลองและผลการทดลองการทำงานของระบบควบคุมแสงส่องสว่างร่วมกับ อุณหภูมิ

5.4.1 การทดลองการทำงานของระบบควบคุมแสงสว่างร่วมกับอุณหภูมิ

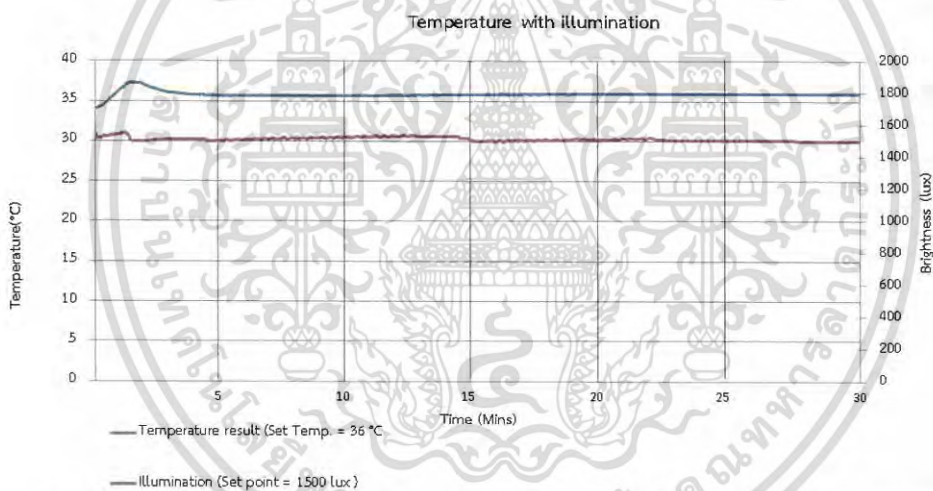
การทดลองการทำงานของระบบควบคุมแสงส่องสว่างร่วมกับอุณหภูมิ เป็นการทดลอง เพื่อทดลองประสิทธิภาพในการทำงานร่วมกันของระบบแสงส่องสว่างและอุณหภูมิ ว่าในการทำงาน ร่วมกันตัวเครื่องสามารถทำงานร่วมกันได้หรือไม่ และรักษาค่าแสงสว่างกับอุณหภูมิภายในตัวเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้มีค่าใกล้เคียงกับค่าที่กำหนด โดยจะทำการทดลองกำหนดค่าความสว่างเท่า 1,500 ลักซ์ และใช้อุณหภูมิเริ่มต้นที่ 33 องศาเซลเซียส สำหรับทดสอบ 36 องศาเซลเซียส

5.4.2 ผลการทดลองการทำงานของระบบควบคุมแสงส่องสว่างร่วมกับอุณหภูมิ

จากการทดลองการทำงานของระบบควบคุมแสงส่องสว่างร่วมกับอุณหภูมิ โดยกำหนดให้แสงสว่างมีค่าคงที่เท่ากับ 1,500 ลักซ์ สำหรับอุณหภูมิจะกำหนดไว้ 36 องศาเซลเซียส ดังแสดงผลทดลองตามรูปที่ 5.6 จากผลการทดลองแสดงให้เห็นว่าตัวเครื่องสามารถทำงานร่วมกันระหว่างแสงส่องสว่างและอุณหภูมิได้ สามารถปรับค่าแสงสว่างและอุณหภูมิภายในตัวเครื่องได้ตามค่าแสงสว่างและอุณหภูมิที่กำหนดไว้ และหลังจากอุณหภูมิภายในตัวเครื่องมีค่าเท่ากับที่ตั้งแล้วสามารถรักษาอุณหภูมิให้มีค่าใกล้เคียงกับค่าที่ตั้งได้ดี ส่วนแสงสว่างในตัวเครื่องหลังจากใช้เวลาในการปรับเพียงเล็กน้อย และสามารถรักษาระดับความสว่างได้ดี



รูปที่ 5.6 ผลการทดลองการทำงานของระบบควบคุมแสงส่องสว่างร่วมกับอุณหภูมิที่ 36 องศาเซลเซียส

บทที่ 6

สรุปผล ปัญหาและข้อเสนอแนะ

6.1 สรุปผล

จากการทดลองการปรับปรุงประสิทธิภาพของเครื่องเย็บแบบควบคุมอุณหภูมิ และแสงส่องสว่างนั้น สามารถควบคุมอุณหภูมิได้สูงสุดถึง 45 องศาเซลเซียส และกำหนดค่าความสว่างไว้ที่ค่าสูงสุด (9,000 ลักซ์) และรักษาระดับของอุณหภูมิให้อยู่ช่วงที่ใกล้เคียงกับอุณหภูมิที่ตั้งไว้ได้ สามารถตรวจวัดปริมาณความเข้มของแสงภายในตัวเครื่องได้ รวมถึงสามารถควบคุมและรักษาระดับความเข้มของแสงให้อยู่ในช่วงใกล้เคียงกับความเข้มแสงที่ตั้งไว้ได้ โดยสามารถตั้งค่าแสงสว่างได้สูงสุดถึง 9,000 ลักซ์โดยเข้าใช้ฟังก์ชันระบบสอบเทียบแสงส่องสว่างก่อน และสามารถเลือกการทำงานในรูปแบบต่างๆ ได้อย่างหลากหลาย เช่น ระบบควบคุมอุณหภูมิ ระบบควบคุมแสงส่องสว่าง และระบบการสอบเทียบแสงส่องสว่าง ซึ่งผลที่ได้เป็นไปตามวัตถุประสงค์ที่กำหนดไว้

6.2 ปัญหาและแนวทางแก้ไข

- 1) ซอฟต์แวร์ของอุปกรณ์ที่ใช้งานนั้น ยังมีการปรับปรุงอยู่เป็นระยะ ๆ ทำให้การเขียนโปรแกรมมีการเปลี่ยนแปลงอยู่ตลอด จึงจำเป็นต้องอัปเดตเฟิร์มแวร์อยู่เรื่อย ๆ
- 2) ปัจจัยภายนอกมีผลต่อการควบคุมอุณหภูมิของเครื่องเป็นอย่างมาก เช่น อุณหภูมิภายนอกมีค่าสูงเกินไป ไม่สามารถลดอุณหภูมิให้มีค่าใกล้เคียงกับค่าที่ตั้งได้ แต่หากต้องการเพิ่มเติมระบบทำความเย็น ก็ต้องใช้ระบบทำความเย็นที่มีประสิทธิภาพในการทำ ความเย็นที่สูงมาก หรืออาจแก้ไขโดยการย้ายเครื่องไปติดตั้งในห้องที่มีเครื่องปรับอากาศ
- 3) เนื่องจากการหยุดการทำงานของเครื่องนั้น เกิดจากการที่ข้อมูลที่ส่งจากเซนเซอร์มายัง อาร์ดูโน มีการส่งข้อมูลผิดพลาดในบางช่วง ซึ่งเป็นผลมาจากความถี่ที่เกิดขึ้นจากตัวบัสลาสต์ที่ป้อนให้กับหลอดไฟไปรบกวนการสื่อสาร ทำให้ในบางครั้งการอ่านค่าจึงเกิดความผิดพลาดทำให้เครื่องเกิดความผิดปกติ แต่ปัญหาดังกล่าวได้จากการแก้ไขแล้ว โดยการเขียนโปรแกรมจำกัดการรับค่าช่วงอุณหภูมิ เฉพาะในช่วงที่มีความน่าจะเป็นว่าเป็นค่าที่อ่านได้จากเซนเซอร์จริง ๆ เท่านั้น

6.3 ข้อเสนอแนะ

การปรับปรุงประสิทธิภาพของเครื่องเย็บแบบควบคุมอุณหภูมิและแสงส่องสว่างนั้น เมื่อนำไปใช้งานในห้องปฏิบัติการที่เปิดโล่งไม่มีเครื่องปรับอากาศ ดังนั้นอุณหภูมิแวดล้อมจะแปรผันไปตามสภาพแวดล้อมภายนอก หากว่าอุณหภูมิสูงเกินจากค่าที่ต้องการควบคุมก็จะประสบกับปัญหาเนื่องจากไม่สามารถลดอุณหภูมิให้อยู่ในช่วงที่ต้องการใช้งานได้ การติดตั้งอุปกรณ์ทำความเย็นเพิ่มเติมเพื่อควบคุมอุณหภูมิภายในตัวเครื่องเย็บให้สามารถทำงานได้ในทุกช่วงที่ต้องการ สามารถทำได้แต่มีค่าใช้จ่ายค่อนข้างสูงและยุ่งยาก หากเปลี่ยนตำแหน่งการใช้งานตัวเครื่องควบคุมอุณหภูมิและแสงส่องสว่างมาอยู่ภายในห้องที่มีเครื่องปรับอากาศ ก็จะช่วยลดความยุ่งยากในการปรับปรุงตัวเครื่องลงได้และไม่มีค่าใช้จ่ายเพิ่มเติม อีกทั้งยังสามารถดำเนินการได้ทันที



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Arduino. "Arduino MEGA 2560." [Online].

Available: <https://www.arduino.cc/en/Main/arduinoBoardMega2560>.2016.

[2] Touchscreens."4d system." [Online].

Available: <http://www.4dsystems.com.au/>

[3] Dallas Semiconductor." Sensor DS18B20." [Online].

Available: <https://www.maximintegrated.com/en/products/analog/sensors-and-sensor-interface/DS18B20.html>

[4] nxp."PCF8591." [Online].

Available: https://www.nxp.com/documents/data_sheet/PCF8591.pdf



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

โปรแกรม Arduino

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

โปรแกรมสำหรับ Arduino

```

#define relay_1 41
#define relay_2 43
#define relay_3 45
#define relay_4 47
#define relay_5 49
#define relay_6 51
#define relay_7 53
int starttemp = 0;
int start = 0;
//-----
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 2
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
int Newmax;
float temp1;
float temp2;
float temp_avg;
float last_temp;
float set_temp ;
float compare_temp;
#define gap_temp 0.5
float R = 70;
float K = 0.7;
//-----
#define PCF8591 (0x90 >> 1)
int i;
int Ain = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int RawValue0 = 0;
int DACout = 0;
float DACoutVolt = 0.0;
float Voltage = 0.0;
float Offset = 0.0;
#include <BH1750FVI.h>
#include <Wire.h>
BH1750FVI LightSensor;
BH1750FVI LightMeter;
float lux1;
float lux2;
float luxA;
float luxB;
float light_avg ;
unsigned int compare_light;
int set_light;
int gap_light;
float ratio1 ;
float ratio2 ;
float ratio11 = 1.0;
float ratio22 = 1.0;
float rationew1;
float rationew2;
//-----
int max_lux;
int min_lux;
int light;
boolean lm;
float lux3 = 0;
float lux4 = 0;
int maxlux = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int A;
int H = 100;
float lux5 = 0;
float lux6 = 0;
int minlux = 0;
int C = 0;
int D = 0;
boolean no;
float Lux_cal;
float lux_10;
float lux_20;
float R1;
float R2;
//-----
#include <genieArduino.h>
int Add1;
int Ca;
int slider_val0 = 0;
int keep0 = 0;
int BUTTON_0;
int Show_BUTTON_0;
int BUTTON_1;
int BUTTON_2;
int Show_BUTTON_1;
int Show_BUTTON_2;
int Show_BUTTON_3;
int Jump;
int WinButton0;
int WinButton1;
int WinButton2;
int WinButton3;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int Asd;
int Cal = 0;
int Ba;
int Show_WinButton0;
int Show_WinButton1;
int Show_WinButton2;
int Show_WinButton3;
int slider_val1 = 0;
int keep1 = 0;
int AddSetpointLight;
int AddSetpointTemp;
int slider_val2;
int AddSetpointCal;
int Da;
int WinButton4;
int Show_WinButton4;
int BUTTON_4;
int Show_BUTTON_4;
int BUTTON_5;
int Show_BUTTON_5;
int BUTTON_6;
int Show_BUTTON_6;
int test;
float test1;
int BUTTON_7;
int Show_BUTTON_7;
Genie genie;
#define RESETLINE 4
//-----
void setup()
{ Serial.begin(19200);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

genie.Begin(Serial);
pinMode(13, OUTPUT);
digitalWrite(13, LOW);
pinMode(RESETLINE, OUTPUT);
digitalWrite(RESETLINE, 1);
delay(100);
digitalWrite(RESETLINE, 0);
delay(5800);
Wire.begin();
sensors.begin();
LightSensor.begin();
LightMeter.begin();
LightSensor.SetAddress(Device_Address_H);
LightMeter.SetAddress(Device_Address_L);
LightSensor.SetMode(Continuous_H_resolution_Mode);
LightMeter.SetMode(Continuous_L_resolution_Mode);
pinMode(relay_1, OUTPUT);
pinMode(relay_2, OUTPUT);
pinMode(relay_3, OUTPUT);
pinMode(relay_4, OUTPUT);
pinMode(relay_5, OUTPUT);
pinMode(relay_6, OUTPUT);
pinMode(relay_7, OUTPUT);
digitalWrite(relay_1, 1);
digitalWrite(relay_2, 0);
digitalWrite(relay_3, 0);
digitalWrite(relay_4, 0);
digitalWrite(relay_5, 0);
digitalWrite(relay_6, 1);
delay(1000);
digitalWrite(relay_6, 0);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

digitalWrite(relay_7, 1);
delay(50);
genie.AttachEventHandler(loop);
genie.WriteStr(0, GENIE_VERSION);
}
//-----
void loop()
{
  if ( start == 1 )
  {
    temp_avg = 0;
    sensors.requestTemperatures();
    temp1 = sensors.getTempCByIndex(0);
    temp2 = sensors.getTempCByIndex(1);
    if ((temp1 > 10 && temp1 < R) && (temp2 < 10 || temp2 > R)) {
      temp1 = temp1 + K;
      temp2 = temp1;
    }
    else if ((temp1 < 10 || temp1 > R) && ( temp2 > 10 && temp2 < R)) {
      temp2 = temp2 + K;
      temp1 = temp2;
    }
    else if ((temp1 < 10 || temp1 > R) && (temp2 < 10 || temp2 > R)) {
      temp1 = last_temp;
      temp2 = last_temp;
    }
    else {
      temp1 = temp1 + K;
      temp2 = temp2 + K;
    }
    temp_avg = ((temp1 * 1.0) + (temp2 * 1.0)) / 2.0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

last_temp = temp_avg;
lux1 = LightSensor.GetLightIntensity();
luxA = (lux1 * ratio11) + C;
lux2 = LightMeter.GetLightIntensity();
luxB = (lux2 * ratio22) + D;
light_avg = (luxA + luxB) / 2;

```

```

if (set_temp > temp_avg) {
    compare_temp = set_temp - temp_avg;
}
else if (set_temp <= temp_avg) {
    compare_temp = temp_avg - set_temp;
}
if (set_light > light_avg) {
    compare_light = set_light - light_avg;
}
else {
    compare_light = light_avg - set_light;
}
if (starttemp == 1 )
{
    if (compare_temp > gap_temp) {
        if (set_temp < temp_avg) {
            digitalWrite(relay_1, 1);
            digitalWrite(relay_7, 0);
        }
        else if (set_temp > temp_avg) {
            digitalWrite(relay_1, 0);
            digitalWrite(relay_7, 1);
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if (compare_temp <= gap_temp) {
    digitalWrite(relay_1, 1);
    digitalWrite(relay_7, 1);
}
}
else if ( starttemp == 0)
{
    digitalWrite(relay_1, 1);
    digitalWrite(relay_7, 1);
}
}
else if (start == 0 )
{
    digitalWrite(relay_1, 1);
    digitalWrite(relay_2, 1);
    digitalWrite(relay_3, 1);
    digitalWrite(relay_4, 1);
    digitalWrite(relay_5, 1);
    digitalWrite(relay_6, 1);
    digitalWrite(relay_7, 1);
}
else if (start == 3 )
{
    digitalWrite(relay_4, 0);
    digitalWrite(relay_5, 0);
    digitalWrite(relay_6, 0);
    lux1 = LightSensor.GetLightIntensity();
    lux2 = LightMeter.GetLightIntensity();
    int i = 255;
    Wire.beginTransaction(PCF8591);
    Wire.write(0x40);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Wire.write(i);
Wire.endTransmission();
if (maxlux == 1) {
  for (int m = 1; m < 26; m++) {
    if (lux1 == 0) {
      lux3 = lux1;
    }
    else if (lux3 - lux1 <= H) {
      lux3 = (lux3 + lux1) / 2;
    }
    else if (lux3 - lux1 > 100) {
      m--;
    }
    if (lux2 == 0) {
      lux4 = lux2;
    }
    else if (lux4 - lux2 <= H) {
      lux4 = (lux4 + lux2) / 2;
    }
    else if (lux4 - lux2 > 100) {
      m--;
    }
  }
}
else if (maxlux == 0) {
}
}
else if (start == 4 )
{
  digitalWrite(relay_4, 0);
  digitalWrite(relay_5, 0);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

digitalWrite(relay_6, 0);
lux1 = LightSensor.GetLightIntensity();
lux2 = LightMeter.GetLightIntensity();
int i = 128;
Wire.beginTransaction(PCF8591);
Wire.write(0x40);
Wire.write(i);
Wire.endTransmission();
if (minlux == 1) {
  for (int m = 1; m < 26; m++) {
    if (lux1 == 0) {
      lux5 = lux1;
    }
    else if (lux5 - lux1 <= H) {
      lux5 = (lux5 + lux1) / 2;
    }
    else if (lux5 - lux1 > 100) {
      m--;
    }
    if (lux2 == 0) {
      lux6 = lux2;
    }
    else if (lux6 - lux2 <= H) {
      lux6 = (lux6 + lux2) / 2;
    }
    else if (lux4 - lux2 > 100) {
      m--;
    }
  }
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if (minlux == 0) {
}
}
if (Cal == 1 )
{
rationew1 = ((max_lux - min_lux) / (lux3 - lux5));
R1 = rationew1 * lux5;
if ((R1) > min_lux) {
C = (R1) - min_lux;
no = 0;
}
else if ((R1) <= min_lux) {
C = min_lux - (R1);
no = 1;
}
rationew2 = ((max_lux - min_lux) / (lux4 - lux6));
R2 = rationew2 * lux6;
if ((R2) > min_lux) {
D = (R2) - min_lux;
no = 0;
}
else if ((R2) <= min_lux) {
D = min_lux - (R2);
no = 1;
}
ratio11 = rationew1;
ratio22 = rationew2;
delay(1000);
if (no == 0) {
delay(1000);
lux_10 = R1 - C;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lux_20 = R2 - D;
Lux_cal = (lux_10 + lux_20) / 2;
}
else {
  delay(1000);
  lux_10 = R1 + C;
  lux_20 = R2 + D;
  Lux_cal = (lux_10 + lux_20) / 2;
}
}
else if ( Cal == 2)
{
}
if (light == 1) {
  if (compare_light >= gap_light) {
    if ((AddSetpointLight * 100 + Ca * 100) < 1000 && (AddSetpointLight * 100
+ Ca * 100) != 0 ) {
      if (lm == 0) {
        i = 40;
        Wire.beginTransmission(PCF8591);
        Wire.write(0x40);
        Wire.write(i);
        Wire.endTransmission();
        lm = 1;
      }
      if (light_avg < (AddSetpointLight * 100 + Ca * 100)) {
        i += 5;
        Wire.beginTransmission(PCF8591);
        Wire.write(0x40);
        Wire.write(i);
        Wire.endTransmission();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    delay(200);
}
else {
    i -= 3;
    Wire.beginTransmission(PCF8591);
    Wire.write(0x40);
    Wire.write(i);
    Wire.endTransmission();
    delay(200);
}
}
else if ((AddSetpointLight * 100 + Ca * 100) >= 1000 && (AddSetpointLight
* 100 + Ca * 100) < 2000) {
    if (lm == 0) {
        i = 53;
        Wire.beginTransmission(PCF8591);
        Wire.write(0x40);
        Wire.write(i);
        Wire.endTransmission();
        lm = 1;
    }
    if (light_avg < (AddSetpointLight * 100 + Ca * 100)) {
        i += 5;
        Wire.beginTransmission(PCF8591);
        Wire.write(0x40);
        Wire.write(i);
        Wire.endTransmission();
        delay(200);
    }
}
else {
    i -= 3;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Wire.beginTransmission(PCF8591);
Wire.write(0x40);
Wire.write(i);
Wire.endTransmission();
delay(200);
}
}
else if ((AddSetpointLight * 100 + Ca * 100) >= 2000 && (AddSetpointLight
* 100 + Ca * 100) < 3000) {
  if (lm == 0) {
    i = 65;
    Wire.beginTransmission(PCF8591);
    Wire.write(0x40);
    Wire.write(i);
    Wire.endTransmission();
    lm = 1;
  }
  if (light_avg < (AddSetpointLight * 100 + Ca * 100)) {
    i += 5;
    Wire.beginTransmission(PCF8591);
    Wire.write(0x40);
    Wire.write(i);
    Wire.endTransmission();
    delay(200);
  }
  else {
    i -= 3;
    Wire.beginTransmission(PCF8591);
    Wire.write(0x40);
    Wire.write(i);
    Wire.endTransmission();
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    delay(200);
  }
}
else if ((AddSetpointLight * 100 + Ca * 100) >= 3000 && (AddSetpointLight
* 100 + Ca * 100) < 4000) {
  if (lm == 0) {
    i = 70;
    Wire.beginTransmission(PCF8591);
    Wire.write(0x40);
    Wire.write(i);
    Wire.endTransmission();
    lm = 1;
  }
  if (light_avg < (AddSetpointLight * 100 + Ca * 100)) {
    i += 5;
    Wire.beginTransmission(PCF8591);
    Wire.write(0x40);
    Wire.write(i);
    Wire.endTransmission();
    delay(200);
  }
  else {
    i -= 3;
    Wire.beginTransmission(PCF8591);
    Wire.write(0x40);
    Wire.write(i);
    Wire.endTransmission();
    delay(200);
  }
}
else if ((AddSetpointLight * 100 + Ca * 100) >= 4000 && (AddSetpointLight

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

* 100 + Ca * 100) < 5000) {
  if (lm == 0) {
    i = 90;
    Wire.beginTransmission(PCF8591);
    Wire.write(0x40);
    Wire.write(i);
    Wire.endTransmission();
    lm = 1;
  }
  if (light_avg < (AddSetpointLight * 100 + Ca * 100)) {
    i += 5;
    Wire.beginTransmission(PCF8591);
    Wire.write(0x40);
    Wire.write(i);
    Wire.endTransmission();
    delay(200);
  }
  else {
    i -= 3;
    Wire.beginTransmission(PCF8591);
    Wire.write(0x40);
    Wire.write(i);
    Wire.endTransmission();
    delay(200);
  }
}
else if ((AddSetpointLight * 100 + Ca * 100) >= 5000 && (AddSetpointLight
* 100 + Ca * 100) < 6000) {
  if (lm == 0) {
    i = 113;
    Wire.beginTransmission(PCF8591);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Wire.write(0x40);
Wire.write(i);
Wire.endTransmission();

lm = 1;
}
if (light_avg < (AddSetpointLight * 100 + Ca * 100)) {
  i += 5;
  Wire.beginTransmission(PCF8591);
  Wire.write(0x40);
  Wire.write(i);
  Wire.endTransmission();
  delay(200);
}
else {
  i -= 3;
  Wire.beginTransmission(PCF8591);
  Wire.write(0x40);
  Wire.write(i);
  Wire.endTransmission();
  delay(200);
}
}
else if ((AddSetpointLight * 100 + Ca * 100) >= 6000 && (AddSetpointLight
* 100 + Ca * 100) < 7000) {
  if (lm == 0) {
    i = 135;
    Wire.beginTransmission(PCF8591);
    Wire.write(0x40);
    Wire.write(i);
    Wire.endTransmission();
    lm = 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
if (i >= 255) {
  i = 250;
}
if (light_avg < (AddSetpointLight * 100 + Ca * 100)) {
  i += 5;
  Wire.beginTransmission(PCF8591);
  Wire.write(0x40);
  Wire.write(i);
  Wire.endTransmission();
  delay(200);
}
else {
  i -= 3;
  Wire.beginTransmission(PCF8591);
  Wire.write(0x40);
  Wire.write(i);
  Wire.endTransmission();
  delay(200);
}
}
else if ((AddSetpointLight * 100 + Ca * 100) >= 7000 && (AddSetpointLight
* 100 + Ca * 100) < 8000) {
  if (lm == 0) {
    i = 174;
    Wire.beginTransmission(PCF8591);
    Wire.write(0x40);
    Wire.write(i);
    Wire.endTransmission();
    lm = 1;
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (i >= 255) {
    i = 250;
}
if (light_avg < (AddSetpointLight * 100 + Ca * 100)) {
    i += 5;
    Wire.beginTransmission(PCF8591);
    Wire.write(0x40);
    Wire.write(i);
    Wire.endTransmission();
    delay(200);
}
else {
    i -= 3;
    Wire.beginTransmission(PCF8591);
    Wire.write(0x40);
    Wire.write(i);
    Wire.endTransmission();
    delay(200);
}
}
else if ((AddSetpointLight * 100 + Ca * 100) >= 8000 && (AddSetpointLight
* 100 + Ca * 100) < 9000) {
    if (lm == 0) {
        i = 200;
        Wire.beginTransmission(PCF8591);
        Wire.write(0x40);
        Wire.write(i);
        Wire.endTransmission();
        lm = 1;
    }
    if (i >= 255) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    i = 250;
}
if (light_avg < (AddSetpointLight * 100 + Ca * 100)) {
    i += 5;
    Wire.beginTransmission(PCF8591);
    Wire.write(0x40);
    Wire.write(i);
    Wire.endTransmission();
    delay(200);
}
else {
    i -= 3;
    Wire.beginTransmission(PCF8591);
    Wire.write(0x40);
    Wire.write(i);
    Wire.endTransmission();
    delay(200);
}
}
else if ((AddSetpointLight * 100 + Ca * 100) >= 9000 && (AddSetpointLight
* 100 + Ca * 100) < 15000) {
    if (lm == 0) {
        i = 220;
        Wire.beginTransmission(PCF8591);
        Wire.write(0x40);
        Wire.write(i);
        Wire.endTransmission();
        lm = 1;
    }
    if (i >= 255) {
        i = 250;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
if (light_avg < (AddSetpointLight * 100 + Ca * 100)) {
  i += 5;
  Wire.beginTransaction(PCF8591);
  Wire.write(0x40);
  Wire.write(i);
  Wire.endTransmission();
  delay(200);
}
else {
  i -= 3;
  Wire.beginTransaction(PCF8591);
  Wire.write(0x40);
  Wire.write(i);
  Wire.endTransmission();
  delay(200);
}
}
else {
  delay(2000);
}
}
else if (light == 0) {
}
Wire.beginTransaction(PCF8591);
Wire.write(0x40);
Wire.write(i);
Wire.endTransmission();
DACout = i;
RawValue0 = analogRead(Ain);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Voltage = (RawValue0 * 5.0 ) / 1024.0;
DACoutVolt = ((DACout * 5.0) / 256.0) - Offset;
if ( DACoutVolt < 0.0) {
    DACoutVolt = 0.0;
}
//-----
test++;
test1++;
genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x01, light_avg);
genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x05, light_avg);
genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x03, temp_avg * 10.0);
genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x07, temp_avg * 10.0);
genie.ReadObject(GENIE_OBJ_USER_LED, 0);
genieFrame Event;
genie.DequeueEvent(&Event);
digitalWrite(13, HIGH);
if (Event.reportObject.cmd == GENIE_REPORT_EVENT)
{
    if (Event.reportObject.object == GENIE_OBJ_SLIDER)
    {
        if (Event.reportObject.index == 0)
        {
            slider_val0 = genie.GetEventData(&Event);
            AddSetpointLight = slider_val0;
            genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x04, (AddSetpointLight * 100
            + Ca * 100));
            genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x00, (AddSetpointLight * 100
            + Ca * 100));
        }
        if (Event.reportObject.index == 1)
        {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

slider_val1 = genie.GetEventData(&Event);
AddSetpointTemp = slider_val1;
genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x06, AddSetpointTemp *
10.0 + Ba);
genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x02, AddSetpointTemp *
10.0 + Ba);

}
if (Event.reportObject.index == 2)
{
slider_val2 = genie.GetEventData(&Event);
AddSetpointCal = slider_val2;
genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x0C, AddSetpointCal * 100 +
Da * 100);
}
}
if (Event.reportObject.object == GENIE_OBJ_4DBUTTON)
{
if (Event.reportObject.index == 0)
{
BUTTON_0 = genie.GetEventData(&Event);
Show_BUTTON_0 = BUTTON_0;
if (Show_BUTTON_0 == 1)
{
start = 1;
digitalWrite(relay_1, 1);
digitalWrite(relay_2, 0);
digitalWrite(relay_3, 0);
digitalWrite(relay_4, 0);
digitalWrite(relay_5, 0);
digitalWrite(relay_6, 0);
}
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    digitalWrite(relay_7, 1);
}
else if ( Show_BUTTON_0 == 0 )
{
    digitalWrite(13, LOW);
    digitalWrite(relay_1, 1);
    digitalWrite(relay_2, 1);
    digitalWrite(relay_3, 1);
    digitalWrite(relay_4, 1);
    digitalWrite(relay_5, 1);
    digitalWrite(relay_6, 1);
    digitalWrite(relay_7, 1);
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x06, 0);
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x02, 0);
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x03, 0);
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x07, 0);
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x00, 0);
}
}
if (Event.reportObject.index == 5)
{
    BUTTON_7 = genie.GetEventData(&Event);
    Show_BUTTON_7 = BUTTON_7;
    if (Show_BUTTON_7 == 1)
    {
        i = 128;
        minlux = 1;
        start = 4;
    }
    else
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (Show_BUTTON_7 == 0)
{
    min_lux = AddSetpointCal * 100 + Da * 100;
    i = 0;
    minlux = 0;
    start = 0;
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x00, 0);
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x0C, 0);
}
}
}
if (Event.reportObject.index == 6)
{
    BUTTON_2 = genie.GetEventData(&Event);
    Show_BUTTON_2 = BUTTON_2;
    if (Show_BUTTON_2 == 1)
    {
        i = 255;
        maxlux = 1;
        start = 3;
    }
    else
    {
        if (Show_BUTTON_2 == 0)
        {
            max_lux = AddSetpointCal * 100 + Da * 100;
            i = 0;
            maxlux = 0;
            start = 0;
            genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x00, 0);
            genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x0C, 0);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
  }
}
if (Event.reportObject.index == 1)
{
  BUTTON_4 = genie.GetEventData(&Event);
  Show_BUTTON_4 = BUTTON_4;
  if (Show_BUTTON_4 == 1)
  {
    light = 1;
  }
  if (Show_BUTTON_4 == 0)
  {
    int i = 0;
    Wire.beginTransmission(PCF8591);
    Wire.write(0x40);
    Wire.write(i);
    Wire.endTransmission();
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x04, 0);
  }
}
if (Event.reportObject.index == 2)
{
  BUTTON_5 = genie.GetEventData(&Event);
  Show_BUTTON_5 = BUTTON_5;
  if (Show_BUTTON_5 == 1)
  {
    starttemp = 1;
    set_temp = (AddSetpointTemp * 10.0 + Ba * 5) / 10.0;
  }
  if (Show_BUTTON_5 == 0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    starttemp = 0;
}
}
if (Event.reportObject.index == 4)
{
    BUTTON_6 = genie.GetEventData(&Event);
    Show_BUTTON_6 = BUTTON_6;
    if (Show_BUTTON_6 == 1)
    {
        Cal = 1;
    }
    if (Show_BUTTON_6 == 0)
    {
        genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x08, 0);
        Cal = 2;
    }
}
}
if (Event.reportObject.object == GENIE_OBJ_WINBUTTON)
{
    if (Event.reportObject.index == 4)
    { WinButton0 = genie.GetEventData(&Event);
      Show_WinButton0 = WinButton0;
      if (Show_WinButton0 == 0 )
      {
          Ca++;
          genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x04, (AddSetpointLight *
            100 + Ca * 100));
          genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x00, (AddSetpointLight *
            100 + Ca * 100));
      }
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
}
if (Event.reportObject.index == 5)
{ WinButton1 = genie.GetEventData(&Event);
  Show_WinButton1 = WinButton1;
  if (Show_WinButton1 == 0 )
  {
    Ca--;
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x04, (AddSetpointLight *
      100 + Ca * 100));
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x00, (AddSetpointLight *
      100 + Ca * 100));
  }
}
if (Event.reportObject.index == 7)
{ WinButton2 = genie.GetEventData(&Event);
  Show_WinButton2 = WinButton2;
  if (Show_WinButton2 == 0 )
  {
    Ba++;
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x06, AddSetpointTemp *
      10.0 + Ba * 5);
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x02, AddSetpointTemp *
      10.0 + Ba * 5);
  }
}
}
if (Event.reportObject.index == 8)
{ WinButton3 = genie.GetEventData(&Event);
  Show_WinButton3 = WinButton3;
  if (Show_WinButton3 == 0 )
  {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

