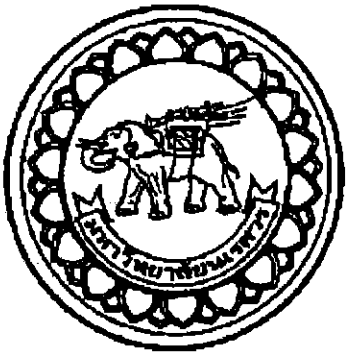


ตู้ควบคุมความชื้นโดยใช้เพลเทียร์

HUMIDITY CONTROL CHAMBER USING PELTIER

นายกิตติพงษ์	ยศอาลัย	รหัส 47380218
นายอุกฤษฎ์	ชัยยา	รหัส 47380277
นายธนุวัฒน์	แก้วกัน	รหัส 47380334

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 5 เม.ย. 2553
เลขทะเบียน 1499936
เลขเรียกหนังสือ ป/5.
ท674 ท 2550

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2550

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	ตู้ควบคุมความชื้นโดยใช้เพลเทียร์		
ผู้ดำเนินโครงงาน	นายกิตติพงษ์	ยศอาลัย	รหัส 47380218
	นายอุกฤษฎ์	ชัยยา	รหัส 47380277
	นายธนุวัฒน์	แก้วกัน	รหัส 47380334
อาจารย์ที่ปรึกษา	ดร.อัครพันธ์	วงศ์กังแห	
สาขาวิชา	วิศวกรรมไฟฟ้า		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2550		

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ
(ดร.อัครพันธ์ วงศ์กังแห)

.....กรรมการ
(ดร.ชัยรัตน์ พินทอง)

.....กรรมการ
(อาจารย์ปิยดนัย ภาชนะพรรณ์)

หัวข้อโครงงาน	ตู้ควบคุมความชื้นโดยใช้เพลเทียร์	
ผู้ดำเนินโครงงาน	นายกิตติพงษ์ ยศอาลัย	รหัส 47380218
	นายอุกฤษฎ์ ชัยยา	รหัส 47380277
	นายธนุวัฒน์ แก้วกัน	รหัส 47380334
อาจารย์ที่ปรึกษา	ดร.อัครพันธ์ วงศ์กังแห	
สาขาวิชา	วิศวกรรมไฟฟ้า	
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์	
ปีการศึกษา	2550	

บทคัดย่อ

ตู้ควบคุมความชื้นโดยใช้เพลเทียร์นี้ สามารถใช้เป็นต้นแบบในการสร้างตู้ควบคุมความชื้นขนาดเล็กหรือขนาดใหญ่ได้ ซึ่งสามารถลดต้นทุนในการซื้อเครื่องควบคุมความชื้นสัมพัทธ์ที่มีราคาแพงจากต่างประเทศ

ตู้ควบคุมความชื้นประกอบด้วยส่วนควบคุมและส่วนที่ถูกควบคุม ส่วนควบคุมจะใช้ไมโครคอนโทรลเลอร์ (MCS-51) ประมวลค่าอุณหภูมิและความชื้นจากไอซีวัดอุณหภูมิและความชื้นเบอร์ SHT15 และจะแสดงผลที่แอลซีดี ในการควบคุมการทำงานของตัวควบบคุมความชื้น จะใช้แหล่งจ่ายไฟฟ้ากระแสตรงจ่ายให้กับ เพลเทียร์ (PELTIER) โดยจะใช้แผงโลหะที่เป็นอลูมิเนียมด้านเย็นเป็นตัวจับความชื้นภายในตู้ควบคุมความชื้น สามารถควบคุมและตั้งค่าเป้าหมายของความชื้นตั้งแต่ 10 ถึง 90 % RH

Project Title	Humidity Control Chamber Using Peltier.		
Name	Mr. Kittipong	Yosarlai	ID. 47380218
	Mr. Ukrit	Chaiya	ID. 47380277
	Mr. Thanuwat	Kaewkun	ID. 47380334
Project Advisor	Dr. Akaraphunt	Vongkunghae	
Major	Electrical Engineering.		
Department	Electrical and Computer Engineering.		
Academic Year	2007		

ABSTRACT

This project is a model for creating a small or large size humidity control chamber. It can reduce the cost to purchase temperature and humidity controllers from abroad.

The humidity controller consists of a controller and a regulator. The control part uses a Microcontroller (MCS-51) for processing humidity values from an IC measuring temperature and humidity no.SHT15 and displays results on a LCD. In controlling the operation of the original humidity, a DC source of electricity is used to power a Peltier by using an aluminum control panel on cool plate to catch the humidity in the chamber. It can control the target value of humidity from 10 to 90 %RH.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้เป็นการศึกษาเกี่ยวกับเรื่องการออกแบบและสร้างตู้ควบคุมความชื้นโดยใช้เพลเทียร์ ซึ่งจะไม่มีทางสำเร็จไปได้ถ้าไม่ได้รับความช่วยเหลือจากบุคคลดังต่อไปนี้

ขอขอบพระคุณ ดร.อัครพันธ์ วงศ์ถังแห อาจารย์ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ มหาวิทยาลัยนเรศวร ผู้เป็นอาจารย์ที่ปรึกษาโครงงานที่ได้ให้ความรู้ ให้คำแนะนำและให้ ความช่วยเหลือแก่คณะผู้จัดทำเป็นอย่างดีตลอดมา

ขอขอบพระคุณ อาจารย์ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ มหาวิทยาลัยนเรศวรทุกท่าน ที่ได้ให้ความรู้และให้คำสั่งสอนจนคณะผู้จัดทำมีความรู้ที่สามารถนำมาประยุกต์ใช้ในการจัดทำ โครงงานในครั้งนี้

และที่สำคัญที่สุดขอขอบพระคุณบิดา มารดา ที่ได้เลี้ยงดูและอบรมสั่งสอนแก่คณะผู้จัดทำจน ทำให้คณะผู้จัดทำทุกคนมีวันนี้ได้ ซึ่งเป็นพระคุณอันหาที่เปรียบไม่ได้

ท้ายนี้คณะผู้จัดทำใคร่ขอขอบพระคุณผู้ที่มีส่วนเกี่ยวข้องทุกท่านที่ไม่ได้กล่าวนามมา ณ ที่นี้ ที่ มีส่วนร่วมในการให้ข้อมูล เป็นที่ปรึกษาในการทำปริญญานิพนธ์ฉบับนี้จนเสร็จสมบูรณ์ คณะผู้จัดทำ จึงขอขอบพระคุณไว้ ณ ที่นี้

คณะผู้จัดทำโครงงาน

นายกิตติพงษ์ ยศอาลัย รหัส 47380218

นายอุกฤษฎ์ ชัยยา รหัส 47380277

นายธนุวัฒน์ แก้วกัน รหัส 47380334

สารบัญ

หน้า

บทคัดย่อภาษาไทย .....ก

บทคัดย่อภาษาอังกฤษ .....ข

กิตติกรรมประกาศ .....ค

สารบัญ .....ง

สารบัญตาราง .....ฉ

สารบัญรูป .....ช

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของโครงงาน.....1

1.2 วัตถุประสงค์ของโครงงาน.....1

1.3 ขอบเขตของโครงงาน.....1

1.4 ขั้นตอนการดำเนินงาน.....2

1.5 ผลที่คาดว่าจะได้รับ.....3

1.6 งบประมาณที่ใช้ทำโครงงาน.....3

บทที่ 2 ทฤษฎีเบื้องต้น

2.1 ความชื้น (Humidity).....4

2.1.1 ความชื้นสัมบูรณ์ (Absolute Humidity).....4

2.1.2 ความชื้นสัมพัทธ์ (Relative Humidity).....4

2.2 ความเป็นมาของเพลเทียร์.....6

2.3 หลักการทำงานของอุปกรณ์เพลเทียร์ (Peltier).....7

2.2.1 ทฤษฎีที่เกี่ยวกับอุปกรณ์เพลเทียร์.....7

2.4 เชื้อราและการเก็บรักษาอุปกรณ์.....11

บทที่ 3 การออกแบบ

3.1 แนวคิดในการออกแบบ.....13

3.1.1 การใช้เพลเทียร์ควบคุมความชื้น.....14

สารบัญ(ต่อ)

หน้า

3.2 การออกแบบวงจร....16
3.2.1 การทำงานของวงจร....17
3.2.2 LCD Module....19
3.3 ขั้นตอนการปฏิบัติงานของวงจรควบคุมอุณหภูมิและความชื้น....20
3.3.1 โปรแกรมหลัก 1 โปรแกรมหลักสำหรับควบคุมความชื้น....21
3.3.2 โปรแกรมย่อยสำหรับตั้งค่าความชื้นค่าสูง....22
3.3.3 โปรแกรมย่อยสำหรับตั้งค่าความชื้นค่าต่ำ....23
3.4 การออกแบบตู้ควบคุมความชื้น (Chamber)....24
3.4.1 โครงสร้างตู้ควบคุมความชื้น....24
3.4.2 ฝาปิดตู้....25

บทที่ 4 การปฏิบัติและการทดลอง
4.1 การสร้างวงจรควบคุมความชื้น....26
4.2 การติดตั้งเพลเทียร์....27
4.3 โครงสร้างของตู้ควบคุมความชื้น....28
4.4 การทดลอง....30
4.5 ผลการทดลอง....30
4.5.1 การทดลองช่วงที่ 1....31
4.5.2 การทดลองช่วงที่ 2....32
4.5.3 การทดลองช่วงที่ 3....33

บทที่ 5 สรุปผลการดำเนินงาน
สรุปผลการดำเนินงาน....34

เอกสารอ้างอิง....35
ภาคผนวก....36
ประวัติผู้เขียนโครงงาน....87

สารบัญตาราง

ตารางที่ หน้า

2.1 การจัดการควบคุมความชื้นที่อุณหภูมิปกติ, ความชื้นต่ำ.....12

3.1 ตารางชื่อและหน้าที่ของสัญญาณต่างๆของ LCD.....19

4.1 ผลการทดลองเมื่อตั้งค่าความชื้น 50-60%.....31

4.2 ผลการทดลองเมื่อตั้งค่าความชื้น 40-50%.....32

4.2 ผลการทดลองเมื่อตั้งค่าความชื้น 30-40%.....33

สารบัญรูป

รูปที่ หน้า

2.1 โนโมกราฟและการใช้งาน....5

2.2 การเกิดซีเบ็ค โวลท์เตจ....6

2.3 วงจรการทดลองเพลเทียร์....6

2.4 ปรากฏการณ์เพลเทียร์....7

2.5 การจัดเรียงตัวของ P-N Junction ภายในตัวเพลเทียร์....9

2.6 ลักษณะของการส่งผ่านพลังงานความร้อนระหว่างเพลตร้อนกับเพลตเย็น....9

2.7 ลักษณะรูปร่างของเพลเทียร์ที่ใช้กันทั่วไป....10

2.8 กราฟการเจริญเติบโตของเชื้อรา....11

2.9 ช่วงที่เหมาะสมต่อการควบคุมความชื้นให้แก่อุปกรณ์ชนิดต่างๆ....12

3.1 การทำงานของการควบคุมความชื้น....13

3.2 Block Diagram ของการเชื่อมต่ออุปกรณ์เข้ากับ Microcontroller....14

3.3 หลักการควบคุมความชื้นของอุปกรณ์เพลเทียร์....15

3.4 วงจรควบคุมความชื้นที่ออกแบบ....16

3.5 Block Diagram ของตัวตรวจจับความชื้นและอุณหภูมิ SHT15....18

3.6 การต่อใช้งานจอแสดงผล LCD....19

3.7 แผนภาพโปรแกรมหลัก 1....21

3.8 แผนภาพโปรแกรมย่อย 1....22

3.9 แผนภาพโปรแกรมย่อย 2....23

3.10 โครงสร้างของตู้ควบคุมความชื้น....24

3.11 โครงสร้างฝาปิดตู้ควบคุมความชื้น....25

4.1 วงจรควบคุมความชื้นที่ประกอบเสร็จแล้ว....26

4.2 การติดตั้งเพลเทียร์เข้ากับ Heat Sink....27

4.3 โครงสร้างของตู้ควบคุมความชื้นที่ทำการประกอบ....28

4.4 โครงสร้างของฝาปิดตู้ควบคุมความชื้น....28

4.5 การติดตั้งอุปกรณ์ควบคุมความชื้นเข้ากับตู้ควบคุมความชื้น....29

4.6 ตู้ควบคุมความชื้นเมื่อประกอบเสร็จแล้ว....29

4.7 การทำงานของตู้ควบคุมความชื้น....30

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.8 กราฟผลการทดลองช่วงที่ 1	31
4.9 กราฟผลการทดลองช่วงที่ 2	32
4.10 กราฟผลการทดลองช่วงที่ 3	33

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ความชื้นเป็นอีกสาเหตุหนึ่งที่ทำให้ประสบปัญหาในการเก็บรักษาอุปกรณ์ต่างๆหรือแม้กระทั่งผลผลิตทางการเกษตร เมื่อมีความชื้นสูงเกินไปหรือต่ำเกินไปอาจสร้างความเสียหายให้แก่อุปกรณ์ได้ เช่น เลนส์ของกล้องถ่ายรูป เมื่อเก็บกล้องไว้ในที่ที่มีความชื้นสูง เชื้อราในบริเวณนั้นจะสร้างความเสียหายแก่เลนส์ของกล้อง เนื่องจากเมื่อมีความชื้นสูงส่งผลให้เชื้อราสามารถเจริญเติบโตได้ดี ดังนั้นการรักษาความชื้นให้อยู่ในสภาวะที่เหมาะสมจึงเป็นสิ่งที่จำเป็นมากในการเก็บรักษาอุปกรณ์ เพราะจะสามารถควบคุมการเจริญเติบโตของเชื้อราได้ เมื่อมีสิ่งที่สามารถควบคุมความชื้นได้ก็จะทำให้เก็บรักษาอุปกณ์ได้นานและมีประสิทธิภาพ

โครงงานนี้ผู้จัดทำได้นำเสนอเกี่ยวกับตู้ควบคุมความชื้น อธิบายถึงคุณสมบัติ หลักการใช้งาน และการนำตู้ควบคุมความชื้นนี้ไปประยุกต์ใช้กับอุปกรณ์ต่างๆได้อย่างเหมาะสม โดยที่ผู้ใช้งานสามารถควบคุมความชื้นได้ตามต้องการและสะดวกต่อการใช้งาน

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อศึกษาการทำงานของวงจรควบคุมความชื้นที่ใช้ควบคุมความชื้นภายในตู้ควบคุมความชื้น

1.2.2 เพื่อนำตู้ควบคุมความชื้นไปประยุกต์ใช้กับอุปกรณ์อื่นๆได้อย่างเหมาะสม

1.2.3 เพื่อศึกษาการเขียนโปรแกรมสำหรับใช้กับวงจรควบคุมความชื้น

1.2.4 เพื่อสามารถควบคุมความชื้นได้ตามความต้องการ

1.2.5 เพื่อนำความรู้ที่ได้มาเผยแพร่ให้กับบุคคลที่สนใจเพื่อใช้ในการศึกษาและพัฒนาต่อไป

1.3 ขอบเขตของโครงงาน

1.3.1 ตู้ควบคุมความชื้นขนาด 20×33×25 เซนติเมตร

1.3.2 ศึกษาเกี่ยวกับช่วงของความชื้นที่ใช้ในการควบคุม

1.3.3 อธิบายหลักการทำงานและคุณสมบัติของ

- อุปกรณ์ต่างๆที่ใช้ในการสร้างตู้ควบคุมความชื้น
- วงจรควบคุมความชื้นที่อยู่ภายในตู้ควบคุมความชื้น

1.3.4 เขียนโปรแกรมเพื่อเชื่อมต่อกับวงจรควบคุมความชื้นโดยใช้ไมโครคอนโทรลเลอร์

1.3.5 ทำการทดสอบโปรแกรมและวงจรควบคุมความชื้น

1.3.6 สรุปผลและนำเสนอโครงงาน

1.4 ขั้นตอนการดำเนินงาน

	2550			2551				
	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1. ศึกษาหลักการทำงาน ของวงจรควบคุมความชื้น และอุปกรณ์ต่างๆที่ใช้ในวงจร	←	→						
2. ทำการศึกษาเกี่ยวกับคุณสมบัติต่างๆของวงจร และการใช้งานของวงจร		←	→					
3. ศึกษาการเขียนโปรแกรมในการเชื่อมต่อเพื่อใช้ควบคุมวงจร			←	→				
4. เขียนโปรแกรมเพื่อเชื่อมต่อกับวงจรควบคุมความชื้น และทำการทดสอบโปรแกรม					←	→		
5. สรุป วิเคราะห์ผล และจัดทำรายงาน							←	→

1.5 ผลที่คาดว่าจะได้รับ

1.5.1 มีความรู้ความเข้าใจเกี่ยวกับวงจรควบคุมความชื้น

1.5.2 มีความเข้าใจเกี่ยวกับการเขียนโปรแกรมเพื่อเชื่อมต่อกับวงจรควบคุมความชื้น

1.5.3 สามารถใช้ตู้ควบคุมความชื้นได้อย่างมีประสิทธิภาพ

1.5.4 สามารถนำวงจรควบคุมความชื้นไปประยุกต์ใช้กับอุปกรณ์อื่นๆได้อย่างเหมาะสม

1.5.5 ใช้เป็นแนวทางในการพัฒนาอุปกรณ์ที่ใช้ในการควบคุมความชื้น และสามารถวิเคราะห์เขียนโปรแกรมไมโครคอนโทรลเลอร์ที่ใช้ควบคุมได้

1.6 งบประมาณที่ใช้ในการทำโครงงาน

ค่าหนังสือประกอบการทำโครงงาน	1,000	บาท
ค่าถ่ายเอกสาร	500	บาท
ค่ากระดาษ	500	บาท
ค่าปริ้นท์งาน	500	บาท
ค่าอุปกรณ์	2,000	บาท
ค่าใช้จ่ายอื่นๆ	1,000	บาท
รวมเป็นเงินทั้งสิ้น	5,500	บาท

บทที่ 2
ทฤษฎีเบื้องต้น

2.1 ความชื้น (Humidity)

ความชื้น (Humidity) หมายถึง ปริมาณไอน้ำที่มีอยู่ในอากาศเมื่อน้ำได้รับความร้อน น้ำจะเปลี่ยนสถานะจากของเหลวกลายเป็นไอเรียกว่า การระเหย ซึ่งความร้อนที่ใช้ในการทำให้น้ำระเหยกลายเป็นไอน้ำ เรียกว่า ความร้อนแฝง (Latent Heat) เมื่ออากาศเย็นลงไอน้ำจะเริ่มกลั่นตัวเป็นละออง และคายความร้อนแฝงออกมาด้วย อากาศจะได้รับไอน้ำได้มากหรือน้อยอยู่กับอุณหภูมิ ดังนั้นอุณหภูมิจึงเป็นตัวกำหนดปริมาณไอน้ำในอากาศ อากาศที่มีอุณหภูมิสูงจะรับไอน้ำได้มากกว่าอากาศที่มีอุณหภูมิต่ำถ้าอากาศไม่สามารถรับไอน้ำได้เรียกว่า ไอน้ำอิ่มตัว (Saturate)

2.1.1 ความชื้นสัมบูรณ์ (Absolute Humidity)

คือน้ำหนักของไอน้ำที่มีอยู่จริงในปริมาตรของอากาศจำนวนหนึ่งคำนวณได้จากน้ำหนักของไอน้ำต่อหนึ่งหน่วยปริมาตรของอากาศ หรือกล่าวอีกนัยหนึ่งว่าความชื้นสัมบูรณ์คือความหนาแน่นของไอน้ำในอากาศหน่วยคิดเป็นกรัมต่อลูกบาศก์เมตร เช่นน้ำหนักของไอน้ำ 25 กรัม ต่อลูกบาศก์เมตร ความชื้นสัมบูรณ์มักจะไม่ค่อยนิยมนำมาใช้ในทางอุตุนิยมวิทยาเพราะเมื่ออากาศลอยตัวขึ้นหรือจมตัวลงจะทำให้ปริมาตรของอากาศเปลี่ยนแปลงเนื่องจากบริเวณรอบ ๆ ความกดอากาศจะเปลี่ยนแปลง แม้ว่าไอน้ำที่มีอยู่ในอากาศจะคงที่ไม่เปลี่ยนแปลงเช่น ไอน้ำที่มีอยู่จำนวนไม่เท่ากันเมื่อปริมาตรเพิ่มขึ้น ความชื้นสัมบูรณ์ลดลง แต่เมื่อปริมาตรลดลง ความชื้นสัมบูรณ์จะเพิ่มขึ้น

2.1.2 ความชื้นสัมพัทธ์ (Relativity Humidity)

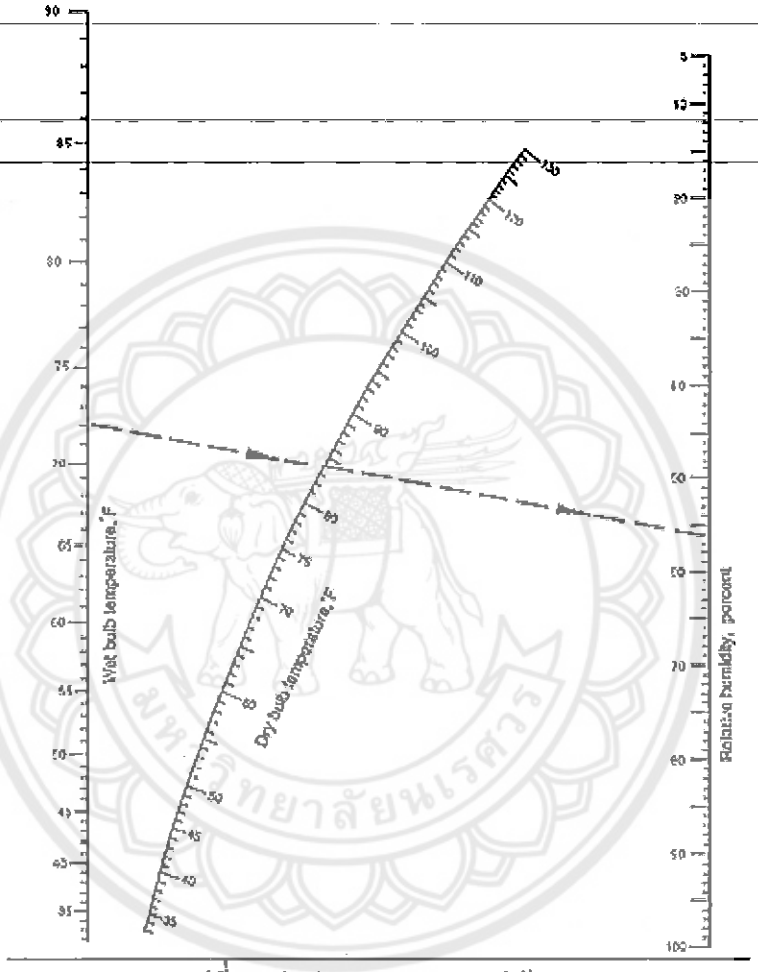
หมายถึง "อัตราส่วนของปริมาณไอน้ำที่มีอยู่จริงในอากาศ ต่อ ปริมาณไอน้ำที่จะทำให้อากาศอิ่มตัว ณ อุณหภูมิเดียวกัน" หรือ "อัตราส่วนของความดันไอน้ำที่มีอยู่จริง ต่อ ความดันไอน้ำอิ่มตัว" ค่าความชื้นสัมพัทธ์แสดงในรูปของร้อยละ (%)

วิธีที่แพร่หลายในการหาความชื้นสัมพัทธ์เรียกว่า วิธีวัดอุณหภูมิกระเปาะเปียก-กระเปาะแห้ง เราใช้เทอร์โมมิเตอร์จำนวนสองตัว ตัวแรกสำหรับวัดอุณหภูมิของอากาศแห้งตามธรรมดา อีกตัวหนึ่งจะถูกหุ้มห่อด้วยวัสดุอุ้มน้ำ และหล่อไว้ด้วยน้ำ

เมื่ออากาศมีความชื้นสัมพัทธ์ต่ำ น้ำจะสามารถระเหยกลายเป็นไอได้มากและ ซึ่งอัตราการระเหยที่เร็ว นี้ จะทำให้อุณหภูมิของกระเปาะเปียกต่ำมากกว่าอุณหภูมิของกระเปาะแห้งมาก ในขณะที่หากความชื้นในอากาศสูงมากขึ้น อัตราการระเหยการเป็นไอของน้ำในกระเปาะเปียกจะน้อยลง ทำให้นำพาความร้อนออกไปได้น้อยลง ผลก็คือความแตกต่างของอุณหภูมิของกระเปาะทั้งสองจะน้อยลง และเมื่อใดที่ความชื้นสัมพัทธ์ของอากาศเป็น 100% อัตราการระเหยของน้ำใน

กระเปาะเปียกจะเป็นศูนย์ด้วย ทำให้ไม่มีความแตกต่างระหว่างอุณหภูมิของกระเปาะเปียกและกระเปาะแห้งทั้งสอง

เมื่อเราทราบอุณหภูมิของกระเปาะเปียกและกระเปาะแห้งแล้ว เราสามารถหาความชื้นสัมพัทธ์ได้จากการเทียบกับชาร์ตที่เรียกว่า โนโมกราฟ (Nomograph) หรือ ไซโคเมตริกชาร์ต (Psychometric chart) โดยที่การหาจากโนโมกราฟจะสะดวกรวดเร็วกว่าในกรณีนี้

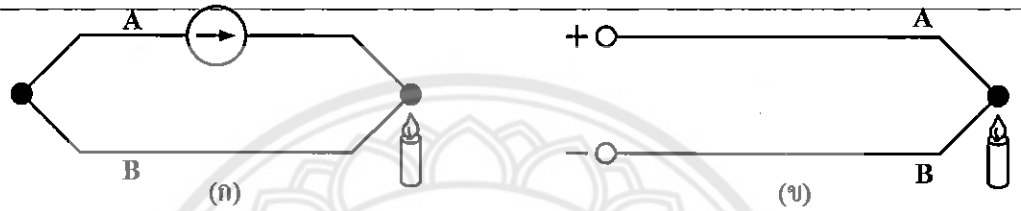


รูปที่ 2.1 โนโมกราฟและการใช้งาน

ตัวอย่างการใช้โนโมกราฟแสดงในรูปข้างล่าง สมมติว่าอุณหภูมิกระเปาะเปียกเป็น 22.2 °C (72 °F) และอุณหภูมิกระเปาะแห้งเป็น 28.8 °C (84 °F) จะหาความชื้นสัมพัทธ์ได้เป็น 56%

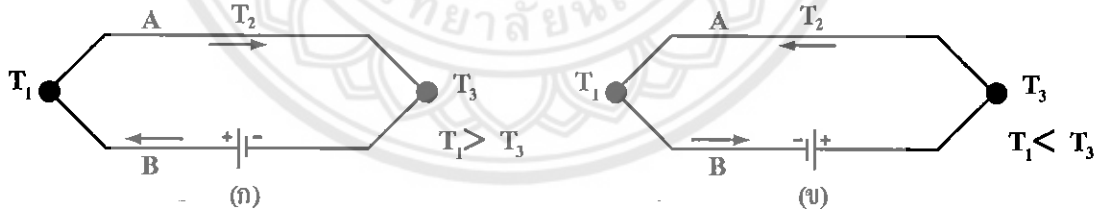
2.2 ความเป็นมาของเพลเทียร์

ในปี ค.ศ. 1821 นักวิทยาศาสตร์ชาวเยอรมัน Thomas Seebeck ทำการศึกษาพบว่าเมื่อนำขดลวดโลหะ 2 เส้นที่ทำด้วยโลหะต่างชนิดกันมาเชื่อมต่อปลายทั้งสองเข้าด้วยกัน ถ้าปลายจุดต่อทั้งสองมีอุณหภูมิต่างกัน จะเกิดกระแสไฟฟ้าไหลในวงจรขดลวดทั้งสองแสดงดังรูปที่2.2 (ก) ปริมาณการไหลของกระแสไฟฟ้าเปลี่ยนแปลงไปตามผลต่างอุณหภูมิที่ปลายจุดต่อทั้งสอง และถ้าเปิดปลายจุดต่อด้านหนึ่งออกจะทำให้เกิดแรงเคลื่อนไฟฟ้าที่ปลายด้านเปิดแสดงดังในรูปที่2.2 (ข) แรงเคลื่อนไฟฟ้านี้เรียกว่า "ซีเบ็คโวลท์เตจ"



รูปที่ 2.2 การเกิดซีเบ็ค โวลท์เตจ

ต่อมาในปี ค.ศ. 1834 Jean C.A. Peltier พบว่าเมื่อจ่ายกระแสไฟฟ้าเข้าไปในวงจรลักษณะเดียวกับซีเบ็คสร้างขึ้นโดยใช้ลวดด้านหนึ่งทำจากบิสมัสและอีกด้านหนึ่งทำจากแอนติโมนิจะทำให้เกิดอุณหภูมิที่ปลายจุดต่อแตกต่างกัน โดยปลายข้างหนึ่งจะร้อนและปลายอีกข้างหนึ่งจะเย็น วงจรการทดลองของเพลเทียร์เป็นแสดงดังรูปที่ 2.3



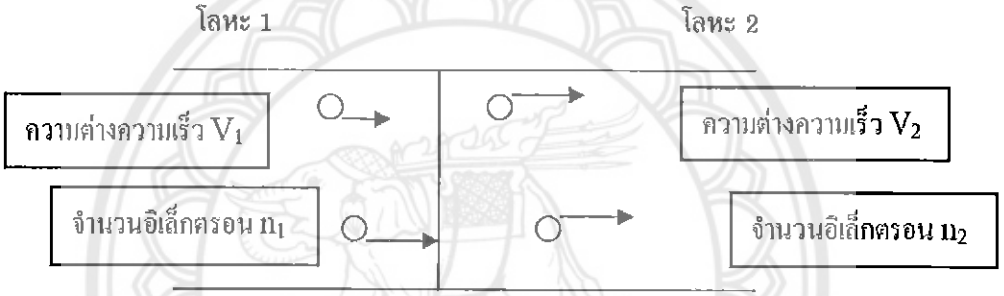
รูปที่ 2.3 วงจรการทดลองเพลเทียร์

เมื่อมีกระแสไฟฟ้าจากแหล่งจ่ายภายนอกไหลผ่านเข้าไปในในวงจรเป็นแสดงดังในรูปที่ 2.3 (ก) จะทำให้จุดต่อ T1 มีอุณหภูมิสูงกว่าจุดต่อ T3 เมื่อมีกระแสไฟฟ้าจากแหล่งจ่ายภายนอกไหลผ่านเข้าไปในในวงจรเป็นแสดงดังในรูปที่ 2.3 (ข) จะทำให้จุดต่อ T3 มีอุณหภูมิสูงกว่าจุดต่อ T1 ซึ่งจากการค้นพบของซีเบ็คและ เพลเทียร์ทำให้เกิดการศึกษาค้นคว้าเกี่ยวกับเทอร์โมอิเล็กตริกต่อมา

2.3 หลักการทำงานของอุปกรณ์เพลเทียร์ (Peltier)

2.3.1 ทฤษฎีเกี่ยวกับอุปกรณ์เพลเทียร์

อุปกรณ์ทำความเย็นเทอร์โมอิเล็กตริกที่ใช้กันอยู่ทั่วๆไปส่วนใหญ่เป็นอุปกรณ์ทางด้านโซลิดสเตตที่สามารถปั๊มความร้อนออกจากพื้นผิวได้รวดเร็วโดยหลักการนี้เหมาะสำหรับการทำความเย็นให้กับอุปกรณ์ขนาดเล็ก เช่น ใช้ทำความเย็นให้กับอุปกรณ์ตรวจจับคู่ควบประจุ หรือ CCD (Charge Coupled Device sensors) โฟโต้ไดโอดและอุปกรณ์กำเนินแสงเลเซอร์ที่ผลิตจากหลักการสารกึ่งตัวนำโดยหลักการทำงานดังกล่าวใกล้เคียงกับปรากฏการณ์ซีเบค (See beck effect) กล่าวคือเมื่อเกิดความแตกต่างของอุณหภูมิระหว่างขั้วต่อ 2ขั้ว ก็จะทำให้เกิดสนามแม่เหล็กไฟฟ้า (Electro magnetic field) เหนี่ยวนำให้เกิดความร้อนและความเย็นที่แตกต่างกันระหว่างขั้วทั้งสอง นั่นก็คือหลักการของคู่ควบความร้อน (thermocouple)



รูปที่ 2.4 ปรากฏการณ์เพลเทียร์

จากรูปที่ 2.4 แสดงให้เห็นว่าอิเล็กตรอนเคลื่อนที่ข้ามรอยต่อโลหะซึ่งมีพลังงาน 2 ฝั่งแตกต่างกัน ทำให้เกิดความร้อนหรือความเย็นก็ได้ ขึ้นอยู่กับทิศทางการไหลของกระแสไฟฟ้า แต่โดยปกติปรากฏการณ์นี้จะใช้ในการสร้างความเย็นมากกว่า

วิธีการของปรากฏการณ์เพลเทียร์ (Peltier) ก็คือจะผ่านกระแสไฟฟ้าเข้าไปในรอยต่อของสารกึ่งตัวนำ P - N ผลที่ได้ทั้งความร้อนและความเย็นที่บริเวณรอยต่อจะขึ้นอยู่กับทิศทางการไหลของกระแสไฟฟ้า ดังแสดงหลักการทำงานพื้นฐานไว้ดังรูปที่ 1 โดยแสดงให้เห็นว่าอิเล็กตรอนได้ไหลจากซ้ายข้ามไปทางขวา ตัดผ่านรอยต่อโลหะไป โดยสนามไฟฟ้า E_1 เป็นค่าแสดงคุณสมบัติของโซลิดสเตต ว่ามีพลังงานเฉลี่ยของโลหะที่ 1 เป็นเท่าไร ส่วนค่าสนามไฟฟ้า E_2 ก็เช่นเดียวกัน จะแสดงคุณสมบัติของโลหะที่ 2 ดังนั้นอิเล็กตรอนในโลหะที่ 1 จะมีพลังงานในการเคลื่อนที่ผ่านรอยต่อเท่ากับ

$$E_{in} = n_1 E_1 v_1 \qquad (2.1)$$

พลังงานเฉลี่ยที่เอาท์พุตออกจากรอยต่อเท่ากับ

$$E_{out} = n_2 E_2 v_2 \tag{2.2}$$

โดยที่ E_{in}, E_{out} คือพลังงานอินพุตและพลังงานเอาท์พุต ตามลำดับ

E_1, E_2 คือสนามไฟฟ้าที่โลหะ 1 และ 2 ตามลำดับ

n_1, n_2 คือจำนวนของอิเล็กตรอนฝั่งโลหะที่ 1 และ 2 ตามลำดับ

v_1, v_2 คือความเร็วในการเคลื่อนที่ของอิเล็กตรอนฝั่งโลหะที่ 1 และ 2 ตามลำดับ

และจากสมการกระแสไฟฟ้า (I) ที่ไหลผ่านรอยต่อก็จะมีค่าเท่ากับ

$$I = n_1 \times v_1 \times q = n_2 \times v_2 \times q \tag{2.3}$$

โดยที่ q คือประจุไฟฟ้า และ I คือกระแสไฟฟ้า โดยสรุปแล้วจะได้ผลรวมของพลังงานความร้อนที่บริเวณรอยต่อมีค่าเท่ากับ

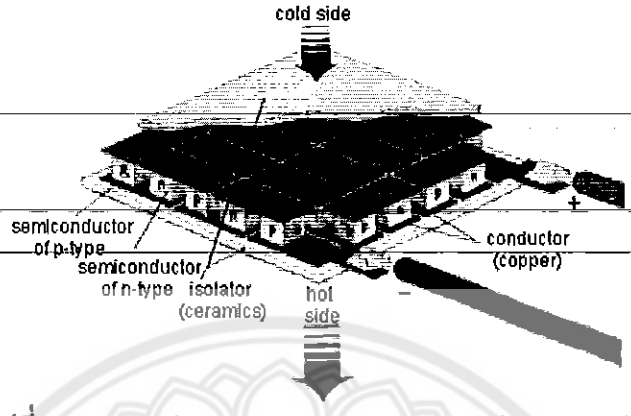
$$E_{out} - E_{in} = I / P(E_1 - E_{out}) \tag{2.4}$$

จากสมการแสดงให้เห็นได้ว่า ความร้อนที่ถูกดึงออกด้วยอุปกรณ์เพลเทียร์ จะเป็นสัดส่วนโดยตรงกับกระแสไฟฟ้าและผลต่างของสนามไฟฟ้าที่โลหะที่ 1 และ 2

ปรากฏการณ์เพลเทียร์นี้จะขึ้นอยู่กับโลหะทั้งสองฝั่งซึ่งปรากฏการณ์ดังกล่าวผลของการสะสมทางความร้อนจะขึ้นอยู่กับค่าความเหนี่ยวนำของอิเล็กตรอนที่เปลี่ยนแปลงข้ามรอยต่อของโลหะดังกล่าวผลตอบสนองด้านความร้อนของปรากฏการณ์นี้กับโลหะธรรมดา เช่น เหล็ก หรือทองแดงนั้น จะมีผลตอบสนองน้อยมากและยากแก่การวัดค่าได้ แต่ในทางปฏิบัติแล้วต้องการความร้อนสูงมาก เพื่อนำไปใช้งาน ดังนั้นจึงต้องเปลี่ยนจากโละธรรมดาเป็นสารกึ่งตัวนำซึ่งจะให้ผลตอบสนองสูงมาก อีกทั้งนำไปใช้งานได้ทั้งความเย็นและความร้อน ขึ้นอยู่กับทิศทางของกระแสไฟฟ้า

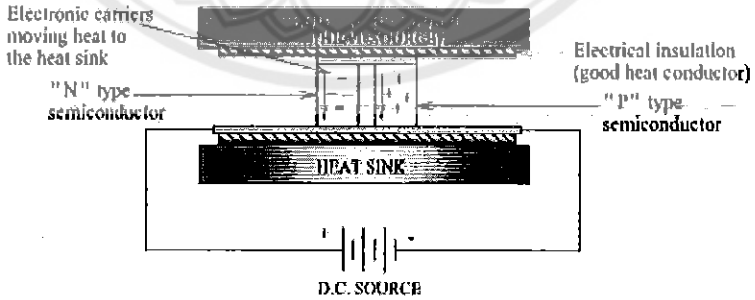
อุปกรณ์ทำความเย็นเทอร์โมอิเล็กตริก จะประกอบด้วยแผงรอยต่อ P-N วางเรียงกันระหว่างแผ่นเพลตเซรามิกตัวนำทั้ง 2 เพลต เมื่อจ่ายกระแสไฟฟ้าเข้าเพลตด้านแรกผ่านรอยต่อ P-N ความเย็นก็จะถูกสะสมที่เพลตนั้น ส่วนความร้อนจะส่งผ่านไปยังรอยต่อ P-N สู่เพลตอีกด้านหนึ่ง ทำให้เพลตอีกด้านมีความร้อนสะสมอยู่สูง โดยความร้อนจะถูกดูดกลืนที่เพลตด้านไบแอสกลับ (reverse biased junction) ส่วนด้านไบแอสตรง (forward biased) จะเป็นด้านปล่อยความร้อนแทน

สารกึ่งตัวนำที่นำมาใช้จึงจำเป็นที่จะต้องมีช่องว่างแบนด์ (band gap) แคบ โดยปกติจะใช้สารบิสมัทเทลลูไลด์ (bismuth telluride) เพื่อที่จะได้ผลิตพาหะส่วนน้อย (minority carriers) เพียงพอ ณ อุณหภูมิปกติ



รูปที่ 2.5 การจัดเรียงตัวของ P-N Junction ภายในตัวเพลเทียร์

ในทางกลับกัน ถ้าจะประยุกต์นำอุปกรณ์ดังกล่าวไปใช้งานเป็นตัวทำความร้อนก็สามารถทำได้เช่นกัน เพียงแต่ใช้งานในเพลตตรงข้ามเท่านั้นเอง แต่ยังคงต้องพยายามรักษาอุณหภูมิต่างๆ ให้คงที่ ถึงแม้อุณหภูมิรอบข้างจะเปลี่ยนแปลงไป สูงต่ำตามค่าที่ตั้งไว้ก็ตาม แต่ส่วนใหญ่แล้วตามมาตรฐานพื้นผิวเพลตทางด้านเย็นจะถูกนำไปใช้เป็นตัวทำความเย็นเทอร์โมอิเล็กตริก โดยจะต่อเข้ากับแผงควบคุมความเย็น ส่วนเพลตทางฝั่งตรงข้ามที่เป็นเพลตความร้อนนั้น มักจะต่อเข้ากับแผ่นระบายความร้อน (Heat sink) เพื่อระบายความร้อนด้วยอากาศขึ้นอยู่กับลักษณะการนำไปประยุกต์ใช้งานว่าจะทำความเย็นหรือความร้อน

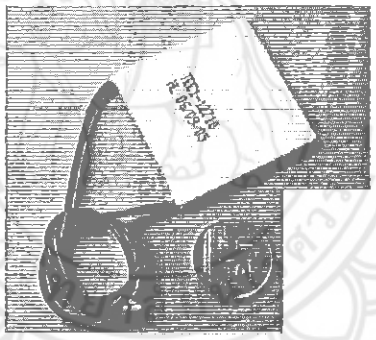


รูปที่ 2.6 ลักษณะของการส่งผ่านพลังงานความร้อนระหว่างเพลตร้อนกับเพลตเย็น

พารามิเตอร์ 2 ตัวที่จะต้องพิจารณาเป็นพิเศษสำหรับตัวทำความเย็นเทอร์โมอิเล็กตริกนี้ก็คืออุณหภูมิสูงสุดที่อุปกรณ์ยังสามารถคงอยู่ได้และความร้อนที่ปั๊มเข้าไป ความแตกต่างของอุณหภูมิสูงสุดที่วัดได้ต้องเป็นค่าที่ไม่มีโหลดความร้อนจากภายนอกบนเพลตความเย็นดังนั้นจะเห็นได้ว่าพลังงานไฟฟ้าอินพุตทั้งหมดที่ส่งเข้าไปยังตัวทำความเย็นจะถูกใช้เอาชนะพลังงานความ

ร้อนภายในที่รั่วไหลออกมาจะใช้ในการต่อต้านความร้อนที่แพร่ทะลุผ่านเพลตความเย็นยิ่งรั่วมากยิ่งทำให้ประสิทธิภาพในการทำความเย็นก็จะลดลงด้วย

อุณหภูมิระหว่างเพลตทั้งสองจะแตกต่างกันมากที่สุด โดยวิธีการนำเพลตความร้อนไปใส่ไว้ในสภาวะสุญญากาศ และใส่ไนโตรเจนแห้งเข้าไปด้วยในสภาพสุญญากาศนี้จะเป็นการกำจัดการแพร่กระจายของความร้อนดังนั้นความร้อนภายในจึงไม่สามารถรั่วออกมาได้ ทำให้ได้เพลตความเย็นที่มีความเย็นสูงที่สุดเมื่อเทียบกับเพลตความร้อนความแตกต่างของอุณหภูมิระหว่างเพลตทั้งสองจะอยู่ระหว่าง 60 ถึง 70 °C ที่สภาวะสุญญากาศผสมไนโตรเจนแห้ง ถ้าจะใช้โดยให้มีอุณหภูมิมากกว่านี้สามารถทำได้โดยใช้อุปกรณ์ชนิดหลายชั้น (multistage) ลักษณะจะคล้ายปิรามิดโดยเพลตความร้อนของชั้นบนก็คือเพลตความเย็นของชั้นล่างที่อยู่ติดกัน-การใช้จะลดอุณหภูมิลงมาตามชั้นพิจารณาเพลตความเย็นของชั้นบนสุดกับเพลตความร้อนของชั้นต่ำสุดผลลัพธ์ที่ได้ก็จะเหมือนมีเพลตแค่ 2 เพลตที่เสมือนเป็นเทอร์โมอิเล็กตริกชั้นเดียวแต่จะมีความแตกต่างของอุณหภูมิระหว่างเพลตสูงกว่าชนิดชั้นเดียว เช่น อุปกรณ์ชนิดสามชั้นจะมีอุณหภูมิต่างกันสูงสุด 100°C แต่อุปกรณ์ชนิดหกชั้นมีอุณหภูมิต่างกันสูงสุดแค่ 130 องศาเซลเซียสเท่านั้นและจากตัวอย่างของอุปกรณ์เพลเทียร์ที่ใช้งานกันทั่วไปได้แสดงดังรูป



รูปที่ 2.7 ลักษณะรูปร่างของเพลเทียร์ที่ใช้กันทั่วไป

พิจารณาปัญหาทางอิเล็กทรอนิกส์แล้วจะเห็นว่าตัวทำความเย็นเทอร์โมอิเล็กตริกสามารถออกแบบได้หลายชนิด เช่น ชนิดไร้รอยต่อ P-N Junction ขนาดเล็กๆหลายๆรอยต่อ หรืออาจจะใช้ขนาดใหญ่เพียงไม่กี่รอยต่อก็ได้ ก็สามารถให้คุณสมบัติทางความร้อนและขนาดความหนาบางเท่าเดิม แต่ราคาแบบหลังจะราคาถูกกว่าและแข็งแรงกว่าด้วย

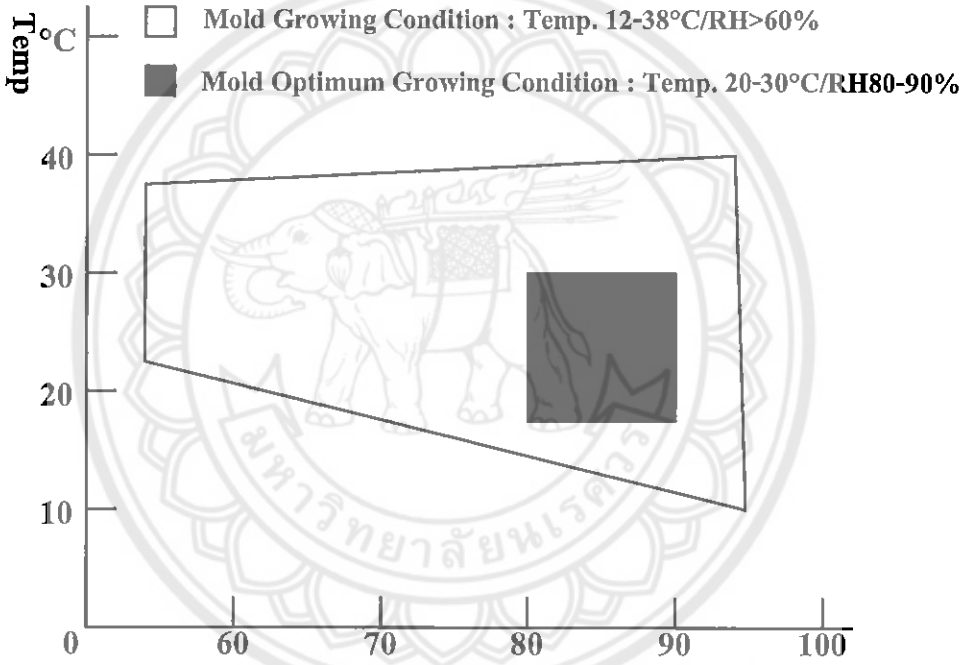
เนื่องจากอุปกรณ์เพลเทียร์มีค่าความต้านทานต่ำมาก ดังนั้นการออกแบบอุปกรณ์อิเล็กทรอนิกส์อื่นๆ ประกอบจึงทำได้ยาก และการออกแบบจะต้องคำนึงเสมอว่าจะส่งผลกระทบต่อส่วนอื่นๆ ด้วยหรือไม่

การติดตั้งอุปกรณ์เพลเทียร์ พื้นฐานของตัวทำความเย็นเทอร์โมอิเล็กตริกมีลักษณะที่เปราะบางแตกง่ายและราคาแพง ดังนั้นแผ่นระบายความร้อนจึงควรมีลักษณะบางแบนราบ และตัวถังต้องอยู่ในสภาพที่ไม่ต้องรับแรงกดจากภายนอก ดังนั้นอุปกรณ์นี้ส่วนใหญ่จะติดตั้งเข้ากับสาย

ที่ต่อเชื่อมเข้ากับเพลตความร้อน เพื่อป้องกันความร้อนรั่วไหลออก ส่วนตัวถังต้องวางไม่ให้เกิดแรงเครียดหรือแรงดันใดๆ และถ้าอุณหภูมิสูงกว่า 100 องศาเซลเซียส จะต้องเชื่อมรอยต่อ P-N เข้าด้วยกันด้วยลวดเชื่อมพิเศษจุดเชื่อมจะเล็กมากและละเอียดไม่ทำอันตรายกับพื้นผิวบริเวณรอยต่อ

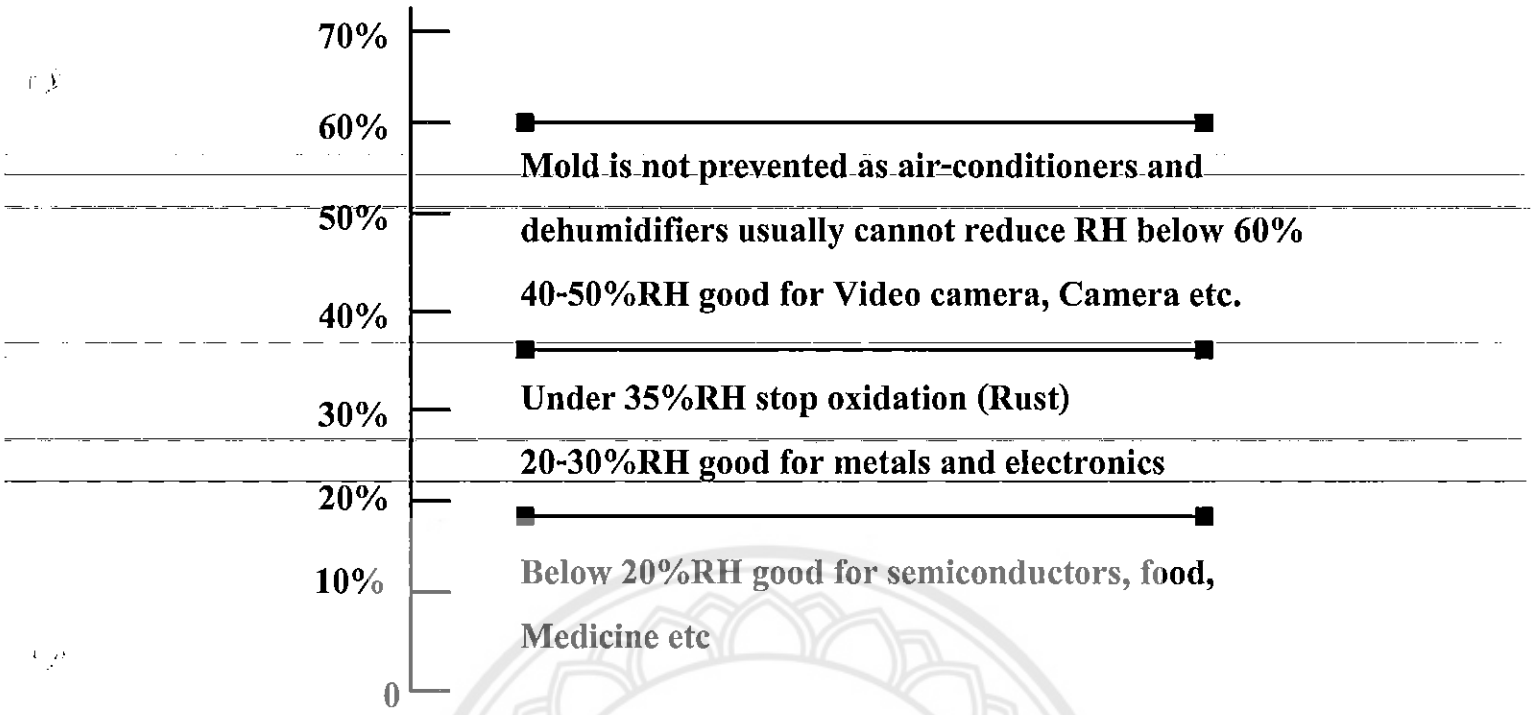
2.4 เชื้อราและการเก็บรักษาอุปกรณ์

เชื้อราเป็นปัจจัยสำคัญที่ทำให้อุปกรณ์ละเครื่องมือต่างๆเสื่อมสภาพ และมีอายุการใช้งานน้อยลง โดยสิ่งสำคัญที่ทำให้เชื้อราเจริญเติบโตคือ ความชื้น หากว่าพื้นที่ดที่มีความชื้นมากย่อมทำให้เชื้อรามีการเจริญเติบโตได้มากเช่นกัน ดังนั้น การลดควบคุมความชื้นจึงเป็นวิธีหนึ่งในการยับยั้งการเพิ่มจำนวนของเชื้อรา



รูปที่ 2.8 กราฟการเจริญเติบโตของเชื้อรา [10]

จากกราฟดังรูปที่ 2.9 จะเห็นได้ว่าเมื่อความชื้นมีค่ามากกว่า 60% จะทำให้เชื้อราเจริญเติบโตได้ เมื่อความชื้นยิ่งมากเชื้อรายิ่งทำให้การเติบโตของเชื้อรามากขึ้นด้วย ดังนั้น การควบคุมความชื้นให้มีค่าต่ำกว่า 60% จะสามารถเก็บรักษาอุปกรณ์หรือเครื่องมือส่วนใหญ่ได้ แต่อุปกรณ์บางชนิด เช่น อุปกรณ์อิเล็กทรอนิกส์ นั้นต้องการการเก็บรักษาที่มีความชื้นต่ำมาก เนื่องจากอุปกรณ์อิเล็กทรอนิกส์มีความเปราะบางและเสียหายง่าย ดังนั้นการเก็บรักษาอุปกรณ์แต่ละชนิดย่อมมีความแตกต่างกัน



รูปที่ 2.9 ช่วงที่เหมาะสมต่อการควบคุมความชื้นให้แก่อุปกรณ์ชนิดต่างๆ [10]

ตารางที่ 2.1 การจัดการควบคุมความชื้นที่อุณหภูมิปกติ, ความชื้นต่ำ [10]

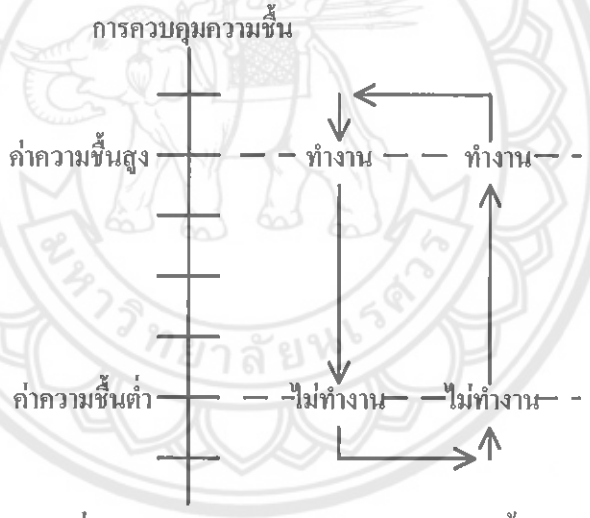
HUMIDITY	Suitable Article
40%RH ~ 50%RH	กล้องวีดิโอ, กล้องถ่ายรูป, เลนส์, กล้องจุลทรรศน์, ฟิล์ม, อุปกรณ์เกี่ยวกับดวงตา, คอมพิวเตอร์,สเตอริโอ,ซีดีและวีซีดี,หนังสือ, ภาพวาด, แสตมป์ ฯลฯ
30%RH ~ 40%RH	อุปกรณ์อิเล็กทรอนิกส์, พีซีบี, Indicator, เครื่องดนตรี, หนังสัตว์, ทองและเงิน, ของสะสมและของมีค่าต่างๆ
Under 30%RH	สมุนไพร, ใบชา,เมล็ดพืช, เครื่องเทศ, ยา,วัสดุแบบผง, อุปกรณ์ทางเคมี, อุปกรณ์สารกึ่งตัวนำ

บทที่ 3

การออกแบบ

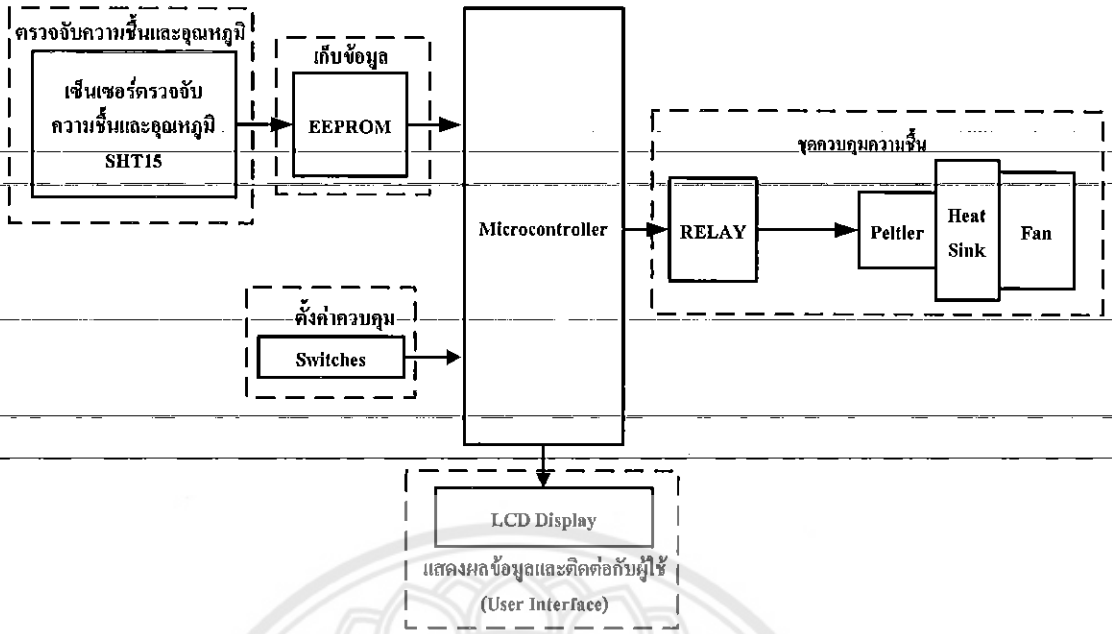
3.1 แนวคิดในการออกแบบ

สำหรับระบบควบคุมความชื้นนี้จะใช้ไมโครคอนโทรลเลอร์ตระกูล MCS51 ของบริษัท ATMEL โดยได้เลือกไมโครคอนโทรลเลอร์เบอร์ AT89C2051 เป็นหัวใจหลักในการควบคุมการทำงานทั้งหมด ซึ่งระบบควบคุมนี้สามารถนำไปประยุกต์ใช้ควบคุมอุณหภูมิได้อีกด้วย เนื่องจากระบบควบคุมนี้ใช้เซ็นเซอร์ ของบริษัท SENSIRION รุ่น SHT15 ที่มีคุณสมบัติตรวจจับได้ทั้งความชื้นและอุณหภูมิ ดังนั้นจึงสามารถตั้งค่าความชื้นและอุณหภูมิสูงและต่ำได้ เพื่อใช้ควบคุมค่าเหล่านั้นให้อยู่ในช่วงที่ต้องการได้อย่างมีประสิทธิภาพ โดยที่ระบบจะแยกชุดควบคุมความชื้นและชุดควบคุมอุณหภูมิไว้คนละส่วน



รูปที่ 3.1 การทำงานของการควบคุมความชื้น

จากรูปที่ 3.1 จะเห็นได้ว่าเมื่อความชื้นมีค่าสูงกว่าค่าความชื้นสูงที่ตั้งไว้ ชุดควบคุมความชื้นก็จะทำงานจนกระทั่งความชื้นลดต่ำลงเท่ากับหรือต่ำกว่าค่าความชื้นที่ตั้งไว้ ชุดควบคุมความชื้นก็จะหยุดทำงาน วนเวียนแบบนี้ไปเรื่อยๆ

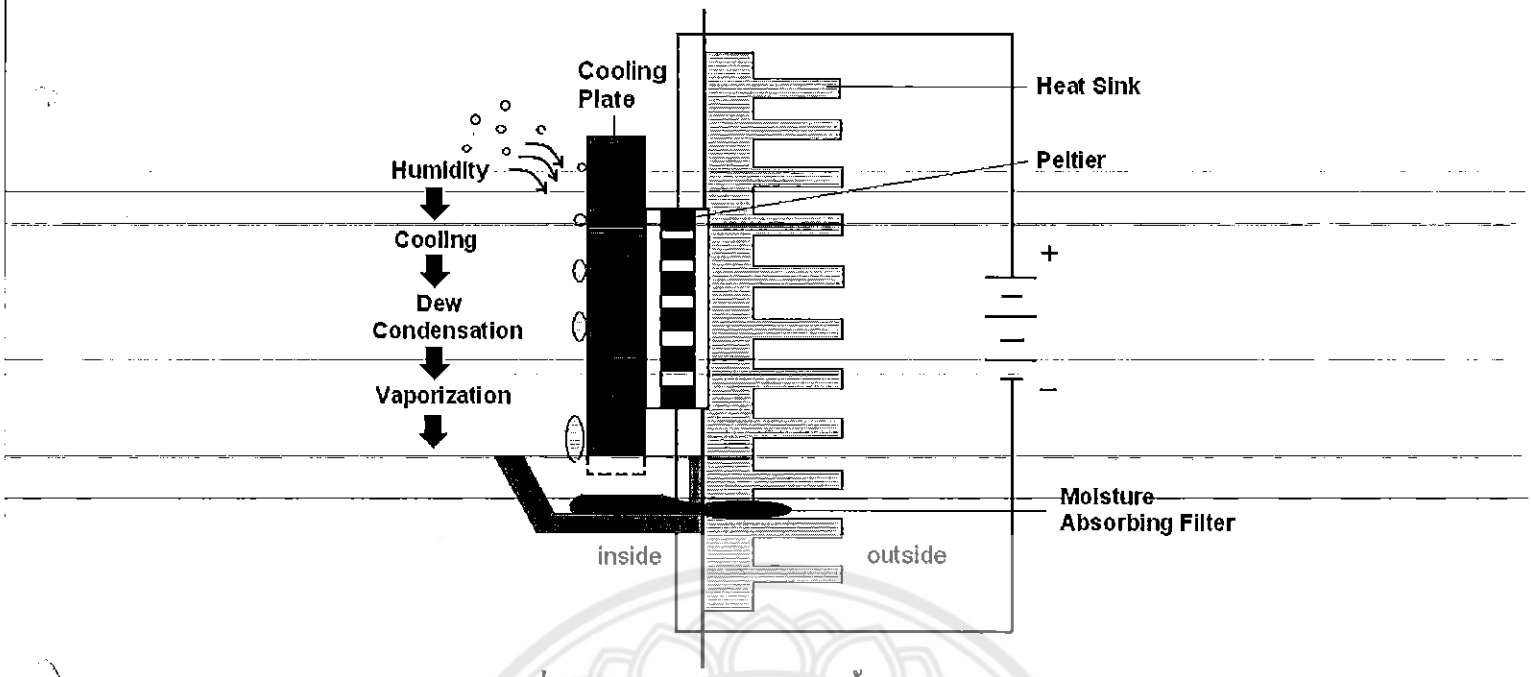


รูปที่ 3.2 Block Diagram ของการเชื่อมต่ออุปกรณ์เข้ากับ Microcontroller

เมื่อพิจารณาจาก Block Diagram ดังรูปที่ 3.2 จะเป็นภาพการทำงานโดยรวมของระบบ ซึ่งเมื่อเซ็นเซอร์ตรวจจับค่าอุณหภูมิและความชื้นได้ จะส่งข้อมูลไปยัง EEPROM ที่เป็นตัวเก็บข้อมูลของเซ็นเซอร์ แล้วส่งไปประมวลผลที่ไมโครคอนโทรลเลอร์ ซึ่งสามารถตั้งค่าควบคุมได้โดยใช้สวิตช์ จากนั้นไมโครคอนโทรลเลอร์จะแสดงผลออกทาง LCD Display และในกรณีที่ค่าอุณหภูมิและค่าความชื้นมีค่าสูงหรือต่ำกว่าค่าที่ตั้งไว้ ไมโครคอนโทรลเลอร์ก็จะสั่งการให้ชุดควบคุมอุณหภูมิและชุดควบคุมความชื้นทำงานโดยผ่านการ Relay

3.1.1 การใช้อุปกรณ์เพลเทียร์ควบคุมความชื้น

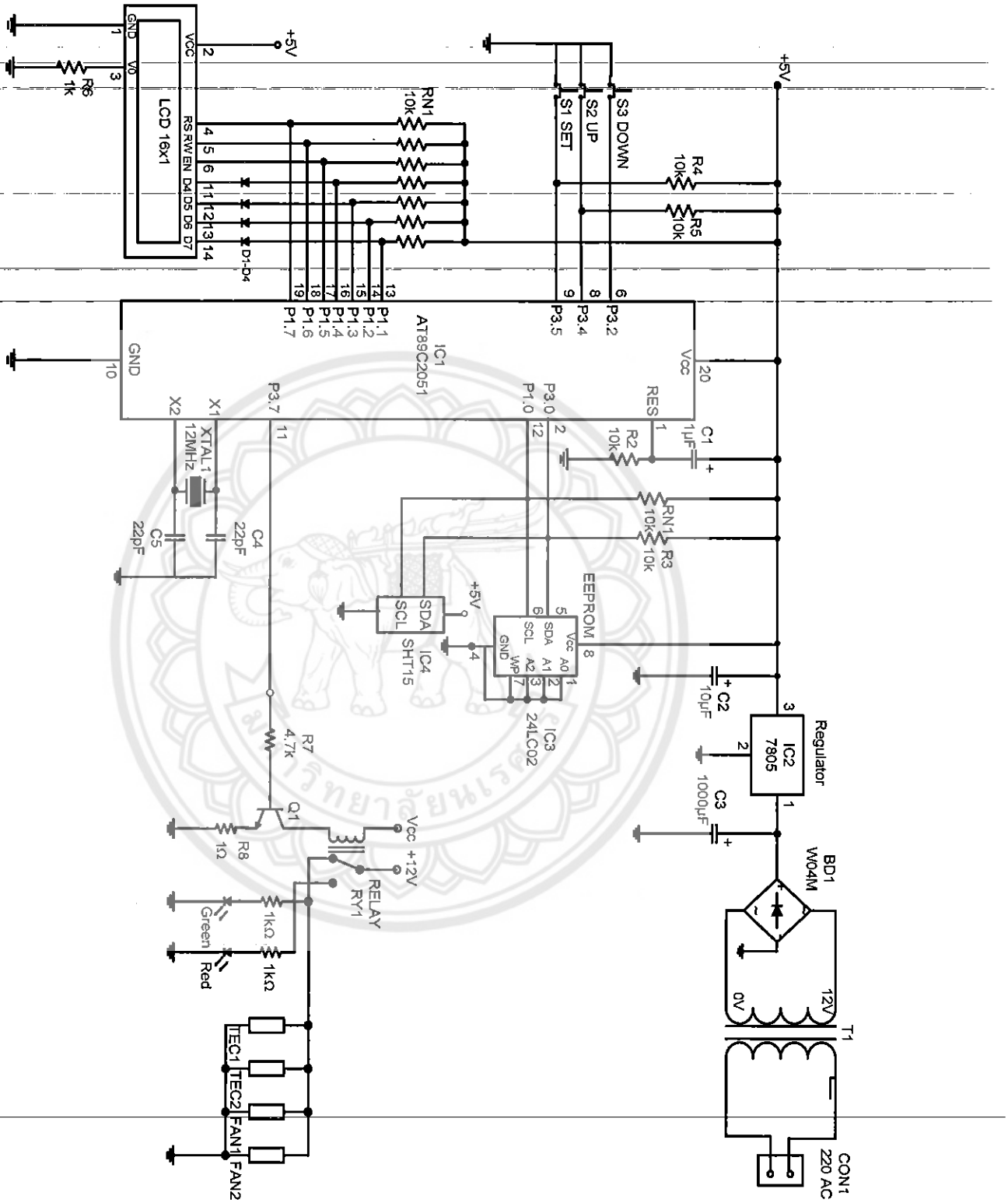
ความชื้นเป็นตัวแปรทางด้านกายภาพ ซึ่งสามารถเปลี่ยนแปลงโดยการเพิ่มขึ้นหรือลดลงได้ตลอดเวลา ซึ่งจะส่งผลกระทบต่อระบบต่างๆที่เราสนใจซึ่งไม่ว่าจะเป็นการทดลองเองหรือในระบบอุตสาหกรรมที่จำเป็นต้องควบคุมและสามารถตรวจจับการเปลี่ยนแปลงของตัวแปรเหล่านี้ ซึ่งเซนเซอร์เป็นอุปกรณ์ซึ่งสามารถใช้ตรวจจับการเปลี่ยนต่างๆ ของระบบ รวมทั้งความชื้น ซึ่งผลที่ได้จากเซนเซอร์ สามารถนำมาควบคุมการทำงานของระบบได้โดยผ่านการควบคุมด้วยไมโครคอนโทรลเลอร์ ซึ่งจะสามารถเพิ่มประสิทธิภาพในการควบคุมความชื้นให้อยู่ในช่วงที่ต้องการได้อย่างแม่นยำ



รูปที่ 3.3 หลักการควบคุมความชื้นของอุปกรณ์เพลเทียร์

จากรูปที่ 3.3 เมื่อจ่ายกระแสให้กับเพลเทียร์จะทำให้เพลตด้านหนึ่งเย็น อีกด้านหนึ่งร้อน ซึ่งด้านที่เย็นนั้นจะใช้เพลตที่มีหน้าสัมผัสที่กว้างกว่าเพลเทียร์เพื่อใช้ในการเพิ่มพื้นที่ผิวกระจายความเย็น ส่วนด้านร้อนจะถูกระบายความร้อนออกด้วย Heat sink เมื่อมีความชื้นมากกว่าค่าที่ได้ตั้งไว้ เพลเทียร์ด้านเย็นจะทำงาน และจะเป็นตัวจับไอน้ำในอากาศให้เกิดการควบแน่นกลายเป็นหยดน้ำแล้วไหลลงสู่แผ่นกรองความชื้นที่ใช้ระบายความชื้นออกไปด้านนอก ทำให้ได้ค่าความชื้นอยู่ในช่วงที่ตั้งไว้

3.2 การออกแบบวงจร



รูปที่ 3.4 วงจรควบคุมความชื้นที่ออกแบบ

3.2.1 การทำงานของวงจร

จากวงจรที่ใช้ควบคุมความชื้นและอุณหภูมิแสดงดังรูปที่ 3.4 หัวใจหลักของวงจรจะอยู่ที่ไมโครคอนโทรลเลอร์เบอร์ AT89C2051 ซึ่งเป็นไอซีที่ใช้สำหรับบรรจุโปรแกรมการทำงานของวงจรทั้งหมด โดยที่การทำงานของวงจรจะเริ่มต้นที่ภาคจ่ายไฟเป็นอันดับแรก แรงดันไฟฟ้ากระแสสลับ 12 โวลต์ จากหม้อแปลงที่ป้อนเข้ามาจะถูกเร็กติไฟล์ (Rectified) โดย บริดจ์ไดโอด (BD1) เพื่อเรียงไฟกระแสสลับให้เป็นไฟกระแสตรง 12 โวลต์ และถูกลดแรงดันให้เหลือ 5 โวลต์ โดยไอซีเรกูเลเตอร์ 7805 โดยจะมีตัวเก็บประจุ C2 และ C3 เป็นตัวช่วยในการกรองแรงดันให้เรียบ

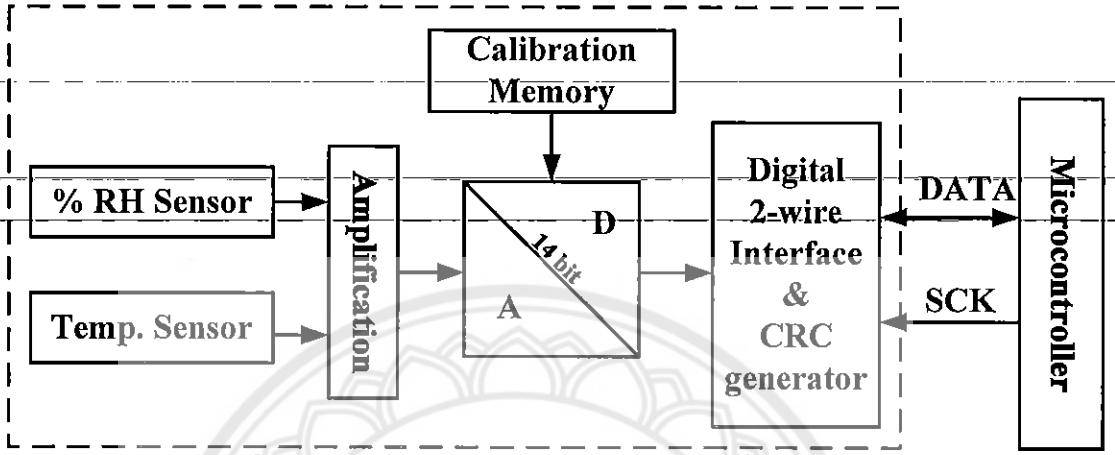
ในส่วนการใช้งาน LCD ซึ่งเป็นส่วนที่ใช้แสดงผล เป็นการต่อขาใช้งานในโหมด 4 บิต ซึ่งจะเห็นว่าขาข้อมูลจะถูกใช้เพียง 4 เส้นเท่านั้น คือขา D4-D7 โดยมีไดโอด D1-D4 เป็นตัวเรียงกระแสให้ข้อมูลผ่านโมดูล LCD อย่างถูกต้อง ส่วนขาควบคุมโมดูล LCD นั้น จะมีอยู่ 3 ขา นั่นคือ ขา EN ทำหน้าที่สั่งการอีนาเบิลโมดูล LCD ส่วนขา RW ทำหน้าที่เลือกการอ่านหรือเขียนข้อมูลลงยังโมดูล LCD และสุดท้าย ขา RS ทำหน้าที่เลือกชนิดของข้อมูลหรือคำสั่งที่จะเขียนให้กับโมดูล LCD ในด้านการแสดงผลของ LCD จะมีตัวต้านทาน R6 ทำหน้าที่จำกัดความสว่างของหน้าจอไว้ หากว่าใช้ R6 ที่มีค่าความต้านทานน้อยๆ หน้าจอก็จะสว่างมาก หากใช้ค่าความต้านทานมาก ความสว่างก็จะน้อยลง

ส่วนของหน่วยความจำ EEPROM ใช้ไอซีเบอร์ 24LC02 (IC3) ใช้เป็นส่วนเก็บค่าอุณหภูมิและค่าความชื้นที่ตั้งไว้เพื่อป้องกันไม่ให้ค่าที่เก็บไว้หายไปหลังจากปิดเครื่อง ดังนั้นจึงมั่นใจได้ว่าเมื่อปิดเครื่องขึ้นมา ค่าต่างๆที่เคยเซตไว้จะยังคงอยู่ โดยไอซี 24LC02 ที่ใช้ในวงจรนี้เป็นไอซีที่ใช้วิธีการรับส่งข้อมูลแบบอนุกรม 8 บิต ด้วยการเชื่อมต่อผ่านระบบบัส I2C โดยใช้ขาสัญญาณเพียง 2 เส้น คือ ขา SCL (ขาสัญญาณนาฬิกา) ซึ่งในที่นี้ถูกต่อเข้ากับพอร์ต P1.0 ส่วนขา SDA (ขาสัญญาณข้อมูล) จะต่อเข้ากับพอร์ต P3.0 ซึ่งข้อดีของการใช้หน่วยความจำแบบ EEPROM ในการเก็บรักษาข้อมูลคือ ไม่ต้องใช้แบตเตอรี่แบ็กอัพข้อมูลเหมือนหน่วยความจำ RAM

ในส่วนของเซ็นเซอร์ตรวจจับอุณหภูมิและความชื้น SHT15 (IC4) มีขาใช้งานอยู่ด้วยกัน 4 ขา เหมือนกับ EEPROM การต่อขาใช้งานของ SHT15 จึงสามารถต่อขนานเข้ากับขาของ 24LC02 ได้เลย เนื่องจากอุปกรณ์สองตัวนี้ทำงานด้วยระบบบัส I2C เหมือนกันทั้งคู่ ส่วนสวิตช์กดติดปล่อยดับ S1, S2 และ S3 มีไว้สำหรับตั้งค่าความชื้นที่ต้องการ โดยที่ไมโครคอนโทรลเลอร์จะรับลอจิก '1' ที่ได้มาจากการกดสวิตช์แต่ละตัว เข้ามายังพอร์ต P3.5, P3.4 และ P3.2 เพื่อนำไปประมวลผลต่อไป

ส่วนสัญญาณเอาท์พุตที่นำไปควบคุมอุปกรณ์ที่ใช้ควบคุมความชื้นนั้น จะต่อมาจากพอร์ต P3.7 ของไมโครคอนโทรลเลอร์ส่งไปป้อนให้กับรีเลย์ โดยที่รีเลย์ (RY1) ทำหน้าที่เป็นตัวตัดต่อไฟเลี้ยงของอุปกรณ์ควบคุมความชื้น โดยมีทรานซิสเตอร์ Q1 เป็นตัวขับให้รีเลย์ทำงาน

เมื่อพอร์ต P3.7 ส่งลอจิก '1' ไปไบแอสขาเบสของทรานซิสเตอร์ Q1 จะทำให้หลอดไฟสีเขียวสว่าง และรีเลย์ RY1 ก็จะทำงาน



รูปที่ 3.5 Block Diagram ของตัวตรวจจับความชื้นและอุณหภูมิ SHT15

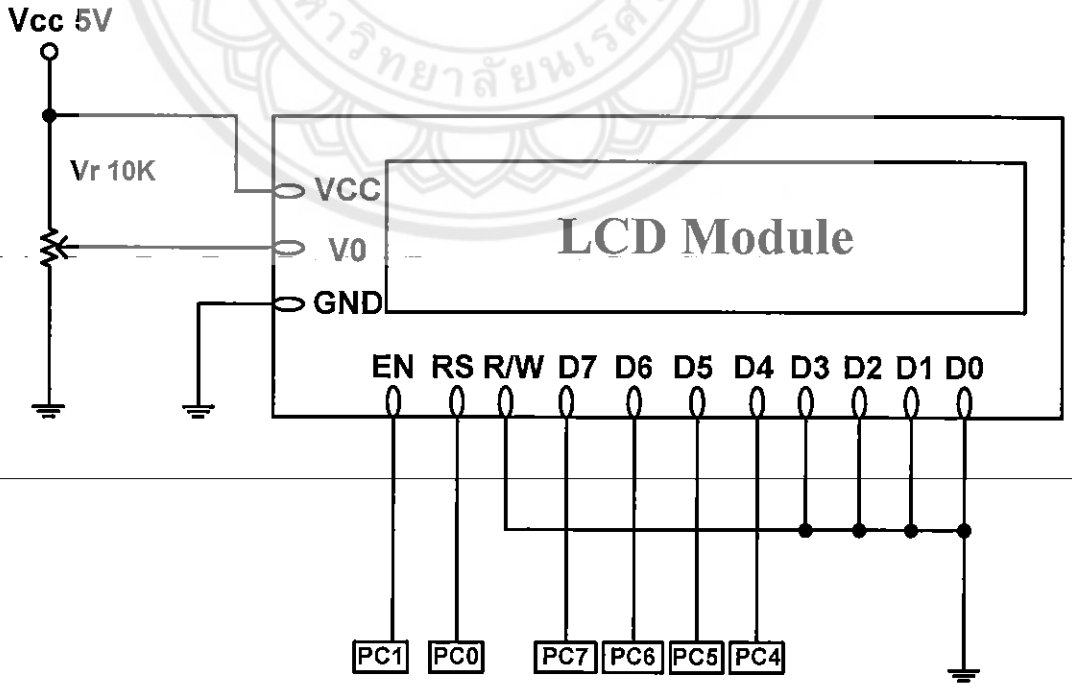
ในส่วนของการเชื่อมต่อข้อมูลระหว่างตัวตรวจจับความชื้นกับอุณหภูมิ SHT-15 กับ Microcontroller จะเป็นการเชื่อมต่อแบบอนุกรม (Serial Interface) ซึ่งตัวตรวจจับนี้จะทำการวัดค่าความชื้นด้วยตัวตรวจจับความชื้นและวัดค่าอุณหภูมิด้วยตัวตรวจจับอุณหภูมิ ซึ่งสองส่วนนี้จะอยู่บน SHT-15 ข้อมูลที่ได้จะถูกส่งออกมาในระบบดิจิตอล ซึ่งจะมีหน่วยที่ใช้ในการแปลงจากสัญญาณอนาลอกเป็นสัญญาณดิจิตอล 14 บิต เราจึงต้องทำการออกแบบโปรแกรมที่สามารถรับส่งข้อมูลระหว่างตัวตรวจจับและ Microcontroller ได้

3.2.2 LCD Module

การติดต่อกับโมดูล LCD 16 ตัวอักษร 2 บรรทัดมีอยู่ด้วยกัน 2 แบบคือ 8 บิต และ 4 บิต โดยโครงงานนี้ใช้วิธีการติดต่อกับจอแสดงผลเป็นแบบ 4 บิต โดยการต่อเป็นไปตามนี้

ตารางที่ 3.1 ตารางชื่อและหน้าที่ของสัญญาณต่างๆของ LCD

ขาที่	ชื่อขา	หน้าที่การทำงาน
1	GND	ต่อกับกราวด์ของวงจร
2	+Vcc	ต่อกับไฟเลี้ยง 5 V
3	Vo	ปรับแรงดันเพื่อปรับแสงสว่าง
4	Rs	เป็นขาเลือกการติดต่อกับรีจิสเตอร์คำสั่งหรือรีจิสเตอร์ข้อมูล 0 : จะติดต่อกับรีจิสเตอร์คำสั่ง 1: จะติดต่อกับรีจิสเตอร์ข้อมูลเพื่อนำข้อมูลไปแสดงผล
5	R/W	ต่อลงกราวด์เพราะต้องการเขียนข้อมูลแสดงผลอย่างเดียว
6	E	เป็นขา Enable การทำงานให้ LCD
7-10	Do-D3	ต่อร่วมกันลงกราวด์เพราะใช้ 4 บิตหลัง
11-14	D4-D7	เป็นขาข้อมูล4บิต



รูปที่ 3.6 การต่อใช้งานจอแสดงผล LCD

• การเขียนข้อมูลคำสั่งไปยังจอแสดงผล

1. ทำให้ขา RS เป็น "0" เพื่อแจ้งให้จอแสดงผลทราบว่าข้อมูลที่ขา DATA เป็นข้อมูลคำสั่ง
2. ส่งข้อมูล (Data) 4 บิตบน (บิต7-บิต4) ที่ต้องการไปยังขา Data ทั้ง 4 เส้น
3. ส่งพัลส์ Enable ไปยังขา E
4. ส่งข้อมูลคำสั่ง 4 บิตล่าง (บิต3-บิต0) ที่ต้องการไปยังขา Data ทั้ง 4 เส้น
5. ส่งพัลส์ Enable ไปยังขา E

• การเขียนข้อมูล (Data) ไปยังจอแสดงผล

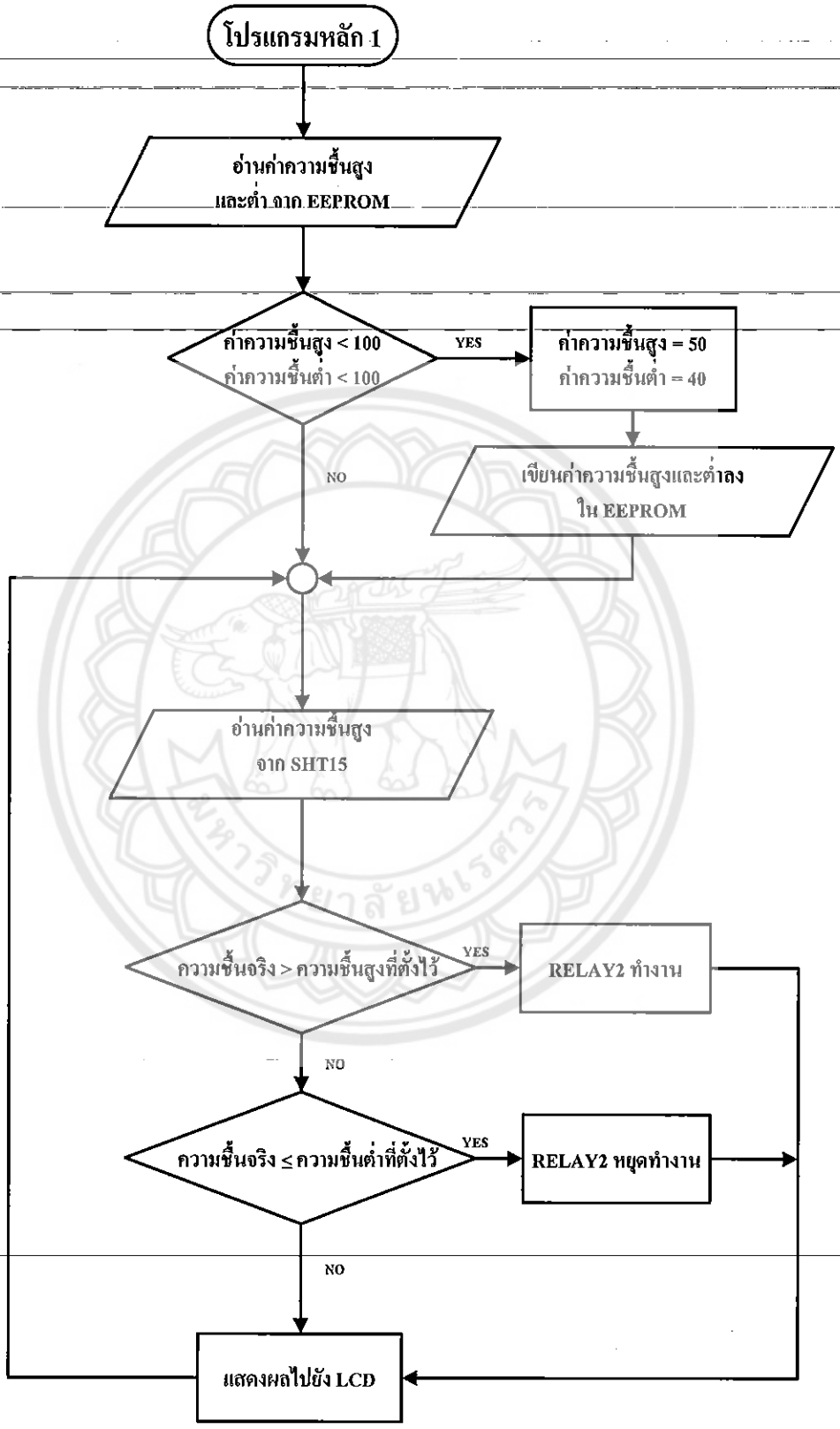
1. ทำให้ขา RS เป็น "1" เพื่อแจ้งให้จอแสดงผลทราบว่าข้อมูลที่ขา DATA เป็นข้อมูลในการแสดงผล
2. ส่งข้อมูล (Data) 4 บิตบน (บิต7-บิต4) ที่ต้องการไปยังขา Data ทั้ง 4 เส้น
3. ส่งพัลส์ Enable ไปยังขา E
4. ส่งข้อมูล (Data) 4 บิตล่าง (บิต3-บิต0) ที่ต้องการไปยังขา Data ทั้ง 4 เส้น
5. ส่งพัลส์ Enable ไปยังขา E

3.3 ขั้นตอนการปฏิบัติงานของวงจรควบคุมอุณหภูมิและความชื้น

ขั้นตอนในการปฏิบัติงานหรือโปรแกรมคำสั่งนั้นจะอธิบายอยู่ในรูปของแผนภาพการปฏิบัติงาน (Flow Chart) ซึ่งมีโปรแกรมย่อยต่างๆดังนี้

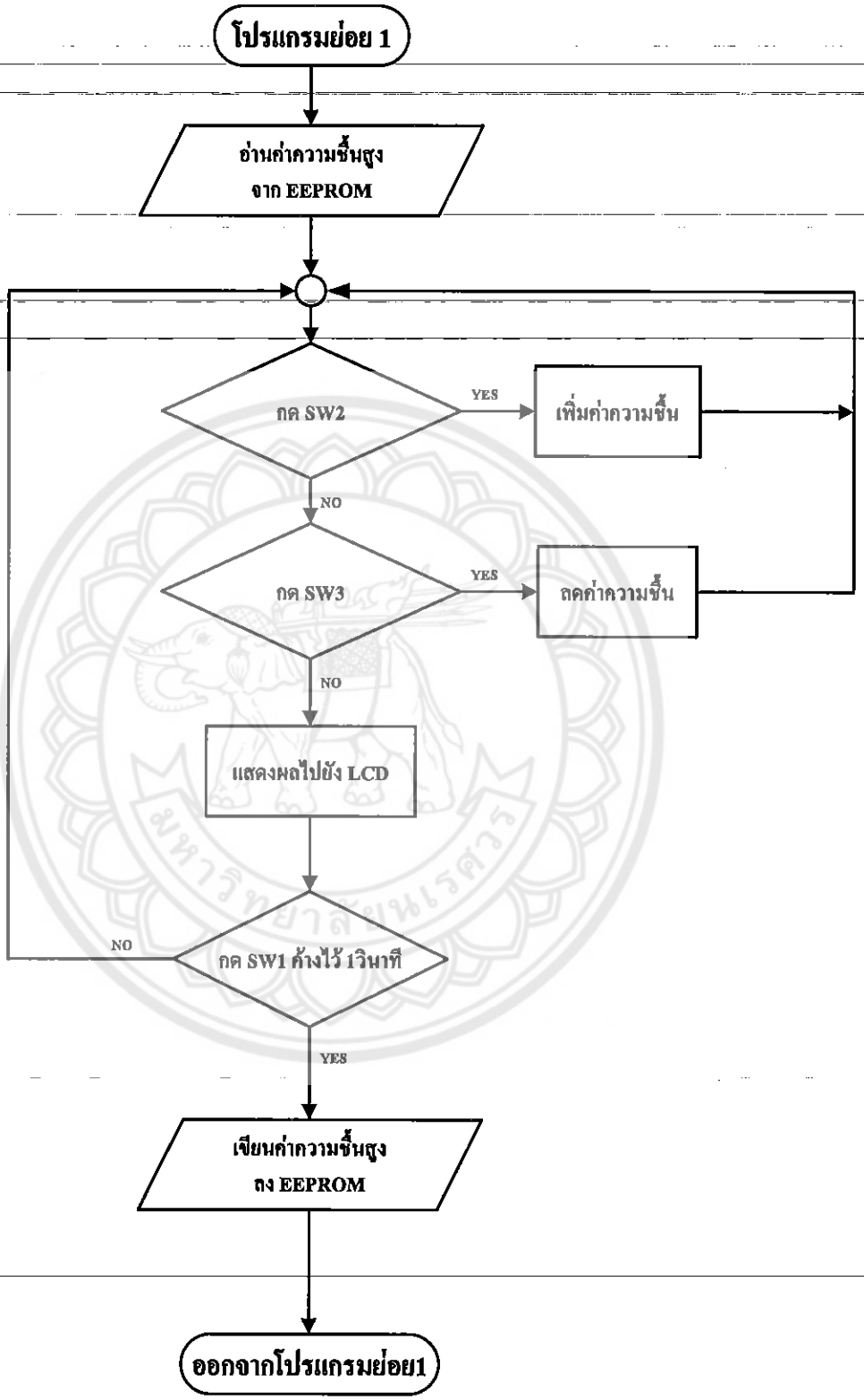
1. โปรแกรมหลัก 1 โปรแกรมหลักสำหรับควบคุมความชื้น
2. โปรแกรมย่อย 1 โปรแกรมย่อยสำหรับตั้งค่าความชื้นค่าสูง
3. โปรแกรมย่อย 2 โปรแกรมย่อยสำหรับตั้งค่าความชื้นค่าต่ำ

3.3.1 โปรแกรมหลัก 1 โปรแกรมหลักสำหรับควบคุมความชื้น



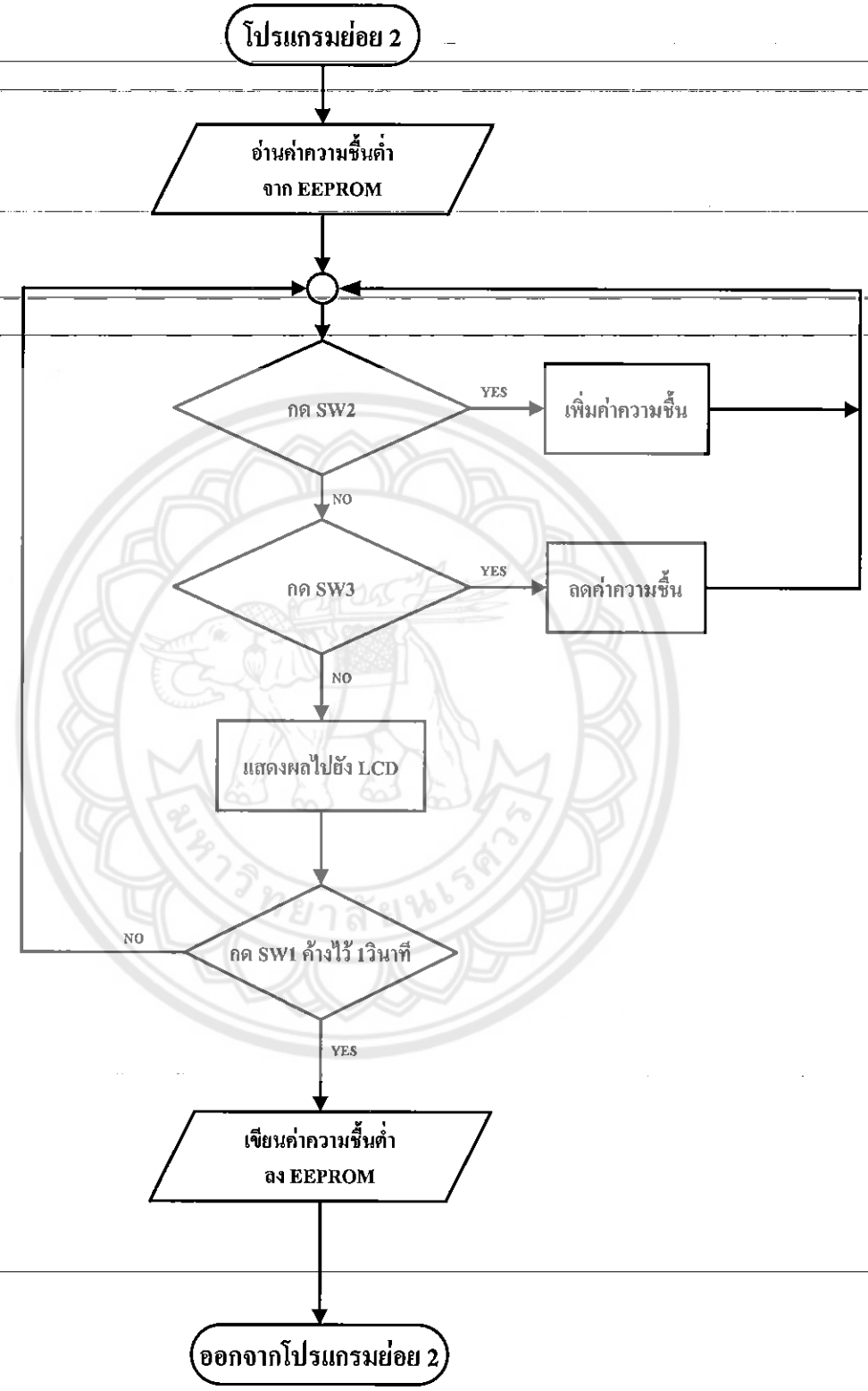
รูปที่ 3.7 แผนภาพโปรแกรมหลัก 1

3.3.2 โปรแกรมย่อยสำหรับตั้งค่าความชื้นค่าสูง



รูปที่ 3.8 แผนภาพโปรแกรมย่อย 1

3.3.3 โปรแกรมย่อยสำหรับตั้งค่าความชื้นค่าต่ำ

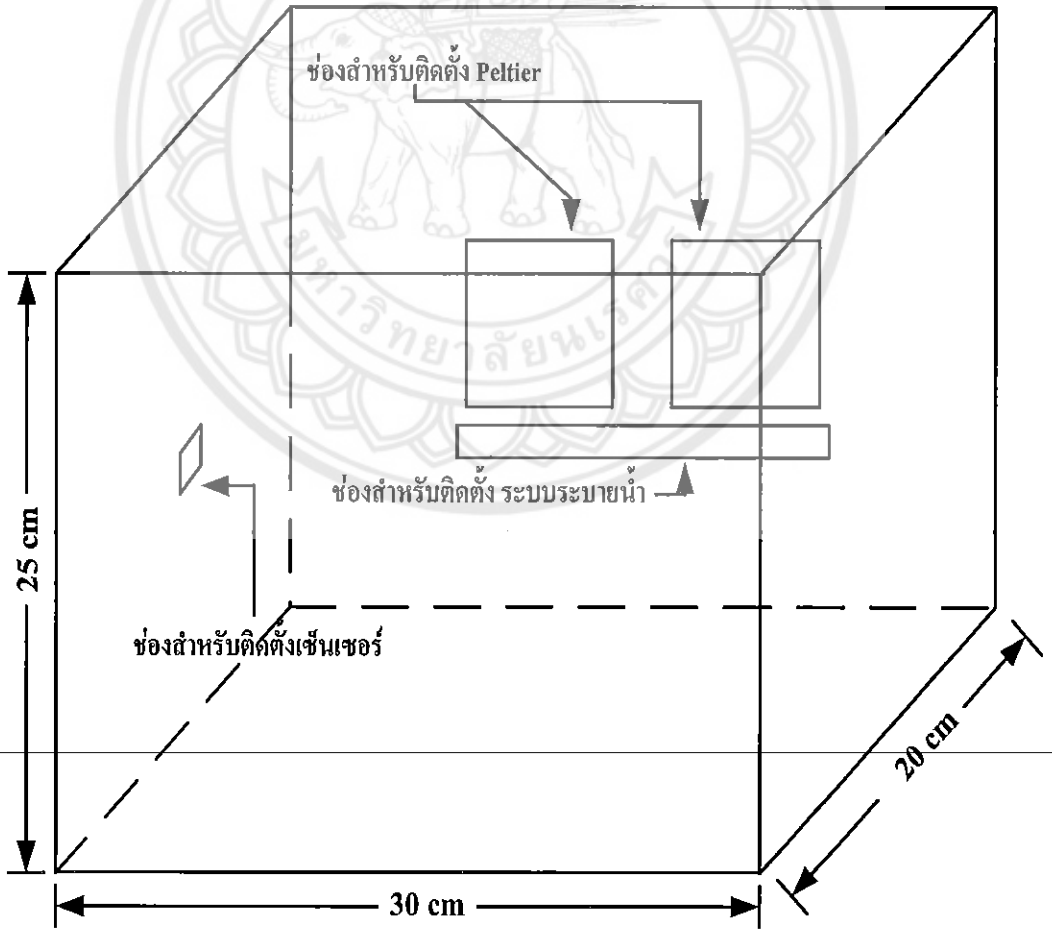


รูปที่ 3.9 แผนภาพโปรแกรมย่อย 2

3.4 การออกแบบตู้ควบคุมความชื้น (Chamber)

ในการออกแบบตู้ที่ใช้ในการควบคุมความชื้นนั้น จะประกอบไปด้วยโครงสร้างที่สำคัญ 2 ส่วนด้วยกันดังต่อไปนี้

3.4.1 โครงสร้างตู้ควบคุมความชื้น ใช้วัสดุพลาสติกแข็งทำเป็นห้องสี่เหลี่ยม ซึ่งแผ่นพลาสติกแข็งที่ใช้นั้นมีความหนาประมาณ 3-5 mm โดยที่ด้านหลังเจาะเว้นช่องสี่เหลี่ยมขนาด 7×7 cmไว้จำนวน 2 ช่องสำหรับ Heat Sink ของทางด้านเพลทเย็น เพื่อใช้ในการควบคุมความชื้นภายในตู้ควบคุมความชื้น และช่องขนาด 15×2 cm เป็นช่องเพื่อใช้ติดตั้งระบบระบายน้ำเพื่อระบายน้ำที่เกิดจากการควบแน่นของอากาศที่ไหลลงมาจากเพลทเย็น ดังรูป

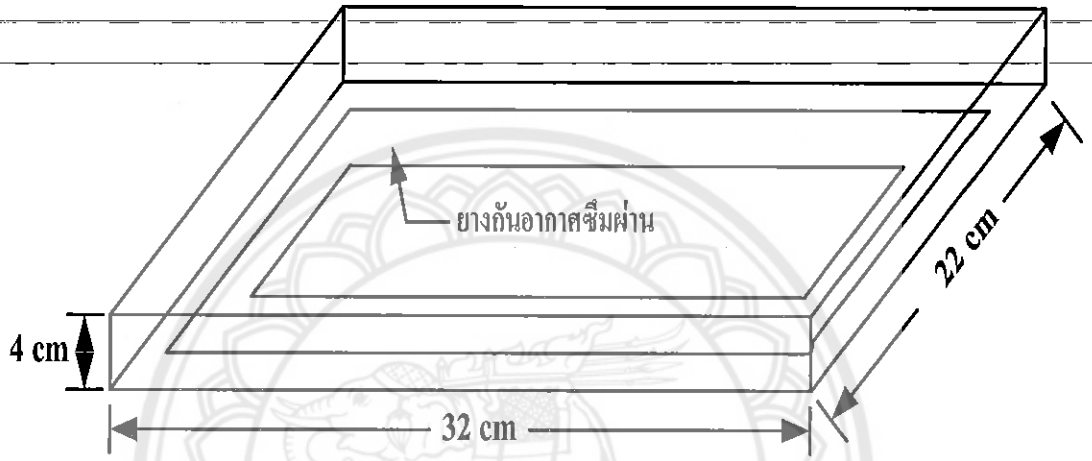


รูปที่ 3.10 โครงสร้างของตู้ควบคุมความชื้น

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

1499836
2/5.
ท674ฎ
2550

3.4.2 ฝาปิดตู้ ใช้วัสดุพลาสติกแข็งทำเป็นตัวฝา ส่วนขอบด้านล่างติดยางกันอากาศซึมผ่านออกจากตู้ควบคุมความชื้น ดังรูป



รูปที่ 3.11 โครงสร้างฝาปิดตู้ควบคุมความชื้น

บทที่ 4
การปฏิบัติและการทดลอง

4.1 การสร้างวงจรควบคุมความชื้น

ทำการสร้างวงจรควบคุมความชื้น ซึ่งได้มาจากการศึกษาการควบคุมความชื้น แล้วนำมาสร้างเป็นวงจรควบคุมความชื้นแบบดิจิตอล

รูปที่ 4.1 วงจรควบคุมความชื้นที่ประกอบเสร็จแล้ว

โดยทำการบัดกรีอุปกรณ์ทั้งหมดที่เป็นส่วนประกอบของวงจร ลงบนแผ่นปริ๊นท์อเนกประสงค์ แล้วทำการทดสอบวงจร ซึ่งได้ผลการทดสอบดังนี้

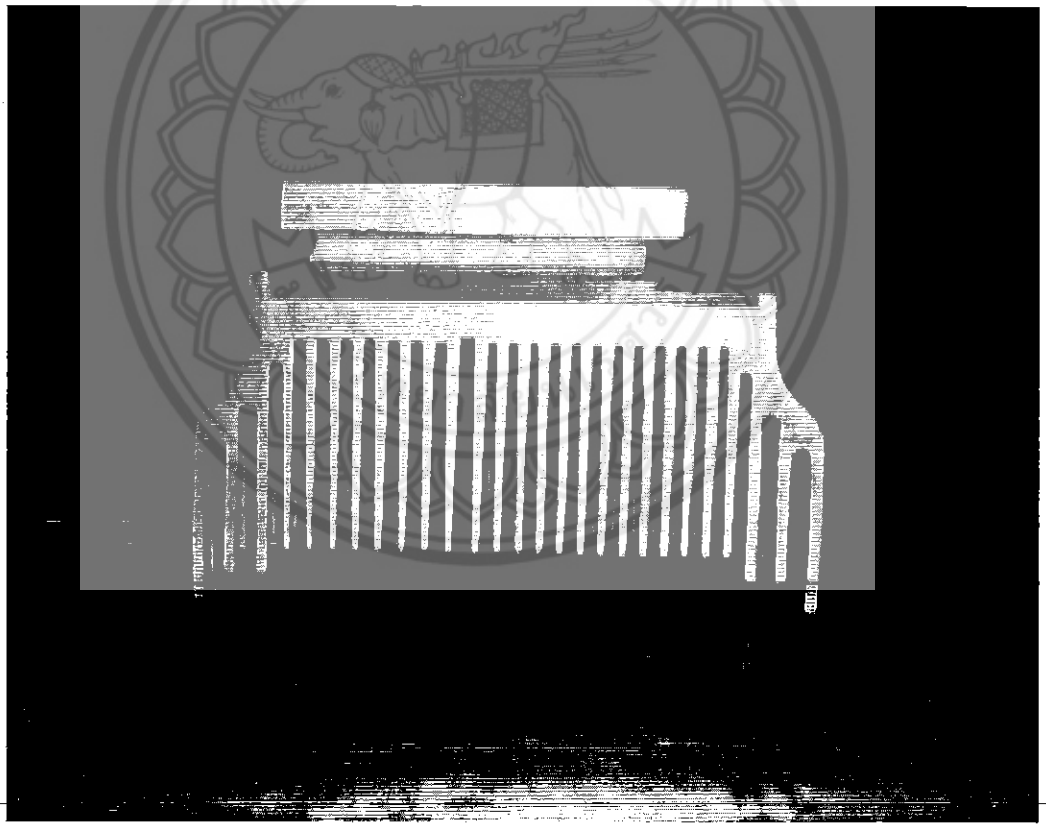
- เมื่อจ่ายไฟเลี้ยง 12 VDC ให้กับวงจร ที่จอแสดงผล LCD จะแสดง "Humidity" อยู่ประมาณ 3 วินาที แล้วจึงอุณหภูมิกับความชื้น ซึ่งค่าอุณหภูมิและค่าความชื้นนั้นคือค่าปกติที่วัดได้โดยรอบของเซ็นเซอร์

- กดปุ่ม SET ค้างไว้ประมาณ 1 วินาที จะเป็นการเซตค่าความชื้นต่ำ
- กดปุ่ม UP หรือ DOWN เพื่อตั้งค่าความชื้นต่ำตามที่ต้องการ
- กดปุ่ม SET อีก 1 ครั้ง ที่ LCD จะแสดงค่าความชื้นสูงที่ต้องการเซต
- กดปุ่ม UP หรือ DOWN เพื่อตั้งค่าความชื้นสูงตามที่ต้องการ
- กดปุ่ม SET อีก 1 ครั้ง จะกลับไปแสดงค่าอุณหภูมิและค่าความชื้นตามปกติ

สังเกตการเปลี่ยนแปลงของความชื้น ซึ่งถ้าหากความชื้นสูงกว่าที่ SET เอาไว้ LED จะเปลี่ยนจากสีแดงกลายเป็นสีเขียว พร้อมทั้ง Relay จะเปลี่ยนสถานะจาก NO เป็น NC แต่ถ้าความชื้นต่ำกว่าที่ SET เอาไว้ LED จะเปลี่ยนจากสีเขียวมาเป็นสีแดงตามเดิม พร้อมทั้ง Relay จะเปลี่ยนสถานะจาก NC เป็น NO แสดงว่าวงจรทำงานได้สมบูรณ์แบบ

4.2 การติดตั้งเพลเทียร์

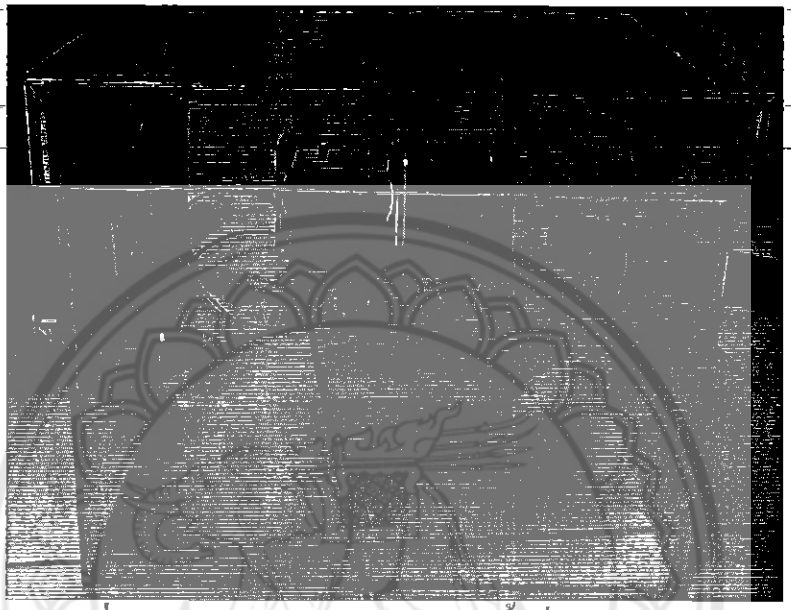
การติดตั้งเพลเทียร์นั้น จะติดตั้งโดยให้ด้านร้อนติดกับ Heat Sink ใช้พัดลมระบายความร้อนให้ Heat Sink ด้วย ส่วนด้านเย็นของเพลเทียร์จะติดกับแผ่นอลูมิเนียม เพื่อเป็นตัวจับไอน้ำในอากาศให้ควบแน่นเป็นหยดน้ำ ซึ่งทำให้ความชื้นลดลงได้



รูปที่ 4.2 การติดตั้งเพลเทียร์เข้ากับ Heat Sink

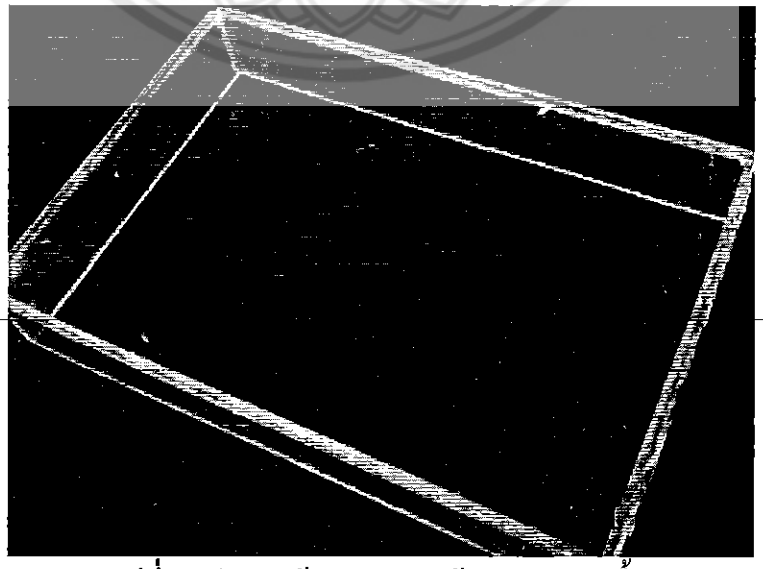
4.3 โครงสร้างของตู้ควบคุมความชื้น

หลังจากที่ได้ออกแบบตู้ควบคุมความชื้นมาแล้วนั้น จึงได้ทำการตัดแผ่นพลาสติกแข็ง ซึ่งเป็นวัสดุที่นำมาใช้สร้างตู้ควบคุมความชื้น ให้ได้ขนาดตามที่ได้ออกแบบไว้ แล้วนำมาประกอบเข้าด้วยกัน โดยที่ด้านข้างจะเจาะรูไว้เพื่อติดตั้งเซ็นเซอร์วัดความชื้น และทางด้านหลังของตู้จะทำช่องไว้เพื่อใช้ติดตั้งเพลเทียร์และชุดระบายความร้อน



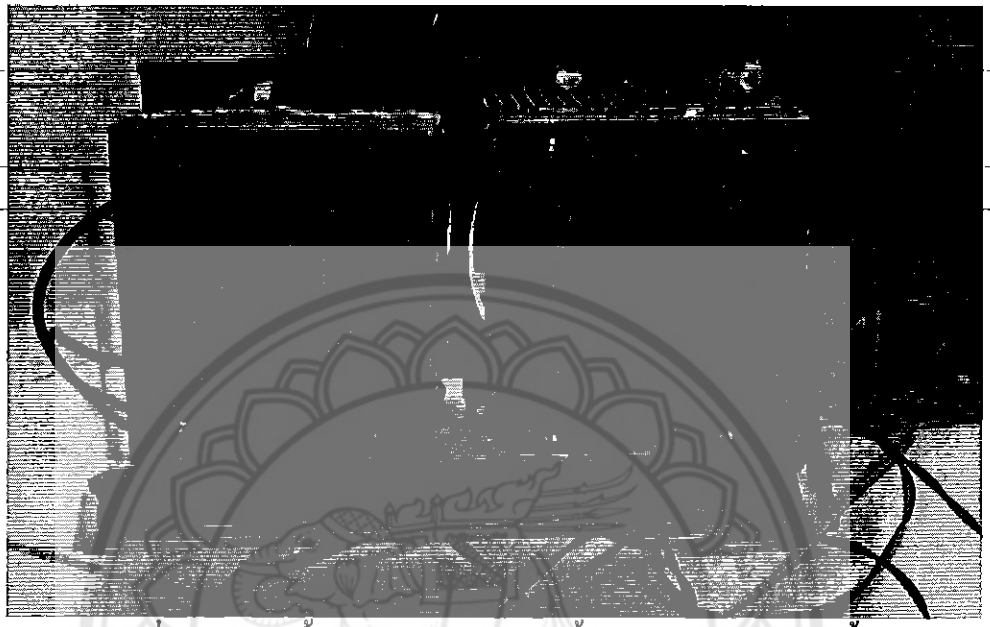
รูปที่ 4.3 โครงสร้างของตู้ควบคุมความชื้นที่ทำการประกอบ

ในส่วนของฝาตู้ก็ใช้พลาสติกแข็งชนิดเดียวกัน แล้วทำการตัดให้ได้ขนาดพอดีกับตัวตู้ เพื่อจะได้ไม่มีอากาศผ่านเข้าไปได้



รูปที่ 4.4 โครงสร้างของฝาปิดตู้ควบคุมความชื้น

เมื่อทำโครงสร้างของตู้เรียบร้อยแล้ว จากนั้นจึงทำการประกอบอุปกรณ์ที่ใช้ในการควบคุมความชื้น ซึ่งประกอบด้วย เพลเทียร์ แผ่นอลูมิเนียม Heat Sink และพัดลมระบายความร้อน โดยจะใช้น็อตยึดเพื่อให้มีความมั่นคง เมื่อประกอบเสร็จแล้ว จึงทำการติดตั้งอุปกรณ์ควบุมความชื้นนี้เข้ากับตู้ควบคุมความชื้นทางด้านหลังที่ทำช่องสำหรับติดตั้งไว้แล้วไว้แล้ว



รูปที่ 4.5 การติดตั้งอุปกรณ์ควบคุมความชื้นเข้ากับตู้ควบคุมความชื้น

รูปที่ 4.6 ตู้ควบคุมความชื้นเมื่อประกอบเสร็จแล้ว

4.4 การทดลอง

4.4.1 นำตู้ควบคุมความชื้นที่ประกอบเสร็จสมบูรณ์แล้วต่อเข้ากับสวิทชิ่งเพาเวอร์ซัพพลาย

4.4.2 เปิดสวิทช์ที่สวิทชิ่งเพาเวอร์ซัพพลายเพื่อจ่ายไฟเข้าเลี้ยงวงจรภายในตัวตู้ควบคุมความชื้น

4.4.3 หลังจากที่จ่ายไฟเลี้ยงให้กับตัวตู้ควบคุมความชื้นแล้ว ไฟแสดงผลของวงจรควบคุมความชื้นแบบดิจิตอล จะแสดงค่าอุณหภูมิและความชื้นปกติภายในตู้ควบคุมความชื้น

4.4.4 จากนั้นทำการตั้งค่าความชื้นโดย

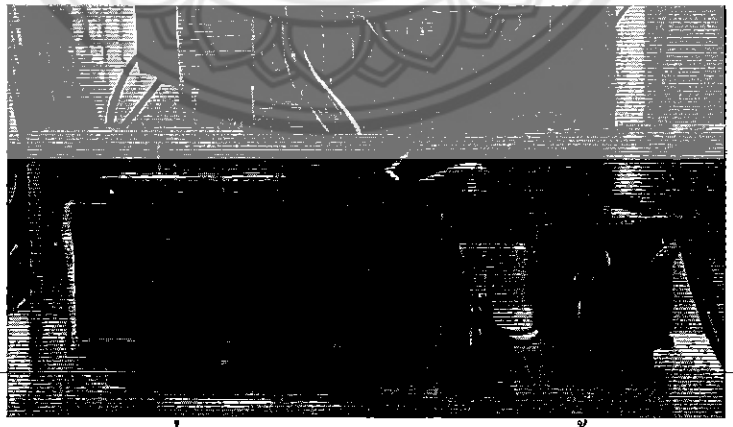
- กดปุ่ม SET ค้างไว้ประมาณ 1 วินาที จะเป็นการเซตค่าความชื้นต่ำ
- กดปุ่ม UP หรือ DOWN เพื่อตั้งค่าความชื้นต่ำตามที่ต้องการ
- กดปุ่ม SET อีก 1 ครั้ง ที่ LCD จะแสดงค่าความชื้นสูงที่ต้องการเซต
- กดปุ่ม UP หรือ DOWN เพื่อตั้งค่าความชื้นสูงตามที่ต้องการ
- กดปุ่ม SET อีก 1 ครั้ง จะกลับไปแสดงค่าอุณหภูมิและค่าความชื้นตามปกติ

4.4.5 ทำการทดลอง 3 ช่วง ณ อุณหภูมิเริ่มต้น 29 °C และความชื้นเริ่มต้น 67%

- การทดลองช่วงที่ 1 ความชื้น 50-60%
- การทดลองช่วงที่ 2 ความชื้น 40-50%
- การทดลองช่วงที่ 3 ความชื้น 30-40%

4.4.6 ทำการบันทึกผลการทดลองทุกๆ 5 นาที

4.5 ผลการทดลอง

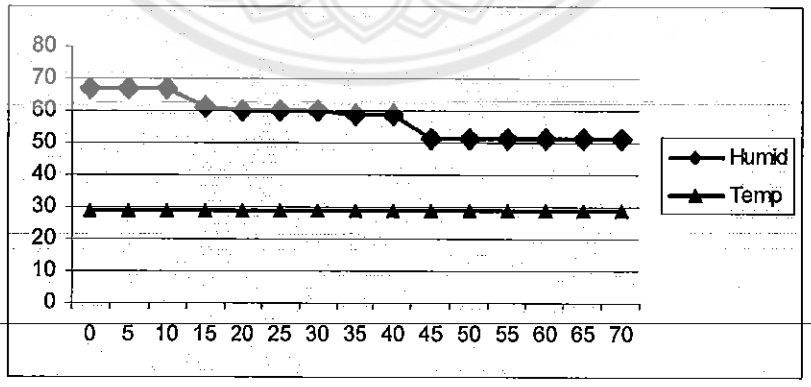


รูปที่ 4.7 การทำงานของตู้ควบคุมความชื้น

4.5.1 การทดลองช่วงที่ 1 เมื่อตั้งค่าความชื้นต่ำที่ 50% และค่าความชื้นสูงที่ 60%

ตาราง 4.1 ผลการทดลองเมื่อตั้งค่าความชื้น 50-60%

เวลา (นาที)	ความชื้น (%)	อุณหภูมิ (°C)
0	67	29
5	67	29
10	67	29
15	61	29
20	60	29
25	60	29
30	60	29
35	59	29
40	59	29
45	51	29
50	51	29
55	51	29
60	51	29
65	51	29
70	51	29

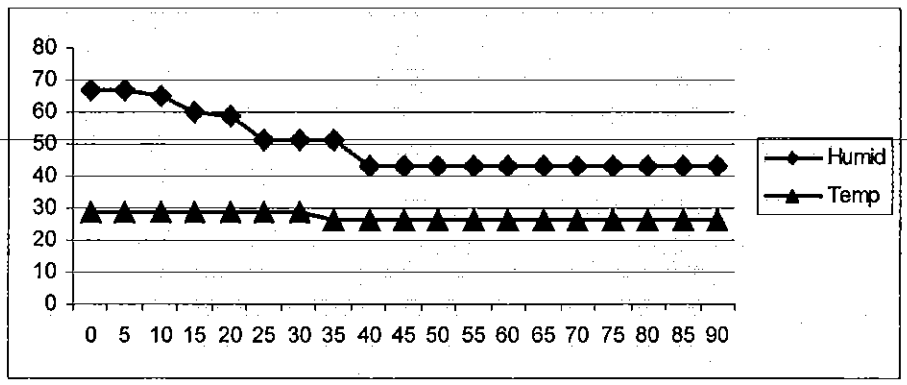


รูปที่ 4.8 กราฟผลการทดลองช่วงที่ 1

4.5.2 การทดลองช่วงที่ 2 เมื่อตั้งค่าความชื้นต่ำที่ 40% และค่าความชื้นสูงที่ 50%

ตาราง 4.2 ผลการทดลองเมื่อตั้งค่าความชื้น 40-50%

เวลา (นาที)	ความชื้น (%)	อุณหภูมิ (°C)
0	67	29
5	67	29
10	61	29
15	60	29
20	59	29
25	51	29
30	51	29
35	51	26
40	43	26
45	43	26
50	43	26
55	43	26
60	43	26
65	43	26
70	43	26
75	43	26
80	43	26
85	43	26
90	43	26

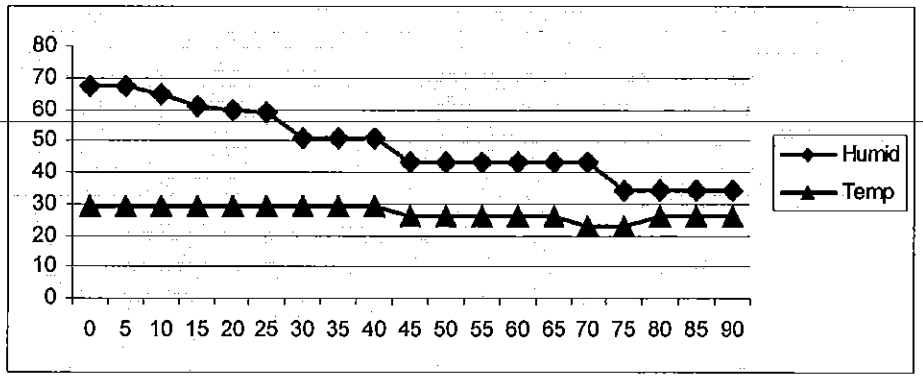


รูปที่ 4.9 กราฟผลการทดลองช่วงที่ 2

4.5.3 การทดลองช่วงที่ 3 เมื่อตั้งค่าความชื้นต่ำที่ 30% และค่าความชื้นสูงที่ 40%

ตาราง 4.3 ผลการทดลองเมื่อตั้งค่าความชื้น 30-40%

เวลา (นาที)	ความชื้น (%)	อุณหภูมิ (°C)
0	67	29
5	67	29
10	65	29
15	61	29
20	60	29
25	59	29
30	51	29
35	51	29
40	51	29
45	43	26
50	43	26
55	43	26
60	43	26
65	43	26
70	43	23
75	34	23
80	34	26
85	34	26
90	34	26



รูปที่ 4.10 กราฟผลการทดลองช่วงที่ 3

บทที่ 5
สรุปผลการดำเนินงาน

จากการศึกษาคุณสมบัติของเพลเทียร์ ทำให้ทราบหลักการทำงานของเพลเทียร์ ซึ่งสามารถนำมาประยุกต์ทำเป็นอุปกรณ์สำหรับใช้งานด้านความเย็นและความร้อนได้หลายอย่าง โดยโครงงานนี้ได้นำเสนอเกี่ยวกับการใช้เพลเทียร์ในการควบคุมความชื้นภายในตู้ควบคุมความชื้น

ซึ่งโครงงานนี้สามารถควบคุมความชื้นได้ต่ำสุดที่ 34%RH ณ อุณหภูมิ 23-26 °C ซึ่งค่าความชื้นระดับนี้ สามารถเก็บรักษาอุปกรณ์หรือเครื่องมือต่างๆได้มากมาย อีกทั้งยังสามารถนำไปประยุกต์ใช้กับตู้ที่มีขนาดใหญ่ได้

ในส่วนของแหล่งจ่ายกระแสไฟฟ้าที่ป้อนกระแสให้กับ เพลเทียร์, วงจรควบคุมและส่วนอื่นๆนั้น ในช่วงแรกได้ใช้หม้อแปลงของวงจรเรียงกระแส ทำให้เกิดปัญหาคือ กระแสจากหม้อแปลงไม่พอสำหรับเพลเทียร์ ทำให้เพลเทียร์ไม่ทำงาน เพราะฉะนั้นจึงได้เลือกใช้ สวิทชิ่งเพาเวอร์ซัพพลาย ของคอมพิวเตอร์ ขนาด 450 วัตต์ ซึ่งสามารถจ่ายกระแสได้สูงเหมาะสมกับโครงงานนี้เป็นอย่างมากเมื่อเทียบกับการใช้หม้อแปลงของวงจรเรียงกระแส

อุปสรรคในการทดลองคือ ขณะที่เพลเทียร์ ทำหน้าที่สร้างความเย็นนั้นจะส่งผลกระทบต่อด้านความร้อนมาก กล่าวคือ ถ้าเราไม่สามรถระบายความร้อนทางด้านเพลตร้อนของเพลเทียร์ ได้ทันจะทำให้ตัวเพลเทียร์ทำความเย็นได้น้อยลงเช่นกัน ทำให้ความชื้นภายในตัวถังของตู้ไม่ลดลงตามที่คาดการณ์เอาไว้ นอกจากนั้นแล้ว สวิทชิ่งเพาเวอร์ซัพพลายที่ใช้ก็ยังไม่สามารถที่จะจ่ายกระแสไฟฟ้าได้ตามที่เราต้องการทำให้วงจรทำงานได้ไม่เต็มประสิทธิภาพ ส่งผลทำให้ประสิทธิภาพในการทำความเย็นลดลงด้วย และในระหว่างการทำการทดลองนั้นก็ได้เกิดความเสียหายขึ้นกับเพลทียร์ตัวหนึ่งด้วย ทำให้ผลการทดลองในช่วงแรกเกิดความคลาดเคลื่อนพอสมควร จึงจำเป็นต้องหาเพลเทียร์ตัวใหม่มาทดแทน เพื่อให้การทดลองเป็นไปอย่างมีประสิทธิภาพ และได้ผลการทดลองที่แม่นยำ

เอกสารอ้างอิง

[1] สมโภชน์ อุทัยวิวัฒน์กุล. **หลักการทำงานของอุปกรณ์เพลเทียร์ในการสร้างความเย็น**. พิมพ์ครั้งที่1. 2541

[2] รศ.ธีรวัฒน์ ประกอบผล. **ภาษาแอสเซมบลีสำหรับ MCS-51**. พิมพ์ครั้งที่ 3. กรุงเทพฯ. สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น). 2546

[3] ขจร อนุดิตย์. **การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ MCS-51 ด้วยภาษา C.** พิมพ์ครั้งที่1. 2550

[4] เจริญพร เลิศสถิตธนกร และ วิชาญ ศรีสุวรรณ์. **"การศึกษาเชิงทดลองของฝาเพดานทำความเย็นแบบเทอร์โมอิเล็กตริก".** การประชุมวิชาการเครือข่ายวิศวกรรมเครื่องกลแห่งประเทศไทย. ครั้งที่ 18. 18-20 ตุลาคม 2547.

[5] เจนศักดิ์ เอกบูรณะวัฒน์ และ สมิต อินทร์ศิริพงษ์. **"ตู้น้ำเย็นเทอร์โมอิเล็กตริก".** การประชุมเชิงวิชาการเครือข่ายพลังงานแห่งประเทศไทย. ครั้งที่ 1. 11-13 พฤษภาคม 2548.

[6] http://www.hebeiltd.com.cn/peltier.datasheet/TEC1-12710.pdf

[7] http://www.sensirion.com/en/01_humidity_sensors/03_humidity_sensor_sht15.htm

[8] http://www.thaimicrotron.com/Sensor/SHT1x.htm

[9] http://www.melcor.com/tec.html

[10] HuiTong Dry-Cabinet Instruction Manual

ภาคผนวก

ภาคผนวก ก
อุปกรณ์ที่ใช้ในวงจรการควบคุม

1.ตัวต้านทานขนาด $\frac{1}{4}$ วัตต์ ±5%

RN1 ตัวต้านทานเน็ตเวิร์ก 9 ขา 10kΩ	1 ตัว
R2, R3, R4, R5 10kΩ	4 ตัว
R6 1kΩ	1 ตัว
R7, R9 4.7kΩ	2 ตัว
R8, R10 390Ω	2 ตัว

2. ตัวเก็บประจุ

C1 - 1μF 16V อิเล็กทรอไลต์	1 ตัว
C2 - 10μF 16V อิเล็กทรอไลต์	1 ตัว
C3 - 1000μF 16V อิเล็กทรอไลต์	1 ตัว
C4, C5 – 22pF เซรามิก	2 ตัว

3. อุปกรณ์สารกึ่งตัวนำ

IC1 – ไอซีเบอร์ AT89C2051 (ที่โปรแกรมแล้ว)	1 ตัว
IC2 – ไอซีเบอร์ 7805	1 ตัว
IC3 – ไอซีเบอร์ 24LC02 (แบบเซอร์เฟสเมาต์)	1 ตัว
IC4 – SHT15	1 ตัว
Q1, Q2 – ทรานซิสเตอร์เบอร์ BC547	2 ตัว
D1, D2, D3, D4 – ไดโอดเบอร์ 1N4148	4 ตัว
D5, D6 – ไดโอดเบอร์ 1N4001	1 ตัว
LED1 – สีแดง ขนาด 3 มม.	1 ตัว
LED2 – สีเขียว ขนาด 3 มม.	1 ตัว
LCD ขนาด 16 ตัวอักษร 1 บรรทัด	1 ตัว

4. อุปกรณ์อื่นๆ

S1, S2, S3 – สวิตช์กดปิดปล่อยดับขนาดเล็ก	3 ตัว
XTAL1 – Crystal 12 MHz.	1 ตัว

RY1 – Relay 12 V 1 contact	2 ตัว
T1 – หม้อแปลง 12 VDC	1 ตัว
ซ็อกเก็ตไอซี 20 ขา	1 ตัว
ซ็อกเก็ตไอซี 8 ขา	1 ตัว
เทอร์มินอลบล็อก 3 ขา	2 ตัว
คอนเน็กเตอร์ตัวผู้ 10 ขา	1 ตัว
คอนเน็กเตอร์ตัวเมีย 10 ขา	1 ตัว
คอนเน็กเตอร์ตัวผู้ 5 ขา	1 ตัว
คอนเน็กเตอร์ตัวเมีย 5 ขา	1 ตัว
คอนเน็กเตอร์ตัวผู้ 2 ขา	1 ตัว
คอนเน็กเตอร์ตัวเมีย 2 ขา	1 ตัว

มหาวิทยาลัยนเรศวร

ภาคผนวก ข
เซ็นเซอร์ตรวจจับความชื้น

เซ็นเซอร์ตรวจจับอุณหภูมิและความชื้น SHT-15

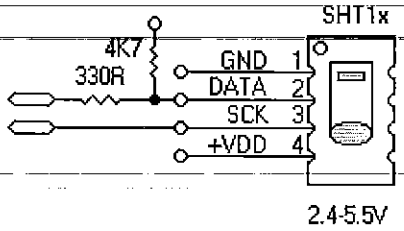
รูปที่ ก.1 เซ็นเซอร์ตรวจจับอุณหภูมิและความชื้น SHT-15

คุณสมบัติ

- อุณหภูมิและความชื้นที่วัดได้มีความสัมพันธ์กัน
- ให้ค่าเอาท์พุทเป็นดิจิตอล
- มีเสถียรภาพดีแม้จะใช้งานนาน
- ไม่ต้องต่ออุปกรณ์เสริมภายนอก
- กินพลังงานไฟฟ้าต่ำ
- มีขนาดเล็ก กะทัดรัด
- Dew point
- Fully calibrate

SHT-15 เป็น chip เดี่ยว ที่มีความสัมพันธ์กันระหว่างอุณหภูมิและความชื้น ซึ่งเป็นตัว sensor ที่ประกอบด้วยอุปกรณ์การวัดความชื้นและอุณหภูมิที่แสดงผลในระบบ digital โดยเป็น application ของ CMOS ที่ทำงานร่วมกับตัว micro machining ซึ่งทำให้เกิดความน่าเชื่อถือสูงเมื่อต่อใช้งานเป็นเวลานาน โดยที่ตัว SHT-15 ประกอบด้วยอุปกรณ์ที่เรียกว่า capacitive polymer sensing ซึ่งเป็นตัวที่ทำหน้าที่วัดความชื้นและอุณหภูมิ ซึ่ง device ตัวนี้เป็นตัวconverter จาก analog เป็น digital แบบ 14bit และตัววงจรภายในต่อกันในรูปแบบอนุกรม ซึ่งทำให้สามารถวัดคุณภาพของสัญญาณได้ดียิ่งขึ้น มีการตอบสนองด้วยเวลาที่

รวดเร็ว และ ไม่ไวต่อสัญญาณรบกวน ซึ่งตัว SHT-15 ถูกออกแบบมาให้มีขนาดเล็กและกินพลังงานไฟฟ้าต่ำ



รูปที่ ก.2 Pin Laout

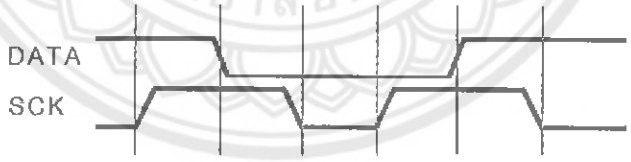
1. Pin Assignment

ตาราง ก.1 Pin Description

Pin	Name	Comment
1	GND	Ground
2	DATA	Serial data, bidirectional
3	SCK	Serial clock, input
4	VDD	Supply 2.4 – 5.5 V
	NC	Remaining pins must be left unconnected

2. ลำดับคำสั่งของ SHT15 (Command Sequence)

2.1 Transmission Start

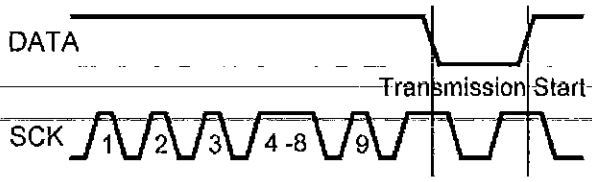


รูปที่ ก.3 Transmission start sequence

ลักษณะเงื่อนไขของสัญญาณมีลักษณะเป็น

- Data เปลี่ยนจาก 1 เป็น 0 ขณะที่ SCK ลูกแรก เป็น 1
- Data เปลี่ยนจาก 0 เป็น 1 ขณะที่ SCK ลูกที่2 เป็น 1

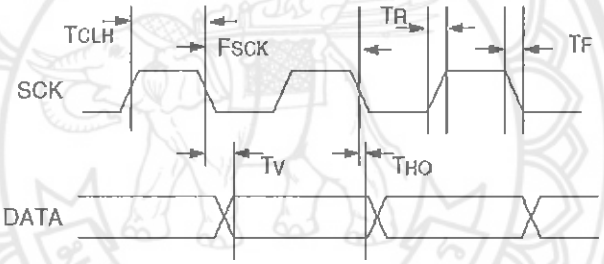
2.2 Connection reset sequence



รูปที่ ก.4 Connection reset sequence

- เมื่อขาดการติดต่อกับอุปกรณ์ ให้ส่ง SCK ไปอย่างน้อย 9 ลูก ขณะที่ Data เป็น 1 แล้วตามด้วย Transmission Start

2.3 I/O Characteristics



รูปที่ ก.5 Timing Diagram

ตาราง ก.2

	Parameter	Conditions	Min	Typ.	Max.	Unit
FSCK	SCK frequency	VDD > 4.5 V			10	MHz
		VDD < 4.5 V			1	MHz
TRFO	DATA fall time	Output load 5 pF	3.5	10	20	ns
		Output load 100 pF	30	40	200	ns
TCLH	SCK high time		100			ns
TCLL	SCK low time		100			ns
TV	DATA valid from			50		ns
THO	Output hold time		0	10		ns
TR/TF	SCK rise/fall time				200	ns

3. รายละเอียดของคำสั่ง

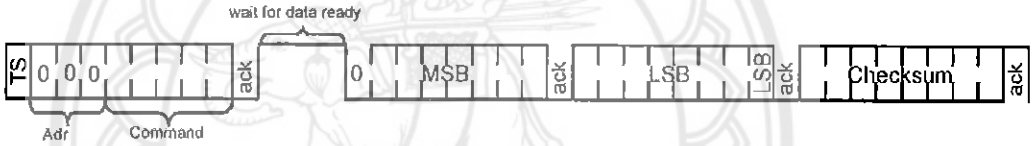
ตาราง ก.3 List of Command

Command	Code	Describtion
Reserved	0000x	Reserved
Measure Temperature	00011	Temperature measurement
Measure Humidity	00101	Humidity measurement
Status Register Read	00111	Read access to the status register (see application note)
Status Register Write	00110	Write access to the status register (see application note)
Reserved	0101x-1110x	Reserved
Soft reset	11110	resets the chip, clears the status register to default values wait 11ms before next command

3.1 การอ่านข้อมูลจากเซ็นเซอร์

ชุดคำสั่งประกอบด้วย Transmission Start + Address + Command

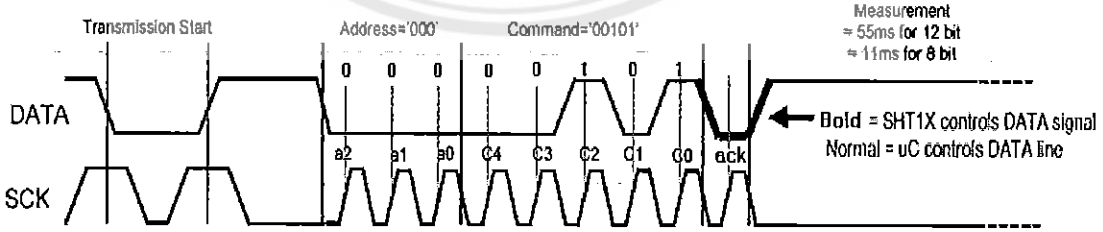
โดย Address = 000 3 + Command 5 bit



รูปที่ ก.6 การรับส่งข้อมูล

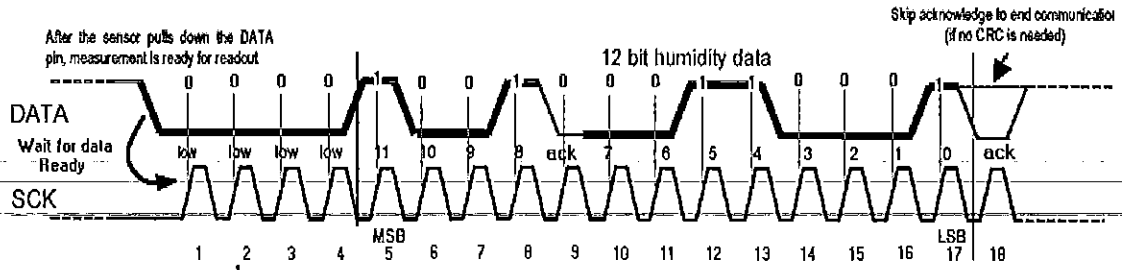
3.2 การอ่านค่าอุณหภูมและความชื้นสัมพัทธ์ (Measurement sequence T and RH)

ตัวอย่างเมื่อไมโครฯ ต้องการอ่านค่า Humidity จาก sensor ซึ่งมี address = 000 และ คำสั่ง = 00101 จะมีชุดคำสั่งดังนี้



รูปที่ ก.7 ลักษณะสัญญาณในการอ่านข้อมูลจาก Sensor

เมื่อ Sensor รับทราบคำสั่งแล้วจะส่ง acknowledge (ACK) ด้วยการดึงขา Data ลงเป็น 0 (เส้นทึบ)



รูปที่ ก.8 ลักษณะสัญญาณในการอ่านข้อมูลT/H 2 byte จาก Sensor

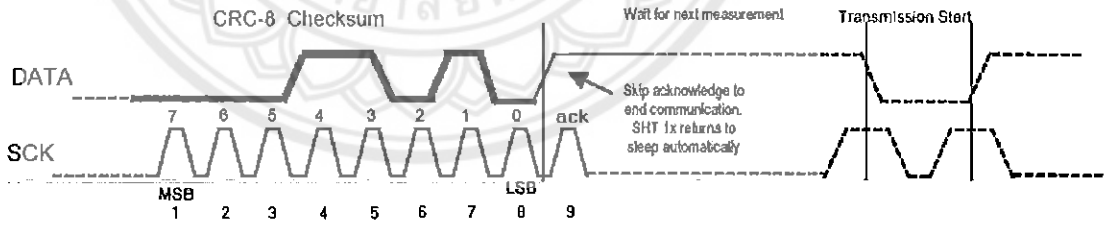
- ข้อมูลของ temperature มีขนาด 14bit และ humidity มีขนาด 12bit (สามารถเปลี่ยนได้เป็น 12 และ 8 bit โดย status register.)

- เมื่อได้รับ acknowledge แล้วให้ไมโครฯ รอสักครู่ประมาณ >210 ms เพื่อให้ Sensor พร้อม แล้วจึงส่งสัญญาณ SCK ต่อไปอีก 2 byte สำหรับรับข้อมูล และ 1 byte สำหรับ ข้อมูลตรวจสอบ ความผิดพลาด (CRC)

จากตัวอย่างจะสามารถอ่านข้อมูล 12 bit ของ Humidity (4 bit แรกเป็น 0 เสมอ) ได้เป็น 0000 1001 0011 0001= 2353 (dec)= 75.79%RH

เมื่อได้ข้อมูลครบแล้วหากต้องการ CRC ให้ตอบ acknowledge ด้วยการดึงขา Data ลงเป็น 0 หากไม่ต้องการก็ข้ามขั้นตอนนี้ไป

3.3 การอ่านข้อมูลเพื่อตรวจสอบข้อผิดพลาด (CRC)



รูปที่ ก.9 ลักษณะสัญญาณในการอ่านข้อมูล CRC8 1byte จาก Sensor

- ส่งสัญญาณ SCK ต่อไปอีก 1 byte (8 bit) สำหรับ ข้อมูล CRC

- หากสิ้นสุดการติดต่อก็ไม่ต้องตอบ Acknowledge ปล่อยให้ Data เป็น 1 ไว้ เมื่อผ่าน SCK ลูกที่ 9

- หากต้องการอ่านข้อมูลต่อ ให้ตอบ Acknowledge ด้วยการดึงขา Data ลงเป็น 0

ตาราง ก.4 ตารางหาค่า d1, d2 ขนาด 14 bit, 12 bit ที่แรงดัน 5 และ 3V

	Celsius		Fahrenheit	
SOT	**d1**	**d2**	**d1**	**d2**
14bit 5V	**-40**	**0.01**	**-40**	**0.018**
12bit 5V	**-40**	**0.04**	**-40**	**0.072**
14bit 3V	**-38.4**	**0.0098**	**-37.1**	**0.0176**
12bit 3V	**-38.4**	**0.0392**	**-37.1**	**0.0704**

3.4 การเปลี่ยนข้อมูล digital

การเปลี่ยนข้อมูล Digital เป็นข้อมูล temperature นั้น ค่าของข้อมูล temperature จะมีลักษณะเป็นเชิงเส้นสามารถคำนวนได้โดยใช้สูตร

$$Temperature = d_1 + d_2 * SO_T \qquad \text{(ข.1)}$$

เมื่อ SOT = Serial Output Temperature d1, d2 เป็น ค่าคงที่เชิงเส้น เช่นอ่านข้อมูล digital ขนาด14 bit ได้ SOT =011010 11111111B = 06911 dec ที่แรงดัน 5 V เมื่อต้องการอ่านค่าเป็นองศา Celsius จะได้ค่า d1= -40, d2 = 0.01 ผลการคำนวน temperature = 29.11C

การเปลี่ยนข้อมูล Digital เป็นข้อมูล humidity นั้น ค่าของข้อมูล humidity จะมีลักษณะเป็นเชิงเส้นสามารถคำนวนได้โดยใช้สูตร

$$RH_{linear} = c_1 + c_2 \bullet SO_{RH} + c_3 \bullet SO_{RH}{}^2 \qquad \text{(ข.2)}$$

เมื่อ RH = Serial Output humidity แบบ Linear และ c_1, c_2, c_3 มีค่าเป็น

$$c_1 = -4 \quad c_2 = 0.0405 \quad c_3 = -2.8*10^{-6} \quad \text{For 12bit } SO_{RH} \qquad \text{(ข.3)}$$

$$c_1 = -4 \quad c_2 = 0.648 \quad c_3 = -7.2*10^{-4} \quad \text{For 8bit } SO_{RH} \qquad \text{(ข.4)}$$

เมื่อได้ ค่า RH-linear และ Temperature แล้ว หาค่า RH true โดยใช้สูตร

$$RH_{true} = (T_{^\circ C} - 25) \bullet (t_1 + t_2 \bullet SO_{RH}) + RH_{linear} \qquad \text{(ข.5)}$$

$$t_1 = 0.01; t_2 = 0.00008 \ , \ t_2 = 0.00128 \quad \text{For 8bit } SO_{RH} \qquad \text{(ข.6)}$$

เช่นอ่านข้อมูล Digital ขนาด 12 bit ได้ RH-linear = 0110 11111111B =1791 dec ที่แรงดัน 5 V อ่านค่า Tc = 29.11C (จากค่า Temperature) เมื่อต้องการหาค่า humidity จะได้ค่า c1= -4, c2 = 0.0405, c3 = -0.0000028

ผลการคำนวนค่า Humidity = 60.18%

ภาคผนวก ค

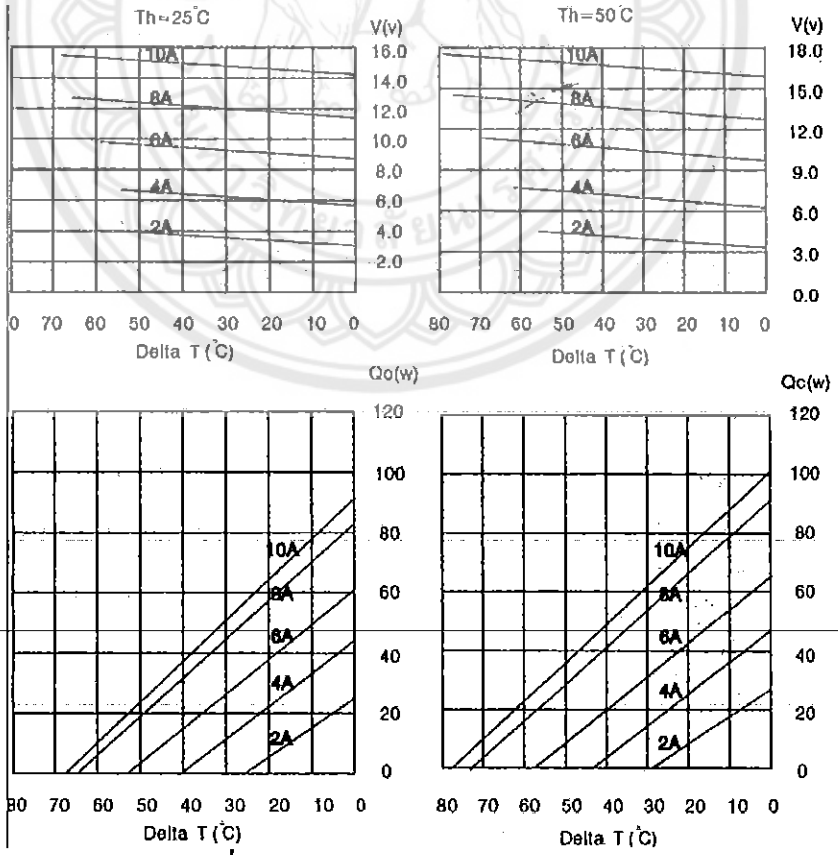
เพลเทียร์ (Peltier)

ประสิทธิภาพของเพลเทียร์

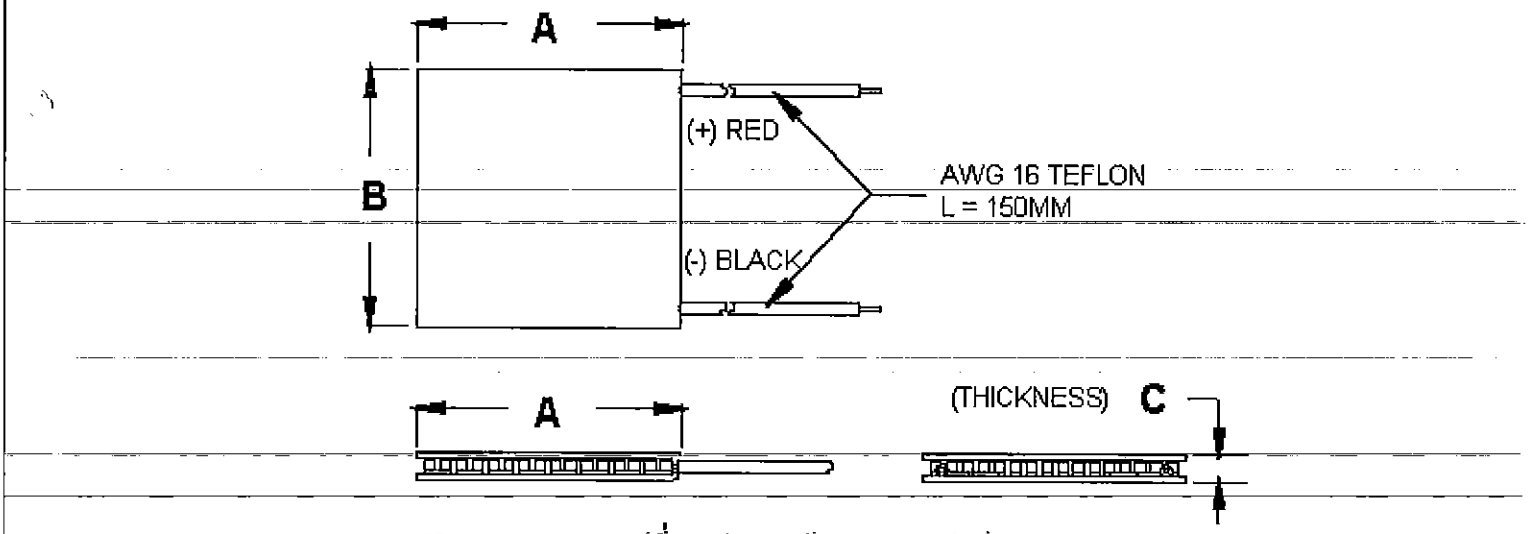
ตาราง ค.1 ประสิทธิภาพของเพลเทียร์

Hot Side Temperature (°C)	25°C	50°C
Qmax (Watts)	85	96
Delta Tmax (°C)	66	75
Imax (Amps)	10.5	10.5
Vmax (Volts)	15.2	17.4
Module Resistance (Ohms)	1.05	1.24

Performance curves



รูปที่ ค.1 กราฟประสิทธิภาพของเพลเทียร์



รูปที่ ค.2 โครงสร้างของเพลเทียร์

ตาราง ค.2 ขนาดของเพลเทียร์

A	B	C
40	40	3.3

Ceramic Material: Alumina (Al2O3)

Solder Construction: 138°C, Bismuth Tin (BiSn)

Operating Tips

- Max. Operating Temperature: 138°C
- Do not exceed Imax or Vmax when operating module.
- Please consult HB for moistureprotection options (seeling).
- Life expectancy: 200,000 hours
- Failure rate based on long time testings: 0.2%.

ภาคผนวก ง

ไฟล์โปรแกรม

```
CPU"8051.TBL"
        HOF"INT8"
        INCL"HEAD51.H"
        ;---------------------------------------
        ; Pin define
        ;---------------------------------------
        sda:    equ     p3.0
        sck:    equ     p1.0

        sw_set: equ     p3.5
        sw_up:  equ     p3.4
        sw_down:equ     p3.2

        PORT_LCD:   EQU   P1          ;Port data to LCD
        RS_LCD:     EQU   P1.1        ;RS select
        E_LCD:      EQU   P1.3        ;Enable LCD
        RW_LCD:     EQU   P1.2        ;R/W select
        RELAY:      EQU   P3.7        ;Relay

        ;-----------------------------------
        ;Bit define
        ;-----------------------------------
        a_bit:  equ     00
        temp_sign:      equ     a_bit
        meas_sel:       equ     a_bit+1
        I2C_ACKNOW:   EQU         a_bit+2
```

```
;-------------------------------------
;Internal RAM define
;-------------------------------------
a_ram:  equ     30h
lead0:  EQU     a_ram
lead1:  EQU     a_ram+1
lead2:  equ     a_ram+2
lead3:  equ     a_ram+3
multi0: EQU     a_ram+4
multi1: EQU     a_ram+5
result0: EQU    a_ram+6
result1: EQU    a_ram+7
result2: EQU    a_ram+8
result3: EQU    a_ram+9

xlead0:         EQU     a_ram+10
xlead1:         EQU     a_ram+11
xlead2:         EQU     a_ram+12
xlead3:         EQU     a_ram+13
xlead4:         EQU     a_ram+14
xlead5:         EQU     a_ram+15
xmulti0:        EQU     a_ram+16
xmulti1:        EQU     a_ram+17
xresult0:       EQU     a_ram+18
xresult1:       EQU     a_ram+19
xresult2:       EQU     a_ram+20
xresult3:       EQU     a_ram+21
xresult4:       EQU     a_ram+22
xresult5:       EQU     a_ram+23

rhlin_0: equ    a_ram+24
rhlin_1: equ    a_ram+25
```

```
rhlin1_0:       equ     a_ram+26
rhlin1_1:       equ     a_ram+27
rhlin2_0:       equ     a_ram+28
rhlin2_1:       equ     a_ram+29

rhtrue0: equ    a_ram+30
rhtrue1: equ    a_ram+31
rhtrue1_0:      equ     a_ram+32
rhtrue1_1:      equ     a_ram+33
rhtrue2_0:      equ     a_ram+34
rhtrue2_1:      equ     a_ram+35
rhtrue3_0:      equ     a_ram+36
rhtrue3_1:      equ     a_ram+37
rhtrue4_0:      equ     a_ram+38
rhtrue4_1:      equ     a_ram+39

sorh0:          equ     a_ram+40
sorh1:          equ     a_ram+41

temp_0:         equ     a_ram+42
temp_1:         equ     a_ram+43
temp1_0:        equ     a_ram+44
temp1_1:        equ     a_ram+45

sot0:           equ     a_ram+46
sot1:           equ     a_ram+47

bcd0:           equ     a_ram+48
bcd1:           equ     a_ram+49
hex_m:          equ     a_ram+50

dspbuf0:        equ     a_ram+51
```

```
dspbuf1:        equ     a_ram+52
dspbuf2:        equ     a_ram+53

tempshow0:      equ     a_ram+54
tempshow1:      equ     a_ram+55
tempshow2:      equ     a_ram+56

rhshow0:        equ     a_ram+57
rhshow1:        equ     a_ram+58
rhshow2:        equ     a_ram+59

rhmin0:         equ     a_ram+60
rhmin1:         equ     a_ram+61
rhmax0:         equ     a_ram+62
rhmax1:         equ     a_ram+63
rh_minset:      equ     a_ram+64
rh_maxset:      equ     a_ram+65

I2C_ADDR:       EQU     a_ram+66
I2C_DATA:       EQU     a_ram+67
TEMP_RAM:       EQU     a_ram+68
DATA_BUF:       EQU     a_ram+69
EEP_ADR_D:      EQU     a_ram+70

;-------------------------------------
;Constant define
;-------------------------------------
meas_temp:      equ     00000011b   ;measure temp code = 03h
meas_humi:      equ     00000101b   ;measure humi code = 05h

EEP_ADR_W:      EQU     0A0H    ;1010xxx0B is code eeprom_wr
EEP_ADR_R:      EQU     0A1H    ;1010xxx1B is code eeprom_rd
```

```
EEP_AD_min:  EQU   10h
eep_ad_max:  equ   20h
;*********************************************************

                ORG   0000h
MAIN:           clr   relay
                mov   b,#128
                mov   r0,#a_ram
clr_ram:        mov   @r0,#00
                inc   r0
                djnz  b,clr_ram

                ;---------------------------------------
                ;อ่านข้อมูลจาก eeprom
                ;---------------------------------------
                mov   a,#EEP_AD_min
                lcall rd_eeprom
                mov   rh_minset,data_buf
                mov   a,#eep_ad_max
                lcall rd_eeprom
                mov   rh_maxset,data_buf
                ;---------------------------

                CLR   RW_LCD
                CLR   E_LCD          ;Disable LCD
                MOV   R2,#0ffH       ;Power-on delay
PWR_DLY:        MOV   R3,#0FFH
                DJNZ  R3,$
                DJNZ  R2,PWR_DLY

                LCALL DELAY
```

```
                    LCALL   INIT_LCD    ;Initial LCD 4 bit 16X1
                    ;1st message
LOOP_LCD:           MOV     A,#4        ;1st location of line 1
                    LCALL   GOTO_LCD
                    LCALL   PRINT_LCD
                    DFB     "Humi"      ;Message on line-1
                    DFB     00H
                    MOV     A,#40H      ;2st location of line 2
                    LCALL   GOTO_LCD
                    LCALL   PRINT_LCD
                    DFB     "dity"      ;Message on line-2
                    DFB     00H
                    mov     r7,#250
                    lcall   delay_10ms
                    mov     r7,#250
                    lcall   delay_10ms

loop_meas:          lcall   sht15_con_rst
                    lcall   sht15_meas  ;อ่านค่าการวัดจาก SHT15
                    lcall   temp_convt  ;เปลี่ยน temp ผลอยู่ใน temp_1,temp_0
                    lcall   rh_convt    ;เปลี่ยน rh ผลอยู่ใน rhtrue1,rhtrue0
                    clr     c
                    mov     a,rh_maxset
                    subb    a,rhtrue1
                    jc      onrelay
                    clr     c
                    mov     a,rh_minset
                    subb    a,rhtrue1
                    jnc     onrelay
                    clr     relay
                    sjmp    wwww
onrelay: setb       relay
```

```
wwww:   mov   hex_m,temp_1   ;เปลี่ยน temp สำหรับส่ง LCD
        lcall bin2bcd
        lcall bcd2asc
        mov   tempshow0,dspbuf0
        mov   tempshow1,dspbuf1
        mov   tempshow2,dspbuf2

        mov   hex_m,rhtrue1   ;เปลี่ยน rh สำหรับส่ง LCD
        lcall bin2bcd
        lcall bcd2asc
        mov   rhshow0,dspbuf0
        mov   rhshow1,dspbuf1
        mov   rhshow2,dspbuf2
        lcall norshow

        clr   c

        JNB   SW_set,swset_chk
        sjmp  cccc

swset_chk: mov  r7,#1
        lCALL delay_10ms
        JB    SW_set;cccc
        lcall set_rh

cccc:   ljmp  loop_meas
        sjmp  $

;++++++++++++++++++++++++++++++++++++++++++++++++++++++++

;//////////////////////////////////////////////

;------------------------------
```

```
; Program read data from EEPROM
; I/P :
; O/P :
; Reg :R0,R2
;--------------------------------
RD_EEPROM:  MOV    I2C_ADDR,#EEP_ADR_W
            ACALL  I2C_TX_ADDR        ; Connect to slave

            MOV    I2C_DATA,a         ; EEPROM ADD. Pointer
            ACALLI2C_TX_DATA

            MOV    I2C_ADDR,#EEP_ADR_R
            ACALL  I2C_TX_ADDR        ; Connect to slave

            MOV    R0,#DATA_BUF       ;ข้อมูลอยู่ใน data_buf
            ACALL  I2C_RX_DATA

            SETB   SDA
            ACALLI2C_DELAY
            ACALLI2C_CLK
            ACALL  I2C_STOP
            RET

;--------------------------------
; I2C Slave Data Read
; I/P :
; O/P :
; Reg :A,R3
;--------------------------------
I2C_RX_DATA:   PUSH        A
          CLR   A
          MOV    R3,#08H
```

```
I2C_DATA_RD_1: ACALL    I2C_DELAY
               SETB   SCk
               ACALLI2C_DELAY
               MOV   C,SDA
               RLC    A
               CLR    SCk
          DJNZ   R3,I2C_DATA_RD_1
               MOV   @R0,A
               POP    A
          RET

;-----------------------------
; Program write data to EEPROM
; I/P :
; O/P :
; Reg :R0,R2
;-----------------------------
WR_EEPROM: MOV   I2C_ADDR,#EEP_ADR_W
               ACALL  I2C_TX_ADDR        ; Connect to slave
               MOV   I2C_DATA,A          ; EEPROM ADD. Pointer
          ACALL       I2C_TX_DATA
          MOV    R0,#DATA_BUF
     MOV    I2C_DATA,@R0
          ACALL  I2C_TX_DATA

          ACALL       I2C_STOP
          RET
```

```
;---------------------------------
; I2C Slave Address Connect
; I/P :
; O/P :
; Reg :A,R3
;---------------------------------
I2C_TX_ADDR:        PUSH   A
                    SETB   I2C_ACKNOW
                    MOV    A,I2C_ADDR
                    ACALL  I2C_START
                    MOV    R3,#08H
I2C_SLAVE_1:        RLC    A
                    MOV    SDA,C
                    ACALL  I2C_CLK
                    DJNZ   R3,I2C_SLAVE_1

                    SETB   SDA          ;Acknowledge bit
                    ACALL  I2C_DELAY
                    SETB   SCk
                    ACALL  I2C_DELAY
                    JB     SDA,I2C_SLAVE_2
                    CLR    I2C_ACKNOW
I2C_SLAVE_2:        CLR    SCk
                    POP    A
                    RET

;---------------------------------
; I2C Slave Data Write
; I/P :
; O/P :
; Reg :A,R3
;---------------------------------
```

```
I2C_TX_DATA:        PUSH   A
                    SETB   I2C_ACKNOW
                    MOV    A,I2C_DATA
                    MOV    R3,#08H
I2C_DATA_WR_1: RLC  A
                    MOV    SDA,C
                    ACALL  I2C_CLK
                    DJNZ   R3,I2C_DATA_WR_1
                SETB   SDA
                    ACALL  I2C_DELAY
                    SETB   SCk
                    ACALL  I2C_DELAY
                    JB     SDA,I2C_DATA_WR_2
                    CLR    I2C_ACKNOW
I2C_DATA_WR_2: CLR  SCK
                    POP    A
                    RET

;-------------------------------
; I2C Start Condition
; I/P :
; O/P :
; Reg :
;-------------------------------
I2C_START:   SETB   SCK
                    SETB   SDA
                    ACALL       I2C_DELAY
                    CLR    SDA
                    ACALL       I2C_DELAY
                    CLR    SCK
                    RET
```

```
;---------------------------------
; I2C Stop Condition
; I/P :
; O/P :
; Reg :
;---------------------------------
I2C_STOP:    CLR    SDA
             ACALL  I2C_DELAY
             SETB   SCK
             ACALL  I2C_DELAY
             SETB   SDA
             RET

;---------------------------------
; I2C Clock
; I/P :
; O/P :
; Reg :
;---------------------------------
I2C_CLK:     ACALL  I2C_DELAY
             SETB   SCK
             ACALL  I2C_DELAY
             CLR    SCK
             RET

;---------------------------------
; I2C Delay
; I/P :
; O/P :
; Reg : R6
;---------------------------------
I2C_DELAY:   MOV    R6,#0CH
```

```
I2C_DELAY_1:   NOP
               NOP
               DJNZ   R6,I2C_DELAY_1
               RET

;////////////////////////////////////////

;-----------------------------------------
; set rh_min
;-----------------------------------------
set_rh:  clr      relay
               lcall    showsetmin

               jnb      sw_set,$  ;///////////////

set_rhmin1:    jnb      sw_up,rhmin_up
               jnb      sw_down,rhmin_down
               JNB      SW_set,set_rhmax
               sjmp     set_rhmin1

rhmin_up:      mov      r7,#1
               lcall    delay_10ms
               jb       sw_up,set_rhmin1
               MOV      R4,rh_minset
               CJNE     R4,#99,INC_rhmin
               sJMP     NOmin_UP
INC_rhmin:     INC      rh_minset

NOmin_UP:      lCALL    showsetmin

               SJMP     set_rhmin1
```

```
rhmin_down:   mov    r7,#1
              lcall  delay_10ms
              jb     sw_down,set_rhmin1
              MOV    R4,rh_minset
              CJNE   R4,#00,dec_rhmin
              sJMP   NOmin_down
dec_rhmin:    dec    rh_minset

NOmin_down:   lCALL  showsetmin

              SJMP   set_rhmin1

set_rhmax:    mov    r7,#1
              lcall  delay_10ms
              jb     sw_set,set_rhmin1
              lcall  showsetmax

              jnb    sw_set,$

set_rhmax1:   jnb    sw_up,rhmax_up
              jnb    sw_down,rhmax_down
              JNB    SW_set,saveset
              sjmp   set_rhmax1

rhmax_up:     mov    r7,#1
              lcall  delay_10ms
              jb     sw_up,set_rhmax1
              MOV    R4,rh_maxset
              CJNE   R4,#99,INC_rhmax
              sJMP   NOmax_UP
INC_rhmax:    INC    rh_maxset
```

```
NOmax_UP:   lCALL  showsetmax
            SJMP   set_rhmax1

rhmax_down: mov    r7,#1
            lcall  delay_10ms
            jb     sw_down,set_rhmax1
            MOV    R4,rh_maxset
            CJNE   R4,#00,dec_rhmax
            sJMP   NOmax_down
dec_rhmax:  dec    rh_maxset

NOmax_down: lCALL  showsetmax
            SJMP   set_rhmax1

saveset: mov r7,#1
            lcall  delay_10ms
            jb     sw_set,set_rhmax1

            mov    r7,#200
saveDELAY:  MOV    TMOD,#01H  ;T0 to timer mode1  DELAY_10mS
            MOV    TH0,#0DCH
            MOV    TL0,#0BFH
            SETB   TR0
jjjj:       jb     sw_set,set_rhmax1
            JNB    TF0,jjjj
            CLR    TR0
            CLR    TF0
            DJNZ   R7,saveDELAY
```

```
                mov     a,#EEP_AD_min
                mov     data_buf,rh_minset
                lcall   wr_eeprom

                mov     r7,#20
                lcall   delay_10ms

                mov     a,#eep_ad_max
                mov     data_buf,rh_maxset
                lcall   wr_eeprom

                lcall   norshow

                jnb     sw_set,$

                RET

;*****************************
; Normal show temp. $ %RH
;*****************************
norshow:        MOV     A,#01H          ;Clear display
                LCALL WR_INS            ;Clear display,Set DD RAM addr=0

                MOV     A,#0            ;1st location of line 1
                LCALL           GOTO_LCD
                LCALL           PRINT_LCD
                DFB     " T="           ;Message on line-1
                ;DFB    0dfh
                DFB     00H
                MOV     A,#40H          ;2st location of line 2
```

```
                LCALL   GOTO_LCD
                LCALL   PRINT_LCD
                DFB     " RH="          ;Message on line-2
                DFB     00H

                MOV     A,#03h          ;แสดง temp on lcd
                LCALL   GOTO_LCD
                mov     a,tempshow2
                cjne    a,#30h,nextshow
                sjmp    nextshow1
nextshow:       lcall   wr_lcd
nextshow1:      mov     a,tempshow1
                cjne    a,#30h,nextshow2
                sjmp    nextshow3
nextshow2:      lcall   wr_lcd
nextshow3:      mov     a,tempshow0
                lcall   wr_lcd
                mov     a,#0dfh
                lcall   wr_lcd

                MOV     A,#44h          ;แสดง temp on lcd
                LCALL   GOTO_LCD
                ;mov    a,rhshow2
                ;lcall  wr_lcd
                mov     a,rhshow1
                cjne    a,#30h,nextshow4
                sjmp    nextshow5
nextshow4:      lcall   wr_lcd
nextshow5:      mov     a,rhshow0
                lcall   wr_lcd
                mov     a,#25h
                lcall   wr_lcd
```

```
                mov     r7,#250
                lcall   delay_10ms
                mov     r7,#250
                lcall   delay_10ms

                ret

;*****************************
; Show set min
;*****************************

showsetmin:     mov     hex_m,rh_minset ;เปลี่ยน temp สำหรับส่ง LCD
                lcall   bin2bcd
                lcall   bcd2asc
                mov     rhmin0,dspbuf0
                mov     rhmin1,dspbuf1


                MOV     A,#01H      ;Clear display
                LCALL WR_INS    ;Clear display,Set DD RAM addr=0

                MOV     A,#0            ;1st location of line 1
                LCALL           GOTO_LCD
                LCALL           PRINT_LCD
                DFB     "Set: Low"      ;Message on line-1
                DFB     00H
                MOV     A,#40H          ;2st location of line 2
                LCALL           GOTO_LCD
                LCALL           PRINT_LCD
                DFB     " RH="  ;Message on line-2
```

```
                DFB     00H

                MOV     A,#44h          ;แสดง set min rh on lcd
                LCALL   GOTO_LCD
                ;mov    a,rhshow2
                ;lcall  wr_lcd
                mov     a,rhmin1
                cjne    a,#30h,dddshow4
                sjmp    dddshow5
dddshow4:       lcall   wr_lcd
dddshow5:       mov     a,rhmin0
                lcall   wr_lcd
                mov     a,#25h
                lcall   wr_lcd

                mov     r7,#50
                lcall   delay_10ms
                ret

;*****************************
; Show set max
;*****************************

showsetmax:     mov     hex_m,rh_maxset     ;เปลี่ยน temp สำหรับส่ง LCD
                lcall   bin2bcd
                lcall   bcd2asc
                mov     rhmax0,dspbuf0
                mov     rhmax1,dspbuf1

                MOV     A,#01H          ;Clear display
                LCALL WR_INS            ;Clear display,Set DD RAM addr=0
```

```
                MOV   A,#0          ;1st location of line 1
                LCALL       GOTO_LCD
                LCALL       PRINT_LCD
                DFB   "Set:High"    ;Message on line-1
                DFB   00H
                MOV   A,#40H        ;2st location of line 2
                LCALL       GOTO_LCD
                LCALL       PRINT_LCD
                DFB   " RH="  ;Message on line-2
                DFB   00H

                MOV   A,#44h        ;แสดง set min rh  on lcd
                LCALL       GOTO_LCD
                ;mov   a,rhshow2
                ;lcall   wr_lcd
                mov   a,rhmax1
                cjne   a,#30h,eeeshow4
                sjmp   eeeshow5
eeeshow4:       lcall   wr_lcd
eeeshow5:       mov   a,rhmax0
                lcall   wr_lcd
                mov   a,#25h
                lcall   wr_lcd

                mov   r7,#50
                lcall   delay_10ms

                ret
;****************************************
; การอ่านข้อมูลจาก SHT15
;****************************************
sht15_meas:     clr   meas_sel          ;อ่าน temp
```

```
                lcall   sht15_rd_meas

                mov     r7,#30
                lcall   delay_10ms
                lcall   sht15_con_rst

                setb    meas_sel        ;อ่าน humi
                lcall   sht15_rd_meas
                ret

;**********************************************************
; SHT15 connection reset
;**********************************************************
sht15_con_rst:  setb    sda
                clr     sck
                mov     b,#09h
con_rst_1:      setb    sck
                lcall   sht15_wait
                clr     sck
                lcall   sht15_wait
                djnz    b,con_rst_1
                lcall   sht15_trans_st
                ret

;**********************************************************
; SHT15 transmission start
;**********************************************************
sht15_trans_st: setb    sda
                clr     sck
                lcall   sht15_wait
```

```
                setb    sck
                lcall   sht15_wait
                clr     sda
                lcall   sht15_wait
                clr     sck
                lcall   sht15_wait
                setb    sck
                lcall   sht15_wait
                setb    sda
                lcall   sht15_wait
                clr     sck
                ret

;*****************************************************
; SHT15 wait
;*****************************************************
sht15_wait:     push    b
                mov     b,#08
sht15_wait_1:   nop
                djnz    b,sht15_wait_1
                pop     b
                ret

;*****************************************************
; SHT15 write command
;*****************************************************
sht15_wr_cmd:   mov     b,#08h
sht15_wrcmd_1:rlc       a
                mov     sda,c
                nop
                nop
                setb    sck
```

```
                lcall   sht15_wait
                clr     sck
                djnz    b,sht15_wrcmd_1
                setb    sda
                setb    sck
                lcall   sht15_wait
                clr     sck
                ret

;*************************************************************
; SHT15         read command
;*************************************************************
sht15_rd_cmd:   ;setb   sda
                mov     b,#08h
sht15_rdcmd_1:  setb    sck
                nop
                nop
                mov     c,sda
                rlc     a
                clr     sck
                nop
                nop
                djnz    b,sht15_rdcmd_1
                clr     sda
                setb    sck
                lcall   sht15_wait
                clr     sck
                ret

;*************************************************************
; Read measure temp. & humidity from SHT15
;*************************************************************
```

```
sht15_rd_meas: lcall   sht15_trans_st
               jb      meas_sel,rd_meas_rh
               mov     a,#meas_temp
               lcall   sht15_wr_cmd
               mov     r7,#30
               lcall   delay_10ms
               lcall   sht15_rd_cmd
               mov     sot1,a
               lcall   sht15_rd_cmd
               setb    sda
               mov     sot0,a
               sjmp    stop_rd_meas
rd_meas_rh:    mov     a,#meas_humi
               lcall   sht15_wr_cmd
               mov     r7,#15
               lcall   delay_10ms
               lcall   sht15_rd_cmd
               mov     sorh1,a
               lcall   sht15_rd_cmd
               setb    sda
               mov     sorh0,a
stop_rd_meas:  ret

;*********************************************
; Delay 10 mS
; I/P : R7
; O/P : None
; Reg. : None
;*********************************************
DELay_10mS:  MOV   TMOD,#01H  ;T0 to timer mode1
             MOV   TH0,#0DCH
             MOV   TL0,#0BFH
```

```
                SETB    TR0
WAIT10mS:       JNB     TF0,WAIT10mS
                CLR     TR0
                CLR     TF0
                DJNZ    R7,DELay_10mS
                RET

;*************************************************************
; convert TEMP. of SHT15 output to Physical values
;*************************************************************
temp_convt:     clr     temp_sign
                mov     lead0,#0f5h     ;d2
                mov     lead1,#28h
                mov     multi0,sot0     ;d2*sot
                mov     multi1,sot1
                lcall   mul16
                mov     temp1_0,result2
                mov     temp1_1,result3

                clr     c
                mov     a,#02h
                subb    a,temp1_1
                jc      temp_con1
                clr     c
                mov     a,temp1_1
                subb    a,#02h
                jc      temp_con2
                clr     c
                mov     a,temp1_0
                subb    a,#80h
                jnc     temp_con1
```

```
temp_con2:      clr     c
                mov     a,#80h
                subb    a,temp1_0
                mov     temp_0,a
                mov     a,#02h
                subb    a,temp1_1
                mov     temp_1,a
                setb    temp_sign
                sjmp    temp_shift_1
temp_con1:      clr     c
                mov     a,temp1_0
                subb    a,#80h
                mov     temp_0,a
                mov     a,temp1_1
                subb    a,#02h
                mov     temp_1,a

temp_shift_1:   mov     b,#4
temp_shift:     clr     c
                mov     a,temp_0
                rlc     a
                mov     temp_0,a
                mov     a,temp_1
                rlc     a
                mov     temp_1,a
                djnz    b,temp_shift
temp_conend:    ret     ;stop temp. convert [temp_1,temp_0 : xx.xx]


;***********************************************************
; convert %RH of SHT15 output to Physical values
;***********************************************************
```

```
rh_convt:   mov   lead0,sorh0     ;sorh^2
            mov   lead1,sorh1
            mov   multi0,sorh0
            mov   multi1,sorh1
            lcall mul16

            mov   xlead0,result0  ;c3*sorh^2
            mov   xlead1,result1
            mov   xlead2,result2
            mov   xlead3,result3

            mov   xmulti0,#0f9h   ;c3
            mov   xmulti1,#2eh
            lcall xmul16
            mov   rhlin1_0,xresult3
            mov   rhlin1_1,xresult4

            mov   lead0,sorh0     ;c2*sorh
            mov   lead1,sorh1
            mov   multi0,#0e3h    ;c2
            mov   multi1,#0a5h
            lcall mul16
            mov   rhlin2_0,result2
            mov   rhlin2_1,result3
            mov   b,#04
rh_shift:   clr   c
            mov   a,rhlin2_0
            rlc   a
            mov   rhlin2_0,a
            mov   a,rhlin2_1
            rlc   a
            mov   rhlin2_1,a
```

```
djnz    b,rh_shift

clr     c               ;c1+c2*sorh
mov     a,rhlin2_1
subb    a,#04h
mov     rhlin_1,a
mov     rhlin_0,rhlin2_0

clr     c               ;c1+c2*sorh+c3*sorh^2
mov     a,rhlin_0
subb    a,rhlin1_0
mov     rhlin_0,a
mov     a,rhlin_1
subb    a,rhlin1_1
mov     rhlin_1,a

;======== calculator rhtrue ====================
clr     c               ;(tc-25)
mov     a,temp_1
subb    a,#19h          ;25d
mov     rhtrue1_1,a
mov     rhtrue1_0,#00h

mov     lead0,sorh0     ;t2*sorh
mov     lead1,sorh1
mov     multi0,#0e2h    ;t2
mov     multi1,#53h
lcall   mul16
mov     rhtrue2_1,result3
mov     rhtrue2_0,result3
```

```
mov     a,rhtrue2_1
swap    a
anl     a,#0fh
mov     rhtrue2_1,a
mov     a,rhtrue2_0
swap    a
anl     a,#0f0h
mov     rhtrue2_0,a

mov     a,rhtrue2_0     ;t1+t2*sorh
add     a,#02H
mov     rhtrue3_0,a
mov     rhtrue3_1,rhtrue2_1

mov     lead0,rhtrue1_0 ;(tc-25)*(t1+t2*sorh)
mov     lead1,rhtrue1_1
mov     multi0,rhtrue3_0
mov     multi1,rhtrue3_1
lcall   mul16
mov     rhtrue4_0,result1
mov     rhtrue4_1,result2

mov     a,rhtrue4_0
add     a,rhlin_0
mov     rhtrue0,a
mov     a,rhtrue4_1
addc    a,rhlin_1
mov     rhtrue1,a
ret     ;stop convert %rh   [rhtrue1,rhtrue0 : xx.xx]
```

```
;************************************************************
;  16 bit multiplier
; register : A,B,R0,R1,R7
; i/p : lead1,lead0,multi1,multi0
; o/p : result3,result2,result1,result0
;************************************************************
mul16:      mov   a,#00
            mov   result0,a
            mov   result1,a
            mov   result2,a
            mov   result3,a

            mov   lead2,#0
            mov   lead3,#0

            mov   b,#16
mul16_re:   clr   c
            mov   a,multi1
            rrc   a
            mov   multi1,a
            mov   a,multi0
            rrc   a
            mov   multi0,a

            jnc   mul16_1

            mov   r7,#4
            clr   c
            mov   r0,#lead0
            mov   r1,#result0
mul16_add:  mov   a,@r0
            addc  a,@r1
```

```
                    mov     @r1,a
                    inc     r0
                    inc     r1
                    djnz    r7,mul16_add
mul16_1:            mov     r7,#4
                    clr     c
                    mov     r0,#lead0
mul16_2:            mov     a,@r0
                    rlc     a
                    mov     @r0,a
                    inc     r0
                    djnz    r7,mul16_2
                    djnz    b,mul16_re
                    ret

;*********************************************************
;  Extra 16 bit multiplier
; register : A,B,R0,R1,R7
; i/p : lead1,lead0,multi1,multi0
; o/p : result3,result2,result1,result0
;*********************************************************
xmul16:             mov     a,#00
                    mov     xresult0,a
                    mov     xresult1,a
                    mov     xresult2,a
                    mov     xresult3,a
                    mov     xresult4,a
                    mov     xresult5,a

                    mov     xlead4,#00
                    mov     xlead5,#00
```

```
                    mov     b,#16
xmul16_re:          clr     c
                    mov     a,xmulti1
                    rrc     a
                    mov     xmulti1,a
                    mov     a,xmulti0
                    rrc     a
                    mov     xmulti0,a

                    jnc     xmul16_1

                    mov     r7,#6
                    clr     c
                    mov     r0,#xlead0
                    mov     r1,#xresult0
xmul16_add:         mov     a,@r0
                    addc    a,@r1
                    mov     @r1,a
                    inc     r0
                    inc     r1
                    djnz    r7,xmul16_add
xmul16_1:           mov     r7,#6
                    clr     c
                    mov     r0,#xlead0
xmul16_2:           mov     a,@r0
                    rlc     a
                    mov     @r0,a
                    inc     r0
                    djnz    r7,xmul16_2
                    djnz    b,xmul16_re
                    ret
```

```
;**************************************
; Binary to BCD
; I/P : hex
; o/p : bcd2,bcd1,bcd0
;**************************************
bin2bcd:        mov     a,#00
                mov     bcd0,a
                mov     bcd1,a
                mov     a,hex_m
                rlc     a
                mov     hex_m,a
                jnc     chk_bit6
                mov     bcd1,#01h
                mov     bcd0,#28h
chk_bit6:       mov     a,hex_m
                rlc     a
                mov     hex_m,a
                jnc     chk_bit5
                mov     a,bcd0
                add     a,#64h
                daa
                mov     bcd0,a
                mov     a,bcd1
                addc    a,#00h
                mov     bcd1,a
chk_bit5:       mov     a,hex_m
                rlc     a
                mov     hex_m,a
                jnc     chk_bit4
                mov     a,bcd0
                add     a,#32h
```

```
                daa
                mov     bcd0,a
                mov     a,bcd1
                addc    a,#00h
                mov     bcd1,a
chk_bit4:       mov     a,hex_m
                rlc     a
                mov     hex_m,a
                jnc     chk_bit3
                mov     a,bcd0
                add     a,#16h
                daa
                mov     bcd0,a
                mov     a,bcd1
                addc    a,#00h
                mov     bcd1,a
chk_bit3:       mov     a,hex_m
                rlc     a
                mov     hex_m,a
                jnc     chk_bit2
                mov     a,bcd0
                add     a,#08h
                daa
                mov     bcd0,a
                mov     a,bcd1
                addc    a,#00h
                mov     bcd1,a
chk_bit2:       mov     a,hex_m
                rlc     a
                mov     hex_m,a
                jnc     chk_bit1
                mov     a,bcd0
```

```
                add     a,#04h
                daa
                mov     bcd0,a
                mov     a,bcd1
                addc    a,#00h
                mov     bcd1,a
chk_bit1:       mov     a,hex_m
                rlc     a
                mov     hex_m,a
                jnc     chk_bit0
                mov     a,bcd0
                add     a,#02h
                daa
                mov     bcd0,a
                mov     a,bcd1
                addc    a,#00h
                mov     bcd1,a
chk_bit0:       mov     a,hex_m
                rlc     a
                mov     hex_m,a
                jnc     stop_bin2bcd
                mov     a,bcd0
                add     a,#01h
                daa
                mov     bcd0,a
                mov     a,bcd1
                addc    a,#00h
                mov     bcd1,a
stop_bin2bcd:   ret
```

```
;*****************************************************
; BCD to ASCii
;*****************************************************
bcd2asc:        mov     a,bcd0
                anl     a,#0fh
                add     a,#30h
                mov     dspbuf0,a
                mov     a,bcd0
                swap    a
                anl     a,#0fh
                add     a,#30h
                mov     dspbuf1,a
                mov     a,bcd1
                anl     a,#0fh
                add     a,#30h
                mov     dspbuf2,a
                ret

;-----------------------------------
;Initial LCD 4 bit interface
;-----------------------------------
INIT_LCD:       MOV     A,#33H          ;DL = 1 , DL = 1
                LCALL WR_INS
                MOV     A,#32H          ;DL = 1 , DL = 0
                LCALL WR_INS
                MOV     A,#28H          ;Function set
                LCALL WR_INS            ;DL=0 4bit,N=0 1line,F=0 5x7
                MOV     A,#0CH          ;Display on/off control
                LCALL WR_INS            ;Display on,cursor off,not blink
                MOV     A,#06H          ;Entry mode set
                LCALL           WR_INS          ;I/D=1
```

```
Increment,S=0 cursor shift
                MOV   A,#01H          ;Clear display
                LCALL WR_INS          ;Clear display,Set DD RAM addr=0
                RET

;---------------------------------------
;Write instruction LCD
;Input : Reg. ACC (Command)
;---------------------------------------
WR_INS:         PUSH  A               ;Save data low byte
                ANL   A,#0F0H
                MOV   PORT_LCD,A      ;High byte & RS=0 Write
                LCALL EN_LCD
                POP   A               ;Restore data low byte
                SWAP  A               ;Shift low byte to high byte
                ANL   A,#0F0H
                MOV   PORT_LCD,A      ;Low byte & RS=0 Write
                LCALL EN_LCD
                MOV   R2,#03H         ;Busy instruction delay
INS_DLY:        MOV   R3,#250
                DJNZ  R3,$
                DJNZ  R2,INS_DLY
                RET

;-----------------------------------------
;Enable pin E of LCD
;-----------------------------------------
EN_LCD:         PUSH  B
                MOV   B,#50           ;Busy delay LCD
                DJNZ  B,$
                POP   B
                SETB  E_LCD           ;Enable LCD (pin E=Px.2=1)
```

```
                NOP
                NOP
                CLR     E_LCD           ;Disable LCD (pin E=Px.2=0)
                RET

;----------------------------------------
;Write ASCII to LCD
;Input: ACC (ASCII)
;----------------------------------------
WR_LCD:         PUSH    A               ;Save data low byte
                ANL     A,#0F0H
                MOV     PORT_LCD,A      ;High byte & RS=1 Write
                SETB    RS_LCD          ;Set RS for write data select
                LCALL EN_LCD
                POP     A               ;Restore data low byte
                SWAP    A               ;Shift low byte to high byte
                ANL     A,#0F0H
                MOV     PORT_LCD,A      ;Low byte & RS=1 Write
                SETB    RS_LCD          ;Set RS for write data select

                LCALL EN_LCD
                RET

;----------------------------------------
;Goto position of LCD
;Input : ACC (address of LCD)
;----------------------------------------
GOTO_LCD:       SETB    0E7H
                LCALL WR_INS
                RET
```

```
;-----------------------------------------
;Print data to LCD display
;-----------------------------------------
PRINT_LCD:  POP   DPH
            POP   DPL
PRINT1:     CLR   A
            MOVC  A,@A+DPTR
            CJNE  A,#00H,PRINT2
            LJMP  PRINT3
PRINT2:     LCALL WR_LCD
            INC   DPTR
            LJMP  PRINT1
PRINT3:     PUSH  DPL
            PUSH  DPH
            RET

;-----------------------------------
; Delay time
;-----------------------------------
DELAY:      MOV   R2,#10
DLY1:       MOV   R3,#0FFH
DLY2:       MOV   R4,#0FFH
            DJNZ  R4,$
            DJNZ  R3,DLY2
            DJNZ  R2,DLY1
            RET

            end
```

ประวัติผู้จัดทำโครงงาน

ชื่อ นายกิตติพงษ์ ยศอาลัย

ภูมิลำเนา 42 หมู่ 8 ต.เปือ อ.เชียงกลาง จ.น่าน 55160

ประวัติการศึกษา

- จบระดับมัธยมจากโรงเรียนเชียงกลาง"ประชาพัฒนา"
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4

สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์

มหาวิทยาลัยนเรศวร

Email : yosarlai@hotmail.com

ชื่อ นายอุกฤษฎ์ ชัยยา

ภูมิลำเนา 104 ต.แม่ต๋ำ อ.เมือง จ.พะเยา 56000

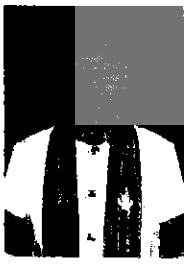
ประวัติการศึกษา

- จบระดับมัธยมจากโรงเรียนพะเยาพิยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4

สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์

มหาวิทยาลัยนเรศวร

Email : ukrit_cha@hotmail.com



ชื่อ นายธนุวัฒน์ แก้วกัน

ภูมิลำเนา 126 หมู่ 16 ต.ท่าพล อ.เมือง จ.เพชรบูรณ์ 62750

ประวัติการศึกษา

- จบระดับมัธยมจากโรงเรียนสวนกุหลาบวิทยาลัย เพชรบูรณ์
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4

สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์

มหาวิทยาลัยนเรศวร

Email : Friday_334@hotmail.com