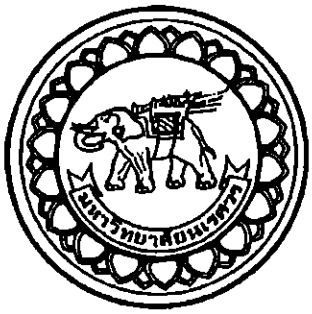


เครื่องควบคุมอุณหภูมิและความชื้น

Temperature and Humidity Controller

นายจักรพันธ์ หวาจ้อย รหัส 47363775

นายปรัชญา จันทร์คามิ รหัส 47363932

i 50ร0836 e 2

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 30 มิ.ย. 2551
เลขทะเบียน 05100030
เลขเรียกหนังสือ ปร.
มหาวิทยาลัยนเรศวร 7225ค
2550

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2550

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	เครื่องควบคุมอุณหภูมิและความชื้น
ผู้ดำเนินโครงงาน	นายจักรพันธ์ หวาจ้อย รหัส 47363775
	นายปรัชญา จันทร์คามิ รหัส 47363932
อาจารย์ที่ปรึกษา	ดร. อัครพันธ์ วงศ์กังแห
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2550

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

คณะกรรมการสอบโครงงานวิศวกรรม

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

.....กรรมการ
(อาจารย์ปิยคนัย ภาชนะพรรณ์)

.....กรรมการ
(ดร. แคทรียา สุวรรณศรี)

หัวข้อโครงงาน	เครื่องควบคุมอุณหภูมิและความชื้น
ผู้ดำเนินโครงงาน	นายจักรพันธ์ หวาจ้อย รหัส 47363775
	นายปรัชญา จันทร์คามิ รหัส 47363932
อาจารย์ที่ปรึกษา	ดร.อัครพันธ์ วงศ์กังแห
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2550

.....

บทคัดย่อ

โครงงานนี้มีจุดประสงค์เพื่อแก้ไขความแปรปรวนเครื่องควบคุมอุณหภูมิของเครื่องปรับอากาศชนิดรวมศูนย์ ของบ้านประหยัดพลังงานที่มีอยู่เดิม แต่ไม่สามารถควบคุมอุณหภูมิและความชื้นได้เที่ยงตรงตามค่าที่ต้องการ ดังนั้นคณะผู้วิจัยจึงได้ใช้เครื่องควบคุมอุณหภูมิและความชื้น รุ่น AP-105 ของบริษัท ศิลารีเสิร์จ จำกัด มาทำการพัฒนาโดยการเปลี่ยนไมโครคอนโทรลเลอร์จากเบอร์ 89C55WD เป็นเบอร์ P89V51RD2FN และสร้างโปรแกรมขึ้นมาใหม่

ผลจากการทดลองเครื่องควบคุมอุณหภูมิและความชื้นมีค่าความเที่ยงตรง แม่นยำ ตรงตามความต้องการของผู้ประกอบการได้ มีผลทำให้การใช้พลังงานของเครื่องปรับอากาศ ฯ ลดลงจากเดิมร้อยละ 27

Project Title	Temperature and Humidity Controller.
Name	Mr. Jakkapan Whajoy ID. 47363775
	Mr. Prachaya Junkami ID. 47363932
Project Advisor	Akaraphunt Vongkunghae, Ph. D.
Major	Electrical Engineering.
Department	Electrical and Computer Engineering.
Acadamic Year	2007

ABSTRACT

This project is to built a controller for and air conditioner system, used in a house. The regular controller can not detect the fault change of temperature and humidity because the sensor is too sensitive. The controller in this project is reprogrammed with moving average filter. There for, it can handle the kid of sudden change.

The result of experiment the temperature and humidity controller have a valuable accuracy follow the requirement of the entrepreneur, which the power usage of air - conditioner decreases 27%, comparing to the regular one as sold in market generally.

กิตติกรรมประกาศ

โครงงานฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีด้วยความช่วยเหลือจากบุคคลหลายฝ่ายโดยเฉพาะอย่างยิ่ง ขอขอบพระคุณ ดร.อัครพันธ์ วงกังแห ที่ได้ให้แนวคิด ตลอดจนเสียสละเวลา ในการตรวจแก้ไข ข้อบกพร่องต่างๆ และสำนักงานกองทุนสนับสนุนการวิจัยฝ่ายอุตสาหกรรมที่ได้ให้ทุนอุดหนุนในการทำโครงงานนี้ นอกจากนี้คณะผู้จัดทำขอขอบพระคุณ คุณเจริญ และคุณอำพร บัวเทศ ผู้จัดการห้างหุ้นส่วนจำกัด เซมดีไซน์และคุณธวัชชัย กลิ่นรอด เจ้าของบ้านประหยัดพลังงาน และผู้มีอุปการคุณทุกท่านเป็นอย่างสูง ที่มีส่วนร่วมในการให้ข้อมูลและช่วยเหลือในหลายๆ ด้านจนโครงงานนี้สำเร็จลุล่วงไปได้ดี

ท้ายสุดนี้ ขอกราบขอบพระคุณบิดร มารดา และ คณาจารย์ทุกท่านที่ได้อบรมสั่งสอน ให้ความรู้ ให้คำแนะนำ ให้คำปรึกษาที่มีคุณค่า แก่คณะผู้วิจัยด้วยดีเสมอมา

คณะผู้จัดทำโครงงาน
นายจักรพันธ์ หวาจ้อย
นายปรัชญา จันทร์คามิ

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ฉ

สารบัญรูป ..... ช

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของโครง 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบข่ายของโครงงาน 1

1.4 ขั้นตอนการดำเนินโครงงาน 2

1.5 แผนการดำเนินโครงงาน 3

1.6 ผลลัพธ์ที่คาดว่าจะได้รับ 4

1.7 งบประมาณของโครงงาน 4

บทที่ 2 ทฤษฎีและหลักการทำงาน

2.1 หลักการทำงานของเครื่องทำความเย็นและปรับอากาศ 5

2.2 วงจรไฟฟ้ากำลังควบคุมระบบเครื่องปรับอากาศ 6

2.3 หลักการเขียนโปรแกรมด้วยภาษาซี 7

2.4 ไมโครคอนโทรลเลอร์ 8

2.5 การเขียนโปรแกรมไมโครคอนโทรลเลอร์ 17

2.6 โปรแกรมแมทแล็บ 17

บทที่ 3 วิธีการดำเนินงาน

3.1 การศึกษาการทำงาน 20

3.2 การออกแบบชิ้นงาน 21

3.3 การสร้างชิ้นงาน 24

สารบัญ (ต่อ)

หน้า

3.4 การทดสอบชิ้นงาน 26
3.5 รวบรวมข้อมูลทั้งหมดเข้ารูปเล่มพร้อมรายงาน 29

บทที่ 4 วิธีการทดสอบและการทดสอบ
4.1 การนำไปใช้งานจริง 30
4.2 ผลการทดลอง 31

บทที่ 5 บทสรุป อภิปรายผล ข้อเสนอแนะ
5.1 สรุปผล 34
5.2 อภิปรายผล 34
5.3 ข้อเสนอแนะ 35

เอกสารอ้างอิง 36

ภาคผนวก
ภาคผนวก ก โปรแกรมการควบคุมการทำงานของเครื่องควบคุมอุณหภูมิ
และความชื้น 38
ภาคผนวก ข การแสดงกราฟอุณหภูมิและความชื้นที่ได้ออกทาง
หน้าจอคอมพิวเตอร์โดยใช้โปรแกรม MATLAB 54

ประวัติผู้เขียนโครงงาน 65

สารบัญตาราง

ตารางที่ หน้า

1.1 แผนการดำเนินโครงงาน 3

2.1 ชื่อขา ตำแหน่งขา ชนิดขา และรายละเอียดการทำงานของ ไอซีไมโครคอนโทรลเลอร์ P89V51RD2FN\BN 13

4.1 ค่าอุณหภูมิที่ทำให้เครื่องปรับอากาศทำงานและไม่ทำงาน 31

4.2 ค่าอุณหภูมิที่ทำให้เครื่องปรับอากาศทำงานและไม่ทำงาน 32

4.3 ค่าความชื้นที่ทำให้พัดลมคอยล์เย็นเครื่องปรับอากาศทำงานและไม่ทำงาน 33

สารบัญรูป

รูปที่ หน้า

2.1 วงจรการทำงานของเครื่องปรับอากาศแบบอัดไอ 6
2.2 ไอซีไมโครคอนโทรลเลอร์ ขนาด 8 บิต 9
2.3 ไอซีไมโครคอนโทรลเลอร์แบบมีหน่วยความจำโปรแกรมที่เป็น EPROM อยู่ภายใน 9
2.4 ไอซีไมโครคอนโทรลเลอร์ของ ATMEL แบบแฟลช 10
2.5 ไอซีไมโครคอนโทรลเลอร์ของ ATMELที่ทำการโปรแกรมในระบบ
ISP (In-system Programming) 10
2.6 ไอซีไมโครคอนโทรลเลอร์ของ ATMEL แบบแฟลชขนาด 20 ขา 11
2.7 ไอซีไมโครคอนโทรลเลอร์ P89V51RD2FN\BN 11
2.8 ชื่อและตำแหน่งขาของไอซีไมโครคอนโทรลเลอร์ P89V51RD2FN\BN 13
3.1 แผนผังขั้นตอนการควบคุมการทำงานด้วยมือ 20
3.2 แผนผังขั้นตอนการควบคุมการทำงานแบบกึ่งอัตโนมัติ 21
3.3 วงจรควบคุมเครื่องปรับอากาศ 21
3.4 ภาพขยาย TC 22
3.5 บอร์ด EX-RELAY V3.0 ของบริษัทศิลารีเสิร์จ 22
3.6 วงจรไฟฟ้ากำลังของเครื่องปรับอากาศ 23
3.7 แผนภาพการไหลของโปรแกรม (Flow chard) 24
3.8 ตู้ควบคุมระบบไฟฟ้าของเครื่องปรับอากาศจำนวน 1ตู้ 24
3.9 เครื่องควบคุมอุณหภูมิและความชื้นที่สร้างขึ้นใหม่ 25
3.10 บ้านประหยัดพลังงานที่ได้ไปทำการติดตั้ง 25
3.11 ท่อดูดลมกลับ 27
3.12 ท่อปล่อยลมออก 27
3.13 แบบประกอบการติดตั้งหัววัดอุณหภูมิและความชื้น 28
4.1 ตู้ควบคุมระบบไฟฟ้าฯ พร้อมกับเครื่องควบคุมอุณหภูมิและความชื้นที่สร้างขึ้นใหม่ 30
4.2 การใช้คอมพิวเตอร์ในการเก็บข้อมูลอุณหภูมิและความชื้น 30
4.3 กราฟแสดงผลค่าอุณหภูมิและความชื้นภายใต้การใช้
เครื่องควบคุมอุณหภูมิ และความชื้นแบบเก่า 31

สารบัญรูป(ต่อ)

รูปที่ หน้า

4.4 กราฟแสดงผลค่าอุณหภูมิและความชื้นภายใต้การใช้
เครื่องควบคุมอุณหภูมิและความชื้นแบบใหม่ 32

ภาคผนวก

รูป ก Flow chard แสดงการทำงานของโปรแกรม 59
รูป ข การ Save file ไว้คนละ Editor 60
รูป ค การเข้า Folder ที่ถูกเก็บไว้ใน Tem&Humi 30 sec 60
รูป ง การลบข้อมูลที่มีอยู่เดิม 61
รูป จ การทำงานของโปรแกรม 61
รูป ฉ กราฟที่ได้จากการทำงานของโปรแกรม 62
รูป ช การหยุดโปรแกรม 62
รูป ซ การเก็บข้อมูล 63
รูป ฌ การเก็บข้อมูลในรูปของ Array 63

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ห้างหุ้นส่วนจำกัด เซมดีไซน์ (SAME DESIGN LIMITED PARTNERSHIP) เป็นผู้ดำเนินธุรกิจเกี่ยวกับการออกแบบ รับเหมาก่อสร้างอาคารประหยัดพลังงานและให้คำปรึกษาด้านการอนุรักษ์พลังงานและสิ่งแวดล้อม และโครงงานเครื่องควบคุมอุณหภูมิและความชื้นนี้ ได้ใช้บ้านที่ หจก.ฯ ก่อสร้างขึ้นมาใหม่ คือ บ้านพักอาศัย ค.ส.ล. 1 ชั้น มีขนาดพื้นที่ใช้งานประมาณ 130 ตารางเมตรนั้น โดยใช้เครื่องปรับอากาศชนิดแยกส่วนขนาด 18,000 บี.ที.ยู. จำนวน 2 เครื่อง ให้สลับกันทำงานในช่วงอุณหภูมิต่ำและทำงานพร้อมกันทั้งสองเครื่องในช่วงอุณหภูมิสูง การทำงานดังกล่าวได้ใช้เครื่องควบคุมอุณหภูมิแบบอิเล็กทรอนิกส์เทอร์โมสตัสที่มีขายตามท้องตลาดมาควบคุมวงจรไฟฟ้าของเครื่องปรับอากาศ ให้ควบคุมอุณหภูมิและความชื้นตามที่ต้องการ แต่เครื่องควบคุมฯ นี้ไม่สามารถควบคุมอุณหภูมิและความชื้นตามที่ต้องการได้ มีผลทำให้ค่าพลังงานไฟฟ้าที่ใช้กับเครื่องปรับอากาศสูงกว่ารายการคำนวณ ดังนั้นทางคณะผู้วิจัยจึงรับเป็นผู้ทำการศึกษาและออกแบบเครื่องดังกล่าว ให้สามารถควบคุมอุณหภูมิและความชื้นได้ตรงได้ตรงตามความต้องการของผู้ประกอบการ

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อสร้างเครื่องควบคุมอุณหภูมิและความชื้นต้นแบบ

1.2.2 เพื่อศึกษาประสิทธิภาพการทำงานของเครื่องควบคุมอุณหภูมิและความชื้น

1.2.3 เพื่อศึกษาความเป็นไปได้ในการนำไปใช้งานจริง

1.3 ขอบข่ายของโครงงาน

1.3.1 สร้างเครื่องควบคุมอุณหภูมิและความชื้นที่ใช้ไมโครคอนโทรลเลอร์

1.3.2 เครื่องควบคุมอุณหภูมิและความชื้นต้นแบบสามารถควบคุมอุณหภูมิให้อยู่ระหว่าง 20 - 30 องศาเซลเซียส และควบคุมความชื้นให้อยู่ระหว่างร้อยละ 60 - 70

1.3.3 สร้างโปรแกรมโดยใช้ภาษาซี

1.3.4 ทดสอบประสิทธิภาพการทำงานจริงของเครื่องควบคุมอุณหภูมิและความชื้น

1.4 ขั้นตอนการดำเนินโครงงาน

1.4.1 ศึกษาการทำงาน

1.4.1.1 ระบบเครื่องปรับอากาศชนิดแยกส่วน

1.4.1.2 วงจรไฟฟ้ากำลังควบคุมระบบเครื่องปรับอากาศ

1.4.1.3 ไมโครคอนโทรลเลอร์

1.4.1.4 หลักการเขียนโปรแกรมด้วยภาษาซี

1.4.2 การออกแบบ

1.4.2.1 วงจรไฟฟ้าควบคุมเครื่องปรับอากาศ

1.4.2.2 โปรแกรมการทำงานของเครื่องควบคุมอุณหภูมิและความชื้นด้วยภาษาซี

1.4.3 การสร้าง

1.4.3.1 ตู้ควบคุมระบบไฟฟ้าของเครื่องควบคุมอุณหภูมิและความชื้น จำนวน 1 ตู้

1.4.3.2 เครื่องควบคุมอุณหภูมิและความชื้นที่เหมาะสม จำนวน 1 เครื่อง

1.4.3.3 โปรแกรมควบคุมเครื่องควบคุมอุณหภูมิและความชื้น

1.4.4 การทดลอง

นำไปติดตั้งใช้งานจริงที่บ้าน ของ

คุณธวัชชัย กลิ่นรอด ตั้งอยู่ที่บ้านเลขที่ 89/1 บ้านบึงถัง ถนนเต็งหนาม - บึงถัง หมู่ที่ 3

ตำบลดอนทอง อำเภอเมือง จังหวัดพิษณุโลก 65000

1.5 แผนการดำเนินโครงงาน

กิจกรรม	ระยะเวลาดำเนินงาน (เดือน/2550-2551)							ผู้รับผิดชอบ/ผู้ปฏิบัติ	สถานที่ในการดำเนินงาน
	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.		
1.5.1 ศึกษาหลักการทำงาน								นายจักรพันธ์ หวาจ้อย นายปรัชญา จันทร์คามิ	
1.5.1.1 ระบบเครื่องปรับอากาศชนิดแยกส่วน	↔								2
1.5.1.2 วงจรไฟฟ้ากำลังควบคุมระบบเครื่องปรับอากาศ	↔	↔							1,2
1.5.1.3 ไมโครคอนโทรลเลอร์		↔	↔						1
1.5.1.4 หลักการเขียนโปรแกรมด้วยภาษาซี		↔	↔	↔					1
1.5.2. การออกแบบ									
1.5.2.1 วงจรไฟฟ้าควบคุมเครื่องปรับอากาศ	↔	↔							1,2
1.5.2.2 โปรแกรมการทำงานของเครื่องควบคุมอุณหภูมิและความชื้นด้วยภาษาซี			↔	↔	↔				1,2
1.5.3. การสร้าง									
1.5.3.1 ตู้ควบคุมระบบไฟฟ้าของเครื่องปรับอากาศจำนวน 1 ตู้					↔	↔			2,3
1.5.3.2 เครื่องควบคุมอุณหภูมิและความชื้น จำนวน 1 เครื่อง					↔	↔			1,2
1.5.3.3 โปรแกรมภาษาซีที่ใช้สำหรับควบคุมอุณหภูมิและความชื้น เพื่อสั่งการให้เครื่องปรับอากาศทำงาน						↔	↔		1
1.5.4. การทดลองโดยการนำไปติดตั้งงานจริงที่บ้านประหยัดพลังงาน						↔	↔		2,3
1.5.5. รวบรวมข้อมูลทั้งหมดจัดรูปเล่มพร้อมรายงาน	↔	↔	↔	↔	↔	↔	↔		1,2,3

1: ปฏิบัติที่ภาควิชา, 2 : ปฏิบัติที่ หจกฯ., 3 : ปฏิบัติที่บ้านประหยัดพลังงาน

1.6 ผลลัพธ์ที่คาดว่าจะได้รับ

1.6.1 เครื่องควบคุมอุณหภูมิและความชื้นต้นแบบ

1.7 งบประมาณของโครงงาน

ซึ่งเป็นงานวิจัยที่ได้รับทุนเป็นจำนวน 100,000 บาท ในการสร้างนวัตกรรมใหม่ๆ จากสำนักงานกองทุนสนับสนุนการทำวิจัย (สกว.) หรือ Industrial and Research Projects for Undergraduate Students (IRPUS) www.irpus.org โดยมีการรายละเอียดดังนี้

1.7.1 งบประมาณทั้งโครงงาน	100,000.00	บาท
1.7.2 จาก สกว.	100,000.00	บาท
ค่าตอบแทนอาจารย์ที่ปรึกษา	20,000.00	บาท
ทุนการศึกษาของนักศึกษา	20,000.00	บาท
ค่าวัสดุ	35,000.00	บาท
ค่าใช้สอย	15,000.00	บาท
อื่นๆ	10,000.00	บาท
รวมทั้งสิ้น	100,000.00	บาท

บทที่ 2
ทฤษฎีและหลักการทำงาน

2.1 หลักการทำงานของเครื่องทำความเย็นและปรับอากาศ

หลักการทำงานของเครื่องปรับอากาศ คือ การนำความร้อนจากภายในอาคารถ่ายเทออกสู่ภายนอกอาคารโดยใช้สารทำความเย็น เครื่องปรับอากาศมีส่วนประกอบที่สำคัญ 4 ส่วน คือ

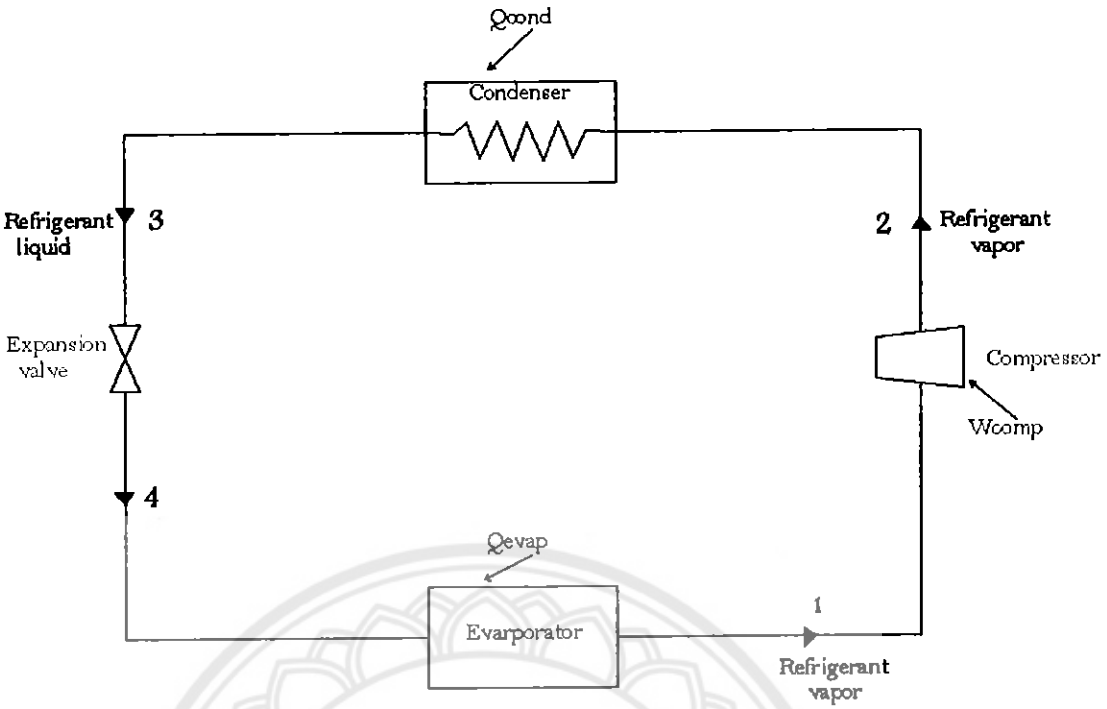
1. คอมเพรสเซอร์ (Compressor) ทำหน้าที่รับสารทำความเย็นที่มีสภาวะเป็นไอเย็นความดันต่ำ เข้ามาเพื่อทำการเพิ่มความดันให้เป็นสารทำความเย็นที่มีสภาวะเป็นไอร้อนความดันสูง

2. คอนเดนเซอร์ (Condenser) ทำหน้าที่รับสารทำความเย็นที่มีสภาวะเป็นไอร้อนความดันสูงมาระบายความร้อนออกเพื่อให้เป็นสารทำความเย็นที่มีสภาวะเป็นของเหลวที่มีความดันสูง

3. เอ็กแพนชั่นวาลว์ (Expansion Valve) ทำหน้าที่รับสารทำความเย็นที่มีสภาวะเป็นของเหลวความดันสูงมาลดแรงดัน แล้วฉีดออกเป็นละอองเพื่อให้เป็นสารทำความเย็นที่มีสภาวะเป็นของเหลวความดันต่ำ

4. อีวาพอเรเตอร์ (Evapporator) ทำหน้าที่รับความร้อนจากระบบมาถ่ายเทให้กับละอองสารทำความเย็น ทำให้ละอองสารทำความเย็นระเหยกลายเป็นสารทำความเย็นที่มีสภาวะเป็นไอเย็นความดันต่ำ

จากรูปที่ 2.1 การทำงานของวัฏจักรการทำความเย็นแบบอัดไอ เริ่มจากคอมเพรสเซอร์ดูดสารทำความเย็นซึ่งอยู่ในสภาวะไอที่จุดที่ (1) และอัดเพื่อให้มีความดันไอสูงขึ้นตามกระบวนการไอเซนทรอปิก (Isentropic Process) เพื่อที่จะทำให้สารทำความเย็นที่จุดที่ (2) จากนั้นสารทำความเย็นที่เป็นแก๊สจะถูกส่งมาถ่ายเทความร้อนที่คอนเดนเซอร์ เพื่อให้สารทำความเย็นกลั่นตัวตามขบวนการคลายความร้อนออก สารทำความเย็นก็จะควบแน่น จนกลายเป็นสารทำความเย็นเหลวที่มีความดันสูงที่จุด (3) จากนั้นถูกส่งผ่านเอ็กเพนชั่นวาล์วเพื่อลดความดันไอและเข้าอีวาปโปเรเตอร์ที่จุดที่ (4) ซึ่งในอีวาปโปเรเตอร์สารทำความเย็นเหลวจะเริ่มเปลี่ยนสถานะกลายเป็นไอ



รูปที่ 2.1 วงจรการทำงานของเครื่องปรับอากาศแบบอัดไอ

2.2 วงจรไฟฟ้ากำลังควบคุมระบบเครื่องปรับอากาศ

ศึกษาการทำงานของอุปกรณ์ที่เกี่ยวข้องแต่ละชนิด เช่น

2.2.1 เซอร์กิตเบรกเกอร์ (Circuit Breaker) มี 2 ประเภท ได้แก่

- เซอร์กิตเบรกเกอร์แบบผกผัน (Inverse Time Circuit Breaker) **ได้แก่ เซอร์กิตเบรกเกอร์ที่ใช้ทั่วไป**

- เซอร์กิตเบรกเกอร์แบบปลดทันที (Instantaneous Circuit Breaker) **แบบนี้เหมาะสำหรับ**วงจรมอเตอร์ สำหรับการเลือกใช้ขนาดของฟิวส์และเซอร์กิตเบรคเกอร์ จะต้องเลือกให้เหมาะสมกับพิกัดของมอเตอร์

2.2.2 แมกเนติกคอนแทคเตอร์ (Magnetic Contactor) เป็นอุปกรณ์ที่อาศัยการทำงานโดยอำนาจแม่เหล็กในการเปิดปิดหน้าสัมผัสในการควบคุมวงจรมอเตอร์หรือเรียกว่าสวิตช์แม่เหล็ก (Magnetic Switch) หรือแมกเนติกคอนแทคเตอร์ (Magnetic Contactor) ก็ได้

2.2.3 โอเวอร์โหลดรีเลย์ (Overload Relay) เป็นอุปกรณ์ป้องกันมอเตอร์ทำงานเกินกำลังหรือป้องกันมอเตอร์ไม่ให้เกิดการเสียหาย เมื่อมีกระแสไหลเกินพิกัดในมอเตอร์

2.2.4 รีเลย์ช่วย (Auxiliary Relay) หรืออาจเรียกว่า รีเลย์ควบคุม (Contol Relay) การทำงานอาศัยอำนาจในการเปิด-ปิดหน้าสัมผัสเหมือนกับหลักการทำงานของแมกเนติกคอนแทคเตอร์ ต่างกันตรงที่รีเลย์ช่วยจะทำงานในวงจรควบคุม แต่แมกเนติกคอนแทคเตอร์จะทำงานในวงจรไฟฟ้ากำลัง

2.2.5 แลทชิ่งรีเลย์ (Latching Relay) คอนแทคจะเปลี่ยนสภาวะจากตำแหน่งเดิมไปตำแหน่งใหม่ ทุกครั้งที่มีอินพุทเข้ามา และจะคงอยู่ในตำแหน่งการใช้งานจนกระทั่งมีอินพุทใหม่มากระตุ้นอีกครั้ง

2.2.6 สวิทช์เลือกการทำงาน (Selector Switch) เป็นสวิตช์สำหรับตัดต่อวงจรไฟฟ้าจากวงจรหนึ่งไปยังอีกวงจรหนึ่ง หรืออีกหลายๆ วงจร

2.2.7 หลอดแสดงสัญญาณ (Pilot Lamp) เป็นหลอดไฟที่ใช้แสดงสถานะการทำงานของวงจร หรืออุปกรณ์ไฟฟ้าที่ต้องการควบคุม เช่น มอเตอร์ เป็นต้น และหลอดแสดงสัญญาณนี้มีหลายสีหลายแบบ

2.2.8 สวิทช์ปุ่มกด (Push Button Switch) อุปกรณ์ที่มีหน้าสัมผัสอยู่ภายใน การเปิด-ปิดหน้าสัมผัสทำได้โดยใช้นิ้วมือกดสวิทช์ปุ่มกดที่ต้องการให้เริ่มทำงาน (Start) เรียกว่า สวิทช์ปกติในสภาวะเปิด (Normally Open ; NO) สวิทช์ปุ่มกดที่ต้องการให้หยุดการทำงาน (Stop) เรียกว่า สวิทช์ปกติในสภาวะปิด (Normally Close; NC)

2.3 หลักการเขียนโปรแกรมด้วยภาษาซี

การเขียนโปรแกรมจะต้องเข้าใจหลักการ 5 ขั้นตอน ดังต่อไปนี้

2.3.1 ทำความเข้าใจและวิเคราะห์ปัญหา

การทำความเข้าใจและการวิเคราะห์ปัญหาเป็นสิ่งที่สำคัญ เพราะผู้เขียนโปรแกรมจะต้องวิเคราะห์ปัญหาร่วมกับนักวิเคราะห์ระบบว่าโจทย์ต้องการผลลัพธ์อะไร การที่จะได้มาซึ่งผลลัพธ์นั้นต้องป้อนข้อมูลอะไรบ้าง และเมื่อป้อนข้อมูลเข้าไปแล้วจะทำการประมวลผลอย่างไร สิ่งเหล่านี้ผู้เขียนโปรแกรมจะต้องทำความเข้าใจให้ถูกต้อง เพราะถ้าผู้เขียนโปรแกรมวิเคราะห์ปัญหาไม่ถูกต้อง ผลลัพธ์ก็อาจจะไม่ตรงกับความต้องการของโจทย์ได้

2.3.2 กำหนดแผนในการแก้ไขปัญหา

ปัญหาการเขียนผังงาน คือ การเขียนแผนภาพที่เป็นลำดับ เพื่อแสดงขั้นตอนการทำงานของโปรแกรมเพื่อให้ง่ายต่อการทำความเข้าใจ การเขียนผังงานมี 3 แบบ คือ แบบเรียงลำดับ (Sequential) แบบมีการกำหนดเงื่อนไข (Condition) และ แบบมีการทำงานวนรอบ (Looping) สัญ - ลักษณ์ของผังงาน (FlowchartSymbol)

2.3.3 เขียนโปรแกรมตามแผนที่กำหนด

เมื่อเขียนผังงานเสร็จเรียบร้อยแล้ว ขั้นตอนต่อไป คือ การเขียนโปรแกรมตามผังงานที่ได้กำหนดเอาไว้ ในกรณีที่เขียนด้วยการเขียนโปรแกรมก็ต้องเป็นไปตามกฎเกณฑ์และโครงสร้างของภาษาซี

2.3.4 ทดสอบและตรวจสอบความถูกต้อง

เมื่อเขียนโปรแกรมเสร็จให้ทดลองคอมไพล์โปรแกรมว่ามีจุดผิดพลาดที่ใดบ้างใน การคอมไพล์โปรแกรมจะใช้วิธีการกดปุ่ม Alt + F9 ในกรณีที่มีข้อผิดพลาดจะแสดงในช่องด้านล่างของหน้าจอเอดิเตอร์ในส่วนของกรอบ message ให้อ่านทำความเข้าใจ และแก้ไขตามที่โปรแกรมแจ้งข้อมูลผิดพลาด เมื่อเสร็จแล้วให้ทดลองรันโปรแกรม

2.3.5 นำโปรแกรมที่ผ่านการทดสอบไปใช้งาน

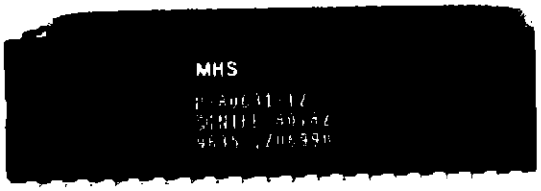
ถ้ารันโปรแกรมแล้วใช้งานได้แสดงว่าจะได้ไฟล์ที่มีส่วนขยายเป็นนามสกุล .HEX เพื่อนำไปทดสอบใช้งานในที่ต่างๆ ถ้านำไปใช้งานแล้วมีปัญหาก็ให้แก้ไขอีกครั้ง แต่ถ้าไม่มีปัญหาแสดงว่าโปรแกรมนี้ใช้งานได้อย่างสมบูรณ์

2.4 ไมโครคอนโทรลเลอร์

เป็นอุปกรณ์ไอซี (IC: Integrated Circuit) ที่สามารถโปรแกรมการทำงานได้ซับซ้อน สามารถรับข้อมูลในรูปสัญญาณดิจิตอลเข้าไปทำการประมวลผลแล้วส่งผลลัพธ์ข้อมูลดิจิตอลออกมาเพื่อนำไปใช้งานตามที่ต้องการได้ ไมโครคอนโทรลเลอร์เป็นไมโครโพรเซสเซอร์ชนิดหนึ่งเช่นเดียวกับหน่วยประมวลผลกลาง (CPU: Central Processing Unit) ที่ใช้ในคอมพิวเตอร์ แต่ได้รับการพัฒนารวมวงจรที่จำเป็น เช่น หน่วยความจำ, ส่วนอินพุต และเอาต์พุต บางส่วนเข้าไปในไอซีตัวเดียวกัน และเพิ่มวงจรบางอย่างเข้าไปเพื่อให้มีความสามารถเหมาะสมกับการใช้ในงานควบคุม เช่น วงจรตั้งเวลา, วงจรการสื่อสารอนุกรม และวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล เป็นต้น

ไมโครคอนโทรลเลอร์มีหลายยี่ห้อ หลายตระกูล และหลายเบอร์ด้วยกัน ซึ่งแต่ละเบอร์ก็จะมีโครงสร้างภายในและความสามารถในการทำงานที่แตกต่างกัน ไมโครคอนโทรลเลอร์ที่นิยมใช้งานคือ MCS51, PIC และ AVR เป็นต้น

ไมโครคอนโทรลเลอร์ในตระกูล MCS51 ที่มีหน่วยประมวลผลกลางขนาด 8 บิต และเป็นหน่วยประมวลผลที่สามารถ เข้าถึงข้อมูลในระดับบิต มีหน่วยความจำ ข้อมูลพื้นฐานที่เป็นแบบแรม มีขาติดต่อที่เป็นแบบสองทิศทาง สามารถใช้งานโดยกำหนดให้เป็นได้ทั้งอินพุต และเอาต์พุต มีวงจรสื่อสารข้อมูลอนุกรมแบบ ฟูลดูเพล็กซ์ รวมทั้งส่วนของวงจรตัวนับจำนวน และตัวตั้งเวลาอยู่ภายในตัวไอซี แต่โครงสร้างไอซีแบบนี้ก็ยังจะต้องมีหน่วยความจำโปรแกรมภายนอก ซึ่งทำหน้าที่กำหนดลำดับคำสั่งการทำงานของไอซี จากคุณสมบัติดังกล่าวเราจะเรียกว่าไอซีไมโครคอนโทรลเลอร์ แสดงได้ดังรูป 2.2 จะเป็น ไอซี ไมโครคอนโทรลเลอร์ของบริษัทอินเทลเบอร์ 80C31 ซึ่งไม่มีหน่วยความจำโปรแกรมภายใน แต่สำหรับไอซีเบอร์ 8051 จะมีหน่วยความจำโปรแกรมที่อยู่ภายในตัวไอซี ซึ่งเป็นหน่วยความจำแบบรอม (ROM) และบรรจุข้อมูลคำสั่งมาจากบริษัท ผู้ผลิตแล้วเราไม่สามารถที่จะทำการแก้ไขข้อมูลได้ ดังนั้นไอซีไมโครคอนโทรลเลอร์เบอร์นี้จึงไม่ค่อยนิยมนำมาใช้ในการออกแบบชิ้นงานที่มีจำนวนไม่มากนัก



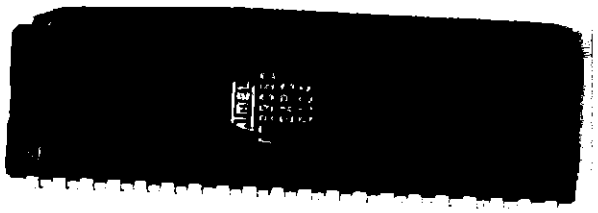
รูป 2.2 ไอซีไมโครคอนโทรลเลอร์ ขนาด 8 บิต

หลังจากนั้นได้มีการพัฒนาเป็นไอซีไมโครคอนโทรลเลอร์ ที่บรรจุหน่วยความจำโปรแกรมอยู่ภายในแพคเกต (Packet) เดียวกันกับไอซีไมโครคอนโทรลเลอร์ และเป็นหน่วยความจำโปรแกรมแบบ EPROM *(Eraser programs read only memory/) ที่มีคุณสมบัติสามารถเขียน ข้อมูลลงไปในตัวไอซีได้ง่ายขึ้น โดยการใช้เครื่องโปรแกรมสำหรับอัดเก็บข้อมูลลงในหน่วยความจำแบบ EPROM และยังสามารถที่จะลบล้างข้อมูล โดยใช้วิธีการฉายด้วยแสงอัลตร้าไวโอเลต ให้กับตัวไอซีในระยะเวลาหนึ่งจะทำให้ไอซีหน่วยความจำ ที่อยู่ภายในไอซีไมโครคอนโทรลเลอร์ตัวเดิมสามารถที่จะนำกลับมาป้อนข้อมูลใช้งานได้ใหม่อีก

ดังนั้นคำสั่งที่เราจะกำหนดการทำงานของไอซีไมโครคอนโทรลเลอร์สามารถที่จะนำไปเก็บไว้ในหน่วยความจำโปรแกรมแบบ EPROM ในตัวไอซีได้เลย ส่งผลให้การนำไอซีไมโครคอนโทรลเลอร์ไปใช้งานมีอุปกรณ์ที่ต่อร่วมในวงจรน้อยลงไป ไอซีไมโครคอนโทรลเลอร์ที่มีคุณสมบัติดังกล่าว ตัวอย่างเช่นไอซีเบอร์ 8751 ซึ่งแสดงได้ดังรูป 2.3

รูป 2.3 ไอซีไมโครคอนโทรลเลอร์แบบมีหน่วยความจำโปรแกรมที่เป็น EPROM อยู่ภายใน

ต่อมาความต้องการใช้งานไอซีไมโครคอนโทรเลอร์มีมากขึ้น รวมทั้งการผลิตชิ้นงานในโรงงาน และทางด้านการศึกษา จึงได้พัฒนาเป็นไอซีไมโครคอนโทรลเลอร์ที่เพิ่มคุณสมบัติของหน่วยความจำโปรแกรมภายในเป็นแบบแฟลช (Flash Memory) ซึ่งมีคุณสมบัติสามารถจะลบล้างข้อมูลได้โดยใช้สัญญาณไฟฟ้า และมีการโปรแกรมใหม่ได้ถึงกว่า 1000 ครั้ง ในการลบข้อมูลแต่ละครั้งจะใช้เวลาไม่ถึง 1 นาที โดยจะนำไอซีไปโปรแกรมข้อมูล โดยผ่านเครื่องโปรแกรมการใช้งานก็สะดวก รวดเร็วมากขึ้น (หากใช้เครื่องล้างข้อมูลจากแสงอัลตร้าไวโอเลต จะต้องใช้เวลาไม่ต่ำกว่า 10 นาที เพื่อจะล้างข้อมูลในหน่วยความจำ แบบ EPROM ภายในไอซี) และที่สำคัญคำสั่งในการใช้งานของไอซียังคงเหมือนเดิม ไอซีไมโครคอนโทรลเลอร์ที่มี คุณสมบัติดังกล่าวคือเบอร์ AT89C51, AT89C52, AT89C55 แสดงได้ดังรูป 2.4



รูป 2.4 ไอซีไมโครคอนโทรลเลอร์ของ ATMEL แบบแฟลช

การเพิ่มคุณสมบัติทางด้านเทคโนโลยีของหน่วยความจำภายในไอซีไมโครคอนโทรลเลอร์ ในการที่จะโปรแกรม ข้อมูลลงตัวไอซี การลบล้างข้อมูล และการพัฒนาระบบเพื่อให้สะดวก รวดเร็วบนเครื่องไมโครคอมพิวเตอร์ส่วนบุคคล ดังนั้นจึงมีการพัฒนาให้ไอซีไมโครคอนโทรลเลอร์ สามารถที่จะทำการโปรแกรมข้อมูลลงในหน่วยความจำโปรแกรมได้ โดยที่ไม่ต้องถอดตัวไอซีไมโครคอนโทรลเลอร์ออกมาทำการโปรแกรมข้างนอกบอร์ดอีก วิธีการนี้เราเรียกว่าการโปรแกรมในระบบ ISP (In-system programming) ซึ่งไม่ต้องใช้เครื่องโปรแกรมที่มีวงจรซับซ้อน และราคาแพง ดังนั้นในการที่จะปรับปรุงและพัฒนาข้อมูลในหน่วยความจำโปรแกรมที่อยู่ภายในตัวไอซี จึงทำได้อย่างสะดวก และรวดเร็ว ตัวอย่างเช่นไอซีไมโครคอนโทรลเลอร์เบอร์ 89S8252 แสดงได้ดังรูป 2.5

รูป 2.5 ไอซีไมโครคอนโทรลเลอร์ของ ATMELที่ทำการโปรแกรมในระบบ ISP (In-system programming)

ขนาดของตัวไอซีไมโครคอนโทรลเลอร์ก็เป็นส่วนหนึ่งของการพัฒนาใช้งาน ดังนั้นจึงมีการลดขนาดของไอซี ให้มีจำนวนของขาใช้งานให้น้อยลงเหลือเพียงขนาด 20 ขา ที่ไว้ใช้งานในกรณีที่ไม่มีความจำเป็นจะต้องใช้จำนวนพอร์ตมาก โดยจะกำหนดให้เหลือเพียง 2 พอร์ต และลดคุณสมบัติบางอย่างลงไป แต่ยังมีคุณสมบัติที่เป็นพื้นฐานเพื่อใช้งานอยู่ครบ คำสั่งและส่วนอื่นๆที่สำคัญก็ยังจะคงเดิม และสามารถที่จะเลือกขนาด เนื้อที่ของหน่วยความจำโปรแกรมเป็นแบบแฟลช และคุณสมบัติส่วนอื่นๆ เพิ่มเติมได้โดยเลือกตามเบอร์ของไอซี เช่นไอซีไมโครคอนโทรลเลอร์เบอร์ AT89C1051 ,AT89C2051 และไอซีเบอร์ AT89C4051 จะมีหน่วยความจำโปรแกรมภายในขนาด 1 Kbytes , 2 Kbytes และ 4 Kbytes ตามลำดับ แสดงได้ดังรูป2.6

รูป 2.6 ไอซีไมโครคอนโทรลเลอร์ของ ATMEL แบบแฟลชขนาด 20 ขา

ดังนั้นจะเห็นได้ว่าการพัฒนาเทคโนโลยีของการผลิตชิฟไอซีได้ก้าวหน้าไปอย่างรวดเร็ว จนถึงในปัจจุบันก็ยังมีไอซีไมโครคอนโทรลเลอร์อีกมากมาย หลายแบบ หลายเบอร์ที่ผลิตขึ้นมา และมีคุณสมบัติอื่นๆ ที่เพิ่มเติมในตัวแต่ไม่ได้กล่าวถึงในที่นี้ แม้กระทั่งภาษาที่จะนำมาเขียนสั่งงานให้กับไอซี ไมโครคอนโทรลเลอร์ เพื่อประมวลผลข้อมูลตามคำสั่ง ก็มีการพัฒนาตามขึ้นไปด้วย เช่น โปรแกรมที่เป็นภาษา C ภาษา BASIC เพื่อให้สะดวกในการใช้งานสำหรับผู้ที่ไม่ชอบเขียนภาษาแอสแซมบลี้ ดังนั้นในการที่เราจะนำไอซีไมโครคอนโทรลเลอร์ตัวใด เบอร์ไหน บริษัทใด หรือโปรแกรมแบบไหนที่จะนำไปใช้งาน จึงขึ้นอยู่กับงานที่เราต้องการออกแบบและข้อมูลที่เราจะสะดวกในการค้นคว้าสร้างชิ้นงาน ที่สำคัญคือต้นทุนในการสร้าง ส่วนข้อดีข้อเสียของแต่ละแบบ ก็คือการแข่งขันของแต่ละบริษัทที่จะต้องผลิตให้ดีกว่ากัน

คุณสมบัติของไมโครคอนโทรลเลอร์ P89V51RD2FN/BN

รูปที่ 2.7ไอซีไมโครคอนโทรลเลอร์ P89V51RD2FN\BN

เป็นไมโครคอนโทรเลอร์ตระกูล MCS – 51 ซึ่งมีหน่วยความจำภายในเป็นแบบแฟลช (Flash memory) ของ Philips Semiconductor ที่มีคุณสมบัติที่โดดเด่นทางเทคนิคดังนี้

- เป็นไมโครคอนโทรลเลอร์ 8 บิต ที่เข้ากันได้กับไมโครคอนโทรลเลอร์ MCS – 51 พื้นฐาน

- หน่วยความจำโปรแกรมภายในตัวไมโครคอนโทรลเลอร์เป็นแบบแฟลช ลบและเขียนใหม่ได้ถึงหนึ่งหมื่นครั้ง ขนาดของหน่วยความจำสูงถึง 64 กิโลไบต์

- หน่วยความจำข้อมูลแรมภายในมีขนาด 1 กิโลไบต์

- โปรแกรมข้อมูลลงในหน่วยความจำโปรแกรมแบบ ในระบบ (ISP : In – system programming)

- ความถี่สัญญาณนาฬิกาสูงสุด 40 MHz ในกรณีทำงานด้วยสัญญาณนาฬิกาภายใน 12 ลูกต่อแมชชีนไซเคิลและ 20 MHz ในกรณีทำงานด้วยสัญญาณนาฬิกาภายใน 6 ลูกต่อแมชชีนไซเคิล

P89V51RD2 ได้รับการกำหนดให้ทำงานเบื้องต้นในโหมดสัญญาณนาฬิกา 12 ลูกต่อแมช - ชีนไซเคิล สามารถเลือกเปลี่ยนเป็น 6 สัญญาณนาฬิกาต่อแมชชีนไซเคิลได้ด้วยการโปรแกรมแบบ ISP แต่เมื่อเปลี่ยนแล้ว หากต้องการเปลี่ยนกลับเป็น 12 สัญญาณนาฬิกาต้องใช้การโปรแกรมแบบขนาน

- ขาพอร์ต 8 บิต 4 พอร์ต แบบกึ่งสองทิศทาง (quasi - bidirectional) เป็นได้ทั้งอินพุตและเอาต์พุต

- อุปกรณ์เพอริเฟอรัลภายในไมโครคอนโทรลเลอร์สามารถทำงานด้วยความเร็ว 12 สัญญาณนาฬิกาต่อแมชชีนไซเคิลได้ แม้ว่าซีพียูจะทำงานด้วยความเร็ว 6 สัญญาณนาฬิกาภายในต่อแมชชีนไซเคิล

- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์

- ไทเมอร์/เคาน์เตอร์ขนาด 16 บิต 3 บิต (ไทเมอร์ 0, 1 และ 2)

- มีรีจิสเตอร์ตัวชี้ตำแหน่งข้อมูลหรือ DPTR 2 ตัว

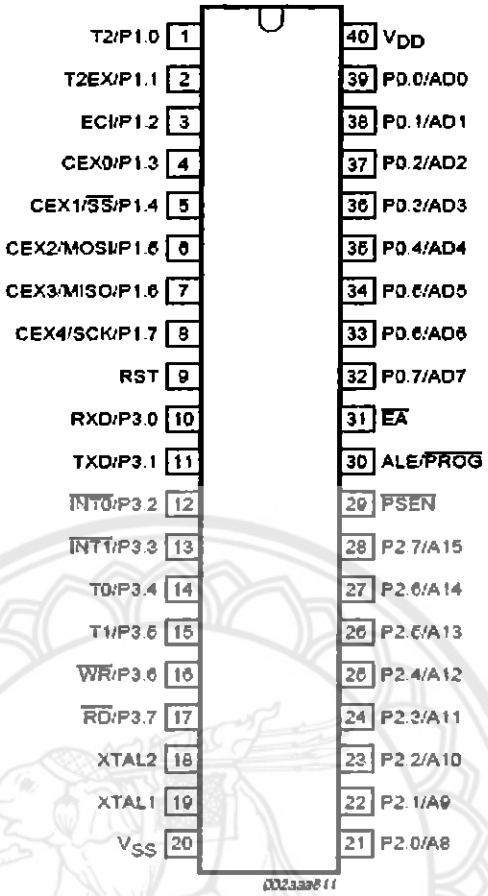
- สามารถรองรับแหล่งแหล่งกำเนิดอินเตอร์รัปต์ได้ 4 ระดับ

- กำหนดนัยสำคัญของการตอบสนองอินเตอร์รัปต์ได้ 4 ระดับ

- สามารถติดต่อหน่วยความจำภายนอกได้สูงสุด 64 กิโลไบต์

- มีวอตช์ด็อกไทเมอร์

- มีโมดูลวงจรนับโปรแกรมได้ (PCA : Programmable Counter Array) ซึ่งบรรจุวงจรตรวจจับสัญญาณ (capture), เปรียบเทียบสัญญาณนาฬิกา (compare), และวอตช์ด็อกไทเมอร์ (watchdog timer)



รูปที่ 2.8 ชื่อและตำแหน่งขาของไอซีไมโครคอนโทรลเลอร์ P89V51RD2FN\BN

ตารางที่ 2.1 ชื่อขา ตำแหน่งขา ชนิดขา และรายละเอียดการทำงานของไอซี ไมโครคอนโทรลเลอร์ P89V51RD2FN\BN

ชื่อขา	ขาที่	ชนิด	หน้าที่และการทำงาน
V_{DD}	40	อินพุต	ต่อไฟเลี้ยง +5V
V_{SS} /GND	20	อินพุต	ต่อกราวด์
P0.0 -P0.7	39-32	อินพุต/เอาต์พุต	- ใช้งานเป็นขาพอร์ตอินพุตเอาต์พุต ถ้าต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ทำให้มีสถานะลอย (float) ค่าอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นพอร์ตขาอินพุตได้ - ใช้ในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0 - A7) และขาข้อมูล (D0 - D7) โดยใช้การมัลติเพล็กซ์เข้าช่วยเพื่อสลับการทำงาน

ชื่อขา	ขาที่	ชนิด	หน้าที่และการทำงาน
			ให้เป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูลในการติดต่อกับหน่วยความจำภายนอก
P1.0-P1.7	1-8	อินพุต/เอาต์พุต	- ใช้งานเป็นขาพอร์ตอินพุตและเอาต์พุตสำหรับการใช้งานทั่วไป เฉพาะขา P1.5-P1.7 สามารถขับกรแสได้สูง 16 mA ต่อขา - เป็นขาสัญญาณของไทเมอร์ 2 และขาสัญญาณของโมดูล PCA ดังมีรายละเอียดต่อไปนี้ T2(P1.0 : ขา 1) เป็นขาอินพุตสำหรับนับค่าของไทเมอร์ 2 และขาเอาต์พุตสัญญาณนาฬิกาโปรแกรมได้ T2EX(P1.1 : ขา 2) เป็นขาอินพุตสำหรับควบคุมการทำงานของไทเมอร์/เคาน์เตอร์ 2 ECI(P1.2 : ขา 3) เป็นขาอินพุตสัญญาณนาฬิกาจากภายนอกสำหรับโมดูล PCA CEX0(P1.3 : ขา 4) เป็นขาอินพุตเอาต์พุตภายนอกของวงจรตรวจจับและเปรียบเทียบสัญญาณสำหรับ PCA โมดูล 0 CEX0(P1.4 : ขา 5) เป็นขาอินพุตเอาต์พุตภายนอกของวงจรตรวจจับและเปรียบเทียบสัญญาณสำหรับ PCA โมดูล 1 CEX0(P1.5 : ขา 6) เป็นขาอินพุตเอาต์พุตภายนอกของวงจรตรวจจับและเปรียบเทียบสัญญาณสำหรับ PCA โมดูล 2 CEX0(P1.6 : ขา 7) เป็นขาอินพุตเอาต์พุตภายนอกของวงจรตรวจจับและเปรียบเทียบสัญญาณสำหรับ PCA โมดูล 3 CEX0(P1.7 : ขา 8) เป็นขาอินพุตเอาต์พุตภายนอกของวงจรตรวจจับและเปรียบเทียบสัญญาณสำหรับ PCA โมดูล 4
P2.0-P2.7	21-28	อินพุต/เอาต์พุต	- ใช้งานเป็นขาพอร์ตอินพุตเอาต์พุตสำหรับใช้งานทั่วไป - ใช้ต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8-A15) เมื่อติดต่อด้วย

ชื่อขา	ขาที่	ชนิด	หน้าที่และการทำงาน
P3.0-P3.7	10-17	อินพุต/เอาต์พุต	- ใช้งานเป็นขาพอร์ตอินพุตเอาต์พุตสำหรับใช้งานทั่วไป - ใช้งานเป็นขาพอร์ตหน้าที่พิเศษ ดังมีรายละเอียดดังต่อไปนี้ $\overline{\text{RxD}}$ (P3.0 : ขา 10) ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม $\overline{\text{TxD}}$ (P3.1 : ขา 11) ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม INT0(P3.2 : ขา 12) ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเทอร์รัปต์จากภายนอกช่อง 0 INT1(P3.3 : ขา 13) ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเทอร์รัปต์จากภายนอกช่อง 1 T0(P3.4 : ขา 14) ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 0 T1(P3.5 : ขา 15) ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 1 $\overline{\text{WR}}$ (P3.6 : ขา 16) ใช้เป็นขาสัญญาณ $\overline{\text{WR}}$ ในกรณีที่เชื่อมต่อกับหน่วยความจำภายนอก $\overline{\text{RD}}$ (P3.6 : ขา 16) ใช้เป็นขาสัญญาณ $\overline{\text{RD}}$ ในกรณีที่เชื่อมต่อกับหน่วยความจำภายนอก
RESET	9	อินพุต	ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณลอจิก "1" อย่างน้อยเป็นเวลา 2 แมชชีนไซเคิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างปกติ
ALE	30	เอาต์พุต	Address Latch Enable ออกมาทุกๆ แมชชีนไซเคิล อย่างไรก็ตามสามารถ disable สัญญาณพัลส์นี้ได้ โดยการเซตบิต 0 ของรีจิสเตอร์ AUXR
$\overline{\text{PSEN}}$	29	เอาต์พุต	Program Store Enable : ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ไมโครคอนโทรลเลอร์จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้ง นอกจากนี้ยังใช้ประกอบในการอ่าน-เขียนข้อมูลใน

ชื่อขา	ขาที่	ชนิด	หน้าที่และการทำงาน
			หน่วยความจำโปรแกรมด้วยกระบวนการ ISP - สำหรับเบอร์ P89C51RD+ ให้ต่อขานี้ลงกราวด์ แล้วป้อนไฟ +12V (±0.5 V) เข้าที่ขา $\overline{EA}$ / V_{PP} - สำหรับเบอร์ P89C51RD2 ให้ต่อขานี้ลงกราวด์, ป้อนลอจิก "1" เข้าที่ขา P2.7 และป้อนแรงดัน +5V เข้าที่ขา $\overline{EA}$ / V_{PP}
$\overline{EA}$ / V_{PP}	31	อินพุต	External Access enable/Programming voltage input : ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ "0" เลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก "1" เลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายใน นอกจากนี้ที่ขานี้ยังใช้เป็นขาอินพุตสำหรับรับแรงดันสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ - สำหรับเบอร์ P89C51RD+ ต้องการแรงดัน +12V (±0.5 V) - สำหรับเบอร์ P89C51RD+ ต้องการแรงดัน +5V
XTAL1	19	อินพุต	ขาอินพุตรับสัญญาณจากวงจรขยายออสซิลเลเตอร์ (ขา XTAL2) และจากภายนอก ในการใช้งานปกติ ขานี้และขา XTAL2 ต่อเข้ากับคริสตอลและตัวเก็บประจุชดเชยค่าน้อยๆ
XTAL2	18	เอาต์พุต	ขาเอาต์พุตของวงจรขยายออสซิลเลเตอร์ภายในไมโครคอนโทรลเลอร์ ในการใช้งานปกติ ขานี้และขา XTAL1 ต่อเข้ากับคริสตอลและตัวเก็บประจุชดเชยค่าน้อยๆ

2.5 การเขียนโปรแกรมไมโครคอนโทรลเลอร์

รูปแบบการเขียนโปรแกรมไมโครคอนโทรลเลอร์สามารถแบ่งได้ 3 แบบคือ

2.5.1 เขียนด้วยภาษาแอสเซมบลี (Assembly) แบบไฟล์เดียว หลังจากนั้นจะทำการคอมไพล์ด้วย Assembler ของ ไมโครคอนโทรลเลอร์ตัวนั้นโดยไฟล์ที่ได้มา มีได้หลายชนิดแต่ส่วนใหญ่จะอยู่ในรูปของ Hex file

2.5.2 ใช้ภาษา Assembly แต่แบ่งเป็นหลายๆ ไฟล์ หลังจากนั้นจะทำการคอมไพล์ แต่ละไฟล์ให้ออกมาเป็น Object files และทำการรวมกันด้วย Linker ในขณะทำการ link ก็จะมี script file ของไมโครคอนโทรลเลอร์ตัวนั้นประกอบ หลังจากทำการ Link แล้วไฟล์ที่ได้จะอยู่ในรูป Hex file

2.5.3 การเขียนด้วยภาษาสูง โดยทั่วไปภาษาที่ใช้เขียนจะเป็นภาษาซี (C) หรือภาษาเบสิก (Basic) เป็นต้น ซึ่งอาจจะเขียนร่วมกับภาษา Assembly โดยไฟล์ที่เขียนจะถูกทำให้กลายเป็น Object files โดย Assembler สำหรับภาษา Assembly และคอมไพล์ โดยตัวคอมไพล์สำหรับภาษาสูง จากนั้นก็ทำการ Link เข้าด้วยกันด้วย Linker ซึ่งขณะทำการ Link ก็จะมีการรวมเอา Library ที่ถูกเรียกใช้ในโปรแกรมเข้าไปรวมด้วยกัน สุดท้ายจะอยู่ในรูป Hex file หลังจากได้ Hex file แล้วจะทำการอัดโปรแกรมเข้าสู่ชิพด้วยตัวโปรแกรมเมอร์ส่วนใหญ่จะมีรูปแบบคือ มี Software บนคอมพิวเตอร์ สำหรับใช้ในการควบคุมการอ่าน เขียน หรือ ลบ โดยส่วนใหญ่จะเชื่อมต่อไปยัง Programmer ด้วยพอร์ตอนุกรม หรือพอร์ตขนาน เมื่ออัดโปรแกรมเข้าไมโครคอนโทรลเลอร์ก็สามารถนำไปใช้งานตามที่ได้ออกแบบไว้

2.6 โปรแกรมแมทแล็บ

MATLAB เป็นภาษาคอมพิวเตอร์ขั้นสูง (High Level Language) ใช้สำหรับการคำนวณทางคณิตศาสตร์ที่ประกอบไปด้วยการคำนวณเชิงตัวเลข กราฟิกที่ซับซ้อน และการจำลองแบบเพื่อให้มองเห็นภาพพจน์ได้ง่ายและชัดเจน ชื่อของ MATLAB ย่อมาจาก matrix laboratory เดิมโปรแกรม MATLAB ได้เขียนขึ้นเพื่อใช้คำนวณทาง matrix หรือเป็น matrix software ที่พัฒนาจากโปรเจ็คที่ชื่อ LINKPACK และ EISPACK

MATLAB ได้พัฒนามาจากปัญหาที่ส่งมาจากหลายๆ ผู้ใช้งาน จึงทำให้ MATLAB มีฟังก์ชันต่างๆให้เลือกใช้อยู่มากมาย ในบางมหาวิทยาลัยได้ใช้โปรแกรม MATLAB เป็นหลักสูตรพื้นฐานในการศึกษาทางด้านคณิตศาสตร์ วิศวกรรม และวิทยาศาสตร์แขนงต่างๆ ตลอดจนในด้านอุตสาหกรรมได้ใช้โปรแกรม MATLAB เป็นเครื่องมือสำหรับใช้ในงานวิจัย พัฒนาและวิเคราะห์

โปรแกรม MATLAB จะมีกล่องเครื่องมือที่ใช้ในการหาคำตอบเรียกว่า Toolbox โดยโปรแกรม MATLAB จะมี Toolbox ในแต่ละสาขา เช่น การประมวลผลสัญญาณ(signal processing toolbox) การประมวลผลภาพ (image processing toolbox) ระบบควบคุม (control

system toolbox) โครงข่ายประสาท (neural networks toolbox) สถิติ (statistics toolbox) ฟัซซี่ลอจิก (fuzzy logic toolbox) เวฟเลท (wavelet toolbox) การติดต่อสื่อสาร (communication toolbox) และสาขาอื่นๆมากมาย ภายใน toolbox แต่ละสาขาจะมีฟังก์ชันต่างๆ ที่เกี่ยวข่องกับการแก้ปัญหาในสาขานั้นๆ ให้เลือกประยุกต์ใช้งานเป็นจำนวนมาก

โครงสร้างของ MATLAB โครงสร้างของโปรแกรม MATLAB ประกอบด้วย 5 ส่วนใหญ่ ๆ คือ

1. ภาษาโปรแกรม MATLAB (The MATLAB language)

MATLAB เป็นโปรแกรมภาษาชั้นสูงที่ใช้ควบคุม flow statements ฟังก์ชัน โครงสร้างข้อมูลอินพุท/เอาท์พุท และลักษณะโปรแกรมใน MATLAB นั้นเป็นทั้งแบบ Structure Programming และ Object-Oriented Programming ทำให้การเขียนโปรแกรมมีความยืดหยุ่นสูงเมื่อเทียบกับการเขียนโปรแกรมด้วยภาษาอื่นๆ เช่น C , Fortran , Java , Basic เป็นต้น

2. สถาปัตยกรรมในการทำงานของ MATLAB (The MATLAB working environment)

MATLAB จะมีกลุ่มของเครื่องมือที่เป็นประโยชน์สำหรับการทำงานของผู้ใช้โปรแกรมหรือโปรแกรมเมอร์ ประโยชน์ที่กล่าวนี้ก็คือการจัดการตัวแปลใน Workspace การนำข้อมูลหรือการผ่านค่าตัวแปรเข้า/ออกและกลุ่มของเครื่องมือต่างๆนี้ ก็จะใช้สำหรับพัฒนา จัดการ ตรวจสอบความผิดพลาดของโปรแกรม (debugging) ที่ได้เขียนขึ้น

3. ฟังก์ชันในการคำนวณทางคณิตศาสตร์ (The MATLAB mathematical function library)

MATLAB จะมีไลบรารีทั่วไปที่ใช้ในการคำนวณอย่างกว้างขวาง เช่น sine , cosine , และพีชคณิตเชิงซ้อน โดยสามารถนำไปประยุกต์ใช้เป็นฟังก์ชันหรือไลบรารีเพิ่มเติมขึ้นจากไลบรารีที่ใช้กันอยู่โดยทั่วไป เช่น ฟังก์ชันในการหา eigenvalues และ eigenvectors การแยกตัวประกอบและส่วนประกอบของเมตริกซ์ด้วยวิธีต่างๆ การวิเคราะห์ข้อมูล การหาความน่าจะเป็นและการแก้ปัญหาของระบบสมการเชิงเส้นที่เป็นพื้นฐานของวิชาสาขาต่างๆ เป็นต้น ทำให้โปรแกรม MATLAB มีฟังก์ชันสำหรับการใช้งานเป็นจำนวนมากและครอบคลุมในรายละเอียดของการคำนวณในสาขาวิชาต่างๆ ได้มากขึ้น

4. Handle Graphics

ระบบกราฟิกของ MATLAB จะประกอบไปด้วยคำสั่งชั้นสูงสำหรับการพล็อตกราฟโดยมีพื้นฐานอยู่บนแนวความคิดที่ว่าทุกๆ สิ่งบนหน้าต่างรูปภาพของโปรแกรม MATLAB จะเป็นวัตถุ (Object) ซึ่งมีเอกลักษณ์เฉพาะตัว Handle Graphics ประกอบด้วยคำสั่งชั้นสูงให้คุณได้เลือกใช้ในการสร้าง Graphics User Interface บนพื้นฐานการประยุกต์ใช้งานของคุณ นอกจากนี้โปรแกรม MATLAB ยังมีฟังก์ชันที่ใช้สำหรับการแสดงภาพสองมิติ ภาพสามมิติและการสร้างภาพเคลื่อนไหว

5. The MATLAB Application Program Interface (API)

API จะใช้เพื่อสนับสนุนการติดต่อจากอุปกรณ์ภายนอกโดยใช่โปรแกรมที่เป็น mex ไฟล์ ซึ่งเป็นไฟล์ที่เขียนขึ้นโดยใช้ mex ฟังก์ชันใน MATLAB ซึ่งจะเรียกใช้รูทีนจากโปรแกรมภาษา C และ Fortran หรืออาจกล่าวได้ว่า API เป็นไลบรารีที่เขียนด้วยโปรแกรมภาษา C และ Fortran ที่มีการเชื่อมต่อกับโปรแกรม MATLAB ด้วยไฟล์ที่เป็น mex ฟังก์ชันอีกทั้ง MATLAB API นี้ยังมีความสามารถสำหรับการเรียก routine จาก MATLAB (dynamic linking) ก็ได้

บทที่ 3
วิธีการดำเนินงาน

ผู้วิจัยได้ดำเนินงานตามลำดับขั้นตอนดังต่อไปนี้

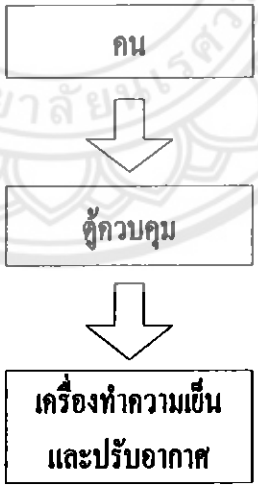
3.1 ศึกษาหลักการทำงาน

ศึกษาหลักการทำงานของเครื่องทำความเย็นและปรับอากาศแบบรวมศูนย์ ศึกษาหลักการทำงานและวงจรการทำงานของเครื่องควบคุมอุณหภูมิและความชื้นของ บริษัทศิลารีเสริช์จ รุ่น AP-105 ศึกษาหลักการทำงานของไมโครคอนโทรเลอร์เบอร์ P89V51RD2FN ศึกษาหลักการเขียนโปรแกรมด้วยภาษาซี และศึกษาหลักการเขียนโปรแกรม MATLAB เพื่อติดต่อกับบอร์ด AP-105 ผ่าน RS232

3.1.1 หลักการควบคุมอุณหภูมิและความชื้นของระบบปรับอากาศแบบรวมศูนย์

ขั้นตอนการควบคุมอากาศแบ่งออกเป็น 2 ส่วน คือ

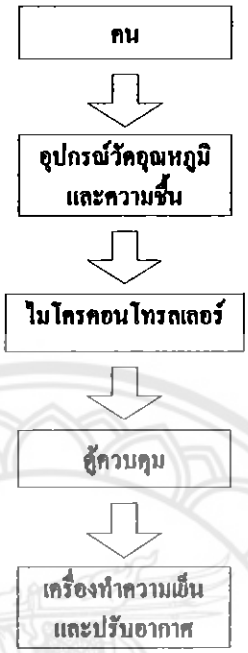
3.1.1.1 หลักการควบคุมระบบฯ ด้วยมือ คือ การควบคุมการทำงานของเครื่องปรับอากาศ โดยอาศัยการตัดสินใจของคนในการเปิด-ปิด เครื่องปรับอากาศ ดูรูปที่ 3.1 ประกอบ



รูปที่ 3.1 แผนผังขั้นตอนการควบคุมการทำงานด้วยมือ

3.1.1.2 หลักการควบคุมระบบฯ แบบกึ่งอัตโนมัติ คือ การควบคุมการทำงานของเครื่องปรับอากาศโดยอาศัยไมโครคอนโทรลเลอร์ซึ่งรับค่าอุณหภูมิและความชื้นภายในบ้านผ่าน

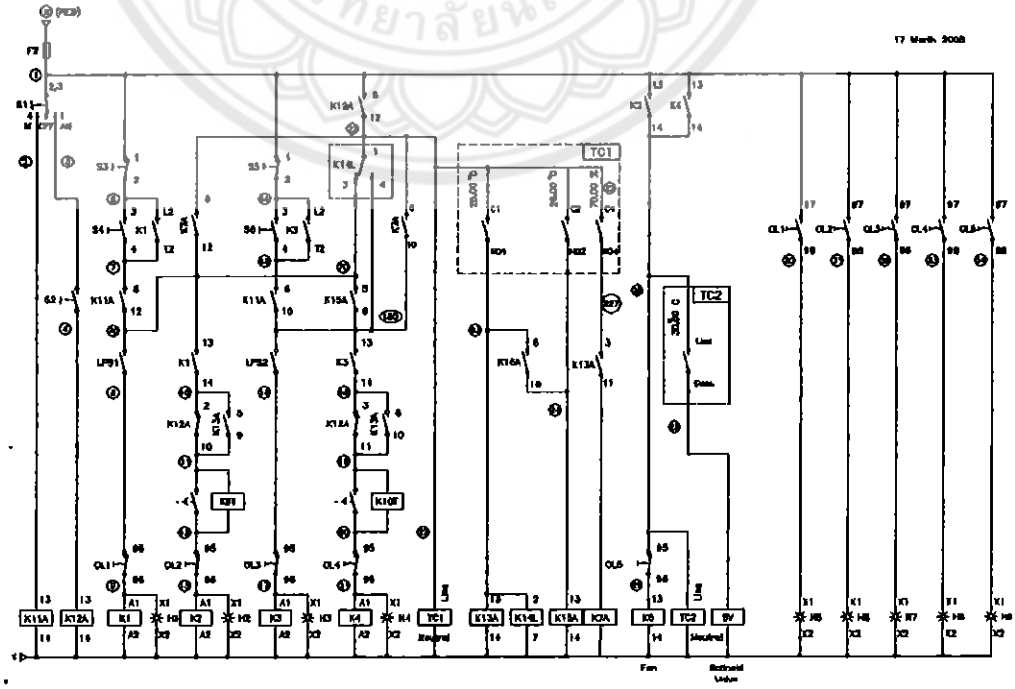
ทางหัววัดอุณหภูมิและความชื้นมาประมวลผลที่ไมโครคอนโทรลเลอร์ในการเปิด-ปิดเครื่องปรับอากาศ ดูรูปที่ 3.2 ประกอบ



รูปที่ 3.2 แผนผังขั้นตอนการควบคุมการทำงานแบบกึ่งอัตโนมัติ

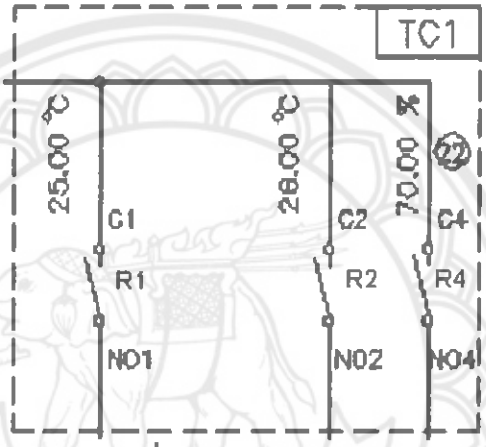
3.2 การออกแบบชิ้นงาน

3.2.1 วงจรไฟฟ้าควบคุมเครื่องปรับอากาศ

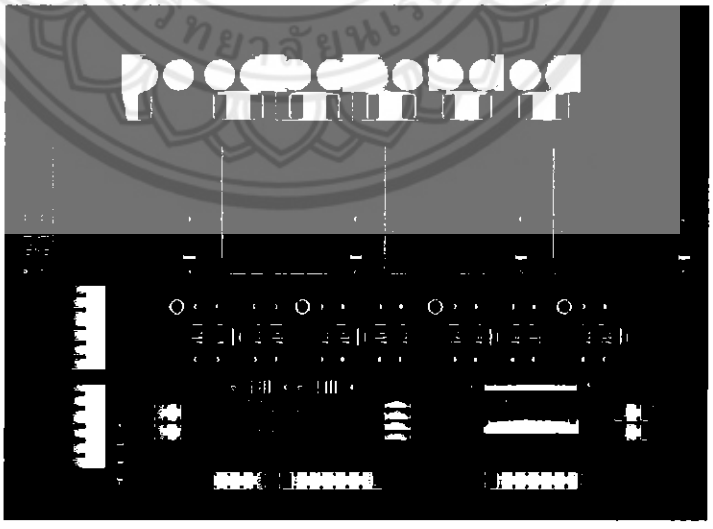


รูปที่ 3.3 วงจรควบคุมเครื่องปรับอากาศ

จากรูปที่ 3.3 เป็นวงจรควบคุมเครื่องปรับอากาศที่มีการทำงานอยู่ด้วยกัน 2 โหมด คือ โหมดควบคุมด้วยมือ และโหมดควบคุมแบบกึ่งอัตโนมัติ โดยเครื่องควบคุมอุณหภูมิและความชื้นที่สร้างขึ้นใหม่ จะเข้าไปทำงานร่วมกันในส่วนของโหมดควบคุมแบบกึ่งอัตโนมัติ โดยมีการต่อรีเลย์เพิ่มเข้าไปในวงจรในส่วนของ TC1 เมื่อรีเลย์ R1, R2 ทำงานแล้วหลังจากนั้นก็จะไปสั่งให้ อีวาโปรเรเตอร์ และคอมเพรสเซอร์ทำงานตามลำดับ สำหรับรีเลย์ R1,R2 นี้จะควบคุมในส่วนของอุณหภูมิ และถ้าอุณหภูมิได้แล้วแต่ความชื้นยังไม่ได้ ก็จะไปสั่งให้รีเลย์ R4 ทำงานแล้วรีเลย์ R4 ก็จะไปสั่งให้อีวาโปรเรเตอร์ 2 ตัวทำงานสลับกันไปเรื่อยๆ ตามค่าของความชื้นที่เราได้ทำการตั้งค่าไว้อีกทีดังภาพขยาย TC1 ในรูปที่ 3.4 ดังนี้



รูปที่ 3.4 ภาพขยาย TC1



รูปที่ 3.5 บอร์ด EX-RELAY V3.0 ของบริษัทศิลารีเสิร์จ

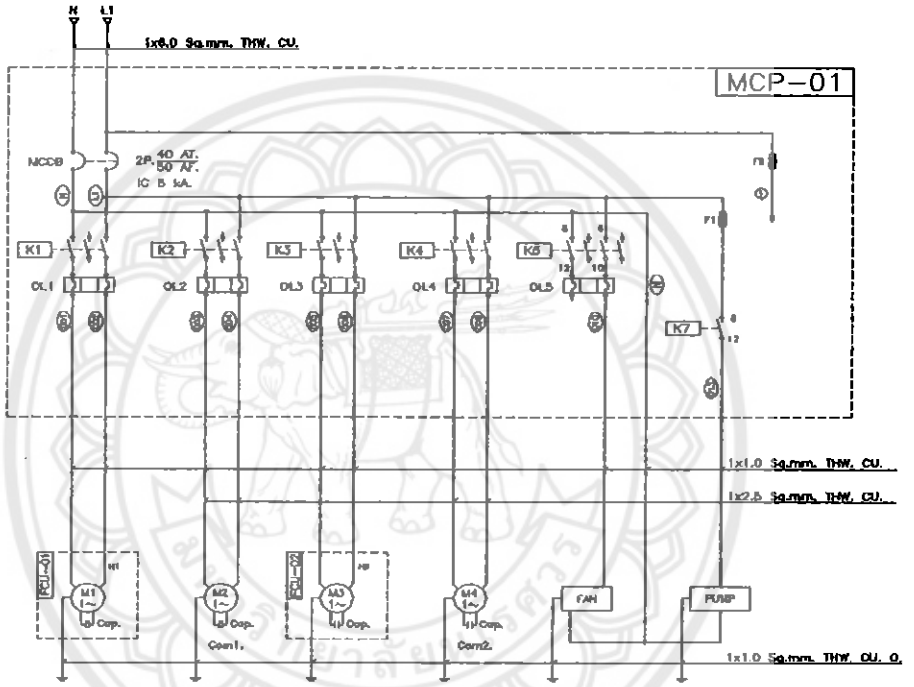
รูป 3.5 คือ ภาพจากบอร์ด EX-RELAY V3.0 ของบริษัทศิลารีเสิร์จ เราจะนำเอา Relay1 (R1), Relay2 (R2) ไปควบคุมในส่วนของอุณหภูมิและเอา Relay4 (R4) ไปควบคุมในส่วนของ

ความชื้นโดยจะมีรายละเอียดดังต่อไปนี้

1. นำขั้วต่อ NO1 และ C1 ไปต่อเข้ากับตู้ควบคุมเครื่องปรับอากาศ ที่หมายเลขขั้ว 22, 23 ตามลำดับ จากวงจรควบคุมในรูปที่ 3.3

2. นำขั้วต่อ NO2 และ C2 ไปต่อเข้ากับตู้ควบคุมเครื่องปรับอากาศ ที่หมายเลขขั้ว 22, 24 ตามลำดับ จากวงจรควบคุมในรูปที่ 3.3

3. นำขั้วต่อ NO4 และ C4 ไปต่อเข้ากับตู้ควบคุมเครื่องปรับอากาศ ที่หมายเลขขั้ว 22, 227 ตามลำดับ จากวงจรควบคุมในรูปที่ 3.3

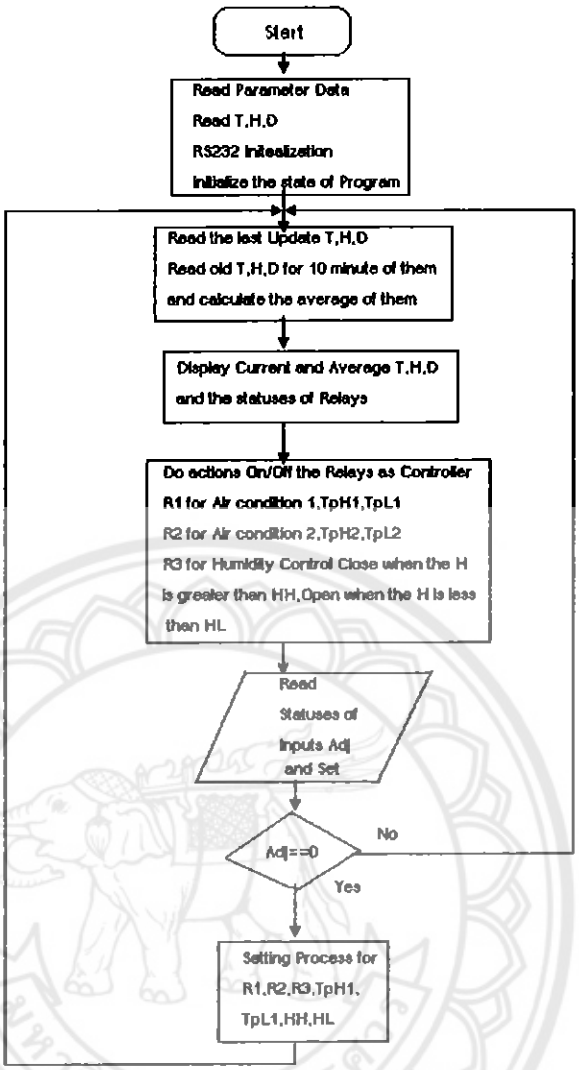


รูปที่ 3.6 วงจรไฟฟ้ากำลังของเครื่องปรับอากาศ

รูปที่ 3.6 คือ วงจรไฟฟ้ากำลังของเครื่องปรับอากาศโดย M1, M3 คือ มอเตอร์พัดลมของชุดคอยล์เย็น และ M2, M4 คือ คอมเพรสเซอร์ของระบบเครื่องปรับอากาศ

3.2.2 โปรแกรมการทำงานของเครื่องควบคุมอุณหภูมิและความชื้นด้วยภาษาซี

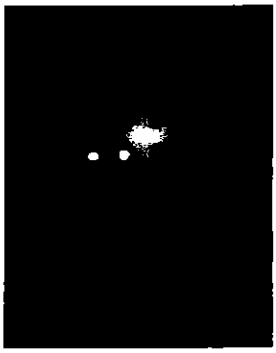
รูปที่ 3.7 แสดงหลักการเขียนโปรแกรมด้วยภาษาซี ด้วยแผนภาพการไหลอย่างคร่าวๆ หลักการก็จะมีอยู่ว่า เมื่อหัววัดอุณหภูมิและความชื้นรับค่าเข้ามา แล้วตัวไมโครคอนโทรเลอร์ก็จะทำการประมวลผลค่าที่รับเข้ามา คือตัวไมโครคอนโทรเลอร์จะทำการบวกค่าที่ได้ทุก ๆ 10 นาที และมาทำการหาค่าเฉลี่ย แล้วถึงจะทำการสั่งให้ตัวรีเลย์ซึ่งติดอยู่กับบอร์ดทำงานและควบคุมเครื่องปรับอากาศต่อไป ในส่วนของการปรับตั้งปุ่ม Adj กับ Set ก็จะสามารถปรับตั้งค่าอุณหภูมิและความชื้นได้ตามความต้องการของผู้ใช้



รูปที่ 3.7 แผนภาพการไหลของโปรแกรม (Flow chard)

3.3 การสร้างชิ้นงาน

3.3.1 ตู้ควบคุมระบบไฟฟ้าของเครื่องปรับอากาศจำนวน 1 ตู้ สร้างตามวงจรในรูปที่ 3.3 และรูปที่ 3.6



ก) รูปด้านหน้าตู้ฯ

ห้องสมุดคณะวิศวกรรมศาสตร์ 05100030

i 5080 836
วร.
9225ค
2580
e.2

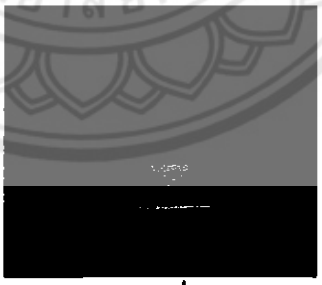
ข) รูปภายในตู้ฯ

รูปที่ 3.8 ตู้ควบคุมระบบไฟฟ้าของเครื่องปรับอากาศจำนวน 1 ตู้

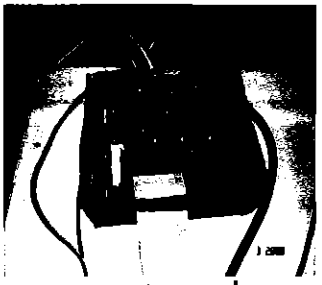
รูปที่ 3.8 จะแสดงรูปภายนอกและภายในตู้ควบคุมระบบไฟฟ้าเครื่องปรับอากาศ ซึ่งการต่อวงจรจะได้ตามรูปที่ 3.3 และ 3.4

3.3.2 เครื่องควบคุมอุณหภูมิและความชื้น จำนวน 1 เครื่อง

ก) รูปด้านหน้าเครื่องฯ



ข) รูปเครื่องฯ



ค) รูปรีเลย์ช่วย เครื่องฯ

รูปที่ 3.9 เครื่องควบคุมอุณหภูมิและความชื้นที่สร้างขึ้นใหม่

หมายเหตุ เครื่องควบคุมอุณหภูมิและความชื้นเป็นของบริษัทศิลารีเสริช์จ แต่เราทำการเปลี่ยนไมโครคอนโทรลเลอร์จากเบอร์ 89C55WD เป็นเบอร์ P89V51RD2FN แล้วทำการเขียนโปรแกรมขึ้นมาใหม่

3.4 การทดสอบชิ้นงาน

โดยนำเครื่องควบคุมอุณหภูมิและความชื้นที่สร้างขึ้นใหม่ไปทำการติดตั้งใช้งานจริงที่บ้านประหยัดพลังงานของ คุณธวัชชัย กลิ่นรอด ตั้งอยู่ที่บ้านเลขที่ 89/1 บ้านบึงถัง ถนนเต็งหนาม - บึงถัง หมู่ที่ 3 ตำบลดอนทอง อำเภอเมือง จังหวัดพิษณุโลก 65000 และทำการเก็บข้อมูลที่ได้จริงจากการทดสอบ

เนื่องจากบ้านหลังนี้นั้นถูกออกแบบโดยผู้ประกอบการ ระบบมีการเดินท่อปล่อยลมออกและท่อดูดลมกลับทั่วทุกห้อง ภาพของบ้านที่ได้ไปทำการติดตั้ง จะเป็นลักษณะดังต่อไปนี้

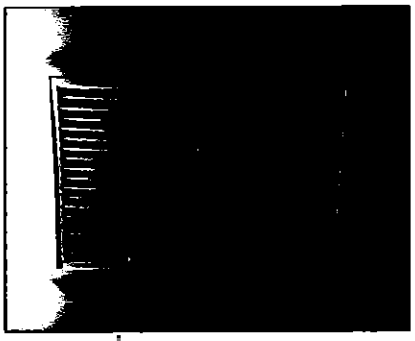
รูปที่ 3.10 บ้านประหยัดพลังงานที่ได้ไปทำการติดตั้ง

รายละเอียดของบ้าน

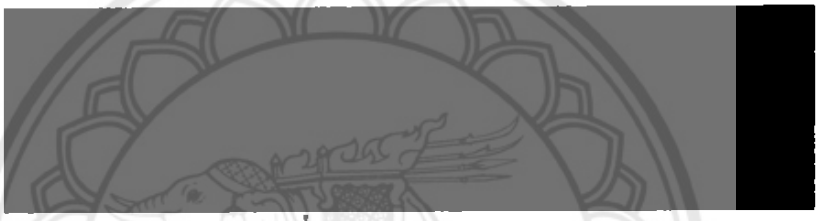
เป็นบ้านชั้นเดียว และมีรายละเอียดดังนี้

- ห้องน้ำ 2 ห้อง
- ห้องนอน 2 ห้อง
- ห้องรับแขก 1 ห้อง
- ห้องครัว 1 ห้อง
- ห้องเก็บของ 1 ห้อง

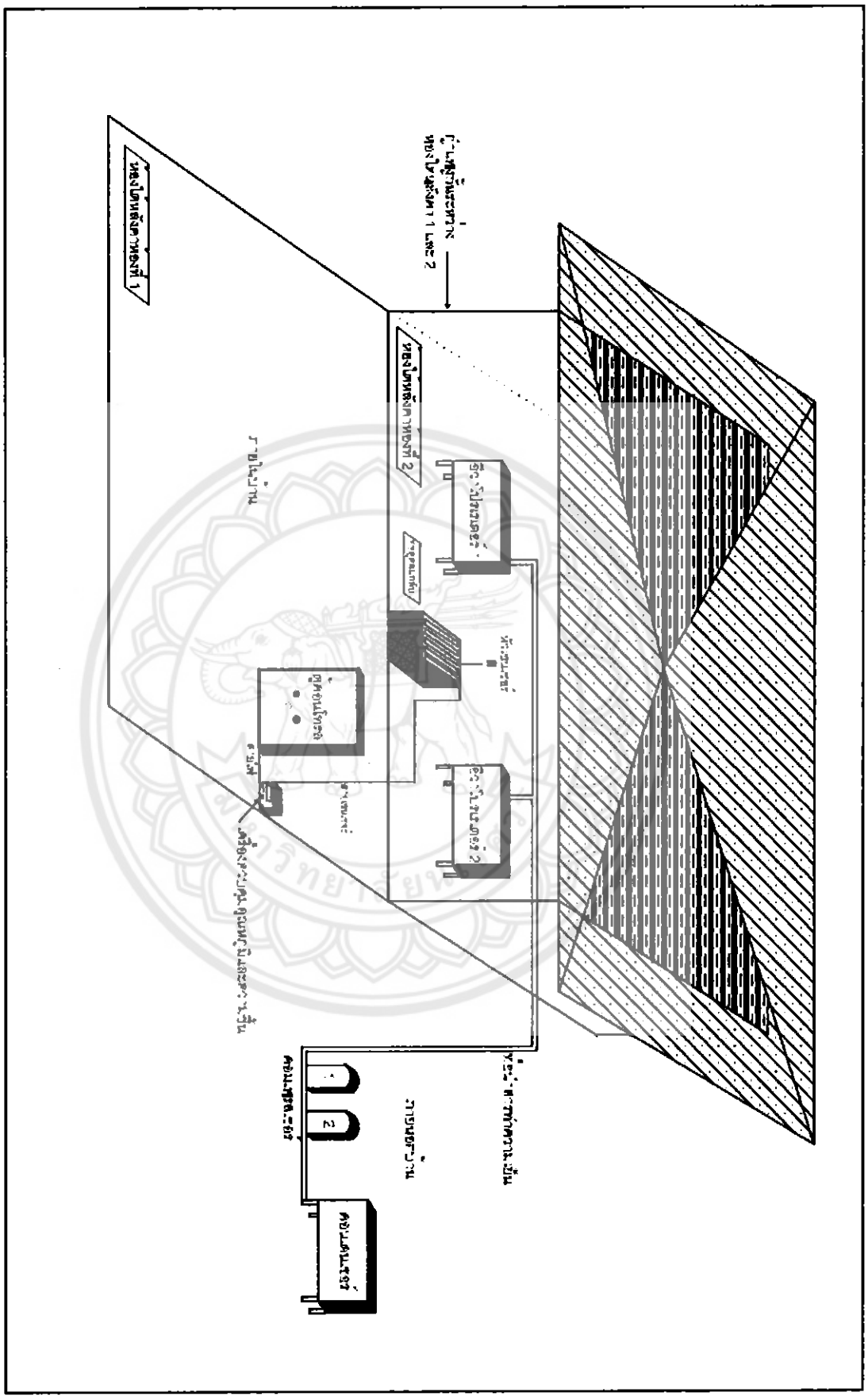
- ห้องเก็บเสื้อผ้า 1 ห้อง
- ห้องใต้หลังคา 2 ห้อง



รูปที่ 3.11 ท่อดูดลมกลับ



รูปที่ 3.12 ท่อปล่อยลมออก



รูปที่ 3.13 แบบประกอบการติดตั้งหัววัดอุณหภูมิและความชื้น

อธิบายแบบประกอบการติดตั้งหัววัดอุณหภูมิและความชื้น

หัววัดอุณหภูมิและความชื้น SHT 15 ของบริษัทศิลา รีเสิร์จ สามารถวัดได้ทั้งค่าอุณหภูมิและความชื้น โดยเราได้ทำการติดตั้งที่บริเวณของท่อดูดลมกลับของระบบเครื่องปรับอากาศ เพราะว่าตรงจุดนี้จะเป็นจุดที่เป็นตัวแทนของค่าอุณหภูมิและความชื้นได้ดีที่สุด โดยเราได้ดึงสายหัววัดอุณหภูมิและความชื้นมาจากห้องใต้หลังคาห้องที่ 1 ซึ่งห้องนั้นได้มีการติดตั้งตู้ควบคุมเครื่องปรับอากาศและเครื่องควบคุมอุณหภูมิและความชื้นรวมอยู่ด้วย

หลักการทำงานของหัววัดอุณหภูมิและความชื้น ก็จะทำการรับค่าอุณหภูมิและความชื้นเข้ามาแล้วไปทำการสั่งให้รีเลย์ที่ติดอยู่กับบอร์ด AP-105 ทำงาน หลังจากนั้นรีเลย์ก็จะไปสั่งให้ระบบเครื่องปรับอากาศทำงานต่อไป

3.5 รวบรวมข้อมูลทั้งหมดเข้ารูปเล่มพร้อมรายงาน

ในหัวข้อนี้เป็นการนำข้อมูลทั้งหมด ซึ่งประกอบไปด้วย ทฤษฎีและหลักการทำงานที่เกี่ยวข้องกับการทำโครงงาน การออกแบบและการสร้างชิ้นงาน และผลการทดสอบการทำงานของชิ้นงาน

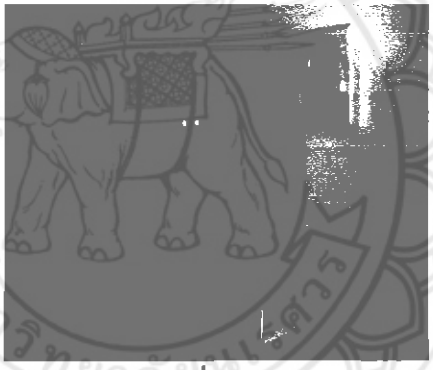
บทที่ 4

วิธีการทดสอบและผลการทดสอบ

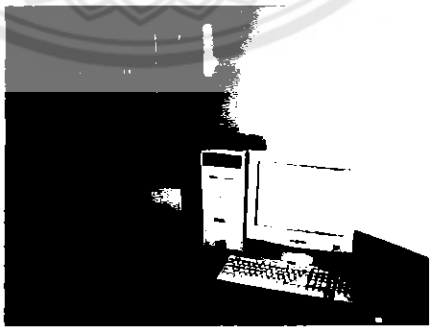
ในบทนี้ จะกล่าวถึงวิธีการทดสอบและผลการทดสอบของเครื่องควบคุมอุณหภูมิที่สร้างขึ้นสำหรับควบคุมเครื่องทำความเย็นและปรับอากาศชนิดแบบรวมศูนย์จำนวน 2 เครื่อง ซึ่งมีการทดสอบดังต่อไปนี้

4.1 การนำไปใช้งานจริง

โดยนำเครื่องควบคุมอุณหภูมิและความชื้นที่สร้างขึ้นใหม่ไปทำการติดตั้งใช้งานจริงที่บ้านประหยัดพลังงานของ คุณธวัชชัย กลิ่นรอด ตั้งอยู่ที่บ้านเลขที่ 89/1 บ้านบึงถัง ถนนเต็งหนาม - บึงถัง หมู่ที่ 3 ตำบลดอนทอง อำเภอเมือง จังหวัดพิษณุโลก 65000



รูปที่ 4.1 ตู้ควบคุมระบบไฟฟ้าฯ พร้อมกับเครื่องควบคุมอุณหภูมิและความชื้นที่สร้างขึ้นใหม่

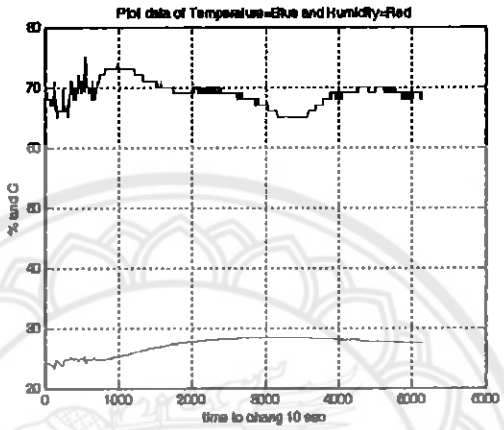


รูปที่ 4.2 การใช้คอมพิวเตอร์ในการเก็บข้อมูลอุณหภูมิและความชื้น

4.2 ผลการทดลอง

จากการนำเครื่องควบคุมอุณหภูมิและความชื้นที่สร้างขึ้นใหม่ไปทำการติดตั้งใช้งานจริง และทำการเก็บค่าอุณหภูมิและความชื้น พร้อมทั้งเขียนเป็นกราฟโดยใช้โปรแกรมแมทแล็บ (MATLAB) พบว่า

4.2.1 การควบคุมเครื่องปรับอากาศภายใต้การใช้เครื่องควบคุมอุณหภูมิแบบเก่า



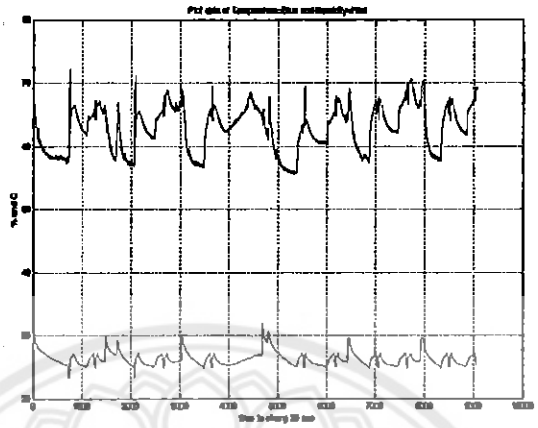
รูปที่ 4.3 กราฟแสดงผลค่าอุณหภูมิและความชื้นภายใต้การใช้เครื่องควบคุมอุณหภูมิและความชื้นแบบเก่า

อธิบาย รูปที่ 4.3 กราฟเส้นบน คือ ความชื้น และกราฟเส้นล่าง คือ อุณหภูมิ จะสังเกตช่วงแรกของกราฟจะได้ว่าเส้นกราฟจะขึ้นๆ-ลงๆ ติดกันมาก จะแสดงให้รู้ว่าช่วงนี้จะมีการตัด-ต่อวงจรของคอมเพรสเซอร์บ่อยเป็นสาเหตุทำให้สิ้นเปลืองพลังงานในช่วงเริ่มเดินเครื่องสูง และจากกราฟจะเห็นได้ว่ามีค่าความแปรปรวนสูงมาก ทำให้สามารถบ่งบอกว่าเครื่องควบคุมไม่มีประสิทธิภาพเท่าที่ควร จากกราฟสามารถอธิบายได้ว่า

ลำดับที่	รายการ	ค่าที่ต้องการ ($^{\circ}c$)		ค่าที่ปรากฏ ($^{\circ}c$)	
		ทำงาน	ไม่ทำงาน	ทำงาน	ไม่ทำงาน
1	เครื่องปรับอากาศเครื่องที่ 1	27	25	25.5	24.5
2	เครื่องปรับอากาศเครื่องที่ 2	28	26	26.5	25.5

ตารางที่ 4.1 ค่าอุณหภูมิที่ทำให้เครื่องปรับอากาศทำงานและไม่ทำงาน

4.2.2 การควบคุมเครื่องปรับอากาศภายใต้การใช้เครื่องเครื่องควบคุมอุณหภูมิและความชื้นแบบใหม่



รูปที่ 4.4 กราฟแสดงผลค่าอุณหภูมิและความชื้นภายใต้การใช้เครื่องควบคุมอุณหภูมิและความชื้นแบบใหม่

อธิบาย รูปที่ 4.4 จากกราฟช่วงแรกเมื่อเปรียบเทียบกับกราฟที่ผ่านมา จะเห็นว่ากราฟจะเป็นเส้นชันลงมา โดยที่ไม่เป็นเส้นชันขึ้นๆ ลงๆ อย่างกราฟที่ผ่านมาแสดงให้เห็นว่า คอมเพรสเซอร์จะตัด-ต่อวงจรน้อยลง จึงมีผลทำให้ประหยัดพลังงานมากขึ้น เส้นที่ชันขึ้นคือช่วงที่คอมเพรสเซอร์หยุดทำงานและจะทำงานอีกครั้งเมื่ออุณหภูมิสูงขึ้น ส่วนเส้นที่ดิ่งขึ้นเป็นเส้นตรงคือช่วงที่อุณหภูมิขึ้นสูงเนื่องจากผู้อยู่อาศัยออกไปนอกบ้านแล้วกลับมาใช้เครื่องปรับอากาศอีกครั้งอุณหภูมิและความชื้นจึงมีค่าสูง จากกราฟสามารถอธิบายได้ว่า

ลำดับที่	รายการ	ค่าที่ต้องการ ($^{\circ}c$)		ค่าที่ปรากฏ ($^{\circ}c$)	
		ทำงาน	ไม่ทำงาน	ทำงาน	ไม่ทำงาน
1	เครื่องปรับอากาศเครื่องที่ 1	27	25	27	25
2	เครื่องปรับอากาศเครื่องที่ 2	28	26	28	26

ตารางที่ 4.2 ค่าอุณหภูมิที่ทำให้เครื่องปรับอากาศทำงานและไม่ทำงาน

ลำดับที่	รายการ	ค่าที่ต้องการ(%)		ค่าที่ปรากฏ(%)	
		ทำงาน	ไม่ทำงาน	ทำงาน	ไม่ทำงาน
1	พัดลมคอยล์เย็นเครื่องปรับอากาศที่ 1	67	63	67	63
2	พัดลมคอยล์เย็นเครื่องปรับอากาศที่ 2	67	63	67	63

ตารางที่ 4.3 ค่าความชื้นที่ทำให้พัดลมคอยล์เย็นเครื่องปรับอากาศทำงานและไม่ทำงาน

หมายเหตุ ค่าอุณหภูมิและความชื้นที่ต้องการนั้นจะตั้งค่าไว้ตามความพอใจของผู้อยู่อาศัย

บทที่ 5
บทสรุป อภิปรายผล ข้อเสนอแนะ

จากผลการทดสอบเครื่องควบคุมอุณหภูมิและความชื้น สามารถสรุปผล อภิปลายผล และมีข้อเสนอแนะดังนี้

5.1 สรุปผล

ผลจากการทดลองพบว่า เครื่องควบคุมอุณหภูมิและความชื้นแบบใหม่ สามารถควบคุมอุณหภูมิและความชื้นได้เที่ยงตรงแม่นยำ ตรงตามความต้องการของผู้ประกอบการ และสามารถลดค่าพลังงานลงจากเดิมได้ ร้อยละ 27

5.2 อภิปรายผล

5.2.1 ความเที่ยงตรงของเครื่องควบคุมอุณหภูมิและความชื้นแบบใหม่

มีผลมาจากการใช้หัววัดอุณหภูมิและความชื้น รุ่น SHT15 ซึ่งมีค่าความผิดพลาดในการวัดค่าอุณหภูมิอยู่ที่ ± 0.4 องศาเซลเซียส และมีค่าความผิดพลาดในการวัดค่าความชื้นอยู่ที่ ± 2.0 % จึงทำให้สามารถวัดค่าอุณหภูมิและความชื้นได้เที่ยงตรงกว่าแบบเก่า และยังใช้หัววัดเพียงหัววัดเดียว ซึ่งเครื่องฯ แบบเก่ามีค่าความผิดผลาดในการวัดค่าอุณหภูมิอยู่ที่ ± 1 องศาเซลเซียส และใช้หัววัดอุณหภูมิ 2 หัววัด ดังนั้นสัญญาณที่เข้ามาจะมี 2 สัญญาณ จึงทำให้มีค่าความเที่ยงตรงน้อยกว่าเครื่องฯ แบบใหม่

5.2.2 ความแตกต่างของเครื่องควบคุมอุณหภูมิและความชื้น

เครื่องควบคุมอุณหภูมิและความชื้นแบบเก่า สามารถควบคุมอุณหภูมิได้เพียงอย่างเดียว แต่เครื่องควบคุมอุณหภูมิและความชื้นแบบใหม่ สามารถควบคุมได้ทั้งอุณหภูมิและความชื้น

5.2.3 การประเมินความเหมาะสมทางด้านเศรษฐศาสตร์

กำหนดให้ค่าพลังงานไฟฟ้า (ค่าไฟฟ้าฐาน + ค่าปรับปรุงต้นทุนการผลิต + ค่าภาษีมูลค่าเพิ่ม) มีราคา 5 บาท/กิโลวัตต์ชั่วโมง จากการทดลองใช้เครื่องควบคุมอุณหภูมิและความชื้นแบบเก่านั้น พบว่าระบบปรับอากาศแบบรวมศูนย์นี้ มีการใช้พลังงานไฟฟ้าเฉลี่ยอยู่ที่ 2.668 กิโลวัตต์ชั่วโมง โดยดูจากมิเตอร์ที่ได้ทำการติดตั้งอยู่ข้างๆตู้คอนโทรล (เป็นจำนวนเงิน 13.34 บาท/ชั่วโมง) หลังจากนำเครื่องฯ แบบใหม่มาทำการควบคุมเครื่องปรับอากาศแล้ว มีผลทำให้พลังงานฯ เฉลี่ยลดลงมาอยู่ที่ 1.951 กิโลวัตต์ชั่วโมง (เป็นจำนวนเงิน 9.75 บาท/ชั่วโมง) ทำให้มีพลังงานฯ เฉลี่ยต่างกัน 0.717 กิโลวัตต์ชั่วโมง (เป็นจำนวนเงิน 3.585 บาท/ชั่วโมง) หรือคิดเป็นร้อยละ 27 ซึ่งเครื่องควบคุมฯ แบบเก่าจำนวน 1 ชุด มีราคา 1,000 บาท ส่วนเครื่องฯ แบบใหม่มีราคา

7,000 บาท ดังนั้นเครื่องฯ ทั้ง 2 มีราคาต่างกันอยู่ที่ 6,000 บาท ถ้าใช้งานเครื่องปรับอากาศประมาณ 1,673 ชั่วโมง เฉลี่ยวันละ 10 ชั่วโมง คิดเป็นระยะเวลาจำนวน 167 วัน หรือจำนวน 5 เดือน ก็จะคืนทุนที่ได้ทำการติดตั้งเครื่องฯ แบบใหม่

5.3 ข้อเสนอแนะ

5.3.1 ศึกษาวิธีการในการป้อนและเก็บข้อมูลออกจากไมโครคอนโทรเลอร์เบอร์ P89V51RD2FN โดยไม่ต้องหยุดการใช้เครื่องปรับอากาศ

5.3.2 ศึกษาวิธีการออกแบบให้ไมโครคอนโทรเลอร์เบอร์ P89V51RD2FN ทำงานแทนวงจรควบคุมระบบไฟฟ้าของเครื่องปรับอากาศ

5.3.3 ระยะเวลาทดลองใช้เครื่องควบคุมอุณหภูมิและความชื้นควรมีระยะเวลาไม่ต่ำกว่า 12 เดือน เพื่อแสดงประสิทธิภาพและความเที่ยงตรงในการเก็บข้อมูลของเครื่องเครื่องควบคุมอุณหภูมิและความชื้น

เอกสารอ้างอิง

[1] รศ.ดร. มนัส สังวรศิลป์ และ วรรัตน์ ภัทรอมรกุล. **คู่มือโปรแกรม MATLAB ฉบับสมบูรณ์** พิมพ์ครั้งที่ 1. นนทบุรี : บริษัท ดวงกระมลสมัย จำกัด. 2543

[2] นคร ภัคดีชาติ,ชัยวัฒน์ ลิ้มพรจิตรวิไล. **ทดลองและใช้งานไมโครคอนโทรลเลอร์ MCS-51 ด้วยโปรแกรมภาษา C ฉบับ P89V51RD2.** พิมพ์ครั้งที่ 1. กรุงเทพฯ : บริษัทอินโนเวตีฟ เอ็กเพ-อริเมนต์ จำกัด. 2521

[3] อุดม รานอก. **ภาษา C สำหรับงานควบคุมไมโครคอนโทรลเลอร์ MCS-51.** นนทบุรี : บริษัท ไอดีซีอินโฟ ดิสทริบิวเตอร์ เซ้นเตอร์ จำกัด. 2537

[4] ผศ. ธีรวัฒน์ ประกอบผล. **คู่มือการเขียนโปรแกรม ภาษา C.** พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : บริษัทซัคเซส มีเดีย จำกัด. 2550

[5] ธีรวัฒน์ จิตต์เนื่อง, วิทยา สงวนวรรณ. **เรียนเพื่อนำไปประยุกต์ใช้งาน อย่างมีประสิทธิภาพ AutoCAD 2002.** กรุงเทพมหานคร : บริษัท เอส.พี.ซี.พริ้นติ้ง จำกัด. 2545

[6] ดร. อัครพันธ์ วงกังแห. ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ มหาวิทยาลัยนเรศวร อำเภอเมือง จังหวัดพิษณุโลก 65000

[7] นายเจริญ บัวเทศ. ห้างหุ้นส่วนจำกัด เซมดีไซน์

ภาคผนวก

ภาคผนวก ก
โปรแกรมการควบคุมการทำงานของเครื่องควบคุมอุณหภูมิและความชื้น

```
#include <reg51RD2.h>
#include <absacc.h>
#include <ctype.h>
#include <intrins.h>
#include <math.h>
#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>


//#include <xdp708b_code.h>
//file xdp708b_code.c
/********** I/O PORT **********/

sbit MXCLK = P3^3;          //กำหนดขาที่ต้องใช้ P1^0;
sbit MXLDB = P3^4;          //กำหนดขาที่ต้องใช้  P1^1;
sbit MXDAT = P3^2;          //กำหนดขาที่ต้องใช้   P1^2;

/********** INT-RAM WORKING AREA **********/

unsigned char DISBUF[8];
unsigned char STRBUF[8+1];

/********** FUNCTION **********/
void mxbyte (unsigned char add,unsigned char dat) {
   unsigned char i;
   for (i=1;i<=8;i++) {          // 8 bit address
      MXDAT = add & 0x80;
      add = add << 1;
      MXCLK = 1;
      MXCLK = 0;
```

```
    }
    for (i=1;i<=8;i++) {          // 8 bit data
      MXDAT = dat & 0x80;
      dat = dat << 1;
      MXCLK = 1;
      MXCLK = 0;
    }
    MXLDB = 1;                    // load clock
    MXLDB = 0;
}

void mxset (void) {               // max7219 setup
    MXCLK = 0;
    MXLDB = 0;
    mxbyte (0x0f,0);              // display - normal
    mxbyte (0x0c,1);              // shutdown - normal
    mxbyte (0x09,0x3f);           // decode mode 0xFF  // decode - no decode 0x00
    mxbyte (0x0a,8);              // intensity
    mxbyte (0x0b,7);              // scan limit - 8 digit
}

void mxload (void) {              // max7219 load to display
    mxset ();
    mxbyte (1,DISBUF[0]);
    mxbyte (2,DISBUF[1]);
    mxbyte (3,DISBUF[2]);
    mxbyte (4,DISBUF[3]);
    mxbyte (5,DISBUF[4]);
    mxbyte (6,DISBUF[5]);
    mxbyte (7,DISBUF[6]);
    mxbyte (8,DISBUF[7]);
}
```

```
void ftosTemp (float dat) {                          // change long to segment
  unsigned char i;  // to disbuf[8]
        unsigned char num;
        unsigned char stl;
  sprintf (STRBUF,"%4.1f",dat);                      // long to string
        stl = strlen(STRBUF)-1 ;
  for (i=0;i<=stl;i++) {                             // string to segment
                DISBUF[stl-i] = 0x0f;
                if (STRBUF[i] == 32)
                                STRBUF[i] = 32;  //DISBUF[stl-i] = 0x0f;
                        else if (STRBUF[i] == 46)
                                DISBUF[stl-i] = 0x8f;
                        else DISBUF[stl-i] = STRBUF[i] - 48;
  }
}

void ftosHumi (float dat) {          // change long to segment
  unsigned char i;                   // to disbuf[8]
        unsigned char num;
        unsigned char stl;
        float datmsb;
        unsigned int datnum;
        datmsb = floor(dat);
  dat = dat-datmsb;
        if (dat >=0.5) datmsb = datmsb + 1;
        datnum = (unsigned int) datmsb;
  sprintf (STRBUF,"%2d",datnum);  // long to string
        stl = strlen(STRBUF)-1;       // +5 because the Humi digits are at 6 and
  for (i=0;i<=stl;i++) {
                                      // string to segment
                DISBUF[stl-i + 4] = 0x0f;
                if (STRBUF[i] == 32)
```

```
                    STRBUF[i] = 32;   //DISBUF[stl-i] = 0x0f;
                else if (STRBUF[i] == 46)
                    DISBUF[stl-i + 4] = 0x8f;
                else DISBUF[stl-i + 4] = STRBUF[i] - 48;
    }
}
//#include <xsht15_on_AP105_code.h>
#define noACK 0
#define ACK   1
                //adr  command  r/w
#define STATUS_REG_W 0x06          //000   0011    0
#define STATUS_REG_R 0x07          //000   0011    1
#define MEASURE_TEMP 0x03          //000   0001    1
#define MEASURE_HUMI 0x05          //000   0010    1
#define RESET     0x1e             //000   1111    0
sbit     DRead = P0^1;
sbit     DATA =  P0^2;
sbit     SCK  = P2^3;
typedef union
{ unsigned int i;
  float f;
} value;
unsigned int TEMP = 1;
unsigned int HUMI = 2;
void dmsec (unsigned long int count) {    // mSec Delay 11.0592 Mhz
   unsigned long int i;                   // Keil v5.2
   while (count) {
     i = 1; while (i>0) i--;               // i = 115
     count--;
   }
}
```

```
/***** writes a byte on the I2C-bus and checks the acknowledge *****/

char s_write_byte(unsigned char value)
{
 unsigned char i,error=0;
 for (i=0x80;i>0;i/=2)                  //shift bit for masking
 { if (i & value) DATA=0;                //masking value with i , write to I2C-BUS
  else DATA=1;
  dmsec(1);
  SCK=0;                                //clk for I2C-BUS
  _nop_();                              //pulswith approx. 5 us
  _nop_();
  _nop_();
  _nop_();
  _nop_();
  _nop_();
        dmsec(1);
  SCK=1;
        dmsec(1);
 }
 dmsec(1);
 DATA=0;                                //release DATA-line
        dmsec(1);                       //nop
 SCK=0;                                 //clk #9 for ack
        dmsec(1);                       //nop
 error=DRead;   // error = DATA;       //check ack (DATA will be pulled down by SHT11)
        dmsec(1);                       //nop
 SCK=1;
        dmsec(1);                       //nop
 return error;                          //error=1 in case of no acknowledge
}
/***** reads a byte form the I2C-bus and gives an acknowledge in case of "ack=1" *****/
```

```
char s_read_byte(unsigned char ack)
{
 unsigned char i,val=0;
 DATA=0;                          //release DATA-line
        dmsec(1);
 for (i=0x80;i>0;i/=2)            //shift bit for masking
 { SCK=0;                         //clk for I2C-BUS
        dmsec(1);
  if (DRead) val=(val | i);       //read bit
        dmsec(1);
  SCK=1;
 }
 dmsec(1);
 DATA=!(!ack);                    //in case of "ack==1" pull down DATA-Line
 dmsec(1);
 SCK=0;                           //clk #9 for ack
 _nop_();                         //pulswith approx. 5 us
 _nop_();
 _nop_();
 _nop_();
 _nop_();
 _nop_();
        dmsec(1);
 SCK=1;
        dmsec(1);
 DATA=0;
        dmsec(1);                 //release DATA-line
 return val;
}

/***** generates a transmission start *****/
 void s_transstart(void)
```

```
{
  DATA=0; dmsec(1);SCK=1;          //Initial state
  _nop_();dmsec(1);
  SCK=0;
  _nop_();dmsec(1);
  DATA=1;
  _nop_();dmsec(1);
  SCK=1;
  _nop_();
  _nop_();
  _nop_();
  _nop_();
  _nop_();
  _nop_();
  dmsec(1);
  SCK=0;
  _nop_();dmsec(1);
  DATA=0;
  _nop_();dmsec(1);
  SCK=1;dmsec(1);
}

/*** communication reset: DATA-line=1 and at least 9 SCK cycles followed by transstart ***/
void s_connectionreset(void)
{
  unsigned char i;
  //Initial state
  DATA=0;
        dmsec(1);
  DRead = ;
        dmsec(1);
  SCK=1;
```

```
        dmsec(1);
 for(i=0;i<9;i++)            //9 SCK cycles
 { SCK=0;
        dmsec(1);
  SCK=1;
        dmsec(1);
 }
 dmsec(1);
 s_transstart();             //transmission start
}

/**** makes a measurement (humidity/temperature) with checksum ****/

char s_measure(unsigned char *p_value, unsigned char *p_checksum, unsigned int mode)
{
 unsigned error=0;
 unsigned int i;

 s_transstart();             //transmission start
 switch(mode){               //send command to sensor
  case 1          : error+=s_write_byte(MEASURE_TEMP); break;
  case 2          : error+=s_write_byte(MEASURE_HUMI); break;
  default     : break;
 }
 for (i=0;i<65535;i++) if(DRead==0) break;  //wait until sensor has finished the measurement
 if(DRead) error+=1;                        // or timeout (~2 sec.) is reached
 *(p_value)  =s_read_byte(ACK);             //read the first byte (MSB)
 *(p_value+1)=s_read_byte(ACK);             //read the second byte (LSB)
 *p_checksum =s_read_byte(noACK);           //read checksum
 return error;
}
```

```
/********** SET RS232 PARAMETER **********/

void init_uart()
{SCON  = 0x52;     //19200 bps @ 11.059 MHz
 TMOD  = 0x20;
 TCON  = 0x69;
 TH1   = 0xfd;   //0xfd;
 TR1   = 1;
}

/********** calculate humidity & temperature **********/
// input :  humi [Ticks] (12 bit)
//          temp [Ticks] (14 bit)
// output:  humi [%RH]
//          temp [°C]

void calc_sht11(float *p_humidity ,float *p_temperature)
{ const float C1=-4.0;                          // for 12 Bit
  const float C2= 0.0405;                       // for 12 Bit
  const float C3=-0.0000028;                    // for 12 Bit
  const float T1=-0.01;                         // for 14 Bit
  const float T2=0.00008;                       // for 14 Bit

  float rh=*p_humidity;                 // rh:     Humidity [Ticks] 12 Bit
  float t=*p_temperature;                   // t:      Temperature [Ticks] 14 Bit
  float rh_lin;                              // rh_lin:  Humidity linear
  float rh_true;                            // rh_true: Temperature compensated humidity
  float t_C;                               // t_C   : Temperature [ïC]

  t_C=t*0.01 - 40;                         //calc. temperature from ticks to [ïC]
  rh_lin=C3*rh*rh + C2*rh + C1;             //calc. humidity from ticks to [%RH]
  rh_true=(t_C-25)*(T1+T2*rh)+rh_lin;      //calc. temperature compensated humidity [%RH]
```

```
 *p_temperature=t_C;                    //return temperature [fC]
 *p_humidity=rh_true;                   //return humidity[%RH]
}

/********** calculate dewpoint **********/
// calculates dew point
// input:   humidity [%RH], temperature [fC]
// output:  dew point [fC]

float calc_dewpoint(float h,float t)
{ float logEx,dew_point;
 logEx=0.66077+7.5*t/(237.3+t)+(log10(h)-2);
 dew_point = (logEx - 0.66077)*237.3/(0.66077+7.5-logEx);
 return dew_point;
}

sbit RL1 = P0^3;
sbit RL2 = P0^4;
sbit RL3 = P0^5;
sbit RL4= P0^6;
sbit RL5 = P0^7;
// Note that the set point THxx is always greater than or equal TLxx
// Also HHxx is always greater than or equal to HLxx

float THR1 = 32;              // Temperature High setpoint
float TLR1 = 30;              // Temperature Low set point
float THR2 = 28;
float TLR2  = 26;
float THR3 = 27;
float TLR3 = 25;
float THR4 = 26;
float TLR4 = 24;
```

```
float HHR5 = 64;              //Humidity High set point
float HLR5 = 60;              //Humidity Low set point

#define numdelay  100 // adjust for 1 second
/*********************************************88
//Note the blink of 5-LED should be one second per a blink.
#define num  60      // Number of Sampling//***********

float Hdata[num], Tdata[num];
float Hav, Tav;
unsigned char interval=10;                //***************
unsigned char nindex,nsec, nRelay;

/* unit in minute
Since the delay per cycle is about 1 sec,
Therefor 60 cycle of program is 1 minite.
*/ /********** MAIN **********/
void main (void)
{
                                          //xdp708b_code.c
  unsigned char a,i,j;
                                        //xsht15_on_AP105_code.c
      value humi_val,temp_val;
      float dew_point;
      unsigned char  checksum;
      unsigned char *hvi;
      unsigned char *tvi;
      unsigned char round;

      init_uart();
      RL1 = 1;                          //กำหนด Relay ต่อใช้งาน RL1 เท่ากับ Relay ตัวที่ 1
      RL2 = 1;
```

```
        RL3 = 1;
        RL4 = 1;
        RL5 = 1;
a = 1;                                  // start segment = 1
for (i=1;i<=24;i++)
        {                                       // 3x8 loop
   for (j=0;j<=7;j++)
                {                               // load to DISBUF
      DISBUF[j] = a;
   }
   mxload ();                                   // load to max7219
   dmsec (250);                                 // delay 0.25 sec
   a = a << 1;                                  // shift segment
   if (a==0) a = 1;
}
                                                // Clear display
for (j=0;j<=5;j++)
        {                                       // load to DISBUF
   DISBUF[j] = 0x0f;
}
        DISBUF[6] = 0x00;
        DISBUF[7] = 0x00;
        mxload();
        s_connectionreset();
        hvi = &humi_val.i;
        tvi = &temp_val.i;
        s_transstart();                         //transmission start
        s_measure( hvi, &checksum, HUMI );      //measure humidity
        s_measure( tvi, &checksum, TEMP);       //measure temperature
        humi_val.f=(float)humi_val.i;           //converts integer to float
        temp_val.f=(float)temp_val.i;           //converts integer to float
        calc_sht11(&humi_val.f,&temp_val.f);    //calculate humidity, temperature
```

```
dew_point=calc_dewpoint(humi_val.f,temp_val.f); //calculate dew point
  ftosTemp(temp_val.f);
ftosHumi(humi_val.f);
DISBUF[6] = nRelay;
mxload();

for (i=0; i <= num-1;i ++)
{
        Tdata[i] = temp_val.f;
        Hdata[i] = humi_val.f;
}
Tav = temp_val.f;
Hav = humi_val.f;

nsec = 0;
nindex = 0;

while (1)
{
   round=60;
        for (i=0; i <= round;i ++)
{
        //xsht15_on_AP105_code.c
s_connectionreset();
        hvi = &humi_val.i;
        tvi = &temp_val.i;
        s_transstart();              //transmission start
        s_measure( hvi, &checksum, HUMI );   //measure humidity
        s_measure( tvi, &checksum, TEMP);    //measure temperature
        humi_val.f=(float)humi_val.i;          //converts integer to float
        temp_val.f=(float)temp_val.i;          //converts integer to float
        calc_sht11(&humi_val.f,&temp_val.f);       //calculate humidity, temperature
```

```
dew_point=calc_dewpoint(humi_val.f,temp_val.f); //calculate dew point
ftosTemp(temp_val.f);
ftosHumi(humi_val.f);
DISBUF[6] = nRelay;
if (nRelay == 0xFF) nRelay = 0x00; else nRelay = 0xFF;
mxload();
nsec = nsec + 1;
if (nsec >= interval-1)
{
        nsec = 0;
        Tdata[nindex] = temp_val.f;
        Hdata[nindex] = humi_val.f;
        if (nindex >= (num-1))
                nindex = 0;
                else
                nindex = nindex + 1;
}
Tav = 0;
Hav = 0;
for (i = 0; i < num;i++)
{
        Tav = Tav + Tdata[i];
        Hav = Hav + Hdata[i];
}
Tav = Tav/num;
Hav = Hav/num;

  // Relay conditions
if (Tav >= THR1) RL1 = 0; else if (Tav < TLR1) RL1  = 1;
if (Tav >= THR2) RL2 = 0; else if (Tav < TLR1) RL2  = 1;
if (Tav >= THR3) RL3 = 0; else if (Tav < TLR3) RL3  = 1;
if (Tav >= THR4) RL4 = 0; else if (Tav < TLR4) RL4  = 1;
```

```
                if (Hav >= HHR5) RL5 = 0; else if (Hav < HLR5) RL5 = 1;
                dmsec(numdelay);
            }
                printf("T:%5.1fH:%5.1f\n",temp_val.f,humi_val.f);
            }
        }
}
```

ภาคผนวก ข
การแสดงกราฟอุณหภูมิและความชื้นที่ได้ออกทางหน้าจอคอมพิวเตอร์โดยใช้โปรแกรม MATLAB

receiver.m

```
function receiver(obj,event)
ct = obj.UserData.ct;
state = obj.UserData.state;
out = fscanf(obj);
ntout = out(1:end-1);
time=clock;
msec=10;
[m,n]=size(ntout);
for i=1:1:14
    if ntout(1,i)=='T'
         temp=str2num(ntout(i+2:i+6));
    end
    if ntout(1,i)=='H'
         humidity=str2num(ntout(i+2:i+6));
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
    time=clock;
    if((time(1,6)>=59.3)&&(time(1,6)<=59.9))
    state=1;
    end
    if((time(1,6)>=30.3)&&(time(1,6)<=30.9))
    state=1;
    end
    if(state==1)
    state=0;
    obj.UserData.state = state;
    ct=ct+1;
    obj.UserData.ct = ct;
    fprintf('date:%d:%d:%d
Time=%d:%d:%.2f\n',time(1,3),time(1,2),time(1,1),time(1,4),time(1,5),
time(1,6));
    fprintf('Temperature:%.2f humidity:%.2f\n',temp,humidity);
    %%%%%%%%%%%%%%%%%% storage data %%%%%%%%%%%%%%%%%%%%%%%
    obj.UserData.ct=ct;
    obj.UserData.time=time;
    obj.UserData.temp=temp;
    obj.UserData.humidity=humidity;
    savedata(ct,time,temp,humidity,obj);
    showdata(obj);
    %%%%%%%%%%%%%%%%%% end storage data %%%%%%%%%%%%%%%%%%%%
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
```

อธิบาย ฟังก์ชั่น receiver นี่เป็นฟังก์ชั่นหลักที่มีการเรียกใช้ฟังก์ชั่นอื่นๆตามมา และตรงส่วนนี้จะมีการกำหนดว่าจะให้กราฟแสดงผลค่าอุณหภูมิและความชื้นห่างกันกี่วินาที ในที่นี้ตั้งไว้ห่างกันที่ 30 วินาที คือทุกๆ 30 วินาทีจะทำการแสดงค่าของอุณหภูมิและความชื้นออกมาอย่างละ 1 ค่า

start.m

```
load s1
fopen(s1)
```

อธิบาย ฟังก์ชั่น start ทำหน้าที่เริ่ม Run Program โดย S1 คือ Serial Port Object : Serial - COM1 เราต้องตั้งค่า S1 ให้เรียบร้อยเพื่อให้คอมพิวเตอร์เชื่อมต่อกับเครื่องวัดอุณหภูมิและความชื้นได้ โดยใช้คำสั่ง get(s1) พิมพ์ลงไปในส่วน Command Window ในโปรแกรม MATLAB ดังแสดงข้างล่างนี้

```
>>get(s1)
    ByteOrder = littleEndian
    BytesAvailable = 0
    BytesAvailableFcn = [1x1 function_handle]
    BytesAvailableFcnCount = 48
    BytesAvailableFcnMode = terminator
    BytesToOutput = 0
    ErrorFcn =
    InputBufferSize = 512
    Name = Serial-COM1
    ObjectVisibility = on
    OutputBufferSize = 512
    OutputEmptyFcn =
    RecordDetail = compact
    RecordMode = overwrite
    RecordName = record.txt
    RecordStatus = off
    Status = closed
    Tag =
    Timeout = 10
    TimerFcn =
    TimerPeriod = 1
    TransferStatus = idle
    Type = serial
    UserData = [1x1 struct]
```

ValuesReceived = 0

ValuesSent = 0

SERIAL specific properties:

BaudRate = 9600

BreakInterruptFcn =

DataBits = 8

DataTerminalReady = on

FlowControl = none

Parity = none

PinStatus = [1x1 struct]

PinStatusFcn =

Port = COM1

ReadAsyncMode = continuous

RequestToSend = on

StopBits = 1

Terminator = CR

เราต้องตั้งค่าต่างๆทั้งหมดตามที่เห็นข้างบนนี้ หรือถ้าต้องการค่า Baudrate ค่าอื่นก็สามารถกระทำได้ จงประยุกต์ใช้ให้เหมาะสมกับงาน

showdata.m

```
function showdata(obj)
load data;
ct = obj.UserData.ct;
t=1:1:ct;
y1=zeros(1,ct);
y2=zeros(1,ct);
temp=data(:,7);
humidity=data(:,8);
for j=1:1:ct
    y1(1,j)=temp(j,1);
end
for j=1:1:ct
    y2(1,j)=humidity(j,1);
end
figure(1)
plot(t,y1,'b',t,y2,'r');grid on
xlabel('Sampling Number (Ts = 30 sec)');
ylabel('% and C');
title('Plot data of Temperature=Blue and Humidity=Red')
```

อธิบาย ฟังก์ชั่น showdata ตรงส่วนนี้จะเป็นการกำหนดพวก ตัวอักษรต่างๆ หรือค่าอื่นๆที่ต้องการแสดงบนกราฟ คือตรงส่วนนี้จะเป็นส่วนที่กำหนด

stop.m

```
fprintf('Stop receiver\n');
fclose(s1)
```

อธิบาย ฟังก์ชั่น stop เป็นฟังก์ชั่นที่ใช้ในการหยุดการทำงานของโปรแกรม

savedata.m

```
function savedata(ct,time,temp,humidity,obj)
load data
ct = obj.UserData.ct;
time = obj.UserData.time;
temp = obj.UserData.temp;
humidity= obj.UserData.humidity;
[m,n]=size(data);
for i=1:1:n-2
    data(ct,i)=time(1,i);
end
data(ct,n-1)=temp;
data(ct,n)=humidity;
save data;
fprintf('round=%d\n',ct);
%data(ct,:)
```

อธิบาย ฟังก์ชั่น savedata เป็นฟังก์ชั่นที่ใช้สำหรับ save ข้อมูล

printsheet.m

```
load data
[m,n]=size(data);
output=fopen('sheetfile.txt','w');
for k=1:1:m;
    for i=1:1:n
      fprintf(output,'%.2f\t',data(k,i));
    end
    fprintf(output,'\n');
end
fclose(output);
fprintf('End print data sheet\n');
```

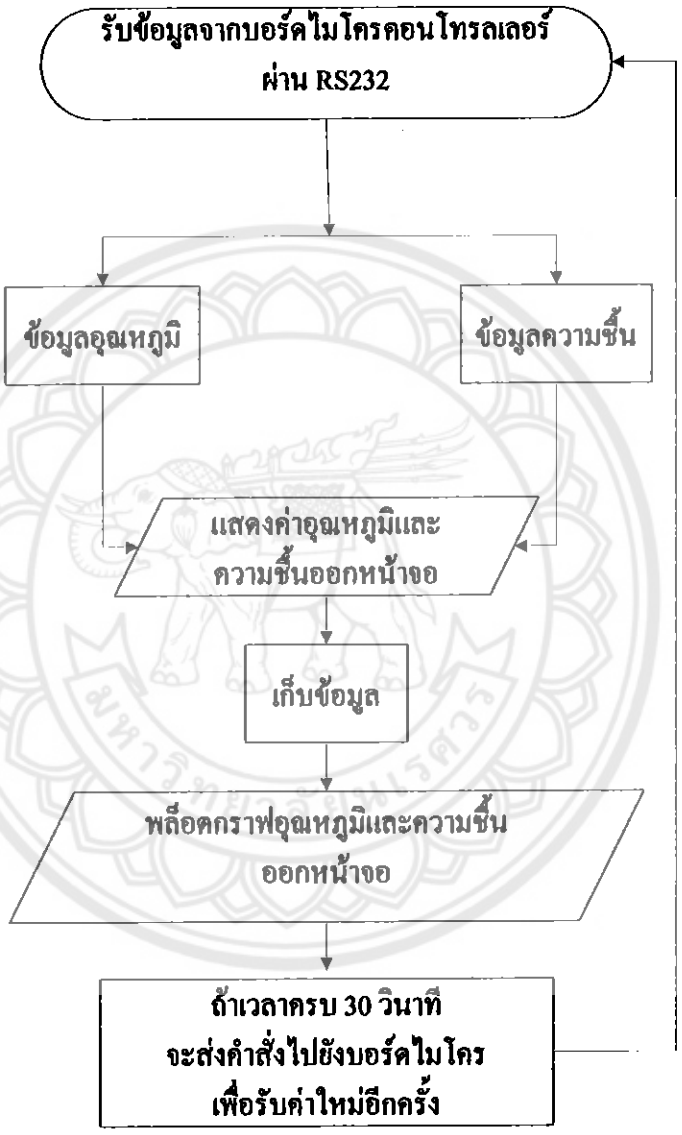
อธิบาย ฟังก์ชั่น printsheet เป็นฟังก์ชั่นที่ใช้สำหรับการสั่งพิมพ์

cleardata.m

```
clc
close all
clear all
data=zeros(30000,8);
save data
load s1
ct=1;
obj.UserData.state=0;
obj.UserData.ct=0;
obj.UserData.time=0;
obj.UserData.temp=0;
obj.UserData.humidity=0;
```

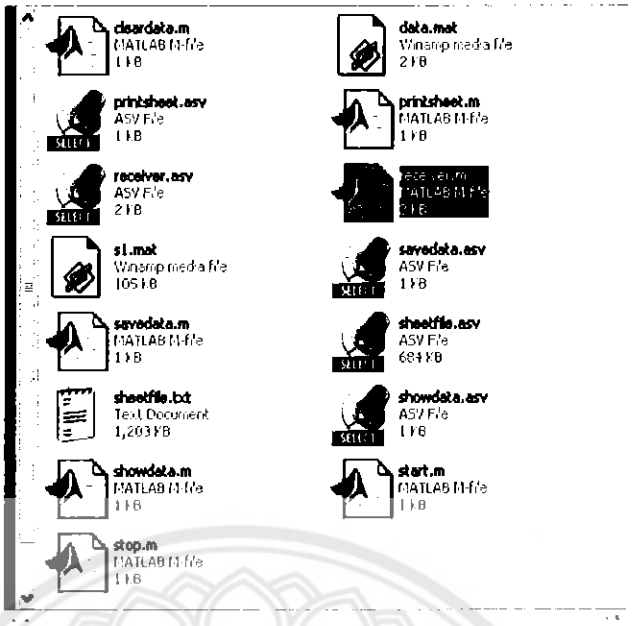
```
obj.UserData
save s1
disp(ct);
fprintf('clear storage data and set start time\n');
```

อธิบาย ฟังก์ชั่น cleardata เป็นฟังก์ชั่นที่ใช้สำหรับลบข้อมูลที่มีการ Run Program ครั้งที่ผ่านมา เพราะถ้าไม่ทำการลบทิ้งก็จะทำให้มีข้อมูลในส่วนที่ไม่ต้องการเยอะเกินไป



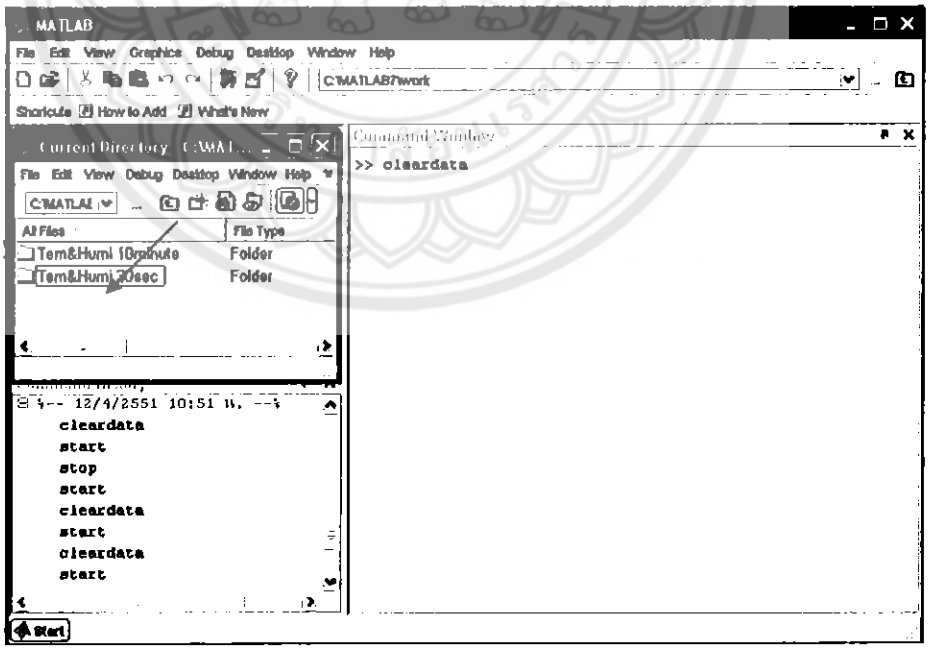
รูป ก Flow chard การทำงานของโปรแกรม MATLAB

หมายเหตุ โปรแกรมทั้งหมดนี้ต้องเก็บไว้ใน Folder ที่ชื่อว่า Work ของโปรแกรม MATLAB เสมอ และถ้าจะเรียกออกมาใช้งานก็ต้องเข้าไปที่ Folder ดังกล่าวเพื่อที่จะเรียกโปรแกรมออกมาใช้งาน และหลักการเขียนโปรแกรม MATLAB นั้น ต้องแยกฟังก์ชั่นแต่ละฟังก์ชั่นออกจากกันหรือทำการ Save คนละ Editor นั้นเอง อย่างที่ผู้เขียนได้ทำตัวหนาไว้เพื่อให้รู้ว่าคนละ Editor



รูป ข การ Save file ไว้คนละ Editor

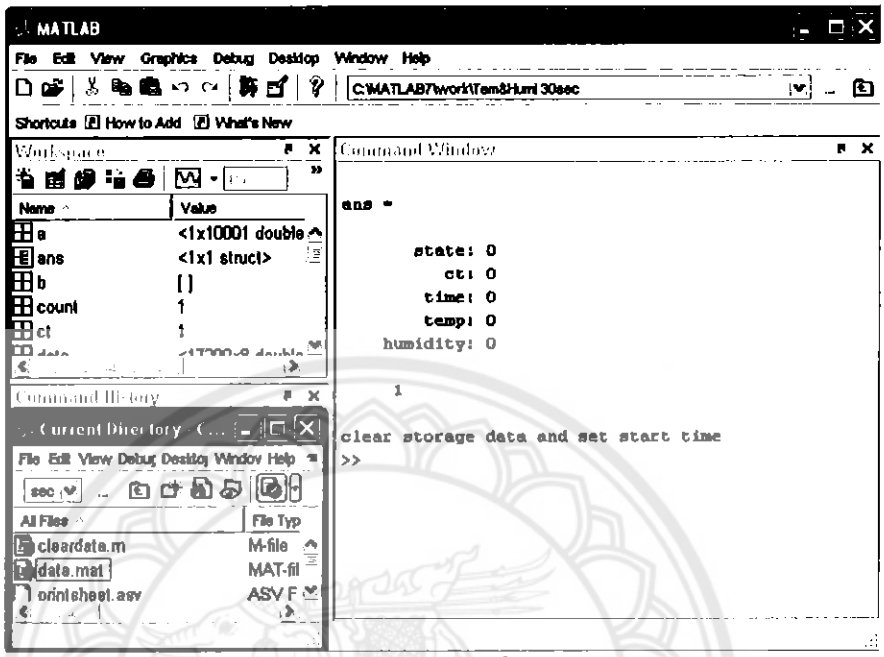
และทุก Editor ที่ทำการ Save ไว้ต้องอยู่ใน Folder เดียวกันทั้งหมด และถ้าเราต้องการเรียกใช้งานต้องเข้าไปที่โปรแกรม MATLAB แล้วเปิด Folder ที่เราทำการ Save ไว้ในที่นี้ชื่อ Folder ว่า Tem & Humi 30 sec ดังภาพลูกศรชี้ข้างล่างนี้ ให้ดับเบิลคลิกได้เลย



รูป ค การเข้า Folder ที่ถูกเก็บไว้ใน Tem & Humi 30 sec

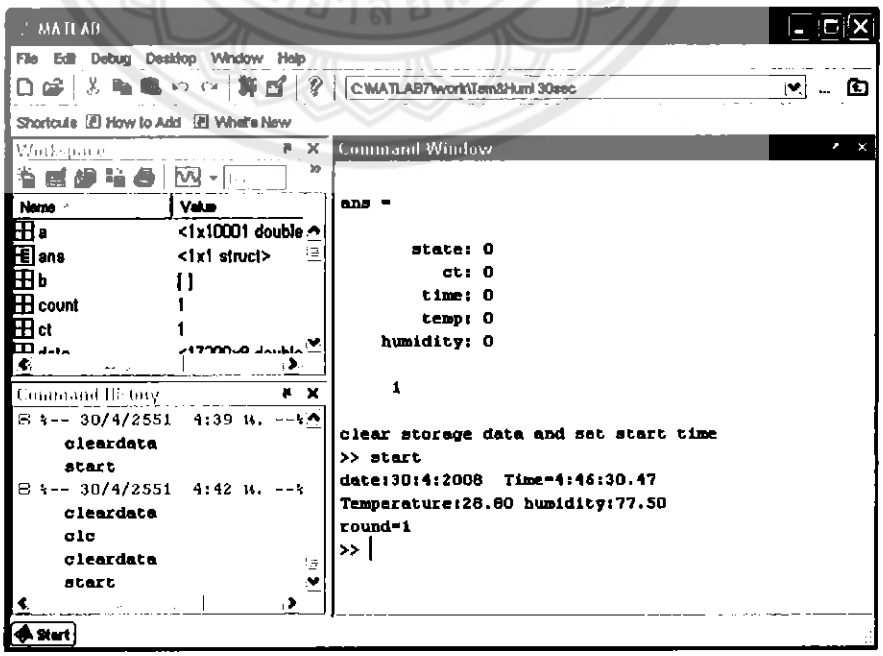
แล้วขั้นตอนต่อไป คือ ต้องต่อเครื่องควบคุมอุณหภูมิและความชื้นเข้ากับคอมพิวเตอร์ให้เรียบร้อยโดยใช้ RS232 เป็นตัวเชื่อม แล้วเมื่อทุกอย่างพร้อมเราก็ทำขั้นตอนดังต่อไปนี้

1.พิมพ์ cleardata ที่หน้าต่าง Command Window (ตรงส่วนนี้จะไม่ขออธิบายอีก เพราะได้อธิบายการทำงานของแต่ละฟังก์ชั่นไว้ในส่วนข้างบนแล้ว) ดังแสดงในรูป ง

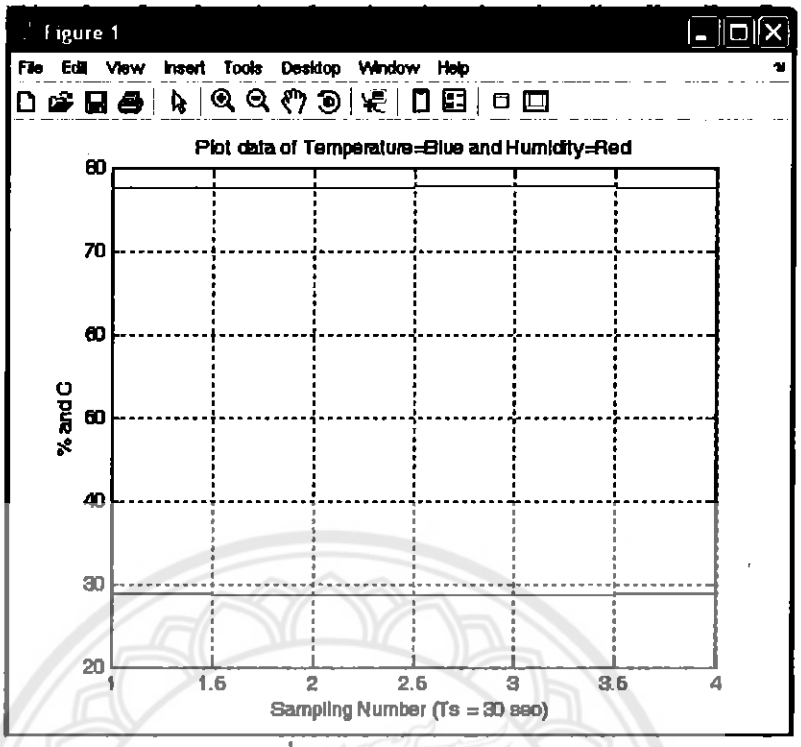


รูป ง การลบข้อมูลที่มีอยู่เดิม

2. พิมพ์ start ที่หน้าต่าง Command Window โปรแกรมก็จะสามารถแกรมทำงาน และหลังจากที่เราได้พิมพ์คำว่า start ลงไปจะได้ผลดังนี้



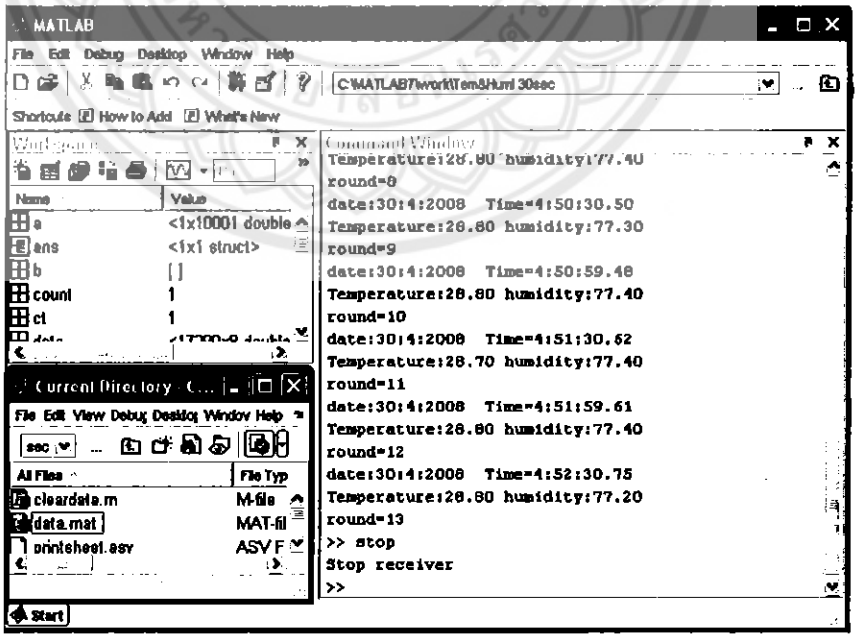
รูป จ การทำงานของโปรแกรม



รูป ฉ กราฟที่ได้จากการทางานของโปรแกรม

กราฟก็จะทำงาน ไปเรื่อยๆตามค่าอุณหภูมิและความชื้นที่เปลี่ยนแปลงทุกๆ 30 วินาที

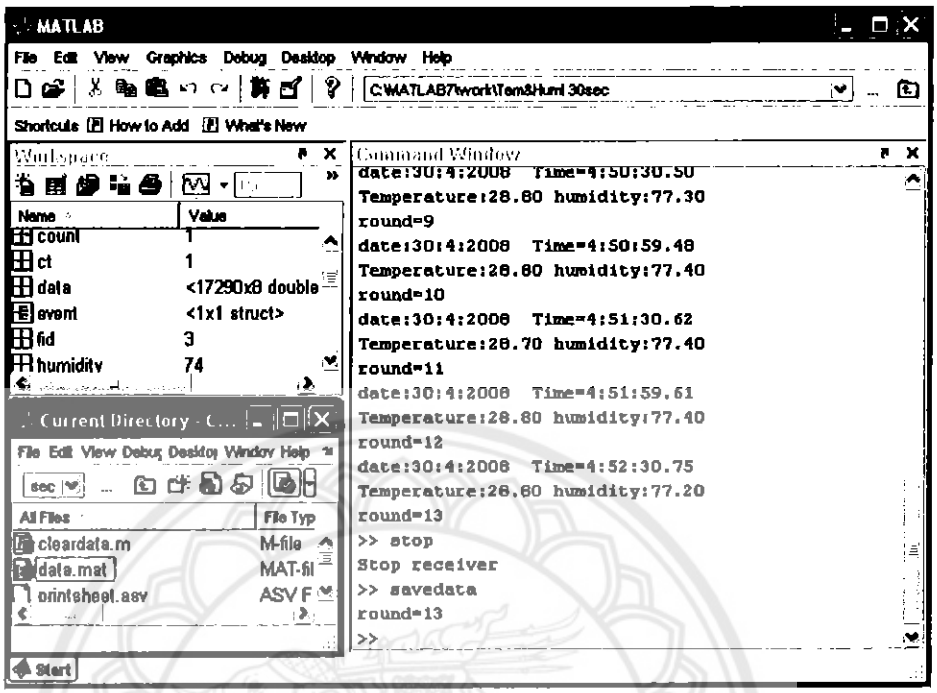
3. พิมพ์ stop ที่หน้าต่าง Command Window เมื่อต้องการให้โปรแกรมหยุดการทำงาน



รูป ช การหยุดโปรแกรม

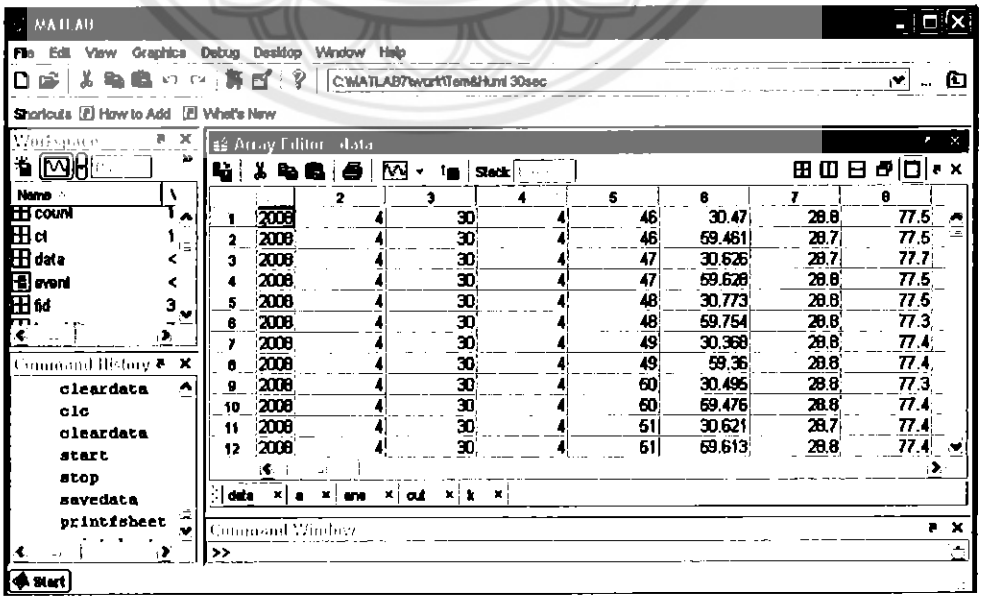
จาก Command Window จะเห็นว่าโปรแกรมจะหยุดทำงานรอบที่ 13 เมื่อเราพิมพ์คำว่า stop

4. พิมพ์ savedata ที่หน้าต่าง Command Window เมื่อต้องการจะ save ข้อมูล วัน/เดือน/ปี/อุณหภูมิ/ความชื้น ที่ได้จากการทำงานของโปรแกรมที่ผ่านมา



รูป ซ การเก็บข้อมูล

เมื่อเราพิมพ์คำว่า savedata ที่ Command Window ดังแสดงในรูป ซ ก็จะได้ว่า เราสามารถเก็บข้อมูลได้ทั้งหมด 13 รอบ แล้วต่อไปเราก็จะไปเอาข้อมูลที่ได้ทำการ save ไว้โดยเข้าไปที่ Workspace ให้คลิกที่คำว่า data ก็จะพบหน้าต่างที่แสดงดังภาพข้างล่างนี้ ซึ่งจะอยู่ในรูปของ Array



รูป ฌ การเก็บข้อมูลในรูปของ Array

จากรูปจะเห็นว่า หลักแรกจะบอกถึงค่าต่างๆดังนี้ จะขออธิบายเฉพาะแถวที่ 1 ส่วนแถวอื่นๆ ก็หลักการเหมือนกัน

หลักที่ 1 ปี ค.ศ. ที่ได้ทำการ Run Program คือปี 2008

หลักที่ 2 จะบอกถึงเดือน Run Program คือ เดือนเมษายน(เดือนที่ 4)

หลักที่ 3 จะบอกวันที่ทำการ Run Program คือวันที่ 30

หลักที่ 4 จะบอกถึงเวลา ที่มีหน่วยเป็นชั่วโมง คือ ตี 4

หลักที่ 5 จะบอกถึงเวลา ที่มีหน่วยเป็นนาที คือ 46 นาที

หลักที่ 6 จะบอกถึงเวลา ที่มีหน่วยเป็นวินาที คือ 30.47 วินาที

หลักที่ 7 จะบอกค่าอุณหภูมิ คือ 28.8 องศา

หลักที่ 8 จะบอกค่าความชื้น คือ 77.5 %

5. พิมพ์ printsheet ที่หน้าต่าง Command Window เมื่อต้องการสั่งพิมพ์ข้อมูลออกมาดู เครื่องก็จะทำงานร่วมกันกับปริ้นเตอร์ที่เราได้ทำการตั้งค่าไว้

หมายเหตุ ถ้าอยากรู้ความหมาย และหน้าที่ของคำสั่งแต่ละคำสั่ง เพิ่มเติมให้ไปดูได้ที่ Help ในโปรแกรม MATLAB

ประวัติผู้เขียนโครงงาน

ชื่อ นายจักรพันธ์ หวาจ้อย

ภูมิลำเนา 39/1 ม.17 ต.ไผ่รอบ อ.โพธิ์ประทับช้าง จ.พิจิตร 66190

ประวัติการศึกษา

- จบระดับมัธยมศึกษาตอนปลาย จากโรงเรียนพิจิตรพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : two_ee@hotmail.com

ชื่อ นายปรัชญา จันทร์คามิ

ภูมิลำเนา 27/1 ม.1 ต.วังม่วง อ.เปือยน้อย จ.ขอนแก่น 40340

ประวัติการศึกษา

- จบระดับมัธยมศึกษาตอนปลาย จากโรงเรียนบ้านไผ่ (ขก.5)
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : prachaya_5@hotmail.com