

ปริญญานิพนธ์

ระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วน

ENERGY CONSUMPTION RECORD AND CONTROL

SYSTEM FOR SPLIT TYPE AIR CONDITION



ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม

สาขาวิชาวิศวกรรมโทรคมนาคม

ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

รฟ.

๖๕๖๖

๖๕๖๖

เลขหมู่.....
 เลขทะเบียน..... 59538
 วัน,เดือน,ปี..... - 7 ส.ย. 2549

ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 b.....
 i.....



ภาควิชาวิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญาโท

ชื่อหัวข้อ ระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วน
Energy Consumption Record and Control System for Split Type Air Condition

ชื่อนักศึกษา	1. นายจิตปกรณ์ เตโชพันธ์	รหัสประจำตัว	46035305
	2. นางสาวปิยารรณ ดิศรพงศ์	รหัสประจำตัว	46035330
	3. นายสุธี ทองแกมแก้ว	รหัสประจำตัว	46035363
	4. นายปิยพงศ์ ชี้อย่าง	รหัสประจำตัว	46035707

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชา วิศวกรรมโทรคมนาคม

อาจารย์ที่ปรึกษา อ.โกศล ตราชู

อาจารย์ที่ปรึกษาร่วม อ.ปิยะ สุภวราสุวัฒน์

คณะกรรมการสอบปริญญาโท	ลายมือชื่อ
1. อ.ไพบูลย์ พวงวงศ์ตระกูล	
2. อ.โกศล ตราชู	
3. อ.วรวิทย์ สมหา	
4. อ.สุชิน อางหาญ	
5. อ.อมรชัย ชัยชนะ	

วัน/เดือน/ปีที่สอบ วันพฤหัสบดีที่ 28 เมษายน พ.ศ. 2548 เวลา 14.00 น.

สถานที่สอบ ห้อง ค.311 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว

ลงนาม.....

(ผศ.สุรสิทธิ์ รัตรี)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่.....เดือน.....พ.ศ.....



<BT4720342>

ระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศ

แบบแยกส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง ระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วน

Energy Consumption Record and Control System for Split Type Air Condition

วัตถุประสงค์

1. เพื่อศึกษาระบบปรับอากาศแบบแยกส่วน หลักการวัดแรงดัน กระแส อุณหภูมิและการสื่อสารข้อมูลแบบดิจิทัล
2. เพื่อออกแบบระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วน
3. เพื่อสร้างระบบบันทึกและควบคุมการใช้พลังงานไฟฟ้าของระบบปรับอากาศแบบแยกส่วน
4. เพื่อทดสอบระบบบันทึกและควบคุมการใช้พลังงานไฟฟ้าของระบบปรับอากาศแบบแยกส่วน
5. เพื่อนำระบบบันทึกและควบคุมการใช้พลังงานไฟฟ้าของระบบปรับอากาศแบบแยกส่วนไปใช้งาน

ประโยชน์ที่คาดว่าจะได้รับ

1. มีความรู้ในระบบปรับอากาศแบบแยกส่วน หลักการวัดแรงดัน กระแส อุณหภูมิและการสื่อสารข้อมูลแบบดิจิทัล
2. ได้ต้นแบบของระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วน
3. ได้ระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วนที่สมบูรณ์
4. ได้ผลการทดสอบระบบบันทึก และควบคุมการใช้พลังงานไฟฟ้า ของระบบปรับอากาศแบบแยกส่วน
5. ได้นำระบบบันทึกและควบคุมการใช้พลังงานไฟฟ้าของระบบปรับอากาศแบบแยกส่วนไปใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

I

ชื่อหัวข้อ	ระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วน	
นักศึกษา	นายจิตปกรณ์	เตโชพันธ์
	นางสาวปิยารรณ	ศิรพงศ์
	นายสุธี	ทองแกมแก้ว
	นายปิยพงศ์	ยี่ซ่าย
อาจารย์ที่ปรึกษา	อาจารย์โกศล	ตราชู
อาจารย์ที่ปรึกษาร่วม	อาจารย์ปิยะ	ศุภวราสุวัฒน์
หลักสูตร	ครุศาสตร์อุตสาหกรรมบัณฑิต	
สาขาวิชา	วิศวกรรมโทรคมนาคม	
ปีการศึกษา	2547	

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอการออกแบบและการสร้าง ระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วน โดยระบบนี้ประกอบด้วย ชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ ชุดแปลงสัญญาณ RS232/RS485 และภาคแสดงผลบนเครื่องคอมพิวเตอร์ ซึ่งสามารถแสดงผลเป็นตัวเลข กราฟ และค่าอุณหภูมิที่เวลาจริงและที่บันทึกไว้ ระบบนี้สามารถช่วยให้การใช้พลังงานของระบบปรับอากาศแบบแยกส่วนถูกควบคุมให้อยู่ในปริมาณที่เหมาะสม โดยไม่ส่งผลกระทบต่อพฤติกรรมการใช้งานตามปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

II

Thesis Title	Energy Consumption Record and Control System for Split Type Air Condition
Students	Mr.Jitpakorn Tachopunth Miss Piyawan Disonpong Mr.sutee Tongkamkaew Mr.Piyapong Yeesai
Advisor	Mr.Koson Trachu
Co-Advisor	Mr.Piya Supavarasuwat
Education Level	Bachelor of Science in Industrial Education
Program in	Telecommunication Engineering
Academic Year	2004

ABSTRACT

This thesis presented a design and development of Energy Consumption Record and Control System for Split Type Air condition. The system consisted of the measure, the control air condition, the RS232/RS485 converter and the display on computer. Which can show the number, graph and temperature at real time? This system control energy of split type air condition with quantity properly and don't have effect with action.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ถูกล่วงไปได้ด้วยดี เนื่องมาจากความร่วมมือของสมาชิกภายในกลุ่มทุกท่าน ขอขอบคุณอาจารย์โกศล ตรีชู อาจารย์ปิยะ สุภวราสุวัฒน์ คณาจารย์ และเจ้าหน้าที่ภาควิชาครุศาสตร์วิศวกรรมทุกท่านที่ให้ความอนุเคราะห์ให้คำแนะนำ แนวความคิด ความรู้ต่างๆ แนวทางการแก้ปัญหาในการจัดทำปริญญานิพนธ์ ขอขอบคุณห้องสมุดคณะครุศาสตร์อุตสาหกรรม สำนักหอสมุดกลาง และเจ้าหน้าที่ห้องปริญญานิพนธ์คณะครุศาสตร์อุตสาหกรรม ที่ช่วยอำนวยความสะดวกและเอื้อเฟื้อสถานที่ในการค้นคว้าข้อมูล สุดท้ายที่ดำริก่อการระลึกถึงอย่างยิ่ง บิดามารดา และผู้มีพระคุณที่เป็นผู้ให้การสนับสนุนด้านการศึกษาและเป็นผู้ให้กำลังใจเมื่อยามรู้สึกท้อแท้ ให้กลับรู้สึกดีขึ้นอีกครั้ง ตั้งแต่อดีตจนถึงปัจจุบัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญรูป	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญ	1
1.2 ชัดความสามารถของโครงการ	1
1.3 เนื้อหาโดยสังเขป	1
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 กล่าวนำ	3
2.2 เครื่องปรับอากาศแบบแยกส่วน	3
2.2.1 ชุดคอยล์น้ำเย็นหรืออีวาพอเรเตอร์	3
2.2.2 ชุดคอนเดนซิงยูนิต	6
2.2.3 วงจรไฟฟ้าเครื่องปรับอากาศแบบแยกส่วน	8
2.3 เครื่องมือวัด	10
2.3.1 โวลต์มิเตอร์กระแสสลับอิเล็กทรอนิกส์	10
2.3.2 แอมมิเตอร์กระแสสลับแบบแคลมป์ออน	14
2.3.3 วงจรเปลี่ยนแรงดันไฟฟ้าเป็นกระแสไฟฟ้า	15
2.3.4 อุปกรณ์ที่ใช้ในการตรวจวัดอุณหภูมิ	17
2.4 การรับส่งข้อมูลแบบอนุกรม	22
2.4.1 พื้นฐานการรับส่งข้อมูล	22
2.4.2 รูปแบบของการรับส่งข้อมูลแบบอนุกรม	24
2.4.3 MCS – 51 กับการรับส่งข้อมูลแบบอนุกรม	25
2.4.4 รีจิสเตอร์ควบคุมพอร์ตอนุกรม	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.4.5 โหมดการรับ - ส่งข้อมูล	28
2.4.6 การกำหนดค่าเริ่มต้นให้รีจิสเตอร์ในการรับส่งข้อมูล	31
2.4.7 อัตราการส่งข้อมูลของพอร์ตอนุกรม	33
2.4.8 การสื่อสารข้อมูลระหว่าง MCS – 51 หลายตัว	36
2.5 การอินเตอร์รัปต์	37
2.5.1 ขบวนการอินเตอร์รัปต์	37
2.5.2 สัญญาณอินเตอร์รัปต์	38
2.5.3 การทำงานของระบบหลังถูกอินเตอร์รัปต์	41
2.5.4 การออกแบบโปรแกรมอินเตอร์รัปต์	42
2.6 การเขียนโปรแกรมด้วยภาษาซี	45
2.6.1 โปรแกรม C51	45
2.6.2 คำสั่งพื้นฐานและการประกาศตัวแปร	45
2.6.3 โครงสร้างของภาษาซี	47
2.6.4 ตัวดำเนินการในภาษาซี	48
2.6.5 ประโยคควบคุมในภาษาซี	50
2.6.6 การทำซ้ำ	52
2.6.7 อาร์เรย์ พอยน์เตอร์ และสตริงเจอร์	54
2.6.8 การเขียนโปรแกรมจัดการหน่วยความจำ	57
บทที่ 3 การออกแบบ การสร้าง และการทดลอง	62
3.1 กล่าวนำ	62
3.2 การออกแบบวงจร	62
3.3 ชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ	64
3.3.1 วงจรควบคุม	64
3.3.2 วงจรตรวจจับอุณหภูมิ	65
3.3.3 วงจรดิจิตอลมิเตอร์	65
3.3.4 วงจรวัดแรงดันไฟฟ้า	67
3.3.5 วงจรวัดกระแสไฟฟ้า	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
3.3.6 วงจรเปิด-ปิดคอมเพรสเซอร์	68
3.3.7 วงจรแหล่งจ่ายไฟ	69
3.4 ชุดแปลงสัญญาณ RS232/RS485	69
3.5 ภาคแสดงผลบนเครื่องคอมพิวเตอร์	70
3.5.1 หน้าจอแสดงข้อมูลการใช้พลังงาน	71
3.5.2 หน้าจอแสดงข้อมูลทั้งหมด	71
บทที่ 4 การทดลองและผลการทดลอง	73
4.1 กล่าวนำ	73
4.2 การทดลองการแสดงผลเมื่อเชื่อมต่อกับ โหลดจำลองของระบบปรับอากาศแบบแยกส่วน	73
4.2.1 การทดลอง	73
4.2.2 ผลการทดลอง	74
4.3 การทดลองการแสดงผลเมื่อเชื่อมต่อชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ	75
4.3.1 การทดลอง	75
4.3.2 ผลการทดลอง	76
4.4 การทดลองการเรียกดูข้อมูลย้อนหลัง	77
4.4.1 การทดลอง	77
4.4.2 ผลการทดลอง	77
บทที่ 5 บทสรุป	78
5.1 บทสรุป	78
5.2 ปัญหาและแนวทางแก้ไข	78
5.3 แนวทางการพัฒนา	79
บรรณานุกรม	80
ภาคผนวก ก เครื่องต้นแบบ	81
ภาคผนวก ข วงจรและแผ่นวงจรพิมพ์	85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
ภาคผนวก ค รายการอุปกรณ์	96
ภาคผนวก ง แผนผังการทำงานและรหัสต้นฉบับของ โปรแกรม	102
ภาคผนวก จ คู่มือการใช้งาน	120
ภาคผนวก ฉ รายละเอียดและคุณสมบัติของอุปกรณ์	128
ประวัติผู้แต่ง	140



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 เปรียบเทียบคุณสมบัติเทอร์โมมิเตอร์แก๊สกับเทอร์โมมิเตอร์ปรอท	18
2.2 คุณสมบัติของอาร์ทีดีที่ทำมาจากโลหะประเภทต่างๆ	19
2.2 (ต่อ) คุณสมบัติของอาร์ทีดีที่ทำมาจากโลหะประเภทต่างๆ	20
2.3 บิตต่างๆ ของรีจิสเตอร์ SCON	27
2.4 โหมดต่างๆ ของการรับส่งแบบอนุกรม	27
2.5 บิตต่างๆ ของรีจิสเตอร์ IE	38
2.6 การจัดลำดับความสำคัญของการอินเตอร์รัปต์	39
2.7 บิตและหน้าที่ต่างๆ ของรีจิสเตอร์ IP	40
2.8 แฟลคที่จะทำงานเมื่อถูกอินเตอร์รัปต์	40
2.8 (ต่อ) แฟลคที่จะทำงานเมื่อถูกอินเตอร์รัปต์	41
2.9 อินเตอร์รัปต์เวกเตอร์ของ MCS-51	41
2.10 การประกาศตัวแปรของภาษาซี	47
4.1 ผลการทดลองการเชื่อมต่อกับ โหลดจำลองของระบบปรับอากาศแบบแยกส่วน	74
ค.1 รายการของอุปกรณ์ของวงจรควบคุม	97
ค.2 รายการของอุปกรณ์ของวงจรดิจิทัลมิเตอร์	97
ค.2 (ต่อ) รายการของอุปกรณ์ของวงจรดิจิทัลมิเตอร์	98
ค.3 รายการของอุปกรณ์ของวงจรวัดแรงดันไฟฟ้า	98
ค.3 (ต่อ) รายการของอุปกรณ์ของวงจรวัดแรงดันไฟฟ้า	99
ค.4 รายการของอุปกรณ์ของวงจรวัดกระแสไฟฟ้า	99
ค.4 (ต่อ) รายการของอุปกรณ์ของวงจรวัดกระแสไฟฟ้า	100
ค.5 รายการของอุปกรณ์ของวงจรเปิด-ปิดคอมเพรสเซอร์	100
ค.6 รายการของอุปกรณ์ของวงจรแหล่งจ่ายไฟ	100
ค.6 (ต่อ) รายการของอุปกรณ์ของวงจรแหล่งจ่ายไฟ	101
ค.7 รายการของอุปกรณ์ของวงจรแปลงสัญญาณ RS232/RS485	101

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 แบบต่างๆ ของชุดคอยล์เย็น	3
2.1 (ต่อ) แบบต่างๆ ของชุดคอยล์เย็น	4
2.2 โครงสร้างภายในของชุดคอยล์เย็น	4
2.3 ใบพัดแบบครอส โพลว์	5
2.4 ชุดคอนเดนซิ่งยูนิท	7
2.5 วงจรไฟฟ้าเครื่องปรับอากาศแบบแยกส่วน	8
2.6 วงจรไฟฟ้าเครื่องปรับอากาศแบบแยกส่วนคอยล์ของ แม่เหล็กคอนแทกเตอร์ใช้ไฟ 24 โวลต์	9
2.7 โวลต์มิเตอร์กระแสสลับแบบออปแอมป์เป็นวงจรขยายตามแรงดันไฟฟ้า	10
2.8 โวลต์มิเตอร์แบบเรียงกระแสไฟฟ้าครึ่งคลื่นที่มีความเที่ยงตรง	11
2.9 โวลต์มิเตอร์กระแสสลับแบบเรียงกระแสไฟฟ้าครึ่งคลื่นที่มีความเที่ยงตรง	11
2.10 โวลต์มิเตอร์กระแสสลับแบบเปลี่ยนแรงดันไฟฟ้าเป็นกระแสไฟฟ้า	12
2.11 โวลต์มิเตอร์กระแสสลับเรียงกระแสไฟฟ้าเต็มคลื่นแบบบริดจ์	13
2.12 โวลต์มิเตอร์กระแสสลับเรียงกระแสไฟฟ้าเต็มคลื่นแบบครึ่งบริดจ์	13
2.13 โครงสร้างของแอมมิเตอร์กระแสสลับแบบแคลมป์ออนมิเตอร์	14
2.14 วงจรของแคลมป์ออน โวลต์ – แอมมิเตอร์	15
2.15 วงจรเปลี่ยนแรงดันไฟฟ้าเป็นกระแสไฟฟ้าแบบโหลดต่อกราวด์	16
2.16 วงจรเปลี่ยนแรงดันไฟฟ้าเป็นกระแสไฟฟ้าแบบโหลดต่อกราวด์	17
2.17 คุณลักษณะของกระแสกับเวลาของเทอร์มิสเตอร์	21
2.18 การรับส่งข้อมูลแบบขนาน	22
2.19 การส่งข้อมูลแบบซิงโครนัส	23
2.20 การส่งข้อมูลแบบอนุกรม	24
2.21 บิตต่างๆ ของข้อมูลที่ส่งแบบอนุกรม	24
2.22 การรับส่งข้อมูลระหว่างรีจิสเตอร์ภายใน	26
2.23 แผนผังเวลาการส่งข้อมูล	29
2.24 แผนผังเวลาการรับข้อมูล	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.25 การรับส่งข้อมูลออกโดยใช้พริจิสเตอร์ช่วย	30
2.26 การรับส่งข้อมูลในโหมด 1	30
2.27 แสดงการกำหนดอัตราในโหมดต่างๆ	34
2.28 การต่อ MCS – 51 แบบ Multiprocessor	36
2.29 ขั้นตอนการทำงานของโปรแกรมเมื่อถูกอินเทอร์รัปต์	37
2.30 รีจิสเตอร์ต่างๆ ที่เกี่ยวข้องกับการอินเทอร์รัปต์	39
2.31 การจัดตำแหน่งโปรแกรมในหน่วยความจำ	43
2.32 ประโยคทดสอบการทำงานของสเตตเมนต์แบบทางเลือกเดียว	50
2.33 ประโยคทดสอบการทำงานของสเตตเมนต์แบบ 2 ทางเลือก	51
2.34 ประโยคตรวจสอบเงื่อนไขก่อนการนำเข้า	53
3.1 (ก) แผนผังการทำงานของระบบบันทึกและควบคุมการใช้พลังงาน ของระบบปรับอากาศแบบแยกส่วน	63
3.1 (ข) ชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ	63
3.2 วงจรควบคุม	64
3.3 วงจรตรวจจับอุณหภูมิ	65
3.4 วงจรดิจิตอลมิเตอร์	66
3.5 วงจรวัดแรงดันไฟฟ้า	68
3.6 วงจรวัดกระแสไฟฟ้า	68
3.7 วงจรเปิด-ปิดคอมเพรสเซอร์	68
3.8 วงจรแหล่งจ่ายไฟ	69
3.9 วงจรแปลงสัญญาณ RS232/RS485	70
3.10 หน้าจอแสดงผลข้อมูลการใช้พลังงาน	71
3.11 หน้าจอแสดงข้อมูลทั้งหมด	72
4.1 การเชื่อมต่อกับโหนดจำลองของระบบปรับอากาศแบบแยกส่วน	73
4.2 การเชื่อมต่อระบบการทดลอง	75
4.3 ผลการทดลองการแสดงผลเมื่อเชื่อมต่อระบบบันทึกและควบคุมการใช้พลังงาน ของระบบปรับอากาศแบบแยกส่วนเข้ากับเครื่องปรับอากาศ	76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.4 ผลการทดลองการเรียกดูข้อมูลย้อนหลัง	77
ก.1 ภาพด้านหน้าและด้านบนของชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ	82
ก.2 ภาพด้านหลังของชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ	82
ก.3 ภาพภายในของชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ	83
ก.4 ภาพด้านหน้าและด้านบนชุดแปลงสัญญาณ RS232/RS485	83
ก.5 ภาพด้านหลังของชุดแปลงสัญญาณ RS232/RS485	84
ก.6 ภาพภายในของชุดแปลงสัญญาณ RS232/RS485	84
ข.1 วงจรควบคุม	86
ข.2 แผ่นวงจรพิมพ์วงจรควบคุม	87
ข.3 ตำแหน่งการวางอุปกรณ์แผ่นวงจรพิมพ์วงจรควบคุม	87
ข.4 วงจรดิจิทัลมิเตอร์	88
ข.5 แผ่นวงจรพิมพ์วงจรดิจิทัลมิเตอร์	89
ข.6 ตำแหน่งการวางอุปกรณ์แผ่นวงจรพิมพ์วงจรดิจิทัลมิเตอร์	89
ข.7 วงจรวัดแรงดันไฟฟ้า	90
ข.8 แผ่นวงจรพิมพ์วงจรวัดแรงดันไฟฟ้า	90
ข.9 ตำแหน่งการวางอุปกรณ์แผ่นวงจรพิมพ์วงจรวัดแรงดันไฟฟ้า	91
ข.10 วงจรวัดกระแสไฟฟ้า	91
ข.11 แผ่นวงจรพิมพ์วงจรวัดกระแสไฟฟ้า	91
ข.12 ตำแหน่งการวางอุปกรณ์แผ่นวงจรพิมพ์วงจรวัดกระแสไฟฟ้า	92
ข.13 วงจรเปิด – ปิดคอมเพรสเซอร์	92
ข.14 แผ่นวงจรพิมพ์วงจรเปิด – ปิดคอมเพรสเซอร์	92
ข.15 ตำแหน่งการวางอุปกรณ์แผ่นวงจรพิมพ์วงจรเปิด – ปิดคอมเพรสเซอร์	93
ข.16 วงจรแหล่งจ่ายไฟ	93
ข.17 แผ่นวงจรพิมพ์วงจรแหล่งจ่ายไฟ	93
ข.18 ตำแหน่งการวางอุปกรณ์แผ่นวงจรพิมพ์วงจรแหล่งจ่ายไฟ	94
ข.19 วงจรแปลงสัญญาณ RS232/RS485	94

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
ข.20 แผ่นวงจรพิมพ์แปลงสัญญาณ RS232/RS485	95
ข.21 ตำแหน่งการวางอุปกรณ์แผ่นวงจรพิมพ์วงจรแปลงสัญญาณ RS232/RS485	95
ง.1 ฟังงานของโปรแกรมหลัก C 51	103
ง.2 ฟังงานของโปรแกรมอ่านแรงดันไฟฟ้าและกระแสไฟฟ้า	104
ง.3 ฟังงานของโปรแกรมวัดอุณหภูมิ	105
ง.4 ฟังงานของโปรแกรมหลักการรับส่งค่าที่เครื่องคอมพิวเตอร์	106
จ.1 ส่วนประกอบและปุ่มควบคุมด้านหน้าของชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ	122
จ.2 ส่วนประกอบและปุ่มควบคุมด้านหลังของชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ	123
จ.3 ส่วนประกอบและปุ่มควบคุมด้านหน้าของชุดแปลงสัญญาณ RS232/RS485	124
จ.4 ส่วนประกอบและปุ่มควบคุมด้านหลังของชุดแปลงสัญญาณ RS232/RS485	124
จ.5 การเชื่อมต่อระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วน	125
จ.6 ค่าต่างๆ ที่วัดได้แสดงผลบนเครื่องคอมพิวเตอร์	126

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

ปัจจุบันระดับการใช้พลังงานไฟฟ้าภายในประเทศได้เพิ่มปริมาณสูงขึ้น โดยเฉพาะในอาคารเรียนทั่วไป จากการสังเกตการใช้พลังงานไฟฟ้าภายในห้องเรียนเพราะพฤติกรรมของผู้ใช้จะเปิดใช้เครื่องปรับอากาศพร้อมกันหลายเครื่อง ในช่วงเวลาคาบเรียนแรก ซึ่งจะส่งผลให้ระดับของพลังงานไฟฟ้าที่ใช้สูงกว่าระดับปกติที่การไฟฟ้ากำหนดไว้ จึงทำให้ต้องเสียค่าใช้จ่ายเพิ่มขึ้นจากระดับปกติ และการปรับตั้งค่าอุณหภูมิของเครื่องปรับอากาศต่ำกว่า 25 °C รวมทั้งการที่ผู้ใช้เปิดเครื่องปรับอากาศทำงานผิดปกติ ส่งผลให้เกิดการสิ้นเปลืองพลังงานไฟฟ้า

1.2 ชีตความสามารถของโครงการ

โครงการนี้มีชีตความสามารถดังนี้

- 1) สามารถแสดงผลข้อมูลเป็นกราฟและตัวเลขออกทางจอเครื่องคอมพิวเตอร์ได้
- 2) สามารถบันทึกและพิมพ์ข้อมูลได้ ตามช่วงเวลา/วัน/สัปดาห์/เดือน/ปี ที่ต้องการ
- 3) สามารถปรับตั้งค่าอุณหภูมิได้จากเครื่องคอมพิวเตอร์ศูนย์กลาง
- 4) สามารถปิดคอมเพรสเซอร์เมื่อตรวจสอบว่าไม่มีประสิทธิภาพในการทำงาน
- 5) สามารถส่งข้อมูลผ่านระบบสื่อสารแบบอนุกรม RS 485
- 6) สามารถแจ้งเตือนให้มีการซ่อมบำรุงจากหน้าจอเครื่องคอมพิวเตอร์ที่ใช้ควบคุม
- 7) ไม่ส่งผลกระทบต่อพฤติกรรมการใช้งานตามปกติของผู้ใช้

1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปฏิญานิพนธ์ฉบับนี้แบ่งออกเป็นบทต่างๆ เพื่อสะดวกต่อการศึกษาและการทำความเข้าใจ โดยในแต่ละบทจะประกอบด้วยเนื้อหาดังต่อไปนี้

บทที่ 1 กล่าวถึงความเป็นมาและความสำคัญของปฏิญานิพนธ์ ชีตความสามารถของโครงการและเนื้อหาในบทต่างๆ โดยสังเขป

บทที่ 2 ประกอบด้วยทฤษฎีต่างๆ เกี่ยวกับ เครื่องปรับอากาศแบบแยกส่วน โวลท์มิเตอร์ กระแสสลับ แอมมิเตอร์กระแสสลับแบบแคลมป์ออน วงจรเปลี่ยนแรงดันไฟฟ้าเป็นกระแสไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ที่ใช้ในการตรวจวัดอุณหภูมิ การรับส่งข้อมูลแบบอนุกรม การอินเทอร์รัพท์ และการเขียนโปรแกรมด้วยภาษาซี

บทที่ 3 กล่าวถึงเนื้อหาที่เกี่ยวกับแผนผังการทำงานของโครงการ ผังวงจรต่างๆ ที่ใช้ในโครงสร้าง ตลอดจนการออกแบบและการสร้างส่วนประกอบต่างๆ เช่น วงจร โวลท์มิเตอร์ แอมป์มิเตอร์ วงจรตรวจวัดอุณหภูมิ วงจรควบคุมการเปิด-ปิดระบบปรับอากาศแบบแยกส่วน และการโปรแกรมควบคุมและประมวลค่าแรงดัน กระแส และอุณหภูมิของระบบปรับอากาศแบบแยกส่วน พร้อมทั้งการทำงานของส่วนประกอบต่างๆ โดยละเอียด

บทที่ 4 ประกอบด้วยการทดลองและผลการทดลองต่างๆ เช่น การทดลองการแสดงผลค่าแรงดัน กระแส และอุณหภูมิ การทดลองควบคุมการทำงานของระบบปรับอากาศแบบแยกส่วน

บทที่ 5 เป็นการสรุปผลการจัดทำโครงการ ปัญหาที่เกิดขึ้นและแนวทางในการแก้ไขรวมทั้งแนวทางในการพัฒนา

ภาคผนวก ก แสดงภาพเครื่องต้นแบบ การติดตั้ง การเชื่อมต่อกับอุปกรณ์อื่นๆ ขณะใช้งานจริง

ภาคผนวก ข ประกอบด้วยผังรายละเอียดวงจรและวงจรพิมพ์

ภาคผนวก ค แสดงรายการอุปกรณ์ที่ใช้ภายในแต่ละวงจร

ภาคผนวก ง แสดงแผนผังการทำงานและรหัสต้นฉบับของโปรแกรมทั้งหมดที่สร้างขึ้นเพื่อประกอบการทำงานของโครงการ

ภาคผนวก จ เป็นคู่มือการใช้งาน โครงการระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วน

ภาคผนวก ฉ แสดงรายละเอียดและคุณสมบัติของอุปกรณ์ที่ใช้ในโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 กล่าวนำ

ในบทนี้จะกล่าวถึงทฤษฎีพื้นฐานต่างๆ ที่นำมาใช้ประกอบในการจัดทำระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วน ซึ่งเนื้อหาต่างๆ จะได้กล่าวถึงต่อไป

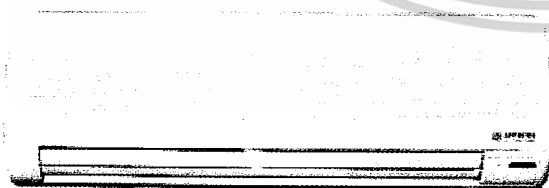
2.2 เครื่องปรับอากาศแบบแยกส่วน

เครื่องปรับอากาศแบบแยกส่วนเป็นแบบที่นิยมใช้กันมากตามบ้านพักอาศัยและสำนักงานในปัจจุบัน เพราะเสียงเงียบกว่าและการติดตั้งสะดวกกว่าเนื่องจากไม่ต้องร้อยหน้าต่างออกเช่นเดียวกับแบบติดหน้าต่าง เพียงแต่เจาะผนังเป็นรูสำหรับร้อยท่อชักชั้น ท่อลิควิด และสายไฟ เพียงเล็กน้อยเท่านั้น เครื่องปรับอากาศแบบแยกส่วนนี้จะแบ่งระบบวงจรน้ำยาเครื่องออกเป็น 2 ส่วนคือ ชุดคอยล์น้ำเย็นหรืออีวาพอเรเตอร์และชุดคอนเดนซิ่งยูนิต

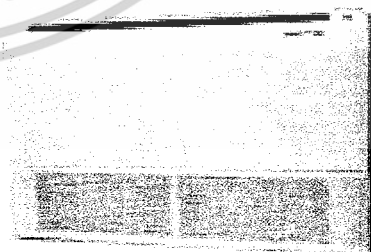
2.2.1 ชุดคอยล์น้ำเย็นหรืออีวาพอเรเตอร์

เป็นส่วนที่ติดตั้งอยู่ภายในห้องปรับอากาศ ซึ่งสามารถแบ่งตามลักษณะการติดตั้งได้เป็น 3 แบบ ดังแสดงในรูปที่ 2.1 คือ

- 1) แบบติดผนัง
- 2) แบบตั้งพื้น
- 3) แบบแขวนเพดาน



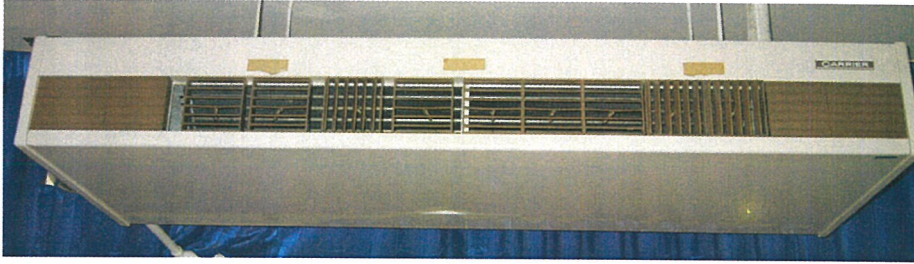
(ก) แบบติดผนัง



(ข) แบบตั้งพื้น

รูปที่ 2.1 แบบต่างๆ ของชุดคอยล์เย็น

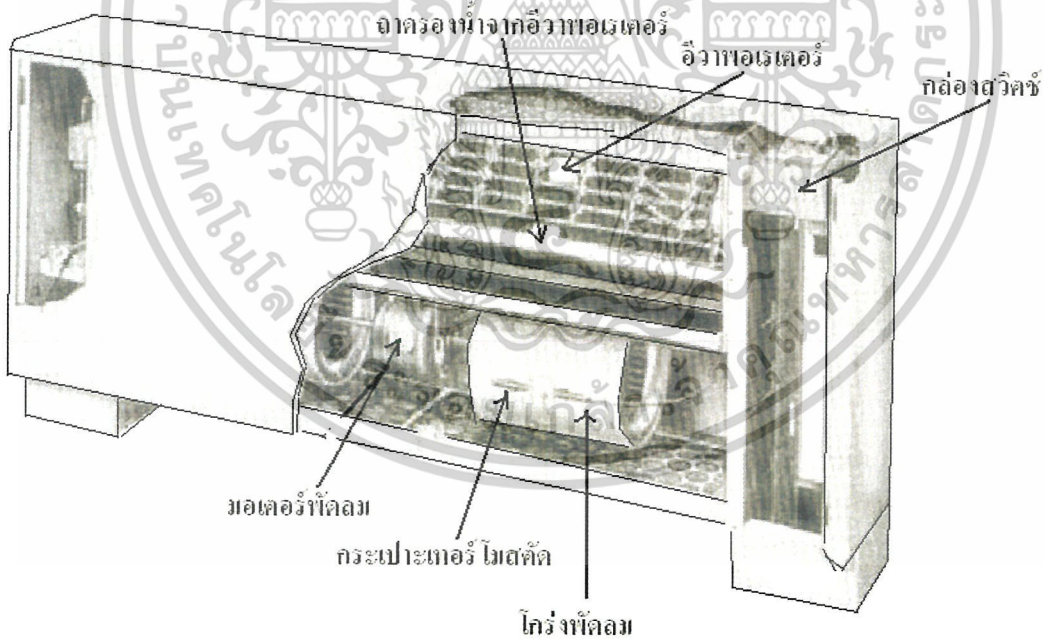
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ค) แบบแขวนเพดาน

รูปที่ 2.1 (ต่อ) แบบต่างๆ ของชุดคอยล์เย็น

โครงสร้างภายในของชุดคอยล์เย็น ไม่ว่าจะเป็นแบบตั้งพื้น แบบแขวนเพดาน หรือติดผนัง จะมีโครงสร้างหลักใหญ่ๆ เหมือนกัน



รูปที่ 2.2 โครงสร้างภายในของชุดคอยล์เย็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างภายในของชุดคอยล์เย็น ประกอบด้วย

1) คอยล์เย็นหรืออีวาพอเรเตอร์คอยล์ เป็นขดท่อทองแดงและมีครีบอลูมิเนียมช่วยเพิ่มพื้นที่ผิวในการดูดซับปริมาณความร้อนจากอากาศในห้อง ขณะที่น้ำยา R-22 ภายในระบบตรงบริเวณนี้ระเหยเปลี่ยนสถานะเป็นแก๊ส จะดูดซับปริมาณความร้อนผ่านผิวท่อทางเดินน้ำยา เข้าไปยังน้ำยาภายในระบบ ทำให้อุณหภูมิโดยรอบคอยล์เย็นลดต่ำลง

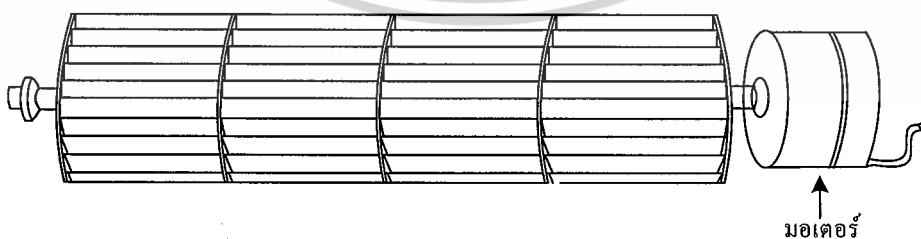
2) เทอร์โมสแตติกเอกซ์แพนชันวาล์ว ส่วนใหญ่ชุดควบคุมการไหลของน้ำยาสำหรับเครื่องปรับอากาศแบบแยกส่วน มักจะใช้เทอร์โมสแตติกเอกซ์แพนชันวาล์ว ซึ่งเป็นแบบที่มีหลักในการปรับควบคุมการไหลของน้ำยาได้ดี

3) ท่อลิควิด เป็นท่อที่ไปต่อเข้ากับชุดคอนเดนซิ่งยูนิต ให้น้ำยาเหลวที่กลั่นตัวเรียบร้อยแล้วจากคอนเดนเซอร์ ส่งผ่านท่อนี้เข้ายังเอกซ์แพนชันวาล์ว

4) ท่อชักชั้น เป็นอีกท่อหนึ่งที่ไปต่อเข้ากับชุดคอนเดนซิ่งยูนิตตรงทางดูดของคอมเพรสเซอร์ ให้น้ำยาในสถานะแก๊สที่มีอุณหภูมิต่ำและความดันต่ำจากคอยล์เย็น ถูกดูดผ่านท่อชักชั้นเข้ายังคอมเพรสเซอร์

5) มอเตอร์พัดลม สำหรับชุดคอยล์เย็นแบบตั้งพื้น แขนวนเพดาน หรือซ่อนฝ้า มอเตอร์พัดลมจะเป็นแบบ 2 แกน หมุนใบพัดหัวท้าย เพื่อดูดอากาศจากภายในห้องมาเป่าผ่านคอยล์เย็น กลับเข้าไปหมุนเวียนภายในห้อง

6) ใบพัด ใบพัดที่ใช้สำหรับดูดเป่าอากาศหมุนเวียนกระจายความเย็นภายในห้องจะเป็นใบพัดแบบกรงกระรอกหรือแบบเซอร์รอก โค เช่นเดียวกับใช้ในเครื่องปรับอากาศแบบหน้าต่าง แต่สำหรับชุดคอยล์เย็นแบบติดผนังรุ่นใหม่ ๆ เนื่องจากการออกแบบต้องการให้มีความบางมากๆ ดังนั้นใบพัดจึงต้องมีขนาดเล็ก แต่เพื่อให้ได้ปริมาณลมที่ดูดเป่าผ่านคอยล์เย็นเท่าเดิม จึงออกแบบใบพัดให้ยาวขึ้น เรียกใบพัดแบบนี้ว่า แบบครอส โฟลว์ (Cross - Flow Fan) ดังแสดงในรูปที่ 2.3



รูปที่ 2.3 ใบพัดแบบครอส โฟลว์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) **ถาดรองน้ำหยด** จะติดตั้งอยู่ด้านล่างของชุดคอยล์เย็น เมื่อไอน้ำในอากาศในห้องปรับอากาศกระทบความเย็น จะกลั่นตัวเป็นหยดน้ำรอบๆ คอยล์เย็น เมื่อน้ำเหล่านี้มีมากขึ้นจะไหลลงสู่ถาดรองน้ำหยดทิ้งภายนอกทางรูท่อน้ำทิ้ง ซึ่งจะเป็นการลดความชื้นของอากาศภายในห้องให้อยู่ในสภาวะที่เหมาะสม

8) **ฟิลเตอร์กรองอากาศ** ติดตั้งอยู่ที่ทางลมกลับ สำหรับกรองฝุ่นละอองของอากาศภายในห้องไม่ให้เข้าไปอุดตันในครีบบของคอยล์น้ำเย็น

นอกจากนี้ในส่วนของชุดคอยล์เย็นยังมีอุปกรณ์ควบคุมทางไฟฟ้าติดอยู่ทางด้านบน หรือด้านหน้าของเครื่อง ประกอบด้วยสวิตช์ควบคุมความเร็วรอบของมอเตอร์พัดลม ซึ่งส่วนใหญ่จะเป็น 3 สปีด คือ H-M-L (High – Medium - Low) ซึ่งผู้ใช้สามารถปรับความเร็วของมอเตอร์พัดลมให้อยู่ในตำแหน่งหนึ่งตำแหน่งใดได้ตามความต้องการ และสวิตช์อีกตัวหนึ่งคือสวิตช์ควบคุมอุณหภูมิหรือเทอร์โมสตัต ผู้ใช้สามารถปรับตั้งอุณหภูมิภายในห้องในช่วงที่ต้องการได้ โดยการปรับที่สวิตช์เทอร์โมสตัตนี้

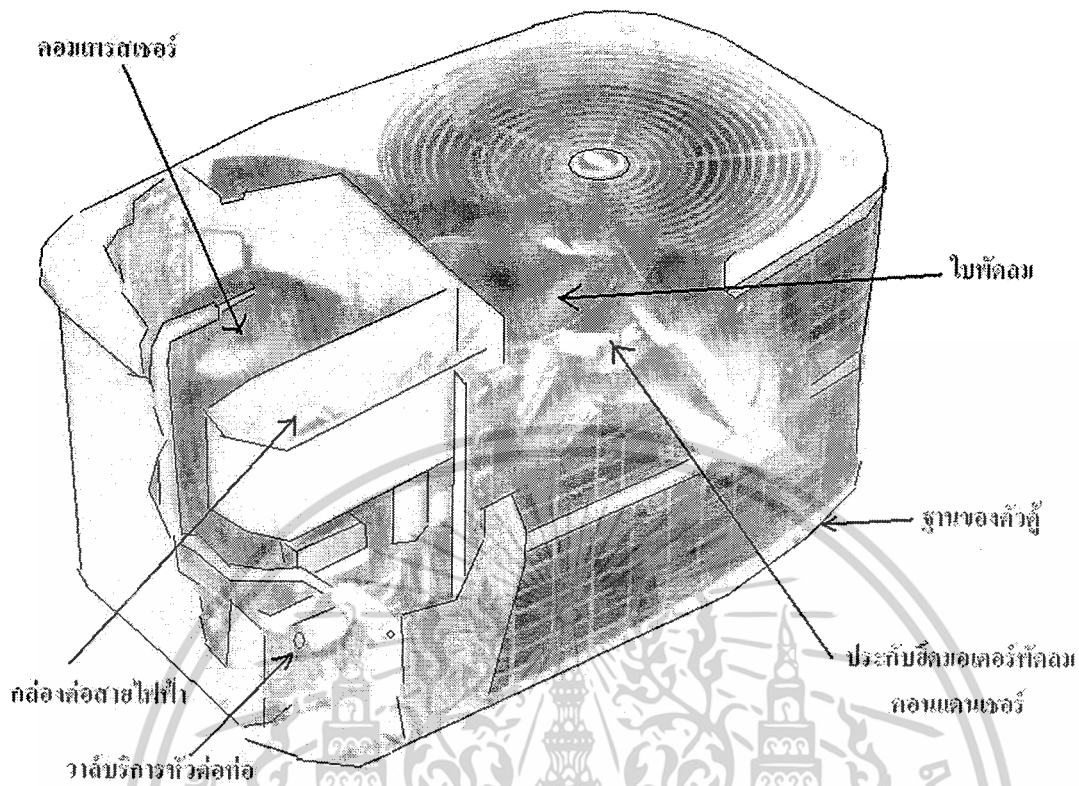
2.2.2 ชุดคอนเดนซิ่งยูนิต

คอนเดนซิ่งยูนิต (Condensing Unit) หมายถึง ส่วนของระบบซึ่งทำหน้าที่ทำให้น้ำภายในระบบกลั่นตัวเปลี่ยนสถานะจากแก๊สเป็นน้ำยาเหลว ซึ่งประกอบด้วยคอมเพรสเซอร์ ท่อทางอัดคอนเดนเซอร์และท่อพิก่น้ำยา รวมกันเป็นส่วนหนึ่งของระบบ ชุดคอนเดนซิ่งยูนิตที่ใช้กับเครื่องทำความเย็น โดยส่วนมากใช้แบบระบายความร้อนด้วยอากาศ เช่น ตู้เย็นและตู้แช่ที่ใช้ในครัวเรือน เครื่องปรับอากาศแบบแยกส่วน (Split type) ที่มีขนาดเล็กกว่า 5 ตัน

โดยปกติคอนเดนซิ่งยูนิตแบ่งออกได้ 2 แบบตามลักษณะของตัวกลางในการระบายความร้อนออกที่คอนเดนเซอร์

1. คอนเดนซิ่งยูนิตแบบระบายความร้อนด้วยอากาศ (Air-cooled Condensing Unit)
2. คอนเดนซิ่งยูนิตแบบระบายความร้อนด้วยน้ำ (Water-cooled Condensing Unit)

เป็นส่วนที่ติดตั้งอยู่นอกห้องปรับอากาศ ระบายความร้อนออกจากร้าน้ำยาเพื่อให้น้ำยาในสถานะแก๊สกลั่นตัวกลับเป็นน้ำยาเหลวอีกครั้งหนึ่ง รูปแบบของคอนเดนซิ่งยูนิตจะออกแบบแตกต่างกันไปตามแต่ละบริษัทผู้ผลิต อย่างไรก็ตาม โครงสร้างอุปกรณ์หลักใหญ่ของชุดคอนเดนซิ่งยูนิตไม่ว่าจะมีรูปแบบใดก็จะเหมือนกัน ดังแสดงในรูปที่ 2.4



รูปที่ 2.4 ชุดคอนเดนเซอร์ยูนิต

อุปกรณ์หลักของชุดคอนเดนเซอร์ยูนิต ประกอบด้วย

- 1) มอเตอร์คอมเพรสเซอร์ มอเตอร์คอมเพรสเซอร์ของเครื่องปรับอากาศแบบแยกส่วนนี้จะ เป็นแบบเฮิร์ตเมติก ซึ่งมีส่วนของมอเตอร์และคอมเพรสเซอร์อยู่ในตัวเรือนเดียวกัน ส่วนของ คอมเพรสเซอร์จะมีอยู่ 2 แบบ คือ แบบลูกสูบและแบบโรตารี เหมือนกันกับมอเตอร์คอมเพรสเซอร์ ของเครื่องปรับอากาศแบบติดหน้าต่าง
- 2) คอนเดนเซอร์ เป็นขดท่อทองแดงและมีครีบช่วยเพิ่มพื้นที่ผิวในการระบายความร้อน ออกจากน้ำยา ในสถานะแก๊สกลั่นตัวเป็นน้ำยาเหลว
- 3) มอเตอร์พัดลม จะช่วยในการระบายความร้อนออกจากน้ำยาในคอมเดนเซอร์ จะเห็นได้ ว่าเครื่องปรับอากาศแบบแยกส่วนมีมอเตอร์พัดลมแยกจากกันเป็น 2 ตัว คือ อยู่ในชุดคอยล์เย็น 1 ตัว และอยู่ที่ชุดคอนเดนเซอร์ยูนิตอีก 1 ตัว
- 4) ใบพัดลม ใบพัดลมสำหรับระบายความร้อนออกที่คอนเดนเซอร์จะเป็นใบพัดแบบ ธรรมชาติเช่นเดียวกับใบพัดระบายความร้อนคอนเดนเซอร์ของเครื่องปรับอากาศแบบติดหน้าต่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

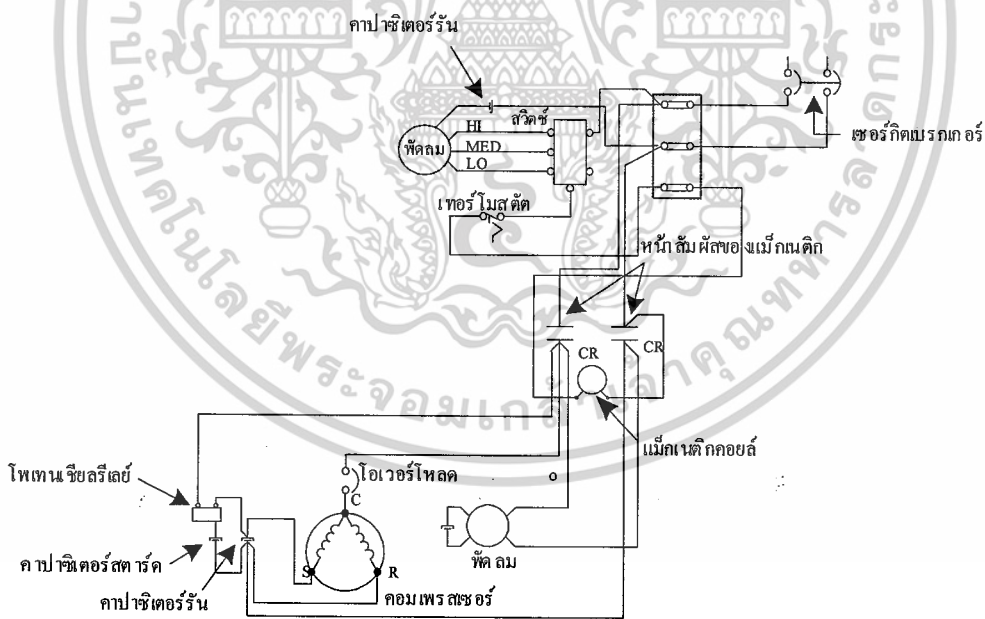
5) วาล์วบริการ ในชุดคอนเดนซึ่งยูนิตจะมีวาล์วบริการอยู่ 2 ตัวคือ วาล์วบริการทางท่อลิกวิดและวาล์วบริการทางท่อซัคชั่น

6) กล่องอุปกรณ์ไฟฟ้า ภายในจะบรรจุแม่เหล็กคอนแทกเตอร์ โอเวอร์โวลด์ รีเลย์ (ส่วนใหญ่เป็นชนิดโพเทนเชียลรีเลย์) คาปาซิเตอร์สตาร์ท คาปาซิเตอร์รัน ติดอยู่ภายในชุดคอนเดนซึ่งยูนิต

2.2.3 วงจรไฟฟ้าเครื่องปรับอากาศแบบแยกส่วน

วงจรไฟฟ้าเครื่องปรับอากาศแบบแยกส่วน จะแยกวงจรออกเป็น 2 ส่วนคือ ส่วนชุดของคอยล์น้ำเย็น และส่วนวงจรไฟของชุดคอน-เดนซึ่งยูนิต ซึ่งจะต้องมีมอเตอร์พัดลมระบายคอนเดนเซอร์เพิ่มขึ้นอีก 1 ตัว ดังรูปที่ 2.5

ในส่วนของวงจรไฟฟ้าชุดคอยล์เย็น จะประกอบด้วยขั้วพิกสาย (Terminal) และซีลิกเตอร์สวิทช์ซึ่งจะต่อเข้ากับมอเตอร์พัดลมที่ความเร็วรอบช้า ปานกลาง และความเร็วสูง มอเตอร์พัดลมนี้จะทำหน้าที่ดูดเป่าอากาศจากภายในห้อง ผ่านคอยล์เย็น และมีเทอร์โมสแตตซึ่งจะคอยควบคุมการทำงานของชุดคอมเดนซึ่งยูนิต

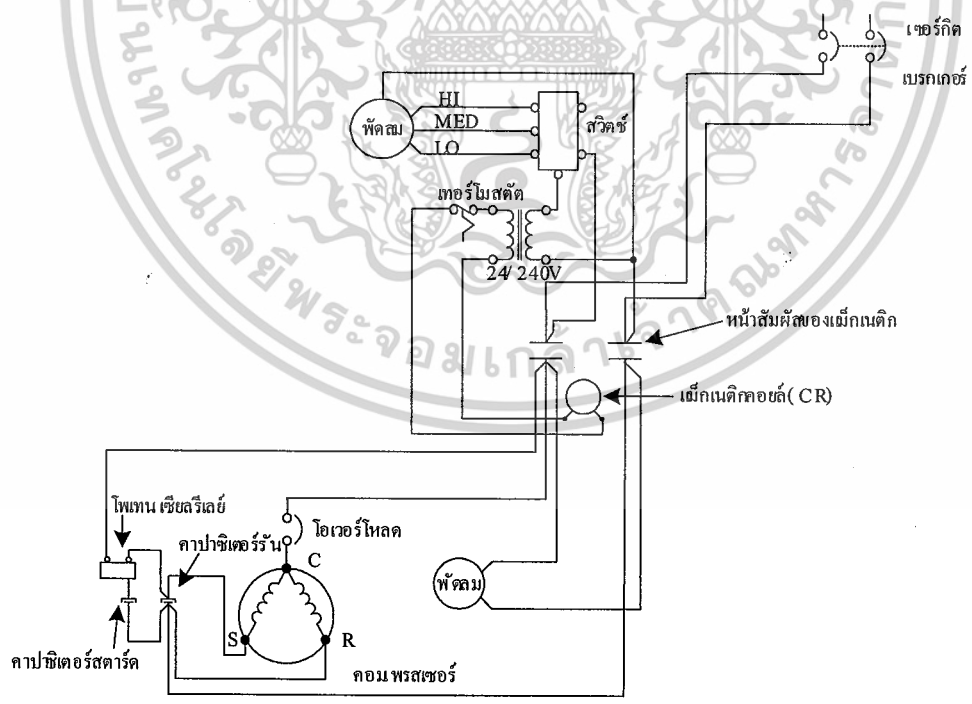


รูปที่ 2.5 วงจรไฟฟ้าเครื่องปรับอากาศแบบแยกส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรไฟฟ้าของชุดคอนเดนซึ่งยูนิตประกอบด้วยแม่เหล็กคอนแทกเตอร์ มอเตอร์คอมเพรสเซอร์และมอเตอร์พัดลมระบายความร้อนคอนเดนเซอร์ วงจรไฟควบคุมจากเทอร์โมสตัดจะผ่านเข้าเลี้ยงคอยล์ของแม่เหล็กคอนแทกเตอร์ ทำหน้าที่สัมผัสต่อ มอเตอร์คอมเพรสเซอร์และมอเตอร์พัดลมระบายคอนเดนเซอร์จะทำงานพร้อมกัน และเมื่ออุณหภูมิของอากาศภายในห้องลดลงถึงจุดที่ปรับตั้งไว้ หน้าสัมผัสของเทอร์โมสตัดจะจาก ตัดไฟที่มาเลี้ยงคอยล์ของแม่เหล็กคอนแทกเตอร์ ทำให้หน้าสัมผัสจาก หยุดการทำงานของอุปกรณ์ทางชุดคอนเดนซึ่งยูนิตทั้งหมด แต่ในขณะเดียวกัน มอเตอร์พัดลมของชุดคอยล์เย็นจะยังคงทำงานตามปกติอยู่จนกว่าอุณหภูมิของอากาศภายในห้องสูงขึ้น จนทำให้หน้าสัมผัสของเทอร์โมสตัดต่ออีกครั้งหนึ่ง ก็จะมีไฟเข้าเลี้ยงคอยล์ของแม่เหล็กคอนแทกเตอร์ อุปกรณ์ทางชุดคอนเดนซึ่งยูนิตก็จะเริ่มทำงานใหม่

ในกรณีที่แม่เหล็กคอนแทกเตอร์เป็นชนิดที่ต้องการแรงดันไฟฟ้า 24 โวลต์ เข้าที่ชุดคอนเดนซึ่งยูนิตจะต้องมีหม้อแปลงไฟฟ้า ลดแรงดันไฟฟ้าจาก 220 โวลต์เหลือ 24 โวลต์ดังรูปที่ 2.6 วงจรไฟควบคุมจากเทอร์โมสตัดที่ชุดคอยล์เย็นซึ่งเป็นไฟ 220 โวลต์ จะผ่านเข้าหม้อแปลงลดแรงดันลงเหลือ 24 โวลต์ เข้าเลี้ยงคอยล์ของแม่เหล็กคอนแทกเตอร์ ทำให้หน้าสัมผัสของแม่เหล็กคอนแทกเตอร์ต่อแรงดันไฟฟ้า 220 โวลต์ เข้าคอมเพรสเซอร์ และมอเตอร์พัดลมที่ชุดคอนเดนซึ่งยูนิตทำงาน



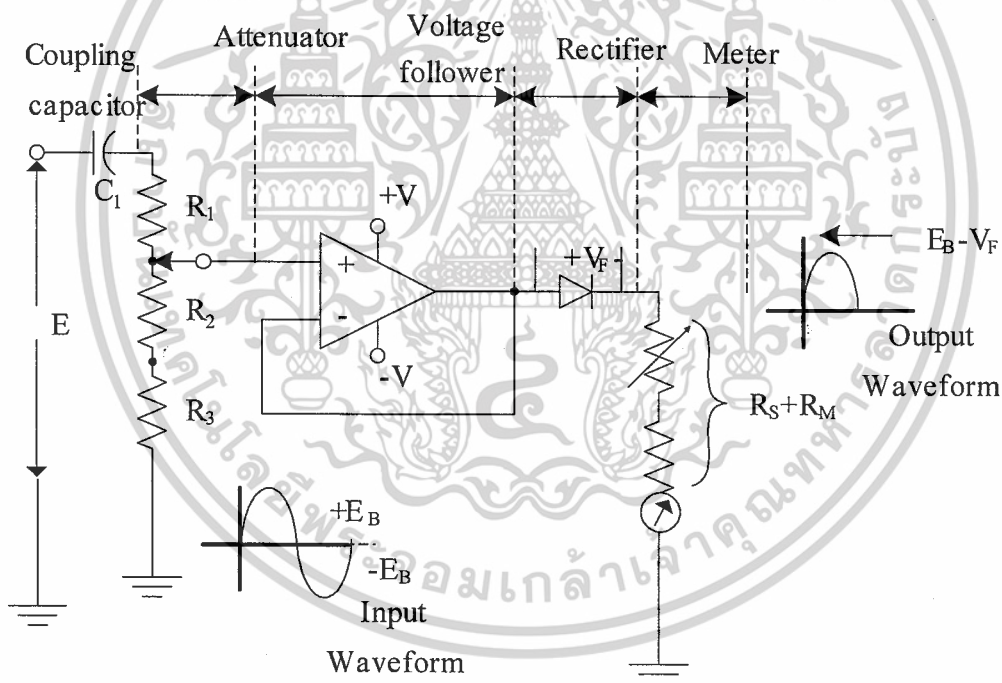
รูปที่ 2.6 วงจรไฟฟ้าเครื่องปรับอากาศแบบแยกส่วนคอยล์ของแม่เหล็กคอนแทกเตอร์ใช้ไฟ 24 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 เครื่องมือวัด

2.3.1 โวลต์มิเตอร์กระแสสลับอิเล็กทรอนิกส์

จากคิซออิเล็กทรอนิกส์ โวลต์มิเตอร์แบบวงจรขยายตามแรงดันไฟฟ้าจะทำให้เป็นวงจร โวลต์มิเตอร์กระแสสลับด้วยการต่อไดโอดอนุกรมกับขดลวดหมุนดังรูปที่ 2.7 จะเป็น โวลต์มิเตอร์กระแสสลับแบบเรียงกระแสไฟฟ้าครั้งคลื่น เมื่อป้อนแรงดันไฟฟ้ากระแสสลับ แรงดัน E จะผ่าน C_1 ที่มีหน้าที่กัน (Block) ไม่ให้แรงดันไฟฟ้ากระแสตรงผ่านเข้ามา แรงดันไฟฟ้ากระแสสลับ แรงดัน E จะถูกลดทอนลงเป็นแรงดันไฟฟ้าอินพุต E_B ป้อนให้ออปแอมป์ซึ่งเป็นแรงดันไฟฟ้าเอาต์พุตของออปแอมป์จะมีค่าเท่ากับแรงดันไฟฟ้าอินพุต E_B การจัดวงจรเช่นนี้จะมีปัญหาค่าผิดพลาดจากแรงดันไฟฟ้า V_F ของไดโอดเพราะว่าแรงดันไฟฟ้า V_0 ที่ตกคร่อมมิเตอร์จะมีค่าเท่ากับ $E_B - V_F = E_B - 0.7$ V ถ้า $E_B < 0.7$ V จะไม่สามารถวัดแรงดันค่าต่ำๆ ได้เลย

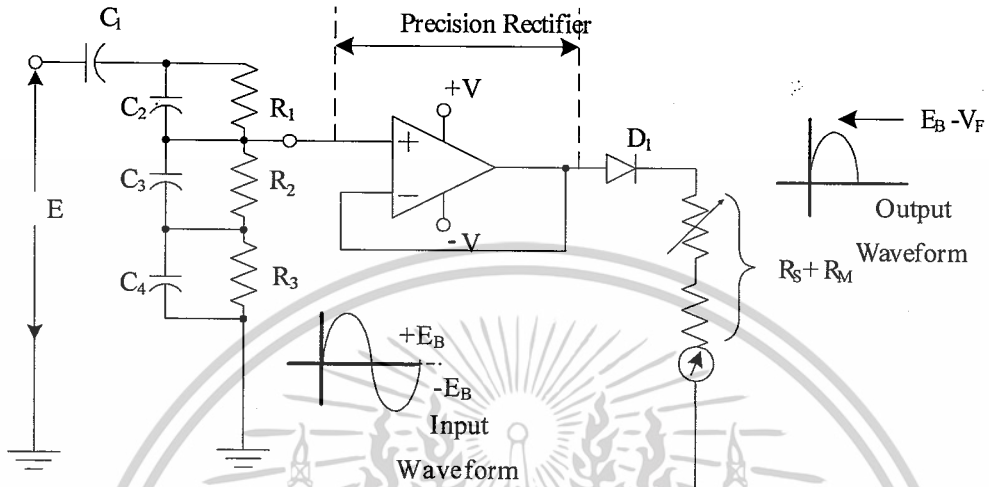


รูปที่ 2.7 โวลต์มิเตอร์กระแสสลับแบบออปแอมป์เป็นวงจรขยายตามแรงดันไฟฟ้า

การป้องกันไม่ให้เกิดค่าผิดพลาดจากแรงดันไฟฟ้า V_F แก้ไขด้วยการย้ายจุดต่อสายสัญญาณการป้อนกลับจากเอาต์พุตของออปแอมป์เปลี่ยนมาต่อที่ขาแคโรดของไดโอดดังรูปที่ 2.8 จะได้แรงดันไฟฟ้า $V_0 = E_B$ ตกคร่อมที่มิเตอร์จะมีความถูกต้องจากการวัดจึงเรียกว่าวงจรเรียงกระแสไฟฟ้าที่

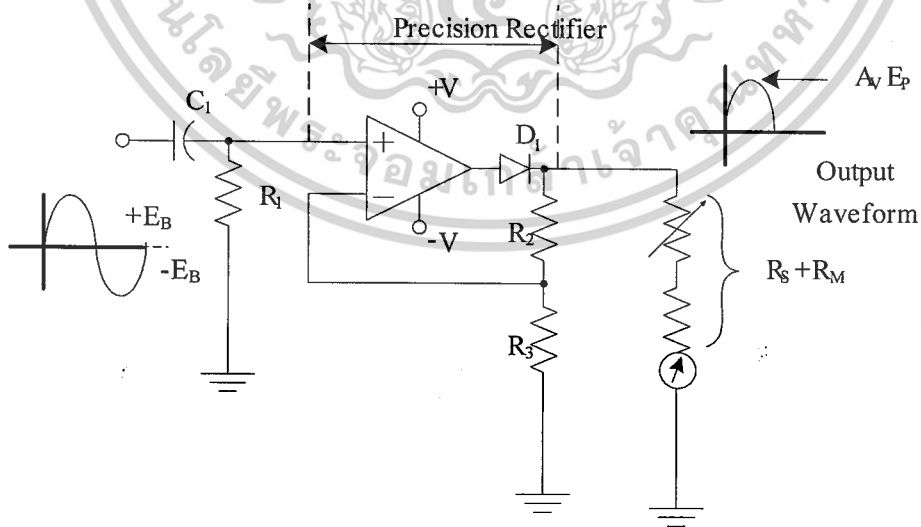
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีความเที่ยงตรง (Precision Rectifier) สังเกตว่ามี C_2, C_3, C_4 ต่อคร่อมตัวต้านทานของวงจรลัดตอน เพื่อชดเชยให้สัญญาณอินพุตไม่ให้ผิดเพี้ยนเรียกว่าคาปาซิเตอร์ชดเชย (Compensation Capacitor)



รูปที่ 2.8 โวลต์มิเตอร์กระแสสลับแบบเรียงกระแสไฟฟ้าครึ่งคลื่นที่มีความเที่ยงตรง

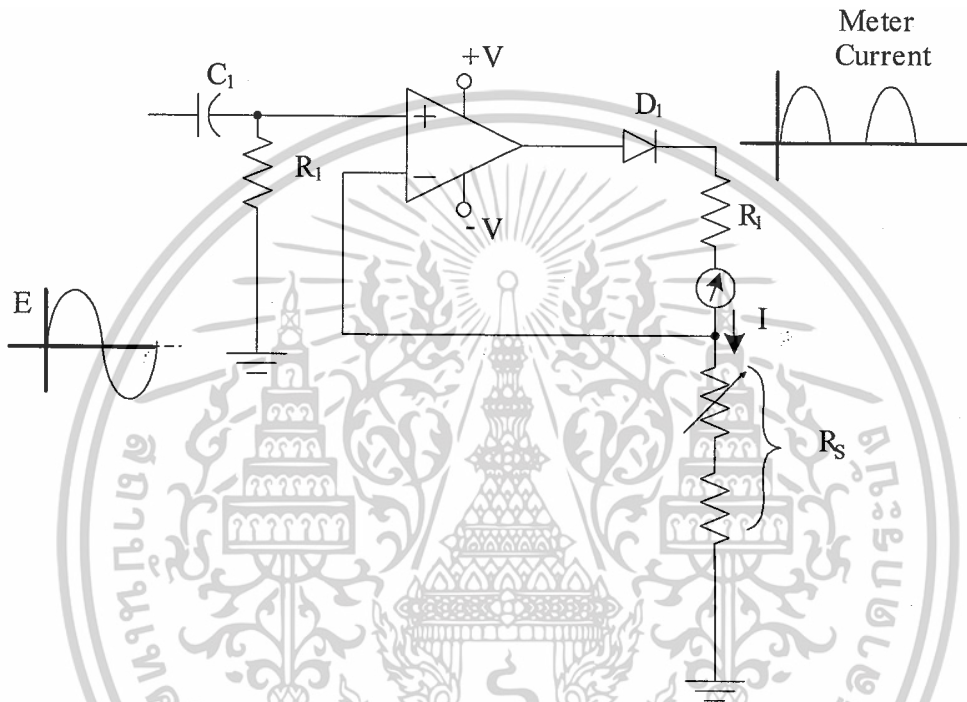
วงจรรูปที่ 2.9 เป็น โวลต์มิเตอร์กระแสสลับที่มีอัตราขยายแรงดันไฟฟ้า $A_V = \frac{R_2}{R_3} + 1$ แรงดันไฟฟ้าอินพุต E_p ป้อนให้วงจรได้ $V_O = A_V E_p$ ตกรวมมิเตอร์



รูปที่ 2.9 โวลต์มิเตอร์กระแสสลับแบบเรียงกระแสไฟฟ้าครึ่งคลื่นที่มีความเที่ยงตรง

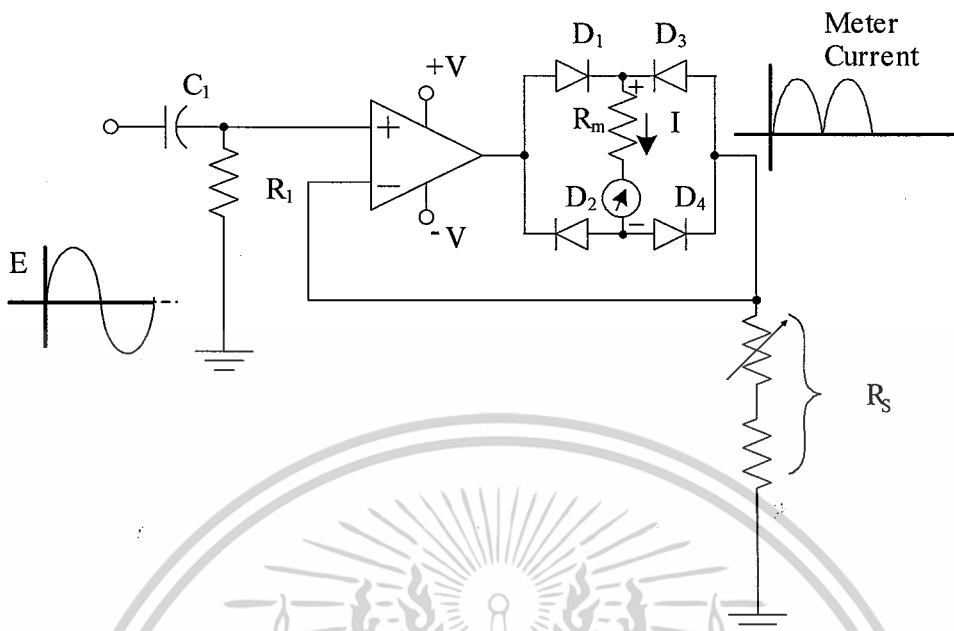
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรรูปที่ 2.10 เป็นวงจร โวลต์มิเตอร์กระแสสลับที่เปลี่ยนแรงดันไฟฟ้าเป็นกระแสไฟฟ้า ต่อขดลวดหมุนแทน R_2 เมื่อแรงดันไฟฟ้าอินพุตครึ่งไซเคิลบวก E_p ถูกป้อนเข้ามาจะทำให้ไดโอดได้รับแรงดันไฟฟ้าไบอัสตรงจึงมีกระแสไฟฟ้า I_p ไหลผ่าน R_3 จะได้ $I_p = \frac{E_p}{R_3}$ และกระแสไฟฟ้าเฉลี่ย $I_{av} = 0.318I_p$



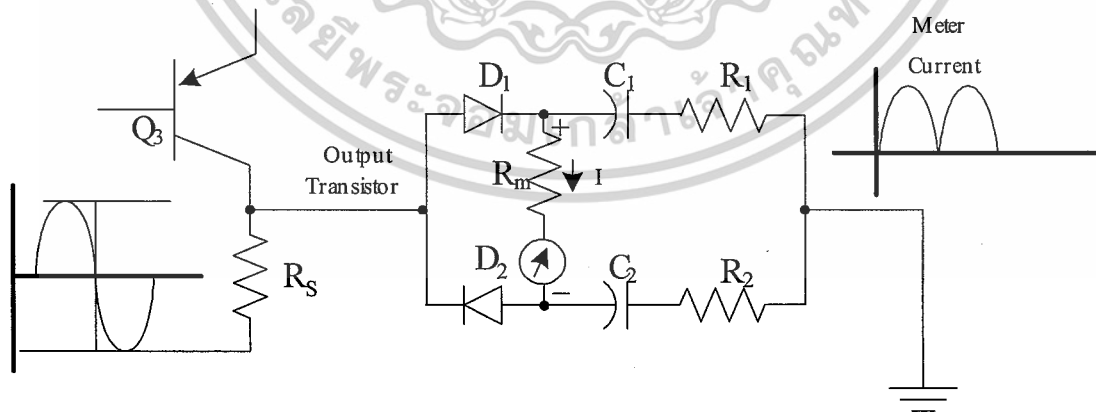
รูปที่ 2.10 โวลต์มิเตอร์กระแสสลับแบบเปลี่ยนแรงดันไฟฟ้าเป็นกระแสไฟฟ้า

วงจรรูปที่ 2.11 เป็น โวลต์มิเตอร์กระแสสลับเรียงกระแสไฟฟ้าเต็มคลื่นแบบบริดจ์ โดยที่ ไดโอด D_1, D_4 จะได้รับแรงดันไฟฟ้าไบอัสตรงขณะที่แรงดันไฟฟ้าอินพุตครึ่งไซเคิลบวกถูกป้อนเข้ามาและ D_2, D_3 จะได้รับแรงดันไฟฟ้าไบอัสตรงขณะที่แรงดันไฟฟ้าอินพุตครึ่งไซเคิลลบถูกป้อนเข้ามาจะได้กระแสไฟฟ้าเฉลี่ย $I_{av} = 0.636I_p$



รูปที่ 2.11 โวลต์มิเตอร์กระแสสลับเรียงกระแสไฟฟ้าเต็มคลื่นแบบบริดจ์

ส่วนรูปที่ 2.12 โวลต์มิเตอร์กระแสสลับที่เรียงกระแสไฟฟ้าเต็มคลื่นแบบครึ่งบริดจ์โดยจะใช้ไดโอด D_1, D_2 เพียง 2 ตัวและมีทรานซิสเตอร์ Q_3 จ่ายกระแสไฟฟ้าให้ R_s ได้แรงดันไฟฟ้าป้อนให้วงจรเรียงกระแสไฟฟ้าแบบครึ่งคลื่น นอกจากนี้ C_1, C_2 มีไว้เพื่อกันกระแสไฟฟ้ากระแสตรงจากมิเตอร์และให้กระแสไฟฟ้ากระแสสลับผ่านไป

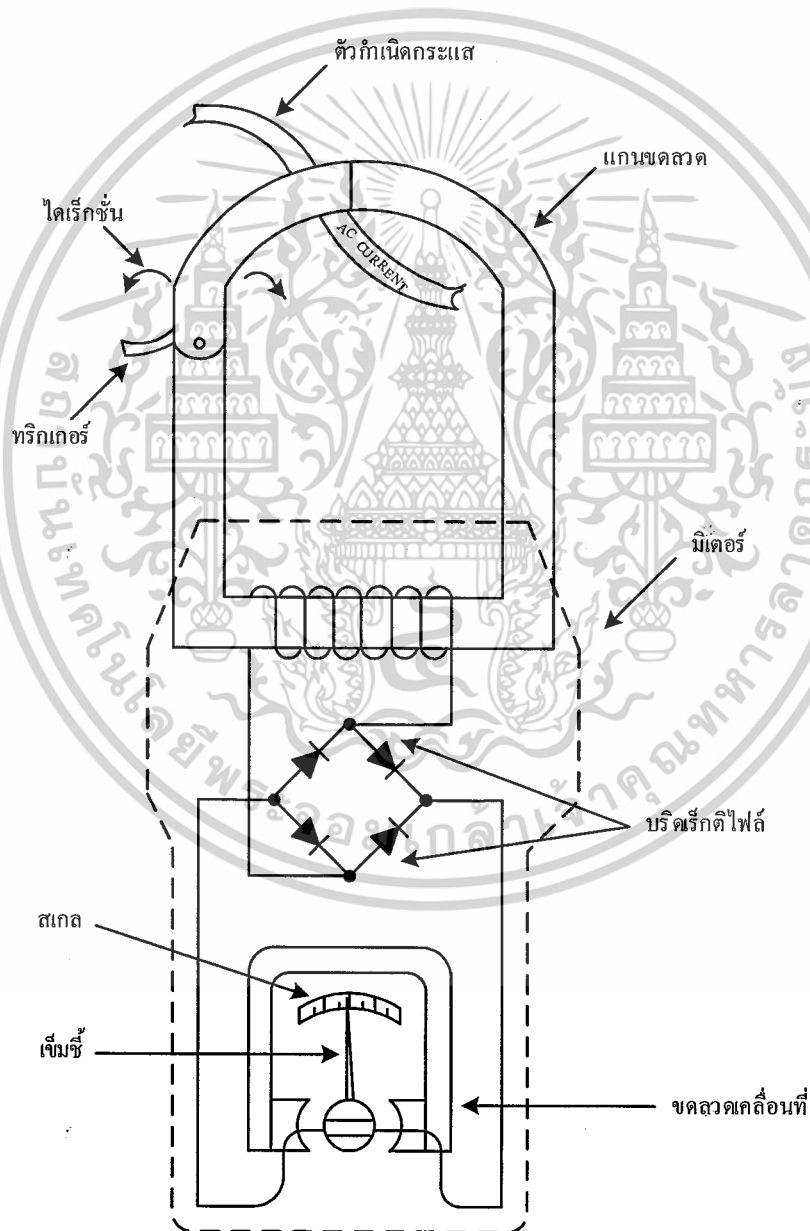


รูปที่ 2.12 โวลต์มิเตอร์กระแสสลับเรียงกระแสไฟฟ้าเต็มคลื่นแบบครึ่งบริดจ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 แอมมิเตอร์กระแสสลับแบบแคลมป์ออน

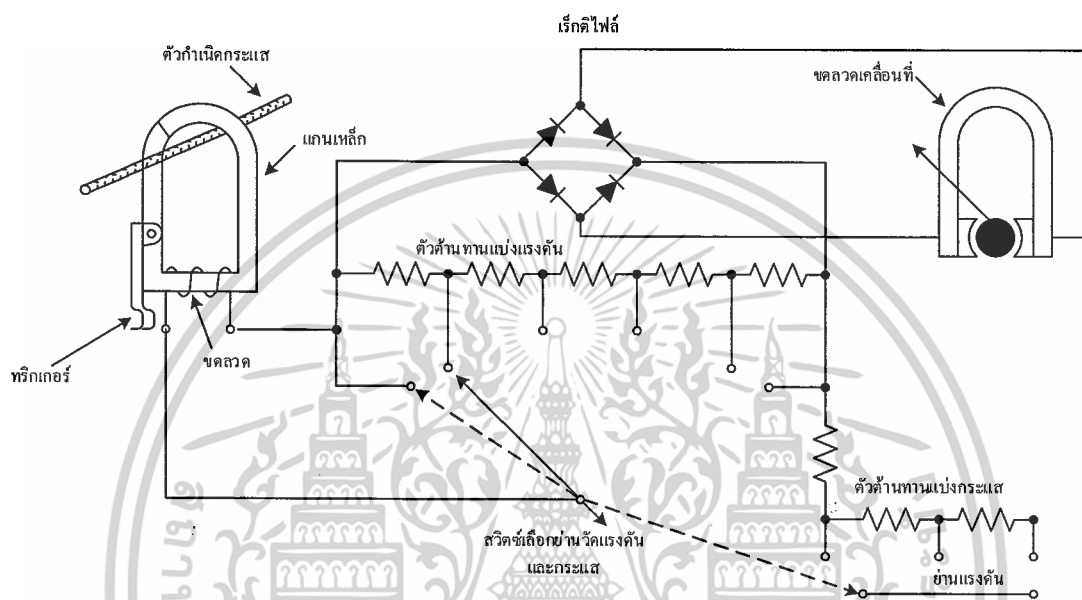
แอมมิเตอร์กระแสสลับแบบแคลมป์ออนดังรูปที่ 2.13 จะทำงานเหมือนกันกับหม้อแปลงกระแสไฟฟ้าโดยมีขดลวดปฐมภูมินำไปคล้องเข้ากับสายไฟฟ้า ทำให้เกิดการเหนี่ยวนำได้ กระแสไฟฟ้าไหลในขดลวดทุติยภูมิ โดยกระแสไฟฟ้านี้จะเป็นสัดส่วนกับกระแสไฟฟ้าที่ขดลวดปฐมภูมิ และนำไปป้อนในหัววงจรเรียงกระแสเพื่อเปลี่ยนไฟฟ้ากระแสสลับเป็นไฟฟ้ากระแสตรงป้อนให้ขดลวดเคลื่อนที่



รูปที่ 2.13 โครงสร้างของแอมมิเตอร์กระแสสลับแบบแคลมป์ออนมิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยปกติแล้วการวัดกระแสไฟฟ้าจะต้องต่อแอมมิเตอร์อนุกรมกับโหลด ถ้าวัดค่ากระแสไฟฟ้าสูงๆ แล้วจะมีอันตรายและไม่สะดวกในการปฏิบัติ จึงใช้แคลมป์ออนมิเตอร์วัดกระแสไฟฟ้า แทน แคลมป์ออนมิเตอร์จะมี 3 ส่วนคือ โอห์มมิเตอร์ แอมมิเตอร์และโวลต์มิเตอร์ จะมีสวิตช์เปลี่ยนย่านวัดเหมือนทั่วไป ดังรูปที่ 2.13 และ 2.14



รูปที่ 2.14 วงจรของแคลมป์ออนโวลต์ – แอมมิเตอร์

2.3.3 วงจรเปลี่ยนแรงดันไฟฟ้าเป็นกระแสไฟฟ้า

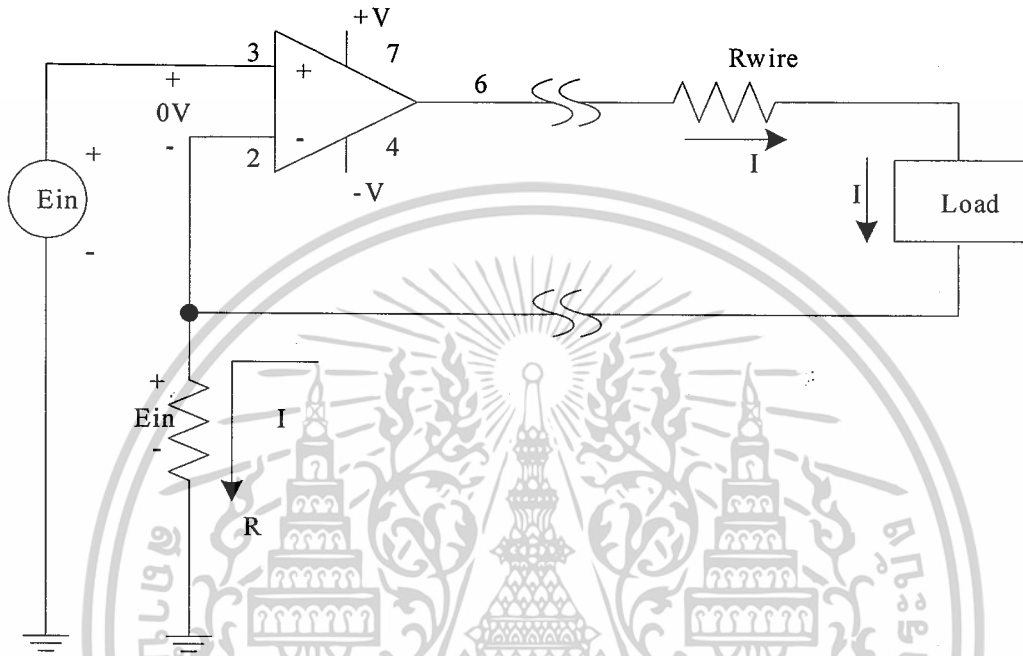
การส่งสัญญาณแรงดันไฟฟ้ามีปัญหาเรื่องอุณหภูมิ จุดต่อไม่ดี (Poor Connection) ความต้านทานสาย (Line Resistance) ที่เกิดระหว่างเอาต์พุตของวงจรกับ โหลด การสูญเสียภายในสายเพียง 2-3 mV นั้น จะทำให้ค่าความผิดพลาดจากการวัดสัญญาณ ดังนั้นจึงเปลี่ยนให้ส่งเป็นสัญญาณกระแสไฟฟ้าแทนเพราะไม่มีปัญหาการสูญเสียในสายและจุดต่อ สัญญาณรบกวนภายในสายนำสัญญาณลดน้อยลง ทำให้มีประสิทธิภาพของการทำงานดีขึ้น

2.3.3.1 วงจรเปลี่ยนแรงดันไฟฟ้าเป็นกระแสไฟฟ้าแบบโหลดต่อกราวด์ (Grounded Load Voltage to Current Conversion)

วงจรเปลี่ยนแรงดันไฟฟ้าเป็นกระแสไฟฟ้าแบบโหลดต่อกราวด์ การใช้งานขึ้นอยู่กับ การนำค่าที่วัดนั้นมาแสดงผลหรือการควบคุมด้วยวงจรอิเล็กทรอนิกส์ที่ต้องการ โดยจะมีหลักการ ทำงานของวงจรดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรขยายความแตกต่างจะใช้ในวงจรแบบโหลดต่อกราวด์เพื่อจ่ายกระแสไฟฟ้าให้โหลด ดังรูปที่ 2.15 ตัวต้านทาน R_1 , R_2 , R_3 และ R_4 จะมีค่าเท่ากัน จะทำให้อัตราขยายเท่ากับ 1 ตัวต้านทาน R_5 ต่อระหว่างเอาต์พุตกับโหลด ส่วน R_4 จะไม่ต่อลงกราวด์แต่จะต่อที่ด้านบนของโหลด



รูปที่ 2.15 วงจรเปลี่ยนแรงดันไฟฟ้าเป็นกระแสไฟฟ้าแบบ โหลดต่อกราวด์

สมการการออกแบบวงจร

หา V_{OUT}

$$V_{out} = V_L + e_2 - e_1 \tag{2.1}$$

หา V_{RS}

แรงดันไฟฟ้าเอาต์พุตของวงจรจะตกคร่อม R_5

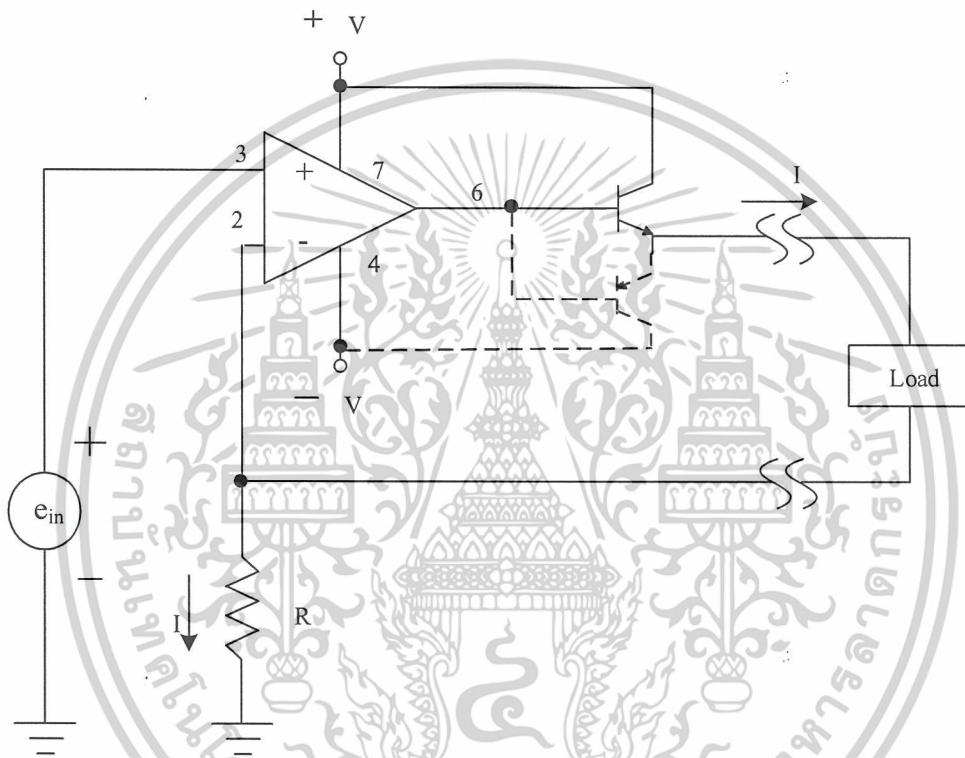
$$\begin{aligned} V_{RS} &= V_{out} - V_L = (V_L + e_2 - e_1) - V_L \\ V_{RS} &= e_2 - e_1 \end{aligned} \tag{2.2}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากกระแสไฟฟ้าไหลผ่าน R_S

$$I_{RS} = \frac{V_{RS}}{R_S}$$

$$I_L = I_{RS} = \frac{e_2 - e_1}{R_S} \quad (2.3)$$



รูปที่ 2.16 วงจรเปลี่ยนแรงดันไฟฟ้าเป็นกระแสไฟฟ้าแบบโหนดต่อกราวด์

จากรูปวงจรที่ 2.16 เอาต์พุตของออปแอมป์สามารถเพิ่มกระแสไฟฟ้าโดยการเพิ่มทรานซิสเตอร์ 1 หรือ 2 ตัวระหว่างเอาต์พุตของออปแอมป์กับช่วงต่อสัญญาณป้อนกลับแบบลบ วงจรขยายความแตกต่างต่อแรงดันไฟฟ้าอินพุต e_m ($e_m = e_2$) ที่ขานอนอินเวอร์ตติ้งและต่อซีโรหรือออฟเซต (Zero หรือ Offset) ที่ขานอนอินเวอร์ตติ้ง มี R_S ไว้ปรับสเกล ในทางปฏิบัติแล้ววงจรขยายความแตกต่างนี้จะนำวงจรขยายการวัด (Instrument Amplifier) มาทำงานแทน

2.3.4 อุปกรณ์ที่ใช้ในการตรวจวัดอุณหภูมิ

เครื่องมือที่ใช้วัดอุณหภูมิเรียกว่าเทอร์โมมิเตอร์ มีสารหลายชนิดที่สามารถนำมาประดิษฐ์เป็นเทอร์โมมิเตอร์ได้ เช่นปรอท แก๊ส อากาศ น้ำ แอลกอฮอล์ เป็นต้น จากตารางที่ 2.1 จะเห็นว่าเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทอร์โมมิเตอร์ที่ทำด้วยปรอท แก๊สไฮโดรเจน และอากาศนั้นต่างก็แบ่งระยะระหว่างจุดเยือกแข็งกับจุดเดือดของน้ำออกเป็น 100 ช่องเหมือนกัน แต่ถ้านำไปวัดที่ระดับอุณหภูมิต่างๆ จะอ่านแตกต่างกันเล็กน้อย ทั้งนี้ขึ้นอยู่กับ การขยายตัวซึ่งเป็นคุณสมบัติเฉพาะของแต่ละสาร แต่โดยทั่วไปพบว่า การขยายตัวหรือหดตัวของแก๊สต่างๆ อากาศที่เปลี่ยนไปนั้น ก่อนข้างคงที่มากกว่าของเหลวและของแข็ง ดังนั้นสามารถใช้เทอร์โมมิเตอร์แก๊สเป็นหลักในการเทียบมาตราส่วนอุณหภูมิ ตารางที่ 2.1 เป็นการเปรียบเทียบเทอร์โมมิเตอร์แก๊สกับเทอร์มิเตอร์ปรอท โดยให้เทอร์โมมิเตอร์ไฮโดรเจนมีคุณสมบัติใกล้เคียงกับแก๊สอุดมคติ

ตารางที่ 2.1 เปรียบเทียบคุณสมบัติเทอร์โมมิเตอร์แก๊สกับเทอร์โมมิเตอร์ปรอท

เทอร์โมมิเตอร์ไฮโดรเจน (ปริมาตรคงที่)	เทอร์โมมิเตอร์อากาศ (ปริมาตรคงที่)	เทอร์โมมิเตอร์ปรอท
0	0	0
20	20.008	20.091
40	40.001	40.111
60	59.990	60.086
80	79.987	80.041
100	100	100

โดยทั่วไป นิยมใช้ปรอทเป็นส่วนประกอบของเทอร์โมมิเตอร์เนื่องจากปรอทมีการขยายตัวและหดตัวค่อนข้างคงที่ทุกๆ องศาที่เปลี่ยนไป การขยายตัวและหดตัวน้อยมากเมื่อเทียบกับแก๊ส มีช่วงของการเป็นของเหลวกว้าง (ปรอทแข็งตัวที่ -38.87°C เดือดที่ 356.9°C) เทอร์โมมิเตอร์ทำด้วยปรอทจึงกะทัดรัด ไม่ต้องใช้หลอดแก้วยาว เทอร์โมมิเตอร์ที่นิยมใช้กันโดยทั่วไป คือ แบบเซลเซียสและฟาเรนไฮต์

จากหลักการในการตรวจวัดอุณหภูมิในหลายๆ วิธีดังกล่าว การตรวจวัดอุณหภูมิโดยอาศัยการเปลี่ยนแปลงคุณสมบัติทางไฟฟ้าถูกนิยมนำมาใช้กันมากที่สุด เพราะสัญญาณที่ได้จากอุปกรณ์เหล่านี้สามารถนำไปต่อรวมกับวงจรไฟฟ้าหรืออิเล็กทรอนิกส์ เพื่อการแสดงผลในเชิงตัวเลขหรือควบคุมระบบกระบวนการที่ต้องการ ดังนั้นในที่นี้จะขอกกล่าวเพียงกลุ่มของอุปกรณ์ที่อาศัยหลักการเปลี่ยนแปลงคุณสมบัติทางไฟฟ้าเป็นสำคัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) อาร์ทีดี (RTD)

เป็นคำย่อมาจาก Resistance Temperature Detector หรือความต้านทานที่ใช้ในการตรวจวัดอุณหภูมิ โดยอาศัยหลักการเปลี่ยนแปลงค่าความต้านทานไฟฟ้าโดยความต้านทานจะเป็นสัดส่วนโดยตรงกับอุณหภูมิ

อาร์ทีดี ทำด้วยโลหะที่มีความยาวค่าหนึ่ง ซึ่งจะทำให้เกิดค่าความต้านทานที่ต้องการที่อุณหภูมิ 0°C ลวดโลหะดังกล่าวนี้พันอยู่รอบแกนที่เป็นฉนวนไฟฟ้าและมีคุณสมบัติทนต่อความร้อน แกนที่ใช้ส่วนมากจะทำมาจากสารประเภทเซรามิก การพันขดลวดจะนิยมทำกันในขณะที่ขดลวดร้อนจนถึงอ่อนตัว หลังจากนั้นจะต้องผ่านกระบวนการอบความร้อนคลายความเครียดที่อยู่ในขดลวดด้วยอุณหภูมิต่ำอย่างน้อย 500°C เป็นเวลานานถึง 24 ชั่วโมง โดยทั่วไปจะถูกบรรจุอยู่ในฝักโลหะ ฉนวนที่ใช้จะเป็นแมกนีเซียมออกไซด์หรืออลูมิเนียมออกไซด์

อาร์ทีดีโดยทั่วไปแล้วทำมาจากโลหะที่มีความต้านทานต่ำ เช่น พลาตินัม ทองแดง นิกเกิล สำหรับทั้งสแตนเลสจะใช้เป็นอุปกรณ์ตรวจวัดอุณหภูมิที่ต้องการย่านวัดสูงๆ แต่เนื่องจากมีความเปราะและแตกหักง่ายจึงไม่ค่อยนิยมนำมาใช้ในงาน ส่วนโลหะที่นิยมใช้ทำอาร์ทีดีและให้ผลตอบแทนที่เป็นเส้นตรง คือ พลาตินัม

ตารางที่ 2.2 คุณสมบัติของอาร์ทีดีทำมาจากโลหะประเภทต่างๆ

ชนิดของโลหะ	กระเปาะที่บรรจุ (Case)	ย่านอุณหภูมิ (องศา)	ความต้านทาน (Ω)	ค่าสัมประสิทธิ์โดยประมาณ (α) $\Omega/\Omega/^{\circ}\text{C}$
พลาสติกหุ้ม ทดลอง	แก้ว	-190 ถึง 540	25 ที่ 0°C	0.0039
พลาตินัม (อุตสาหกรรม)	สแตนเลส เหล็ก	-200 ถึง 125	25 ที่ 0°C	0.0039
		-18 ถึง 540	25 ที่ 0°C	0.0039
พลาตินัม (แบบแผ่นฟิล์ม)	เซรามิก อะลูมินาและแก้ว	-50 ถึง 600	1000 ที่ 0°C	0.0039
		-272 ถึง 200	26 ที่ 0°C	0.0037

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 (ต่อ) คุณสมบัติของอาร์ทีดีที่ทำมาจากโลหะประเภทต่างๆ

ชนิดของโลหะ	กระเปาะที่ บรรจุ (Case)	ย่านอุณหภูมิ (องศา)	ความต้านทาน (Ω)	ค่าสัมประสิทธิ์ โดยประมาณ (α) $\Omega/\Omega/^{\circ}\text{C}$
โรเดียม	ทองเหลือง	-75 ถึง 120	10 ที่ 0 $^{\circ}\text{C}$	0.0038
เหล็ก	ทองเหลือง	0 ถึง 120	10 ที่ 0 $^{\circ}\text{C}$	0.0038
ทองแดง				

จำเป็นอย่างยิ่งที่ต้องพิจารณาสายที่ต่อระหว่างอาร์ทีดีกับวงจรบริดจมากเป็นพิเศษ หากสายที่ใช้ต่อ มีความยาวมาก จะส่งผลต่ออุณหภูมิและค่าความต้านทานที่เกิดขึ้นกับวงจร ทำให้ผลของการวัดเกิดความผิดพลาด จึงจำเป็นต้องมีการแก้ไขปัญหาที่เกิดขึ้นเพื่อชดเชยค่าที่ผิดพลาด ซึ่งสามารถทำได้หลายวิธีด้วยกัน การต่อวงจรอาร์ทีดีแบบ 3 สาย เป็นแบบมาตรฐานที่นิยมใช้กันมากที่สุด ทั้งนี้สายตัวนำทั้ง 3 ต้องมีความยาวและขนาดเท่ากัน รวมทั้งอยู่ในสภาพแวดล้อมที่มีอุณหภูมิเดียวกันตลอด

2) เทอร์มิสเตอร์

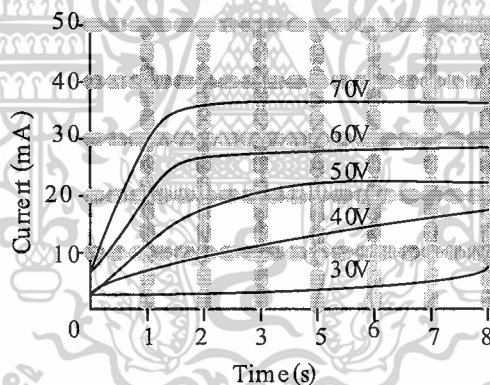
เป็นอุปกรณ์ตรวจวัดอุณหภูมิ ที่อาศัยการเปลี่ยนแปลงค่าความต้านทานเช่นเดียวกับอาร์ทีดี แต่เทอร์มิสเตอร์ทำจากคาร์บอนและสารกึ่งตัวนำ เช่น ออกไซด์ของโลหะ นิเกิล โคบอลต์ เหล็ก ทองแดง เยอมาเนียม แมกนีเซียม และไทเทเนียม ซึ่งส่วนใหญ่จะนิยมใช้ออกไซด์ของแมกกาไนต์กับทองแดงและออกไซด์ของนิเกิลกับทองแดง สัมประสิทธิ์การเปลี่ยนแปลงค่าความต้านทานของเทอร์มิสเตอร์จะมีค่าสูง ข้อสำคัญที่ทำให้ต่างจากอาร์ทีดี คือ การเปลี่ยนแปลงค่าความต้านทานจะกลับกันกับอาร์ทีดี กล่าวง่าย ๆ คือ ค่าความต้านทานจะลดลงเมื่ออุณหภูมิสูงขึ้น ซึ่งทั้งนี้เนื่องจากคุณสมบัติของสารกึ่งตัวนำ

เมื่อนำเทอร์มิสเตอร์มาใช้งานร่วมกับวงจรไฟฟ้า แรงดันตกคร่อมจะเพิ่มขึ้นตามกระแส แต่เมื่อถึงที่จุดหนึ่ง แรงดันไฟฟ้าตกคร่อมจะลดลงในขณะที่กระแสเพิ่มขึ้นเรื่อยๆ เนื่องจากคุณสมบัติของเทอร์มิสเตอร์มีค่าความต้านทานเป็นลบ ถ้าแรงดันไฟฟ้าที่ป้อนให้กับเทอร์มิสเตอร์มีค่าน้อย กระแสจะมีค่าน้อยตามไปด้วย ดังนั้นความร้อนที่เกิดขึ้นจากกระแสนี้ก็ยังไม่เพียงพอที่จะทำให้อุณหภูมิของเทอร์มิเตอร์มีค่าสูงตามอุณหภูมิของสภาพแวดล้อมในขณะนั้นได้ ภายใต้สภาวะเช่นนี้ กระแสจะเป็นสัดส่วนกับแรงดันไฟฟ้าตามกฎของโอห์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อแรงดันไฟฟ้าที่ป้อนมีค่าสูงขึ้น ทำให้กระแสสูงตามไปด้วย จะเกิดความร้อนทำให้อุณหภูมิเทอร์มิสเตอร์สูงเกินกว่าอุณหภูมิของสภาพแวดล้อมในขณะนั้นและค่าความต้านทานของเทอร์มิสเตอร์จะลดลง ซึ่งผลอันนี้จะทำให้กระแสไหลมากขึ้นและค่าความต้านทานยังลดลงไปเรื่อยๆ ค่ากระแสนี้จะค่อยๆ เพิ่มขึ้นเรื่อยๆ จนกระทั่งค่าสูญเสียของเทอร์มิสเตอร์ที่มีค่าเท่ากับพลังงานหรือกำลังที่ป้อนให้ภายใต้สิ่งแวดล้อมอุณหภูมิคงที่ ค่าความต้านทานของเทอร์มิสเตอร์จะเป็นฟังก์ชันของพลังงานที่สิ้นเปลืองภายในตัวของมันเอง จะต้องมีพลังงานที่เพียงพอที่จะทำให้อุณหภูมิของมันสูงกว่าอุณหภูมิของสภาพแวดล้อม

ในรูปที่ 2.17 แสดงให้เห็นถึงคุณลักษณะของกระแสกับเวลาของเทอร์มิสเตอร์ กระแสจะมีค่าสูงสุดเมื่อเวลาผ่านไปเพียงเล็กน้อย และขึ้นอยู่กับขนาดของแรงดันไฟฟ้าที่จ่ายให้ด้วย เมื่อเกิดความร้อนในตัวเทอร์มิสเตอร์ จะใช้เวลาช่วงหนึ่งที่จะทำให้อุณหภูมิสูงขึ้นและกระแสไฟฟ้าจะมีค่าคงที่ ซึ่งในช่วงเวลาดังกล่าวนี้ ถือได้ว่าเป็นเวลาที่ใช้ในการตอบสนองนั่นเอง ช่วงเวลานี้จะแปรผันไปตามแรงดันไฟฟ้าที่จ่ายให้



รูปที่ 2.17 คุณลักษณะของกระแสกับเวลาของเทอร์มิสเตอร์

เนื่องจากสัมประสิทธิ์การเปลี่ยนแปลงของเทอร์มิสเตอร์มีค่าสูง การใช้เทอร์มิสเตอร์ร่วมกับวงจรบริดจ์เหมือนอาร์ทีดีนั้น ทำให้สามารถอ่านค่าได้ละเอียดมากได้ในช่วงอุณหภูมิแคบๆ โดยสังเกตการณ์เปลี่ยนแปลงอุณหภูมิแม้เพียง 0.005°C เท่านั้น ดังนั้นวงจรการวัด โดยทั่วไปจึงหันมาใช้วงจรแบ่งแรงดันและสายที่ต่อจากเทอร์มิสเตอร์มายังวงจรก็ไม่มีส่วนสร้างความผิดพลาดเหมือนอาร์ทีดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ไอซี DS1820

ไอซี DS1820 เป็นไอซีระบบการสื่อสารข้อมูลอนุกรมแบบหนึ่งสายซึ่งถือได้ว่าเป็นระบบที่มีความชาญฉลาดและใช้จำนวนสายสัญญาณเพียง 1 เส้นเท่านั้น โดยไม่ต้องมีสายสัญญาณนาฬิกา มาควบคุมจังหวะการถ่ายทอดข้อมูลเหมือนกับระบบสื่อสารข้อมูลอนุกรมในแบบอื่นสายข้อมูลจะทำหน้าที่เสมือนเป็นสายสัญญาณนาฬิกาในตัว ส่วนค่าของข้อมูลจะพิจารณาจากลักษณะของรูปสัญญาณที่ปรากฏบนสายสัญญาณ ในแต่ละช่องของเวลาซึ่งเรียกว่า ไทม์สล็อต (Time Slot) โดยคาบเวลาดำสุดและสูงสุดของสถานะต่างๆ ในการสื่อสารข้อมูลในแต่ละไทม์สล็อตมีการกำหนดขอบเขตไว้อย่างชัดเจนในแต่ละไทม์สล็อตนั้น รูปแบบการถ่ายทอดข้อมูลจะเป็นอะซิงโครนัสในระดับบิต ไม่มีการสื่อสารข้อมูลระหว่างไอซีแผงวงจรเดี่ยว

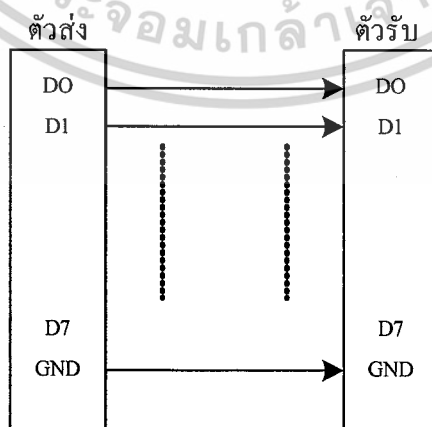
2.4 การรับส่งข้อมูลแบบอนุกรม

2.4.1 พื้นฐานการรับส่งข้อมูล

การรับส่งข้อมูลในระบบคอมพิวเตอร์ หมายถึง การรับส่งข้อมูลเป็นจำนวนไบต์ๆ ให้กับอุปกรณ์ที่เกี่ยวข้องกับคอมพิวเตอร์ ซึ่งแบ่งประเภทของการรับส่งข้อมูลได้ 2 แบบ

- 1) การรับส่งข้อมูลแบบขนาน (Parallel)
- 2) การรับส่งข้อมูลแบบอนุกรม (Serial)

การรับส่งข้อมูลแบบขนานเป็นการรับส่งข้อมูล จำนวน 1 ไบต์ ออกไปทางพอร์ต ในเวลาเดียวกันในระบบคอมพิวเตอร์ 1 ไบต์จะมีจำนวน 8 บิต คือ D0 – D7 ถ้ามีการส่งข้อมูลแบบขนานจะใช้สายสัญญาณอย่างน้อย 9 เส้น คือ สาย Data 8 เส้น และสายกราวด์ 1 เส้นดังรูปที่ 2.18



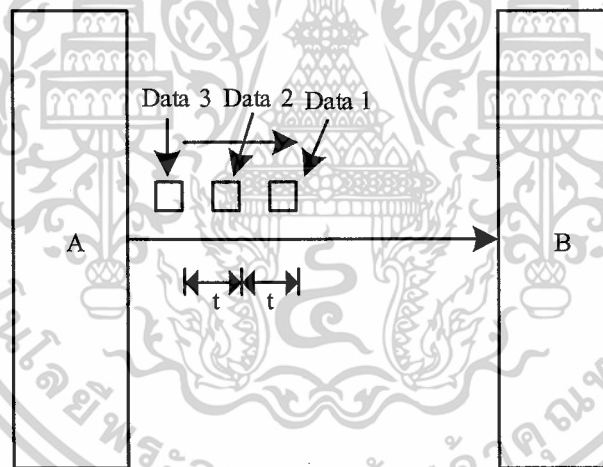
รูปที่ 2.18 การรับส่งข้อมูลแบบขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรับส่งข้อมูลแบบอนุกรม คือ การรับส่งข้อมูลที่ละบิต จนครบ 1 ไบต์ ถ้าต้องการส่งข้อมูล 1 ไบต์ คือ D0 – D7 อาจจะส่งบิต D0 ออกไปก่อนแล้วตามด้วย D1 ไปเรื่อยๆ จนถึง D7 การส่งข้อมูลทั้งสองแบบมีข้อดีข้อเสียแตกต่างกันคือ การส่งข้อมูลแบบขนานสามารถส่งข้อมูลได้เร็ว คือส่งเป็นระยะๆ จะสิ้นเปลืองสายสัญญาณ ถ้าเป็นการส่งแบบอนุกรม เมื่อต้องการส่งข้อมูลเป็นระยะๆ จะช่วยประหยัดสายสัญญาณเนื่องจากจะใช้สายอย่างน้อยเพียง 2 เส้น คือ สายสัญญาณกับสายกราวด์ แต่การรับส่งข้อมูลจะใช้เวลาเนื่องจากเป็นการส่งทีละบิต ในบทนี้จะกล่าวถึงพื้นฐานการรับส่งข้อมูลแบบอนุกรม โดยจะเน้นที่ตัว MSC – 51 เป็นสำคัญ

1) การรับส่งข้อมูลแบบซิงโครนัส (Synchronous Input / Output)

การรับส่งข้อมูลแบบนี้ไม่ว่าจะเป็นการส่งแบบอนุกรมหรือขนาน ข้อมูลแต่ละไบต์ในการรับส่งจะมีช่วงห่างกันแน่นอน เช่น การส่งข้อมูลจาก A ไป B ดังรูปที่ 2.19 Data 1 จะห่างจาก Data 2 เป็นเวลาและ Data 3 จะห่างจาก Data 2 เป็นเวลา t เช่นกัน ระบบนี้เหมาะสมกับงานที่ไม่มีความยุ่งยากมาก



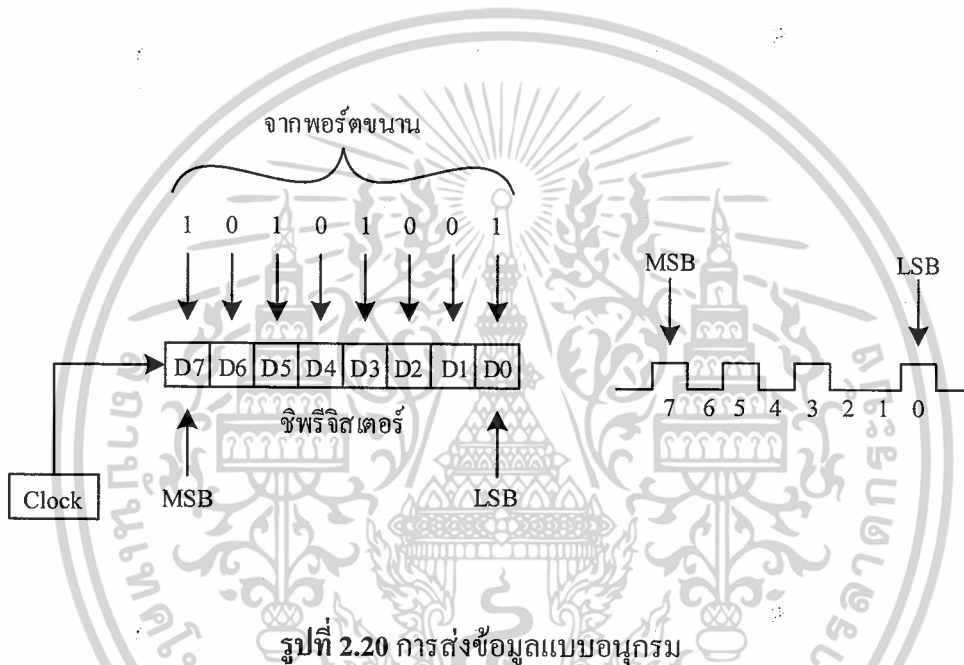
รูปที่ 2.19 การส่งข้อมูลแบบซิงโครนัส

2) การรับส่งข้อมูลแบบอะซิงโครนัส (Asynchronous Input / Output)

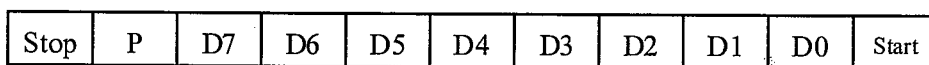
การรับส่งข้อมูลแบบนี้ ข้อมูลในการรับส่งจะไม่มีเวลาที่แน่นอน ซึ่งจะขึ้นอยู่กับความพร้อมของผู้ส่งและผู้รับ โดยจะมีสายสัญญาณตรวจสอบความพร้อมของระบบทั้งสองว่าพร้อมที่จะติดต่อกันหรือไม่ โดยสัญญาณที่เพิ่มขึ้นมาจากระบบแบบซิงโครนัส เรียกว่า สายสเตตัส (Status Line)

2.4.2 รูปแบบของการรับส่งข้อมูลแบบอนุกรม

เมื่อไมโครคอมพิวเตอร์ต้องการจะรับส่งข้อมูลแบบอนุกรม ตัวไมโครคอมพิวเตอร์จะส่งข้อมูลออกไปทางพอร์ทซึ่งเป็นพอร์ทแบบขนานก่อน จากนั้นจะมีอุปกรณ์มาต่อที่พอร์ท เพื่อแปลงข้อมูลแบบขนานให้เป็นแบบอนุกรมอีกทีหนึ่ง (Parallel – to – Serial Conversion) ตัวแปลงข้อมูลนี้อาจจะพิจารณาได้ง่ายๆ ว่าเป็นชิพริจิสเตอร์ ดังรูปที่ 2.20 เมื่อข้อมูลที่ส่งอยู่ในชิพริจิสเตอร์ แล้วตัวสัญญาณนาฬิกาจะเป็นตัวกระตุ้นให้ข้อมูลบิตต่ำออกไปในเวลาแรก จากนั้นจะส่งบิตต่อไปตามออกมา จากรูปที่ 2.20 จะเป็นการส่งข้อมูล A9H ออกไป



สำหรับตัวรับข้อมูลแบบอนุกรมนั้นเมื่อตัวรับข้อมูลทำงานจะเป็นการรับเข้ามาใน Shift Register แล้วส่งข้อมูลให้ไมโครคอมพิวเตอร์แบบขนาน (Serial – to – Parallel) ระบบคอมพิวเตอร์ในปัจจุบันจะมีตัวแปลง Parallel – to – Serial และ Serial – to – Parallel อยู่ในชิพไอซี เรียกว่า Universal Asynchronous Receiver Transmitter (UART) การส่งข้อมูลแบบอนุกรมต้องมีการเพิ่มเติมข้อมูลบางอย่างเข้าไปเพื่อให้การรับส่งข้อมูลสามารถทำงานได้ถูกต้องโดยมีการเติมค่าบิตต่าง ๆ ลงไปตามรูปที่ 2.20



รูปที่ 2.21 บิตต่างๆ ของข้อมูลที่ส่งแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งข้อมูลแบบ 8 บิต จะต้องส่งบิตแรกออกไปก่อน เรียกว่า บิตต้น (Start Bit) ถ้ามีการส่งข้อมูลหลายๆ ไบต์ออกมา บิตจะเป็นตัวบอกว่ามีข้อมูลใหม่มา โดยทั่วไปบิตเริ่มต้นมักมีระดับลอจิกเป็น “0” ถัดไปจะเป็นข้อมูลบิต D0 ถึง D7 จากนั้นจะตามด้วยบิตตรวจสอบความถูกต้อง (Parity Bit) ถ้าข้อมูล 8 บิตที่ส่งออกมา จำนวนของบิตที่มีค่าเป็น “1” เป็นจำนวนคู่ บิตนี้จะมีค่าเป็น 0 แต่ถ้าจำนวนของบิตที่มีค่าเป็น “1” เป็นจำนวนคี่ บิตนี้จะมีค่าเป็น 1 จากนั้นข้อมูลที่ส่งออกไปจะตามด้วยบิตสิ้นสุด (Stop Bit) เพื่อเป็นการบอกว่าข้อมูลที่ส่งมา 8 บิตนั้นหมดแล้ว ตัวบิตสิ้นสุดอาจมีจำนวนมากกว่า 1 บิตก็ได้ เช่น $1\frac{1}{2}$ บิต, 2 บิต

บิตตรวจสอบความถูกต้องหรือบิตพาริตีจะมีสองลักษณะคือพาริตีคู่ (Even Parity) และพาริตีคี่ (Odd Parity) ซึ่งสามารถเลือกได้ ถ้าหากระบุเป็นพาริตีคู่หมายความว่าข้อมูลที่ส่งไปหรือไบต์นั้นมีจำนวนลอจิก “1” รวมกับบิตพาริตีเป็นจำนวนคู่บิต ถ้าหากระบุเป็นพาริตีคี่หมายความว่าข้อมูลที่ส่งไปหรือไบต์นั้นมีจำนวนลอจิก “1” ของไบต์ข้อมูลที่ส่งไปรวมกับค่าบิตพาริตีเป็นจำนวนคี่บิต ตัวอย่างเช่นถ้าเราส่งไบต์ข้อมูลที่มีค่าเป็น B2H หรือ 10110010B ออกไป ถ้าระบบระบุว่าเป็นพาริตีคี่แล้วค่าของบิตพาริตีจะต้องเป็นลอจิก “1” ซึ่งในไบต์ข้อมูลจะมีลอจิก “1” จำนวน 4 บิต และบิตพาริตีเป็น “1” ดังนั้นมีลอจิก “1” ทั้งหมด 5 บิต ซึ่งเป็นคี่บิตแต่ถ้าระบบระบุว่าเป็นระบบพาริตีคู่แล้วค่าพาริตีจะต้องเป็นลอจิก “0” เมื่อทางภาครับได้รับข้อมูลเข้ามาก็ตรวจสอบว่าข้อมูลทั้งหมดมีลอจิก “1” เป็นคู่หรือคี่บิต ตรงกับการทำงานของระบบซึ่งออกแบบไว้หรือไม่ ซึ่งจะเป็นการตรวจสอบความถูกต้องได้ระดับหนึ่ง

การสื่อสารข้อมูลแบบอนุกรม การกำหนดอัตราเร็วในการรับส่งข้อมูลจะบอกเป็นบิตต่อวินาที (Bit Per Second: BPS) ที่เรียกว่าอัตราบอดหรือบอดเรท (Baud Rate) โดยค่ามาตรฐานที่ใช้กันมีหลายค่าได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600 และ 19200 บิตต่อวินาที

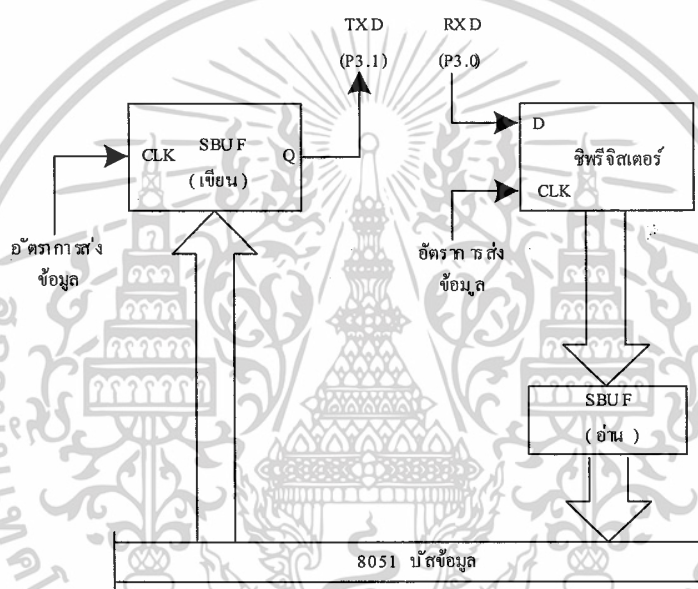
ใน MCS – 51 นั้นจะมีพอร์ตสำหรับรับส่งข้อมูลแบบอนุกรมอยู่ในชิพ โดยไม่ต้องต่ออุปกรณ์ภายนอกเพิ่ม และเป็นพอร์ตสื่อสารอนุกรมแบบฟูลดูเพล็กซ์ (Full – Duplex) คือสามารถรับส่งข้อมูลได้สองทิศทางในเวลาเดียว การควบคุมการใช้งานพอร์ตอนุกรมของ MCS – 51 นี้สามารถโปรแกรมให้ทำงานรูปแบบต่างๆ ได้ 4 ประเภทหรือ 4 โหมดการทำงาน โดยโปรแกรมผ่านทางรีจิสเตอร์ SCON โดยโหมด 0 จะเป็นการสื่อสารแบบซิงโครนัสหรือแบบเข้าจังหวะเวลา ส่วนโหมด 1, 2 และ 3 เป็นการสื่อสารแบบอะซิงโครนัส

2.4.3 การรับส่งข้อมูลแบบอนุกรมใน MCS – 51

พอร์ตอนุกรมของ MCS – 51 จะใช้ขา TXD และ RXD ในการรับส่งข้อมูล โดยขาทั้งสองจะอยู่ในพอร์ต 3 คือ P3.1 หรือขา 11 TXD และ P3.0 หรือขา 10 เป็น RXD พอร์ตอนุกรมของ

MCS-51 สามารถทำงานแบบ Full Duplex ได้ คือสามารถส่งและรับข้อมูลในเวลาเดียวกันได้ โดยในการรับและส่งข้อมูลจะมีบัฟเฟอร์สำหรับเก็บข้อมูลให้ใช้

รีจิสเตอร์ที่สำคัญในการรับส่งข้อมูล คือ SBUF และ SCON ซึ่งเป็นรีจิสเตอร์ที่อยู่ใน Special Function Registers (SFR) โดยรีจิสเตอร์ Serial Port Buffer (SBUF) จะอยู่ในตำแหน่ง 99H ถ้าเขียนข้อมูลไปที่ตำแหน่งนี้จะเป็นการส่งข้อมูลออกทางพอร์ตอนุกรม และถ้าอ่านข้อมูลจากตำแหน่งนี้จะเป็นการรับข้อมูลจากพอร์ตอนุกรม โดยใน SBUF จะประกอบด้วยบัฟเฟอร์ 2 ตัว สำหรับส่งและรับข้อมูล ดังรูปที่ 2.22



รูปที่ 2.22 การรับส่งข้อมูลระหว่างรีจิสเตอร์ภายใน

สำหรับ Serial Port Register (SCON) ซึ่งอยู่ในตำแหน่ง 98H จะเป็นรีจิสเตอร์ที่สามารถเข้าถึงข้อมูลระดับได้รีจิสเตอร์นี้จะทำหน้าที่ควบคุมและบอกสถานะต่างๆ ของการรับส่งข้อมูลแบบอนุกรม

สำหรับความเร็วของการส่งข้อมูล (Baud Rate) สามารถหาได้จากการหารสัญญาณนาฬิกาที่ใช้กับ MCS-51

2.4.4 รีจิสเตอร์ควบคุมพอร์ตอนุกรม

MCS-51 มีโหมดการทำงานของพอร์ตอนุกรมหลายโหมด ซึ่งสามารถโปรแกรมการทำงานได้โดยเขียนข้อมูลไปยังรีจิสเตอร์ SCON ความหมายของแต่ละบิตแสดงดังตารางที่ 2.3 และ

2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 บิตต่างๆ ของรีจิสเตอร์ SCON

บิต	ชื่อ	ตำแหน่ง	ความหมาย
SCON.7	SM0	9FH	บิตเลือกโหมดการทำงานบิต 0
SCON.6	SM1	9EH	บิตเลือกโหมดการทำงานบิต 1
SCON.5	SM2	9DH	บิตเลือกโหมดการทำงานบิต 2
SCON.4	REN	9CH	บิตเฟลกกำหนดยอมให้มีการรับข้อมูล
SCON.3	TB8	9BH	ค่าของบิต 9 สำหรับการส่งข้อมูลในโหมด 2 และ 3 สามารถ Set และ Clear ได้โดยโปรแกรม
SCON.2	RB8	9AH	ค่าของบิต 9 เมื่อรับข้อมูลเข้ามา
SCON.1	TI	99H	บิตเฟลกแสดงการอินเทอร์รัพท์ภายหลังการส่งข้อมูลออกโดยจะ Set เมื่อส่งข้อมูลออกไปหมดแล้วและสามารถ Clear ได้ด้วยโปรแกรม
SCON.0	RI	98H	เฟลกแสดงการอินเทอร์รัพท์ภายหลังรับข้อมูลเข้ามาสามารถ Clear ได้ด้วยโปรแกรม

ตารางที่ 2.4 โหมดต่างๆ ของการรับส่งแบบอนุกรม

SM0	SM1	MODE	ความหมาย	Baud Rate
0	0	0	Shift Register	เปลี่ยนไม่ได้ (Oscillator Frequency $\div 12$)
0	1	1	8-bit UART	สามารถเปลี่ยนแปลงได้โดยการกำหนดจาก Timer
1	0	2	9-bit UART	เปลี่ยนไม่ได้ (Oscillator Frequency $\div 32$ หรือ $\div 64$)
1	1	3	9-bit UART	สามารถเปลี่ยนแปลงได้โดยการกำหนดจาก Timer

ก่อนที่จะพอร์ทอนุกรมจะต้อง โปรแกรมให้กับ SCON เสียก่อนเพื่อกำหนดโหมดการทำงานและลักษณะต่างๆ เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOV SCON, #01010010B

เป็นการกำหนดให้พอร์ตอนุกรมทำงานในโหมด 1 และอินาเบิตให้มีการรับข้อมูลพร้อมกับการกำหนดให้ TI เป็น 1

ในการส่งข้อมูลทุกโหมดสามารถทำได้โดยเขียนข้อมูลไปยัง SBUF เมื่อข้อมูลถูกส่งไปแล้วบิต TI จะถูกเซตเป็น “1” ในการส่งข้อมูล จะต้องคอยตรวจสอบบิต TI เพราะว่าถ้า TI ยังไม่เป็น “1” แสดงว่าข้อมูลยังส่งไปไม่หมด ถ้าหากมีการเขียนข้อมูลไปติดต่อไปยัง SBUF จะทำให้เกิดข้อผิดพลาดขึ้น สำหรับในการรับส่งข้อมูลบิต REN จะต้องเซตให้เป็น “1” ยกเว้น โหมด 0 เพื่ออนุญาตให้รับข้อมูลได้ เมื่อข้อมูลรับเข้ามาเรียบร้อยแล้ว บิต RI จะถูกเซตเป็น “1”

2.4.5 โหมดการรับ - ส่งข้อมูล

การรับส่งข้อมูลกับไมโครคอนโทรลเลอร์ MCS-51 นั้น ภายในชิพ MCS-51 จะมี UART อยู่ในตัว ซึ่งเป็นข้อดีของไมโครคอนโทรลเลอร์ ถ้าเป็นไมโครโปรเซสเซอร์ เช่น เบอร์ Z-80 ถ้าต้องการรับส่งทางพอร์ตอนุกรม โดยโหมดการรับส่งข้อมูลทางพอร์ตอนุกรมจะประกอบด้วย 4 โหมดการทำงานดังนี้

- 1) 8 - Bit Shift Register (Mode 0)
- 2) 8 - Bit UART with Variable Baud Rate (Mode 1)
- 3) 9 - Bit UART with Fixed Baud Rate (Mode 2)
- 4) 9 -Bit UART with Variable Baud Rate (Mode 3)

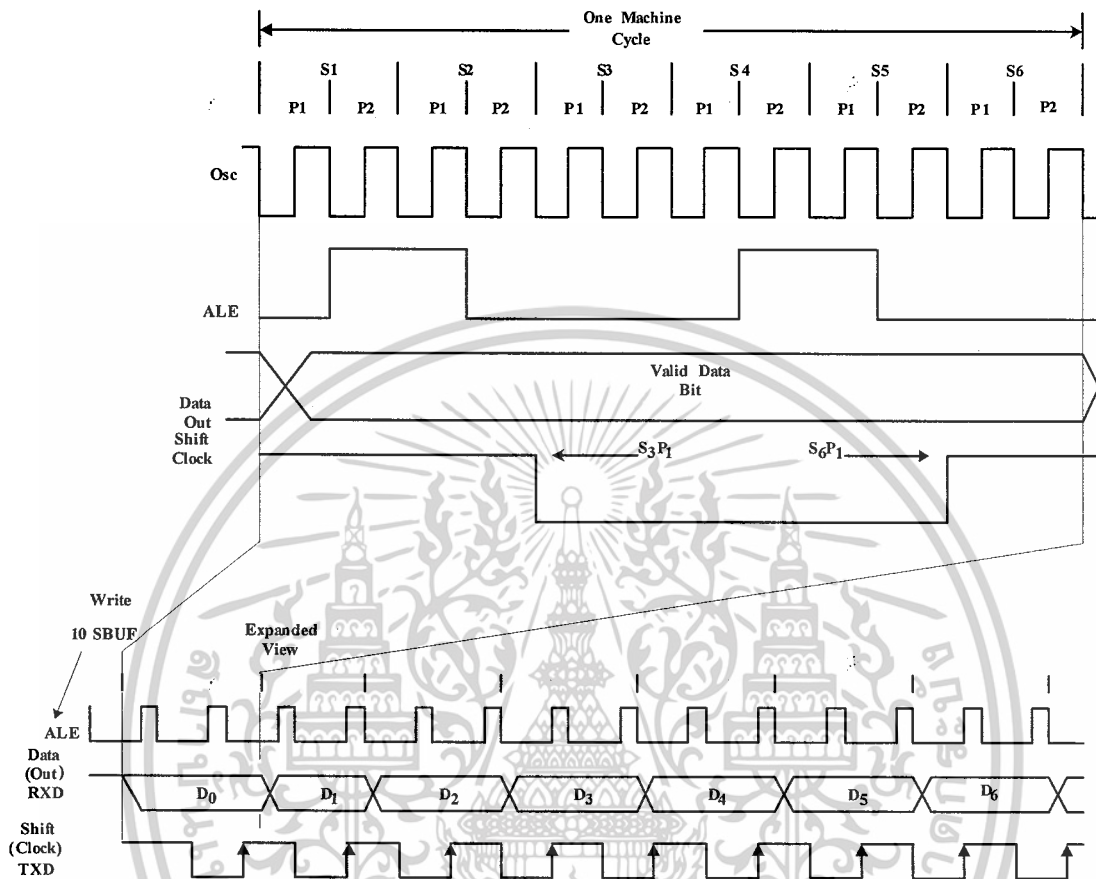
ใน MCS - 51 การสื่อสารทางพอร์ตอนุกรมจะมีอยู่ 4 ประเภท หรือ 4 โหมด ซึ่งกำหนดได้ที่บิต SM0 และ SM1 ใน SCON โดยจะมี 3 โหมด เป็นการสื่อสารแบบทางเดียวในลักษณะของข้อมูลที่ส่งจะมีบิตเริ่มต้น (Start Bit) และบิตจบ (Stop Bit) คล้ายกับการสื่อสารแบบ RS - 232 ในระบบคอมพิวเตอร์ อีกโหมดหนึ่งจะเป็นการใช้พอร์ตอนุกรมในลักษณะซีพรีจิสเตอร์

1) 8 - Bit Shift Register (Mode 0)

การทำงานในโหมดนี้จะใช้ขา RXD ในการรับส่งข้อมูล โดยต่อกับซีพรีจิสเตอร์ภายนอก ส่วนขา TXD จะเป็นสัญญาณนาฬิกาทางด้านเอาต์พุต เพื่อกระตุ้นรีจิสเตอร์ภายนอกให้เลื่อนบิตถ้ามีการส่งข้อมูลหรือรับข้อมูล 8 บิต จะเริ่มที่บิตต่ำสุดก่อน โดยมีค่าอัตราบอดเท่ากับ 1/2 ของความถี่ที่ใช้บนชิพ

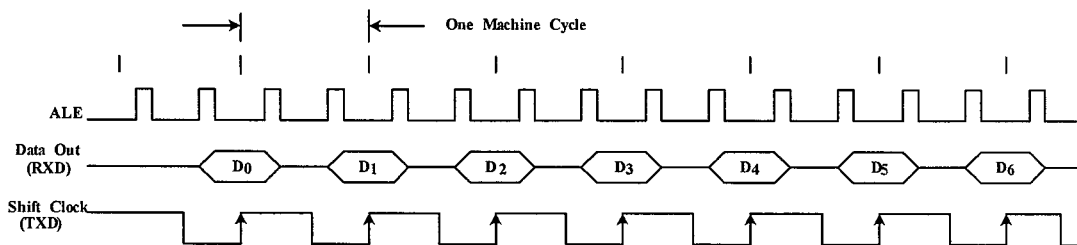
ในการส่งข้อมูลจะทำโดยเขียนข้อมูลไปที่รีจิสเตอร์ SBUF ข้อมูลจะถูกส่งออกมาทางขา RXD (P3.0) โดยจะสอดคล้องกับสัญญาณที่ออกมาทางขา TXD ซึ่งสัญญาณทางขา TXD จะถูกส่ง

ออกมาทุกๆ รูปคลื่น โดยจะเป็นลอจิก “0” ใน S3P1 และจะกลับเป็นลอจิก “1” ใน S6P1 ซึ่งแสดงได้ดังรูปที่ 2.23



รูปที่ 2.23 แผนผังเวลาการส่งข้อมูล

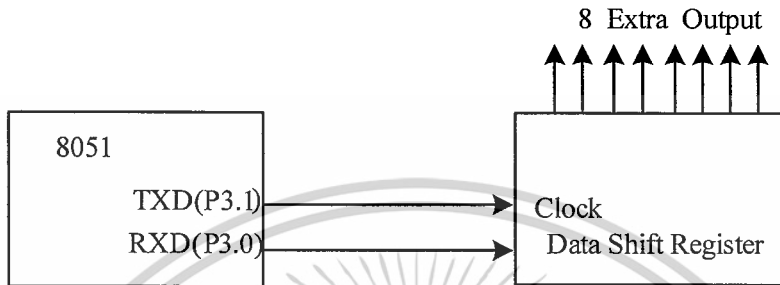
สำหรับการรับข้อมูลจะรับได้เมื่อเซตขา Receiver Enable Bit (REN) เป็น “1” และเคลียร์ขา Receiver Interrupt Bit (RI) เป็น “0” ข้อมูลจะเข้าสู่ MCS – 51 เมื่อสัญญาณนาฬิกาถูกส่งออกไปทาง TXD ที่ขอขาขึ้นของสัญญาณนาฬิกาบิตต่ำจะถูกส่งเข้ามาก่อนดังรูปที่ 2.23



รูปที่ 2.24 แผนผังเวลาการรับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

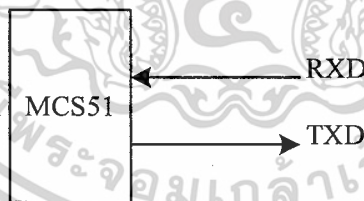
ในการประยุกต์ใช้งาน โหมดนี้จะต้องมีไอซีชิฟต์รีจิสเตอร์มาต่อภายนอก เช่น ถ้าหากต้องการส่งข้อมูลออกมาทางพอร์ตอนุกรม อาจต่อวงจรได้ดังรูปที่ 2.24 โดยใช้ไอซี Serial – to Parallel Shift Register โดยข้อมูลส่งออกมาทาง RXD และใช้ TXD เป็น Clock



รูปที่ 2.25 การรับส่งข้อมูลออกโดยใช้ชิฟต์รีจิสเตอร์ช่วย

2) 8 - Bit UART with Variable Baud Rate (Mode 1)

ในโหมดนี้จะเป็นการรับส่งข้อมูลแบบ 10 บิตซึ่งประกอบด้วยบิตเริ่มต้น (เป็น “0”) ข้อมูล 8 บิตและบิตจบ (เป็น “1”) นอกจากนี้ยังสามารถกำหนดค่าอัตราบอดได้โดยค่าอัตราบอดนี้จะแปรตามตัวจับเวลาที่ 1 ในโหมดนี้ จะส่งข้อมูลออกทาง TXD และรับข้อมูลเข้าทาง RXD ถ้าเป็นการรับข้อมูลเข้าตัว Stop Bit จะเข้ามายังบิต RB8 ใน SCON



รูปที่ 2.26 การรับส่งข้อมูลในโหมด 1

ค่าอัตราบอดที่ใช้ในการรับส่งข้อมูลจะกำหนดโดย Timer 1 หลังจากโปรแกรมไปใน Timer 1 แล้วสามารถเลือกค่าอัตราบอดได้อีกสองค่าคือ ค่าจาก Timer 1 Overflowหาร 32 กับค่าจาก Timer 1 Overflowหาร 16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรับสร้างข้อมูลทำได้โดยการเขียนข้อมูล 8 บิตไปที่ SBUF โดยที่ 9 (Stop Bit) ให้เขียนลงใน TB8 ใน SCON จากนั้นข้อมูลจะถูกส่งออกมาทางขา TXD โดยส่งบิตเริ่มต้นออกมาก่อน ตามด้วยข้อมูล 8 บิตและจบด้วยบิตสิ้นสุดเมื่อข้อมูลถูกส่งออกไปหมดแล้วบิต Interrupt Flag (TI) จะเป็น “1” ดังนั้นในการเขียนข้อมูลใหม่ลงไปจะต้องตรวจสอบบิตนี้

ในการรับข้อมูล จะเริ่มเมื่อมีการเปลี่ยนแปลงลอจิกจาก 1 เป็น 0 ทางขา RXD หมายความว่าเริ่มรับบิตเริ่มต้น จากนั้นข้อมูลอีก 8 บิตจะถูกเก็บลงใน SBUF และบิตสิ้นสุดจะถูกเก็บในบิต RB8 ของรีจิสเตอร์ SCON เมื่อข้อมูลเข้ามาครบแล้ว บิต Interrupt Flag (RI) จะถูกเซตดังนั้นในการอ่านข้อมูลจะอ่านได้เมื่อบิต RI ถูกเซตแล้ว เมื่ออ่านข้อมูลไปแล้วจะต้องเคลียร์บิตนี้

3) 9 - Bit UART with Fixed Baud Rate (Mode 2)

การทำงานในโหมดนี้ไม่สามารถกำหนดค่าอัตราบอดได้ซึ่งค่าอัตราบอดจะมีสองค่าคือ 1/64 และ 1/32 ของสัญญาณนาฬิกาบนชิพ การรับส่งข้อมูลจะเป็นชุดข้อมูล 9 บิต บิตเริ่มต้น บิตหยุด รวมเป็น 11 บิต โดยข้อมูล 9 บิตจะเป็นจำนวนข้อมูล 8 บิตและบิตที่โปรแกรมอีก 1 บิต โดยบิตนี้จะเป็นบิตที่ 9 ซึ่งจะเป็นพาริตีบิตในการส่งข้อมูลจะต้องเขียนไปที่ บิต TB8 ในรีจิสเตอร์ SCON สำหรับการรับส่งข้อมูลบิตที่ 9 จะถูกเก็บในบิต RB8

4) 9 -Bit UART with Variable Baud Rate (Mode 3)

การทำงานใน โหมดนี้คล้ายกับ โหมด 2 แต่สามารถกำหนดค่าอัตราบอดได้โดยการโปรแกรมไปที่ Timer 1 หลังจากโปรแกรมแล้วยังสามารถเลือกได้อีก 2 ค่าคือ ความถี่ค่าเกินของ Timer 1 ทหารด้วย 16 และหารด้วย 32

2.4.6 การกำหนดค่าเริ่มต้นให้รีจิสเตอร์ในการรับส่งข้อมูล

การรับข้อมูล ถ้าจะให้ MCS - 51 รับข้อมูลทางพอร์ทอนุกรมจะต้องโปรแกรมไปที่บิต Receiver Enable (REN) ในรีจิสเตอร์ SCON ให้เป็นลอจิก “1” ซึ่งอาจทำได้สองวิธีดังนี้

$$\text{REN} = 1$$

เป็นการเซตบิต REN ให้เป็น “1” หรืออาจทำโดยใช้คำสั่ง

$$\text{SCON} = \text{xxx1xxxxB}$$

ซึ่งเป็นการย้ายข้อมูลที่ทำให้บิต REN เป็น 1 สำหรับค่า x หมายความว่า เป็นอะไรก็ได้ขึ้นกับการใช้งานในโหมดต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลแบบ 9 บิต ในการรับส่งข้อมูลที่มีบิตข้อมูลแบบ 9 บิต ได้แก่ การใช้งานในโหมด 2 และโหมด 3 การส่งข้อมูลบิตที่ 9 จะถูกเขียนในบิต TB8 โดยการเขียนโปรแกรม สำหรับการรับข้อมูลเข้ามาถึงบิตที่ 9 จะถูกเขียนลงในบิต RB8

การเพิ่มบิต Parity การส่งข้อมูลแบบ 9 บิต สามารถใช้บิตที่ 9 เป็นบิตพาริตีได้ ซึ่งบิตพาริตีจะอยู่ใน Program Status Word (PSW) โดยจะถูกเซตหรือเคลียร์ทุกๆ แมชชีนไซเคิลที่เกี่ยวข้องกับ Accumulator เช่น ถ้าจะส่งข้อมูลแบบ 8 บิต ตามด้วยบิตพาริตีคู่เป็นบิตที่ 9 สามารถเขียนโปรแกรมได้ดังนี้

```
MOV C, P           ; อ่านค่าบิต P มาเก็บใน C
MOV TB8, C        ; นำค่าบิตพาริตีเขียนลงใน TB8
MOV SBUF, A       ; ส่งข้อมูลไปทางพอร์ทอนุกรม
```

ถ้าเป็นแบบ Odd Parity ให้แก้ไขข้อมูลที่อ่านได้จากบิตพาริตีเสียก่อนที่จะส่งออกไป ซึ่งเขียนโปรแกรมดังนี้

```
MOV C, P           ; อ่านค่าบิตพาริตีมาเก็บใน C
CPL C             ; กลับค่าให้เป็น Odd Parity
MOV TB8, C        ; เขียนค่าลงใน TB8
MOV SBUF, A       ; ส่งข้อมูลออกทางพอร์ทอนุกรม
```

การส่งข้อมูลแบบมีพาริตีบิตด้วยไม่ว่าจะส่งได้แบบ 9 บิต หรือโหมด 2 และ 3 เท่านั้น ในโหมด 1 ซึ่งส่งข้อมูลแบบ 8 บิตก็สามารถทำได้ อย่างเช่นการส่งรหัส ASCII จะใช้บิตข้อมูล 7 บิต สำหรับบิตที่เหลืออีกหนึ่งบิตจะเป็นบิตพาริตีรวมเป็น 8 บิต ซึ่งสามารถเขียนโปรแกรมได้ดังนี้

```
CLR ACC.7         ; เคลียร์ค่าบิต 7 เพื่อใช้เป็นพาริตีบิต
MOV C, P         ; นำบิตพาริตีมาเก็บใน C
MOV ACC.7, C     ; เขียนค่าบิตพาริตีลงในรีจิสเตอร์ A
MOV SBUF, A      ; ส่งข้อมูลออกทางพอร์ทอนุกรม
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แฟลกอินเทอร์รัปต์ เมื่อมีการรับส่งข้อมูลเสร็จสิ้นจะมีผลต่อแฟลกอินเทอร์รัปต์ (RI และ TI) ในรีจิสเตอร์ SCON ซึ่งบิตเหล่านี้จะถูกเซตโดยอุปกรณ์แต่ต้องเคลียร์ด้วยโปรแกรม

บิต RI ถ้าถูกเซตหมายความว่าบัพเฟอร์ที่ไว้รับข้อมูลเต็มให้อ่านข้อมูลไปได้แล้ว และบิตนี้สามารถใช้อินเทอร์รัปต์ MCS – 51 ได้ แต่ถ้าเขียนโปรแกรมจะใช้วิธีตรวจเช็คบิตนี้ ถ้าเป็น “1” จะสามารถอ่านข้อมูลมาเก็บในรีจิสเตอร์ A ได้ แต่ก่อนอ่านจะต้องเคลียร์ RI ก่อนเพื่อจะได้รับข้อมูลถัดไปได้ ซึ่งเขียนโปรแกรมได้ดังนี้

```
WAIT : JNB  NI , WAIT      ; ถ้าบิตนี้ไม่เป็น “1” จะทำงานอยู่ที่เดิม
      CLR  RI              ; เคลียร์ RI
      MOV  A , SBUF       ; อ่านค่ามาเก็บใน A
```

บิต TI เมื่อส่งข้อมูลออกไปแล้วบิตนี้จะถูกเซต เป็นการบอกว่าบัพเฟอร์ส่งข้อมูลว่างแล้วให้ส่งข้อมูลใหม่เข้าไปได้ ซึ่งสามารถใช้บิตนี้อินเทอร์รัปต์ MCS – 51 ได้เช่นกัน แต่ถ้าเขียนโปรแกรมจะเขียนได้ดังนี้

```
WAIT : JNB  RI , WAIT     ; ตรวจบิต TI ว่าเป็น “1” หรือยัง
      CLR  RI              ; เคลียร์ TI
      MOV  SBUF , A       ; เขียนข้อมูลลงไป
```

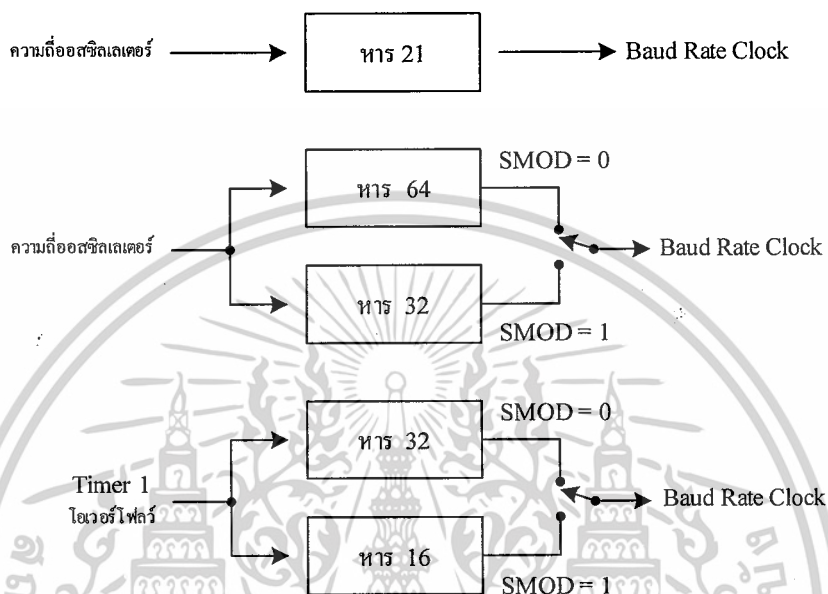
2.4.7 อัตราการส่งข้อมูลของพอร์ตอนุกรม

การรับส่งข้อมูลในโหมดต่างๆ พบว่าในโหมด 0 โหมด 2 ไม่สามารถกำหนดอัตราบอดเองได้ โดยในโหมด 0 ค่าอัตราบอดจะมีค่าเท่ากับความเร็วของตัวกำเนิดความถี่หารด้วย 12 ในโหมด 1 จะมีสองค่าคือ ความถี่ของตัวกำเนิดความถี่หารด้วย 32 และหารด้วย 64 สองค่านี้เรียกว่า SMOD 0 และ SMOD 1 ซึ่งสามารถกำหนดได้ในรีจิสเตอร์ PCON บิตที่ 7 ในรีจิสเตอร์ PCON นี้ไม่สามารถเข้าถึงข้อมูลระดับบิตได้ การเขียนข้อมูลลงไปทีละบิตจะต้องใช้วิธีที่เรียกว่า “Read – Modify – Write” คือ อ่านขึ้นมาแก้ไขแล้วเขียนลงไปใหม่ ตัวอย่างเช่น

```
MOV  A , PCON      ; อ่านค่าจาก PCON มาเก็บในรีจิสเตอร์ A
SETB ACC.7         ; เซตบิต 7 (SMOD)
MOV  PCON , A      ; เขียนค่าลงไปใหม่ใน PCON
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับโหมด 1 และโหมด 3 สามารถกำหนดค่าอัตราบอดได้โดยการโปรแกรมใน Timer1 ในการโปรแกรมแต่ละครั้งจะมี SMOD สองค่าเช่นกัน ค่าอัตราบอดของโหมดต่างๆ แสดงได้ดังรูปที่ 2.27



รูปที่ 2.27 การกำหนดอัตราในโหมดต่างๆ

การใช้ Timer 1 กำหนด Baud Clock

การกำหนดค่าลงใน Timer 1 ทำได้โดยการโปรแกรมไปที่ TMOD ให้ทำงานแบบ 8 bit Auto Reload Mode (โหมด 2) โดยเขียนค่าเขียนที่ TH 1 ซึ่งโปรแกรมไปรีจิสเตอร์ TMOD ได้ดังนี้

```
MOV TMOD, #0010xxxxB
```

ค่า x หมายความว่า เป็นอะไรก็ได้เพราะบิตเหล่านี้ใช้ใน Timer 0

ถ้าต้องการอัตราบอดต่างๆ สามารถใช้ 16 Bit Mode ได้ โดยโปรแกรมเป็น TMOD = 0001xxxxB ค่าอัตราบอดที่ส่งออกจะมีค่าเท่ากับ ความถี่ของ Timer 1 เกิดค่าเกินหารด้วย 32 (หรือ หารด้วย 16 ถ้าเป็น SMOD = 1)

การคำนวณหาค่าอัตราบอดที่กำหนดด้วย Timer 1 สามารถทำได้โดยใช้สมการต่อไปนี้

$$\text{Baud Rate} = \frac{2^{SMOD}}{32} \times \frac{\text{ความถี่ออสซิลเลเตอร์}}{12 \times [256 - TH1]} \quad (2.4)$$

โดยที่ SMOD เป็นค่าของบิตภายในรีจิสเตอร์ PCON ซึ่งอาจมีค่าเป็น 0 หรือ 1

TH เป็นค่าภายในรีจิสเตอร์ TH1 ซึ่งใช้สำหรับย้อนกลับไปเรียกค่าของการนับเวลา

รูปแบบทั่วไปของการหาอัตราบอดในโหมด 1 และ 3 สามารถทำได้ดังนี้

$$\text{Baud Rate} = \frac{\text{Timer 1 Overflow Rate}}{32} \quad (2.5)$$

ถ้าต้องการอัตราบอดเท่ากับ 1,200 สามารถคำนวณค่าความถี่ค่าเกินของ Timer 1 ได้ดังนี้

$$1200 = \frac{\text{Timer 1 Overflow Rate}}{32} \quad (2.6)$$

จะได้ Timer 1 Overflow Rate เท่ากับ 38 kHz

ถ้าระบบ MCS-51 ใช้ความถี่สัญญาณนาฬิกาจากแร่กำเนิดความถี่ เท่ากับ 12 MHz ตัว Timer 1 จะได้รับ Clock เท่ากับ 1 MHz หรือ 1,000 kHz ถ้าเราต้องการ Timer 1 ค่าเกินเท่ากับ 38.4 kHz ดังนั้น ค่าอัตราค่าเกินมีค่าเท่ากับ $1,000/38.4 = 26.04$ Clock โดยค่าเกินจะเกิดเมื่อเกิดการเปลี่ยนจาก FFH เป็น 00H ดังนั้นจะต้องให้ Timer 1 นับไป 26 Count ดังนั้นค่าที่จะให้รีจิสเตอร์ TH1 มีค่าเท่ากับ -26 ซึ่งใช้ค่าที่ย้อนกลับไปดังนั้นเขียนคำสั่งได้ดังนี้

```
MOV TH1, #-26
```

ตัวโปรแกรมแอสเซมบลอร์ทั่วไปจะแปลงค่า -26 เป็น 0E6H เอง จากที่ผ่านมาจะเห็นว่า ความถี่อัตราบอดจะมีความสัมพันธ์กับค่าสัญญาณนาฬิกาที่ใช้ จากแร่กำเนิดความถี่ในตารางที่ 3 จะเป็นค่าที่ต้องกำหนดใน Timer 1 เมื่อต้องการอัตราบอดต่างๆ จะเห็นว่าถ้าใช้ความถี่ 12,000 MHz

ค่า อัตราบอด

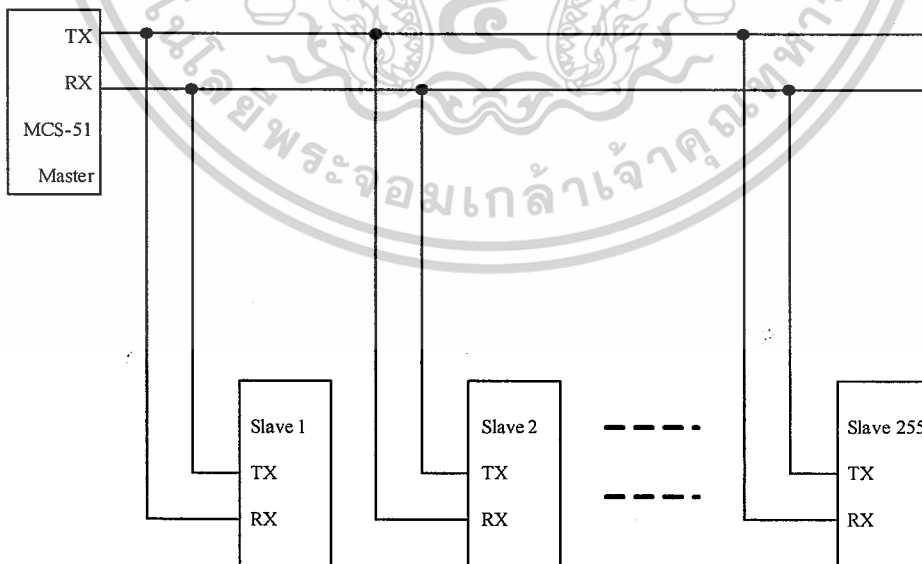
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.8 การสื่อสารข้อมูลระหว่าง MCS – 51 หลายตัว

การใช้งานพอร์ตอนุกรมในโหมด 2 และโหมด 3 ของระบบ MCS – 51 สามารถนำมาใช้กับการสื่อสารของ MCS – 51 หลายๆ ตัวได้ ที่เรียกกันว่า การสื่อสาร แบบระบบมัลติโพรเซสเซอร์ (Multiprocessor System) โดยในระบบจะมี MCS – 51 ตัวหนึ่งเป็นตัวควบคุมการทำงานหลักเรียกว่ามาสเตอร์ (Master) ส่วนตัวอื่นๆ ที่ต่ออยู่ในระบบจะเรียกว่าสลาฟ (Slave) โดยตัวสลาฟแต่ละตัวจะถูกกำหนดค่าแอดเดรสประจำตัวของมันถ้าข้อมูลที่ส่งมีค่าแอดเดรสตรงกับสลาฟตัวใดก็จะทำให้สลาฟตัวนั้นสามารถรับข้อมูลได้

เมื่อเริ่มต้นการทำงาน MCS – 51 ทุกตัวจะถูก โปรแกรมให้พอร์ตอนุกรมทำงานในโหมดเดียวกัน และ MCS – 51 ตัวที่เป็นสลาฟนั้นบิต SM2 จะต้องถูกเซตเป็น “1” ซึ่งจะทำให้สลาฟแต่ละตัวไม่มีการรับส่งข้อมูลซึ่งกันและกัน เมื่อตัวมาสเตอร์ต้องการส่งข้อมูลให้สลาฟจะทำการส่งค่าแอดเดรสของตัวที่ต้องการติดต่อออกมาก่อน ซึ่งจะมีทั้งหมด 9 บิต โดยค่าบิต TB8 จะเป็น “1” เรียกค่าข้อมูล 9 บิตนี้ว่าบัสตำแหน่ง ส่วนตัวสลาฟทุกตัวพร้อมที่จะรับข้อมูลเข้ามาได้เนื่องจากบิต SM2 เป็น “1” เมื่อรับข้อมูลเข้ามาแล้วค่าในบิต RB8 จะมีค่าเป็น “1” จากนั้นตัวสลาฟจะอ่านค่าข้อมูลใน SBUF มาเปรียบเทียบกับแอดเดรสของตัวเองหรือไม่ ถ้าตัวใดตรงกันก็จะทำให้ SM2 เป็น “0” จากนั้นจะเป็นการรับส่งข้อมูลในโหมดปกติ

จะเห็นว่าการกำหนดแอดเดรสของ MCS 51 แต่ละตัวมีขนาด 1 ไบต์ ดังนั้นสามารถต่อตัวสลาฟได้ทั้งหมด 255 ตัว ดังรูปที่ 2.27



รูปที่ 2.28 การต่อ MCS – 51 แบบ Multiprocessor

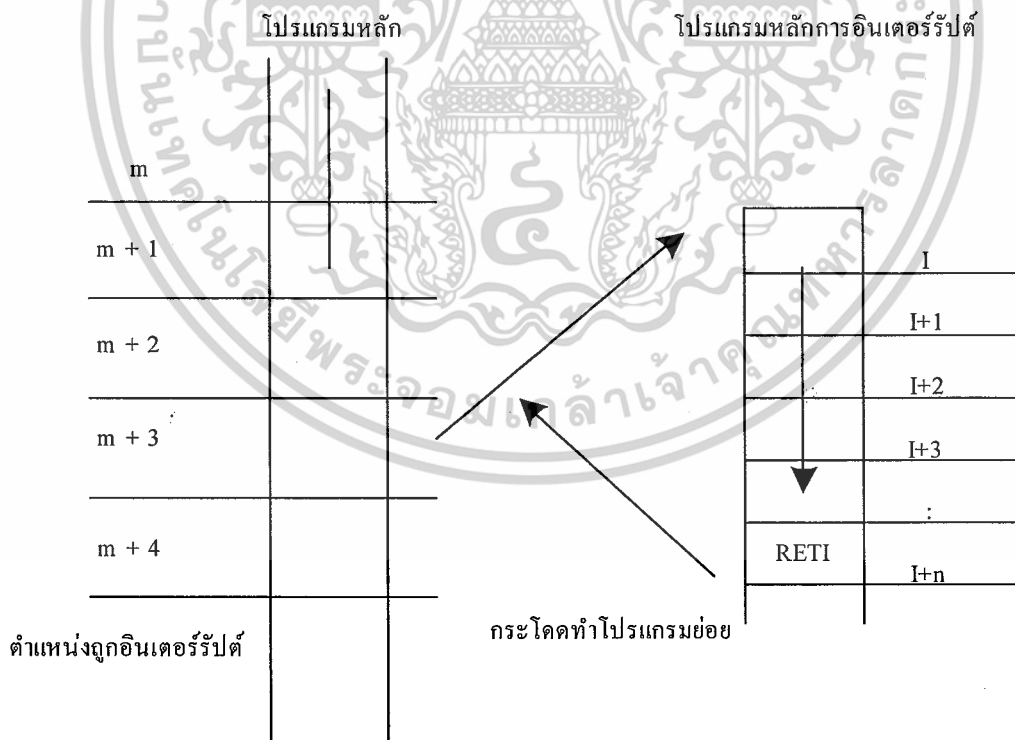
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 การอินเทอร์รัปต์

2.5.1 กระบวนการอินเทอร์รัปต์

ถ้าหากคอมพิวเตอร์กำลังทำงาน โปรแกรมหลักอยู่ เมื่อมีการอินเทอร์รัปต์เข้ามา คอมพิวเตอร์จะหยุดโปรแกรมหลัก ไปทำงานโปรแกรมตอบสนองการอินเทอร์รัปต์ (Interrupt Service Routine) เมื่อทำโปรแกรมตอบสนองอินเทอร์รัปต์เสร็จ คอมพิวเตอร์จะกลับมาทำโปรแกรมเดิม

ถ้า CPU กำลังทำงาน โปรแกรมหลักอยู่ เช่นกำลังทำคำสั่งในตำแหน่งของหน่วยความจำที่ $m, m+1, m+2$ ไปเรื่อยๆ โดย PC จะชี้ที่ตำแหน่งที่จะอ่านคำสั่งถัดมา เมื่อโปรแกรมทำงานถึงตำแหน่งที่ $m+3$ แล้วเกิดการอินเทอร์รัปต์ขึ้น (ขณะนั้น PC อยู่ที่ $m+4$) โปรแกรมจะต้องทำงานโปรแกรมตอบสนองการอินเทอร์รัปต์โดยย้าย PC ไปที่ตำแหน่งที่เก็บโปรแกรมตอบสนองการอินเทอร์รัปต์ จากนั้นจะเก็บค่า PC เดิมในหน่วยความจำสแตค เมื่อคอมพิวเตอร์ทำงานโปรแกรมตอบสนองการอินเทอร์รัปต์เสร็จสิ้นลง จะคืนค่าในสแตค ($m+4$) ให้กับ PC เพื่อให้ PC ทำโปรแกรมหลักต่อไป



รูปที่ 2.29 ขั้นตอนการทำงานของ โปรแกรมเมื่อถูกอินเทอร์รัปต์

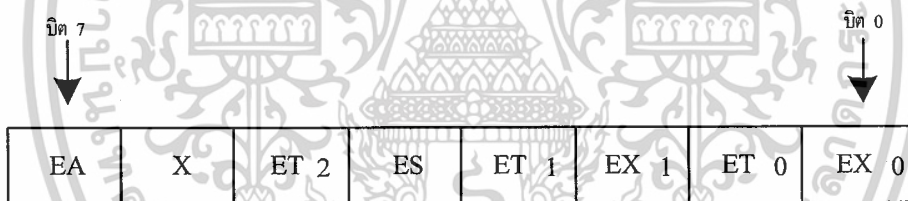
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 สัญญาณอินเทอร์รัปต์

แหล่งกำเนิดสัญญาณอินเทอร์รัปต์ที่ใช้กับ MCS-51 มีสองชนิดคือ อินเทอร์อินเทอร์รัปต์ภายในและภายนอก โดยอินเทอร์รัปต์ภายในจะเกิดขึ้นภายในตัว MCS-51 เอง ได้แก่สัญญาณจาก ไทมเมอร์แฟลค 0 (TF0) ไทมเมอร์แฟลค 1 (TF1) และพอร์ทอนุกรม สำหรับอินเทอร์รัปต์ภายนอกเกิดจากสัญญาณที่กระตุ้นเข้ามาทาง INTO และ INT1 เมื่อมีสัญญาณอินเทอร์รัปต์จากแหล่งต่างๆเข้ามา จะสามารถโปรแกรมได้ว่าจะให้ MCS-51 ยอมให้มีการอินเทอร์รัปต์ได้หรือไม่ โดยการโปรแกรมไปที่ รีจิสเตอร์ IE (Interrupt Enable) และถ้ามีสัญญาณอินเทอร์รัปต์มาจากแหล่งต่างๆ หลายแหล่งพร้อมกันก็จะสามารถจัดลำดับได้ว่า จะให้อินเทอร์รัปต์ใดเกิดก่อน โดยการโปรแกรมไปที่ อินเทอร์รัปต์ไฟโอริตี IP (Interrupt Priority) รีจิสเตอร์ทั้งสองตัวมีรายละเอียดดังต่อไปนี้

1) Interrupt Enables

เป็นรีจิสเตอร์ที่สามารถเข้าถึงข้อมูลระดับบิตได้ ใช้สำหรับกำหนดค่าว่าถ้าเกิดการอินเทอร์รัปต์จากแหล่งต่างๆ จะทำการอินเทอร์รัปต์เหล่านั้นหรือไม่ โดยรายละเอียดของบิตต่างๆ มีดังตารางที่ 2.5



ตารางที่ 2.5 บิตต่างๆ ของรีจิสเตอร์ IE

บิต	ชื่อบิต	ตำแหน่งบิต	รายละเอียด
IE.7	EA	AFH	ถ้าเซตยอมให้มีการอินเทอร์รัปต์
IE.6	-	AEH	สงวนไว้ใช้งานในอนาคต
IE.5	ET2	ADH	Enable อินเทอร์รัปต์จาก Timer 2 (ใช้กับ8052)
IE.4	ES	ACH	Enable อินเทอร์รัปต์จากพอร์ทอนุกรม
IE.3	ET1	ABH	Enable อินเทอร์รัปต์จาก Timer 1
IE.2	EX1	AAH	Enable อินเทอร์รัปต์จาก INT1
IE.1	ET0	A9H	Enable อินเทอร์รัปต์จาก Timer 0
IE.0	EX0	A8H	Enable อินเทอร์รัปต์จาก INTO

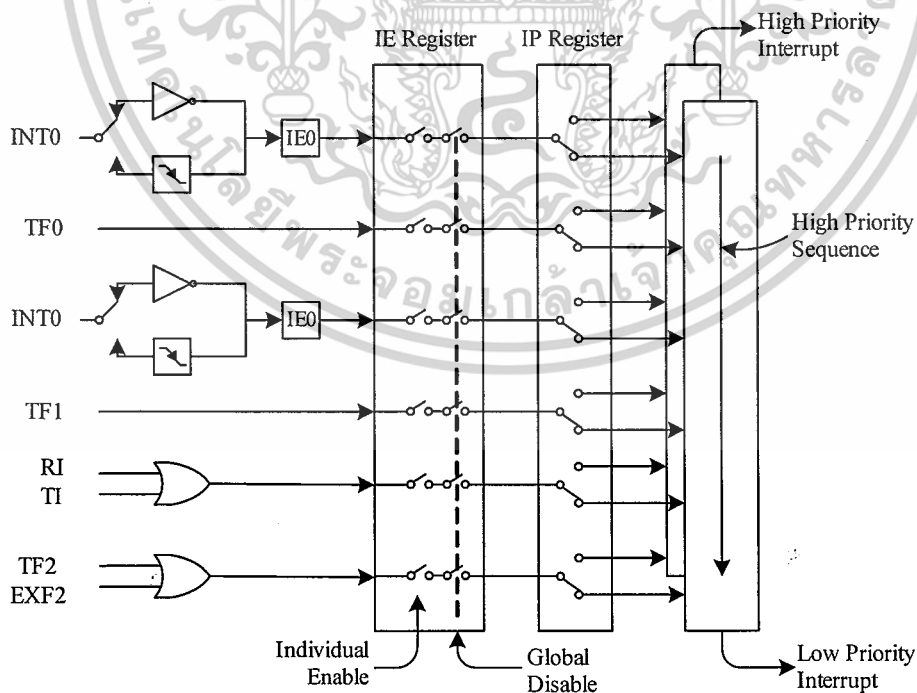
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) Interrupt Priority

เป็นรีจิสเตอร์ที่สามารถเข้าถึงข้อมูลระดับบิตได้ ใช้ในการจัดลำดับความสำคัญของการอินเตอรัปต์ซึ่งสามารถจัดได้สองลำดับ ถ้าเป็น “1” หมายความว่ามีความสูงสุดเป็น “0” หมายความว่ามีความสำคัญต่ำสุด ความหมายของบิตต่างๆ แสดงได้ดังตารางที่ 2.5 ถ้าหากกำหนดให้มีความสำคัญเป็น “1” เหมือนกันหมด MCS-51 จะจัดลำดับความสำคัญใหม่ดังนี้

ตารางที่ 2.6 การจัดลำดับความสำคัญของการอินเตอรัปต์

ลำดับ	อินเตอรัปต์
1 (สูงสุด)	IE0
2	TF0
3	IE1
4	TF1
5 (ต่ำสุด)	Serial Port



รูปที่ 2.30 รีจิสเตอร์ต่างๆ ที่เกี่ยวข้องกับการอินเตอรัปต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.7 บิตและหน้าที่ต่างๆ ของรีจิสเตอร์ IP

บิต	ชื่อบิต	ตำแหน่งบิต	รายละเอียด
IP.7	-	-	ไม่ใช้งาน
IP.6	-	-	ไม่ใช้งาน
IP.5	PT2	0BDH	ใช้กับ Timer 2 (8052)
IP.4	PS	0BCH	ใช้กับพอร์ทอนุกรม
IP.3	PT1	0BBH	ใช้กับ Timer 1
IP.2	PX1	0BAH	ใช้กับอินเตอร์รัปต์จาก INT1
IP.1	PT0	0B9H	ใช้กับ Timer 0
IP.0	PX0	0B8H	ใช้กับอินเตอร์รัปต์จาก INTO

จากรูปที่ 2.7 แสดงการอินเตอร์รัปต์จากแหล่งต่างๆ ที่มีผลกับ MCS-51 ถ้าเป็นเบอร์ 8051, 8031 จะถูกอินเตอร์รัปต์ได้ 5 แหล่ง ถ้าเป็นเบอร์ 8052 , 8032 จะถูกอินเตอร์รัปต์ได้ 6 แหล่ง โดยเพิ่มอินเตอร์รัปต์จาก Timer 2 ในรูปที่ 2.30 จะแสดงให้เห็นว่า ถ้า MCS-51 จะถูกอินเตอร์รัปต์ได้จะต้องเซตค่า Global Enable ในรีจิสเตอร์ IE นอกจากนี้ยังกำหนดได้ว่าจะให้อินเตอร์รัปต์ใดเกิดขึ้นได้ โดยการเซตค่า Interrupt Enable ของอินเตอร์รัปต์จากแหล่งต่างๆ ในรีจิสเตอร์ IE จากรูปยังแสดงให้เห็นอีกว่าเมื่อมีการอินเตอร์รัปต์เข้าจะมีผลต่อแฟล็กใด เช่นถ้า INTO เป็น “1” บิต IE0 จะเป็น “1” หมายความว่าถูกอินเตอร์รัปต์ โดยแฟล็กต่างๆ ที่มีผลจากการถูกอินเตอร์รัปต์แสดงได้ดังตารางที่ 2.8

ตารางที่ 2.8 แฟล็กที่จะทำงานเมื่อถูกอินเตอร์รัปต์

อินเตอร์รัปต์	แฟล็ก	ประกอบอยู่ในรีจิสเตอร์
External 0	IE0	TCON.1
External 1	IE1	TCON.3
Timer 1	TF1	TCON.7
Timer 0	TF0	TCON.5
Serial port	T1	SCON.1
Serial port	RI	SCON.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.8 (ต่อ) แฟลคที่จะทำงานเมื่อถูกอินเทอร์รัปต์

อินเทอร์รัปต์	แฟลค	ประกอบอยู่ในรีจิสเตอร์
Timer 1	TF2	T2CON.7 (8052)
Timer 2	EXF2	T2CON.6 (8052)

จากตารางจะเห็นว่า ถ้ามีการอินเทอร์รัปต์จากภายนอกเข้า ตัวที่จะอินเทอร์รัปต์ MCS-51 คือ บิตแฟลค IE0 ซึ่งอยู่ในรีจิสเตอร์ TCON ถ้ามีการสื่อสารแบบอนุกรม เมื่อข้อมูลถูกส่งไปหมดแล้วจะอินเทอร์รัปต์ MCS-51 ทางบิตแฟลค TI ถ้ารับข้อมูลหมดแล้วจะอินเทอร์รัปต์ MCS-51 ทางบิตแฟลค RI ซึ่งอยู่ในรีจิสเตอร์ SCON และถ้าใช้ Timer 0 ในการนับเมื่อเกิดค่าเกินสามารถอินเทอร์รัปต์ MCS-51 ได้ทางบิต TF0

2.5.3 การทำงานของระบบหลังถูกอินเทอร์รัปต์

เมื่อ MCS-51 ถูกอินเทอร์รัปต์ จะต้องกระโดดไปทำโปรแกรมตอบสนองการอินเทอร์รัปต์ โดยตำแหน่งที่กระโดดไปเรียกว่า อินเทอร์รัปต์เวกเตอร์ (Interrupt Vectors) เมื่อทำการโปรแกรมตอบสนองการอินเทอร์รัปต์เรียบร้อยแล้ว MCS-51 จะกระโดดมาทำงานยังตำแหน่งเดิม โดยก่อนที่จะกระโดดไปทำโปรแกรมตอบสนองการอินเทอร์รัปต์จะต้องเก็บค่าตำแหน่งเดิมไว้ โดยเก็บค่า PC ลงหน่วยความจำสแตคซึ่งอยู่ที่หน่วยความจำที่ถูกชี้โดยรีจิสเตอร์ SP เมื่อทำโปรแกรมตอบสนองการอินเทอร์รัปต์เสร็จแล้วจะคืนค่าในหน่วยความจำสแตคให้ PC ตามเดิม ค่าอินเทอร์รัปต์เวกเตอร์ของ MCS-51 แสดงได้ดังตารางที่ 2.9

ตารางที่ 2.9 อินเทอร์รัปต์เวกเตอร์ของMCS-51

อินเทอร์รัปต์	อินเทอร์รัปต์เวกเตอร์
System Reset	0000H
External 0	0003H
Timer 0	000BH
External 1	0013H
Timer 1	001BH
Serial Port	0023H
Timer 2	002BH

จากตารางจะเห็นว่าถ้าระบบถูกอินเทอร์รัปต์จากภายนอกทาง INT0 ตัว MCS-51 จะกระโดดไปทำงานที่ตำแหน่ง 0003H ถ้าระบบถูกอินเทอร์รัปต์จาก Timer 0 จะกระโดดไปทำงานตำแหน่ง 000BH

2.5.4 การออกแบบโปรแกรมอินเทอร์รัปต์

การเขียนโปรแกรมหลัก (Main Program) จะต้องกำหนดค่าให้ MCS-51 ถูกอินเทอร์รัปต์ด้วยอะไร และจะให้ MCS-51 ถูกอินเทอร์รัปต์ได้หรือไม่ โดยการโปรแกรมค่าต่างๆ ใน IE รีจิสเตอร์ ถ้ามีการอินเทอร์รัปต์จากสองแหล่งขึ้นไปควรมีการจัดลำดับความสำคัญในรีจิสเตอร์ IP ดังนั้นโปรแกรมหลักจะต้องมีการโปรแกรมดังนี้

1) โปรแกรมค่าในรีจิสเตอร์ IE

2) โปรแกรมค่าในรีจิสเตอร์ IP

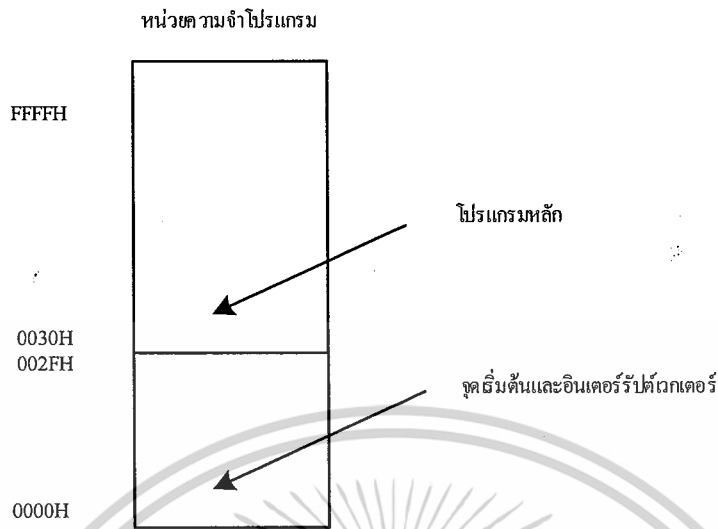
สำหรับโปรแกรมการตอบสนองการอินเทอร์รัปต์ถือว่าเป็นโปรแกรมย่อยโปรแกรมหนึ่ง แต่จะต้องจบโปรแกรมย่อยด้วยคำสั่ง RETI (Return from Interrupt)

จากตารางอินเทอร์รัปต์เวกเตอร์ จะเห็นว่าถ้ากรีเซตหรือให้ระบบเริ่มทำงาน โปรแกรมจะเริ่มทำงานที่ตำแหน่ง 0000 H และตำแหน่งที่เก็บ โปรแกรมหลักมีโอกาสที่จะทับกับหน่วยความจำ โปรแกรมที่เก็บค่าอินเทอร์รัปต์เวกเตอร์ที่ตำแหน่ง 0003 H ถ้าโปรแกรมายาวมากอาจจะไปทับตำแหน่ง 000BH ได้ซึ่งเป็นตำแหน่งของอินเทอร์รัปต์เวกเตอร์ของ Timer 0 ดังนั้นในการเขียนโปรแกรมหลัก ภายใน 3 ตำแหน่งแรกคือ 0000H, 0001H, 0002H จะต้องกระโดดไปที่อื่นก่อนเพื่อให้ข้ามอินเทอร์รัปต์เวกเตอร์ไปซึ่งอาจเขียนโปรแกรมได้ดังนี้

```

ORG    0000H    ; เริ่มต้น โปรแกรม
LJMP   MAIN    ; กระโดดไปโปรแกรมหลัก
.....        ; เพื่อหนีอินเทอร์รัปต์เวกเตอร์
.....
ORG    0030H    ; ตำแหน่งเริ่มต้นของ โปรแกรม
MAIN:  .....   ; เริ่มต้น โปรแกรมหลัก
.....

```



รูปที่ 2.31 การจัดตำแหน่งโปรแกรมในหน่วยความจำ

จากตัวอย่างโปรแกรมจะเห็นว่า เมื่อเริ่มต้นโปรแกรมหรือระบบบูทรีเซต ระบบจะทำงานตำแหน่งแรกคือคำสั่งกระโดดไปโปรแกรมหลัก ซึ่งอยู่ต่อจากโปรแกรมตอบสนอง โดยการอินเทอร์รัปต์ที่อยู่ตำแหน่ง 0030H

1) โปรแกรมการตอบการอินเทอร์รัปต์แบบสั้น

จากตารางอินเทอร์รัปต์เวกเตอร์ จะเห็นว่าที่เก็บโปรแกรมอินเทอร์รัปต์แต่ละแห่งจะห่างกัน 8 ไบต์ ดังนั้นถ้ามีการอินเทอร์รัปต์จากแหล่งต่าง ๆ หลายแหล่งและโปรแกรมตอบสนองการอินเทอร์รัปต์บางโปรแกรมมีขนาดยาวเกิน 8 ไบต์ จะทำให้โปรแกรมจะไปทับกับตำแหน่งของโปรแกรมตอบสนองการอินเทอร์รัปต์ของอินเทอร์รัปต์ถัดไป แต่ถ้าโปรแกรมการตอบสนองการอินเทอร์รัปต์ไม่ยาวมากเกินไปสามารถเขียนไปในตำแหน่งนั้นได้เลยตั้งโปรแกรมต่อไปนี้

```

ORG 0000H
LJMP MAIN ; กระโดดไปโปรแกรมหลัก

ORG 000BH ; ตำแหน่งเริ่มต้นของอินเทอร์รัปต์ Timer 0
TOISR : .....

.....

RETI ; กลับโปรแกรมหลัก

MAIN : ..... ; โปรแกรมหลัก

.....

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างโปรแกรมจะใช้อินเทอร์รัปต์จาก Timer 0 เมื่อระบบเริ่มทำงานจะทำตำแหน่ง 0000H โดยกระโดดไปโปรแกรมหลักซึ่งอยู่ที่ตำแหน่งต่อจากโปรแกรมตอบสนองการอินเทอร์รัปต์เมื่อมีการอินเทอร์รัปต์เวกเตอร์ Timer 0 ระบบจะทำโปรแกรมตำแหน่งที่ 000BH ซึ่งเป็นอินเทอร์รัปต์เวกเตอร์ของ Timer 0 โดยโปรแกรมตอบสนองการอินเทอร์รัปต์จะจบด้วยคำสั่ง RETI เพื่อกลับสู่โปรแกรมหลักต่อไป

2) โปรแกรมตอบสนองการอินเทอร์รัปต์ขนาดใหญ่

ในกรณีที่มีการอินเทอร์รัปต์จากหลายแหล่ง และโปรแกรมตอบสนองการอินเทอร์รัปต์แต่ละโปรแกรมยาวเกิน 8 ไบต์ จะไม่สามารถเขียนโปรแกรมตอบสนองการอินเทอร์รัปต์เวกเตอร์ได้ ซึ่งจะแก้ปัญหานี้ได้โดยกำหนดให้ตำแหน่งของอินเทอร์รัปต์เวกเตอร์ให้ทำโปรแกรมกระโดด โดยกระโดดไปที่ตำแหน่งเก็บโปรแกรมตอบสนองการอินเทอร์รัปต์เขียนไว้ที่ตำแหน่งอื่นดังตัวอย่างต่อไปนี้

```

ORG 0000H      ; เริ่มโปรแกรมของระบบ
LJMP MAIN     ; กระโดดไปโปรแกรมหลัก
ORG 000BH      ; ตำแหน่งของอินเทอร์รัปต์ Timer 0
LJMP LED1     ; กระโดดไปโปรแกรมตอบสนองการอินเทอร์รัปต์ชื่อ
                LED1
ORG 0030H      ; ตำแหน่งหลังอินเทอร์รัปต์เวกเตอร์
MAIN:          ; โปรแกรมหลัก
.....
LED:           ; โปรแกรมการตอบสนองการอินเทอร์รัปต์ Timer 1
.....
RETI          ; กลับสู่โปรแกรมหลัก

```

จากโปรแกรมจะเห็นว่า เมื่อระบบทำงานจะต้องทำที่ตำแหน่ง 0000H โดยกระโดดไปทำโปรแกรมหลักที่ตำแหน่งต่อจาก 0030H เพราะตำแหน่งดังกล่าวข้ามอินเทอร์รัปต์เวกเตอร์จากตำแหน่งต่างๆ ไปแล้วเมื่อมีการอินเทอร์รัปต์จาก Timer 0 โปรแกรมจะต้องทำงานที่ตำแหน่ง 000BH แต่โปรแกรมตอบสนองการอินเทอร์รัปต์ยาวมากที่ตำแหน่ง 000BH จึงทำให้โปรแกรมกระโดดไปที่โปรแกรมตอบสนองการอินเทอร์รัปต์ชื่อ LED1 ซึ่งอยู่ท้ายโปรแกรม เมื่อจบโปรแกรมจะจบด้วยคำสั่ง RETI เพื่อกลับไปโปรแกรมหลักต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 การเขียนโปรแกรมด้วยภาษาซี

2.6.1 โปรแกรม C51

โปรแกรม C51 เป็นซีคอมไพเลอร์ที่ออกแบบมาสำหรับการเขียนภาษาซี กับไมโครคอนโทรลเลอร์เบอร์ MCS-51 โดยทั่วไปนักพัฒนาไมโครคอนโทรลเลอร์ที่เขียนด้วยภาษาแอสเซมบลีจะเขียนโปรแกรมและแปลให้เป็นออบเจกต์ไฟล์ (นามสกุล .OBJ) หรือเฮกไฟล์ (นามสกุล .HEX) จากนั้นจะนำโปรแกรมลงบนชิพไมโครคอนโทรลเลอร์ แต่การเขียนโปรแกรมด้วยภาษาซีจะมีขั้นตอนต่างๆ มากขึ้นดังนี้

- 1) เขียนชุดคำสั่งบน โปรแกรม Editor ให้มีนามสกุลเป็น .C
- 2) กำหนดพรีโพรเซสเซอร์ (Preprocessor) สำหรับกำหนดการคอมไพล์โปรแกรม ซึ่งอาจกำหนดไว้ใน Source Code หรือกำหนดตอนคอมไพล์ก็ได้
- 3) คอมไพล์โปรแกรม ในที่นี้จะใช้โปรแกรม C51 ในการคอมไพล์ และจะได้ไฟล์ออบเจกต์ที่มีนามสกุลเป็น .OBJ ออกมา
- 4) เชื่อมต่อชุดคำสั่งเสริม (Link) โดยใช้โปรแกรม L51 ซึ่งจะทำการรวมชุดคำสั่งต่างๆ ที่อ้างอิงถึงกัน และโปรแกรมจะสร้างออบเจกต์ที่สมบูรณ์ออกมา
- 5) แปลงเป็นไฟล์ Hexadecimal File ด้วยโปรแกรม OHS51 ซึ่งจะได้ไฟล์นามสกุล .HEX ออกมา

ถ้าหากใช้คอมไพเลอร์ตัวอื่นๆ ขั้นตอนเหล่านี้อาจแตกต่างกันไป

2.6.2 คำสั่งพื้นฐานและการประกาศตัวแปร

```
# include<stdio.h>
main( )
{
    printf ( " Hello Program C \ n" );
}
```

เมื่อรันโปรแกรมคอมพิวเตอร์จะแสดงว่า Hello Program C ออกทางจอภาพ จากโปรแกรมจะพบว่าบรรทัดแรกจะเป็นการเรียกฟังก์ชันที่ใช้ติดต่อกับไอโอมาตรฐานออกมา จากนั้นก็เรียกใช้ printf ที่เก็บอยู่ใน Stdio.h นั้น สำหรับการเขียนโปรแกรมกับไมโครคอนโทรลเลอร์การทำงานต่างๆ จะเป็นการกระทำกับพอร์ต รีจิสเตอร์ และหน่วยความจำต่างๆ ดังนั้นผลลัพธ์จากการทำงานของโปรแกรมจะเกี่ยวข้องกับการย้ายข้อมูลระหว่างรีจิสเตอร์ และหน่วยความจำต่างๆ เป็นส่วนใหญ่

ถ้าหากต้องการเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ส่งค่าตั้งแต่ 0 ถึง 255 ออกไปทางพอร์ต P1 สามารถเขียนโปรแกรมด้วยภาษาแอสเซมบลีได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ORG    0000H
MOV    R1 , #00D
LOOP : MOV    P1 , R1
      INC    R1
      CJNE  R1 , #255D , LOOP

```

แต่ถ้าหากเขียนโปรแกรมเป็นภาษาซีจะได้เป็น

```

#include < reg 52.h >
main ( )
{
    unsigned char i;
    for ( i = 0; i <= 255; i++)
        P1 = i
}

```

ในบรรทัดแรกจะคล้ายกับบรรทัดแรกของการเขียนโปรแกรมบนเครื่องคอมพิวเตอร์ PC โดยจะบอกให้คอมไพเลอร์รู้จักตัวแปรและรีจิสเตอร์ต่างๆ ของ MCS-51 โดยในไฟล์ reg52.h จะบรรจุชื่อรีจิสเตอร์และหน่วยความจำต่างๆ ของ 8052 เอาไว้

บรรทัดที่สองจะเป็นชื่อฟังก์ชันการทำงานหลัก (main) โดยคำสั่งการทำงานต่างๆ จะอยู่ในเครื่องหมายปีกกาเปิดและปีกกาปิด ต่อมาจะเป็นการประกาศตัวแปร i ให้เป็นตัวแปรแบบ char ไม่คิดเครื่องหมายซึ่งจะเก็บข้อมูลได้ตั้งแต่ 0 ถึง 255 บรรทัดต่อไปจะเป็นประโยคคำสั่งวนลูปซึ่งจะทำให้โปรแกรมทำงานต่อเนื่องโดยนำค่า 0 ถึง 255 ส่งออกไปทางพอร์ต P1 ครั้งละค่า

การเขียนโปรแกรมด้วยภาษาซีนี้สามารถเขียนไคเร็กทีฟควบคุม (Control Directive) ได้ซึ่งจะเป็นการบอกว่าจะให้ทำการใดๆ ขณะคอมไพล์โปรแกรม โดยอาจเขียนตอนคอมไพล์ หรือเขียนภายในโปรแกรมก็ได้ โดยการกำหนดในส่วนของ Preprocess ซึ่งจะใช้คำว่า #pragma นำหน้าส่วนของไคเร็กทีฟควบคุมนั้นมีอยู่หลายคำสั่งสำหรับรายละเอียดต่างๆ จะกล่าวในภายหลัง สำหรับคำสั่งที่ใช้กันมากที่สุดคือคำสั่ง code ซึ่งจะบอกให้สร้างภาษาแอสเซมบลีขณะที่ทำการแปลโปรแกรมด้วย

2.6.3 โครงสร้างของภาษาซี

ด้วยภาษาซีเป็นภาษาที่สามารถเขียนโปรแกรมเป็นแบบโครงสร้างได้ โดยโปรแกรมจะแบ่งการทำงานต่างๆ ออกเป็นกลุ่มๆ หรือฟังก์ชัน โดยฟังก์ชันเหล่านั้นสามารถเรียกขึ้นมาใช้ใหม่ได้ ในการเขียนโปรแกรมจะต้องระบุไว้ว่าในโปรแกรมนั้นมีฟังก์ชันใดให้ใช้บ้าง แต่ทุกโปรแกรมจะต้องมีฟังก์ชันหลักที่มีชื่อว่า main() เสมอ

2.6.3.1 ตัวแปรและค่าคงที่

การใช้งานตัวแปรและค่าคงที่ต่างๆ จะต้องมีการประกาศชื่อตัวแปรขึ้นมาเสียก่อนเมื่อมีการคอมไพล์โปรแกรมตัวคอมไพเลอร์จะเตรียมพื้นที่ในหน่วยความจำแรมเอาไว้สำหรับเก็บตัวแปรและค่าคงที่เหล่านั้นในการประกาศตัวแปรสามารถทำได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเภทของข้อมูล

ชื่อตัวแปร [...];

การประกาศตัวแปรจะต้องเริ่มด้วยชื่อประเภทของข้อมูล และตามด้วยชื่อตัวแปร โดยจะประกาศครั้งละกี่ตัวก็ได้ ส่วนชื่อประเภทของข้อมูล และตามด้วยชื่อตัวแปร โดยจะประกาศครั้งละกี่ตัวก็ได้ ส่วนชื่อของตัวแปรนั้นจะซ้ำกันไม่ได้และต้องไม่ซ้ำกับชื่อของคำสงวน (Keywords) ของคอมไพเลอร์ตัวนั้นๆ สำหรับประเภทของข้อมูลในการประกาศตัวแปรเป็นดังนี้

ตารางที่ 2.10 การประกาศตัวแปรของภาษาซี

ประเภทของข้อมูล	ขนาด (บิต)	ค่าที่เก็บได้
bit	1	0 ถึง 1
char	8	-128 ถึง 127
unsigned char	8	0 ถึง 255
int	16	-32768 ถึง 32767
unsigned int	16	0 ถึง 65535
long	32	-2147483648 ถึง +2147483647
unsigned long	32	0 ถึง 4294967295
float	32	-1.17549e -38 ถึง e +3.402823 +38

สำหรับคำสงวนเป็นคำที่คอมไพเลอร์รู้จักและถูกใช้งานเฉพาะ ไม่สามารถนำมาตั้งชื่อตัวแปรและฟังก์ชันได้ คำสงวนของโปรแกรม C51 เป็นดังต่อไปนี้

at	idata	sfr
alien	interrupt	sfr18
bdata	large	small
bit	pdata	_task_
code	_priority_	using
compact	reentrant	xdata
data	sbit	

2.6.4 ตัวดำเนินการในภาษาซี

ตัวดำเนินการจะเป็นตัวที่ใช้กระทำกับตัวแปร ค่าคงที่ต่างๆ ให้รวมเป็นค่าเดียวกันโดยอาจกระทำทางคณิตศาสตร์หรือกระทำทางลอจิกก็ได้ ในการเขียนโปรแกรมด้วยภาษานั้น ตัวดำเนินการเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจะแบ่งออกเป็นสองกลุ่มใหญ่ๆ คือตัวดำเนินการกระทำกับตัวถูกกระทำตัวเดียว (Single Operand Operators หรือ unary operator) และตัวดำเนินการ ที่กระทำกับตัวถูกกระทำสองตัว (Two Operands Operators หรือ binary operator)

1) ตัวดำเนินการกระทำกับตัวถูกกระทำตัวเดียว

ตัวดำเนินการประเภทนี้จะกระทำกับตัวถูกกระทำเพียงตัวเดียว ประกอบด้วยตัวดำเนินการต่างๆ ดังนี้

- ลบ (Negate)
- ~ กลับค่าลอจิกของบิตข้อมูล (Bit Wise Complement)
- ! กลับค่าทางลอจิก (Logic Complement)
- ++ เพิ่มค่าขึ้นหนึ่งค่า (Increment)
- ลดค่าลงหนึ่งค่า (Decrement)
- * ตัวดำเนินการทางพอยน์เตอร์
- & ตำแหน่งหน่วยความจำของตัวแปร

2) ตัวดำเนินการที่กระทำกับตัวถูกกระทำสองตัว

ตัวดำเนินการประเภทนี้จะกระทำกับตัวถูกกระทำสองตัว ถ้าหากมีตัวถูกกระทำหลายๆ ตัวสามารถนำมาเขียนรวมกันเป็นประโยคได้ ซึ่งประกอบไปด้วยตัวดำเนินการที่ใช้กำหนดค่า ตัวดำเนินการทดสอบค่าซึ่งจะให้ผลลัพธ์เป็นค่าทางลอจิก (จริง, เท็จ) ตัวดำเนินการทางคณิตศาสตร์ และตัวดำเนินการทางลอจิกดังต่อไปนี้

- = กำหนดค่าในประโยค (Assignment)
- + บวก
- ลบ
- * คูณ
- / หหาร
- % เศษที่เหลือจากการหาร (Modulo)
- && การแอนด์ (Logical AND)
- || กรออร์ (Logical OR)
- & การแอนด์แบบบิตต่อบิต (Bit Wise AND)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	การออร์แบบบิตต่อบิต (Bit Wise OR)
^	การเอ็กคลูซีฟออร์แบบบิตต่อบิต (Bit Wise Exclusive OR)
<<	เลื่อนบิตไปทางซ้าย
>>	เลื่อนบิตไปทางขวา
==	ทดสอบว่าเท่ากันหรือไม่
!=	ทดสอบว่าไม่เท่ากันหรือไม่
>	ทดสอบว่ามากกว่าหรือไม่
<	ทดสอบว่าน้อยกว่าหรือไม่
>=	ทดสอบว่ามากกว่าหรือเท่ากับหรือไม่
<=	ทดสอบว่าน้อยกว่าหรือเท่ากับหรือไม่

นอกจากนี้ตัวดำเนินการบางประเภทสามารถนำมารวมกันเป็น Compound Operators ได้ ตัวอย่างเช่น ถ้ามีประโยค $y = y * 2$; อาจเขียนตัวดำเนินการรวมเป็น $y *= 2$; รูปแบบของตัวดำเนินการที่รวมกันได้เป็นดังนี้

+=	บวกและให้เท่ากับบวกด้วย
-=	ลบและให้เท่ากับลบด้วย
*=	หารและให้เท่ากับหารด้วย
%=	หารแบบบวกและให้เท่ากับ
&=	ทำการแอนด์และให้เท่ากับ
=	ทำการออร์และให้เท่ากับ
^=	ทำการเอ็กคลูซีฟออร์และให้เท่ากับ
<<=	เลื่อนบิตไปทางซ้ายและไม่เท่ากับ
>>=	เลื่อนบิตทางขวาและไม่เท่ากับ

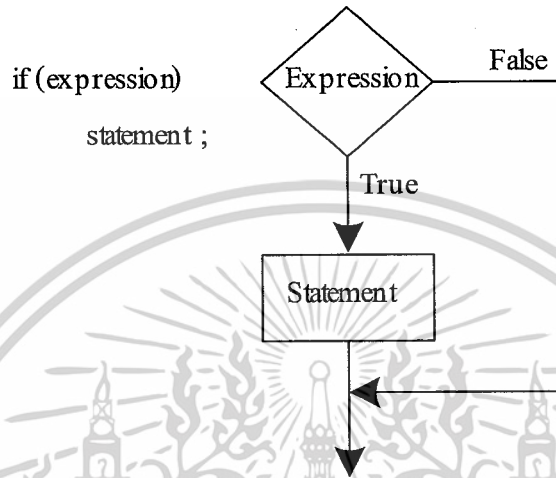
2.6.5 ประโยคควบคุมในภาษาซี

การทำงานของโปรแกรมนั้นจะทำคำสั่งแต่ละคำสั่งเรียงลำดับกันไป และเราสามารถให้โปรแกรมตัดสินใจในการเลือกได้ หรือให้ทำงานใดงานหนึ่งซ้ำๆ ตามเงื่อนไขที่กำหนดได้โดยใช้คำสั่งควบคุมในภาษาซีนั้นจะมีประโยคคำสั่งควบคุมที่ใช้ในการเลือกทำและทำงานซ้ำๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) ประโยค IF / ELSE

ประโยคคำสั่งนี้จะใช้ควบคุมทิศทางการทำงานของโปรแกรม โดยจะถูกแปลออกมาเป็นคำสั่งในภาษาแอสเซมบลีดังนี้ jz , jnz , jb , jnb , jc และ jnc โดยมีรูปแบบประโยคคำสั่งดังนี้



รูปที่ 2.32 ประโยคทดสอบการทำงานของสแตตเมนต์แบบทางเลือกเดียว

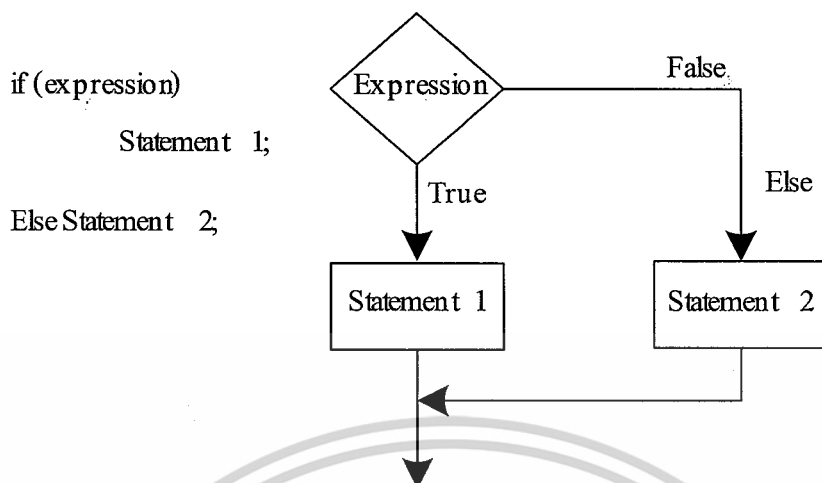
ประโยคนี้ใช้ในการทดสอบว่าจะทำสแตตเมนต์ที่ตามมาหรือไม่ ถ้าค่าใน Expression เป็นจริงหรือมีค่าไม่เป็นศูนย์จะทำสแตตเมนต์ที่ตามมา ถ้าเป็นเท็จหรือมีค่าเป็นศูนย์จะไม่ทำ และสแตตเมนต์ที่จะทำงานนั้นอาจเป็นประโยคคำสั่งประโยคเดียว หรือเป็นสแตตเมนต์ซ้อนก็ได้ (ต้องมีปีกกาคลุม) ตัวอย่างเช่น ถ้าค่าของพอร์ต P1 มีค่าไม่เป็น 0 ให้ตัวแปร C เป็นศูนย์ จะเขียนได้ดังนี้

```

if (P1 != 0)
C = 0

```

ถ้าหากเป็นการทำงานเลือกทำแบบมีสองทางเลือก และต้องการทำงานเพียงอย่างใดอย่างหนึ่งจะใช้ประโยค if - else ซึ่งมีรูปแบบดังนี้



รูปที่ 2.33 ประโยคทดสอบการทำงานของสแตตเมนต์แบบ 2 ทางเลือก

2) ประโยค Switch

การเลือกทำที่มีทางเลือกหลายๆ ทางเลือกนั้นเราสามารถนำประโยค if – else มาซ้อนกันได้ แต่จะทำให้มองดูเข้าใจได้ยาก ในภาษาซีมีประโยค switch ที่ใช้ในการเลือกทำอย่างใดอย่างหนึ่งจากหลายๆ ทางเลือกโดยมีรูปแบบของประโยคดังนี้

```

switch(k)
{
    case 1:      statement 1:
                break;
    case 2:      statement 2:
                break;
    case 3:      statement 3:
                break;
                :
                :
    default:    statement n;
}
  
```

การทำงานของโปรแกรมจะนำค่าในตัวแปร k ที่อยู่ในวงเล็บหลัง switch มาเปรียบเทียบกับเท่ากับค่าคงที่หลังคำสั่ง case ตัวใด และจะทำงานสแตตเมนต์ที่ตามหลัง case นั้น และจะออกนอกปีกกาของ switch เมื่อพบคำสั่ง break โดยสแตตเมนต์ที่ทำงานนั้นจะเป็นสแตตเมนต์ซ้อนก็ได้ แต่ถ้าค่าในตัวแปร k ไม่เท่ากับค่าคงที่ค่าใดเลยโปรแกรมจะทำสแตตเมนต์ที่ตามหลัง default

2.6.6 การทำซ้ำ

คำสั่งให้โปรแกรมทำงานซ้ำถือว่าเป็นประโยคคำสั่งควบคุมอย่างหนึ่ง การทำซ้ำหรือที่เรียกว่าการทำลูปนั้นจะมีประโยคคำสั่งอยู่สามประเภทคือ for , while และ do – while ซึ่งแต่ละแบบจะต่างกันตรงเงื่อนไขของการทำซ้ำ

1) ประโยค for

ประโยคคำสั่งนี้จะใช้กรณีที่มีจำนวนรอบของการทำซ้ำที่แน่นอน โดยมีรูปแบบดังนี้

```
for (initialization ; condition ; increment)
    statement ;
```

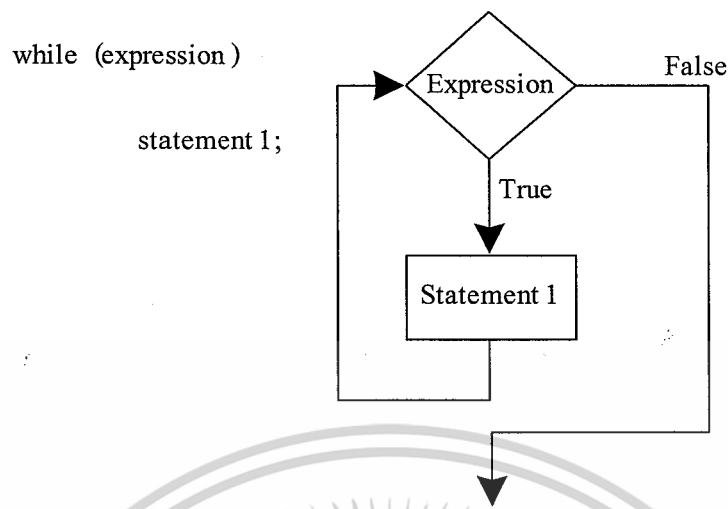
โดยที่ initialization เป็นค่ากำหนดเริ่มต้นให้กับตัวแปรของการทำลูป condition เป็นเงื่อนไขที่ใช้ทดสอบการทำซ้ำครั้งต่อไป ซึ่งจะเป็นการกระทำลูป increment เป็นการเพิ่มค่าให้ตัวแปรในการทำซ้ำแต่ละครั้ง สำหรับ statement จะเป็นสแตตเมนต์ของคำสั่งที่จะทำซ้ำ ซึ่งอาจเป็น สแตตเมนต์รวมก็ได้ ตัวอย่างเช่น

```
unsigned char x;
for (x = 0; x <= 255; x++)
    P1 = x;
```

การทำงานตามชุดคำสั่งข้างบนจะเป็นการส่งค่า 0 ถึง 255 ออกมาทางพอร์ต P1 โดยเริ่มต้นจะประกาศตัวแปร x สำหรับนับการทำซ้ำ และให้ x เท่ากับ 0 ต่อมาส่ง x ออกทาง P1 และตรวจสอบว่า x น้อยกว่าหรือเท่ากับ 255 จริงหรือไม่ ถ้าจริงให้เพิ่มค่า x ขึ้นอีกหนึ่ง (x++) และทำซ้ำเรื่อยๆ

2) ประโยค while

การทำซ้ำแบบนี้จะตรวจสอบเงื่อนไขก่อนการทำซ้ำ ถ้าเงื่อนไขก่อนการทำซ้ำ ถ้าเงื่อนไขเป็นจริงจะทำสแตตเมนต์ที่กำหนด และทดสอบเงื่อนไขใหม่ ถ้าเงื่อนไขเป็นเท็จจะออกจากการทำซ้ำทันที โดยมีรูปแบบดังนี้



รูปที่ 2.34 ประโยคตรวจสอบเงื่อนไขก่อนการทำซ้ำ

ในส่วนของ Expression นั้นสามารถตรวจสอบค่าคงที่ได้ด้วย ถ้าค่าคงที่ได้ด้วย ถ้าค่าไม่เท่ากับศูนย์ จะทำซ้ำ ถ้าค่าเท่ากับศูนย์จะไม่ทำซ้ำ ตัวอย่างเช่น ถ้าหากต้องการให้ค่าลอจิกทางพอร์ต P1 ของ MCS - 51 ทุกบิตมีค่าลอจิกกลับไปมาจะเขียนคำสั่งได้ดังนี้

```

while (1)
{
    P1 = 0x55;
    P1 = 0x0AA;
}
  
```

3) ประโยค do - while

การทำซ้ำประเภทนี้จะตรวจสอบเงื่อนไขภายหลังการทำสแตตเมนต์แต่ละครั้ง ถ้าหากเงื่อนไขเป็นเท็จจะออกจากการทำซ้ำทันทีโดยมีรูปแบบดังนี้

```

void delay (unsigned int x)
{
    unsigned char j;
    while (x--)
    {
        for (j = 0; j < 125; j ++);
    }
}
  
```

จากชุดคำสั่งด้านบนจะสร้างฟังก์ชันชื่อ delay ขึ้นมาและมีการประกาศตัวแปร j สำหรับใช้ภายในฟังก์ชันและจะผ่านค่าเข้าไปในฟังก์ชันผ่านทางตัวแปร x ต่อมาเมื่อพบกับประโยค while จะเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลดค่า x ลงหนึ่งค่า ถ้าหากค่า x ยังไม่เป็นศูนย์จะทำสแตตเมนต์ที่ตามมาโดยวนลูป for จำนวน 125 ครั้ง และกลับไปลดค่า x ในประโยค while ใหม่จนกว่าค่า x จะเป็นศูนย์

2.6.7 อาร์เรย์ พอยน์เตอร์ และสตรักเจอร์

ในการเขียนโปรแกรมด้วยภาษาซีถ้าหากต้องการใช้งานตัวแปรหลายตัวเราสามารถประกาศชื่อตัวแปรออกมาหลายตัวได้ เช่น x1, x2, x3, ..., xn แต่ถ้าหากตัวแปรทุกตัวใช้เก็บข้อมูลประเภทเดียวกันเราสามารถประกาศเป็นตัวแปรแบบอาร์เรย์ (Array) ได้ การประกาศตัวแปรแบบอาร์เรย์สามารถทำได้ดังรูปแบบต่อไปนี้

ประเภทของข้อมูล ชื่ออาร์เรย์ [ขนาดอาร์เรย์];

ถ้าเป็นอาร์เรย์แบบสองมิติสามารถประกาศได้ดังนี้

ประเภทของข้อมูล ชื่ออาร์เรย์ [ขนาดอาร์เรย์] [ขนาดอาร์เรย์];

ตัวอย่างเช่น

```
char x[8];
```

เป็นการประกาศตัวแปรชื่อ x จำนวน 8 เซลล์ แต่ละเซลล์จะเก็บข้อมูลประเภทตัวอักษร การอ้างถึงอาร์เรย์ x แต่ละเซลล์จะใช้อินเด็กซ์เป็นตัวอ้าง เช่น x[0] เป็นการอ้างถึงเซลล์แรก นอกจากนี้การประกาศตัวแปรอาร์เรย์สามารถทำกำหนดค่าข้อมูลเข้าไปในตัวแปรอาร์เรย์ได้เลย ตัวอย่างเช่น

```
unsigned char ab[] = {0xa, 0x9, 0x5, 0x6};
```

จะเห็นว่าในการประกาศตัวแปรอาร์เรย์ ab จะไม่ระบุขนาดของอาร์เรย์ ระบบจะจองหน่วยความจำเท่ากับค่าที่กำหนด การประกาศแบบนี้เซลล์แรก ab[0] จะเก็บค่า 0A ตามลำดับหกขนาดหนึ่งไบต์ได้

ในระบบไมโครคอนโทรลเลอร์ เมื่อมีการประกาศตัวแปรอาร์เรย์ ระบบจะจองหน่วยความจำสำหรับเก็บตัวแปรอาร์เรย์นั้น ตัวแปรประเภทนี้สามารถนำไปประยุกต์ใช้งานในการเขียนโปรแกรมแบบเปิดตารางได้

ในการอ้างถึงข้อมูลแต่ละเซลล์ในอาร์เรย์สามารถใช้ตัวแปรพอยน์เตอร์ชี้ไปที่ตำแหน่งของอาร์เรย์ได้โดยตรง โดยพอยน์เตอร์จะเป็นตัวแปรที่ใช้เก็บแอดเดรสหรือตำแหน่งหน่วยความจำ ตัวดำเนินการที่ใช้กับพอยน์เตอร์คือ & และ * ตัวอย่างเช่น ถ้ามีการประกาศตัวแปรเป็น

```
char j;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเป็นการประกาศหน่วยความจำชื่อ j สำหรับเก็บตัวอักษร เราสามารถอ้างแอดเดรสของตัวแปร j ได้ดังนี้

```
*j;
```

ถ้าหากมีการประกาศตัวแปรพอยน์เตอร์สำหรับเก็บแอดเดรสของตัวแปร j จะทำได้ดังนี้

```
char *dptr;
dptr = &j;
```

เป็นการประกาศตัวแปรพอยน์เตอร์ชื่อ $dptr$ และให้ชี้ไปที่แอดเดรสของตัวแปร j ถ้าหากมีการประกาศตัวแปรเป็น

```
int ax[20];
```

หมายความว่าอาร์เรย์ชื่อ ax จะมีทั้งหมด 20 เซล แต่ละเซลล์จะเก็บเลขจำนวนเต็ม การอ้างถึงข้อมูลแต่ละเซลล์จะเขียนเป็น $ax[i]$ โดยที่ i เป็นค่าอินเด็กซ์ และถ้าประกาศตัวแปร ip ให้เป็นตัวแปรพอยน์เตอร์ชี้ไปที่อาร์เรย์ของเลขจำนวนเต็มสามารถทำได้ดังต่อไปนี้

```
int *ip
ip = &ax[0];
```

จำทำให้ตัวแปร ip ชี้ไปที่ตำแหน่งของเซลล์ $ax[0]$ ถ้าหากต้องการนำข้อมูลที่อยู่ในตัวแปร $ax[0]$ มาใส่ในตัวแปร x สามารถทำได้ดังนี้

```
x = *ip;
```

และถ้า ip ชี้ไปที่ $ax[0]$ ถ้าหากมีการอ้างเป็น $ip + 1$ จะเป็นการชี้ไปที่ $ax[1]$ ดังนั้นถ้าหากมีการอ้างเป็น $ip + i$ จะเป็นการอ้างไปที่อาร์เรย์ $ax[i]$ ได้

ในระบบไมโครคอนโทรลเลอร์บางครั้งจะต้องมีการออกแบบหน่วยความจำ ROM และ RAM ต่ออยู่ภายนอกตัวไมโครคอนโทรลเลอร์ และอุปกรณ์ต่างๆ ที่ต่อกับไมโครคอนโทรลเลอร์จะมีตำแหน่งหรือแอดเดรสที่แน่นอน สามารถใช้พอยน์เตอร์ชี้ไปที่ตำแหน่งความจำภายนอกได้ ตัวอย่างเช่น

```
char *abs_ptr = 0x8000;
```

เป็นการประกาศตัวแปรพอยน์เตอร์ชื่อ abs_ptr ให้ชี้ไปที่ตำแหน่ง 8000H

เอกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่ผ่านมาจะพบว่าตัวแปรประเภทอาร์เรย์นั้นอาจมองว่าเป็นกลุ่มของข้อมูลได้ โดยข้อมูลในกลุ่มนั้นจะเป็นข้อมูลประเภทเดียวกัน ถ้าหากต้องการประกาศตัวแปรเป็นกลุ่มของข้อมูลในกลุ่มนั้นเป็นชนิดต่างกัน จะต้องประกาศตัวแปร เป็นแบบโครงสร้าง หรือสตัคเจอร์ (Structure) ซึ่งมีรูปแบบดังนี้

```
struct {
    ประเภทของข้อมูล    ชื่อตัวแปร;
    .....
    .....
}ชื่อโครงสร้าง;
```

โดยกลุ่มของข้อมูลที่ประกาศขึ้นนั้นจะอยู่ในเครื่องหมายปีกกา และเราสามารถอ้างไปที่ข้อมูลตัวใดๆ ได้ โดยใช้เครื่องหมาย (.) ตัวอย่างเช่น

```
struct {
    unsigned long s;
    unsigned int t;
    unsigned char done;
}state;
```

จะเป็นการประกาศตัวแปร โครงสร้างชื่อ state ซึ่งจะใช้หน่วยความจำทั้งหมด 7 ไบต์ (long 4 + int 2 + char 1) ถ้าหากต้องการใส่ข้อมูลค่า 321 ให้กับตัวแปร t ในโครงสร้างจะทำได้ดังนี้

```
state.t = 321;
```

ถ้าหากมีการประกาศตัวแปรดังต่อไปนี้จะทำให้ผลลัพธ์มีค่าเหมือนกับตัวแปรแบบโครงสร้างในตัวอย่างที่ผ่านมา

```
#define uchar unsigned char
#define uint unsigned    ←   ประกาศตัวแปรใหม่
struct stateform{
    unsigned long s;
    uint t;        ←   เรียกใช้ในโครงสร้าง
    uchar done;
};
```

จะเห็นว่าจะใช้ #define ประกาศตัวแปรประเภทของข้อมูลใหม่ขึ้นมา และถูกเรียกใช้ในตัวแปรโครงสร้าง ถ้าหากต้องการให้ตัวแปรโครงสร้างนี้มีชื่อว่า state จะทำได้ดังนี้

```

struct stateform      state;
  ↑                   ↑
ข้อมูลแบบโครงสร้าง  ชื่อตัวแปร

```

ถ้าหากต้องการประกาศตัวแปรแบบโครงสร้างหลายตัว สามารถทำได้ในรูปของอาร์เรย์ของโครงสร้าง (Array of Structures) ดังตัวอย่างต่อไปนี้

```

#define uchar unsigned char
#define uint unsigned
struct stateform {
    unsigned long s;
    uint done;
};
struct stateform state[20]

```

จะเป็นการประกาศตัวแปรอาร์เรย์ชื่อ state ที่มีทั้งหมด 20 เซลล์ แต่ละเซลล์จะเป็นตัวแปรแบบโครงสร้างขนาด 7 ไบต์ ทำให้ตัวแปรนี้ใช้หน่วยความจำทั้งหมด 140 ไบต์ (20×7)

2.6.8 การเขียนโปรแกรมจัดการหน่วยความจำ

การจัดการหน่วยความจำของไมโครคอนโทรลเลอร์ MCS – 51 จะใช้การจัดการหน่วยความจำแบบ Harvard โดยแยกหน่วยความจำออกเป็นส่วนของหน่วยความจำโปรแกรมและหน่วยความจำข้อมูล สำหรับหน่วยความจำข้อมูลนั้นยังถูกแบ่งออกเป็นหน่วยความจำภายในตัว MCS – 51 เอง และหน่วยความจำภายนอก ในการขยายพอร์ตเพิ่มเพื่อใช้กับตัว MCS – 51 ตำแหน่งของพอร์ตจะเป็นตำแหน่งของหน่วยความจำภายนอกนี้ด้วย

สำหรับหน่วยความจำภายในของ MCS – 51 ยังถูกแบ่งออกเป็น หน่วยความจำ 128 ไบต์แรก (ตำแหน่ง 00 -7FH) ซึ่งสามารถเข้าถึงข้อมูล ได้แบบ Direct และ Indirect ส่วนตำแหน่ง 80H ถึง FFH จะเป็นรีจิสเตอร์ฟังก์ชันพิเศษ หรือ SFR (Special Function Register) ซึ่งจะใช้การเข้าถึงข้อมูลแบบ Direct รีจิสเตอร์ฟังก์ชันพิเศษ ซึ่งจะใช้การเข้าถึงข้อมูลแบบ Direct

ในการใช้ภาษาซีกับ MCS – 51 จะต้องมีการจัดวางหน่วยความจำให้ตัวแปรต่างๆ ที่ถูกประกาศขึ้นนั้นสามารถอยู่ในหน่วยความจำที่เหมาะสมได้ โดยการจัดการหน่วยความจำของตัวคอมไพเลอร์มีรูปแบบดังนี้

TINY เป็นการจัดการหน่วยความจำให้ตัวคอมไพเลอร์มองว่าไม่มีหน่วยความจำแรมภายนอก ซึ่งจะใช้กับ MCS – 51 แบบที่เป็นซิงเกิลชิป เช่น เบอร์ 89C8051

SMALL หน่วยความจำแบบนี้ออมไพเลอร์จะมองว่าแรมและรอมอยู่ซ้อนกันภายในพื้นที่ 64 กิโลไบต์ ตัวแปรต่างๆ ที่ถูกประกาศใช้งาน หรือประกาศเป็นแบบโลเคิลจะถูกกำหนดอยู่ในหน่วยความจำข้อมูลภายใน (Internal Data Memory) ขนาดของหน่วยความจำแรมจะถูกกำหนดตามที่ใช้งานจริง รูปแบบหน่วยความจำใหม่นี้จะเป็นค่าที่โปรแกรมคอมไพเลอร์ได้กำหนดไว้แล้ว

COMPACT หน่วยความจำแบบนี้จะมีลักษณะเดียวกับหน่วยความจำแบบ SMALL แต่ตัวแปรที่ถูกประกาศใช้งาน หรือประกาศแบบโลเคิลจะถูกกำหนดอยู่ในหน่วยความจำภายนอก ทำให้ตัวแปรสามารถมีได้มากกว่าแบบ SMALL แต่การทำงานของโปรแกรมที่ต้องประมวลผลกับตัวแปรจะทำงานได้ช้ากว่าแบบ SMALL เพราะแบบ SMALL ตัวแปรจะอยู่ในแรมภายในชิป

LARGE หน่วยความจำแบบนี้จะมีขนาดใหญ่ ตัวแปรทั้งหมดจะถูกกำหนดอยู่ในหน่วยความจำข้อมูลภายนอก

การเขียนโปรแกรมด้วยภาษาซีให้กับไมโครคอนโทรลเลอร์ด้วย C51 นี้สามารถประกาศตัวแปรต่างๆ แบบเจาะจงได้ โดยใช้เมโครที่กำหนดไว้ในไฟล์ `absacc.h` ดังต่อไปนี้

CODE	แทนพื้นที่หน่วยความจำโปรแกรมภายนอก
DATA	แทนพื้นที่หน่วยความจำข้อมูลภายใน 128 ไบต์แรก (ตำแหน่ง 00H ถึง 7FH)
IDATA	แทนเนื้อหาที่หน่วยความจำข้อมูลภายใน 256 ไบต์
BDATA	แทนตำแหน่งของหน่วยความจำระดับบิต 128 บิต (20H ถึง 2FH)-ของหน่วยความจำภายใน
XDATA	แทนตำแหน่งหน่วยความจำภายนอก
PDATA	แทนตำแหน่งหน่วยความจำภายนอก 256 ไบต์แรก

ในการกำหนดชื่อตัวแปรต่างๆ เพื่อแทนตำแหน่งหน่วยความจำนั้นมีรูปแบบดังนี้

ชนิดของตัวแปร	ชื่อเมโคร	ตัวแปร
---------------	-----------	--------

ตัวอย่างเช่น

```
char CODE text [ ] = "COMPUTER"
unsigned char XDATA AB[100]
unsigned int IDATA x, y, z
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดบรรทัดแรกจะทำให้คำว่า COMPUTER ถูกเก็บอยู่ในหน่วยความจำโปรแกรม ภายนอกบรรทัดที่สองจะทำให้อาร์เรย์ AB ถูกประกาศเอาไว้ในหน่วยความจำข้อมูลภายนอก บรรทัดที่สามจะทำให้ตัวแปร x, y และ z ถูกประกาศเอาไว้ในหน่วยความจำข้อมูลภายใน

ในการเขียน โปรแกรมให้ไมโครคอนโทรลเลอร์ติดต่อกับพอร์ตจะต้องทำการโอนย้ายข้อมูลกับพอร์ตตำแหน่งแอดเดรสต่างๆ ซึ่งการอ้างแอดเดรสของพอร์ตสามารถใช้เมโครได้เช่นกัน ทำให้สามารถแทนพอร์ตด้วยตัวแปรต่างๆ ได้โดยเมโครที่ใช้จะเก็บอยู่ในไฟล์ absacc.h โดยมีชื่อดังต่อไปนี้

CBYTE , CWORD	แทนไบต์หรือเวิร์ดของหน่วยความจำโปรแกรม
DBYTE , DWORD	แทนไบต์หรือเวิร์ดของหน่วยความจำข้อมูลภายใน
PBYTE , PWORD	แทนไบต์หรือเวิร์ดของหน่วยความจำภายนอก 256 ตำแหน่งแรก
XBYTE , XWORD	แทนไบต์หรือเวิร์ดของหน่วยความจำภายนอกทั้งหมด

ในการกำหนดตัวแปรให้แทนตำแหน่งต่างๆ จะต้องใช้ร่วมกับไคเร็กทีฟ #define ดังตัวอย่างต่อไปนี้

```
#define port8 XBYTE [0x8000
```

จะเป็นการแทนตัวแปร port8 ด้วยหน่วยความจำภายนอกตำแหน่ง 8000H ถ้าหากต้องการส่งค่า FF ไปยังหน่วยความจำตำแหน่ง 8000H สามารถทำได้ดังนี้

```
port8 = 0xFF;
```

ถ้าหากต้องการอ่านข้อมูลจากตำแหน่ง 8000H มาเก็บในตัวแปร x สามารถทำได้ดังนี้

```
char x;  
x = port8;
```

การเข้าถึงข้อมูลระดับบิตและรีจิสเตอร์พิเศษ

ในโปรแกรม C51 ถ้ามีการประกาศตัวแปรและกำหนดให้เป็นหน่วยความจำแบบ BDATA จะเป็นการใช้งานในแอดเดรสหน่วยความจำที่สามารถเข้าถึงข้อมูลในระดับบิตได้ นอกจากนี้ยังสามารถใช้คำสั่ง C51 กำหนดตัวแปรแบบบิตขึ้นมาได้อีกด้วย ตัวอย่างเช่น

```
int BDATA ibase; /* Bit - addressable int */
char BDATA bary[4]; /* Bit - addressable array */
sbit mybit0 = ibase^0; /* Bit 0 of ibase */
sbit mybit15 = ibase^15; /* Bit 15 of ibase */
sbit Ary07 = bary[0]^7 /* Bit 7 of bary[0] */
```

บรรทัดแรกจะเป็นการประกาศตัวแปรชื่อ `ibase` สำหรับเก็บจำนวนเต็มลงในหน่วยความจำพื้นที่ที่สามารถเข้าถึงข้อมูลในระดับบิตได้ บรรทัดต่อมาประกาศตัวแปรอาร์เรย์จำนวน 4 ไบต์ลงในหน่วยความจำระดับบิต ในบรรทัดที่สามจะใช้คำสั่ง `sbit` กำหนดตัวแปรขึ้นมาใหม่ แทนบิตต่ำสุด โดยใช้สัญลักษณ์ `carat` ('^') ซึ่งจำทำให้ตัวแปรชื่อ `mybit0` แทนบิตที่ 0 ของหน่วยความจำชื่อ `ibase` บรรทัดที่สี่จะเป็นตัวแปรชื่อ `mybit15` แทนบิตสูงสุดของตัวแปร `ibase` และบรรทัดสุดท้ายจะให้ตัวแปรชื่อ `Ary07` แทนบิตที่ 7 ของ `bary[0]` สำหรับขนาดของบิตของตัวแปร จะขึ้นกับประเภทของข้อมูลที่กำหนด เช่น ถ้าเป็นข้อมูลแบบ `char` และ `unsigned char` ข้อมูลบิตจะอยู่ในช่วง 0 ถึง 7 บิต ถ้าเป็นแบบ `int`, `unsigned int`, `short` และ `unsigned short` จะอยู่ในช่วง 0 ถึง 15 บิต

ถ้าต้องการกำหนดตัวแปรชื่อ `OUT` แทนบิตสูงสุดของพอร์ต P1 ของ MCS-51 สามารถทำได้ดังนี้

```
sbit OUT = P1^7;
```

พื้นที่หน่วยความจำภายในของ MCS-51 ส่วนหนึ่งจะเป็นพื้นที่ของรีจิสเตอร์ฟังก์ชันพิเศษ (SFR) เช่น รีจิสเตอร์ควบคุมไทมเมอร์ รีจิสเตอร์ตัวนับ พอร์ตอนุกรม พอร์ตอินพุตเอาต์พุตต่างๆ ซึ่งอยู่ในหน่วยความจำภายในตั้งแต่ตำแหน่ง `0x80` ถึง `0xFF` ถ้าหากต้องการกำหนดชื่อใหม่แทนสัญลักษณ์เหล่านั้นสามารถใช้คำสั่ง `sfr` ในการประกาศได้ โดยใช้เครื่องหมายเท่ากับ (=) ระบุค่าแอดเดรสให้กับตัวแปรนั้น เช่น

```
sfr P0 = 0x80; /* Port_0 address 80H */
sfr P1 = 0x90; /* Port_1 address 90H */
sfr P2 = 0xA0; /* Port_2 address A0H */
sfr P3 = 0xB0; /* Port_3 address B0H */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ถ้าหากต้องการกำหนดชื่อแทนรีจิสเตอร์ฟังก์ชันพิเศษที่มีขนาด 16 บิต จะต้องใช้คำสั่ง sfr 16 และระบุแอดเดรสไบต์ต่ำให้กับตัวแปรนั้น ตัวอย่างเช่น

```
sfr16 T2 = 0xcc /* Timer 2 : T2L 0CCH , T2H 0CDH */
```

เป็นการให้ T2 แทนไทมเมอร์ 2 ซึ่งรีจิสเตอร์นี้จะมีขนาด 2 ไบต์ ไบต์ต่ำหรือ T2L จะอยู่ในแอดเดรส 0CCH และไบต์สูง T2H จะอยู่ในแอดเดรส 0CDH

สำหรับการเขียนโปรแกรมด้วย C51 นั้นจะเห็นว่ารีจิสเตอร์บางตัว ตำแหน่งบิตบางตำแหน่งเราสามารถอ้างอิงชื่อขึ้นมาได้เลย เนื่องจากในไฟล์ res51.h ที่ได้ include ขึ้นมาเป็นไฟล์ที่กำหนดรีจิสเตอร์ต่างเอาไว้แล้ว เช่น P0, P1, P2 และ P3 สำหรับตำแหน่งบิตของรีจิสเตอร์ที่สามารถเข้าถึงข้อมูลระดับบิตได้ เช่น TF1, TR1, TF0, TR0, IE1, IT1, IE0 และ IT0 ซึ่งเป็นบิตที่อยู่ในรีจิสเตอร์ TCON ก็กำหนดไว้แล้วเช่นกัน โดยในไฟล์ reg51.h ส่วนหนึ่งได้กำหนดไว้ดังนี้

```
/* BYTE register */
sfr P0 = 0x80;
sfr P1 = 0x90;
sfr P2 = 0xA0;
sfr P3 = 0xB0;
/* BIT register */
/* TCON */
sbit TF1 = 0x8F;
sbit TR1 = 0x8E;
sbit TF0 = 0x8D;
sbit TR0 = 0x8C;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบ การสร้าง และการทำงาน

3.1 กล่าวนำ

การออกแบบและการสร้างระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วนประกอบไปด้วย 3 ส่วนคือ ชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ ชุดแปลงสัญญาณ RS232/RS485 และภาคแสดงผลบนเครื่องคอมพิวเตอร์ วงจรของชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศประกอบด้วย วงจรควบคุม วงจรตรวจจับอุณหภูมิ วงจรดิจิตอลมิเตอร์ วงจรวัดแรงดันไฟฟ้า วงจรวัดกระแสไฟฟ้า วงจรเปิด-ปิดคอมเพรสเซอร์ และวงจรแหล่งจ่ายไฟ ส่วนของภาคแสดงผลบนเครื่องคอมพิวเตอร์ใช้โปรแกรมวิซวลเบสิกในการสร้าง

3.2 การออกแบบวงจร

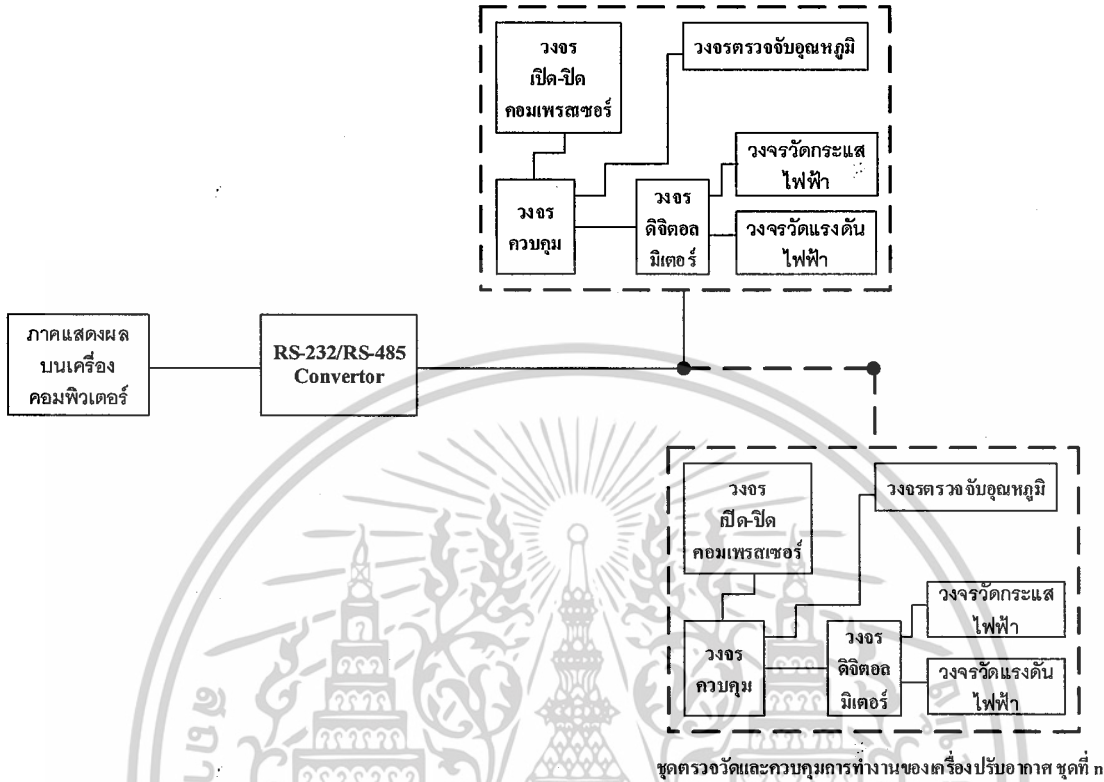
วิธีการออกแบบวงจรที่ใช้งานในระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วนออกแบบโดยการกำหนดขั้นตอนการทำงานในแผนผังการทำงาน ดังรูปที่ 3.1

การออกแบบวงจรแบ่งออกเป็นส่วนใหญ่ๆ ได้ดังรูปที่ 3.1 (ก) ระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วนนี้ใช้มาตรฐานการเชื่อมต่อแบบอนุกรม RS-485 เชื่อมต่อกับชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศและทำการแปลงสัญญาณมาเป็น RS-232 เพื่อติดต่อสื่อสารกับข้อมูลกับคอมพิวเตอร์และแสดงผลบนเครื่องคอมพิวเตอร์

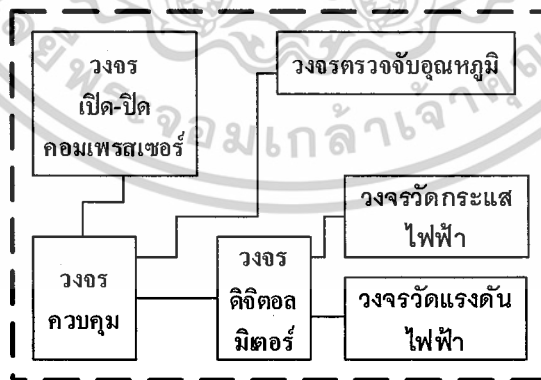
ในรูปที่ 3.1 (ข) เป็นชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ ภายในชุดนี้จะประกอบด้วยวงจรต่างๆ ดังนี้ วงจรควบคุม วงจรเปิด-ปิดคอมเพรสเซอร์ วงจรตรวจจับอุณหภูมิ วงจรดิจิตอลมิเตอร์ วงจรวัดแรงดันไฟฟ้า และวงจรวัดกระแสไฟฟ้า โดยมีวงจรนี้จะทำหน้าที่ตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ แล้วนำค่าที่ได้นั้นส่งไปยังชุดแปลงสัญญาณ RS-485 เป็น RS-232 เพื่อทำการติดต่อและแสดงผลบนหน้าจอคอมพิวเตอร์ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ ชุดที่ 1



รูปที่ 3.1 (ก) แผนผังการทำงานของระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วน



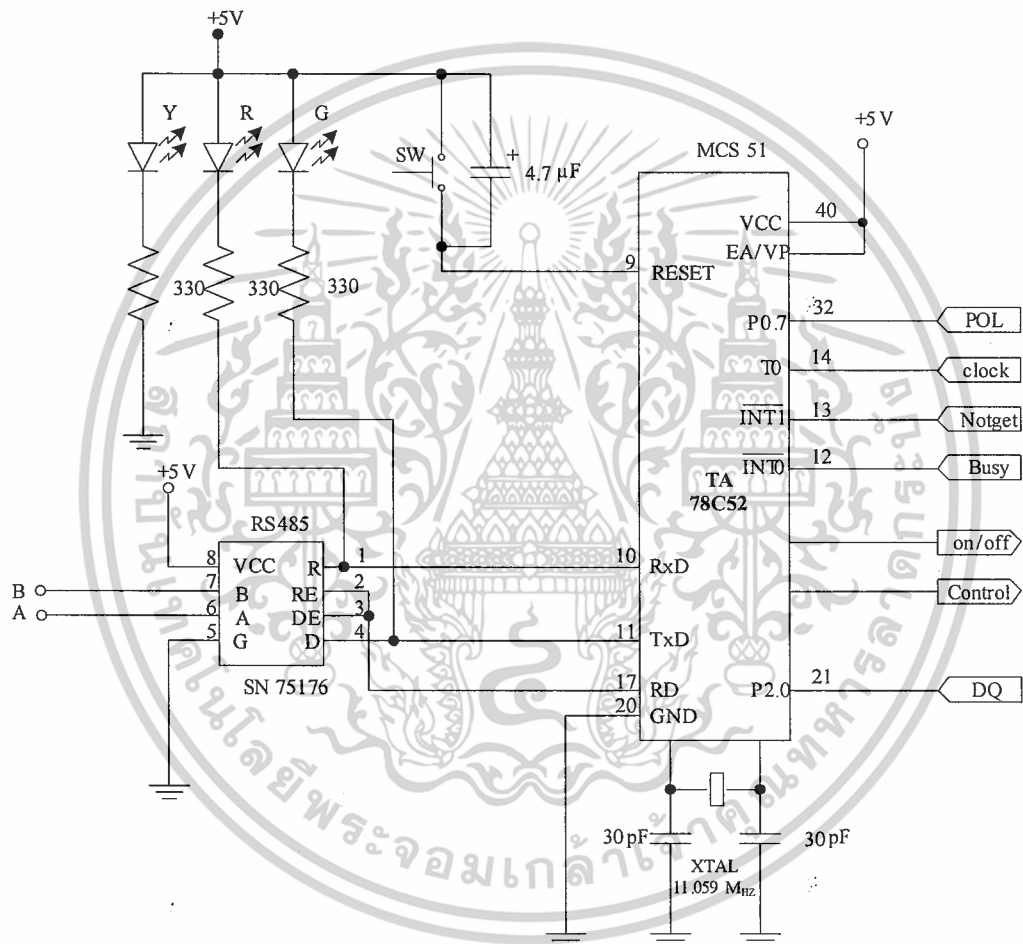
รูปที่ 3.1 (ข) ชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ

3.3.1 วงจรควบคุม

วงจรควบคุมใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เป็นตัวควบคุมหลักในการประมวลผลค่าข้อมูล โดยมีวงจรรวม (IC) ตรวจวัดอุณหภูมิเบอร์ DS1820 ทำการตรวจวัดอุณหภูมิแล้วส่งค่าที่ตรวจวัดได้มายังไมโครคอนโทรลเลอร์เพื่อทำการประมวลผล



รูปที่ 3.2 วงจรควบคุม

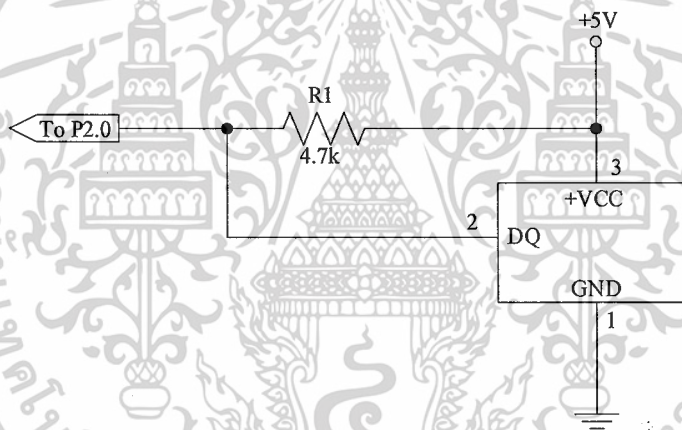
การทำงานของวงจร เมื่อ DS1820 ส่งข้อมูลมายังขาที่ 21 ของไมโครคอนโทรลเลอร์ ไมโครคอนโทรลเลอร์ก็จะทำการประมวลผล และส่งข้อมูลที่ประมวลผลได้มาแสดงผลที่หน้าจอคอมพิวเตอร์ ในขณะเดียวกันก็ทำหน้าที่รับส่งข้อมูลซึ่งเป็นค่าข้อมูลต่างๆ ส่งผ่านพอร์ทอนุกรม RS-485 โดยไอซีที่ใช้ในการรับส่งผ่านพอร์ทอนุกรม RS-485 คือไอซีเบอร์ SN75176 เชื่อมต่อกับขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Tx, Rx, และ RD ของไมโครคอนโทรลเลอร์ ข้อมูลที่ส่งมาจะถูกแปลงเพื่อส่งผ่านมาตรฐานอนุกรมแบบ RS-232 ด้วยวงจรแปลง RS-232/Rs-485

3.3.2 วงจรตรวจจับอุณหภูมิ

วงจรตรวจจับอุณหภูมิ DS1820 ทำหน้าที่เป็นอุปกรณ์ตรวจวัดอุณหภูมิ การติดต่อระหว่างวงจรตรวจวัดอุณหภูมิกับวงจรประมวลผลค่าอุณหภูมิ และรับ-ส่งข้อมูลผ่าน RS-485 การเชื่อมต่อวงจรรวม DS1820 กับไมโครคอนโทรลเลอร์เป็นการเชื่อมต่อข้อมูลแบบสายสัญญาณเพียงเส้นเดียว วงจรรวมหมายเลข DS1820 มีขาต่อใช้งานเพียง 3 ขา คือ ขา DQ ซึ่งเป็นขาที่เชื่อมต่อเพื่อส่งข้อมูลในระบบบัส ขาไฟเลี้ยงและขาราวด์ เนื่องจากสายสัญญาณของระบบบัสนี้ต้องทำการกำหนดสถานะปกติไว้ที่ลอจิกสูง ดังนั้นในการเชื่อมต่อใช้งานจึงต้องทำการต่อตัวความต้านทานค่า 4.7 กิโลโอห์ม พูลอัพกับไฟเลี้ยงในวงจรด้วย รูปวงจรตรวจจับอุณหภูมิ แสดงดังรูปที่ 3.3



รูปที่ 3.3 วงจรตรวจจับอุณหภูมิ

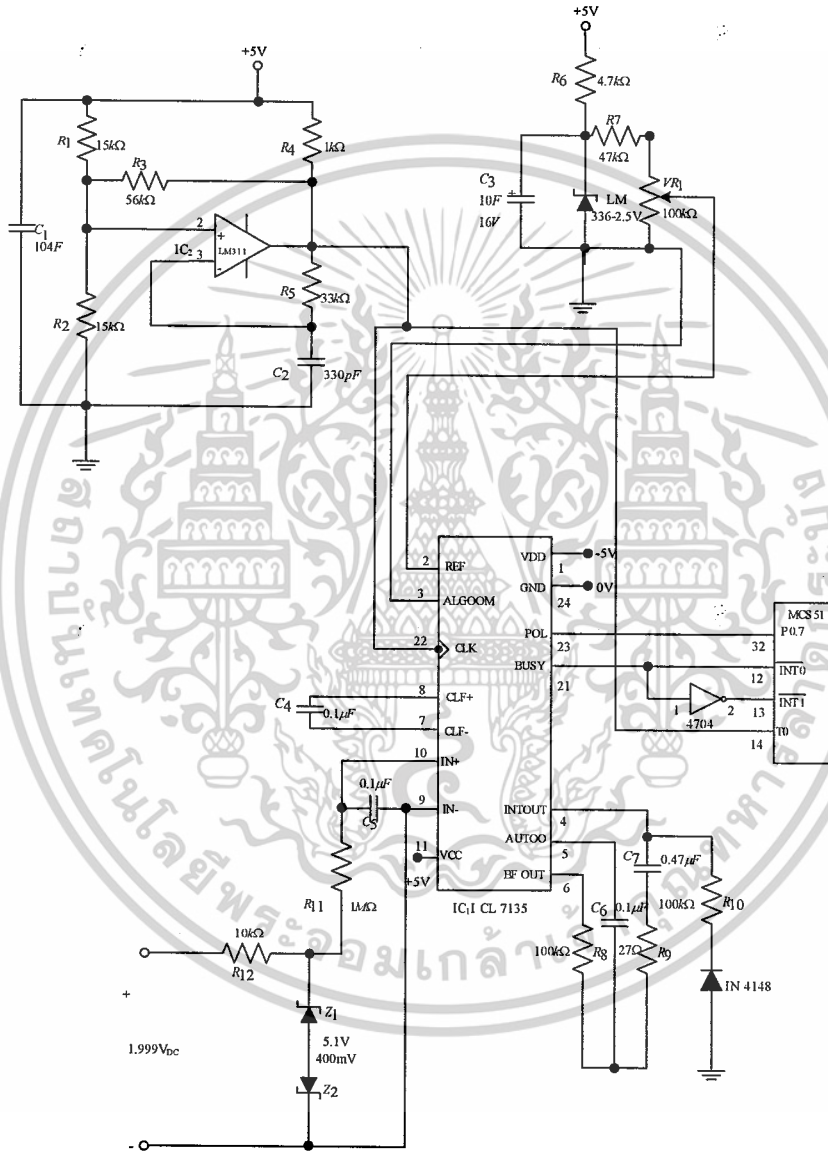
3.3.3 วงจรดิจิตอลมิเตอร์

วงจรถิจิตอลมิเตอร์เป็นส่วนประกอบที่สำคัญวงจรหนึ่ง ซึ่งจะประกอบด้วยไอซี 2 ตัว IC2 เบอร์ LM 311 จะทำหน้าที่ผลิตสัญญาณนาฬิกาที่มีความถี่เท่ากับ 165 กิโลเฮิร์ตซ์ ส่งให้กับขา 22 ของ IC1 เบอร์ ICL7135 (ซึ่งเป็นขา Clock In) แล้วก็ยังส่งไปวงจรของ MCS-51 เข้าขา 14 (TO) เพื่อนำไปเปรียบเทียบกับสัญญาณอื่นๆ

เมื่อมีสัญญาณอินพุตเข้ามาขั้วบวกจะผ่าน R11 , R12 เข้าที่ขา 10 ของ IC1 (ซึ่งเป็นขา IN+) และขั้วลบจะเข้าที่ขา 9 ของ IC1 (ซึ่งเป็นขา IN-) หลังจากนั้นก็จะมีสัญญาณ BUSY ออกที่ขา 21 ของ IC1 ไปวงจร MCS-51 เข้าที่ขา 12 และสัญญาณ BUSY อีกส่วนก็จะผลิตสัญญาณไปเข้าที่ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอตเกท เพื่อทำการกลับสถานะแล้วส่งสัญญาณที่ได้ไปยังวงจร MCS-51 เข้าที่ขา 13 และจะมีสัญญาณออกที่ขา 23 ของ IC1 (ซึ่งเป็นขา Pol) ทำหน้าที่บอกขั้วสัญญาณไฟฟ้า ไปยังวงจร MCS-51 ส่งสัญญาณเข้าไปที่ขา 32 (ซึ่งเป็นขา P 0.7) ส่วนในด้านของ VR1 ทำหน้าที่ปรับแรงดันให้กับขา 2 ของ IC1 (ซึ่งเป็นขา REF) ดังจะแสดงไว้ในรูปที่ 3.4

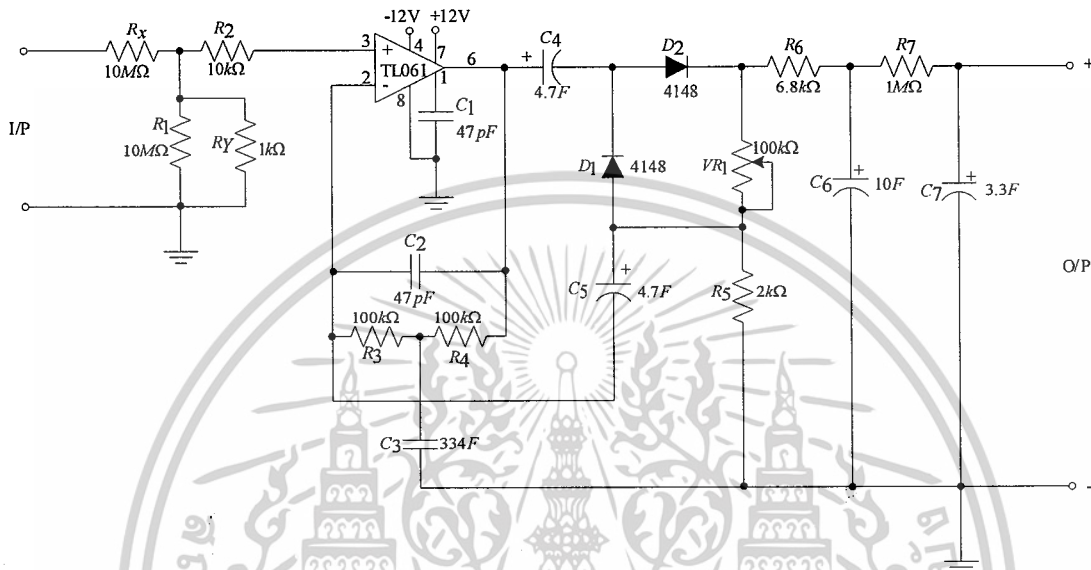


รูปที่ 3.4 วงจรดิจิตอลมิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.4 วงจรวัดแรงดันไฟฟ้า

วงจรวัดแรงดันทำหน้าที่วัดแรงดันไฟฟ้าจากเครื่องปรับอากาศ เพื่อที่จะรับค่าที่ได้นั้นส่งต่อไปยังสายนำสัญญาณเพื่อที่จะแสดงผลออกหน้าจอต่อไป



รูปที่ 3.5 วงจรวัดแรงดันไฟฟ้า

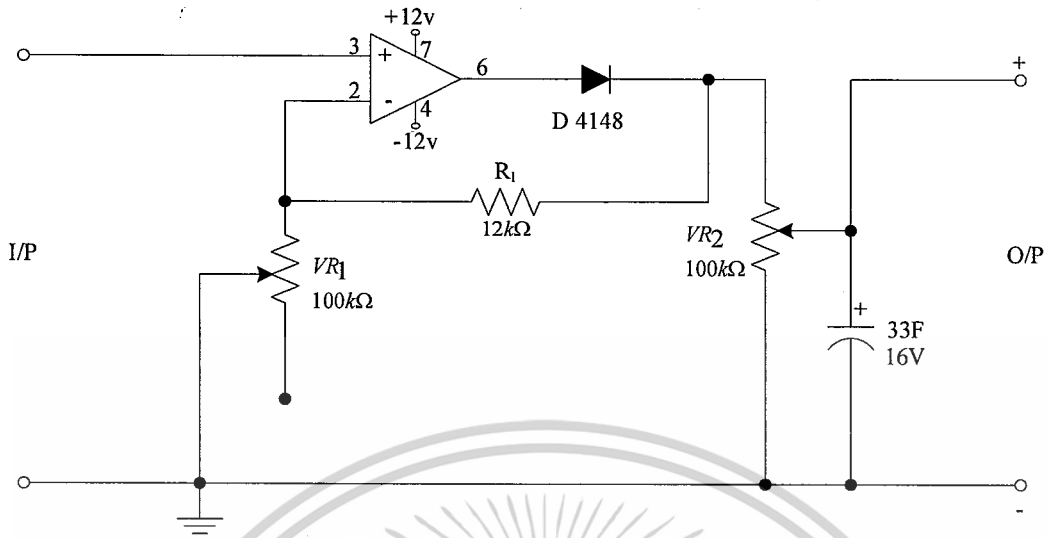
จากรูปที่ 3.5 วงจรวัดแรงดันไฟฟ้ามีลักษณะการทำงานของวงจร คือ สัญญาณจะถูกป้อนให้กลับวงจรแบ่งแรงดันไฟฟ้า แล้วผ่าน R2 ไปยังขา 3 ของไอซีเบอร์ TL061 โดยมี R1 ทำหน้าที่เป็นอินพุตทอมพิวเตอร์ของวงจร ไอซีเบอร์ TL061 จะทำหน้าที่ขยายสัญญาณอินพุตแล้วป้อนให้กับวงจรเรกติไฟเออร์แบบ โวลต์เตจดับเบิล ได้เป็นแรงดันไฟฟ้ากระแสตรงผ่าน R7 ไปเข้าวงจรต่อไป เอาท์พุทที่ได้จะไม่เกิน 2 โวลต์ โดยจะถูกควบคุมจากการปรับค่าของ VR1 ให้เหมาะสม

3.3.5 วงจรวัดกระแสไฟฟ้า

วงจรวัดกระแสไฟฟ้าทำหน้าที่วัดกระแสไฟฟ้าจากเครื่องปรับอากาศ เพื่อที่จะรับค่าที่ได้ นั้นส่งต่อไปยังสายนำสัญญาณเพื่อที่จะแสดงผลออกหน้าจอต่อไป

ในรูปที่ 3.6 วงจรวัดกระแสไฟฟ้ามีลักษณะการทำงานของวงจร คือ สัญญาณอินพุทที่ได้ จาก CT (Current Transformer) เข้าที่ขา 3 ของไอซีออปแอมป์เบอร์ OP07 จะทำหน้าที่ขยายสัญญาณอินพุทแล้วป้อนให้กับวงจรเรกติไฟเออร์แบบ โวลต์เตจซิงเกิ้ล ให้เป็นแรงดันไฟฟ้ากระแสตรง โดยมี C ทำหน้าที่เป็นตัวฟิลเตอร์ และ VR2 ทำหน้าที่ปรับแรงดันไฟฟ้ากระแสตรงในด้าน เอาต์พุทไม่ให้เกิน 2 โวลต์

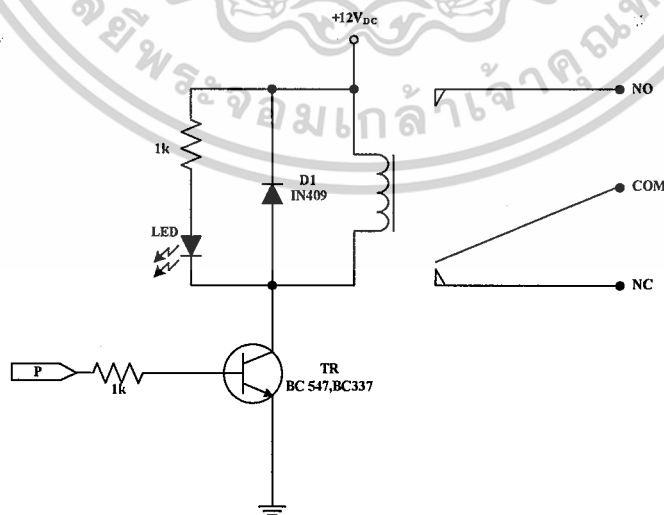
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 วงจรวัดกระแสไฟฟ้า

3.3.6 วงจรเปิด-ปิดคอมเพรสเซอร์

วงจรเปิด-ปิดคอมเพรสเซอร์จะมีทรานซิสเตอร์เบอร์ BC 337 ทำหน้าที่เป็นตัวสวิตช์เปิด-ปิด โดยจะมีสัญญาณจากพอร์ทของ MCS-51 ไบอัสให้กับขา B ของทรานซิสเตอร์ทำให้กระแสไหลผ่านตัวทรานซิสเตอร์ โดยเมื่อจ่ายแรงดันขั้วบวกให้กับขั้วที่ขา 1 เมื่อ MCS-51 สั่งให้ดีเลย์ทำงาน ทรานซิสเตอร์จะถูกวงจร ไบอัสให้กระแสสามารถไหลผ่านขา E ลงกราวด์ได้ และสามารถทำให้ดีเลย์ทำงานได้ โดยแสดงรูปวงจรเปิด-ปิดคอมเพรสเซอร์ไว้ในรูปที่ 3.7

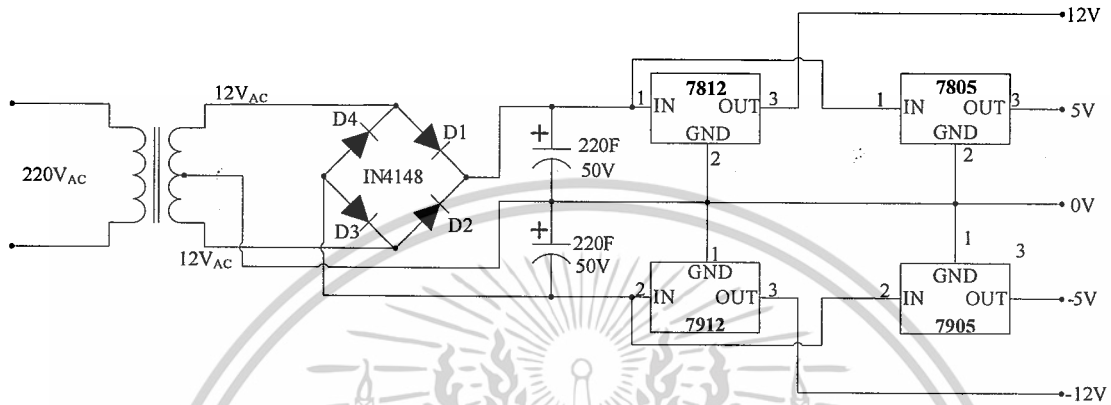


รูปที่ 3.7 วงจรเปิด-ปิดคอมเพรสเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.7 วงจรแหล่งจ่ายไฟ

วงจรแหล่งจ่ายไฟทำหน้าที่จ่ายแรงดันไฟฟ้าให้กับชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ

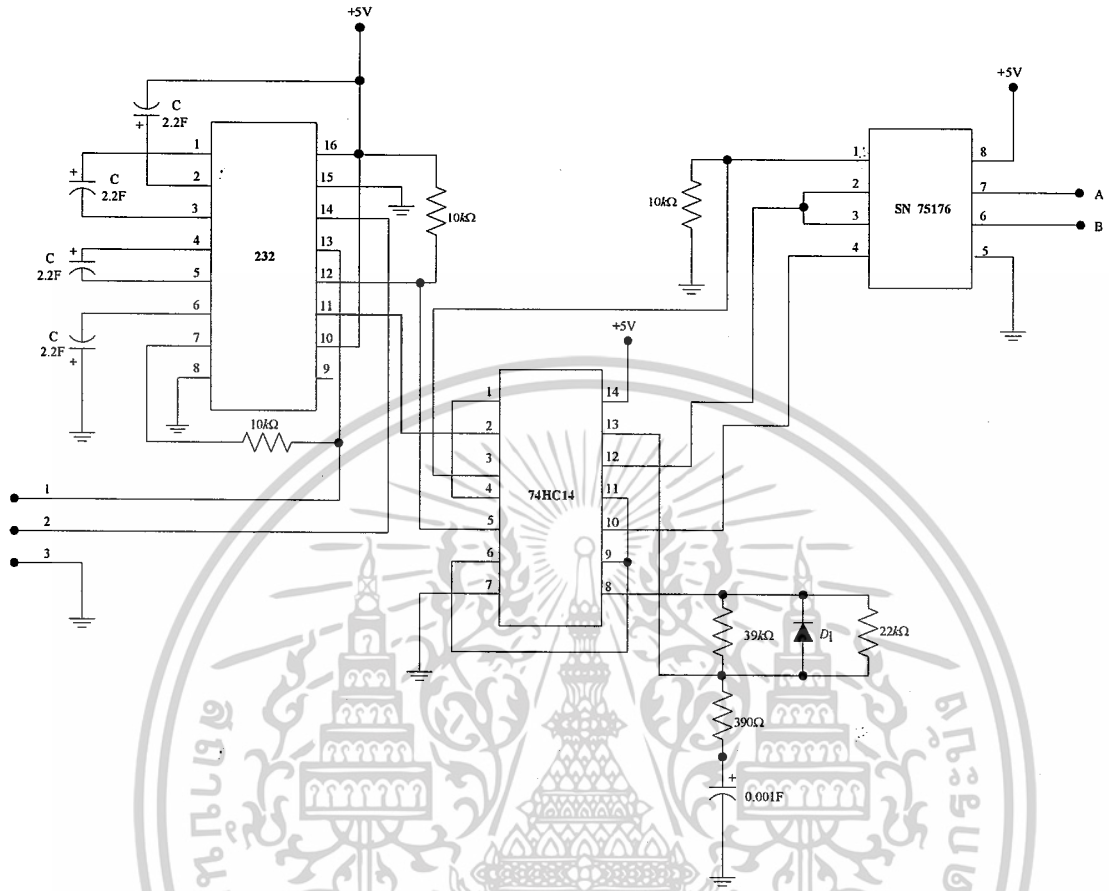


รูปที่ 3.8 วงจรแหล่งจ่ายไฟ

จากรูปที่ 3.8 วงจรแหล่งจ่ายไฟมีลักษณะการทำงานของวงจร คือ ทำหน้าที่แปลงสัญญาณไฟฟ้ากระแสสลับเป็นแรงดันไฟฟ้ากระแสตรง โดยมีวงจรไดโอดบริดจ์เป็นวงจรเรียงกระแส แล้วมีตัวเก็บประจุค่า 2200 ไมโครฟารัดเป็นตัวฟิลเตอร์แล้วใช้ไอซีเบอร์ 7812 เป็นตัวพิกัดแรงดันขนาด 12 โวลต์ ไอซีเบอร์ 7805 เป็นตัวพิกัดแรงดัน 5 โวลต์ ไอซีเบอร์ 7912 เป็นตัวพิกัดแรงดันขนาด -12 โวลต์ และไอซีเบอร์ 7905 เป็นตัวพิกัดขนาด -5 โวลต์

3.4 ชุดแปลงสัญญาณ RS232/RS485

ไอซีเบอร์ 232 ทำหน้าที่ในการส่งผ่านสัญญาณจากพอร์ทอนุกรมของคอมพิวเตอร์ ซึ่งอยู่ในรูปของการสื่อสารแบบมาตรฐานของ RS 232 และไอซีเบอร์ SN 75176 ทำหน้าที่ในการส่งผ่านสัญญาณในรูปของการสื่อสารแบบมาตรฐานของ RS 485 โดยมีไอซีเบอร์ 74HC14 ทำหน้าที่เป็นตัวควบคุมการแปลงสัญญาณของ RS 232 กับ RS 485 เพื่อให้สามารถทำการติดต่อสื่อสารกันได้



รูปที่ 3.9 วงจรแปลงสัญญาณ RS232/RS485

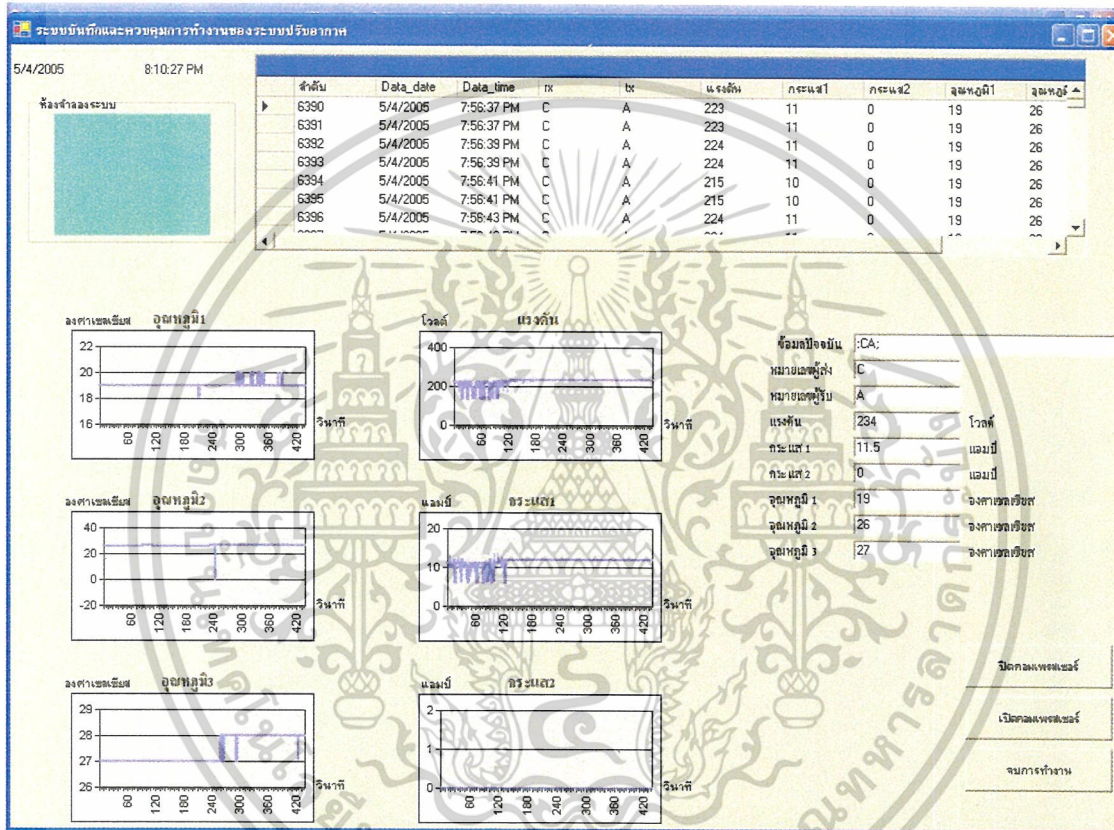
3.5 ภาคแสดงผลบนเครื่องคอมพิวเตอร์

ในการสร้างรูปแบบหน้าจอแสดงผลได้เลือกใช้โปรแกรมวิชวลเบสิกในการสร้าง การออกแบบหน้าจอแสดงผลที่เครื่องคอมพิวเตอร์ของระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วนที่สร้างขึ้นจะเริ่มต้น โดยการออกแบบหน้าจอหลักของโปรแกรม ที่หน้าจอหลักของโปรแกรมจะมีเมนูคำสั่งที่จะเลือกใช้คำสั่งย่อยต่างๆ รูปแบบของหน้าต่างของคำสั่งย่อยนั้นการการออกแบบก็จะแตกต่างกันตามลักษณะการใช้งานของ โปรแกรมย่อยนั้นๆ รายละเอียดของการออกแบบของทุกหน้าต่างจะกล่าวถึงดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.1 หน้าจอแสดงข้อมูลการใช้พลังงาน

หน้าจอแสดงข้อมูลการใช้พลังงานจะทำหน้าที่ในการแสดงผลค่าของแรงดันไฟฟ้า กระแสไฟฟ้า และค่าอุณหภูมิ ซึ่งในการใช้งานสามารถแสดงค่าต่างๆ เหล่านี้เป็นตัวเลขและกราฟได้ โดยสามารถเรียกดูข้อมูลที่เป็นปัจจุบันและย้อนหลังได้อีกด้วย หน้าจอแสดงผลข้อมูลการใช้พลังงานแสดงดังรูปที่ 3.10



รูปที่ 3.10 หน้าจอแสดงผลข้อมูลการใช้พลังงาน

3.5.2 หน้าจอแสดงข้อมูลทั้งหมด

ในการทำงานของโปรแกรมต่างๆ ข้อมูลจะถูกเก็บไว้ในฐานข้อมูล และสามารถที่จะเรียกดูข้อมูลต่างๆ เหล่านี้ได้จากหน้าจอแสดงผล โดยจะแสดงในรูปที่ 3.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับ	Data_date	Data_time	rx	tx	แรงดัน	กระแส1	กระแส2	อุณหภูมิ1	อุณหภูมิ2	อุณหภูมิ3
6205	5/4/2005	7:25:45 PM	C	A	53	26	0	20	27	86
6206	5/4/2005	7:25:45 PM	C	A	53	26	0	20	27	86
6207	5/4/2005	7:25:47 PM	C	A	234	12	0	20	27	26
6208	5/4/2005	7:25:47 PM	C	A	234	12	0	20	27	26
6209	5/4/2005	7:25:53 PM	C	A	235	12	0	20	27	26
6210	5/4/2005	7:25:53 PM	C	A	235	12	0	20	27	26
6211	5/4/2005	7:25:59 PM	C	A	235	12	0	20	27	26
6212	5/4/2005	7:25:59 PM	C	A	235	12	0	20	27	26
6213	5/4/2005	7:26:05 PM	C	A	235	12	0	20	27	26
6214	5/4/2005	7:26:05 PM	C	A	235	12	0	20	27	26
6217	5/4/2005	7:27:05 PM	C	A	234	12	0	20	27	26
6218	5/4/2005	7:27:05 PM	C	A	234	12	0	20	27	26
6219	5/4/2005	7:27:07 PM	C	A	235	12	0	20	27	26
6220	5/4/2005	7:27:07 PM	C	A	235	12	0	20	27	26
6221	5/4/2005	7:27:13 PM	C	A	236	12	0	20	27	26
6222	5/4/2005	7:27:13 PM	C	A	236	12	0	20	27	26
6223	5/4/2005	7:27:19 PM	C	A	235	12	0	20	27	26
6224	5/4/2005	7:27:19 PM	C	A	235	12	0	20	27	26
6225	5/4/2005	7:27:22 PM	C	A	235	12	0	20	27	26
6226	5/4/2005	7:27:22 PM	C	A	235	12	0	20	27	26
6229	5/4/2005	7:28:28 PM	C	A	233	12	0	19	27	26
6230	5/4/2005	7:28:28 PM	C	A	233	12	0	19	27	26
6231	5/4/2005	7:28:30 PM	C	A	233	12	0	20	27	26
6232	5/4/2005	7:28:30 PM	C	A	233	12	0	20	27	26
6233	5/4/2005	7:28:32 PM	C	A	233	12	0	20	27	26
6234	5/4/2005	7:28:32 PM	C	A	233	12	0	20	27	26
6235	5/4/2005	7:28:34 PM	C	A	233	12	0	20	27	26
6236	5/4/2005	7:28:34 PM	C	A	233	12	0	20	27	26
6237	5/4/2005	7:28:36 PM	C	A	233	12	0	20	27	26
6238	5/4/2005	7:28:36 PM	C	A	233	12	0	20	27	26
6239	5/4/2005	7:28:38 PM	C	A	233	12	0	20	27	26
6240	5/4/2005	7:28:38 PM	C	A	233	12	0	20	27	26
6241	5/4/2005	7:28:40 PM	C	A	233	12	0	20	27	26
6242	5/4/2005	7:28:40 PM	C	A	233	12	0	20	27	26

รูปที่ 3.11 หน้าจอแสดงข้อมูลทั้งหมด

ค่าที่เก็บไว้ในฐานข้อมูลจะประกอบด้วย ผู้ส่ง ผู้รับ ค่าแรงดันไฟฟ้า ค่ากระแสไฟฟ้าตัวที่ 1 ค่ากระแสไฟฟ้าตัวที่ 2 ค่าอุณหภูมิจุดที่ 1 และค่าอุณหภูมิจุดที่ 2 โดยสามารถที่จะเรียกดูข้อมูลที่เป็นปัจจุบันและย้อนหลังได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 กล่าวนำ

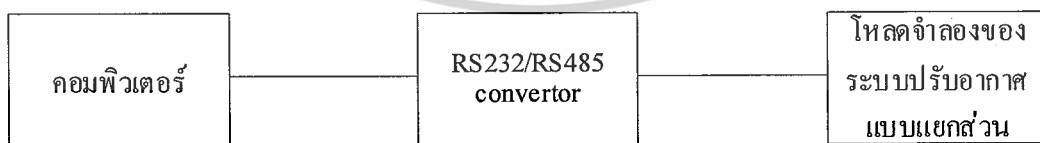
ระบบบันทึกและควบคุมการทำงานของระบบปรับอากาศแบบแยกส่วนที่สร้างขึ้นประกอบด้วย 3 ส่วน คือ ชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ ชุด RS232/RS485 Converter และภาคแสดงผลบนเครื่องคอมพิวเตอร์ โดยชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศจะประกอบด้วย วงจรควบคุม วงจรตรวจจับอุณหภูมิ วงจรดิจิตอล มิเตอร์ วงจรวัดแรงดันไฟฟ้า วงจรวัดกระแสไฟฟ้า วงจรเปิด-ปิดคอมเพรสเซอร์ และวงจรแหล่งจ่ายไฟ โดยมีชุด RS232/RS485 Converter เป็นตัวเชื่อมต่อ ส่วนของภาคแสดงผลบนเครื่องคอมพิวเตอร์ใช้โปรแกรมวิซวลเบสิกในการสร้าง การทดลองของระบบนี้เพื่อหาความสามารถของระบบว่าเป็นไปตามขีดความสามารถหรือไม่ ระบบมีประสิทธิภาพเพียงใด มีส่วนการทำงานใดบกพร่องบ้าง เพื่อที่จะได้แก้ไขและปรับปรุงให้ดีขึ้นต่อไป ซึ่งการทดลองและผลการทดลองเป็นดังต่อไปนี้

4.2 การทดลองการแสดงผลเมื่อเชื่อมต่อกับโหนดจำลองของระบบปรับอากาศแบบแยกส่วน

4.2.1 การทดลอง

ลำดับขั้นตอนการทดลอง

- 1) เชื่อมต่อระบบดังรูปที่ 4.1



รูปที่ 4.1 การเชื่อมต่อกับ โหนดจำลองของระบบปรับอากาศแบบแยกส่วน

- 2) เปิดสวิตซ์การทำงานของ โหนดจำลองของระบบปรับอากาศแบบแยกส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) เข้าสู่โปรแกรมของระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วน โดยจะแสดงผลบนหน้าจอคอมพิวเตอร์

4) ทำการวัดค่ากำลังไฟฟ้าที่โหลดจำลองของระบบปรับอากาศแบบแยกส่วน

4.2.2 ผลการทดลอง

ตารางที่ 4.1 ผลการทดลองการเชื่อมต่อกับ โหลดทดลองจำลองของระบบปรับอากาศแบบแยกส่วน

จำนวนหลอดไฟ	ค่ากำลังไฟฟ้าที่รับได้ (วัตต์)
1	98
2	195
3	294
4	396
5	492
6	593
7	697
8	795
9	892
10	990
11	1093
12	1194
13	1290
14	1396
15	1496
16	1590
17	1695
18	1794
19	1894
20	1992

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

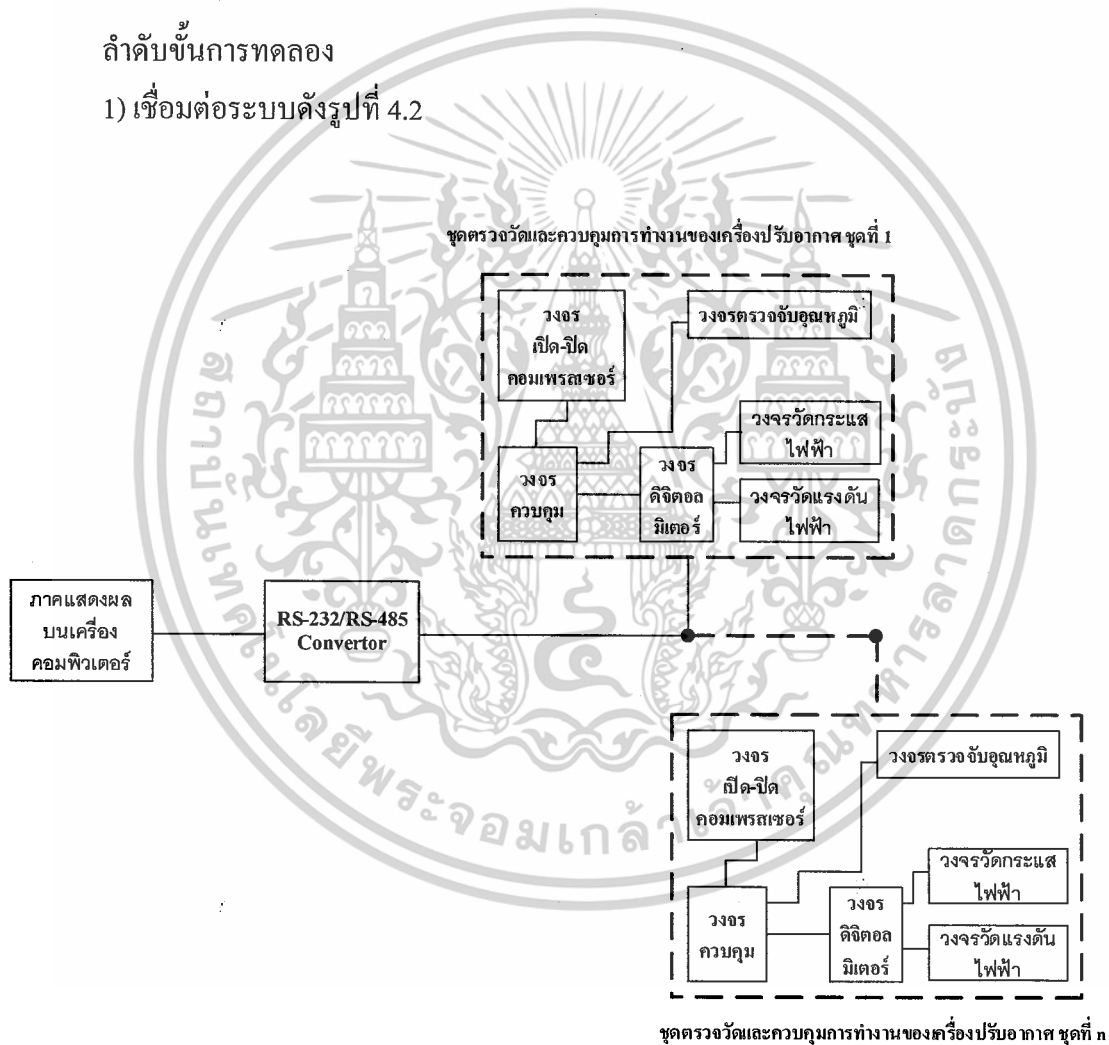
จากผลการทดลองการเชื่อมต่อกับโหนดจำลองของระบบปรับอากาศแบบแยกส่วนค่าที่ได้จากผลการทดลองนั้นเป็นไปตามจำนวนโหนดที่ใช้จริง เช่นหลอดไฟหนึ่งหลอดก็จะมีค่ากำลังไฟฟ้าเท่ากับ 100 วัตต์ เป็นต้น ซึ่งสามารถทำให้ทราบถึงการรับส่งข้อมูลได้

4.3 การทดลองการแสดงผลเมื่อเชื่อมต่อระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วนเข้ากับเครื่องปรับอากาศ

4.3.1 การทดลอง

ลำดับขั้นการทดลอง

- 1) เชื่อมต่อระบบดังรูปที่ 4.2



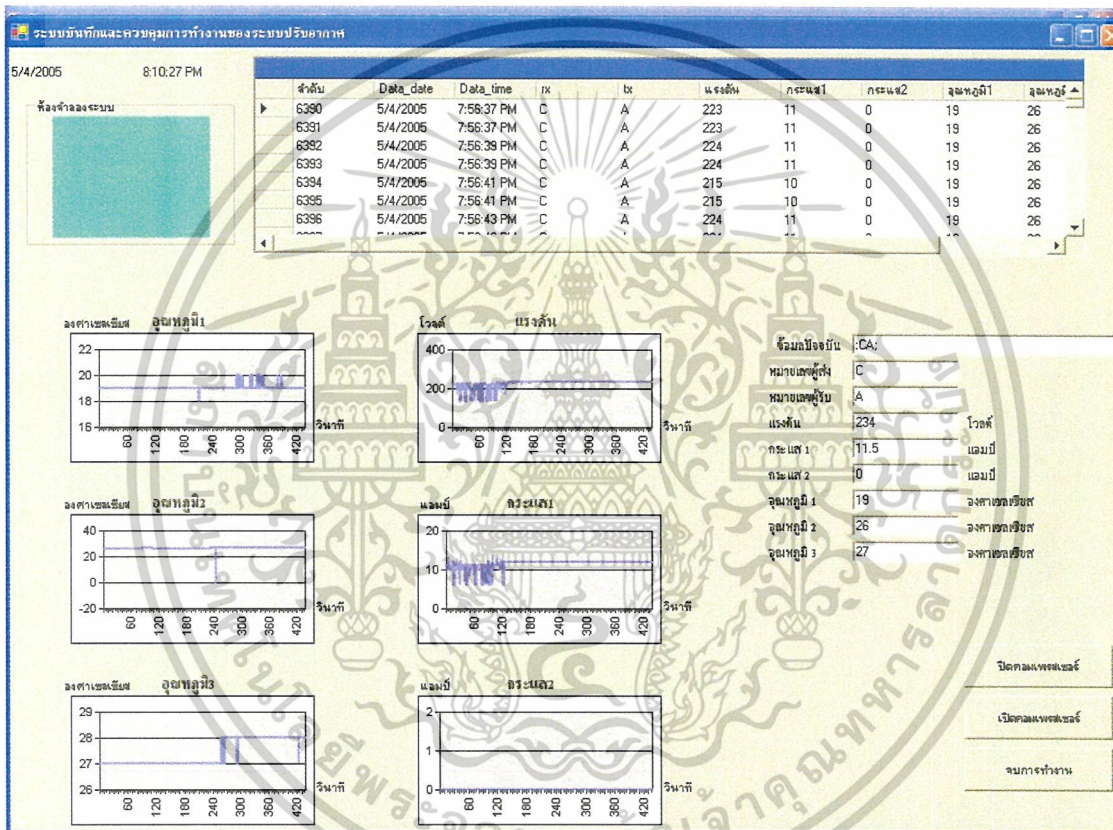
รูปที่ 4.2 การเชื่อมต่อระบบการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2) เปิดสวิตซ์การทำงานของชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ
- 3) เข้าสู่โปรแกรมของระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วน โดยจะแสดงผลบนหน้าจอคอมพิวเตอร์
- 4) ทำการวัดค่าต่างๆ

4.3.2 ผลการทดลอง

ผลการทดลองแสดงได้ดังรูปที่ 4.3



รูปที่ 4.3 ผลการทดลองการแสดงผลเมื่อเชื่อมต่อกับระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วนเข้ากับเครื่องปรับอากาศ

จากผลการทดลองระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วนที่ต่อเข้ากับเครื่องปรับอากาศนี้ สามารถที่จะแสดงค่าที่รับได้เป็นตัวเลขและกราฟ ข้อมูลจะมีการเปลี่ยนแปลงทุกๆ 2 วินาที ข้อมูลที่รับได้นั้นจะนำไปเก็บไว้ในฐานข้อมูล โดยสามารถเรียกดูค่าต่างๆ เหล่านี้ย้อนหลังได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

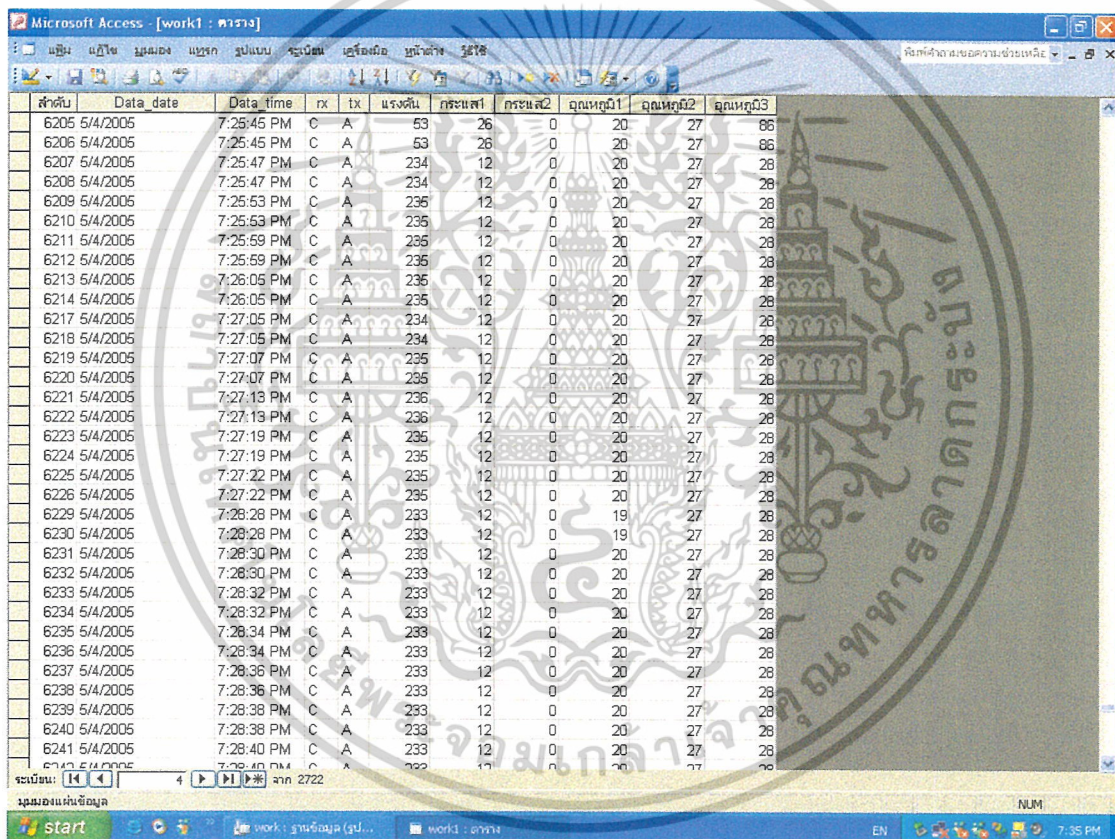
4.4 การทดลองการเรียกดูข้อมูลย้อนหลัง

4.4.1 การทดลอง

ทำการทดลองโดยการวัดค่าต่างๆ ทิ้งไว้ แล้วสังเกตการเปลี่ยนแปลงแล้วทำการดูข้อมูลย้อนหลังได้จากฐานข้อมูล

4.4.2 ผลการทดลอง

ผลการทดลองแสดงดังรูปที่ 4.4



ลำดับ	Data_date	Data_time	rx	tx	แรงดัน	กระแส	กระแส2	อุณหภูมิ1	อุณหภูมิ2	อุณหภูมิ3
6205	5/4/2005	7:25:45 PM	C	A	53	26	0	20	27	86
6206	5/4/2005	7:25:45 PM	C	A	53	26	0	20	27	86
6207	5/4/2005	7:25:47 PM	C	A	234	12	0	20	27	28
6208	5/4/2005	7:25:47 PM	C	A	234	12	0	20	27	28
6209	5/4/2005	7:25:53 PM	C	A	235	12	0	20	27	28
6210	5/4/2005	7:25:53 PM	C	A	235	12	0	20	27	28
6211	5/4/2005	7:25:59 PM	C	A	235	12	0	20	27	28
6212	5/4/2005	7:25:59 PM	C	A	235	12	0	20	27	28
6213	5/4/2005	7:26:05 PM	C	A	235	12	0	20	27	28
6214	5/4/2005	7:26:05 PM	C	A	235	12	0	20	27	28
6217	5/4/2005	7:27:05 PM	C	A	234	12	0	20	27	26
6218	5/4/2005	7:27:05 PM	C	A	234	12	0	20	27	26
6219	5/4/2005	7:27:07 PM	C	A	235	12	0	20	27	28
6220	5/4/2005	7:27:07 PM	C	A	235	12	0	20	27	28
6221	5/4/2005	7:27:13 PM	C	A	236	12	0	20	27	26
6222	5/4/2005	7:27:13 PM	C	A	236	12	0	20	27	26
6223	5/4/2005	7:27:19 PM	C	A	235	12	0	20	27	26
6224	5/4/2005	7:27:19 PM	C	A	235	12	0	20	27	26
6225	5/4/2005	7:27:22 PM	C	A	235	12	0	20	27	26
6226	5/4/2005	7:27:22 PM	C	A	235	12	0	20	27	26
6229	5/4/2005	7:28:28 PM	C	A	233	12	0	19	27	26
6230	5/4/2005	7:28:28 PM	C	A	233	12	0	19	27	26
6231	5/4/2005	7:28:30 PM	C	A	233	12	0	20	27	26
6232	5/4/2005	7:28:30 PM	C	A	233	12	0	20	27	26
6233	5/4/2005	7:28:32 PM	C	A	233	12	0	20	27	26
6234	5/4/2005	7:28:32 PM	C	A	233	12	0	20	27	26
6235	5/4/2005	7:28:34 PM	C	A	233	12	0	20	27	26
6236	5/4/2005	7:28:34 PM	C	A	233	12	0	20	27	26
6237	5/4/2005	7:28:36 PM	C	A	233	12	0	20	27	26
6238	5/4/2005	7:28:36 PM	C	A	233	12	0	20	27	26
6239	5/4/2005	7:28:38 PM	C	A	233	12	0	20	27	26
6240	5/4/2005	7:28:38 PM	C	A	233	12	0	20	27	26
6241	5/4/2005	7:28:40 PM	C	A	233	12	0	20	27	26
6242	5/4/2005	7:28:40 PM	C	A	233	12	0	20	27	26

รูปที่ 4.4 ผลการทดลองการเรียกดูข้อมูลย้อนหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุป

5.1 สรุป

ระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วนประกอบไปด้วย 3 ส่วนคือ ชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ ชุดแปลงสัญญาณ RS232/RS485 และภาคแสดงผลบนเครื่องคอมพิวเตอร์ วงจรของชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศประกอบด้วย วงจรควบคุม วงจรตรวจจับอุณหภูมิ วงจรดิจิทัลไมโครคอนโทรลเลอร์ วงจรวัดแรงดันไฟฟ้า วงจรวัดกระแสไฟฟ้า วงจรเปิด-ปิดคอมเพรสเซอร์ และวงจรแหล่งจ่ายไฟ โดยระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วนนี้สามารถตรวจวัดค่าของกระแสไฟฟ้า แรงดันไฟฟ้า และอุณหภูมิ นำไปแสดงผลบนหน้าจอคอมพิวเตอร์ โดยที่หน้าจอคอมพิวเตอร์นั้นจะแสดงค่าข้อมูลต่างๆ ในลักษณะที่เป็นตัวเลข กราฟ โดยค่าของข้อมูลเหล่านี้จะมีการเปลี่ยนแปลงอยู่ตลอดเวลา และสามารถที่จะเรียกดูผลของข้อมูลย้อนหลังได้ ตามช่วงเวลา/วัน/สัปดาห์/เดือน/ปี ที่ต้องการ โดยระบบนี้สามารถส่งข้อมูลผ่านระบบสื่อสารแบบอนุกรม RS 485 ได้อีกด้วย

ระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วนนี้เหมาะสำหรับผู้ที่สนใจในการประหยัดพลังงานไฟฟ้า เพื่อที่จะเป็นส่วนช่วยในการลดค่าใช้จ่ายของค่าไฟฟ้า และเสริมสร้างลักษณะนิสัยที่ดีในการใช้พลังงานอีกด้วย

5.2 ปัญหาและแนวทางแก้ไข

จากการดำเนินการสร้างและทดสอบ โครงการงานพบว่า มีปัญหาเกิดขึ้นในการทำงานหลายประการ ซึ่งสรุปได้ดังนี้

1. ปัญหา วงจรวัดแรงดันเกิดการระเบิดบ่อยครั้งเนื่องจากแรงดันที่เข้ามามากเกินไป

แนวทางแก้ไข เพิ่มหม้อแปลงไฟฟ้าให้กับวงจรเพื่อที่จะเปลี่ยนแปลงแรงดันให้ลดน้อยลง

2. ปัญหา โปรแกรม Visual Basic.Net ที่ใช้ส่วนใหญ่จะมีเนื้อหาในหนังสือไม่เพียงพอในการใช้เขียนโปรแกรม

แนวทางแก้ไข ทำการค้นคว้าหาข้อมูลจากอินเทอร์เน็ต แล้วนำข้อมูลที่ได้นั้นมาใช้ในการเขียนโปรแกรมต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ปัญหา การส่งสัญญาณในบางครั้ง ข้อมูลไม่แสดงที่บนหน้าจอเครื่องคอมพิวเตอร์
แนวทางแก้ไข เปลี่ยนสายนำสัญญาณของพอร์ท RS 232 ใหม่ เนื่องจากสายตรงหัวต่อหลุด

5.3 แนวทางการพัฒนา

1. สามารถนำชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศไปใช้กับห้องอื่นๆ
ได้อีกด้วย
2. เปลี่ยนจากการส่งสัญญาณจากคู่สายโทรศัพท์เป็นการส่งในระบบ Local Area Network
(LAN)
3. ระบบที่สร้างขึ้นนี้ใช้กับระบบปรับอากาศแบบแยกส่วนได้เพียงอย่างเดียว ควรพัฒนาให้
มีการใช้งานในงานอื่นๆ ได้ด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- ธีรวัฒน์ ประกอบผล. การประยุกต์ใช้งานไมโครคอนโทรลเลอร์. พิมพ์ครั้งที่ 7. กรุงเทพฯ : สำนักพิมพ์ ส.ส.ท. 2546.
- ธีรวัฒน์ ประกอบผล. การพัฒนาไมโครคอนโทรลเลอร์ด้วยภาษาซี. พิมพ์ครั้งที่ 2. กรุงเทพฯ : สำนักพิมพ์ ส.ส.ท. 2545.
- ศุภชัย สมพานิช. Database Programming กับ VB.NET. นนทบุรี : สำนักพิมพ์อินโฟเพรส. 2545.
- ศักรินทร์ โสภนันทะ. เครื่องมือวัดและการวัดทางไฟฟ้า. กรุงเทพฯ : ซีเอ็ดดูเคชั่น. 2545.
- สมศักดิ์ สุโมตยกุล. หลักการทำงานและเทคนิคการตรวจซ่อมเครื่องทำความเย็นและเครื่องปรับอากาศ. กรุงเทพฯ : ซีเอ็ดดูเคชั่น. 2545.
- สุรสิทธิ์ คิวประสพศักดิ์ และนนท์ แนวงโสภณ. อินไซต์ Visual Basic .NET. กรุงเทพฯ : สำนักพิมพ์โปรวิชั่น. 2546



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

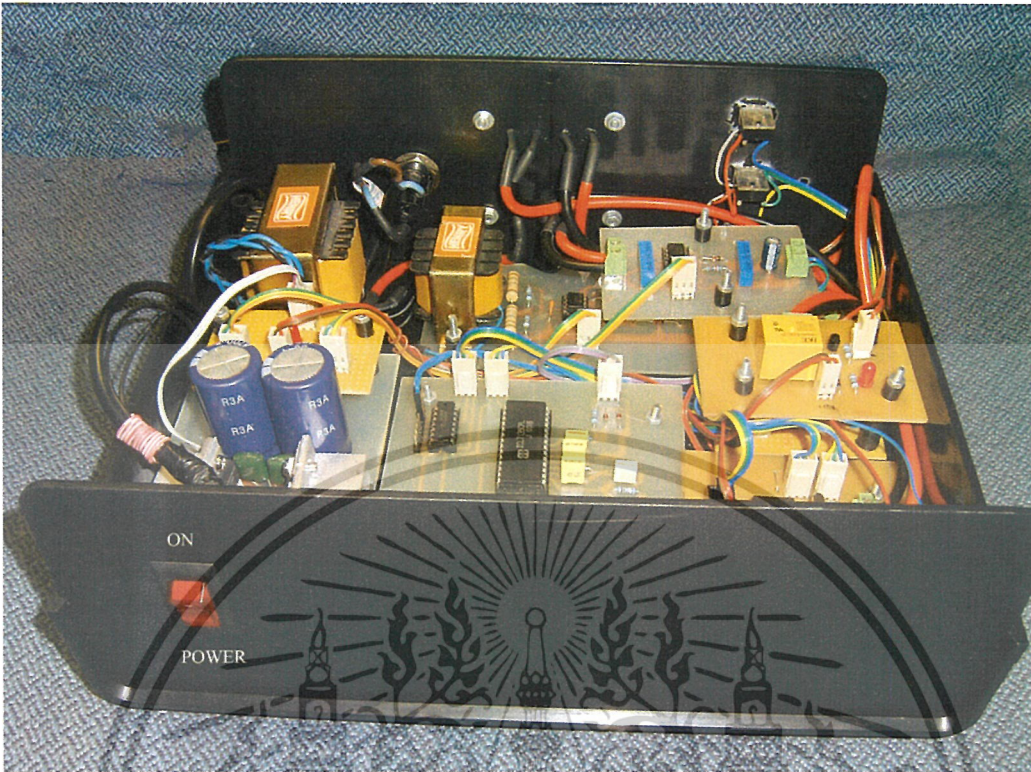


รูปที่ ก.1 ภาพด้านหน้าและด้านบนของชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ



รูปที่ ก.2 ภาพด้านหลังของชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.3 ภาพภายในของชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ



รูปที่ ก.4 ภาพด้านหน้าและด้านบนชุดRS232/RS485 Converter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.5 ภาพด้านหลังของชุดRS232/RS485 Converter

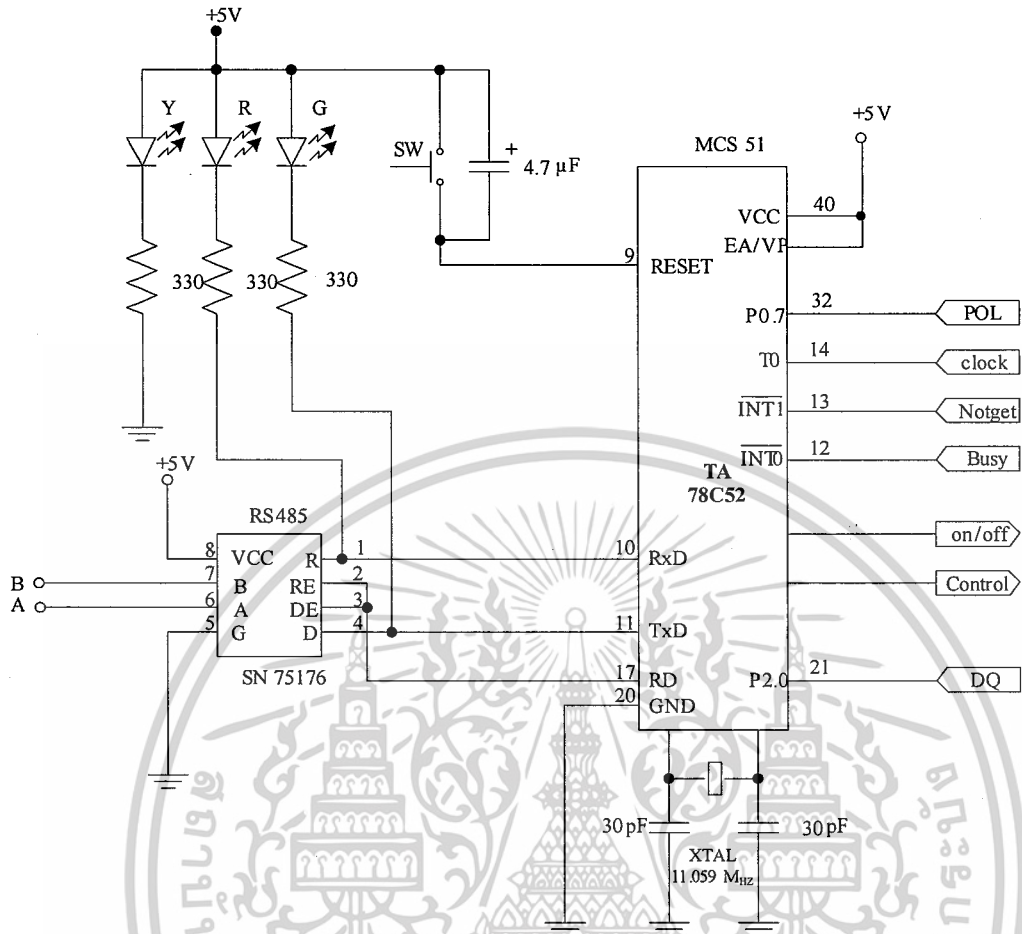


รูปที่ ก.6 ภาพภายในของชุดRS232/RS485 Converter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

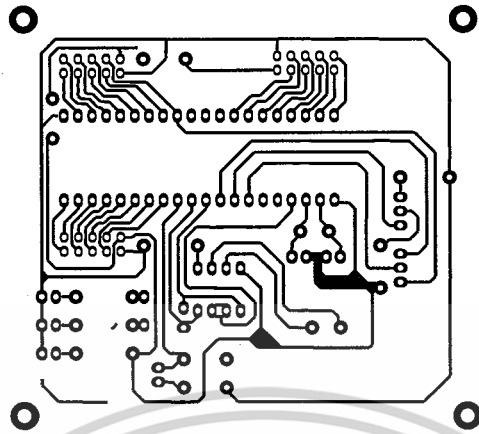


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.1 วงจรควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

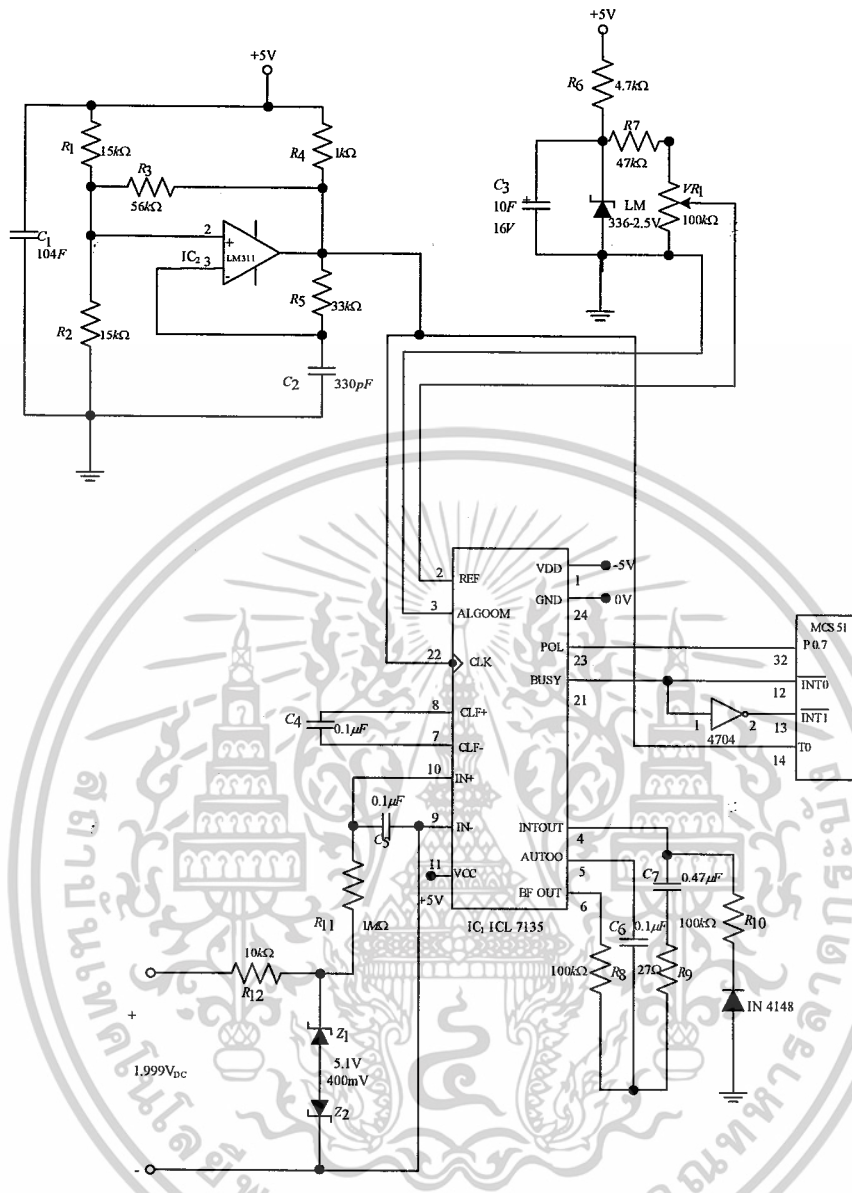


รูปที่ ข.2 แผงวงจรพิมพ์วงจรควบคุม



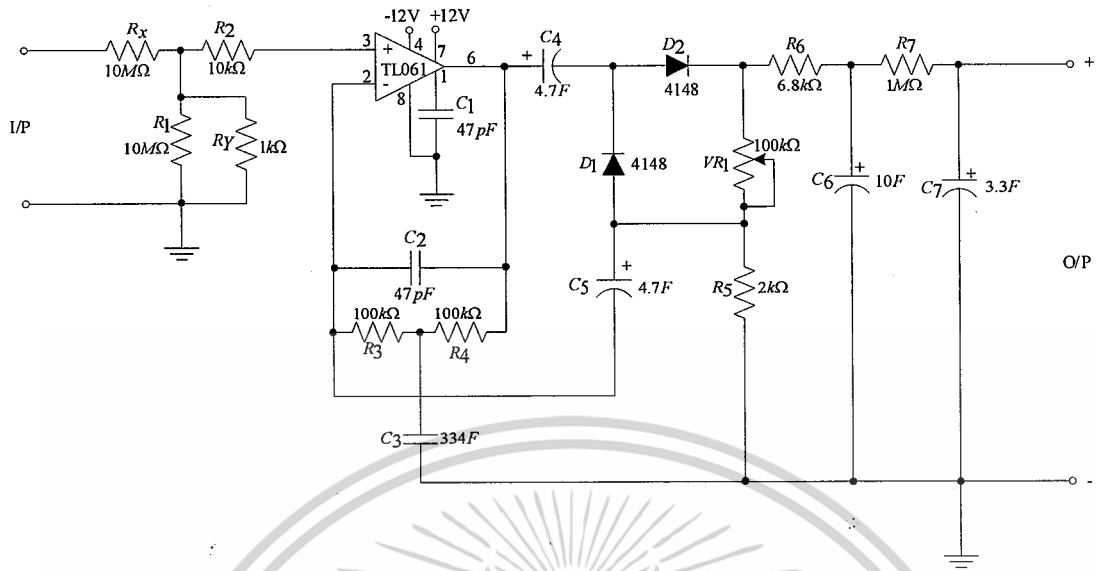
รูปที่ ข.3 ตำแหน่งการวางอุปกรณ์แผงวงจรพิมพ์วงจรควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

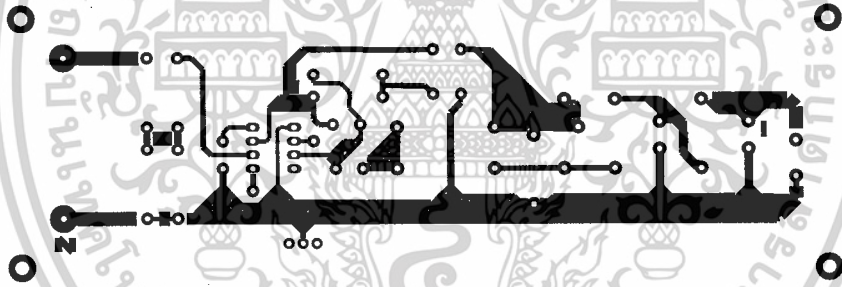


รูปที่ ข.4 วงจรดิจิตอลมิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

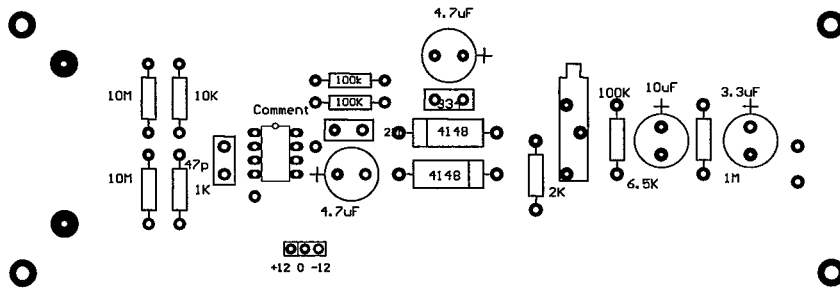


รูปที่ ข.7 วงจรวัดแรงดันไฟฟ้า

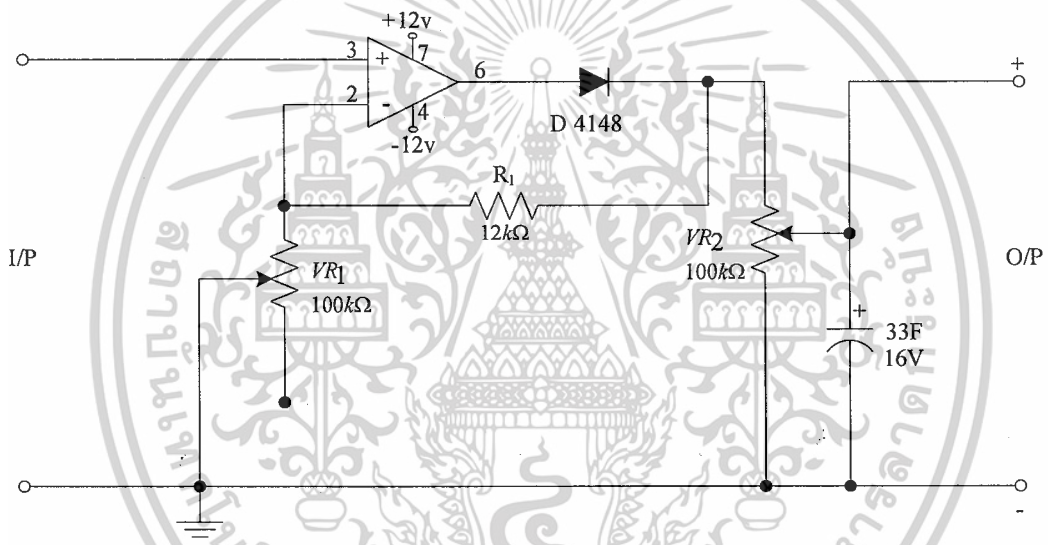


รูปที่ ข.8 แผ่นวงจรพิมพ์วงจรวัดแรงดันไฟฟ้า

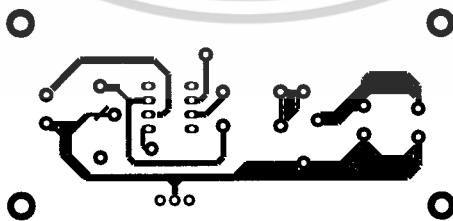
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.9 ตำแหน่งการวางอุปกรณ์แผ่นวงจรพิมพ์วงจรมีฟังก์ชันแรงดันไฟฟ้า

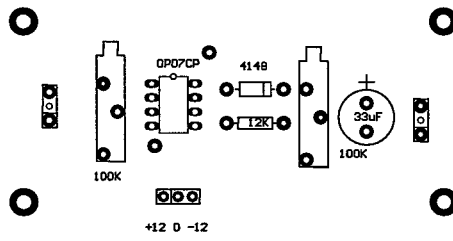


รูปที่ ข.10 วงจรมีฟังก์ชันแรงดันไฟฟ้า

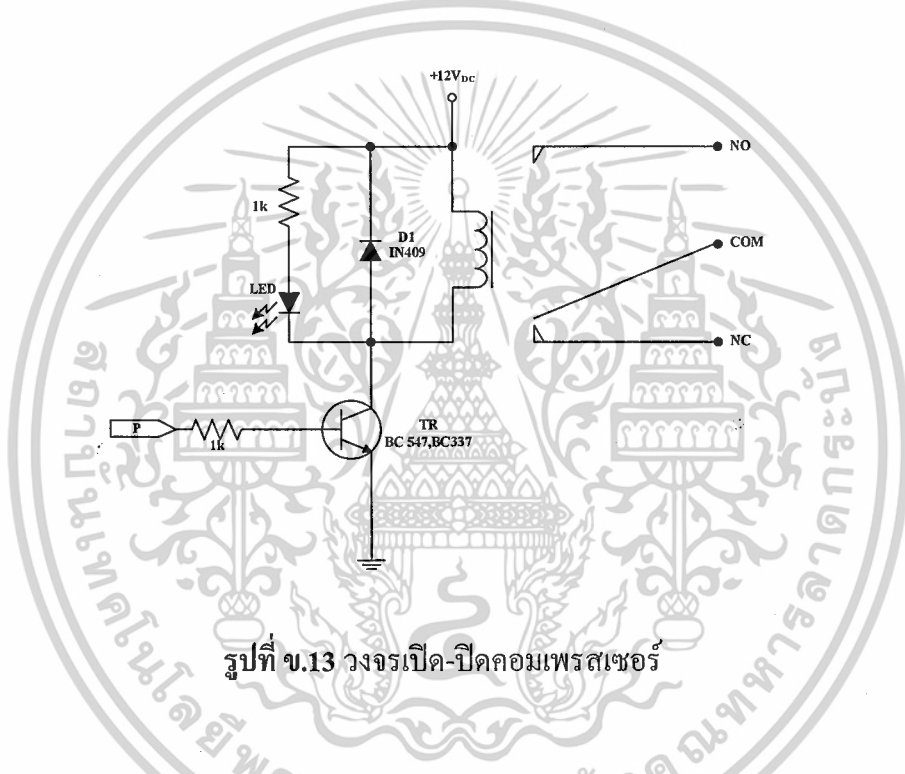


รูปที่ ข.11 แผ่นวงจรพิมพ์วงจรมีฟังก์ชันแรงดันไฟฟ้า

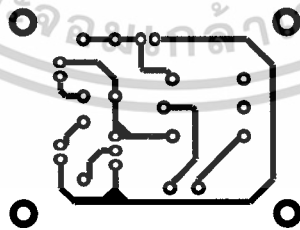
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.12 ตำแหน่งการวางอุปกรณ์แผ่นวงจรพิมพ์วงจรวัดกระแสไฟฟ้า

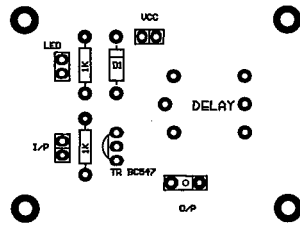


รูปที่ ข.13 วงจรเปิด-ปิดคอมเพรสเซอร์

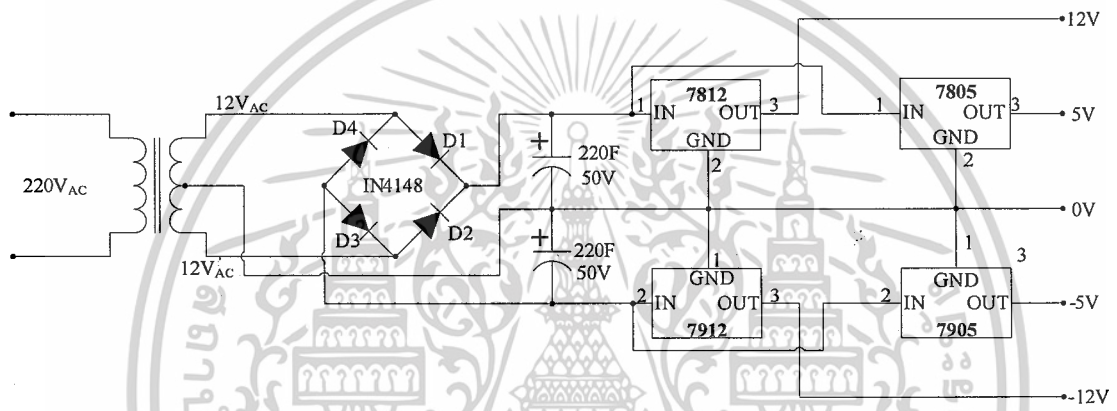


รูปที่ ข.14 แผ่นวงจรพิมพ์วงจรเปิด-ปิดคอมเพรสเซอร์

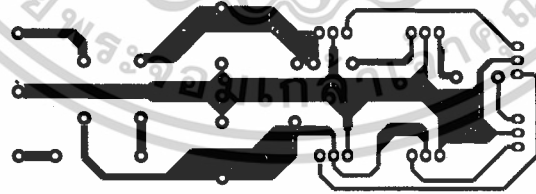
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.15 ตำแหน่งการวางอุปกรณ์แผ่นวงจรพิมพ์วงจรเปิด-ปิดคอมเพรสเซอร์

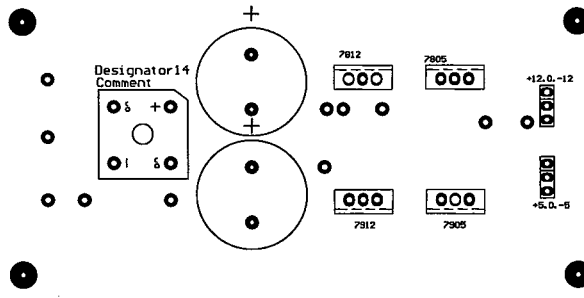


รูปที่ ข.16 วงจรแหล่งจ่ายไฟ

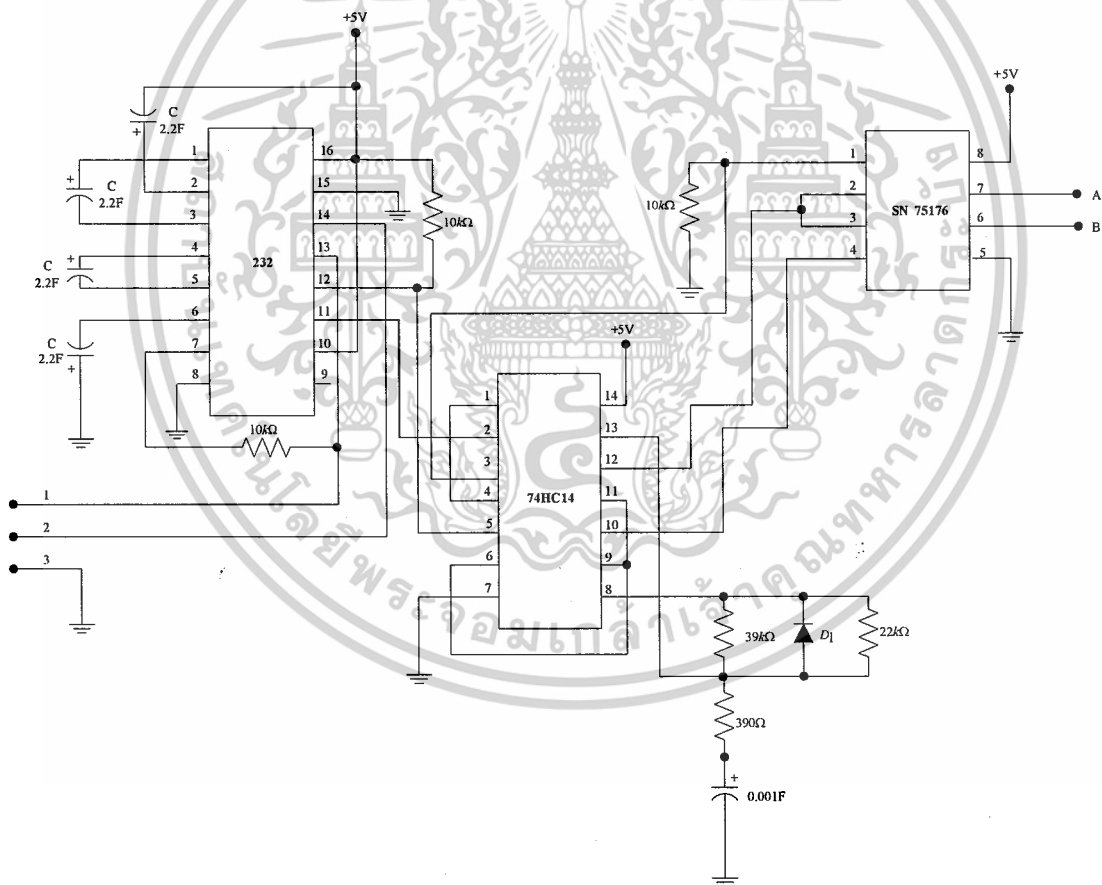


รูปที่ ข.17 แผ่นวงจรพิมพ์วงจรแหล่งจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

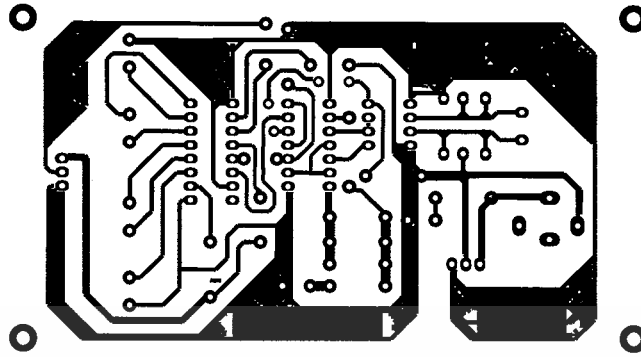


รูปที่ ข.18 ตำแหน่งการวางอุปกรณ์แผ่นวงจรพิมพ์วงจรแหล่งจ่ายไฟ



รูปที่ ข.19 วงจรแปลงสัญญาณ RS232/RS485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.20 แผ่นวงจรพิมพ์วงจรแปลงสัญญาณ RS232/RS485



รูปที่ ข.21 ตำแหน่งการวางอุปกรณ์แผ่นวงจรพิมพ์วงจรแปลงสัญญาณ RS232/RS485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ค
รายการอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 รายการของอุปกรณ์ของวงจรควบคุม

ชื่ออุปกรณ์	รายละเอียด	จำนวน
วงจรรวม		
IC1	SN75176	1 ตัว
IC2	TA 78C52	1 ตัว
อุปกรณ์สารกึ่งตัวนำ		
LED	สีแดง สีเหลือง สีเขียว	3 ตัว
ตัวเก็บประจุ		
C1	4.7 μ F	1 ตัว
C2,C3	30 pF	2 ตัว
ตัวความต้านทาน		
R1,R2,R3	330 Ω	3 ตัว
อุปกรณ์อื่นๆ		
SW	กดติดปล่อยดับ	1 ตัว
XTAL	11.0592 MHz	1 ตัว

ตารางที่ ค.2 รายการของอุปกรณ์ของวงจรดิจิทัลมิเตอร์

ชื่ออุปกรณ์	รายละเอียด	จำนวน
วงจรรวม		
IC1	7135	1 ตัว
IC2	LM311	1 ตัว
IC3	4704	1 ตัว
อุปกรณ์สารกึ่งตัวนำ		
Z1,Z2	5.1V,400 mV	2 ตัว
Z3	LM336,2.5V	1 ตัว
D1	IN4148	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.2 (ต่อ) รายการของอุปกรณ์ของวงจรดิจิทัลคอมพิวเตอร์

ชื่ออุปกรณ์	รายละเอียด	จำนวน
ตัวเก็บประจุ		
C1	104 F	1 ตัว
C2	330 pF	1 ตัว
C3	10 μ F	1 ตัว
C4,C6	1 μ F	2 ตัว
C5	0.1 μ F	1 ตัว
C7	0.47 μ F	1 ตัว
ตัวความต้านทาน		
R1,R2	15 k Ω	2 ตัว
R3	56 k Ω	1 ตัว
R4	1 k Ω	1 ตัว
R5	33 k Ω	1 ตัว
R6	4.7 k Ω	1 ตัว
R7	47 k Ω	1 ตัว
R8,R10	100 k Ω	1 ตัว
R9	27 Ω	2 ตัว
R11	1 M Ω	1 ตัว
R12	10 k Ω	1 ตัว
VR1	100 k Ω	1 ตัว

ตารางที่ ค.3 รายการของอุปกรณ์ของวงจรวัดแรงดันไฟฟ้า

ชื่ออุปกรณ์	รายละเอียด	จำนวน
วงจรรวม		
IC1	TL061	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.3 (ต่อ) รายการของอุปกรณ์ของวงจรวัดแรงดันไฟฟ้า

ชื่ออุปกรณ์	รายละเอียด	จำนวน
อุปกรณ์สารกึ่งตัวนำ		
D1-D2	4148	2 ตัว
ตัวเก็บประจุ		
C1	47 pF	1 ตัว
C2	22 pF	1 ตัว
C3	334 F	1 ตัว
C4,C5	4.7 F	2 ตัว
C6	10 F	1 ตัว
C7	3. F	1 ตัว
ตัวความต้านทาน		
Rx,R1	10 M Ω	2 ตัว
Ry	1 k Ω	1 ตัว
R2	10 k Ω	1 ตัว
R3,R4	100 k Ω	2 ตัว
R5	2 k Ω	1 ตัว
R6	6.8 k Ω	1 ตัว
R7	1 M Ω	1 ตัว
VR1	100 k Ω	1 ตัว

ตารางที่ ค.4 รายการของอุปกรณ์ของวงจรวัดกระแสไฟฟ้า

ชื่ออุปกรณ์	รายละเอียด	จำนวน
วงจรรวม		
IC1	OP07	1 ตัว
อุปกรณ์สารกึ่งตัวนำ		
D1	4148	1ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.4 (ต่อ) รายการของอุปกรณ์ของวงจรวัดกระแสไฟฟ้า

ชื่ออุปกรณ์	รายละเอียด	จำนวน
ตัวเก็บประจุ		
C1	33 μ F 16V	1 ตัว
ตัวความต้านทาน		
R1	12 k Ω	1 ตัว
VR1,VR2	100 k Ω	2 ตัว

ตารางที่ ค.5 รายการของอุปกรณ์ของวงจรเปิด-ปิดคอมเพรสเซอร์

ชื่ออุปกรณ์	รายละเอียด	จำนวน
อุปกรณ์สารกึ่งตัวนำ		
D1	IN409	1 ตัว
LED	สีแดง	1 ตัว
ตัวความต้านทาน		
R1,R2	1 k Ω	2 ตัว
อุปกรณ์อื่นๆ		
TR1	BC337	1 ตัว
รีเลย์	รีเลย์	1 ตัว

ตารางที่ ค.6 รายการของอุปกรณ์ของวงจรแหล่งจ่ายไฟ

ชื่ออุปกรณ์	รายละเอียด	จำนวน
วงจรรวม		
IC1	7812	1 ตัว
IC2	7805	1 ตัว
IC3	7912	1 ตัว
IC4	7905	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.6 (ต่อ) รายการของอุปกรณ์ของวงจรแหล่งจ่ายไฟ

ชื่ออุปกรณ์	รายละเอียด	จำนวน
อุปกรณ์สารกึ่งตัวนำ D1-D4	IN4148	4 ตัว
ตัวเก็บประจุ C1,C2	220 μ F 50V	2 ตัว
อุปกรณ์อื่นๆ T1	หม้อแปลง 220/12-0-12	1 ตัว

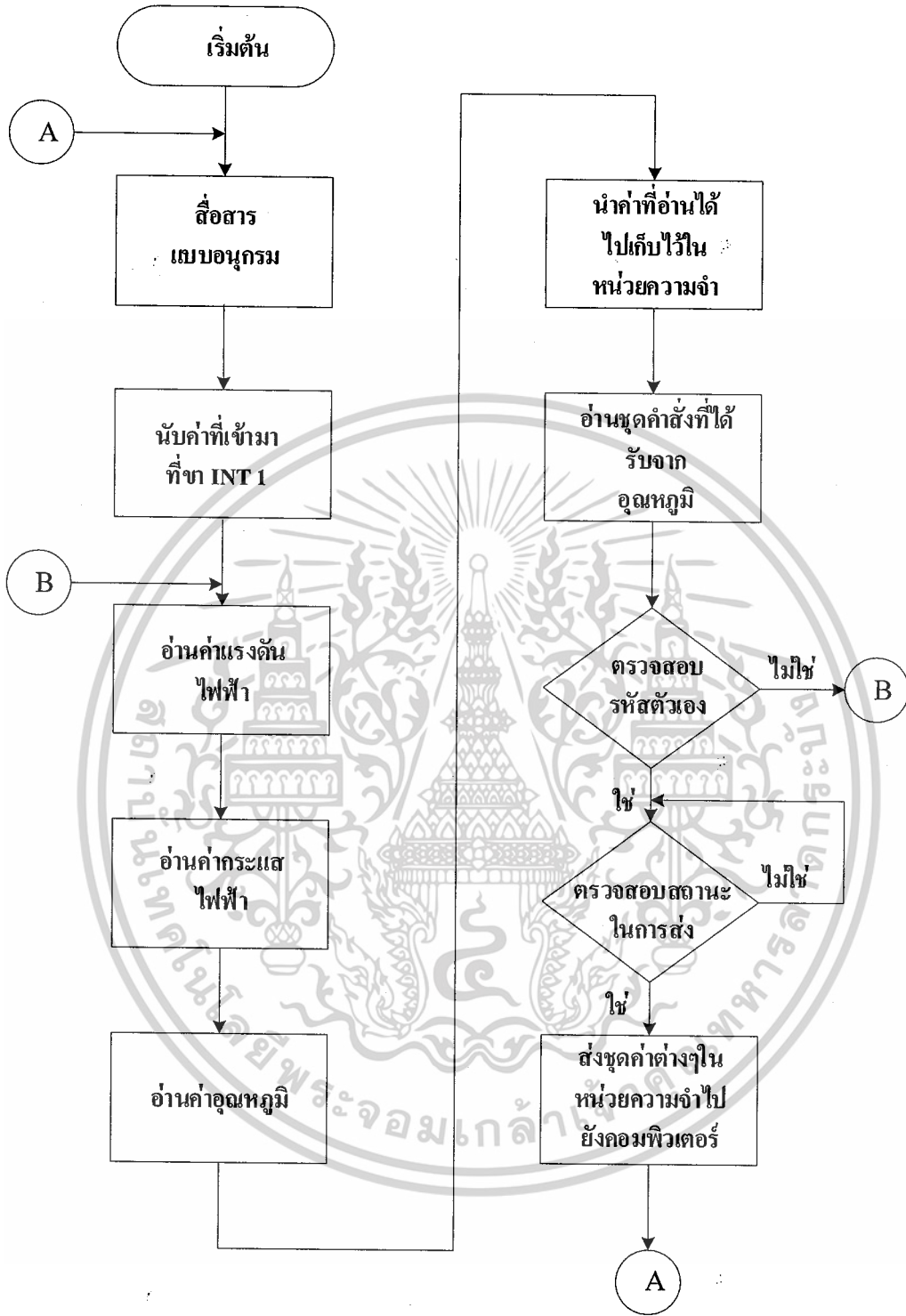
ตารางที่ ค.7 รายการของอุปกรณ์ของวงจรแปลงสัญญาณ RS232/RS485

ชื่ออุปกรณ์	รายละเอียด	จำนวน
วงจรรวม		
IC1	232	1 ตัว
IC2	SN75176	1 ตัว
IC3	74HC14	1 ตัว
อุปกรณ์สารกึ่งตัวนำ		
D1	4148	1 ตัว
ตัวเก็บประจุ		
C1-C4	2.2 μ F	4 ตัว
C5	0.001 μ F	1 ตัว
ตัวความต้านทาน		
R1,R2,R3	10 k Ω	3 ตัว
R4	39 k Ω	1 ตัว
R5	390 Ω	1 ตัว
R6	22 k Ω	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

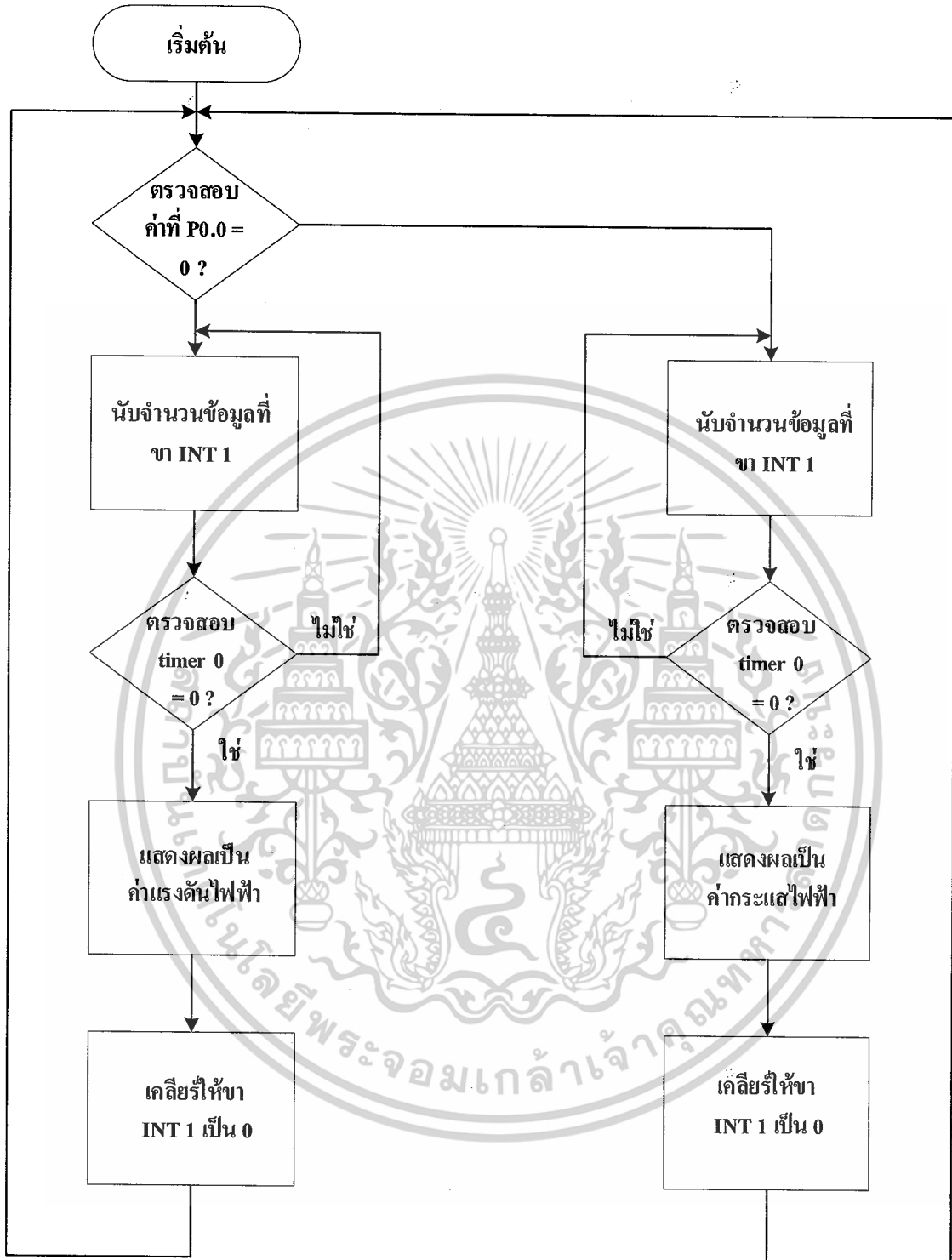


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



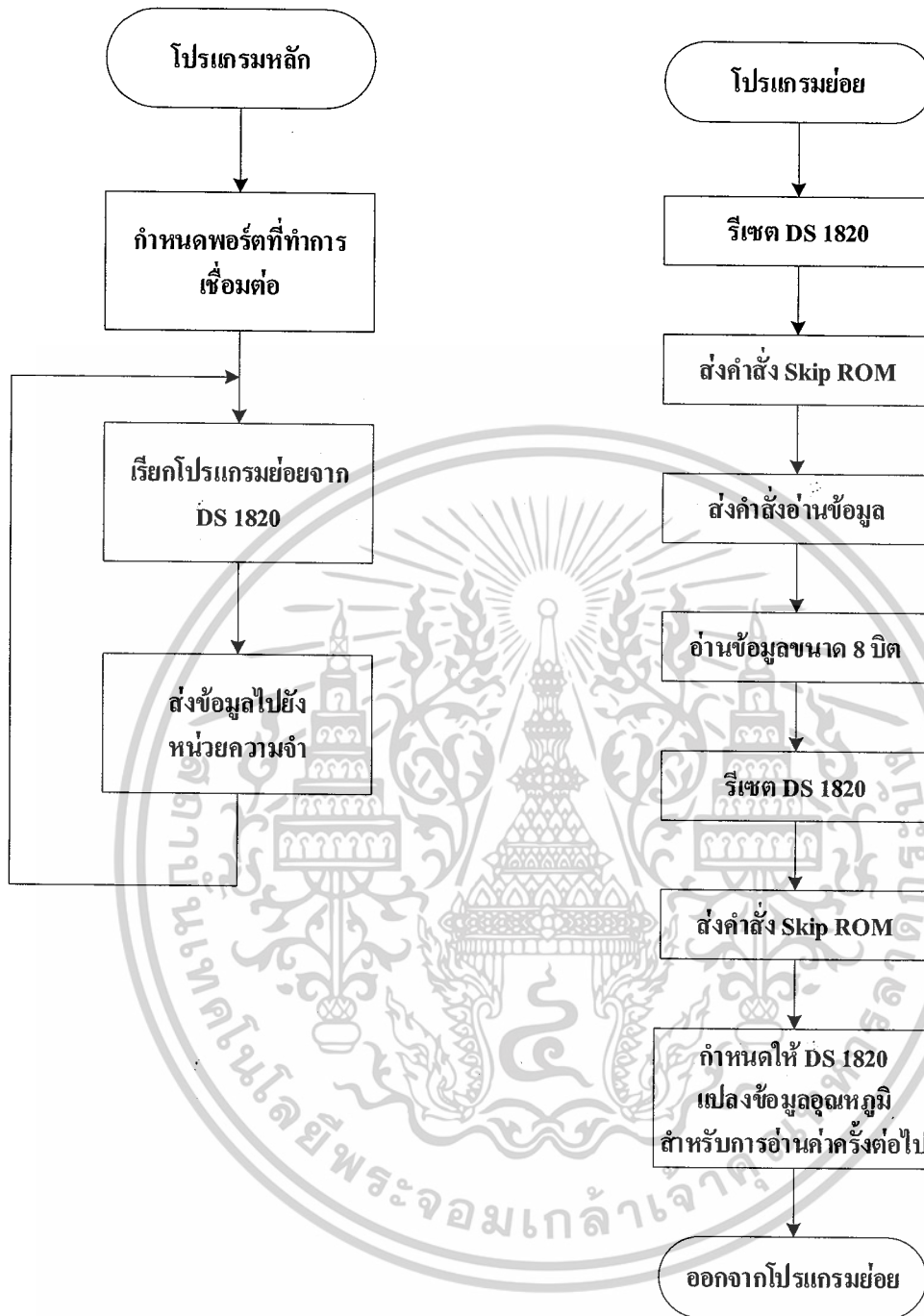
รูปที่ ง.1 ผังงานของโปรแกรมหลัก C 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



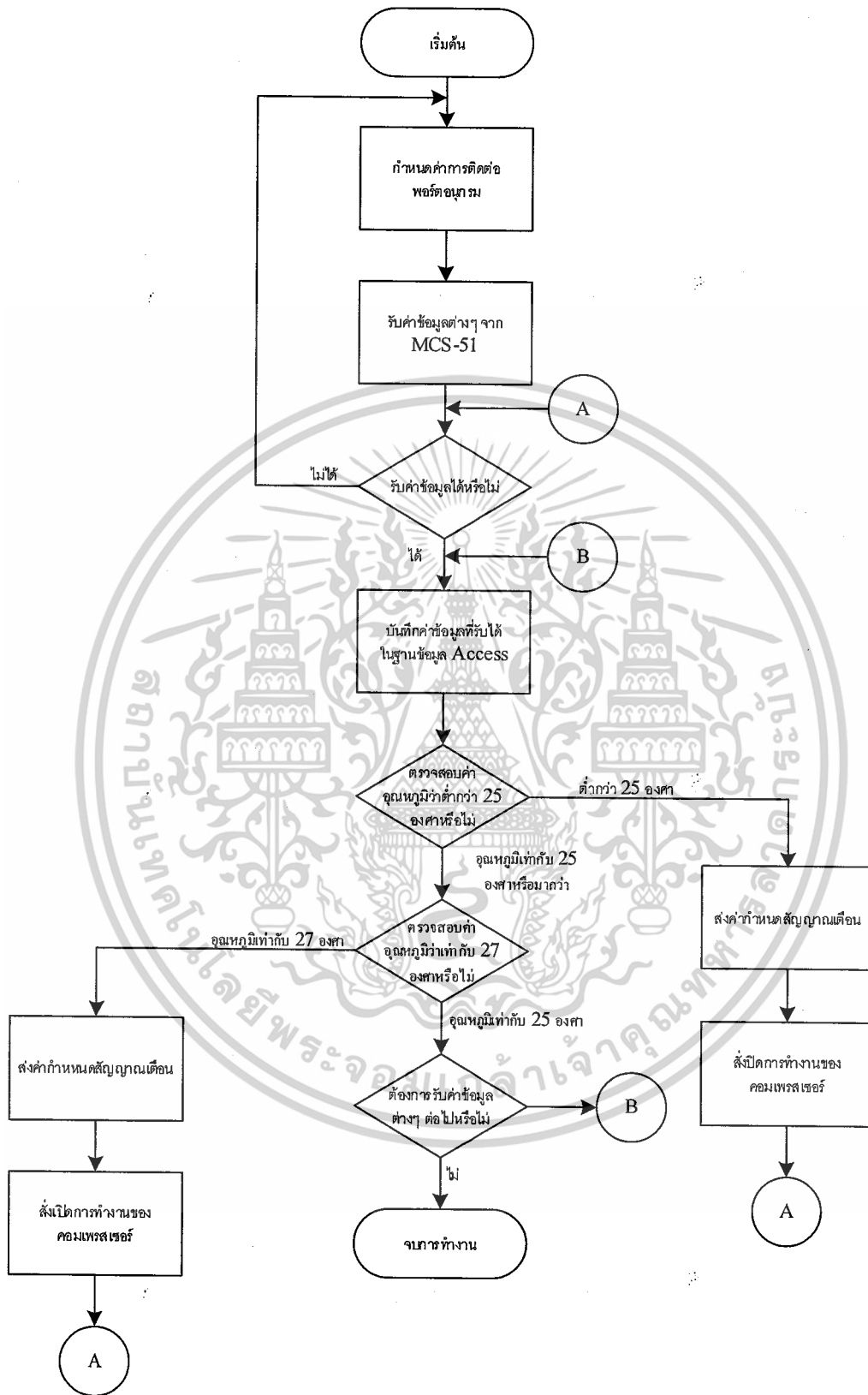
รูปที่ 2.2 ฝั่งงานของโปรแกรมอ่านแรงดันไฟฟ้าและกระแสไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 ผังงานของโปรแกรมวัดอุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 ฟังงานของโปรแกรมหลักการรับส่งค่าที่เครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการทำงานของชุดตรวจวัดค่าต่างๆ

```

#include <reg51.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>
#define TIMER0_COUNT 0xDC11 /* 1000h - ((11,059,200Hz / (12 *
                             FREQ)) - 17) */
static unsigned timer0_tick; /* timer tick variable */
#define My_ID 'B' /* ID for this room */
#define STX 0x3A /* start of text = : */
#define ETX 0x3B /* End of text = ; */
unsigned int a_count = 0;
bit data_ready;
unsigned char inputbuf[16];
unsigned char reportbuf[32];
unsigned char TEMBUF[2];
sbit statusLED = P0^0;
sbit BUSY = P0^1;
sbit HI_IO = P0^2;
sbit TXCTRL = P3^7;
sbit TMDAT = P2^0;
sbit TMDAT1 = P2^1;
sbit TMDAT2 = P2^2;
sbit compressor = P0^3;
static void int0_isr (void) interrupt 0 using 1 /* int1 vector */
{
    unsigned i;
    TR0 = 0;
    a_count = TL0+(TH0<<8);
    TL0 = 0;
    TH0 = 0;
    data_ready = 1;
}

static void int1_isr (void) interrupt 2 using 1 /* int1 vector */
{
    TR0 = 1;
}

void counter0_initialize (void)
{
    EA = 0;
    timer0_tick = 0;
    TR0 = 0;
    TL0 = 0; /* counter 8 bit low = 0000 */
    TH0 = 0; /* counter 8 bit high = 0000 */
    TMOD &= ~0x0F; /* TMOD = TMOD and (1111 0000) */
    TMOD |= 0x05; /* TMOD (C/T0 = Counter 16 bit) */
    PTO = 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ETO = 1;          /* enable interrupt from T0 pin */
EX0 = 1;          /* enable interrupt from INTO pin */
EX1 = 1;          /* enable interrupt from INTO pin */
IT0 = 1;          /* falling edge int 0 */
IT1 = 1;          /* falling edge int 0 */
TRO = 1;
EA = 1;
}

void dmsec (unsigned int count)
{
    unsigned int i;          /* mSec Delay 11.0592 Mhz */
    while (count)           /* Keil v5.2 */
    {
        i = 200; while (i>0) i--;
        count--;
    }
}

void tmreset (unsigned char chipno)
{
    unsigned int i;          /* Reset TX */
    switch (chipno)
    {
    case 0:
        TMDAT = 0;          /* mov p2.0,0 */
        break;
    case 1:
        TMDAT1 = 0;        /* mov p2.1,0 */
        break;
    case 2:
        TMDAT2 = 0;        /* mov p2.2,0 */
        break;
    }
    i = 103;
    while (i>0)             /* 103 loop = 900 uS */
        i--;                /* Approx 900 uS */
    switch (chipno)
    {
    case 0:
        TMDAT = 1;          /* mov p2.0,0 */
        break;
    case 1:
        TMDAT1 = 1;        /* mov p2.1,0 */
        break;
    case 2:
        TMDAT2 = 1;        /* mov p2.2,0 */
        break;
    }

    i = 4;
    while (i>0)             /* delay */
        i--;
}

void tmpre (unsigned char chipno)
{
    /* Wait for Presence RX
*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unsigned int i;
switch(chipno)
{
case 0:
    while (TMDAT) ;           /* loop until tmdat = 0 */
    while (~TMDAT);          /* loop until tmdat = 1 */
    break;
case 1:
    while (TMDAT1) ;         /* loop until tmdat1 = 0 */
    while (~TMDAT1);        /* loop until tmdat1 = 1 */
    break;

case 2:
    while (TMDAT2) ;         /* loop until tmdat2 = 0 */
    while (~TMDAT2);        /* loop until tmdat2 = 1 */
    break;
}

    i = 4;
    while (i>0)
        i--;
}

bit tmrbit (unsigned char chipno)
{
    unsigned int i;           /* read one bit */
    bit dat;
    switch (chipno)
    {
case 0:
        TMDAT = 0; i++;
        TMDAT = 1; i++; i++;
        dat = TMDAT;
        break;
case 1:
        TMDAT1 = 0; i++;
        TMDAT1 = 1; i++; i++;
        dat = TMDAT1;
        break;
case 2:
        TMDAT2 = 0; i++;
        TMDAT2 = 1; i++; i++;
        dat = TMDAT2;
        break;
    }

    i = 8;
    while (i>0) i--;         /* Approx 65 uS */
    return (dat);
}

unsigned char tmrbyte (unsigned char chipno)
{
    unsigned char i,j,dat;    /* read one byte */
    dat = 0;
    for (i=1;i<=8;i++)
    {
        j = tmrbit (chipno);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        dat = (j << 7) | (dat >> 1);
    }
    return (dat);
}

void tmwbyte (unsigned char chipno, unsigned char dat)
{
    /* write one byte */
    unsigned int i;
    unsigned char j;
    bit testb;
    for (j=1;j<=8;j++)
    {
        testb = dat & 0x01;
        dat = dat >> 1;
        if (testb)
        {
            switch(chipno)
            {
            case 0:
                TMDAT = 0; /* Write 1 */
                i++;
                i++; /* Approx 4 uS */
                TMDAT = 1;
                i = 8;
                while (i>0) i--; /* Approx 65 uS */
                break;
            case 1:
                TMDAT1 = 0; /* Write 1 */
                i++;
                i++; /* Approx 4 uS */
                TMDAT1 = 1;
                i = 8;
                while (i>0) i--; /* Approx 65 uS */
                break;
            case 2:
                TMDAT2 = 0; /* Write 1 */
                i++;
                i++; /* Approx 4 uS */
                TMDAT2 = 1;
                i = 8;
                while (i>0) i--; /* Approx 65 uS */
                break;
            }
        }
    }
    else
    {
        switch(chipno)
        {
        case 0:
            TMDAT = 0; /* Write 0 */
            i = 8;
            while (i>0) i--; /* Approx 65 uS */
            TMDAT = 1;
            i++;
            i++; /* Approx 4 uS */
            break;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 1:
    TMDAT1 = 0;                /* Write 0 */
    i = 8;
    while (i>0) i--;          /* Approx 65 uS */
    TMDAT1 = 1;
    i++;
    i++;                        /* Approx 4 uS */
    break;
case 2:
    TMDAT2 = 0;                /* Write 0 */
    i = 8;
    while (i>0) i--;          /* Approx 65 uS */
    TMDAT2 = 1;
    i++;
    i++;                        /* Approx 4 uS */
    break;
}
}
}
}

void tmstart (unsigned char chipno)
{
    /* ds1820 start convert */
    tmreset (chipno);
    tmpre (chipno);
    dmsec (1);
    tmwbyte (chipno, 0xcc);    /* skip rom */
    tmwbyte (chipno, 0x44);    /* convert */
}

void tmrtemp (unsigned char chipno)
{
    /* read temp */
    unsigned char a,b;
    TEMBUF[0] = 0;
    TEMBUF[1] = 0;
    tmreset (chipno);
    tmpre (chipno);
    dmsec (1);
    tmwbyte (chipno, 0xcc);    /* skip rom */
    tmwbyte (chipno, 0xbe);    /* convert */
    a = tmrbyte (chipno);      /* LSB */
    b = tmrbyte (chipno);      /* MSB */
    if (b==1) return;         /* don't care negative temp */
    TEMBUF[1] = a & 0x1;       /* 0=x.0 1=x.5 */
    a = a >> 1;
    TEMBUF[0] = a + 1;         /* adjust for Thailand*/

    if(TEMBUF[1] == 0)
    printf("chipno %bd tmpbuf0 %bx tmpbuf1
    %bx\n", chipno, TEMBUF[0], TEMBUF[1]);
}

void serialport initialize (void)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    /* serial port use C/T#1 */
    /* Digital Meter (ICL7135) use port C/T0
*/
    SCON = 0x50; /* SCON: mode 1, 8-bit UART, enable rcvr */
    TMOD |= 0x20; /* TMOD: timer 1, mode 2, 8-bit reload */
    TH1 = 0xfa; /* TH1: reload value for 9600 baud */
    TR1 = 1; /* TR1: timer 1 run */
    TI = 1; /* TI: set TI to send first char of UART
*/
}

void delay(int i)
{
    while(i>0)
        i--;
}

unsigned char read_volt()
{
    unsigned char volt;
    HI_LO = 0;
    delay(1000);
    volt = (a_count-1001)/10;
    HI_LO = 1;
    delay(1000);
    BUSY =!BUSY;
    sprintf(&(reportbuf+4), "%03d", (int)volt);
}

unsigned char read_amp1()
{
    unsigned char amp1;
    HI_LO = 1;
    delay(2000);
    amp1 = (a_count-1001)/10;
    BUSY =!BUSY;
    sprintf(&(reportbuf+7), "%02d", (int)amp1);
}

void read_amp2()
{
    unsigned char amp2 =0;
    sprintf(&(reportbuf+9), "%02d", (int)amp2);
}

void read_temp(unsigned char chipno)
{
    unsigned int i;
    tmstart (chipno);
    i = 150; /* scan display & delay approx 1 sec
*/
    while (i>0)
    {
        i--;
    }

    tmrtemp (chipno); /* read temperature */
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

switch (chipno)
{
case 0:
    sprintf(&(reportbuf+11),"%bd", (unsigned
char)TEMBUF[0]);
    break;
case 1:
    sprintf(&(reportbuf+13),"%bd", (unsigned
char)TEMBUF[0]);
    break;
case 2:
    sprintf(&(reportbuf+15),"%bd", (unsigned
char)TEMBUF[0]);
    break;
}
}
/*
todo: write function
read_volt()
read_amp1()
read_amp2()
read_temp()
read_temp1()
read_temp2()
read_package()
send_package()
*/
char getch()
{
    char c;
    while(!RI);
    c = SBUF;
    RI = 0;
    return (c);
}
unsigned char read_package()
{
    char j=0;
    inputbuf[0]=getch();
    if(inputbuf[0]==STX)
    {
        j = 1;
        while((inputbuf[j]=getch())!=ETX)
        {
            j++;
        }
        inputbuf[j] = '\0';
    }
    else
    {
        j = -1;
        inputbuf[0] = '\0';
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        return j;
    }

    unsigned char check_ID()
    {
        if( inputbuf[1] == My_ID )
            return 1;
        else
            return 0;
    }

    unsigned char check_command()
    {
        if( inputbuf[1] == My_ID )
            return inputbuf[3];
        else
            return -1;
    }

    void CompressorOn(bit onoff)
    {
        if(onoff == 0)
            compressor = 1; /* off compressor */
        else
            compressor = 0; /* on compressor */
    }

    void send_report()
    {
        TXCTRL = 1;
        printf("%s\n", reportbuf);
        TXCTRL = 0;
    }

    void compose_report()
    {
        reportbuf[0] = ':';
        reportbuf[1] = 'A';
        reportbuf[2] = My_ID;
        reportbuf[3] = ' ';
        reportbuf[17] = ':';
        reportbuf[18] = '\0';
    }

    void main (void)
    {
        unsigned int mode, present_mode, next_mode;
        char mode_type[3];
        int i;
        unsigned char numchar;
        unsigned char command char;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mode = 0;
serialport_initialize ();          /* initial serial port */
counter0_initialize ();           /* initial digital meter
*/
printf("ID=%c Version 1.20  ready\n",My_ID);
statusLED = 1;
TXCTRL = 0;
data_ready = 0;
while(1)
{
    read_volt();
    read_amp1();
    read_amp2();
    read_temp(0);
    read_temp(1);
    read_temp(2);

    compose_report();
    read_package();
    if((numchar = read_package()) != -1)
    {
        BUSY = !BUSY;
    }

    command_char = check_command();
    switch(command_char)
    {
    case '0':
        CompressorOn(0);
        break;
    case '1':
        CompressorOn(1);
        break;
    }

    if(check_ID() == 1)
    {
        i = 100;
        while(i--); // delay

        statusLED = !statusLED;
        send_report();
        inputbuf[1] = '\0';
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการแสดงผลบนจอเครื่องคอมพิวเตอร์

```
Imports System
Imports System.Drawing
Imports System.Data
Imports System.Data.OleDb
Imports System.Windows.Forms
Imports System.Threading
Imports System.Collections
Imports System.ComponentModel
Imports System.IO
Namespace TestSubstr
    Public Class Form1
        Inherits System.Windows.Forms.Form
        Dim inPackage As String
        Dim runmode As Integer
        Dim Conn As OleDbConnection = New OleDbConnection
        Dim olecmd As OleDbCommand = New OleDbCommand
        Dim ds As DataSet = New DataSet
        Dim dataAdap As OleDbDataAdapter
        Dim sqlAdd As String = " "
        Dim strConn As String =
        "Provider=Microsoft.JET.OLEDB.4.0;data source=D:\\project\\work.mdb"
        Dim ff As Integer
        Dim counter As Integer
        Dim hasdata As Boolean = False

        #Region " Windows Form Designer generated code "

        Private Sub Form1_Load(ByVal sender As System.Object, ByVal
        e As System.EventArgs) Handles MyBase.Load
            PortCom1.Open("COM1")
            Timer1.Start()
            Timer1.Enabled = True
        End Sub

        Private Sub PortController1_OnDataReceived(ByVal sender As
        System.Object, ByVal e As System.EventArgs) Handles
        PortCom1.OnDataReceived
            Me.inPackage = PortCom1.Read()
            TextBox1.AppendText(Me.inPackage)
            Trace.WriteLine(TextBox1.Text)
        End Sub

        'ปุ่มจบการทำงาน
        Private Sub Button4_Click(ByVal sender As System.Object,
        ByVal e As System.EventArgs) Handles Button4.Click
            Timer1.Stop()
            Me.Close()
            'จบการทำงาน
            Timer1.Enabled = False
        End Sub

        Sub Fill_Database(ByVal string1 As String)
            Dim str_tx, str_rx, str_volt, str_amp1, str_amp2,
            str_tmp1, str_tmp2, str_tmp3 As String
            string1 = TextBox1.Text
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Trace.WriteLine(TextBox1.Text)
If string1.Length > 30 Then
    TextBox1.ResetText()
End If
If string1.Length < 25 Then
Else
    str_tx = string1.Substring(10, 1) ' rx
    str_rx = string1.Substring(9, 1) ' tx
    str_volt = string1.Substring(12, 3) 'volt
    str_amp1 = string1.Substring(15, 2) / 2 'amp1
    str_amp2 = string1.Substring(17, 2) / 2 'amp2
    str_tmp1 = string1.Substring(19, 2) 'temp1
    str_tmp2 = string1.Substring(21, 2) 'temp2
    str_tmp3 = string1.Substring(23, 2) 'temp3
    TextBox2.Text = str_tx
    TextBox3.Text = str_rx
    TextBox4.Text = str_volt
    TextBox5.Text = str_amp1
    TextBox6.Text = str_amp2
    TextBox7.Text = str_tmp1
    TextBox8.Text = str_tmp2
    TextBox9.Text = str_tmp3

    hasdata = True
    TextBox1.ResetText()

    sqlAdd = "Insert into
work1(Data_date,Data_time,rx,tx,แรงดัน,กระแส1,กระแส2,อุณหภูมิ1,อุณหภูมิ2,อุณหภูมิ3)"
    sqlAdd &= " values('" & Label22.Text & "', '" &
Label24.Text & "', '" & TextBox2.Text & "', '" & TextBox3.Text & "'
, '" & TextBox4.Text & "', '" & TextBox5.Text & "', '" &
TextBox6.Text & "', '" & TextBox7.Text & "', '" & TextBox8.Text &
"', '" & TextBox9.Text & "')"

    Try
        With Conn
            If .State = ConnectionState.Open Then
.Close()
                .ConnectionString = strConn
                .Open()
            End With

            olecmd = New OleDbCommand
            With olecmd
                .CommandType = CommandType.Text
                .CommandText = sqlAdd
                .Connection = Conn
                .ExecuteNonQuery()
            End With

        Catch errorToAdd As System.Exception

            Timer1.Stop()
            PortCom1.Close()
            PortCom1.Open("Com1")
            Timer1.Start()
            Return

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End Try

dataAdap = New OleDbDataAdapter(sqlAdd, Conn)
dataAdap.Fill(DataSet11, "work1") 'ใส่ข้อมูลใน dataset
ชื่อ Serial
    TextBox1.Update()
    TextBox2.Update()
    TextBox3.Update()
    TextBox4.Update()
    TextBox5.Update()
    TextBox6.Update()
    TextBox7.Update()
    TextBox8.Update()
    OleDbDataAdapter1.Fill(DataSet11, "work1")
    DataGrid1.Refresh()
    AxChartSpace1.Update()
    AxChartSpace2.Update()
    AxChartSpace3.Update()
    AxChartSpace4.Update()
    AxChartSpace5.Update()
    AxChartSpace6.Update()
End If

End Sub

Private Sub Timer1_Elapsed(ByVal sender As System.Object,
ByVal e As System.Timers.ElapsedEventArgs) Handles Timer1.Elapsed
    PortCom1.Write(":CA;")
    Label22.Text = Now.ToShortDateString
    Label24.Text = Now.ToLongTimeString
    Fill_Database(inPackage)

    counter = counter + 1
End Sub

Private Sub Timer2_Elapsed(ByVal sender As System.Object,
ByVal e As System.Timers.ElapsedEventArgs) Handles Timer2.Elapsed

    ff = counter Mod 2

    If TextBox7.Text = counter Then
        Panell1.BackColor = Color.Lime
        If ff = 1 Then
            Panell1.BackColor = Color.LimeGreen
        End If
    End If

    If counter > 10 Then
        If TextBox7.Text > 19 Then
            Panell1.BackColor = Color.Yellow
        End If
        If ff = 1 Then
            Panell1.BackColor = Color.Goldenrod
        End If
    End If
    'ทำงานปกติ

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        If TextBox7.Text = 19 Then
            Panell.BackColor = Color.Lime
            If ff = 1 Then
                Panell.BackColor = Color.LimeGreen
            End If
        End If
    End If

    If counter > 15 Then
        If TextBox7.Text > 19 Then
            Panell.BackColor = Color.DarkRed
        End If
        If ff = 1 Then
            Panell.BackColor = Color.Red
        End If
        'ทำงานปกติ
        If TextBox7.Text = 19 Then
            Panell.BackColor = Color.Lime
            If ff = 1 Then
                Panell.BackColor = Color.LimeGreen
            End If
        End If
    End If
    ' End If

End Sub

Private Sub btopen_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btopen.Click
    Timer1.Stop()
    PortCom1.Write(":CA1;")
    Timer1.Start()
End Sub

Private Sub btclose_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btclose.Click
    Timer1.Stop()
    PortCom1.Write(":CA0;")
    Timer1.Start()
End Sub

End Class
End Namespace

```

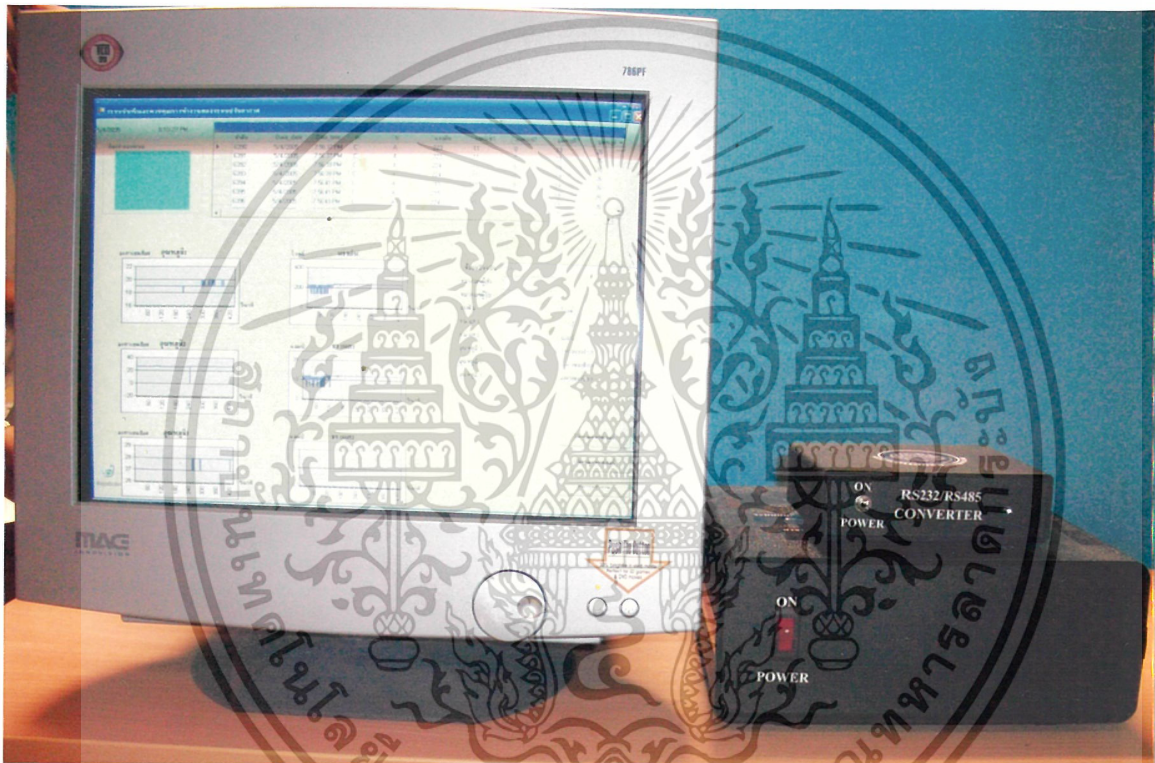
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก จ
คู่มือการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งาน
ระบบบันทึกและควบคุมการใช้พลังงานของ
ระบบปรับอากาศแบบแยกส่วน



ภาควิชาครุศาสตร์วิศวกรรม
 คณะครุศาสตร์อุตสาหกรรม
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. คำแนะนำเบื้องต้น

ก่อนที่จะลงมือใช้งานระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วน ควรทำการศึกษาคู่มือการใช้งานให้เข้าใจ เพื่อที่จะใช้งานได้ถูกต้อง และป้องกันการเสียหายที่จะเกิดขึ้นกับระบบบันทึกและควบคุมการใช้งานของระบบปรับอากาศแบบแยกส่วน

2. ส่วนประกอบและปุ่มควบคุม

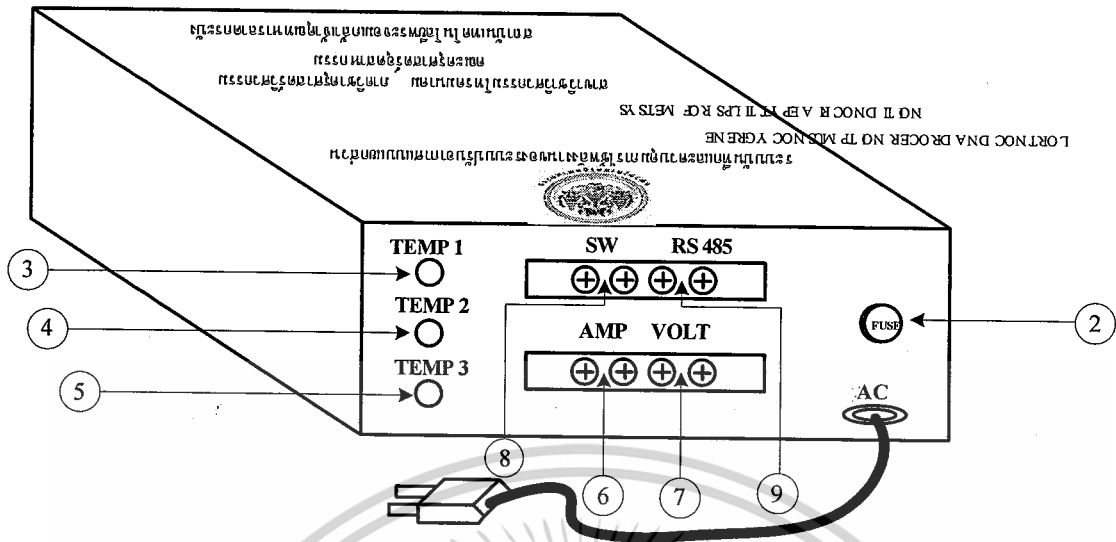
ระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วนประกอบด้วย 3 ส่วน ดังนี้

2.1 ส่วนของชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ



รูปที่ จ.1 ส่วนประกอบและปุ่มควบคุมด้านหน้าของชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



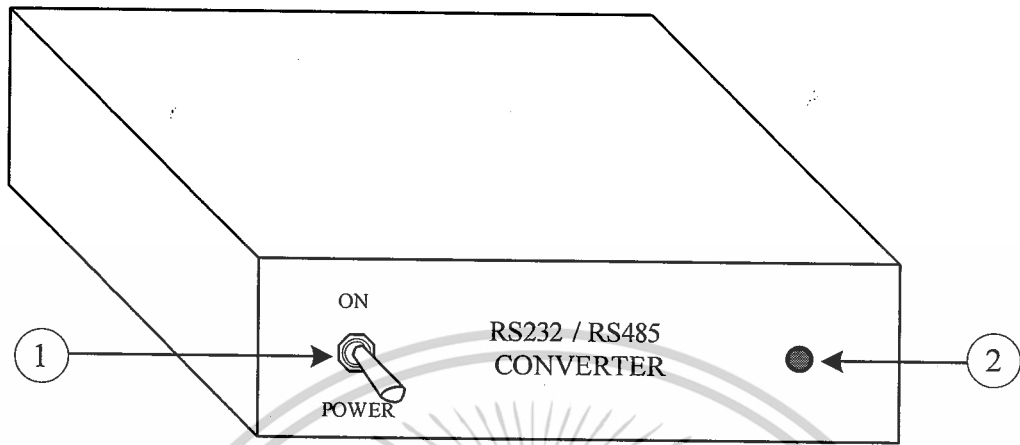
รูปที่ จ.2 ส่วนประกอบและปุ่มควบคุมด้านหลังของชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ

จากรูปที่ จ.1 และ จ.2 มีรายละเอียดต่างๆ ดังนี้

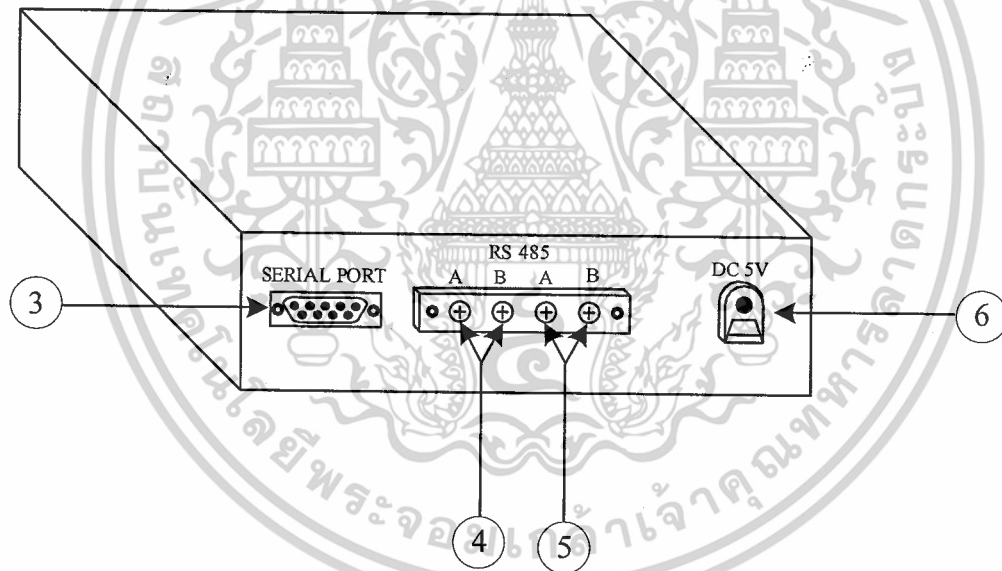
- ① สวิตช์เปิด-ปิดเครื่อง
- ② ฟิวส์
- ③ ตัวตรวจจับอุณหภูมิจุดที่ 1
- ④ ตัวตรวจจับอุณหภูมิจุดที่ 2
- ⑤ ตัวตรวจจับอุณหภูมิจุดที่ 3
- ⑥ กระแสไฟฟ้า
- ⑦ แรงดันไฟฟ้า
- ⑧ สวิตช์
- ⑨ RS 485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ส่วนของชุดแปลงสัญญาณ RS232/RS485



รูปที่ จ.3 ส่วนประกอบและปุ่มควบคุมด้านหน้าชุดแปลงสัญญาณ RS232/RS485



รูปที่ จ.4 ส่วนประกอบและปุ่มควบคุมด้านหลังส่วนของชุดแปลงสัญญาณ RS232/RS485

จากรูปที่ จ.3 และ จ.4 มีรายละเอียดต่างๆ ดังนี้

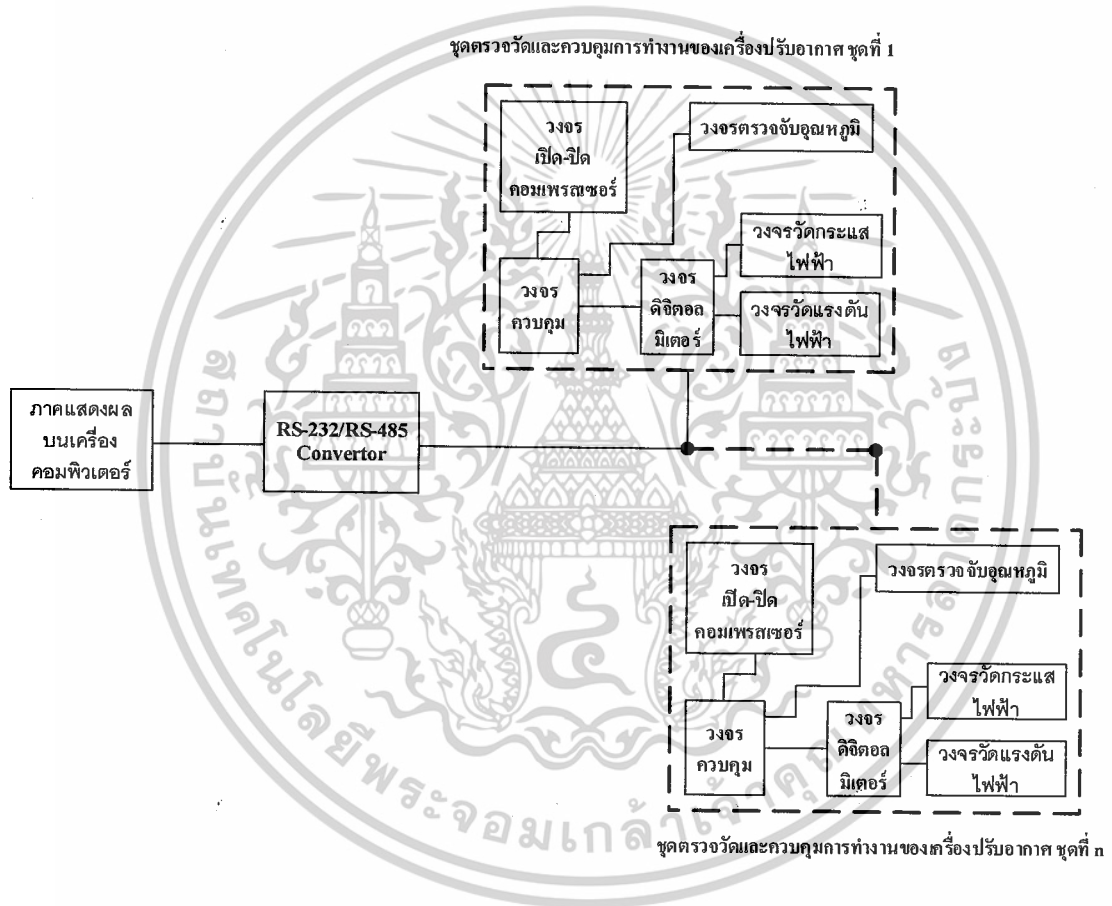
- ① สวิตช์เปิด-ปิดเครื่อง
- ② หลอดแอลอีดีที่แสดงสถานะการทำงาน
- ③ จุดต่อสายไปยังเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4 จุดต่อสาย
- 5 จุดต่อสาย
- 6 แหล่งจ่ายไฟฟ้ากระแสตรง 5 โวลต์

3 การติดตั้งและการใช้งาน

3.1 เชื่อมต่อระบบดังรูปที่ จ.6



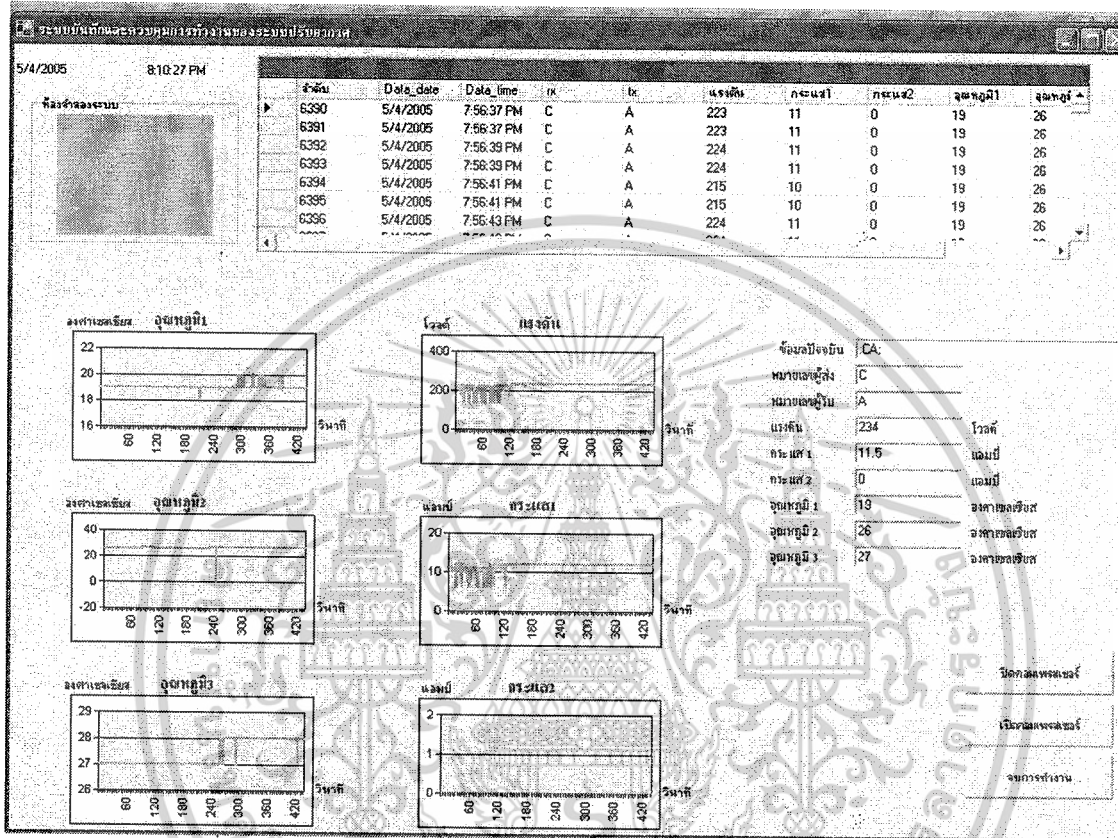
รูปที่ จ.5 การเชื่อมต่อระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วน

- 3.2 เชื่อมต่อกับส่วนควบคุมของระบบปรับอากาศแบบแยกส่วน
- 3.3 ติดตั้งจุดวัดอุณหภูมิภายในห้อง
- 3.4 ทำการเปิดสวิทช์ของชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศและชุด RS232/RS485 Converter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 เปิดโปรแกรมของระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วน

3.6 ค่าที่ตรวจวัดได้จะแสดงออกมาทั้งตัวเลขและกราฟ



รูปที่ 6.6 ค่าต่างๆ ที่วัดได้แสดงบนเครื่องคอมพิวเตอร์

4.การแก้ปัญหาเบื้องต้น

เมื่อท่านประสบปัญหาในการใช้งานระบบบันทึกและควบคุมการใช้พลังงานของระบบปรับอากาศแบบแยกส่วน สามารถตรวจสอบแนวทางแก้ไขปัญหาเบื้องต้นได้จากตารางต่อไปนี้

อาการ	สาเหตุและ/หรือวิธีแก้ไข
ค่าต่างๆ ไม่แสดงที่หน้าจอคอมพิวเตอร์	ตรวจสอบจุดเชื่อมต่อต่างๆ , ตรวจสอบการตั้งค่าพอร์ตของคอมพิวเตอร์ว่าถูกต้องหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อาการ	สาเหตุและ/หรือวิธีแก้ไข
ไม่มีการส่งสัญญาณ จากชุด RS232/RS485 Converter	ตรวจเช็คไฟเลี้ยง 5 โวลต์ว่ามีการจ่ายไฟหรือไม่

5. การดูแลรักษาและข้อควรระวัง

5.1 การดูแลรักษา

- ควรปิดสวิทช์และดึงปลั๊กออกทุกครั้งเมื่อไม่ใช้งาน
- ตรวจเช็คและทำความสะอาดชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศ และชุด RS232/RS485 Converter ให้อยู่ในสภาพพร้อมใช้งานอยู่เสมอ

5.2 ข้อควรระวัง

- ชุดตรวจวัดและควบคุมการทำงานของเครื่องปรับอากาศและชุด RS232/RS485 Converter ก่อนที่จะถอดปลั๊กออกควรปิดสวิทช์ก่อนทุกครั้ง

6. ข้อมูลจำเพาะ

คุณสมบัติ	รายละเอียด
ไอซีตรวจวัดอุณหภูมิ	เบอร์ DS 1820
การแสดงผลบนเครื่องคอมพิวเตอร์	ตัวเลขและกราฟ
แหล่งจ่ายพลังงาน	ไฟฟ้ากระแสสลับ 220 โวลต์ ความถี่ 50 เฮิร์ตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก จ
รายละเอียดและคุณสมบัติของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4¹/₂ Digit, BCD Output, A/D Converter

The Intersil ICL7135 precision A/D converter, with its multiplexed BCD output and digit drivers, combines dual-slope conversion reliability with ± 1 in 20,000 count accuracy and is ideally suited for the visual display DVM/DPM market. The 2.0000V full scale capability, auto-zero, and auto-polarity are combined with true ratiometric operation, almost ideal differential linearity and true differential input. All necessary active devices are contained on a single CMOS IC, with the exception of display drivers, reference, and a clock.

The ICL7135 brings together an unprecedented combination of high accuracy, versatility, and true economy. It features auto-zero to less than 10 μ V, zero drift of less than 1 μ V/ $^{\circ}$ C, input bias current of 10pA (Max), and rollover error of less than one count. The versatility of multiplexed BCD outputs is increased by the addition of several pins which allow it to operate in more sophisticated systems. These include STROBE, OVERRANGE, UNDERRANGE, RUN/HOLD and BUSY lines, making it possible to interface the circuit to a microprocessor or UART.

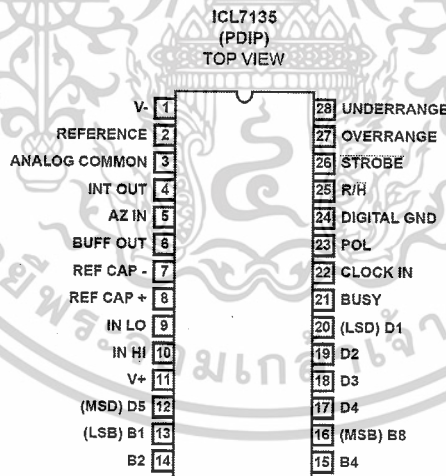
Features

- Accuracy Guaranteed to ± 1 Count Over Entire ± 20000 Counts (2.0000V Full Scale)
- Guaranteed Zero Reading for 0V Input
- 1pA Typical Input Leakage Current
- True Differential Input
- True Polarity at Zero Count for Precise Null Detection
- Single Reference Voltage Required
- Overage and Underrange Signals Available for Auto-Range Capability
- All Outputs TTL Compatible
- Blinking Outputs Gives Visual Indication of Overage
- Six Auxiliary Inputs/Outputs are Available for Interfacing to UARTs, Microprocessors, or Other Circuitry
- Multiplexed BCD Outputs

Ordering Information

PART NUMBER	TEMP. RANGE ($^{\circ}$ C)	PACKAGE	PKG. NO.
ICL7135CP1	0 to 70	28 Ld PDIP	E28.6

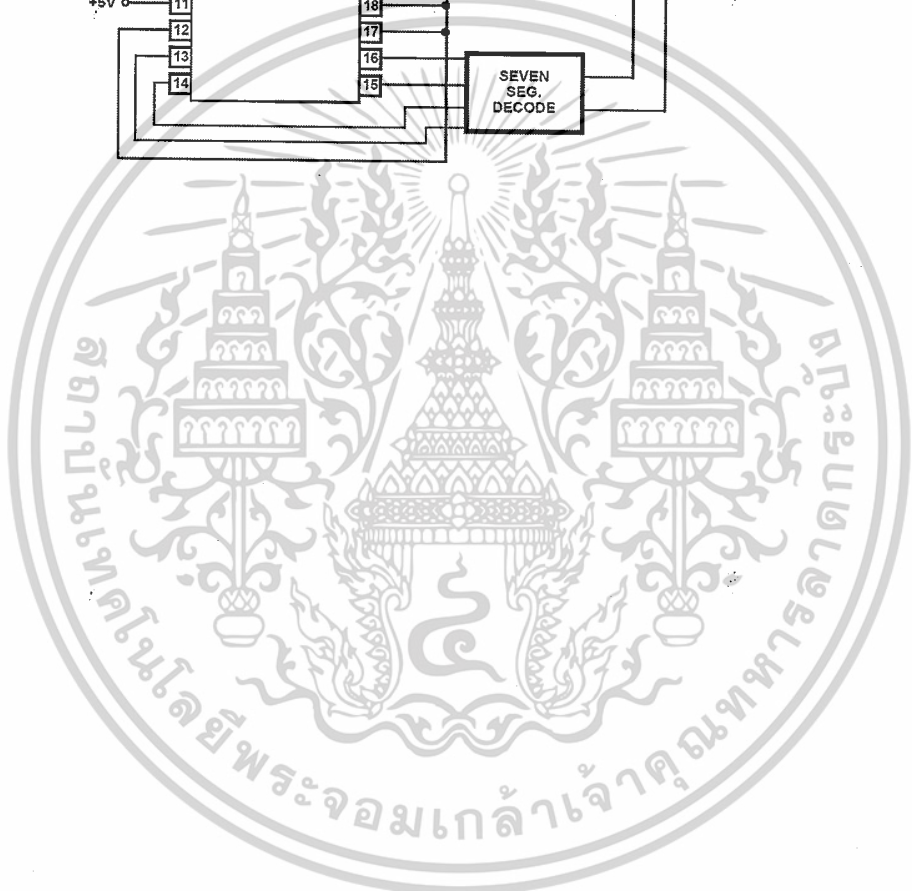
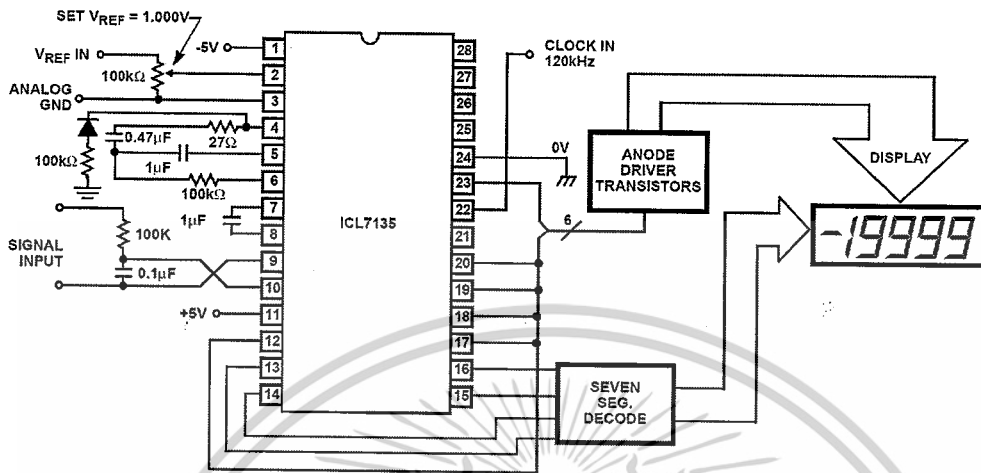
Pinout



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ICL7135

Typical Application Schematic



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ICL7135

Absolute Maximum Ratings

Supply Voltage V+	+6V
V-	-9V
Analog Input Voltage (Either Input) (Note 1)	V+ to V-
Reference Input Voltage (Either Input)	V+ to V-
Clock Input Voltage	GND to V+

Operating Conditions

Temperature Range	0°C to 70°C
-------------------	-------------

CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

NOTES:

1. Input voltages may exceed the supply voltages provided the input current is limited to +100 μ A.
2. θ_{JA} is measured with the component mounted on a low effective thermal conductivity test board in free air. See Tech Brief TB379 for details.

Thermal Information

Thermal Resistance (Typical, Note 2)	θ_{JA} (°C/W)
PDIP Package	55
Maximum Junction Temperature	150°C
Maximum Storage Temperature Range	-65°C to 150°C
Maximum Lead Temperature (Soldering 10s)	300°C

Electrical Specifications V+ = +5V, V- = -5V, T_A = 25°C, f_{CLK} Set for 3 Readings/s, Unless Otherwise Specified

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNITS	
ANALOG (Notes 3, 4)						
Zero Input Reading	V _{IN} = 0V, V _{REF} = 1.000V	-00000	+00000	+00000	Counts	
Ratiometric Error (Note 4)	V _{IN} = V _{REF} = 1.000V	-3	0	+3	Counts	
Linearity Over \pm Full Scale (Error of Reading from Best Straight Line)	-2V \leq V _{IN} \leq +2V	-	0.5	1	LSB	
Differential Linearity (Difference Between Worst Case Step of Adjacent Counts and Ideal Step)	-2V \leq V _{IN} \leq +2V	-	0.01	-	LSB	
Rollover Error (Difference in Reading for Equal Positive and Negative Voltage Near Full Scale)	-V _{IN} \approx +V _{IN} \approx 2V	-	0.5	1	LSB	
Noise (Peak-to-Peak Value Not Exceeded 95% of Time), e _N	V _{IN} = 0V, Full scale = 2.000V	-	15	-	μ V	
Input Leakage Current, I _{ILK}	V _{IN} = 0V	-	1	10	pA	
Zero Reading Drift (Note 7)	V _{IN} = 0V, 0°C to 70°C	-	0.5	2	μ V/°C	
Scale Factor Temperature Coefficient, T _C (Notes 5 and 7)	V _{IN} = +2V, 0°C to 70°C Ext. Ref. 0ppm/°C	-	2	5	ppm/°C	
DIGITAL INPUTS						
Clock In, Run/Hold (See Figure 2)	V _{INH}	2.8	2.2	-	V	
	V _{INL}	-	1.6	0.8	V	
	I _{INL}	V _{IN} = 0V	-	0.02	0.1	mA
	I _{INH}	V _{IN} = +5V	-	0.1	10	μ A
DIGITAL OUTPUTS						
All Outputs, V _{OL}	I _{OL} = 1.6mA	-	0.25	0.40	V	
B1, B2, B4, B8, D1, D2, D3, D4, D5, V _{OH}	I _{OH} = -1mA	2.4	4.2	-	V	
BUSY, STROBE, OVERRANGE, UNDERRANGE, POLARITY, V _{OH}	I _{OH} = -10 μ A	4.9	4.99	-	V	
SUPPLY						
+5V Supply Range, V+		+4	+5	+6	V	
-5V Supply Range, V-		-3	-5	-8	V	
+5V Supply Current, I+	f _C = 0	-	1.1	3.0	mA	
-5V Supply Current, I-	f _C = 0	-	0.8	3.0	mA	
Power Dissipation Capacitance, C _{PD}	vs Clock Frequency	-	40	-	pF	
CLOCK						
Clock Frequency (Note 6)		DC	2000	1200	kHz	

NOTES:

3. Tested in 4^{1/2} digit (20,000 count) circuit shown in Figure 3. (Clock frequency 120kHz.)
4. Tested with a low dielectric absorption integrating capacitor, the 27 Ω INT OUT resistor shorted, and R_{INT} = 0. See Component Value Selection Discussion.
5. The temperature range can be extended to 70°C and beyond as long as the auto-zero and reference capacitors are increased to absorb the higher leakage of the ICL7135.
6. This specification relates to the clock frequency range over which the ICL7135 will correctly perform its various functions. See "Max Clock Frequency" section for limitations on the clock frequency range in a system.
7. Parameter guaranteed by design or characterization. Not production tested.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

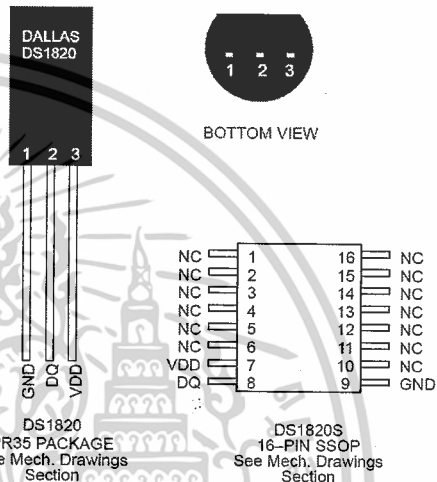
DALLAS SEMICONDUCTOR

DS1820 1-Wire™ Digital Thermometer

FEATURES

- Unique 1-Wire™ interface requires only one port pin for communication
- Multidrop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line
- Zero standby power required
- Measures temperatures from -55°C to $+125^{\circ}\text{C}$ in 0.5°C increments. Fahrenheit equivalent is -67°F to $+257^{\circ}\text{F}$ in 0.9°F increments
- Temperature is read as a 9-bit digital value.
- Converts temperature to digital word in 200 ms (typ.)
- User-definable, nonvolatile temperature alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

PIN ASSIGNMENT



PIN DESCRIPTION

GND	- Ground
DQ	- Data In/Out
V _{DD}	- Optional V _{DD}
NC	- No Connect

DESCRIPTION

The DS1820 Digital Thermometer provides 9-bit temperature readings which indicate the temperature of the device.

Information is sent to/from the DS1820 over a 1-Wire interface, so that only one wire (and ground) needs to be connected from a central microprocessor to a DS1820. Power for reading, writing, and performing temperature conversions can be derived from the data line itself with no need for an external power source.

Because each DS1820 contains a unique silicon serial number, multiple DS1820s can exist on the same 1-Wire bus. This allows for placing temperature sensors in many different places. Applications where this feature is useful include HVAC environmental controls, sensing temperatures inside buildings, equipment or machinery, and in process monitoring and control.

DS1820

DETAILED PIN DESCRIPTION

PIN 16-PIN SSOP	PIN PR35	SYMBOL	DESCRIPTION
9	1	GND	Ground.
8	2	DQ	Data Input/Output pin. For 1-Wire operation: Open drain. (See "Parasite Power" section.)
7	3	V _{DD}	Optional V _{DD} pin. See "Parasite Power" section for details of connection.

DS1820S (16-pin SSOP): All pins not specified in this table are not to be connected.

OVERVIEW

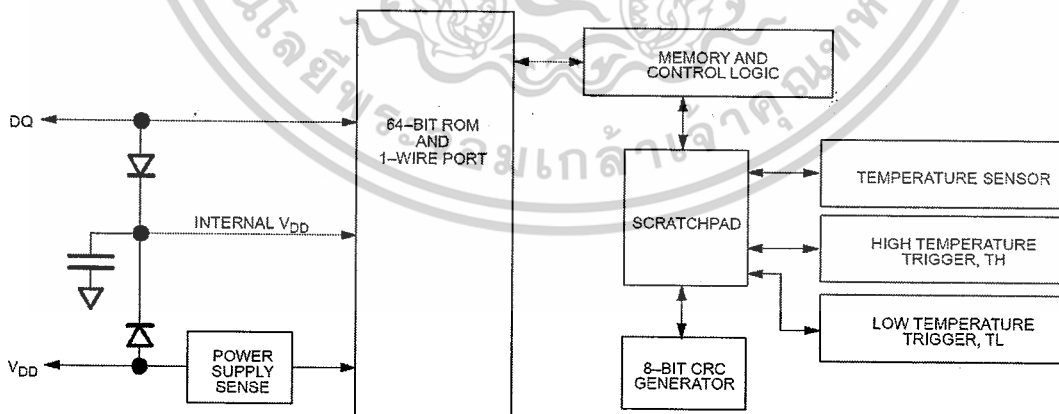
The block diagram of Figure 1 shows the major components of the DS1820. The DS1820 has three main data components: 1) 64-bit lasered ROM, 2) temperature sensor, and 3) nonvolatile temperature alarm triggers TH and TL. The device derives its power from the 1-Wire communication line by storing energy on an internal capacitor during periods of time when the signal line is high and continues to operate off this power source during the low times of the 1-Wire line until it returns high to replenish the parasite (capacitor) supply. As an alternative, the DS1820 may also be powered from an external 5 volts supply.

Communication to the DS1820 is via a 1-Wire port. With the 1-Wire port, the memory and control functions will not be available before the ROM function protocol has been established. The master must first provide one of five ROM function commands: 1) Read ROM, 2) Match ROM, 3) Search ROM, 4) Skip ROM, or 5) Alarm Search. These commands operate on the 64-bit lasered ROM portion of each device and can single out

a specific device if many are present on the 1-Wire line as well as indicate to the Bus Master how many and what types of devices are present. After a ROM function sequence has been successfully executed, the memory and control functions are accessible and the master may then provide any one of the six memory and control function commands.

One control function command instructs the DS1820 to perform a temperature measurement. The result of this measurement will be placed in the DS1820's scratchpad memory, and may be read by issuing a memory function command which reads the contents of the scratchpad memory. The temperature alarm triggers TH and TL consist of one byte EEPROM each. If the alarm search command is not applied to the DS1820, these registers may be used as general purpose user memory. Writing TH and TL is done using a memory function command. Read access to these registers is through the scratchpad. All data is read and written least significant bit first.

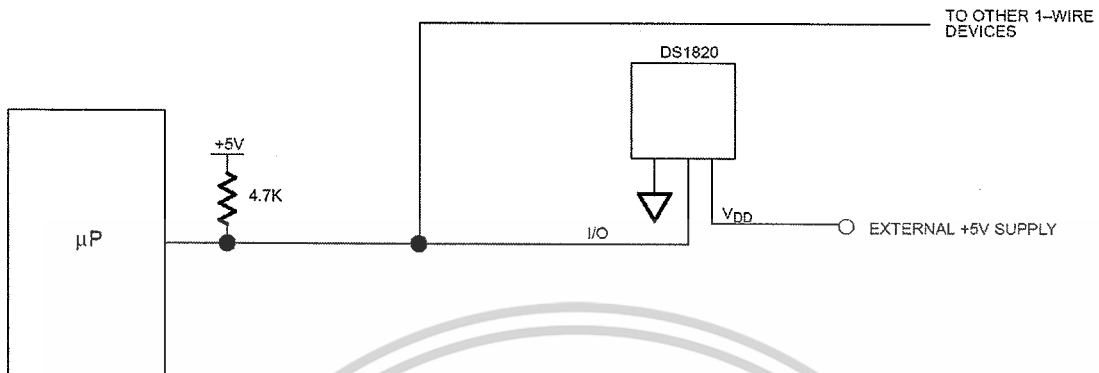
DS1820 BLOCK DIAGRAM Figure 1



030598 2/27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

USING V_{DD} TO SUPPLY TEMPERATURE CONVERSION CURRENT Figure 3



OPERATION – MEASURING TEMPERATURE

The DS1820 measures temperature through the use of an on-board proprietary temperature measurement technique. A block diagram of the temperature measurement circuitry is shown in Figure 4.

The DS1820 measures temperature by counting the number of clock cycles that an oscillator with a low temperature coefficient goes through during a gate period determined by a high temperature coefficient oscillator. The counter is preset with a base count that corresponds to -55°C . If the counter reaches zero before the gate period is over, the temperature register, which is also preset to the -55°C value, is incremented, indicating that the temperature is higher than -55°C .

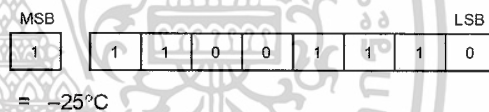
At the same time, the counter is then preset with a value determined by the slope accumulator circuitry. This circuitry is needed to compensate for the parabolic behavior of the oscillators over temperature. The counter is then clocked again until it reaches zero. If the gate period is still not finished, then this process repeats.

The slope accumulator is used to compensate for the non-linear behavior of the oscillators over temperature, yielding a high resolution temperature measurement. This is done by changing the number of counts necessary for the counter to go through for each incremental degree in temperature. To obtain the desired resolution, therefore, both the value of the counter and the number of counts per degree C (the value of the slope accumulator) at a given temperature must be known.

Internally, this calculation is done inside the DS1820 to provide 0.5°C resolution. The temperature reading is

provided in a 16-bit, sign-extended two's complement reading. Table 1 describes the exact relationship of output data to measured temperature. The data is transmitted serially over the 1-Wire interface. The DS1820 can measure temperature over the range of -55°C to $+125^{\circ}\text{C}$ in 0.5°C increments. For Fahrenheit usage, a lookup table or conversion factor must be used.

Note that temperature is represented in the DS1820 in terms of a $1/2^{\circ}\text{C}$ LSB, yielding the following 9-bit format:

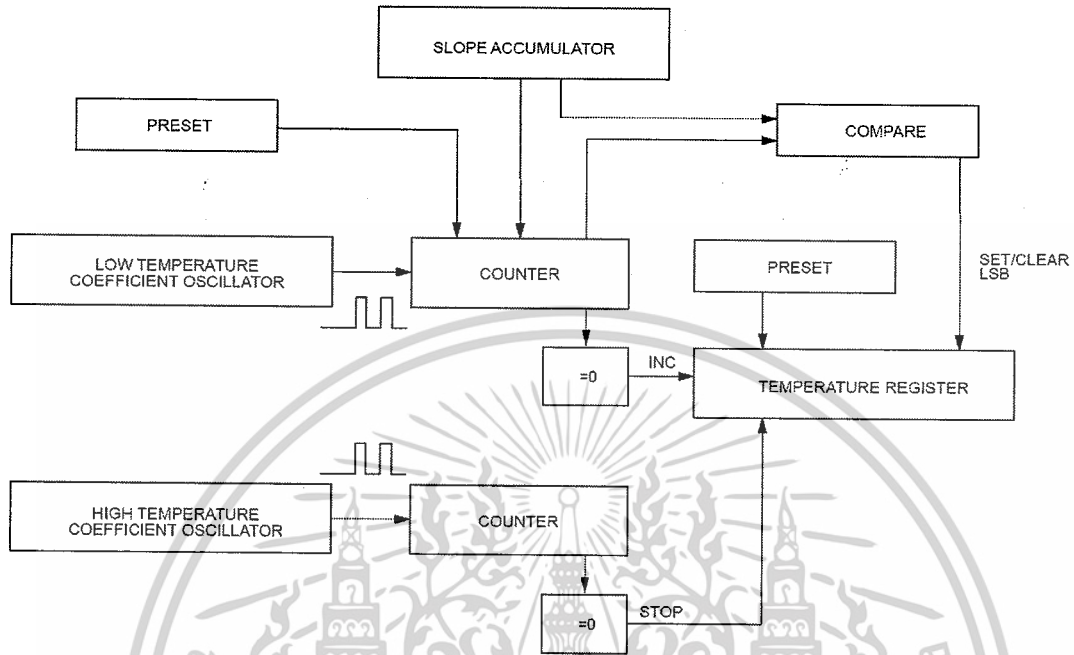


The most significant (sign) bit is duplicated into all of the bits in the upper MSB of the two-byte temperature register in memory. This "sign-extension" yields the 16-bit temperature readings as shown in Table 1.

Higher resolutions may be obtained by the following procedure. First, read the temperature, and truncate the 0.5°C bit (the LSB) from the read value. This value is TEMP_READ. The value left in the counter may then be read. This value is the count remaining (COUNT_REMAIN) after the gate period has ceased. The last value needed is the number of counts per degree C (COUNT_PER_C) at that temperature. The actual temperature may then be calculated by the user using the following:

$$\text{TEMPERATURE} = \text{TEMP_READ} - 0.25 + \frac{(\text{COUNT_PER_C} - \text{COUNT_REMAIN})}{\text{COUNT_PER_C}}$$

TEMPERATURE MEASURING CIRCUITRY Figure 4



TEMPERATURE/DATA RELATIONSHIPS Table 1

TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+125°C	00000000 11111010	00FA
+25°C	00000000 00110010	0032h
+1/2°C	00000000 00000001	0001h
+0°C	00000000 00000000	0000h
-1/2°C	11111111 11111111	FFFFh
-25°C	11111111 11001110	FFCEh
-55°C	11111111 10010010	FF92h

OPERATION – ALARM SIGNALING

After the DS1820 has performed a temperature conversion, the temperature value is compared to the trigger values stored in TH and TL. Since these registers are 8-bit only, the 0.5°C bit is ignored for comparison. The most significant bit of TH or TL directly corresponds to the sign bit of the 16-bit temperature register. If the result of a temperature measurement is higher than TH or lower than TL, an alarm flag inside the device is set.

This flag is updated with every temperature measurement. As long as the alarm flag is set, the DS1820 will respond to the alarm search command. This allows many DS1820s to be connected in parallel doing simultaneous temperature measurements. If somewhere the temperature exceeds the limits, the alarming device(s) can be identified and read immediately without having to read non-alarming devices.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AppKit: Using the SN75176 in an RS485 Network

This AppKit shows how to create a bi-directional RS485 network using the Texas Instruments SN75176 Differential Bus Transceiver chip and a Parallax BASIC Stamp single-board computer.

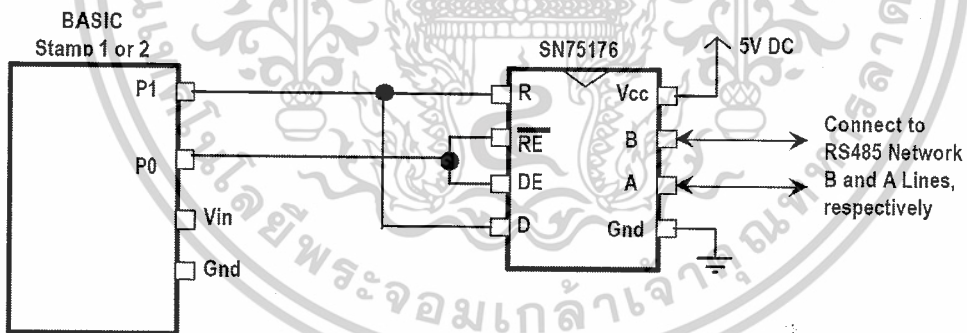
Description

RS485 is commonly used to provide strong serial signals able to withstand long cable distances (up to 4000 feet) at high baud rates in potentially noisy electrical environments. Two wires carrying an RS485 signal (the A and B lines) provide a signal base from which many devices can communicate. Twisted pair wire is recommended for long distances, but for short distances we recommend using 24 gauge wire. Up to 32 devices can be connected to an RS485 data line with the SN75176 and communicate using a data protocol. This is referred to as an RS485 network, or an RS485 drop network.

The heart of this communication is the Texas Instruments SN75176 Differential Bus Transceiver chip. This chip converts RS485 signals to RS232 TTL-level signals allowing devices that traditionally communicate over standard RS232 serial connections to communicate over a single two-wire RS485 network.

Hardware Interface

The SN75176 chip has only 8 pins. Here is how it is connected to the BASIC Stamp:



The SN75176 requires 30 mA, therefore we recommend driving only one of these chips from the Stamp's Vdd pin provided no other components require current. Setting RE (Not Receiver Enable) to Low, the R (Receiver) pin is enabled, allowing the Stamp to receive any data coming over the A and B RS485 network lines. Setting DE (Driver Enable) to High allows the Stamp to transmit data over the RS485 network. Vcc and Vdd are electronic terms with the same meaning.

PARALLAX 7

sales / technical support (916) 624-8333 • fax (916) 624-8003
stamptech@parallaxinc.com

Using the SN75176 in an RS 485 Network

Receiving and transmitting serial data using the BASIC Stamp works like this:

To Transmit data:

- 1) Set P0 to High.
- 2) Use SEROUT on P1 to transmit the data.

To Receive Data:

- 1) Set P0 to Low.
- 2) Use SERIN to receive serial data on P1

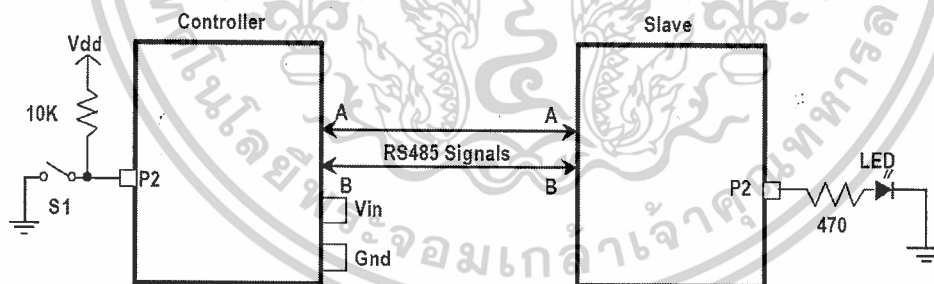
Setting Up the Network

An RS485 network operates with a controller/slave communication protocol. One controller commands everything, and one or more slaves respond to the commands. This example uses one BASIC Stamp as the controller, and one BASIC Stamp as the slave. The circuit shown below can easily be expanded to have up to 32 BASIC Stamps communicating on one RS485 network by connecting additional slaves to the network A and B lines.

Construct the network by following these steps (see figure below):

- 1) Build two identical circuits as shown below. Designate one to be the controller and the other to be the slave.
- 2) On the controller connect the supplied switch between the BASIC Stamp's P2 pin and ground, and the 10K resistor between P2 and Vcc as a pull-up resistor. This makes the line +5 V unless a button is pushed in which case it goes to 0 V.
- 3) On the slave connect the supplied LED and resistor in series between the BASIC Stamp's P2 pin and ground.
- 4) Program the controller Stamp with the supplied program "control.bas" if you are using a Stamp 1, or "control.bs2" of you are using a Stamp 2.
- 5) Program the slave BASIC Stamp with the supplied program "slave.bas" if you are using a Stamp 1, or "slave.bs2" if you are using a Stamp 2.
- 6) Flick the switch on the Controller. The Slave LED will turn on or off depending on the controller's switch position.

See the program listings for details of the software communication process.



PARALLAX 3

sales / technical support (916) 624-8333 • fax (916) 624-8003
stamptech@parallaxinc.com

Page 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Highly Flexible Voltage Comparators

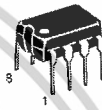
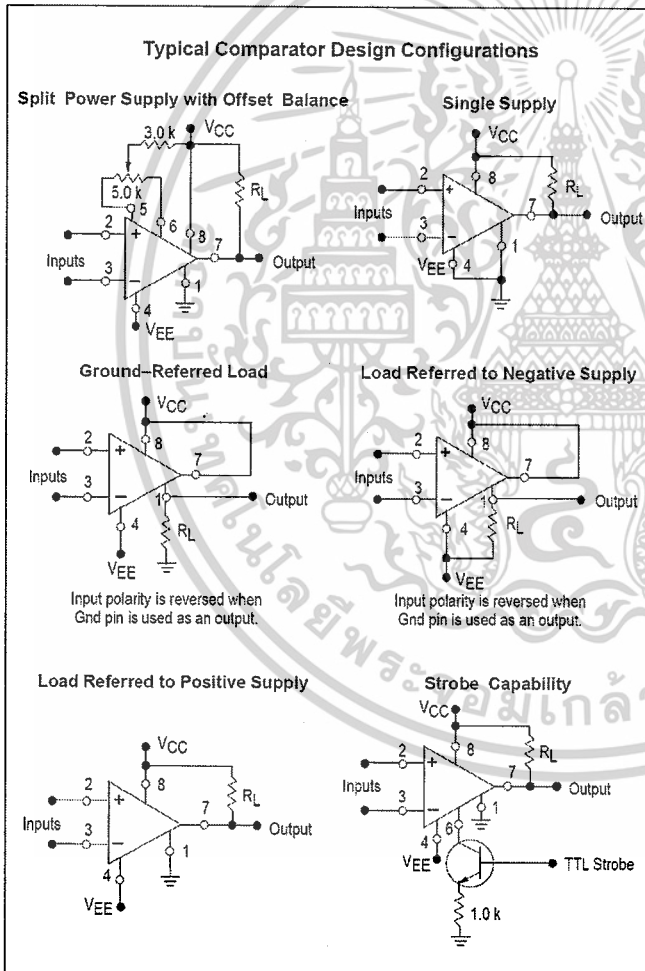
The ability to operate from a single power supply of 5.0 V to 30 V or ± 15 V split supplies, as commonly used with operational amplifiers, makes the LM211/LM311 a truly versatile comparator. Moreover, the inputs of the device can be isolated from system ground while the output can drive loads referenced either to ground, the V_{CC} or the V_{EE} supply. This flexibility makes it possible to drive DTL, RTL, TTL, or MOS logic. The output can also switch voltages to 50 V at currents to 50 mA. Thus the LM211/LM311 can be used to drive relays, lamps or solenoids.

Order this document by LM311/D

LM311 LM211

HIGH PERFORMANCE VOLTAGE COMPARATORS

SEMICONDUCTOR
TECHNICAL DATA

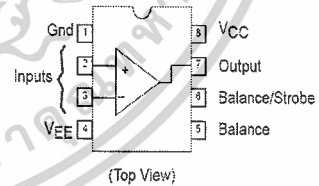


N SUFFIX
PLASTIC PACKAGE
CASE 626



D SUFFIX
PLASTIC PACKAGE
CASE 751
(SO-8)

PIN CONNECTIONS



ORDERING INFORMATION

Device	Operating Temperature Range	Package
LM211D	$T_A = 25^\circ$ to $+85^\circ\text{C}$	SO-8
LM311D LM311N	$T_A = 0^\circ$ to $+70^\circ\text{C}$	SO-8 Plastic DIP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LM311 LM211

MAXIMUM RATINGS ($T_A = +25^\circ\text{C}$, unless otherwise noted.)

Rating	Symbol	LM211	LM311	Unit
Total Supply Voltage	$V_{CC} + V_{EE} $	36	36	Vdc
Output to Negative Supply Voltage	$V_O - V_{EE}$	50	40	Vdc
Ground to Negative Supply Voltage	V_{EE}	30	30	Vdc
Input Differential Voltage	V_{ID}	± 30	± 30	Vdc
Input Voltage (Note 2)	V_{in}	± 15	± 15	Vdc
Voltage at Strobe Pin	-	V_{CC} to $V_{CC}-5$	V_{CC} to $V_{CC}-5$	Vdc
Power Dissipation and Thermal Characteristics Plastic DIP Derate Above $T_A = +25^\circ\text{C}$	P_D $1/\theta_{JA}$	625 5.0		mW mW/°C
Operating Ambient Temperature Range	T_A	-25 to +85	0 to +70	°C
Operating Junction Temperature	$T_{J(max)}$	+150	+150	°C
Storage Temperature Range	T_{stg}	-65 to +150	-65 to +150	°C

ELECTRICAL CHARACTERISTICS ($V_{CC} = +15\text{ V}$, $V_{EE} = -15\text{ V}$, $T_A = 25^\circ\text{C}$, unless otherwise noted [Note 1].)

Characteristic	Symbol	LM211			LM311			Unit
		Min	Typ	Max	Min	Typ	Max	
Input Offset Voltage (Note 3) $R_S \leq 50\text{ k}\Omega$, $T_A = +25^\circ\text{C}$ $R_S \leq 50\text{ k}\Omega$, $T_{low} \leq T_A \leq T_{high}^*$	V_{IO}	-	0.7	3.0	-	2.0	7.5	mV
Input Offset Current (Note 3) $T_A = +25^\circ\text{C}$ $T_{low} \leq T_A \leq T_{high}^*$	I_{IO}	-	1.7	10	-	1.7	50	nA
Input Bias Current $T_A = +25^\circ\text{C}$ $T_{low} \leq T_A \leq T_{high}^*$	I_{IB}	-	45	100	-	45	250	nA
Voltage Gain	A_V	40	200	-	40	200	-	V/mV
Response Time (Note 4)	-	-	200	-	-	200	-	ns
Saturation Voltage $V_{ID} \leq -5.0\text{ mV}$, $I_O = 50\text{ mA}$, $T_A = 25^\circ\text{C}$ $V_{ID} \leq -10\text{ mV}$, $I_O = 50\text{ mA}$, $T_A = 25^\circ\text{C}$ $V_{CC} \geq 4.5\text{ V}$, $V_{EE} = 0$, $T_{low} \leq T_A \leq T_{high}^*$ $V_{ID} \leq 6.0\text{ mV}$, $I_{sink} \leq 8.0\text{ mA}$ $V_{ID} \leq 10\text{ mV}$, $I_{sink} \leq 8.0\text{ mA}$	V_{OL}	-	0.75	1.5	-	0.75	1.5	V
Strobe "On" Current (Note 5)	I_S	-	3.0	-	-	3.0	-	mA
Output Leakage Current $V_{ID} \geq 5.0\text{ mV}$, $V_O = 35\text{ V}$, $T_A = 25^\circ\text{C}$, $I_{strobe} = 3.0\text{ mA}$ $V_{ID} \geq 10\text{ mV}$, $V_O = 35\text{ V}$, $T_A = 25^\circ\text{C}$, $I_{strobe} = 3.0\text{ mA}$ $V_{ID} \geq 5.0\text{ mV}$, $V_O = 35\text{ V}$, $T_{low} \leq T_A \leq T_{high}^*$	-	-	0.2	10	-	0.2	50	nA nA μA
Input Voltage Range ($T_{low} \leq T_A \leq T_{high}^*$)	V_{ICR}	-14.5	-14.7 to 13.8	+13.0	-14.5	-14.7 to 13.8	+13.0	V
Positive Supply Current	I_{CC}	-	+2.4	+6.0	-	+2.4	+7.5	mA
Negative Supply Current	I_{EE}	-	-1.3	-5.0	-	-1.3	-5.0	mA

* $T_{low} = -25^\circ\text{C}$ for LM211 $T_{high} = +85^\circ\text{C}$ for LM211= 0°C for LM311= $+70^\circ\text{C}$ for LM311

- NOTES:**
- Offset voltage, offset current and bias current specifications apply for a supply voltage range from a single 5.0 V supply up to $\pm 15\text{ V}$ supplies.
 - This rating applies for $\pm 15\text{ V}$ supplies. The positive input voltage limit is 30 V above the negative supply. The negative input voltage limit is equal to the negative supply voltage or 30 V below the positive supply, whichever is less.
 - The offset voltages and offset currents given are the maximum values required to drive the output within a volt of either supply with a 1.0 mA load. Thus, these parameters define an error band and take into account the "worst case" effects of voltage gain and input impedance.
 - The response time specified is for a 100 mV input step with 5.0 mV overdrive.
 - Do not short the strobe pin to ground; it should be current driven at 3.0 mA to 5.0 mA.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อ-สกุล	นายจิตปกรณ์ เตโชพันธ์
วัน เดือน ปีเกิด	28 กุมภาพันธ์ พ.ศ. 2526
ภูมิลำเนา	25/1 หมู่ 2 ตำบลสองคอน อำเภอแก่งคอย จังหวัดสระบุรี 18110
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนวัดสองคอนกลาง จังหวัดสระบุรี
มัธยมศึกษาตอนต้น	โรงเรียนแก่งคอย จังหวัดสระบุรี
ประกาศนียบัตรวิชาชีพ	วิทยาลัยเทคนิคสระบุรี
ประกาศนียบัตรวิชาชีพชั้นสูง	วิทยาลัยเทคนิคสระบุรี
ปริญญาตรี	สาขาวิชาวิศวกรรมโทรคมนาคม ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สจล.
คติพจน์	การที่เราเข้าใจในสิ่งที่คิดว่าเลวร้ายที่สุด บางครั้งก็ไม่ได้ ทำให้ชีวิตดีขึ้น แต่อย่างน้อยก็ทำให้เรามีความสุข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อ-สกุล	นางสาวปิยารรณ ดิศรพงศ์
วัน เดือน ปีเกิด	19 มกราคม พ.ศ. 2526
ภูมิลำเนา	154/2 หมู่ 1 ตำบลสะเตงนอก อำเภอเมือง จังหวัดยะลา 95000
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนอนุบาลยะลา จังหวัดยะลา
มัธยมศึกษาตอนต้น	โรงเรียนสตรียะลา จังหวัดยะลา
ประกาศนียบัตรวิชาชีพ	วิทยาลัยเทคนิคยะลา
ประกาศนียบัตรวิชาชีพชั้นสูง	วิทยาลัยเทคนิคยะลา
ปริญญาตรี	สาขาวิชาวิศวกรรมโทรคมนาคม ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สจล.
คติพจน์	ปัจจุบันทำเพื่อพ่อแม่และตนเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อ-สกุล	นายสุธี ทองแกมแก้ว
วัน เดือน ปีเกิด	25 ตุลาคม พ.ศ. 2525
ภูมิลำเนา	33 หมู่.1 ตำบลคลองอยู่ตะเภา อำเภอหาดใหญ่ จังหวัดสงขลา 90110
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนเทศบาล 4 (วัดคลองเรียน) จังหวัดสงขลา
มัธยมศึกษาตอนต้น	โรงเรียนหาดใหญ่วิทยาลัยสมบูรณ์กุลกันยา จังหวัดสงขลา
ประกาศนียบัตรวิชาชีพ	วิทยาลัยการอาชีพหลวงประธานราษฎร์นิกร จังหวัดสงขลา
ประกาศนียบัตรวิชาชีพชั้นสูง	วิทยาลัยเทคนิคหาดใหญ่ จังหวัดสงขลา
ปริญญาตรี	สาขาวิชาวิศวกรรมโทรคมนาคม ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สจล.
คติพจน์	จะไม่รู้จักคำว่าดี ถ้าไม่เคยทำผิด แต่ผิดแล้วต้องแก้ไข มิใช่แก้ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อ-สกุล	นายปิยพงศ์ ยี่ซ่าย
วัน เดือน ปีเกิด	26 มีนาคม พ.ศ. 2525
ภูมิลำเนา	23/1 หมู่ 4 ตำบลกระแสดินธุ์ อำเภอกระแสดินธุ์ จังหวัดสงขลา 90270
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนชุมชนวัดเชิงแส จังหวัดสงขลา
มัธยมศึกษาตอนต้น	โรงเรียนกระแสดินธุ์วิทยา จังหวัดสงขลา
ประกาศนียบัตรวิชาชีพ	โรงเรียนสงขลาเทคโนโลยี จังหวัดสงขลา
ประกาศนียบัตรวิชาชีพชั้นสูง	วิทยาลัยเทคนิคหาดใหญ่ จังหวัดสงขลา
ปริญญาตรี	สาขาวิชาวิศวกรรมโทรคมนาคม ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สจล.
คติพจน์	จงภูมิใจในผลงานของตนเอง ที่ได้มาจากความคิด ความสร้างสรรค์ และพรสวรรค์ที่มีอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้