

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

วัตต์มิเตอร์ไร้สายเก็บข้อมูลการใช้กำลังงานไฟฟ้า

WIRELESS WATTMETER DATA COLLECTION



เลขหมู่..... 72916
เลขทะเบียน..... 25 ส.ย. 2550
วัน,เดือน,ปี..... 25 ส.ย. 2550

b. 112224222
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาดามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

ผ่านการตรวจรูปเล่มแล้ว
(ลงชื่อ).....ผู้ตรวจ

ผ่านการตรวจชิ้นงานแล้ว
(ลงชื่อ).....ผู้ตรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วัดต์มิเตอร์ไร้สายเก็บข้อมูลการใช้กำลังงานไฟฟ้า
WIRELESS WATTMETER DATA COLLECTION



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง วัดต์มิเตอร์ไร้สายเก็บข้อมูลการใช้กำลังงานไฟฟ้า

WIRELESS WATTMETER DATA COLLECTION

ผู้จัดทำ

1. นางสาวสุกัญญา วิชัย 47015070

2. นายสุทธิศักดิ์ ยอดอ่อน 47015071


..... อาจารย์ที่ปรึกษา
(ผศ. เกรียงไกร วงศ์โรจนกรณ์)


..... อาจารย์ที่ปรึกษา
(รศ.ดร. สุวิทย์ ดิทธิชีวกาศ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วัดคัมมิเตอร์ไร้สายเก็บข้อมูลการใช้กำลังงานไฟฟ้า

WIRELESS WATTMETER DATA COLLECTION

โดย นางสาว สุกัญญา วิชัย 47015070

นาย . สุทธิศักดิ์ ขอดอ่อน 47015071

อาจารย์ที่ปรึกษา ผศ. เกรียงไกร วงศ์โรจนภรณ์

รศ.ดร. สุวิพล สิริทธิวิภาค

บทคัดย่อ

โครงการนี้ทำขึ้นมาเพื่อตรวจสอบและเก็บข้อมูลการใช้กำลังงานไฟฟ้า ข้อมูลการใช้กำลังงานไฟฟ้าจะประมวลผลโดยไมโครคอนโทรลเลอร์แล้วส่งข้อมูลกำลังงานไฟฟ้าทางคลื่นวิทยุไปสู่ตัวรับ ซึ่งตัวรับจะรับข้อมูลบันทึกข้อมูลส่งถ่ายข้อมูลไปยังคอมพิวเตอร์เพื่อประมวลผลต่อไป

ABSTRACT

This thesis is about testing and collecting information of power-electricity usage which is processed by microcontroller and then sent to the receiver by radio-wave signal. The data will be saved by the receiver and can be displayed by the monitor later.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ทางคณะผู้จัดทำขอขอบคุณ ผศ. เกียรติกร วงศ์โรจนภรณ์ และ รศ. ดร. สุวิมล สิทธิชีวภาค สำหรับคำแนะนำ ความคิดเห็น แนวทางต่างๆ และขอบคุณผู้ให้คำที่ปรึกษาในส่วนของเนื้อหาวิชาในส่วนนั้นๆ และอำนวยความสะดวกแก่คณะผู้จัดทำในด้านอุปกรณ์และเครื่องมือต่างๆ ในการทดลอง รวมทั้งบุคลากรที่ให้กำลังใจเสมอมา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 แนวคิดในการออกแบบ	1
1.4 ขอบเขตของโครงการ	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 กำลังงานไฟฟ้ากระแสสลับ	3
2.1.1 กำลังงานไฟฟ้าจริง (True Power)	3
2.1.2 กำลังงานไฟฟ้าที่ปรากฏ (Apparent Power)	3
2.1.3 กำลังงานไฟฟ้ารีแอกทีฟ (Reactive Power)	3
2.1.4 กำลังงานไฟฟ้า	4
2.1.5 ค่าสูงสุดหรือค่ายอดคลื่นของรูปคลื่นไซน์	4
2.1.6 ค่ายอดถึงยอดของรูปคลื่นไซน์	5
2.1.7 ค่าเฉลี่ยของรูปคลื่นไซน์	5
2.1.8 ค่า rms ของรูปคลื่นไซน์	6
2.3 นิยามค่าต่างๆ	6
2.4 หลักการพื้นฐานของออปแอมป์ (Op-Amp Basics)	9
2.4.1 การใช้งานออปแอมป์ในวงจรขยายสัญญาณ	10
2.4.2 วงจรเรียงกระแสเต็มรูป	11
2.5 โครงสร้างและสถาปัตยกรรมของไมโครคอนโทรลเลอร์ ตระกูล MCS-51	12
2.5.1 คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51	13
2.5.2 คุณสมบัติของไมโครคอนโทรลเลอร์ โดยบริษัท Atmel	14
2.6 ไมโครคอนโทรลเลอร์ตระกูล PIC	14
2.6.1 คุณสมบัติของ PIC16F877	15
2.6.2 โครงสร้างภายในชิป	16
2.6.3 หน้าที่ของพอร์ตที่ใช้งาน	16
2.6.4 พอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ PIC16F877	17
2.6.5 ชนิดของ PIC	19
2.6.6 การจัดสรรหน่วยความจำโปรแกรมของ PIC	20
2.6.7 การจัดสรรหน่วยความจำข้อมูลแรมและรีจิสเตอร์ไฟล์	20
2.6.8 ความเร็วของ PIC	21
2.7 การแปลงสัญญาณอนาล็อกเป็นดิจิทัล	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.7.1 โมดูลการแปลงสัญญาณอนาล็อกเป็นดิจิตอลของ PIC16F877	23
2.7.2 การทำงานของโมดูลการแปลงสัญญาณอนาล็อกเป็นดิจิตอล	23
2.7.3 การเลือกความถี่สัญญาณนาฬิกาของการแปลงสัญญาณ	24
2.7.4 กระบวนการแปลงสัญญาณอนาล็อกเป็นดิจิตอล	24
2.8 แอลซีดีโมดูล	24
2.8.1 ประเภทของแอลซีดีโมดูล	25
2.8.2 โครงสร้างภายในตัวควบคุมแอลซีดีโมดูล แบบอักษร	26
2.8.3 ส่วนประกอบหลักๆ แอลซีดีโมดูล	27
2.9 ไอซีฐานเวลา (Real Time Clock)	27
2.9.1 คุณสมบัติของไอซีฐานเวลา	27
2.9.2 ลักษณะการทำงานของไอซีฐานเวลา	27
2.10 การเชื่อมต่อไมโครคอนโทรลเลอร์กับ RS-232 ผ่าน MAX232	28
2.11 คีย์แพดและไอซีสแกนคีย์สวิตช์	29
2.11.1 การใช้งานคีย์แพด	29
2.11.2 การใช้งานไอซีสแกนคีย์สวิตช์ (MM74C922N)	30
2.12 เซนเซอร์กระแส (ไอซีเบอร์ ACS706ELC-20A)	30
2.13 RF-Wireless โมดูล	32
2.14 รูปแบบของการสื่อสาร	33
2.14.1 การติดต่อทางเดียว (Simplex)	33
2.14.2 การติดต่อทั้งสองทาง (Half-duplex)	33
2.14.3 การติดต่อสองทาง (Full-duplex)	33
บทที่ 3 การคำนวณและการสร้าง	35
3.1 หลักการเบื้องต้น	35
3.2 ส่วนของฮาร์ดแวร์	35
3.3 ฮาร์ดแวร์ภาคส่ง	36
3.3.1 วงจรตรวจจับแรงดัน (Voltage Sensor)	36
3.3.2 วงจรตรวจจับกระแส (Current Sensor)	40
3.3.3 การทำงานของไมโครคอนโทรลเลอร์ PIC16F877	41
3.4 ฮาร์ดแวร์ภาครับ	42
3.5 การทำงานของภาค RF-Wireless โมดูล	43
3.5.1 การทำงานใน เซตอัปโหมด	43
3.5.2 การทำงานแบบ RF Auto Direction ในรีโมต	44

สารบัญ (ต่อ)

	หน้า
3.6 แหล่งจ่ายไฟ	45
3.7 ส่วนของซอฟต์แวร์	48
3.7.1 Flow Chat การทำงานของภาคส่ง	48
3.7.2 Flow Chat การทำงานของภาครับ	49
3.7.3 Flow Chat การทำงานของการตั้งวันที่	50
3.7.4 Flow Chat การทำงานของการตั้งเวลา	51
3.7.5 Flow Chat การทำงานของการอ่านข้อมูล	52
3.7.6 Flow Chat การทำงานของการเก็บข้อมูล	53
3.7.7 Flow Chat การทำงานของการส่งข้อมูลไปยังคอมพิวเตอร์	54
บทที่ 4 การทดลองและผลการทดลอง	55
4.1 ผลการทดลองฮาร์ดแวร์	55
4.1.1 ผลการทดลองวงจรตรวจจับแรงดัน	55
4.1.2 ผลการทดลองรูปสัญญาณวงจรตรวจจับแรงดัน	56
4.1.3 ผลการทดลองวงจรตรวจจับกระแส	60
4.1.4 ผลการทดลองรูปสัญญาณวงจรตรวจจับกระแส	61
4.2 ผลการทดลองซอฟต์แวร์	65
4.2.1 การสร้างแรงดันอ้างอิง	65
4.2.2 แสดงสัญญาณที่ไม่ต่อโหลด	66
4.2.3 เวลาที่ใช้แปลงสัญญาณอนาล็อกเป็นดิจิทัล	66
4.2.4 กำลังงานไฟฟ้าขณะต่อ โหลดและไม่ต่อ โหลด	67
4.2.5 แสดงผลการทำงานของภาครับ	69
4.2.6 แสดงการต่อใช้งานจริงของภาคส่งและภาครับ	70
บทที่ 5 บทสรุปและวิจารณ์	72
5.1 การทำงานของภาครับและภาคส่ง	72
5.2 ปัญหาที่พบในการทดลอง	72
5.3 แนวทางการแก้ไขปัญหา	72
5.4 แนวทางในการพัฒนา	72
ภาคผนวก	73
กิตติกรรมประกาศ	
บรรณานุกรม	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 1.1 แสดงหลักการเบื้องต้นของโครงการงาน	2
รูปที่ 2.1 แสดงค่าสูงสุดหรือค่ายอดคลื่นของรูปคลื่น	4
รูปที่ 2.2 แสดงค่ายอดถึงยอดของรูปคลื่นไซน์	5
รูปที่ 2.3 แสดงค่าเฉลี่ยของรูปคลื่นไซน์	6
รูปที่ 2.4 แสดงค่า rms ของรูปคลื่นไซน์	6
รูปที่ 2.5 แสดงสัญลักษณ์พื้นฐานของออปแอมป์	9
รูปที่ 2.6 แสดงวงจรเทียบเคียงของออปแอมป์	10
รูปที่ 2.7 แสดงวงจรบัฟเฟอร์แรงดันไฟฟ้า	10
รูปที่ 2.8 แสดงวงจรขยายสัญญาณแบบกลับเฟส	10
รูปที่ 2.9 แสดงวงจรขยายสัญญาณแบบไม่กลับเฟส	11
รูปที่ 2.10 แสดงบล็อกไดอะแกรมหลักการทำงานและวงจร	12
รูปที่ 2.11 แสดงรายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชของ ATMEL เบอร์ AT89C52	13
รูปที่ 2.12 แสดงการจัดขามมาตรฐานของไมโครคอนโทรลเลอร์ตระกูล PIC เบอร์ PIC16F877	15
รูปที่ 2.13 แสดงรายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ตระกูล PIC เบอร์ PIC16F877	16
รูปที่ 2.14 แสดงกราฟแรงดันเอาต์พุตที่ได้จากการสัญญาณแปลงอนาล็อกเป็นดิจิทัล	22
รูปที่ 2.15 แสดงการแปลงสัญญาณอนาล็อกเป็นดิจิทัลแบบแฟลช	23
รูปที่ 2.16 แสดงลักษณะของตัวแอลซีดี โมดูลแบบ 16 ตัวอักษร 2 แถว	24
รูปที่ 2.17 แสดงขั้นตอนการควบคุมแอลซีดี โมดูล	25
รูปที่ 2.18 แสดงจอภาพแอลซีดี โมดูล แบบกราฟฟิก	26
รูปที่ 2.19 แสดงลักษณะการจับขาของไอซีฐานเวลา	27
รูปที่ 2.20 แสดงการต่อสายสัญญาณ SDA และ SCL กับอุปกรณ์ต่าง ๆ	28
รูปที่ 2.21 แสดงลักษณะการต่อใช้งาน MAX232 ร่วมกับ DB9	29
รูปที่ 2.22 แสดงลักษณะการต่อของวงจรของสวิทช์แบบเมตริกซ์หรือคีย์แพคแบบ 4 x 4	30
รูปที่ 2.23 แสดงการต่อใช้งานคีย์แพคกับไอซีสแกนคีย์สวิทช์ MM74C922N	30
รูปที่ 2.24 แสดงบล็อกไดอะแกรมการทำงานของ ACS706ELC-20A	31
รูปที่ 2.25 แสดงกราฟความเป็นเชิงเส้นระหว่างกระแสอินพุตกับแรงดันเอาต์พุต	32
รูปที่ 2.26 แสดงลักษณะตัวอุปกรณ์เซนเซอร์กระแส ACS706ELC-20A	32
รูปที่ 2.27 แสดงรูปแบบของการสื่อสาร	34
รูปที่ 3.1 แสดงบล็อกไดอะแกรมภาคชุดรับ-ส่งวัดคัมมิเตอร์ไร้สายเก็บข้อมูลการใช้กำลังงานไฟฟ้า	36
รูปที่ 3.2 แสดงการต่อวงจรตรวจจับแรงดัน	36
รูปที่ 3.3 แสดงวงจรขยายผลต่าง	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 3.4 แสดงวงจรเรียงกระแสเต็มรูป	39
รูปที่ 3.5 แสดงวงจรขยายแบบกลับเฟส ที่มีไดโอดต่อร่วมด้วย	39
รูปที่ 3.6 แสดงวงจรขยายผลบวก	40
รูปที่ 3.7 แสดงลักษณะการต่อวงจรตรวจจับกระแส	41
รูปที่ 3.8 แสดงลักษณะ โหมดการทำงานหน้าหลักของฟังก์ชันการทำงานต่างๆ	43
รูปที่ 3.9 แสดงค่าต่างๆ ที่ กำหนด ของ RF-Wireless โมดูล	44
รูปที่ 3.10 แสดงการต่อสาย RS-232 เพื่อใช้งานกับ RF-Wireless โมดูล ในโหมด Auto-Direction	44
รูปที่ 3.11 แสดงวงจรแหล่งจ่ายไฟของภาครับ	45
รูปที่ 3.12 แสดงวงจรแหล่งจ่ายไฟของภาคส่ง	45
รูปที่ 3.13 แสดงวงจรภาคส่งวัดคีมิตเตอร์ไร้สายเก็บข้อมูลการใช้กำลังงานไฟฟ้า	46
รูปที่ 3.14 แสดงวงจรภาครับวัดคีมิตเตอร์ไร้สายเก็บข้อมูลการใช้กำลังงานไฟฟ้า	47
รูปที่ 3.15 แสดง Flow Chat การทำงานของภาคส่ง	48
รูปที่ 3.16 แสดง Flow Chat การทำงานของภาครับ	49
รูปที่ 3.17 แสดง Flow Chat การทำงานของการวันที่	50
รูปที่ 3.18 แสดง Flow Chat การทำงานของการตั้งเวลา	51
รูปที่ 3.19 แสดง Flow Chat การทำงานของการอ่านข้อมูล	52
รูปที่ 3.20 แสดง Flow Chat การทำงานของการเก็บข้อมูล	53
รูปที่ 3.21 แสดง Flow Chat การทำงานของการส่งข้อมูลไปยังคอมพิวเตอร์	54
รูปที่ 4.1 แสดงวงจรตรวจจับแรงดันแบบสมบรูณ์	55
รูปที่ 4.2 แสดงสัญญาณแรงดันระหว่าง V1 กับ V2 เมื่อป้อนแรงดันอินพุต 220 โวลต์	57
รูปที่ 4.3 แสดงสัญญาณเอาต์พุตของวงจรเรียงกระแสเต็มรูป เมื่อป้อนแรงดันอินพุต 220 โวลต์	57
รูปที่ 4.4 แสดงสัญญาณแรงดันระหว่าง V1 กับ V2 เมื่อป้อนแรงดันอินพุต 200 โวลต์	58
รูปที่ 4.5 แสดงสัญญาณเอาต์พุตของวงจรเรียงกระแสเต็มรูป เมื่อป้อนแรงดันอินพุต 200 โวลต์	58
รูปที่ 4.6 แสดงสัญญาณแรงดันระหว่าง V1 กับ V2 เมื่อป้อนแรงดันอินพุต 180 โวลต์	59
รูปที่ 4.7 แสดงสัญญาณเอาต์พุตวงจรเรียงกระแสเต็มรูป เมื่อป้อนแรงดันอินพุต 180 โวลต์	59
รูปที่ 4.8 แสดงความเป็นเชิงเส้นของการตรวจจับแรงดัน	60
รูปที่ 4.9 แสดงวงจรตรวจจับกระแสแบบสมบรูณ์	60
รูปที่ 4.10 แสดงสัญญาณแรงดันเอาต์พุตวงจรตรวจจับกระแส เมื่อต่อ โหลด 100 วัตต์	61
รูปที่ 4.11 แสดงสัญญาณเอาต์พุตไฟฟ้ากระแสตรงที่ระดับอ้างอิง 2.5 โวลต์ เมื่อต่อโหลดไฟเป็น โหลด	62
รูปที่ 4.12 แสดงสัญญาณแรงดันเอาต์พุตวงจรตรวจจับกระแส เมื่อต่อ โหลด 635 วัตต์	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 4.13 แสดงสัญญาณเอาต์พุตไฟฟ้ากระแสตรงที่ระดับอ้างอิง 2.5 โวลต์ เมื่อต่อกาน้ำร้อนเป็น โหลด	63
รูปที่ 4.14 แสดงสัญญาณแรงดันเอาต์พุตวงจรตรวจจับกระแส เมื่อต่อโหลด 735 วัตต์	63
รูปที่ 4.15 แสดงสัญญาณเอาต์พุตไฟฟ้ากระแสตรงที่ระดับอ้างอิง 2.5 โวลต์ เมื่อต่อกาน้ำร้อนและ หลอดไฟเป็นโหลด	64
รูปที่ 4.16 แสดงสัญญาณแรงดันเอาต์พุตวงจรตรวจจับกระแส เมื่อไม่มีการต่อโหลด	64
รูปที่ 4.17 แสดงสัญญาณเอาต์พุตไฟฟ้ากระแสตรงที่ระดับอ้างอิง 2.5 โวลต์ เมื่อไม่มีการต่อโหลด	65
รูปที่ 4.18 แสดงตัวอย่างการสุ่มค่าสูงสุดในแต่ละครั้งของแรงดันอ้างอิง 2.5 โวลต์	65
รูปที่ 4.19 แสดงการทำงานของโปรแกรมขณะไม่ต่อโหลด	66
รูปที่ 4.20 แสดงเวลาที่ใช้ในกระบวนการสุ่มสัญญาณ 2 คาบเวลา	67
รูปที่ 4.21 แสดงกำลังงานไฟฟ้าเมื่อเปิดพัดลมปุม 1 หลังจากนั้นปิดพัดลม	67
รูปที่ 4.22 แสดงกำลังงานไฟฟ้าเมื่อต่อเตาเริด	68
รูปที่ 4.23 แสดงกำลังงานไฟฟ้าขณะไม่ต่อโหลด	68
รูปที่ 4.24 แสดงการทำงานหน้าแรกของภาครับ	69
รูปที่ 4.25 แสดงการทำงานการตั้งเวลาและวันที่	69
รูปที่ 4.26 แสดงการทำงานให้เลือกเก็บหรืออ่านข้อมูล	69
รูปที่ 4.27 แสดงการรับค่ากำลังงานไฟฟ้าจากตัวส่ง	70
รูปที่ 4.28 แสดงการต่อวงจรใช้งานจริงภาคส่ง	70
รูปที่ 4.29 แสดงอุปกรณ์ที่เสร็จสมบูรณ์ของภาคส่ง	71
รูปที่ 4.30 แสดงการต่อวงจรใช้งานจริงภาครับ	71
รูปที่ 4.31 แสดงอุปกรณ์ที่เสร็จสมบูรณ์ของภาครับ	71

สารบัญตาราง

	หน้า
ตารางที่ 2.1 Versions of 8051 from Atmel (All ROM Flash)	14
ตารางที่ 2.2 Various Speeds of 8051 from Atmel	14
ตารางที่ 4.1 แสดงผลการทดลองวงจรตรวจจับแรงดัน	56
ตารางที่ 4.2 แสดงผลการทดลองวงจรตรวจจับกระแส	61



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ในสภาวะปัจจุบันได้มีการนำพลังงานในรูปแบบต่างๆมาใช้กันอย่างแพร่หลาย เช่น พลังงานลม พลังงานแสงอาทิตย์ (Solar Cell) หรือพลังงานรูปแบบอื่นๆ แต่จะคิดเป็นส่วนน้อยเมื่อเทียบกับพลังงานไฟฟ้า ซึ่งมีการผลิตขึ้นมาใช้กันนานแล้ว โดยในการผลิตพลังงานไฟฟ้านั้นจำเป็นต้องใช้วัตถุดิบต่างๆในการผลิต วัตถุดิบในการผลิตพลังงานไฟฟ้านั้นไม่ว่าจะเป็น ถ่านหิน น้ำมัน และแก๊สธรรมชาติ กำลังมีปริมาณลดลงอย่างต่อเนื่อง ในเวลาอีกไม่นานนี้อาจจะถึงสภาวะขาดแคลน ดังนั้นในปัจจุบันหลายฝ่ายกำลังหันมาให้ความสำคัญในเรื่องการอนุรักษ์พลังงานไฟฟ้า โดยจะเห็นได้จากมีการรณรงค์ให้ใช้วิธีการต่างๆ เพื่อช่วยกันลดปริมาณการใช้พลังงานไฟฟ้าในส่วนที่ฟุ่มเฟือยลง ซึ่งจะทำให้สามารถที่จะประหยัดรายจ่ายในส่วนนี้ได้

ในเมื่อทุกคนจำเป็นต้องใช้เครื่องใช้ไฟฟ้าเพื่ออำนวยความสะดวกในการดำรงชีวิตในแต่ละวัน ดังนั้นเราคงปฏิเสธไม่ได้ว่าเครื่องใช้ไฟฟ้ามีส่วนเกี่ยวข้องกับกิจกรรมที่เราทำเกือบทุกเรื่อง ซึ่งมีความจำเป็นมาก จึงได้มีการผลิตเครื่องมือสำหรับวัดกำลังงานไฟฟ้ามาใช้ ดังนั้นจึงต้องการค้นคว้าและผลิตเป็นต้นแบบที่มีต้นทุนต่ำ โดยนำไปประยุกต์ใช้งานได้หลายประเภท ซึ่งสามารถพัฒนาต่อไปได้อีก

1.2 วัตถุประสงค์ของโครงการ

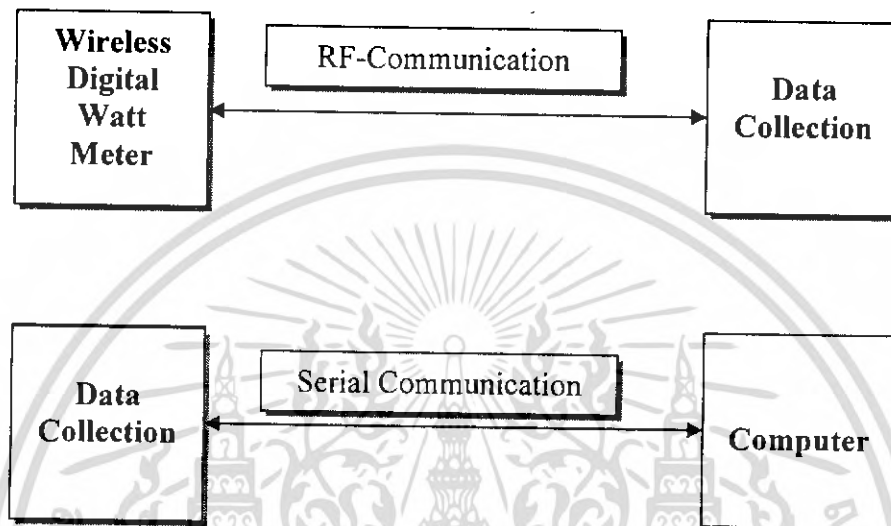
- 1.2.1 เพื่อช่วยเป็นแรงจูงใจในการอนุรักษ์พลังงานไฟฟ้า
- 1.2.2 เพื่อสังเกตว่าอุปกรณ์ไฟฟ้าใช้กำลังงานไฟฟ้าตามปกติหรือไม่
- 1.2.3 เพื่อสร้างแนวคิดในการออกแบบอุปกรณ์ที่มีประโยชน์ต่อสังคม
- 1.2.4 เพื่อสร้างวัดคีมเตอร์ที่สามารถเก็บข้อมูลการใช้กำลังงานไฟฟ้าแบบไร้สายที่มีความสะดวกและทันสมัยได้

1.3 แนวคิดในการออกแบบ

แนวทางการออกแบบของโครงการเล่มนี้ สร้างขึ้นมาเพื่อให้ผู้ใช้งานสามารถตรวจสอบค่ากำลังงานไฟฟ้าได้อย่างสะดวกด้วยการเสียบสายเครื่องใช้ไฟฟ้าเข้ากับชิ้นงานที่สร้างขึ้น หรือใช้เป็นวัดคีมเตอร์แบบดิจิทัลที่ใช้ตามหอพัก ซึ่งอุปกรณ์จะเก็บข้อมูลการใช้กำลังงานไฟฟ้าเทียบกับเวลาไว้ จากนั้นอุปกรณ์ชุดส่งจะทำการส่งข้อมูลดิจิทัลที่มีการมอดูเลตแบบจีโอเฟอสเคต (Gaussian Frequency Shift Keying) เป็นการส่งแบบไร้สายด้วยคลื่นวิทยุความถี่ 2.4 กิกะเฮิรตซ์ ทำให้การติดตั้งสะดวกและทันสมัย จากแนวคิดของกำลังงานไฟฟ้ากระแสตรง $P=VI$ นำมาวัดกำลังงานไฟฟ้ากระแสสลับที่ใช้อยู่ทั่วไป จากสูตร $S=VI$ คือกำลังงานไฟฟ้าที่ปรากฏ และ $P=VI\cos\phi$ เป็นกำลังงานไฟฟ้าที่ใช้อยู่จริง ซึ่งมุมระหว่างกระแสและแรงดันจะมีค่าต่างกันเพียงเล็กน้อย ดังนั้นค่ากำลังงานไฟฟ้าที่ปรากฏและค่ากำลังงานไฟฟ้าที่ใช้อยู่จริงจึงมีค่าใกล้เคียงกัน จึงนำแนวคิดนี้มาใช้ในการออกแบบโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 1.1 สามารถพิจารณาได้ว่าในโครงการนี้จะมีส่วนประกอบที่สำคัญอยู่สองส่วนคือ ส่วนของฮาร์ดแวร์และส่วนของซอฟต์แวร์ โดยในส่วนของฮาร์ดแวร์จะประกอบด้วย เครื่องวัดกำลังงานไฟฟ้าดิจิทัลแบบไร้สาย ส่วนของซอฟต์แวร์จะประกอบด้วยโปรแกรมการควบคุมต่างๆและการแสดงผลทางคอมพิวเตอร์สามารถแสดงดังรูปที่ 1.1



รูปที่ 1.1 แสดงหลักการเบื้องต้นของโครงการ

1.4 ขอบเขตของโครงการ

โครงการนี้เป็นการศึกษาและออกแบบการสร้างวัดคิมิเตอร์แบบดิจิทัล ใช้สำหรับไฟฟ้าระบบ 1 เฟส 2 สาย มีแรงเคลื่อนไฟฟ้าเท่ากับ 220 โวลต์ และย่านการวัดอยู่ในช่วงระหว่าง 0 ถึง 15 แอมแปร์

- 1.4.1 สามารถตรวจจับกระแสและแรงดันที่ไหลได้
- 1.4.2 สามารถแสดงปริมาณของกำลังงานไฟฟ้าได้
- 1.4.3 สามารถรับ-ส่งข้อมูลดิจิทัลแบบไร้สายด้วยคลื่นวิทยุความถี่ 2.4 กิกะเฮิร์ต ได้
- 1.4.4 สามารถเก็บข้อมูลการใช้กำลังงานไฟฟ้าและส่งถ่ายข้อมูลไปยังคอมพิวเตอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 กำลังงานไฟฟ้ากระแสสลับ

การพิจารณาและคำนวณเกี่ยวกับกำลังงานไฟฟ้ากระแสสลับ ส่วนใหญ่จะเกี่ยวข้องกับการใช้งานในวงจรไฟฟ้ากระแสสลับที่แท้จริงในทางปฏิบัติ โดยมีเรื่องการแก้ค่าเพาเวอร์แฟกเตอร์ของวงจรให้สูงขึ้นเพื่อต้องการให้กำลังงานไฟฟ้าที่ใช้ไปลดลงนั้นจึงเป็นเรื่องที่สำคัญมาก เพราะจะสามารถนำไปใช้กันมากในโรงงานอุตสาหกรรมขนาดใหญ่ที่ใช้กำลังงานไฟฟ้ามาก ๆ เพื่อต้องการลดกำลังไฟฟ้าที่สูญเสียไปในรูปของรีแอกทีฟ อย่างไรก็ตาม ในการพิจารณาและคำนวณเกี่ยวกับค่ากำลังงานไฟฟ้ากระแสสลับจะแบ่งการพิจารณาออกเป็น 3 ลักษณะเช่นเดียวกัน ดังนี้คือ

2.1.1 กำลังงานไฟฟ้าจริง (True Power)

กำลังงานไฟฟ้าจริงหมายถึง กำลังงานไฟฟ้าที่แท้จริงซึ่งทำให้เกิดกำลังงานหรือเกิดงานขึ้นมาในวงจร ซึ่งมีค่าเท่ากับ $VI \cos \phi$ และใช้ตัว P เป็นสัญลักษณ์ โดยสามารถเขียนเป็นสมการได้ดังนี้คือ

$$P = VI \cos \phi \quad (2.1)$$

ในที่นี้

P = กำลังงานไฟฟ้าจริง (วัตต์)

V = แรงดันไฟฟ้า (โวลต์)

I = กระแสไฟฟ้า (แอมแปร์)

$\cos \phi$ = \cos มุมเฟสของวงจร = เพาเวอร์แฟกเตอร์

2.1.2 กำลังงานไฟฟ้าที่ปรากฏ (Apparent Power)

กำลังงานไฟฟ้าที่ปรากฏหมายถึง ค่ากำลังงานไฟฟ้าที่จ่ายให้แก่วงจรและเป็นค่ากำลังงานไฟฟ้าที่ปรากฏให้เห็นทางโวลต์มิเตอร์และแอมมิเตอร์ กล่าวคือเป็นกำลังงานไฟฟ้าที่ได้จากการคูณแรงดันไฟฟ้าและกระแสที่วัดได้จากโวลต์มิเตอร์และแอมมิเตอร์ ซึ่งมีค่าเท่ากับ VI โดยใช้ตัวอักษร S เป็นสัญลักษณ์เขียนแทน สามารถเขียนเป็นสมการได้ดังนี้คือ

$$S = VI \quad (2.2)$$

ในที่นี้

S = กำลังงานไฟฟ้าที่ปรากฏ (โวลต์แอมป์)

2.1.3 กำลังงานไฟฟารีแอกทีฟ (Reactive Power)

กำลังงานไฟฟารีแอกทีฟหมายถึง กำลังงานไฟฟ้าที่สูญเสียไปในวงจร จะอยู่ในรูปของรีแอกทีฟซึ่งเป็นกำลังงานไฟฟ้าที่ไม่สามารถนำไปใช้ให้เกิดประโยชน์ได้ กล่าวคือเป็นกำลังงานไฟฟ้าที่ไม่สามารถทำให้เกิดงานหรือเกิดกำลังงานไฟฟ้าได้นั่นเอง มีค่าเท่ากับ $VI \sin \phi$ โดยใช้ตัวอักษร Q เป็นสัญลักษณ์เขียนแทน ซึ่งสามารถเขียนเป็นสมการได้ดังนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ (2.3) การคำนวณกำลังงานไฟฟ้าจริง $P = VI \cos \phi$ ไม่สามารถทำได้ถ้าหากไม่ทราบค่าของ $\cos \phi$ ซึ่งสามารถหาได้จากค่าของ $\sin \phi$ โดยใช้ความสัมพันธ์ $\sin^2 \phi + \cos^2 \phi = 1$ ดังนั้นหากทราบค่าของ $\sin \phi$ ก็สามารถหาค่าของ $\cos \phi$ ได้ อย่างไรก็ตาม การคำนวณกำลังงานไฟฟ้าที่ปรากฏ $S = VI$ นั้นสามารถทำได้โดยไม่ต้องทราบค่าของ $\cos \phi$ หรือ $\sin \phi$ เลยก็ตาม อย่างไรก็ตาม การคำนวณกำลังงานไฟฟารีแอกทีฟ $Q = VI \sin \phi$ นั้นจำเป็นต้องทราบค่าของ $\sin \phi$ หรือ $\cos \phi$ เสียก่อนจึงจะสามารถคำนวณค่าของ Q ได้

ในที่นี้ $Q =$ กำลังงานไฟฟ้ารีแอกทีฟ (วาร์)
 $\sin \phi = \sin$ มุมเฟสของวงจร

ข้อสังเกต หน่วยของกำลังงานไฟฟ้ารีแอกทีฟ คือ วาร์

2.1.4 กำลังงานไฟฟ้า

นิยามคือ ค่าของพลังงานที่ใช้ในช่วงเวลาหนึ่ง มีสูตรดังนี้ คือ

$$P = \frac{dW}{dt} = VI \quad (2.4)$$

ในที่นี้ $W =$ พลังงาน (จูล)
 $t =$ เวลา (วินาที)

Watt-hour power หมายถึง ค่ากำลังงานไฟฟ้าที่คูณเทียบกับเวลา โดยใช้หน่วยเป็นชั่วโมง สำหรับในประเทศไทยใช้หน่วยนี้ในการคิดคำนวณค่าไฟฟ้า

$$P_{\text{watt-hour}} = P_{\text{avg}} \times t_{\text{hour}} \quad (2.5)$$

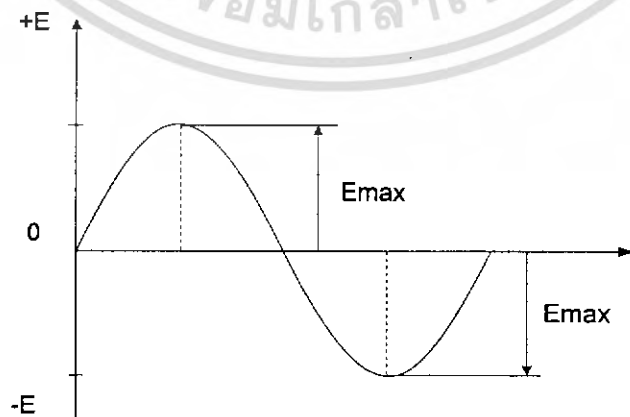
Average Power หมายถึงค่าเฉลี่ยของกำลังงาน โดยคิดจากสูตร

$$P = \frac{1}{T} \int_{t+T}^t P dt \quad (2.6)$$

2.1.5 ค่าสูงสุดหรือค่ายอดคลื่นของรูปคลื่นไซน์

ค่าสูงสุดหรือค่ายอดคลื่นของรูปคลื่นไซน์หมายถึง ขนาดที่สูงสุดทั้งทางด้านบวกและทางด้านลบ เมื่อเทียบกับแกนนอน (ที่แกนนอนขนาดเท่าศูนย์)

รูปคลื่นไซน์ที่อยู่ในลักษณะสมมาตร จะเห็นได้ว่าครึ่งไซเคิลบวกและครึ่งไซเคิลลบจะมีลักษณะเหมือนกันทุกประการ ดังนั้นค่ายอดคลื่นในทางด้านบวกและทางด้านลบจะมีขนาดเดียวกัน ดังรูปที่ 2.1 แสดงให้เห็นค่าสูงสุดหรือค่ายอดคลื่นของรูปคลื่นไซน์ที่มีค่าเท่ากันทั้งในครึ่งไซเคิลบวกและลบ



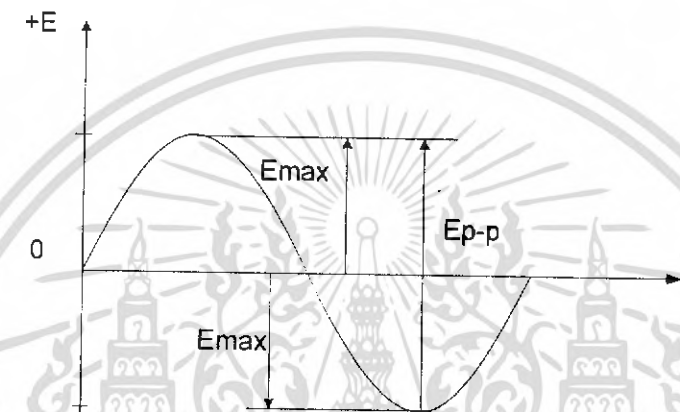
รูปที่ 2.1 แสดงค่าสูงสุดหรือค่ายอดคลื่นของรูปคลื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาค้นคว้าเท่านั้น เมื่อผู้ผู้ใดนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6 ค่ายอดถึงยอดของรูปคลื่นไซน์

ค่ายอดถึงยอดของรูปคลื่นไซน์ (Peak to Peak Value) เป็นผลรวมขนาดของรูปคลื่นทั้งในครึ่งไซเคิลบวกและครึ่งไซเคิลลบ ซึ่งพิจารณาจากค่ายอดคลื่นทางด้านบวกมายังค่ายอดคลื่นทางด้านลบ แสดงดังรูปที่ 2.2 จะได้ว่าค่าจากยอดถึงยอดคือ

$$\begin{aligned} E_{p-p} &= E_{\max}(\text{ครึ่งไซเคิลบวก}) - E_{\max}(\text{ครึ่งไซเคิลลบ}) \\ &= 2 E_{\max} \end{aligned} \quad (2.7)$$



รูปที่ 2.2 แสดงค่ายอดถึงยอดของรูปคลื่นไซน์

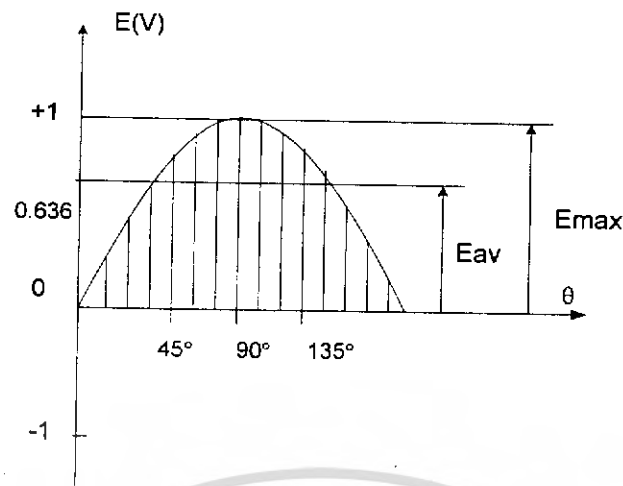
2.1.7 ค่าเฉลี่ยของรูปคลื่นไซน์

ค่าเฉลี่ยหรือค่าระหว่างกลางในหนึ่งไซเคิลของรูปคลื่นไซน์ที่แท้จริงนั้นมีค่าเท่ากับศูนย์ ดังนั้นค่าเฉลี่ยที่แท้จริงของคลื่นรูปไซน์จึงไม่มีใช้ในทางปฏิบัติ

การหาค่าเฉลี่ยของรูปคลื่นไซน์ทั้งในครึ่งไซเคิลบวกและครึ่งไซเคิลลบนั้น ค่าที่หามาได้สามารถนำไปใช้ประโยชน์ในการสร้างเครื่องวัดทางไฟฟ้ากระแสสลับได้ ปกติแล้วเครื่องวัดไฟฟ้ากระแสสลับจะประกอบไปด้วยเครื่องวัดไฟฟ้ากระแสตรงที่มีการเคลื่อนที่ของขดลวดในตำแหน่งคงที่ แล้วผ่านวงจรเรกติไฟเออร์ ทำหน้าที่แปลงไฟฟ้ากระแสสลับเป็นไฟฟ้ากระแสตรงครึ่งคลื่นแล้วเข็มของเครื่องวัดที่เคลื่อนที่ไปบนสเกลจะชี้ค่าเฉลี่ยของขนาดรูปคลื่นครึ่งคลื่นที่ผ่านวงจรเรกติไฟเออร์ดังกล่าวข้างต้น

ค่าเฉลี่ยทั้งหมดของรูปคลื่นไซน์ที่กล่าวมาหมายถึงค่าเฉลี่ยขนาดของครึ่งไซเคิลบวกหรือครึ่งไซเคิลลบ จากรูปที่ 2.3 แสดงถึงการหาค่าเฉลี่ยของรูปคลื่นไซน์ในครึ่งไซเคิลบวก ส่วนในครึ่งไซเคิลลบนั้นก็มีหลักการพิจารณาเดียวกัน

$$E_{av} = 0.636 E_{\max} \quad (2.8)$$



รูปที่ 2.3 แสดงค่าเฉลี่ยของรูปคลื่นไซน์

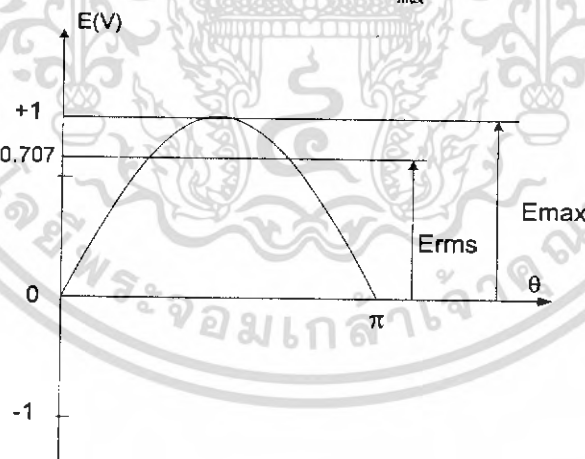
2.1.8 ค่า rms ของรูปคลื่นไซน์

กำลังงานไฟฟ้าที่ได้จากกระแสหรือแรงดันของไฟฟ้ากระแสตรงที่เป็นค่าเรียบและค่าคงที่จะมีค่าเท่ากับกำลังงานไฟฟ้าที่ได้จากไฟฟ้ากระแสตรงหรือแรงดันรูปคลื่นไซน์ของไฟฟ้ากระแสสลับ ซึ่งค่าของรูปคลื่นไซน์ดังกล่าวนี้ก็คือ ค่า rms หรือค่าประสิทธิผล (Effective Value) ของกระแสหรือแรงดันของไฟฟ้ากระแสสลับนั่นเอง

$$E_{rms} = \frac{1}{\sqrt{2}} E_{max}$$

$$0.707 E_{max}$$

(2.9)



รูปที่ 2.4 แสดงค่า rms ของรูปคลื่นไซน์

2.3 นิยามค่าต่างๆ

Watt-hour

วัตต์ต่อชั่วโมงมีสัญลักษณ์คือ Wh เป็นหน่วยของพลังงานหนึ่งวัตต์ที่ใช้ใน 1 ชั่วโมง ค่าวัตต์ต่อชั่วโมงนี้ไม่ใช่หน่วยมาตรฐานในระบบที่เป็นทางการ แต่เรามักจะนำมาใช้ในการประยุกต์ทางไฟฟ้า

การหาค่าในหน่วยของจูล เมื่อเราทราบค่าวัตต์ต่อชั่วโมงก็สามารถนำค่าวัตต์ต่อชั่วโมงมาคูณกับ 3.6×10^6 ส่วนการหาค่าในหน่วยวัตต์ต่อชั่วโมงเมื่อทราบค่าจูล ให้นำค่าจูลมาคูณกับ 2.778×10^{-6} ขึ้นด้านการคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทั่วไปค่าพลังงานจะมีค่าเท่ากับกำลังงานไฟฟ้าคูณกับเวลา ซึ่งการกำหนดค่าพลังงานเป็นวัตต์ต่อชั่วโมงต้องใช้ค่าพลังงานเป็นหน่วยวัตต์และใช้เวลาเป็นหน่วยของชั่วโมง ค่าวัตต์ต่อชั่วโมงยากที่จะใช้ในการกำหนดพลังงานในรูปแบบอื่นทางไฟฟ้า ปริมาณของพลังงานศักย์ในแก๊สโซลีน น้ำมัน หรือถ่านหิน จะปลดปล่อยพลังงานออกมาเมื่อเชื้อเพลิงได้รับการเผาไหม้ พลังงานความร้อนจะเป็นผลมาจากการเผาไหม้ของเชื้อเพลิง

พลังงาน

พลังงานมีหลายรูปแบบทั้งในการเคลื่อนที่ทางกล ความร้อน หรือพลังงานไฟฟ้าที่ถูกสร้างขึ้น แล้วออกมา สูญเสียไป หรือได้รับกลับเป็นพลังงานทั้งสิ้น สัญลักษณ์ทั่วไปของพลังงานแทนด้วย E ในระบบมาตรฐาน (The Standard Unit) จะมีหน่วยเป็นจูล ในบางระบบเช่น ระบบ Btu จะแทนด้วย 1 Btu เท่ากับ 1,055 จูล

1 จูล คือพลังงานที่เท่ากับ 1 วัตต์ของกำลังงานสำหรับ 1 วินาที พลังงานในวงจรที่เป็นไฟฟ้ากระแสตรงจะใช้ค่าแรงดันในหน่วยของโวลต์และกระแสในหน่วยของแอมแปร์ ส่วนในวงจรที่เป็นไฟฟ้ากระแสสลับจะคิดค่าพลังงานที่ซับซ้อนกว่า สัญลักษณ์โดยทั่วไปของกระแสคือ I เมื่อแหล่งกำเนิดแรงดันที่เป็นไฟฟ้ากระแสตรงจะให้ค่ากระแสเป็นแอมแปร์สำหรับเวลาในหน่วยวินาที ดังนั้นพลังงานนี้จึงใช้หน่วยเป็นจูล คือ

$$E = VI * t \quad (2.10)$$

เมื่อกระแสในหน่วยแอมแปร์วิ่งผ่านความต้านทานในหน่วยของโอห์ม สำหรับเวลาในหน่วยของวินาทีนั้นพลังงานสูญเสียจะได้

$$E = I^2 R * t \quad (2.11)$$

เมื่อแรงดันไฟฟ้ากระแสตรงตกคร่อมตัวต้านทานที่มีหน่วยเป็นโอห์ม สำหรับเวลาในหน่วยของวินาทีนั้นพลังงานดูดซับจะได้

$$E = \frac{V^2 * t}{R} \quad (2.12)$$

วัตต์

วัตต์เป็นหน่วยมาตรฐานของกำลังงานไฟฟ้า (พลังงานต่อหน่วยของเวลา) และมีค่าเท่ากับ 1 จูลต่อวินาที ค่าวัตต์นี้ใช้ในการกำหนดอัตราของพลังงานไฟฟ้าที่สูญเสีย หรืออัตราพลังงานสนามแม่เหล็กที่แผ่ออกมา ดูดซับเข้าไป หรือสูญเสีย

$$P = VI$$

$$P = \frac{V^2}{R}$$

$$P = I^2 R$$

(2.13)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในระบบที่ใช้งานความถี่คลื่นวิทยุหรือคลื่นความถี่สูงนั้น ค่ากำลังงานไฟฟ้าจะไม่ขึ้นกับค่าความต้านทานเท่านั้นแต่จะขึ้นกับค่าความนำไฟฟ้าด้วย ดังนั้นจึงไม่สามารถใช้สูตรการคำนวณที่เป็นค่าคงที่ได้ จำเป็นต้องทำการวัดค่าโดยตรง ในกรณีที่กำลังงานสูงหรือต่ำมากทำให้อัตราการสูญเสียกำลังงานจะสูงขึ้น

กำลังงาน

กำลังงานไฟฟ้า คืออัตราการเปลี่ยนแปลงของพลังงานไฟฟ้าไปเป็นพลังงานในรูปแบบอื่น เช่น การเคลื่อนที่ ความร้อน หรือสนามแม่เหล็ก สัญลักษณ์โดยทั่วไปของกำลังงานแทนด้วย P มีหน่วยเป็นวัตต์ ในวงจรส่วนใหญ่จะใช้เป็นกิโลวัตต์ โดย 1 กิโลวัตต์ เท่ากับ 1,000 วัตต์ ส่วน 1 วัตต์เป็นผลของกำลังงานจากพลังงานที่สูญเสียหรือเปลี่ยนแปลงเท่ากับ 1 จูลต่อวินาที ค่ากำลังงานนี้บางครั้งเรียกว่า วัตต์เดจ ในไฟฟ้ากระแสตรงจะเป็นค่าแรงดันในหน่วยโวลต์และกระแสในหน่วยแอมแปร์ ซึ่งเป็นกฎโดยทั่วไป คือที่ความถี่ต่ำจะไม่มีการเก็บสะสมหรือปลดปล่อยพลังงาน ดังนั้นค่ากำลังงานจึงมีการคำนวณที่ซับซ้อน

ในวงจรไฟฟ้ากระแสตรงค่ากำลังงานจะคิดเป็นปริมาณสเกลาร์ ส่วนในวงจรไฟฟ้ากระแสสลับค่ากำลังงานจะคิดเป็นปริมาณเวกเตอร์ แต่หากเราสมมติว่าไม่มีค่าความนำไฟฟ้าเราจะใช้สูตรการคำนวณเป็นแบบไฟฟ้ากระแสตรงได้เสีย แต่ถ้าใช้แบบไฟฟ้ากระแสสลับแล้วเราจะต้องคำนวณโดยใช้เป็นค่า rms ของค่าทางไฟฟ้ากระแสสลับ

เวลา

เวลาเป็นปรากฏการณ์ที่มีความหมายทางการเปลี่ยนแปลงของสภาพแวดล้อมในจักรวาล ซึ่งตามความหมายที่แท้จริงแล้วนั้นยากที่จะอธิบายได้ หน่วยพื้นฐานในระบบ SI คือวินาที โดยใน 1 วินาทีจะมีค่าเท่ากับ 9,192,631,770 ของรอบการแผ่รังสีของซีเซียม 133 อะตอม

จูล

จูลเป็นหน่วยพลังงานมาตรฐานในวิชาอิเล็กทรอนิกส์และวิทยาศาสตร์ประยุกต์ทั่วไป 1 จูล คือพลังงานที่ปล่อยออกมาเมื่อมีแรง 1 นิวตัน ทำให้เกิดการเคลื่อนที่ 1 เมตร 1 จูล เสมือนกับ 1 วัตต์ของกำลังงานที่แผ่หรือสูญเสียใน 1 วินาที

แรงดัน

แรงดัน มีความหมายได้ทั้งเป็นพลังงานศักย์ของประจุไฟฟ้าและพลังงานศักย์ดกคร่อม ถ้าตัวนำหรือสารกึ่งตัวนำที่อยู่ระหว่าง 2 จุดซึ่งมีศักย์ที่แตกต่างกันจะมีกระแสไฟฟ้าไหลผ่าน สัญลักษณ์ที่เหมือนกันคือ V หรือ E หน่วยมาตรฐานคือโวลต์ โดย 1 โวลต์คือแรงดันที่ทำให้ประจุ 1 คูลอมบ์เคลื่อนที่ผ่านในเวลา 1 วินาที แรงดันนี้เป็นได้ทั้งไฟฟ้ากระแสตรงและไฟฟ้ากระแสสลับ แรงดันไฟฟ้ากระแสตรงจะมีทิศทางเดียวตลอดเวลา ในแรงดันไฟฟ้ากระแสสลับนั้นจะมีการเปลี่ยนแปลงทิศทางไปตามคาบของเวลา จำนวนรอบที่สมบูรณ์ต่อเวลาซึ่งจะเป็นวินาทีคือความถี่ซึ่งวัดเป็นหน่วยเฮิร์ต เฮาต์พูดของเพาเวอร์ซัพพลายที่กรองแล้วจะเป็นตัวอย่างของไฟฟ้ากระแสตรง

ความต่างศักย์จะสร้างสนามไฟฟ้าสถิตก็ต่อเมื่อไม่มีกระแสไหลผ่าน เมื่อแรงดันเพิ่มขึ้นระหว่าง 2 จุด จะถูกแบ่งโดยระยะทางที่จำเพาะ สนามไฟฟ้าสถิตที่เข้มข้นกว่าเดิมจากการเพิ่มขึ้นของความต่างศักย์ระหว่าง 2 จุด ความหนาแน่นของฟลักซ์ไฟฟ้าสถิตจะมีค่าลดลงในช่วงที่วัดมุมประจุเดียวล้อมรอบ โดยสนามไฟฟ้าสถิตที่ความจุเป็นสัดส่วนโดยตรงกับแรงดันของวัตถุต่อวัตถุที่อยู่ข้างเคียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ผู้เห็นไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

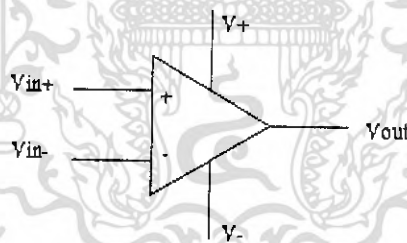
กระแสไฟฟ้า

กระแสไฟฟ้าคือการไหลของประจุไฟฟ้า โดยปกติแล้วจะเป็นอิเล็กตรอนหรืออะตอมที่สูญเสียอิเล็กตรอน จะใช้สัญลักษณ์แทนด้วย I หน่วยมาตรฐานคือแอมแปร์ 1 แอมแปร์ของกระแสไฟฟ้า คือการเคลื่อนที่ของประจุไฟฟ้าขนาด 1 คูลอมบ์ (6.24×10^{18} ประจุ) ผ่านจุดๆหนึ่งใน 1 วินาที นักฟิสิกส์กำหนดว่ากระแสไฟฟ้าไหลจากจุดประจุบวกไปยังจุดประจุลบกระแส โดยปกติที่ตัวเคลื่อนที่คืออิเล็กตรอนทำให้เกิดการเคลื่อนที่เป็นการเคลื่อนที่จากลบไปบวก กระแสไฟฟ้าสามารถเป็นได้ทั้งไฟฟ้ากระแสตรงและไฟฟ้ากระแสสลับ โดยไฟฟ้ากระแสตรงนั้นสามารถไหลผ่านจุดทุกจุดในทิศทางเดิมตลอดเวลาแม้ว่าขนาดของกระแสจะเปลี่ยนแปลงก็ตาม แต่ในไฟฟ้ากระแสสลับการไหลของประจุไฟฟ้าจะไหลกลับทิศทางกันเป็นช่วงๆ จะมีจำนวนรอบของไฟฟ้ากระแสสลับที่สมบูรณ์ต่อวินาทีคือความถี่ในหน่วยของเฮิรตซ์

กระแสต่อพื้นที่ตัดขวางคือความหนาแน่นกระแสมีหน่วยเป็นแอมแปร์ต่อพื้นที่ ซึ่งโดยทั่วไปแล้วกระแสนขนาดใหญ่ในตัวนำจะมีค่าสูงกว่าความหนาแน่นกระแส แต่อย่างไรก็ตามความหนาแน่นกระแสจะแปรตามส่วนที่แตกต่างกันทางไฟฟ้า

2.4 หลักการพื้นฐานของออปแอมป์ (Op-Amp Basics)

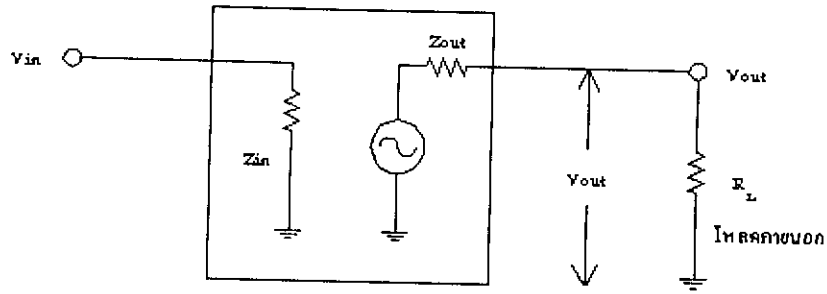
ออปแอมป์เป็นอุปกรณ์อิเล็กทรอนิกส์ที่มีอัตราขยายและอินพุตอิมพีแดนซ์ Z_i หรือ R_i สูงมาก (มักมีค่าเป็นเมกะโอห์ม) และมีเอาต์พุตอิมพีแดนซ์ Z_o หรือ R_o ต่ำ (ต่ำกว่า 100 โอห์ม) วงจรอย่างง่ายของออปแอมป์ประกอบด้วยขั้วอินพุต 2 ขั้ว และขั้วเอาต์พุต 1 ขั้ว แสดงดังรูปที่ 2.5



รูปที่ 2.5 แสดงสัญลักษณ์พื้นฐานของออปแอมป์

เราทราบว่าถ้าจ่ายสัญญาณอินพุตให้ขั้วบวกของออปแอมป์จะได้สัญญาณเอาต์พุตที่มีมุมอินเฟสกับสัญญาณอินพุต ถ้าจ่ายสัญญาณอินพุตให้กับขั้วลบของออปแอมป์จะได้สัญญาณเอาต์พุตที่มีมุมต่างเฟสกับสัญญาณอินพุต 180 องศา หรือมีมุมสัญญาณอินพุตตรงข้ามกับสัญญาณเอาต์พุต

จากคุณสมบัติด้านความต้านทานของออปแอมป์ จึงทำให้สามารถเทียบเคียงวงจรของออปแอมป์ในทางปฏิบัติได้ดังรูปที่ 2.6 โดยที่อินพุตอิมพีแดนซ์ มีค่าสูงมากจนประมาณได้ว่า กระแสอินพุตเท่ากับ 0 แอมแปร์ เอาต์พุตอิมพีแดนซ์มีค่าต่ำมากจนประมาณได้ว่าเอาต์พุตอิมพีแดนซ์เท่ากับ 0 โอห์ม



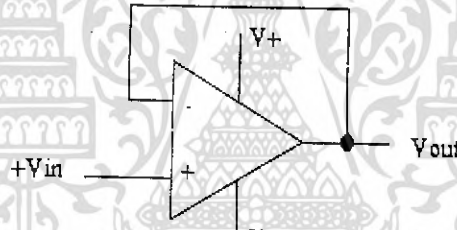
รูปที่ 2.6 แสดงวงจรเทียบเคียงของออปแอมป์

2.4.1 การใช้งานออปแอมป์ในวงจรขยายสัญญาณ

1. วงจรตามสัญญาณยูนิตี้ (Unity Follower) หรือวงจรบัฟเฟอร์แรงดันไฟฟ้า (Voltage Buffer) คือวงจรที่อัตราขยายเท่ากับ 1 และสัญญาณเอาต์พุตไม่มีการกลับเฟสกับสัญญาณอินพุต

$$A_v = \frac{V_{out}}{V_{in}} = 1 \tag{2.14}$$

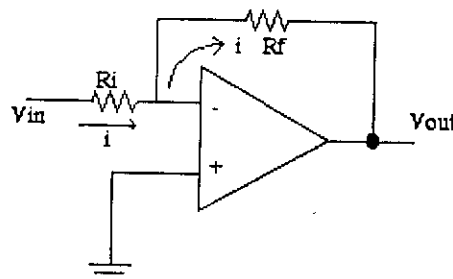
$$v_{out} = v_{in} \tag{2.15}$$



รูปที่ 2.7 แสดงวงจรบัฟเฟอร์แรงดันไฟฟ้า

2. วงจรขยายสัญญาณแบบกลับเฟส (Inverting Amplifier) เป็นวงจรขยายสัญญาณที่อัตราขยายขึ้นอยู่กับค่าของตัวต้านทานวงจรและได้รับความนิยมมากแบบหนึ่ง ความสัมพันธ์ระหว่าง v_{in+} และ v_{in-} มีค่าเท่ากับ 0 โวลต์ ดังนั้นศักย์ไฟฟ้าที่ขา v_{in-} เท่ากับ 0 โวลต์

$$v_{out} = -\frac{R_f}{R_i} v_{in} \tag{2.16}$$



รูปที่ 2.8 แสดงวงจรขยายสัญญาณแบบกลับเฟส

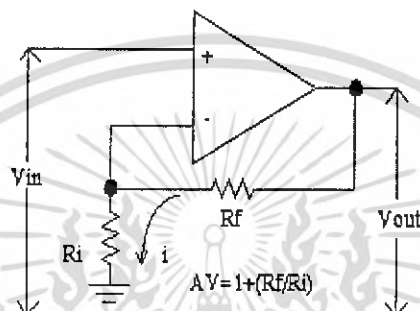
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. วงจรขยายสัญญาณแบบไม่กลับเฟส (Non-inverting Amplifier) จะหาค่า v_{in} ได้โดยใช้กฎการแบ่งแรงดันไฟฟ้า ดังนี้คือ

$$v_{in} = \frac{R_f}{R_i + R_f} v_{out} \tag{2.17}$$

นอกจากนี้จะได้

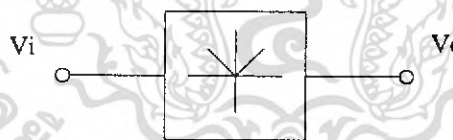
$$\frac{v_{out}}{v_{in}} = \frac{R_i + R_f}{R_i} = 1 + \frac{R_f}{R_i} \tag{2.18}$$



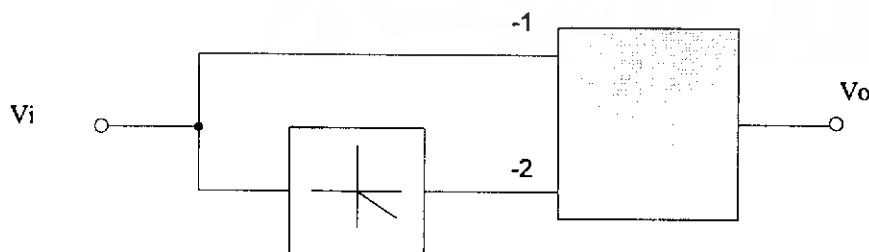
รูปที่ 2.9 แสดงวงจรขยายสัญญาณแบบไม่กลับเฟส

2.4.2 วงจรเรียงกระแสเต็มรูป

ถ้าหากต้องการวงจรเรียงกระแสเต็มรูปหรือวงจรที่ให้ค่าสัมบูรณ์ $v_o = |v_i|$ จะใช้อุปกรณ์อีกตัวหนึ่งต่อเป็นวงจรบวก ดังรูปที่ 2.10 (ก)

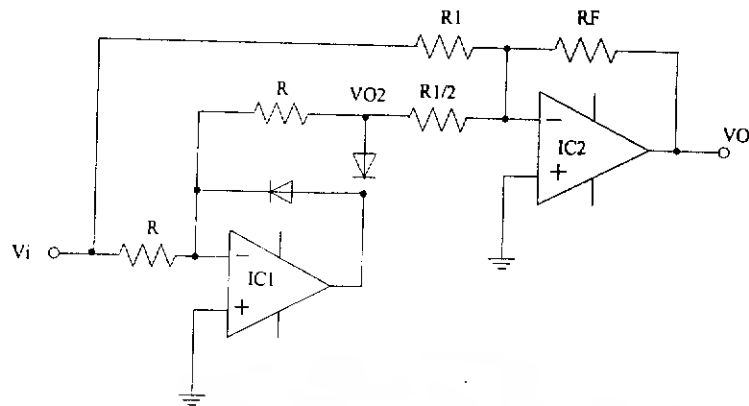


(ก) แผนภาพบล็อกของวงจรเรียงกระแสเต็มรูปหรือวงจรค่าสัมบูรณ์



(ข) แผนภาพบล็อกแสดงหลักการทำงานของวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ค) วงจร

รูปที่ 2.10 แสดงบล็อกโคอะแกรมหลักการทำงานและวงจร

หลักการทำงานของวงจรก็คือวงจรจะทำการเรียงกระแสครึ่งรูปแล้วคูณด้วย 2 ก่อนจะไปบวกกับแรงดันขาเข้า ดังรูปที่ 2.10 (ข)

$$\text{เมื่อ } v_i > 0 \quad v = -v_i \quad \text{และ} \quad v_o = -v_i - (-2v_i) = v_i$$

$$\text{เมื่อ } v_i < 0 \quad v = 0 \quad \text{และ} \quad v_o = -v_i$$

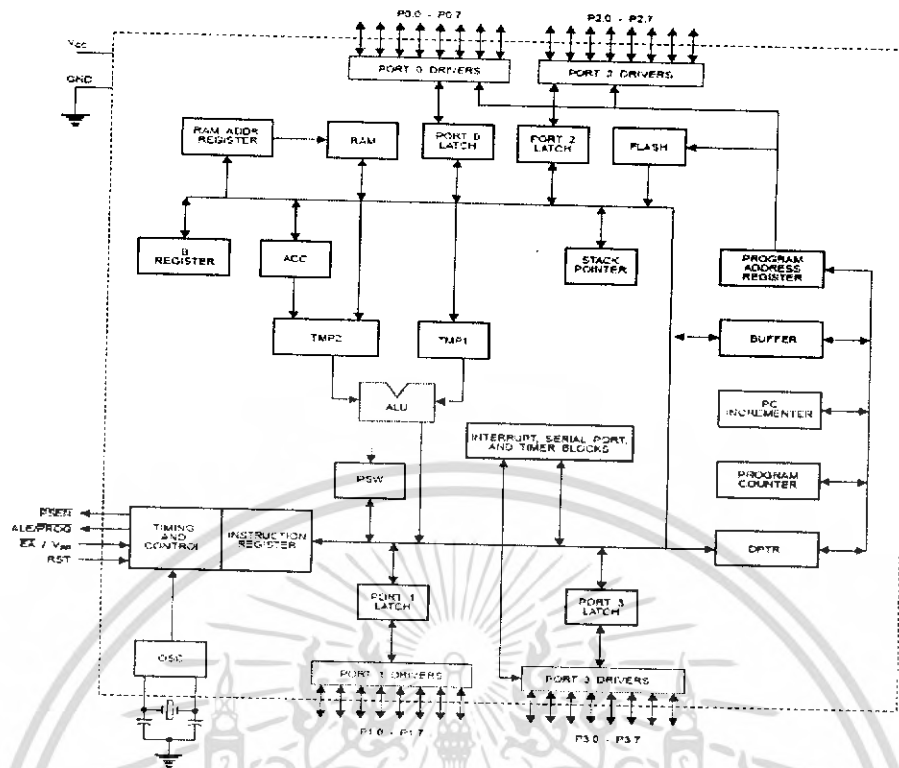
จะเห็นว่า $v_o = |v_i|$ ตามที่ต้องการ

วงจรในรูปที่ 2.10 (ค) เป็นการต่อตามวงจรตามแผนภาพบล็อกในรูปที่ 2.10 (ข) ถ้าเราไม่ต้องการขยาย ให้เลือก $R_f = R_i$ ก็จะได้ $v_o = |v_i|$

2.5 โครงสร้างและสถาปัตยกรรมของไมโครคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์คืออุปกรณ์อิเล็กทรอนิกส์ประเภทหนึ่งซึ่งรวมเอาหน่วยประมวลผล วงจรรับสัญญาณอินพุต หน่วยคำนวณวงจรกำเนิดสัญญาณนาฬิกา หน่วยคำนวณทางคณิตศาสตร์และลอจิก ไว้ด้วยกัน จึงสามารถนำไปใช้งานแทนวงจรอิเล็กทรอนิกส์ที่มีความซับซ้อนได้เป็นอย่างดี และยังช่วยลดจำนวนอุปกรณ์กับขนาดของระบบให้มีปริมาณลดลง ขณะที่มีความสามารถสูงขึ้นภายใต้งบประมาณที่เหมาะสม

ไมโครคอนโทรลเลอร์มาจากการรวมกันของคำ 2 คำคือ ไมโคร ซึ่งหมายถึงไมโครโปรเซสเซอร์ที่เป็นอุปกรณ์ประมวลผลข้อมูลขนาดเล็กภายในประกอบด้วย หน่วยประมวลผลกลางหรือซีพียู หน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรเชื่อมต่อหน่วยความจำ และวงจรสัญญาณนาฬิกา และอีกคำหนึ่งคือ คอนโทรลเลอร์ หมายถึงอุปกรณ์ที่ใช้ในการควบคุม ดังนั้นไมโครคอนโทรลเลอร์จึงเป็นอุปกรณ์ที่ใช้ในการควบคุมโดยที่สามารถเขียนโปรแกรมเพื่อกำหนดรูปแบบการควบคุมได้อย่างอิสระ



รูปที่ 2.11 แสดงรายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชของ ATMEL เบอร์ AT89C52

2.5.1 คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51

1. เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียู ขนาด 8 บิต
2. ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลชสามารถลบและเขียนใหม่ได้พันครั้ง
3. หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม แต่ในบางเบอร์จะมีหน่วยความจำแบบอีพรอมเพิ่มเติม
4. ขาพอร์ตเป็นแบบสองทิศทางสามารถใช้งานได้ทั้งอินพุตพอร์ตและเอาต์พุตพอร์ต
5. มีวงจรติดต่อสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
6. ไทเมอร์/เคาน์เตอร์ มีขนาด 16 บิต อย่างน้อย 2 ตัว
7. สามารถรองรับแหล่งกำเนิดอินเทอร์รัปต์ได้ 6 ประเภท
8. สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
9. มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิป
10. มีวงจรสื่อสารอนุกรมแบบ SPI สำหรับในอนุกรม AT89Sxx
11. มีวอตช์ดีอกไทเมอร์ในตัว สำหรับในอนุกรม AT89Sxx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 คุณสมบัติของไมโครคอนโทรลเลอร์ โดยบริษัท Atmel

ตารางที่ 2.1 Versions of 8051 from Atmel (All ROM Flash)

Part Number	ROM	RAM	I/O pins	Timer	Interrupt	V _{cc}	Packaging
AT89C51	4K	128	32	2	6	5V	40
AT89LV51	4K	128	32	2	6	3V	40
AT89C1051	1K	64	15	1	3	3V	20
AT89C2051	2K	128	15	2	6	3V	20
AT89C52	8K	128	32	3	8	5V	40
AT89LV52	8K	128	32	3	8	3V	40

89C51 มีความเร็วของซีพียู 12 16 20 เมกะเฮิร์ต ดังแสดงในตารางที่ 2.2

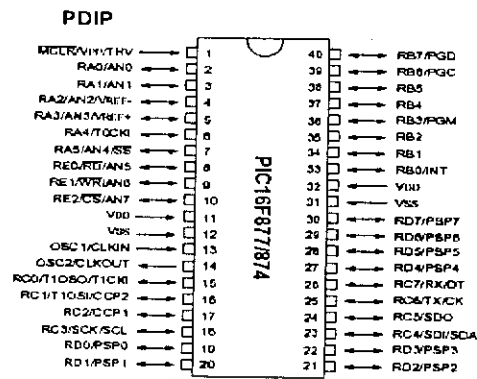
ตารางที่ 2.2 Various Speeds of 8051 from Atmel

Part Number	Speed	Pins	Packaging	Use
AT89C51-12PC	12 MHZ	40	Dip Plastic	Commercial
AT89C51-16PC	16 MHZ	40	Dip Plastic	Commercial
AT89C51-20PC	20 MHZ	40	Dip Plastic	Commercial

2.6 ไมโครคอนโทรลเลอร์ตระกูล PIC

ไมโครคอนโทรลเลอร์มีการพัฒนาขึ้นอย่างต่อเนื่อง ปัจจุบันนี้ไมโครคอนโทรลเลอร์ตระกูลใหม่ๆเกิดขึ้นมามากมายแล้วแต่มีศักยภาพในการทำงานสูงด้วยกันทั้งสิ้น โดยไมโครคอนโทรลเลอร์ตระกูล PIC ของบริษัทไมโครชิปได้รับความนิยมแพร่หลายมากในบ้านเรา เนื่องจากเป็นไมโครคอนโทรลเลอร์ที่มีความสามารถและเฟียบพร้อมไปด้วยทรัพยากรหรือฟังก์ชันการใช้งานต่างๆมากมาย เช่น โมดูลการแปลงสัญญาณอนาล็อกเป็นดิจิทัล ไทเมอร์/คาน์เตอร์ USART, SPI, I²C, PWM และอื่นๆ ซึ่งองค์ประกอบต่างๆเหล่านี้จะถูกสร้างมารวมอยู่ในตัวซีพียูเดียว ทำให้ซีพียูเพียงตัวเดียวก็สามารถทำงานได้หลายๆอย่างและสามารถลดฮาร์ดแวร์บางส่วนลงได้ ความเร็วของซีพียูตระกูลนี้จะใช้เวลาในการกระทำคำสั่งต่างๆ เพียง 1 หรือ 2 ไซเคิลต่อคำสั่งเท่านั้น โดยการทำงานจะเป็นลักษณะไปป์ไลน์ ทำให้มีความเร็วในการทำงานมากกว่าซีพียูโดยทั่วไปที่ความถี่เดียวกัน ซึ่งไมโครคอนโทรลเลอร์ที่กล่าวถึงนี้คือเบอร์ PIC 16F877 ซึ่งเป็นเบอร์ที่ได้รับความนิยมมากอีกเบอร์หนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 แสดงการจัดขามมาตรฐานของไมโครคอนโทรลเลอร์ตระกูล PIC เบอร์ PIC16F877

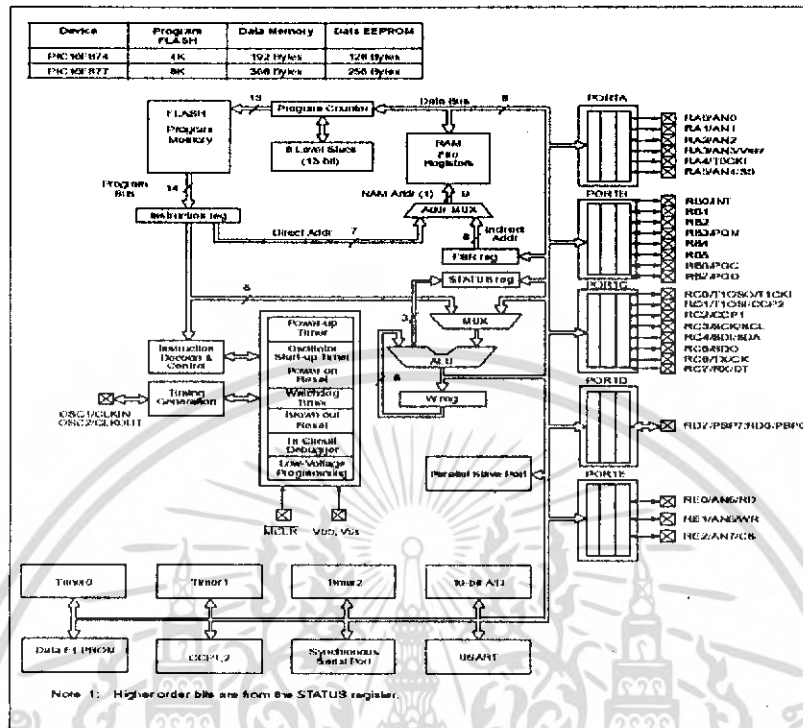
2.6.1 คุณสมบัติของ PIC16F877

คุณสมบัติต่างๆ ของไมโครคอนโทรลเลอร์ PIC16F877 สามารถสรุปได้ดังนี้

1. มี 35 คำสั่ง
2. ในการปฏิบัติตามคำสั่งต่างๆจะใช้ 1 และ 2 ไชเคิล ในคำสั่งที่เป็นการกระโดด
3. ความถี่สูงสุดที่ทำงานได้คือ 20 เมกะเฮิร์ต
4. การทำงานจะเป็นลักษณะ ไปป์ไลน์ทำให้การทำงานเร็วขึ้น
5. หน่วยความจำโปรแกรม (Program Memory) มีขนาด 8 กิโลไบต์
6. หน่วยความจำข้อมูลแรม 368 ไบต์
7. หน่วยความจำข้อมูลอีอีพรอม 256 ไบต์
8. สามารถตอบสนองการอินเตอร์รัปต์ได้ถึง 14 แหล่ง
9. สแตก 8 ระดับ
10. เพาเวอร์อนรีเซต เพาเวอร์อัพไทเมอร์
11. วอตช์ด็อกไทเมอร์
12. สามารถเลือกการป้องกันข้อมูลได้ (Code Protection)
13. โหมดประหยัดพลังงาน (Sleep Mode)
14. เลือกโหมดของสัญญาณนาฬิกาได้หลายโหมด
15. สามารถโปรแกรมได้โดยใช้แรงดันไฟ 5 โวลต์ได้
16. ฟังก์ชันการโปรแกรมแบบ ICSP (In-Circuit Serial Program)
17. ทำงานที่ไฟเลี้ยง 2.0 โวลต์ ถึง 5.5 โวลต์
18. กระแสทั้งซิงค์และซอร์สของพอร์ตคือ 25 มิลลิแอมป์
19. ไทเมอร์/คาน์เตอร์ จำนวน 3 ตัว คือ ไทเมอร์0 ไทเมอร์1 และไทเมอร์2
20. โมดูล แคลเจอร์/Compare/PWM จำนวน 2 ชุด
21. โมดูลการแปลงสัญญาณอนาล็อกเป็นดิจิตอล ความละเอียด 10 บิตมี 8 แชนแนล ภายในตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2 โครงสร้างภายในชิป



รูปที่ 2.13 แสดงรายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ตระกูล PIC เบอร์ PIC16F877

จากผังจะมีรีจิสเตอร์สำคัญๆ คือ W ซึ่งเป็นรีจิสเตอร์ที่ใช้ในการทำเป็นอินพุตให้กับ ALU และเป็นตัวเก็บผลลัพธ์จากการทำงานของ ALU STATUS เป็นรีจิสเตอร์ที่ใช้เก็บสถานะของการทำงานในคำสั่ง ว่าเมื่อคำสั่งทำงานเสร็จแล้วเกิดอะไรขึ้นบ้าง ซึ่งมีประโยชน์ในการเขียนโปรแกรมแบบมีเงื่อนไข โปรแกรมเกาน์เตอร์จะเป็นรีจิสเตอร์อีกตัวหนึ่งที่มีความสำคัญ เนื่องจากใช้สำหรับเป็นตัวชี้ว่า คำสั่งที่จะนำมาประมวลผลนั้นอยู่ ณ ตำแหน่งใดในหน่วยความจำ

2.6.3 หน้าที่ของพอร์ตที่ใช้งาน

1. RA0-RA3 และ RA5 จะใช้งานเป็น I/O ปกติ และทำหน้าที่เป็นขาอินพุตของสัญญาณอนาล็อก (AN0-AN4)

2. RA4 เป็นขา I/O

3. RA6/OSC2/CLKO จะทำหน้าที่ในหลายส่วน คือ เป็นขา OSC2 และ CLKO จะนำมาใช้เป็นขาสัญญาณ I/O ได้ก็ต่อเมื่อเราใช้คริสตอลออสซิลเลเตอร์แบบโมดูลสำเร็จ ซึ่งสามารถต่อเข้ากับขา OSC1/CLKIN ได้โดยไม่ต้องต่อกับขา RA6/OSC2 ทำให้ ขา RA6 ว่างและนำไปใช้เป็น I/O ได้

4. RB0-RB7 สามารถใช้งานเป็นแบบ I/O แต่มีคุณสมบัติพิเศษคือ มีวงจรพูลอัพภายในและเป็นแหล่งกำเนิดสัญญาณอินเทอร์รัปต์ต่างๆ ดังนี้

- RB0/INT0 เป็นขาสัญญาณอินเทอร์รัปต์ภายนอก 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

- RB1/INT1 เป็นขาสัญญาณอินเทอร์รัปต์ภายนอก 1
- RB2/INT2 เป็นขาสัญญาณอินเทอร์รัปต์ภายนอก 2
- RB3/INT3 เป็นขาสัญญาณอินเทอร์รัปต์ภายนอก 3 (18F442)
- RB4-RB7 เป็นขาที่สามารถกำเนิดสัญญาณอินเทอร์รัปต์ได้

2.6.4 พอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ PIC16F877

PIC16F877 มีพอร์ตใช้งานตั้งแต่ 3 ถึง 5 พอร์ต และจำนวน 20 ถึง 30 บิต ซึ่งขึ้นอยู่กับเบอร์ของไมโครคอนโทรลเลอร์และด้วยความสามารถในตัวของ PIC16F877 ที่สามารถทำงานได้หลายอย่าง

ความสามารถในการกระจายกระแสเอาต์พุตของพอร์ตที่ไฟเลี้ยง 5 โวลต์ คือ 25 มิลลิแอมป์ต่อขาทั้งกระแสซิงค์และกระแสซอร์ส ในขณะที่กระแสเอาต์พุตรวมของพอร์ต A, B และ E มีค่าสูงสุด 200 มิลลิแอมป์ ส่วนกระแสเอาต์พุตรวมของพอร์ต C และ D มีค่าสูงสุดได้ 200 มิลลิแอมป์ ในการออกแบบเพื่อขับโหลดทางด้านเอาต์พุตรวมของพอร์ตต้องระวังเรื่องเอาต์พุตรวมที่ไมโครคอนโทรลเลอร์สามารถขับโหลดได้ด้วย

พอร์ต A

มีทั้งสิ้น 6 ช่องหรือ 6 บิต กำหนดให้ชื่อขาเป็น RA0-RA5 รีจิสเตอร์ที่ใช้ในการเก็บข้อมูลคือ พอร์ต A มีแอดเดรสอยู่ที่ 0x05 (แแบงค์ 0) เป็นรีจิสเตอร์ขนาด 8 บิต แต่ใช้งานจริงเพียง 6 บิต ที่เหลือ 2 บิต ต้องกำหนดให้เป็น "0" ส่วนการกำหนดทิศทางของพอร์ตนี้จะกระทำผ่านทางรีจิสเตอร์ TRISA ซึ่งมีแอดเดรสอยู่ที่ 0x85 (แแบงค์ 1) มีขนาด 8 บิต และใช้เพียง 6 บิตเช่นกัน 2 บิตบนคือบิต 6 และบิต 7 ต้องกำหนดให้เป็น "0" บิตของ TRISA ใช้กำหนดทิศทางของ RA0 ไล่เรียงลำดับจนถึงบิต 5 ของ TRISA ใช้กำหนดทิศทางของขาพอร์ต RA5 หากต้องการกำหนดให้บิตใดในพอร์ตเป็นอินพุตต้องเขียนข้อมูล "1" ไปยังบิตนั้น และตรงกันข้ามหากต้องการกำหนดให้เป็นขาเอาต์พุตให้เขียนข้อมูลเป็น "0" ไปยังบิตนั้น

โครงสร้างทางฮาร์ดแวร์ของพอร์ต A สามารถทำงานเป็นขาพอร์ตอินพุตปกติและเป็นขาอินพุตสัญญาณอนาล็อกสำหรับวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลขนาด 10 บิตภายในไมโครคอนโทรลเลอร์ โดยขา RA0-RA3 และ RA5 จะมีการทำงานที่เหมือนกัน ส่วน RA4 จะแตกต่างตรงที่ขานี้นอกจากจะเป็นขาพอร์ตสำหรับไทเมอร์ 0 ภายในไมโครคอนโทรลเลอร์ด้วย และขา RA4 ไม่สามารถใช้งานเป็นขาอินพุตรับสัญญาณอนาล็อกได้ เมื่อขาพอร์ต RA0-RA3 ทำงานเป็นขาพอร์ตอินพุตดิจิตอลจะสามารถรับสัญญาณดิจิตอลระดับทีทีแอล (0 ถึง 5 โวลต์) ได้โดยตรง และหากการทำงานเป็นเอาต์พุตจะสามารถขับโหลดที่ต้องการกระแส 20 มิลลิแอมป์ได้ หากนำมาขับ LED ต้องต่อตัวต้านทานจำกัดกระแสหรือใช้ไฟเลี้ยง 3 โวลต์ก็จะขับ LED ได้โดยตรง

พอร์ต B

มี 8 บิต กำหนดชื่อขาเป็น RB0-RB7 รีจิสเตอร์ที่ใช้ในการเก็บข้อมูลคือ พอร์ต B มีแอดเดรสอยู่ที่ 0x06 (แแบงค์ 0) เป็นรีจิสเตอร์ขนาด 8 บิต การกำหนดทิศทางของพอร์ตนี้จะกระทำผ่านทางรีจิสเตอร์ TRISB ซึ่งมีแอดเดรสอยู่ที่ 0x86 (แแบงค์ 1) และ 0x186 (แแบงค์ 3) มีขนาดเช่นเดียวกับพอร์ต A บิต 0 ของ TRISB ไล่เรียงลำดับจนถึงบิต 7 ของ TRISB ใช้กำหนดทิศทางของขาพอร์ต RB7 หากต้องการกำหนดให้ขาพอร์ต

บิตใดเป็นอินพุตต้องเขียนข้อมูลเป็น “1” ไปยังบิตนั้น ในทางตรงกันข้ามหากต้องการกำหนดให้เป็นขาเอาต์พุตให้เขียนข้อมูลเป็น “0” ไปยังบิตนั้น โดยโครงสร้างทางฮาร์ดแวร์ของพอร์ต B สามารถใช้งานในลักษณะต่างๆได้ 5 แบบ คือ

1. เป็นขาอินพุตเอาต์พุตปกติ
2. เป็นขาพอร์ตอินพุตสัญญาณอินเทอร์รัปต์จากภายนอก โดยใช้ขา RB0/INT
3. เป็นขาพอร์ตอินพุตสำหรับรับแรงดันโปรแกรมระดับต่ำ (Low Voltage Programming) โดยใช้ขา RB3/PGM
4. เป็นขาข้อมูลอนุกรมและสัญญาณนาฬิกา สำหรับการ โปรแกรมหน่วยความจำในไมโครคอนโทรลเลอร์ ซึ่งใช้ 2 ขา คือ RB7/PGD และ RB6/PGC
5. ใช้เป็นแหล่งกำเนิดสัญญาณอินเทอร์รัปต์แบบตรวจสอบการเปลี่ยนแปลงข้อมูลหรือระดับของสัญญาณที่ขา RB4-RB7

วงจรอินพุตบัฟเฟอร์ที่ขาพอร์ตนี้มีทั้งแบบที่ทีแอลและซิมิต์ทริกเกอร์ เพื่อจะจัดการให้สัญญาณอินพุตที่เข้ามามีความเหมาะสมและสมบูรณ์มากที่สุด สามารถที่จะรองรับการพูลอัปภายในแบบอัตโนมัติได้ ในกรณีการเอนเอเบิลตอบสนองอินเทอร์รัปต์แบบตรวจสอบการเปลี่ยนแปลงลอจิกที่ขาพอร์ต RB4-RB7 หากเกิดการอินเทอร์รัปต์ขึ้น บิต RBIF (บิต 0 ในรีจิสเตอร์ INTCON) จะเซต หลังจากการตอบสนองการอินเทอร์รัปต์แล้ว ต้องเคลียร์บิต RBIF ด้วยกระบวนการทางซอฟต์แวร์เสมอ

พอร์ต C

มีทั้งสิ้น 8 บิต กำหนดชื่อขาเป็น RC0-RC7 รีจิสเตอร์ที่ใช้เก็บข้อมูลคือ พอร์ต C มีแอดเดรสอยู่ที่ 0x07 (แแบงค์ 0) เป็นรีจิสเตอร์ขนาด 8 บิต ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ TRISC ซึ่งมีแอดเดรสอยู่ที่ 0x87 (แแบงค์ 1) มีขนาด 8 บิตเช่นเดียวกับพอร์ต A และ B ของขา TRISC ใช้กำหนดทิศทางของขาพอร์ต RC0 ไปเรียงลำดับจนถึงบิต 7 ของ TRISC หากต้องการกำหนดให้ขาพอร์ตบิตใดเป็นอินพุตต้องเขียนข้อมูล “1” ไปยังบิตนั้น และในทางตรงกันข้ามหากต้องการกำหนดให้เป็นขาเอาต์พุตให้เขียนข้อมูลเป็น “0” ไปยังบิตนั้น

พอร์ต C สามารถใช้งานในลักษณะต่างๆ ได้หลายรูปแบบ เป็นขาพอร์ตที่มีความสามารถมาก ไม่ว่าจะเป็นขาพอร์ตอินพุตเอาต์พุตปกติ ขาเอาต์พุตออสซิลเลเตอร์ ของโมดูลไทเมอร์1 ขาอินพุตสำหรับรับสัญญาณนาฬิกาของโมดูลไทเมอร์ 1 ขาเชื่อมต่อระบบบัส ขาเชื่อมต่อแบบ SPI ขาเชื่อมต่ออนุกรมแบบ USART ขาอินพุตของวงจรแคปเจอร์หรือวงจรตรวจจับสัญญาณขาเอาต์พุตของวงจร Compare และ ขาเอาต์พุตวงจร PWM หรืออาจกล่าวได้ว่าพอร์ต C เป็นพอร์ตสำหรับเชื่อมต่ออุปกรณ์ภายนอกที่มีความสมบูรณ์มากที่สุด

พอร์ต D

มี 8 บิต กำหนดชื่อขาเป็น RD0-RD7 รีจิสเตอร์ที่ใช้ในการเก็บข้อมูลคือ พอร์ต D มีแอดเดรสอยู่ที่ 0x08 (แแบงค์ 0) เป็นรีจิสเตอร์ขนาด 8 บิต ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ TRISD เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนสื่อใดๆ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งมีแอดเดรสอยู่ที่ 0x88 (แองก์ 1) มีขนาด 8 บิต หากต้องการกำหนดให้ขาพอร์ตในบิตใดเป็นอินพุตต้องเขียนข้อมูล "1" ไปยังบิตนั้น และในทางตรงกันข้ามหากต้องการกำหนดให้เป็นขาเอาต์พุตให้เขียนข้อมูลเป็น "0" ไปยังบิตนั้น สำหรับ พอร์ต D นี้จะมีเฉพาะไมโครคอนโทรลเลอร์ตระกูล PIC รุ่น 40 ขา เท่านั้น

พอร์ต D สามารถใช้งานได้ 2 โหมดคือ เป็นขาพอร์ตอินพุตเอาต์พุตปกติและเป็นส่วนขยายพอร์ตแบบขนาน (Parallel Slave Port) สำหรับใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอกที่จัดระบบบัสแบบไมโคร-โปรเซสเซอร์คือ สายสัญญาณควบคุมการอ่าน เขียน และเลือกอุปกรณ์ (Chip Select) การเลือกโหมดการทำงานของพอร์ต D นี้ขึ้นกับบิตโหมดขยายพอร์ตแบบขนานถ้าเป็น "0" เป็นการกำหนดให้พอร์ต D เป็นพอร์ตปกติ และถ้าเป็น "1" พอร์ต D จะทำงานในโหมดขยายพอร์ตแบบขนาน

พอร์ต E

มี 3 บิต กำหนดชื่อขาเป็น RE0-RE2 รีจิสเตอร์ที่ใช้ในการเก็บข้อมูลคือ พอร์ต E มีแอดเดรสอยู่ที่ 0x09 (แองก์ 0) เป็นรีจิสเตอร์ขนาด 8 บิต แต่ใช้งานได้เพียง 3 บิตล่างคือ บิต 0 บิต 2 เท่านั้น ที่เหลือกำหนดเป็น "0" ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ TRISE มีแอดเดรสอยู่ที่ 0x89 (แองก์ 1) มีขนาด 8 บิต โดยใช้ 3 บิตล่างในการกำหนดทิศทางของพอร์ต E ส่วนที่เหลือใช้ควบคุมการทำงานในโหมดขยายพอร์ตแบบขนานของพอร์ต E สามารถใช้เป็นขาพอร์ตอินพุตเอาต์พุตปกติ ขาอินพุตอนาล็อก จะแปลงสัญญาณอนาล็อกเป็นดิจิทัล และขาควบคุมการติดต่อกับอุปกรณ์ภายนอกแบบ ขยายพอร์ตแบบขนานทั้งนี้ขึ้นอยู่กับการกำหนดข้อมูลของรีจิสเตอร์ที่ใช้ควบคุมการทำงานของพอร์ตนี้ เช่นเดียวกับพอร์ต D และพอร์ต E จะมีเฉพาะในไมโครคอนโทรลเลอร์ตระกูล PIC รุ่น 40 ขา เท่านั้น

โครงสร้างทางฮาร์ดแวร์ของพอร์ต E นั้นบิต RE0-RE2 เมื่อทำงานในโหมดพอร์ตอินพุตเอาต์พุตปกติจะเห็นได้ว่าความคล้ายคลึงกับพอร์ตนี้จะเป็นแบบขมิดด์ทริกเกอร์ ในขณะที่ทำงานในโหมดของการแปลงสัญญาณอนาล็อกเป็นดิจิทัล วงจรอินพุตจะเปลี่ยนเป็นแบบทีทีแอล แม้ว่าพอร์ต E ใน PIC 16F877 มีจำนวนน้อยเพียงแค่ 3 บิต แต่สามารถเลือกรูปแบบการทำงานได้มากถึง 3 แบบคือ เป็นขาพอร์ตอินพุตเอาต์พุตปกติ อินพุตสำหรับวงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัลขนาด 10 บิต และพอร์ตสัญญาณควบคุมสำหรับการติดต่อกับอุปกรณ์ในโหมดขยายพอร์ตแบบขนาน ดังนั้นในการเลือกรูปแบบการทำงานต้องระมัดระวังเช่นเดียวกับพอร์ต C ที่ได้กล่าวมาแล้ว

2.6.5 ชนิดของ PIC

ไมโครคอนโทรลเลอร์ในตระกูล PIC นั้นถ้าแบ่งออกตามชนิดของหน่วยความจำโปรแกรมแล้วสามารถแบ่งได้เป็น 3 แบบ ด้วยกันดังนี้คือ

1. OTP (One Time Programmable) เป็นชิปที่มีราคาถูกที่สุดในสามประเภท สาเหตุก็มาจากว่าชิปแบบ OTP จะสามารถทำการโปรแกรมได้แค่ครั้งเดียวเท่านั้น หลังจากชิปได้ถูกโปรแกรมไปแล้วจะไม่สามารถใช้โปรแกรมใหม่ได้อีก ดังนั้นชิปประเภทนี้นิยมใช้หลังจากที่ได้มีการพัฒนาโปรแกรมจนกระทั่งจุดบกพร่องต่างๆในโปรแกรมไม่มีอีกแล้ว เพราะจะมีต้นทุนต่ำกว่าเมื่อเทียบกับตัวหน่วยความจำประเภทอื่นๆ

2. อีพรอมเป็นชิปที่มีหน่วยความจำโปรแกรมที่เมื่อเขียนโปรแกรมเข้าไปแล้ว สามารถโปรแกรมใหม่ได้ด้วยการลบโปรแกรมเดิม โดยใช้แสงยูวีส่องผ่านเข้าไปยังชิปใช้เวลาประมาณ 5 ถึง 10 นาที ดังนั้นที่ด้านบนของชิปจะมีกรอบกระจกเพื่อให้แสงยูวีสามารถจะส่องผ่านเข้าไปในตัวชิปได้ แต่ก็มีจำนวนครั้งในการลบโปรแกรมเช่นกัน เมื่อลบโปรแกรมด้วยแสงยูวีมากๆ ตัวชิปก็จะหมดอายุเร็วขึ้น

3. อีอีพรอมเป็นชิปที่ออกมาไม่กี่ปีเอง ส่วนของหน่วยความจำโปรแกรมสามารถอ่านหรือเขียนด้วยสัญญาณทางไฟฟ้า ซึ่งใช้เวลาในการลบข้อมูลไม่กี่วินาทีสามารถลบและเขียนใหม่ได้หลายพันครั้ง ทำให้เป็นที่นิยมที่สุดในสามประเภท

2.6.6 การจัดสรรหน่วยความจำโปรแกรมของ PIC

หน่วยความจำโปรแกรมเป็นส่วนสำคัญมากเพราะเป็นที่เก็บข้อมูลคำสั่งทั้งหมด ซึ่งใช้กำหนดให้ไมโครคอนโทรลเลอร์ทำงาน หน่วยความจำโปรแกรมของ PIC16F877 เป็นแบบเฟลช ทำให้สามารถลบและเขียนใหม่ได้เป็นพันครั้ง แต่อย่างไรก็ตามโดยปกติหน่วยความจำโปรแกรม หลังจากทำการเขียนในขั้น ตอนของการ โปรแกรมแล้วก็จะมิไว้สำหรับอ่านออกมาได้เพียงทางเดียว

PIC16F877 มีโปรแกรมเคาน์เตอร์ขนาด 13 บิต เพื่อกำหนดการเข้าถึงหน่วยความจำโปรแกรมมีขนาด 8 กิโลเวิร์ด เนื่องจากไมโครคอนโทรลเลอร์ในอนุกรมนี้มีขนาดหน่วยความจำที่ค่อนข้างใหญ่มาก จึงต้องจัดสรรเป็นเพจหรือเป็นหน้า โดยแต่ละเพจจะมีขนาด 2 กิโลเวิร์ด ทั้งนี้เนื่องจากชุดคำสั่งเกี่ยวกับการกระโดดของไมโครคอนโทรลเลอร์ตระกูล PIC สามารถเข้าถึงหน่วยความจำได้สูงสุด 2,048 ตำแหน่ง

สำหรับ PIC16F877 มีขนาดหน่วยความจำโปรแกรม 8 กิโลเวิร์ด มีการสงวนแอดเดรส 0x0000 และ 0x0004 ไว้เช่นกัน หรืออาจกล่าวได้ว่าเป็นรูปแบบมาตรฐานของการเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์สำหรับ PIC16F877 มีการแบ่งหน่วยความจำออกเป็น 4 เพจ ดังนี้คือ

เพจ 0 มีแอดเดรสในช่วง 0x0000-0x07FF (โดยสงวนแอดเดรส 0x0000 และ 0x0004 ไว้)

เพจ 1 มีแอดเดรสในช่วง 0x0800-0x0FFF

เพจ 2 มีแอดเดรสในช่วง 0x1000-0x17FF

เพจ 3 มีแอดเดรสในช่วง 0x1800-0x1FFF

นอกจากนั้นใน PIC16F877 ยังมีพื้นที่หน่วยความจำพิเศษสำหรับเก็บค่าของโปรแกรมชั่วคราวไว้ขนาด 13 บิต โดยเรียกว่า สแตก ซึ่งจะมิบทบาทมากสำหรับการกระโดดไปทำงานยังโปรแกรมย่อยของ PIC16F877 โดยที่เมื่อกระทำคำสั่งแล้วให้กระโดดไปทำงานยังโปรแกรมย่อย ซีพียูจะเก็บค่าในโปรแกรมเคาน์เตอร์ ในขณะที่ไว้ในสแตก จากนั้นจะกระโดดไปทำงานยังโปรแกรมย่อย เมื่อทำงานเสร็จเรียบร้อยแล้วซีพียูจะอ่านค่าโปรแกรมเคาน์เตอร์จากสแตกกลับมา แล้วทำตามกระบวนการในโปรแกรมหลักต่อไป สำหรับสแตกใน PIC16F877 มีขนาด 13 บิต สามารถเก็บค่าของโปรแกรมเคาน์เตอร์ได้ 8 ระดับ

2.6.7 การจัดสรรหน่วยความจำข้อมูลแรมและรีจิสเตอร์ไฟล์

ใน PIC16F877 มีหน่วยความจำข้อมูลแรมสำหรับใช้งานทั่วไป 368 ไบต์ และมีรีจิสเตอร์ไฟล์ 8 บิต 57 ตัว แต่ละแบงค์มีขนาดสูงสุด 128 ไบต์ แต่มีการใช้งานได้จริงในแต่ละแบงค์ต่างกัน โดยแต่ละแบงค์มีการจัดสรรพื้นที่ดังนี้

แบงค์ 0 มีช่วงแอดเดรส 0x00-0x7F

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอดเดรส 0x00-0x1F เป็นพื้นที่ของรีจิสเตอร์ไฟล์

แอดเดรส 0x1F -0x7F เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 96 ไบต์
 แบนด์ 1 มีช่วงแอดเดรส 0x80-0xFF

แอดเดรส 0x80-0x9F เป็นพื้นที่ของรีจิสเตอร์ไฟล์ แต่มีบางแอดเดรสไม่ใช้งาน

แอดเดรส 0xA0-0xEF เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 80 ไบต์
 แบนด์ 2 มีช่วงแอดเดรส 0x100-0x17F

แอดเดรส 0x100-0x10F เป็นพื้นที่ของรีจิสเตอร์ไฟล์ แต่มีบางแอดเดรสไม่ใช้งาน

แอดเดรส 0x110-0x11F เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 16 ไบต์

แอดเดรส 0x120-0x16F เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 80 ไบต์

แอดเดรส 0x170-0x17F บรรจุข้อมูลเหมือนกับในแอดเดรส 0x70-0x7F ในแบนด์ 0 เพื่อช่วยให้สามารถใช้ข้อมูลจากแอดเดรส 0x00-0x7F ได้ง่ายขึ้น โดยไม่ต้องเปลี่ยนแบนด์

แบนด์ 3 มีช่วงแอดเดรส 0x100-0x17F

แอดเดรส 0x180-0x1FF เป็นพื้นที่ของรีจิสเตอร์ไฟล์ แต่มีบางแอดเดรสไม่ใช้งาน

แอดเดรส 0x190-0x11F เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 16 ไบต์

แอดเดรส 0x1A0-0x1AF เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 80 ไบต์

แอดเดรส 0x1F0-0x1FF บรรจุข้อมูลเหมือนกับในแอดเดรส 0x70-0x7F ในแบนด์ 0 เพื่อช่วยให้สามารถใช้ข้อมูลจากแอดเดรส 0x00-0x7F ได้ง่ายขึ้น โดยไม่ต้องเปลี่ยนแบนด์

2.6.8 ความเร็วของ PIC

ภาคของควมดีสัญญาณนาฬิกา ปัจจุบันสามารถทำสัญญาณนาฬิกาได้ที่ 20 เมกะเฮิร์ต ซึ่งทำให้หนึ่งคำสั่งของ PIC ใช้เวลาทำงานเพียง 0.25 ไมโครวินาที แต่อย่างไรก็ตามได้มีบริษัทอื่นได้ซื้อลิขสิทธิ์ PIC จาก ไมโครชิป และได้สร้างชิปที่มีความเร็วได้มากกว่าเดิมขึ้นไปอีก

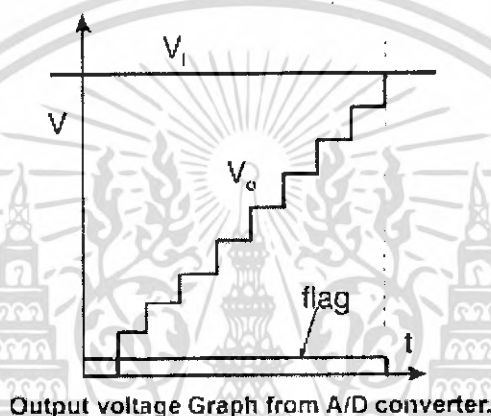
2.7 การแปลงสัญญาณอนาล็อกเป็นดิจิตอล

สัญญาณที่ใช้ในอุปกรณ์อิเล็กทรอนิกส์มี 2 ชนิด คือ สัญญาณอนาล็อกและสัญญาณดิจิตอล โดยสัญญาณอนาล็อกจะใช้กับอุปกรณ์ทั่วไปและใช้การควบคุมแบบเก่า

ปัจจุบันไมโครคอนโทรลเลอร์เข้ามาช่วยในการควบคุมอุปกรณ์ต่างๆมากมาย ทำให้การควบคุมทำได้ง่ายและรวดเร็วมากยิ่งขึ้น แต่ในการควบคุมเราจำเป็นต้องใช้สัญญาณดิจิตอลในการติดต่อสื่อสารกับไมโครคอนโทรลเลอร์ แต่ในความเป็นจริงแล้วใช้สัญญาณอนาล็อกในการควบคุม ดังนั้นจึงต้องเปลี่ยนสัญญาณอนาล็อกให้เป็นสัญญาณดิจิตอลแล้วนำสัญญาณที่ได้เข้ามาสู่ไมโครคอนโทรลเลอร์ เพื่อจะใช้ในการควบคุมระบบต่อไป

แม้ว่าสัญญาณอนาล็อกจะมีความแน่นอนและความแม่นยำสูง แต่สัญญาณอนาล็อกจะควบคุมได้ยาก เนื่องจากในสภาพแวดล้อมที่มีสัญญาณรบกวนอยู่มาก และวงจรในการควบคุมสัญญาณแบบอนาล็อกไม่ว่องไวเท่าที่ควร อีกทั้งยังมีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

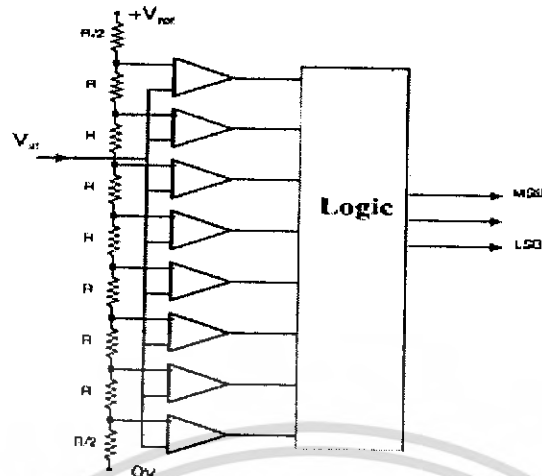
จะมีความซับซ้อนมากเมื่อเทียบกับการควบคุมแบบดิจิทัล จึงเป็นสาเหตุที่ทำให้การควบคุมแบบอนาล็อกสามารถทำได้ยาก อย่างไรก็ตาม สัญญาณดิจิทัลก็ไม่สามารถทดแทนความละเอียดของสัญญาณอนาล็อกได้อย่างสมบูรณ์แบบ แต่สามารถทำให้การควบคุมนั้นทำให้ง่ายและสะดวกมากยิ่งขึ้น วงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัล เมื่อรับค่าที่นับเพิ่มขึ้นทีละหนึ่งจากตัวนับ แล้วจะแปลงค่าให้เป็นสัญญาณอนาล็อกที่มีค่าความต่างศักย์ค่าๆหนึ่ง จากนั้นส่งเข้าไปที่อุปกรณ์ตัวเปรียบเทียบจะเปรียบเทียบค่าความต่างศักย์ของสัญญาณอินพุตและค่าจากตัวนับ ถ้าหากว่าสัญญาณทั้งสองมีค่าเท่ากันก็จะส่งค่าความต่างศักย์ 0 โวลต์ออกมา (ลอจิก 0) ถ้าไม่ค่าเท่ากันก็จะส่งความต่างศักย์ที่ไม่ใช่ 0 โวลต์ออกมา (ลอจิก 1)



รูปที่ 2.14 แสดงกราฟแรงดันเอาต์พุตที่ได้จากการแปลงสัญญาณอนาล็อกเป็นดิจิทัล

ค่าความต่างศักย์ที่ออกมาจะนำมาเข้าลอจิกเกตและสัญญาณนาฬิกาจะได้ค่าลอจิกออกมา ถ้าหากผลลัพธ์ที่ได้เป็นสัญญาณนาฬิกา ก็แสดงว่ายังไม่ได้ผลลัพธ์ตามที่ต้องการ สัญญาณนาฬิกาจะเข้าไปทำให้ตัวนับเริ่มนับเพิ่มขึ้นต่อไป และเมื่อได้ค่าผลลัพธ์ดิจิทัลที่ต้องการแล้วค่าที่ได้จากตัวเปรียบเทียบจะให้ความต่างศักย์เป็น 0 โวลต์ (ลอจิก 0) เมื่อนำมาเข้าลอจิกเกตและสัญญาณนาฬิกาเข้ามาแล้วก็จะให้ลอจิก 0 ทำให้ตัวนับไม่นับเพิ่มอีก ดังนั้นจึงจะได้ค่าดิจิทัลจากตัวนับที่ต้องการ

ข้อเสียของวิธีนี้ คือการนับต้องเริ่มนับที่ 0 เสมอและนับเพิ่มขึ้นเรื่อยๆ จะทำให้ช้า เพราะเอาต์พุตที่ได้จะมีการหน่วงเวลาไว้ จึงไม่เป็นที่นิยมใช้เท่าที่ควร จึงได้มีการเปลี่ยนตัวนับใหม่ซึ่งเป็นแบบนับลงได้ด้วย จึงจะอ้างอิงระดับจากระดับเก่า ทำให้ไม่จำเป็นต้องนับ 0 ใหม่ทุกครั้งเมื่อมีการเปลี่ยนอินพุตแต่ให้อ้างอิงกับผลลัพธ์เดิม ทำให้ได้ผลลัพธ์เร็วขึ้น



A "Flash" converter.

รูปที่ 2.15 แสดงการแปลงสัญญาณอนาล็อกเป็นดิจิทัลแบบแฟลช

หลักการของการแปลงสัญญาณอนาล็อกเป็นดิจิทัลแบบแฟลช ก็คือการใช้การแบ่งแรงดันเป็นหลายๆ ค่า แล้วเปรียบเทียบกับ V_{in} เป็นคู่ๆ พร้อมกัน แล้วทำการทางลอจิก 1 จากรูปที่ 2.15 มีแรงดันอินพุตเปรียบเทียบกับกัน 8 บิต ค่าความต่างศักย์จะเพิ่มขึ้นเรื่อยๆ จากค่าความต้านทานที่ต่อเพิ่มขึ้นความต่างศักย์ที่ได้นั้น เมื่อนำไปเปรียบเทียบกับ V_{in} แล้วเมื่อมีค่ามากกว่าก็จะปล่อยลอจิกออกมา ถ้ามากกว่าก็จะให้ลอจิก 1 แต่ถ้าน้อยกว่าหรือเท่ากันก็จะให้ลอจิก 0 วิธีการแปลงสัญญาณอนาล็อกเป็นดิจิทัลแบบแฟลชนี้จะเร็วที่สุดแต่ใช้ได้กับอุปกรณ์ทางฮาร์ดแวร์มากกว่าแบบอื่นๆ

2.7.1 โมดูลการแปลงสัญญาณอนาล็อกเป็นดิจิทัลของ PIC16F877

อีกหนึ่งโมดูลสำคัญที่ไม่โครคอนโทรลเลอร์สมัยใหม่ต้องมีคือโมดูลการแปลงสัญญาณอนาล็อกเป็นดิจิทัล และสำหรับไมโครคอนโทรลเลอร์ PIC16F877 มีความละเอียด 8 ช่องอินพุต โดยขาพอร์ตที่ใช้งานร่วมกันคือขาพอร์ต RA0-RA3, RA5 และ RE0-RE2 การทำงานเพื่อรองรับอินพุตจำนวนมากจะใช้วิธีการมัลติเพล็กซ์ ซึ่งควบคุมกระบวนการทางซอฟต์แวร์ สำหรับโมดูลการแปลงสัญญาณอนาล็อกเป็นดิจิทัล ใน PIC16F877 เป็นแบบซิงเซลซีฟ แอปพริอ็อกซิเมชัน หรือแบบการประมาณสี่บิตเนื่อง

2.7.2 การทำงานของโมดูลการแปลงสัญญาณอนาล็อกเป็นดิจิทัล

ขาพอร์ตที่ใช้งานเป็นอินพุตอนาล็อกต้องได้รับการกำหนดจากบิต PCFG3:PCFG0 ในรีจิสเตอร์ ADCON1 ว่าจะให้ขาพอร์ตใดเป็นพอร์ตรับสัญญาณอนาล็อกและพอร์ตใดรับแรงดันอ้างอิง หลังจากนั้นสัญญาณอนาล็อกที่ต้องการแปลงจะส่งผ่านวงจรเลือกช่องสัญญาณและจะถูกควบคุมโดยบิต CHS2:CHS0 ในรีจิสเตอร์ ADCON0 ภาคอินพุตของวงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัลใน PIC16F877 ประกอบด้วยวงจรสุ่มและเก็บสัญญาณที่มีตัวเก็บประจุค่า 120 พิโคฟารัด จะทำหน้าที่เก็บแรงดันอินพุตไว้ให้กับวงจรการแปลงสัญญาณอนาล็อกเป็นดิจิทัล และวงจรเปรียบเทียบภายในหรือประจุแรงดันอินพุตให้เต็มก่อน จึงจะทำให้ผลลัพธ์ที่ได้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.3 การเลือกความถี่สัญญาณนาฬิกาของการแปลงสัญญาณ

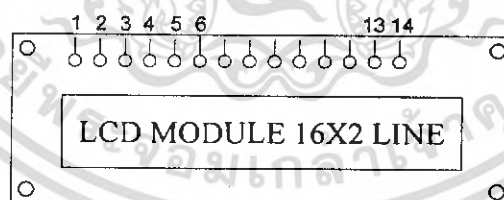
การเลือกความถี่สัญญาณนาฬิกาของการแปลงสัญญาณมีความสำคัญมาก เนื่องจากมีผลต่อเวลารวมในการแปลงสัญญาณ โดยในโมดูลแปลงสัญญาณอนาล็อกเป็นดิจิทัลของ PIC16F877 จะใช้เวลาแปลงสัญญาณอนาล็อกเป็นข้อมูลดิจิทัล 10 บิตเท่ากับ $2T_{AD}$ โดยค่าเวลา T_{AD} คือค่าเวลาที่ใช้ในการแปลงสัญญาณต่อบิต และเวลา T_{AD} ที่น้อยที่สุดยังคงให้ผลการแปลงสัญญาณที่ถูกต้อง คือ 1.6 ไมโครวินาที ดังนั้นการเลือกความถี่สัญญาณนาฬิกาจึงต้องคำนึงถึงปัจจัยนี้ด้วย

2.7.4 กระบวนการแปลงสัญญาณอนาล็อกเป็นดิจิทัล

เมื่อสั่งให้โมดูลการแปลงสัญญาณอนาล็อกเป็นดิจิทัลจำนวน 10 บิต GO/DONE ที่ใช้ในการแสดงสถานะการทำงานจะเซต ถ้ามีการเคลียร์บิตนี้เกิดขึ้นในขณะที่การแปลงสัญญาณยังดำเนินอยู่ การแปลงสัญญาณจะหยุดลงทันที แต่ข้อมูลการแปลงสัญญาณที่ไม่เสร็จสมบูรณ์จะไม่ได้รับการนำไปใช้งาน นั่นคือ ถ้าแปลงไม่สมบูรณ์ค่าในรีจิสเตอร์ ADRESH และ ADRESL จะยังเป็นค่าเดิมก่อนหน้า

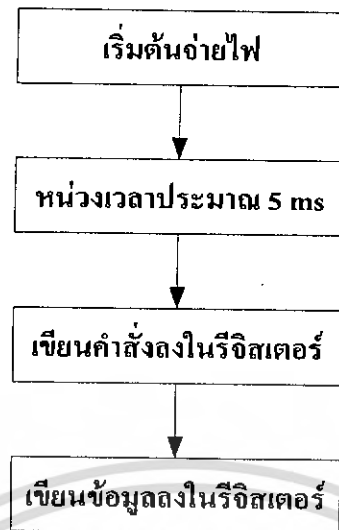
2.8 แอลซีดีโมดูล

โครงสร้างภายในจะประกอบด้วยจอภาพแบบแอลซีดี วงจรขับสัญญาณ และตัวควบคุมแอลซีดีโมดูล การที่จะทำให้แอลซีดีโมดูลแสดงผลออกมาในลักษณะต่างๆ ได้นั้นจะต้องมีการเขียนคำสั่งควบคุมและเขียนข้อมูลให้กับแอลซีดีโมดูล โดยใช้ไมโครคอนโทรลเลอร์ในการควบคุมการทำงานต่างๆ ตามที่เราต้องการ ในการที่จะติดต่อกับแอลซีดีโมดูลจะต้องหน่วงเวลาหลังจากที่ทำการส่งรหัสคำสั่งหรือข้อมูล เนื่องจากต้องรอให้ไมโครคอนโทรลเลอร์ภายในตัวของแอลซีดีโมดูลแปลความหมายของรหัสคำสั่งและทำงานตามคำสั่งให้เรียบร้อยเสียก่อน จากนั้นจึงจะรับข้อมูลหรือดำเนินการต่อไปได้ โดยในที่นี้จะใช้แอลซีดีโมดูล แบบ 16 ตัวอักษร 2 แถว และ 16 ตัวอักษร 4 แถว เพื่อออกคำสั่งต่างๆ ซึ่งจะมีคำสั่งในการตั้งวันที่ เวลา การรับค่าจากเครื่องส่ง การส่งค่าออกสู่จอคอมพิวเตอร์ และการแสดงค่ากำลังงานไฟฟ้า



รูปที่ 2.16 แสดงลักษณะของตัวแอลซีดีโมดูลแบบ 16 ตัวอักษร 2 แถว

ในการใช้งานแอลซีดีโมดูลจะต้องทำความเข้าใจคำสั่งต่างๆ ด้วย โดยการเริ่มต้นใช้งานจะต้องส่งรหัสควบคุมให้กับรีจิสเตอร์ และส่งข้อมูลให้กับรีจิสเตอร์ โดยมีลำดับการทำงานดังรูปที่ 2.17



รูปที่ 2.17 แสดงขั้นตอนการควบคุมแอลซีดี โมดูล

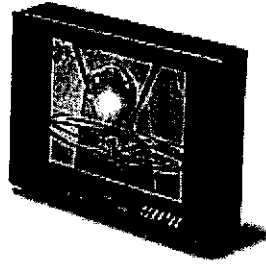
2.8.1 ประเภทของแอลซีดีโมดูล

แอลซีดีโมดูลเป็นอุปกรณ์แสดงผลแบบหนึ่งที่มีนิยมนำใช้กันมาก แอลซีดีโมดูลสามารถแบ่งประเภทตามลักษณะการแสดงผลได้ 3 รูปแบบดังนี้คือ แอลซีดีโมดูลแบบอักษร แอลซีดีโมดูลแบบกราฟฟิก และแอลซีดีโมดูลแบบเซกเมนต์

1. แอลซีดีโมดูลแบบอักษร เป็นแอลซีดีโมดูลที่แสดงตัวอักษรหรือตัวเลข และเครื่องหมายต่างๆ ได้ โดยสร้างจากจุดเล็กๆทางแนวตั้งและแนวนอนหรือเรียกว่าดอตเมตริกซ์ โดยทั่วไปจะมี 2 ขนาดด้วยกัน คือ 5x7 จุดและ 5x10 จุด (ตัวเลขตัวแรกเป็นจำนวนจุดทางแนวนอน ส่วนตัวหลังเป็นจำนวนจุดทางแนวตั้ง) นอกจากนี้แอลซีดีโมดูลยังสามารถแสดงข้อความได้ 1 บรรทัดหรือมากกว่าก็ได้ขึ้นอยู่กับรุ่นของแอลซีดีโมดูลนั้นๆ

2. แอลซีดีโมดูลแบบกราฟฟิก มีโครงสร้างคล้ายกับแอลซีดีโมดูลแบบอักษร สามารถแสดงข้อมูลเป็นตัวอักษร ตัวเลข เครื่องหมาย และรูปภาพได้ ความละเอียดของภาพจะขึ้นอยู่กับความละเอียดของดอตเมตริกซ์ในแอลซีดีโมดูล

จุดเด่นของแอลซีดีโมดูลคือสามารถแสดงผลได้แม้ในสิ่งแวดล้อมที่มีแสงจ้าหรือกลางแจ้ง การมองเห็นทำได้อย่างชัดเจนไม่อาจใช้แรงดันไฟฟ้าขั้วที่แรงดันต่ำ ใช้งาน CMOS ที่ทำงานเพียง 3 โวลต์ ก็สามารถขับแอลซีดีโมดูลได้ ดังนั้นจึงนิยมนำใช้ในวงจรคอมพิวเตอร์หรือวงจรดิจิทัลทั่วไป แหล่งจ่ายไฟสำหรับแอลซีดีโมดูลจะใช้แหล่งจ่ายเดียว และแรงดันไฟฟ้าระดับเดียวจึงไม่ยุ่งยากหรือซับซ้อนในการใช้งานการแสดงผลของแอลซีดีโมดูลมีความคมชัด ไม่มีการกระพริบหรือภาพสั่นไหวไม่สร้างสัญญาณเสียงรบกวน มีขนาดกะทัดรัด น้ำหนักเบา เบนราบ ขนาดแสดงผลมีขนาดเหมาะสมกับการประยุกต์เข้ากับอุปกรณ์ต่างๆ ผู้ออกแบบการแสดงผลทำได้ตามต้องการ



รูปที่ 2.18 แสดงจอภาพแอลซีดีโมดูล แบบกราฟฟิก

3. แอลซีดี โมดูลแบบเซกเมนต์ เป็นข้อมูลแอลซีดีโมดูลแบบที่เล็กที่สุด ซึ่งมีลักษณะการแสดงผลเป็นเซกเมนต์คล้ายกับ LED 7 ส่วน โดยปกติมักจะมีมากกว่า 1 หลัก จะพบเห็นได้ทั่วไป ในดิจิตอลมัลติมิเตอร์

2.8.2 โครงสร้างภายในตัวควบคุมแอลซีดีโมดูล แบบอักษร

ในการใช้งานแอลซีดีโมดูลนั้นจำเป็นต้องทำความเข้าใจเกี่ยวกับ โครงสร้างและคำสั่งที่ใช้ในการควบคุมให้ดีเสียก่อน โดยที่แอลซีดีโมดูลแบบอักษรประกอบด้วย

1. บัฟเฟอร์อินพุตเอาต์พุตเป็นส่วนที่ใช้ในการติดต่อรับ-ส่งข้อมูลกับอุปกรณ์ภายนอก เพื่อที่จะส่งถ่ายข้อมูลเข้าออกภายในตัวควบคุม
2. รีจิสเตอร์คำสั่ง เป็นรีจิสเตอร์ที่ใช้ในการรับข้อมูลคำสั่งจากอุปกรณ์ภายนอกเพื่อนำไปควบคุมการแสดงผล
3. รีจิสเตอร์ข้อมูล (Data Register) เป็นรีจิสเตอร์ที่รับข้อมูลจากอุปกรณ์ภายนอกเข้ามา เพื่อส่งต่อไปยังหน่วยความจำที่ทำหน้าที่เก็บข้อมูลไปแสดงผลหรือนำข้อมูลไปสร้างตัวอักษรเพิ่มเติมในแรมเก็บตัวอักษร
4. แรมเก็บข้อมูลแสดงผล (Display Data RAM) เป็นหน่วยความจำแบบแรมที่ทำหน้าที่เก็บข้อมูลที่มาจากรีจิสเตอร์ข้อมูล ตัวควบคุมจะนำข้อมูลในแรมเก็บข้อมูลแสดงผล นี้ไปเปิดตารางของตัวอักษรที่เก็บไว้ในหน่วยความจำรอม และแรมจะเก็บตัวอักษรเพื่อนำไปแสดงที่ตัวแสดงผลต่อไป
5. รอมเก็บตัวอักษร (Character Generator ROM) เป็นหน่วยความจำรอมที่ใช้เก็บข้อมูลตัวอักษรหรือสัญลักษณ์ที่สามารถอ่านออกไปแสดงที่ตัวแสดงผลได้ มีขนาด 9,900 บิต โดยจะถูกอ่านด้วยค่าของข้อมูลในแรมเก็บข้อมูลแสดงผล
6. แรมเก็บตัวอักษร (Character Generator RAM) เป็นหน่วยความจำแรมที่ใช้เก็บอักษรที่มีการสร้างเพิ่มเติมขึ้นมา
7. เฟล็ก BUSY ทำหน้าที่แจ้งสถานะการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกทราบว่าตัวควบคุมพร้อมที่จะรับข้อมูลหรือคำสั่งหรือไม่ ดังนั้นก่อนจะส่งข้อมูลหรือคำสั่งมายังตัวควบคุมต้องตรวจสอบสถานะของเฟล็ก BUSY นี้เสียก่อน

2.8.3 ส่วนประกอบหลักๆ แอลซีดีโมดูล

ในแอลซีดีโมดูล มีส่วนประกอบหลักๆ 3 ส่วนดังนี้

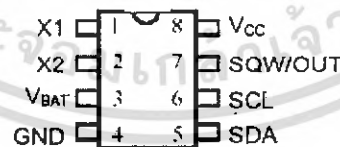
1. ตัวแสดงผล (Display) ภายในจะมีส่วนประกอบเป็นผลึกเหลวที่สามารถแสดงผลให้เห็นโดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงผลบนจอแอลซีดีโมดูล
2. ตัวควบคุม (Controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของแอลซีดีโมดูล เช่น ลบจอภาพ แสดงตัวอักษร หรือเลื่อนเคอร์เซอร์ เป็นต้น
3. ตัวขับ (Driver) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้แสดงผลแสดงข้อมูลตามที่กำหนด

2.9 ไอซีฐานเวลา (Real Time Clock)

สำหรับในระบบนี้ที่ใช้ไมโครคอนโทรลเลอร์ควบคุมจะต้องมีเวลาเข้ามาเกี่ยวข้องกับคีย์ ซึ่งจะบอกเวลาเป็นชั่วโมง นาที และวินาที หรือมีการบอกวัน เดือน และปีให้กับระบบด้วย ดังนั้นการสร้างฐานเวลาจะต้องถูกต้องและแม่นยำ แม้ว่าระบบหยุดทำงานและเริ่มทำงานใหม่ค่าเวลาต่างๆ จะต้องถูกต้องด้วย

2.9.1 คุณสมบัติของไอซีฐานเวลา

1. นับค่าเวลาจริงคิดเป็นวินาที นาที ชั่วโมง วันที่ของแต่ละเดือน เดือน วันต่างๆในแต่ละสัปดาห์ และ ปี ซึ่งสามารถนับได้ถึงปี ค.ศ. 2100
2. มีแรม 56 ไบต์ สำหรับเก็บข้อมูลต่าง ๆ
3. สายข้อมูล 2 เส้นส่งข้อมูลแบบอนุกรม
4. ส่งสัญญาณแบบสี่เหลี่ยม
5. ตรวจสอบแรงดันไฟตกอัตโนมัติ
6. มีแบตเตอรี่แบบคอปที่ใช้กระแสต่ำกว่า 500 นาโนแอมป์ สำหรับการผลิตความถี่
7. สามารถใช้ในสถานที่ ที่มีอุณหภูมิตั้งแต่ -40°C ถึง 85°C
8. มีขาทั้งหมด 8 ขาคัดงัดแบบ DIP และ SOIC



DS1307 8-Pin DIP (300 mil)

รูปที่ 2.19 แสดงลักษณะการจัดขาของไอซีฐานเวลา

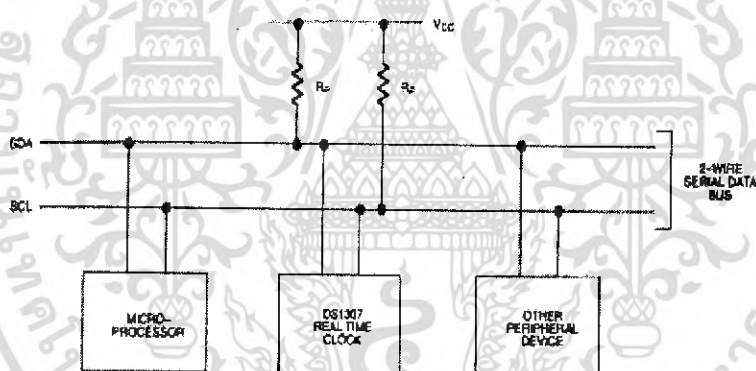
2.9.2 ลักษณะการทำงานของไอซีฐานเวลา

เป็นไอซีฐานเวลาที่ใช้กำลังงานต่ำ มีปฏิทินแสดงเวลาเป็นรหัส BCD และมีหน่วยความจำสำหรับการใช้งานทั่วไป 56 ไบต์ ตำแหน่งละข้อมูลสามารถสื่อสารแบบอนุกรมซึ่งใช้สาย 2 เส้นสามารถส่งข้อมูลได้ 2 ทิศทาง การแสดงเวลาและปฏิทินประกอบด้วย วินาที นาที ชั่วโมง วัน วันที่ เดือน และ ปี วันสุดท้ายของเดือนสามารถบอกได้อย่างอัตโนมัติ รวมถึงการบอกปีได้อย่างถูกต้องแม่นยำ ทั้งนี้ยังสามารถบอกเวลาเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ทั้งแบบ 24 ชั่วโมงและ 12 ชั่วโมง หรือแบบ AM/PM โดยไอซีนี้ยังมีวงจรตรวจจับพลังงานเมื่อแรงดันไม่เพียงพอ

การทำงานจะเริ่มขึ้นเมื่อถูกต่อใช้งานเป็นอุปกรณ์สเลฟ โดยถูกติดตั้งบนบัสที่ส่งผ่านข้อมูลแบบอนุกรม การเริ่มสื่อสารกับไอซีฐานเวลาต้องอยู่ภายใต้เงื่อนไขเริ่มต้น จากนั้นไอซีจะทำการติดต่อสื่อสารจนกระทั่งมีเงื่อนไขเกิดขึ้นทำให้ไอซีหยุดการสื่อสาร หากไฟเลี้ยงมีค่าต่ำกว่า $1.25 \times V_{BAT}$ ก็จะควบคุมให้ไอซีหยุดการทำงาน แล้วจะรีเซ็ตค่าตัวนับแอดเคสภายใน จึงทำให้ไม่สามารถติดต่อกับไอซีได้ แต่ถ้าหากไฟเลี้ยงมีค่าต่ำกว่า V_{BAT} ไอซีจะเข้าสู่โหมดสำรองข้อมูลกระแสต่ำทันที และจะไม่มีกระแสสัญญาณพัลส์ออกมาจนกว่าแรงดัน V_{CC} จะมากกว่า $1.25 \times V_{BAT}$

โดยไอซีฐานเวลาจะสนับสนุนการสื่อสาร 2 ทิศทาง ซึ่งจะใช้สายเพียง 2 เส้นในการเชื่อมต่อกับไมโครคอนโทรลเลอร์ อุปกรณ์ที่ส่งข้อมูลลงในบัสเรียกว่าเครื่องส่ง ส่วนอุปกรณ์ที่รับข้อมูลเข้ามาจากบัสเรียกว่าเครื่องรับ อุปกรณ์ที่ทำหน้าที่ควบคุมการส่งข้อมูลเรียกว่ามาสเตอร์ ส่วนอุปกรณ์ที่ถูกควบคุมโดยมาสเตอร์เรียกว่าสเลฟ โดยบัสต้องถูกควบคุมด้วยอุปกรณ์ที่เป็นมาสเตอร์จะกำเนิดสัญญาณนาฬิกาผ่านขา SCL เพื่อควบคุมการเข้าถึงข้อมูล กำเนิดสัญญาณ START และสัญญาณ STOP



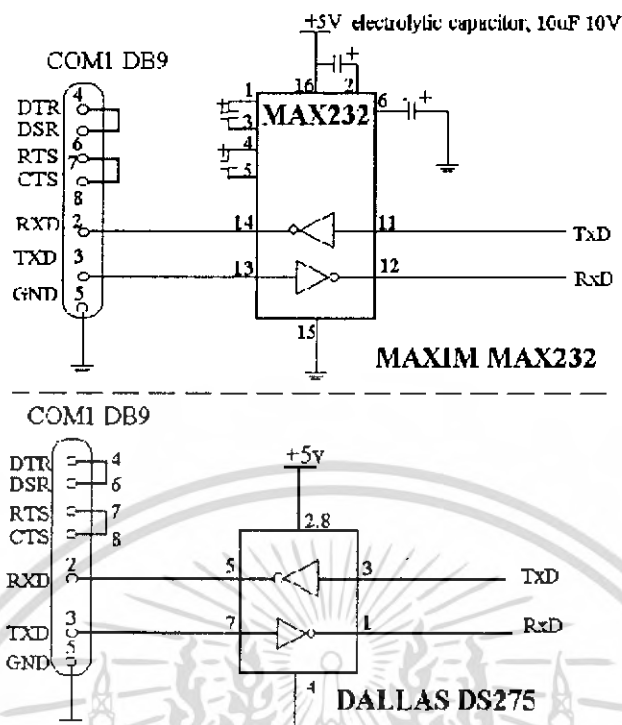
รูปที่ 2.20 แสดงการต่อสายสัญญาณ SDA และ SCL กับอุปกรณ์ต่างๆ

2.10 การเชื่อมต่อไมโครคอนโทรลเลอร์กับ RS-232 ผ่าน MAX232

โดยการต่อพอร์ตอนุกรมของ MCS-51 กับมาตรฐาน RS-232 ระดับสัญญาณของการสื่อสาร RS-232 ไม่เป็นไปตามมาตรฐานแรงดันที่ทีแอล เราจึงนำชิป MAX232 มาช่วยปรับระดับแรงดันให้กับ MCS-51 ได้

พอร์ตที่ใช้ในการรับ-ส่งข้อมูลแบบอนุกรมของ MCS-51 จะรับ-ส่งออกมาทางขา RxD และ ขา TxD ซึ่งอยู่ในพอร์ต 3 (P3.0 และ P3.1) โดย RxD จะเป็นขาที่ 10 TxD จะเป็นขาที่ 11 การรับ-ส่งข้อมูลกับมาตรฐาน RS-232 เราจะต้องติดต่อผ่านขาทั้งสองนี้โดยสามารถนำไอซี MAX232 มาเชื่อมต่อได้ โดยไอซีตัวนี้จะใช้ไฟเลี้ยง +5 โวลต์ เท่ากับไฟเลี้ยงไมโครคอนโทรลเลอร์ แต่สามารถยกระดับแรงดันตั้งแต่ -25 โวลต์ ถึง +25 โวลต์ ได้โดยไม่ต้องใช้แหล่งจ่ายไฟแบบคู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 แสดงลักษณะการต่อใช้งาน MAX232 ร่วมกับ DB9

ความเร็วในการรับ-ส่งข้อมูลแบบพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 สามารถติดต่อสื่อสารกับคอมพิวเตอร์ส่วนบุคคลได้แต่ต้องกำหนดอัตราเร็วให้เท่ากัน ในการสื่อสารพอร์ตอนุกรมผ่าน RS-232 ของคอมพิวเตอร์ได้กำหนดค่าอัตราเร็วไว้หลายค่าตั้งแต่ 100 ถึง 9600 บิตต่อวินาที สำหรับคอมพิวเตอร์ที่ใช้ไมโครโปรเซสเซอร์เพนเทียมสามารถส่งข้อมูลได้ความเร็วสูงถึง 56 กิโลบิตต่อวินาที

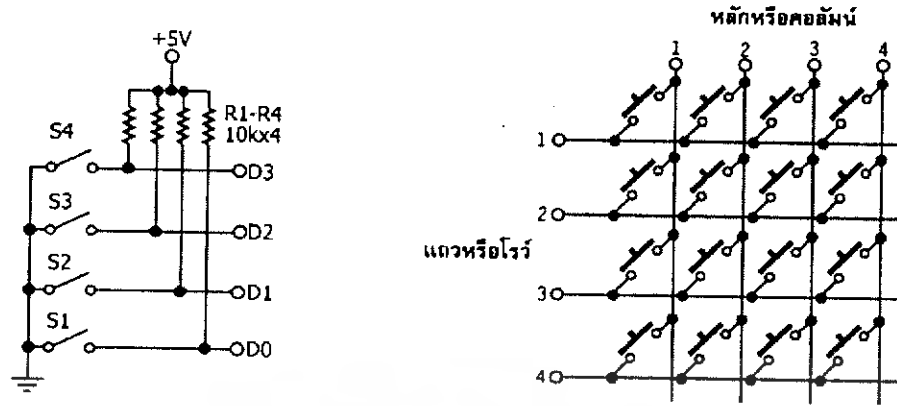
2.11 คีย์แพดและไอซีสแกนคีย์สวิตช์

คีย์แพดหรือสวิตช์แบบเมตริกซ์นี้จะเปรียบเหมือนสวิตช์ที่ถูกต้องกันในแนวตั้ง (Column) และในแนวนอน (Row) ดังนั้นค่าของสวิตช์จะต้องประกอบด้วย ตำแหน่งในแนวหลักและแถว กระบวนการที่จะทำได้มา ซึ่งค่าของสวิตช์มีขั้นตอนซับซ้อนพอควร แต่วงจรแบบสวิตช์นี้มีข้อดีคือสามารถรองรับการเพิ่มของสวิตช์ได้สะดวก เพียงแค่เพิ่มเติมจำนวนสวิตช์และแก้ไขซอฟต์แวร์เพียงเล็กน้อยเท่านั้น จึงทำให้วงจรสวิตช์แบบเมตริกซ์เป็นที่นิยมใช้กันแพร่หลายในระบบควบคุมอัตโนมัติหรือกึ่งอัตโนมัติ โดยในที่นี้ใช้คีย์แพด ขนาด 4 x 4 จะได้สวิตช์ทั้งหมดถึง 16 ตัว หรือ 16 คีย์

2.11.1 การใช้งานคีย์แพด

กระบวนการในการอ่านค่าของสวิตช์แบบนี้จะใช้วิธีการเขียนโปรแกรมในไมโครคอนโทรลเลอร์ ในทุกครั้งที่มีการส่งข้อมูลไปยังสายด้านแถวของคีย์แพด ไมโครคอนโทรลเลอร์จะทำการอ่านค่าที่ด้านหลักเข้ามาด้วย หากไม่มีการกดค่าด้านหลักก็จะเป็น "1" ทั้งหมด ถ้าหากว่ามีมีการกดคีย์แพดค่าของด้านหลักก็จะเป็น "0" อีกต่อไป ซึ่งเป็นการแจ้งให้ทราบว่ามีการกดคีย์แพด จากนั้นไมโครคอนโทรลเลอร์จะค้นหาตำแหน่งต่อไป ในการค้นหาตำแหน่งนั้นสิ่งที่ได้มาอย่างแรกคือตำแหน่งของคีย์นั้น จากนั้น

ก็จะหาตำแหน่งนั้นไปเปิดตารางข้อมูล เพื่อที่จะได้ค่าที่ต้องการนำไปแสดงผลที่หน้าจอใช้ประโยชน์ด้านการคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



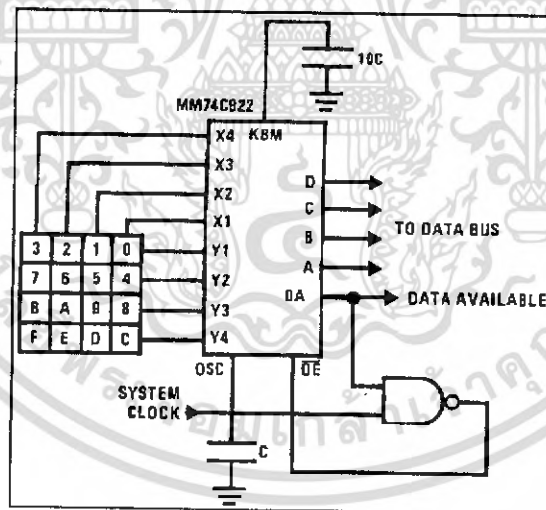
(ก) วงจรของสวิตช์ที่ต่อไฟเลี้ยงและกราวด์

(ข) วงจรของสวิตช์แบบเมตริกซ์แบบ 4 x 4

รูปที่ 2.22 แสดงลักษณะการต่อของวงจรของสวิตช์แบบเมตริกซ์หรือคีย์แพคแบบ 4 x 4

2.11.2 การใช้งานไอซีสแกนคีย์สวิตช์ (MM74C922N)

ในปัจจุบันได้มีการออกแบบไอซีสแกนคีย์สวิตช์แบบเมตริกซ์ โดยเอาต์พุตที่ออกมาจากไอซีจะเป็นเลขไบนารี จากนั้นจะนำไปต่อกับพอร์ตไมโครคอนโทรลเลอร์ทำให้ระบบไมโครคอนโทรลเลอร์ไม่ต้องเสียเวลาในการทำโปรแกรมสแกนคีย์สวิตช์ ซึ่งไอซีสแกนคีย์สวิตช์เบอร์ MM74C922N จะเป็นแบบ 16-Key Encoder ดังรูปที่ 2.23

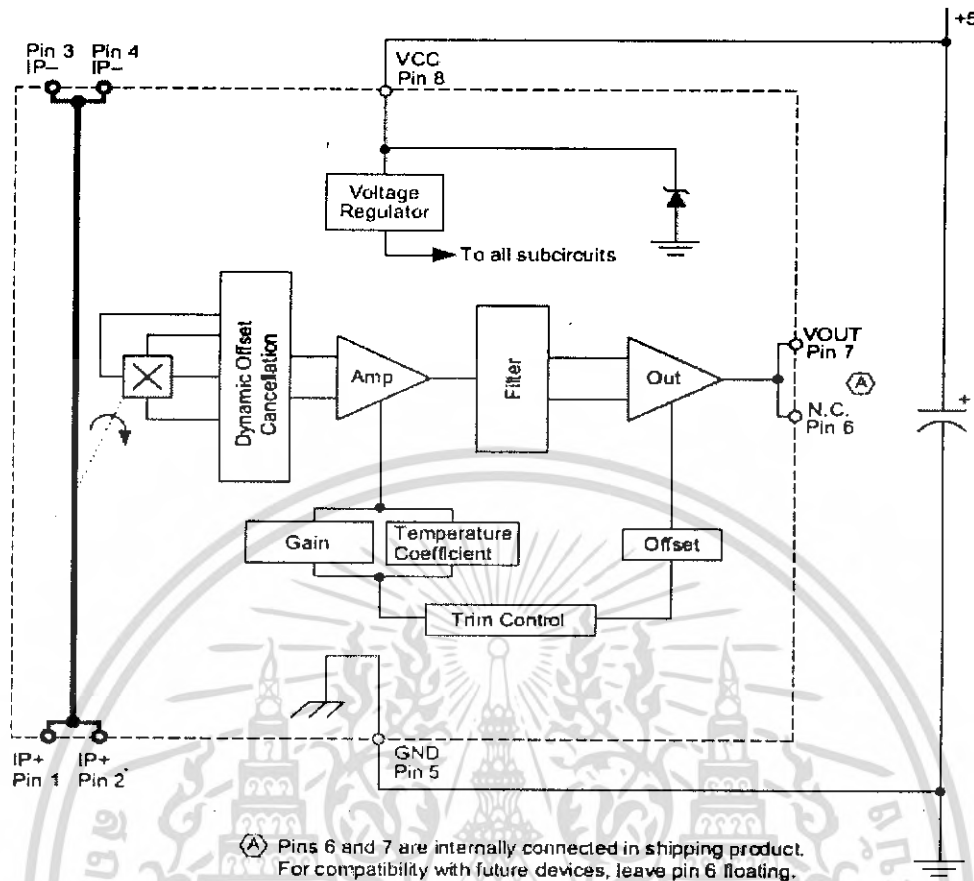


รูปที่ 2.23 แสดงการต่อใช้งานคีย์แพคกับไอซีสแกนคีย์สวิตช์ MM74C922N

2.12 เซนเซอร์กระแส (ไอซีเบอร์ ACS706ELC-20A)

เซนเซอร์กระแสนี้มีคุณสมบัติของความเป็นเชิงเส้น สำหรับกระแสอินพุตที่มีค่าอยู่ระหว่าง 0 ถึง 20 แอมแปร์ เป็นอุปกรณ์ที่เหมาะสมกับการใช้งานประเภทควบคุม ตัวเซนเซอร์สามารถวัดกระแสได้อย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 แสดงบล็อกโคแอมป์การทำงานของ ACS706ELC-20A

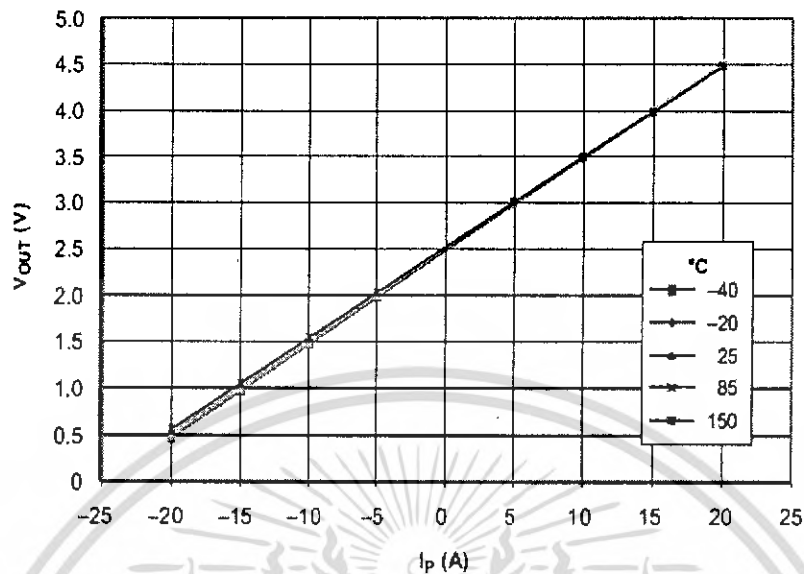
ถูกต้องแม่นยำสูง วงจรภายในเป็นแบบ Linear Hall Sensor ซึ่งการต่อวงจรจะต้องเดินลายทองแดงบนผิว เนื่องจากชิปของไอซีมีขนาดเล็กมาก (เพียง 5 มิลลิเมตร) การทำงานของไอซีจะเริ่มขึ้นเมื่อมีกระแสไหล ผ่านก็จะอาศัยคุณสมบัติของสนามแม่เหล็กไฟฟ้าที่มีอยู่ภายในชิปจึงทำให้แรงดันเอาต์พุตที่ได้เป็นสัดส่วน โดยตรงกับค่ากระแสอินพุตที่ป้อนให้กับไอซี แรงดันเอาต์พุตที่สภาวะปกติจะมีค่าไม่เกิน $VCC/2$ และภายในตัวไอซีสร้างมาจากอุปกรณ์สารกึ่งตัวนำจำพวก BiCMOS จากกราฟดังรูปที่ 2.25 นั้นแสดงให้เห็น ลักษณะของแรงดันเอาต์พุตต่ำสุดและสูงสุดที่ออกจากตัวเซนเซอร์ โดยสภาวะต่ำสุดคือเมื่อกระแสอินพุต มีค่าเป็น 0 แอมแปร์ จะได้แรงดันมีค่าเป็นเอาต์พุต 2.5 โวลต์ และสภาวะสูงสุดคือเมื่อกระแสอินพุตมีค่า เป็น 20 แอมแปร์ จะได้แรงดันมีค่าเป็นเอาต์พุต 4.5 โวลต์

การต่อใช้งานไอซีเซนเซอร์กระแส ACS706ELC-20A

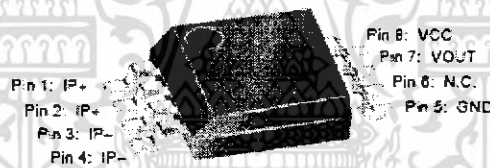
- ขา 1, 2 และขา 3, 4 เป็นขาอินพุตที่ต่อกับแหล่งจ่ายไฟฟ้ากระแสสลับ
- ขา 5 ต่อลงกราวด์ไฟฟ้ากระแสตรง
- ขา 7 เป็นขาแรงดันเอาต์พุต
- ขา 8 ต่อกับแหล่งจ่ายไฟฟ้ากระแสตรง +5 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Output Voltage versus Primary Current
 $V_{CC} = 5\text{ V}$



รูปที่ 2.25 แสดงกราฟความเป็นเชิงเส้นระหว่างกระแสอินพุตกับแรงดันเอาต์พุต



รูปที่ 2.26 แสดงลักษณะตัวอุปกรณ์เซนเซอร์กระแส ACS706ELC-20A

2.13 RF-Wireless โมดูล

RF-Wireless โมดูล ซึ่งมีการมอดูเลตแบบจีโอเอฟเอสเค โดยเป็นชุด Signal Converter สำหรับใช้ในการแปลงสัญญาณระหว่าง RS-232 และ RF-Wireless โดยในโหมดการทำงานของการส่งข้อมูล เครื่องส่งจะทำหน้าที่ที่รองรับข้อมูลจากพอร์ตสื่อสารอนุกรม RS-232 จากขา TxD แล้วแปลงเป็นสัญญาณความถี่จีโอเอฟเอสเคส่งออกไปในอากาศ และในทางกลับกันในโหมดการทำงานแบบรับ เครื่องรับที่ชุด RF-Wireless โมดูล ก็จะทำหน้าที่ที่คอยตรวจจับข้อมูลที่อยู่ในรูปของสัญญาณความถี่จีโอเอฟเอสเคจากด้าน RF เพื่อแปลงข้อมูลกลับโดยผ่าน RS-232 ส่งออกไปทางขา RxD เข้าไมโครคอนโทรลเลอร์

สามารถนำไปต่อใช้งานร่วมกับพอร์ตสื่อสารอนุกรม RS-232 เพื่อใช้งานในลักษณะการสื่อสารอนุกรมแบบไร้สายได้โดยตรง ซึ่งจะมีข้อดีกว่าคือสามารถรับ-ส่งข้อมูลได้ในระยะทางที่ไกลกว่า RS-232 หลายเท่าตัว และอีกประการสำคัญคือไม่จำเป็นต้องใช้สายสัญญาณที่เป็นตัวนำสัญญาณทางไฟฟ้าในการสื่อสารข้อมูลกัน ทำให้สามารถเปลี่ยนแปลงหรือเคลื่อนย้ายจุดรับ-ส่งข้อมูลได้ตลอดเวลา ถ้าเป็นการรับ-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งข้อมูลด้วยระบบ RS-232 แบบที่ใช้สายสัญญาณก็จะเกิดความยุ่งยากในเรื่องของการติดตั้งสายสัญญาณเป็นอย่างมาก

2.14 รูปแบบของการสื่อสาร

รูปแบบของการสื่อสารคือ ลักษณะของการติดต่อสื่อสารระหว่าง DTE คู่หนึ่ง ซึ่งแบ่งออกได้เป็น 3 แบบ คือ

2.14.1 การติดต่อทางเดียว (Simplex) การสื่อสารแบบทางเดียว ก็คือ มีการส่งสัญญาณจากอุปกรณ์เพียงด้านเดียว และสัญญาณเดินทางไปทิศทางเดียว

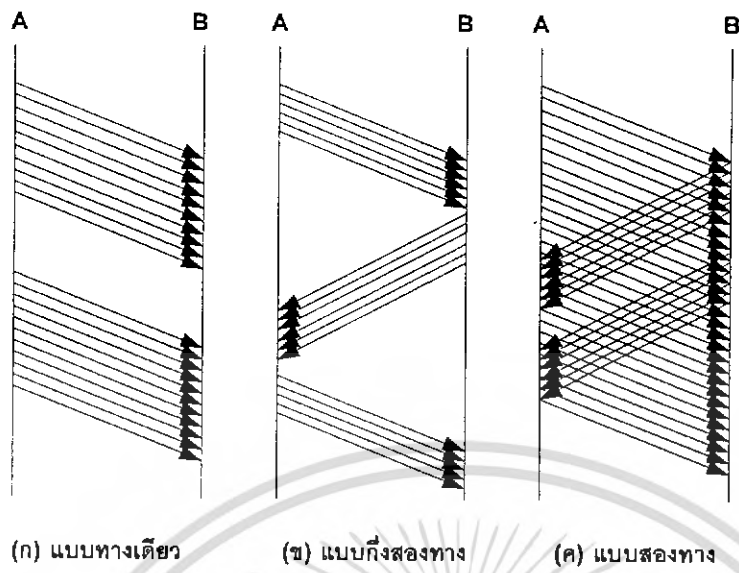
กรณีที่รูปแบบของการสื่อสารเป็นแบบทางเดียว อุปกรณ์ทางด้านส่งจะไม่มีโอกาสรู้ว่าอุปกรณ์ทางด้านรับรับสัญญาณข้อมูลได้ถูกต้องหรือไม่ ดังนั้นเพื่อการประกันความเชื่อถือของระบบจึงจำเป็นต้องส่งรหัสแก้ไขการผิดพลาดที่ถี่ๆ ฟังไปด้วย หรือไม่ก็ส่งสัญญาณข้อมูลชุดเดียวกันซ้ำกันหลายๆ ครั้ง ซึ่งวิธีการทั้งสองนี้ล้วนแต่ทำให้ประสิทธิภาพของการใช้ระบบเลวลงอย่างเห็นได้ชัดทั้งสิ้น

2.14.2 การติดต่อทั้งสองทาง (Half-duplex) ส่วนแบบสองทางนั้น การส่งสัญญาณจะเกิดขึ้นได้ทั้งสองด้าน แต่เวลาจะไม่ตรงกันคือจะผลัดกันส่งและผลัดกันรับ เช่น A และ B ต้องการจะพูดคุยกันถ้า A พูด B ต้องรอให้ A พูดให้จบก่อน B ถึงจะพูดตอบกลับไปได้

2.14.3 การติดต่อสองทาง (Full-Duplex) สำหรับแบบสองทิศทางนั้น การส่งสัญญาณจากทั้งสองด้านจะเกิดขึ้นได้พร้อมกัน เช่น A และ B พูดคุยกัน A และ B สามารถพูดพร้อมกันได้เลยโดยไม่ต้องรอให้อีกฝ่ายหยุดพูด

กรณีที่รูปแบบของการสื่อสารเป็นแบบทั้งสองทางและแบบสองทางนั้น การส่งสัญญาณทำได้ทั้งสองทิศทาง ดังนั้นถ้าแบ่งสัญญาณข้อมูลออกเป็นบล็อกที่มีความยาวพอเหมาะจะสามารถใช้รหัสตรวจจับการผิดพลาดแบบง่ายๆ ได้ และเมื่อปลายทางพบที่มีการผิดพลาดเกิดขึ้นก็จะส่งสัญญาณมาบอกทางด้านต้นทางให้ส่งบล็อกของสัญญาณที่ผิดพลาดไปอีกครั้ง ส่วนระบบสองทางนั้นอุปกรณ์ทางด้านส่งในขณะที่ทำการส่งอยู่ จะได้รับข่าวสารจากทางด้านรับเกี่ยวกับบล็อกข้อมูลที่ส่งไปก่อน ซึ่งทำให้การตอบสนองทำได้รวดเร็ว นอกจากนี้ยังสามารถทำการส่งข้อมูลปริมาณมากๆ จากอุปกรณ์ทั้งสองด้านในเวลาเดียวกันได้ จึงนับว่าเป็นระบบที่มีประสิทธิภาพในการทำงานสูง

รูปแบบการสื่อสารจะสัมพันธ์กับความน่าเชื่อถือของการส่งผ่านข่าวสารและประสิทธิภาพในเชิงการใช้เวลา รหัสข้อมูลที่ใช้ในการสื่อสารข้อมูลนั้นโดยปกติจะพยายามใช้จำนวนบิตที่น้อยที่สุดจึงทำให้เกิดปัญหาด้านความซ้ำซ้อน (Redundancy) คือถ้าในระหว่างทางที่ส่งอยู่เกิดมีการผิดพลาดขึ้นเพียง 1 บิต อุปกรณ์ปลายทางจะตีความหมายรหัสผิดพลาดได้ ระบบส่งสัญญาณข้อมูลโดยทั่วไปมักจะมีการผิดพลาดเกิดขึ้นระหว่างการส่งได้เสมอ ดังนั้นการหามาตรการจัดการกับความผิดพลาดที่อาจจะเกิดขึ้นได้นี้จึงเป็นหัวข้อที่สำคัญในงานด้านสื่อสารข้อมูล วิธีทั่วไปอาจจะใช้วิธีส่งรหัสตรวจจับการผิดพลาดหรือรหัสแก้ไขการผิดพลาดพ่วงเข้าไปด้วย นอกจากนี้อาจจะใช้วิธีควบคุมการส่งข้อมูลระหว่าง DTE ในการแก้ไขความผิดพลาดที่เกิดขึ้นได้



รูปที่ 2.27 แสดงรูปแบบของการสื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

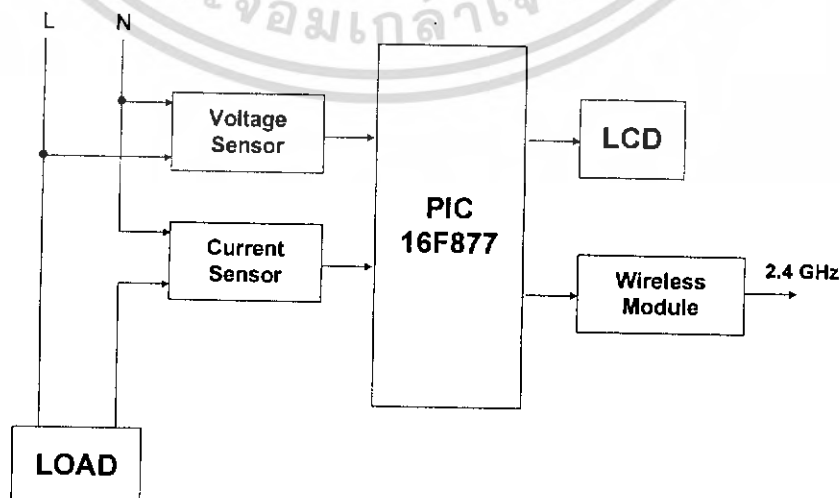
การคำนวณและการสร้าง

3.1 หลักการเบื้องต้น

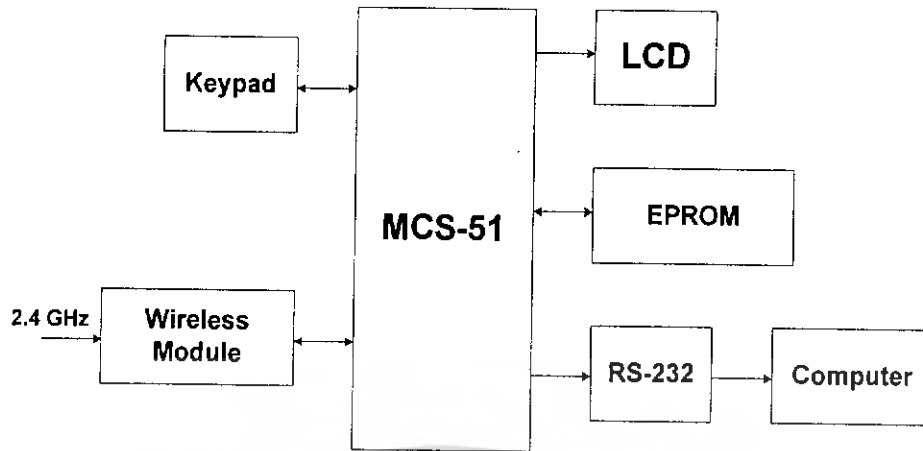
การคำนวณและการสร้างวัดคัมมิเตอร์ไร้สายเก็บข้อมูลการใช้กำลังงานไฟฟ้าสามารถแบ่งหลักการทำงานออกเป็น 2 ส่วน คือ ฮาร์ดแวร์และซอฟต์แวร์ ส่วนของฮาร์ดแวร์นั้นแบ่งการทำงานออกอีก 2 ส่วน คือ ชุดภาคส่งและชุดภาครับ และส่วนของซอฟต์แวร์จะมีโปรแกรมการคำนวณค่ากำลังงานไฟฟ้า การแสดงผล และการควบคุมต่างๆ ที่ใช้ในการวัดกำลังงานไฟฟ้าที่ปรากฏและแสดงผลในหน่วยของวัตต์ จึงสามารถกล่าวได้ว่าใช้หลักในการแปลงแรงดันไฟฟ้ากระแสสลับมาอยู่ในรูปของแรงดันไฟฟ้ากระแสตรงก่อน และนำค่าเอาต์พุตของวงจรตรวจจับกระแสป้อนให้กับไมโครคอนโทรลเลอร์ PIC16F877 เพื่อใช้ประมวลผลการคำนวณค่ากำลังงานไฟฟ้าที่ปรากฏเมื่อคิดเทียบกับเวลา โดยหลักในการคำนวณจะกล่าวถึงต่อไป

3.2 ส่วนของฮาร์ดแวร์

ส่วนของฮาร์ดแวร์ประกอบด้วยชุดภาคส่งและชุดภาครับ ในชุดภาคส่งจะมีวงจรตรวจจับแรงดันและวงจรตรวจจับกระแส จากเอาต์พุตที่ได้ของวงจรทั้งสองเป็นสัญญาณแบบอนาล็อก ดังนั้นจึงนำไปป้อนเป็นอินพุตให้กับไมโครคอนโทรลเลอร์ PIC เนื่องจากภายในของไมโครคอนโทรลเลอร์ PIC มีคุณสมบัติการแปลงสัญญาณอนาล็อกเป็นดิจิทัลในตัว จากนั้นไมโครคอนโทรลเลอร์จะประมวลผลก่อนส่งออกไปแสดงที่จอแอลซีดีโมดูล ถ้ามีการร้องขอข้อมูลจากภาครับ ภาคส่งก็จะส่งข้อมูลกำลังงานไฟฟ้าออกผ่านทาง RF-Wireless โมดูล สำหรับชุดภาครับเมื่อได้รับสัญญาณคลื่นวิทยุความถี่ 2.4 กิกะเฮิร์ตซ์ แล้วจะนำสัญญาณที่ได้ไปเก็บไว้ในหน่วยความจำ โดยใช้ไมโครคอนโทรลเลอร์ MCS-51 ประมวลผลและควบคุมการทำงานทางด้านชุดภาครับ ถ้าต้องการจะถ่ายโอนข้อมูลไปยังคอมพิวเตอร์ก็จะส่งข้อมูลที่ได้รับเข้ามาผ่านทางพอร์ต RS-232 เพื่อแสดงผลข้อมูลการใช้กำลังงานไฟฟ้าผ่านไฮเปอร์เทอร์มินอลบนจอคอมพิวเตอร์ การทำงานของวัดคัมมิเตอร์ไร้สายเก็บข้อมูลการใช้กำลังงานไฟฟ้าที่ประกอบด้วยชุดรับ-ส่ง สามารถแสดงได้จากบล็อกไดอะแกรม ดังรูปที่ 3.1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ (ก) บล็อกไดอะแกรมทางภาคชุดส่ง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



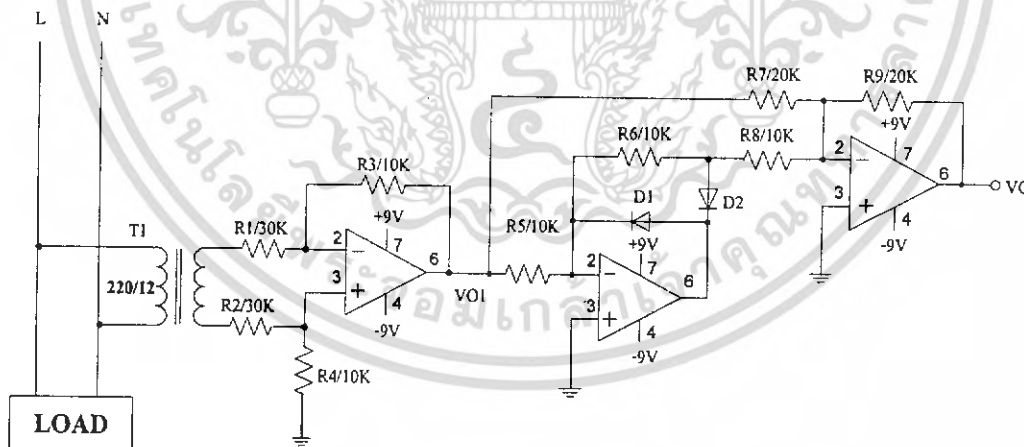
(ข) บล็อกไดอะแกรมทางภาคชุดรับ

รูปที่ 3.1 แสดงบล็อกไดอะแกรมภาคชุดรับ-ส่ง วัตต์มิเตอร์ไร้สายเก็บข้อมูลการใช้กำลังงานไฟฟ้า

3.3 ฮาร์ดแวร์ภาคส่ง

3.3.1 วงจรตรวจจับแรงดัน (Voltage Sensor)

วงจรตรวจจับแรงดันใช้แนวคิดในการลดปริมาณของแรงดันไฟฟ้ากระแสสลับ 220 โวลต์ โดยการใช่มือแปลงไฟฟ้าขนาด 220:12 เพื่อลดอันตรายจากการต่อวงจร โดยตรงกับแรงดันไฟฟ้าที่มีปริมาณสูงและยังช่วยป้องกันความเสียหายหากเกิดความผิดพลาดภายในวงจรได้



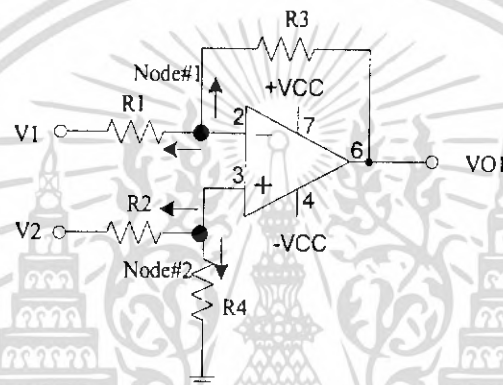
รูปที่ 3.2 แสดงการต่อวงจรตรวจจับแรงดัน

การทำงานของวงจรตรวจจับแรงดันสามารถแบ่งเป็นวงจรย่อยๆ ได้ 2 วงจร ดังนี้

1. วงจรขยายผลต่าง (Voltage Subtracting Amplifier)

เมื่อทราบว่าแรงดันเอาต์พุตของหม้อแปลงมีค่าแรงดันไฟฟ้ากระแสสลับ 12 โวลต์ โดยจะนำแรงดันส่วนนี้ไปอินพุตให้เป็นแรงดันอินพุตให้กับวงจรขยายผลต่างนั่นคือ V1 และ V2 เมื่อแรงดันอินพุตเป็นเอกสักรวมเป็นเอกสักรหัสลงวินโดวส์ให้คลิกขวาเพื่อแก้ไขให้มัน เมื่อผู้ดูแลเห็นแบบนี้เป็นการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟฟ้ากระแสสลับ 12 โวลต์ แล้วจึงทำให้สามารถพิจารณาได้ 2 กรณีคือ กรณีแรกจะได้ $V_1=0$ โวลต์ กับ $V_2=12$ โวลต์และกรณีที่สองจะได้ $V_1=12$ โวลต์กับ $V_2=0$ โวลต์ สามารถสังเกตได้ว่าแรงดันทั้งสองจะมีค่าสลับกันเช่นนี้อยู่ตลอดเวลาเพราะว่าเป็นแรงดันไฟฟ้ากระแสสลับ ในการออกแบบพารามิเตอร์ค่าความต้านทานแต่ละตัวภายในวงจรขยายผลต่างเพื่อต้องการให้ค่าแรงดันเอาต์พุต V_{O1} ที่มีค่าได้ไม่เกิน 5 โวลต์ เพราะจุดประสงค์ของการกำหนดแรงดันเอาต์พุตวงจรรขยายผลต่างนี้ ก็เพื่อต้องการนำไปป้อนเป็นแรงดันอินพุตให้กับไมโครคอนโทรลเลอร์ PIC16F877 แรงดันทางด้านอินพุตที่ป้อนให้กับไมโครคอนโทรลเลอร์ต้องมีขอบเขตที่จำกัดเพราะว่าแรงดันไฟฟ้าที่ป้อนต้องเป็นซิกบวกเท่านั้น การแปลงสัญญาณไฟฟ้ากระแสสลับให้มีค่าเฉพาะช่วงบวกนี้จะกล่าวถึงในวงจรเรียงกระแสเต็มรูป สำหรับแนวคิดในการคำนวณค่าพารามิเตอร์ต่างๆ ของวงจรมีดังนี้



รูปที่ 3.3 แสดงวงจรรขยายผลต่าง

- พิจารณา Node#1 จะได้

$$\frac{V_x - V_1}{R_1} + \frac{V_x - V_{O1}}{R_3} = 0$$

$$\left(\frac{R_1 + R_3}{R_1 R_3} \right) \cdot V_x - \frac{V_1}{R_1} - \frac{V_{O1}}{R_3} = 0$$

$$V_{O1} = \left(\frac{R_1 + R_3}{R_1 R_3} \right) \cdot V_x - \frac{R_3}{R_1} \cdot V_1 \quad (3.1)$$

- พิจารณา Node#2 จะได้

$$\frac{V_x - V_2}{R_2} + \frac{V_x}{R_4} = 0$$

$$\left(\frac{R_2 + R_4}{R_2 R_4} \right) \cdot V_x - \frac{V_2}{R_2} = 0$$

$$V_x = \left(\frac{R_4}{R_2 + R_4} \right) \cdot V_2 \quad (3.2)$$

- แทนค่าสมการที่ (2) ในสมการ (1) จะได้

$$V_{O1} = \frac{R_4(R_1 + R_3)}{R_1(R_2 + R_4)} \cdot V_2 - \frac{R_3}{R_1} \cdot V_1 \quad (3.3)$$

จากสมการที่ (3.3) เมื่อได้ค่าของเอาต์พุต V_{O1} จึงสามารถกำหนดค่าความต้านทานต่างๆ ภายในวงจรได้ด้วยการเลือกพิจารณาแรงดันที่ V_1 และ V_2 สลับกัน โดยต้องการแรงดันเอาต์พุต $V_{O1} = 4$ โวลต์ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะทำให้สามารถเลือกกำหนดค่าความต้านทานให้ได้ตามสัดส่วนลงตัวพอดีและยังสามารถช่วยป้องกัน หากแรงดันไฟฟ้าที่ V_{o1} ให้มีค่ามากกว่า 4 โวลต์ แต่สามารถเป็นแรงดันอินพุตที่ทำให้ไมโครคอนโทรลเลอร์แปลงสัญญาณอนาล็อกเป็นดิจิทัลยังทำงานได้อยู่ ในส่วนของการแปลงแปลงสัญญาณอนาล็อกเป็นดิจิทัลต้องการแรงดันอินพุตที่อยู่ในช่วงระหว่าง 0 ถึง 5 โวลต์

- กรณีแรก $V_1=0$ โวลต์ $V_2=12$ โวลต์

$$4 = -\frac{R_3}{R_1} (12V)$$

$$\frac{1}{3} = -\frac{R_3}{R_1}$$

$$R_1 = 3R_3 \quad (3.4)$$

- กรณีที่สอง $V_1=12$ โวลต์ $V_2=0$ โวลต์

$$4 = \frac{R_4(R_1+R_3)}{R_1(R_2+R_4)} \cdot 12V$$

$$\frac{1}{3} = \frac{R_4(1+3)}{3(R_2+R_4)}$$

$$1 = \frac{4R_4}{R_2+R_4}$$

$$R_2+R_4 = 4R_4$$

$$R_2 = 3R_4 \quad (3.5)$$

ดังนั้นจึงได้ว่า $R_1=3R_3$ และ $R_2=3R_4$ ทำให้สามารถกำหนดค่าความต้านทานได้ดังนี้คือ

$$R_1=R_2=30K\Omega$$

$$R_3=R_4=10K\Omega \quad (3.6)$$

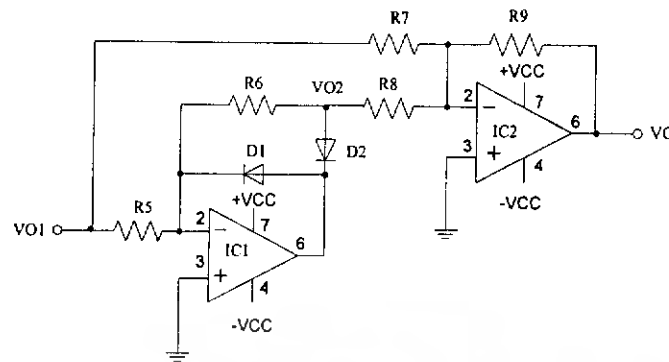
เมื่อกำหนดค่าความต้านทานทุกตัวแล้วสามารถเขียนรูปสมการใหม่ได้คือ

$$V_{o1} = \frac{1}{3}(V_2 - V_1) \quad (3.7)$$

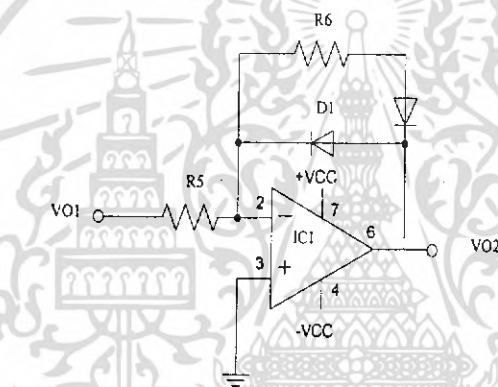
2. วงจรเรียงกระแสเต็มรูป

วงจรเรียงกระแสเต็มรูปมีการทำงานร่วมกันอยู่ 2 วงจรคือ วงจรขยายแบบกลับเฟสที่มีอัตราขยายเท่ากับ 1 และภายในวงจรมีไดโอดต่อรวมอยู่ด้วยเพื่อนำเฉพาะแรงดันเอาต์พุตที่เป็นแรงดันไฟฟ้าช่วงซีกลบไปคำนวณเท่านั้น และอีกวงจรหนึ่ง คือวงจรขยายผลบวก

การทำงานของวงจรขยายแบบกลับเฟสคือ เมื่อมีแรงดันอินพุตป้อนเข้ามาทางด้าน V_{o1} จากวงจรตรวจจับแรงดัน จากนั้นวงจรเรียงกระแสเต็มรูปจะทำงานโดย IC1 จะอาศัยคุณสมบัติของวงจรขยายแบบกลับเฟส



รูปที่ 3.4 แสดงวงจรเรียงกระแสเต็มรูป



รูปที่ 3.5 แสดงวงจรขยายแบบกลับเฟส ที่มีไดโอดค่อร่วมด้วย

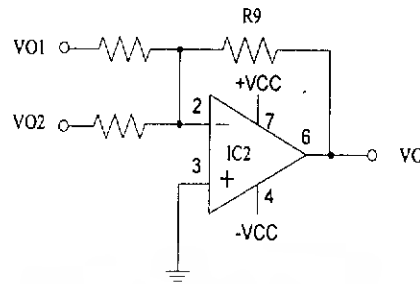
จากรูปที่ 3.5 สามารถแสดงเป็นสมการได้คือ

$$\begin{aligned}
 V_{O2} &= -\frac{R_6}{R_5} \cdot V_{O1} \\
 &= -\frac{10\text{ K}}{10\text{ K}} \cdot 4\text{V} \\
 &= -4\text{V}
 \end{aligned}
 \tag{3.8}$$

ดังนั้น $V_{o2} = -V_{o1}$ เป็นแรงดันไฟลบจึงสามารถไหลผ่านไดโอด D2 ได้ เพราะว่าไดโอด D2 ต่อแบบไบอัสกลับ (Reverse Bias) และต่อไดโอด D1 แบบไบอัสตรง (Forward Bias) เพื่อป้องกันไม่ให้แรงดันไฟลบที่เอาต์พุต V_{o2} ไหลผ่านไดโอด D1 ได้และอีกทางมีไดโอด D1 ต่อกับขาอินเวอร์ตติ้งของออปแอมป์ เพื่อป้องกันแรงดันไฟบวกไม่ให้ผ่านออกไปได้ เพื่อต้องการแรงดันเอาต์พุต V_{o2} ที่เป็นค่าลบเพราะจะนำไปคำนวณหาค่าของ V_o แล้วทำให้ได้ค่าที่มีแต่เฉพาะช่วงบวกเท่านั้น โดยค่าที่คิดลบจะหักล้างกันไปเองในภายหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของวงจรขยายผลบวกคือ IC2 จะรับแรงดันอินพุตมา 2 ค่า ได้แก่ V_{O1} และ V_{O2} โดยสามารถคำนวณได้ดังนี้



รูปที่ 3.6 แสดงวงจรขยายผลบวก

$$\begin{aligned} V_O &= - \left(\frac{R_9}{R_7} \cdot V_{O1} + \frac{R_9}{R_8} \cdot V_{O2} \right) \\ &= - \left(\frac{20K}{20K} \cdot 4V + \frac{20K}{10K} \cdot (-4V) \right) \\ &= - (4V + 2(-4V)) \\ &= - (4V - 8V) \\ &= - (-4V) \\ &= 4V \end{aligned}$$

ดังนั้นจะได้สมการ $V_O = |V_{O1}|$ (3.9)

จากการคำนวณข้างต้นสามารถสรุปได้ว่า ถ้าต้องการแรงดันเอาต์พุตของวงจรเรียงกระแสเต็มรูปให้มีค่าเท่าไรก็ได้ตามต้องการเพียงแต่กำหนดแรงดันอินพุตที่ป้อนเข้ามา เนื่องจากวงจรเรียงกระแสเต็มรูปจะให้ค่าแรงดันเอาต์พุตเท่ากับแรงดันอินพุตแต่แรงดันเอาต์พุตจะมีค่าเฉพาะช่วงซีกบวก

จากสมการเอาต์พุตของวงจรเรียงกระแสเต็มรูปเป็น $V_O = \frac{R_9}{R_7} |V_{O1}|$ แต่ถ้าไม่ต้องการให้มีการขยายสัญญาณจะได้สมการใหม่เป็น $V_O = |V_{O1}|$ โดยสามารถกำหนดค่าได้ดังนี้คือ ความต้านทาน $R_9 = R_7$ และความต้านทาน $R_8 = R_7/2$ ส่วนของวงจรขยายแบบกลับเฟสนั้นสามารถเลือกแบบใดก็ได้ที่มีอัตราขยายเท่ากับ 1

การกำหนดแรงดันเอาต์พุตของวงจรตรวจจับแรงดันให้มีค่า 4 โวลต์เฉพาะช่วงบวก เนื่องจากการแปลงสัญญาณอนาล็อกเป็นดิจิทัลของไมโครคอนโทรลเลอร์ PIC16F877 จะต้องแปลงสัญญาณเอาต์พุตจากวงจรตรวจจับแรงดันที่อยู่ในรูปของสัญญาณที่มีทั้งซีกบวกและซีกลบ ให้เหลือสัญญาณเฉพาะซีกบวกเท่านั้น (สัญญาณที่มีเฉพาะซีกบวกคือสัญญาณหลังจากผ่านวงจรเรียงกระแสเต็มรูป) เพราะการแปลงสัญญาณอนาล็อกเป็นดิจิทัลต้องมีแรงดันอินพุตเฉพาะซีกบวกจึงจะทำให้ไมโครคอนโทรลเลอร์ PIC16F877 ทำงานได้

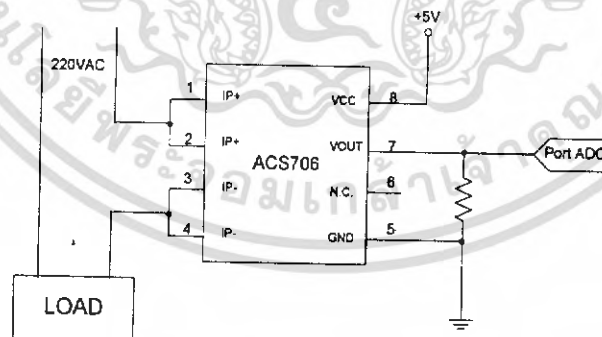
3.3.2 วงจรตรวจจับกระแส (Current Sensor)

วงจรตรวจจับกระแสซึ่งใช้ไอซีเบอร์ ACS706ELC-20A เนื่องจากไอซีนีสามารถใช้เป็นเซนเซอร์กระแสได้เหมือนกับ ซีที แต่จะแตกต่างกันตรงที่เซนเซอร์กระแสนี้มีความเป็นเชิงเส้นและมีความถูกต้อง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แม่นยำในการวัดกระแสมากเมื่อเทียบกับ ซีที ที่ให้เอาต์พุตออกมาไม่เป็นเชิงเส้น และอินพุตกับเอาต์พุตจะมีความเป็นอิสระต่อกัน โดยใช้หลักการของ Hall Effect ดังนั้นจึงเลือกไอซีเบอร์นี้มาใช้ในวงจรตรวจจ่ายกระแส สำหรับย่านที่ใช้ในการวัดค่ากระแสอินพุตของไอซีให้อยู่ในช่วง 0 ถึง 15 แอมแปร์ และจะได้เอาต์พุตเป็นแรงดันไฟฟ้าที่มีค่าเปลี่ยนแปลงได้ไม่เกิน 2 โวลต์ คือระหว่าง 2.5 ถึง 4.5 โวลต์ เพราะที่ขณะไม่มีการต่อโหลดจะได้แรงดันเอาต์พุต 2.5 โวลต์ แต่ถ้าต่อโหลดจะได้แรงดันเอาต์พุตที่มีค่าเปลี่ยนแปลงตามค่ากระแสอินพุตที่ได้จากโหลดนั้นๆ เนื่องจากแรงดันเอาต์พุตเป็นสัดส่วนกับกระแสอินพุตโดยตรง วงจรตรวจจ่ายกระแสจะทำงานได้ก็ต่อเมื่อมีการต่อโหลดใช้งานเท่านั้น ถึงแม้จะจ่ายแรงดันอินพุต 220 โวลต์เข้ามาแล้วก็ตาม หากไม่ต่อโหลดก็จะไม่มีกระแสไหลผ่านที่ตัวเซนเซอร์กระแส ถึงแม้ว่าจะมีแรงดันเอาต์พุตออก 2.5 โวลต์ แต่แรงดันนั้นไม่มีการเปลี่ยนแปลงก็ไม่เกิดประโยชน์อะไร แล้วยังส่งผลถึงค่ากำลังงานไฟฟ้าที่ไม่สามารถเกิดขึ้นได้เช่นกัน เนื่องจากกำลังงานไฟฟ้านี้เกิดจากการคูณกันของค่าแรงดันกับกระแสเทียบกับเวลา เมื่อไม่มีโหลดก็ไม่เกิดกระแสจึงเปรียบได้กับว่ามีค่าเป็นศูนย์นั่นเอง

การทำงานของวงจรตรวจจ่ายกระแส

เมื่อนำโหลดลักษณะต่างๆมาต่อกับแรงดันไฟฟ้ากระแสสลับ 220 โวลต์และจ่ายไฟเลี้ยง 5 โวลต์ให้แก่วงจรตรวจจ่ายกระแสแล้ว คุณสมบัติของตัวไอซีจะมีอัตราส่วนของกระแสดังนี้คือ กระแสอินพุต 1 แอมแปร์ได้แรงดันเอาต์พุต 100 มิลลิโวลต์ โดยการต่อใช้งานจะต่อขา 1 และ 2 เข้าด้วยกันเพื่อแบ่งกระแสที่ไหลผ่านตัวไอซีให้ไหลได้มากขึ้น (ไหลได้ทั้งสองขา) ดังรูปที่ 3.7 และกระแสจะไหลจากขา 1, 2 ไป 3, 4 เกิดการเหนี่ยวนำสนามแม่เหล็กไฟฟ้าออกมาที่ขา 7 อย่างเป็นสัดส่วนต่อกัน จากปกติที่ขณะไม่ต่อโหลดจะมีแรงดันเอาต์พุตไฟฟ้ากระแสตรงที่ระดับ 2.5 โวลต์ แต่เมื่อต่อโหลดเข้าไปเกิดการเปลี่ยนแปลงด้านอินพุต ทำให้แรงดันเอาต์พุตที่เป็นไฟฟ้ากระแสสลับเปลี่ยนแปลงรูปสัญญาณอยู่บนระดับแรงดันอ้างอิงที่เป็นไฟฟ้ากระแสตรง หลังจากนั้นจึงนำแรงดันเอาต์พุตที่ได้ป้อนให้กับไมโครคอนโทรลเลอร์เพื่อใช้แปลงเป็นสัญญาณอนาล็อกเป็นดิจิตอลต่อไป



รูปที่ 3.7 แสดงลักษณะการต่อวงจรตรวจจ่ายกระแส

3.3.3 การทำงานของไมโครคอนโทรลเลอร์ PIC16F877

ลักษณะการทำงานของไมโครคอนโทรลเลอร์ PIC เลือกใช้ไอซีเบอร์ PIC16F877 สำหรับทำงานในส่วนต่างๆ ทางภาคส่งของโครงการ มีดังนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ภาคควบคุมการคำนวณค่ากำลังงานไฟฟ้าเทียบกับเวลา โดยการนำค่าแรงดันเอาต์พุตจากวงจรตรวจับกระแสและตรวจับแรงดันที่มีค่าไม่เกิน 5 โวลต์ สัญญาณที่ได้มีเฉพาะช่วงบวกเท่านั้น เพื่อนำสัญญาณไปป้อนให้แก่ไมโครคอนโทรลเลอร์ PIC16F877 ใช้ในการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลแรงดันอินพุตที่มีค่าอยู่ระหว่าง 0 ถึง 5 โวลต์ โดยนำเอาต์พุตจากวงจรตรวจับแรงดันจะป้อนเข้าที่ขา 4 หรือบิต RA2 ของไมโครคอนโทรลเลอร์ และเอาต์พุตจากวงจรตรวจับกระแสจะป้อนเข้าที่ขา 3 หรือบิต RA1 ของไมโครคอนโทรลเลอร์เพื่อใช้ในการประมวลผล การคำนวณค่ากำลังงานไฟฟ้านั้นจะใช้การเขียนโปรแกรมภาษาซี โดยที่ใช้วงจรเอชดีที่มีอยู่ภายในตัวของไมโครคอนโทรลเลอร์แปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล แล้วนำผลที่ได้จากอินพุตทั้งสองคูณกันและคิดเทียบกับเวลา 1 วินาที จากโปรแกรมจะสั่งให้แอมป์ลิฟาย์ทุกคาบเวลาของเอาต์พุตทั้งสอง และในแต่ละคาบเวลาจะเลือกค่าที่สูงสุดของแต่ละอินพุตเก็บไว้และนำมาประมวลผล เพื่อนำค่าที่ได้ไปคำนวณคิดค่ากำลังงานไฟฟ้าที่ปรากฏ

2. ภาคควบคุมการแสดงผลแอลซีดีโมดูล ซึ่งการเชื่อมต่อแอลซีดีโมดูลขนาด 16 ตัวอักษร 2 แถวกับไมโครคอนโทรลเลอร์นั้นผ่านทางพอร์ต B โดยให้แอลซีดีโมดูลแสดงค่ากำลังงานไฟฟ้าที่ได้เทียบกับเวลาในหน่วยของวัตต์ และหมายเลขของเครื่องนั้นๆ

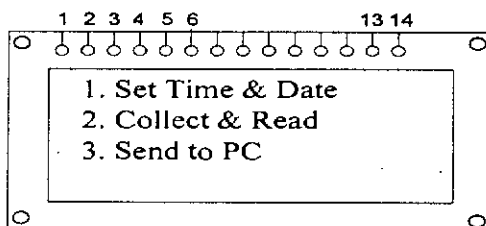
3. ควบคุมการส่งข้อมูลออกให้กับ RF-Wireless โมดูล เพื่อส่งข้อมูลออกไปให้กับชุดภาครับแบบไร้สายด้วยความถี่ 2.4 กิกะเฮิร์ตซ์ เมื่อมีการร้องขอข้อมูลที่ตัวคัมมิเตอร์ตัวนั้นๆ ต่อผ่านขา TxD ทางขา 25 บิต RC6 ของไมโครคอนโทรลเลอร์ โดยข้อมูลที่ส่งออกคือข้อมูลการใช้กำลังงานไฟฟ้าทั้งหมด และทางภาครับจะต่อผ่านขา RxD ออกทางขา 26 บิต RC7 ของไมโครคอนโทรลเลอร์

3.4 ฮาร์ดแวร์ภาครับ

1. ไมโครคอนโทรลเลอร์ MCS-51 ได้ถูกนำมาใช้ในการควบคุมอุปกรณ์ต่างๆ เพื่อให้สามารถทำงานร่วมกับอุปกรณ์เชื่อมต่อภายนอกได้อย่างมีประสิทธิภาพตามที่ต้องการ โดยใช้ไอซีเบอร์ AT89C52 ซึ่งจะทำงานร่วมกับจอแสดงผลแอลซีดี (แบบ 16 ตัวอักษร 4 แถว) ไอซีฐานเวลา หน่วยความจำภายนอก คีซีแพด MAX232 DB9 RF-Wireless โมดูล ไอซีสแกนคีย์สวิตช์ และคอมพิวเตอรืที่ใช้ในการแสดงผล การใช้งานจริงของไมโครคอนโทรลเลอร์เบอร์นี้ต่อคริสตอลค่า 11.059 เมกะเฮิร์ตซ์ คร่อมที่ขา 18 และ 19 และต่อคาปาซิเตอร์ค่า 30 พิโคฟารัด เพื่อสร้างสัญญาณนาฬิกาป้อนให้ไมโครคอนโทรลเลอร์ ส่วนขา 31 และขา 9 จะต่อคาปาซิเตอร์ค่า 1 ไมโครฟารัด ที่ขา 9 ต่อความต้านทาน 10 กิโลโอห์มต่อลงกราวด์ ขาแรงดันไฟเลี้ยงวงจร 5 โวลต์ต่อที่ขา 40 และขากราวด์ต่อที่ขา 20 เป็นการต่อวงจรใช้งานไมโครคอนโทรลเลอร์ MCS-51 แบบพื้นฐาน

2. ภาคการทำงานร่วมกันระหว่างคีซีแพดและแสดงผลทางจอแอลซีดีโมดูล สำหรับภาครับนี้จะใช้แอลซีดีโมดูลขนาด 16 ตัวอักษร 4 แถว และคีซีแพดขนาด 4x4 เพื่อใช้แสดงรูปแบบฟังก์ชันและโหมดการทำงานต่างๆ ของระบบ เมื่อเริ่มต้นการทำงานที่หน้าจอลแอลซีดีโมดูลจะปรากฏข้อความหน้าหลักหรือข้อความต้อนรับปรากฏขึ้นมา จากนั้นจะเลือกกดปุ่มต่างๆ บนคีซีแพดตามที่ตั้งไว้ก็จะเข้าไปยังโหมดการทำงานย่อยที่มีอยู่ภายใน ซึ่งจะแสดงข้อความดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงลักษณะโหมดการทำงานหน้าหลักของฟังก์ชันการทำงานต่างๆ

บรรทัดแรกแสดงคำว่า 1. Set Time & Date หมายความว่าหากต้องการจะตั้งค่าวันที่และเวลาใหม่ ให้กดคีย์แพดเลขที่ 1 จากนั้นการทำงานจะเข้าสู่โหมดของการตั้งวันที่และเวลา

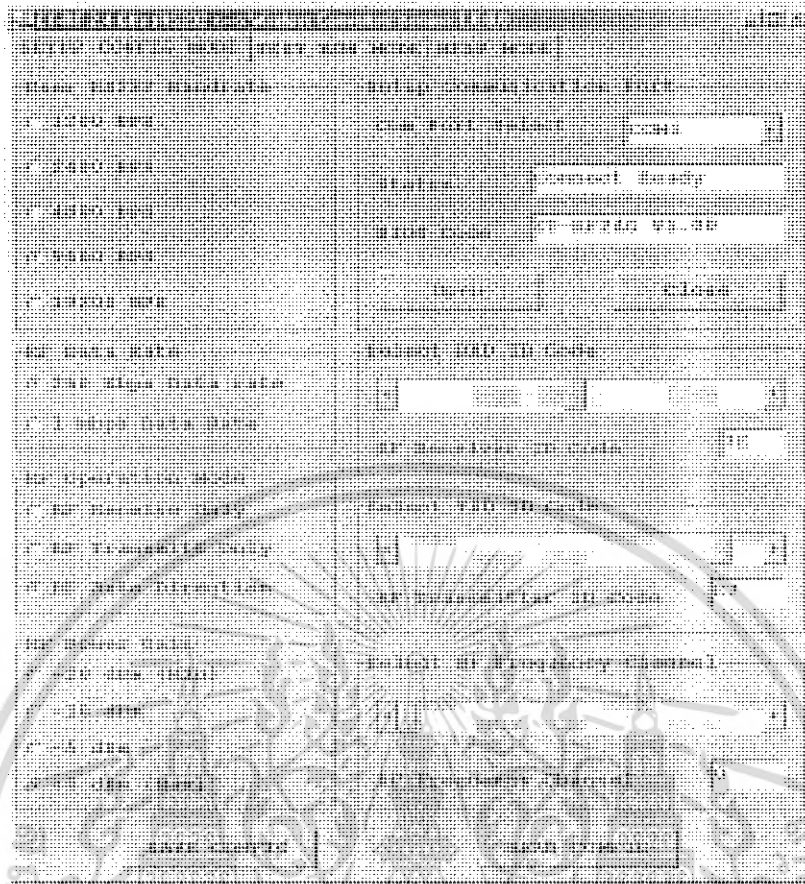
บรรทัดที่สองแสดงคำว่า 2. Collect & Read หมายความว่าเลือกโหมดการอ่านข้อมูลหรือเก็บข้อมูลจากวัดคัมมิเตอร์ไร้สายเก็บข้อมูลการใช้กำลังงานไฟฟ้า หากเราต้องการอ่านและเก็บข้อมูลนั้นให้กดคีย์แพดที่เลข 2 จากนั้นการทำงานจะเข้าสู่โหมดการอ่านและเก็บข้อมูล

บรรทัดที่สามแสดงคำว่า 3. Send to PC หมายความว่าต้องการนำข้อมูลที่ได้รับมาส่งไปแสดงผลที่คอมพิวเตอร์หรือไม่ ถ้าต้องการก็กดคีย์แพดเลขที่ 3 จากนั้นการทำงานจะเข้าสู่โหมดของการส่งข้อมูลไปแสดงผลที่โปรแกรมไฮเปอร์เทอร์มินอลทางจอคอมพิวเตอร์

3.5 การทำงานของภาค RF-Wireless โมดูล

3.5.1 การทำงานใน เซตอัปโหมด

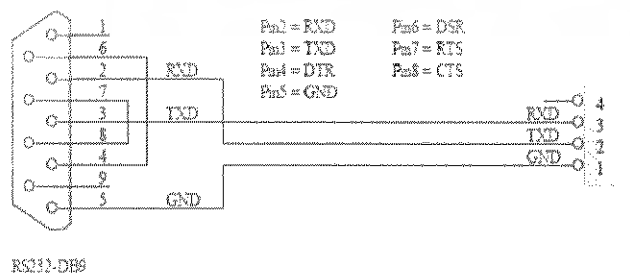
การใช้งาน RF-Wireless โมดูล ในเซตอัปโหมด ซึ่งเป็นโหมดสำหรับที่ใช้ในการกำหนดค่าต่างๆ สำหรับควบคุมการทำงานของ RF-Wireless โมดูล จะใช้ในขณะที่เครื่องทำงานอยู่ในรันโหมด โดยคุณสมบัติของค่าต่างๆ มีดังนี้



รูปที่ 3.9 แสดงค่าต่างๆ ที่กำหนดของ RF-Wireless โมดูล

3.5.2 การทำงานแบบ RF Auto Direction ใน รั้นโหมด

เป็นการทำงานชนิด 2 ทิศทาง เรียกว่าแบบผลัดกันรับผลัดกันส่ง ซึ่งสามารถใช้รับ-ส่งข้อมูลระหว่างต้นทางและปลายทางได้ โดยใช้ RF-Wireless โมดูล ด้านละ 1 ชุด เท่านั้น เพียงแต่การรับ-ส่งข้อมูลแบบนี้จะไม่สามารถส่งข้อมูลสวนทางกันได้เหมือนกับแบบฟูลดูเพล็กซ์จะต้องใช้วิธีการผลัดกันรับข้อมูลและส่งข้อมูลแทน โดยเมื่อฝ่ายรับทำการรับข้อมูลได้จนครบแล้วจึงจะสลับหน้าที่เป็นฝ่ายส่งเพื่อส่งข้อมูลย้อนกลับไป

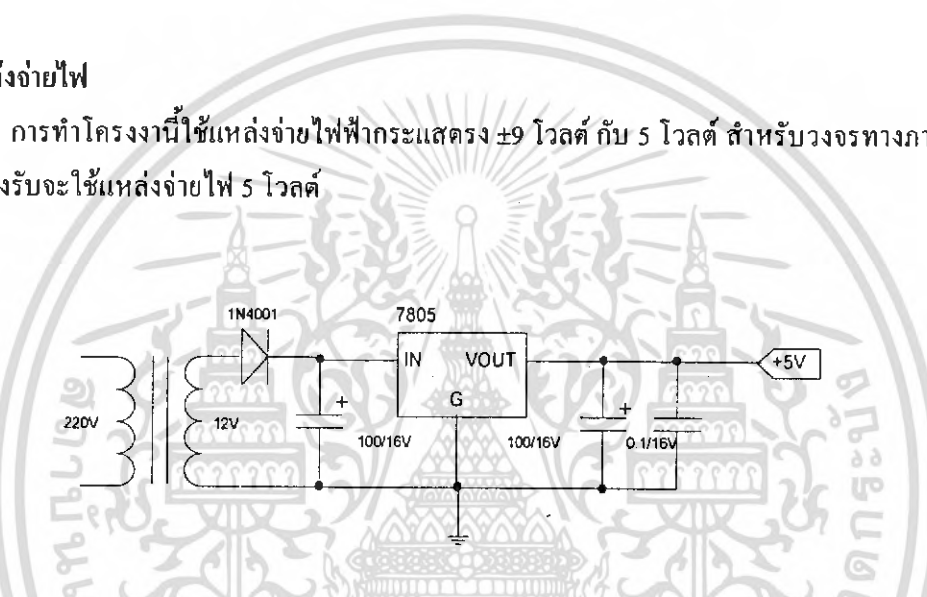


รูปที่ 3.10 แสดงการต่อสาย RS-232 เพื่อใช้งานกับ RF-Wireless โมดูล ในโหมด Auto Direction เอกสารนี้เป็นเอกสารที่ สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

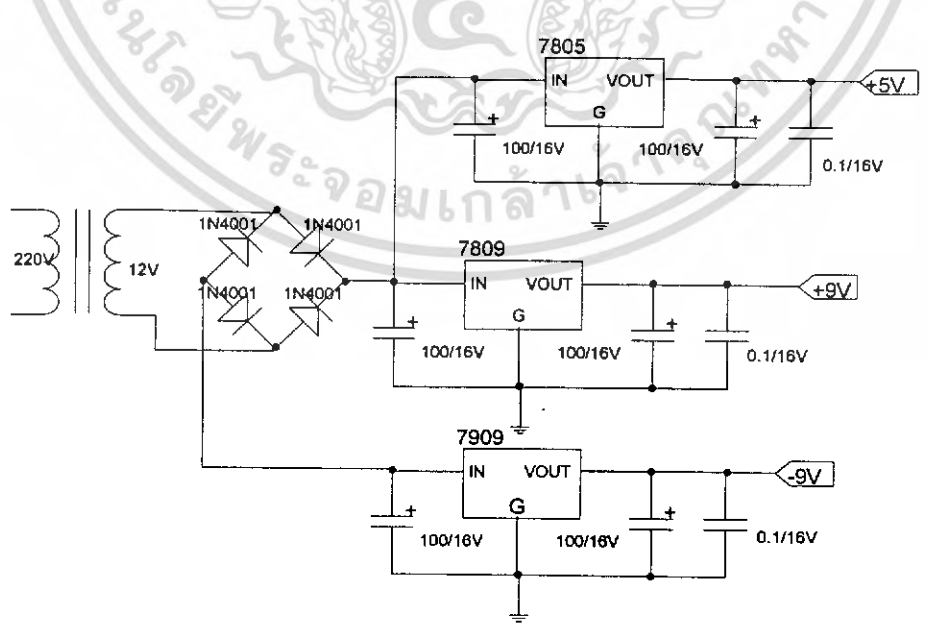
สำหรับความสามารถในการรับข้อมูลจาก RS-232 ของ RF-Wireless โมดูล ในโหมดการทำงานนี้จะสามารถรับข้อมูลได้อย่างต่อเนื่องสูงสุดไม่เกิน 64 ไบต์ ดังนั้นในกรณีที่มีการส่งข้อมูลจากด้าน RS-232 ด้วยข้อมูลจำนวนมากกว่า 64 ไบต์ ต่อเนื่องกันนั้นควรทำการแบ่งข้อมูลออกเป็นชุดๆ โดยให้มีขนาดชุดละไม่เกิน 64 ไบต์ หลังจากทำการส่งข้อมูลอย่างต่อเนื่องไปได้ 1 ชุด (64 ไบต์) ควรทำการหน่วงเวลาไว้ชั่วขณะหนึ่งอย่างน้อย 1 มิลลิวินาที แล้วจึงเริ่มส่งข้อมูลชุดถัดไปสลับกับการหน่วงเวลาอย่างนี้เรื่อยๆ เพื่อให้ RF-Wireless โมดูล สามารถนำข้อมูลที่รับได้จากด้าน RS-232 ส่งออกไปทางด้าน RF ได้ทันที ซึ่งถ้าทำการส่งข้อมูลอย่างต่อเนื่องโดยไม่มีกรหน่วงเวลาเลยอาจทำให้ข้อมูลบางไบต์จะเกิดการสูญหายไป

3.6 แหล่งจ่ายไฟ

การทำโครงงานนี้ใช้แหล่งจ่ายไฟฟ้ากระแสตรง ± 9 โวลต์ กับ 5 โวลต์ สำหรับวงจรทางภาคส่งและวงจรทางรับจะใช้แหล่งจ่ายไฟ 5 โวลต์

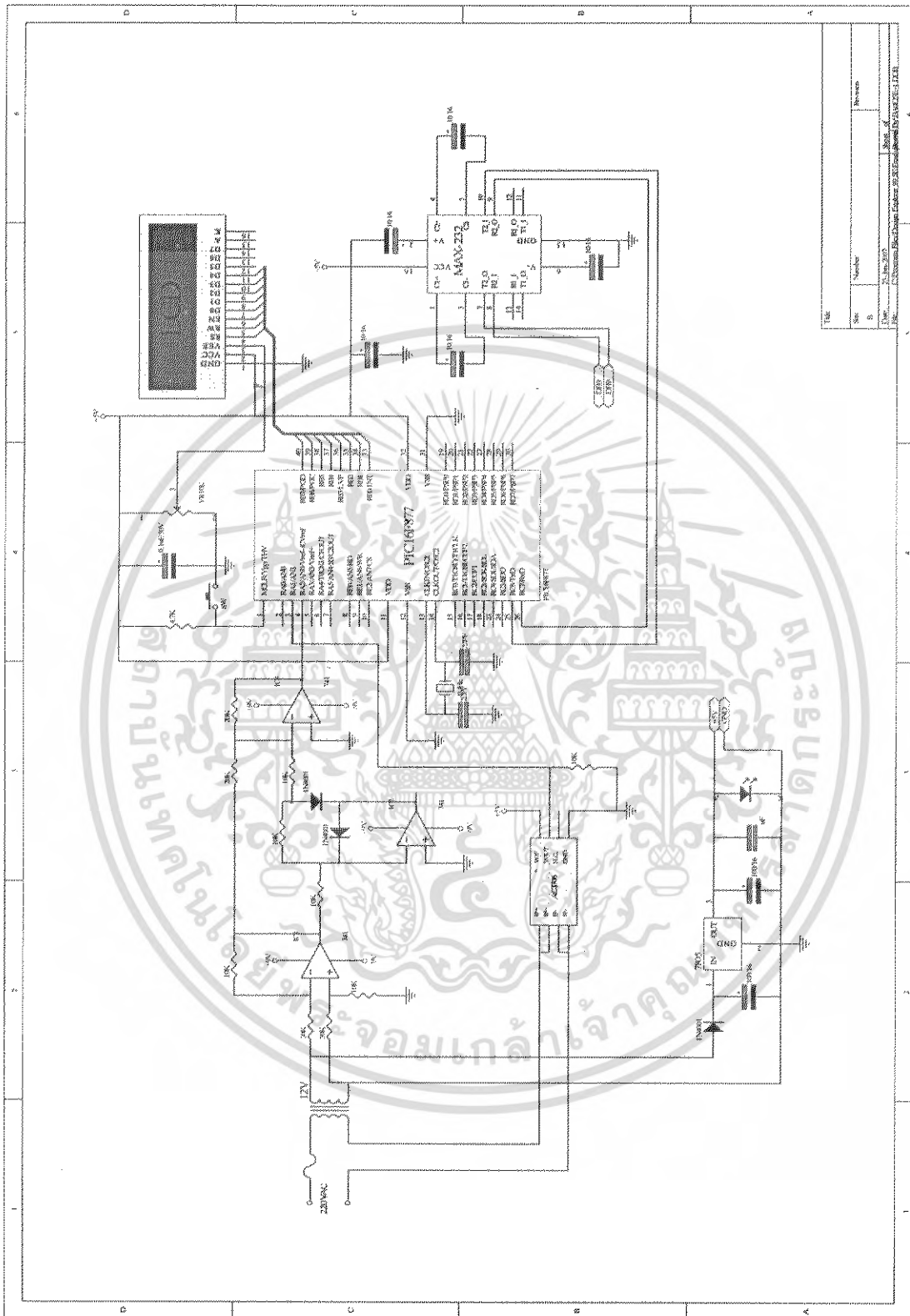


รูปที่ 3.11 แสดงวงจรแหล่งจ่ายไฟของภาครับ



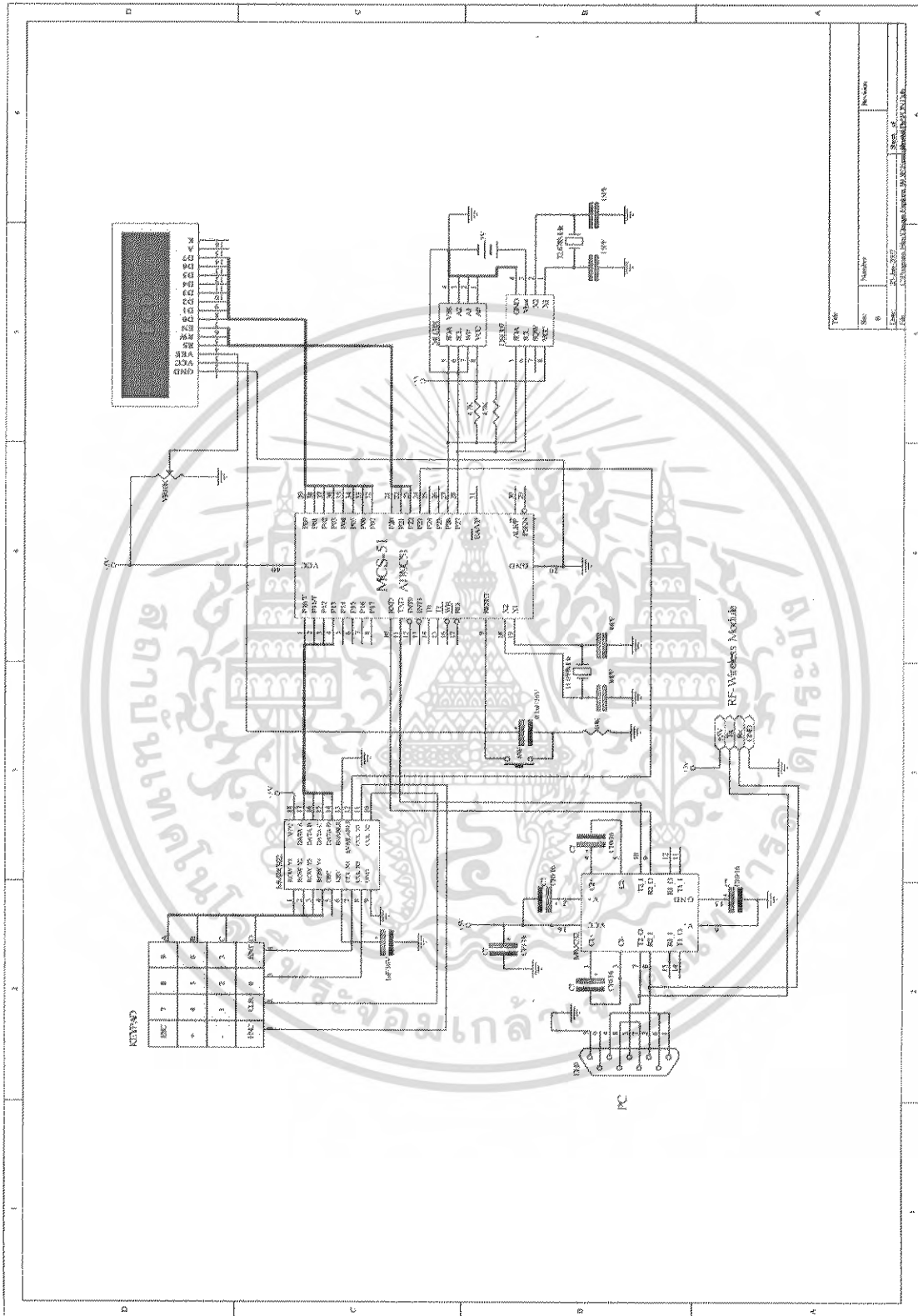
รูปที่ 3.12 แสดงวงจรแหล่งจ่ายไฟของภาคส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ในเชิงพาณิชย์เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แสดงวงจรภาคส่งวัดคัมมิเตอร์ไร้สายเก็บข้อมูลการใช้กำลังงานไฟฟ้า

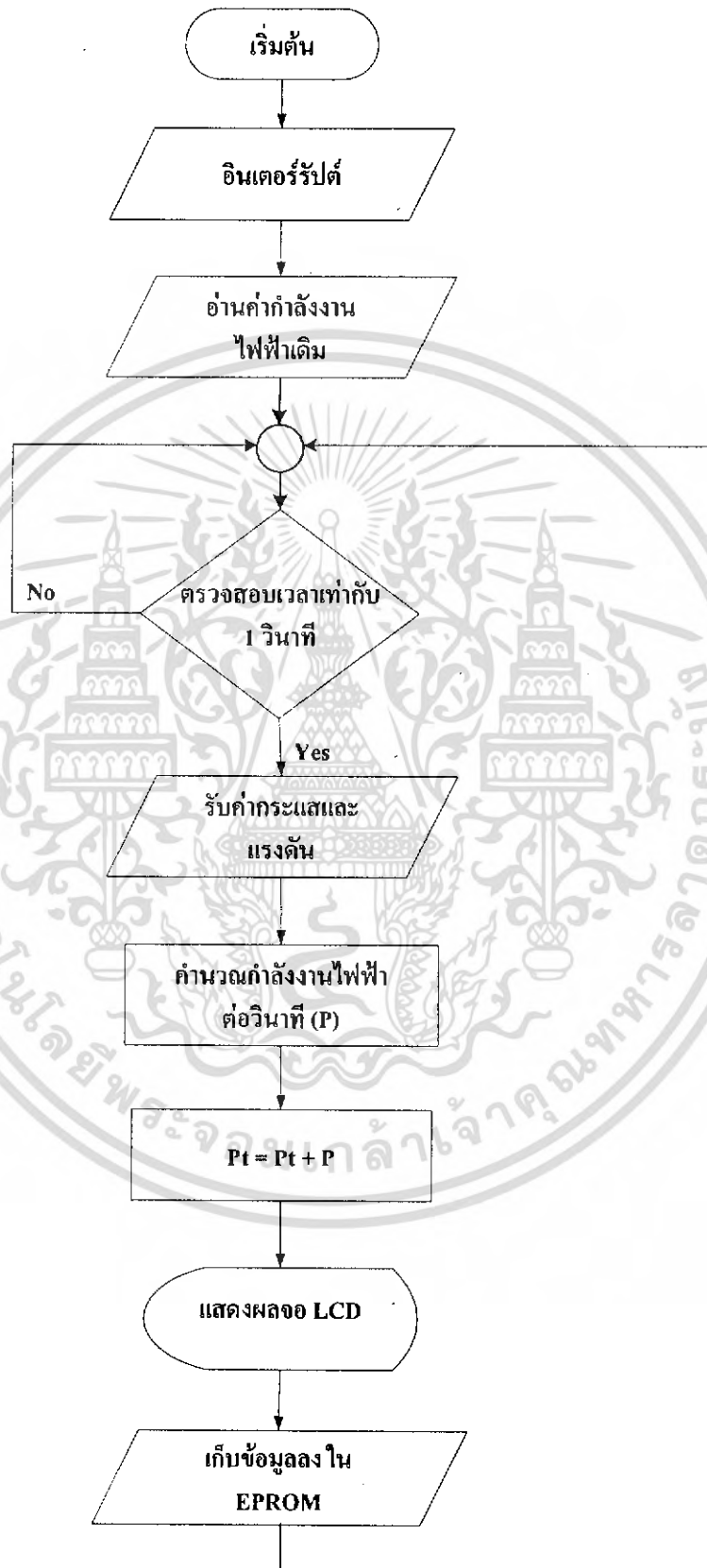
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 แสดงวงจรการรับวัดตัวไมโครไรสายเก็บข้อมูลการใช้กำลังงานไฟฟ้า เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 ส่วนของซอฟต์แวร์

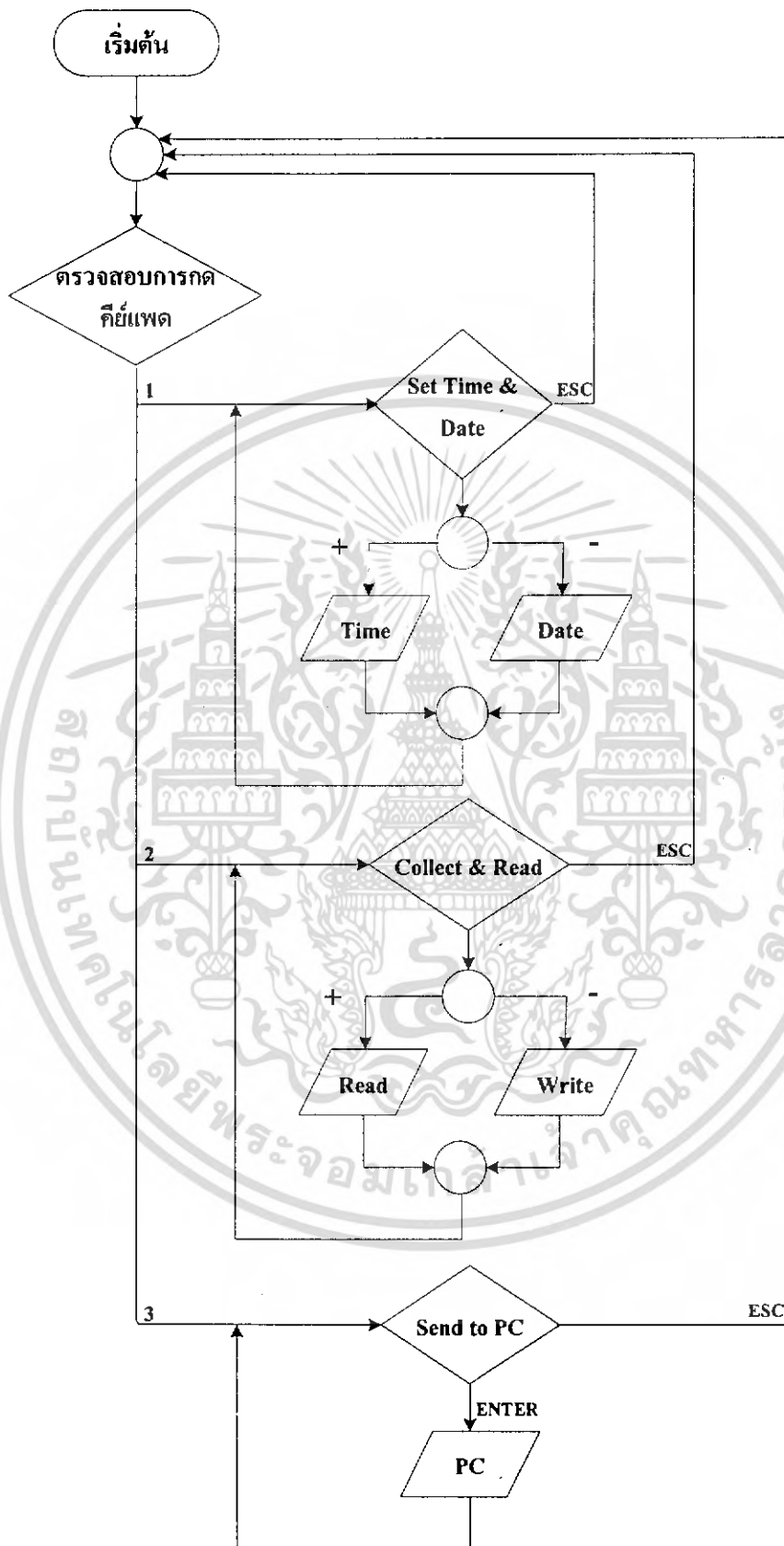
3.7.1 Flow Chat การทำงานของภาคส่ง



รูปที่ 3.15 แสดง Flow Chat การทำงานของภาคส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.2 Flow Chat การทำงานของภากรับ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

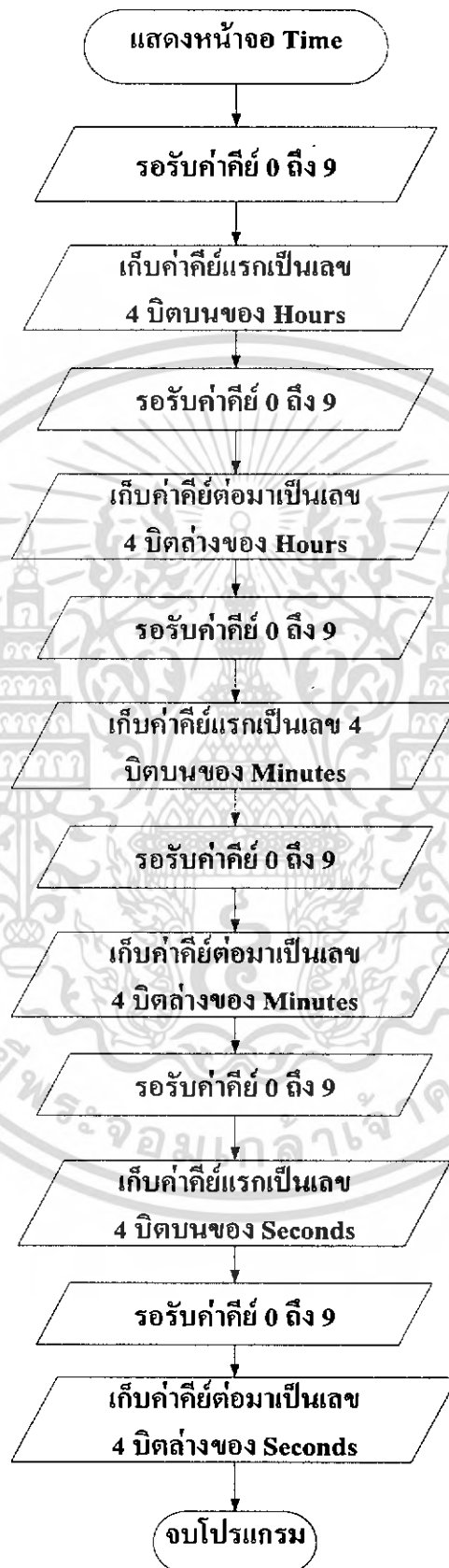
3.7.3 Flow Chat การทำงานของการตั้งวันที่



รูปที่ 3.17 แสดง Flow Chat การทำงานของการตั้งวันที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

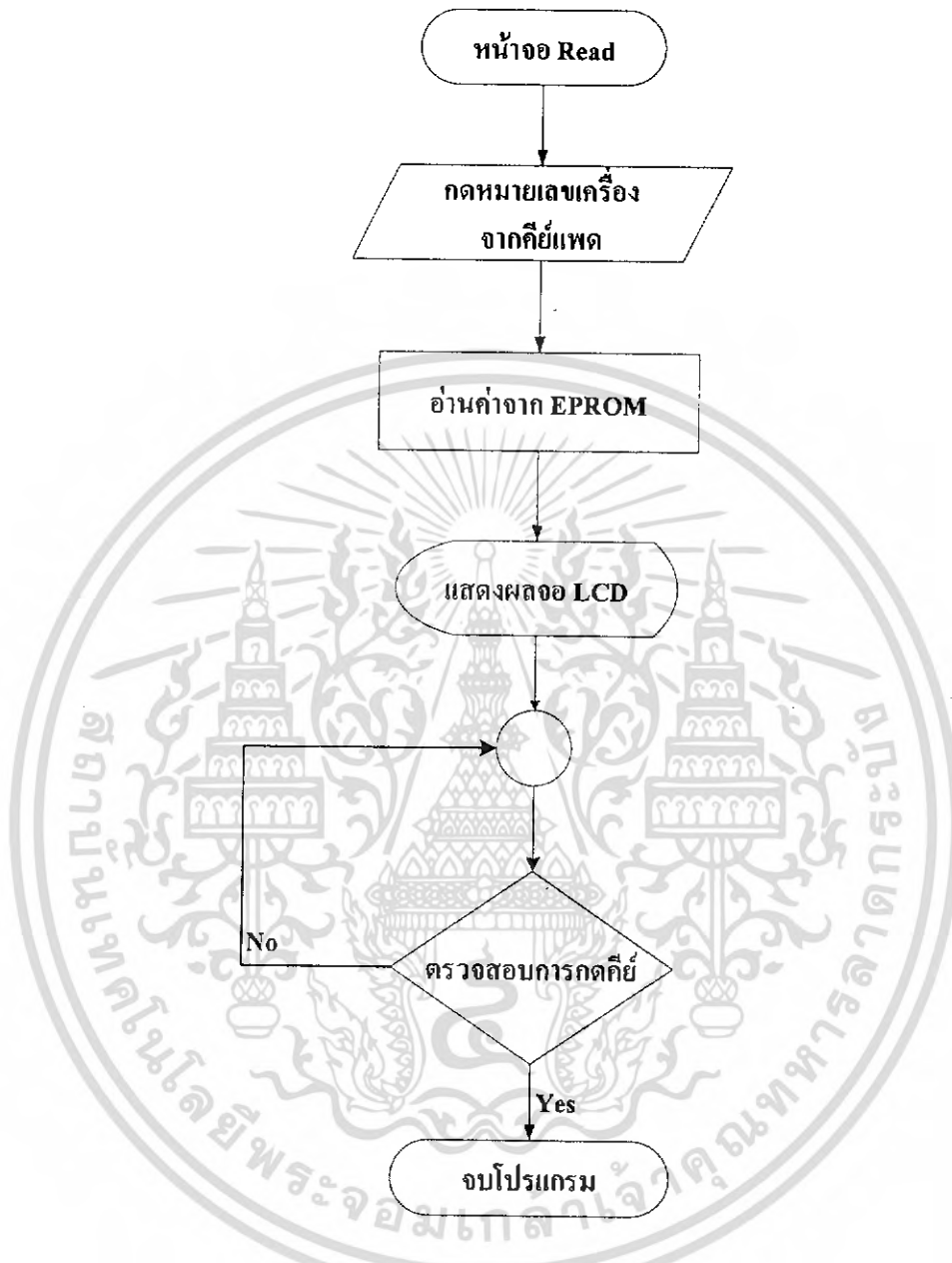
3.7.4 Flow Chat การทำงานของการตั้งเวลา



รูปที่ 3.18 แสดง Flow Chat การทำงานของการตั้งเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

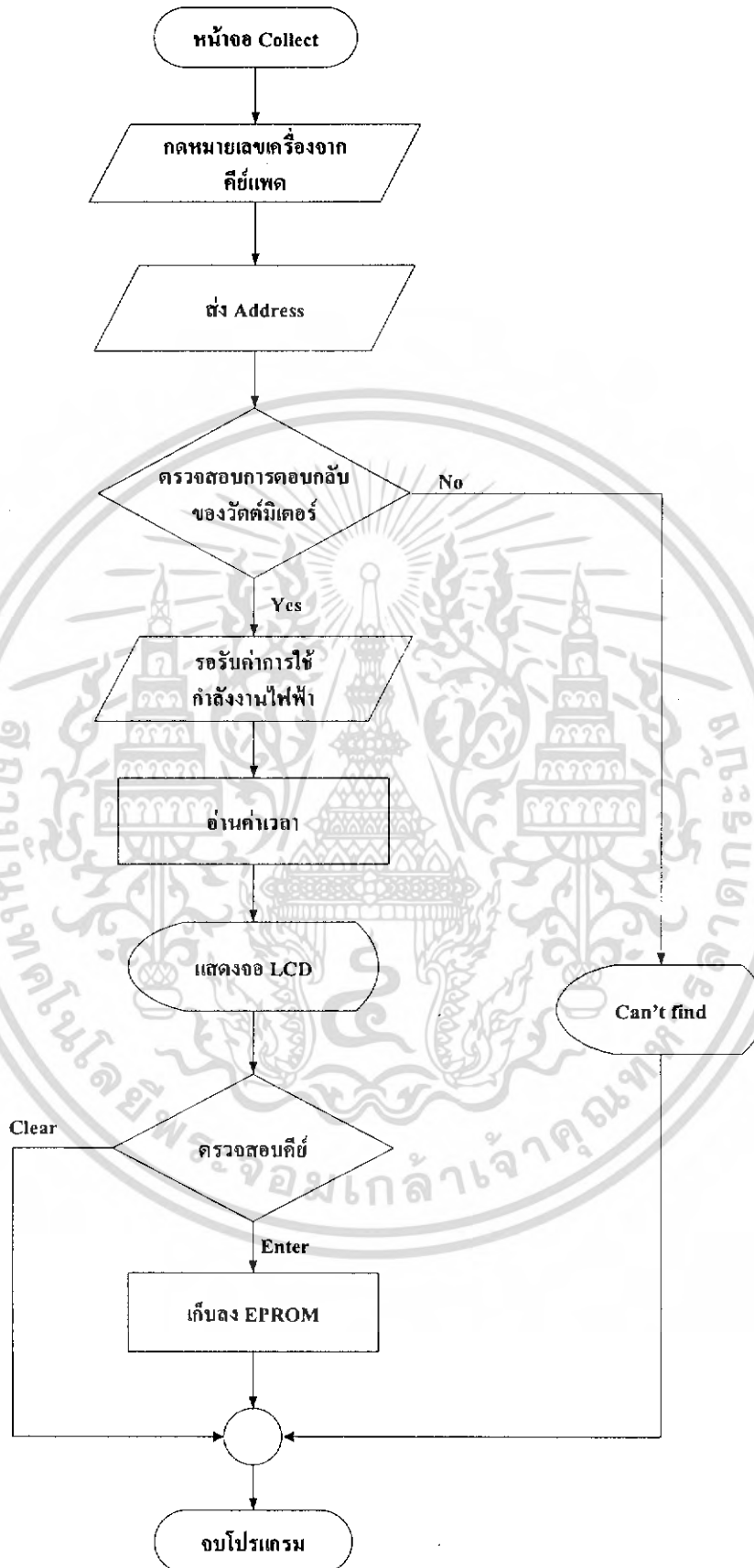
3.7.5 Flow Chat การทำงานของการอ่านข้อมูล



รูปที่ 3.19 แสดง Flow Chat การทำงานของการอ่านข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

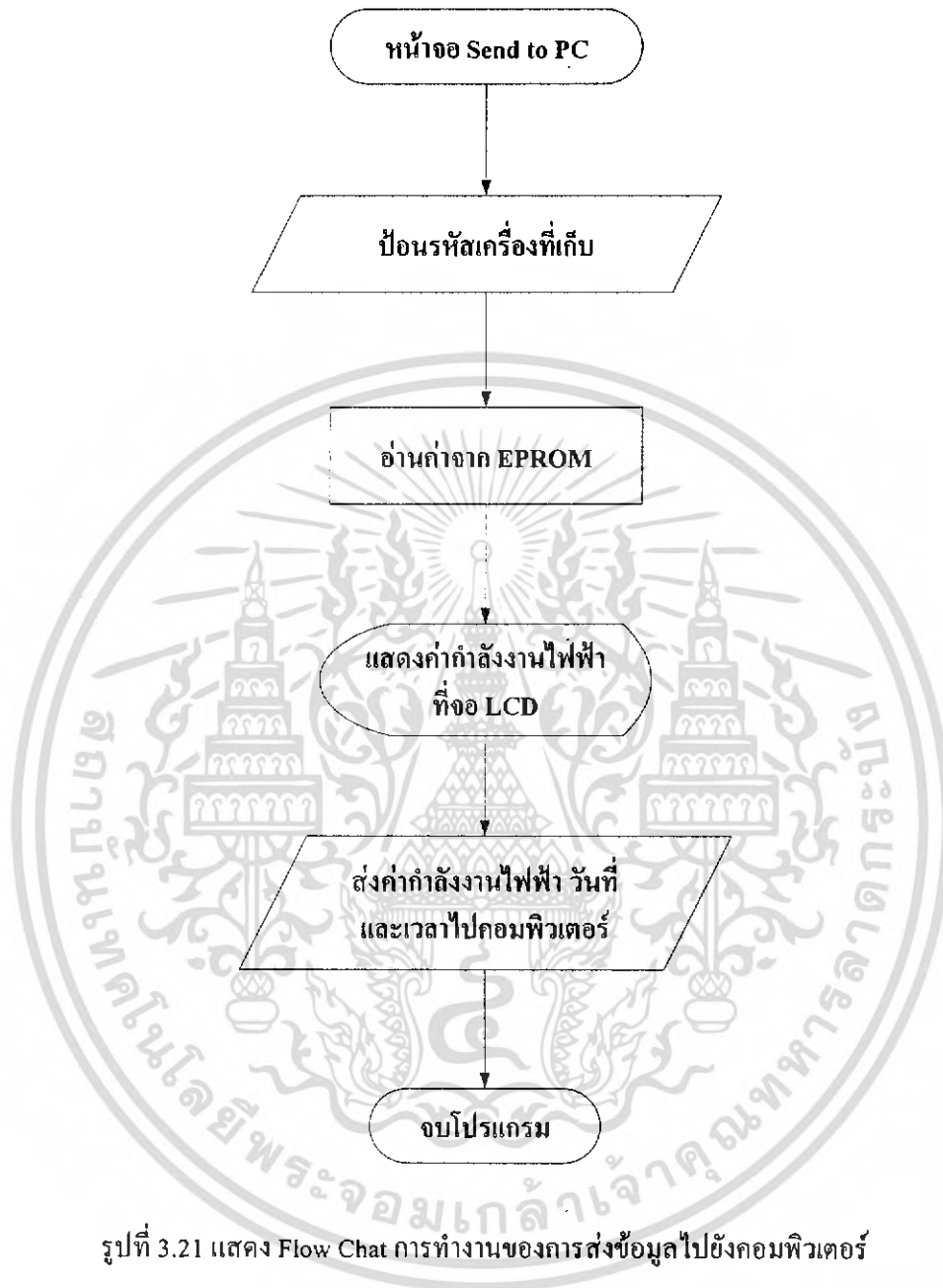
3.7.6 Flow Chat การทำงานของการเก็บข้อมูล



รูปที่ 3.20 แสดง Flow Chat การทำงานของการเก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.7 Flow Chat การทำงานของการส่งข้อมูลไปยังคอมพิวเตอร์



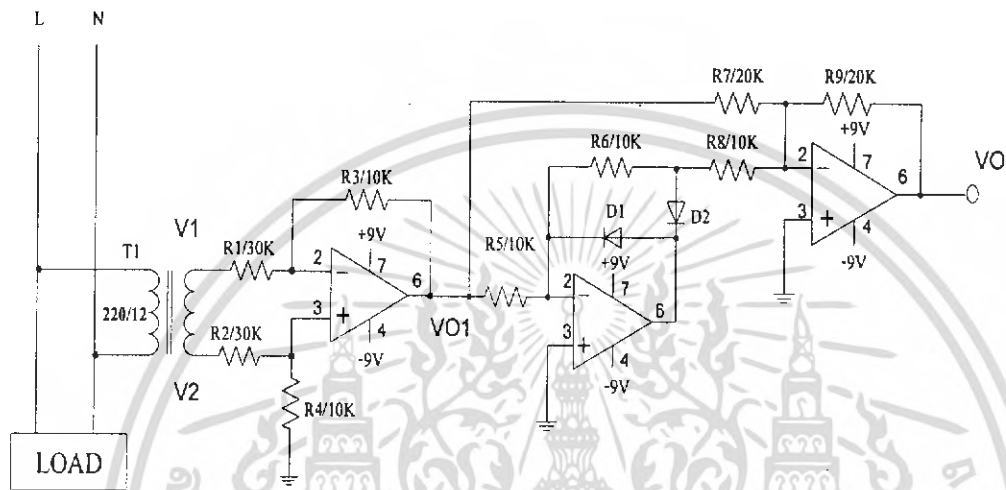
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 ผลการทดลองฮาร์ดแวร์

4.1.1 ผลการทดลองวงจรตรวจจับแรงดัน



รูปที่ 4.1 แสดงวงจรตรวจจับแรงดันแบบสมบูรณ์

จากรูปที่ 4.1 จะแสดงวงจรรวมของการตรวจจับแรงดัน โดยจะนำอ้างอิงจุดต่างๆที่ทำการวัดสัญญาณออกมาเปรียบเทียบกับเป็นไปตามการคำนวณและการสร้างในบทที่ 3 หรือไม่ ซึ่งจะวัดสัญญาณดังจุดต่อไปนี้

1. สัญญาณที่ได้หลังหม้อแปลง 220/12 นั่นก็คือผลต่างระหว่าง V1 กับ V2 และเป็นแรงดันอินพุตของวงจรขยายผลต่าง
2. สัญญาณที่จุด VO1 จะเป็นเอาต์พุตของวงจรขยายผลต่างที่ออกแบบให้ลดแรงดันอินพุตของวงจรเหลือ 1 ใน 3 เท่า
3. สัญญาณที่จุด VO เป็นเอาต์พุตของวงจรตรวจจับแรงดันที่เป็นไฟฟ้ากระแสตรง และจะนำไปป้อนให้กับไมโครคอนโทรลเลอร์ PIC16F877 ใช้แปลงสัญญาณอนาล็อกเป็นดิจิทัล

จากจุดต่างๆ ของสัญญาณที่วัดผลการทดลองนั้นสามารถบอกให้ทราบว่า การออกแบบถูกต้อง เพราะผลลัพธ์ที่ได้เป็นไปตามที่ออกแบบและสามารถแสดงผลการทดลองที่ได้จากการวัดด้วยมัลติมิเตอร์

ตารางที่ 4.1 แสดงผลการทดลองวงจรตรวจจับแรงดัน

แรงดันอินพุต (VAC)	แรงดันผลต่างของ V1,V2 (VAC)	VO1 (VAC)	VO (VDC)
190	10.75	3.53	3.54
193	10.93	3.63	3.61
196	11.09	3.69	3.70
199	11.27	3.74	3.74
202	11.44	3.80	3.79
205	11.60	3.85	3.85
208	11.78	3.91	3.91
211	11.94	3.96	3.95
214	120.8	4.01	4.02
217	12.28	4.08	4.08
220	12.45	4.13	4.12
223	12.58	4.18	4.17
226	12.78	4.23	4.23
229	12.95	4.30	4.29
232	13.10	4.35	4.34
235	13.28	4.40	4.40
238	13.43	4.46	4.46
241	13.60	4.52	4.51

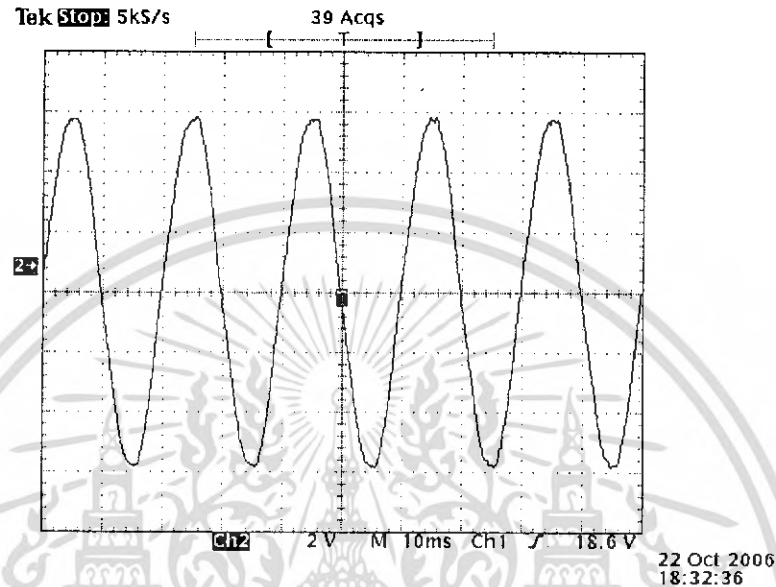
4.1.2 ผลการทดลองรูปสัญญาณวงจรตรวจจับแรงดัน

จากผลการทดลองรูปสัญญาณของวงจรตรวจจับแรงดันจะแสดงให้เห็นถึงความเปลี่ยนแปลงของสัญญาณที่ป้อนให้แก่วงจรขยายผลต่าง (VO1) และแรงดันเอาต์พุตที่ได้จากวงจรเรียงกระแสเต็มรูป (VO)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

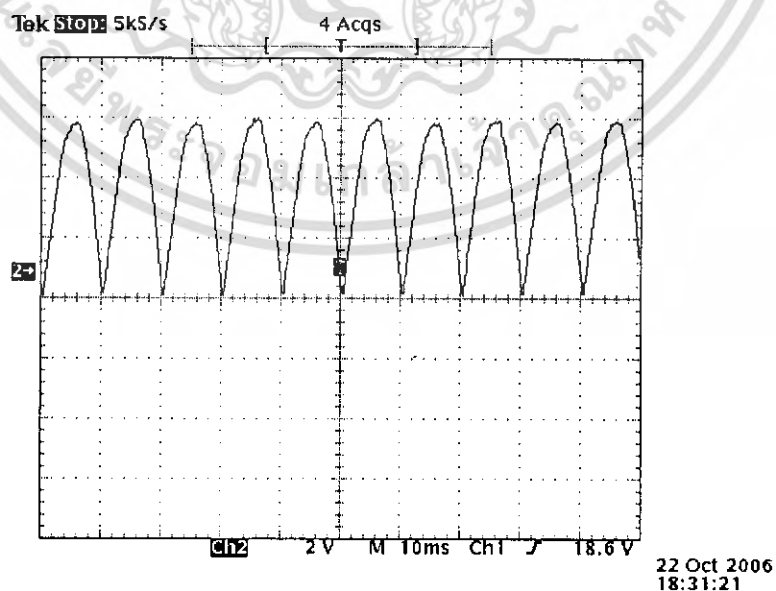
แรงดันเอาต์พุตจะเปรียบเสมือนเป็นการยกระดับจากสัญญาณรูปคลื่นไซน์ที่เป็นไฟฟ้ากระแสสลับให้เป็นไฟฟ้ากระแสตรงที่มีเฉพาะสัญญาณช่วงบวกแบบฟูลเวฟเรกติไฟต์

1. สัญญาณที่ได้จากเอาต์พุตของวงจรขยายผลต่างระหว่าง V_1 กับ V_2 ที่เป็นสัญญาณ ไซน์ ความถี่ 50 เฮิร์ต เมื่อป้อนแรงดันอินพุต 220 โวลต์ ใช้มิเตอร์วัดแรงดันได้ 12.45 โวลต์ แสดงดังรูปที่ 4.2



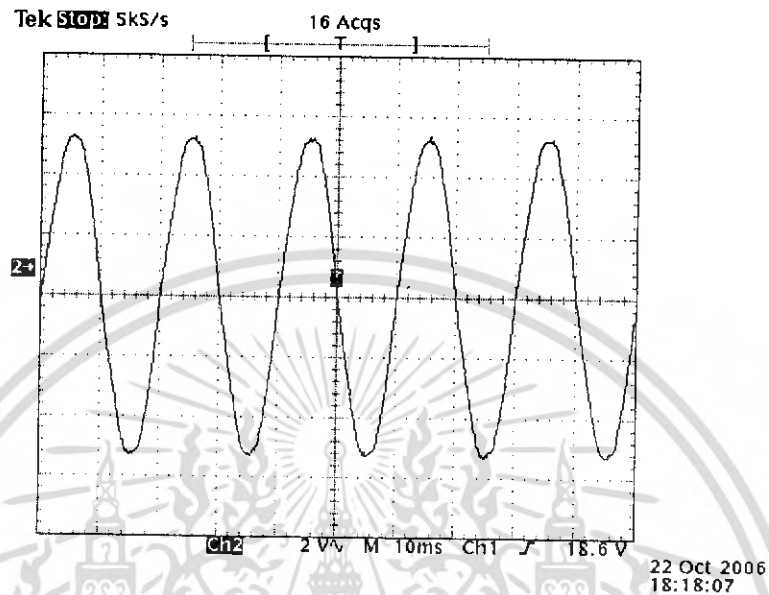
รูปที่ 4.2 แสดงสัญญาณแรงดันระหว่าง V_1 กับ V_2 เมื่อป้อนแรงดันอินพุต 220 โวลต์

2. สัญญาณแรงดันเอาต์พุตจากวงจรเรียงกระแสเต็มรูปจะมีค่าลดลงมาเป็น 3 เท่าจากแรงดันที่ได้หลังหม้อแปลงและมีค่าเฉพาะช่วงบวก (สัญญาณไฟฟ้ากระแสตรง) แรงดันที่วัดได้จากมิเตอร์ 4.12 โวลต์ แสดงดังรูปที่ 4.3



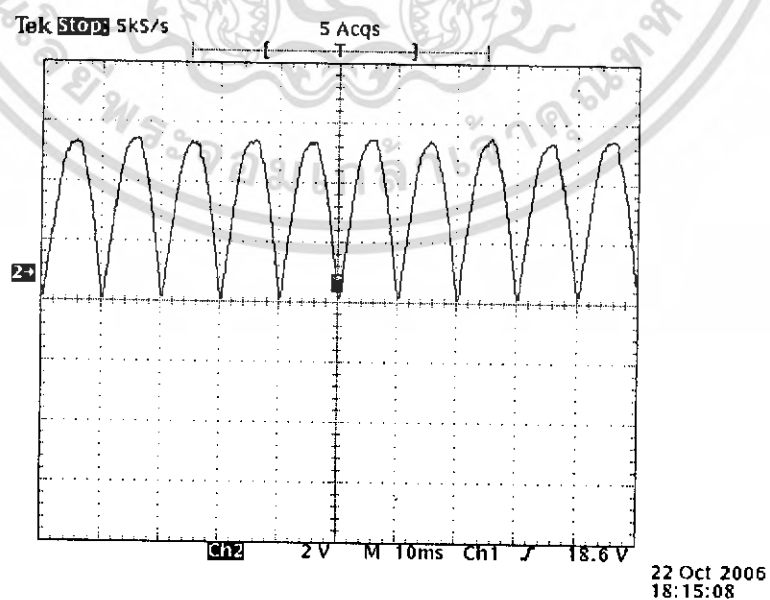
รูปที่ 4.3 แสดงสัญญาณเอาต์พุตวงจรเรียงกระแสเต็มรูป เมื่อป้อนแรงดันอินพุต 220 โวลต์ เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้กับโรงเรียนเพื่อใช้เพื่อการศึกษาเท่านั้น เมื่อผู้ใดเห็นเว็บไซต์นี้เป็นการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สัญญาณที่ได้จากเอาต์พุตของวงจรขยายผลต่างระหว่าง V1 กับ V2 เมื่อป้อนแรงดันอินพุต 200 โวลต์ ใช้มิเตอร์วัดแรงดันได้ 11.32 โวลต์ แสดงดังรูปที่ 4.4



รูปที่ 4.4 แสดงสัญญาณแรงดันระหว่าง V1 กับ V2 เมื่อป้อนแรงดันอินพุต 200 โวลต์

4. สัญญาณแรงดันเอาต์พุตจากวงจรเรียงกระแสเต็มรูปจะมีค่าเฉพาะช่วงบวก สามารถวัดแรงดันได้ 3.45 โวลต์ แสดงดังรูปที่ 4.5

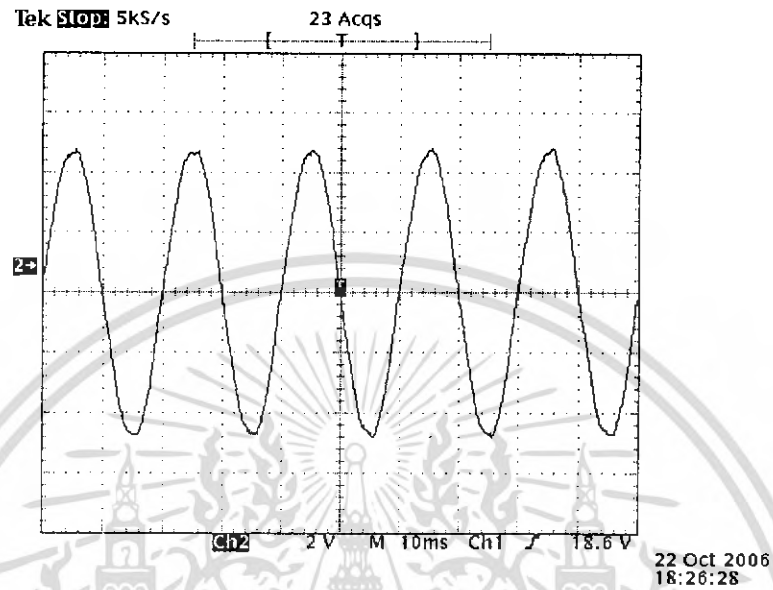


รูปที่ 4.5 แสดงสัญญาณเอาต์พุตวงจรเรียงกระแสเต็มรูป เมื่อป้อนแรงดันอินพุต 200 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการเรียนการสอนเท่านั้น การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย ผู้ที่ฝ่าฝืนจะถูกลงโทษตามกฎหมาย

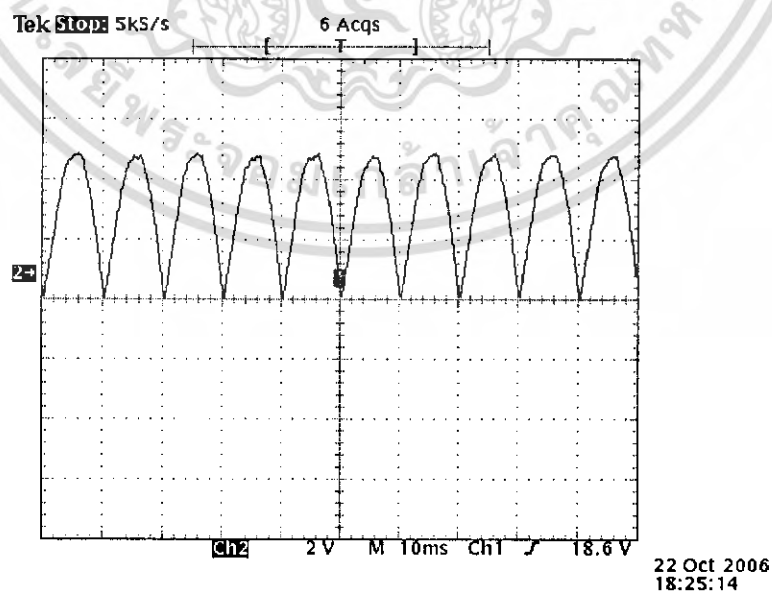
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. สัญญาณเอาต์พุตของวงจรขยายผลต่างระหว่าง V1 กับ V2 เมื่อป้อนแรงดันอินพุต 180 โวลต์
ใช้มิเตอร์วัดแรงดันได้ 10.20 โวลต์ แสดงดังรูปที่ 4.6



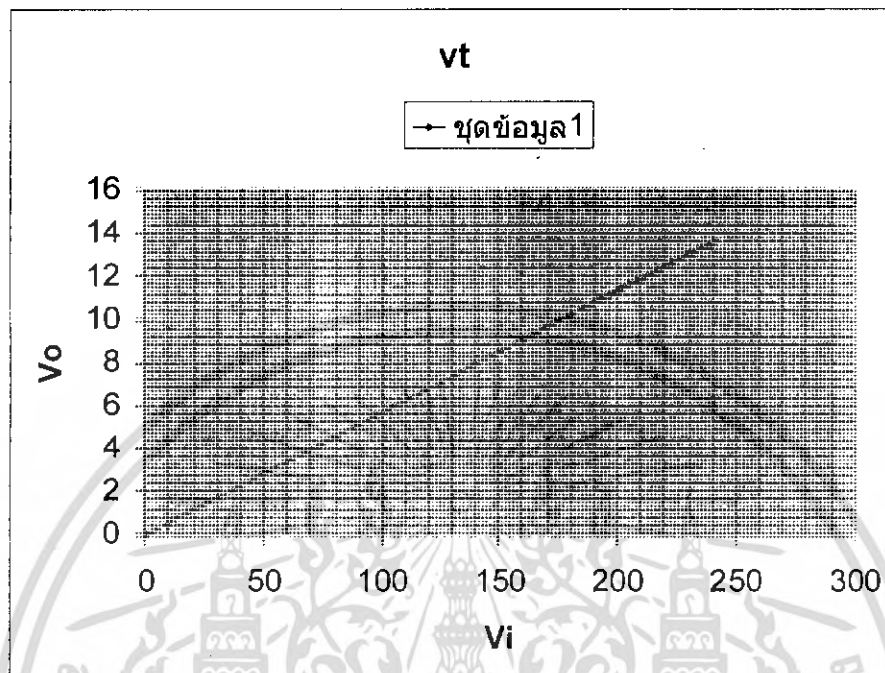
รูปที่ 4.6 สัญญาณแรงดันระหว่าง V1 กับ V2 เมื่อป้อนแรงดันอินพุต 180 โวลต์

6. สัญญาณแรงดันเอาต์พุตจากวงจรเรียงกระแสเต็มรูปจะมีค่าเฉพาะช่วงบวก สามารถวัดแรงดัน
ได้ 3.35 โวลต์ แสดงดังรูปที่ 4.7



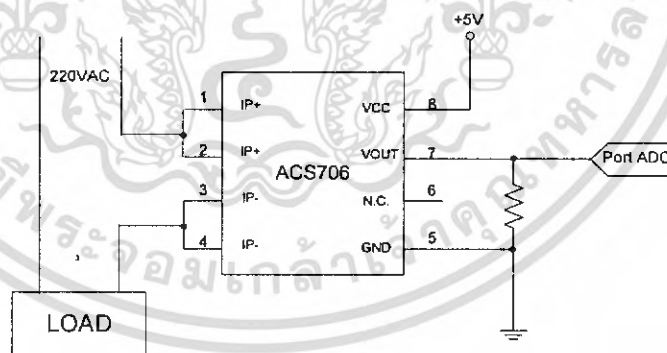
รูปที่ 4.7 แสดงสัญญาณเอาต์พุตวงจรเรียงกระแสเต็มรูป เมื่อป้อนแรงดันอินพุต 180 โวลต์
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น เมื่อผู้ผู้เห็นใบโฆษณาหรือการนำ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.8 แสดงลักษณะกราฟที่เป็นเชิงเส้นของเอาต์พุตจากวงจรตรวจจับแรงดันต่อแรงดันอินพุตที่ป้อนเข้ามา ซึ่งจะเป็นสัดส่วนต่อกัน โดยตรง



รูปที่ 4.8 แสดงความเป็นเชิงเส้นของการตรวจจับแรงดัน

4.1.3 ผลการทดลองวงจรตรวจจับกระแส



รูปที่ 4.9 แสดงวงจรตรวจจับกระแสแบบสมบูรณ์

จากรูปที่ 4.9 แสดงวงจรใช้งานจริงของการตรวจจับกระแสที่รับแรงดันอินพุต 220 โวลต์ แล้วต่อโหลดให้มีค่าเปลี่ยนแปลงไปเรื่อยๆทำให้เห็นรูปสัญญาณเกิดการเปลี่ยนแปลงแบบเป็นเชิงเส้น ทั้งนี้ค่าที่ได้จากผลการทดลองจะขึ้นอยู่กับค่าโหลดที่นำมาต่อ สามารถแสดงดังตารางที่ 4.2

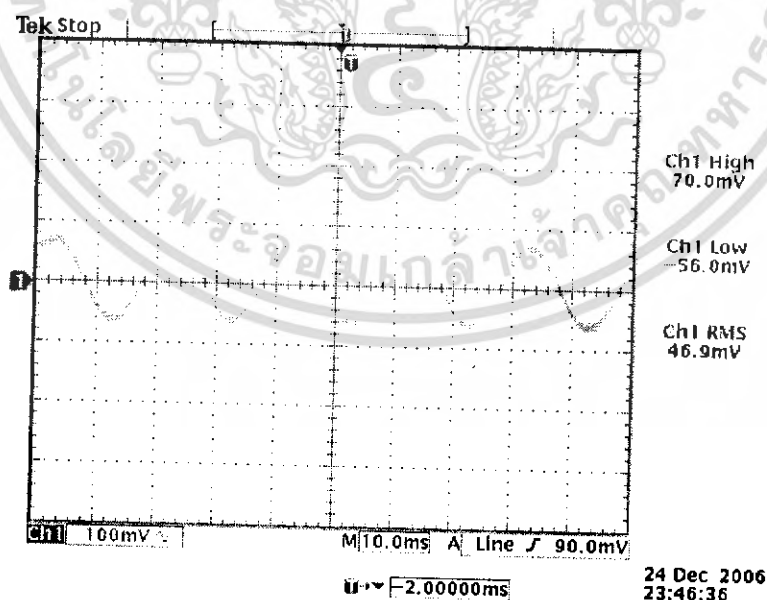
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 แสดงผลการทดลองวงจรตรวจจับกระแส

ชนิดของโหลด (วัตต์)	กระแสไหลออกจาก การคำนวณ (แอมแปร์)	กระแสไหล จากมิเตอร์ (แอมแปร์)	แรงดันเอาต์พุตวงจร ตรวจจับกระแสที่วัดได้ (มิลลิโวลต์)
พัดลม (กดปุ่ม 3)	0.245	0.247	24.9
หลอดไฟขนาด 60 วัตต์	0.273	0.278	28.6
หลอดไฟขนาด 100 วัตต์	0.455	0.455	46.9
หลอดไฟขนาด 60 วัตต์ และ 100 วัตต์	0.727	0.725	72.1
หลอดไฟขนาด 100 วัตต์ 2 หลอด	0.909	1.001	107
กาน้ำร้อน	2.886	2.900	295
กาน้ำร้อนและหลอดไฟขนาด 100 วัตต์	3.341	3.355	335.8

4.1.4 ผลการทดลองรูปสัญญาณวงจรตรวจจับกระแส

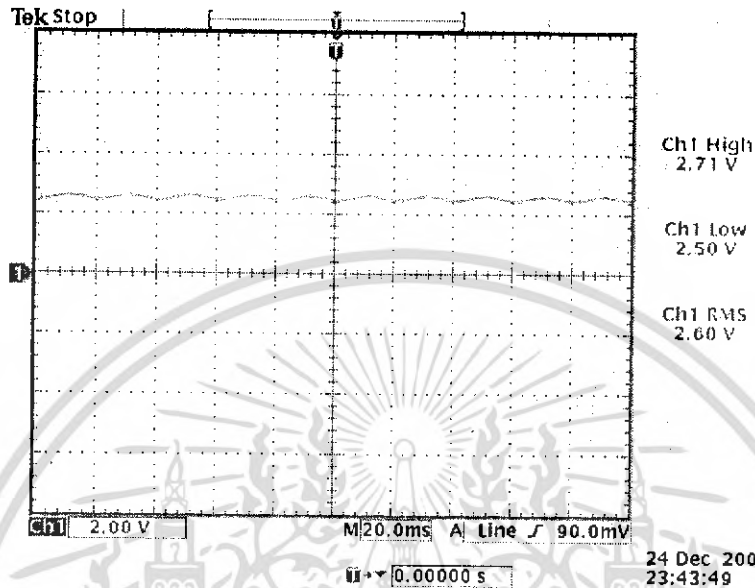
1. สัญญาณเอาต์พุตวงจรตรวจจับกระแส เมื่อต่อโหลดเป็นหลอดไฟขนาด 100 วัตต์ ใช้มิเตอร์วัดแรงดันเอาต์พุตไฟฟ้ากระแสสลับได้ 46.9 มิลลิโวลต์ แสดงดังรูปที่ 4.10



รูปที่ 4.10 แสดงสัญญาณแรงดันเอาต์พุตวงจรตรวจจับกระแส เมื่อต่อโหลด 100 วัตต์

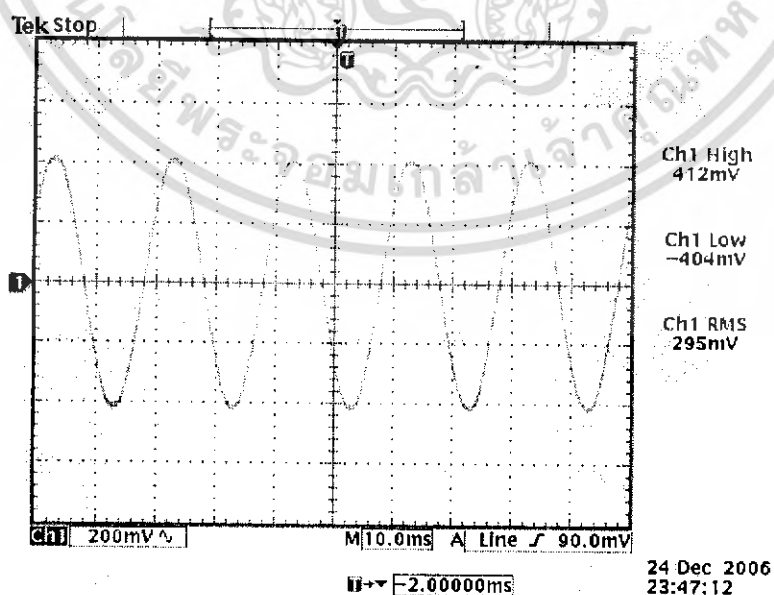
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. สัญญาณเอาต์พุตวงจรตรวจจับกระแส เมื่อต่อ โหลดเป็นหลอดไฟขนาด 100 วัตต์ จะได้แรงดันเอาต์พุตที่มีการกระเพื่อมของไฟฟ้ากระแสสลับบนไฟฟ้ากระแสตรงที่ระดับอ้างอิงค่า 2.5 โวลต์ แสดงดังรูปที่ 4.11



รูปที่ 4.11 แสดงสัญญาณเอาต์พุตไฟฟ้ากระแสตรงที่ระดับอ้างอิง 2.5 โวลต์ เมื่อต่อหลอดไฟเป็นโหลด

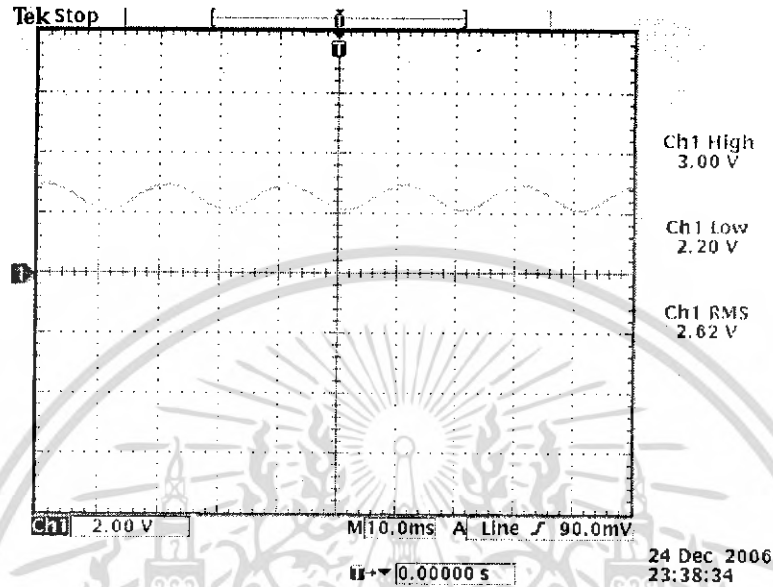
3. สัญญาณเอาต์พุตวงจรตรวจจับกระแส เมื่อต่อกาน้ำร้อนขนาด 635 วัตต์ เป็นโหลดให้กับวงจรใช้มิเตอร์วัดแรงดันเอาต์พุตไฟฟ้ากระแสสลับได้ 295 มิลลิโวลต์ แสดงดังรูปที่ 4.12



รูปที่ 4.12 แสดงสัญญาณแรงดันเอาต์พุตวงจรตรวจจับกระแส เมื่อต่อ โหลด 635 วัตต์

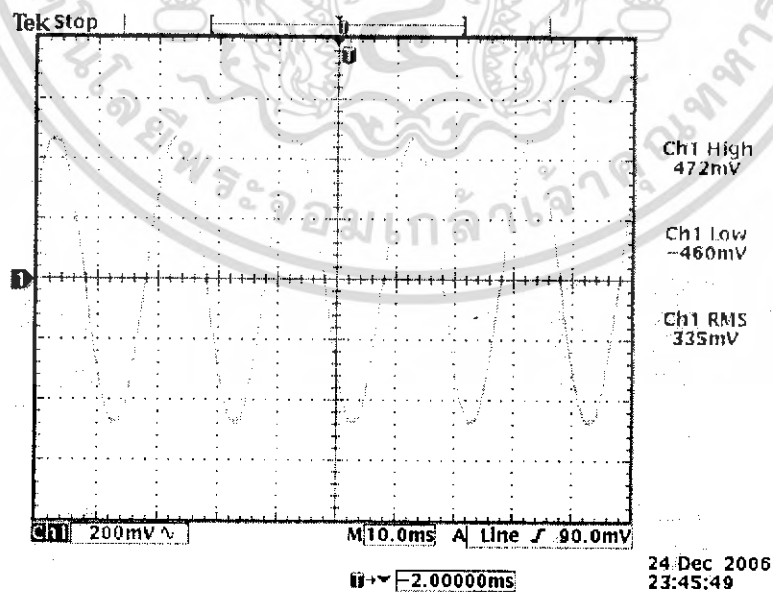
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น เมื่อผู้ใดเห็นแจ้งโดยมิชอบดำเนินการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. สัญญาณเอาต์พุตวงจรตรวจจับกระแส เมื่อต่อภาณ์น้ำร้อนขนาด 635 วัตต์ เป็นโหลดให้กับวงจร แรงดันเอาต์พุตจะมีการกระเพื่อมของไฟฟ้้ากระแสสลับบนไฟฟ้้ากระแสตรง ที่ค่าระดับอ้างอิง 2.5 โวลต์ แสดงดังรูปที่ 4.13



รูปที่ 4.13 แสดงสัญญาณเอาต์พุตไฟฟ้้ากระแสตรงที่ระดับอ้างอิง 2.5 โวลต์ เมื่อต่อภาณ์น้ำร้อนเป็นโหลด

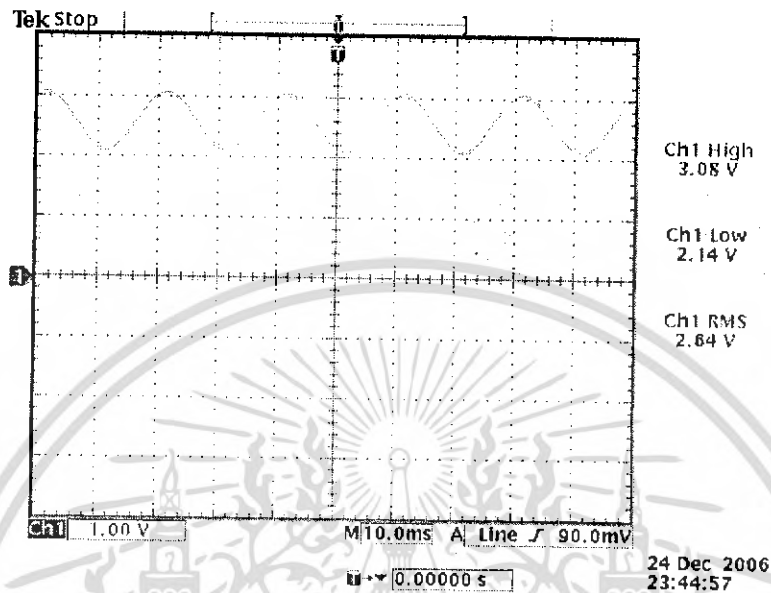
5. สัญญาณเอาต์พุตวงจรตรวจจับกระแส เมื่อต่อภาณ์น้ำร้อนและหลอดไฟรวมกันขนาด 735 วัตต์ เป็นโหลดให้กับวงจร ใช้มิเตอร์วัดแรงดันเอาต์พุตไฟฟ้้ากระแสสลับได้ 335 มิลลิโวลต์ แสดงดังรูปที่ 4.14



รูปที่ 4.14 แสดงสัญญาณแรงดันเอาต์พุตวงจรตรวจจับกระแส เมื่อต่อโหลด 735 วัตต์

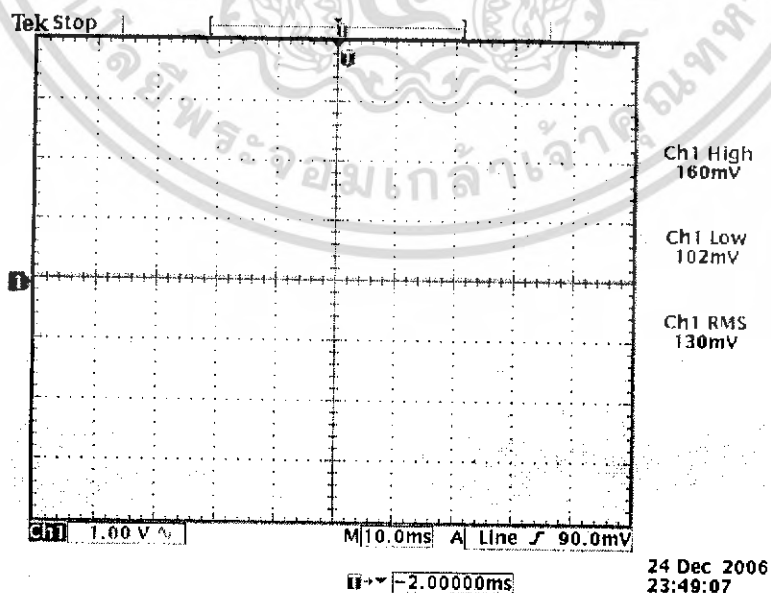
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. สัญญาณเอาต์พุตวงจรตรวจจับกระแส เมื่อต่อภาแน้ำร้อนและหลอดไฟรวมกันมีค่าขนาด 635 วัตต์ แรงดันเอาต์พุตจะมีการกระเพื่อมของไฟฟ้ากระแสสลับบนไฟฟ้ากระแสตรงได้ที่ระดับอ้างอิงค่า 2.5 โวลต์ แสดงดังรูปที่ 4.15



รูปที่ 4.15 แสดงสัญญาณเอาต์พุตไฟฟ้ากระแสตรงที่ระดับอ้างอิง 2.5 โวลต์ เมื่อต่อภาแน้ำร้อนและหลอดไฟเป็นโหลด

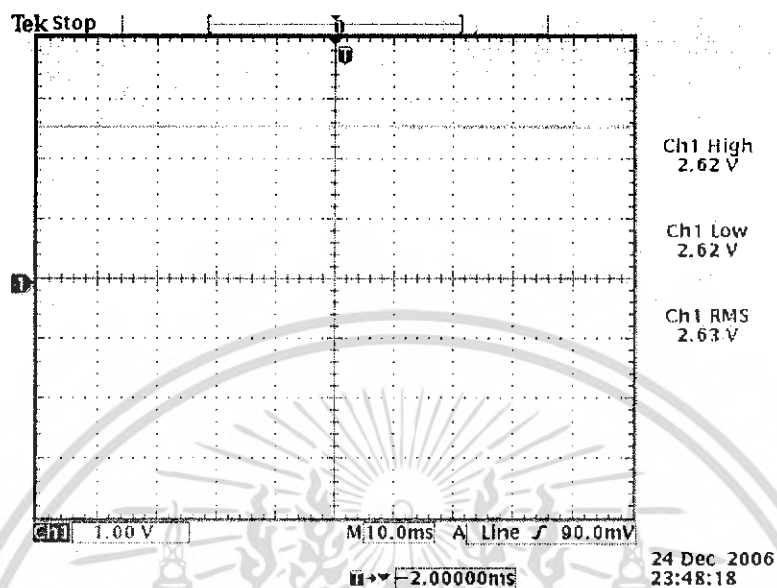
7. สัญญาณเอาต์พุตวงจรตรวจจับกระแส เมื่อไม่มีการต่อโหลดให้กับวงจร ใช้มิเตอร์วัดแรงดันเอาต์พุตไฟฟ้ากระแสสลับได้ 0 มิลลิโวลต์ แสดงดังรูปที่ 4.16



รูปที่ 4.16 แสดงสัญญาณแรงดันเอาต์พุตวงจรตรวจจับกระแส เมื่อไม่มีการต่อโหลด

เอกสารนี้เป็นเอกสารที่สงวนเวลาให้รับการใช้งานเพื่อการศึกษายเท่านั้น เมื่อผู้ใดเห็นประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. สัญญาณเอาต์พุตวงจรตรวจจับกระแส เมื่อไม่มีการต่อโหลดให้กับวงจร แรงดันเอาต์พุตของวงจรจะมีค่าเฉพาะไฟฟ้ากระแสตรงที่ระดับ 2.5 โวลต์ แสดงดังรูปที่ 4.17



รูปที่ 4.17 แสดงสัญญาณเอาต์พุตไฟฟ้ากระแสตรงที่ระดับ 2.5 โวลต์ เมื่อไม่มีการต่อโหลด

4.2 ผลการทดลองซอร์ฟแวร์

4.2.1 การสร้างแรงดันอ้างอิง

สามารถแสดงค่าจากการกำหนดระดับแรงดันอ้างอิง 2.5 โวลต์ ขณะที่ไม่มีการต่อโหลด สามารถทดสอบได้ด้วยการเขียนโปรแกรมภาษาซีควบคุมไมโครคอนโทรลเลอร์ PIC16F877 และเมื่อกดสวิตซ์รีเซต โปรแกรมจะสุ่มค่าที่มากที่สุดแล้วแสดงผลออกทางไซปอร์ตเทอร์มินอล แสดงผลที่ได้ดังรูปที่ 4.18

```

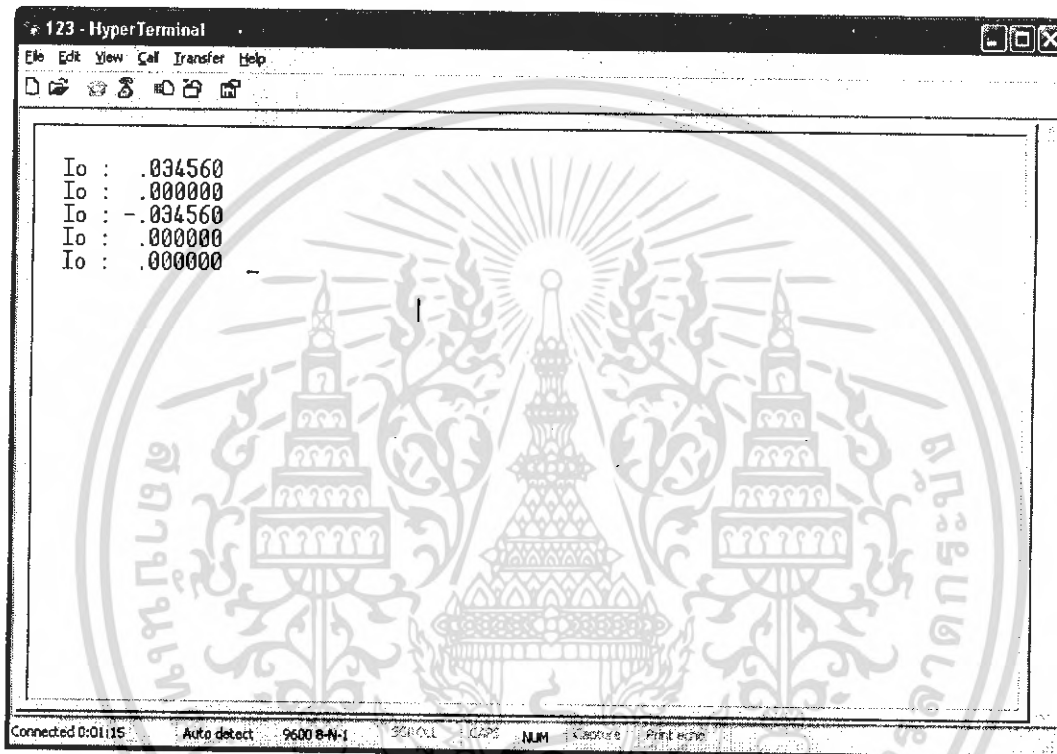
C:\HyperTerminal
File Edit View Help Device Control
D ๘ ๙ ๐ ๒ ๕
Iref : 2.551319
Iref : 2.551319
Iref : 2.551319
Iref : 2.551319
Iref : 2.551319
Connected 0:01:16 Auto detect 9600 B-N-1
  
```

รูปที่ 4.18 แสดงตัวอย่างการสุ่มค่าสูงสุดในแต่ละครั้งของแรงดันอ้างอิง 2.5 โวลต์

เอกสารนี้เป็นเอกสารลิขสิทธิ์ในชื่อสำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตเห็นประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 แสดงสัญญาณที่ไม่ต่อโหลด

ขณะไม่ต่อโหลดแรงดันเอาต์พุตจากวงจรตรวจจับกระแสจะมีค่าเท่ากับ 2.5 โวลต์ จากสูตรในการออกแบบเขียนโปรแกรมกำหนดให้ $I_O = 0.707 (I_{cur} - I_{ref}) * G$ สมการนี้จะแสดงให้เห็นว่าถ้าไม่ต่อโหลดแล้วจะทำให้สมการกลายเป็นศูนย์ และเกิดอีกกรณีหนึ่งคือเมื่อเอาต์พุตจากวงจรตรวจจับกระแสมีค่าน้อยกว่า 2.5 โวลต์ขณะไม่ต่อโหลดโปรแกรมจะไม่ทำงาน.แสดงดังรูปที่ 4.19



รูปที่ 4.19 แสดงการทำงานของโปรแกรมขณะไม่ต่อโหลด

4.2.3 เวลาที่ใช้แปลงสัญญาณอนาล็อกเป็นดิจิทัล

จากรูปแสดงเวลาที่ใช้สุ่มสัญญาณเพื่อเลือกค่าสูงสุดก่อนจะนำไปคำนวณ จากการสุ่มสัญญาณนี้จะทำงานใน 2 คาบเวลาจะได้จำนวนครั้งในการแซมปลิงเท่ากับ 2500 ครั้ง ก็สามารถทำให้ค่าละเอียดได้ เพราะถ้าสุ่มสัญญาณเพียงคาบเวลาเดียวอาจมีปัญหาเรื่องการเหลื่อมเฟสกันและไม่เลือกสุ่มทุกคาบเวลา เนื่องจากรูปสัญญาณจะซ้ำเดิมไปเรื่อยๆ จึงไม่มีความจำเป็นต้องทำทุกคาบเวลา ดังนั้นเลือกพิจารณาแค่ 2 คาบเวลา แสดงดังรูปที่ 4.20

```

time : 40483.998358 us
time : 40321.998596 us
time : 40341.998338 us
time : 40400.998592 us
time : 40243.998765 us
time : 40288.998186 us
time : 40260.998308 us
time : 40285.998582 us
time : 40304.998159 us
time : 40340.999066 us

```

Connected 0:00:22 Auto detect 9600 8-N-1

รูปที่ 4.20 แสดงเวลาที่ใช้ในกระบวนการสุ่มสัญญาณ 2 คาบเวลา

4.2.4 กำลังงานไฟฟ้าขณะที่ต่อโหลดและไม่ต่อโหลด

แสดงรูปสัญญาณที่ได้จากเครื่องวัดกำลังงานไฟฟ้า เมื่อเปิดพัดลมปุ่ม 1 หลังจากนั้นปิดพัดลม

แสดงดังรูปที่ 4.21

```

voltage : 221.999996 Volt
power : 9.677906 Watt
current : .172802 Amp
voltage : 221.999996 Volt
power : 9.688563 Watt
current : .000000 Amp
voltage : 221.999996 Volt
power : 9.688563 Watt
current : .000000 Amp
voltage : 221.999996 Volt
power : 9.688563 Watt
current : .000000 Amp
voltage : 221.999996 Volt

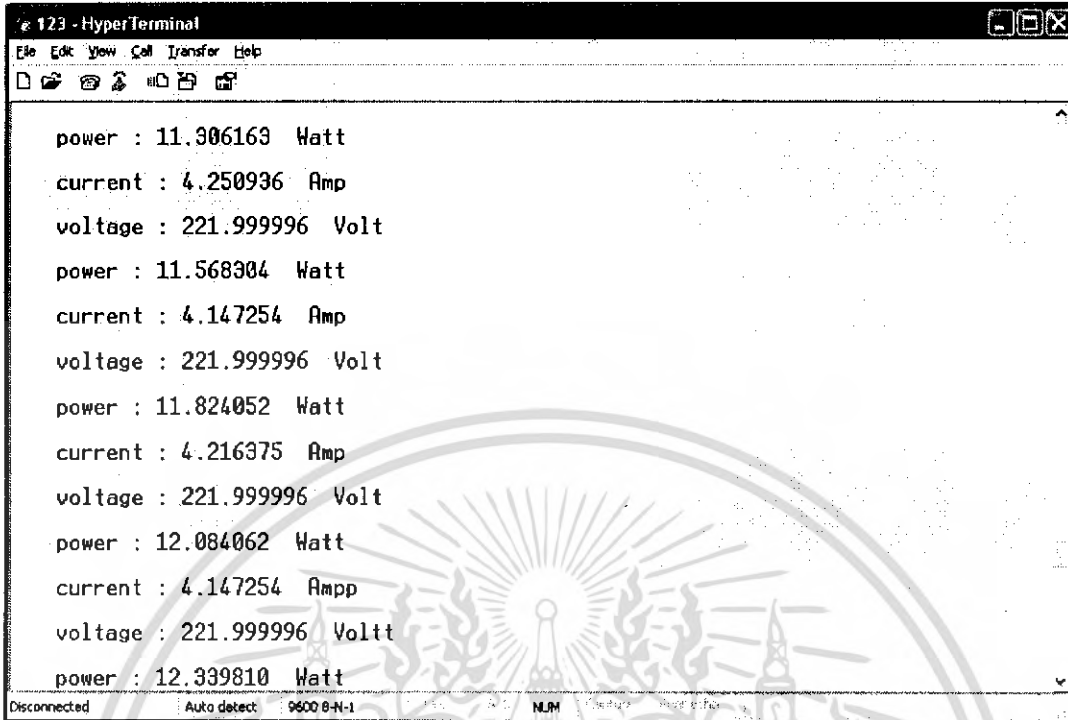
```

Disconnected Auto detect 9600 8-N-1

รูปที่ 4.21 แสดงกำลังงานไฟฟ้าเมื่อเปิดพัดลมปุ่ม 1 หลังจากนั้นปิดพัดลม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงรูปสัญญาณที่ได้จากเครื่องวัดกำลังงานไฟฟ้า เมื่อต่อเตาโรค แสดงดังรูปที่ 4.22



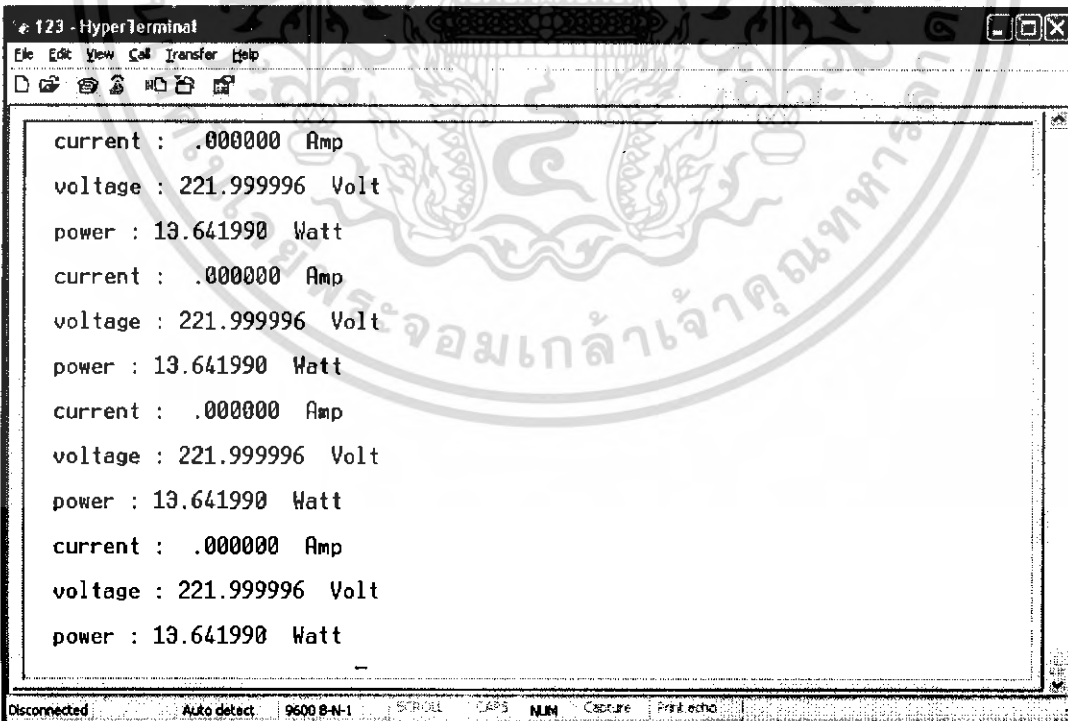
```

123 - HyperTerminal
File Edit View Call Transfer Help
power : 11.906169 Watt
current : 4.250936 Amp
voltage : 221.999996 Volt
power : 11.568904 Watt
current : 4.147254 Amp
voltage : 221.999996 Volt
power : 11.824052 Watt
current : 4.216375 Amp
voltage : 221.999996 Volt
power : 12.084062 Watt
current : 4.147254 Amp
voltage : 221.999996 Volt
power : 12.339810 Watt
Disconnected Auto detect 9600 8-N-1 SERIAL CAPS NUM Capture Print echo

```

รูปที่ 4.22 แสดงกำลังงานไฟฟ้าเมื่อต่อเตาโรค

แสดงรูปสัญญาณที่ได้จากเครื่องวัดกำลังงานไฟฟ้า ขณะไม่ต่อโหลด แสดงดังรูปที่ 4.23



```

123 - HyperTerminal
File Edit View Call Transfer Help
current : .000000 Amp
voltage : 221.999996 Volt
power : 13.641990 Watt
current : .000000 Amp
voltage : 221.999996 Volt
power : 13.641990 Watt
current : .000000 Amp
voltage : 221.999996 Volt
power : 13.641990 Watt
current : .000000 Amp
voltage : 221.999996 Volt
power : 13.641990 Watt
Disconnected Auto detect 9600 8-N-1 SERIAL CAPS NUM Capture Print echo

```

รูปที่ 4.23 แสดงกำลังงานไฟฟ้าขณะไม่ต่อโหลด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

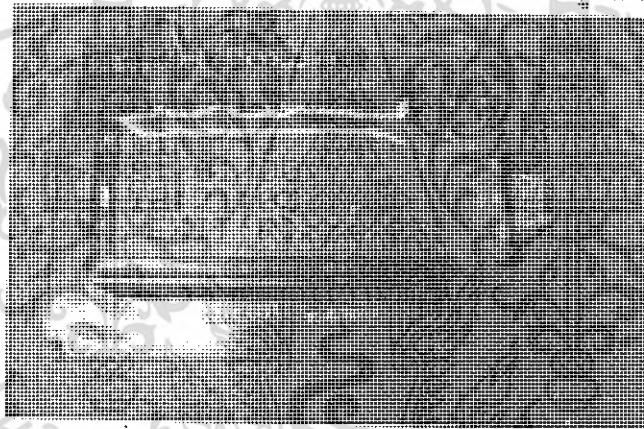
4.2.5 แสดงผลการทำงานของภาครับ

เมื่อเปิดเครื่องทางด้านภาครับ จะแสดงการทำงานหน้าหลัก ดังรูปที่ 4.24



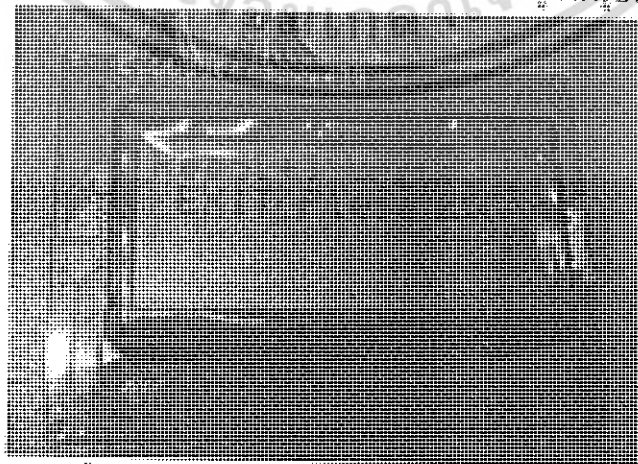
รูปที่ 4.24 แสดงการทำงานหน้าหลักของภาครับ

เมื่อกดคีย์แพคเลข 1 จะเข้าสู่การทำงานของการตั้งเวลาและวันที่ ดังรูปที่ 4.25



รูปที่ 4.25 แสดงการทำงานการตั้งเวลาและวันที่

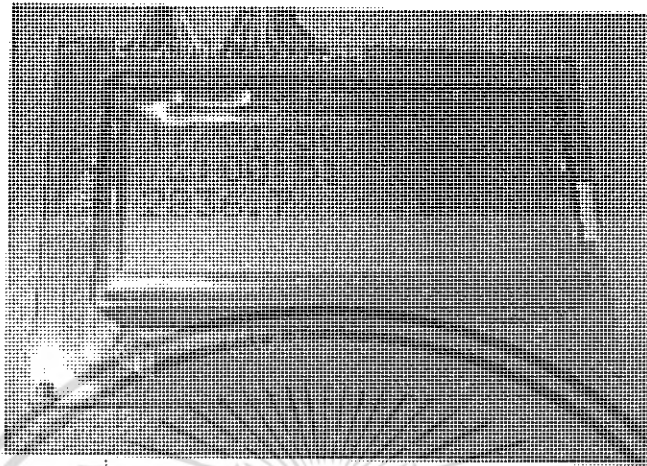
เมื่อกดคีย์แพคเลข 2 จะเข้าสู่การทำงานให้เลือกเก็บหรืออ่านข้อมูล ดังรูปที่ 4.26



รูปที่ 4.26 แสดงการทำงานให้เลือกเก็บหรืออ่านข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

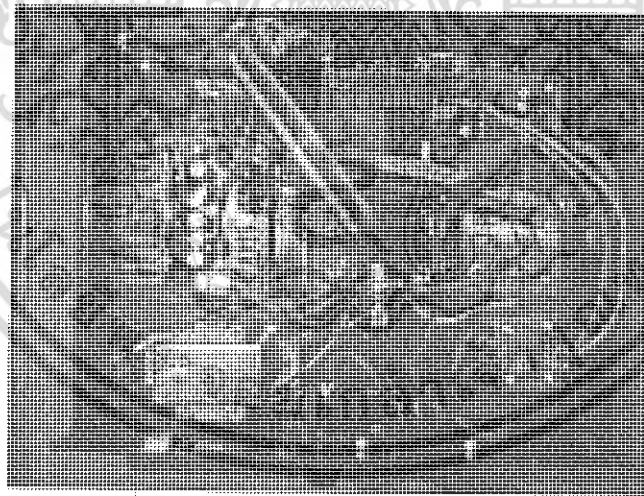
เมื่อเข้าสู่โหมดของการรับข้อมูลจากเครื่องส่ง เครื่องรับจะแสดงวันที่ เวลา และค่ากำลังงานไฟฟ้าที่หน้าจอแอลซีดีโมดูล ดังรูปที่ 4.27



รูปที่ 4.27 แสดงการรับค่ากำลังงานไฟฟ้าจากตัวส่ง

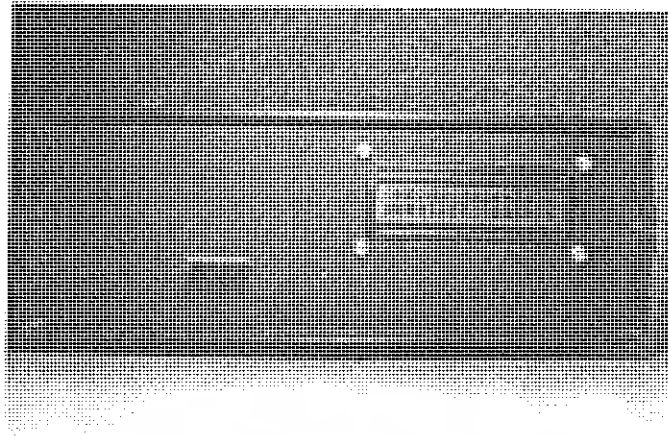
4.2.6 แสดงการต่อใช้งานจริงของภาคส่งและภาครับ

ส่วนฮาร์ดแวร์ของเครื่องส่งที่ออกแบบ ในการต่อใช้งานจริง แสดงดังรูปที่ 4.28 และแสดงรูปร่างของเครื่องส่ง ดังรูปที่ 4.29



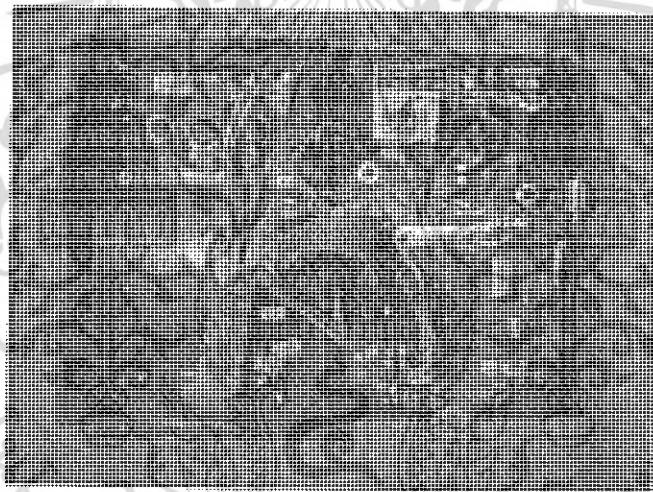
รูปที่ 4.28 แสดงการต่อวงจรใช้งานจริงภาคส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

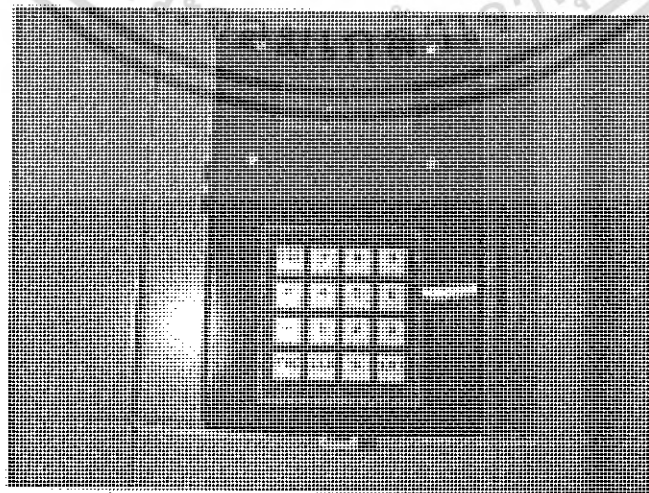


รูปที่ 4.29 แสดงอุปกรณ์ที่เสร็จสมบูรณ์ของภาคส่ง

ส่วนฮาร์ดแวร์ของเครื่องรับที่ออกแบบ ในการต่อใช้งานจริง แสดงดังรูปที่ 4.30 และแสดงรูปร่างของเครื่องส่ง ดังรูปที่ 4.31



รูปที่ 4.30 แสดงการต่อวงจร ใช้งานจริงภาครับ



รูปที่ 4.31 แสดงอุปกรณ์ที่เสร็จสมบูรณ์ของภาครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] สมยศ จุณณะปิยะ การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ ตระกูล MCS-51 พิมพ์ครั้งที่ 5: กรุงเทพมหานคร:ภาควิชาวิศวกรรมโทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2546
- [2] มงคล ทองสงคราม อิเล็กทรอนิกส์ 2 พิมพ์ครั้งที่ 5: กรุงเทพมหานคร, 2544
- [3] ไมตรี วรอุฉจรยากุล ทฤษฎีวงจรไฟฟ้า เล่ม 3 พิมพ์ครั้งที่ 3: ศูนย์การพิมพ์พลชัย กรุงเทพมหานคร, 2540
- [4] ไมตรี วรอุฉจรยากุล ทฤษฎีวงจรไฟฟ้า เล่ม 4 พิมพ์ครั้งที่ 3: ศูนย์การพิมพ์พลชัย กรุงเทพมหานคร, 2540
- [5] พันศักดิ์ พุฒิมานิตพงษ์ ทฤษฎีอิเล็กทรอนิกส์เบื้องต้น พิมพ์ครั้งที่ 1: สำนักพิมพ์ ศูนย์ส่งเสริมวิชาการ กรุงเทพมหานคร, 2538
- [6] รศ.ธีรวัฒน์ ประกอบผล ภาษาแอสเซมบลีสำหรับ MCS-51 พิมพ์ครั้งที่ 3: สำนักพิมพ์ ส.ส.ท. กรุงเทพมหานคร, 2546
- [7] ประจัน พลังสันติกุล เรียนรู้และใช้งาน CCS C คอมไพเลอร์เขียนโปรแกรมภาษา C ควบคุมไมโครคอนโทรลเลอร์ PIC
- [8] วรพงษ์ กรแก้ววัฒนกุล, ชัยวัฒน์ ลิ้มพรจิตรวิไล เรียนรู้และปฏิบัติการ ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ฌบับ AT89C5x ของ Atmel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//*****
//*****
//   โปรแกรมคำนวณ ค่ากำลังงาน ของตัวส่ง
//*****
//*****
#define _PIC16F877_
#include <16F877.h>
#define TxD    PIN_C6
#define RxD    PIN_C7
#define CLOCK_SP 4000000
#define MAX_DATA_EEPROM 256           // Data EEPROM in PIC16F877
//*****
#fuses    HS
#fuses    NOLVP, NOWDT
#fuses    NOPROTECT
//*****
#device   ADC=10
//*****
#use     delay (clock=CLOCK_SP)
#use     rs232 (baud=9600, xmit=TxD, rcv=RxD)
#use     fast_io(a)
#use     fast_io(b)
#use     fast_io(d)

#include<math.h>
#include<stdlib.h>
#include "INPUT.C"
//*****
#define Vbe    0.0048875855327468230694037145650049
#define Cur    10
#define Gt     52.5           // gain=17.5*3
#define Ref    2.551319648
#define rms    0.7071067812
#define hh     3600           // 1 ชม.
//*****
//     lcd
//*****
#locate   port_b=0x06
#bit     lcd_RW= port_b.1
#bit     lcd_RS= port_b.0
#bit     lcd_EN= port_b.2

#bit     lcd_D4= port_b.4
#bit     lcd_D5= port_b.5
#bit     lcd_D6= port_b.6
#bit     lcd_D7= port_b.7

#define msg3 " watt "
#define msg4 " NO.GB 3924-24 "

float Pt=0;
byte buffer[10];

int16 overflow=0;
#INT_TIMER1
void Int_TIMER1_isr(void){
    overflow++;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//*****
//      function prototype
//*****
float current();
float voltage();
float mal_num();
void number(float pt);
float cal_powereeprom();
void ReadIntEEPROM();
void WriteIntEEPROM(byte addr, byte value);
void lcd_init(void);
void lcd_clear(void);
void lcd_msg_title(void);
void lcd_clk(void);
void set_lcd_add(unsigned char add);
void lcd_command(unsigned char com);
void lcd_putc(unsigned char data);
void lcd_clear(void);
void lcd_str(char *s);
void power_lcd(float power);

//*****
//  main
//
//*****
void main(void){
    float power;
    set_tris_a(0x0F);
    set_tris_b(0x00);
    delay_ms(10);
    lcd_init();

    enable_interrupts(GLOBAL);
    enable_interrupts(INT_RDA);
    enable_interrupts(INT_TIMER1);
    setup_timer_1(T1_INTERNAL | T1_DIV_BY_4);
    set_timer1(15536);

    pt=cal_powereeprom();
    lcd_msg_title();           // show title lcd
    while (TRUE){

        if(overflow >4){
            overflow = 0;

            power=mal_num();    //คำนวณค่า power
            number(power);     // collect to eeprom
            power_lcd(power);  // lcd

        }

    }

}

//*****
// calculate p=v*i
//*****
float mal_num(){
    float c,v,p;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        c=current();
        v=voltage();
        p=(v*c)/hh;
        Pt=Pt+p;
        return(Pt);
    }

    /*******
    // อ่านค่า ADC จากเซนเซอร์กระแส
    /*******
    float current() {
        int16 value,i,ct=0;
        float curr,Io;

        setup_port_a(ALL_ANALOG);
        setup_adc(ADC_CLOCK_INTERNAL);
        set_adc_channel(1);
        delay_us(10);
        set_tris_a(0x0F);
        for(i=0;i<2500;i++)
        {
            value = read_ADC(); //--- A/D --- value
            if(value>ct)
            {
                ct=value;
            }
            delay_us(5);
        }
        delay_us(10);
        curr = Vbe*(float)ct;
        Io = (curr-Ref)*Cur*rms;
        if(Io<=0.034560){
            Io=0;
        }
        return(Io);
    }

    /*******
    // อ่านค่า ADC จากเซนเซอร์โวลต์
    /*******
    float voltage() {
        int16 value1,i,vt=0;
        float volt,Vo;

        setup_port_a(ALL_ANALOG);
        setup_adc(ADC_CLOCK_INTERNAL);
        set_adc_channel(2);
        delay_us(10);
        set_tris_a(0x0F);
        for(i=0;i<1250;i++)
        {
            value1 = read_ADC(); //--- A/D --- value
            if(value1>vt)
            {
                vt=value1;
            }
            delay_us(5);
        }
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    delay_us(10);
    volt = Vbe*(float)vt;
    Vo = (volt)*Gt*rms;
    return(Vo);
}

/*****
*   อ่านค่าpower ใน eeprom ที่เก็บไว้ล่าสุด
*
*****/
float cal_powereeprom(){
    float Ptotal=0,x=10000.0,pto=0;
    int i;
        ReadIntEEPROM();

        for(i=0;i<9;i++) {
            Ptotal=(float)buffer[i]*x;
            pto=pto+Ptotal;
            x=x/10.0;
        }
    return(pto);
}

/*****
* FUNCTION:      ReadIntEEPROM
* DESCRIPTION:   Read Internal EEPROM
* PARAMETERS:    nothing
* RETURNED:     nothing
*****/
void ReadIntEEPROM(void) {
    int volume;
    for(volume=0; volume<9; volume++){
        buffer[volume] = read_eeprom(volume);
    }
}

/*****
// จำนวนจำนวนตัวเลข เก็บ eeprom
*****/
void number(float pt1){
    int i;
    int32 ptt,ter=100000000;
    char num[10];
        ptt=pt1*10000;

        for( i=0; i<9;i++){
            num[i]=ptt/ter;          //x=100000
            ptt=fmod(ptt,ter);
            WriteIntEEPROM(i,num[i]); //Write EEPROM
            ter=ter/10;
        }
}

/*****
* FUNCTION:      WriteIntEEPROM
* DESCRIPTION:   Write Internal EEPROM
* PARAMETERS:    nothing
* RETURNED:     nothing
*****/

```

```

void WriteIntEEPROM(byte addr, byte value) {
    write_eeprom(addr,value);           //write eeprom
}

//*****
//      function lcd
//*****
void lcd_command(unsigned char com){
    lcd_RS=0;
    if(com & 0x80) lcd_D7=1;else lcd_D7=0;
    if(com & 0x40) lcd_D6=1;else lcd_D6=0;
    if(com & 0x20) lcd_D5=1;else lcd_D5=0;
    if(com & 0x10) lcd_D4=1;else lcd_D4=0;
    lcd_clk();

    if(com & 0x08) lcd_D7=1;else lcd_D7=0;
    if(com & 0x04) lcd_D6=1;else lcd_D6=0;
    if(com & 0x02) lcd_D5=1;else lcd_D5=0;
    if(com & 0x01) lcd_D4=1;else lcd_D4=0;
    lcd_clk();
}

//*****
//      msg char data
//*****
void lcd_putc(unsigned char data){
    lcd_RS=1;
    if(data & 0x80) lcd_D7=1;else lcd_D7=0;
    if(data & 0x40) lcd_D6=1;else lcd_D6=0;
    if(data & 0x20) lcd_D5=1;else lcd_D5=0;
    if(data & 0x10) lcd_D4=1;else lcd_D4=0;
    lcd_clk();

    if(data & 0x08) lcd_D7=1;else lcd_D7=0;
    if(data & 0x04) lcd_D6=1;else lcd_D6=0;
    if(data & 0x02) lcd_D5=1;else lcd_D5=0;
    if(data & 0x01) lcd_D4=1;else lcd_D4=0;
    lcd_clk();
}

//*****
//      clear lcd
//*****
void lcd_clear(void){
    lcd_command(0x01);
}

//*****
//      set start lcd
//*****
void set_lcd_add(unsigned char add){
    add=add | 0x80;
    lcd_command(add);
}

//*****
//      lcd initital
//*****
void lcd_init(void){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

delay_ms(100);
lcd_EN=0;
lcd_RW=0;
lcd_RS=0;
lcd_command(0x33); //send mode 4 bit
lcd_command(0x32); //send mode 4 bit
lcd_command(0x28);
lcd_command(0x0c);
lcd_command(0x01);
delay_ms(20);
lcd_clear();
}
//*****
// send clock pulse
//*****
void lcd_clk(void){
    lcd_EN=1;
    delay_ms(2);
    lcd_EN=0;
}

//*****
// copy string
//*****
void lcd_str(char *s){
    while(*s!=0){
        lcd_putc(*s++);
    }
}

//*****
// show msg title
//*****
void lcd_msg_title(void){
    unsigned char ss[16];

    lcd_clear();
    strcpy(ss,msg4);
    set_lcd_add(0x00);
    lcd_str(ss);
    delay_us(10);
    strcpy(ss,msg3);
    set_lcd_add(0x48);
    lcd_str(ss);
}

//*****
// show power on lcd
//*****
void power_lcd(float po){
    int i,b=0x30;
    int32 ptt,ter=1000000;
    char num[10];
    ptt=po*100;
    set_lcd_add(0x40);
    for( i=0; i<7;i++){
        num[i]=ptt/ter; //x=100000
        ptt=fmod(ptt,ter);
        WriteIntEEPROM(i,num[i]); //Write EEPROM
        ter=ter/10;
        num[i]=num[i]|b;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        lcd_putc(num[i]);
        if(i==4) {
            lcd_putc(".");
        }
    }

}

//*****
//      interrupte TxD & RxD
//*****
#INT_RDA
void rs232_data(void){
    char  number1=0,num[7];
    int  i,j,number,value;
    value=getc();
    delay_ms(10);

    if(value==1){
        for(i=0; i<6; i++) {
            num[i] = read_eeprom(i);
        }

        if(num[0]!=0){
            number=7;
        }else if(num[1]!=0){
            number=6;
        }else if(num[2]!=0){
            number=5;
        }else if(num[3]!=0){
            number=4;
        }else {
            number=3;
        }

        delay_ms(10);
        putc(number); //send total data

        for(j=7-number;j<6;j++){
            delay_ms(100);
            num[j]=num[j] | 0x30;
            putc(num[j]);

            if(j==4) {
                delay_ms(100);
                putc('.');
            }
        }
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*****
; PROGRAM : data collection
; DESCRIPTION
; PROGRAM : Read & Write EEPROM 24LC256
; PROGRAM : REAL TIME CLOCK
; PROGRAM : SEND TO PC
;*****

```

```

;----- Define Port& pin Name -----

```

```

SDA          BIT    P2.6
SCL          BIT    P2.7
LCD_EN      BIT    P2.2
LCD_RS      BIT    P2.0
LCD_RW      BIT    P2.1

```

```

;----- Define User Register -----

```

```

FLAG        EQU    02FH
I2C_ACK     BIT    FLAG.0
ADDRESS     EQU    02EH
O0          BIT    ADDRESS.0
O1          BIT    ADDRESS.1
O2          BIT    ADDRESS.2
O3          BIT    ADDRESS.3
O4          BIT    ADDRESS.4
O5          BIT    ADDRESS.5
O6          BIT    ADDRESS.6
O7          BIT    ADDRESS.7

```

```

;----- Define User Register -----

```

```

LCD_ADDR    EQU    030H
LCD_DATA    EQU    031H
LCD_PTR     EQU    032H
I2C_ADDR    EQU    033H
I2C_DATA    EQU    034H
KPAD_DATA   EQU    035H
BUFFER      EQU    036H

```

```

SECONDS     EQU    040H
MINUTES     EQU    041H
HOURS       EQU    042H
DAY         EQU    043H
DATE        EQU    044H
MONTH       EQU    045H
YEAR        EQU    046H
CONTROL     EQU    047H

```

```

SECONDS_1   EQU    050H
MINUTES_1   EQU    049H
HOURS_1     EQU    04AH
DAY_1       EQU    04BH
DATE_1      EQU    04CH
MONTH_1     EQU    04DH
YEAR_1      EQU    04EH

```

```

SECONDS_2   EQU    051H
MINUTES_2   EQU    052H
HOURS_2     EQU    053H
DATE_2      EQU    055H
MONTH_2     EQU    056H
YEAR_2      EQU    057H
RR          EQU    059H
NUMBER_ADDRESS EQU    05AH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;----- Define User Register memory -
M_0 EQU 054H
M_1 EQU 038H
M_2 EQU 039H
M_3 EQU 03AH
M_4 EQU 03BH
M_5 EQU 03CH
M_6 EQU 03DH
M_7 EQU 03EH
;***** Define I2C Slave Address*****
EPROM_ID EQU 1010000B
RTC_ID EQU 11010000B
;*****
;----- Main Program -----
;*****
ORG 0000H
MOV P0,#00H
MOV PCON,#00H
MOV TMOD,#21H
MOV TL1,#0FDH
MOV TH1,#0FDH
SETB TR1 ;set timer 1
SETB SDA
SETB SCL
LCALL DELAY_10ms
MAIN:
MAIN_LOOP:
LCALL INIT_LCD
LCALL LCD_CLR
MOV LCD_ADDR,#080H
LCALL SET_ADDR_LCD
MOV DPTR,#TITLE_8 ;-
***** 1
LCALL WRLINE_LCD
LCALL DELAY_10ms
MOV LCD_ADDR,#0C0H
LCALL SET_ADDR_LCD
MOV DPTR,#TITLE_4 ;-
***** 2
LCALL WRLINE_LCD
LCALL DELAY_10ms
MOV LCD_ADDR,#090H
LCALL SET_ADDR_LCD
MOV DPTR,#TITLE_5 ;-
***** 3
LCALL WRLINE_LCD
LCALL DELAY_10ms
LCALL WAIT_KEY ;---->>Check key
LCALL DELAY_100ms
LCALL WAIT_KEYPRESSED
LCALL DELAY_100ms
MOV A,KPAD_DATA
CJNE A,#1,SET_1
LCALL DELAY_100ms
LJMP LOOP_TIM_DAT ;---->> JUMP SET_TIME&DATE
SET_1:
CJNE A,#2,SET_2 ;-->>read & write
LCALL DELAY_100ms
SJMP LOOP

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SET_2:          CJNE  A,#3,MAIN_LOOP    ;--->>send pc
                LCALL DELAY_100ms
                LJMP  SEND_PC
;*****
WRITE_E_1:     CJNE  A,#64H,LOOP
                LJMP  WRITE_E
;*****
;--->>SET_1     READ&WRITE
;*****

LOOP:          LCALL LCD_CLR
                MOV   LCD_ADDR,#080H
                LCALL SET_ADDR_LCD
                LCALL DELAY_10ms
                MOV   DPTR,#TITLE_3    ;----->> Read[+] Write[-]
                LCALL WRLINE_LCD
                LCALL DELAY_10ms
                MOV   LCD_ADDR,#0C0H   ;--->>EXIT PLEASE KEY
                LCALL SET_ADDR_LCD
                MOV   DPTR,#TITLE_12
                LCALL WRLINE_LCD

                LCALL WAIT_KEY
                LCALL DELAY_100ms
                LCALL WAIT_KEYPRESSED
                MOV   A,KPAD_DATA

                CJNE  A,#45H,L_0
                LJMP  MAIN_LOOP

L_0:          CJNE  A,#61H,WRITE_E_1    ;----->> Read[+]
                ;----->> Read eeprom

                LCALL DELAY_100ms
                LCALL LCD_CLR

                ACALL LOOP_1

                MOV   LCD_ADDR,#090H
                LCALL SET_ADDR_LCD
                LCALL DELAY_10ms

                ACALL READ_EE           ;----->> CALL READ EEPROM
                LCALL DELAY_100ms

                MOV   A,RR

                CJNE  A,#7,TEST_60
                SJMP  S_70

TEST_60:      CJNE  A,#6,TEST_50
                SJMP  S_60

TEST_50:      CJNE  A,#5,TEST_40
                SJMP  S_50

TEST_40:      CJNE  A,#4,TEST_30
                SJMP  S_40

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกริใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TEST_30:          CJNE  A,#3,TEST_20
                  SJMP  S_30

TEST_20:          CJNE  A,#2,TEST_10
                  SJMP  S_20

TEST_10:          SJMP  S_10

S_70:             MOV   LCD_DATA,M_6           ;--->> Show data
10G00             LCALL WRCHAR_LCD           ;--->> Show data

S_60:             MOV   LCD_DATA,M_5           ;--->> Show data 1000
                  LCALL WRCHAR_LCD           ;--->> Show data

S_50:             MOV   LCD_DATA,M_4           ;--->> Show data 100
                  LCALL WRCHAR_LCD           ;--->> Show data

S_40:             MOV   LCD_DATA,M_3           ;--->> Show data 10
                  LCALL WRCHAR_LCD           ;--->> Show data

S_30:             MOV   LCD_DATA,M_2           ;--->> Show data 1
                  LCALL WRCHAR_LCD           ;--->> Show data

S_20:             MOV   LCD_DATA,M_1           ;--->> Show data .
                  LCALL WRCHAR_LCD           ;--->> Show data

S_10:             MOV   LCD_DATA,M_0           ;--->> Show data .1
                  LCALL WRCHAR_LCD           ;--->> Show data

                  MOV   LCD_ADDR,#098H
                  LCALL SET_ADDR_LCD
                  MOV   DPTR,#TITLE_9         ;--->> WATT
                  LCALL WRLINE_LCD8
                  LCALL DELAY_10ms

                  LCALL SHOW_TIME_MEMO

                  LCALL WAIT_KEY
                  LCALL DELAY_100ms
                  LJMP  LOOP

;*****
;   SHOW_LCD_DATA
;*****
SHOW_LCD_DATA:    MOV   LCD_DATA,A
                  LCALL WRCHAR_LCD           ;--->>SHOW_LCD_DATA
                  LCALL DELAY_10ms
                  RET

;*****
;   can't find
;*****
ERROR:           LCALL LCD_CLR
                  MOV   LCD_ADDR,#0C0H
                  LCALL SET_ADDR_LCD
                  MOV   DPTR,#TITLE_2         ; ---   can't find
                  LCALL WRLINE_LCD
                  LCALL DELAY_1
                  LJMP  LOOP

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*****
;---->> Receiver and WRITE_EEPROM
;*****

WRITE_E:          LCALL DELAY_10ms          ;---->> WRITE[-]

MOV              SCON,#40H                ;-->>Serial Model REN=0-
>SEND

ACALL LOOP_1          ;----->> ADDRESS -----

LCALL DELAY_10ms
MOV  A,NUMBER_ADDRESS
ACALL SEND

SETB REN          ;--->>Serial Model REN=1

LCALL DELAY_100ms

JNB  RI,ERROR          ;->> Receiver Total Data
send

MOV  A,SBUF
CLR  RI

MOV  RR,A          ; -->> RR=TOTAL DATA

MOV  LCD_ADDR,#090H
LCALL SET_ADDR_LCD

MOV  A,RR

CJNE A,#7,TEST_6
SJMP S_7

TEST_6:          CJNE A,#6,TEST_5
SJMP S_6

TEST_5:          CJNE A,#5,TEST_4
SJMP S_5

TEST_4:          CJNE A,#4,TEST_3
SJMP S_4

TEST_3:          CJNE A,#3,TEST_2
SJMP S_3

TEST_2:          CJNE A,#2,TEST_1
SJMP S_2

TEST_1:          SJMP S_1
;.....
S_7:             ACALL REC
MOV  M_6,A
ACALL SHOW_LCD_DATA

S_6:             ACALL REC
MOV  M_5,A
ACALL SHOW_LCD_DATA

S_5:             ACALL REC
MOV  M_4,A
ACALL SHOW_LCD_DATA

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

S_4:          ACALL REC
              MOV   M_3,A
              ACALL SHOW_LCD_DATA

S_3:          ACALL REC
              MOV   M_2,A
              ACALL SHOW_LCD_DATA

S_2:          ACALL REC
              MOV   M_1,A
              ACALL SHOW_LCD_DATA

S_1:          ACALL REC
              MOV   M_0,A
              ACALL SHOW_LCD_DATA

              MOV   LCD_ADDR,#098H
              LCALL SET_ADDR_LCD
              MOV   DPTR,#TITLE_9           ;---->> WATT
              LCALL WRLINE_LCD8
              LCALL DELAY_10ms

              LCALL RTC_RD                 ;---->> READ TIME&DATE

              MOV   LCD_ADDR,#080H
              LCALL SET_ADDR_LCD
              LCALL DELAY_10ms
              ACALL SHOW_DATELCD          ;---->> SHOW TIME&DATE

              MOV   LCD_ADDR,#0C0H
              LCALL SET_ADDR_LCD
              LCALL DELAY_10ms
              ACALL SHOW_TIMELCD          ;---->> SHOW TIME&DATE

              LCALL WAIT_KEY               ;--push key save or
clear
              LCALL DELAY_100ms
              LCALL WAIT_KEYPRESSED
              LCALL DELAY_100ms
              MOV   A,KPAD_DATA
              CJNE A,#59H,JUMP_CLR
              ACALL WRITE_EE
              LJMP  LOOP

;*****
;   show msg clear
;*****

JUMP_CLR:    LCALL LCD_CLR
              MOV   LCD_ADDR,#080H
              LCALL SET_ADDR_LCD
              MOV   DPTR,#TITLE_11        ; --- Clear
              LCALL WRLINE_LCD
              LCALL DELAY_1
              LCALL DELAY_1
              LJMP  LOOP

;*****
;   SEND TO SBUF
;*****
SEND:        MOV   SBUF,A                 ;-->> send
WAIT1:       JNB  TI,WAIT1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                CLR    TI
                RET
;*****
; Receiver TO SBUF
;*****
REC:            JNB    RI,REC                ;-->>receiver
                MOV    A,SBUF
                CLR    RI
                RET
;*****
SELECT_33:     LJMP   SELECT_3
;*****
;----->> address Adjust level bit
;*****
LOOP_1:        LCALL  LCD_CLR
                MOV    LCD_ADDR,#080H
                LCALL  SET_ADDR_LCD
                MOV    DPTR,#TITLE_6        ; --- ADDRESS
                LCALL  WRLINE_LCD
                MOV    R1,#8

                MOV    LCD_ADDR,#0C0H
                LCALL  SET_ADDR_LCD
                MOV    A,#00H
                ACALL  SELECT_1            ; -----START
                CJNE   A,#0,H1
                CLR    O7
                SJMP   H2
H1:            CJNE   A,#1,SELECT_33
                SETB   O7
                SJMP   H2
;.....
H2:            ACALL  SELECT_1
                CJNE   A,#0,H3
                CLR    O6
                SJMP   H4
H3:            CJNE   A,#1,SELECT_3
                SETB   O6
                SJMP   H4
;.....
H4:            ACALL  SELECT_1
                CJNE   A,#0,H5
                CLR    O5
                SJMP   H6
H5:            CJNE   A,#1,SELECT_3
                SETB   O5
                SJMP   H6
;.....
H6:            ACALL  SELECT_1
                CJNE   A,#0,H7
                CLR    O4
                SJMP   H8
H7:            CJNE   A,#1,SELECT_3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        SETB  O4
        SJMP  H8
        ;.....

H8:      ACALL SELECT_1
        CJNE  A,#0,H9
        CLR   O3
        SJMP  H10

H9:      CJNE  A,#1,SELECT_3
        SETB  O3
        SJMP  H10
        ;.....

H10:     ACALL SELECT_1
        CJNE  A,#0,H11
        CLR   O2
        SJMP  H12

H11:     CJNE  A,#1,SELECT_3
        SETB  O2
        SJMP  H12
        ;.....

H12:     ACALL SELECT_1
        CJNE  A,#0,H13
        CLR   O1
        SJMP  H14

H13:     CJNE  A,#1,SELECT_3
        SETB  O1
        SJMP  H14
        ;.....

H14:     ACALL SELECT_1
        CJNE  A,#0,H15
        CLR   O0
        SJMP  H16

H15:     CJNE  A,#1,SELECT_3
        SETB  O0
        SJMP  H16
        ;.....

H16:     LCALL LCD_CLR
        MOV   NUMBER_ADDRESS,ADDRESS
        MOV   A,ADDRESS      ;--->> Resersh channel
eprom
        MOV   B,#014H
        MUL  AB
        MOV   ADDRESS,A
        RET

;*****
; no 0-1 return
;*****

SELECT_3:  MOV   DPTR,#TITLE_7
           LCALL WRLINE_LCD
           LCALL DELAY_100ms

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        LCALL DELAY_1
        LJMP LOOP_1

;*****
;---->> set level bit from key pad
;*****

SELECT_1:    LCALL LCD_BLINK
             LCALL WAIT_KEY           ;----Chack Key
             LCALL DELAY_100ms
             LCALL WAIT_KEYPRESSED
             MOV  R0,KPAD_DATA
             LCALL KPAD2LCD
             MOV  A,R0
             LCALL DELAY_100ms
             RET

;*****
;---->> SET TIME & DATE
;*****

LOOP_TIM_DAT:  LCALL LCD_CLR
              MOV  LCD_ADDR,#080H      ;---->>SET TIME & DATE
              LCALL SET_ADDR_LCD
              MOV  DPTR,#TITLE_10
              LCALL WRLINE_LCD
              MOV  LCD_ADDR,#0D0H      ;---->>EXIT PLESE KEY
              LCALL SET_ADDR_LCD
              MOV  DPTR,#TITLE_12
              LCALL WRLINE_LCD

LOOP_WAIT:    LCALL DELAY_10ms
             LCALL RTC_RD             ;---->> READ TIME&DATE
             MOV  LCD_ADDR,#0C0H
             LCALL SET_ADDR_LCD
             LCALL DELAY_10ms
             ACALL SHOW_DATELCD       ;---->> SHOW DATE

             MOV  LCD_ADDR,#090H
             LCALL SET_ADDR_LCD
             LCALL DELAY_10ms
             ACALL SHOW_TIMELCD       ;---->> SHOW TIME

             JNB P2.3,LOOP_WAIT       ;-----Check Key

             LCALL DELAY_10ms
             LCALL WAIT_KEYPRESSED

             MOV  A,KPAD_DATA
             LCALL DELAY_10ms
             CJNE A,#61H,MM_1         ;----> +
             LCALL DELAY_10ms
             ACALL SET_TIME
             SJMP LOOP_TIM_DAT

MM_1:        CJNE A,#64H,MM_2         ;----> -
             LCALL DELAY_10ms
             ACALL SET_DATE
             LJMP LOOP_TIM_DAT

MM_2:        CJNE A,#45H,MM_3         ;----> ESC

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LJMP MAIN_LOOP

MM_3:      LCALL LCD_CLR
           LCALL DELAY_10ms
           MOV LCD_ADDR,#083H
           LCALL DELAY_100ms
           LCALL SET_ADDR_LCD
           MOV LCD_DATA,#'F'
           LCALL WRCHAR_LCD
           MOV LCD_DATA,#'a'
           LCALL WRCHAR_LCD
           MOV LCD_DATA,#'1'
           LCALL WRCHAR_LCD
           MOV LCD_DATA,#'s'
           LCALL WRCHAR_LCD
           MOV LCD_DATA,#'e'
           LCALL WRCHAR_LCD
           LCALL DELAY_1
           LJMP LOOP_TIM_DAT

;*****
;      EPROM WRITE
;*****

WRITE_EE:  MOV I2C_ADDR,#EPROM_ID
           LCALL I2C_SLAVE

           MOV I2C_DATA,#00H           ;ADDRESS EPROM high
           ACALL I2C_DATA_WR

           MOV I2C_DATA,ADDRESS       ;ADDRESS EPROM lows
           ACALL I2C_DATA_WR
           LCALL DELAY_10ms

           MOV I2C_DATA,RR            ;-->>Total data
           ACALL I2C_DATA_WR
           LCALL DELAY_10ms

           MOV I2C_DATA,M_0           ;---->> .1
           ACALL I2C_DATA_WR

           DJNZ RR,NEXT_1
           LJMP STO_P

NEXT_1:    LCALL DELAY_10ms
           MOV I2C_DATA,M_1           ;---->> .
           ACALL I2C_DATA_WR
           DJNZ RR,NEXT_2
           LJMP STO_P

NEXT_2:    LCALL DELAY_10ms
           MOV I2C_DATA,M_2           ;---->> 1
           ACALL I2C_DATA_WR
           DJNZ RR,NEXT_3
           LJMP STO_P

NEXT_3:    LCALL DELAY_10ms
           MOV I2C_DATA,M_3           ;---->> 10
           ACALL I2C_DATA_WR
           DJNZ RR,NEXT_4
           LJMP STO_P

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

NEXT_4:      LCALL DELAY_10ms
              MOV   I2C_DATA,M_4           ;--->> 100
              ACALL I2C_DATA_WR
              DJNZ  RR,NEXT_5
              SJMP  STO_P

NEXT_5:      LCALL DELAY_10ms
              MOV   I2C_DATA,M_5           ;--->> 1000
              ACALL I2C_DATA_WR
              DJNZ  RR,NEXT_6
              SJMP  STO_P

NEXT_6:      LCALL DELAY_10ms
              MOV   I2C_DATA,M_6           ;--->> 10000
              ACALL I2C_DATA_WR

STO_P:       LCALL DELAY_10ms
              MOV   I2C_DATA,DATE_1       ;--->>DATE FIRST
              ACALL I2C_DATA_WR

              LCALL DELAY_10ms
              MOV   I2C_DATA,DATE_2       ;--->>DATE Second
              ACALL I2C_DATA_WR

              LCALL DELAY_10ms
              MOV   I2C_DATA,MONTH_1      ;--->>MONTH FIRST
              ACALL I2C_DATA_WR

              LCALL DELAY_10ms
              MOV   I2C_DATA,MONTH_2      ;--->>MONTH
              ACALL I2C_DATA_WR

              LCALL DELAY_10ms
              MOV   I2C_DATA,YEAR_1       ;--->>YEAR FIRST
              ACALL I2C_DATA_WR

              LCALL DELAY_10ms
              MOV   I2C_DATA,YEAR_2       ;--->>YEAR
              ACALL I2C_DATA_WR

              LCALL DELAY_10ms
              MOV   I2C_DATA,HOURS_1      ;--->>hh First
              ACALL I2C_DATA_WR

              LCALL DELAY_10ms
              MOV   I2C_DATA,HOURS_2      ;--->>hh
              ACALL I2C_DATA_WR

              LCALL DELAY_10ms
              MOV   I2C_DATA,MINUTES_1     ;--->>mm First
              ACALL I2C_DATA_WR

              LCALL DELAY_10ms
              MOV   I2C_DATA,MINUTES_2     ;--->>mm
              ACALL I2C_DATA_WR

              LCALL DELAY_10ms
              MOV   I2C_DATA,SECONDS_1     ;--->>ss First
              ACALL I2C_DATA_WR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        LCALL DELAY_10ms
        MOV   I2C_DATA,SECONDS_2      ;--->>ss
        ACALL I2C_DATA_WR

        LCALL I2C_STOP
        RET

;*****
;   EPROM READ
;*****

READ_EE:  MOV   I2C_ADDR,#EPROM_ID      ;-->> ID=10100000B
          LCALL I2C_SLAVE
          MOV   I2C_DATA,#00H          ;-->> ADDRESS EPROM high
          ACALL I2C_DATA_WR

          MOV   I2C_DATA,ADDRESS      ;-->> ADDRESS EPROM lows
          ACALL I2C_DATA_WR

          MOV   I2C_ADDR,#EPROM_ID+1  ;-->> ID=10100001B
          LCALL I2C_SLAVE
          LCALL DELAY_10ms

          ACALL I2C_DATA_RD            ;-->> Read total data
          MOV   RR,I2C_DATA            ;-->> RR= total data
          LCALL I2C_ACK_BIT
          MOV   A,RR
          LCALL DELAY_10ms

          ACALL I2C_DATA_RD            ;-->> Read
          MOV   M_0,I2C_DATA           ;-->> .1
          LCALL I2C_ACK_BIT
          DEC   A
          JZ    GO
          LCALL DELAY_10ms

          ACALL I2C_DATA_RD            ;-->> Read
          MOV   M_1,I2C_DATA           ;-->> .
          LCALL I2C_ACK_BIT
          DEC   A
          JZ    GO
          LCALL DELAY_10ms

          ACALL I2C_DATA_RD            ;-->> Read
          MOV   M_2,I2C_DATA           ;-->> 1
          LCALL I2C_ACK_BIT
          DEC   A
          JZ    GO
          LCALL DELAY_10ms

          ACALL I2C_DATA_RD            ;-->> Read
          MOV   M_3,I2C_DATA           ;-->> 10
          LCALL I2C_ACK_BIT
          DEC   A
          JZ    GO
          LCALL DELAY_10ms

          ACALL I2C_DATA_RD            ;-->> Read
          MOV   M_4,I2C_DATA           ;-->> 100

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL I2C_ACK_BIT
DEC A
JZ GO
LCALL DELAY_10ms

ACALL I2C_DATA_RD ;-->> Read
MOV M_5,I2C_DATA ;-->> 1000
LCALL I2C_ACK_BIT
DEC A
JZ GO
LCALL DELAY_10ms

ACALL I2C_DATA_RD ;-->> Read
MOV M_6,I2C_DATA ;-->> 10000
LCALL I2C_ACK_BIT

GO: ACALL I2C_DATA_RD ;-->> Read
MOV DATE_1,I2C_DATA ;-->> DATE
LCALL I2C_ACK_BIT
LCALL DELAY_10ms

ACALL I2C_DATA_RD ;-->> Read
MOV DATE_2,I2C_DATA ;-->>
LCALL I2C_ACK_BIT
LCALL DELAY_10ms

ACALL I2C_DATA_RD ;-->> Read
MOV MONTH_1,I2C_DATA ;-->> MONTH
LCALL I2C_ACK_BIT
LCALL DELAY_10ms

ACALL I2C_DATA_RD ;-->> Read
MOV MONTH_2,I2C_DATA ;-->>
LCALL I2C_ACK_BIT
LCALL DELAY_10ms

ACALL I2C_DATA_RD ;-->> Read
MOV YEAR_1,I2C_DATA ;-->> YEAR
LCALL I2C_ACK_BIT
LCALL DELAY_10ms

ACALL I2C_DATA_RD ;-->> Read
MOV YEAR_2,I2C_DATA ;-->>
LCALL I2C_ACK_BIT
LCALL DELAY_10ms

ACALL I2C_DATA_RD ;-->> Read
MOV HOURS_1,I2C_DATA ;-->> hh
LCALL I2C_ACK_BIT
LCALL DELAY_10ms

ACALL I2C_DATA_RD ;-->> Read
MOV HOURS_2,I2C_DATA ;-->>
LCALL I2C_ACK_BIT
LCALL DELAY_10ms

ACALL I2C_DATA_RD ;-->> Read
MOV MINUTES_1,I2C_DATA ;-->> mm
LCALL I2C_ACK_BIT
LCALL DELAY_10ms

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ACALL I2C_DATA_RD          ;-->> Read
MOV  MINUTES_2,I2C_DATA   ;-->>
LCALL I2C_ACK_BIT
LCALL DELAY_10ms

ACALL I2C_DATA_RD          ;-->> Read
MOV  SECONDS_1,I2C_DATA   ;-->> ss
LCALL I2C_ACK_BIT
LCALL DELAY_10ms

ACALL I2C_DATA_RD          ;-->> Read
MOV  SECONDS_2,I2C_DATA   ;-->>

LCALL I2C_NACK_BIT
LCALL DELAY_10ms
LCALL I2C_STOP
RET

;*****
;  SHOW_TIME_DATE TO LCD
;*****
SHOW_DATELCD:
MOV  LCD_DATA,DATE
ACALL BCD2LCD
MOV  DATE_1,BUFFER
MOV  DATE_2,BUFFER+1

MOV  LCD_DATA,#'/'
LCALL WRCHAR_LCD

MOV  LCD_DATA,MONTH
ACALL BCD2LCD
LCALL DELAY_10ms
MOV  MONTH_1,BUFFER
MOV  MONTH_2,BUFFER+1

MOV  LCD_DATA,#'/'
LCALL WRCHAR_LCD

MOV  LCD_DATA,YEAR
ACALL BCD2LCD
LCALL DELAY_10ms
MOV  YEAR_1,BUFFER
MOV  YEAR_2,BUFFER+1
RET

-----
SHOW_TIMELCD:
MOV  A,HOURS
ANL  A,#00110000B
JZ   WRITE_TIME_HN
SWAP A
ADD  A,#030H
AJMP WRITE_TIME_HH

WRITE_TIME_HN:
MOV  A,#' '
WRITE_TIME_HH:
MOV  LCD_DATA,A
MOV  HOURS_1,A
LCALL WRCHAR_LCD
LCALL DELAY_10ms

MOV  A,HOURS
ANL  A,#00001111B
ADD  A,#030H
MOV  LCD_DATA,A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในหน่วยงานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV    HOURS_2,A
        LCALL WRCHAR_LCD
        LCALL DELAY_10ms

MOV    LCD_DATA,#': '
        LCALL WRCHAR_LCD
        LCALL DELAY_10ms

MOV    LCD_DATA,MINUTES
        ACALL BCD2LCD
        LCALL DELAY_10ms
MOV    MINUTES_1,BUFFER
MOV    MINUTES_2,BUFFER+1

MOV    LCD_DATA,#': '
        LCALL WRCHAR_LCD

MOV    LCD_DATA,SECONDS
        ACALL BCD2LCD
        LCALL DELAY_10ms
MOV    SECONDS_1,BUFFER
MOV    SECONDS_2,BUFFER+1
RET

;*****
;    I2C Data Write
;*****
I2C_DATA_WR:    PUSH    ACC
                SETB   I2C_ACK
                MOV    A,I2C_DATA
                MOV    R5,#008
I2C_DATA_WR_1:  RLC    A
                MOV    SDA,C
                LCALL  I2C_CLK
                DJNZ  R5,I2C_DATA_WR_1
                SETB  SDA
                LCALL  I2C_DELAY
                SETB  SCL
                LCALL  I2C_DELAY
                JB    SDA,I2C_DATA_WR_2
                CLR   I2C_ACK
I2C_DATA_WR_2:  CLR   SCL
                POP   ACC
                RET

;*****
;    I2C Data Read
;*****
I2C_DATA_RD:    PUSH    ACC
                CLR    A
                MOV    R5,#008
I2C_DATA_RD_1:  LCALL  I2C_DELAY
                SETB  SCL
                LCALL  I2C_DELAY
                MOV   C,SDA
                RLC   A

                CLR   SCL
                DJNZ  R5,I2C_DATA_RD_1
                MOV   I2C_DATA,A
                POP   ACC
                RET

;*****

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;      BCD code to show LCD
;*****
BCD2LCD:          PUSH  ACC
                  PUSH  B
                  MOV   A,LCD_DATA
                  MOV   B,A
                  ANL   A,#11110000B
                  SWAP  A
                  ADD   A,#030H
                  MOV   BUFFER,A
                  MOV   LCD_DATA,A
                  LCALL WRCHAR_LCD
                  MOV   A,B
                  ANL   A,#00001111B
                  ADD   A,#030H
                  MOV   BUFFER+1,A
                  MOV   LCD_DATA,A
                  LCALL WRCHAR_LCD
                  POP   B
                  POP   ACC
                  RET

;*****
;----- SET TIME -----
;*****
SET_TIME:        LCALL LCD_CLR
                  MOV   LCD_ADDR,#080H
                  LCALL SET_ADDR_LCD
                  LCALL DELAY_100ms
                  MOV   DPTR,#SCR_SET_TIME
                  LCALL WRLINE_LCD
                  LCALL DELAY_1

                  MOV   LCD_ADDR,#0C0H
                  LCALL SET_ADDR_LCD
                  LCALL LCD_BLINK

                  LCALL WAIT_KEY
                  LCALL DELAY_100ms
                  LCALL WAIT_KEYPRESSED
                  MOV   BUFFER,KPAD_DATA
                  LCALL KPAD2LCD
                  LCALL DELAY_100ms
                  LCALL LCD_BLINK

                  LCALL WAIT_KEY
                  LCALL DELAY_100ms
                  LCALL WAIT_KEYPRESSED
                  MOV   BUFFER+1,KPAD_DATA
                  LCALL KPAD2LCD
                  LCALL DELAY_100ms

                  LCALL BUFFER2ACC
                  MOV   HOURS,A

                  MOV   LCD_ADDR,#0C3H
                  LCALL SET_ADDR_LCD
                  LCALL LCD_BLINK

                  LCALL WAIT_KEY
                  LCALL DELAY_100ms
                  LCALL WAIT_KEYPRESSED

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV    BUFFER, KPAD_DATA
LCALL  KPAD2LCD
LCALL  DELAY_100ms

LCALL  LCD_BLINK
LCALL  WAIT_KEY
LCALL  DELAY_100ms
LCALL  WAIT_KEYPRESSED
MOV    BUFFER+1, KPAD_DATA
LCALL  KPAD2LCD
LCALL  DELAY_100ms

LCALL  BUFFER2ACC
MOV    MINUTES, A

MOV    LCD_ADDR, #0C6H
LCALL  SET_ADDR_LCD
LCALL  LCD_BLINK

LCALL  WAIT_KEY
LCALL  DELAY_100ms
LCALL  WAIT_KEYPRESSED
MOV    BUFFER, KPAD_DATA
LCALL  KPAD2LCD
LCALL  DELAY_100ms

LCALL  LCD_BLINK
LCALL  WAIT_KEY
LCALL  DELAY_100ms
LCALL  WAIT_KEYPRESSED
MOV    BUFFER+1, KPAD_DATA
LCALL  KPAD2LCD
LCALL  DELAY_100ms

LCALL  BUFFER2ACC
MOV    SECONDS, A

LCALL  DELAY_100ms
LCALL  RTC_WR
LCALL  LCD_CLR
RET

;*****
;----- SET_DATE -----
;*****
SET_DATE:
LCALL  LCD_CLR
MOV    LCD_ADDR, #080H
LCALL  SET_ADDR_LCD
MOV    DPTR, #SCR_SET_DATE
LCALL  WRLINE_LCD
LCALL  DELAY_100ms

MOV    LCD_ADDR, #0C0H
LCALL  SET_ADDR_LCD
LCALL  LCD_BLINK

LCALL  WAIT_KEY
LCALL  DELAY_100ms
LCALL  WAIT_KEYPRESSED
MOV    BUFFER, KPAD_DATA
LCALL  KPAD2LCD

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ในกิจการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL DELAY_100ms

LCALL LCD_BLINK
LCALL WAIT_KEY
LCALL DELAY_100ms
LCALL WAIT_KEYPRESSED
MOV  BUFFER+1, KPAD_DATA
LCALL KPAD2LCD
LCALL DELAY_100ms

LCALL BUFFER2ACC
MOV  DATE, A

MOV  LCD_ADDR, #0C3H
LCALL SET_ADDR_LCD
LCALL LCD_BLINK

LCALL WAIT_KEY
LCALL DELAY_100ms
LCALL WAIT_KEYPRESSED
MOV  BUFFER, KPAD_DATA
LCALL KPAD2LCD
LCALL DELAY_100ms

LCALL LCD_BLINK
LCALL WAIT_KEY
LCALL DELAY_100ms
LCALL WAIT_KEYPRESSED
MOV  BUFFER+1, KPAD_DATA
LCALL KPAD2LCD
LCALL DELAY_100ms

LCALL BUFFER2ACC
MOV  MONTH, A

MOV  LCD_ADDR, #0C6H
LCALL SET_ADDR_LCD
LCALL LCD_BLINK

LCALL WAIT_KEY
LCALL DELAY_100ms
LCALL WAIT_KEYPRESSED
MOV  BUFFER, KPAD_DATA
LCALL KPAD2LCD
LCALL DELAY_100ms

LCALL LCD_BLINK
LCALL WAIT_KEY
LCALL DELAY_100ms
LCALL WAIT_KEYPRESSED
MOV  BUFFER+1, KPAD_DATA
LCALL KPAD2LCD
LCALL DELAY_100ms

LCALL BUFFER2ACC
MOV  YEAR, A
LCALL RTC_WR
LCALL LCD_CLR
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*****
;           I2C RTC Read
;*****
RTC_RD:      MOV    I2C_ADDR,#RTC_ID
              LCALL I2C_SLAVE

              MOV    I2C_DATA,#000H
              LCALL I2C_DATA_WR

              MOV    I2C_ADDR,#RTC_ID+1
              LCALL I2C_SLAVE
              LCALL I2C_DATA_RD

              MOV    SECONDS,I2C_DATA
              LCALL I2C_ACK_BIT

              LCALL I2C_DATA_RD
              MOV    MINUTES,I2C_DATA
              LCALL I2C_ACK_BIT

              LCALL I2C_DATA_RD
              MOV    HOURS,I2C_DATA
              LCALL I2C_ACK_BIT

              LCALL I2C_DATA_RD
              MOV    DAY,I2C_DATA
              LCALL DELAY_10ms
              LCALL I2C_ACK_BIT

              LCALL I2C_DATA_RD
              MOV    DATE,I2C_DATA
              LCALL I2C_ACK_BIT

              LCALL I2C_DATA_RD
              MOV    MONTH,I2C_DATA
              LCALL I2C_ACK_BIT

              LCALL I2C_DATA_RD
              MOV    YEAR,I2C_DATA
              LCALL I2C_ACK_BIT

              LCALL I2C_DATA_RD
              MOV    CONTROL,I2C_DATA
              LCALL I2C_NACK_BIT
              LCALL I2C_STOP
              RET

;*****
;           I2C RTC Write
;*****
RTC_WR:      MOV    I2C_ADDR,#RTC_ID
              LCALL I2C_SLAVE

              MOV    I2C_DATA,#000H
              LCALL I2C_DATA_WR

              MOV    I2C_DATA,SECONDS
              LCALL I2C_DATA_WR

              MOV    I2C_DATA,MINUTES
              LCALL I2C_DATA_WR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV    I2C_DATA, HOURS
LCALL I2C_DATA_WR

MOV    I2C_DATA, DAY
LCALL I2C_DATA_WR

MOV    I2C_DATA, DATE
LCALL I2C_DATA_WR

MOV    I2C_DATA, MONTH
LCALL I2C_DATA_WR

MOV    I2C_DATA, YEAR
LCALL I2C_DATA_WR

MOV    I2C_DATA, CONTROL
LCALL I2C_DATA_WR

LCALL I2C_STOP
RET
;*****
;   Write Character to show LCD
;*****
WRCHAR_LCD:    ACALL LCD_ON
                MOV    P0, LCD_DATA
                SETB  LCD_RS
                CLR   LCD_RW
                ACALL LCD_CLK
                RET
;*****
;   I2C Slave Connect
;*****
I2C_SLAVE:    PUSH  ACC
                SETB I2C_ACK
                MOV  A, I2C_ADDR
                ACALL I2C_START
                MOV  R5, #008
I2C_SLAVE_1:  RLC   A
                MOV  SDA, C
                ACALL I2C_CLK
                DJNZ R5, I2C_SLAVE_1
                SETB SDA
                ACALL I2C_DELAY
                SETB SCL
                ACALL I2C_DELAY
                JB   SDA, I2C_SLAVE_2
                CLR  I2C_ACK
I2C_SLAVE_2:  CLR  SCL
                POP  ACC
                RET
;*****
;   I2C Strat Condition
;*****
I2C_START:    JNB   SCL, I2C_START_1
                CLR  SCL
I2C_START_1:  SETB SDA
                SETB SCL
                ACALL I2C_DELAY
                CLR  SDA
                ACALL I2C_DELAY
                CLR  SCL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                                RET
;*****
;      I2C StOP Condition
;*****
I2C_STOP:          JNB     SCL,I2C_STOP_1
                  CLR     SCL
I2C_STOP_1:       CLR     SDA
                  ACALL  I2C_DELAY
                  SETB   SCL
                  ACALL  I2C_DELAY
                  SETB   SDA
                  RET

;*****
;      I2C Clock
;*****
I2C_CLK:          LCALL  I2C_DELAY
                  SETB   SCL
                  ACALL  I2C_DELAY
                  CLR     SCL
                  RET

;*****
;      I2C Acknowledge
;*****
I2C_ACK_BIT:      CLR     SDA
                  ACALL  I2C_DELAY
                  ACALL  I2C_CLK
                  SETB   SDA
                  RET

;*****
;      I2C Not Acknowledge
;*****
I2C_NACK_BIT:     SETB   SDA
                  ACALL  I2C_DELAY
                  ACALL  I2C_CLK
                  SETB   SCL
                  RET

;*****
;      LCD Initialize
;*****
INIT_LCD:         ACALL  DELAY_100ms
                  CLR     LCD_RS
                  MOV     PO,#38H
                  ACALL  LCD_CLK
                  ACALL  DELAY_10ms

                  ACALL  LCD_OFF
                  ACALL  LCD_CLR
                  MOV     PO,#06H
                  ACALL  LCD_CLR
                  ACALL  LCD_HOME
                  ACALL  LCD_CLR
                  RET

;*****
;      LCD CLEAR DISPLAY
;*****

LCD_CLR:          MOV     PO,#01H
                  CLR     LCD_RS
                  CLR     LCD_RW
                  ACALL  LCD_CLK
                  RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*****
;   LCD RETURN HOME
;*****

LCD_HOME:          MOV    P0,#02H
                   CLR    LCD_RS
                   CLR    LCD_RW
                   ACALL  LCD_CLK
                   RET

;*****
;   LCD DISPLAY
;*****

LCD_OFF:           MOV    P0,#08H
                   CLR    LCD_RS
                   CLR    LCD_RW
                   ACALL  LCD_CLK
                   RET

;*****
;   LCD CLK
;*****

LCD_CLK:           CLR    LCD_RW
                   CLR    LCD_EN
                   SETB   LCD_EN
                   LCALL  LCD_DELAY
                   CLR    LCD_EN
                   LCALL  LCD_DELAY
                   RET

;*****
;   LCD DISPLAY ON
;*****

LCD_ON:            MOV    P0,#0CH
                   CLR    LCD_RS
                   CLR    LCD_RW
                   ACALL  LCD_CLK
                   RET

;*****
;   LCD CURSOR ON
;*****

LCD_BLINK:         MOV    P0,#0FH
                   CLR    LCD_RS
                   CLR    LCD_RW
                   ACALL  LCD_CLK
                   RET

;*****
;   Set LCD Adress
;*****
SET_ADDR_LCD:     MOV    A,LCD_ADDR
                   SETB  ACC.7
                   MOV    P0,A
                   CLR    LCD_RS
                   CLR    LCD_RW
                   LCALL  LCD_CLK
                   RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*****
;   BUFFER2ACC
;*****

BUFFER2ACC:      MOV    A,BUFFER
                  ANL    A,#00FH
                  SWAP   A
                  MOV    B,A
                  MOV    A,BUFFER+1
                  ANL    A,#00FH
                  ADD    A,B
                  RET

;*****
;   KPAD2LCD
;*****

KPAD2LCD:        MOV    A,KPAD_DATA
                  ADD    A,#030H
                  MOV    LCD_DATA,A
                  LCALL  WRCHAR_LCD
                  RET

;*****
;   WRLINE_LCD8
;*****
WRLINE_LCD8:     MOV    R0,#0
WRLINE_LCD_11:  CLR    A
                  MOVC   A,@A+DPTR
                  MOV    P0,A
                  SETB   LCD_RS
                  CLR    LCD_RW
                  LCALL  LCD_CLK
                  INC    DPTR
                  INC    R0
                  CJNE   R0,#8,WRLINE_LCD_11
                  LCALL  LCD_ON
                  RET

;*****
;   Write LINE 16 Character to show LCD
;*****
WRLINE_LCD:      MOV    R0,#0
WRLINE_LCD_1:   CLR    A
                  MOVC   A,@A+DPTR
                  MOV    P0,A
                  SETB   LCD_RS
                  CLR    LCD_RW
                  LCALL  LCD_CLK
                  INC    DPTR
                  INC    R0
                  CJNE   R0,#8,WRLINE_LCD_1

WRLINE_LCD_2:   CLR    A
                  MOVC   A,@A+DPTR
                  MOV    P0,A
                  SETB   LCD_RS
                  CLR    LCD_RW
                  LCALL  LCD_CLK
                  INC    DPTR
                  INC    R0
                  CJNE   R0,#16,WRLINE_LCD_2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                LCALL LCD_ON
                RET
;*****
;   WAIT KEY[check key]
;*****

WAIT_KEY:      JNB P2.3,WAIT_KEY
                RET

;*****
;   SCAN KEY
;*****

WAIT_KEYPRESSED:  MOV   A,P1
                  ANL   A,#00001111B
                  LJMP  TER_0

TER_0:          CJNE  A,#00000000B,TER_1
                  MOV   KPAD_DATA,#46H           ;FNC---->F
                  RET
TER_1:          CJNE  A,#00000001B,TER_2
                  MOV   KPAD_DATA,#43H           ;CLR---->C
                  RET
TER_2:          CJNE  A,#00000010B,TER_3
                  MOV   KPAD_DATA,#0             ;0
                  RET
TER_3:          CJNE  A,#00000011B,TER_4
                  MOV   KPAD_DATA,#59H           ;ENT---->Y
                  RET
TER_4:          CJNE  A,#00000100B,TER_5
                  MOV   KPAD_DATA,#64H           ;- ---->(d)
                  RET
TER_5:          CJNE  A,#00000101B,TER_6
                  MOV   KPAD_DATA,#1            ;1
                  RET
TER_6:          CJNE  A,#00000110B,TER_7
                  MOV   KPAD_DATA,#2            ;2
                  RET
TER_7:          CJNE  A,#00000111B,TER_8
                  MOV   KPAD_DATA,#3            ;3
                  RET
TER_8:          CJNE  A,#00001000B,TER_9
                  MOV   KPAD_DATA,#61H         ;+ ---->(a)
                  RET
TER_9:          CJNE  A,#00001001B,TER_10
                  MOV   KPAD_DATA,#4            ;4
                  RET
TER_10:         CJNE  A,#00001010B,TER_11
                  MOV   KPAD_DATA,#5            ;5
                  RET
TER_11:         CJNE  A,#00001011B,TER_12
                  MOV   KPAD_DATA,#6            ;6
                  RET
TER_12:         CJNE  A,#00001100B,TER_13
                  MOV   KPAD_DATA,#45H         ;ESC---->E
                  RET
TER_13:         CJNE  A,#00001101B,TER_14
                  MOV   KPAD_DATA,#7            ;7
                  RET
TER_14:         CJNE  A,#00001110B,TER_15
                  MOV   KPAD_DATA,#8            ;8

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                                RET
TER_15:                        MOV    KPAD_DATA,#9                ;9
                                RET
;*****
;    SHOW_TIME_DATE INTO MEMORY eeprom
;*****

SHOW_TIME_MEMO:                ACALL  DELAY_100ms
                                MOV    LCD_ADDR,#080H
                                ACALL  SET_ADDR_LCD

                                MOV    LCD_DATA,DATE_1
                                LCALL  WRCHAR_LCD

                                MOV    LCD_DATA,DATE_2
                                LCALL  WRCHAR_LCD
                                ACALL  DELAY_10ms

                                MOV    LCD_DATA,#'/'
                                LCALL  WRCHAR_LCD

                                MOV    LCD_DATA,MONTH_1
                                LCALL  WRCHAR_LCD
                                ACALL  DELAY_10ms
                                MOV    LCD_DATA,MONTH_2
                                LCALL  WRCHAR_LCD
                                ACALL  DELAY_10ms

                                MOV    LCD_DATA,#'/'
                                LCALL  WRCHAR_LCD

                                MOV    LCD_DATA,YEAR_1
                                LCALL  WRCHAR_LCD
                                ACALL  DELAY_10ms
                                MOV    LCD_DATA,YEAR_2
                                LCALL  WRCHAR_LCD
                                ACALL  DELAY_10ms

                                MOV    LCD_ADDR,#0C0H
                                ACALL  SET_ADDR_LCD
                                ACALL  DELAY_10ms

                                MOV    LCD_DATA,HOURS_1
                                LCALL  WRCHAR_LCD
                                ACALL  DELAY_10ms
                                MOV    LCD_DATA,HOURS_2
                                LCALL  WRCHAR_LCD
                                ACALL  DELAY_10ms

                                MOV    LCD_DATA,#':'
                                LCALL  WRCHAR_LCD
                                ACALL  DELAY_10ms

                                MOV    LCD_DATA,MINUTES_1
                                LCALL  WRCHAR_LCD
                                ACALL  DELAY_10ms
                                MOV    LCD_DATA,MINUTES_2
                                LCALL  WRCHAR_LCD
                                ACALL  DELAY_10ms

                                MOV    LCD_DATA,#':'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL WRCHAR_LCD
ACALL DELAY_10ms

MOV LCD_DATA,SECONDS_1
LCALL WRCHAR_LCD
ACALL DELAY_10ms
MOV LCD_DATA,SECONDS_2
LCALL WRCHAR_LCD
ACALL DELAY_10ms
RET

;*****
;---->>>SEND_PC
;*****
SEND_PC:      LCALL LCD_CLR
              MOV LCD_ADDR,#080H
              LCALL SET_ADDR_LCD
              MOV DPTR,#TITLE_1      ;--SEND & EXIT
              LCALL WRLINE_LCD
              ACALL WAIT_KEY
              ACALL DELAY_10ms
              LCALL WAIT_KEYPRESSED
              LCALL DELAY_10ms

              MOV A,KPAD_DATA
              CJNE A,#59H,NEX_T1      ;--ent
              SJMP START_SEND

NEX_T1:      CJNE A,#45H,SEND_PC      ;--esc
              LJMP MAIN_LOOP

;.....
START_SEND:  ACALL DELAY_100ms
              MOV SCON,#40H

              MOV A,#0AH
              LCALL SEND
              MOV A,#0DH
              LCALL SEND

              LCALL LOOP_1
              LCALL READ_EE

              MOV LCD_ADDR,#090H
              LCALL SET_ADDR_LCD
              MOV A,RR

              CJNE A,#7,TEST_600
              SJMP S_700

TEST_600:   CJNE A,#6,TEST_500
              SJMP S_600

TEST_500:   CJNE A,#5,TEST_400
              SJMP S_500

TEST_400:   CJNE A,#4,TEST_300
              SJMP S_400

TEST_300:   CJNE A,#3,TEST_200
              SJMP S_300

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกระใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TEST_200:      CJNE  A,#2,TEST_100
                SJMP  S_200

TEST_100:      SJMP  S_100

;.....

S_700:         MOV   LCD_DATA,M_6           ;--->> Show data
10000
                LCALL WRCHAR_LCD          ;--->> Show data
                MOV   A,M_6
                LCALL SEND

S_600:         MOV   LCD_DATA,M_5           ;--->> Show data 1000
                LCALL WRCHAR_LCD          ;--->> Show data
                MOV   A,M_5
                LCALL SEND

S_500:         MOV   LCD_DATA,M_4           ;--->> Show data 100
                LCALL WRCHAR_LCD          ;--->> Show data
                MOV   A,M_4
                LCALL SEND

S_400:         MOV   LCD_DATA,M_3           ;--->> Show data 10
                LCALL WRCHAR_LCD          ;--->> Show data
                MOV   A,M_3
                LCALL SEND

S_300:         MOV   LCD_DATA,M_2           ;--->> Show data 1
                LCALL WRCHAR_LCD          ;--->> Show data
                MOV   A,M_2
                LCALL SEND

S_200:         MOV   LCD_DATA,M_1           ;--->> Show data .
                LCALL WRCHAR_LCD          ;--->> Show data
                MOV   A,M_1
                LCALL SEND

S_100:         MOV   LCD_DATA,M_0           ;--->> Show data .1
                LCALL WRCHAR_LCD          ;--->> Show data
                MOV   A,M_0
                LCALL SEND

                MOV   LCD_ADDR,#098H
                LCALL SET_ADDR_LCD
                MOV   DPTR,#TITLE_9       ;--->>WATT
                LCALL WRLINE_LCD8
                LCALL DELAY_10ms
                MOV   DPTR,#TITLE_9
                ACALL TX_TEXT             ;---->>TO PC

                LCALL SHOW_TIME_MEMO
                MOV   A,DATE_1             ;---->>DATE
                LCALL SEND
                MOV   A,DATE_2
                LCALL SEND

                ACALL TX_LABEL            ;;;;;;;;;;;;; /

                MOV   A,MONTH_1           ;--->>MONTH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL SEND
MOV  A,MONTH_2
LCALL SEND

ACALL TX_LABEL          ;;;;;;;;;;;;;; /

MOV  A,YEAR_1          ;---->>YEAR
LCALL SEND
MOV  A,YEAR_2
LCALL SEND

MOV  A,#' '

LCALL SEND

MOV  A,HOURS_1        ;---->>HOURS
LCALL SEND
MOV  A,HOURS_2
LCALL SEND

ACALL TX_LABEL1       ;;;;;;;;;;;;;; :

MOV  A,MINUTES_1     ;---->>MINUTES
LCALL SEND
MOV  A,MINUTES_2
LCALL SEND

ACALL TX_LABEL1       ;;;;;;;;;;;;;; :

MOV  A,SECONDS_1     ;---->>SECONDS
LCALL SEND
MOV  A,SECONDS_2
LCALL SEND

MOV  A,#0AH
LCALL SEND
MOV  A,#0DH
LCALL SEND

LJMP SEND_PC
;*****
TX_TEXT:      CLR  TI
TX_LOOP:     CLR  A
             MOVC A,@A+DPTR
             INC  DPTR
             CJNE A,#00H,TX_CHAR
             RET
TX_CHAR:     MOV  SBUF,A
             JNB  TI,$
             CLR  TI
             ACALL DELAY_100ms
             SJMP TX_LOOP

;*****
TX_LABEL:     MOV  A,#'/'
             LCALL SEND
             RET

;*****
TX_LABEL1:    MOV  A,#':'
             LCALL SEND

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                                RET
;*****
;                                ALL DELAY
;*****
I2C_DELAY:                      MOV    R6,#0CH
I2C_DELAY_1:                    NOP
                                NOP
                                DJNZ  R6,I2C_DELAY_1
                                RET

LCD_DELAY:                      MOV    R7,#02
LCD_DELAY_1:                    MOV    R6,#0E6H
LCD_DELAY_2:                    NOP
                                NOP
                                DJNZ  R6,LCD_DELAY_2
                                DJNZ  R7,LCD_DELAY_1
                                RET

DELAY_10ms:                     MOV    R7,#010
DELAY_10ms_1:                   MOV    R6,#0E6H
DELAY_10ms_2:                   NOP
                                NOP
                                DJNZ  R6,DELAY_10ms_2
                                DJNZ  R7,DELAY_10ms_1
                                RET

DELAY_100ms:                    MOV    R7,#0100
DELAY_100ms_1:                  MOV    R6,#0E6H
DELAY_100ms_2:                  NOP
                                NOP
                                DJNZ  R6,DELAY_100ms_2
                                DJNZ  R7,DELAY_100ms_1
                                RET

DELAY_1:                         MOV    R5,#0100
DELAY_1s_1:                     LCALL DELAY_10ms
                                DJNZ  R5,DELAY_1s_1
                                RET

;*****
; Define Constant
;*****

TITLE_1:  DB    ' SEND or EXIT '
TITLE_2:  DB    ' can "t find '
TITLE_3:  DB    'Read[+] Write[-]'
TITLE_4:  DB    '2. Collec & Read'
TITLE_5:  DB    '3. Send to PC '
TITLE_6:  DB    ' ADDRESS '
TITLE_7:  DB    ' ERROR '
TITLE_8:  DB    '1. set Time&Date'
TITLE_9:  DB    ' W ',00H
TITLE_10: DB    'Time[+] Date[-]'
TITLE_11: DB    ' Clear '
TITLE_12: DB    ' EXIT Plese ESC '
SCR_SET_DATE: DB    ' Date : dd/mm/yy'
SCR_SET_TIME: DB    ' Time : hh/mm/SS'

```

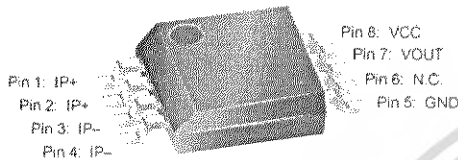
END

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ACS706ELC-20A

Bidirectional 1.5 mΩ Hall Effect Based Linear Current Sensor with Voltage Isolation and 20 A Dynamic Range

Package LC



Pins 6 and 7 are internally connected in shipping product. For compatibility with future devices, leave pin 6 floating.

Nominal Operating Temperature, T_A
 Range E..... -40 to 85°C
 Overcurrent Transient Tolerance*, I_p 60 A
 *100 total pulses, 250 ms duration each, applied at a rate of 1 pulse every 100 seconds.

ABSOLUTE MAXIMUM RATINGS

Supply Voltage, V_{CC} 16 V
 Reverse Supply Voltage, V_{RCC} -16 V
 Output Voltage, V_{OUT} 16 V
 Reverse Output Voltage, V_{ROUT} -0.1 V
 Output Current Source, $I_{OUT(SOURCE)}$ 3 mA
 Output Current Sink, $I_{OUT(SINK)}$ 10 mA
 Maximum Transient Sensed Current*, $I_{R(max)}$... 100 A
 Operating Temperature,
 Maximum Junction, $T_{J(max)}$ 165°C
 Storage Temperature, T_S -65 to 170°C
 *Junction Temperature, $T_J < T_{J(max)}$.



TUV America
 Certificate Number:
 U8V 04 12 54214 005

The Allegro ACS706 family of current sensors provides economical and precise solutions for current sensing in industrial, automotive, commercial, and communications systems. The device package allows for easy implementation by the customer. Typical applications include motor control, load detection and management, switched-mode power supplies, and overcurrent fault protection.

The device consists of a precision, low-offset linear Hall sensor circuit with a copper conduction path located near the surface of the die. Applied current flowing through this copper conduction path generates a magnetic field which is sensed by the integrated Hall IC and converted into a proportional voltage. Device accuracy is optimized through the close proximity of the magnetic signal to the Hall transducer. A precise, proportional voltage is provided by the low-offset, chopper-stabilized BiCMOS Hall IC, which is programmed for accuracy at the factory.

The output of the device has a positive slope ($>V_{CC}/2$) when an increasing current flows through the primary copper conduction path (from pins 1 and 2, to pins 3 and 4), which is the path used for current sensing. The internal resistance of this conductive path is typically 1.5 mΩ, providing low power loss. The thickness of the copper conductor allows survival of the device at up to 3× overcurrent conditions. The terminals of the conductive path are electrically isolated from the sensor leads (pins 5 through 8). This allows the ACS706 family of sensors to be used in applications requiring electrical isolation without the use of opto-isolators or other costly isolation techniques.

The ACS706 is provided in a small, surface mount SOIC8 package. The leadframe is plated with 100% matte tin, which is compatible with standard lead (Pb) free printed circuit board assembly processes. Internally, the flip-chip uses high-temperature Pb-based solder balls, currently exempt from RoHS. The device is fully calibrated prior to shipment from the factory.

Features and Benefits

- Small footprint, low-profile SOIC8 package
- 1.5 mΩ internal conductor resistance
- Excellent replacement for sense resistors
- 1600 V_{RMS} minimum isolation voltage between pins 1-4 and 5-8
- 4.5 to 5.5 V, single supply operation
- 50 kHz bandwidth
- 100 mV/A output sensitivity and 20 A dynamic range
- Output voltage proportional to ac and dc currents
- Factory-trimmed for accuracy
- Extremely stable output offset voltage
- Near-zero magnetic hysteresis
- Ratiometric output from supply voltage

Use the following complete part number when ordering:

Part Number	Package
ACS706ELC-20A	SOIC8 surface mount

ACS706ELC20A-DS, Rev. 2

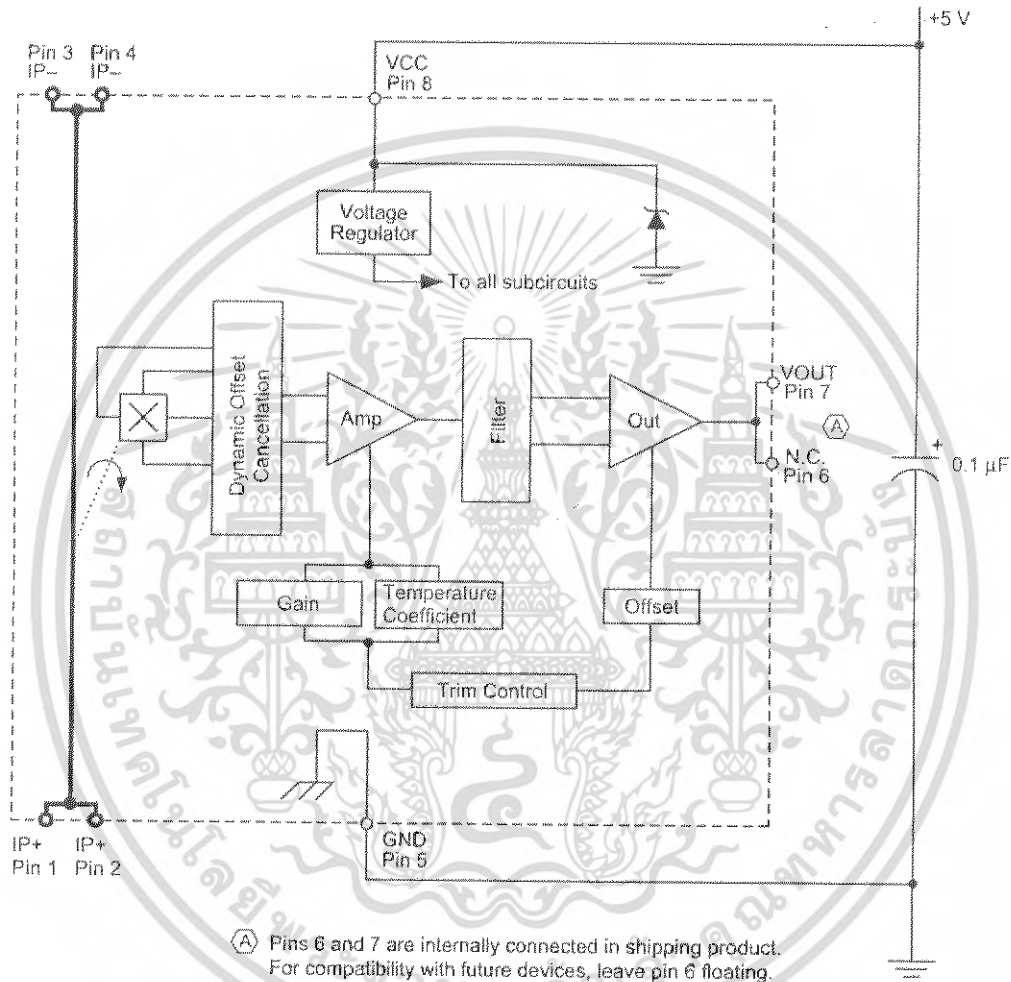


115 Northeast Corridor, Box 15036
 Worcester, Massachusetts 01615-0306 (508) 853-5000
www.allegromicro.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ACS706ELC-20A

Functional Block Diagram



ACS706ELC-20A

OPERATING CHARACTERISTICS

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
ELECTRICAL CHARACTERISTICS, over operating ambient temperature range unless otherwise specified						
Optimized Accuracy Range	I_P		-20	-	20	A
Linear Sensing Range	I_R		-20	-	20	A
Supply Voltage	V_{CC}		4.5	5.0	5.5	V
Supply Current	I_{CC}	$V_{CC} = 5.0$ V, output open	5	8	10	mA
Output Resistance	R_{OUT}	$I_{OUT} = 1.2$ mA	-	1	2	Ω
Output Capacitance Load	C_{LOAD}	VOUT to GND	-	-	10	nF
Output Resistive Load	R_{LOAD}	VOUT to GND	4.7	-	-	k Ω
Primary Conductor Resistance	$R_{PRIMARY}$	$T_A = 25^\circ\text{C}$	-	1.5	-	m Ω
RMS Isolation Voltage	V_{ISORMS}	Pins 1-4 and 5-8; 60 Hz, 1 minute	1600	2500	-	V
DC Isolation Voltage	V_{ISODC}		-	5000	-	V
PERFORMANCE CHARACTERISTICS, over operating ambient temperature range unless otherwise specified						
Propagation Time	t_{PROP}	$I_P = \pm 20$ A, $T_A = 25^\circ\text{C}$	-	3.15	-	μs
Response Time	$t_{RESPONSE}$	$I_P = \pm 20$ A, $T_A = 25^\circ\text{C}$	-	6	-	μs
Rise Time	t_r	$I_P = \pm 20$ A, $T_A = 25^\circ\text{C}$	-	6.56	-	μs
Frequency Bandwidth	f	-3 dB, $T_A = 25^\circ\text{C}$; I_P is 10 A peak-to-peak; no external filter	-	50	-	KHz
Sensitivity	Sens	Over full range of I_P , I_P applied for 5 ms; $T_A = 25^\circ\text{C}$	-	100	-	mV/A
		Over full range of I_P , I_P applied for 5 ms	94	-	106	mV/A
Noise	V_{NOISE}	Peak-to-peak, $T_A = 25^\circ\text{C}$, no external filter	-	70	-	mV
		Root Mean Square, $T_A = 25^\circ\text{C}$, no external filter	-	12.5	-	mV
Linearity	E_{LIN}	Over full range of I_P , I_P applied for 5 ms	-	± 1	± 3.5	%
Symmetry	E_{SYM}	Over full range of I_P , I_P applied for 5 ms	98	100	102	%
Zero Current Output Voltage	$V_{OUT(0)}$	$I_P = 0$ A, $T_A = 25^\circ\text{C}$	-	$V_{CC}/2$	-	V
Electrical Offset Voltage	V_{OE}	$I_P = 0$ A, $T_A = 25^\circ\text{C}$	-15	-	15	mV
		$I_P = 0$ A	-50	-	50	mV
Magnetic Offset Error	I_{ERROM}	$I_P = 0$ A, after excursion of 20 A	-	± 0.01	± 0.05	A
Total Output Error ¹	E_{TOT}	$I_P = \pm 20$ A, I_P applied for 5 ms; $T_A = 25^\circ\text{C}$	-	± 1.5	-	%
		$I_P = \pm 20$ A, I_P applied for 5 ms	-	-	± 8.4	%

THERMAL CHARACTERISTICS^{2,3}, $T_A = -40^\circ\text{C}$ to 125°C , $V_{CC} = 5$ V unless otherwise specified

			-	Value	-	Units
Junction-to-Lead Thermal Resistance	$R_{\theta JL}$	Mounted on the Allegro ASEQ 70x evaluation board; additional information about reference boards and tests is available on the Allegro Web site	-	5	-	$^\circ\text{C/W}$
Junction-to-Ambient Thermal Resistance	$R_{\theta JA}$	Mounted on the Allegro ASEQ 70x evaluation board; additional information about reference boards and tests is available on the Allegro Web site	-	41	-	$^\circ\text{C/W}$

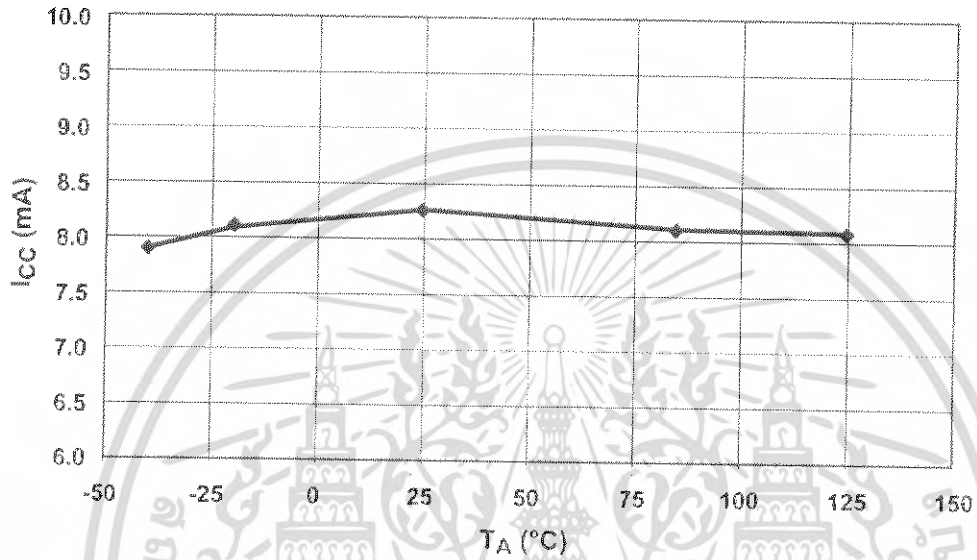
¹Percentage of I_P with $I_P = 20$ A. Output filtered. Up to a 2.0% shift in E_{TOT} may be observed at end-of-life for this device.

²The Allegro evaluation board has 1500 mm² of 2 oz. copper on each side, connected to pins 1 and 2, and to pins 3 and 4, with thermal vias connecting the layers. Performance values include the power consumed by the PWB. Further details on the board are available from the ACS704 Frequently Asked Questions document on our website. Further information about board design and thermal performance also can be found on pages 16 and 17 of this datasheet.

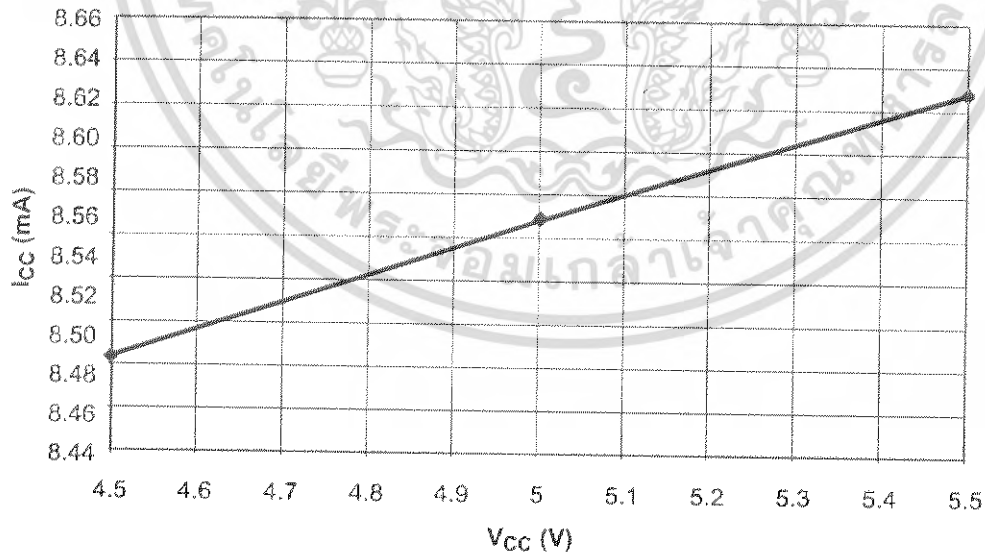
³ $R_{\theta JA}$ values shown in this table are typical values, measured on the Allegro evaluation board. The actual thermal performance depends on the board design, the airflow in the system, and thermal interactions between the sensor and surrounding components through the PCB and the ambient air. To improve thermal performance, see our applications material on the Allegro Web site.

ACS706ELC-20A

Typical Performance Characteristics Supply Current versus Ambient Temperature $V_{CC} = 5\text{ V}$

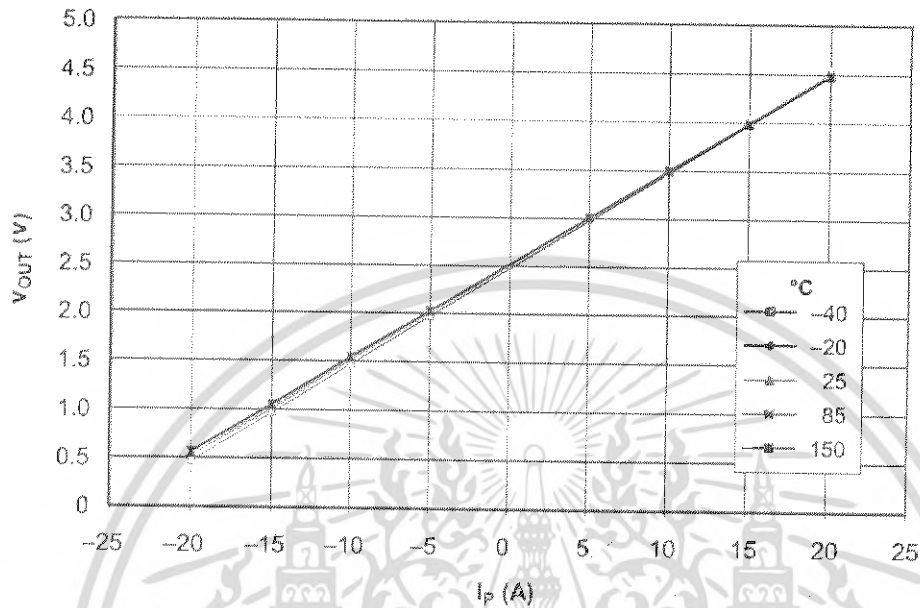


Supply Current versus Applied V_{CC}

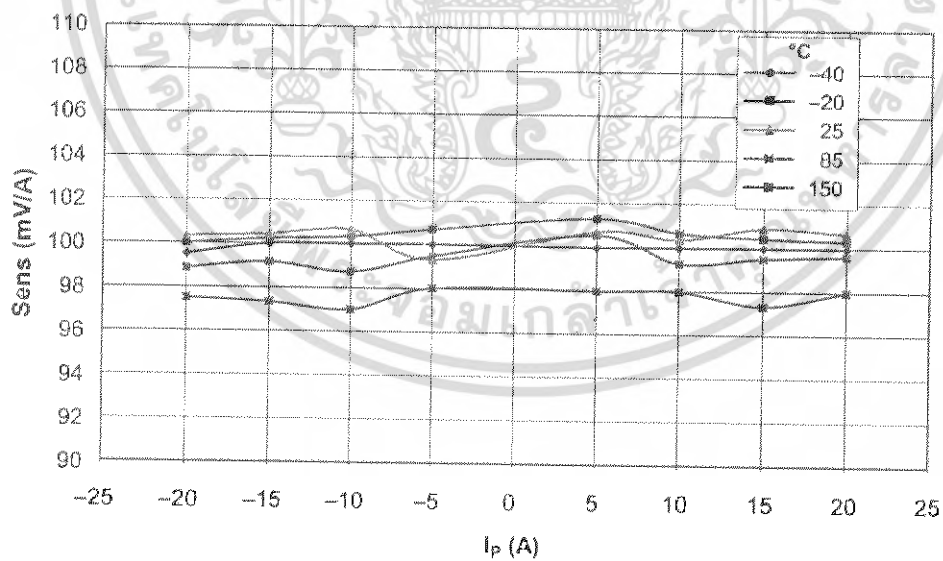


ACS706ELC-20A

Output Voltage versus Primary Current
 $V_{CC} = 5\text{ V}$



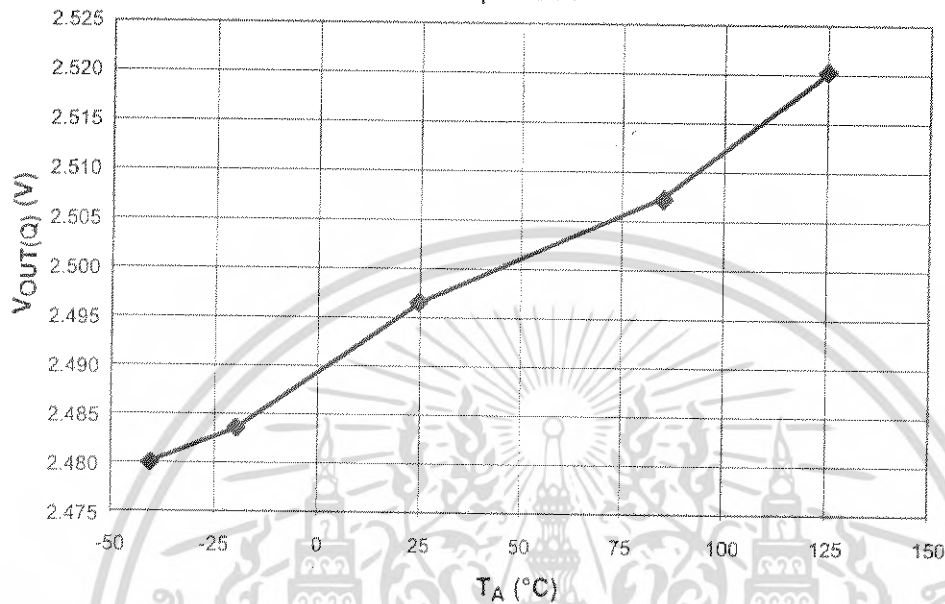
Sensitivity versus Primary Current
 $V_{CC} = 5\text{ V}$



ACS706ELC-20A

Zero Current Output Voltage vs. Ambient Temperature

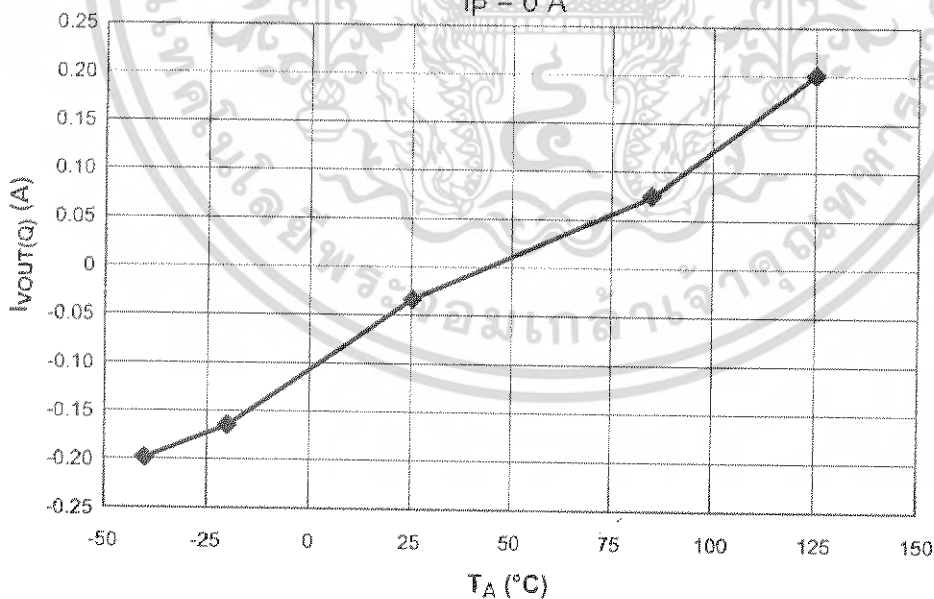
$I_p = 0\text{ A}$



Zero Current Output Current versus Ambient Temperature

(Data in above chart converted to amperes)

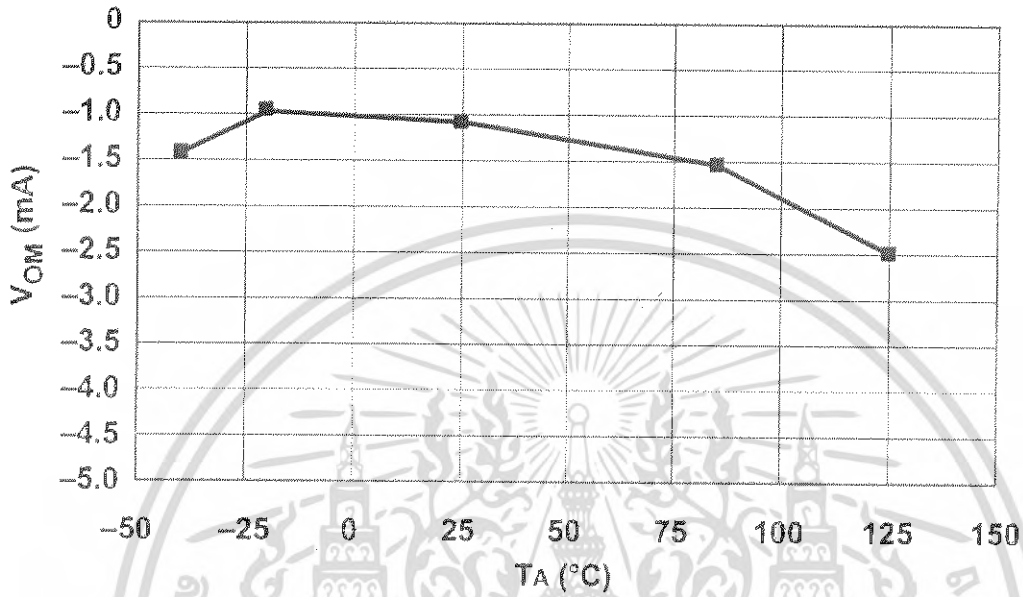
$I_p = 0\text{ A}$



ACS706ELC-20A

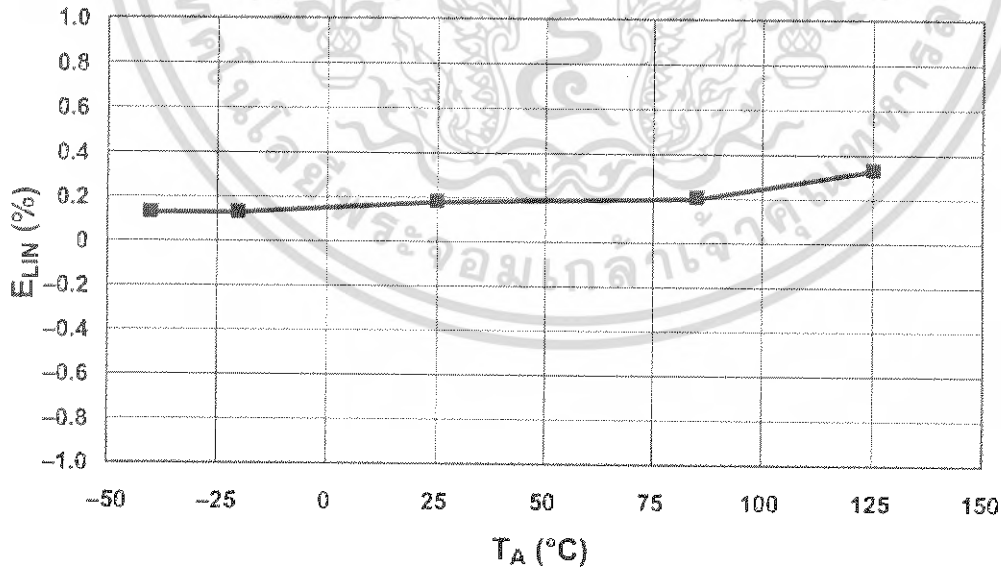
Magnetic Offset versus Ambient Temperature

$V_{CC} = 5\text{ V}$; $I_P = 0\text{ A}$, after excursion to 20 A



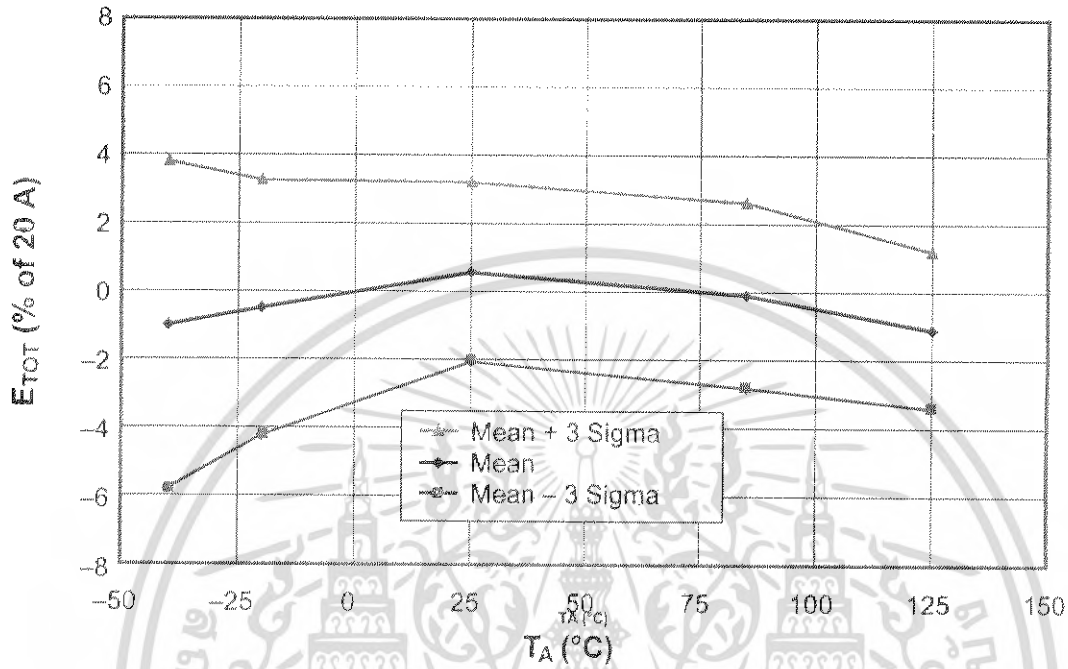
Nonlinearity versus Ambient Temperature

$V_{CC} = 5\text{ V}$ $I_P = 20\text{ A}$



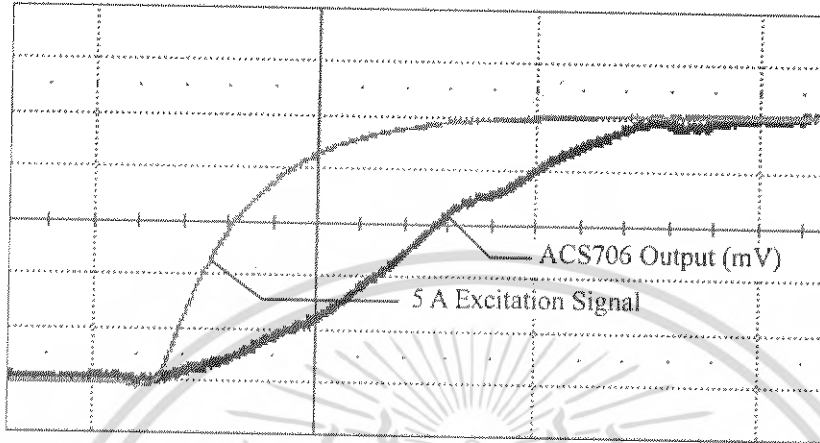
ACS706ELC-20A

Typical Percentage Error versus Ambient Temperature
Measurements at $T_A = -40, -20, 25, 85, \text{ and } 125^\circ\text{C}$



ACS706ELC-20A

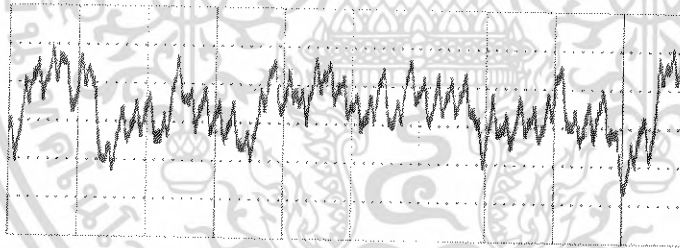
Step Response of ACS706ELC-20A at $T_A=25^\circ\text{C}$



Measure	P2.pkpk(C3)	P5.rise(C3)
value	534 mV	7.23652 μs
mean	571.25 mV	7.1411227 μs
min	534 mV	5.43890 μs
max	619 mV	10.66561 μs

Time = 5 $\mu\text{s}/\text{div}$.
Excitation signal = 1.00 A/div.
Output = 100 mV/div.

Typical Peak-to-Peak Noise of ACS706ELC-20A at $T_A=25^\circ\text{C}$



Measure	P2.pkpk(C3)
value	91.9 mV
mean	91.159 mV
min	68.1 mV
max	123.1 mV

Time = 20 $\mu\text{s}/\text{div}$.
Noise = 20.0 mV/div.

ACS706ELC-20A

ACS706ELC-20A Noise Filtering and Frequency Response Performance

Break Frequency of Filter on Output (kHz)	Resistance (kΩ)	Capacitance (μF)	Nominal Programmed Sensitivity (mV/A)	Filtered Peak-to-Peak Noise (mV)	Resolution with Filtering (A)	Rise Time for 5A Step, Filtered (μs)
Unfiltered	—	—	—	70.0	0.700	6.56
80	0.200	0.01	100	58.8	0.588	7.82
50	0.320			49.9	0.499	9.55
40	0.392			46.3	0.463	10.25
20	0.800			32.9	0.329	16.15
10	1.6			21.9	0.219	30.14
7.0	3.15			13.3	0.133	53.29
3.3	4.8			9.8	0.098	79.73
0.6	26			1.3	0.013	394.66
0.3	53			0.58	0.00583	724.73

Definitions of Accuracy Characteristics

Sensitivity (Sens). The change in sensor output in response to a 1 A change through the primary conductor. The sensitivity is the product of the magnetic circuit sensitivity (G/A) and the linear IC amplifier gain (mV/G). The linear IC amplifier gain is programmed at the factory to optimize the sensitivity (mV/A) for the full-scale current of the device.

Noise (V_{NOISE}). The product of the linear IC amplifier gain (mV/G) and the noise floor for the Allegro Hall effect linear IC (≈ 1 G). The noise floor is derived from the thermal and shot noise observed in Hall elements. Dividing the noise (mV) by the sensitivity (mV/A) provides the smallest current that the device is able to resolve.

Linearity (E_{LIN}). The degree to which the voltage output from the sensor varies in direct proportion to the primary current through its full-scale amplitude. Nonlinearity in the output can be attributed to the saturation of the flux concentrator approaching the full-scale current. The following equation is used to derive the linearity:

$$100 \left\{ 1 - \left[\frac{(V_{\text{out_full-scale amperes}} - V_{\text{OUT(Q)}})}{2(V_{\text{out_half-scale amperes}} - V_{\text{OUT(Q)}})} \right] \right\}$$

where $V_{\text{out_full-scale amperes}}$ = the output voltage (V) when the sensed current approximates full-scale $\pm I_p$.

Symmetry (E_{SYM}). The degree to which the absolute voltage output from the sensor varies in proportion to either a positive or negative full-scale primary current. The following formula is used to derive symmetry:

$$100 \left(\frac{V_{\text{out_+full-scale amperes}} - V_{\text{OUT(Q)}}}{V_{\text{OUT(Q)}} - V_{\text{out_full-scale amperes}}} \right)$$

Quiescent output voltage (V_{OUT(Q)}). The output of the sensor when the primary current is zero. For a unipolar supply voltage, it nominally remains at $V_{CC}/2$. Thus, $V_{CC} = 5$ V translates into $V_{\text{OUT(Q)}} = 2.5$ V. Variation in $V_{\text{OUT(Q)}}$ can be attributed to the resolution of the Allegro linear IC quiescent voltage trim and thermal drift.

Electrical offset voltage (V_{OE}). The deviation of the device output from its ideal quiescent value of $V_{CC}/2$ due to nonmagnetic causes. To convert this voltage to amperes, divide by the device sensitivity, Sens.

Accuracy (E_{TOT}). The accuracy represents the maximum deviation of the actual output from its ideal value. This is also known as the total output error. The accuracy is illustrated graphically in the Output Voltage versus Current chart on the following page.

Accuracy is divided into four areas:

- **0 A at 25°C.** Accuracy of sensing zero current flow at 25°C, without the effects of temperature.
- **0 A over Δ temperature.** Accuracy of sensing zero current flow including temperature effects.
- **Full-scale current at 25°C.** Accuracy of sensing the full-scale current at 25°C, without the effects of temperature.
- **Full-scale current over Δ temperature.** Accuracy of sensing full-scale current flow including temperature effects.

Ratiometry. The ratiometric feature means that its 0 A output, $V_{\text{OUT(Q)}}$, (nominally equal to $V_{CC}/2$) and sensitivity, Sens, are proportional to its supply voltage, V_{CC} . The following formula is used to derive the ratiometric change in 0 A output voltage, $\Delta V_{\text{OUT(Q)RAT}}$ (%):

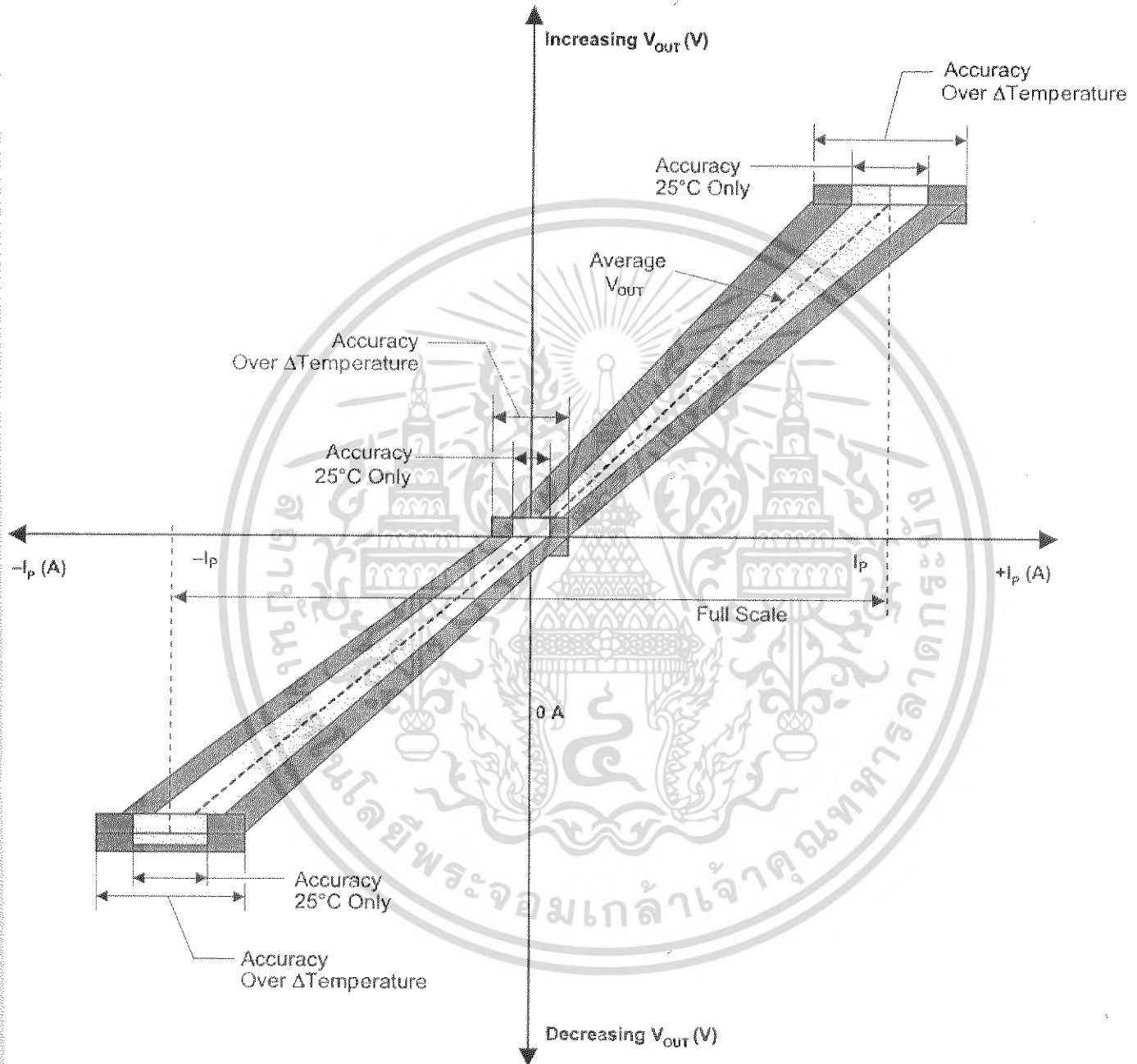
$$100 \left(\frac{V_{\text{OUT(Q) } V_{CC}} / V_{\text{OUT(Q) } 5V}}{V_{CC} / 5 V} \right)$$

The ratiometric change in sensitivity, $\Delta \text{Sens}_{\text{RAT}}$ (%), is defined as:

$$100 \left(\frac{\text{Sens}_{V_{CC}} / \text{Sens}_{5V}}{V_{CC} / 5 V} \right)$$

ACS706ELC-20A

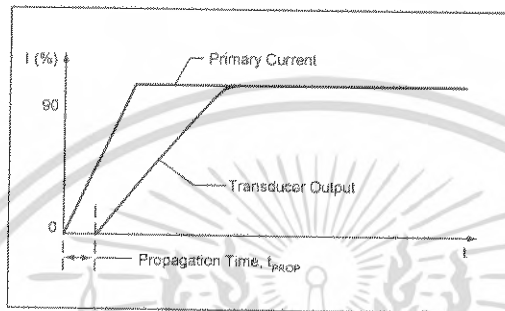
Output voltage vs. current, illustrating sensor accuracy at 0 A and at full-scale current



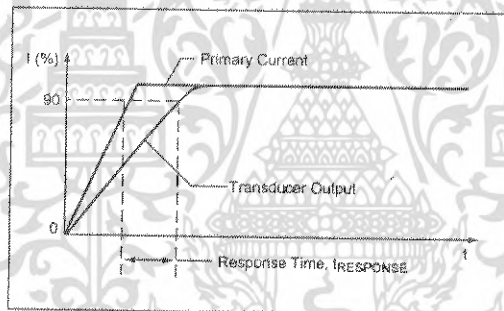
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Definitions of Dynamic Response Characteristics

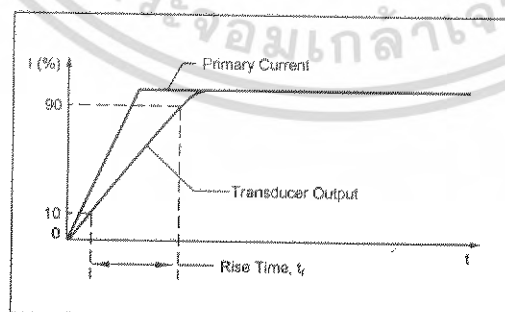
Propagation delay (t_{PROP}): The time required for the sensor output to reflect a change in the primary current signal. Propagation delay is attributed to inductive loading within the linear IC package, as well as in the inductive loop formed by the primary conductor geometry. Propagation delay can be considered as a fixed time offset and may be compensated.



Response time ($t_{RESPONSE}$): The time interval between a) when the primary current signal reaches 90% of its final value, and b) when the sensor reaches 90% of its output corresponding to the applied current.



Rise time (t_r): The time interval between a) when the sensor reaches 10% of its full scale value, and b) when it reaches 90% of its full scale value. The rise time to a step response is used to derive the bandwidth of the current sensor, in which $f(-3 \text{ dB}) = 0.35/t_r$. Both t_r and $t_{RESPONSE}$ are detrimentally affected by eddy current losses observed in the conductive IC ground plane.



ACS706ELC-20A

Standards and Physical Specifications

Parameter	Specification
Flammability (package molding compound)	UL recognized to UL 94V-0
Fire and Electric Shock	UL60950-1:2003 EN60950-1:2001 CAN/CSA C22.2 No. 60950-1:2003

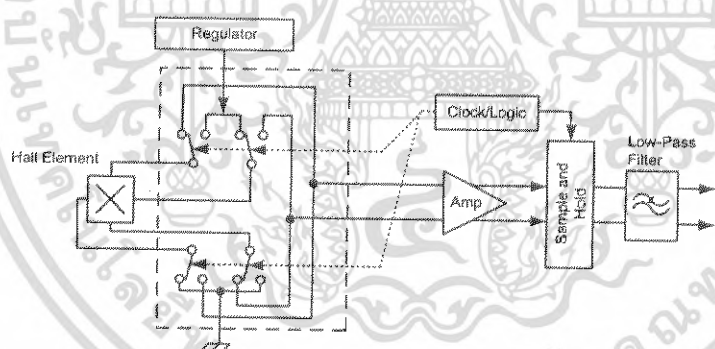
Device Branding Key (Two alternative styles are used)

ACS706T ELC20A YYWWA	ACS	Allegro Current Sensor
	704	Device family number
	T	Indicator of 100% matte tin leadframe plating
	E	Operating ambient temperature range code
	LC	Package type designator
	20A	Primary sensed current
	YY	Manufacturing date code: Calendar year (last two digits)
	WW	Manufacturing date code: Calendar week
	A	Manufacturing date code: Shift code
ACS706T ELC20A L...L YYWW	ACS	Allegro Current Sensor
	704	Device family number
	T	Indicator of 100% matte tin leadframe plating
	E	Operating ambient temperature range code
	LC	Package type designator
	20A	Primary sensed current
	L...L	Manufacturing lot code
	YY	Manufacturing date code: Calendar year (last two digits)
	WW	Manufacturing date code: Calendar week

Chopper Stabilization Technique

Chopper Stabilization is an innovative circuit technique that is used to minimize the offset voltage of a Hall element and an associated on-chip amplifier. Allegro patented a Chopper Stabilization technique that nearly eliminates Hall IC output drift induced by temperature or package stress effects. This offset reduction technique is based on a signal modulation-demodulation process. Modulation is used to separate the undesired dc offset signal from the magnetically induced signal in the frequency domain. Then, using a low-pass filter, the modulated dc offset is suppressed while the magnetically induced signal passes through the filter. As a result of this chopper stabilization approach, the output voltage from the Hall IC is desensitized to the effects of temperature and mechanical stress. This technique produces devices that have an extremely stable Electrical Offset Voltage, are immune to thermal stress, and have precise recoverability after temperature cycling.

This technique is made possible through the use of a BiCMOS process that allows the use of low-offset and low-noise amplifiers in combination with high-density logic integration and sample and hold circuits.

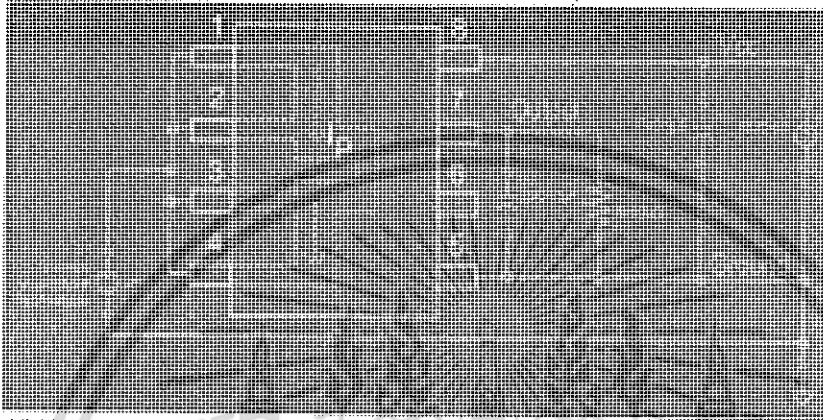


Concept of Chopper Stabilization Technique

Applications Information

Transient Common-Mode Voltage Rejection in the ACS706

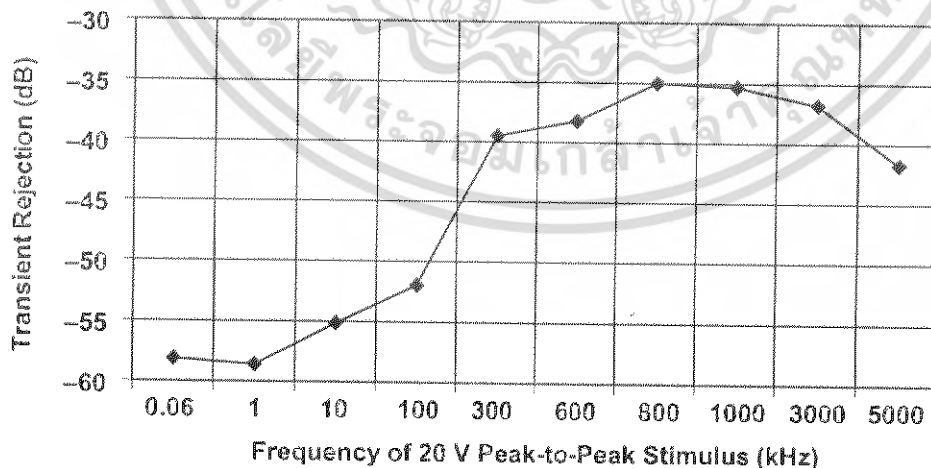
In order to quantify transient common-mode voltage rejection for the ACS706, a device was soldered onto a printed circuit board. A 0.1 μF bypass capacitor and a 5 V dc power supply were connected between VCC and GND (pins 8 and 5) for this device. A 10 k Ω load resistor and a 0.01 μF capacitor were connected in parallel between the VOUT pin and the GND pin of the device (pins 7 and 5).



ACS706 Schematic Diagram of the Circuit used to Measure Transient Rejection

A function generator was connected between the primary current conductor (pins 1 thru 4) and the GND pin of the device (pin 5). This function generator was configured to generate a 10 V peak (20 V peak-to-peak) sine wave between pins 1-4 and pin 5. Note that the sinusoidal stimulus was applied such that no electrical current would flow through the copper conductor composed of pins 1-4 of this device.

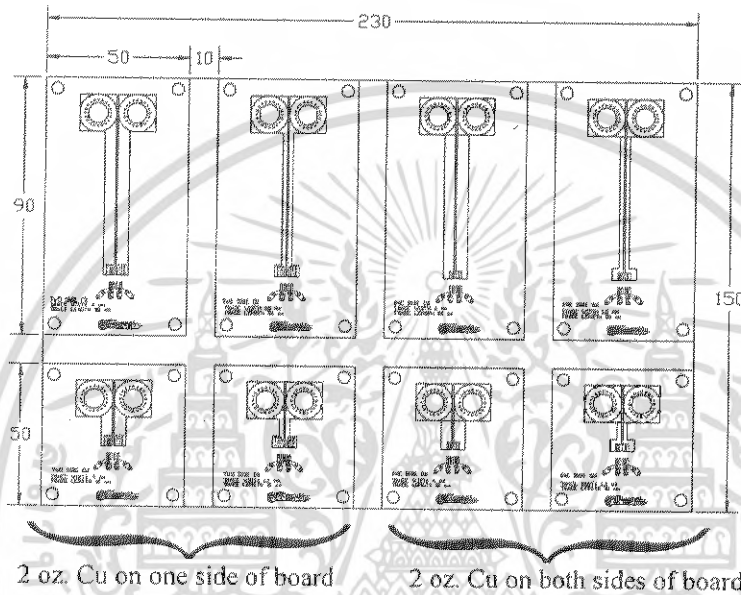
The frequency of this sine wave was varied from 60 Hz to 5 MHz in discrete steps. At each frequency, the statistics feature of an oscilloscope was used to measure the voltage variations (noise) on the ACS706 output in mV (peak to peak). The noise was measured both before and after the application of the stimulus. Transient common-mode voltage rejection as a function of frequency is shown in the following figure.



ACS706ELC-20A

The Effect of PCB Layout on ACS706 Thermal Performance

Eight different PC boards were fabricated to characterize the effect of PCB design on the operating junction temperature of the Hall-effect IC inside of the ACS706. These PC boards are shown in the figure below.



An ACS706 device was soldered on to each PCB for thermal testing. The results of the testing are shown in the following table.

Test Results on Eight Thermal Characterization PCBs

Tested at 15A, $T_A = 20^\circ\text{C}$, still air, 2 oz. copper traces, current carried on and off board by 14 gauge wires

PC Boards Sides with Traces	Trace Width (mm)	Trace Length (mm)	Temperature Rise Above Ambient ($^\circ\text{C}$)
1	4	50	90
	1.5	50	Overheated
	4	10	48
	1.5	10	110
2	4	50	53
	1.5	50	106
	4	10	38
	1.5	10	54

ACS706ELC-20A

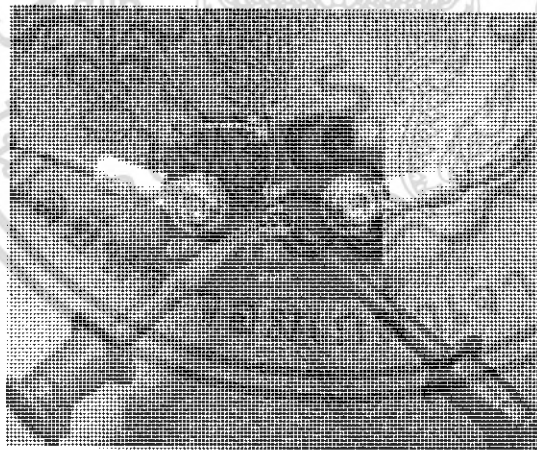
Improved PC Board Designs

The eight PC boards in the figure above do not represent an ideal PC board for use with the ACS706. The ACS706 evaluation boards, for sale at the Allegro Web site On-Line Store, represent a more optimal PC board design (see photo below). On the evaluation boards, the current to be sensed flows through very wide traces that were fabricated using 2 layers of 2 oz. copper. Thermal management tests were conducted on the Allegro evaluation boards and all tests were performed using the same test conditions described in the bulleted list above. The results for these thermal tests are shown in the table below. When using the Allegro evaluation boards we see that even at an applied current of 20 A the junction temperature of the ACS706 is only ≈ 30 degrees above ambient temperature.

Test Results on Eight Electrical Characterization PCBs

Tested at $T_A = 20^\circ\text{C}$, still air

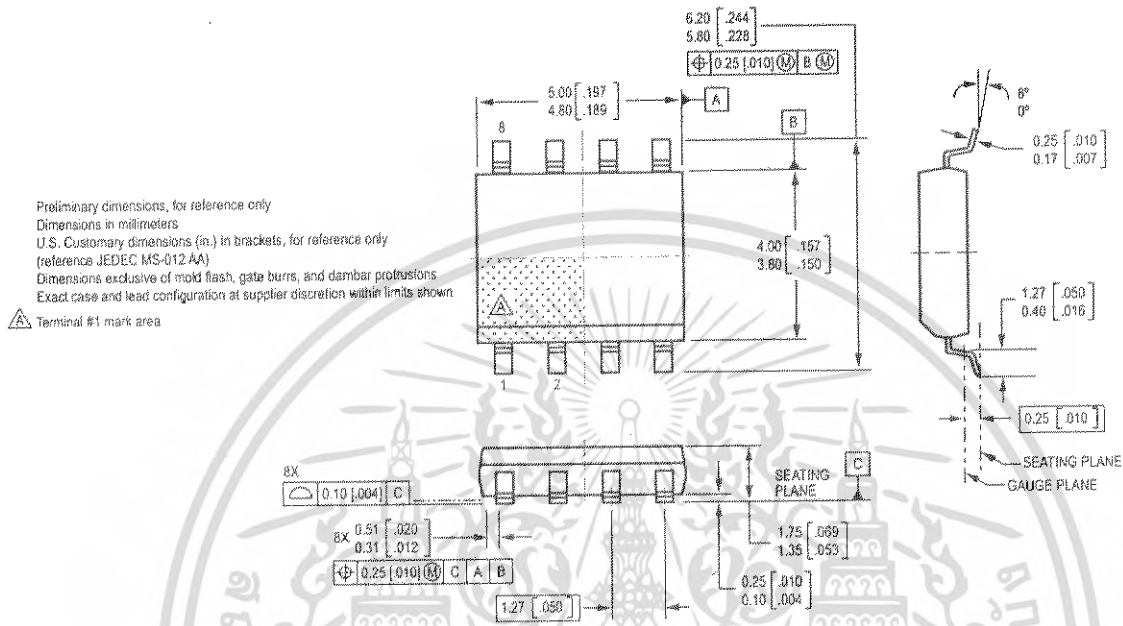
Applied Current (A)	Temp Rise Above Ambient ($^\circ\text{C}$)
15	22
20	31



Allegro Current sensor evaluation board with ACS706 and external connections.

ACS706ELC-20A

Package LC, 8-pin SOIC



The products described herein are manufactured under one or more of the following U.S. patents: 5,045,920; 5,264,783; 5,442,283; 5,389,889; 5,581,179; 5,517,112; 5,619,137; 5,621,319; 5,650,719; 5,686,894; 5,694,038; 5,729,130; 5,917,320; and other patents pending.

Allegro MicroSystems, Inc. reserves the right to make, from time to time, such departures from the detail specifications as may be required to permit improvements in the performance, reliability, or manufacturability of its products. Before placing an order, the user is cautioned to verify that the information being relied upon is current.

Allegro products are not authorized for use as critical components in life-support devices or systems without express written approval.

The information included herein is believed to be accurate and reliable. However, Allegro MicroSystems, Inc. assumes no responsibility for its use; nor for any infringement of patents or other rights of third parties which may result from its use.

Copyright©2005, Allegro MicroSystems, Inc.

ACS706ELC20A-DS, Rev. 2



115 Northwest Cutoff, Box 15036
 Worcester, Massachusetts 01615-0036 (508) 853-5000
 www.allegromicro.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้