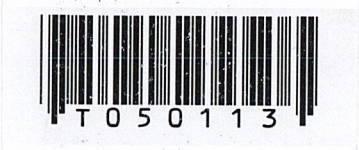


# อุปกรณ์ระบุตำแหน่งวัตถุโดยใช้คลื่นอัลตราโซนิก

## ULTRASONIC OBJECT POSITIONING



โดย

นายกิตติชัย	สุรพัทธ์ภิญโญ
นายชาญชัย	เสรีพรเจริญกุล
นายณัฐพล	สืบอ่ำ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

พ.ศ.

๒๕๕๐

๒๕๕

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....

เลขทะเบียน...50113...

วัน,เดือน,ปี. 2 1 เม.ย. 2547

b.....
i.....

๒๕๕๐๒๕๖

อุปกรณ์ระบุตำแหน่งวัตถุโดยใช้คลื่นอัลตราโซนิก

ULTRASONIC OBJECT POSITIONING

โดย

นายกิตติชัย สุรพัทตร์ภิญโญ 42010020

นายชาญชัย เสรีพรเจริญกุล 42010081

นายณัฐพล สืบอ่ำ 42010106

อาจารย์ที่ปรึกษา

อ.สมเกียรติ ฤกษ์วัลญญู

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชาวิศวกรรมโทรคมนาคม

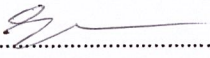
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง อุปกรณ์ระบุตำแหน่งด้วยคลื่นอัลตราโซนิก

ULTRASONIC OBJECT POSITIONING

ผู้จัดทำ

1. นายกิตติชัย สุรพัทศรีภิญโญ 42010020
2. นายชาญชัย เสรีพรเจริญกุล 42010081
3. นายณัฐพล สืบอ่ำ 42010106

.....  อาจารย์ที่ปรึกษา

( อ.สมเกียรติ ฤกษ์วีระบุญ )

# อุปกรณ์ระบุตำแหน่งวัตถุโดยใช้คลื่นอัลตราโซนิก

## Ultrasonic Object Positioning

โดย นาย กิตติชัย สุรพักตร์ภิญโญ 42010020

นาย ชาณชัย เสรีพรเจริญกุล 42010081

นาย ณัฐพล สืบอ่ำ 42010106

อาจารย์ที่ปรึกษา อ.สมเกียรติ ฤกษ์วีระบุญ

### บทคัดย่อ

ปริญญานิพนธ์นี้เป็นการศึกษาและนำคลื่นอัลตราโซนิกมาประยุกต์ใช้ในการระบุตำแหน่งของวัตถุหรือสิ่งกีดขวางในระนาบแนวนอน แล้วนำมาแสดงผลตำแหน่งเป็นพิกัดของระยะทางและองศา ซึ่งปริญญานิพนธ์นี้สามารถนำไปเป็นแนวทางในการพัฒนาอุปกรณ์ต่างๆ ที่ต้องใช้ความสามารถในการระบุตำแหน่งของวัตถุเช่นตาของหุ่นยนต์ เป็นต้น

### Abstract

This thesis is to study about ultrasonic and apply it to position object or obstacle in azimuth plane. Then, display in form of distance and degree coordinate. In further application, this thesis could be used to develop various equipments such as robot's eye, which positioning ability is necessary.

## สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 ทฤษฎีคุณสมบัติของคลื่นอัลตราโซนิก	2
2.1.1 คุณสมบัติของคลื่นอัลตราโซนิก	2
2.1.2 การทำงานของทรานสดิวเซอร์ (Transducer)	2
2.2 ทฤษฎีและหลักการของวงจร โดยรวม	3
2.3 ทฤษฎีและหลักการขององค์ประกอบย่อย	8
2.3.1 วงจรกำเนิดสัญญาณความถี่ 40 กิโลเฮิร์ตซ์ (Oscillator)	8
2.3.2 วงจรเข้ารหัสแบบชิฟต์รีจิสเตอร์ (Shift Register)	10
2.3.3 วงจรคูณสัญญาณ แบบกิลเบิร์ต (Gilbert Multiplier Circuit)	12
2.3.4 วงจรขยาย (Amplifier)	15
2.3.5 วงจรอินเวอร์ตติ้ง แอมป์ลิไฟเออร์ (Non – Inverting Amplifier)	17
2.3.6 สเตปปิ้งมอเตอร์ (Stepping Motor)	18
2.3.7 วงจรพรีแอมป์ (Pre-Amp) ภาครับ	22
2.3.8 วงจรแบนพาสฟิลเตอร์ (Band Pass Filter)	24
2.3.9 Automatic Gain Control (AGC)	27
2.3.10 วงจรเอนเวลโลป ดีเทคเตอร์ (Envelope detector)	28
2.3.11 วงจรกรองความถี่ต่ำผ่าน (Low Pass Filter, LPF)	28
2.3.12 การแปลงสัญญาณอนาลอกเป็นดิจิตอล (Analog to Digital Converter)	30
2.3.13 ไมโครคอนโทรลเลอร์ (Microcontroller)	32
2.3.14 พอร์ทขนาน (Parallel Port)	35
2.3.15 การเขียนโปรแกรมด้วยภาษา Visual Basic 6.0 (VB)	38
2.3.16 การเขียนโปรแกรมด้วยภาษา Visual C++ 6.0 (VC++)	39
บทที่ 3 การคำนวณและสร้าง	42
3.1 วงจรภาคส่ง	42
3.1.1 วงจรกำเนิดความถี่ (Oscillator)	42
3.1.2 วงจรเข้ารหัสแบบชิฟต์รีจิสเตอร์ (Shift Register Coder)	43
3.1.3 วงจรคูณสัญญาณ	44
3.1.4 วงจรอินเวอร์ตติ้ง แอมป์ลิไฟเออร์ (Inverting Amplifier)	44
3.1.5 วงจรลดทอนสัญญาณ (Attenuater)	45
3.1.6 วงจรบัฟเฟอร์ (Buffer)	46

## สารบัญ (ต่อ)

เรื่อง	หน้า
3.1.7 วงจรแปลงสัญญาณไฟ (Regulator)	46
3.1.8 วงจรขับสเตปป์จัมเปอร์	47
3.2 วงจรส่วนภาครับ	49
3.2.1 การสร้างวงจรพรีแอมป์ภาครับ (Pre-Amplifier)	49
3.2.2 การสร้างวงจร Bandpass Filter	49
3.2.3 การสร้างวงจร Automatic Gain Control (AGC)	54
3.2.4 การสร้างวงจรเอนเวลโลปดีเทคเตอร์ (Envelope detector)	54
3.2.5 วงจรกรองความถี่ต่ำผ่าน (Low Pass Filter, LPF)	55
3.2.6 การควบคุมด้วยไมโครคอนโทรลเลอร์ MCS-51	56
3.2.7 การเขียนโปรแกรมการทำงานในภาคแสดงผล	58
3.2.8 การต่อวงจรรวม	59
บทที่ 4 การทดลองและผลการทดลอง	62
4.1 การทดลองและผลการทดลอง	62
4.1.1 ทรานซิสเตอร์ ตัวส่ง และ ตัวรับ	62
4.1.2 การทดลองตัวแปลงระดับไฟโวลต์เตจเรกกูเรเตอร์	64
4.1.3 วงจรออสซิลเลเตอร์ของภาคส่ง	67
4.1.4 วงจรเข้ารหัสแบบดิจิทัลรีจิสเตอร์	68
4.1.5 วงจรคูณสัญญาณ	69
4.1.6 วงจรพรีแอมป์	71
4.1.7 วงจรแบนพาสฟิลเตอร์	73
4.1.8 วงจร AGC	75
4.1.9 วงจรเอนเวลโลปดีเทคเตอร์	77
4.1.10 การรับส่งข้อมูลระหว่าง MCS-51 กับคอมพิวเตอร์ ทาง Parallel Port	79
บทที่ 5 บทวิจารณ์และบทสรุป	81
ภาคผนวก	
Pseudo Code	
ภาพแสดงความสำเร็จของชิ้นงาน	
ภาพแสดงลายปริ้นของชิ้นงาน	
Data sheet	
หนังสืออ้างอิง	

## สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 ลักษณะของลำคลื่นในการส่ง (Beam Angle) ของทรานสดิวเซอร์	3
รูปที่ 2.2 แสดงการส่งคลื่นอัลตราโซนิกไปกระทบสิ่งกีดขวาง และสะท้อนกลับมายังเครื่องรับ	3
รูปที่ 2.3 รูปแสดงการจัดเรียงทรานสดิวเซอร์ตัวรับ	4
รูปที่ 2.4 รูปแสดงวิธีการคำนวณมุมและระยะห่างของวัตถุกับอุปกรณ์	4
รูปที่ 2.5 รูปแสดงบล็อกไดอะแกรมภาคส่งของอุปกรณ์ระบุตำแหน่งวัตถุโดยคลื่นอัลตราโซนิก	6
รูปที่ 2.6 รูปแสดงบล็อกไดอะแกรมภาครับของอุปกรณ์ระบุตำแหน่งวัตถุโดยคลื่นอัลตราโซนิก	6
รูปที่ 2.7 รูปแสดงตัวอย่างการทำงานของ โปรแกรมการแสดงผล	7
รูปที่ 2.8 รูปแสดงวงจรกำเนิดสัญญาณความถี่ 40 กิโลเฮิร์ตซ์	8
รูปที่ 2.9 รูปแสดงวงจรภายในของวงจรซีพียูเตอร์	10
รูปที่ 2.10 รูปแสดงการทำงานของวงจรซีพียูเตอร์	11
รูปที่ 2.11 รูปแสดงวงจรภายในและการทำงานของวงจรคุณสมบัติแบบกิลเบิร์ต	12
รูปที่ 2.12 รูปแสดงการทำงานของวงจรขยายแบบอินเวอร์ตติ้ง แอมพลิฟายเออร์	15
รูปที่ 2.13 รูปแสดงวงจรภายในของวงจรขยายแบบอินเวอร์ตติ้ง แอมพลิฟายเออร์	15
รูปที่ 2.14 รูปแสดงการทำงานของวงจรขยายแบบนอน-อินเวอร์ตติ้ง แอมพลิฟายเออร์	17
รูปที่ 2.15 รูปวงจรบัฟเฟอร์ที่ได้จากประยุกต์จากนอน-อินเวอร์ตติ้ง แอมพลิฟายเออร์	17
รูปที่ 2.16 โครงสร้างภายในสเตปปีงมอเตอร์	18
รูปที่ 2.17 วงจรขับรีเลย์โดยใช้ทรานซิสเตอร์	21
รูปที่ 2.18 แสดงการต่อวงจรรีเลย์	22
รูปที่ 2.19 แสดงสัญญาณรูปไซน์ที่ถูกตัดออกไปทำให้มีลักษณะคล้ายสัญญาณรูปสี่เหลี่ยม	22
รูปที่ 2.20 แสดงวงจรคอมมอนอีมิเตอร์ ในแสดงที่ 1	23
รูปที่ 2.21 แสดงวงจรคอมมอนอีมิเตอร์ ในแสดงที่ 3	24
รูปที่ 2.22 แสดงความถี่ที่ฟิลเตอร์กำหนดให้ผ่านได้	25
รูปที่ 2.23 วงจร Delyiannis & Friend	27
รูปที่ 2.24 วงจรเอนเวลโลป ดีเทคเตอร์	28
รูปที่ 2.25 วงจร Sallen & Key	28
รูปที่ 2.26 ผังเวลาการทำงานของ ADC 0804	30
รูปที่ 2.27 แสดงการเชื่อมต่อระหว่าง MCS-51 กับ ADC 0804	31
รูปที่ 2.28 รูปแสดงขาของ Parallel Port	35
รูปที่ 2.29 แสดงการรันโปรแกรมเพื่อทำการรับ-ส่งข้อมูล	40
รูปที่ 2.30 แสดงวิธีการต่อวงจรเพื่ออ่านค่าข้อมูล 40 bits เข้า parallel port	41
รูปที่ 3.1 รูปแสดงวงจรกำเนิดความถี่ 40 กิโลเฮิร์ตซ์	42

## สารบัญรูปภาพ(ต่อ)

	หน้า
รูปที่ 3.2 รูปแสดงการต่อวงจรเข้ารหัสพรีจิสเตอร์	43
รูปที่ 3.3 รูปแสดงการต่อวงจรคูณสัญญาณด้วย IC MC1496	44
รูปที่ 3.4 รูปแสดงการต่อวงจรขยายแบบอินเวอร์ตติง แอมป์ลีฟายเออร์	45
รูปที่ 3.5 รูปแสดงการต่อวงจรลดทอนสัญญาณ	45
รูปที่ 3.6 รูปแสดงการต่อวงจรบัฟเฟอร์ด้วยออป-แอมป์ เบอร์ LM6361	46
รูปที่ 3.7 แสดงการต่อตัวแปลงระดับสัญญาณไฟ (Voltage Regulator)	46
รูปที่ 3.8 แสดงการทำงานของโปรแกรมส่วนควบคุมการหมุนของสเตปปีงมอเตอร์	47
รูปที่ 3.9 วงจรขับสเตปปีงมอเตอร์ผ่าน พอร์ตขนานด้วยคอมพิวเตอรื	48
รูปที่ 3.10 การต่อตัวรับด้วยทรานซิสเตอร์แบบคอมมอนอิมิตเตอร์	49
รูปที่ 3.11 รูปแสดงตำแหน่ง โพลของวงจรแบนด์พาสฟิลเตอร์	50
รูปที่ 3.12 แสดงคุณสมบัติแต่ละสเตจของฟิลเตอร์	51
รูปที่ 3.13 แสดงการต่อวงจรฟิลเตอร์ตามที่ออกแบบได้	51
รูปที่ 3.14 แสดงการเปลี่ยน R แต่ละสเตจเพื่อปรับค่าอัตราขยาย	52
รูปที่ 3.15 แสดงการต่อวงจรฟิลเตอร์ที่ผ่านการปรับอัตราขยายแล้ว	52
รูปที่ 3.16 แสดงการต่อวงจรฟิลเตอร์ที่ผ่านการ scaling แล้ว	53
รูปที่ 3.17 แสดงการต่อวงจร AGC	54
รูปที่ 3.18 แสดงการต่อวงจรเอนเวล โลป ดีเทกเตอร์	54
รูปที่ 3.19 วงจรกรองความถี่ต่ำผ่านที่ออกแบบแล้ว	55
รูปที่ 3.20 โพลซาร์ตแสดงการทำงานของโปรแกรม MCS-51	57
รูปที่ 3.21 โพลซาร์ตแสดงการทำงานของโปรแกรมรับ-ส่ง 40 Bits	58
รูปที่ 3.22 แสดงการต่อวงจรรวมในส่วนภาคส่ง	59
รูปที่ 3.23 แสดงการต่อวงจรรวมในส่วนภาครับ	60
รูปที่ 3.24 แสดงวิธีต่อวงจรเพื่ออ่านค่าข้อมูล 40 บิตจาก A/Dมาเข้า parallel portร่วมกับMCS-51	61
รูปที่ 4.1 แสดงการทดสอบทรานสดิวเซอร์ ตัวส่งและตัวรับ ในระยะส่ง 30 เซนติเมตร	62
รูปที่ 4.2 แสดงการทดสอบทรานสดิวเซอร์ ตัวส่งและตัวรับใส่กรวย ในระยะส่ง 30 เซนติเมตร	63
รูปที่ 4.3 แสดงการทดสอบ โดยใช้ตัวส่ง 3 ตัวและตัวรับใส่กรวย ในระยะส่ง 30 เซนติเมตร	63
รูปที่ 4.4 แสดงระดับสัญญาณไฟ +12 โวลต์ที่ได้จากวงจร โวลต์เตจเรกกูเรเตอร์	64
รูปที่ 4.5 แสดงระดับสัญญาณไฟ -12 โวลต์ที่ได้จากวงจร โวลต์เตจเรกกูเรเตอร์	65
รูปที่ 4.6 แสดงระดับสัญญาณไฟ -8 โวลต์ที่ได้จากวงจร โวลต์เตจเรกกูเรเตอร์	65
รูปที่ 4.7 แสดงระดับสัญญาณไฟ +5 โวลต์ที่ได้จากวงจร โวลต์เตจเรกกูเรเตอร์	66

## สารบัญรูปภาพ(ต่อ)

	หน้า
รูปที่ 4.8 แสดงระดับสัญญาณไฟ -5 โวลต์ที่ได้จากวงจรโวลต์เคจเรกกูเรเตอร์	66
รูปที่ 4.9 แสดงสัญญาณจากออสซิลเลเตอร์ 40 กิโลเฮิร์ตซ์	67
รูปที่ 4.10 แสดงสัญญาณจากออสซิลเลเตอร์ 40 กิโลเฮิร์ตซ์ หลังผ่านวงจรลดทอนสัญญาณ	67
รูปที่ 4.11 แสดงสัญญาณที่ผ่านการเข้ารหัส 1010 ด้วยการวัดที่ ขา 10 ของวงจรเข้ารหัส เทียบ กราวด์ เทียบ กับสัญญาณพัลส์จากขา 1 ของไมโครคอนโทรลเลอร์	68
รูปที่ 4.12 แสดงสัญญาณเอาต์พุตของวงจรคูณสัญญาณ	69
รูปที่ 4.13 แสดงสัญญาณเอาต์พุตของวงจรคูณสัญญาณ หลังขยายแล้ว	70
รูปที่ 4.14 รูปแสดงเอาต์พุตของวงจรพีแอมป์ที่วัดได้	71
รูปที่ 4.15 รูปแสดงเอาต์พุตของวงจรพีแอมป์ที่วัดได้ วัดอยู่ที่ระยะ 3 เมตร	72
รูปที่ 4.16 รูปแสดงเอาต์พุตของวงจรพีแอมป์ที่วัดได้ วัดอยู่ที่ระยะ 4.5 เมตร	72
รูปที่ 4.17 รูปแสดงเอาต์พุตของวงจรแบนด์พาสฟิลเตอร์ เทียบกับ สัญญาณอินพุตรูปไซน์	73
รูปที่ 4.18 รูปแสดงเอาต์พุตของวงจรแบนด์พาสฟิลเตอร์ เทียบกับ สัญญาณอินพุตรูปสี่เหลี่ยม	74
รูปที่ 4.19 รูปแสดงเอาต์พุตของวงจรแบนด์พาสฟิลเตอร์ เทียบกับสัญญาณอินพุตรูปสามเหลี่ยม	74
รูปที่ 4.20 รูปแสดงเอาต์พุตของวงจร AGC เทียบกับ สัญญาณอินพุตที่ขนาด 3 โวลต์	75
รูปที่ 4.21 รูปแสดงเอาต์พุตของวงจร AGC เทียบกับ สัญญาณอินพุตที่ขนาด 7 โวลต์	76
รูปที่ 4.22 รูปแสดงเอาต์พุตของวงจรเอนเวลโลปดีเทคเตอร์ เทียบกับอินพุต 1.5 โวลต์	77
รูปที่ 4.23 รูปแสดงเอาต์พุตของวงจรเอนเวลโลปดีเทคเตอร์ เทียบกับอินพุต 3 โวลต์	78
รูปที่ 4.24 รูปแสดงเอาต์พุตของวงจรเอนเวลโลปดีเทคเตอร์ เทียบกับอินพุตจาก AGC	78
รูปที่ 4.25 รูปแสดงสัญญาณตรวจสอบซึ่งกันและกันระหว่าง MCS-51 กับคอมพิวเตอร์	79
รูปที่ 4.26 รูปแสดงสัญญาณตรวจสอบซึ่งกันและกันระหว่าง MCS-51 กับคอมพิวเตอร์	80
รูปที่ 4.27 รูปแสดงการรับข้อมูลผ่านวงจร Selector ร่วมกับ MCS-51 ผ่านทางคอมพิวเตอร์	80
รูปที่ 6.1 แสดงชิ้นงาน ภาคขับสเตปปีงมอเตอร์	107
รูปที่ 6.2 แสดงชิ้นงาน ภาคส่ง และส่วนอินเตอร์เฟส	107
รูปที่ 6.3 แสดงชิ้นงาน ภาครับ	108
รูปที่ 6.4 แสดงชิ้นงาน โดยรวม	108

## สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงลำดับการทำงานของสแตปปีงมอเตอร์ แบบครึ่งเฟส	19
ตารางที่ 2.2 แสดงลำดับการทำงานของสแตปปีงมอเตอร์ แบบ 2 เฟส	20
ตารางที่ 2.3 แสดงลำดับการทำงานของสแตปปีงมอเตอร์ แบบครึ่งเฟส	21
ตารางที่ 2.4 แสดงความสัมพันธ์ระหว่าง $V_{cc}/2$ กับ $V_{in}$ ของ ADC 0804	31
ตารางที่ 2.5 ตารางการถอดรหัสหน่วยความจำ	35
ตารางที่ 2.6 แสดงสถานะต่างๆของพอร์ตขนาน	37
ตารางที่ 3.1 ตารางแสดงสเตปการควบคุมสแตปปีงมอเตอร์ แบบ ไบโพลาร์	47
ตารางที่ 5.1 ตารางแสดงความคลาดเคลื่อนของอุปกรณ์อันเกิดจากอุณหภูมิ	81

## บทที่ 1

### บทนำ

ในปัจจุบันเทคโนโลยีต่างๆ ได้ถูกพัฒนาขึ้นอย่างรวดเร็ว โดยเฉพาะเทคโนโลยีคลื่นเสียงนั้นได้ถูกนำมาใช้ในการพัฒนาคุณภาพชีวิตของมนุษย์ในปัจจุบันอย่างมาก และมีการพัฒนาไปอย่างไม่หยุดยั้ง ไม่ว่าจะเป็นนำไปใช้ในการสื่อสาร , อุปกรณ์ทำความสะอาด , เครื่องควบคุมระยะไกล , เครื่องวัดระยะทาง และเทคโนโลยีอื่นๆ ที่นำไปใช้ในการประดิษฐ์อุปกรณ์ต่างๆ อีกมากมาย เราจึงเกิดแนวคิดในการใช้เทคโนโลยีทางด้านเสียงมาประยุกต์ใช้ในการระบุตำแหน่งของวัตถุหรือสิ่งกีดขวางในระนาบแนวนอน แล้วนำมาแสดงผลตำแหน่งเป็นพิกัดของระยะทางและมุม

อุปกรณ์ระบุตำแหน่งวัตถุโดยใช้คลื่นอัลตราโซนิก เป็นเครื่องมือที่นำมาใช้ โดยอาศัยหลักการของคลื่นสะท้อน ไมโครคอนโทรลเลอร์ (Microcontroller) และความรู้ทางคอมพิวเตอร์ โดยมีหลักการทำงานจากการส่งคลื่น ในรูปของการส่งคลื่นไปสะท้อนที่พื้นผิววัตถุแล้วกลับมายังเครื่องรับ เพื่อตรวจสอบตำแหน่งของวัตถุ ถ้าหากมีวัตถุอยู่ในระยะขอบเขตที่เราสามารถตรวจหาวัตถุได้ ตัวรับจะสามารถรับสัญญาณสะท้อนกลับได้ แล้วส่งไปให้ไมโครคอนโทรลเลอร์ทำการเก็บข้อมูลของสัญญาณ แล้วส่งต่อไปยังคอมพิวเตอร์เพื่อทำการประมวลผลหาระยะทางและมุมของวัตถุและนำมาแสดงผลบนจอภาพ ซึ่งรูปแบบคลื่นอัลตราโซนิก (Ultrasonic Wave) ที่เรานำมาใช้นี้เป็นคลื่นที่มีทิศทาง และสามารถกำหนดจุดที่คลื่นจะเคลื่อนที่ไปยังเป้าหมายได้อย่างแน่นอน คลื่นอัลตราโซนิกที่เรานำมาใช้นั้น ได้เลือกค่าความถี่ที่ 40 กิโลเฮิร์ตซ์

เราอาจนำหลักการจากอุปกรณ์ระบุตำแหน่งวัตถุโดยใช้คลื่นอัลตราโซนิกนี้ ไปประยุกต์ใช้ในงานด้านต่างๆ ได้ อาทิ ใช้แทนตาของหุ่นยนต์ ให้หุ่นยนต์เคลื่อนที่เข้าหาวัตถุที่ตรวจพบหรือหลบวัตถุที่กีดขวางอยู่ ใช้ในระบบคลื่นโซนาร์ ที่ใช้หลักการสะท้อนของคลื่นเช่นกัน

นอกจากนี้อุปกรณ์ระบุตำแหน่งวัตถุโดยใช้คลื่นอัลตราโซนิกยังสามารถนำมาประยุกต์ใช้ในรูปแบบอื่นได้อย่างกว้างขวาง ซึ่งทั้งนี้ก็แล้วแต่ความประสงค์ของผู้ใช้งาน และความสะดวก ซึ่งอาจมีการพัฒนาต่อไปในอนาคต

## บทที่ 2

### ทฤษฎีและหลักการ

#### 2.1 ทฤษฎีคุณสมบัติของคลื่นอัลตราโซนิก

##### 2.1.1 คุณสมบัติของคลื่นอัลตราโซนิก

อัลตราโซนิกหมายถึง คลื่นที่มีความถี่สูงเกินกว่าหูของมนุษย์จะได้ยิน โดยทั่วไปแล้ว หูของมนุษย์จะได้ยินในย่านความถี่ 30 เฮิรตซ์ ถึง 30 กิโลเฮิรตซ์ อัลตราโซนิกที่กล่าวถึงโดยทั่วไปจึงหมายถึง คลื่นที่มีความถี่สูงกว่า 30 กิโลเฮิรตซ์ขึ้นไป

อัลตราโซนิกเป็นคลื่นที่เกิดจากการเปลี่ยนแปลงพลังงานรูปอื่นให้มาเป็นพลังงานกล โดยการสั่นไปมา หรือเกิดการเปลี่ยนแปลงระหว่างพลังงานไฟฟ้ากับพลังงานกลทำให้เกิดคลื่นอัลตราโซนิกกระจายออกไปในอากาศ ดังนั้นจึงถือได้ว่าคลื่นที่เกิดขึ้นเป็นคลื่นกล (Mechanical wave) คลื่นอัลตราโซนิกสามารถสร้างได้ด้วยทรานสดิวเซอร์ ซึ่งเป็นอุปกรณ์ที่ใช้ในการเปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล โดยจะทำการแปลงไปมาระหว่างพลังงานไฟฟ้ากับพลังงานกล และมีความถี่เรโซแนนท์ (Resonant Frequency) คงที่อยู่ค่าหนึ่ง นอกจากนี้ยังมีคุณสมบัติต่างๆของคลื่น เช่น การสะท้อนของคลื่น ซึ่งเป็นที่ คุณสมบัติสำคัญที่เรานำมาประยุกต์ใช้ใน โครงการงานนี้

ความเร็วของคลื่นอัลตราโซนิกที่ยอมรับได้ในอากาศ จะมีความสัมพันธ์ดังสมการ

$$v = 331 + 0.6T \text{ เมตรต่อวินาที} \quad (2.1)$$

เมื่อ  $v$  คือ ความเร็วของคลื่นอัลตราโซนิก เมตรต่อวินาที (m/s)

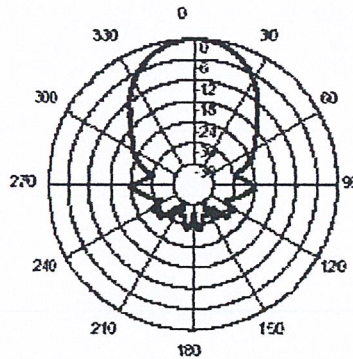
$T$  คือ อุณหภูมิของอากาศ องศาเซลเซียส (C)

##### 2.1.2 การทำงานของทรานสดิวเซอร์ (Transducer)

ทรานสดิวเซอร์ เป็นอุปกรณ์ที่สามารถใช้แปลงพลังงานไฟฟ้าให้เป็นพลังงานกล โดยการสั่นไปมาทำให้เกิดคลื่นเหนือเสียง และก็เป็นอุปกรณ์ที่ทำกรรับคลื่นเสียงเข้ามาแล้วทำการแปลงกลับไปเป็นสัญญาณไฟฟ้าได้ ดังนั้นในการทำงานจึงต้องมีทรานสดิวเซอร์ 2 ตัว โดยทำหน้าที่เป็นตัวส่งและตัวรับอย่างละตัวในระบบเดียวกัน ซึ่งทรานสดิวเซอร์จะมีความถี่เรโซแนนท์ (Resonant Frequency) อยู่ค่าหนึ่ง และจะต้องใช้คู่กันเสมอ

สำหรับทางด้านตัวส่ง เมื่อทรานสดิวเซอร์ได้รับสัญญาณแรงดันมาครบรอบ จะทำให้ชิ้นสารวัสดุโค้งงอ ทำให้เกิดการบีบอัดอากาศโดยรอบเกิดเป็นคลื่นเหนือเสียงขึ้นมา ดังนั้นถ้าป้อนสัญญาณเป็นห้วงๆ (Pulse) จากออสซิลเลเตอร์ ก็จะทำให้ชิ้นสารโค้งงอเล็กน้อย ก่อให้เกิดการบีบอัดอากาศเกิดเป็นคลื่นเหนือเสียงออกไป โดยทั่วไปกำลังเอาท์พุท ที่ออกมาจะสูงก็ต่อเมื่อความถี่ของสัญญาณออสซิลเลเตอร์ที่ป้อนเข้า ทรานสดิวเซอร์ตรงกับความถี่เรโซแนนท์ของทรานสดิวเซอร์นั้นๆ ส่วนที่ความถี่อื่นๆ กำลังของเอาท์พุทจะลดลง ทั้งนี้จะมีลักษณะของลำคลื่นในการส่ง (Beam Angle) ดังรูปที่ 2.1

ส่วนการทำงานของด้านตัวรับ มีการทำงานตรงกันข้ามกับด้านตัวส่ง คือ เมื่อมีคลื่นเหนือเสียงที่มีความถี่ตรงกับความถี่เรโซแนนซ์ของขั้วรับเข้ามา ก็จะทำให้ขั้วรับ โกงตัวไปมาและเกิดสัญญาณเป็นแรงดัน ไฟฟ้าขึ้นมาคร่อมขั้วทั้งสองของขั้วรับ



รูปที่ 2.1 ลักษณะของลำคลื่นในการส่ง (Beam Angle) ของทรานสดิวเซอร์

### 2.2 ทฤษฎีและหลักการของวงจรโดยรวม

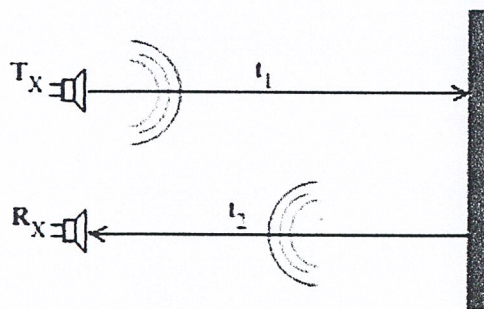
จากการที่เราทราบแล้วว่า คลื่นอัลตราโซนิค นั้น มีคุณสมบัติเป็นคลื่นกลที่เกิดการสะท้อนได้ เช่นเดียวกับคลื่นเสียง แต่ยังมีความเร็วเท่ากับคลื่นเสียง โดยมีความเร็วแปรผันตามอุณหภูมิและความสัมพันธ์(2.1)และจากสมการความสัมพันธ์ระหว่างความเร็วและระยะทางกับเวลา

$$s = vt \tag{2.2}$$

เมื่อ  $s$  คือ ระยะทางคลื่นเดินทางได้ (เมตร)

$t$  คือ เวลา (วินาที)

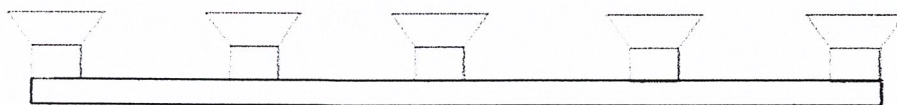
ดังนั้น เราจะเห็นว่า หลักการโดยคร่าวๆ ในการเดินทางของคลื่นอัลตราโซนิค เมื่อเราส่งคลื่นอัลตราโซนิค ไปกระทบสิ่งกีดขวาง และสะท้อนกลับมายังเครื่องรับ ดังรูปที่ 2.2 จะได้ว่า



รูปที่ 2.2 แสดงการส่งคลื่นอัลตราโซนิค ไปกระทบสิ่งกีดขวาง และสะท้อนกลับมายังเครื่องรับ

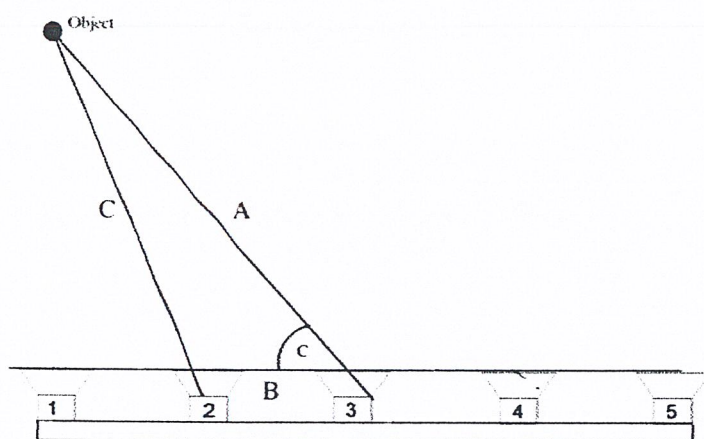
หากเราทำการจับเวลาที่คลื่นเดินทางไป หรือเวลาที่คลื่นเดินทางกลับได้ เราก็จะสามารถคำนวณหาค่าระยะทางระหว่างวัตถุ กับแหล่งกำเนิดคลื่น เพื่อนำค่าระยะทางนั้นมาพิจารณาต่อไปได้

อุปกรณ์ระบุตำแหน่งวัตถุชิ้นนี้ ได้นำเอาเทคนิคการหาระยะของวัตถุ ดังวิธีการข้างต้น มาประยุกต์ใช้ในการระบุตำแหน่ง โดยการใส่ตัวรับคลื่นอัลตราโซนิคเป็น 5 ชุด โดยตัวส่ง 1 ชุด โดยใช้ในการบ่งชี้ทิศทางการหมุนด้วยสเตรปป์มอเตอร์ มาวางเรียงกันในแนวระนาบ ดังรูปที่ 2.3



รูปที่ 2.3 รูปแสดงการจัดเรียงทรานสดิวเซอร์ตัวรับ

จากการเรียงของชุดอุปกรณ์รับส่งอัลตราโซนิก ดังที่กล่าวมา ทำให้อุปกรณ์สามารถระบุพิกัดของวัตถุเป็นระยะและมุม ในแนวระนาบได้ด้วยวิธีการทางเรขาคณิตดังต่อไปนี้



รูปที่ 2.4 รูปแสดงวิธีการคำนวณมุมและระยะห่างของวัตถุกับอุปกรณ์

พิจารณาจากรูปที่ 2.3 เมื่อคลื่นอัลตราโซนิคเข้าสู่ตัวรับทั้ง 5 ตัว เราจะทราบระยะทางระหว่างวัตถุกับตัวรับทั้งหมด เมื่อเราทราบระยะทางระหว่างวัตถุไปผู้ตัวรับ 2 ตัวที่ติดกัน (เช่น ตัวที่ 2 กับ 3 ) กำหนดให้เป็นระยะ A กับ C และเรารู้ระยะห่างระหว่างตัวรับทั้งสอง ( ด้าน B ) ทำให้เราทราบว่าวัตถุนั้น ทำมุมกับตัวรับ เราจะหามุมภายในของสามเหลี่ยมที่เกิดขึ้นได้ด้วยความสัมพันธ์ดังต่อไปนี้

$$C^2 = A^2 + B^2 - 2AB \cdot \cos(\theta_C) \quad (2.3)$$

$$\text{และ } \frac{A}{\sin(\theta_A)} = \frac{B}{\sin(\theta_B)} = \frac{C}{\sin(\theta_C)} \quad (2.4)$$

ในกรณีเดียวกันเราจึงจะสามารถหามุมที่ทำกับทรานสดิวเซอร์ตัวต่อๆไปได้ โดยมุมและระยะห่างของวัตถุที่ทำกับ ทรานสดิวเซอร์ตัวที่ 3 ( ตัวกลาง ) จะนำมาอ้างอิงเป็นมุมระยะที่วัตถุทำกับชุดอุปกรณ์ โดยทั้งนี้จะต้องนำมาพิจารณากับมุมและระยะที่เกิดจากทรานสดิวเซอร์ข้างเคียงเพื่อความถูกต้องด้วย ทำให้เราหาระยะและมุมเพื่อแสดงต่อยังภาคแสดงผลได้

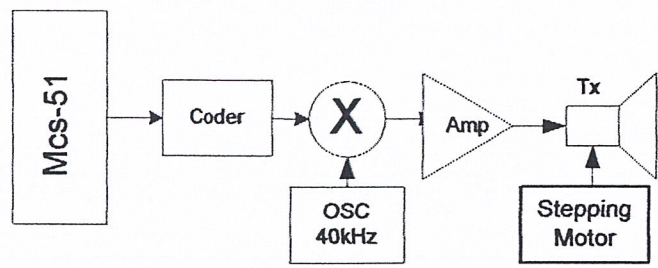
การทำงานของอุปกรณ์ชิ้นนี้ จะนำหลักการดังกล่าว ซึ่งจะมีบล็อกไดอะแกรม (Block Diagram) ดังรูปที่ 2.5 - 2.6 โดยหลักการทำงานของวงจรจะแบ่งออกเป็น 3 ส่วนคือ ภาคส่ง ภาครับ และภาคแสดงผล ซึ่งจะนำมาอธิบายได้ต่อไปนี้

#### ภาคส่ง

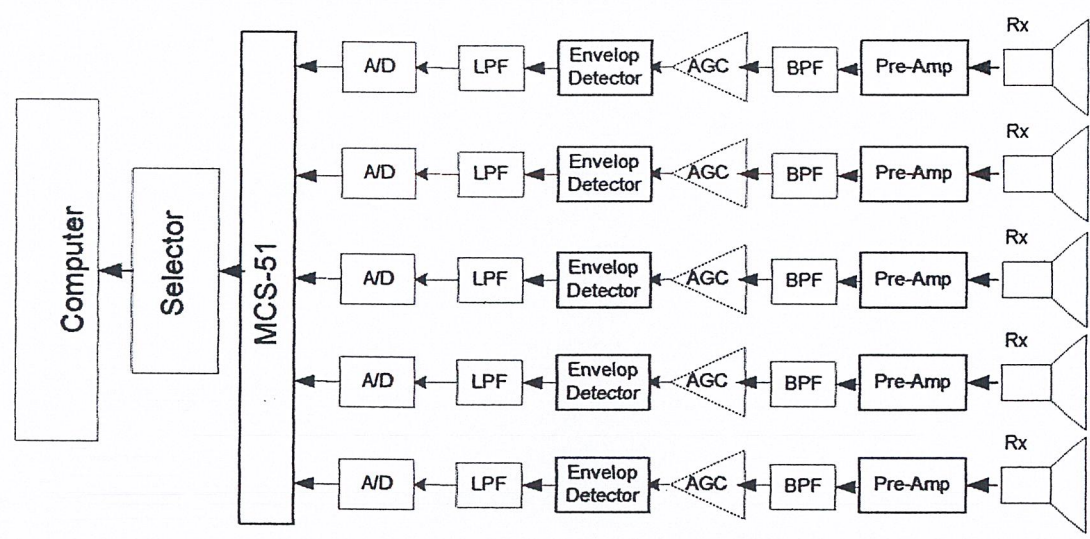
จะมาจากการสร้างสัญญาณพิกษาจาก MCS-51 มาเพื่อ Enable หรือ Disable วงจรเข้ารหัสซีพรีจิเตอร์ ให้สร้างรหัสของสัญญาณของแต่ละชุดส่ง มาคูณกับสัญญาณรูปไซน์ความถี่ 40 กิโลเฮิร์ตซ์ จากวงจรกำเนิดความถี่ แล้วผ่านวงจรถ่ายก่อนส่งผ่านทรานสดิวเซอร์ ให้ทำหน้าที่แปลงสัญญาณไฟฟ้าเป็นคลื่นอัลตราโซนิคที่ความถี่ 40 กิโลเฮิร์ตซ์ เพื่อทำการส่งสัญญาณออกมา โดยเริ่มส่งจากมุม 0 องศา ส่งออกไปกระทบกับวัตถุที่อยู่ในขอบเขตของบีมของทรานสดิวเซอร์ของตัวส่งในมุมสเตปแรก แล้วจะสะท้อนกลับมายังตัวรับ จากนั้นจึงทำการหมุนสเตปปีงมอเตอร์ไปยังสเตปต่อไปอีก 18 องศา เพื่อทำส่งครั้งต่อไป จนครบ 180 องศา

#### ภาครับ

เมื่อชุดรับทำการรับสัญญาณที่สะท้อนกลับมาได้ ก็จะนำเอาสัญญาณที่ได้มาขยายด้วยวงจร พรีแอมป์ แล้วนำมาผ่านวงจรแบนด์พาสฟิลเตอร์ โดยจะกรองเอาสัญญาณรบกวนที่ความถี่อื่นๆออกไปเหลือไว้แต่ความถี่ 40 กิโลเฮิร์ตซ์ จากนั้นนำมาผ่านวงจร AGC (Automatic Gain Control ) เพื่อคงระดับสัญญาณไม่ให้เกิน 5 โวลต์ แล้วนำไปเข้าสู่วงจรเอนเวลโลป ดีเทคเตอร์ (Envelop Detector) เพื่อกู้สัญญาณรหัสเดิมออกมา จากนั้นนำมาผ่านวงจรกรองความถี่ต่ำผ่านเพื่อตัดสัญญาณรบกวนที่ความถี่สูงทำให้สัญญาณที่ได้มีความเรียบมากขึ้น แล้วนำมาทำการแซมปลิงสัญญาณ โดยวงจร A/D (Analog to Digital Converter) เพื่อนำไปเปรียบเทียบกับรหัสที่ส่งออกมา ซึ่งสัญญาณที่ได้จากการกู้สัญญาณนี้ จะเกิดดีเลย์ (Delay) จากการเดินทางของคลื่นอัลตราโซนิค หากเราทราบความแตกต่างของเวลานั้นแล้ว เราจะสามารถหาระยะทางได้ ด้วยการเปรียบเทียบสัญญาณกับรหัสเดิม โดยเราจะทราบความแตกต่างของเวลานี้ได้จาก แล้วจะส่งค่าที่แซมปลิงไปยังคอมพิวเตอร์ เพื่อทำการเปรียบเทียบกับรหัสที่ใช้ในการส่ง มาคำนวณระยะทางและมุมที่ถูกต้อง เพื่อนำมาแสดงในจอภาพแสดงผลต่อไป



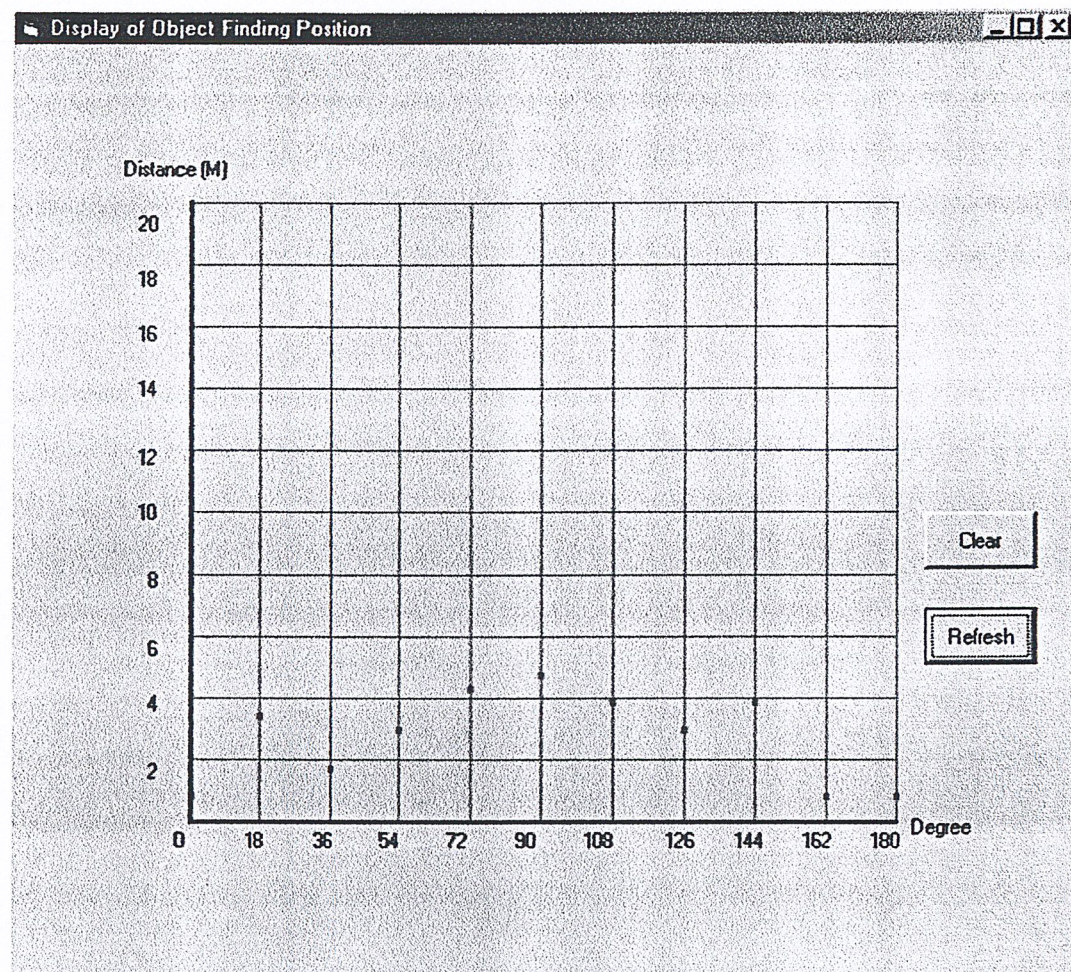
รูปที่ 2.5 รูปแสดงบล็อกไดอะแกรมภาคส่งของอุปกรณ์ระบุตำแหน่งวัตถุโดยใช้คลื่นอัลตราโซนิก



รูปที่ 2.6 รูปแสดงบล็อกไดอะแกรมภาคส่งของอุปกรณ์ระบุตำแหน่งวัตถุโดยใช้คลื่นอัลตราโซนิก

### ภาคแสดงผล

ในส่วนของภาคแสดงผลจะนำเอาค่าที่อ่านได้จากการชมปลิงจาก A/D มาคำนวณและประมวลผลยังคอมพิวเตอร์ โดยในที่นี้ได้เลือกใช้โปรแกรมที่เขียนจาก Visual Basic มารับค่าและแสดงผลออกมาทางจอภาพ ซึ่งจะมีตัวอย่างการแสดงผลดังต่อไปนี้



รูปที่ 2.7 รูปแสดงตัวอย่างการทำงานของโปรแกรมการแสดงผล

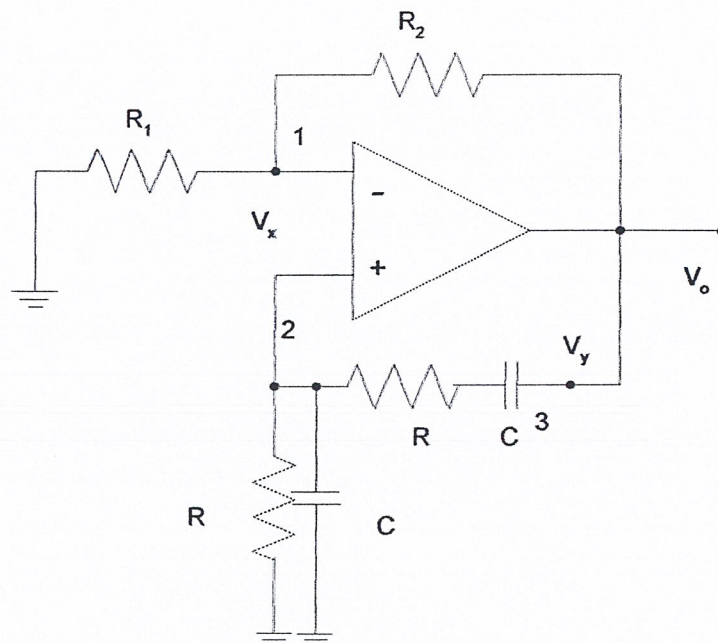
## 2.3 ทฤษฎีและหลักการขององค์ประกอบย่อย

เนื่องจากการทำงานของอุปกรณ์ประกอบด้วยวงจรจำนวนมาก จึงขอจำแนกวงจรออกเป็น 3 ส่วน เพื่ออำนวยความสะดวกเข้าใจ โดยแยกออกเป็นภาคส่ง ภาครับ และภาคแสดงผล ดังต่อไปนี้

### วงจรภาคส่ง

#### 2.3.1 วงจรกำเนิดสัญญาณความถี่ 40 กิโลเฮิรตซ์ ( Oscillator )

เป็นวงจรถูกกำเนิดคลื่นรูปไซน์ (Sine Wave) เพื่อนำมาทำเป็นสัญญาณคลื่นพาห้ เพื่อนำไปเข้ารหัส ก่อนป้อนเข้าทรานสดิวเซอร์ตัวส่งเพื่อส่งคลื่นออกไปอีกทีหนึ่ง ในที่นี้ใช้วงจรออสซิลเลเตอร์ ที่สร้างด้วย วงจรเวินบริดจ์ออสซิลเลเตอร์ (Wein Bridge Oscillator) มีลักษณะดังรูปที่ 2.8



รูปที่ 2.8 รูปแสดงวงจรถูกกำเนิดสัญญาณความถี่ 40 กิโลเฮิรตซ์

เริ่มพิจารณาโดยพิจารณาจาก โหนด(node) ที่ 1, โหนด ที่ 2 และ โหนด ที่ 3 จะได้สมการดังนี้

$$V_x = \frac{V_o R_1}{R_1 + R_2} = kV_o \quad (2.5)$$

$$\frac{V_x - V_y}{R} + \frac{V_x}{R} + V_x SC = 0 \quad (2.6)$$

$$\frac{V_y - V_x}{R} + (V_y - V_o)SC = 0 \quad (2.7)$$

จาก (2.6) จะได้

$$V_x \left( \frac{2}{R} + SC \right) - \frac{V_y}{R} = 0 \quad (2.8)$$

จาก (2.7) จะได้

$$V_y \left( \frac{1}{R} + SC \right) - \frac{V_x}{R} - V_o SC = 0 \quad (2.9)$$

นำค่า  $V_x$  จาก (2.5) แทนใน (2.8) และ (2.9) ได้

$$RkV_o \left( \frac{2}{R} + SC \right) = V_y \quad (2.10)$$

$$V_y \left( \frac{1}{R} + SC \right) - \frac{kV_o}{R} - V_o SC = 0 \quad (2.11)$$

แทนค่า  $V_y$  จาก (2.10) ลงไปใน (2.11) จะได้

$$V_o Rk \left( \frac{2}{R} + SC \right) \left( \frac{1}{R} + SC \right) - \frac{kV_o}{R} - V_o SC = 0$$

$$\frac{d^2 V_o}{dt^2} + \frac{dV_o}{dt} \left( \frac{3k-1}{kRC} \right) + \frac{V_o}{R^2 C^2} = 0$$

หากเลือกให้  $k = \frac{1}{3}$  จะได้

$$\frac{d^2 V_o}{dt^2} + \frac{V_o}{R^2 C^2} = 0$$

$$V_o = Ae^{j/RC} + Be^{-j/RC}$$

เลือกให้  $A = B$

$$V_o = 2A \left( \frac{e^{j/RC} + e^{-j/RC}}{2} \right)$$

$$V_o = 2A \cos \left( \frac{t}{RC} \right)$$

ดังนั้นเราสามารถกำหนดค่าความถี่การออสซิลเลต ได้โดยความสัมพันธ์

$$f = \frac{1}{2\pi RC} \quad (2.12)$$

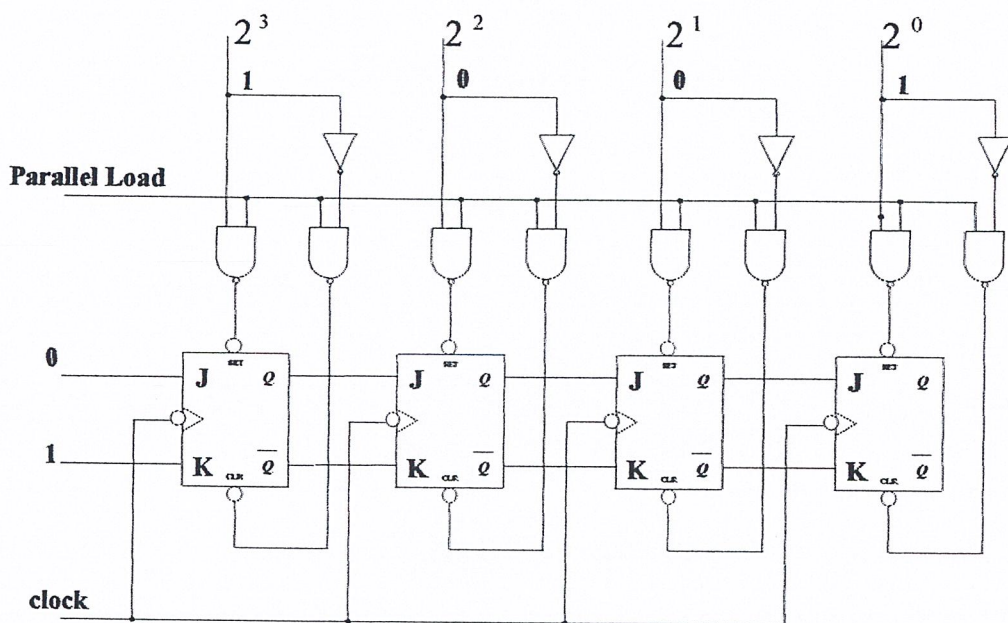
และกำหนดให้  $k = \frac{1}{3}$  ได้โดยกำหนดให้

$$\frac{R_2}{R_1} = 2 \quad (2.13)$$

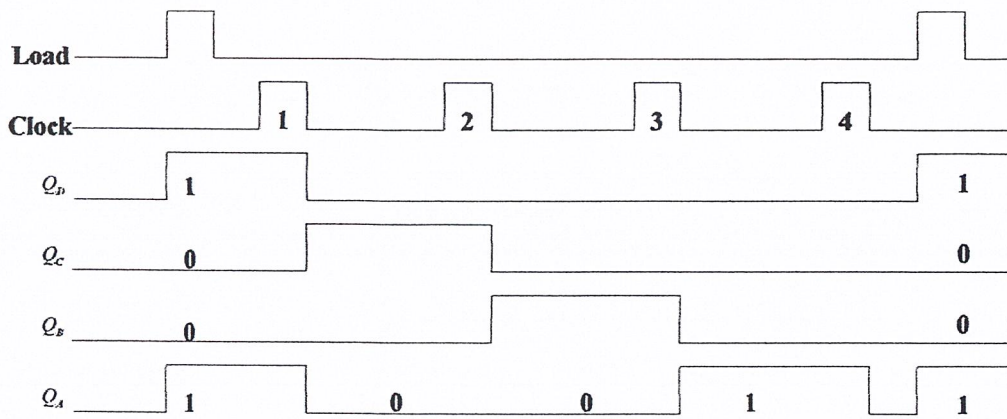
### 2.3.2 วงจรเข้ารหัสแบบชิฟท์รีจิสเตอร์ ( Shift Register )

เราต้องเข้ารหัสสัญญาณก่อนที่จะนำไปมอดูเลต ( Modulate ) กับความถี่ 40 กิโลเฮิร์ตซ์ แทนที่จะส่งคลื่นความถี่ 40 กิโลเฮิร์ตซ์ออกไปเลย เพื่อให้เราสามารถนำมาพิจารณาในการวัดระยะทางได้ไกลขึ้น เพราะที่ด้านรับ ตัวรับจะสามารถแยกแยะความแตกต่างระหว่างสัญญาณกับสัญญาณรบกวน ( Noise ) ได้มากขึ้น เราสามารถใช้วงจรชิฟท์รีจิสเตอร์ แบบ Parallel-in Serial-out มาเป็นวงจรเข้ารหัสสัญญาณได้

การทำงานของชิฟท์รีจิสเตอร์สามารถอธิบายได้ตามรูปที่ 2.9 และ 2.10 ได้ดังนี้คือ วงจร ฟลิปฟลอป A ( Flip-Flop ) มีค่าอินพุต J และ K เป็น 0 และ 1 ตามลำดับ ทำให้ค่า Q ของฟลิปฟลอป A จะกลายเป็น "0" หลังจากมีสัญญาณขอบขาลงเข้ามาที่ขา Clock Input ถ้าชิฟท์รีจิสเตอร์ ถูกป้อนสัญญาณนาฬิกาเข้า 4 ครั้ง ค่า Q ของฟลิปฟลอปทุกตัวจะกลายเป็น "0" และจะยังคงค่า "0" จนกระทั่งมีการโหลดแบบขนานเข้าชิฟท์รีจิสเตอร์อีกครั้ง



รูปที่ 2.9 รูปแสดงวงจรภายในของวงจรชิฟท์รีจิสเตอร์

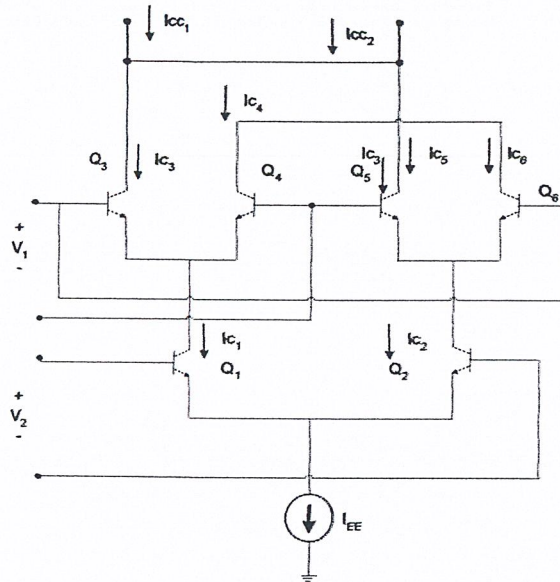


รูปที่ 2.10 รูปแสดงการทำงานของวงจรรีจิสเตอร์

ซึ่งการ โหลดค่าเข้ารีจิสเตอร์นี้ทำได้โดย การป้อนเลขรหัสไบนารีเข้าอินพุตแบบขนาน แล้วทำให้ค่า "1" เข้าที่ Parallel Load หรือ คอนโทรล จะเป็นการ enable แนนเกต ทั้งต่อกับ  $\overline{PRESET}$  และ  $\overline{CLEAR}$  ของฟลิปฟลอปแต่ละตัว เพราะมีการอินเวอร์เตอร์ระหว่างอินพุตของ แนนเกต ทั้งสอง ทำให้เอาท์พุตของแนนเกต 2 ตัว จะไม่มีทางมีค่าตรงกัน เมื่อแนนเกตถูก enable จะส่งผลให้ค่า Q นั้นขึ้นอยู่กับค่าอินพุตที่ต่อขนานเข้าสู่รีจิสเตอร์ ค่า Q จะไม่เปลี่ยน ทรานไซด์ที่มี Parallel Load เป็น "1" เมื่อ Parallel Load เป็น "0" แนนเกตจะถูก inhibited ทำให้ค่า Q จะเป็นค่าที่ถูกเลื่อนค่อๆกันไปเมื่อมีการป้อนสัญญาณนาฬิกา

### 2.3.3 วงจรคูณสัญญาณ แบบกิลเบอร์ต ( Gilbert Multiplier Circuit )

เราต้องการคูณสัญญาณระหว่างรหัสของชุดส่งแต่ละชุดกับสัญญาณคลื่นรูปไซน์ 40 กิโลเฮิร์ตซ์ ซึ่งจะนำไปแปลงเป็นคลื่นอัลตราโซนิค ในที่นี้เลือกใช้ วงจรคูณแบบกิลเบอร์ต ซึ่งแสดงได้รูปที่ 2.11



รูปที่ 2.11 รูปแสดงวงจรภายในและการทำงานของวงจรคูณสัญญาณแบบกิลเบอร์ต

จากวงจรจะได้ว่า

$$I_{C3} = \frac{I_{C1}}{1 + e^{-\left(\frac{V_1}{V_T}\right)}} \quad (2.14)$$

$$I_{C4} = \frac{I_{C1}}{1 + e^{\left(\frac{V_1}{V_T}\right)}} \quad (2.15)$$

$$I_{C5} = \frac{I_{C2}}{1 + e^{\left(\frac{V_1}{V_T}\right)}} \quad (2.16)$$

$$I_{C6} = \frac{I_{C2}}{1 + e^{-\left(\frac{V_1}{V_T}\right)}} \quad (2.17)$$

$$I_{C1} = \frac{I_{EE}}{1 + e^{-\left(\frac{V_2}{V_T}\right)}} \quad (2.18)$$

$$I_{C2} = \frac{I_{EE}}{1 + e^{\left(\frac{V_2}{V_T}\right)}} \quad (2.19)$$

ดังนั้นเราจะได้ว่า

$$I_{C3} = \frac{I_{EE}}{\left(1 + e^{-\left(\frac{V_2}{V_T}\right)}\right) \left(1 + e^{-\left(\frac{V_1}{V_T}\right)}\right)} \quad (2.20)$$

$$I_{C4} = \frac{I_{EE}}{\left(1 + e^{-\left(\frac{V_2}{V_T}\right)}\right) \left(1 + e^{\left(\frac{V_1}{V_T}\right)}\right)} \quad (2.21)$$

$$I_{C5} = \frac{I_{EE}}{\left(1 + e^{\left(\frac{V_2}{V_T}\right)}\right) \left(1 + e^{-\left(\frac{V_1}{V_T}\right)}\right)} \quad (2.22)$$

$$I_{C6} = \frac{I_{EE}}{\left(1 + e^{\left(\frac{V_2}{V_T}\right)}\right) \left(1 + e^{\left(\frac{V_1}{V_T}\right)}\right)} \quad (2.23)$$

จากวงจรเราจะพบอีกว่า

$$I_{CC1} = I_{C3} + I_{C5} \quad (2.24)$$

$$I_{CC2} = I_{C4} + I_{C6} \quad (2.25)$$

ดังนั้น

$$\Delta I = I_{CC1} - I_{CC2}$$

$$\Delta I = I_{C3} + I_{C5} - I_{C4} - I_{C6}$$

$$\Delta I = (I_{C3} - I_{C6}) - (I_{C4} - I_{C5}) \quad (2.26)$$

จะได้ว่า

$$\begin{aligned} I_{C3} - I_{C6} &= \frac{I_{EE}}{1 + e^{-\left(\frac{V_1}{V_T}\right)}} \left( \frac{1}{1 + e^{-\left(\frac{V_2}{V_T}\right)}} - \frac{1}{1 + e^{\left(\frac{V_2}{V_T}\right)}} \right) \\ &= \frac{I_{EE}}{1 + e^{-\left(\frac{V_1}{V_T}\right)}} \tanh\left(\frac{V_2}{2V_T}\right) \end{aligned} \quad (2.27)$$

ในทำนองเดียวกัน

$$\begin{aligned} I_{C4} - I_{C5} &= \frac{I_{EE}}{1 + e^{\left(\frac{V_1}{V_T}\right)}} \left( \frac{1}{1 + e^{-\left(\frac{V_2}{V_T}\right)}} - \frac{1}{1 + e^{\left(\frac{V_2}{V_T}\right)}} \right) \\ &= \frac{I_{EE}}{1 + e^{\left(\frac{V_1}{V_T}\right)}} \tanh\left(\frac{V_2}{2V_T}\right) \end{aligned} \quad (2.28)$$

ดังนั้น

$$\Delta I = I_{EE} \tanh\left(\frac{V_2}{2V_T}\right) \left( \frac{1}{1+e^{-\left(\frac{V_1}{V_T}\right)}} - \frac{1}{1+e^{\left(\frac{V_1}{V_T}\right)}} \right)$$

$$\Delta I = I_{EE} \tanh\left(\frac{V_2}{2V_T}\right) \tanh\left(\frac{V_1}{2V_T}\right) \quad (2.29)$$

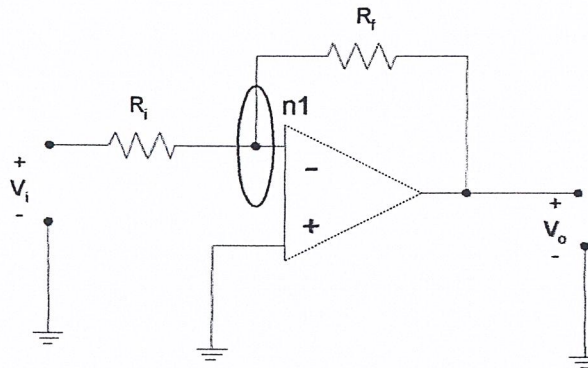
หาก  $\frac{V_1}{2V_T} \ll 1$  และ  $\frac{V_2}{2V_T} \ll 1$  จะได้ว่า

$$\Delta I = I_{EE} \left(\frac{V_1}{2V_T}\right) \left(\frac{V_2}{2V_T}\right) = kV_1V_2 \quad (2.30)$$

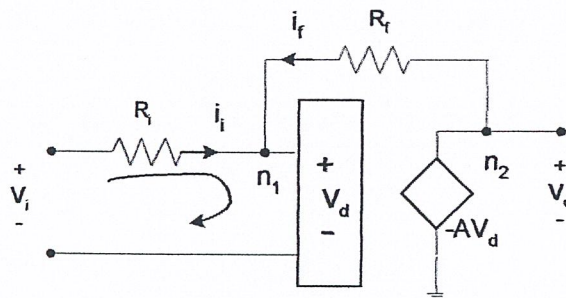
### 2.3.4 วงจรขยาย (Amplifier)

ในที่นี้เราได้เลือกใช้วงจรขยายแบบอินเวอร์ตติ้ง แอมพลิฟายเออร์ (Inverting Amplifier) เนื่องจากเป็นวงจรขยายที่สร้างได้ง่าย และไม่ซับซ้อน เหตุผลที่ไม่จำเป็นต้องใช้เพาเวอร์ แอมพลิฟายเออร์ (Power Amplifier) เพราะว่าการที่ทรานซิสเตอร์ จะกำเนิดคลื่นอัลตราโซนิคแรงเท่าไร นั้นขึ้นอยู่กับค่าความต่างศักย์ตกคร่อมตัวทรานซิสเตอร์เท่านั้น ไม่ได้ขึ้นอยู่กับกำลังของสัญญาณไฟฟ้าที่เข้ามา (พิจารณาได้จาก Datasheet ในภาคผนวก)

อินเวอร์ตติ้ง แอมพลิฟายเออร์ มีหลักการทำงานดังต่อไปนี้



รูปที่ 2.12 รูปแสดงการทำงานของวงจรวจรขยายแบบอินเวอร์ตติ้ง แอมพลิฟายเออร์



รูปที่ 2.13 รูปแสดงวงจรภายในของวงจรวจรขยายแบบอินเวอร์ตติ้ง แอมพลิฟายเออร์

จากวงจรภายในจะคำนวณได้ว่า

$$\text{KCL ;} \quad -V_i + R_i i_i - V_d = 0 \quad (2.31)$$

$$-V_d + i_f R_f - A V_d = 0 \quad (2.32)$$

$$\text{KCL ;} \quad i_i + i_f = 0 \quad (2.33)$$

$$i_i = -i_f$$

จากสมการ (2.31) ;

$$i_i = \frac{(V_i + V_d)}{R_i} \quad (2.34)$$

จากสมการ (2.33) ;

$$i_f = \frac{-(V_i + V_d)}{R_i} \quad (2.35)$$

แทนค่า  $i_f$  จาก (2.35) ลงในสมการ (2.32)

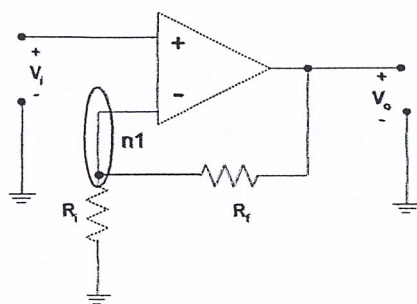
$$-V_d + \frac{-(V_i + V_d)}{R_i} R_f - A V_d = 0$$

$$\frac{V_o}{V_i} = \frac{-R_f / R_i}{\left( \frac{1}{A} + \frac{R_f}{A R_i} + 1 \right)}$$

ในกรณีที่ค่า A มีค่ามากๆ เราจะอนุมานได้ว่า

$$\frac{V_o}{V_i} = \frac{-R_f}{R_i} \quad (2.36)$$

2.3.5 วงจรนอน-อินเวอร์ตติ้ง แอมป์ลิฟายเออร์ (Non – Inverting Amplifier )

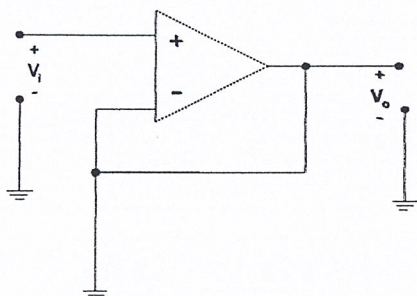


รูปที่ 2.14 รูปแสดงการทำงานของวงจรถ่ายแบบนอน-อินเวอร์ตติ้ง แอมป์ลิฟายเออร์

พิจารณาที่ โหนด 1 จะได้ว่า

$$\frac{V_i - 0}{R_i} + \frac{V_i - V_o}{R_f} = 0$$

$$\frac{V_o}{V_i} = \left( \frac{R_f + R_i}{R_i} \right) = 1 + \frac{R_f}{R_i}$$



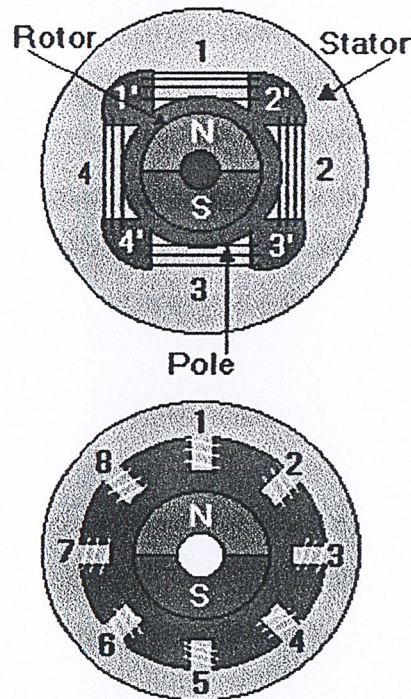
รูปที่ 2.15 รูปวงจรวอร์ฟเฟอร์ที่ได้จากประยุกต์จากนอน-อินเวอร์ตติ้ง แอมป์ลิฟายเออร์

เนื่องจากคุณสมบัตินี้ เราจึงนำมาประยุกต์ให้เป็นวงจรวอร์ฟเฟอร์ โดยการกำหนดให้ค่า  $R_f$  มีค่าเท่ากับ 0 เพื่อช่วยคงค่าสัญญาณให้เสถียรมากขึ้น และมีอัตราขยายเป็น 1 จึงไม่ทำให้สัญญาณมีการผิดเพี้ยนไป

### 2.3.6 สเตปป์ิงมอเตอร์ (Stepping Motor)

เป็นมอเตอร์ที่มีลักษณะการหมุนเป็นสเตป เมื่อเราป้อนไฟฟ้าให้กับมอเตอร์ทำให้หมุนเพียงเล็กน้อยตามเส้นรอบวงและหยุด ซึ่งต่างจากมอเตอร์ทั่วไปที่จะหมุนทันทีและตลอดเวลาเมื่อป้อนแรงดันไฟฟ้าข้อดีของสเตปมอเตอร์ สามารถกำหนดตำแหน่งของการหมุนด้วยตัวเลข (องศาหรือระยะทาง) ได้อย่างละเอียดโดยใช้คอมพิวเตอร์หรือ ไมโครคอนโทรลเลอร์เป็น เครื่องกำหนด

โครงสร้างของขั้วแม่เหล็กบนสเตเตอร์ทำมาจากแผ่นเหล็กวงแหวนที่มีซี่ยื่นออกมาประกบกันเป็นชั้นๆ โดยที่แต่ละซี่นั้นจะมีคอยล์(ขดลวด)พันสวมอยู่ เมื่อมีการป้อนกระแสผ่านคอยล์ทำให้เกิดสนามแม่เหล็กไฟฟ้า (Electromagnetic)



รูปที่ 2.16 โครงสร้างภายในสเตปป์ิงมอเตอร์

โดยแนวทางสเตปมอเตอร์เป็นอุปกรณ์จ่ายพวกเชิงกลทางไฟฟ้า โดยมีกรุปของไบนารีโวลต์เตจเป็นอินพุตและการเคลื่อนที่แบบเชิงมุมเป็นเอาต์พุต หรือว่าหมุนทีละสเตปซึ่งอยู่ระหว่าง 0.1 - 30 องศาอยู่ที่โครงสร้างของสเตปมอเตอร์ โดยตามสัญญาณพัลส์ที่จ่ายให้กับขดสเตเตอร์ทำให้เกิดแรงผลักแก่โรเตอร์หมุนไป สเตปมอเตอร์มีขดลวดหลายชุดในที่นี้เราเรียกว่า Phase(เฟส) ดังนั้นสัญญาณที่ต่อเนื่องเป็น Sequence(ซีควีน) ลักษณะของ Binary(ไบนารี) ซึ่งจะต้องไปผ่านวงจร Driver(ไดรเวอร์) ก็จะทำให้โรเตอร์หมุนไปอย่างต่อเนื่อง

การพันขดลวดบนสเตเตอร์ของสเตปมอเตอร์ การพันมีด้วยกัน 2 วิธี คือ แบบไบโพลาร์(Bipolar) กับแบบยูนิโพลาร์ (Unipolar)

### แบบไบโพลาร์

จะมีการพันขดลวดหนึ่งขด ในแต่ละขั้วแม่เหล็กของสเตเตอร์ โดยขั้วแม่เหล็กที่เกิดขึ้น ที่สเตเตอร์ จะถูกกำหนดโดยทิศทางของการไหลของกระแสไฟฟ้า ซึ่งสามารถทำให้เกิดขั้วแม่เหล็กในทิศทางตรงกันข้ามได้เพียง การกลับทิศทางของการไหลในกระแสไฟฟ้า โดยมาจากการควบคุมของวงจรวัดซึ่งให้กลับขั้วไฟฟ้า

### แบบยูนิโพลาร์

แบบนี้มี 2 ขด บนแต่ละขั้วแม่เหล็กของสเตเตอร์ ทำให้แต่ละขดลวดเกิดขั้วแม่เหล็กในทิศทางตรงกันข้าม การกลับทิศทางขั้วแม่เหล็กทำได้โดยใช้วงจรวัดซึ่งให้สลับหนึ่งไปยังอีกขั้วหนึ่งแทนกัน แบบยูนิโพลาร์จะทำให้เกิดแรงบิดน้อยกว่าแบบไบโพลาร์

### การสั่งงานควบคุมการหมุนของสเตปมอเตอร์

การควบคุมและสั่งงานให้สเตปมอเตอร์ทำงาน ไปที่ละสเตปสามารถทำได้โดยการจ่ายกำลังไฟไปยังขดลวด ในแต่ละขอบบนสเตเตอร์ โดยการป้อนจะทำในลักษณะเป็นลำดับหรือเรียกว่า ซีควเอนเชียล ในรูปที่ถูกคือ ซึ่งจะแบบ ได้เป็น 3 รูปแบบ คือ แบบเวฟ(wave) แบบ 2 เฟส(2 phase) และแบบครึ่งสเตป (half step) ทั้ง 3 แบบนี้ก็มีข้อดีและข้อเสียต่างกันออกไป

### แบบเวฟ (wave)

จะเป็นการกระตุ้นแบบที่ง่ายที่สุด ซึ่งจะทำการกระตุ้นขดลวดทีละขดในเวลาหนึ่งๆเรียงกันไป ทั้งนี้ขึ้นอยู่กับทิศทางที่เราต้องให้มอเตอร์หมุนไป วงจรที่นำมากระตุ้นนั้นจะมีราคาค่อนข้างจะถูกกว่าและง่ายกว่า

เราสามารถเขียนขั้นตอนการทำงานเป็นตารางออกมาได้ดังนี้

Step No.	Phase 1	Phase 2	Phase 3	Phase 4
1	ON			
2		ON		
3			ON	
4				ON
5	ON			
6		ON		

ตารางที่ 2.1 แสดงลำดับการทำงานของสเตปมอเตอร์ แบบครึ่งเฟส

### แบบ 2 เฟส(2 Phase)

วิธีนี้จะคล้ายกับการกระตุ้นในแบบเวฟแต่จะต่างกันตรงที่แบบ 2 เฟส จะกระตุ้นทีละ 2 ขด ที่อยู่ใกล้กันใน เวลาเดียวกัน และจะเรียงลำดับกันไป ดังเช่นแบบเดียวกับแบบเวฟ กล่าวถึงข้อดีข้อเสียของแบบ 2 เฟส แล้วมีดังนี้

**ข้อดี** การที่เราจะเพิ่มจำนวนขดลวดที่ถูกกระตุ้นจะทำให้แรงบิดได้มากกว่าแบบเวฟ ซึ่ง โรเตอร์ จะหมุนด้วยแรง ดึงแบเต็มที่แรงจาก ทั้ง 2 ขดลวดที่กระตุ้นพร้อมกัน

**ข้อเสีย** แบบ 2 เฟส จะกระตุ้นขดลวดนั้นต้องใช้กำลังไฟมากขึ้นเป็น 2 เท่าของแบบเวฟ ก็เป็นไปได้ตามธรรมชาติ ได้อย่างที่ค่อนข้างเสียอย่าง นั้นล่ะครับ

เราสามารถเขียนลำดับการกระตุ้นของขดลวดแบบ 2 เฟส ได้ดังในตารางต่อไปนี้

Step No.	Phase 1	Phase 2	Phase 3	Phase 4
1	ON	ON		
2		ON	ON	
3			ON	ON
4	ON			ON
5	ON	ON		
6		ON	ON	

ตารางที่ 2.2 แสดงลำดับการทำงานของสเตปป์มอเตอร์ แบบ 2 เฟส

### แบบครึ่งสเตป

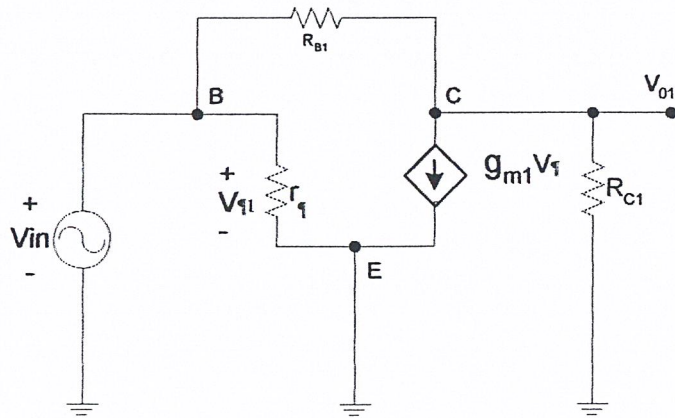
แบบนี้เป็นรูปแบบผสมผสานของการกระตุ้นระหว่าง แบบเวฟ กับ แบบ 2 เฟส เพื่อให้จำนวนรอบของสเตปป์ให้ มากขึ้นเป็น 2 เท่า ซึ่งในระบบนี้จะทำการกระตุ้นขดลวดเรียงกันไปเรื่อยๆ โดยมีข้อดี และข้อเสียดังนี้

**ข้อดี** การกระตุ้นแบบนี้จะให้แรงบิดที่เพิ่มมากขึ้น เนื่องจากช่วงสเตปป์ที่มีระยะสั้นลงอีกประการหนึ่งแต่ละ สเตปป์เกิดแรงดึงจากขดลวด 2 ขดที่ถูกกระตุ้นพร้อมกันเป็นผลให้ค่าตำแหน่งความถูกต้องมากขึ้นไปด้วย

**ข้อเสีย** ก็คงจะเช่นเดียวกับแบบ 2 เฟสล่ะครับ ที่ต้องจ่ายกำลังไฟเป็น 2 เท่าของแบบเวฟหรือจะใช้เท่ากับแบบ 2 เฟส นั้นเอง

ดังนั้นเราสามารถนำลำดับการทำงานของสเตปป์มอเตอร์ แบบครึ่งเฟส ในรูปของตาราง ได้ดังนี้

ในที่นี้เราใช้วงจรคอมมอนอีมิตเตอร์ (Common emitter) ต่อเป็น 3 แสดงดังรูปที่ 2.18 พิจารณาแสดงที่ 1 เมื่อเราคิดคิซีคอนดิชัน (DC Condition) แล้วจะได้  $V_{\pi 1}$  และ  $g_{m1}$  แล้วนำมาคิดเอซีคอนดิชัน ได้ดังรูปที่ 2.20



รูปที่ 2.20 แสดงวงจรคอมมอนอีมิตเตอร์ ในแสดงที่ 1

พิจารณาที่ โหนด (take node) ที่จุด  $V_{O1}$  จะได้

$$\begin{aligned} \frac{V_{O1}}{R_{C1}} + g_{m1}V_{\pi 1} + \frac{V_{O1} - V_{\pi}}{R_{B1}} &= 0 \\ \frac{(R_{C1} + R_{B1})V_{O1}}{R_{C1}R_{B1}} &= -g_{m1}V_{\pi 1} + \frac{V_{\pi 1}}{R_{B1}} \\ Gain_1 = \frac{V_{O1}}{V_{\pi 1}} = \frac{V_{O1}}{V_{IN}} &= \left(\frac{R_{C1}R_{B1}}{R_{C1} + R_{B1}}\right)\left(-g_{m1} + \frac{1}{R_B}\right) \end{aligned} \tag{2.37}$$

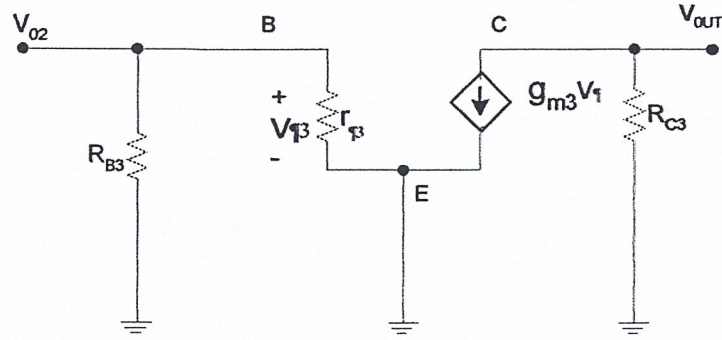
ถ้าเราเลือกให้  $R_{B1}$  มีค่ามากกว่า  $R_{C1}$  มากๆ จะได้

$$Gain_1 = \frac{V_{O1}}{V_{IN}} = -g_{m1}R_{C1} \tag{2.38}$$

พิจารณาแสดงที่ 2 จะเห็นว่า มีลักษณะเช่นเดียวกันกับแสดงแรก พิจารณาเช่นเดียวกันจะได้ว่า

$$Gain_2 = \frac{V_{O2}}{V_{O1}} = -g_{m2}R_{C2} \tag{2.39}$$

พิจารณาแสดงที่ 3 เพื่อพิจารณาเงื่อนไขของคิซี(DC) แล้วจะได้  $V_{\pi 3}$  และ  $g_{m3}$  แล้วนำมาคิดเงื่อนไขของเอซี(AC) ได้ดังรูปที่ 2.21



รูปที่ 2.21 แสดงวงจรคอมมอนอีมิเตอร์ ในเสตจที่ 3

$$V_{OUT} = -g_{m3} V_{\pi3} R_{C3} = -g_m V_{O2} R_{C3}$$

$$Gain_3 = \frac{V_{OUT}}{V_{O2}} = -g_{m3} R_{C3} \quad (2.40)$$

$$\therefore Gain_{total} = Gain_1 \cdot Gain_2 \cdot Gain_3$$

$$= (-g_{m1} R_{C1}) (-g_{m2} R_{C2}) (-g_{m3} R_{C3})$$

$$= -g_{m1} g_{m2} g_{m3} R_{C1} R_{C2} R_{C3} \quad (2.41)$$

### 2.3.8 วงจรแบนด์พาสฟิลเตอร์ (Band Pass Filter)

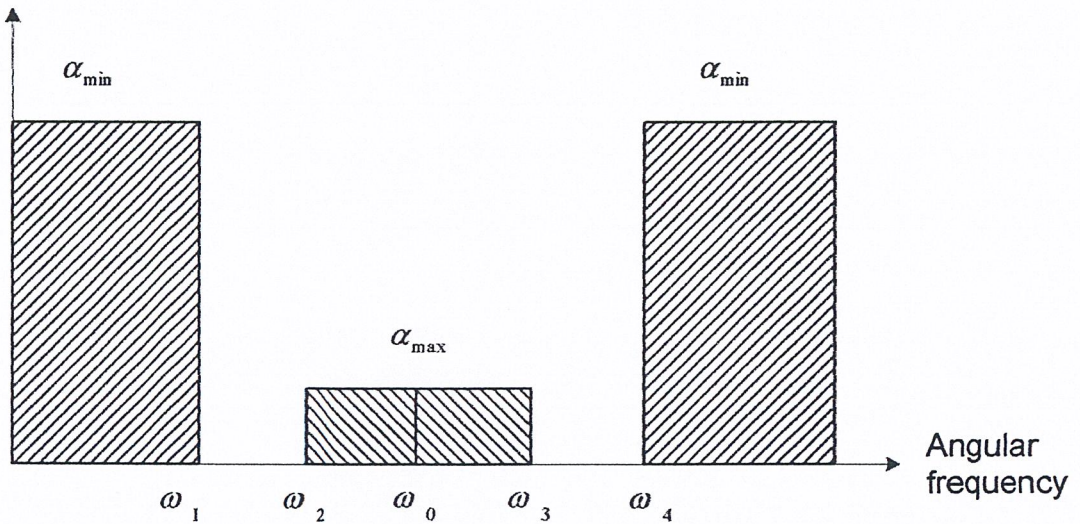
สัญญาณที่ผ่านวงจรพรีแอมป์ ซึ่งมีค่าอัตราการขยายสูงมาแล้วนั้น อาจจะขยายสัญญาณรบกวน (Noise) อื่นๆเข้ามา อีกทั้งอาจจะมาจากตัววงจรพรีแอมป์เอง นอกจากนี้ด้วยอัตราการขยายที่สูงนั้น ทำให้สัญญาณที่ได้ผิดเพี้ยนไป จนเป็นสัญญาณรูปสี่เหลี่ยม (Square Wave) ซึ่งจะก่อให้เกิดฮาร์โมนิก (Harmonic) อื่นๆของสัญญาณความถี่ 40 กิโลเฮิร์ตซ์ ด้วย แต่เราต้องการที่จะสัญญาณอัลตราโซนิค ความถี่ 40 กิโลเฮิร์ตซ์ ที่ส่งออกไปสะท้อนกับวัตถุเท่านั้น ทำให้เราจำเป็นต้องนำสัญญาณมาผ่านวงจรแบนด์พาสฟิลเตอร์ เพื่อกรองเอาสัญญาณรบกวนอื่นๆออกไป

ในที่นี้เราเลือกใช้แบนด์พาสฟิลเตอร์ แบบแอคทีฟ (Active) เนื่องจากเหมาะสมกับการใช้งานความถี่ต่ำ และยังสามารถเลือกปรับอัตราการขยายเพิ่มเติมจากวงจรพรีแอมป์ได้อีก และได้เลือกใช้แบนด์พาสฟิลเตอร์ แบบ บัตเตอร์วอร์คซ์ ในการออกแบบ เพราะตัวกรองสัญญาณมีความคมเพียงพอ และยังเกิดการกระเพื่อมในช่วงสัญญาณพาสแบน (Pass band) น้อยที่สุด (Minimum Ripple) หลังจากที่เรารู้จำนวน Order ค่า Q และ ความถี่กลาง (Center Frequency,  $f_0$ ) ของแต่ละสแตจแล้ว จึงเลือกใช้วงจร Delyianis & friends ในการสร้างแบนด์พาสฟิลเตอร์ เพราะเป็นการง่าย และสะดวกในการออกแบบ

### วิธีการออกแบบแบนด์พาส บัตเตอร์เวิร์ทซ์ ฟิลเตอร์

เมื่อเราทราบค่าการกระเพื่อมสูงสุดในพาสแบน ( $\alpha_{\max}$ ) และค่าการลดทอนต่ำสุดในสโตปแบน ( $\alpha_{\min}$ ) แล้ว เลือกค่าความถี่ที่พาสแบนด์  $\omega_1, \omega_2, \omega_3, \omega_4$  และค่าความถี่กลาง  $\omega_0$  ดังรูป

Attenuation



รูปที่ 2.22 แสดงความถี่ที่ฟิลเตอร์กำหนดให้ผ่านได้

การเลือกความถี่ เราต้องเลือกให้สอดคล้องกับความสัมพันธ์ ดังนี้

$$\omega_1 \omega_4 = \omega_2 \omega_3 = \omega_0^2 \quad (2.42)$$

จากนั้น ให้แปลงความถี่ของ LPF ซึ่งใช้คิดเป็นวงจรมันแบบ ดังนี้

$$\Omega_p = 1 \quad (2.43)$$

$$\Omega_s = \frac{\omega_4 - \omega_3}{\omega_2 - \omega_1} \quad (2.44)$$

เราจะได้ค่า Order ของฟิลเตอร์ ตามความสัมพันธ์

$$n = \frac{\log[(10^{\alpha_{\max}/10} - 1)/(10^{\alpha_{\min}/10} - 1)]}{2 \log \Omega_s} \quad (2.45)$$

จะได้  $\Omega_0$  ของ LPF ดังนี้

$$\Omega_0 = \frac{1}{(10^{\alpha_{\max}/10} - 1)^{1/2n}} \quad (2.46)$$

เมื่อทราบ Order ก็จะสามารถหาค่าตำแหน่งโพล (Pole) ใน s-plane ได้โดย การหามุมที่โพลกระทำกับแกน  $+\sigma$  (มุม  $\theta_k$ )

$$\theta_k = 90^\circ \left( \frac{2k + n - 1}{n} \right) \quad k = 1, 2, \dots, 2n$$

แต่เราจะเลือกจุดโพลที่อยู่ทางด้านซ้ายของระนาบ s-plane เนื่องจากจะทำให้วงจรเสถียร แล้วหาค่ามุมเหล่านั้นใหม่โดยให้เป็นมุมที่จุดโพลเหล่านั้นกระทำกับแกน  $-\sigma$  (มุม  $\psi_k$ ) จากนั้นแล้วหาพิกัดของโพลให้อยู่ในรูป

$$poles = \Sigma \pm j\Omega$$

$$\text{โดยที่ } \Sigma = \cos \psi \text{ และ } \Omega = \sin \psi$$

จากนั้นให้ใช้ Geffe's Algorithm หาค่า Q และ  $f_0$  ของแต่ละกรณีได้โดย

$$q_c = \frac{\omega_0}{BW}$$

$$C = \Sigma^2 + \Omega^2$$

$$D = \frac{2\Sigma}{q_c}$$

$$E = 4 + \frac{C}{q_c^2}$$

$$G = \sqrt{E^2 - 4D^2}$$

$$Q = \frac{1}{D} \sqrt{\frac{1}{2}(E + G)}$$

$$K = \frac{\Sigma Q}{q_c}$$

$$W = K + \sqrt{K^2 - 1}$$

$$\omega_1 = \omega_0 W$$

$$\omega_2 = \omega_0 / W$$

โดยแต่ละสแตจจะมี Transfer function ดังนี้

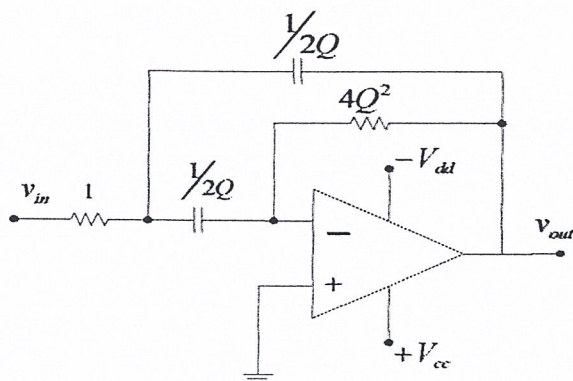
$$T_i(j\omega) = \frac{-(\omega_0 / Q)s}{s^2 + (\omega_0 / Q)s + \omega_0^2} \quad (2.47)$$

และอัตราขยายเป็น

$$|T_i(j\omega)| = \frac{(2Q_i \omega_{0i} \omega)^2}{\sqrt{(\omega_{0i}^2 - \omega^2)^2 + \left(\frac{\omega_{0i} \omega}{Q_i}\right)^2}} \quad (2.48)$$

โดย i คือ หมายเลขสแตจ

จากนั้นใช้วงจร Delyiannis & Friend เพื่อแปลงให้เป็นแบนด์พาสฟิลเตอร์ ดังรูป



รูปที่ 2.23 วงจร Delyiannis & Friend

จากนั้นใช้การ scaling ให้ได้ความถี่ที่ต้องการ โดยการเลือกค่า  $K_m$  จะได้

$$R_{new} = K_m \quad (2.49)$$

$$k_f = 2\pi f_0 \quad (2.50)$$

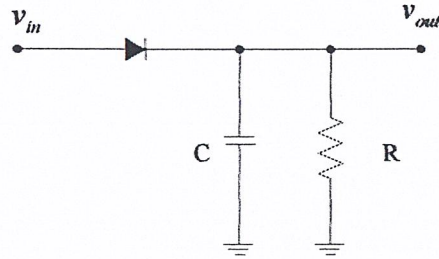
$$C_{new} = \frac{1}{k_m k_f} C_{old} \quad (2.51)$$

### 2.3.9 Automatic Gain Control (AGC)

เนื่องจากสัญญาณที่ได้รับนั้นมีขนาดที่แตกต่างกันไปมากเมื่อระยะทางที่สะท้อนวัตถุเปลี่ยนไป โดยขนาดของสัญญาณแปรผกผันกับระยะทาง ทำให้การนำสัญญาณไปประมวลผลในขั้นต่อไปนั้น จะเกิดปัญหาในการแปลงสัญญาณดิจิทัล (digital) เพื่อเปรียบเทียบ code ต่อไป หากวัตถุที่สะท้อนอยู่ใกล้เมื่อสัญญาณผ่านพรีแอมป์และ แบนด์พาสฟิลเตอร์ แล้วขนาดของสัญญาณก็อาจจะเกิน 5 โวลต์ ซึ่งเกินช่วงที่ A/D Converter จะสามารถรับได้ ในขณะที่วัตถุที่สะท้อนอยู่ไกล เมื่อผ่านพรีแอมป์ และแบนด์พาสฟิลเตอร์ แล้ว ขนาดของสัญญาณก็อาจจะต่ำเกินไป เราต้องการให้ขนาดสัญญาณคงที่ไม่ว่าวัตถุที่สะท้อนอยู่ที่ระยะใด เพราะจะได้ง่ายต่อการเปรียบเทียบรหัส จึงต้องใส่วงจร Automatic Gain Control เพื่อควบคุมให้ขนาดของสัญญาณคงที่ ถึงแม้อินพุตจะเปลี่ยนแปลงก็ตาม

### 2.3.10 วงจรเอนเวลโลป ดีเทคเตอร์ (Envelope detector)

หลังจากที่สัญญาณผ่านวงจร AGC แล้ว เราต้องการกู้รหัสกลับ จึงใช้วงจรเอนเวลโลป ดีเทคเตอร์ เพราะง่ายต่อการประกอบ โดยมีรูปวงจดังรูป



รูปที่ 2.24 วงจรเอนเวลโลป ดีเทคเตอร์

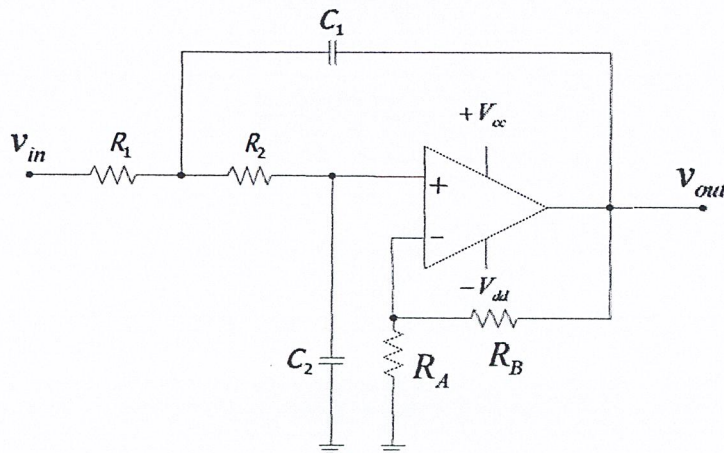
โดยมีความสัมพันธ์ระหว่าง ค่า  $RC$  และความถี่พาหะดังนี้

$$f_c = \frac{1}{2\pi RC} \quad (2.52)$$

### 2.3.11 วงจรกรองความถี่ต่ำผ่าน (Low Pass Filter, LPF)

หลังจากที่สัญญาณผ่านวงจรเอนเวลโลป ดีเทคเตอร์แล้ว เราจะได้สัญญาณรหัสกลับมา แต่เราต้องใส่วงจรกรองความถี่ต่ำผ่าน เพื่อให้สัญญาณรหัสที่ได้เรียบขึ้น และเป็นการกำจัดสัญญาณรบกวนที่อาจเกิดขึ้นจาก สัญญาณความถี่สูงได้

ในที่นี้เลือกใช้วงจร Sallen & Key เพราะง่ายต่อการออกแบบ มีรูปวงจดังนี้



รูปที่ 2.25 วงจร Sallen & Key

เราสามารถพิสูจน์หา transfer function ของวงจรได้ดังนี้

พิจารณาที่ โหนด A ของ op-amp

$$\frac{v_r - v_o}{R_B} + \frac{v_r - 0}{R_A} = 0$$

$$v_r = \frac{v_o}{R_B \left( \frac{1}{R_B} + \frac{1}{R_A} \right)} = \frac{R_A v_o}{R_A + R_B} = k v_o \quad (2.53)$$

$$\text{เมื่อ } k = \frac{R_A}{R_A + R_B}$$

พิจารณาที่ โหนด B ของ op-amp

$$\frac{v_r - v_1}{R_2} + \frac{v_r - 0}{1/SC_2} = 0$$

$$v_1 = R_2 v_r \left( \frac{1}{R_2} + SC_2 \right) \quad (2.54)$$

ที่ โหนด C

$$\frac{v_1 - v_i}{R_1} + \frac{v_1 - v_r}{R_2} + (v_1 - v_o)SC_1 = 0$$

$$v_1 \left( \frac{1}{R_1} + \frac{1}{R_2} + SC_1 \right) - \frac{v_i}{R_1} - \frac{v_r}{R_2} = v_o SC_1 \quad (2.55)$$

แทนค่า  $v_r = k v_o$  ลงใน (2.54)

$$v_1 = k v_o R_2 \left( \frac{1}{R_2} + SC_2 \right) \quad (2.56)$$

แทนค่า  $v_1$  จาก (2.56) และ  $v_r$  จาก (2.54) ลงใน (2.55)

$$k v_o R_2 \left( \frac{1}{R_2} + SC_2 \right) \left( \frac{1}{R_1} + \frac{1}{R_2} + SC_1 \right) - \frac{v_i}{R_1} - \frac{k v_o}{R_2} = v_o SC_1$$

$$\frac{v_o}{v_i} = \frac{\frac{1}{k C_1 C_2 R_1 R_2}}{S^2 + S \frac{[C_1 R_1 (k-1) + k C_2 (R_2 + R_1)]}{k C_1 C_2 R_1 R_2} + \frac{1}{C_1 C_2 R_1 R_2}}$$

และเมื่อเปรียบเทียบกับสมการในรูปทั่วไปของวงจรกรองความถี่ต่ำผ่าน ดังสมการที่ (2.57)

$$T(s) = \frac{K \omega_0^2}{s^2 + (\omega_0/Q)s + \omega_0^2} \quad (2.57)$$

∴ จะได้ว่า

$$\omega_0 = \sqrt{\frac{1}{C_1 C_2 R_1 R_2}} \quad \text{เมื่อให้ } C_1 = C_2 \text{ และ } R_1 = R_2 \quad (2.58)$$

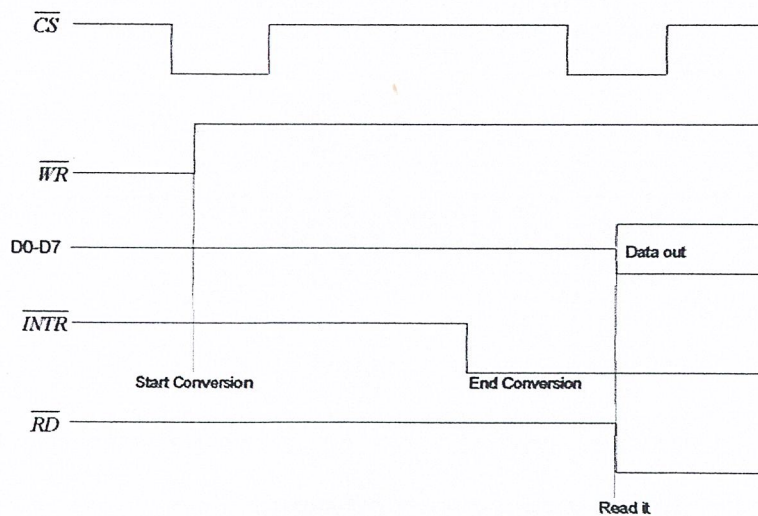
และได้อัตราการขยายเป็น

$$\text{Gain} = 1 + \frac{R_B}{R_A} \quad (2.59)$$

### 2.3.12 การแปลงสัญญาณอนาลอกเป็นดิจิทัล (Analog to Digital Converter)

เมื่อสัญญาณที่สะท้อนกลับมาถูกกรองจนเหลือโค้ด (code) ที่ค่อนข้างสมบูรณ์และมีแอมพลิจูดไม่เกิน 5 โวลต์ แล้วเราจะนำ code นั้นไปผ่าน IC ADC0804 เพื่อแปลงโค้ด จากอนาลอกให้เป็นดิจิทัล 8 บิตเพื่อส่งผ่านพอร์ตแบบขนานไปให้คอมพิวเตอร์ประมวลผลต่อไป โดยเราจะควบคุมขั้นตอนในการแปลงสัญญาณด้วย MCS-51

ADC 0804 เป็น A/D ขนาด 8 บิต มีความเร็วในการแปลงสัญญาณ (Conversion Time)  $110 \mu\text{s}$  วิธีสาร์ท A/D ทำได้โดยส่งสัญญาณเข้าขา  $\overline{WR}$  และ  $\overline{CS}$  ด้วยศักดาค่ำ (Low) แล้วให้รออีก  $110 \mu\text{s}$  หลังจากนั้น A/D จะส่งสัญญาณออกมาที่ขา  $\overline{INTR}$  (คือสัญญาณที่บอกว่าแปลงสัญญาณเสร็จแล้ว) มีระดับสัญญาณเป็น Low ขานี้มักจะต่อกับขา  $\overline{INTR}$  ของ CPU เพื่อบอกให้ CPU ทำขบวนการอินเตอร์รัปต์เพื่อนำข้อมูลที่แปลงแล้วไปเก็บ ในการอ่านข้อมูลไปเก็บ CPU ต้องส่งสัญญาณ  $\overline{RD}$  มาเข้า  $\overline{RD}$  ของ A/D (โดยขา  $\overline{CS}$  ต้องเป็น Low อยู่ก่อนหน้านี) โดยมีผังเวลาในการควบคุมดังรูปที่ 2.26



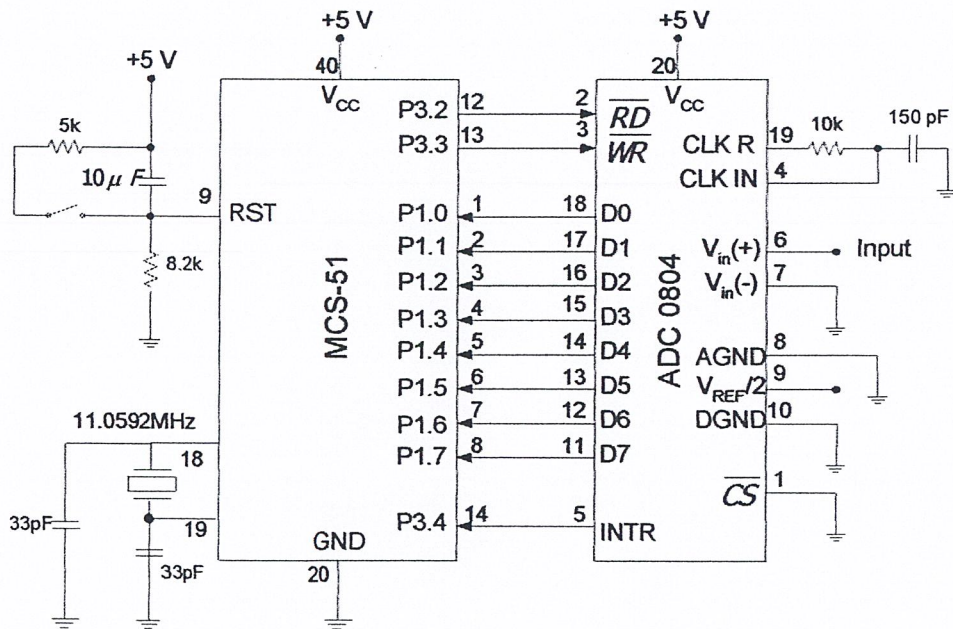
รูปที่ 2.26 ผังเวลาการทำงานของ ADC 0804

ซึ่งสัญญาณที่ถูกแปลงออกมาจะมีทั้งหมด 256 ระดับ (8 บิต) ซึ่งแต่ละระดับนั้นจะมีค่าเท่าไรนั้นขึ้นอยู่กับ  $V_{ref}/2$  ที่เราป้อนให้ ดังตารางที่ 2.4 ในที่นี้ปล่อยให้ขานี้ลอยซึ่งทำให้ range ของอินพุตอยู่ที่ 0-5 โวลต์

$V_{ref}/2$	$V_{in}$ (V)	Step Size(mV)
Not Connect	0 to 5	$5/256 = 19.53$
2.0	0 to 4	$4/256 = 15.62$
1.5	0 to 3	$3/256 = 11.71$
1.28	0 to 2.56	$2.56/256 = 10$
1.0	0 to 2	$2/256 = 7.81$
0.5	0 to 1	$1/256 = 3.90$

ตารางที่ 2.4 แสดงความสัมพันธ์ระหว่าง  $V_{ref}/2$  กับ  $V_{in}$  ของ ADC 0804

ตัวอย่างการเชื่อมต่อระหว่าง MCS-51 กับ ADC 0804 เป็นดังรูป



รูปที่ 2.27 แสดงการเชื่อมต่อระหว่าง MCS-51 กับ ADC 0804

### 2.3.13 ไมโครคอนโทรลเลอร์ (Microcontroller)

ในโครงการนี้ได้นำไมโครคอนโทรลเลอร์ MCS-51 มาประยุกต์ใช้ด้วย โดยใช้ในการกำเนิดสัญญาณที่จะนำไปเข้ารหัสในภาคส่ง, เก็บข้อมูลของสัญญาณที่รับเข้ามาในภาครับและนำสัญญาณที่เก็บได้ส่งต่อไปยังคอมพิวเตอร์

#### โครงสร้างพื้นฐาน

ไมโครคอนโทรลเลอร์ เป็นอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่งซึ่งรวมเอาหน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรรับสัญญาณอินพุต วงจรขับสัญญาณเอาต์พุต หน่วยความจำ วงจรกำเนิดสัญญาณนาฬิกาไว้ด้วยกัน ทำให้สามารถใช้งานแทนวงจรรีเลย์ที่ซับซ้อนได้เป็นอย่างดี ช่วยลดจำนวนอุปกรณ์และขนาดของระบบ ในขณะที่มีขีดความสามารถสูงขึ้น ภายใต้งบประมาณที่เหมาะสม

ในปัจจุบันนิยมใช้ไมโครคอนโทรลเลอร์ที่ใช้หน่วยความจำภายในเป็นแบบแฟลช ซึ่งทำให้สามารถลบและเขียนโปรแกรมใหม่ได้นับพันครั้ง จึงสามารถใช้งานในรูปแบบของไมโครคอนโทรลเลอร์ชิปเดี่ยวไม่ต้องใช้หน่วยความจำภายนอก ส่งผลให้สามารถใช้งานพอร์ตอินพุตและเอาต์พุตของไมโครคอนโทรลเลอร์ได้อย่างมีประสิทธิภาพ ซึ่งในโครงการนี้ได้นำไมโครคอนโทรลเลอร์ที่มีหน่วยความจำภายในแบบแฟลช เบอร์ 89C51 มาใช้

#### การทำงานของพอร์ต

ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีพอร์ตให้ใช้งานทั้งสิ้น 4 พอร์ต คือ พอร์ต 0 ถึง พอร์ต 3 แต่ละพอร์ตมีขนาด 8 บิต เป็นพอร์ตแบบ 2 ทิศทาง กล่าวคือ สามารถเป็นได้ทั้งอินพุตสำหรับรับสัญญาณข้อมูลเข้าและเอาต์พุตสำหรับส่งสัญญาณข้อมูลออก ทุกพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีวงจรถอดและวงจรถักข้อมูลจบบัฟเฟอร์อินพุต

ที่พอร์ต 0 และพอร์ต 2 จะใช้งานเป็นพอร์ตอินพุตและเอาต์พุตสำหรับงานทั่วไป และใช้ในการติดต่อกับหน่วยความจำภายนอก สำหรับพอร์ต 3 ทั้งพอร์ต และพอร์ต 1 บางขานอกจากจะใช้เป็นขาพอร์ตอินพุตเอาต์พุตตามปกติแล้วยังสามารถใช้งานในหน้าที่พิเศษได้อีกขึ้นอยู่กับว่าเป็นไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ใด

#### จังหวะการทำงานของไมโครคอนโทรลเลอร์ MCS-51

ในการใช้งานไมโครคอนโทรลเลอร์ MCS-51 จะต้องทำความเข้าใจถึงจังหวะการทำงานของซีพียูและลำดับขั้นตอนการประมวลผลคำสั่ง ในการประมวลผลคำสั่งของซีพียูนั้นจะมีขั้นตอนหลักๆ 2 ขั้นตอนคือ กระบวนการเฟตช์ (fetch) เป็นการเรียกคำสั่งออกจากหน่วยความจำโปรแกรมแล้วทำการแปลรหัสคำสั่งนั้นเป็นภาษาเครื่องเพื่อเตรียมการประมวลผล ขั้นตอนต่อมาคือ กระบวนการเอ็กซีคิวต์ (execute) เป็นการกระทำตามคำสั่งที่กำหนดหรือตามที่เฟตช์ขึ้นมาโดยกระบวนการก่อนหน้า เมื่อทำการเอ็กซีคิวต์คำสั่งเรียบร้อยแล้ว ก็จะไปเริ่มกระบวนการเฟตช์คำสั่งใหม่ต่อไป

ในการทำงาน 1 รอบหรือ 1 แมกซีนไซเคิล ซีพียูในไมโครคอนโทรลเลอร์ MCS-51 จะใช้เวลา 12 คาบเวลาของสัญญาณนาฬิกา นั่นคือ เวลาในการทำงาน 1 ไซเคิลมีค่าเท่ากับ 1 มิลลิวินาที หรือมีความเร็วในการทำงานภายใน 1 เมกกะเฮิร์ตซ์ ในกรณีที่ใช้ความถี่สัญญาณนาฬิกา 12 เมกกะเฮิร์ตซ์ ดังนั้นถ้า

ต้องการทราบความเร็วของการทำงานภายในไมโครคอนโทรลเลอร์ MCS-51 สามารถหาได้จาก ค่าความถี่สัญญาณนาฬิกาหารด้วย 12 และถ้าต้องการหาค่าเวลาของ 1 รอบการทำงานหรือ 1 แมกซ์ไซเคิล สามารถทำได้โดยการหาส่วนกลับของความเร็ว

#### กระบวนการรีเซ็ต ในไมโครคอนโทรลเลอร์ MCS-51

การรีเซ็ต (reset) เป็นกระบวนการที่ทำให้ซีพียูในไมโครคอนโทรลเลอร์กลับไปเริ่มต้นทำงานใหม่ที่แอดเดรส 0000H ในหน่วยความจำโปรแกรม พร้อมกันนั้นยังส่งผลให้เกิดการเคลียร์ค่าของข้อมูลในรีจิสเตอร์ภายในไมโครคอนโทรลเลอร์ และทำให้รีจิสเตอร์ทุกตัวกลับไปสู่สถานะเริ่มต้นทำงานใหม่

การเกิดรีเซ็ตในไมโครคอนโทรลเลอร์ MCS-51 เกิดขึ้นได้เพียงกรณีเดียวคือ ที่ขา RST (ขา 9) ได้รับลอจิก “1” อย่างน้อยเป็นเวลา 2 แมกซ์ไซเคิล หรือ 24 คาบเวลาของสัญญาณนาฬิกา โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานอยู่ เมื่อเกิดการรีเซ็ตขึ้น ไมโครคอนโทรลเลอร์จะสร้างสัญญาณรีเซ็ตภายในขึ้นเพื่อตอบสนองการรีเซ็ตจากภายนอก

#### การจัดหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51

ในไมโครคอนโทรลเลอร์ MCS-51 มีหน่วยความจำหลักๆ อยู่ 2 ส่วนคือ หน่วยความจำโปรแกรม และหน่วยความจำข้อมูล

##### หน่วยความจำโปรแกรม (Program Memory)

ใช้เก็บข้อมูลของโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์ หรือที่เรียกว่า โปรแกรมมอนิเตอร์ (monitor program) สำหรับ MCS-51แบบแฟลช เบอร์ 89C51 สามารถติดต่อหน่วยความจำโปรแกรมได้สูงสุด 64 กิโลไบต์ โดยสามารถเลือกใช้หน่วยความจำโปรแกรมภายในอย่างเดียวหรือรวมกับภายนอกหรือเลือกใช้หน่วยความจำภายนอกอย่างเดียวก็ได้ โดยภายใน 89C51 มีหน่วยความจำภายใน 4 กิโลไบต์

##### หน่วยความจำข้อมูล (Data Memory)

มี 2 แบบคือ หน่วยความจำข้อมูลภายนอกและภายใน โดย 89C51 สามารถติดต่อกับหน่วยความจำข้อมูลภายนอกได้สูงสุด 64 กิโลไบต์ และมีหน่วยความจำข้อมูลภายในเป็นแบบแรม (RAM : Random Access Memory) ขนาด 128 ไบต์ จัดสรรแบ่งเป็น 3 ส่วน คือ หน่วยความจำข้อมูลส่วนล่าง, ส่วนบน และรีจิสเตอร์ฟังก์ชันพิเศษ (SFR : Special Function Register) แต่ละส่วนมีขนาด 128 ไบต์

รีจิสเตอร์ฟังก์ชันพิเศษ เป็นรีจิสเตอร์ที่ใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์ มี 22 ตัว สำหรับ 89C51 มีแอดเดรสอยู่ระหว่าง 80H-FFH ในพื้นที่ของหน่วยความจำข้อมูลส่วนบน สามารถเข้าถึงได้โดยตรง จะขอกกล่าวถึงเฉพาะเพียงรีจิสเตอร์ตัวที่ใช้งานทั่วไป

- รีจิสเตอร์แสดงสถานะของโปรแกรม (Program Status Word : PSW) เป็นรีจิสเตอร์ขนาด 8 บิตสามารถเข้าถึงได้ในระดับบิต มีแอดเดรสอยู่ที่ 00H ทำหน้าที่เก็บสถานะการทำงานของโปรแกรมในขณะนั้น จะเรียกสถานะต่างๆของโปรแกรมว่า แฟล็ก (flag) เมื่อซีพียูกระทำคำสั่งทางคณิตศาสตร์และลอจิกแล้วเกิดการเปลี่ยนแปลงสถานะขึ้น ผลของการเปลี่ยนแปลงจะมาปรากฏที่บิตต่างๆ ของ PSW

- แอควิวเลเตอร์ (Accumulator : ACC) มีขนาด 8 บิต สามารถเข้าถึงระดับบิตได้ มีแอดเดรสอยู่ที่ EOH เป็นรีจิสเตอร์ที่ใช้สำหรับเก็บข้อมูลหรือผลลัพธ์ที่ได้จากการทำงานของไมโครคอนโทรลเลอร์ โดยเฉพาะอย่างยิ่งการคำนวณทางคณิตศาสตร์และลอจิก ก่อนที่จะส่งไปให้ซีพียูต่อไป เรียกสั้นๆ ว่า รีจิสเตอร์ A

- รีจิสเตอร์ B มีขนาด 8 บิต สามารถเข้าถึงระดับบิตได้ อยู่ที่แอดเดรส FOH มีหน้าที่เก็บข้อมูลทั่วไป และมีหน้าที่พิเศษคือ หากต้องการคูณหารทางคณิตศาสตร์ ต้องนำข้อมูลที่ต้องการหารหรือคูณมาเก็บไว้ในรีจิสเตอร์ B แล้วจึงกระทำคำสั่งคูณหรือหารกับรีจิสเตอร์ A ต่อไป

รีจิสเตอร์ SFR ตัวอื่นๆจะกล่าวถึงในเรื่องต่อไป

**การเชื่อมโยง MCS-51 กับหน่วยความจำข้อมูลภายนอก (Data Memory)**

หน่วยความจำข้อมูลภายนอก เป็นหน่วยความจำที่เก็บข้อมูลได้ถึง 64 กิโลไบต์ เป็นหน่วยความจำแรม (RAM) สัญญาณจาก MCS-51 ที่ใช้ในการอ่านข้อมูลจากแรม คือ  $\overline{RD}$  และสัญญาณที่ใช้ในการเขียนข้อมูลลงแรม คือ  $\overline{WR}$

แรม (RAM : Random Access Memory) เป็นหน่วยความจำที่สามารถเขียนและอ่านข้อมูลได้ ข้อมูลจะสูญหายทันทีที่ขาดไฟเลี้ยง แบ่งเป็น 2 ชนิด คือ ไดนามิกแรม (Dynamic RAM) และ สตาทิกแรม (Static RAM)

สตาทิกแรม เป็นหน่วยความจำชนิดหนึ่งที่ย่ายในการนำมาใช้งานแต่ราคาจะสูงเมื่อเทียบกับไดนามิกแรม ซึ่งราคาถูกกว่าแต่ยุ่งยากขึ้นมากก็คือต้องทำการ รีเฟรช (Refresh) หน่วยความจำตลอดเวลา ส่วนสตาทิกแรมที่มีใช้อย่างแพร่หลายมีอยู่ด้วยกันหลายเบอร์ตามขนาดความจุของหน่วยความจำ เบอร์ที่นิยมกันมากสำหรับงานที่ไม่ใหญ่มากนักคือ เบอร์ TMS4016 หรือ 6116 ซึ่งมีขนาดความจุ  $2K \times 8$

**การถอดรหัสตำแหน่งของหน่วยความจำ**

การถอดรหัสตำแหน่งของหน่วยความจำมีความจำเป็นอย่างยิ่งเพราะว่าหน่วยความจำที่นำมาใช้มีขนาดเล็กกว่า 64 กิโลไบต์ ดังนั้น จึงต้องนำมาเรียงกันแล้วใช้วงจรถอดรหัสตำแหน่งของหน่วยความจำแยกเป็นแต่ละช่วง ถ้าใช้หน่วยความจำขนาด 4 กิโลไบต์ จำนวน 8 ตัว จะได้เท่ากับ 32 กิโลไบต์ ถ้าให้ตัวแรกเริ่มที่แอดเดรส 8000H ตัวที่ 2 ก็จะเริ่มที่แอดเดรส 9000H ดังรายละเอียดดังตารางที่ 2.5

EPROM ตัวที่	แอดเดรส (HEX)	A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub> .....A <sub>0</sub>
1	8000-8FFF	1	0	0	0	X.....X
2	9000-9FFF	1	0	0	1	X.....X
3	A000-AFFF	1	0	1	0	X.....X
4	B000-BFFF	1	0	1	1	X.....X
5	C000-CFFF	1	1	0	0	X.....X
6	D000-DFFF	1	1	0	1	X.....X
7	E000-EFFF	1	1	1	0	X.....X
8	F000-FFFF	1	1	1	1	X.....X

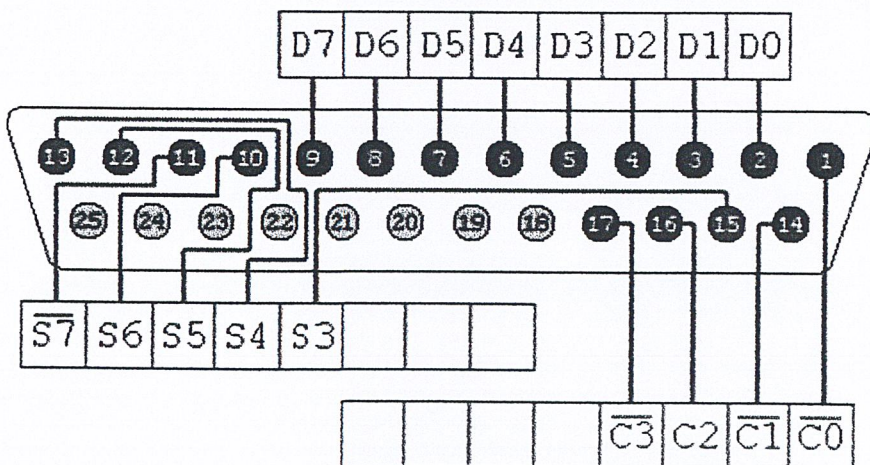
ตารางที่ 2.5 ตารางการถอดรหัสหน่วยความจำ

ส่วนการถอดรหัสตำแหน่งของหน่วยความจำจะใช้ไอซีถอดรหัส 3 to 8 line Decoder เบอร์ 74LS138

#### 2.3.14 พอร์ทขนาน (Parallel Port)

การเลือกใช้งานในพอร์ทชนิดนี้ สามารถทำการประยุกต์ใช้งานได้ดีเพราะสามารถทำงานได้ที่ละ 8 บิตในการติดต่อข้อมูลกับอุปกรณ์ภายนอกให้ระดับแรงดันที่ใช้กับอุปกรณ์ TTL ที่สัญญาณลอจิกเป็น "1" เท่ากับ +5 โวลต์ และลอจิกเป็น "0" เท่ากับ 0 โวลต์ และมีความเร็วในการรับ-ส่งข้อมูลในอัตราที่สูงประมาณ 1.3 Mbytes/sec มีข้อเสียคือไม่สามารถทำงานในระยะทางที่ไกลๆระหว่างอุปกรณ์ที่ติดต่อกัน เพราะจะเกิดความผิดพลาดของข้อมูลขึ้นง่ายเนื่องจากระดับแรงดันไม่สม่ำเสมออีกทั้งสิ้นเปลืองค่าใช้จ่ายในการต้องใช้สายสัญญาณหลายเส้น

พอร์ทขนานนั้นประกอบไปด้วย 25 ขา และ 3 พอร์ต โดยการอ่านและเขียนค่า จะต้องกำหนด Address ของพอร์ตให้ถูกต้อง โดยปกติจะเป็น &H378 สามารถแสดงตำแหน่งของแต่ละขาได้ดังนี้



รูปที่ 2.28 รูปแสดงขาของ Parallel Port

D=Data Port (Address=&H378)

S=Status Port (Address=&H379)

C=Control Port (Address=&H37A)

#### **Data Port**

สามารถใช้ในการส่งข้อมูลได้ทั้ง 2 ทาง ทั้งรับและส่ง ใน EPP Mode โดยการกำหนดด้วย Control bit 5 ของ พอร์ตนานเอง โดยปกติแล้วจะมีค่าเริ่มต้นเป็น 00000000

#### **Status Port**

เราสามารถใช้งานได้เพียง 5 บิต ในการรับค่าอินพุตจากภายนอก โดยปกติแล้วจะมีค่าตั้งต้นเป็น 21 ในแต่ละ ขา แต่จะมีขา 7 ที่เริ่มต้นเป็น 0 หากต่อลงกราวด์จะมีค่าเป็น 1

#### **Control Port**

เราสามารถใช้งาน Port control แทน Status Port หรือ Data Port ได้ตามที่ต้องการ แต่โดยปกติแล้ว จะใช้ในการส่งข้อมูลออกเพื่อทำการเชื่อมต่อกับอุปกรณ์ภายนอก สามารถใช้งานได้ 4 บิต นอกนั้นเป็น บิต สวอนที่ถูกเก็บไว้ในการควบคุมด้วยตัวคอมพิวเตอรืเอง เช่นเดียวกับ Status Port อีก 3 ขาที่หายไป

Pin	Job	D/S/C	Name
1	Output	C0	-Strobe
2	Output	D0	+Data Bit 0
3	Output	D1	+Data Bit 0
4	Output	D2	+Data Bit 0
5	Output	D3	+Data Bit 0
6	Output	D4	+Data Bit 0
7	Output	D5	+Data Bit 0
8	Output	D6	+Data Bit 0
9	Output	D7	+Data Bit 0
10	Input	S6	-Acknowledge
11	Input	S7	+Busy
12	Input	S5	+Paper End
13	Input	S4	+Select In
14	Output	C1	-Auto Feed
15	Input	S3	-Error
16	Output	C2	-Initialize
17	Output	C3	-Select
18-25	Ground	-	Ground

ตารางที่ 2.6 แสดงสถานะต่างๆของพอร์ตขนาน

### 2.3.15 การเขียนโปรแกรมด้วยภาษา Visual Basic 6.0 ( VB )

เขียนโปรแกรมติดต่อ I/O ผ่านทาง Port ของเครื่องคอมพิวเตอร์ ผ่านทาง Parallel Port หรือ Printer Port นั้น เราสามารถใช้ VB ทำงานด้าน Interface Hardware ได้ไม่แพ้ภาษาตัวอื่นๆและง่ายกว่า จึงทำให้สามารถทำความเข้าใจเพื่อจะนำไปใช้งานได้สะดวกและรวดเร็ว

Visual Basic 6 นั้นจะไม่มีฟังก์ชันสำหรับติดต่อพอร์ตโดยตรง สามารถติดต่อทางพอร์ตได้โดยใช้ API(Application Programming Interface) โดยจะต้องมี ไฟล์ DLL ไว้สำหรับเรียกฟังก์ชันเพื่อติดต่อพอร์ตโดยจะเขียนด้วยภาษา VC++, C++, Pascal เป็นต้น

เราสามารถเขียนโปรแกรมโดยใช้ Visual Basic เพื่อควบคุมฮาร์ดแวร์ ผ่านพอร์ตขนาน(Parallel Port) ได้ทั้งการส่งค่าไปควบคุม(Output) และการรับค่าตรวจสอบ(Input) ทั้งนี้ต้องทำความเข้าใจในการอ้างอิง Address Printer Port เพื่อสื่อสารกับคอมพิวเตอร์ให้ถูกต้องเสียก่อน ซึ่งในปกติแล้วการใช้ฟังก์ชันทั้ง Inp32 และ Out32 จำเป็นอย่างยิ่งที่จะต้องอ้างอิง Address Printer Port ในการที่จะ ไปตรวจสอบและส่งค่าในแต่ละบิตของ Printer Port โดยจะแบ่งออกเป็น 3 ส่วนคือ

1) Data คือส่วนที่เป็นส่งข้อมูลที่มีจำนวน 8 บิต ให้กับ Printer (คอมพิวเตอร์ด้วยกันก็ได้) อีกทั้งอันนี้แหละที่เราจะควบคุมสัญญาณแต่ละบิต เพื่อนำไปใช้ต่อกับอุปกรณ์ภายนอกสำหรับตำแหน่งอ้างอิงของ Data Port จะเท่ากับตำแหน่งของ base printer port

LPT1 Data=H378

2) Status คือส่วนที่เป็นตัวคอยตรวจสอบสถานะของ Printer เช่น การที่เครื่อง Error, กระดาษหมด, ส่งสัญญาณพร้อมทำงาน เป็นต้น

ทั้งนี้เราสามารถนำเรา Status Port นี้มาใช้ในงานควบคุมอุปกรณ์ได้โดยเป็นตัวรับสัญญาณ Input(+5 Vdc) สำหรับตำแหน่งอ้างอิงของ Data Port จะเท่ากับตำแหน่งของ base printer port+1

LPT1 Status=H379 (H378+1)

3) Control คือส่วนที่ควบคุมการทำงานของ Printer เช่น Strobe, Auto linefeed, Select printer เป็นต้น สามารถจะรับและส่งข้อมูลได้ 6 บิต สำหรับตำแหน่งอ้างอิงของ Data Port จะเท่ากับตำแหน่งของ base printer port+1

LPT1 Status=H37A(H378+2)

### 2.3.16 การเขียนโปรแกรมด้วยภาษา Visual C++ 6.0 (VC++)

จากการที่เราไม่สามารถเขียน โปรแกรมด้วยวิซวลเบสิกให้สามารถ ติดต่อผ่านทางพอร์ต Parallel ได้โดยตรงนั้น เราจึงจะต้องสร้างไฟล์ DLL(Dynamic Link Library) เพื่อขอยืมฟังก์ชันติดต่อ I/O port ของภาษาอื่นมาใช้ ซึ่งสามารถจะเขียนด้วยภาษาใดก็ได้ที่มีฟังก์ชันติดต่อพอร์ตได้ ในที่นี้เราเลือกใช้ภาษา Visual C++ 6.0 โดยมีฟังก์ชัน ติดต่อกับ Input และ Output ของพอร์ตดังนี้

1. `_Inp` ซึ่งเป็นฟังก์ชันที่จะอ่านค่าจากตำแหน่งอินพุตพอร์ต ซึ่งมีฟังก์ชันมีดังนี้

`_Inp` = อ่านค่าเป็น ไบต์ เราจะกำหนดชื่อฟังก์ชันเพื่อใช้ใน VB ---> `inpVB`

`_Inpw` = อ่านค่าเป็นเวิร์ด เราจะกำหนดชื่อฟังก์ชันเพื่อใช้ใน VB ---> `inpwVB`

`_Inpd` = อ่านค่าเป็นดับเบิลเวิร์ดเราจะกำหนดชื่อฟังก์ชันเพื่อใช้ใน VB ---> `inpdwVB`

ซึ่งถ้าเวลาเราเรียกใช้ฟังก์ชันที่เขียน Code ด้วย ภาษา C ตามที่กล่าวมาแล้วนั้น ในโปรแกรม Visual Basic ต้องประกาศฟังก์ชัน ซึ่งมีรูปแบบดังนี้

```
Declare Function InpVB LIB "DLLPORT" (Byval port As Integer) As Long
```

```
Declare Function InpwVB LIB "DLLPORT" (Byval port As Integer) As Integer
```

```
Declare Function InpdwVB LIB "DLLPORT" (Byval port As Integer) As Long
```

ถ้าใช้ประกาศภายใน Form ไม่ใช่ที่โมดูลไฟล์ อาจจะต้องใส่ คำว่า Private ดังเช่น

```
Private Declare Function InpVB Lib "DLLPORT"(ByVal port As Integer) As Long
```

2. `_outp` คือฟังก์ชันส่งค่าไปที่ตำแหน่งเอาต์พุตพอร์ต ตามค่าตำแหน่งที่กำหนด ซึ่งมีฟังก์ชันดังนี้

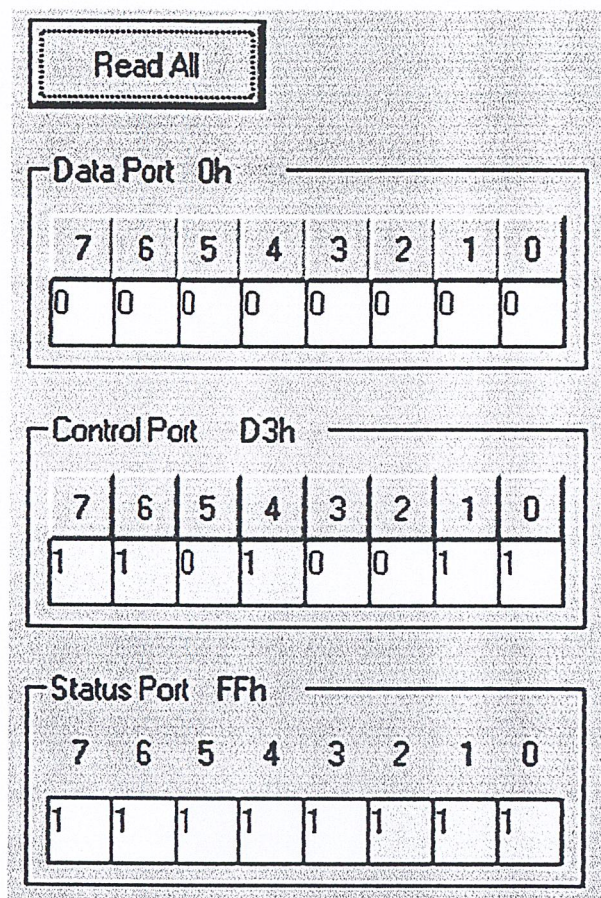
`_outp` = ส่งค่าเป็น ไบต์ เราจะกำหนดชื่อฟังก์ชันเพื่อใช้ใน VB ---> `outVB`

`_outpw` = ส่งค่าเป็นเวิร์ด เราจะกำหนดชื่อฟังก์ชันเพื่อใช้ใน VB ---> `outwVB`

`_outpd` = ส่งค่าเป็นดับเบิลเวิร์ด เราจะกำหนดชื่อฟังก์ชันเพื่อใช้ใน VB ---> `outdwVB`

สำหรับ โค้ดที่จะเขียนเป็นฟังก์ชันที่เรากำหนดชื่อ ไว้แล้วนั้นสามารถเขียนเป็น code ภาษา C ใน MS VC++ 6 จะกล่าวไว้ในภาคผนวก

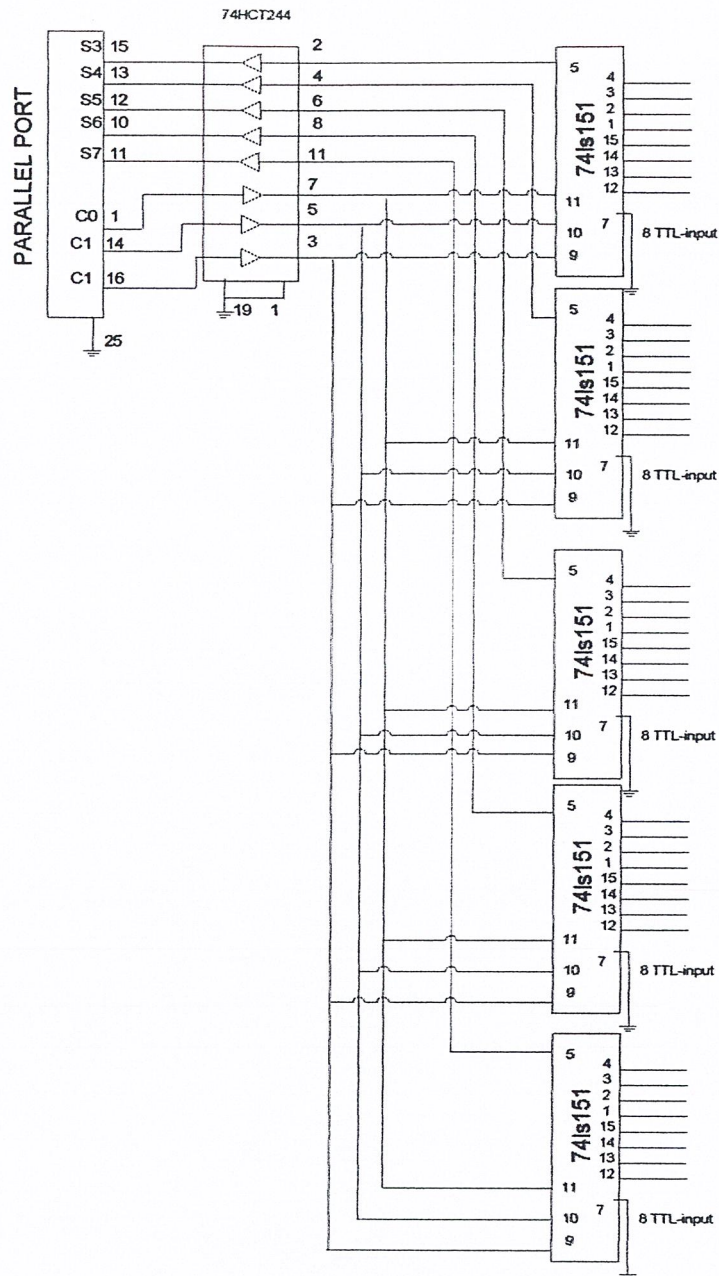
สามารถนำมาเขียนเป็นโปรแกรมแสดงการรับส่งข้อมูลด้วยพอร์ตนานได้ (Pseudo code ในภาคผนวก) ซึ่งจะแสดงให้เห็นจากการรันโปรแกรมดังนี้



รูปที่ 2.29 แสดงการรันโปรแกรมเพื่อทำการรับ-ส่งข้อมูล

การใช้พอร์ตนาน รับ-ส่งข้อมูลร่วมกับอุปกรณ์ภายนอกนั้น จำเป็นจะต้องเลือกใช้ประเภทของพอร์ตนานให้เหมาะสมกับการใช้งาน ในที่นี้เราต้องการรับ-ส่งข้อมูลด้วยความเร็วสูงและมีความแน่นอน เราจึงใช้พอร์ทแบบ PS/2-type (Simple Bidirectional) โดยนำสัญญาณเข้าที่ขา Status 5 ขา control port 4 ขา และ รับส่งข้อมูลด้วย ขา data port อีก 8 ขา

จากที่เราทราบว่าเราสามารถนำเข้าสู่สัญญาณ ไบนารีด้วยพอร์ตนาน สามารถนำเข้าด้วยวิธีต่าง ในที่นี้ได้เลือกใช้การนำสัญญาณเข้า Computer เป็นจำนวน 40 บิต โดยใช้วิธี อ่านค่าทีละ 5 บิต ด้วยขา Status 5 ขา ทั้งนี้จะให้การควบคุมการอ่านค่าด้วย ขา control 3 ขา โดยมีหลักการทำงานร่วมกับวงจรเลือกรับค่าสัญญาณ 40 บิต และเขียน โปรแกรมรับค่า ดังต่อไปนี้



รูปที่ 2.30 แสดงวิธีการต่อวงจรเพื่ออ่านค่าข้อมูล 40 บิต เข้าพอร์ตขนาน

หลักการของวงจรนี้คือ จะใช้ค่าที่ออกจาก ขาคอนโทรล ทั้ง 3 ของ พอร์ตขนาน ในการเลือกค่า อินพุต 1 ใน 8 ตัวของ 74LS151 แต่ละตัวเพื่อส่งค่าผ่านบัฟเฟอร์ (74HC244) ไปยังขาสแตตัส ของ พอร์ต ขนาน ทั้ง 5 ขา

โปรแกรมจะทำหน้าที่ในการอ่านค่าที่ได้ทั้ง 40 บิต มาจัดเรียงข้อมูลใหม่ ให้เป็นข้อมูล 8 บิต ที่ ได้รับแต่ละตัวจาก 74LS151

### บทที่ 3

#### การคำนวณและสร้าง

#### 3.1 วงจรภาคส่ง

##### 3.1.1 วงจรกำเนิดความถี่ (Oscillator)

ในโครงการชิ้นนี้เลือกใช้วงจรถ่ายความถี่แบบเวินบริดจ์ (Wein Bridge Oscillator) มากำเนิดคลื่นสัญญาณไซน์ ความถี่ 40 กิโลเฮิรตซ์ ซึ่งเป็นความถี่ที่ทรานซิสเตอร์อัลตราโซนิคตอบสนองได้ดีที่สุด ทั้งนี้เพราะวงจรถ่ายความถี่แบบเวินบริดจ์ มีโครงสร้างไม่ซับซ้อนและง่ายต่อการออกแบบเลือกความถี่ ซึ่งออกแบบให้ได้ความถี่ 40 กิโลเฮิรตซ์นั้นทำได้โดยการเลือกความต้านทาน และตัวเก็บประจุ ซึ่งได้พิสูจน์ไว้แล้วในสมการ (2.12) ที่หัวข้อ 2.3.1 ว่า

$$f = \frac{1}{2\pi \cdot RC}$$

ในที่นี้เลือกใช้ ออป-แอมป์ เบอร์ LM6361 และกำหนดค่า C ก่อนว่าเป็น  $0.1\mu F$  จึงได้ว่า  $R=40\Omega$  และจากสมการที่ (2.13)ทำให้เราทราบว่า

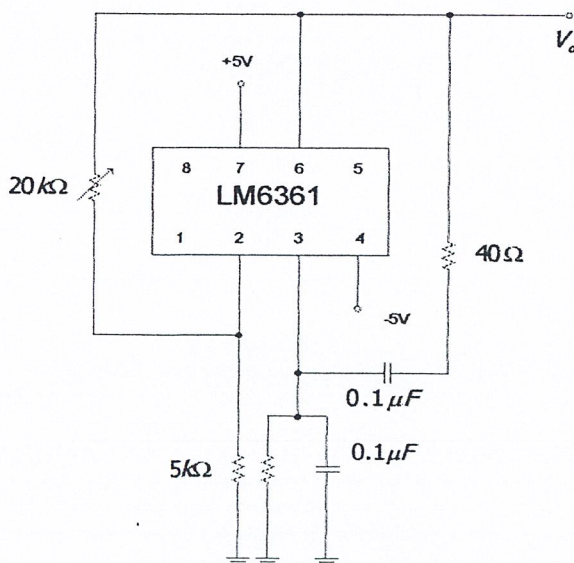
$$\frac{R_2}{R_1} = 2$$

จึงเลือกให้ ตัวต้านทาน  $R_1 = 5k\Omega$

และ ให้ตัวต้านทาน  $R_2$  เป็น ตัวความต้านทานปรับค่าได้ขนาด  $20k\Omega$

เนื่องจากตัวต้านทาน และตัวเก็บประจุแต่ละตัวย่อมมีความคลาดเคลื่อน จึงต้องใช้  $R_2$  เป็น ตัวต้านทานปรับค่าได้เพื่อการปรับความคลาดเคลื่อนเหล่านั้นเพื่อให้เกิดการออสซิลเลต

เมื่อเลือกความต้านทานและตัวเก็บประจุต่างๆ แล้วจึงต่อได้วงจรดังรูปที่ 3.1

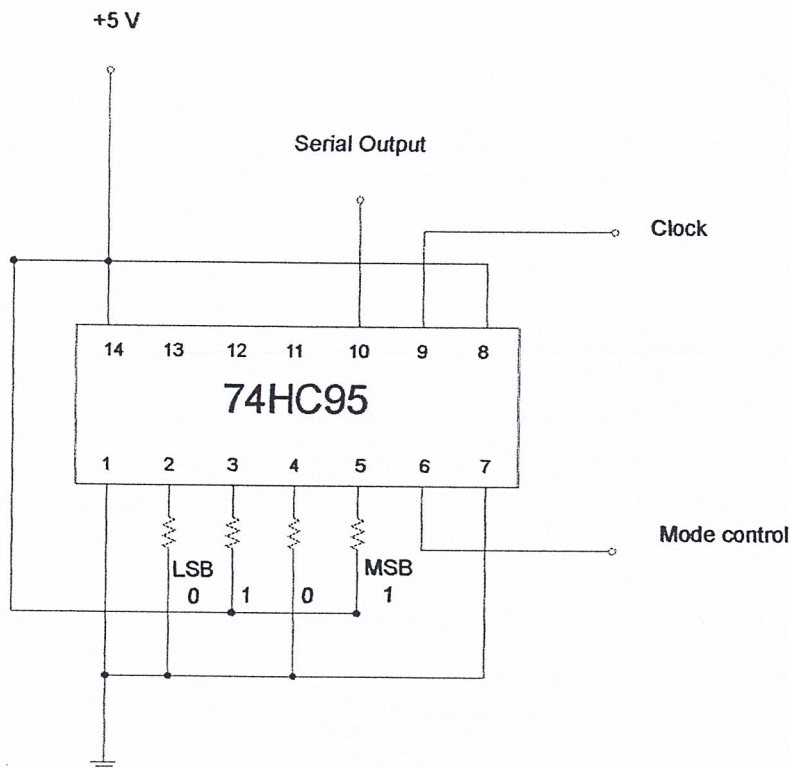


รูปที่ 3.1 รูปแสดงวงจรถ่ายความถี่ 40 กิโลเฮิรตซ์

### 3.1.2 วงจรเข้ารหัสแบบชิฟท์รีจิสเตอร์ ( Shift Register Coder )

ในโครงการนี้เราเลือกใช้ชิฟท์รีจิสเตอร์ เบอร์ 74HC95 มาทำการกำเนิดรหัสบิตดิจิทัล สำหรับชุดตัวส่งแต่ละชุดให้แตกต่างกัน ก่อนจะนำไปคูณกับสัญญาณรูปไซน์ ความถี่ 40 กิโลเฮิรตซ์ แล้วส่งผ่านไปให้ทรานสดิวเซอร์ตัวส่ง เพื่อแปลงสัญญาณไฟฟ้าเป็นคลื่นอัลตราโซนิก เพื่อส่งออกอากาศต่อไป

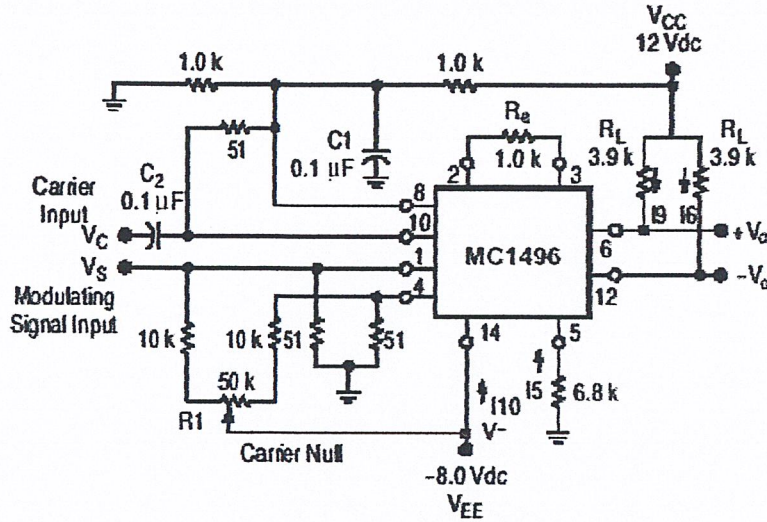
เหตุผลที่ใช้ชิฟท์รีจิสเตอร์มาสร้างรหัสนั้น เพราะสามารถออกแบบสัญญาณเป็นรหัสที่เราต้องการได้โดยง่าย และสัญญาณที่เราต้องการนั้นเป็นค่ารหัสที่ตายตัวอยู่แล้วโดยวิธีการที่เราจะป้อนระดับ “1” และ “0” ใต้นั้น จะอาศัยการตัวต้านทานเข้าไฟ +5 โวลต์ และ กราวด์ ทำให้ง่ายต่อการสร้าง ซึ่งต่อออกมาได้รูปวงจรดังนี้



รูปที่ 3.2 รูปแสดงการต่อวงจรเข้ารหัสชิฟท์รีจิสเตอร์

### 3.1.3 วงจรคูณสัญญาณ

ในโครงงานนี้เลือกใช้ IC เบอร์ MC1496 ซึ่งมีวงจรภายในเป็นวงจรคูณแบบกิลเบิร์ต ตามที่ได้พิสูจน์มาแล้วข้างต้น มาต่อเป็นวงจรดังรูปที่ 3.3 ในที่นี้จะนำมาใช้ในวงจรเข้ารหัส (Coder) และวงจรถอดรหัส (Decoder)



รูปที่ 3.3 รูปแสดงการต่อวงจรคูณสัญญาณด้วย IC MC1496

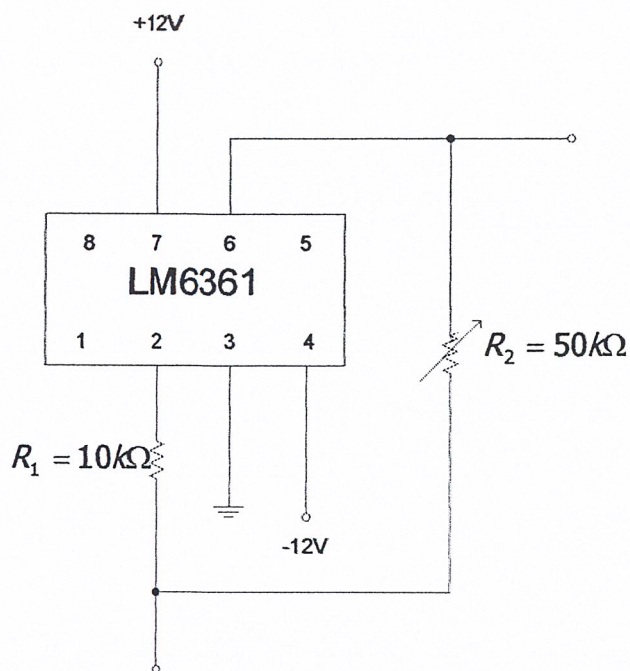
### 3.1.4 วงจรอินเวอร์ตติ้ง แอมพลิฟายเออร์ (Inverting Amplifier)

เราใช้วงจรอินเวอร์ตติ้ง แอมพลิฟายเออร์นี้ มาทำการขยายสัญญาณก่อนส่งเข้าทรานซิสเตอร์ตัวส่งคลื่นอัลตราโซนิค เหตุผลที่ไม่จำเป็นต้องใช้เพาเวอร์ แอมพลิฟายเออร์ (Power Amplifier) เพราะว่าการที่ทรานซิสเตอร์ จะกำเนิดคลื่นอัลตราโซนิคแรงเท่าไร นั้นขึ้นอยู่กับค่าความต่างศักย์ตกคร่อมตัวทรานซิสเตอร์เท่านั้น ไม่ได้ขึ้นอยู่กับกำลังของสัญญาณไฟฟ้าที่เข้ามา

จากสมการที่ (2.36) ในหัวข้อที่ 2.3.4 เราทราบว่า ค่าอัตราขยายจะเท่ากับ

$$Gain = \frac{-R_2}{R_1}$$

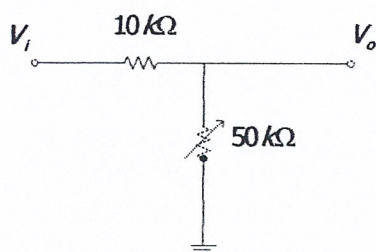
จากการทดลอง เราได้ว่าเอาท์พุทที่ได้จากวงจรคูณ มีขนาดประมาณ  $6 V_{pp}$  แต่เราต้องการส่งสัญญาณไฟฟ้าไปยังทรานซิสเตอร์ด้วยขนาด  $24 V_{pp}$  จึงเลือกใช้  $R_1 = 10k\Omega$  และใช้  $R_2$  เป็นตัวต้านทานปรับค่าได้ขนาด  $50k\Omega$  ได้รูปวงจรดังรูปที่ 3.4



รูปที่ 3.4 รูปแสดงการต่อวงจรขยายแบบอินเวอร์ตติง แอมป์ไฟเออร์

### 3.1.5 วงจรลดทอนสัญญาณ (Attenuator)

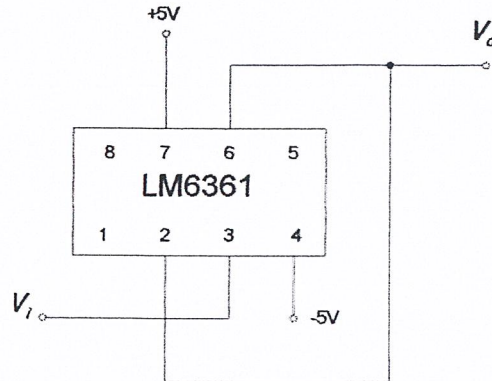
เนื่องจากวงจรรวม IC เบอร์ MC1496 สัญญาณคลื่นรูปไซน์ความถี่ 40 กิโลเฮิรตซ์ที่นำเข้าไป คุณต้องมีขนาดเล็กไม่เกิน  $300\text{ mV}_{pp}$  จึงต้องต่อวงจรลดทอนสัญญาณก่อน ดังรูปที่ 3.5



รูปที่ 3.5 รูปแสดงการต่อวงจรลดทอนสัญญาณ

### 3.1.6 วงจรบัฟเฟอร์ (Buffer)

เนื่องจากสัญญาณคลื่นรูปไซน์ 40 กิโลเฮิร์ตซ์ที่ได้จากวงจรกำเนิดสัญญาณนั้น ต้องนำไปต่อกับวงจรคูณหลายชุด ทำให้ความแรงของสัญญาณที่ได้ตกลง จึงต้องต่อวงจรบัฟเฟอร์ เพื่อป้องกันการตกลงของขนาดสัญญาณ นอกจากนี้แล้ว เรายังนำวงจรนี้มาต่อก่อนเข้าทรานซิสเตอร์อีกด้วย เพื่อคงขนาดของสัญญาณเช่นเดียวกัน ซึ่งจะแสดงดังรูปที่ 3.6



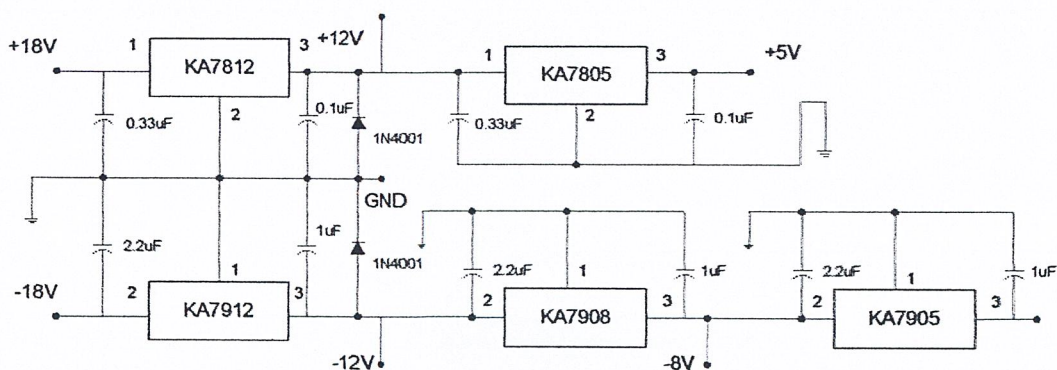
รูปที่ 3.6 รูปแสดงการต่อวงจรบัฟเฟอร์ด้วยออป-แอมป์ เบอร์ LM6361

### 3.1.7 วงจรแปลงสัญญาณไฟ (Voltage Regulator)

การต่อวงจรต่างๆในระบบนั้นจำเป็นจะต้องใช้ไฟเลี้ยงหลายระดับด้วยกัน ดังนั้นจึงต้องทำการแปลงระดับไฟก่อน เพื่อให้เลือกใช้ไฟเลี้ยงในวงจรได้อย่างเหมาะสม ในที่นี้เราจะใช้การแปลงระดับไฟฟ้าออกเป็นระบบดังต่อไปนี้

$$+18\text{ V} \rightarrow +12\text{ V} \rightarrow +5\text{ V}$$

$$-18\text{ V} \rightarrow -12\text{ V} \rightarrow -8\text{ V} \rightarrow -5\text{ V}$$



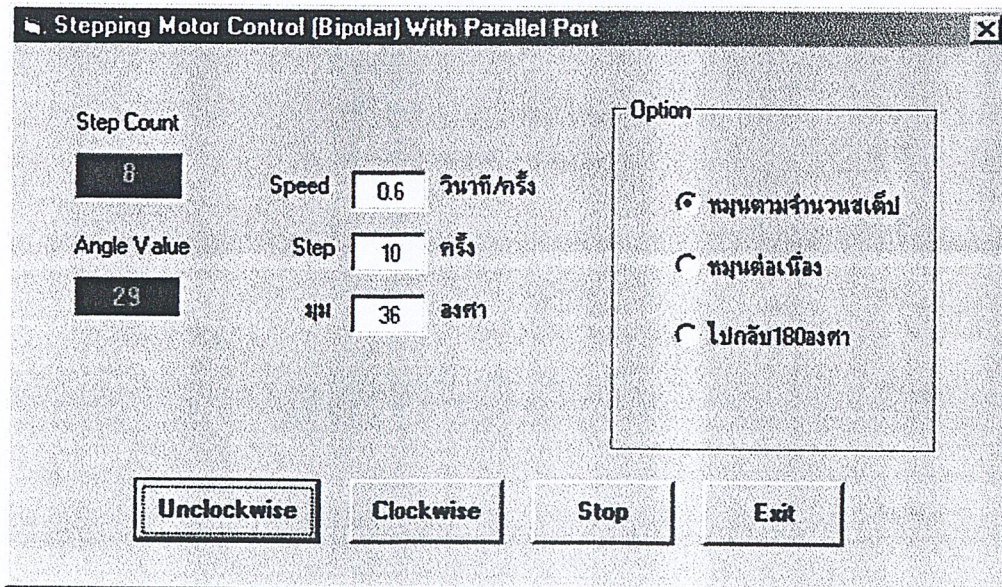
รูปที่ 3.7 แสดงการต่อตัวแปลงระดับสัญญาณไฟ (Voltage Regulator)

3.1.8 วงจรขับสเตปป์มอเตอร์

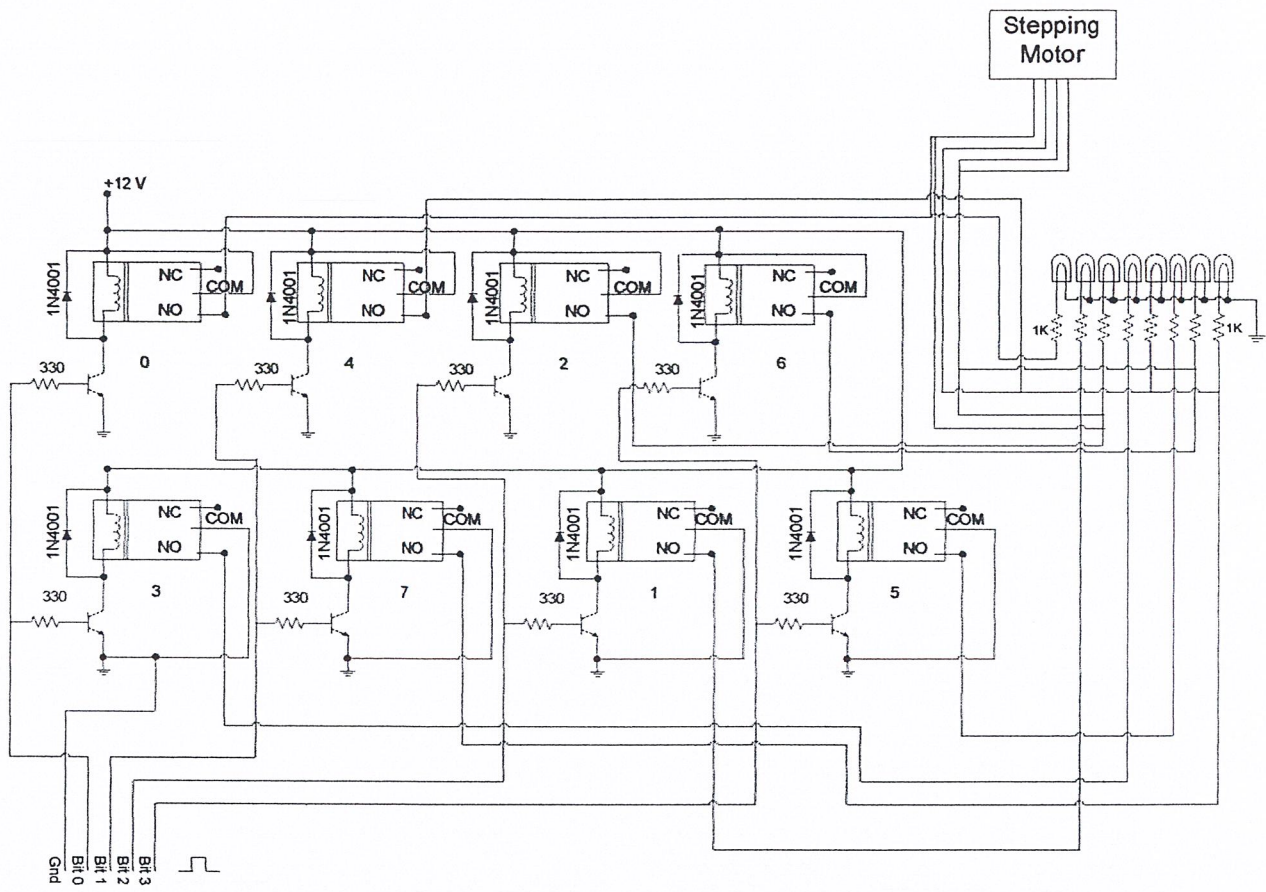
การบังคับการหมุนของสเตปป์มอเตอร์นั้น ในที่นี้เราได้เลือกใช้คอมพิวเตอร์ในการควบคุม โดยมาจากพอร์ตขนาน เข้าที่ขาอินพุท ของวงจรถานซิสเตอร์ไคร์เวอร์ เพื่อจะนำสัญญาณไปทริกขาของรีเลย์ให้ทำงานจากนั้น จึงจะคอยสวิทช์และจ่ายไฟ 12 โวลต์ ของแหล่งจ่าย ไปให้กับขั้วของสเตปป์มอเตอร์ในแต่ละขั้วซึ่งจะทำงานเรียงตามลำดับหรือในที่นี้เราเขียนโดยใช้ โปรแกรมสั่งให้พอร์ตขนาน ส่งค่าออกไปที่พอร์ตตั้งแต่ บิต 1 - 4 เป็น ไปตามตารางแสดงการขับสเตปป์มอเตอร์แบบยูนิโพลาร์ หมุนสเตป ๓.6 องศา (Pseudo code ดังภาคผนวก) ควบคุมตามตาราง

Clockwise	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
1	1			1				
2					1			1
3		1	1					
4						1	1	
Unclockwise	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
1		1	1					
2					1			1
3	1			1				
4						1	1	

ตารางที่ 3.1 ตารางแสดงสเตปป์การควบคุมสเตปป์มอเตอร์ แบบไบโพลาร์



รูปที่ 3.8 แสดงการทำงานของโปรแกรมส่วนควบคุมการหมุนของสเตปป์มอเตอร์

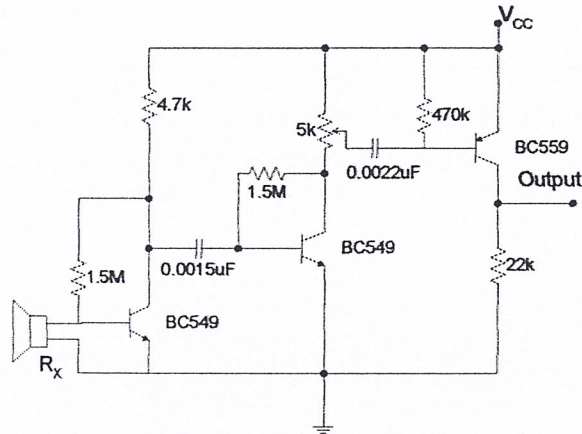


รูปที่ 3.9 วงจรขับสเตปป์มอเตอร์ผ่าน พอร์คขนานด้วยคอมพิวเคอร์

3.2 วงจรส่วนภาครับ

3.2.1 การสร้างวงจรพรีแอมป์ภาครับ (Pre-Amplifier)

จากที่กล่าวมาแล้วในหัวข้อที่ 2.3.7 เราควรเลือกให้  $R_B$  มีค่ามากกว่า  $R_C$  มากๆ เพื่อง่ายต่อการหาอัตราขยายในที่นี้จึงเลือกใช้  $R_B = 1.5$  เมกกะโอห์ม ซึ่งมากกว่า  $R_C$  มากๆ ดังรูปที่ 3.10



รูปที่ 3.10 การต่อตัวรับด้วยทรานซิสเตอร์แบบคอมมอนอีมิเตอร์

3.2.2 การสร้างวงจรแบนด์พาสฟิลเตอร์ (Bandpass Filter)

จากสมการที่ (2.42) ในหัวข้อที่ 2.3.8

$$\omega_1 \omega_4 = \omega_2 \omega_3 = \omega_0^2$$

เราเลือกให้  $\omega_0 = 38.73$  kHz

$$\omega_1 = 30 \text{ kHz}$$

$$\omega_2 = 50 \text{ kHz}$$

$$\omega_3 = 21.43 \text{ kHz}$$

$$\omega_4 = 70 \text{ kHz}$$

จาก (2.43) และ (2.44) จะได้

$$\Omega_p = 1$$

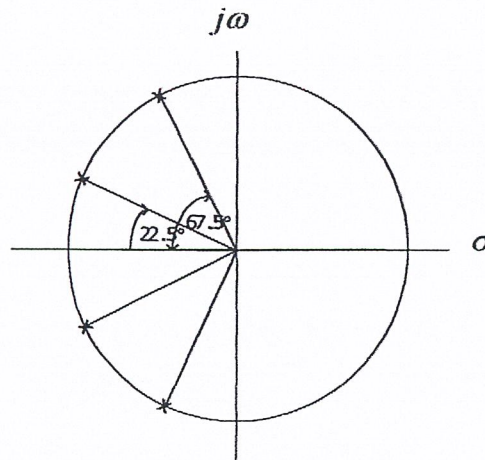
$$\Omega_s = 2.4285$$

จาก (2.45) และ (2.46) จะได้

$$n = 3.77 \approx 4$$

$$\Omega_0 = 1.9365$$

เมื่อทราบ  $n$  แล้วจะได้ตำแหน่งโพลดังนี้



รูปที่ 3.11 รูปแสดงตำแหน่งโพลของวงจรแบนด์พาสฟิลเตอร์

แบ่งออกเป็น 2 กรณี คือ

Case 1

pole อยู่ที่  $-1.2017 \pm j0.4978$

$$\Sigma = 1.2017, \Omega = 0.4978$$

จาก Geffe's Algorithm ในหัวข้อ 2.3.8 จะได้

$$q_c = 1.9365$$

$$C = 1.6919$$

$$D = 1.2411$$

$$E = 4.4512$$

$$G = 3.6948$$

$$Q = 1.6261$$

$$K = 1.0091$$

$$W = 1.1442$$

$$f_{01} = 44.3132 \text{ kHz}$$

$$f_{02} = 33.8503 \text{ kHz}$$

Case 2

pole อยู่ที่  $-0.4978 \pm j1.2017$

$$\Sigma = 0.4978, \Omega = 1.2017$$

จาก Geffe's Algorithm ในหัวข้อ 2.3.8 จะได้

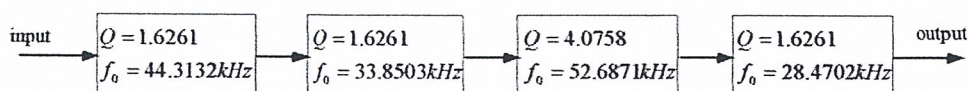
$$q_c = 1.9365$$

$$C = 1.6919$$

$$D = 0.5141$$

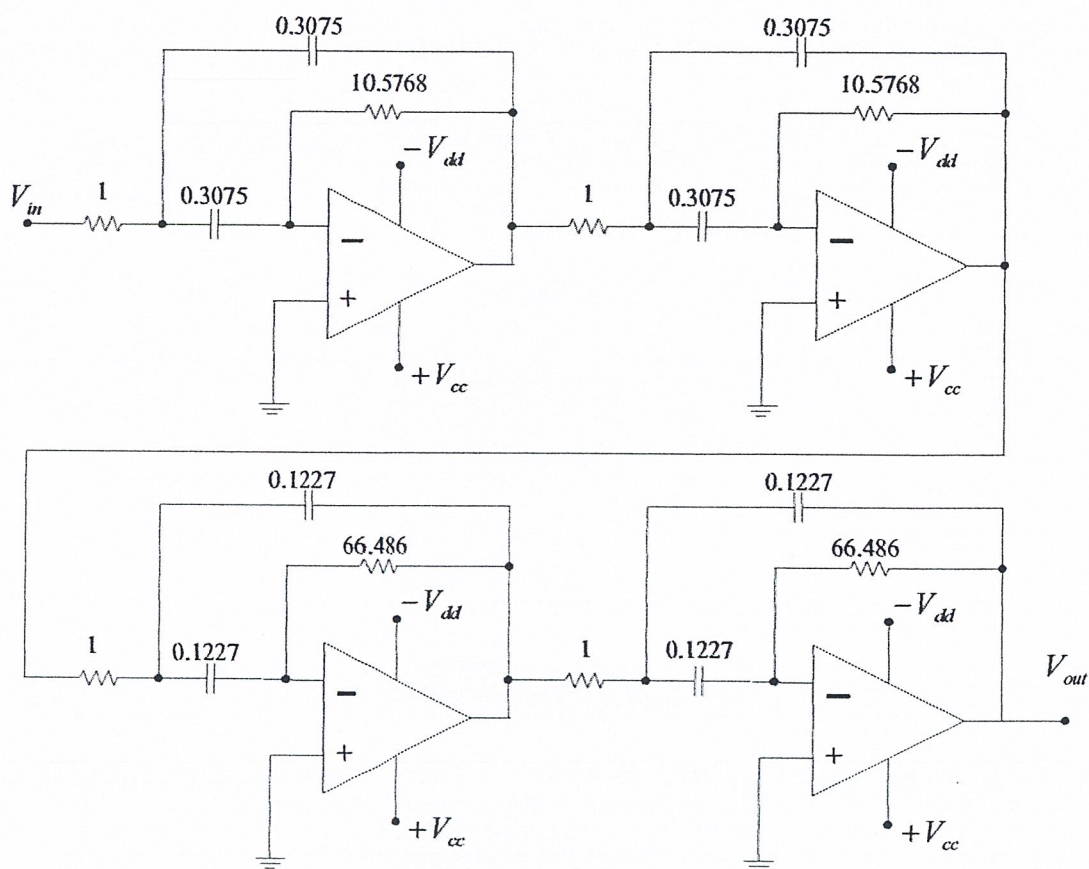
$$\begin{aligned}
 E &= 4.4512 \\
 G &= 4.3308 \\
 Q &= 4.0758 \\
 K &= 4.0758 \\
 W &= 1.3604 \\
 f_{03} &= 52.6871 \text{ kHz} \\
 f_{04} &= 28.4701 \text{ kHz}
 \end{aligned}$$

นั่นหมายถึงเราจะต้องออกแบบฟิลเตอร์ 4 สเตจ โดยแต่ละสเตจมีค่า  $Q$  และ  $f_0$  ดังรูปที่ 3.12



รูปที่ 3.12 แสดงคุณสมบัติแต่ละสเตจของฟิลเตอร์

และจากรูปที่ 3.12 เราจะได้วงจรที่ยังไม่ผ่านการ scaling ดังรูปที่ 3.13



รูปที่ 3.13 แสดงการต่อวงจรฟิลเตอร์ตามที้ออกแบบได้

จากสมการที่ (2.48) เราจะคำนวณค่าอัตรการขยายในแต่ละสเตจที่ความถี่ 40 กิโลเฮิรตซ์ ดังนี้

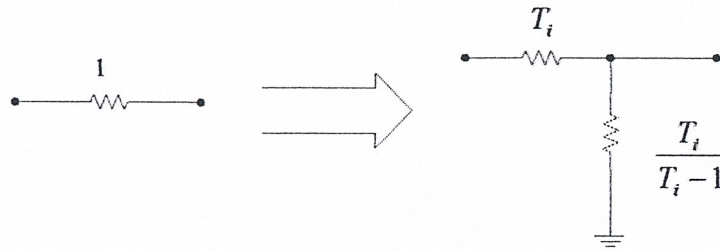
$$|T_1(j\omega)| = 4.84179$$

$$|T_2(j\omega)| = 4.84185$$

$$|T_3(j\omega)| = 12.1358$$

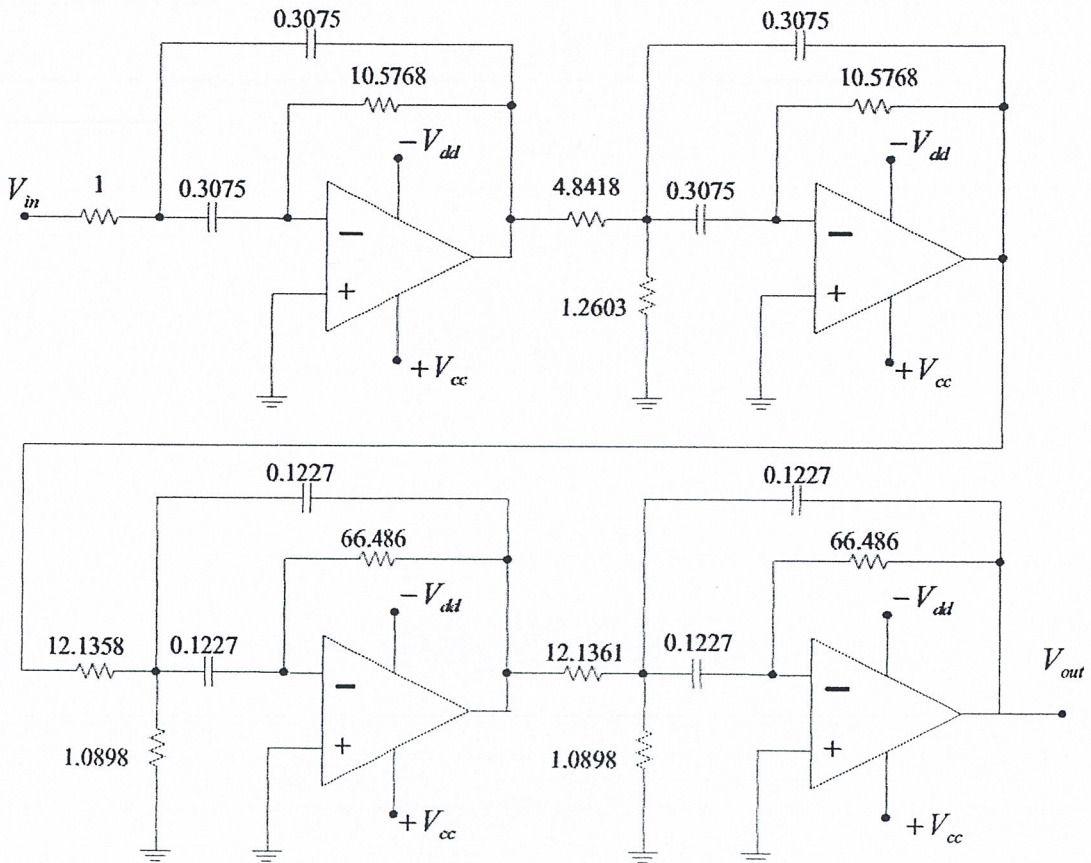
$$|T_4(j\omega)| = 12.1361$$

แต่เนื่องจากเราต้องการค่า อัตรการขยาย ประมาณ 5 เท่า จึงลดทอนสัญญาณลงมาจากสเตจ ยกเว้นสเตจแรกโดยการเปลี่ยน R แต่ละสเตจ เป็นลักษณะดังรูป



รูปที่ 3.14 แสดงการเปลี่ยน R แต่ละสเตจเพื่อปรับค่าอัตรการขยาย

จึงได้วงจรดังรูปที่



รูปที่ 3.15 แสดงการต่อวงจรฟิลเตอร์ที่ผ่านการปรับอัตรการขยายแล้ว

จากนั้นทำการ scaling โดยการใช้ความสัมพันธ์ตามสมการที่ (2.49),(2.50),(2.51)

โดยเลือกค่า  $k_m, k_f$  ในแต่ละสเตจ ดังนี้

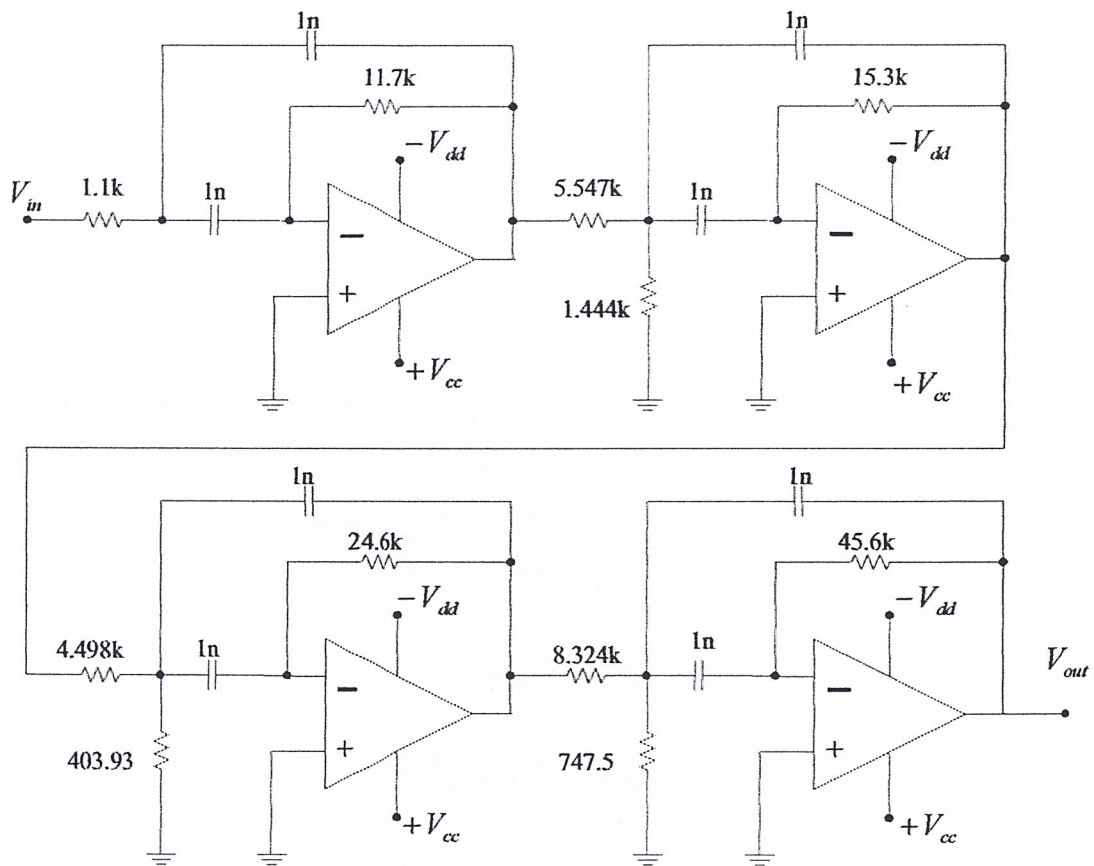
Stage 1  $k_m = 1104.4146$   $k_f = 278428.0472$

Stage 2  $k_m = 1445.7817$   $k_f = 212687.7076$

Stage 3  $k_m = 370.6469$   $k_f = 331042.8126$

Stage 4  $k_m = 685.9211$   $k_f = 178883.5423$

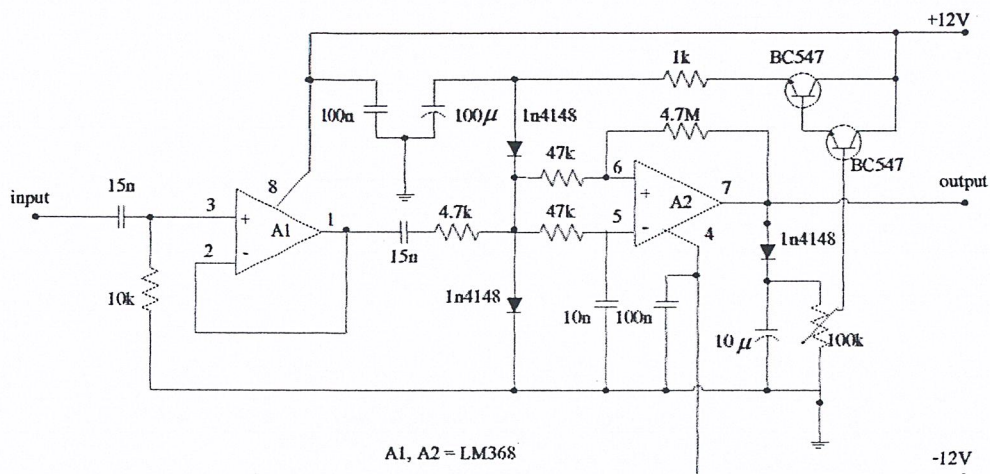
จะได้วงจร บัตเตอร์เวิร์ทซ์แบนด์พาสฟิลเตอร์ ดังนี้



รูปที่ 3.16 แสดงการต่อวงจรฟิลเตอร์ที่ผ่านการ scaling แล้ว

### 3.2.3 การสร้างวงจร Automatic Gain Control (AGC)

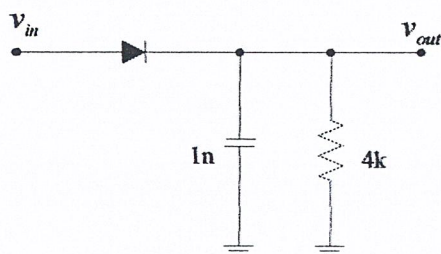
ตัวอย่างวงจร AGC ลักษณะดังรูป 3.17 ซึ่งวงจร AGC นี้ จะทำให้ค่าเอาต์พุตคงที่ที่ระดับประมาณ 3 โวลต์ แต่รูปสัญญาณจะเกิดการบิดเบี้ยว (distortion) กลายเป็นสัญญาณลักษณะสามเหลี่ยม จึงต้องนำไปผ่านวงจรกรองความถี่ในขั้นต่อไป



รูปที่ 3.17 แสดงการต่อวงจร AGC

### 3.2.4 การสร้างวงจรเอนเวลโลป ดีเทคเตอร์ (Envelope detector)

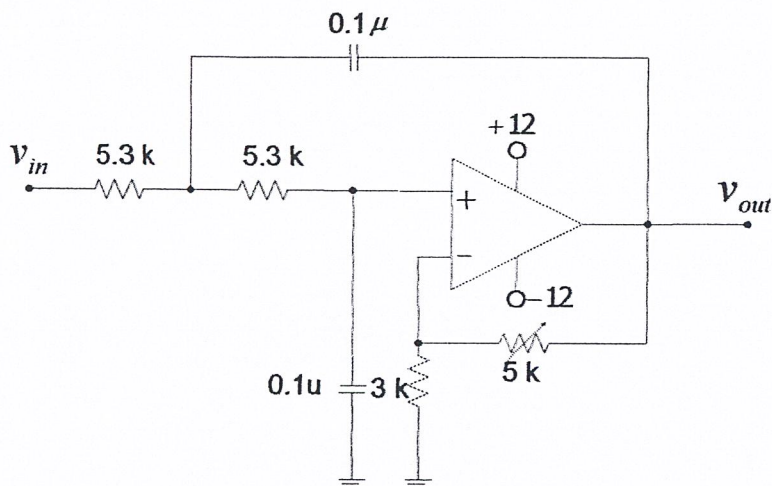
จากรูปที่ 2.24 และความสัมพันธ์ที่ (2.52) เมื่อ  $f_c = 40 \text{ kHz}$  จะได้วงจรดังรูปที่ 3.18



รูปที่ 3.18 แสดงการต่อวงจรเอนเวลโลป ดีเทคเตอร์

### 3.2.5 วงจรกรองความถี่ต่ำผ่าน (Low Pass Filter, LPF)

จากสมการ(2.57) เมื่อกำหนดให้มีความถี่คัตออฟ = 300 กิโลเฮิรตซ์ เนื่องจากสัญญาณรบกวนที่มีความกว้าง 1 บิต เท่ากับ 2.5 มิลลิวินาที หรือความถี่เท่ากับ 250 เฮิรตซ์ และเมื่อกำหนดให้ C มีค่าเท่ากับ  $0.1 \mu F$  จะได้  $R = 5.3 k\Omega$  ส่วน  $R_B$  และ  $R_A$  เราเลือกให้เป็น R ปรับค่าได้ ดังรูป เพื่อปรับขนาดของอัตราขยาย ให้มีเอาต์พุตที่มีระดับไม่เกิน 5 โวลต์ เพื่อเข้าสู่วงจร A/D ต่อไป



รูปที่ 3.19 วงจรกรองความถี่ต่ำผ่านที่ออกแบบแล้ว

### 3.2.6 การควบคุมด้วยไมโครคอนโทรลเลอร์ MCS-51

หน้าที่ของ MCS-51

เราใช้ไมโครคอนโทรลเลอร์ทำหน้าที่ 4 อย่างคือ

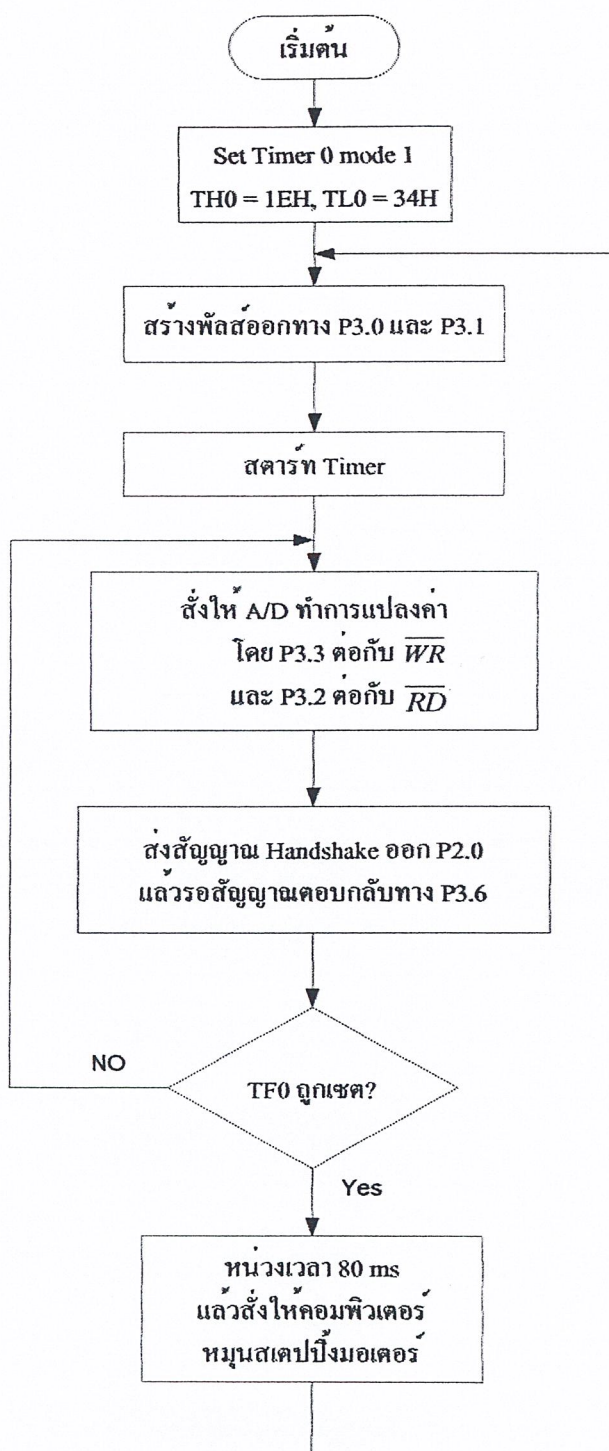
1. สั่งให้คอมพิวเตอร์ทำการหมุนสแตปปีงมอเตอร์
2. สร้างพัลส์ไปให้แก่วงจรซีฟทีรีจิสเตอร์เพื่อสร้างรหัสของตัวส่ง
3. ควบคุมการทำงานของ A/D ให้แปลงสัญญาณ
4. สั่งให้คอมพิวเตอร์รับสัญญาณดิจิทัลจาก A/D ไปประมวลผล

ขั้นตอนการทำงานของ MCS-51

1. initial ไทเมอร์ 0 โหมด 1 ให้ TH0 = 1EH และ TL0 = 34H
2. สร้างพัลส์ขนาด 1 ms ออกทาง P3.0 และ P3.1 เพื่อควบคุมวงจซีฟทีรีจิสเตอร์
3. ให้ไทเมอร์ 0 เริ่มทำการนับเวลา เพื่อนับเวลาทั้งหมดที่ A/D จะต้องทำการแปลงข้อมูล
4. สั่งให้ A/D ทำการแปลงค่าเพื่อส่งไปยังคอมพิวเตอร์โดยให้ค่าลอจิก "0" และ "1" แก่ P3.3 ที่ต่อกับ  $\overline{WR}$  ของ A/D แล้วทำการหน่วงเวลาไป  $110 \mu s$  เพื่อรอให้ A/D แปลงค่าให้เสร็จ แล้วจึงให้ลอจิก "0" P3.2 ต่อกับ  $\overline{RD}$  ของ A/D เพื่อให้ A/D ค่าที่แปลงได้ออกมา
5. ส่งสัญญาณไปบอกให้คอมพิวเตอร์ทำการรับค่าที่แปลงเป็นดิจิทัลไปเก็บทาง P2.0 แล้วรอรับสัญญาณตอบรับเมื่อคอมพิวเตอร์ทำการเก็บค่าเสร็จแล้วทาง P3.6
6. ตรวจสอบว่า บิต TFO เซตหรือยังถ้ายังจะกลับไปทำในขั้นตอนที่ 4 ใหม่ ถ้าถูกเซตแล้วจะทำการเคลียร์ TR0 และ TF0 แล้วหน่วงเวลาเพื่อรอการประมวลผลของคอมพิวเตอร์ แล้วจึงสั่งให้ สแตปปีงมอเตอร์ทำการหมุน ก่อนจะกลับไปทำในขั้นตอนที่ 2 ต่อไป

จะได้โฟลวชาร์ตการทำงานของ MCS-51 เป็นดังนี้

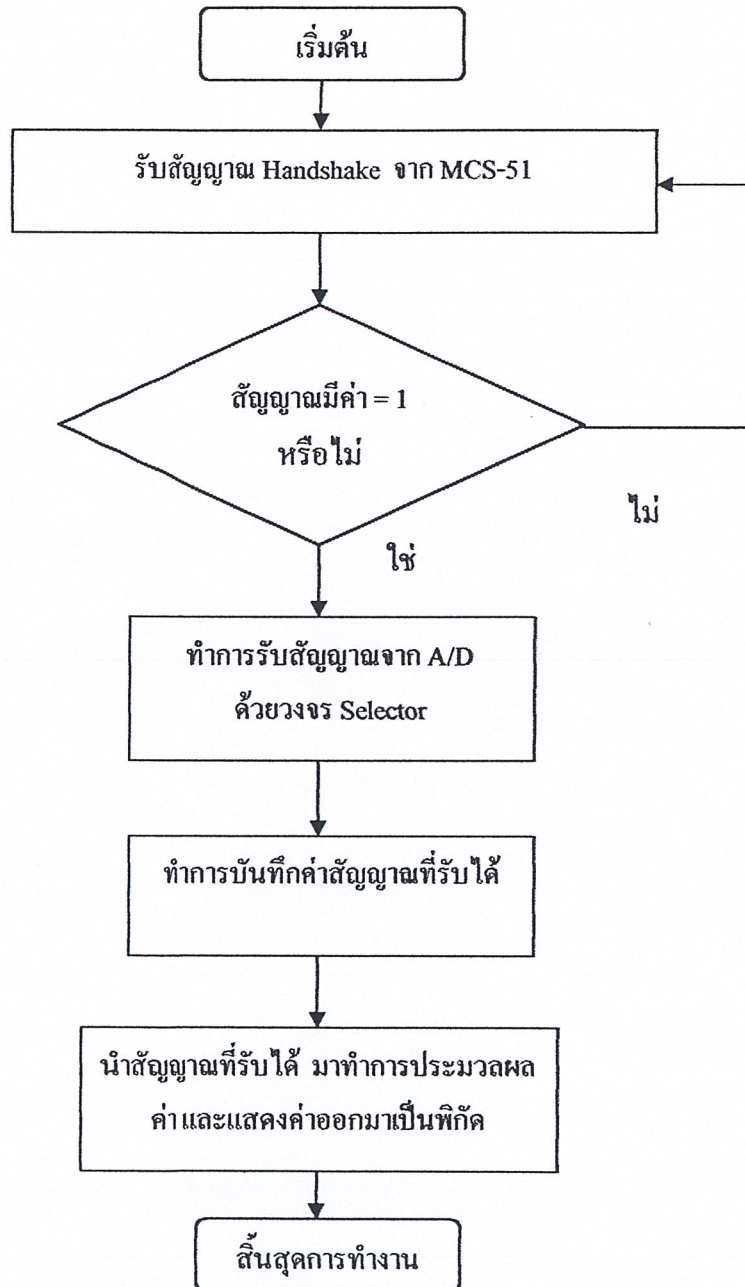
## Flowchart



รูปที่ 3.20 โฟลวชาร์ตแสดงการทำงานของโปรแกรม MCS-51

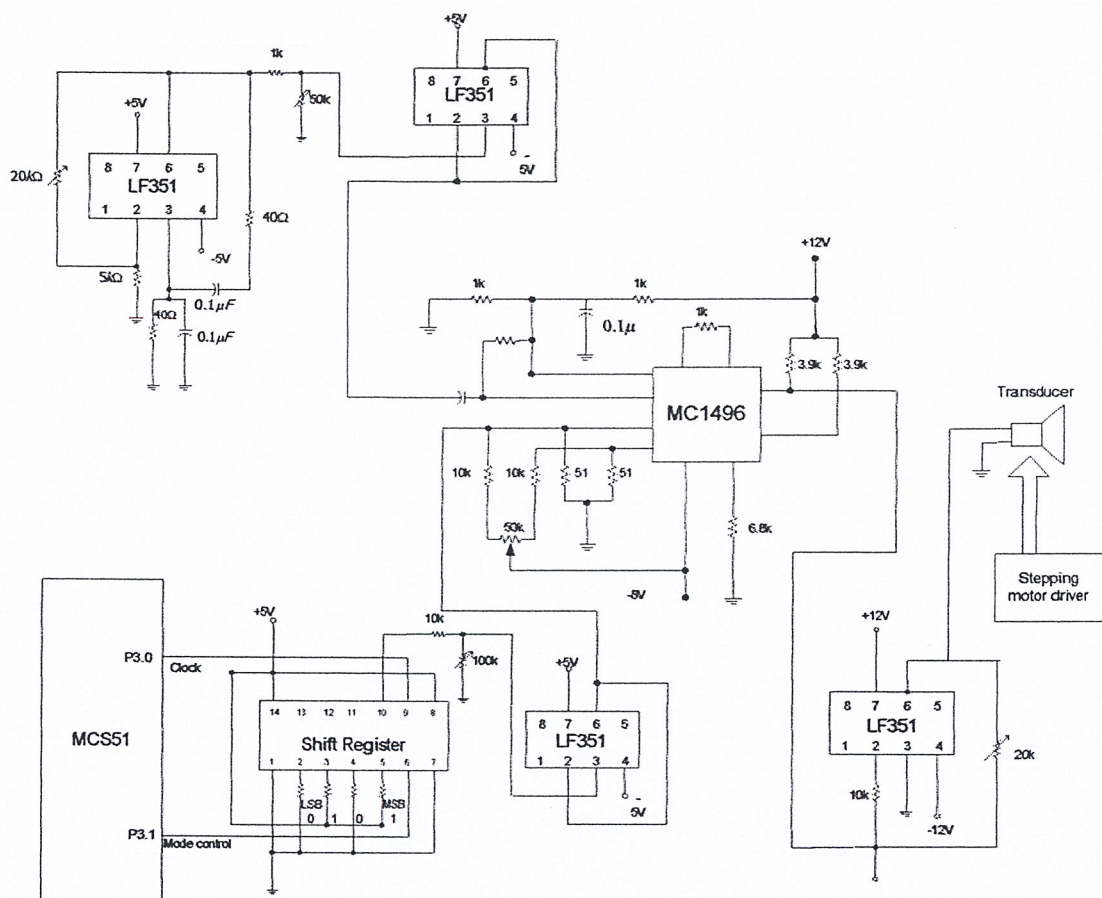
### 3.2.7 การเขียนโปรแกรมการทำงานในภาคแสดงผล

การเขียนวงจรการทำงาน ในภาคแสดงผลนั้น ได้เลือกใช้การเขียนด้วยภาษา Visual Basic เนื่องจากที่เป็นภาษาที่ใกล้เคียงกับภาษามนุษย์ ทำให้ทำการศึกษาและทำความเข้าใจได้ง่าย และยัง สามารถทำงานได้ด้วยอินเตอร์เฟซที่สวยงาม นอกจากนี้ยังสามารถทำงานได้บนระบบปฏิบัติการ Windows ทำให้เราเลือกการเขียน โปรแกรมด้วย ภาษา Visual Basic มาใช้ การเขียนโปรแกรมนั้นแสดงให้เห็นดัง โฟลวชาร์ตต่อไปนี้



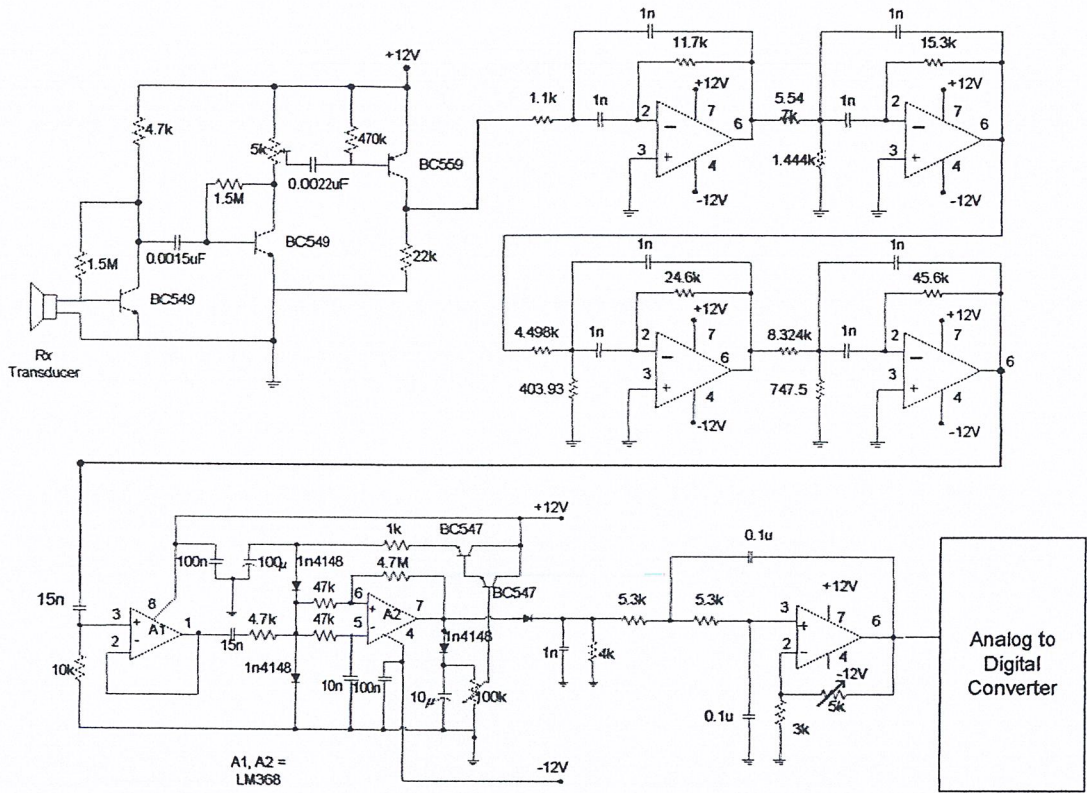
รูปที่ 3.21 โฟลวชาร์ตแสดงการทำงานของโปรแกรมรับ-ส่ง 40 Bits

### 3.2.8 การต่อวงจรรวม วงจรส่วนภาคส่ง



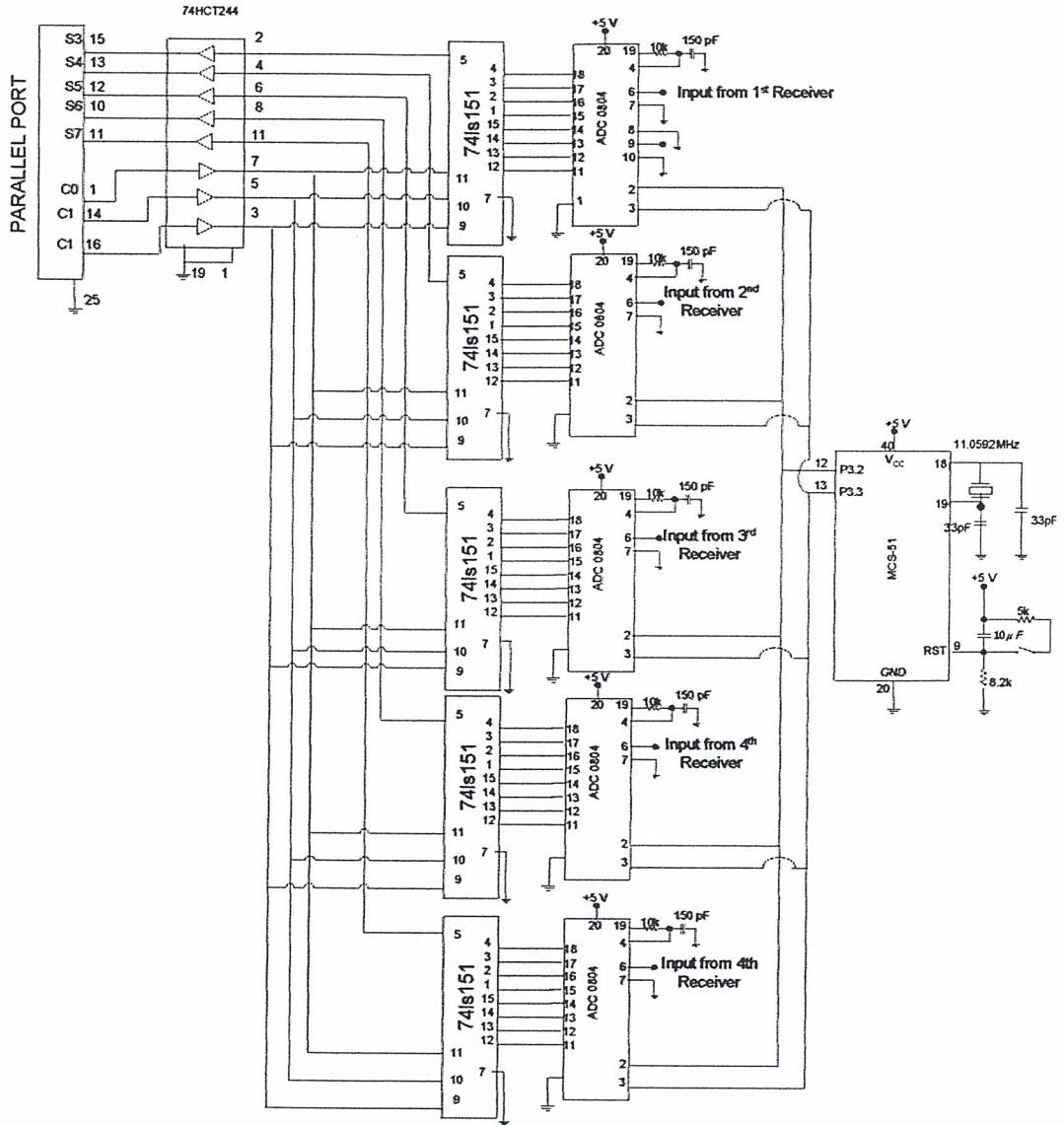
รูปที่ 3.22 แสดงการต่อวงจรรวมในส่วนภาคส่ง

วงจรส่วนภาครับ



รูปที่ 3.23 แสดงการต่อวงจรรวมในส่วนภาครับ ชุด

วงจรส่วนภาคอินเทอร์เฟส



รูปที่ 3.24 แสดงวิธีการต่อวงจรเพื่ออ่านค่าข้อมูล 40 บิต จาก A/D มาเข้าพอร์ทขนาน ร่วมกับ MCS-51

## บทที่ 4

### การทดลองและผลการทดลอง

#### 4.1 การทดลองและผลการทดลอง

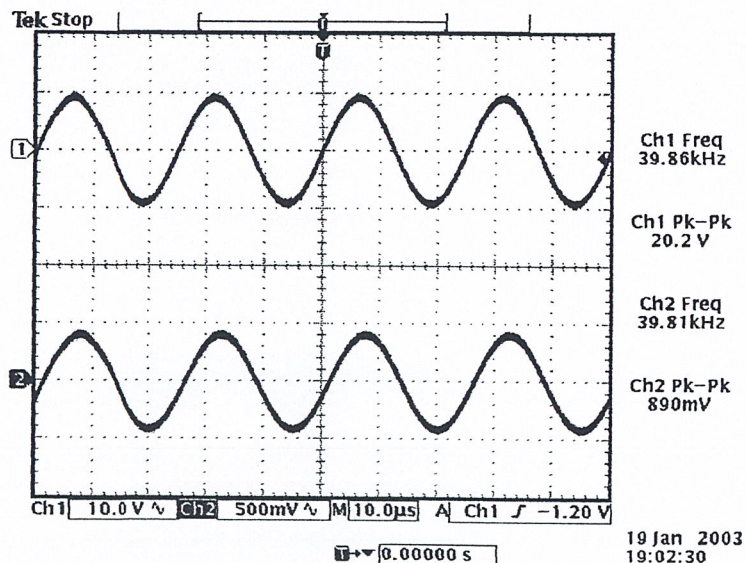
##### ผลการทดลองในภาคส่ง

##### 4.1.1 ทรานซิสเวออร์ ตัวส่ง และ ตัวรับ

###### วิธีการทดลอง

1. ป้อนสัญญาณรูปไซน์ จากฟังก์ชันเจนเนอเรเตอร์ (Function Generator) ขนาด  $9 V_{pp}$  เข้าทรานซิสเวออร์ตัวส่งโดยตรง
2. นำออสซิลโลสโคป (Oscilloscope) วัดสัญญาณที่ขาของทรานซิสเวออร์ตัวรับโดยตรง และเนื่องจากทรานซิสเวออร์ตัวรับมีขั้ว ซึ่งขาลบจะมีโลหะหนุนอยู่ ส่วนอีกขาหนึ่งเป็นขาบวก จึงควรจะต้องระวังในจุดนี้ ในส่วนของทรานซิสเวออร์ตัวส่งนั้น สามารถใช้สลับกันได้
3. นำทรานซิสเวออร์ตัวส่งและรับถือห่างกันเป็นระยะประมาณ 30 เซนติเมตร โดยหันทรานซิสเวออร์ตัวส่งกับตัวรับเข้าหากัน ให้ตรงกันมากที่สุด เพื่อให้ได้ไดเรกทีวิตี (Directivity) สูงสุด ควรจะได้ผลการทดลองออกมาดังรูปที่ 4.1
4. นำกรวยมาใส่ที่ตัวรับเพื่อเปรียบเทียบความแรงของสัญญาณ เมื่อมีตัวช่วยรวมโฟกัสของสัญญาณ ซึ่งปรากฏว่ามีสัญญาณที่แรงขึ้นมา ดังรูปที่ 4.2
5. นำตัวส่งมารวมกันเป็นแหล่งกำเนิดเสมือน จำนวน 3 ตัว ส่งสัญญาณมายังตัวรับที่ใส่กรวย เพื่อเปรียบเทียบความแรงของสัญญาณ ปรากฏว่ามีความแรงของสัญญาณแรงขึ้นมาก ซึ่งปรากฏผลออกมาดังรูปที่ 4.3

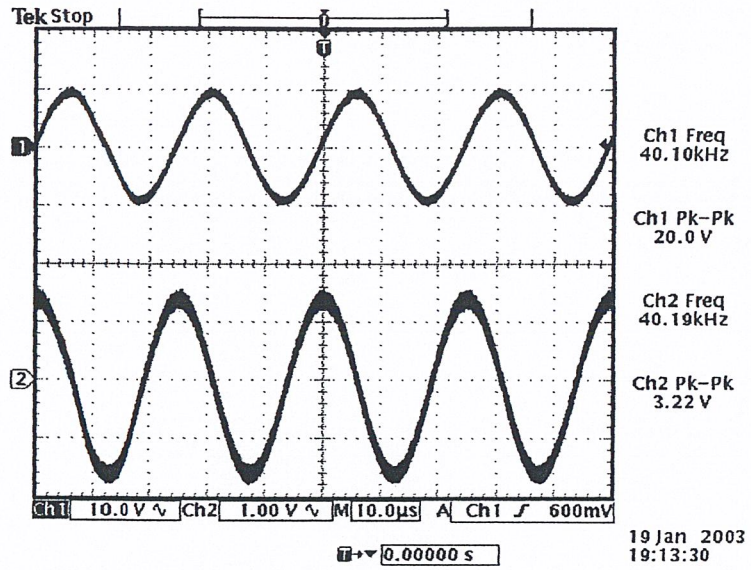
###### ผลการทดลอง



รูปที่ 4.1 แสดงการทดสอบทรานซิสเวออร์ ตัวส่งและตัวรับ ในระยะส่ง 30 เซนติเมตร

Ch1. จับที่ทรานซิสเวออร์ตัวส่ง

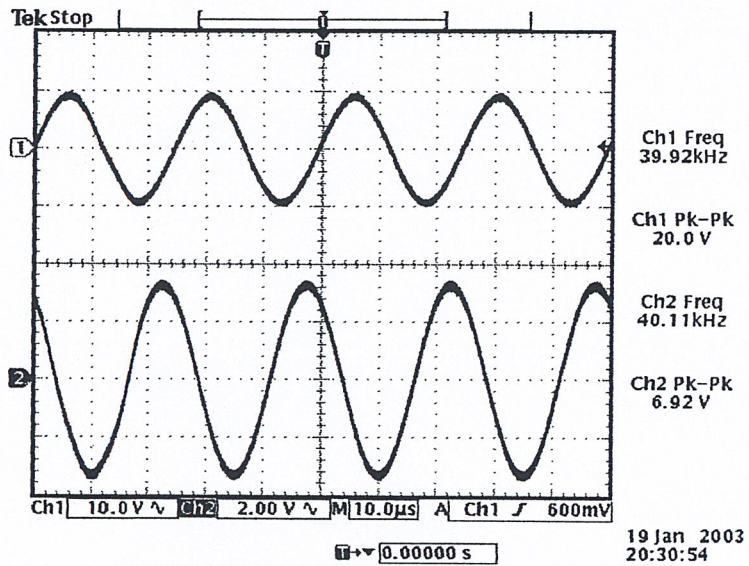
Ch2 จับที่ทรานซิสเวออร์ตัวรับ



รูปที่ 4.2 แสดงการทดสอบทรานสดิวเซอร์ ตัวส่งและตัวรับใส่กรวย ในระยะส่ง 30 เซนติเมตร

Ch1. จับที่ทรานสดิวเซอร์ตัวส่ง

Ch2. จับที่ทรานสดิวเซอร์ตัวรับ



รูปที่ 4.3 แสดงการทดสอบโดยใช้ตัวส่ง 3 ตัวและตัวรับใส่กรวย ในระยะส่ง 30 เซนติเมตร

Ch1. จับที่ทรานสดิวเซอร์ตัวส่ง

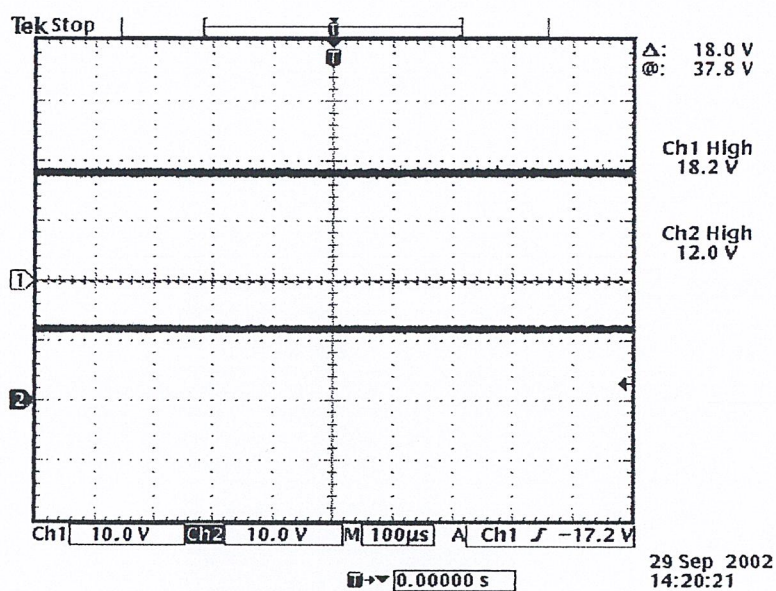
Ch2. จับที่ทรานสดิวเซอร์ตัวรับ

#### 4.1.2 การทดลองตัวแปลงระดับไฟโวลต์เตจเรกกูเรเตอร์

##### วิธีการทดลอง

1. ต่อวงจรดังรูปที่ 3.7 และป้อนไฟ DC +18 (+Vcc) และ -18 (-Vcc) โวลต์ เข้าวงจร
2. วัดเอาต์พุตของวงจรที่ขา 3 ของ KA7812 ควรได้ระดับไฟ DC+12 โวลต์ ดังรูปที่ 4.4
3. วัดเอาต์พุตของวงจรที่ขา 3 ของ KA7912 ควรได้ระดับไฟ DC-12 โวลต์ ดังรูปที่ 4.5
4. วัดเอาต์พุตของวงจรที่ขา 3 ของ KA7908 ควรได้ระดับไฟ DC-8 โวลต์ ดังรูปที่ 4.6
5. วัดเอาต์พุตของวงจรที่ขา 3 ของ KA7805 ควรได้ระดับไฟ DC+5 โวลต์ ดังรูปที่ 4.7
6. วัดเอาต์พุตของวงจรที่ขา 3 ของ KA7905 ควรได้ระดับไฟ DC-5 โวลต์ ดังรูปที่ 4.8

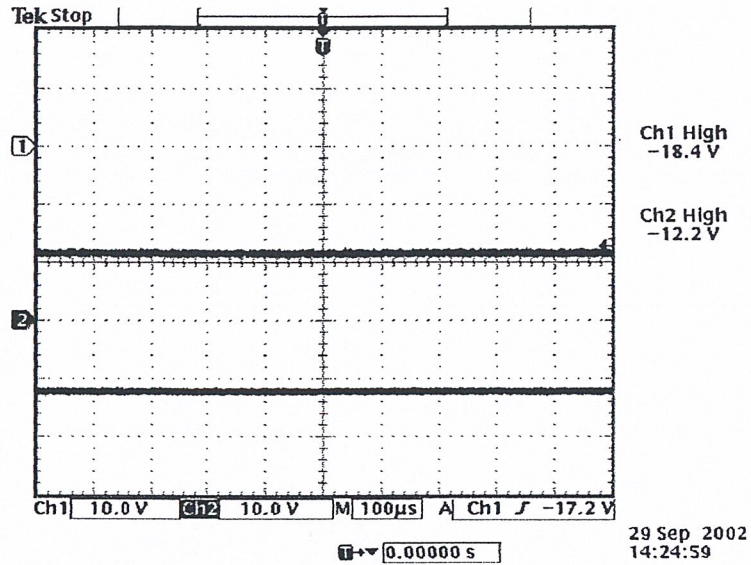
##### ผลการทดลอง



รูปที่ 4.4 แสดงระดับสัญญาณไฟ +12 โวลต์ที่ได้จากวงจรโวลต์เตจเรกกูเรเตอร์

Ch1. จับที่อินพุตไฟ DC +18 V

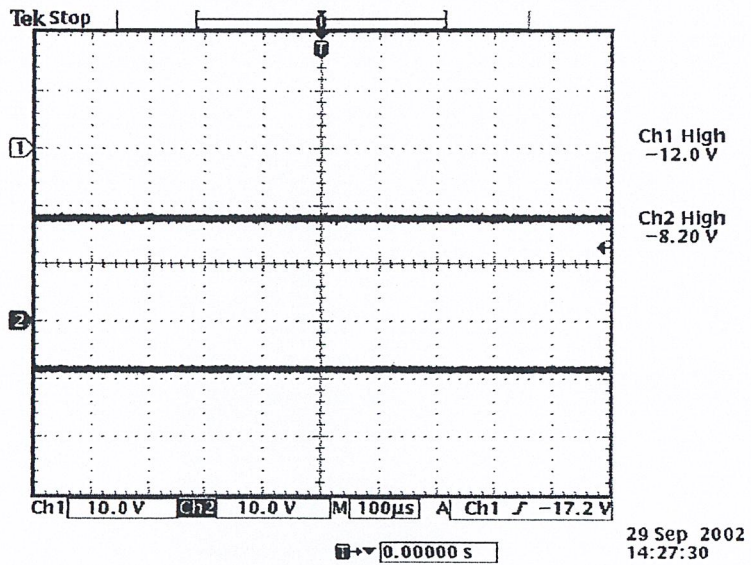
Ch2. จับที่เอาต์พุตของโวลต์เตจเรกกูเรเตอร์ ได้ DC +12V



รูปที่ 4.5 แสดงระดับสัญญาณไฟ -12 โวลต์ที่ได้จากวงจรโวลต์เตจเรกกูเรเตอร์

Ch1. จับที่อินพุตไฟ DC -18 V

Ch2. จับที่เอาต์พุตของโวลเตจเรกกูเรเตอร์ ได้ DC -12V



รูปที่ 4.6 แสดงระดับสัญญาณไฟ -8 โวลต์ที่ได้จากวงจรโวลต์เตจเรกกูเรเตอร์

Ch1. จับที่อินพุตไฟ DC -12 V

Ch2. จับที่เอาต์พุตของโวลเตจเรกกูเรเตอร์ ได้ DC -8 V

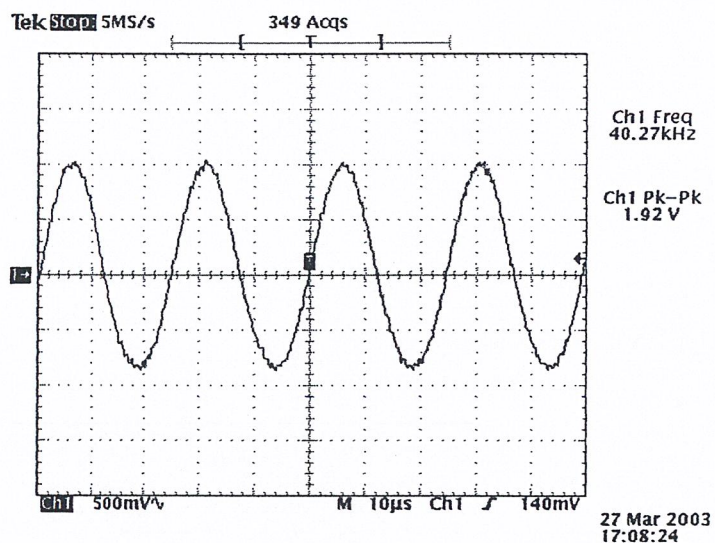


### 4.1.3 วงจรออสซิลเลเตอร์ของภาคส่ง

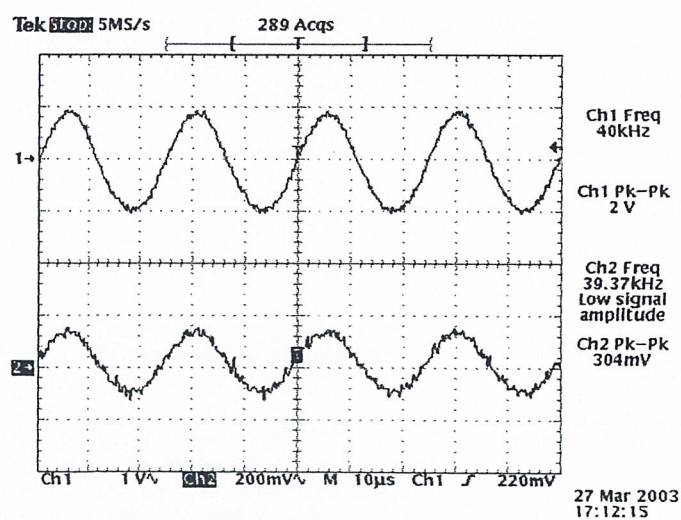
#### วิธีทดลอง

1. ต่อวงจรดังรูปที่ 3.1
2. ใช้ออสซิลโลสโคป วัดค่าสัญญาณเอาต์พุต ซึ่งควรจะได้ลักษณะเป็นสัญญาณ AC รูปไซน์ ความถี่ 40 กิโลเฮิร์ตซ์ ดังรูปที่ 4.9
3. ต่อจากนั้นนำมาต่อกับวงจรลดทอนสัญญาณ เนื่องจากในขั้นต่อไปต้องนำไปต่อกับวงจรคูณ ซึ่งต้องการสัญญาณอินพุต เป็นสัญญาณขนาดเล็ก (Small Signal) สัญญาณเอาต์พุตที่ได้ ออกมา ควรจะได้ลักษณะเป็นสัญญาณ AC รูปไซน์ ความถี่ 40 กิโลเฮิร์ตซ์ ขนาดไม่เกิน 300 มิลลิโวลต์ ดังรูปที่ 4.10

#### ผลการทดลอง



รูปที่ 4.9 แสดงสัญญาณจากออสซิลเลเตอร์ 40 กิโลเฮิร์ตซ์



รูปที่ 4.10 แสดงสัญญาณจากออสซิลเลเตอร์ 40 กิโลเฮิร์ตซ์ หลังผ่านวงจรลดทอนสัญญาณ

Ch1. จับที่สัญญาณไซน์ 40 กิโลเฮิร์ตซ์ที่สร้างขึ้นได้

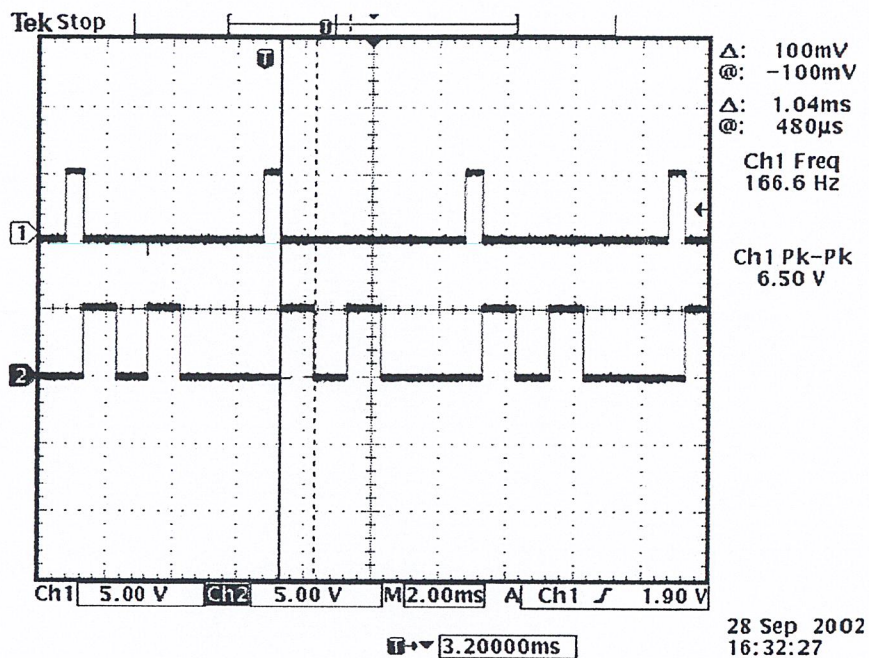
Ch2. จับที่สัญญาณไซน์ 40 กิโลเฮิร์ตซ์ที่ผ่านวงจรลดทอนสัญญาณแล้ว

#### 4.1.4 วงจรเข้ารหัสแบบชิฟท์รีจิสเตอร์

##### วิธีทดลอง

1. ต่อวงจรเข้ารหัสแบบชิฟท์รีจิสเตอร์ดังรูปที่ 3.2 สำหรับรหัส 1010
2. นำสัญญาณพัลส์ที่ได้จาก ขา 1 ของไมโครคอนโทรลเลอร์ มาเข้าขา Mode Control หรือขา 6 ของวงจรเข้ารหัส และชุดสัญญาณพัลส์เพื่อการทดสอบวงจรเข้ารหัสที่ได้จากขา 2 ของไมโครคอนโทรลเลอร์ มาเข้าขา Clock R-Shift หรือขา 9 ของวงจรเข้ารหัส
3. วัดสัญญาณเอาต์พุตที่ได้จาก ขา 10 ของวงจรเข้ารหัส ซึ่งจากกรณีนี้ ควรจะได้สัญญาณออกมาเป็นสัญญาณพัลส์ที่มีระดับเป็น 1001 (ในที่นี้ได้ออกแบบให้สัญญาณ 1 บิตมีความกว้าง 1 มิลลิวินาที) ดังรูปที่ 4.11

##### ผลการทดลอง



รูปที่ 4.11 แสดงสัญญาณที่ผ่านการเข้ารหัส 1010 ด้วยการวัดที่ ขา 10 ของวงจรเข้ารหัส เทียบ  
กราวด์ เทียบ กับสัญญาณพัลส์จากขา 1 ของไมโครคอนโทรลเลอร์

Ch1. สัญญาณพัลส์จากขา 1 ของไมโครคอนโทรลเลอร์

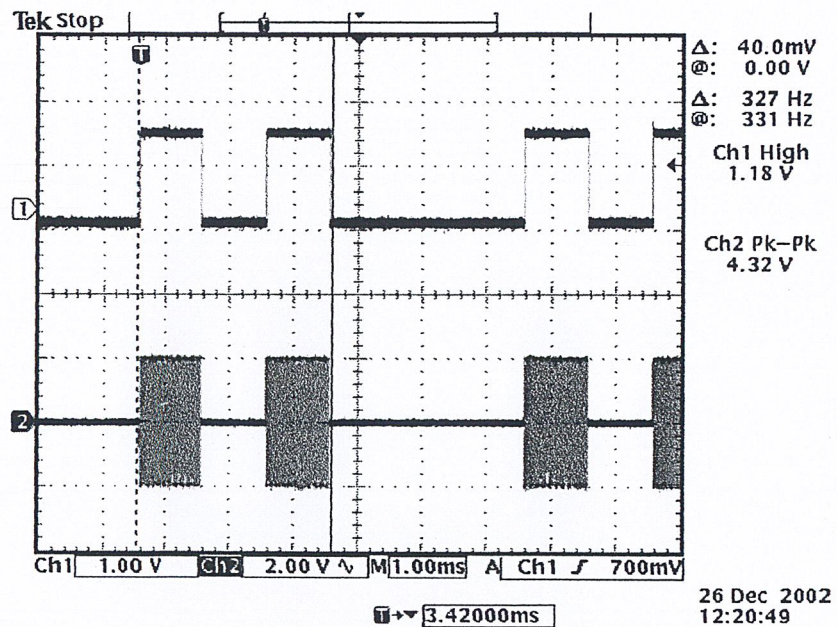
Ch2. สัญญาณรหัสที่ได้

#### 4.1.5 วงจรคูณสัญญาณ

##### วิธีการทดลอง

1. ต่อวงจรคูณสัญญาณดังรูปที่ 3.3
2. นำสัญญาณไซน์ความถี่ 40 กิโลเฮิรตซ์ที่ลดทอนแล้ว จากวงจรกำเนิดสัญญาณในหัวข้อ 4.1.3 มาป้อนเข้าเป็นคลื่นพาส (V<sub>o</sub>) ในวงจรคูณสัญญาณ
3. นำสัญญาณรหัสที่ได้มาจากวงจรเข้ารหัส มาป้อนเป็นสัญญาณข่าวสาร (V<sub>s</sub>) ในวงจรคูณสัญญาณ (ในที่นี้ใช้สัญญาณรหัส 1010)
4. วัดสัญญาณเอาต์พุตที่ได้ ที่ ขา 6 (+V<sub>o</sub>) สัญญาณเอาต์พุตควรมีลักษณะดังรูป 4.12
5. ทำการขยายสัญญาณที่ผ่านการคูณสัญญาณแล้วด้วยวงจรขยายแบบอินเวอร์ตติง แอมป์ตีฟายเออร์ รูปที่ 3.4 เพื่อทำการเพิ่มความแรงของสัญญาณในการส่ง ซึ่งจะได้ผลออกมาดังรูปที่ 4.13

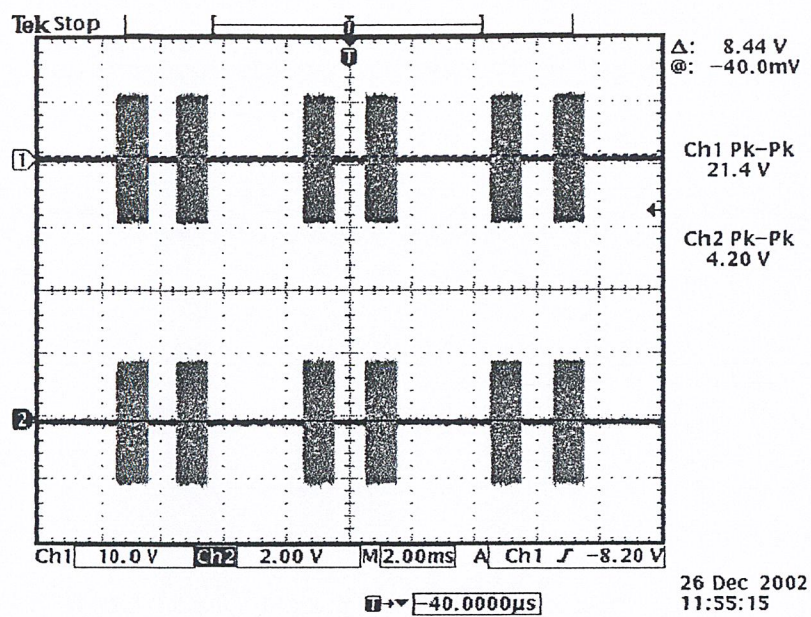
##### ผลการทดลอง



รูปที่ 4.12 แสดงสัญญาณเอาต์พุตของวงจรคูณสัญญาณ

Ch1. สัญญาณรหัสที่นำมาคูณกับสัญญาณความถี่ 40 กิโลเฮิรตซ์

Ch2. สัญญาณที่ผ่านการคูณสัญญาณแล้ว



รูปที่ 4.13 แสดงสัญญาณเอาต์พุตของวงจรคูณสัญญาณ หลังขยายแล้ว

Ch1. สัญญาณเอาต์พุตของวงจรคูณสัญญาณ หลังขยายแล้ว

Ch2. แสดงสัญญาณเอาต์พุตของวงจรคูณสัญญาณ ก่อนขยาย

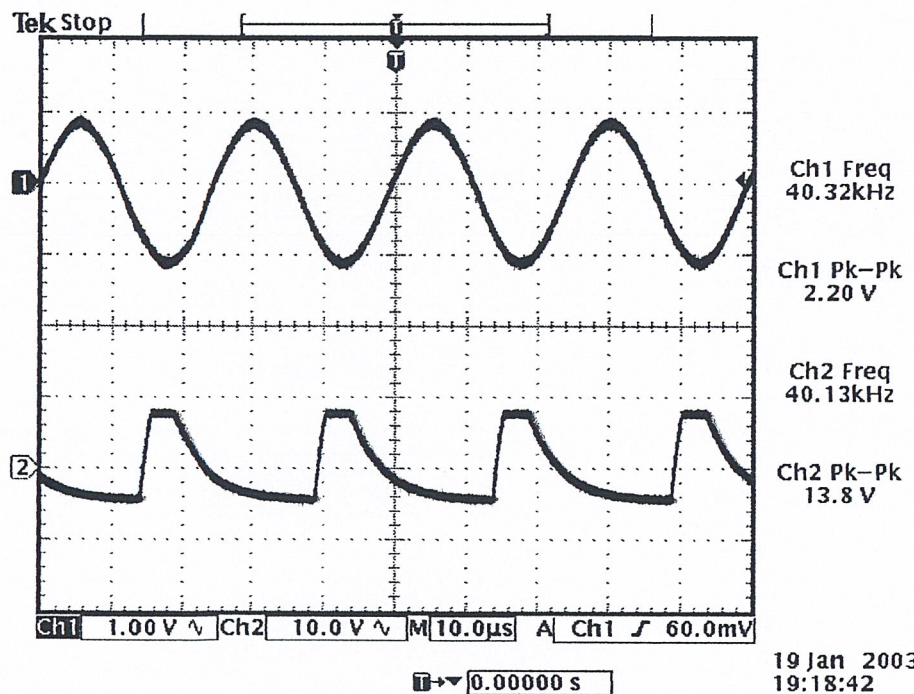
## ผลการทดลองในภาครับ

### 4.1.6 วงจรพีแอมป์

#### วิธีการทดลอง

1. ต่อวงจรพีแอมป์ดังรูปที่ 3.10
2. ทำการทดสอบการทำงานของวงจรพีแอมป์ด้วยการส่งสัญญาณความถี่ 40 กิโลเฮิร์ต ขนาด 20 Vpp จากเครื่องกำเนิดสัญญาณ โดยส่งและรับผ่านทางทรานสดิวเซอร์ที่ระยะ 60 เซนติเมตร แล้ววัดค่าสัญญาณเอาต์พุตที่ได้ ผลที่ได้จะปรากฏดังรูปที่ 4.14
3. ทำการส่งสัญญาณรหัสที่คูณกับสัญญาณความถี่ 40 กิโลเฮิร์ต ขนาด 20 Vpp สะท้อนวัตถุ แล้ววัดความแรงของสัญญาณที่ได้ผ่านทรานสดิวเซอร์ด้วยวงจรพีแอมป์ โดยเลื่อนวัตถุไปที่ระยะต่างๆ ในที่นี้ทดสอบที่วัตถุระยะ 3 เมตร และ 4.5 เมตร ผลที่ได้ปรากฏดังรูปที่ 4.15 และ 4.16ตามลำดับ

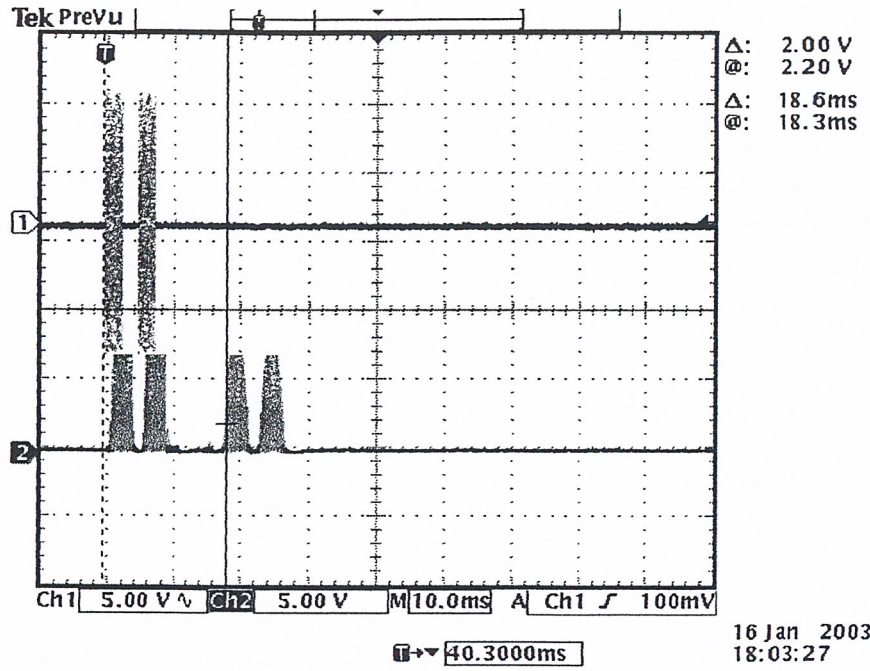
#### ผลการทดลอง



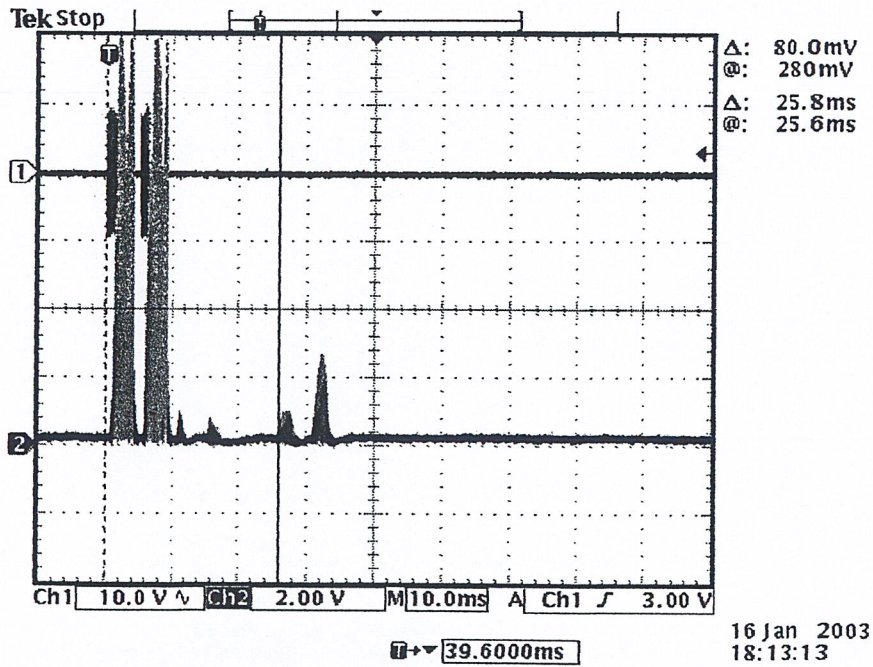
รูปที่ 4.14 รูปแสดงเอาต์พุตของวงจรพีแอมป์ที่วัดได้

Ch1. สัญญาณ 40 กิโลเฮิร์ต ที่ส่ง โดยเครื่องกำเนิดสัญญาณ

Ch2. แสดงสัญญาณเอาต์พุตของวงจรพีแอมป์



รูปที่ 4.15 รูปแสดงเอาต์พุตของวงจรพรีแอมป์ที่วัดได้ วัดดูที่ระยะ 3 เมตร



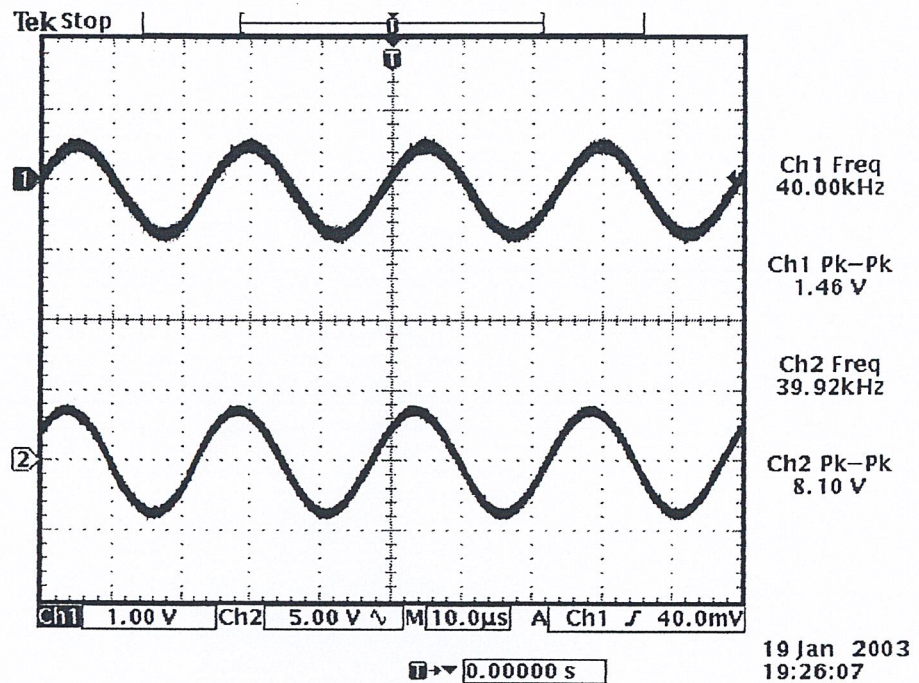
รูปที่ 4.16 รูปแสดงเอาต์พุตของวงจรพรีแอมป์ที่วัดได้ วัดดูที่ระยะ 4.5 เมตร

#### 4.1.7 วงจรแบนด์พาสฟิลเตอร์

##### วิธีการทดลอง

1. ต่อวงจรแบนด์พาสฟิลเตอร์ดังรูปที่ 3.16
2. ทดสอบวงจรแบนด์พาสฟิลเตอร์ที่ออกแบบด้วยการป้อนสัญญาณรูปไซน์ สัญญาณรูปสี่เหลี่ยม และสัญญาณรูปสามเหลี่ยมที่มีความถี่ 40 กิโลเฮิร์ตซ์ ตามลำดับ
3. สังเกตสัญญาณเอาต์พุตที่ได้จากวงจรแบนด์พาสฟิลเตอร์ โดยการจับที่ขาเอาต์พุตของแบนด์พาสฟิลเตอร์ สังเกตและบันทึกผลการทดลอง ซึ่งจะเห็นได้ว่า สามารถตัดองค์ประกอบอื่นของสัญญาณและความถี่อื่นที่ไม่ใช่ความถี่ 40 กิโลเฮิร์ตซ์ออกไปได้ ปรากฏดังรูปที่ 4.17 – 4.19

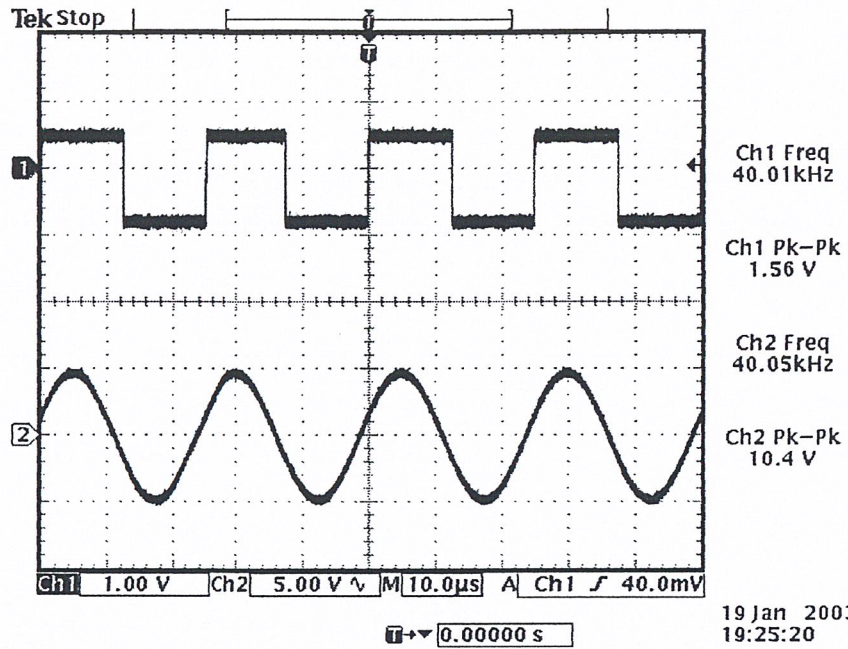
##### ผลการทดลอง



รูปที่ 4.17 รูปแสดงเอาต์พุตของวงจรแบนด์พาสฟิลเตอร์ เทียบกับ สัญญาณอินพุตรูปไซน์

Ch1. สัญญาณ 40 กิโลเฮิร์ตซ์ รูปไซน์ ที่ส่ง โดยเครื่องกำเนิดสัญญาณ

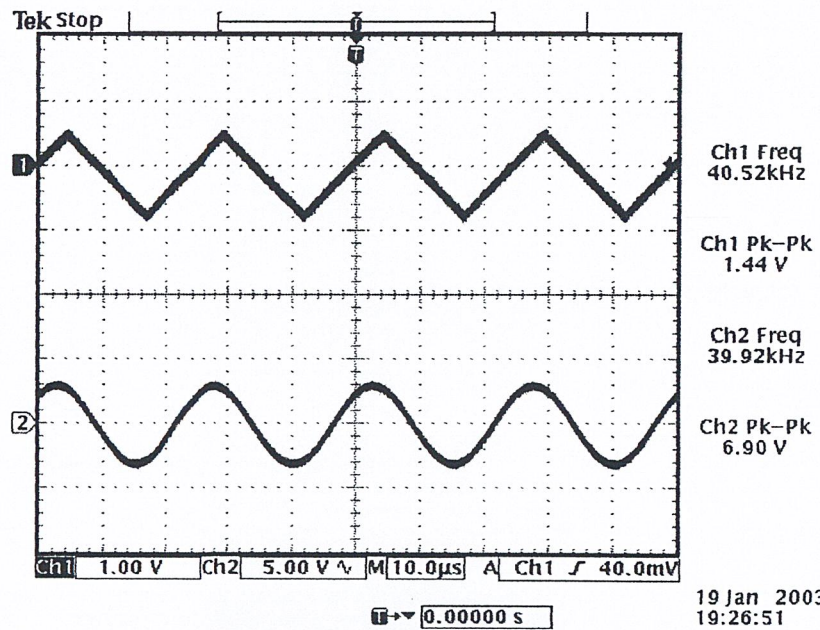
Ch2. แสดงสัญญาณเอาต์พุตของวงจรแบนด์พาสฟิลเตอร์



รูปที่ 4.18 รูปแสดงเอาต์พุตของวงจรแบนด์พาสฟิลเตอร์ เทียบกับ สัญญาณอินพุตรูปสี่เหลี่ยม

Ch1. สัญญาณ 40 กิโลเฮิรตซ์ รูปสี่เหลี่ยม ที่ส่งโดยเครื่องกำเนิดสัญญาณ

Ch2. แสดงสัญญาณเอาต์พุตของวงจรแบนด์พาสฟิลเตอร์



รูปที่ 4.19 รูปแสดงเอาต์พุตของวงจรแบนด์พาสฟิลเตอร์ เทียบกับ สัญญาณอินพุตรูปสามเหลี่ยม

สามเหลี่ยม

Ch1. สัญญาณ 40 กิโลเฮิรตซ์ รูปสามเหลี่ยมที่ส่ง โดยเครื่องกำเนิดสัญญาณ

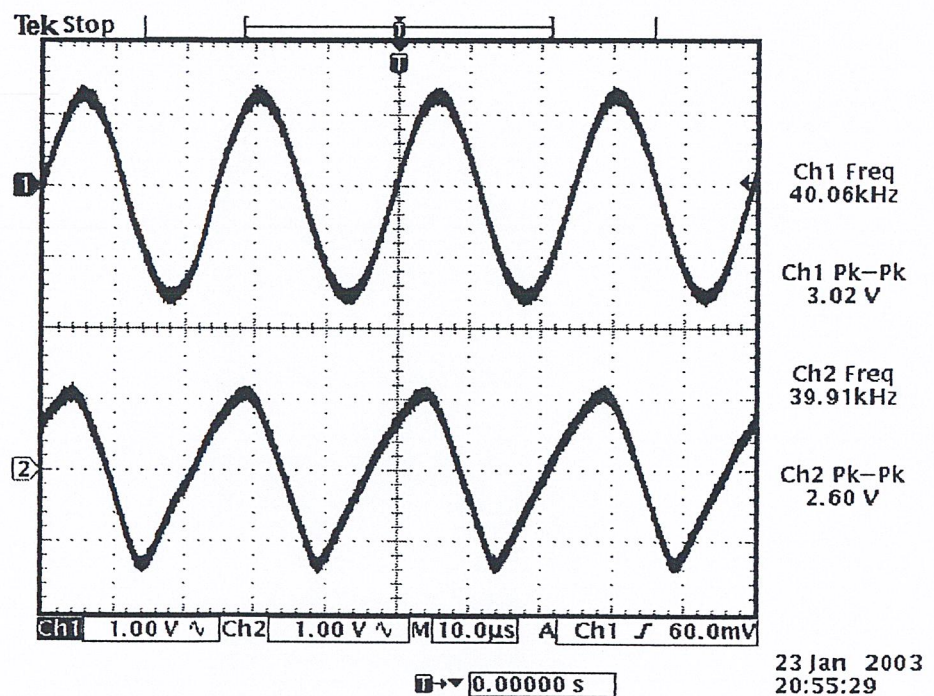
Ch2. แสดงสัญญาณเอาต์พุตของวงจรแบนด์พาสฟิลเตอร์

#### 4.1.8 วงจร AGC

##### วิธีการทดลอง

1. ต่อวงจร AGC ตามรูปที่ 3.17
2. ทดสอบวงจร AGC ด้วยการป้อนสัญญาณอินพุตที่มีความถี่ 40 กิโลเฮิร์ตส์ ด้วยเครื่องกำเนิดสัญญาณที่ขนาดตั้งแต่ 100 มิลลิโวลต์ ขึ้นมาเรื่อย สังเกตความเปลี่ยนแปลงที่ขาเอาต์พุตของวงจร AGC
3. ปรับขนาดแรงดันของสัญญาณที่ขนาด 3 โวลต์ และขนาด 7 โวลต์ตามลำดับ บันทึกผลสังเกตจะเห็นได้ว่าการใช้ AGC จะสามารถใช้ในการควบคุมขนาดของสัญญาณที่ป้อนเข้ามาได้เป็นอย่างดี ในขนาดสัญญาณที่เข้ามาต่ำจะมีอัตราขยายที่สูง จนกระทั่งถึงจุดๆ หนึ่งจะเกิดการคงที่ของสัญญาณไม่ว่าจะป้อนสัญญาณที่มีแรงดันขนาดเท่าใดเข้ามา ทำให้สามารถควบคุมอัตราขยายของสัญญาณได้คงที่ และระดับสัญญาณที่ได้นั้นจะมีค่าไม่เกิน 5 โวลต์ ปรากฏดังรูปที่ 4.20 และ 4.21

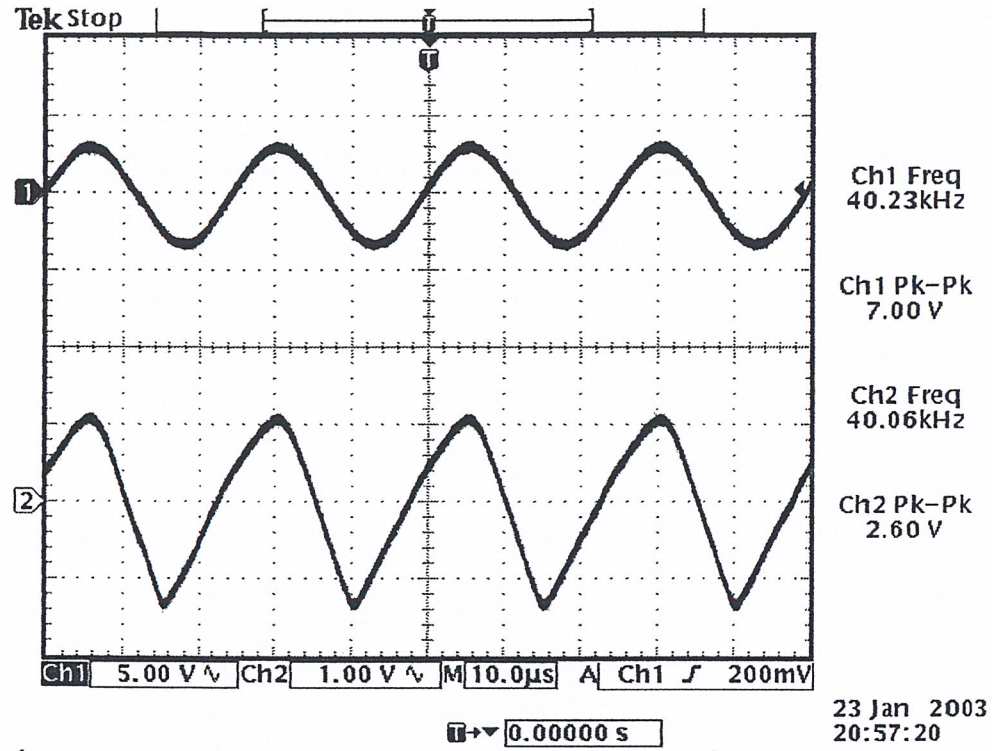
##### ผลการทดลอง



รูปที่ 4.20 รูปแสดงเอาต์พุตของวงจร AGC เทียบกับ สัญญาณอินพุตที่ขนาด 3 โวลต์

Ch1. สัญญาณ 40 กิโลเฮิร์ตซ์ ที่ขนาด 3 โวลต์

Ch2. แสดงสัญญาณเอาต์พุตของ AGC



รูปที่ 4.21 รูปแสดงเอาต์พุตของวงจร AGC เทียบกับ สัญญาณอินพุตที่ขนาด 7 โวลต์

Ch1. สัญญาณ 40 กิโลเฮิรตซ์ ที่ขนาด 7 โวลต์

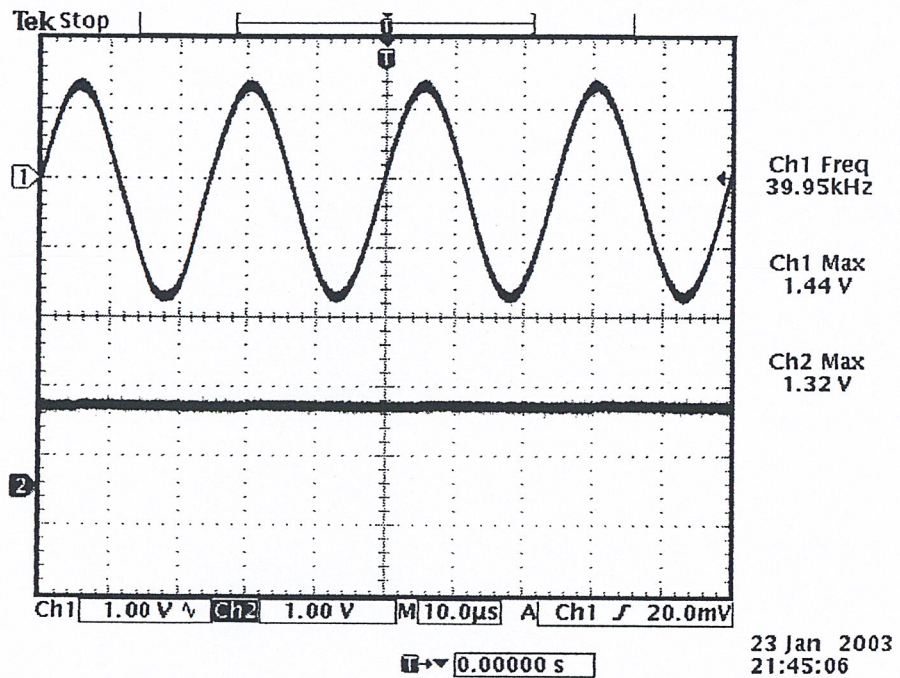
Ch2. แสดงสัญญาณเอาต์พุตของ AGC

#### 4.1.9 วงจรเอนเวลโลปดีเทคเตอร์

##### วิธีการทดลอง

1. ทำการต่อวงจรเอนเวลโลปดีเทคเตอร์ ตามรูปวงจรที่ 3.18
2. ทำการทดสอบวงจร ด้วยการป้อนสัญญาณอินพุตรูปไซน์ ที่ความถี่ 40 กิโลเฮิรตซ์ ขนาดต่างๆ สังเกตความเปลี่ยนแปลงจากการ วัดเทียบอินพุตของวงจร ในที่นี้จะแสดงให้เห็นค่าที่วัดได้ที่สัญญาณขนาด 1.4 โวลต์ และขนาด 3.4 โวลต์ ตามลำดับ ปรากฏผลที่ได้จากการทดลองตามรูปที่ 4.22 – 4.23
3. ทำการวัดค่าสัญญาณที่วัดได้ ด้วยการวัดที่เอาต์พุตของ วงจรเอนเวลโลป ดีเทคเตอร์ เทียบกับวงจรที่รับสัญญาณอินพุตจาก สัญญาณเอาต์พุตของวงจร AGC ที่ได้ จากการใช้งานจริงได้ ปรากฏผลการทดลองดังรูปที่ 4.24

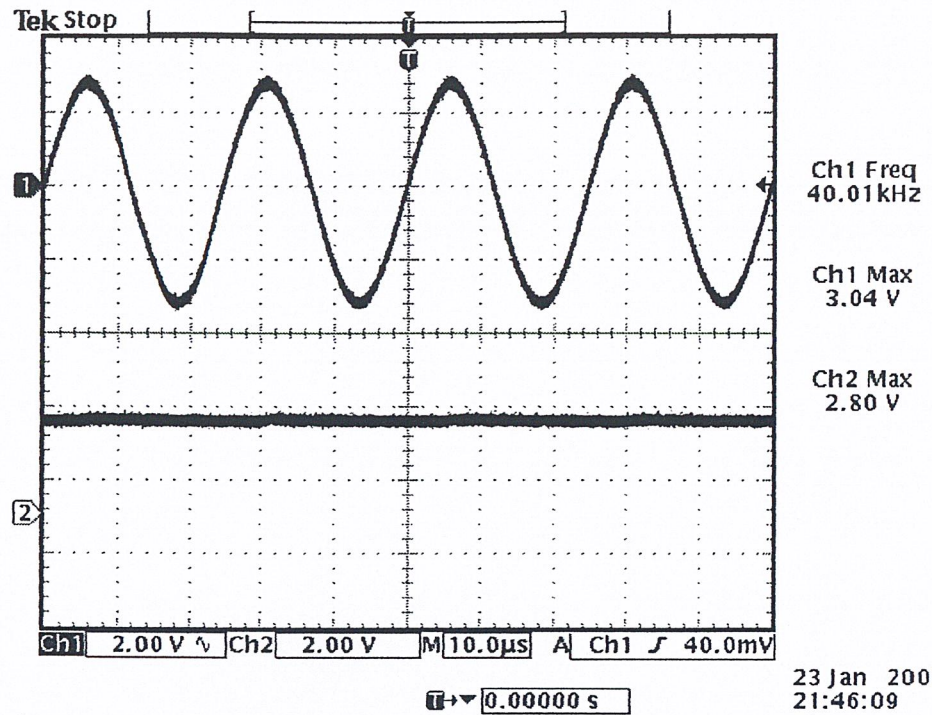
##### ผลทดลอง



รูปที่ 4.22 รูปแสดงเอาต์พุตของวงจรเอนเวลโลปดีเทคเตอร์ เทียบกับอินพุต 1.5 โวลต์

Ch1. สัญญาณ 40 กิโลเฮิรตซ์ ที่ขนาด 1.5 โวลต์

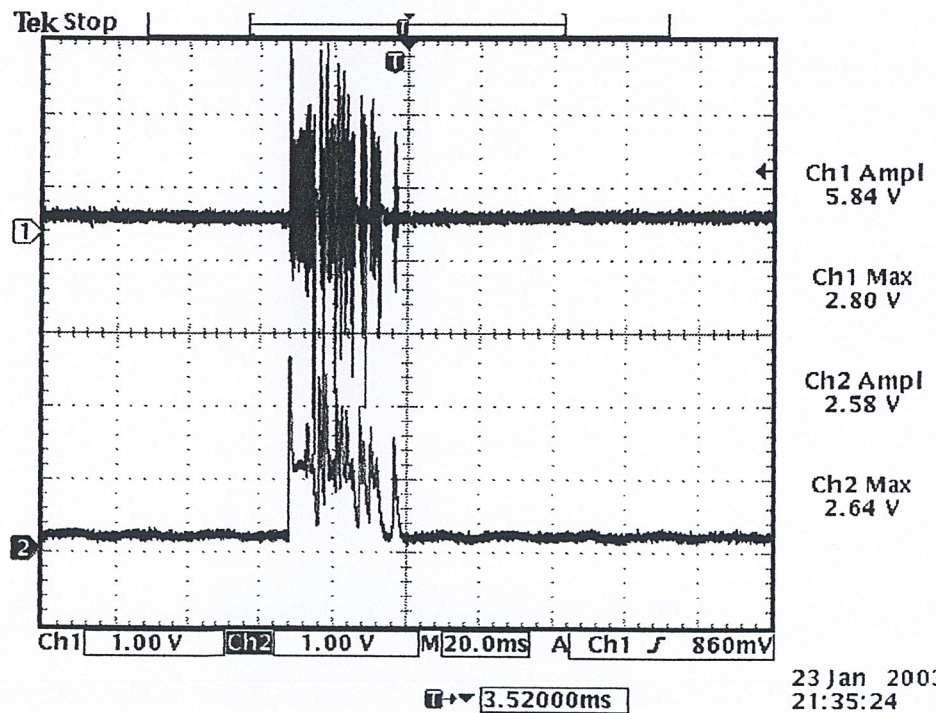
Ch2. แสดงสัญญาณเอาต์พุตของวงจรเอนเวลโลปดีเทคเตอร์



รูปที่ 4.23 รูปแสดงเอาต์พุตของวงจรเอนเวลโลปดีเทคเตอร์ เทียบกับอินพุต 3 โวลต์

Ch1. สัญญาณ 40 กิโลเฮิร์ต ที่ขนาด 3 โวลต์

Ch2. แสดงสัญญาณเอาต์พุตของวงจรเอนเวลโลปดีเทคเตอร์



รูปที่ 4.24 รูปแสดงเอาต์พุตของวงจรเอนเวลโลปดีเทคเตอร์ เทียบกับอินพุตจาก AGC

Ch1. สัญญาณเอาต์พุต จากวงจร AGC

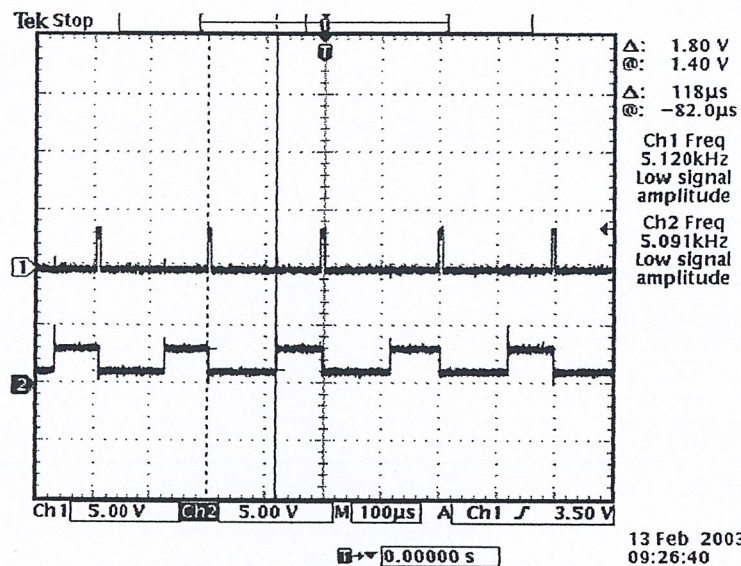
Ch2. แสดงสัญญาณเอาต์พุตของวงจรเอนเวลโลปดีเทคเตอร์

#### 4.1.10 การรับส่งข้อมูลระหว่าง MCS-51 กับคอมพิวเตอร์ ทาง Parallel Port

##### วิธีการทดลอง

1. ทำการเขียนโปรแกรมทดสอบสัญญาณ Hand check ระหว่าง MCS-51 และ Computer ผ่านทาง Parallel Port ด้วยภาษา VB ทดสอบด้วยการวัดสัญญาณโดยการวัดที่ขาที่มีการส่งสัญญาณตรวจสอบสถานะซึ่งกันและกัน ที่บริเวณขาของ MCS-51
2. ทำการวัดเวลาจากกระบวนการแปลงจากสัญญาณ Analog เป็นสัญญาณ Digital
3. ทำการวัดเวลาจากกระบวนการรับ-ส่งข้อมูล และบันทึกค่าสัญญาณที่วัดได้เพื่อประมวลผลด้วยข้อมูลในรูปของเลขฐาน 2 เพื่อนำมาใช้แสดงผลภาพต่อไป ดังจะเห็นได้จากการผลการทดลองในรูปที่ 4.25 – 4.27

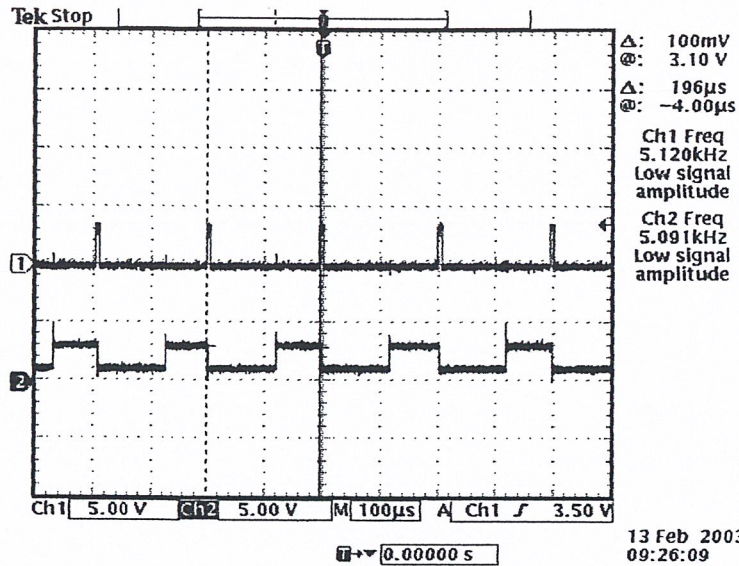
##### ผลการทดลอง



รูปที่ 4.25 รูปแสดงสัญญาณตรวจสอบซึ่งกันและกันระหว่าง MCS-51 กับคอมพิวเตอร์

Ch1. สัญญาณตรวจสอบจากคอมพิวเตอร์

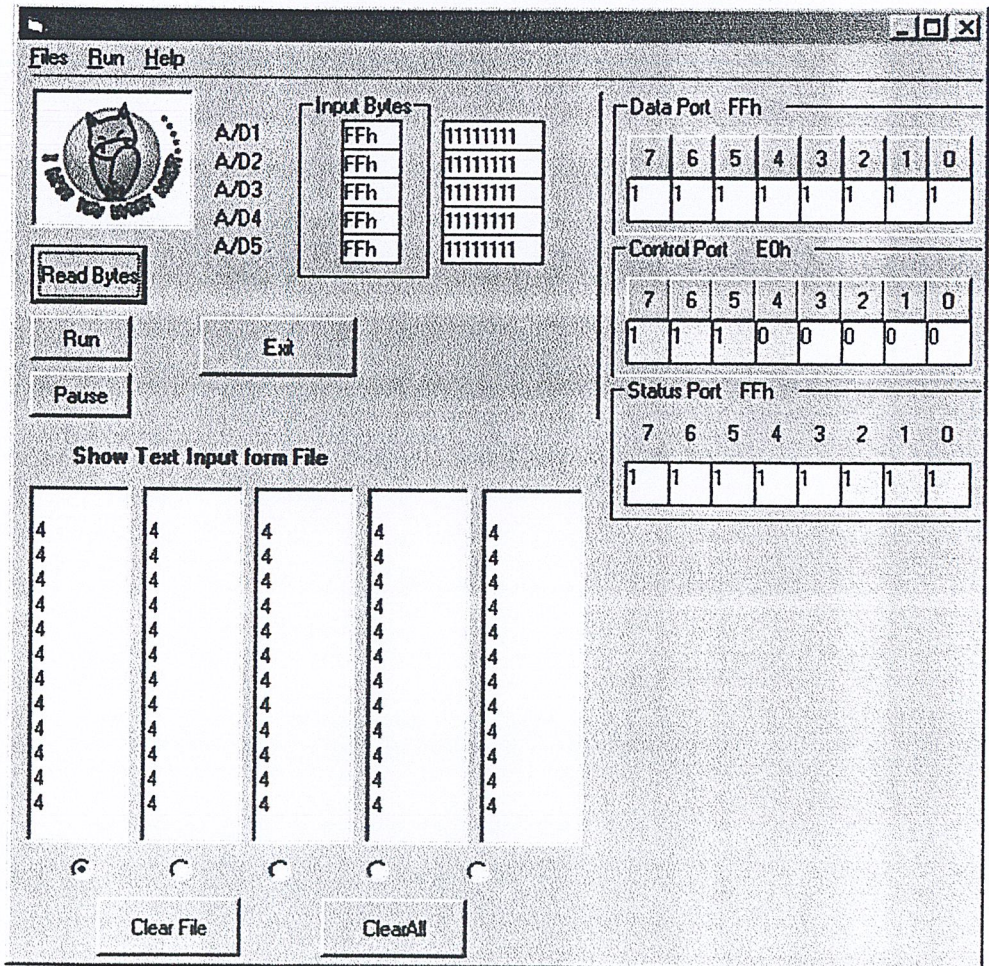
Ch2. สัญญาณตรวจสอบจาก MCS-51 และแสดงเวลาในการแปลงสัญญาณ



รูปที่ 4.26 รูปแสดงสัญญาณตรวจสอบซึ่งกันและกันระหว่าง MCS-51 กับคอมพิวเตอร์

Ch1. สัญญาณตรวจสอบจากคอมพิวเตอร์

Ch2. สัญญาณตรวจสอบจาก MCS-51 และอัตราการแซมปลิ่ง



รูปที่ 4.27 รูปแสดงการรับข้อมูลผ่านวงจร Selector ร่วมกับ MCS-51 ผ่านทางคอมพิวเตอร์

## บทที่ 5

### บทวิจารณ์และบทสรุป

จากการทดลองพบว่าปัญหาที่เกิดขึ้นคือ

1. ไม่สามารถตรวจจับวัตถุที่เป็นวัตถุคู่ควบเสียงได้ เนื่องจากวัตถุเหล่านั้นมีสัมประสิทธิ์การคู่ควบสูง ทำให้คลื่นอัลตราโซนิกถูกคู่ควบไป คลื่นอัลตราโซนิกที่สะท้อนกลับมาจึงมีความแรงของสัญญาณต่ำทำให้วัฏระยะวัตถุเหล่านี้ไม่ได้
2. ไม่สามารถใช้งานในห้องปิดทั่วไปได้ เพราะเกิดการสะท้อน (echo) มากเกินจนไม่สามารถแยกแยะสัญญาณได้ จึงสามารถใช้งานได้ในพื้นที่โล่งเปิดเท่านั้น
3. อัตราการแซมปลิง (Sampling rate) ที่ได้จากการแปลงของ A/D แล้วนำข้อมูลไปเก็บไว้ในคอมพิวเตอร์มีค่าค่อนข้างต่ำ เนื่องจากโปรแกรมคอมพิวเตอร์ที่นำมาใช้คือ วิชาลเบสิก (Visual Basic) เป็นภาษาที่ทำงานค่อนข้างช้า ทำให้กินเวลาในการเก็บข้อมูลไว้ในคอมพิวเตอร์ค่อนข้างสูง
4. ความผิดพลาดที่เกิดจากความไม่แน่นอนของสัญญาณจากการสะท้อนของคลื่น เนื่องจากปัญหาที่เกิดขึ้นมาจากความผิดพลาดในการสะท้อนของตัวคลื่นเอง โดยเราจะพบว่าการสะท้อนของคลื่นกลับมาในขนาดที่ไม่แน่นอน บางช่วงเวลาไม่สามารถตรวจจับสัญญาณที่เกิดจากการสะท้อนได้เลย เนื่องจากสัญญาณในช่วงนั้นหายไป ทำให้ต้องเพิ่มครั้งในการตรวจจับเพื่อหาค่าเฉลี่ย แต่ก็ยังพบที่มีความผิดพลาดอยู่มาก และใช้เวลาในการตรวจจับวัตถุนานกว่ามาก ด้วยเหตุนี้ทำให้เราไม่สามารถแก้ไขปัญหานี้เองมาจากตัวคลื่นเองได้
5. ปัญหาอันเนื่องมาจากสภาพแวดล้อมที่ไม่แน่นอน เช่น การมีลมพัด หรือ สภาพภูมิอากาศที่เปลี่ยนแปลงไปอย่างรวดเร็ว และไม่แน่นอน ทำให้การตรวจจับวัตถุเกิดความผิดพลาดสูงมาก อันเนื่องมาจากการเปรียบเทียบสัญญาณในช่วงเวลาที่แตกต่างกัน

# ภาคผนวก

## Pseudo Code

## MCS-51

```

                                ORG 0000H
START:                          MOV TMOD,#01H;      (2) TIMER 0 MODE 1
MAIN:                            MOV TH0,#1EH;      (2) SET UP TIMER
                                MOV TL0,#34H;      (2)

                                MOV R0,#04H;      (1) SIGNAL TO GENERATE CODE
                                SETB P3.0;        (1)
                                ACALL DELAY;      (2)
                                CLR P3.0;        (1)
                                ACALL DELAY;      (2)
PULSE:                            SETB P3.1;      (1)
                                ACALL DELAY;      (2)
                                CLR P3.1;        (1)
                                ACALL DELAY;      (2)
                                DJNZ R0,PULSE;    (2)

SETB TR0;                          (1) START TIMER
AD:                                CLR P3.3;      (1) START A/D
                                SETB P3.3;      (1)
                                ACALL D110;      (2)
                                CLR P3.2;        (1)
                                SETB P2.0;      (1)
COM:                                JNB P3.6,COM;   (2)
                                CLR P2.0;      (1)

                                JNB TF0,AD;      (2)
                                CLR TF0         (1)
                                CLR TR0         (1)
                                ACALL DEL;      (2)
                                SJMP MAIN;      (2)

DELAY:                              MOV R1,#04H;      (1) DELAY 2 mS
DELAY1:                            MOV R2,#71H;      (1)
DELAY2:                            DJNZ R2,DELAY2;  (2)
                                DJNZ R1,DELAY1;  (2)
                                NOP;            (1)
                                RET;            (2)

D110:                              MOV R7,#30H;      (1) DELAY 110 uS
D1101:                             DJNZ R7,D1101;    (2)
                                NOP;            (1)
                                RET;            (2)

DEL:                                MOV R3,#0FAH;    (1) DELAY 80 mS
DEL1:                              MOV R4,#92H;     (1)
DEL2:                              DJNZ R4,DEL2;    (2)
                                DJNZ R3,DEL1;    (2)
                                RET;
                                END;

```

## Pseudo Code

## Visual C++

การสร้างไฟล์ DLL เพื่อให้ VisualBasic สามารถติดต่อกับพอร์ทขนานของคอมพิวเตอร์

```
#include <stdio.h>
#include <conio.h> // สำหรับกำหนดไฟล์อ้างอิง _inp และ _outp
    BOOL inpVB(WORD PORT)
    {
        return ( _inp(PORT));
    };

    BOOL APIENTRY outVB(WORD port,BOOL bytedata)
    {
        return ( _outp(port,bytedata)) ;
    }
```

Pseudo Code  
VB

```
'Option Explicit
Dim DataIn%(0 To 7)
Dim DataByte%(0 To 4)
'-----
'-----'

'Part 0 Menu Program
'-----
'-----'

Private Sub AboutMenu_Click(Index As Integer)
About.Show
End Sub

Private Sub MotorMenu_Click(Index As Integer)
Motor.Show
End Sub

Private Sub PlotMenu_Click(Index As Integer)
Plot.Show
End Sub

Private Sub PortMenu_Click(Index As Integer)
frmSelectPort.Show
End Sub

Private Sub OnExit_Click()
ShutDown
End
End Sub
' ปิดโปรแกรม และ Reset ค่า ที่ป้อนให้เป็น 0

Private Sub Writefile()
'-----
'-----'
' เก็บข้อมูลลงเป็น Text file และ แสดงข้อมูล Text file ออกมาทางหน้าจอ
'-----
'-----'

Dim ByteNumber%
For ByteNumber = 0 To 4
Select Case ByteNumber
Case 0
FileName = "c:\testfile1.txt"
Const ForReading = 1, ForWriting = 2, ForAppending = 8
Dim fs, f
Set fs = CreateObject("Scripting.FileSystemObject")
Set f = fs.OpenTextFile(FileName, ForAppending, TristateFalse)
Set R = fs.OpenTextFile(FileName, ForReading, TristateFalse)
f.WriteLine (BinToDec(lblByteIn(5).Caption) * 5 \ 256)
'f.WriteLine lblByteIn(5).Caption
f.Close
Text11(0).Text = R.ReadAll
R.Close
Case 1
FileName = "c:\testfile2.txt"
Set fs = CreateObject("Scripting.FileSystemObject")
Set f = fs.OpenTextFile(FileName, ForAppending, TristateFalse)
Set R = fs.OpenTextFile(FileName, ForReading, TristateFalse)
' f.WriteLine lblByteIn(6).Caption
f.WriteLine (BinToDec(lblByteIn(6).Caption) * 5 \ 256)
```

```

    f.Close
    Text11(1).Text = R.ReadAll
    R.Close
Case 2
    FileName = "c:\testfile3.txt"
    Set fs = CreateObject("Scripting.FileSystemObject")
    Set f = fs.OpenTextFile(FileName, ForAppending, TristateFalse)
    Set R = fs.OpenTextFile(FileName, ForReading, TristateFalse)
    'f.Writeline lblByteIn(7).Caption
    f.Writeline (BinToDec(lblByteIn(7).Caption) * 5 \ 256)
    f.Close
    Text11(2).Text = R.ReadAll
    R.Close
Case 3
    FileName = "c:\testfile4.txt"
    Set fs = CreateObject("Scripting.FileSystemObject")
    Set f = fs.OpenTextFile(FileName, ForAppending, TristateFalse)
    Set R = fs.OpenTextFile(FileName, ForReading, TristateFalse)
    f.Writeline (BinToDec(lblByteIn(8).Caption) * 5 \ 256)
    'f.Writeline lblByteIn(8).Caption
    f.Close
    Text11(3).Text = R.ReadAll
    R.Close
Case Else
    FileName = "c:\testfile5.txt"
    Set fs = CreateObject("Scripting.FileSystemObject")
    Set f = fs.OpenTextFile(FileName, ForAppending, TristateFalse)
    Set R = fs.OpenTextFile(FileName, ForReading, TristateFalse)
    f.Writeline (BinToDec(lblByteIn(9).Caption) * 5 \ 256)
    'f.Writeline lblByteIn(9).Caption
    f.Close
    Text11(4).Text = R.ReadAll
    R.Close
End Select
Next ByteNumber
End Sub

Private Sub Form_Load()
StartUp
ReadPorts (BaseAddress)
    'Out &H378, 0
    Dim ControlPortData As Integer
    'ControlPortData = ControlPortRead(BaseAddress)
    BitSet ControlPortData, 5
    ControlPortWrite BaseAddress, ControlPortData
    ReadPorts (BaseAddress)

End Sub

Private Sub Form_Unload(Cancel As Integer)
ShutDown
End
End Sub

Private Sub mnuPort_Click(Index As Integer)
frmSelectPort.Show
End Sub

Private Sub Cleartext_Click()
    If Option1.Value = True Then
        FileName = "c:\testfile1.txt"
        n = 0

```

```

        ElseIf Option2.Value = True Then
        FileName = "c:\testfile2.txt"
        n = 1
        ElseIf Option3.Value = True Then
        FileName = "c:\testfile3.txt"
        n = 2
        ElseIf Option4.Value = True Then
        FileName = "c:\testfile4.txt"
        n = 3
        ElseIf Option5.Value = True Then
        FileName = "c:\testfile5.txt"
        n = 4
        End If
        Const ForReading = 1, ForWriting = 2, ForAppending = 8
        Set fs = CreateObject("Scripting.FileSystemObject")
        Set a = fs.OpenTextFile(FileName, ForWriting, TristateFalse)
        a.Writeline ""
        Text11(n).Text = ""
        a.Close
    End Sub

Private Sub GetBytesFromDataIn()
'Bits 3-7 of the 8 bytes contain data.
'To make 5 data bytes from these bits,
'each data byte contains one bit from each byte read.
'For example, data byte 0 contains 8 "bit 3s," one from each byte
read.
Dim ByteNumber%
Dim BitNumber%
Dim BitToAdd%
For ByteNumber = 0 To 4
    DataByte(ByteNumber) = 0
    'BitRead gets the selected bit value (ByteNumber + 3)
    'from the selected byte read (DataIn(BitNumber)).
    'To get the bit value for the created data byte, multiply times
    2^BitNumber.
    'Add each bit value to the created byte.
    For BitNumber = 0 To 7
        BitToAdd = (BitRead(DataIn(BitNumber), ByteNumber + 3)) * 2 ^
    BitNumber
        DataByte(ByteNumber) = DataByte(ByteNumber) + BitToAdd
    Next BitNumber
Next ByteNumber
End Sub

Private Sub cmdReadBytes_Click()
Dim BitNumber%
'The control port selects a bit number to read.
'The status port holds the data to be read.
For BitNumber = 0 To 7
    ControlPortWrite BaseAddress, BitNumber
    DataIn(BitNumber) = StatusPortRead(BaseAddress)
Next BitNumber
GetBytesFromDataIn
DisplayResults
Writefile
End Sub

Private Sub DisplayResults()
Dim ByteNumber%
For ByteNumber = 0 To 4

```

```

lblByteIn(ByteNumber).Caption = Hex$(DataByte(ByteNumber)) & "h"
lblByteIn(ByteNumber + 5).Caption = DecToBin$(DataByte(ByteNumber))
Next ByteNumber
End Sub
Private Sub cmdControlBitToggle_Click(Index As Integer)
'toggle a bit at the control port
Dim ControlPortData As Integer
ControlPortData = ControlPortRead(BaseAddress)
BitToggle ControlPortData, Index
ControlPortWrite BaseAddress, ControlPortData
ReadPorts (BaseAddress)
End Sub
Private Sub cmdDataBitToggle_Click(Index As Integer)
'toggle a bit at the data port
Dim DataPortData As Integer
DataPortData = DataPortRead(BaseAddress)
BitToggle DataPortData, Index
DataPortWrite BaseAddress, DataPortData
ReadPorts (BaseAddress)
End Sub

Private Sub Run_Click_Click()

n = 1
Dim ControlPortData As Integer
If n > 0 Then
Do
DoEvents
ReadPorts (BaseAddress)
a = Inp(&H378)

If a = 255 Then
'b = Inp(&H378 + 2)
'Debug.Print b
'Out (&H378 + 2), (b + 8)
'Out (&H378 + 2), (b - 8)
BitToggle ControlPortData, 3
ControlPortWrite BaseAddress, ControlPortData
BitToggle ControlPortData, 3
ControlPortWrite BaseAddress, ControlPortData
' b = b + 1
'Debug.Print b
cmdReadBytes_Click

End If

Loop Until n = 0
End If
End Sub

Private Sub Stop_Click_Click()
Dim ControlPortData As Integer
ControlPortData = ControlPortRead(BaseAddress)
Dim DataPortData As Integer
DataPortData = DataPortRead(BaseAddress)
BitSet ControlPortData, 5
BitToggle ControlPortData, 5
ControlPortWrite BaseAddress, ControlPortData
BitToggle DataPortData, 2
DataPortWrite BaseAddress, DataPortData

```

```
ReadPorts (BaseAddress)
End Sub
Private Sub ClearAll_Click()

    Const ForReading = 1, ForWriting = 2, ForAppending = 8

    For n = 0 To 4
        If n = 0 Then
            FileName = "c:\testfile1.txt"
            Set fs = CreateObject("Scripting.FileSystemObject")
            Set a = fs.OpenTextFile(FileName, ForWriting, TristateFalse)

            a.Writeline ""
            a.Close
            ElseIf n = 1 Then
                FileName = "c:\testfile2.txt"
                Set fs = CreateObject("Scripting.FileSystemObject")
                Set a = fs.OpenTextFile(FileName, ForWriting, TristateFalse)

            a.Writeline ""
            a.Close
            ElseIf n = 2 Then
                FileName = "c:\testfile3.txt"
                Set fs = CreateObject("Scripting.FileSystemObject")
                Set a = fs.OpenTextFile(FileName, ForWriting, TristateFalse)

            a.Writeline ""
            a.Close
            ElseIf n = 3 Then
                FileName = "c:\testfile4.txt"
                Set fs = CreateObject("Scripting.FileSystemObject")
                Set a = fs.OpenTextFile(FileName, ForWriting, TristateFalse)

            a.Writeline ""
            a.Close
            ElseIf n = 4 Then
                FileName = "c:\testfile5.txt"
                Set fs = CreateObject("Scripting.FileSystemObject")
                Set a = fs.OpenTextFile(FileName, ForWriting, TristateFalse)

            a.Writeline ""
            a.Close
        End If
        Text11(n).Text = ""
    Next
End Sub
```

Pseudo Code  
VB  
ควบคุม Stepping motor

```

Dim ValExit As Boolean ' ตัวแปรสำหรับหยุดการหมุนของมอเตอร์
Dim valStep, valCount As Long ' ตัวแปรของค่าสเตป, ส่งไปควบคุมบิต, นับจำนวนสเตป
Dim BitStep(4) As Integer
Dim valSpeed As Single ' หน่วงเวลาการแต่ละสเตป
Dim PortAddress As Long ' กำหนดตำแหน่งพอร์ตที่เราใช้ต่อควบคุมสเตปมอเตอร์

Private Sub Command1_Click()
' เมื่อมีการคลิกปุ่มให้มอเตอร์หมุนซ้ำ
ValExit = False
valSpeed = Val(Text1.Text) ' กำหนดความเร็วในแต่ละสเตป
valStep = Val(Text2.Text) ' กำหนดจำนวนสเตป
valCount = 0
If Option1.Value = True And valAngle <> 0 Then ' ถ้าเลือกการหมุนแบบตามจำนวนมุมที่กำหนด
    valStep = Val(Text2.Text)
    BitStep(1) = 2
    BitStep(2) = 4
    BitStep(3) = 1
    BitStep(4) = 8
    Y = 1
    Do
        DoEvents ' ปล่อยให้ CPU ไปประมวลผลอย่างอื่นบ้าง
        valCount = valCount + 1 ' นับจำนวนสเตป
        ' ส่งค่าออกพอร์ตไปสั่งให้โซลิตสเตปทำงานพลาจจ่ายแรงดันให้ขดลวดเริ่มต้นที่บิตแรก
        Out &H378 + 2, BitStep(Y)
        DelayTime (valSpeed) ' หน่วงเวลาเพื่อควบคุมความเร็วรอบแต่ละสเตป
        Y = Y + 1
        If Y > 4 Then Y = 1 ' เมื่อถึงบิตที่ 4 ให้เรากลับไปที่บิต 1 ใหม่
        Label1.Caption = valCount ' แสดงค่าจำนวนสเตปที่ Label 1
        Label6.Caption = ConvertStep (Label1.Caption)
    Loop Until valCount = valStep ' จบการทำงานเมื่อค่าที่เท่ากับค่าจำนวนสเตปที่ตั้งไว้
End If
If Option2.Value = True Then ' ถ้าเลือกการหมุนแบบสเตปต่อเนื่อง
    BitStep(1) = 2
    BitStep(2) = 4
    BitStep(3) = 1
    BitStep(4) = 8
    Do
        DoEvents
        ' ส่งค่าออกพอร์ตไปสั่งให้โซลิตสเตปทำงานพลาจจ่ายแรงดันให้ขดลวด
        Out &H378 + 2, BitStep(Y)
        valCount = valCount + 1
        Call DelayTime (valSpeed) ' หน่วงเวลาเพื่อควบคุมความเร็วรอบแต่ละสเตป
        Y = Y + 1
        If Y > 4 Then Y = 1 ' เมื่อถึงบิตที่ 4 ให้เรากลับไปที่บิต 1 ใหม่
        Label1.Caption = valCount ' แสดงค่าจำนวนสเตปที่ Label 1
        Label6.Caption = ConvertStep (Label1.Caption)
    Loop Until ValExit = True ' จบการทำงานเมื่อคลิกปุ่มหยุดหมุนโดยจะรับค่า ValExit=True
End If

If Option3.Value = True Then ' ถ้าเลือกการหมุนแบบตามจำนวนมุมที่กำหนด
Do Until ValExit = True

```

```

valStep = Val (Text2.Text)
  BitStep(1) = 2
  BitStep(2) = 4
  BitStep(3) = 1
  BitStep(4) = 8
  Y = 1
  Do
    DoEvents      ' ปล่อยให้ CPU ไปประมวลผลอย่างอื่นบ้าง
    valCount = valCount + 1      ' นับจำนวนตลับ
    ' ส่งค่าออกพอร์ตไปตั้งให้โชติศตแดงทำงานเพาธจ่ายแรงดันให้หลอดรีดคันท์ที่บิตแรก
    Out &H378 + 2, BitStep(Y)

    Call DelayTime(valSpeed)      ' หน่วงเวลาเพื่อควบคุมความเร็วรอบแต่ละตลับ
    Y = Y + 1
    If Y > 4 Then Y = 1 ' เมื่อถึงบิตที่ 4 ให้รีเซ็ตค่าไปที่บิต 1 ใหม่
    Label1.Caption = valCount ' แสดงค่าจำนวนตลับที่ Label 1
    Label6.Caption = ConvertStep(Label1.Caption)
  Loop Until valCount = 50 Or ValExit = True ' จบการทำงานเมื่อคลิกปุ่มหยุดหมุนโดยจะรับค่า
ValExit=True

  If valCount = 50 Then
    BitStep(1) = 1
    BitStep(2) = 4
    BitStep(3) = 2
    BitStep(4) = 8
  Do
    DoEvents
    ' ส่งค่าออกพอร์ตไปตั้งให้โชติศตแดงทำงานเพาธจ่ายแรงดันให้หลอด
    Out &H378 + 2, BitStep(Y)
    valCount = valCount - 1
    Call DelayTime(valSpeed)      ' หน่วงเวลาเพื่อควบคุมความเร็วรอบแต่ละตลับ
    Y = Y + 1
  If Y > 4 Then Y = 1 ' เมื่อถึงบิตที่ 4 ให้รีเซ็ตค่าไปที่บิต 1 ใหม่
  Label1.Caption = valCount ' แสดงค่าจำนวนตลับที่ Label 1
  Label6.Caption = ConvertStep(Label1.Caption)

  Loop Until valCount = 0 Or ValExit = True ' จบการทำงานเมื่อคลิกปุ่มหยุดหมุนโดยจะรับค่า
ValExit=True

End If
Loop
  End If

End Sub

Private Sub Command2_Click()
  valCount = valStep
  ValExit = True ' ตั้งให้เมื่อคลิกปุ่มหยุดหมุนโดยจะตั้งค่า ValExit=True
  Out &H378 + 2, 0
End Sub

Public Sub DelayTime(DelayTime) ' รูทีนการหน่วงเวลาหน่วยวินาที
  Dim LoopTime As Long ' กำหนดจำนวนรอบเวลา
  Const SecInDay = 86400 ' จำนวนวินาทีใน 1 วัน
  LoopTime = Timer + DelayTime ' กำหนดจำนวนรอบเวลารวมกับค่าที่เรตติ้งการตั้ง
  If LoopTime > SecInDay Then ' เมื่อไรที่ค่าจำนวนรอบเวลายังน้อยกว่าจำนวนวินาทีใน 1 วัน
    LoopTime = LoopTime - SecInDay ' ให้นำจำนวนวินาทีใน 1 วันลบค่าจำนวนรอบเวลา

```

```

Do While Timer > LoopTime ' เมื่อไรที่ค่าเวลาของเครื่องยิ่งมากกว่าค่าจำนวนรอบเวลา
    DoEvents ' ปล่อยให้ CPU ไปประมวลผลอย่างอื่นบ้าง
    Loop ' กลับไปตรวจสอบเงื่อนไขใหม่
End If
Do While Timer < LoopTime ' เมื่อไรที่ค่าเวลาของเครื่องยังน้อยกว่าค่าจำนวนรอบเวลา
    DoEvents ' ปล่อยให้ CPU ไปประมวลผลอย่างอื่นบ้าง
Loop
' กลับไปตรวจสอบเงื่อนไขใหม่ ถ้าตรงเงื่อนไขก็เป็นอันว่าจบการตั้งเวลาออกจากรูทีน
End Sub
Private Sub Command3_Click()
' เมื่อมีการคลิกปุ่มให้มอเตอร์หมุนขวา
ValExit = False
valSpeed = Val(Text1.Text) ' กำหนดความเร็วในแต่ละสเตป
valStep = Val(Text2.Text) ' กำหนดจำนวนสเตป
valCount = 0
If Option1.Value = True And valAngle <> 0 Then ' ถ้าเลือกการหมุนแบบตามจำนวนมุมที่กำหนด
    BitStep(1) = 1
    BitStep(2) = 4
    BitStep(3) = 2
    BitStep(4) = 8
    Y = 1
    Do
        DoEvents ' ปล่อยให้ CPU ไปประมวลผลอย่างอื่นบ้าง
        valCount = valCount + 1 ' นับจำนวนสเตป
        ' ส่งค่าออกพอร์ตไปตั้งให้โซลิตเตดจทำงานเพาจอ้งแรงดันให้หลอดเริ่มขึ้นที่บิตแรก
        Out &H378 + 2, BitStep(Y)
        DelayTime (valSpeed) ' หน่วงเวลาเพื่อควบคุมความเร็วรอบแต่ละสเตป
        Y = Y + 1
        If Y > 4 Then Y = 1 ' เมื่อถึงบิตที่ 4 ให้รีเซ็ตค่าไปที่บิต 1 ใหม่
        Label1.Caption = valCount ' แสดงค่าจำนวนสเตปที่ Label 1
        Label6.Caption = ConvertStep (Label1.Caption)
    Loop Until valCount = valStep ' จบการทำงานเมื่อค่านี้เท่ากับจำนวนสเตปที่ตั้งไว้
End If
If Option2.Value = True Then ' ถ้าเลือกการหมุนแบบสเตปต่อเนื่อง
    BitStep(1) = 1
    BitStep(2) = 4
    BitStep(3) = 2
    BitStep(4) = 8
    Do
    DoEvents
    ' ส่งค่าออกพอร์ตไปตั้งให้โซลิตเตดจทำงานเพาจอ้งแรงดันให้หลอด
        Out &H378 + 2, BitStep(Y)
        valCount = valCount + 1
        Call DelayTime (valSpeed) ' หน่วงเวลาเพื่อควบคุมความเร็วรอบแต่ละสเตป
        Y = Y + 1
        If Y > 4 Then Y = 1 ' เมื่อถึงบิตที่ 4 ให้รีเซ็ตค่าไปที่บิต 1 ใหม่
        Label1.Caption = valCount ' แสดงค่าจำนวนสเตปที่ Label 1
        Label6.Caption = ConvertStep (Label1.Caption)
    Loop Until ValExit = True ' จบการทำงานเมื่อคลิกปุ่มหยุดหมุนโดยจะรับค่า ValExit=True
End If
If Option3.Value = True Then ' ถ้าเลือกการหมุนแบบตามจำนวนมุมที่กำหนด
Do Until ValExit = True
    valStep = Val(Text2.Text)
    BitStep(1) = 1

```

```

BitStep(2) = 4
BitStep(3) = 2
BitStep(4) = 8
Y = 1
Do
    DoEvents      ' ปลดปล่อย CPU ไปประมวลผลอย่างอื่นบ้าง
    valCount = valCount + 1      ' นับจำนวนวนวน
        ' ส่งค่าออกพอร์ตไปยังให้โวลิตแสดงทำงานพาทายจ่ายแรงดันให้ขดลวดเริ่มต้นที่บิตแรก
    Out &H378 + 2, BitStep(Y)

    Call DelayTime(valSpeed)      ' หน่วงเวลาเพื่อความนุ่มนวลความเร็วรอบแต่ละสเตป
        Y = Y + 1
    If Y > 4 Then Y = 1      ' เมื่อถึงบิตที่ 4 ให้รีเซ็ตค่าไปที่บิต 1 ใหม่
    Label1.Caption = valCount      ' แสดงค่าจำนวนวนวนที่ Label 1
    Label6.Caption = ConvertStep(Label1.Caption)
Loop Until valCount = 50 Or ValExit = True      ' จบการทำงานเมื่อคลิกปุ่มหยุดหมุนโดยจะรับค่า
ValExit=True

If valCount = 50 Then
    BitStep(1) = 2
    BitStep(2) = 4
    BitStep(3) = 1
    BitStep(4) = 8
    Do
    DoEvents
        ' ส่งค่าออกพอร์ตไปยังให้โวลิตแสดงทำงานพาทายจ่ายแรงดันให้ขดลวด
        Out &H378 + 2, BitStep(Y)
        valCount = valCount - 1
    Call DelayTime(valSpeed)      ' หน่วงเวลาเพื่อความนุ่มนวลความเร็วรอบแต่ละสเตป
        Y = Y + 1
    If Y > 4 Then Y = 1      ' เมื่อถึงบิตที่ 4 ให้รีเซ็ตค่าไปที่บิต 1 ใหม่
    Label1.Caption = valCount      ' แสดงค่าจำนวนวนวนที่ Label 1
    Label6.Caption = ConvertStep(Label1.Caption)

    Loop Until valCount = 0 Or ValExit = True      ' จบการทำงานเมื่อคลิกปุ่มหยุดหมุนโดยจะรับค่า
ValExit=True
End If
Loop
End If
End Sub
Private Sub Form_Load()
    Out &H378 + 2, 0      ' ตั้งค่าโฟลต์ที่เทอร์มคควบคุม ของ Card I/O 8255
End Sub

Private Sub valAngle_Change()
Text2.Text = ConvertAngle(Val(valAngle.Text))
End Sub

Private Sub Text2_Change()
valAngle.Text = ConvertStep(Val(Text2.Text))
End Sub

Private Sub OnExit_Click()      ' ปิดโปรแกรม และ Reset ค่าที่ป้อนให้เป็น 0
valCount = valStep
ValExit = True      ' ตั้งให้บิตเตอร์หยุดหมุน โดยจะส่งค่า ValExit=True
Out &H378 + 2, 0
End
End Sub

```

## Module ต่างๆ ที่ใช้ร่วมกัน

```
'Lptutil.bas
'General purpose routines for use with the PC's parallel port.
'Port accesses require a DLL with Inp and Out.
Option Explicit
```

```
Function DataPortRead%(BaseAddress%)
'Reads a parallel port's data port.
DataPortRead = Inp(BaseAddress)
End Function
```

```
Sub DataPortWrite(BaseAddress%, Value%)
'Writes a byte to a parallel port's data port.
Out BaseAddress, Value
End Sub
```

```
Function StatusPortRead%(BaseAddress%)
'Reads a parallel port's status port.
'Calculates the status-port address from the port's
'base address, and inverts bit 7 of the byte read.
'The status-port hardware reinverts these bits,
'so the value read matches the value at the connector.
StatusPortRead = (Inp(BaseAddress + 1) Xor &H80)
End Function
```

```
Function BitRead%(Variable%, BitNumber%)
'Returns the value (0 or 1) of the requested bit in a Variable.
Dim BitValue%
BitValue = 2 ^ BitNumber 'the value of the requested bit
BitRead = (Variable And BitValue) \ BitValue
End Function
```

```
Sub BitReset(Variable%, BitNumber%)
'Resets (clears) the requested bit in a Variable.
Dim BitValue, CurrentValue%
'The value of the requested bit.
BitValue = 2 ^ BitNumber
Variable = Variable And (&HFFFF - BitValue)
End Sub
```

```
Sub BitSet(Variable%, BitNumber%)
'Sets the requested bit in a Variable.
Dim BitValue, CurrentValue%
'The value of the requested bit.
BitValue = 2 ^ BitNumber
Variable = Variable Or BitValue
End Sub
```

```
Sub BitToggle(Variable%, BitNumber%)
'Toggles the requested bit in a Variable.
Dim BitValue, CurrentValue%
'The value of the requested bit.
BitValue = 2 ^ BitNumber
'Is the current value 0 or 1?
CurrentValue = Variable And BitValue
Select Case CurrentValue
    Case 0
        'If current value = 0, set it
        Variable = Variable Or BitValue
    Case Else
```

```

        'If current value = 1, reset it
        Variable = Variable And (&HFFFF - BitValue)
End Select
End Sub

Function ControlPortRead%(BaseAddress%)
'Reads a parallel port's control port.
'Calculates the control-port address from the port's
'base address, and inverts bits 0, 1, & 3 of the byte read.
'The control-port hardware reinverts these bits,
'so the value read matches the value at the connector.
ControlPortRead = (Inp(BaseAddress + 2) Xor &HB)
End Function

Sub ControlPortWrite(BaseAddress%, Value%)
'Writes a Value to a parallel port's control port.
'Calculates the control-port address from the port's
'base address, and inverts bits 0, 1, & 3.
'The control-port hardware reinverts these bits,
'so Value is written to the port connector.
Out BaseAddress + 2, Value Xor &HB
End Sub

Option Explicit

Sub ReadPorts(BaseAddress As Integer)
'Read the data, status, and control ports of selected port.
'Display the byte read and each bit in the byte.
Dim ByteRead As Integer
Dim BitNumber As Integer
Dim BitValue As Integer

ByteRead = DataPortRead(BaseAddress)
frmMain.lblDataPortByte(0).Caption = Hex$(ByteRead) + "h"
For BitNumber = 0 To 7
    BitValue = BitRead(ByteRead, BitNumber)
    frmMain.lblDataBit(BitNumber).Caption = BitValue
Next BitNumber

ByteRead = StatusPortRead(BaseAddress)
frmMain.lblStatusPortByte(0).Caption = Hex$(ByteRead) + "h"
For BitNumber = 0 To 7
    BitValue = BitRead(ByteRead, BitNumber)
    frmMain.lblStatusBit(BitNumber).Caption = BitValue
Next BitNumber

ByteRead = ControlPortRead(BaseAddress)
frmMain.lblControlPortByte(0).Caption = Hex$(ByteRead) + "h"
For BitNumber = 0 To 7
    BitValue = BitRead(ByteRead, BitNumber)
    frmMain.lblControlBit(BitNumber).Caption = BitValue
Next BitNumber

End Sub
'Declare Inp and Out for port I/O
'Two versions, for 16-bit and 32-bit Visual Basic 4 programs.

#If Win32 Then
'DLL procedure names are case-sensitive in VB4.
'Use Aliases so Inp and Out don't have to have matching case in VB.

```

```

Public Declare Function Inp Lib "inpout32.dll" _
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Public Declare Sub Out Lib "inpout32.dll" _
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)

#Else
Public Declare Function Inp Lib "inpout16.Dll" _
Alias "Inp16" (ByVal PortAddress As Integer) As Integer
Public Declare Sub Out Lib "inpout16.Dll" _
Alias "Out16" (ByVal PortAddress As Integer, ByVal Value As Integer)
#End If

```

These two declarations for the inpout32dll are for getting data from the  
' parallel port, Inp, and sending data to the parallel port, Out.

```

Public Declare Function Inp Lib "inpout32dll" _
Alias "Inp32"(ByVal PortAddress As Integer) As Integer

Public Declare Sub Out Lib "inpout32dll" _
Alias "Out32"(ByVal PortAddress As Integer, ByVal Value As Integer)

```

```

Dim PortAddress As String
Function BinToDec%(BinNumber$)
    Dim Weight%
    Dim Dec%
    Dim i%
    Weight% = 1
    Dec% = 0
    'Reset decimal number

    If BinNumber$ <> "0000000" Then
        For i% = Len(BinNumber$) To 1 Step -1
            If Mid$(BinNumber$, i%, 1) = "1" Then
                Dec% = Dec% + Weight% 'If bit=1 then add weight
            factor
            End If
            Weight% = Weight% * 2 'Multiply weight factor by 2
        Next
        BinToDec% = Dec% 'Store result
    Else
        BinToDec% = 0
    End If
End Function

```

```

Function DecToBin$(Decnumber%)
    'Conversion of decimal number (0..255) to 8 bit binary string.
    '-----
    Dim Bin$
    Dim Faktor%, i%

    Bin$ = ""
    Faktor% = 128

    If Decnumber% <> 0 Then
        For i% = 1 To 8
            If Faktor% > Decnumber% Then
                Bin$ = Bin$ + "0"
            Else

```

```

        Bin$ = Bin$ + "1"
        Decnumber% = Decnumber% - Faktor%
    End If
    Faktor% = Faktor% \ 2
Next
DecToBin$ = Bin$
Else
    DecToBin$ = "00000000"
End If
End Function

' These two declarations for the inpout32.dll are for getting data
from the
' parallel port, Inp, and sending data to the parallel port, Out.

Public Declare Function Inp Lib "inpout32.dll" _
Alias "Inp32"(ByVal PortAddress As Integer) As Integer

Public Declare Sub Out Lib "inpout32.dll" _
Alias "Out32"(ByVal PortAddress As Integer, ByVal Value As Integer)

Dim PortAddress As String
Function ConvertAngle%(Angle$)
    If Angle$ <> "0" Then
        ConvertAngle% = Angle$ / 3.6
    End If
End Function
Function ConvertStep%(StepAngle$)
    If StepAngle$ <> "0" Then
        ConvertStep% = StepAngle$ * 3.6
    End If
End Function

Option Explicit
#If Win32 Then
Declare Function GetPrivateProfileStringByKeyName& Lib "Kernel32" _
Alias "GetPrivateProfileStringA" _
(ByVal lpApplicationName$, ByVal lpszKey$, ByVal lpszDefault$, _
ByVal lpszReturnBuffer$, ByVal cchReturnBuffer&, ByVal lpszFile$)

Declare Function WritePrivateProfileString& Lib "Kernel32" _
Alias "WritePrivateProfileStringA" _
(ByVal lpApplicationName$, ByVal lpKeyName$, ByVal lpString$, ByVal
lpFileName$)

Declare Function GetWindowsDirectory& Lib "Kernel32" _
Alias "GetWindowsDirectoryA" (ByVal lpBuffer$, ByVal nSize%)

#Else

Declare Function GetPrivateProfileStringByKeyName% Lib "Kernel" _
Alias "GetPrivateProfileString" _
(ByVal lpApplicationName$, ByVal lpKeyName$, ByVal lpDefault$, _
ByVal lpReturnedString$, ByVal nSize%, ByVal lpFileName$)

```

```

Declare Function WritePrivateProfileString% Lib "Kernel" _
  (ByVal _
  lpApplicationName$, ByVal lpKeyName$, ByVal lpString$, ByVal
  lpFileName$)

Declare Function GetWindowsDirectory% Lib "Kernel" (ByVal lpBuffer$,
  ByVal nSize%)

#End If

Type PortData
  Name As String
  Address As Integer
  PortType As String
  EcpModeDescription As String
  EcpModeValue As Integer
  Enabled As Integer
End Type
Global Port(0 To 3)As PortData
Global BaseAddress%
Global PortType$
Global IniFile$

Global EcrAddress%
Global EcrData%
Global EcpDataPortAddress%
Global EppDataPort0Address%
Global IndexOfSelectedPort%
Global PortDescription$

Global EcpExists%
Global SppExists%
Global PS2Exists%
Global EppExists%

Function GetEcpModeDescription$(EcpModeValue%)
Select Case EcpModeValue
  Case 0
    GetEcpModeDescription = "SPP"
  Case 1
    GetEcpModeDescription = "PS/2"
  Case 10
    GetEcpModeDescription = "Fast Centronics"
  Case 11
    GetEcpModeDescription = "ECP"
  Case 100
    GetEcpModeDescription = "EPP"
  Case 110
    GetEcpModeDescription = "Test"
  Case 111
    GetEcpModeDescription = "Configuration"
End Select
End Function

Sub GetIniData()
'Use the Windows API call GetPrivateProfileString to read user
information
'from an ini file.

```

```

Dim NumberOfCharacters%
Dim ReturnBuffer As String * 128
Dim Index%
Dim WindowsDirectory$

'The API functions read data into fixed-length string buffers
'and return the number of characters in the entry.
'Use Left$ to discard all characters to the right of the final
character in the entry.

'Get the path of the Windows directory, where the ini file is stored.
NumberOfCharacters = GetWindowsDirectory(ReturnBuffer, 127)
WindowsDirectory = Left$(ReturnBuffer, NumberOfCharacters)
IniFile = WindowsDirectory + "\lptprogs.ini"

'If the ini file doesn't exist, don't try to read it.
If Not Dir$(IniFile) = "" Then
    'The port addresses:
    Port(0)Address = CInt(VbGetPrivateProfileString("lptdata",
"Port0Address", IniFile))
    Port(1)Address = CInt(VbGetPrivateProfileString("lptdata",
"Port1Address", IniFile))
    Port(2)Address = CInt(VbGetPrivateProfileString("lptdata",
"Port2Address", IniFile))
    Port(3)Address = CInt(VbGetPrivateProfileString("lptdata",
"Port3Address", IniFile))

    'The port types:
    Port(0).PortType = VbGetPrivateProfileString("lptdata",
"Port0Type", IniFile)
    Port(1).PortType = VbGetPrivateProfileString("lptdata",
"Port1Type", IniFile)
    Port(2).PortType = VbGetPrivateProfileString("lptdata",
"Port2Type", IniFile)
    Port(3).PortType = VbGetPrivateProfileString("lptdata",
"Port3Type", IniFile)

    'Port enabled?
    Port(0).Enabled = CInt(VbGetPrivateProfileString("lptdata",
"Port0Enabled", IniFile))
    Port(1).Enabled = CInt(VbGetPrivateProfileString("lptdata",
"Port1Enabled", IniFile))
    Port(2).Enabled = CInt(VbGetPrivateProfileString("lptdata",
"Port2Enabled", IniFile))
    Port(3).Enabled = CInt(VbGetPrivateProfileString("lptdata",
"Port3Enabled", IniFile))

    'The selected port
    IndexOfSelectedPort = CInt(VbGetPrivateProfileString("lptdata",
"IndexOfSelectedPort", IniFile))
End If
End Sub

Function ReadEcpMode%(TestAddress%)
'The Ecr mode is in bits 5, 6, and 7 of the ECR.
EcrAddress = TestAddress + &H402

```

```

EcrData = Inp(EcrAddress)
ReadEcpMode = (EcrData And &HE0)\ &H20
End Function

Function ReadEppTimeoutBit%(BaseAddress%)
'Reads and clears the EPP timeout bit (status port bit 0).
'Should be done after each EPP operation.
'The method for clearing the bit varies, so try 3 ways:
'1.Write 1 to it.
'2.Write 0 to it.
'3.Read it again.
Dim StatusPortAddress%
Dim ReadAgain%
StatusPortAddress = BaseAddress + 1
ReadEppTimeoutBit = BitRead(StatusPortRead(BaseAddress), 0)
Out StatusPortAddress, 1
Out StatusPortAddress, 0
ReadAgain = BitRead(StatusPortRead(BaseAddress), 0)
End Function

Sub SetEcpMode(EcpModeValue%)
'Store the Ecp mode's value and description in the Port array.
Port(IndexOfSelectedPort).EcpModeValue = EcpModeValue
Port(IndexOfSelectedPort).EcpModeDescription =
GetEcpModeDescription(EcpModeValue)
EcrAddress = BaseAddress + &H402
'Read the ECR & clear bits 5, 6, 7.
EcrData = Inp(EcrAddress) And &HF
'Write the selected value to bits 5, 6, 7.
EcrData = EcrData + EcpModeValue * &H20
Out EcrAddress, EcrData
End Sub

Sub ShutDown()
WriteIniData
End
End Sub

Sub StartUp()
Dim PortExists%
Dim Index%
'Get information from the ini file.
GetIniData

'Load the forms.
Load frmSelectPort
frmSelectPort.optPortName(IndexOfSelectedPort).Value = True
frmMain.Show
End Sub

Function TestForEcp%(TestAddress%)
'Test for the presence of an ECP.
'If the ECP is idle and the FIFO empty,
'in the ECP's Ecr (at Base Address+402h),
'bit 4(Fifo full)=0, and bit 4(Fifo empty)=1.
'The first test is to see if these bits differ from the
'corresponding bits in the control port (at Base Address+2).

```

```

'If so, a further test is to write 34h to the Ecr,
'then read it back. Bit 1 is read/write, and bit 0 is read-only.
'If the value read is 35h, the port is an ECP.
Dim EcrBit0%
Dim EcrBit1%
Dim ControlBit0%
Dim ControlBit1%
Dim ControlPortData%
Dim TestEcrAddress%
Dim OriginalEcrData%
TestForEcp = False
EcrAddress = TestAddress + &H402

'Read ECR bits 0 & 1 and Control Port bit 1.
EcrData = Inp(EcrAddress)
EcrBit0=BitRead(EcrData, 0)
EcrBit1=BitRead(EcrData, 1)
ControlPortData = ControlPortRead(TestAddress)
ControlBit1=BitRead(ControlPortData, 1)
If EcrBit0=1 And EcrBit1=0 Then
    'Compare control bit 1 to ECR bit 1.
    'Toggle the control bit if necessary,
    'to be sure the two registers are different.
    If ControlBit1=0 Then
        ControlPortWrite TestAddress, &HF
        ControlPortData = ControlPortRead(TestAddress)
        ControlBit1=BitRead(ControlPortData, 1)
    End If
    If EcrBit1 <> ControlBit1 Then
        OriginalEcrData = EcrData
        Out EcrAddress, &H34
        EcrData = Inp(EcrAddress)
        If EcrData = &H35 Then
            TestForEcp = True
        End If
        'Restore the ECR to its original value.
        Out EcrAddress, OriginalEcrData
    End If
End If
End Function

Function TestForEpp%(TestAddress%)
'Write to an Epp register, then read it back.
'If the reads match the writes, it's probably an Epp.
Dim ByteRead%
Dim StatusPortData%
Dim EppAddressPort%
Dim TimeoutBit%
Dim StatusPortAddress%
StatusPortAddress = TestAddress + 1
TestForEpp = False
'Use EppAddressPort for testing.
'SPPs, ECPs, and PS/2 ports don't have this register.
EppAddressPort = TestAddress + 3
Out EppAddressPort, &H55
'Clear the timeout bit after each EPP operation.
TimeoutBit = ReadEppTimeoutBit%(TestAddress%)

```

```

ByteRead = Inp(EppAddressPort)
TimeoutBit = ReadEppTimeoutBit%(TestAddress%)
If ByteRead = &H55 Then
    Out EppAddressPort, &HAA
    TimeoutBit = ReadEppTimeoutBit%(TestAddress%)
    ByteRead = Inp(EppAddressPort)
    TimeoutBit = ReadEppTimeoutBit%(TestAddress%)
    If ByteRead = &HAA Then
        TestForEpp = True
    End If
End If
End Function

```

```

Function TestForPS2%(TestAddress%)
'Tests a parallel port's data port for bidirectional ability.
'First, try to tri-state (disable) the data outputs by
'setting bit 5 of the Control port.
'Then write 2 values to the data port and read each back
'If the values match, the data outputs are not disabled,
'and the port is not bidirectional.
'If the values don't match,
'the data outputs are disabled and the port is bidirectional.
Dim DataInput%
Dim ControlPortData%
Dim OriginalControlPortData%
Dim OriginalDataPortData%

'Set Control port bit 5.
ControlPortWrite TestAddress, &H2F
TestForPS2=False
'Write the first byte and read it back:
DataPortWrite TestAddress, &H55
DataInput = DataPortRead(TestAddress)
'If it doesn't match, the port is bidirectional.
If Not DataInput = &H55 Then TestForPS2=True
'If it matches, write another and read it back.
If DataInput = &H55 Then
    DataPortWrite TestAddress, &HAA
    DataInput = DataPortRead(TestAddress)
    'If it doesn't match, the port is bidirectional
    If Not DataInput = &HAA Then
        TestForPS2=True
    End If
End If
'Reset Control port bit 5
ControlPortWrite TestAddress, &HF
End Function

```

```

Function TestForSpp%(TestAddress%)
'Write two bytes and read them back.
'If the reads match the writes, the port exists.
Dim ByteRead%
'Be sure that control port bit 5=0(data outputs enabled).
ControlPortWrite TestAddress, &HF
TestForSpp = False
DataPortWrite TestAddress, &H55
ByteRead = DataPortRead(TestAddress)
If ByteRead = &H55 Then

```

```

    DataPortWrite TestAddress, &HAA
    ByteRead = DataPortRead(TestAddress)
    If ByteRead = &HAA Then
        TestForSpp = True
    End If
End If
End Function

Function TestPort%(PortIndex%)
'Test for a port's presence, and if it exists, the type of port.
'In order, check for the presence of an ECP, EPP, SPP, and PS/2 port.
'Update the information in the Port array and the display.
Dim EcpModeDescription$
Dim EcpModeValue%
Dim TestAddress%
TestPort = False
EcpExists = False
EppExists = False
SppExists = False
PS2Exists = False
PortType = ""
TestAddress = Port(PortIndex).Address
'Begin by hiding all port details.
frmSelectPort.lblAddress(PortIndex).Visible = False
frmSelectPort.lblType(PortIndex).Visible = False
frmSelectPort.cboEcpMode(PortIndex).Visible = False

EcpExists = TestForEcp(TestAddress)
If EcpExists Then
    PortType = "ECP"
    'Read the current Ecp mode.
    EcpModeValue = ReadEcpMode(TestAddress)
Else
    'If it's not an ECP, look for an EPP.
    'If TestAddress = 3BCh, skip the EPP test.
    'EPPs aren't allowed at 3BCh due to possible conflict
    'with video memory.
    frmSelectPort.cboEcpMode(PortIndex).Visible = False
    If TestAddress = &H3BC Then
        EppExists = False
    Else
        EppExists = TestForEpp(TestAddress)
    End If
    If EppExists Then
        PortType = "EPP"
    Else
        'If it's not an EPP, look for an SPP.
        SppExists = TestForSpp(TestAddress)
        If SppExists Then
            'Test for a PS/2 port only if the SPP exists
            '(because if the port doesn't exist, it will pass the
PS/2 test!)
            PS2Exists = TestForPS2(TestAddress)
            If PS2Exists Then
                PortType = "PS/2"
            Else
                PortType = "SPP"
            End If
        Else
            PortType = "SPP"
        End If
    End If
End If
End Function

```

```

        PortType = ""
    End If

End If
End If

If PortType = "" Then
    frmSelectPort.optPortName(PortIndex).Enabled = False
    Port(PortIndex).PortType = ""
    Port(PortIndex).Address = 0
    Port(PortIndex).Enabled = False
Else
    TestPort = True
    Port(PortIndex).Enabled = True
    Port(PortIndex).PortType = PortType
    Port(PortIndex).Enabled = True
    If EcpExists Then
        Port(PortIndex).EcpModeValue = EcpModeValue
        Port(PortIndex).EcpModeDescription =
GetEcpModeDescription(EcpModeValue)
    End If
End If
UpdateLabels
End Function

Sub UpdateLabels()
'Use the information in the Port array to update the on-screen
display.
Dim Index%
Dim EcpModeValue%
For Index = 0 To 3
    frmSelectPort.lblAddress(Index).Caption =
Hex$(Port(Index).Address) + "h"
    If Port(Index).Enabled = True Then
        frmSelectPort.optPortName(Index).Enabled = True
        frmSelectPort.lblAddress(Index).Visible = True
        frmSelectPort.lblType(Index).Caption = Port(Index).PortType
        frmSelectPort.lblType(Index).Visible = True
        If Port(Index).PortType = "ECP" Then
            EcpModeValue = ReadEcpMode(Port(Index).Address)
            frmSelectPort.cboEcpMode(Index).ListIndex = EcpModeValue
            Port(Index).EcpModeValue = EcpModeValue
            Port(Index).EcpModeDescription =
GetEcpModeDescription(EcpModeValue)
            frmSelectPort.cboEcpMode(Index).Visible = True
        Else
            frmSelectPort.cboEcpMode(Index).Visible = False
        End If
    Else
        frmSelectPort.optPortName(Index).Enabled = False
        frmSelectPort.lblAddress(Index).Visible = False
        frmSelectPort.lblType(Index).Visible = False
        frmSelectPort.cboEcpMode(Index).Visible = False
    End If
End If
Next Index
End Sub

Function VbGetPrivateProfileString$(section$, key$, file$)
    Dim KeyValue$
    Dim Characters%

```

```

        KeyValue = String$(128, 0)
        Characters = GetPrivateProfileStringByKeyName(section, key, "",
        KeyValue, 127, file)
        KeyValue = Left$(KeyValue, Characters)
        VbGetPrivateProfileString = KeyValue
    End Function

Sub WriteIniData()
    Dim BaseAddressWrite%
    Dim PortTypeWrite%
    Dim Index%
    Dim IniWrite%

    'Use Windows API call WritePrivateProfileString to save
    initialization information.
    'If the ini file doesn't exist, it will be created and stored in the
    Windows directory.
    'The port addresses:
    IniWrite = WritePrivateProfileString("lptdata", "Port0Address",
    CStr(Port(0).Address), IniFile)
    IniWrite = WritePrivateProfileString("lptdata", "Port1Address",
    CStr(Port(1).Address), IniFile)
    IniWrite = WritePrivateProfileString("lptdata", "Port2Address",
    CStr(Port(2).Address), IniFile)
    IniWrite = WritePrivateProfileString("lptdata", "Port3Address",
    CStr(Port(3).Address), IniFile)

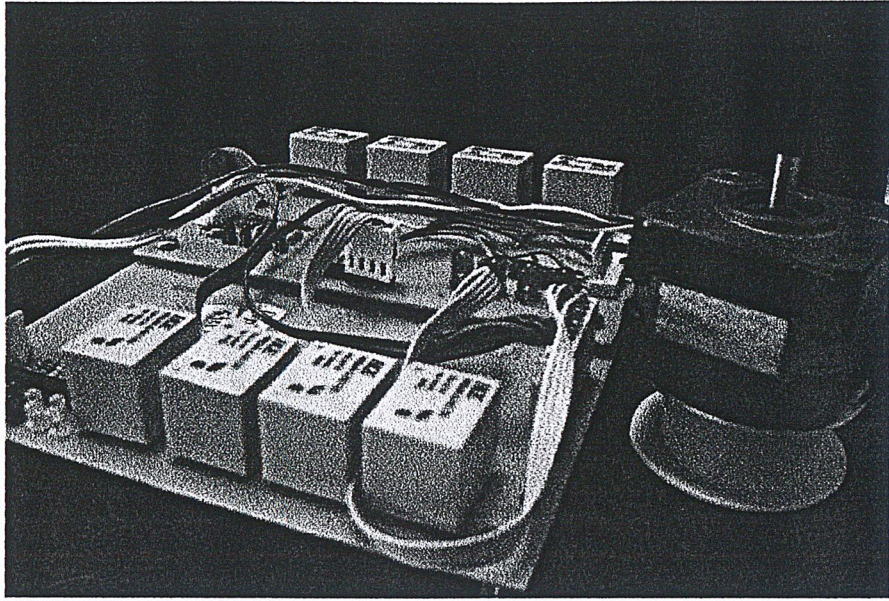
    'The port types:
    IniWrite = WritePrivateProfileString("lptdata", "Port0Type",
    Port(0).PortType, IniFile)
    IniWrite = WritePrivateProfileString("lptdata", "Port1Type",
    Port(1).PortType, IniFile)
    IniWrite = WritePrivateProfileString("lptdata", "Port2Type",
    Port(2).PortType, IniFile)
    IniWrite = WritePrivateProfileString("lptdata", "Port3Type",
    Port(3).PortType, IniFile)

    'Port enabled?
    IniWrite = WritePrivateProfileString("lptdata", "Port0Enabled",
    CStr(Port(0).Enabled), IniFile)
    IniWrite = WritePrivateProfileString("lptdata", "Port1Enabled",
    CStr(Port(1).Enabled), IniFile)
    IniWrite = WritePrivateProfileString("lptdata", "Port2Enabled",
    CStr(Port(2).Enabled), IniFile)
    IniWrite = WritePrivateProfileString("lptdata", "Port3Enabled",
    CStr(Port(3).Enabled), IniFile)
    'Find the selected port and save it:
    Index = 4
    Do
        Index = Index - 1
    Loop Until (frmSelectPort.optPortName(Index).Value = True) Or Index =
    0
    IniWrite = WritePrivateProfileString("lptdata",
    "IndexOfSelectedPort", CStr(Index), IniFile)

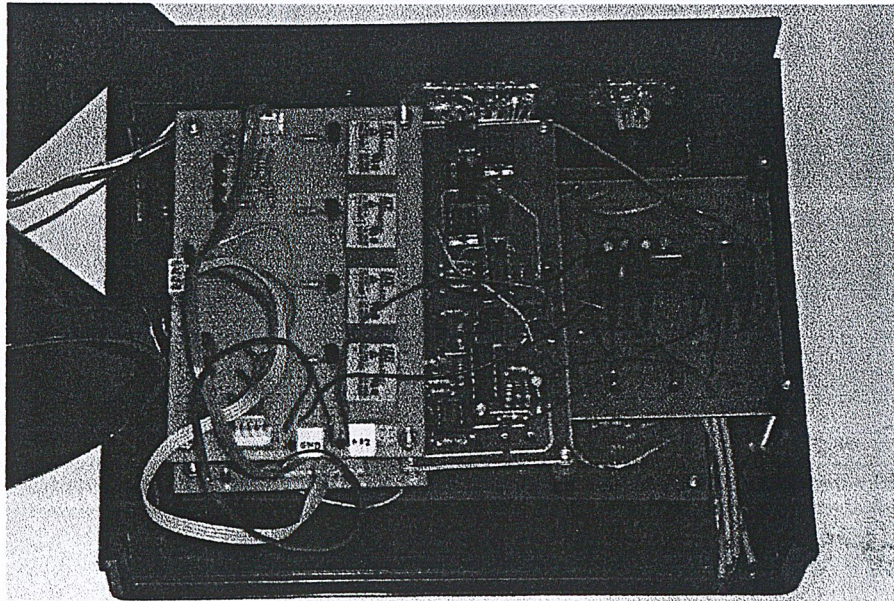
End Sub

```

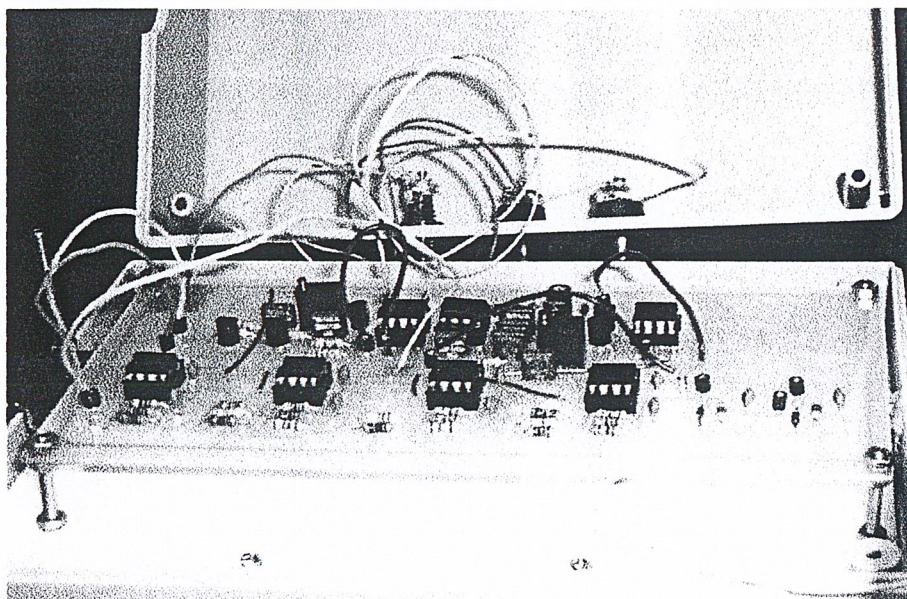
# ภาพชิ้นงาน



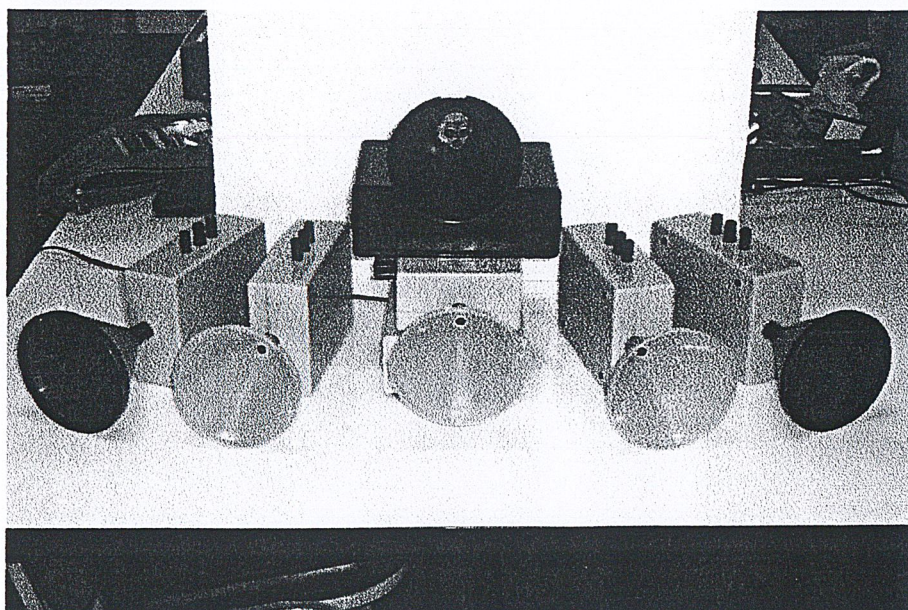
รูปที่ 6.1 แสดงชิ้นงาน ภาคขับสเตปิ่งมอเตอร์



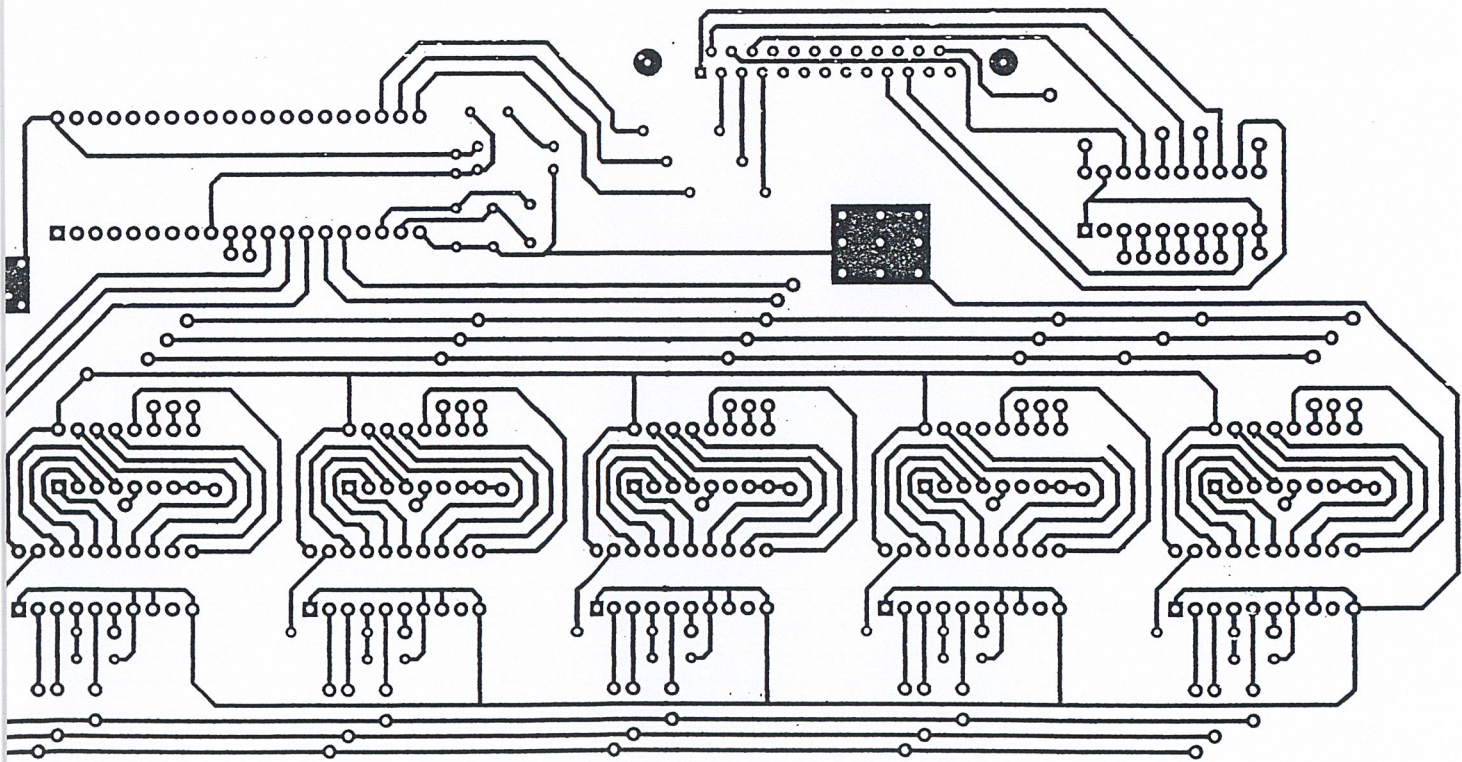
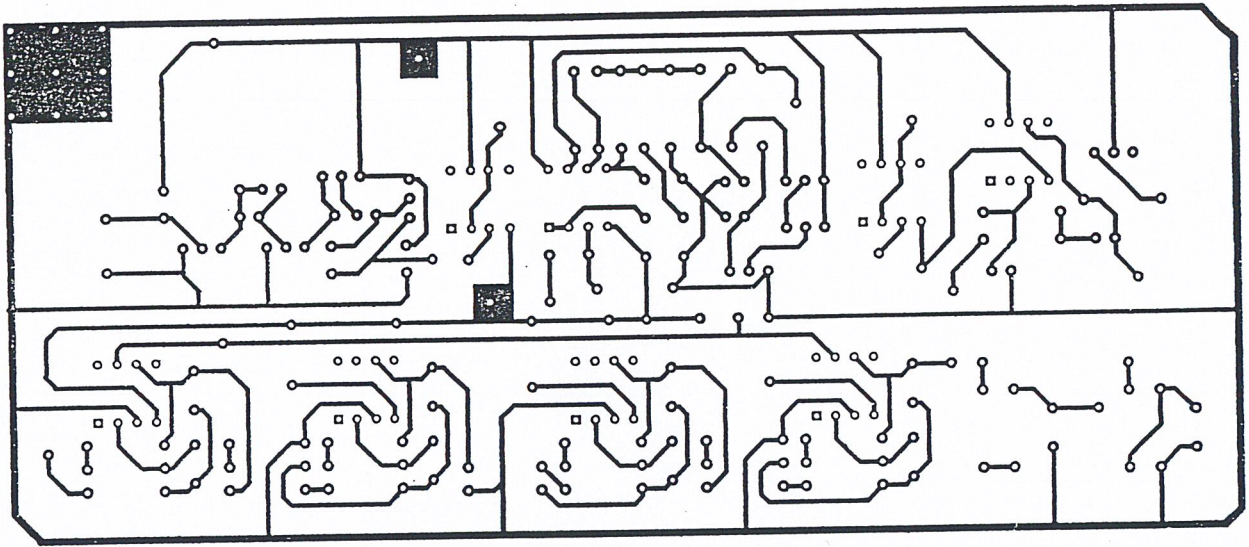
รูปที่ 6.2 แสดงชิ้นงาน ภาคส่ง และส่วนอินเตอร์เฟซ

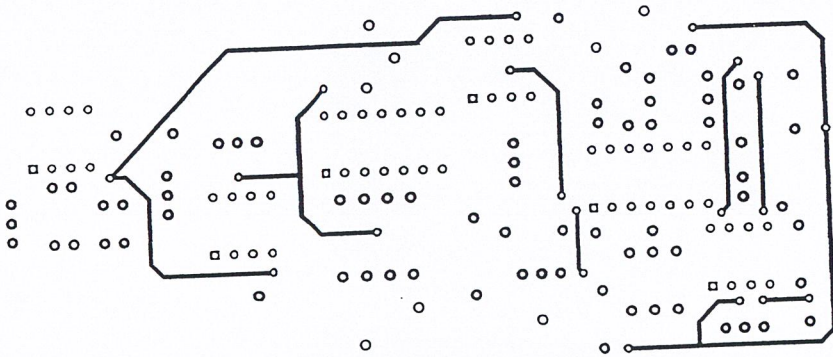
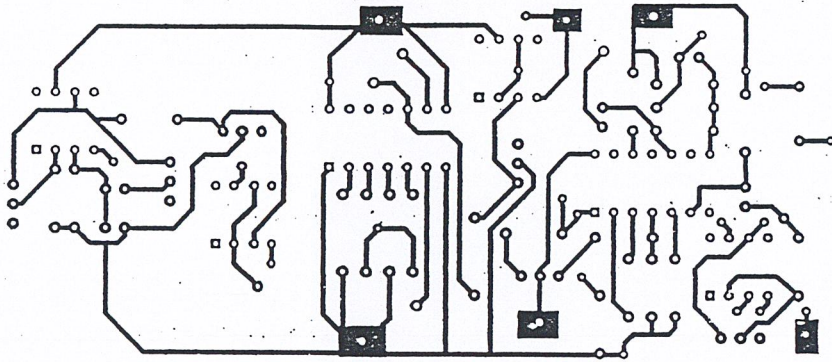
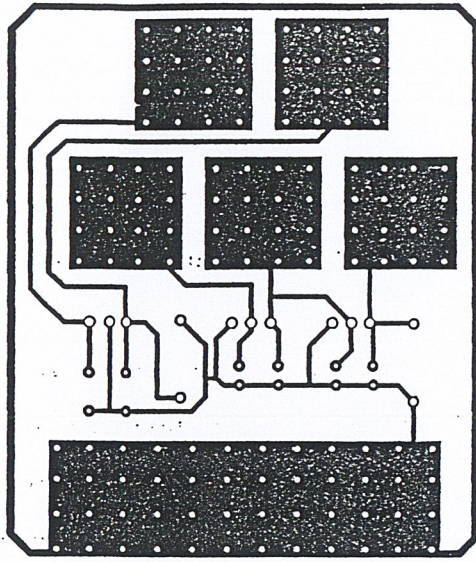


รูปที่ 6.3 แสดงชิ้นงาน ภาครับ



รูปที่ 6.4 แสดงชิ้นงาน โดยรวม





## DM74LS151 1-of-8 Line Data Selector/Multiplexer

### General Description

This data selector/multiplexer contains full on-chip decoding to select the desired data source. The DM74LS151 selects one-of-eight data sources. The DM74LS151 has a strobe input which must be at a low logic level to enable these devices. A high level at the strobe forces the W output HIGH, and the Y output LOW.

The DM74LS151 features complementary W and Y outputs.

### Features

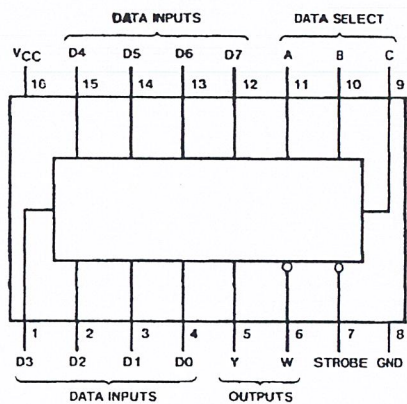
- Select one-of-eight data lines
- Performs parallel-to-serial conversion
- Permits multiplexing from N lines to one line
- Also for use as Boolean function generator
- Typical average propagation delay time data input to W output 12.5 ns
- Typical power dissipation 30 mW

### Ordering Code:

Order Number	Package Number	Package Description
DM74LS151M	M16A	16-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150 Narrow
DM74LS151SJ	M16D	16-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS151N	N16E	16-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

### Connection Diagram



### Truth Table

Inputs				Outputs	
Select			Strobe S	Y	W
C	B	A			
X	X	X	H	L	H
L	L	L	L	D0	$\overline{D0}$
L	L	H	L	D1	$\overline{D1}$
L	H	L	L	D2	$\overline{D2}$
L	H	H	L	D3	$\overline{D3}$
H	L	L	L	D4	$\overline{D4}$
H	L	H	L	D5	$\overline{D5}$
H	H	L	L	D6	$\overline{D6}$
H	H	H	L	D7	$\overline{D7}$

H = HIGH Level  
 L = LOW Level  
 X = Don't Care  
 D0, D1...D7 = the level of the respective D input

DM74LS151 1-of-8 Line Data Selector/Multiplexer

# DM74LS151

## 1-of-8 Line Data Selector/Multiplexer

### General Description

This data selector/multiplexer contains full on-chip decoding to select the desired data source. The DM74LS151 selects one-of-eight data sources. The DM74LS151 has a strobe input which must be at a low logic level to enable these devices. A high level at the strobe forces the W output HIGH, and the Y output LOW.

The DM74LS151 features complementary W and Y outputs.

### Features

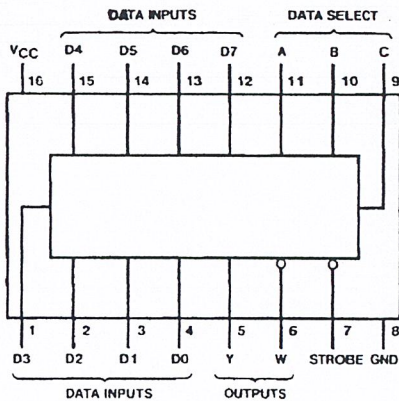
- Select one-of-eight data lines
- Performs parallel-to-serial conversion
- Permits multiplexing from N lines to one line
- Also for use as Boolean function generator
- Typical average propagation delay time data input to W output 12.5 ns
- Typical power dissipation 30 mW

### Ordering Code:

Order Number	Package Number	Package Description
DM74LS151M	M16A	16-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150 Narrow
DM74LS151SJ	M16D	16-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS151N	N16E	16-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

### Connection Diagram



### Truth Table

Inputs				Outputs	
Select			Strobe S	Y	W
C	B	A			
X	X	X	H	L	H
L	L	L	L	D0	$\overline{D0}$
L	L	H	L	D1	$\overline{D1}$
L	H	L	L	D2	$\overline{D2}$
L	H	H	L	D3	$\overline{D3}$
H	L	L	L	D4	$\overline{D4}$
H	L	H	L	D5	$\overline{D5}$
H	H	L	L	D6	$\overline{D6}$
H	H	H	L	D7	$\overline{D7}$

H = HIGH Level  
L = LOW Level  
X = Don't Care  
D0, D1...D7 = the level of the respective D input

## Switching Characteristics

at  $V_{CC} = 5V$  and  $T_A = 25^\circ C$

Symbol	Parameter	From (Input) To (output)	$R_L = 2\text{ k}\Omega$				Units
			$C_L = 15\text{ pF}$		$C_L = 50\text{ pF}$		
			Min	Max	Min	Max	
$t_{PLH}$	Propagation Delay Time LOW-to-HIGH Level Output	Select (4 Levels) to Y		43		46	ns
$t_{PHL}$	Propagation Delay Time HIGH-to-LOW Level Output	Select (4 Levels) to Y		30		36	ns
$t_{PLH}$	Propagation Delay Time LOW-to-HIGH Level Output	Select (3 Levels) to W		23		25	ns
$t_{PHL}$	Propagation Delay Time HIGH-to-LOW Level Output	Select (3 Levels) to W		32		40	ns
$t_{PLH}$	Propagation Delay Time LOW-to-HIGH Level Output	Strobe to Y		42		44	ns
$t_{PHL}$	Propagation Delay Time HIGH-to-LOW Level Output	Strobe to Y		32		40	ns
$t_{PLH}$	Propagation Delay Time LOW-to-HIGH Level Output	Strobe to W		24		27	ns
$t_{PHL}$	Propagation Delay Time HIGH-to-LOW Level Output	Strobe to W		30		36	ns
$t_{PLH}$	Propagation Delay Time LOW-to-HIGH Level Output	D0 thru D7 to Y		32		35	ns
$t_{PHL}$	Propagation Delay Time HIGH-to-LOW Level Output	D0 thru D7 to Y		26		33	ns
$t_{PLH}$	Propagation Delay Time LOW-to-HIGH Level Output	D0 thru D7 to W		21		25	ns
$t_{PHL}$	Propagation Delay Time HIGH-to-LOW Level Output	D0 thru D7 to W		20		27	ns

# LM158, LM158A, LM258, LM258A LM358, LM358A, LM2904, LM2904Q DUAL OPERATIONAL AMPLIFIERS

SLOS068E – JUNE 1976 – REVISED SEPTEMBER 2002

- **Wide Range of Supply Voltages:**
  - Single Supply . . . 3 V to 30 V (LM2904 and LM2904Q . . . 3 V to 26 V) or
  - Dual Supplies
- **Low Supply-Current Drain Independent of Supply Voltage . . . 0.7 mA Typ**
- **Common-Mode Input Voltage Range Includes Ground, Allowing Direct Sensing Near Ground**
- **Low Input Bias and Offset Parameters:**
  - Input Offset Voltage . . . 3 mV Typ  
A Versions . . . 2 mV Typ
  - Input Offset Current . . . 2 nA Typ
  - Input Bias Current . . . 20 nA Typ  
A Versions . . . 15 nA Typ
- **Differential Input Voltage Range Equal to Maximum-Rated Supply Voltage . . .  $\pm 32$  V (LM2904 and LM2904Q . . .  $\pm 26$  V)**
- **Open-Loop Differential Voltage Amplification . . . 100 V/mV Typ**
- **Internal Frequency Compensation**

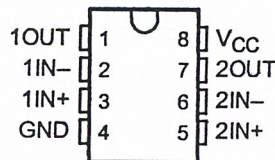
## description/ordering information

These devices consist of two independent, high-gain, frequency-compensated operational amplifiers designed to operate from a single supply over a wide range of voltages. Operation from split supplies also is possible if the difference between the two supplies is 3 V to 30 V (3 V to 26 V for the LM2904 and LM2904Q), and  $V_{CC}$  is at least 1.5 V more positive than the input common-mode voltage. The low supply-current drain is independent of the magnitude of the supply voltage.

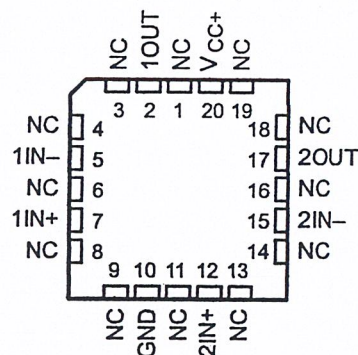
Applications include transducer amplifiers, dc amplification blocks, and all the conventional operational amplifier circuits that now can be implemented more easily in single-supply-voltage systems. For example, these devices can be operated directly from the standard 5-V supply used in digital systems and easily provide the required interface electronics without additional  $\pm 5$ -V supplies.

The LM2904Q is manufactured to demanding automotive requirements.

LM158, LM158A . . . JG PACKAGE  
LM258 . . . D OR P PACKAGE  
LM258A . . . P PACKAGE  
LM358 . . . D, P, PS, OR PW PACKAGE  
LM358A . . . D OR P PACKAGE  
LM2904 . . . D, P, PS, OR PW PACKAGE  
LM2904Q . . . D PACKAGE  
(TOP VIEW)



LM158, LM158A . . . FK PACKAGE  
(TOP VIEW)



NC – No internal connection



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

**TEXAS  
INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 2002, Texas Instruments Incorporated  
On products compliant to MIL-PRF-38535, all parameters are tested unless otherwise noted. On all other products, production processing does not necessarily include testing of all parameters.

**LM158, LM158A, LM258, LM258A**  
**LM358, LM358A, LM2904, LM2904Q**  
**DUAL OPERATIONAL AMPLIFIERS**  
 SLOS068E – JUNE 1976 – REVISED SEPTEMBER 2002

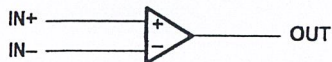
**description/ordering information (continued)**

**ORDERING INFORMATION**

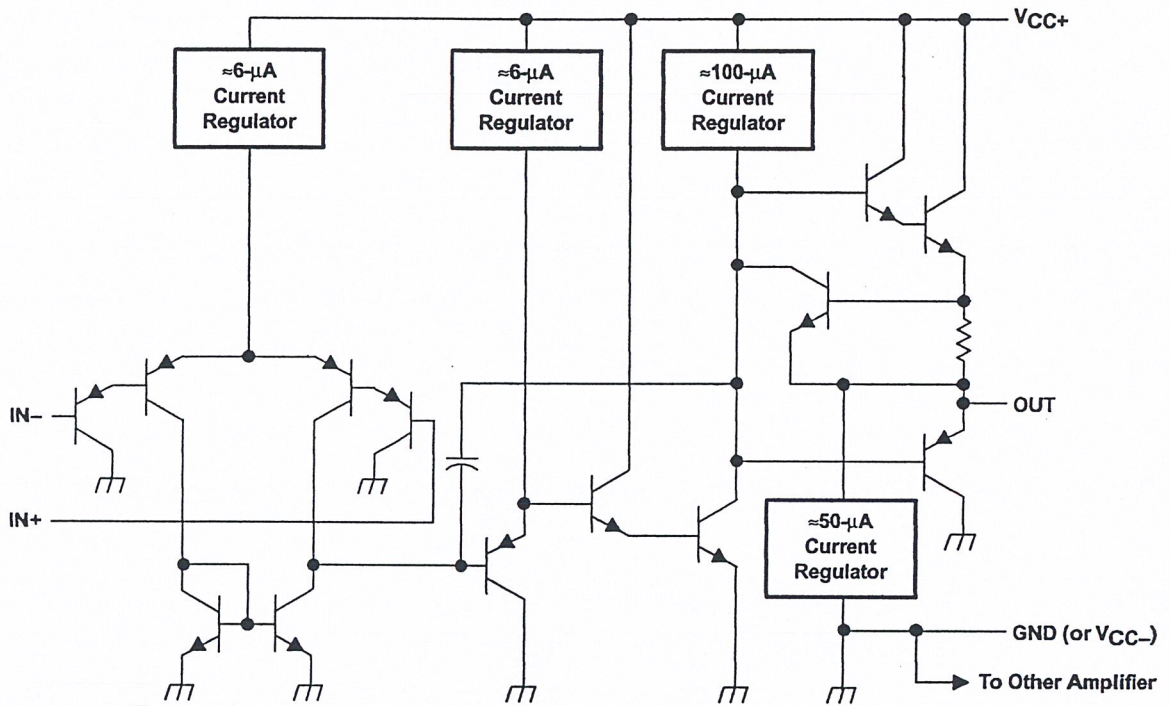
TA	V <sub>IO</sub> max AT 25°C	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	7 mV	PDIP (P)	Tube	LM358P	LM358P
		SOIC (D)	Tube	LM358D	LM358
			Tape and reel	LM358DR	
		SOP (PS)	Tape and reel	LM358PSR	L358
	TSSOP (PW)	Tape and reel	LM358PWR	L358	
	3 mV	PDIP (P)	Tube	LM358AP	LM358AP
SOIC (D)		Tube	LM358AD	LM358A	
		Tape and reel	LM358ADR		
-25°C to 85°C	5 mV	PDIP (P)	Tube	LM258P	LM258P
		SOIC (D)	Tube	LM258D	LM258
			Tape and reel	LM258DR	
	3 mV	PDIP (P)	Tube	LM258AP	LM258AP
-40°C to 125°C	7 mV	PDIP (P)	Tube	LM2904P	LM2904P
		SOIC (D)	Tube	LM2904D	LM2904
			Tape and reel	LM2904DR	
			Tube	LM2904QD	2904Q
			Tape and reel	LM2904QDR	
		SOP (PS)	Tape and reel	LM2904PSR	L2904
TSSOP (PW)	Tape and reel	LM2904PWR	L2904		
-55°C to 125°C	5 mV	CDIP (JG)	Tube	LM158JG	LM158JG
			Tube	LM158JGB	LM158JGB
		LCCC (FK)	Tube	LM158FKB	LM158FKB
	2 mV	CDIP (JG)	Tube	LM158AJG	LM158AJG
			Tube	LM158AJGB	LM158AJGB
		LCCC (FK)	Tube	LM158AFKB	LM158AFKB

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at [www.ti.com/sc/package](http://www.ti.com/sc/package).

**symbol (each amplifier)**



schematic (each amplifier)



COMPONENT COUNT	
Epi-FET	1
Diodes	2
Resistors	7
Transistors	51
Capacitors	2

**LM158, LM158A, LM258, LM258A  
LM358, LM358A, LM2904, LM2904Q  
DUAL OPERATIONAL AMPLIFIERS**

SLOS068E – JUNE 1976 – REVISED SEPTEMBER 2002

**absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†**

		LM158, LM158A LM258, LM258A LM358, LM358A	LM2904 LM2904Q	UNIT
Supply voltage, $V_{CC}$ (see Note 1)		32	26	V
Differential input voltage, $V_{ID}$ (see Note 2)		$\pm 32$	$\pm 26$	V
Input voltage, $V_I$ (either input)		-0.3 to 32	-0.3 to 26	V
Duration of output short circuit (one amplifier) to ground at (or below) 25°C free-air temperature ( $V_{CC} \leq 15$ V) (see Note 3)		Unlimited	Unlimited	
Operating virtual junction temperature, $T_J$		150	150	°C
Package thermal impedance, $\theta_{JA}$ (see Notes 4 and 5)	D package	97	97	°C/W
	P package	85	85	
	PS package	95	95	
	PW package	149	149	
Package thermal impedance, $\theta_{JC}$ (see Notes 6 and 7)	FK package	5.61		°C/W
	JG package	14.5		
Operating free-air temperature range, $T_A$	LM158, LM158A	-55 to 125		°C
	LM258, LM258A	-25 to 85		
	LM358, LM358A	0 to 70		
	LM2904, LM2904Q		-40 to 125	
Case temperature for 60 seconds	FK package	260		°C
Lead temperature 1,6 mm (1/16 inch) from case for 60 seconds	JG package	300	300	°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	P package	260	260	°C
Storage temperature range, $T_{stg}$		-65 to 150	-65 to 150	°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

- NOTES:
1. All voltage values, except differential voltages and  $V_{CC}$  specified for measurement of  $I_{OS}$ , are with respect to the network ground terminal.
  2. Differential voltages are at  $IN+$  with respect to  $IN-$ .
  3. Short circuits from outputs to  $V_{CC}$  can cause excessive heating and eventual destruction.
  4. Maximum power dissipation is a function of  $T_J(\max)$ ,  $\theta_{JA}$ , and  $T_A$ . The maximum allowable power dissipation at any allowable ambient temperature is  $P_D = (T_J(\max) - T_A)/\theta_{JA}$ . Operating at the absolute maximum  $T_J$  of 150°C can affect reliability.
  5. The package thermal impedance is calculated in accordance with JESD 51-7.
  6. Maximum power dissipation is a function of  $T_J(\max)$ ,  $\theta_{JC}$ , and  $T_C$ . The maximum allowable power dissipation at any allowable case temperature is  $P_D = (T_J(\max) - T_C)/\theta_{JC}$ . Operating at the absolute maximum  $T_J$  of 150°C can affect reliability.
  7. The package thermal impedance is calculated in accordance with MIL-STD-883.



**LM158, LM158A, LM258, LM258A**  
**LM358, LM358A, LM2904, LM2904Q**  
**DUAL OPERATIONAL AMPLIFIERS**  
 SLOS068E – JUNE 1976 – REVISED SEPTEMBER 2002

electrical characteristics at specified free-air temperature,  $V_{CC} = 5\text{ V}$  (unless otherwise noted)

PARAMETER	TEST CONDITIONST	$T_A$ ‡	LM158 LM258			LM358			UNIT	
			MIN	TYP§	MAX	MIN	TYP§	MAX		
$V_{IO}$	Input offset voltage	$V_{CC} = 5\text{ V to MAX}$ , $V_{IC} = V_{ICR}(\text{min})$ , $V_O = 1.4\text{ V}$	25°C	3	5	3	7	mV		
			Full range	7			9			
$\alpha_{V_{IO}}$	Average temperature coefficient of input offset voltage	Full range	7			7			$\mu\text{V}/^\circ\text{C}$	
$I_{IO}$	Input offset current	$V_O = 1.4\text{ V}$	25°C	2	30	2	50	nA		
			Full range	100			150			
$\alpha_{I_{IO}}$	Average temperature coefficient of input offset current	Full range	10			10			$\text{pA}/^\circ\text{C}$	
$I_{IB}$	Input bias current	$V_O = 1.4\text{ V}$	25°C	-20	-150	-20	-250	nA		
			Full range	-300			-500			
$V_{ICR}$	Common-mode input voltage range	$V_{CC} = 5\text{ V to MAX}$	25°C	0 to $V_{CC}-1.5$		0 to $V_{CC}-1.5$		V		
			Full range	0 to $V_{CC}-2$		0 to $V_{CC}-2$				
$V_{OH}$	High-level output voltage	$R_L \geq 2\text{ k}\Omega$	25°C	$V_{CC}-1.5$		$V_{CC}-1.5$		V		
			25°C	$V_{CC}-1.5$		$V_{CC}-1.5$				
			Full range	$R_L = 2\text{ k}\Omega$	26		26			
				$R_L \geq 10\text{ k}\Omega$	27	28	27		28	
$V_{OL}$	Low-level output voltage	$R_L \leq 10\text{ k}\Omega$	Full range	5	20	5	20	mV		
$A_{VD}$	Large-signal differential voltage amplification	$V_{CC} = 15\text{ V}$ , $V_O = 1\text{ V to }11\text{ V}$ , $R_L = \geq 2\text{ k}\Omega$	25°C	50	100	25	100	V/mV		
			Full range	25			15			
CMRR	Common-mode rejection ratio	$V_{CC} = 5\text{ V to MAX}$ , $V_{IC} = V_{ICR}(\text{min})$	25°C	70	80	65	80	dB		
$k_{SVR}$	Supply-voltage rejection ratio ( $\Delta V_{DD}/\Delta V_{IO}$ )	$V_{CC} = 5\text{ V to MAX}$	25°C	65	100	65	100	dB		
$V_{O1}/V_{O2}$	Crosstalk attenuation	$f = 1\text{ kHz to }20\text{ kHz}$	25°C	120			120			dB
$I_O$	Output current	$V_{CC} = 15\text{ V}$ , $V_{ID} = 1\text{ V}$ , $V_O = 0$	25°C	-20	-30	-20	-30	mA		
			Full range	-10			-10			
		$V_{CC} = 15\text{ V}$ , $V_{ID} = -1\text{ V}$ , $V_O = 15\text{ V}$	25°C	10	20	10	20			
			Full range	5			5			
		$V_{ID} = -1\text{ V}$ , $V_O = 200\text{ mV}$	25°C	12	30	12	30	$\mu\text{A}$		
$I_{OS}$	Short-circuit output current	$V_{CC}$ at 5 V, GND at -5 V, $V_O = 0$	25°C	$\pm 40$	$\pm 60$	$\pm 40$	$\pm 60$	mA		
$I_{CC}$	Supply current (two amplifiers)	$V_O = 2.5\text{ V}$ , No load	Full range	0.7	1.2	0.7	1.2	mA		
		$V_{CC} = \text{MAX}$ , $V_O = 0.5\text{ V}$ , No load	Full range	1	2	1	2			

† All characteristics are measured under open-loop conditions with zero common-mode input voltage, unless otherwise specified. MAX  $V_{CC}$  for testing purposes is 26 V for the LM2904 and 30 V for others.

‡ Full range is  $-55^\circ\text{C}$  to  $125^\circ\text{C}$  for LM158,  $-25^\circ\text{C}$  to  $85^\circ\text{C}$  for LM258,  $0^\circ\text{C}$  to  $70^\circ\text{C}$  for LM358, and  $-40^\circ\text{C}$  to  $125^\circ\text{C}$  for LM2904 and LM2904Q.

§ All typical values are at  $T_A = 25^\circ\text{C}$ .



**LM158, LM158A, LM258, LM258A  
LM358, LM358A, LM2904, LM2904Q  
DUAL OPERATIONAL AMPLIFIERS**

SLOS068E – JUNE 1976 – REVISED SEPTEMBER 2002

electrical characteristics at specified free-air temperature,  $V_{CC} = 5\text{ V}$  (unless otherwise noted)

PARAMETER	TEST CONDITIONST	$T_A$ ‡	LM2904 LM2904Q		UNIT	
			MIN	TYP§		MAX
$V_{IO}$ Input offset voltage	$V_{CC} = 5\text{ V to MAX,}$ $V_{IC} = V_{ICR(\text{min})}, V_O = 1.4\text{ V}$	25°C	3	7	mV	
		Full range		10		
$\alpha_{V_{IO}}$ Average temperature coefficient of input offset voltage		Full range	7		$\mu\text{V}/^\circ\text{C}$	
$I_{IO}$ Input offset current	$V_O = 1.4\text{ V}$	25°C	2	50	nA	
		Full range		300		
$\alpha_{I_{IO}}$ Average temperature coefficient of input offset current		Full range	10		$\text{pA}/^\circ\text{C}$	
$I_{IB}$ Input bias current	$V_O = 1.4\text{ V}$	25°C	-20	-250	nA	
		Full range		-500		
$V_{ICR}$ Common-mode input voltage range	$V_{CC} = 5\text{ V to MAX}$	25°C	0 to $V_{CC}-1.5$		V	
		Full range	0 to $V_{CC}-2$			
$V_{OH}$ High-level output voltage	$R_L \geq 2\text{ k}\Omega$	25°C			V	
		25°C	$V_{CC}-1.5$			
		Full range	$R_L = 2\text{ k}\Omega$	26		
			$R_L \geq 10\text{ k}\Omega$	23		24
$V_{OL}$ Low-level output voltage	$R_L \leq 10\text{ k}\Omega$	Full range	5	20	mV	
$A_{VD}$ Large-signal differential voltage amplification	$V_{CC} = 15\text{ V}, V_O = 1\text{ V to }11\text{ V,}$ $R_L = \geq 2\text{ k}\Omega$	25°C	25	100	V/mV	
		Full range	15			
CMRR Common-mode rejection ratio	$V_{CC} = 5\text{ V to MAX,}$ $V_{IC} = V_{ICR(\text{min})}$	25°C	50	80	dB	
$k_{SVR}$ Supply-voltage rejection ratio ( $\Delta V_{DD}/\Delta V_{IO}$ )	$V_{CC} = 5\text{ V to MAX}$	25°C	65	100	dB	
$V_{O1}/V_{O2}$ Crosstalk attenuation	$f = 1\text{ kHz to }20\text{ kHz}$	25°C		120	dB	
$I_O$ Output current	$V_{CC} = 15\text{ V}, V_{ID} = 1\text{ V}, V_O = 0$	25°C	-20	-30	mA	
		Full range	-10			
	$V_{CC} = 15\text{ V}, V_{ID} = -1\text{ V}, V_O = 15\text{ V}$	25°C	10	20		
		Full range	5			
$I_{OS}$ Short-circuit output current	$V_{CC}$ at 5 V, GND at -5 V, $V_O = 0$	25°C	±40	±60	mA	
$I_{CC}$ Supply current (two amplifiers)	$V_O = 2.5\text{ V,}$ No load	Full range	0.7	1.2	mA	
	$V_{CC} = \text{MAX,}$ $V_O = 0.5\text{ V,}$ No load	Full range	1	2		

† All characteristics are measured under open-loop conditions with zero common-mode input voltage, unless otherwise specified. MAX  $V_{CC}$  for testing purposes is 26 V for the LM2904 and 30 V for others.

‡ Full range is  $-55^\circ\text{C}$  to  $125^\circ\text{C}$  for LM158,  $-25^\circ\text{C}$  to  $85^\circ\text{C}$  for LM258,  $0^\circ\text{C}$  to  $70^\circ\text{C}$  for LM358, and  $-40^\circ\text{C}$  to  $125^\circ\text{C}$  for LM2904 and LM2904Q.

§ All typical values are at  $T_A = 25^\circ\text{C}$ .



**LM158, LM158A, LM258, LM258A**  
**LM358, LM358A, LM2904, LM2904Q**  
**DUAL OPERATIONAL AMPLIFIERS**  
SLOS068E – JUNE 1976 – REVISED SEPTEMBER 2002

**electrical characteristics at specified free-air temperature,  $V_{CC} = 5\text{ V}$  (unless otherwise noted)**

PARAMETER	TEST CONDITION <sup>†</sup>	$T_A$ <sup>‡</sup>	LM158A			LM258A			UNIT	
			MIN	TYP <sup>§</sup>	MAX	MIN	TYP <sup>§</sup>	MAX		
$V_{IO}$	Input offset voltage $V_{CC} = 5\text{ V to }30\text{ V}$ , $V_{IC} = V_{ICR(\text{min})}$ , $V_O = 1.4\text{ V}$	25°C			2		2	3	mV	
		Full range			4			4		
$\alpha_{V_{IO}}$	Average temperature coefficient of input offset voltage	Full range		7	15*		7	15	$\mu\text{V}/^\circ\text{C}$	
$I_{IO}$	Input offset current $V_O = 1.4\text{ V}$	25°C		2	10		2	15	nA	
		Full range			30			30		
$\alpha_{I_{IO}}$	Average temperature coefficient of input offset current	Full range		10	200		10	200	$\text{pA}/^\circ\text{C}$	
$I_{IB}$	Input bias current $V_O = 1.4\text{ V}$	25°C		-15	-50		-15	-80	nA	
		Full range			-100			-100		
$V_{ICR}$	Common-mode input voltage range $V_{CC} = 30\text{ V}$	25°C		0 to $V_{CC}-1.5$			0 to $V_{CC}-1.5$		V	
		Full range		0 to $V_{CC}-2$			0 to $V_{CC}-2$			
$V_{OH}$	High-level output voltage $R_L \geq 2\text{ k}\Omega$ $V_{CC} = 30\text{ V}$	25°C		$V_{CC}-1.5$			$V_{CC}-1.5$		V	
		Full range	$R_L = 2\text{ k}\Omega$	26		26				
			$R_L \geq 10\text{ k}\Omega$	27	28	27	28			
$V_{OL}$	Low-level output voltage $R_L \leq 10\text{ k}\Omega$	Full range		5	20		5	20	mV	
$A_{VD}$	Large-signal differential voltage amplification $V_{CC} = 15\text{ V}$ , $V_O = 1\text{ V to }11\text{ V}$ , $R_L \geq 2\text{ k}\Omega$	25°C	50	100		50	100		V/mV	
		Full range	25			25				
CMRR	Common-mode rejection ratio	25°C	70	80		70	80		dB	
kSVR	Supply-voltage rejection ratio ( $\Delta V_{DD}/\Delta V_{IO}$ )	25°C	65	100		65	100		dB	
$V_{O1}/V_{O2}$	Crosstalk attenuation $f = 1\text{ kHz to }20\text{ kHz}$	25°C		120			120		dB	
$I_O$	Output current $V_{CC} = 15\text{ V}$ , $V_{ID} = 1\text{ V}$ , $V_O = 0$	25°C	-20	-30	-60	-20	-30	-60	mA	
		Full range	-10			-10				
		25°C	$V_{CC} = 15\text{ V}$ , $V_{ID} = -1\text{ V}$ , $V_O = 15$		10	20		10	20	$\mu\text{A}$
				Full range	5			5		
$I_{OS}$	Short-circuit output current $V_{CC}$ at 5 V, GND at -5 V, $V_O = 0$	25°C		$\pm 40$	$\pm 60$		$\pm 40$	$\pm 60$	mA	
		Full range		0.7	1.2		0.7	1.2		
$I_{CC}$	Supply current (two amplifiers) $V_O = 2.5\text{ V}$ , No load	Full range		0.7	1.2		0.7	1.2	mA	
		Full range		1	2		1	2		

\*On products compliant to MIL-PRF-38535, this parameter is not production tested.

<sup>†</sup> All characteristics are measured under open-loop conditions with zero common-mode input voltage, unless otherwise specified. MAX  $V_{CC}$  for testing purposes is 26 V for LM2904 and 30 V for others.

<sup>‡</sup> Full range is -55°C to 125°C for LM158A, -25°C to 85°C for LM258A, and 0°C to 70°C for LM358A.

<sup>§</sup> All typical values are at  $T_A = 25^\circ\text{C}$ .



**LM158, LM158A, LM258, LM258A  
LM358, LM358A, LM2904, LM2904Q  
DUAL OPERATIONAL AMPLIFIERS**  
SLOS068E – JUNE 1976 – REVISED SEPTEMBER 2002

electrical characteristics at specified free-air temperature,  $V_{CC} = 5\text{ V}$  (unless otherwise noted)

PARAMETER	TEST CONDITION <sup>†</sup>	$T_A$ <sup>‡</sup>	LM358A			UNIT
			MIN	TYP <sup>§</sup>	MAX	
$V_{IO}$ Input offset voltage	$V_{CC} = 5\text{ V to }30\text{ V}$ , $V_{IC} = V_{ICR(\text{min})}$ , $V_O = 1.4\text{ V}$	25°C		2	3	mV
		Full range			5	
$\alpha_{V_{IO}}$ Average temperature coefficient of input offset voltage		Full range		7	20	$\mu\text{V}/^\circ\text{C}$
$I_{IO}$ Input offset current	$V_O = 1.4\text{ V}$	25°C		2	30	nA
		Full range			75	
$\alpha_{I_{IO}}$ Average temperature coefficient of input offset current		Full range		10	300	$\text{pA}/^\circ\text{C}$
$I_{IB}$ Input bias current	$V_O = 1.4\text{ V}$	25°C		-15	-100	nA
		Full range			-200	
$V_{ICR}$ Common-mode input voltage range	$V_{CC} = 30\text{ V}$	25°C		0 to $V_{CC}-1.5$		V
		Full range		0 to $V_{CC}-2$		
$V_{OH}$ High-level output voltage	$R_L \geq 2\text{ k}\Omega$ $V_{CC} = 30\text{ V}$	25°C		$V_{CC}-1.5$		V
		Full range	$R_L = 2\text{ k}\Omega$	26		
			$R_L \geq 10\text{ k}\Omega$	27	28	
$V_{OL}$ Low-level output voltage	$R_L \leq 10\text{ k}\Omega$	Full range		5	20	mV
$A_{VD}$ Large-signal differential voltage amplification	$V_{CC} = 15\text{ V}$ , $V_O = 1\text{ V to }11\text{ V}$ , $R_L \geq 2\text{ k}\Omega$	25°C		25	100	V/mV
		Full range		15		
CMRR Common-mode rejection ratio		25°C		65	80	dB
$k_{SVR}$ Supply-voltage rejection ratio ( $\Delta V_{DD}/\Delta V_{IO}$ )		25°C		65	100	dB
$V_{O1}/V_{O2}$ Crosstalk attenuation	$f = 1\text{ kHz to }20\text{ kHz}$	25°C		120		dB
$I_O$ Output current	$V_{CC} = 15\text{ V}$ , $V_{ID} = 1\text{ V}$ , $V_O = 0$	25°C		-20	-30	mA
		Full range		-10		
	$V_{CC} = 15\text{ V}$ , $V_{ID} = -1\text{ V}$ , $V_O = 15\text{ V}$	25°C		10	20	
		Full range		5		
	$V_{ID} = -1\text{ V}$ , $V_O = 200\text{ mV}$	25°C		30		$\mu\text{A}$
$I_{OS}$ Short-circuit output current	$V_{CC}$ at 5 V, GND at -5 V, $V_O = 0$	25°C		$\pm 40$	$\pm 60$	mA
$I_{CC}$ Supply current (two amplifiers)	$V_O = 2.5\text{ V}$ , No load	Full range		0.7	1.2	mA
	$V_{CC} = \text{MAX}$ , $V_O = 0.5\text{ V}$ No load	Full range		1	2	

<sup>†</sup> All characteristics are measured under open-loop conditions with zero common-mode input voltage, unless otherwise specified. MAX  $V_{CC}$  for testing purposes is 26 V for LM2904 and 30 V for others.

<sup>‡</sup> Full range is -55°C to 125°C for LM158A, -25°C to 85°C for LM258A, and 0°C to 70°C for LM358A.

<sup>§</sup> All typical values are at  $T_A = 25^\circ\text{C}$ .



LM158, LM158A, LM258, LM258A  
 LM358, LM358A, LM2904, LM2904Q  
 DUAL OPERATIONAL AMPLIFIERS  
 SLOS068E – JUNE 1976 – REVISED SEPTEMBER 2002

operating conditions,  $V_{CC} = \pm 15\text{ V}$ ,  $T_A = 25^\circ\text{C}$

PARAMETER		TEST CONDITIONS	TYP	UNIT
SR	Slew rate at unity gain	$R_L = 1\text{ M}\Omega$ , $C_L = 30\text{ pF}$ , $V_I = \pm 10\text{ V}$ (see Figure 1)	0.3	$\text{V}/\mu\text{s}$
$B_1$	Unity-gain bandwidth	$R_L = 1\text{ M}\Omega$ , $C_L = 20\text{ pF}$ (see Figure 1)	0.7	MHz
$V_n$	Equivalent input noise voltage	$R_S = 100\ \Omega$ , $V_I = 0\text{ V}$ , $f = 1\text{ kHz}$ (see Figure 2)	40	$\text{nV}/\sqrt{\text{Hz}}$

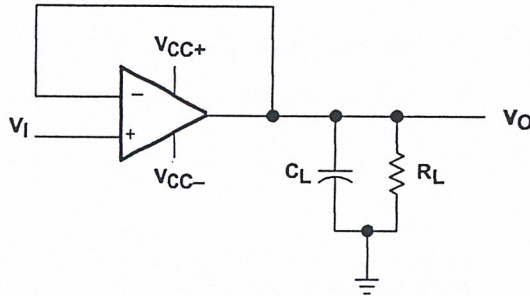


Figure 1. Unity-Gain Amplifier

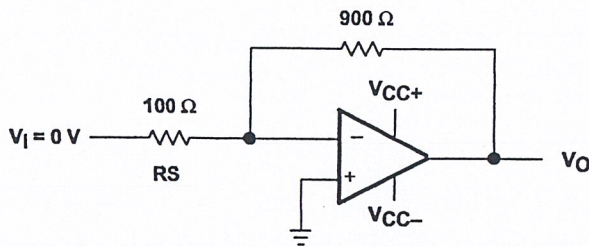


Figure 2. Noise-Test Circuit

## MM74HCT240 • MM74HCT244

### Inverting Octal 3-STATE Buffer • Octal 3-STATE Buffer

#### General Description

The MM74HCT240 and MM74HCT244 3-STATE buffers utilize advanced silicon-gate CMOS technology and are general purpose high speed inverting and non-inverting buffers. They possess high drive current outputs which enable high speed operation even when driving large bus capacitances. These circuits achieve speeds comparable to low power Schottky devices, while retaining the low power consumption of CMOS. All three devices are TTL input compatible and have a fanout of 15 LS-TTL equivalent inputs.

MM74HCT devices are intended to interface between TTL and NMOS components and standard CMOS devices. These parts are also plug-in replacements for LS-TTL devices and can be used to reduce power consumption in existing designs.

The MM74HCT240 is an inverting buffer and the MM74HCT244 is a non-inverting buffer. Each device has two active low enables (1G and 2G), and each enable independently controls 4 buffers.

All inputs are protected from damage due to static discharge by diodes to  $V_{CC}$  and Ground.

#### Features

- TTL input compatible
- Typical propagation delay: 14 ns
- 3-STATE outputs for connection to system buses
- Low quiescent current: 80  $\mu$ A
- High output drive current: 6 mA (min)

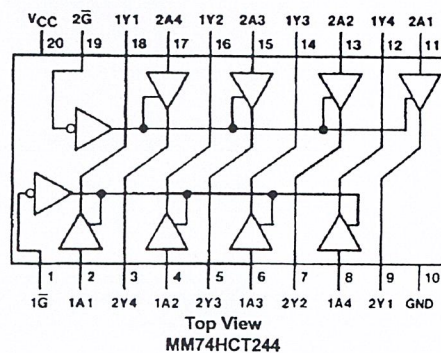
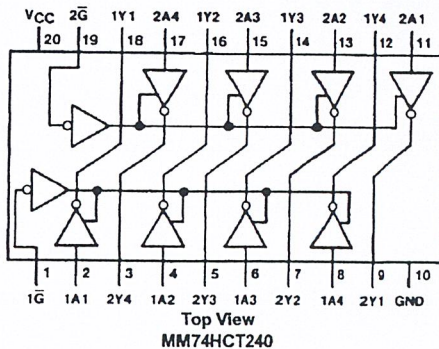
#### Ordering Code:

Order Number	Package Number	Package Description
MM74HCT240WM	M20B	20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300" Wide
MM74HCT240SJ	M20D	20-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
MM74HCT240MTC	MTC20	20-Lead Thin Shrink Small Outline Package (TSSOP), JEDEC MO-153, 4.4mm Wide
MM74HCT240N	N20A	20-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide
MM74HCT244WM	M20B	20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300" Wide
MM74HCT244SJ	M20D	20-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
MM74HCT244MTC	MTC20	20-Lead Thin Shrink Small Outline Package (TSSOP), JEDEC MO-153, 4.4mm Wide
MM74HCT244N	N20A	20-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

#### Connection Diagrams

Pin Assignments for DIP, SOIC, SOP and TSSOP



### Truth Tables

MM74HCT240

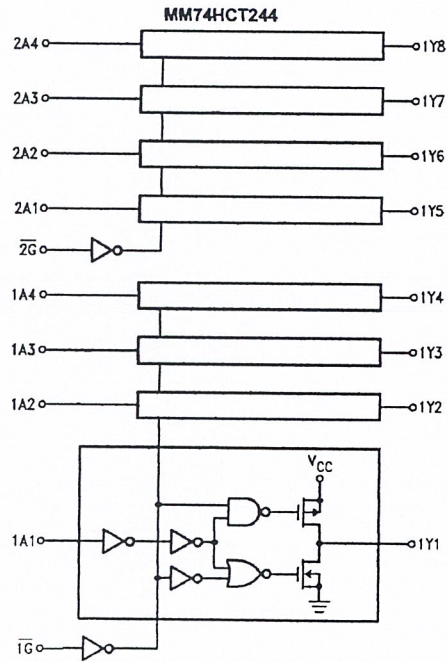
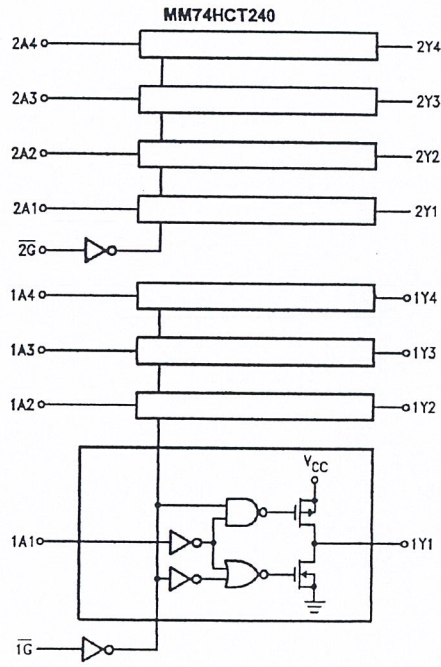
$\overline{1G}$	1A	1Y	$\overline{2G}$	2A	2Y
L	L	H	L	L	H
L	H	L	L	H	L
H	L	Z	H	L	Z
H	H	Z	H	H	Z

H = HIGH Level  
 L = LOW Level  
 Z = High Impedance

MM74HCT244

$\overline{1G}$	1A	1Y	$\overline{2G}$	2A	2Y
L	L	L	L	L	L
L	H	H	L	H	H
H	L	Z	H	L	Z
H	H	Z	H	H	Z

### Logic Diagrams



### Absolute Maximum Ratings (Note 1)

(Note 2)

Supply Voltage ( $V_{CC}$ )	-0.5 to +7.0V
DC Input Voltage ( $V_{IN}$ )	-1.5 to $V_{CC} + 1.5V$
DC Output Voltage ( $V_{OUT}$ )	-0.5 to $V_{CC} + 0.5V$
Clamp Diode Current ( $I_{IK}, I_{OK}$ )	$\pm 20$ mA
DC Output Current, per pin ( $I_{OUT}$ )	$\pm 35$ mA
DC $V_{CC}$ or GND Current, per pin ( $I_{CC}$ )	$\pm 70$ mA
Storage Temperature Range ( $T_{STG}$ )	-65°C to +150°C
Power Dissipation ( $P_D$ )	
(Note 3)	600 mW
S.O. Package only	500 mW
Lead Temperature ( $T_L$ )	
(Soldering 10 seconds)	260°C

### Recommended Operating Conditions

	Min	Max	Units
Supply Voltage ( $V_{CC}$ )	4.5	5.5	V
DC Input or Output Voltage ( $V_{IN}, V_{OUT}$ )	0	$V_{CC}$	V
Operating Temperature Range ( $T_A$ )	-40	+85	°C
Input Rise or Fall Times ( $t_r, t_f$ )		500	ns

Note 1: Absolute Maximum Ratings are those values beyond which damage to the device may occur.

Note 2: Unless otherwise specified all voltages are referenced to ground.

Note 3: Power Dissipation temperature derating — plastic "N" package: -12 mW/°C from 65°C to 85°C.

### DC Electrical Characteristics

$V_{CC} = 5V \pm 10\%$  (unless otherwise specified)

Symbol	Parameter	Conditions	$T_A = 25^\circ C$			Units	
			Typ	Guaranteed Limits			
$V_{IH}$	Minimum HIGH Level Input Voltage			2.0	2.0	2.0	V
$V_{IL}$	Maximum LOW Level Input Voltage			0.8	0.8	0.8	V
$V_{OH}$	Minimum HIGH Level Output Voltage	$V_{IN-EE} = V_{IH}$ or $V_{IL}$	$V_{CC}$	$V_{CC}-0.1$	$V_{CC}-0.1$	$V_{CC}-0.1$	V
		$ I_{OUT}  = 20 \mu A$	4.2	3.98	3.84	3.7	V
		$ I_{OUT}  = 6.0$ mA, $V_{CC} = 4.5V$	5.2	4.98	4.84	4.7	V
$V_{OL}$	Maximum LOW Level Voltage	$V_{IN} = V_{IH}$ or $V_{IL}$	0	0.1	0.1	0.1	V
		$ I_{OUT}  = 20 \mu A$	0.2	0.26	0.33	0.4	V
		$ I_{OUT}  = 6.0$ mA, $V_{CC} = 4.5V$	0.2	0.26	0.33	0.4	V
$I_{IN}$	Maximum Input Current	$V_{IN} = V_{CC}$ or GND, $V_{IH}$ or $V_{IL}$		$\pm 0.05$	$\pm 0.5$	$\pm 1.0$	$\mu A$
$I_{OZ}$	Maximum 3-STATE Output Leakage Current	$V_{OUT} = V_{CC}$ or GND $\bar{G} = V_{IH}$ $G = V_{IL}$		$\pm 0.25$	$\pm 2.5$	$\pm 10$	$\mu A$
$I_{CC}$	Maximum Quiescent Supply Current	$V_{IN} = V_{CC}$ or GND $I_{OUT} = 0 \mu A$		4.0	40	160	$\mu A$
		$V_{IN} = 2.4V$ or $0.5V$ (Note 4)	0.6	1.0	1.3	1.5	mA

Note 4: Measured per Input. All other inputs at  $V_{CC}$  or GND.

### AC Electrical Characteristics

MM74HCT240, MM74HCT244  $V_{CC} = 5.0V$ ,  $t_r = t_f = 6$  ns,  $T_A = 25^\circ C$  (unless otherwise specified)

Symbol	Parameter	Conditions	Typ	Guaranteed Limits	Units
$t_{PHL}$ , $t_{PLH}$	Maximum Output Propagation Delay	$C_L = 45$ pF	14	18	ns
$t_{PZL}$ , $t_{PZH}$	Maximum Output Enable Time	$C_L = 45$ pF $R_L = 1$ k $\Omega$	20	30	ns
$t_{PLZ}$ , $t_{PHZ}$	Maximum Output Disable Time	$C_L = 5$ pF $R_L = 1$ k $\Omega$	16	25	ns

### AC Electrical Characteristics

MM74HCT240, MM74HCT244  $V_{CC} = 5.0V \pm 10\%$ ,  $t_r = t_f = 6$  ns (unless otherwise specified)

Symbol	Parameter	Conditions	$T_A = 25^\circ C$		$T_A = -40$ to $85^\circ C$	$T_A = -55^\circ$ to $125^\circ C$	Units	
			Typ	Guaranteed Limits				
$t_{PHL}$ , $t_{PLH}$	Maximum Output Propagation Delay	$C_L = 50$ pF	14	20	25	30	ns	
		$C_L = 150$ pF	20	28	35	42	ns	
$t_{PZH}$ , $t_{PZL}$	Maximum Output Enable Time	$R_L = 1$ k $\Omega$	$C_L = 50$ pF	21	30	38	45	ns
			$C_L = 150$ pF	26	42	53	63	ns
$t_{PHZ}$ , $t_{PLZ}$	Maximum Output Disable Time	$R_L = 1$ k $\Omega$		16	25	32	38	ns
		$C_L = 50$ pF						
$t_{THL}$ , $t_{TLH}$	Maximum Output Rise and Fall Time	$C_L = 50$ pF	6	12	15	18	ns	
$C_{IN}$	Maximum Input Capacitance		10	15	15	15	pF	
$C_{OUT}$	Maximum Output Capacitance		15	20	20	20	pF	
$C_{PD}$	Power Dissipation Capacitance (Note 5)	(per buffer) $\bar{G} = V_{CC}$ , $G = GND$	5				pF	
		$\bar{G} = GND$ , $G = V_{CC}$	90				pF	

Note 5:  $C_{PD}$  determines the no load dynamic power consumption,  $P_D = C_{PD} V_{CC}^2 f + I_{CC} V_{CC}$ , and the no load dynamic current consumption,  $I_S = C_{PD} V_{CC} f + I_{CC}$ .

# ADC0801/ADC0802/ADC0803/ADC0804/ADC0805

## 8-Bit $\mu$ P Compatible A/D Converters

### General Description

The ADC0801, ADC0802, ADC0803, ADC0804 and ADC0805 are CMOS 8-bit successive approximation A/D converters that use a differential potentiometric ladder—similar to the 256R products. These converters are designed to allow operation with the NSC800 and INS8080A derivative control bus with TRI-STATE output latches directly driving the data bus. These A/Ds appear like memory locations or I/O ports to the microprocessor and no interfacing logic is needed.

Differential analog voltage inputs allow increasing the common-mode rejection and offsetting the analog zero input voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

### Features

- Compatible with 8080  $\mu$ P derivatives—no interfacing logic needed - access time - 135 ns
- Easy interface to all microprocessors, or operates "stand alone"

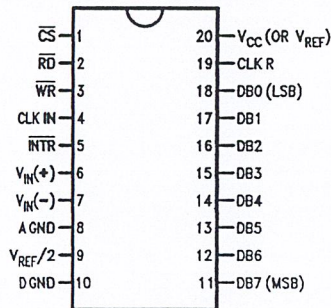
- Differential analog voltage inputs
- Logic inputs and outputs meet both MOS and TTL voltage level specifications
- Works with 2.5V (LM336) voltage reference
- On-chip clock generator
- 0V to 5V analog input voltage range with single 5V supply
- No zero adjust required
- 0.3" standard width 20-pin DIP package
- 20-pin molded chip carrier or small outline package
- Operates ratiometrically or with 5  $V_{DC}$ , 2.5  $V_{DC}$ , or analog span adjusted voltage reference

### Key Specifications

- Resolution 8 bits
- Total error  $\pm 1/4$  LSB,  $\pm 1/2$  LSB and  $\pm 1$  LSB
- Conversion time 100  $\mu$ s

### Connection Diagram

ADC080X  
Dual-In-Line and Small Outline (SO) Packages



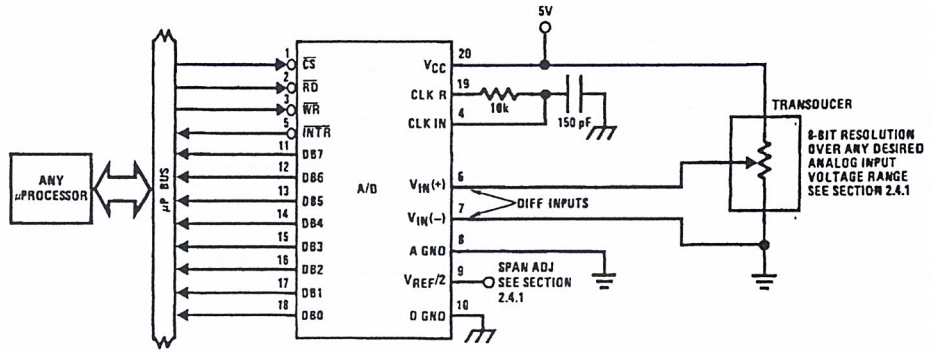
See Ordering Information

### Ordering Information

TEMP RANGE		0°C TO 70°C	0°C TO 70°C	-40°C TO +85°C
ERROR	$\pm 1/4$ Bit Adjusted			ADC0801LCN
	$\pm 1/2$ Bit Unadjusted	ADC0802LCWM		ADC0802LCN
	$\pm 1/2$ Bit Adjusted			ADC0803LCN
	$\pm 1$ Bit Unadjusted	ADC0804LCWM	ADC0804LCN	ADC0805LCN/ADC0804LCJ
PACKAGE OUTLINE		M20B—Small Outline	N20A—Molded DIP	

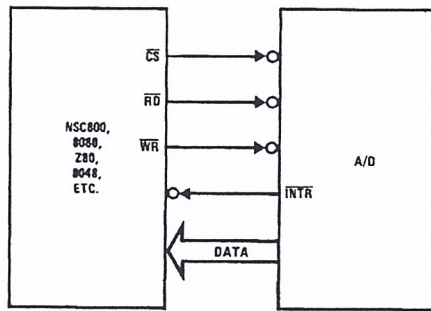
Z-80\* is a registered trademark of Zilog Corp.

## Typical Applications



DS005671-1

### 8080 Interface



DS005671-31

Error Specification (Includes Full-Scale, Zero Error, and Non-Linearity)			
Part Number	Full-Scale Adjusted	$V_{REF/2}=2.500 V_{DC}$ (No Adjustments)	$V_{REF/2}$ =No Connection (No Adjustments)
ADC0801	$\pm 1/4$ LSB		
ADC0802		$\pm 1/2$ LSB	
ADC0803	$\pm 1/2$ LSB		
ADC0804		$\pm 1$ LSB	
ADC0805			$\pm 1$ LSB

### Absolute Maximum Ratings (Notes 1, 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage ( $V_{CC}$ ) (Note 3)	6.5V
Voltage	
Logic Control Inputs	-0.3V to +18V
At Other Input and Outputs	-0.3V to ( $V_{CC}+0.3V$ )
Lead Temp. (Soldering, 10 seconds)	
Dual-In-Line Package (plastic)	260°C
Dual-In-Line Package (ceramic)	300°C
Surface Mount Package	
Vapor Phase (60 seconds)	215°C

Infrared (15 seconds)	220°C
Storage Temperature Range	-65°C to +150°C
Package Dissipation at $T_A=25^\circ\text{C}$	875 mW
ESD Susceptibility (Note 10)	800V

### Operating Ratings (Notes 1, 2)

Temperature Range	$T_{MIN} \leq T_A \leq T_{MAX}$
ADC0804LCJ	-40°C $\leq T_A \leq$ +85°C
ADC0801/02/03/05LCN	-40°C $\leq T_A \leq$ +85°C
ADC0804LCN	0°C $\leq T_A \leq$ +70°C
ADC0802/04LCWM	0°C $\leq T_A \leq$ +70°C
Range of $V_{CC}$	4.5 $V_{DC}$ to 6.3 $V_{DC}$

### Electrical Characteristics

The following specifications apply for  $V_{CC}=5 V_{DC}$ ,  $T_{MIN} \leq T_A \leq T_{MAX}$  and  $f_{CLK}=640$  kHz unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
ADC0801: Total Adjusted Error (Note 8)	With Full-Scale Adj. (See Section 2.5.2)			$\pm 1/4$	LSB
ADC0802: Total Unadjusted Error (Note 8)	$V_{REF}/2=2.500 V_{DC}$			$\pm 1/2$	LSB
ADC0803: Total Adjusted Error (Note 8)	With Full-Scale Adj. (See Section 2.5.2)			$\pm 1/2$	LSB
ADC0804: Total Unadjusted Error (Note 8)	$V_{REF}/2=2.500 V_{DC}$			$\pm 1$	LSB
ADC0805: Total Unadjusted Error (Note 8)	$V_{REF}/2$ -No Connection			$\pm 1$	LSB
$V_{REF}/2$ Input Resistance (Pin 9)	ADC0801/02/03/05	2.5	8.0		k $\Omega$
	ADC0804 (Note 9)	0.75	1.1		k $\Omega$
Analog Input Voltage Range	(Note 4) $V(+)$ or $V(-)$	Gnd-0.05		$V_{CC}+0.05$	$V_{DC}$
DC Common-Mode Error	Over Analog Input Voltage Range		$\pm 1/16$	$\pm 1/8$	LSB
Power Supply Sensitivity	$V_{CC}=5 V_{DC} \pm 10\%$ Over Allowed $V_{IN}(+)$ and $V_{IN}(-)$ Voltage Range (Note 4)		$\pm 1/16$	$\pm 1/8$	LSB

### AC Electrical Characteristics

The following specifications apply for  $V_{CC}=5 V_{DC}$  and  $T_{MIN} \leq T_A \leq T_{MAX}$  unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
$T_C$	Conversion Time	$f_{CLK}=640$ kHz (Note 6)	103		114	$\mu\text{s}$
$T_C$	Conversion Time	(Notes 5, 6)	66		73	$1/f_{CLK}$
$f_{CLK}$	Clock Frequency	$V_{CC}=5V$ , (Note 5)	100	640	1460	kHz
	Clock Duty Cycle		40		60	%
CR	Conversion Rate in Free-Running Mode	$\overline{INTR}$ tied to $\overline{WR}$ with $\overline{CS}=0 V_{DC}$ , $f_{CLK}=640$ kHz	8770		9708	conv/s
$t_{W(WR)L}$	Width of $\overline{WR}$ Input (Start Pulse Width)	$\overline{CS}=0 V_{DC}$ (Note 7)	100			ns
$t_{ACC}$	Access Time (Delay from Falling Edge of $\overline{RD}$ to Output Data Valid)	$C_L=100$ pF		135	200	ns
$t_{1H}, t_{0H}$	TRI-STATE Control (Delay from Rising Edge of $\overline{RD}$ to Hi-Z State)	$C_L=10$ pF, $R_L=10k$ (See TRI-STATE Test Circuits)		125	200	ns
$t_{WI}, t_{RI}$	Delay from Falling Edge of $\overline{WR}$ or $\overline{RD}$ to Reset of $\overline{INTR}$			300	450	ns
$C_{IN}$	Input Capacitance of Logic Control Inputs			5	7.5	pF

## AC Electrical Characteristics (Continued)

The following specifications apply for  $V_{CC}=5 V_{DC}$  and  $T_{MIN} \leq T_A \leq T_{MAX}$  unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
$C_{OUT}$	TRI-STATE Output Capacitance (Data Buffers)			5	7.5	pF
<b>CONTROL INPUTS</b> [Note: CLK IN (Pin 4) is the input of a Schmitt trigger circuit and is therefore specified separately]						
$V_{IN}(1)$	Logical "1" Input Voltage (Except Pin 4 CLK IN)	$V_{CC}=5.25 V_{DC}$	2.0		15	$V_{DC}$
$V_{IN}(0)$	Logical "0" Input Voltage (Except Pin 4 CLK IN)	$V_{CC}=4.75 V_{DC}$			0.8	$V_{DC}$
$I_{IN}(1)$	Logical "1" Input Current (All Inputs)	$V_{IN}=5 V_{DC}$		0.005	1	$\mu A_{DC}$
$I_{IN}(0)$	Logical "0" Input Current (All Inputs)	$V_{IN}=0 V_{DC}$	-1	-0.005		$\mu A_{DC}$
<b>CLOCK IN AND CLOCK R</b>						
$V_{T+}$	CLK IN (Pin 4) Positive Going Threshold Voltage		2.7	3.1	3.5	$V_{DC}$
$V_{T-}$	CLK IN (Pin 4) Negative Going Threshold Voltage		1.5	1.8	2.1	$V_{DC}$
$V_H$	CLK IN (Pin 4) Hysteresis ( $V_{T+}$ )-(V <sub>T-</sub> )		0.6	1.3	2.0	$V_{DC}$
$V_{OUT}(0)$	Logical "0" CLK R Output Voltage	$I_O=360 \mu A$ $V_{CC}=4.75 V_{DC}$			0.4	$V_{DC}$
$V_{OUT}(1)$	Logical "1" CLK R Output Voltage	$I_O=-360 \mu A$ $V_{CC}=4.75 V_{DC}$	2.4			$V_{DC}$
<b>DATA OUTPUTS AND INTR</b>						
$V_{OUT}(0)$	Logical "0" Output Voltage Data Outputs INTR Output	$I_{OUT}=1.6 mA, V_{CC}=4.75 V_{DC}$ $I_{OUT}=1.0 mA, V_{CC}=4.75 V_{DC}$			0.4 0.4	$V_{DC}$ $V_{DC}$
$V_{OUT}(1)$	Logical "1" Output Voltage	$I_O=-360 \mu A, V_{CC}=4.75 V_{DC}$	2.4			$V_{DC}$
$V_{OUT}(1)$	Logical "1" Output Voltage	$I_O=-10 \mu A, V_{CC}=4.75 V_{DC}$	4.5			$V_{DC}$
$I_{OUT}$	TRI-STATE Disabled Output Leakage (All Data Buffers)	$V_{OUT}=0 V_{DC}$ $V_{OUT}=5 V_{DC}$	-3		3	$\mu A_{DC}$ $\mu A_{DC}$
$I_{SOURCE}$		$V_{OUT}$ Short to Gnd, $T_A=25^\circ C$	4.5	6		$mA_{DC}$
$I_{SINK}$		$V_{OUT}$ Short to $V_{CC}$ , $T_A=25^\circ C$	9.0	16		$mA_{DC}$
<b>POWER SUPPLY</b>						
$I_{CC}$	Supply Current (Includes Ladder Current)  ADC0801/02/03/04LCJ/05 ADC0804LCN/LCWM	$f_{CLK}=640 kHz,$ $V_{REF}/2=NC, T_A=25^\circ C$ and $\overline{CS}=5V$			1.1 1.9	1.8 2.5 mA mA

**Note 1:** Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

**Note 2:** All voltages are measured with respect to Gnd, unless otherwise specified. The separate A Gnd point should always be wired to the D Gnd.

**Note 3:** A zener diode exists, internally, from  $V_{CC}$  to Gnd and has a typical breakdown voltage of  $7 V_{DC}$ .

**Note 4:** For  $V_{IN(-)} \rightarrow V_{IN(+)}$  the digital output code will be 0000 0000. Two on-chip diodes are tied to each analog input (see block diagram) which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the  $V_{CC}$  supply. Be careful, during testing at low  $V_{CC}$  levels (4.5V), as high level analog inputs (5V) can cause this input diode to conduct—especially at elevated temperatures, and cause errors for analog inputs near full-scale. The spec allows 50 mV forward bias of either diode. This means that as long as the analog  $V_{IN}$  does not exceed the supply voltage by more than 50 mV, the output code will be correct. To achieve an absolute 0  $V_{DC}$  to 5  $V_{DC}$  input voltage range will therefore require a minimum supply voltage of 4.950  $V_{DC}$  over temperature variations, initial tolerance and loading.

**Note 5:** Accuracy is guaranteed at  $f_{CLK} = 640 kHz$ . At higher clock frequencies accuracy can degrade. For lower clock frequencies, the duty cycle limits can be extended so long as the minimum clock high time interval or minimum clock low time interval is no less than 275 ns.

**Note 6:** With an asynchronous start pulse, up to 8 clock periods may be required before the internal clock phases are proper to start the conversion process. The start request is internally latched, see Figure 4 and section 2.0.

## AC Electrical Characteristics (Continued)

**Note 7:** The  $\overline{CS}$  input is assumed to bracket the  $\overline{WR}$  strobe input and therefore timing is dependent on the  $\overline{WR}$  pulse width. An arbitrarily wide pulse width will hold the converter in a reset mode and the start of conversion is initiated by the low to high transition of the  $\overline{WR}$  pulse (see timing diagrams).

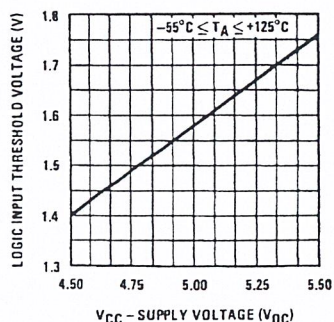
**Note 8:** None of these A/Ds requires a zero adjust (see section 2.5.1). To obtain zero code at other analog input voltages see section 2.5 and Figure 7.

**Note 9:** The  $V_{REF/2}$  pin is the center point of a two-resistor divider connected from  $V_{CC}$  to ground. In all versions of the ADC0801, ADC0802, ADC0803, and ADC0805, and in the ADC0804LCJ, each resistor is typically 16 k $\Omega$ . In all versions of the ADC0804 except the ADC0804LCJ, each resistor is typically 2.2 k $\Omega$ .

**Note 10:** Human body model, 100 pF discharged through a 1.5 k $\Omega$  resistor.

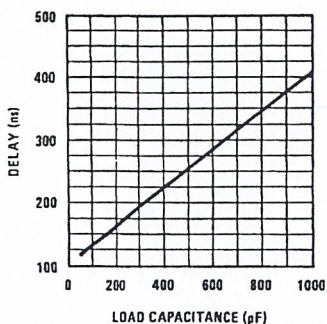
## Typical Performance Characteristics

**Logic Input Threshold Voltage vs. Supply Voltage**



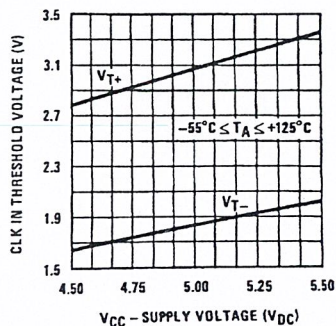
DS005671-38

**Delay From Falling Edge of RD to Output Data Valid vs. Load Capacitance**



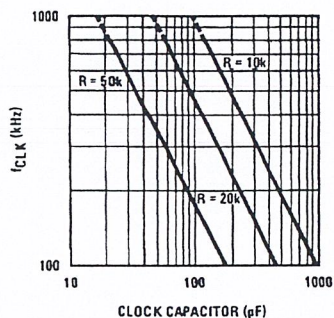
DS005671-39

**CLK IN Schmitt Trip Levels vs. Supply Voltage**



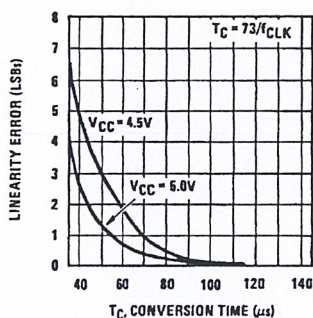
DS005671-40

**f<sub>CLK</sub> vs. Clock Capacitor**



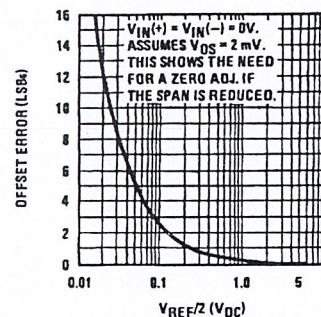
DS005671-41

**Full-Scale Error vs Conversion Time**



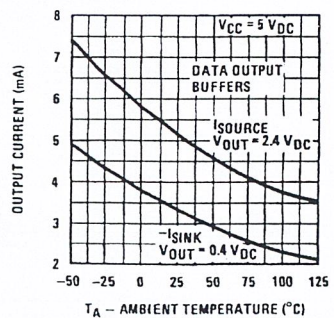
DS005671-42

**Effect of Unadjusted Offset Error vs. V<sub>REF/2</sub> Voltage**



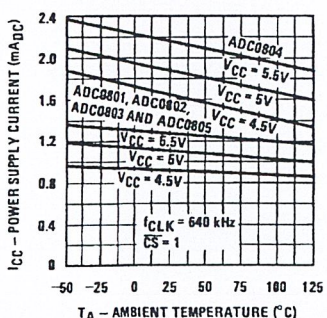
DS005671-43

**Output Current vs Temperature**



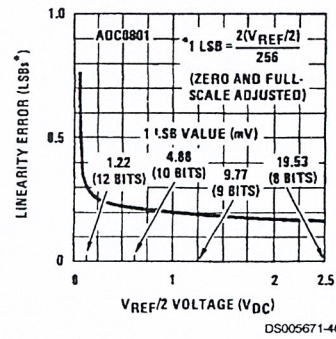
DS005671-44

**Power Supply Current vs Temperature (Note 9)**



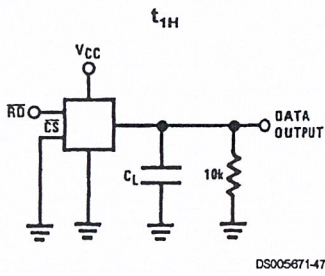
DS005671-45

**Linearity Error at Low V<sub>REF/2</sub> Voltages**

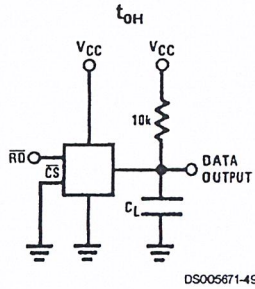
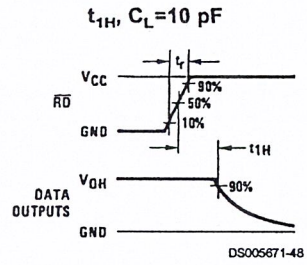


DS005671-46

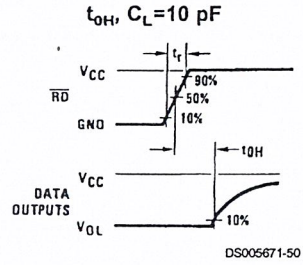
### TRI-STATE Test Circuits and Waveforms



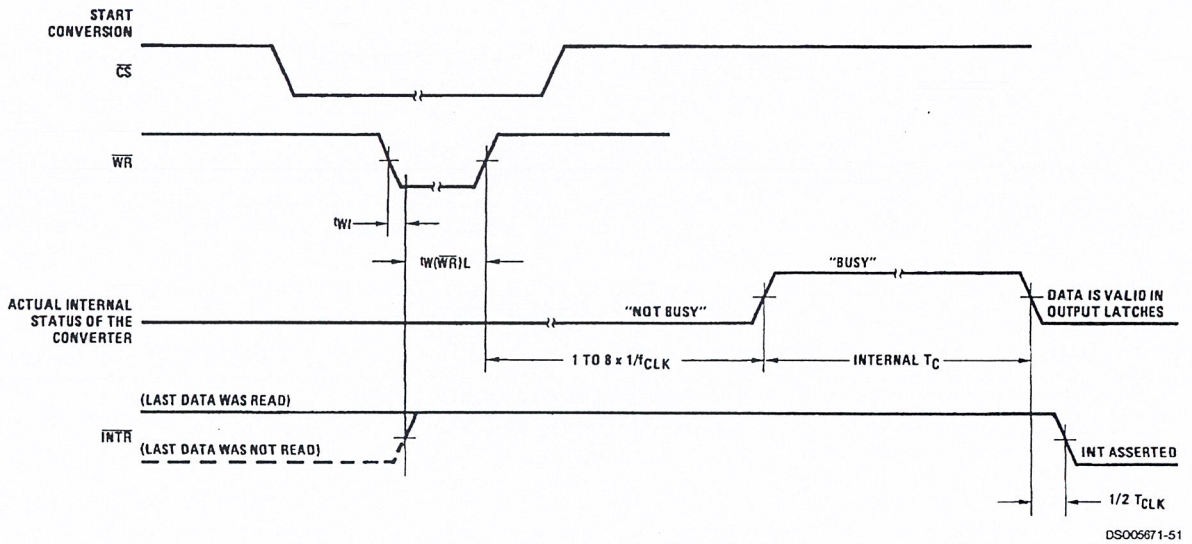
$t_r = 20 \text{ ns}$



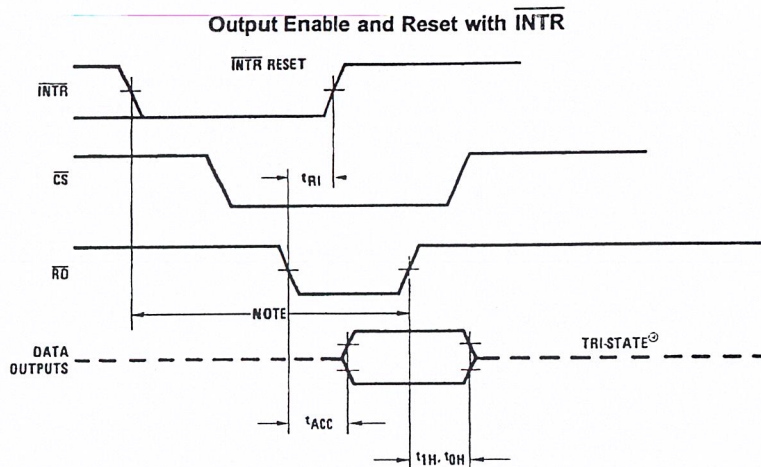
$t_r = 20 \text{ ns}$



### Timing Diagrams (All timing is measured from the 50% voltage points)



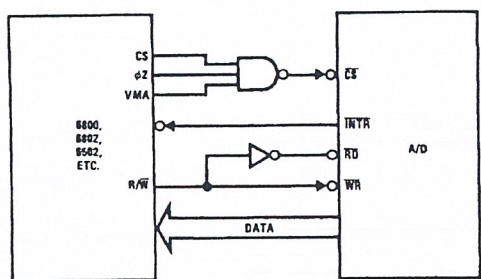
## Timing Diagrams (All timing is measured from the 50% voltage points) (Continued)



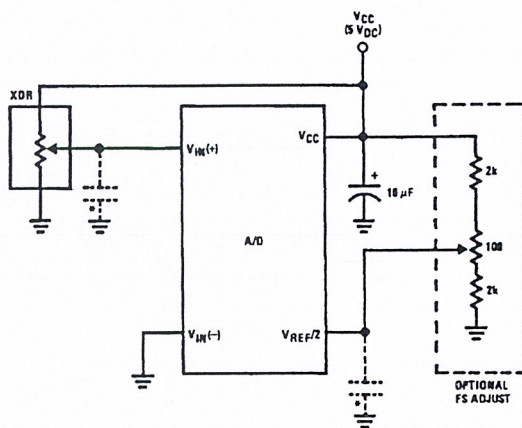
Note: Read strobe must occur 8 clock periods ( $8/f_{\text{CLK}}$ ) after assertion of interrupt to guarantee reset of  $\overline{\text{INTR}}$ .

## Typical Applications

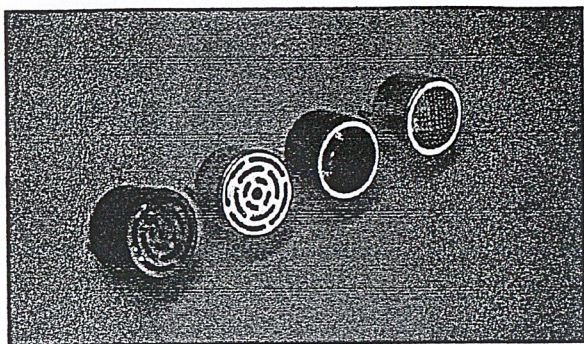
6800 Interface



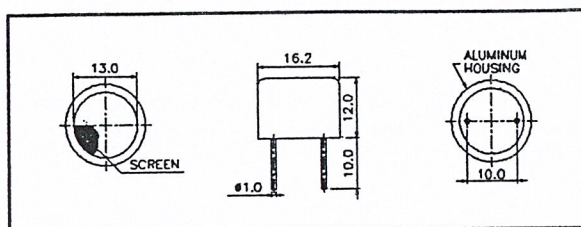
Ratiometric with Full-Scale Adjust



Note: before using caps at  $V_{\text{IN}}$  or  $V_{\text{REF}/2}$ , see section 2.3.2 Input Bypass Capacitors.



Dimensions: dimensions are in mm



Specification

<b>400ST160</b>	Transmitter
<b>400SR160</b>	Receiver
<b>Center Frequency</b>	40.0±1.0Khz
<b>Bandwidth (-6dB)</b>	400ST160: 2.0Khz 400SR160: 2.5Khz
<b>Transmitting Sound Pressure Level</b> at 40.0Khz; 0dB re 0.0002µbar per 10Vrms at 30cm	120dB min.
<b>Receiving Sensitivity</b> at 40.0Khz 0dB = 1 volt/µbar	-65dB min.
<b>Capacitance at 1Khz ±20%</b>	2400 pF
<b>Max. Driving Voltage (cont.)</b>	20Vrms
<b>Total Beam Angle -6dB</b>	55° typical
<b>Operation Temperature</b>	-40 to 85°C
<b>Storage Temperature</b>	-40 to 85°C

All specification taken typical at 25°C  
Closer frequency tolerance can be supplied upon request.

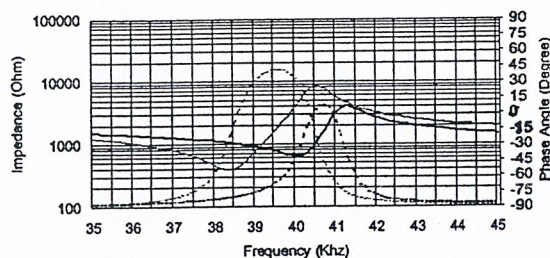
Models available:

1	400ST/R160	Aluminum Housing
2	400ST/R16B	Black Al. Housing
2	400ST/R10P	Plastic Housing
3	400ST/R16F	Al. Housing w/Solid Grid

Impedance/Phase Angle vs. Frequency

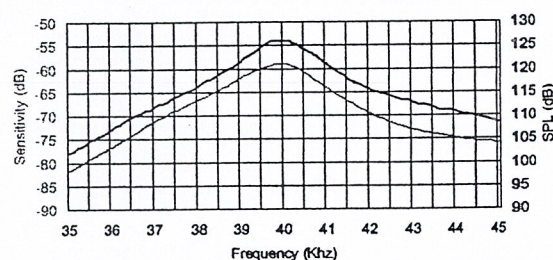
Tested under 1Vrms Oscillation Level

400SR160 Impedance \_\_\_\_\_  
 400SR160 Phase \_\_\_\_\_  
 400ST160 Impedance \_\_\_\_\_  
 400ST160 Phase \_\_\_\_\_

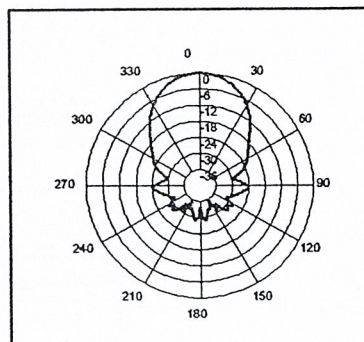


Sensitivity/Sound Pressure Level

Tested under 10Vrms @30cm



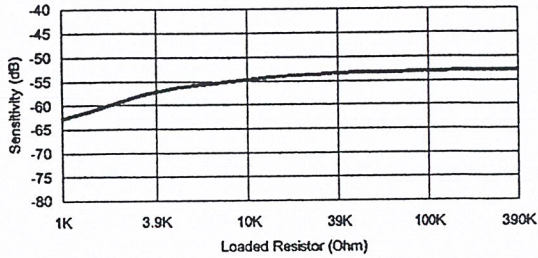
Beam Angle: Tested at 40.0Khz frequency



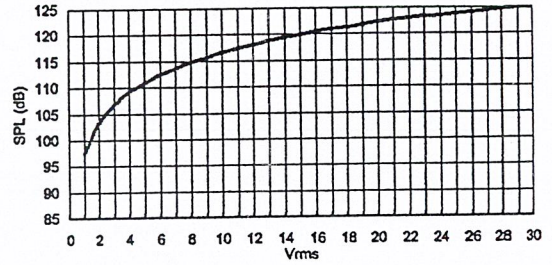
**400SR160 Receiver**

**400ST160 Transmitter**

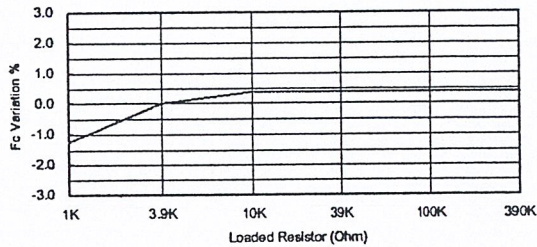
**Sensitivity Variation vs. Loaded Resistor**



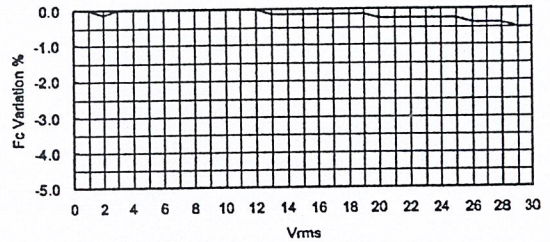
**SPL Variation vs. Driving Voltage**



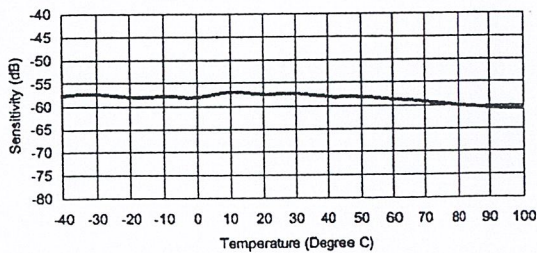
**Center Frequency Shift vs. Loaded Resistor**



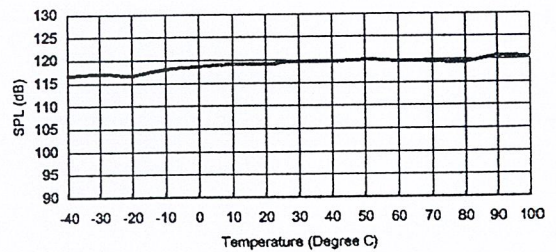
**Center Frequency Shift vs. Driving Voltage**



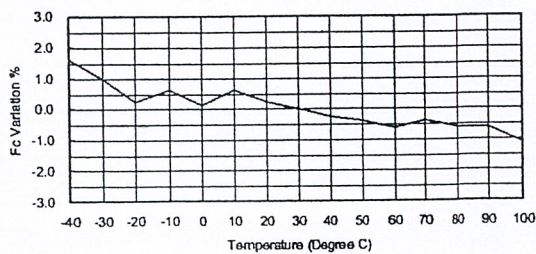
**Sensitivity Variation vs. Temperature**



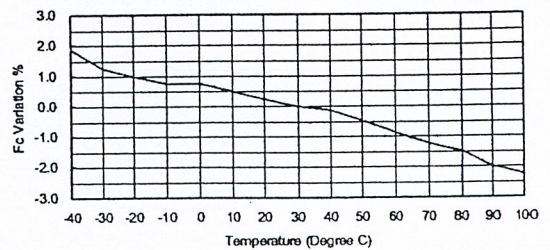
**SPL Variation vs. Temperature**



**Center Frequency Shift vs. Temperature**



**Center Frequency Shift vs. Temperature**



## หนังสืออ้างอิง

1. Sedra / Smith , “Microelectronic Circuits” , Oxford University Press , 1998
2. James Bignell / Robert Donovan . “Digital Electronic” , Delmar Publisher Inc. , 1994
3. รศ. สมยศ จุณณะปิยะ , “การประยุกต์ใช้งานไมโครคอนโทรลเลอร์” , สจล. , พ.ศ.2543
4. วรพจน์ กรแก้ววัฒนกุล / ชัยวัฒน์ ถิมพรจิตวิไล , “เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช” , บริษัท อินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด
5. จูรพงษ์ นามแดง , “Hobby Electronic – ตลับเมตรไร้สาย” , SE-ED Publishing , พ.ศ. พ.ศ.2540
6. นิรุช อำนวยศิลป์ , “คู่มือการเขียนโปรแกรม Microsoft Visual C++ Version 6.0” , บริษัท ชัคเชส มีเดีย จำกัด.
7. พีรภัทร์ สว่างเพ็ชร , “เทคนิคการเขียนโปรแกรมและเกมด้วย Visual C++” , บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน) .
8. ชาริน สิทธิธรรมขารี , “คู่มือเขียนโปรแกรม Microsoft Visual Basic Version 6.0” , บริษัท ชัคเชส มีเดีย จำกัด.
9. Jan Axelson , “Parallel Port Complete” Lakeview Research Madison, WI 53704, 2000
10. M.E.VAN VALKENBURG, “Analog Filter Design”, CBS College Publishing