การออกแบบโปรแกรมและวงจรกรองสัญญาณความถี่แบบเรียลไทม์

FREQUENCY FILTER DESIGN AND APPLICATION IN REAL TIME

นายภัทรพงศ์ เพชรรัตน์ รหัส 48380242

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 12 ก.ย. 2556
เลขทะเบียน 16434514
เลขเรียกหนังสือ ปร.
มหาวิทยาลัยนเรศวร ภ3แก
2555

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2555

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน ออกแบบโปรแกรมและวงจรกรองสัญญาณความถี่แบบเรียลไทม์

ผู้ดำเนินโครงงาน นายภัทรพงศ์ เพชรรัตน์ รหัส 48380242

ที่ปรึกษาโครงงาน ดร.มุฑิตา สงฆ์จันทร์

สาขาวิชา วิศวกรรมไฟฟ้า

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2555

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....ที่ปรึกษาโครงงาน

(ดร.มุฑิตา สงฆ์จันทร์)

.....กรรมการ

(ผศ.ดร.ศุภวรรณ พลพิทักษ์ชัย)

.....กรรมการ

(ดร.ชัยรัตน์ พินทอง)

ชื่อหัวข้อโครงงาน	การออกแบบโปรแกรมและวงจรกรองสัญญาณความถี่แบบเรียลไทม์
ผู้ดำเนินโครงงาน	นายภัทรพงศ์ เพชรรัตน์ รหัส 48380242
ที่ปรึกษาโครงงาน	ดร.มุฑิตา สงฆ์จันทร์
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2555

.....

บทคัดย่อ

ปริญญานิพนธ์เล่มนี้ได้ศึกษาเกี่ยวกับการประมวลผลแบบเวลาจริง (Real-Time Signal - Processing) โดยสัญญาณที่ประมวลผลนำมาจากวงจรผลิตสัญญาณความถี่ (FET) แบบเวนบริดจ์ออสซิลเลเตอร์ (Wien-Bridge Oscillator) ซึ่งให้ผลตอบสนองทางเอาท์พุตออกมาเป็นสัญญาณรูปไซน์ที่มีความถี่หนึ่งกิโลเฮิร์ต ทำการศึกษาสัญญาณโดยเทียบกับวงจรกรองความถี่แบบต่างๆ เช่น วงจรกรองความถี่ต่ำ (low – pass filter: LPF) วงจรกรองความถี่สูง (high – pass filter: HPF) วงจรตัวกรองความถี่แถบผ่าน (band – pass filter: BPF) และวงจรกรองความถี่แถบหยุด (band – stop filter: BSF) เพื่อวิเคราะห์และศึกษาการทำงานของวงจรกรองความถี่ จากการวิเคราะห์สัญญาณที่ออกมาแล้วได้สัญญาณอะนาล็อก จึงได้พัฒนาการประมวลผลโดยการใช้ไมโครคอนโทรลเลอร์เป็นตัวเชื่อมต่อกับคอมพิวเตอร์ทำให้การประมวลผลสัญญาณแบบเวลาจริง (Real-Time Signal Processing) สามารถเก็บข้อมูล และเข้าใจได้ง่ายขึ้น นอกจากนี้ยังสามารถพัฒนาไปยังความถี่อื่นๆ ได้อีกด้วย

Project title	Frequency Filter Design and Application in Real-Time
Name	Mr.Pattarapong Phetcharat ID. 48380242
Project advisor	Miss Mutita Songjun, Ph.D.
Major	Electrical Engineering
Department	Electrical and Computer Engineering
Academic year	2012

Abstract

This project studies is about Real-time Signal Processing Wien-Bridge Oscillator generate the output sinusoidal at 1 KHz. The signal will input to the different type of filter as low-pass filter, high-pass filter, band-pass filter and band-stop filter. The output will also be shown in computer via Delphi program. To analyze and study the performance of the filter frequency from the output signal to be analog signal. So that development process using a microcontroller is connected to the computer. To make real-time signal processing, data storage and more easily understood. It is also possible to develop other frequencies as well.

กิตติกรรมประกาศ

โครงงานวิศวกรรมเรื่องการออกแบบโปรแกรมและวงจรกรองสัญญาณความถี่แบบเรียลไทม์สำเร็จได้ด้วยดี ผู้ดำเนินโครงงานขอกราบขอบพระคุณ ดร.มุฑิตา สงฆ์จันทร์ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงานผู้ให้คำแนะนำ ให้ความช่วยเหลือ และข้อคิดเห็นต่างๆ อันเป็นประโยชน์ในการทำโครงงานนี้ ขอขอบพระคุณอาจารย์ประจำภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ทุกๆท่านที่ประสิทธิ์ประสาทวิชาความรู้กับผู้ดำเนินโครงงาน และขอขอบคุณคณาจารย์คณะวิศวกรรมศาสตร์ทุกท่านที่ให้ความรู้ตลอดการเรียนที่ผ่านมา

ขอขอบคุณ นายวัฒนะ ค่านวิไล ผู้เสียสละ ตั้งแต่เริ่มจัดทำโครงงาน จนโครงงานนี้สำเร็จได้ด้วยดี ขอบคุณพี่น้องคณะวิศวกรรมศาสตร์ทุกท่านที่คอยให้ความช่วยเหลือ และให้กำลังใจที่ดีเสมอมา

ขอบคุณภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ และคณะวิศวกรรมศาสตร์ที่ให้ความเอื้อเฟื้ออุปกรณ์และสถานที่ในการทำโครงงาน

นายภัทรพงศ์ เพชรรัตน์

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์ ..... ก
บทคัดย่อภาษาไทย ..... ข
บทคัดย่อภาษาอังกฤษ ..... ค
กิตติกรรมประกาศ ..... ง
สารบัญ ..... จ
สารบัญตาราง ..... ช
สารบัญรูป ..... ซ

บทที่ 1 บทนำ
1.1 ที่มาและความสำคัญ 1
1.2 วัตถุประสงค์ 1
1.3 ขอบข่ายการศึกษา 1
1.4 ขั้นตอนการดำเนินงาน 2
1.5 ประโยชน์ที่ได้รับ 2
1.6 งบประมาณ 2

บทที่ 2 หลักการและทฤษฎี
2.1 พื้นฐานการออกแบบวงจรผลิตความถี่ 3
2.2 วงจรกรองความถี่ 7
2.3 ไมโครคอนโทรลเลอร์ ATMEGA8 19
2.4 รีเลย์ 25
2.5 สวิตชิ่งเพาเวอร์ซัพพลาย 27

บทที่ 3 วิธีการดำเนินงาน
3.1 ศึกษาและรวบรวมข้อมูลเกี่ยวกับโปรแกรมและตัวกรองความถี่ 30
3.2 การออกแบบวงจรกรองสัญญาณความถี่แบบเรียลไทม์ 30
3.3 แหล่งจ่ายไฟกระแสสลับให้กับวงจร 31
3.4 การสร้างวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์ 32

สารบัญ (ต่อ)

หน้า

3.5 การสร้างวงจรกรองความถี่ต่ำผ่าน 33
3.6 การสร้างวงจรกรองความถี่สูงผ่าน 34
3.7 การสร้างวงจรกรองความถี่แถบผ่าน 35
3.8 การสร้างวงจรกรองความถี่แถบหยุด 36
3.9 การประกอบวงจรกรองความถี่พร้อมใช้งานแบบเรียลไทม์ 38

บทที่ 4 ผลการทดลอง
4.1 หน้าต่างโปรแกรมเคลไฟ 39
4.2 สัญญาณที่เกิดจากวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์ 40
4.3 สัญญาณที่เกิดจากวงจรกรองความถี่ต่ำผ่าน 41
4.4 สัญญาณที่เกิดจากวงจรกรองความถี่สูงผ่าน 42
4.5 สัญญาณที่เกิดจากวงจรกรองความถี่แถบผ่าน 43
4.6 สัญญาณที่เกิดจากวงจรกรองความถี่แถบหยุด 44

บทที่ 5 สรุปและอภิปรายผลการทดลอง
5.1 สรุปผลการดำเนินงาน 46
5.2 ปัญหาและข้อเสนอแนะ 46
5.3 ประเมินผล 46
5.4 แนวทางในการพัฒนาโครงงานต่อไป 47

เอกสารอ้างอิง 48

ภาคผนวก ก 49
ภาคผนวก ข 61
ภาคผนวก ค 88

ประวัติผู้ดำเนินโครงงาน 106

สารบัญตาราง

ตารางที่ หน้า

1.1 ขั้นตอนการดำเนินงาน 2
2.1 หน้าที่การทำงานของขาสัญญาณทั้งหมดของ ATMEGA8 21

สารบัญรูป

รูปที่ หน้า

2.1 วงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์ 3
2.2 สัญญาณเอาท์พุตของวงจรแบบโดเมนทางเวลา (Time Domain) 5
2.3 สัญญาณเอาท์พุตของวงจรแบบโดเมนทางความถี่ (Frequency Domain) 5
2.4 การออกแบบวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์ 5
2.5 สัญญาณเอาท์พุทของวงจรแบบโดเมนทางเวลา 7
2.6 สัญญาณเอาท์พุตของวงจรแบบโดเมนทางความถี่ 7
2.7 วงจรกรองสัญญาณความถี่ต่ำและกราฟแสดงอัตราขยายแรงดันเชิงความถี่ 8
2.8 วงจรกรองสัญญาณความถี่สูงและกราฟแสดงผลการตอบสนองต่ออัตราขยายเชิงความถี่ 10
2.9 วงจรกรองสัญญาณช่วงความถี่และกราฟแสดงผลตอบสนองอัตราขยายเชิงความถี่ 12
2.10 กราฟแสดงการตอบสนองความถี่ของวงจรผสม 16
2.11 แสดงวงจรลดทอนสัญญาณช่วงความถี่และผลตอบสนองต่อความถี่ 17
2.12 การจัดขาของ ATMEGA8 20
2.13 รีเลย์และสัญลักษณ์ของรีเลย์ 25
2.14 สภาวะการทำงานของรีเลย์ 26
2.15 หน้าสัมผัสและการเรียกจำนวนหน้าสัมผัส 27
2.16 สวิตชิ่งเพาเวอร์ซัพพลาย 27
2.17 องค์ประกอบพื้นฐานของสวิตชิ่งเพาเวอร์ซัพพลาย 28
2.18 วงจรสวิตชิ่งเพาเวอร์ซัพพลาย 29
3.1 ขั้นตอนการทำงานของวงจรกรองสัญญาณความถี่แบบเรียลไทม์ 31
3.2 แบบวงจรแหล่งจ่ายไฟ 31
3.3 แหล่งจ่ายไฟกระแสสลับ 31
3.4 แบบวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์ 32
3.5 วงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์ 33
3.6 แบบวงจรตัวกรองความถี่ต่ำผ่าน 33
3.7 วงจรกรองความถี่ต่ำผ่าน 34
3.8 แบบวงจรกรองความถี่สูงผ่าน 34
3.9 วงจรกรองความถี่สูงผ่าน 35
3.10 แบบวงจรกรองความถี่แถบผ่าน 35
3.11 วงจรกรองความถี่แถบผ่าน 36

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.12 แบบวงจรกรองความถี่แถบหยุดเฉพาะช่วงแคบผ่าน 37
3.13 วงจรกรองความถี่แถบหยุดเฉพาะช่วงแคบผ่าน 37
3.14 วงจรกรองสัญญาณความถี่พร้อมใช้งานแบบเรียลไทม์ 38
4.1 หน้าต่างของโปรแกรมเดลไฟ 39
4.2 สัญญาณจากวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์ 41
4.3 สัญญาณที่เกิดจากวงจรกรองความถี่ต่ำผ่าน 41
4.4 กราฟแสดงการวิเคราะห์ฟังก์ชั่นโครงข่ายเทียบกับความถี่ของสัญญาณความถี่ที่มาจากวงจรกรองความถี่ต่ำผ่าน 42
4.5 สัญญาณที่เกิดจากวงจรกรองความถี่สูงผ่าน 42
4.6 กราฟแสดงการวิเคราะห์ฟังก์ชั่นโครงข่ายเทียบกับความถี่ของสัญญาณความถี่ที่มาจากวงจรกรองความถี่สูงผ่าน 43
4.7 สัญญาณที่เกิดจากวงจรกรองความถี่แถบผ่าน 43
4.8 กราฟแสดงการวิเคราะห์ฟังก์ชั่นโครงข่ายเทียบกับความถี่ของสัญญาณความถี่ที่มาจากวงจรกรองความถี่แถบผ่าน 44
4.9 สัญญาณวงจรกรองความถี่แถบหยุดเฉพาะช่วงแคบผ่าน 44
4.10 กราฟแสดงการวิเคราะห์ฟังก์ชั่นโครงข่ายเทียบกับความถี่ของสัญญาณความถี่ที่มาจากวงจรกรองความถี่แถบหยุดเฉพาะช่วงแคบผ่าน 45

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

ในระบบสื่อสาร วงจรกรองความถี่เป็นวงจรทางอิเล็กทรอนิกส์ที่มีความสำคัญอย่างมาก ด้วยคุณสมบัติในการทำงาน หรือผลตอบสนองในการทำงานทางด้านความถี่จึงได้มีการนำไปประยุกต์ใช้งานในด้านต่างๆ เช่นระบบอิเล็กทรอนิกส์ ระบบสื่อสาร ระบบควบคุม และเครื่องมือแพทย์ วงจรกรองความถี่แบ่งเป็นสองแบบคือ วงจรกรองความถี่แบบอนาลอก (Analog Filter) และวงจรกรองความถี่แบบดิจิตอล (Digital Filter) วงจรกรองความถี่ทำหน้าที่จำแนกความถี่ตามความต้องการ ซึ่งแบ่งตามคุณลักษณะของผลตอบสนองความถี่ (frequency response) ได้แก่ วงจรกรองความถี่ต่ำผ่าน (low-pass filter ; LPF) วงจรกรองความถี่สูงผ่าน (high-pass filter ; HPF) วงจรกรองแถบความถี่ผ่าน (band-pass filter ; BPF) และวงจรกรองแถบความถี่หยุดผ่าน (band-stop filter ; BSF) ซึ่งใช้สำหรับกำจัดสัญญาณที่ไม่ต้องการออก และแยกสัญญาณที่มีความซับซ้อนออกมาเป็นส่วนๆ เพื่อป้อนเข้าสู่วงจรต่างๆ ของระบบต่อไป ซึ่งโดยพื้นฐานแล้ววงจรกรองสัญญาณไฟฟ้านั้นจัดเป็นเน็ตเวอร์ก (Network) ทางไฟฟ้าโดยประกอบขึ้นจากอุปกรณ์พื้นฐานได้แก่ ตัวต้านทาน ตัวเก็บประจุ และตัวเหนี่ยวนำ เพื่อให้ได้คุณสมบัติตามที่ต้องการ

ดังนั้นการศึกษาเกี่ยวกับคุณลักษณะของตัวกรองความถี่โดยใช้วิธีการทดสอบจากวงจรผลิตความถี่ (FET) รวมถึงทฤษฎีการวิเคราะห์ตัวกรองไมโครคอนโทรลเลอร์จึงมีความสำคัญอย่างยิ่งโดยมุ่งเน้นไปที่ผลตอบสนองของตัวกรองความถี่ซึ่งสามารถประยุกต์กับการใช้งานจริงได้

1.2 วัตถุประสงค์

สร้างวงจรกรองความถี่แบบเรียลไทม์และแสดงผลออกทางหน้าจอคอมพิวเตอร์

1.3 ขอบข่ายการศึกษา

1. สร้างวงจรผลิตความถี่ขนาดหนึ่งกิโลเฮิร์ต
2. สร้างวงจรกรองความถี่แบบต่ำผ่าน สูงผ่าน แถบผ่าน และแถบหยุด
3. แสดงผลผ่านทางหน้าจอคอมพิวเตอร์โดยใช้โปรแกรมเคลไฟ

1.4 ขั้นตอนการดำเนินงาน

ตารางที่ 1.1 ขั้นตอนการดำเนินงาน

กิจกรรม	ปี 2555							
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	
1.ศึกษาและรวบรวมข้อมูลเกี่ยวกับโปรแกรมและตัวกรองความถี่								
2.ออกแบบวงจรที่ใช้ในการวิเคราะห์ตัวกรองความถี่แบบเรียลไทม์								
	ปี 2556							
	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.
3.สร้างตัวกรองความถี่แบบเรียลไทม์								
4.ทดสอบตัวกรองความถี่แบบเรียลไทม์								
5.รวบรวมผลการทดสอบและจัดรูปเล่ม								

1.5 ประโยชน์ที่คาดว่าจะได้รับ

เข้าใจเกี่ยวกับคุณสมบัติและคุณลักษณะของตัวกรองความถี่แบบเรียลไทม์ สามารถนำความรู้ทางคณิตศาสตร์มาประยุกต์ใช้กับการวิเคราะห์ตัวกรองความถี่แบบเรียลไทม์ หาโครงสร้างที่เหมาะสมสำหรับการใช้งาน และสามารถใช้โปรแกรมพร้อมการเชื่อมต่อระหว่างฮาร์ดแวร์และซอฟแวร์สำหรับวิเคราะห์ตัวกรองความถี่แบบเรียลไทม์

1.6 งบประมาณ

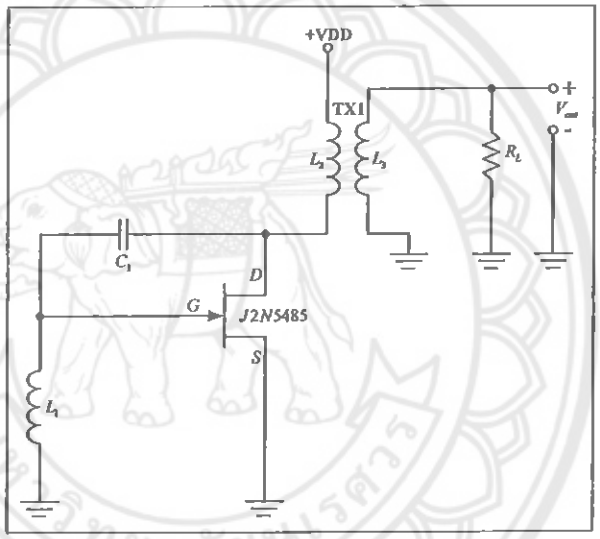
ค่าวัสดุอุปกรณ์อิเล็กทรอนิกส์	4,000 บาท
ค่าถ่ายเอกสารและเข้าเล่มปริญญานิพนธ์	1,000 บาท
รวมเป็นเงินทั้งสิ้น (ห้าพันบาทถ้วน)	5,000 บาท

หมายเหตุ: ถัวเฉลี่ยทุกรายการ

บทที่ 2
หลักการและทฤษฎี

2.1 พื้นฐานการออกแบบวงจรผลิตความถี่

พื้นฐานการออกแบบวงจรผลิตความถี่ด้วยอุปกรณ์ต่างๆ นำหลักการทำงานและคุณสมบัติของแต่ละอุปกรณ์มาใช้ในการสร้างวงจรผลิตความถี่ เพื่อทำให้เกิดความเข้าใจในการออกแบบวงจรผลิตความถี่มากขึ้น การออกแบบวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์ (Wien-Bridge Oscillator) ดังแสดงในรูปที่ 2.1 มีวิธีการออกแบบดังนี้



รูปที่ 2.1 วงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์

จากรูปที่ 2.1 วงจรผลิตความถี่จะใช้อุปกรณ์เฟตตัวเก็บประจุ ตัวเหนี่ยวนำ หม้อแปลง และตัวต้านทานมาสร้างเป็นวงจร ซึ่งวงจรผลิตความถี่แบบฮาร์ทเลย์ (Hartley) จะให้ผลตอบสนองทางเอาท์พุตออกมาเป็นสัญญาณรูปซายน์ (Sinusoidal) โดยสามารถเขียนสมการที่ใช้ในการออกแบบวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์ได้ดังสมการที่ 2.1 และสมการที่ 2.2

$$\omega = \frac{1}{\sqrt{C(L_1+L_2)}} \qquad (2.1)$$

เมื่อกำหนดให้ $\omega = 2\pi f$

$$R = R_L \left(\frac{n_2}{n_3}\right)^2 = \frac{L_2}{gmL_1} \qquad (2.2)$$

เมื่อกำหนดให้ $gm = \frac{1}{\omega^2 L_1 R C_1} - \frac{1}{R} = \frac{L_2}{R L_1}$

การออกแบบวงจรส่วนใหญ่จะกำหนดให้ $L_1 = L_2$ ดังนั้นสามารถเขียนสมการที่ใช้ในการคำนวณหาค่าของตัวเก็บประจุ (C_1) ได้ดังสมการที่ 2.3

$$C_1 = \frac{1}{\omega^2(L_1+L_2)} = \frac{1}{(2\pi f)^2(L_1+L_2)} \quad (2.3)$$

เมื่อพิจารณาค่าแรงดันจากวงจร $V_{DS} = V_{DD}$ และ $V_{GS} = 0$ โวลต์ ข้อมูลของอุปกรณ์เฟตเบอร์ 2N5485 ที่นำมาใช้ในการออกแบบวงจรนี้จะดูค่าของทรานสคอนดักแตนซ์ (Transconductance: gm) ว่ามีค่าเท่าไร เพื่อจะนำมาใช้ในการออกแบบวงจร ซึ่งมีค่าเท่ากับ 3500 ไมโครวินาทีถึง 7000 ไมโครวินาที โดยการออกแบบจะนำค่าต่ำสุดมาใช้คือ 3500 ไมโครวินาที ซึ่งทำให้สามารถคำนวณหาค่าของ R ได้ดังสมการที่ 2.4 สัญญาณเอาท์พุต (V_{out}) ที่ได้จากวงจรแสดงได้ดังรูปที่ 2.2 และรูปที่ 2.3 และการออกแบบวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์ ดังรูปที่ 2.4

$$R = \frac{L_2}{gmL_1} \quad (2.4)$$

ขั้นตอนการคำนวณหาค่า Turns Ratio สามารถคำนวณได้ดังสมการที่ 2.5

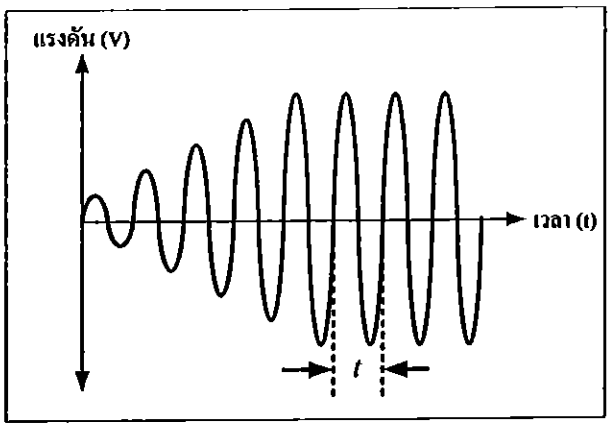
$$\frac{n_2}{n_1} = \sqrt{\frac{R}{R_L}} \quad (2.5)$$

ขั้นตอนการคำนวณหาค่า Coupling สามารถคำนวณได้ดังสมการที่ 2.6

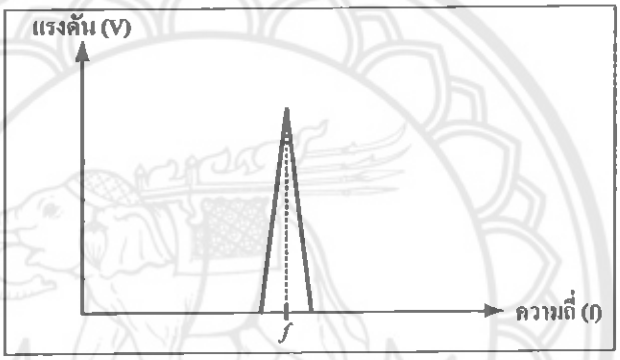
$$Coupling = \frac{n_3}{n_2} = \frac{1}{(n_2/n_1)} \quad (2.6)$$

ขั้นตอนการคำนวณหาค่า L_3 สามารถคำนวณได้ดังสมการที่ 2.7

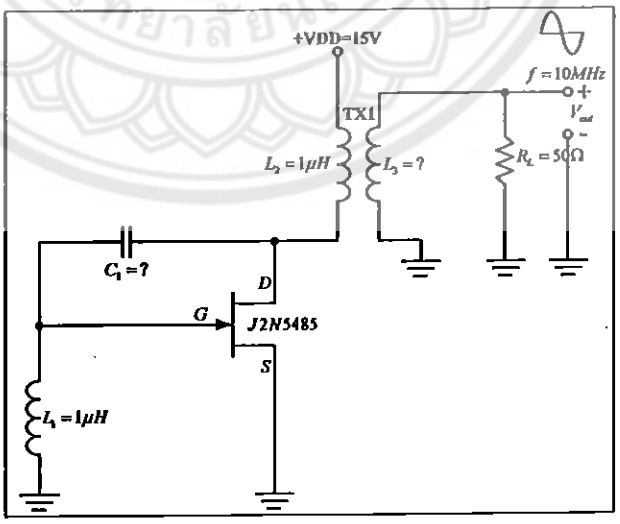
$$L_3 = L_2\left(\frac{n_3}{n_2}\right)^2 = L_2\left(\frac{1}{(n_2/n_1)}\right)^2 \quad (2.7)$$



รูปที่ 2.2 สัญญาณเอาท์พุตของวงจรแบบโดเมนทางเวลา (Time Domain)



รูปที่ 2.3 สัญญาณเอาท์พุตของวงจรแบบโดเมนทางความถี่ (Frequency Domain)



รูปที่ 2.4 การออกแบบวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์

จากรูปที่ 2.4 เป็นการออกแบบวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์ซึ่งค่าความถี่เท่ากับ 10 เมกกะเฮิร์ต โดยกำหนดให้ใช้เฟตเบอร์ J2N5485 ป้อนแรงดันให้กับวงจรมีค่าเท่ากับ +VDD = 15 โวลต์ กำหนดให้ใช้ตัวเหนี่ยวนำ (L_1 = L_2 = 1 ไมโครเฮิร์ต) และกำหนดให้ใช้ตัวต้านทาน (R_L) เท่ากับ 50 โอห์ม จากข้อมูลที่กำหนดสามารถคำนวณหาค่าของตัวเก็บประจุ (C_1) ได้ดังสมการที่ 2.3

$$C_1 = \frac{1}{\omega^2(L_1+L_2)} = \frac{1}{(2\pi f)^2(L_1+L_2)}$$

$$C_1 = \frac{1}{(2\pi\times10^8)^2}$$

$$C_1 = 126.7\ PF$$

คำนวณหาค่าของตัวต้านทาน R ได้ดังสมการที่ 2.4

$$R = \frac{L_2}{gmL_1} = \frac{1\mu H}{3500\mu s \times 1\mu H}$$

$$R = 285.7\Omega \approx 300 \text{ โอห์ม}$$

คำนวณหาค่าอัตราส่วนจำนวนรอบ (Turns Ratio) สามารถคำนวณได้ดังสมการที่ 2.5

$$\frac{n_2}{n_1} = \sqrt{\frac{R}{R_L}} = \sqrt{\frac{300\Omega}{50\Omega}} = 2.45$$

คำนวณหาค่าการเชื่อมต่อ (Coupling) สามารถคำนวณได้ดังสมการที่ 2.6

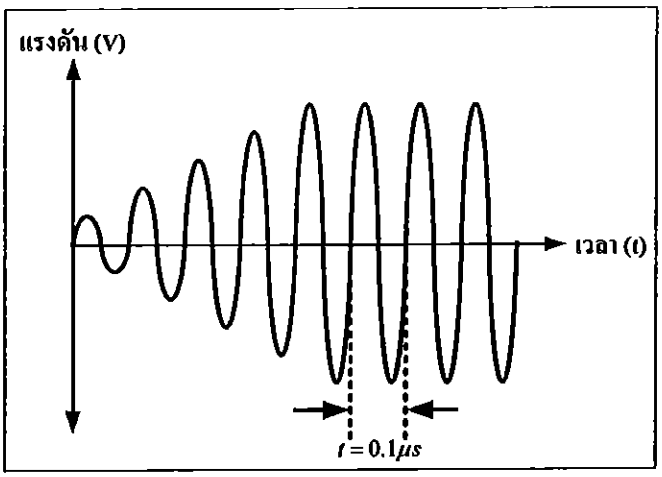
$$Coupling = \frac{n_3}{n_2} = \frac{1}{(n_2/n_1)} = \frac{1}{2.45} = 0.408$$

คำนวณหาค่าขดลวดเหนี่ยวนำที่ L_3 ซึ่งสามารถคำนวณได้ดังสมการที่ 2.7

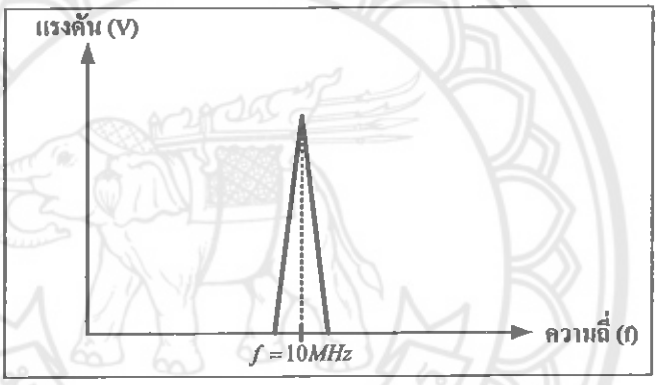
$$L_3 = L_2\left(\frac{n_3}{n_2}\right)^2 = L_2\left(\frac{1}{(n_2/n_1)}\right)^2$$

$$L_3 = 1\mu H\left(\frac{1}{(2.45)}\right)^2 = 0.1667\mu H$$

ค่าอุปกรณ์ต่างๆ สามารถนำมาสร้างเป็นวงจรผลิตความถี่ที่ต้องการได้ โดยผลตอบสนองของวงจรสามารถเขียนแสดงได้ดังรูปที่ 2.5 และรูปที่ 2.6



รูปที่ 2.5 สัญญาณเอาท์พุตของวงจรแบบโดเมนทางเวลา



รูปที่ 2.6 สัญญาณเอาท์พุตของวงจรแบบโดเมนทางความถี่

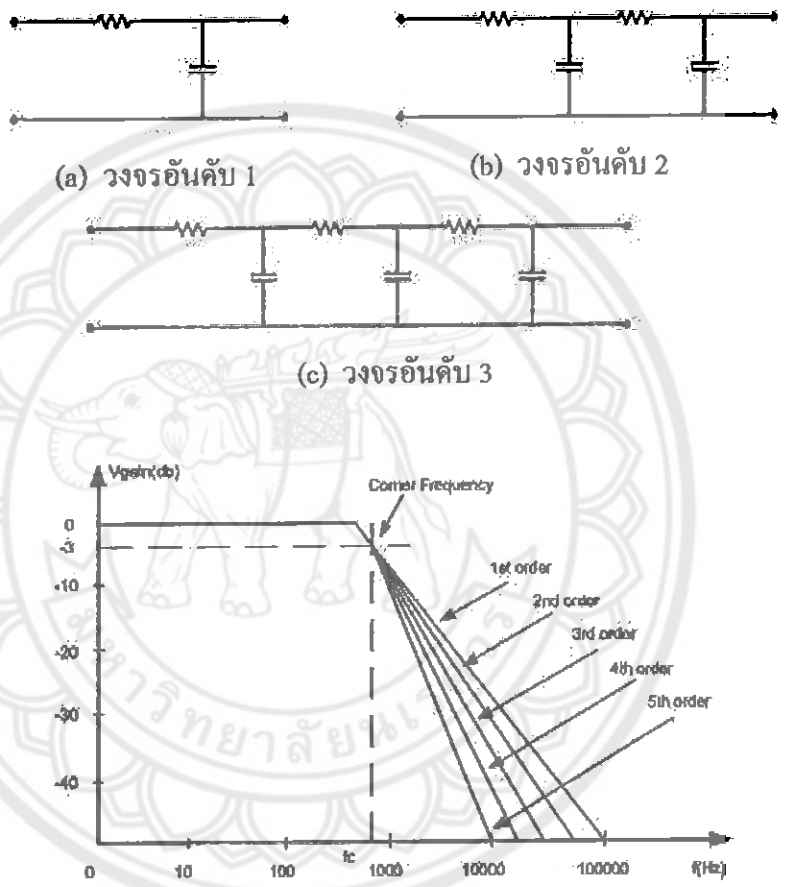
2.2 วงจรกรองความถี่ (Filter Circuit)

วงจรกรองความถี่หรือฟิลเตอร์ (Filter) คือวงจรไฟฟ้าที่ยอมให้สัญญาณไฟฟ้าที่ความถี่ใดๆ ความถี่หนึ่งหรือช่วงความถี่ใดความถี่หนึ่งเท่านั้นผ่านไปได้ ส่วนความถี่อื่นหรือช่วงความถี่อื่นๆ นอกเหนือจากที่กำหนดจะถูกลดทอนไปซึ่งจะเป็นช่วงความถี่ใดนั้นจะขึ้นอยู่กับการออกแบบวงจร ซึ่งมี 2 แบบได้แก่ แบบพาสซีฟคือวงจรที่ประกอบขึ้นด้วยอุปกรณ์แบบพาสซีฟ อุปกรณ์นี้สามารถทำงานได้โดยไม่ต้องมีการกระตุ้นด้วยไฟฟ้าเพื่อให้ทำงาน เช่นอุปกรณ์ประเภทตัวต้านทาน (R) ตัวเก็บประจุ (C) และขดลวดเหนี่ยวนำ (L) แบบที่สองคือแบบแอกทีฟ เป็นวงจรที่ประกอบขึ้นด้วยอุปกรณ์ที่ต้องการไฟฟ้า เพื่อกระตุ้นการทำงานของตัวอุปกรณ์ เช่นอุปกรณ์ประเภททรานซิสเตอร์หรือไอซี

2.2.1 วงจรกรองความถี่แบ่งได้ 4 ประเภทคือ

ก. วงจรกรองสัญญาณความถี่ต่ำ (Low Pass Filter: LPF)

วงจรกรองสัญญาณความถี่ต่ำ หมายถึงวงจรที่ยอมให้สัญญาณความถี่ตั้งแต่ 0 เฮิรตซ์ถึงความถี่ที่กำหนดผ่านไปได้ ส่วนความถี่ตั้งแต่ที่กำหนดถึงความถี่ที่สูงกว่าจะถูกลดทอนไปตามลำดับ ลักษณะของวงจรมีหลายอันดับและกราฟแสดงอัตราขยายแรงดันเชิงความถี่ (dB) ดังรูปที่ 2.7



รูปที่ 2.7 วงจรกรองสัญญาณความถี่ต่ำและกราฟแสดงอัตราขยายแรงดันเชิงความถี่

จากรูป 2.7 เมื่อป้อนความถี่ต่ำเข้าวงจรที่ตัวต้านทานทำให้มีค่าความต้านทานคาปาซิเตอร์สูง ซึ่งความถี่ต่ำไหลผ่านตัวต้านทานมีค่าความต้านทานน้อยกว่าค่าความต้านทานคาปาซิเตอร์ระดับสัญญาณขาออกจึงสามารถผ่านได้มาก แต่เมื่อความถี่สูงกว่าจุดที่กำหนด ค่าความต้านทานคาปาซิเตอร์ จะลดลง ทำให้ความถี่ผ่านขดลวดได้ลดลง และบางส่วนที่ผ่านไปได้ก็จะถูกตัวเก็บประจุดึงลงกราวด์ระดับสัญญาณขาออกจึงผ่านได้น้อยมาก ในทางทฤษฎีวงจรกรองความถี่ต่ำจะยอมให้สัญญาณใดๆ ที่มีความถี่ตั้งแต่ 0 เฮิรตซ์ จนถึงความถี่ที่กำหนดซึ่งเรียกว่าความถี่คัตออฟ หรือความถี่ขอบเขต (Cutoff Frequency) ผ่านไปได้โดยไม่มีการลดทอนของสัญญาณ และถ้าความถี่ของ

สัญญาณเข้ามีค่าเกินที่กำหนดไว้ สัญญาณออกควรจะมีค่าเป็นศูนย์ แต่ในทางปฏิบัติไม่สามารถจะทำเช่นนั้นได้เนื่องจากการตอบสนองสัญญาณที่ความถี่ต่างๆ ของอุปกรณ์ประเภทพาสซีฟจะเป็นแบบค่อยเป็นค่อยไปไม่เปลี่ยนแปลงทันทีทันใด ดังนั้นเมื่อสัญญาณมีความถี่สูงขึ้นวงจรจะลดสัญญาณลงไปเรื่อยๆ จนกระทั้งจะลดลงในอัตราคงที่ค่าหนึ่ง

ขั้นตอนการคำนวณวงจรกรองความถี่ต่ำค่าความถี่คัตออฟ (ω_c) ที่ต้องการใช้งานหาได้จากสมการ

$$\omega_c = \frac{1}{RC} = 2\pi f_c \tag{2.8}$$

โดยที่ ω_c มีหน่วยเป็นเรเดียนต่อวินาที (rad/s)

f_c มีหน่วยเป็น เฮิรตซ์ (Hz)

R มีหน่วยเป็น โอห์ม (Ω)

C มีหน่วยเป็น ฟารัด (F)

เมื่อกำหนด ω_c แล้วต้องการจะหาค่า R จึงได้เป็น

$$R = \frac{1}{\omega_c C} = \frac{1}{2\pi f_c} \tag{2.9}$$

ตัวอย่างที่ 1 กำหนดให้ ตัวต้านทานเท่ากับ 10 กิโลวัตต์และตัวเก็บประจุเท่ากับ 0.001 ไมโครฟารัด จงหาความถี่คัตออฟ

วิธีทำ

จากสมการ $\omega_c = \frac{1}{RC} = 2\pi f_c$ แทนค่า R และ C ลงในสมการ

จะได้ $\omega_C = \frac{1}{(10\times10^3)(0.001\times10^{-6})}$ = 100 กราดต่อวินาที

หรือ

$$f_c = \frac{\omega_C}{6.2\%} = \frac{100\times10^3}{6.2\%} = 15.9 \text{ กิโลเฮิร์ต}$$

ตัวอย่างที่ 2 จากวงจรกรองความถี่ต่ำผ่าน จงคำนวณหาค่าตัวต้านทาน เมื่อกำหนดให้ความถี่ คัตออฟ มีค่าเท่ากับ 2 กิโลเฮิรตซ์ และ ตัวเก็บประจุเท่ากับ 0.005 ไมโครฟารัด

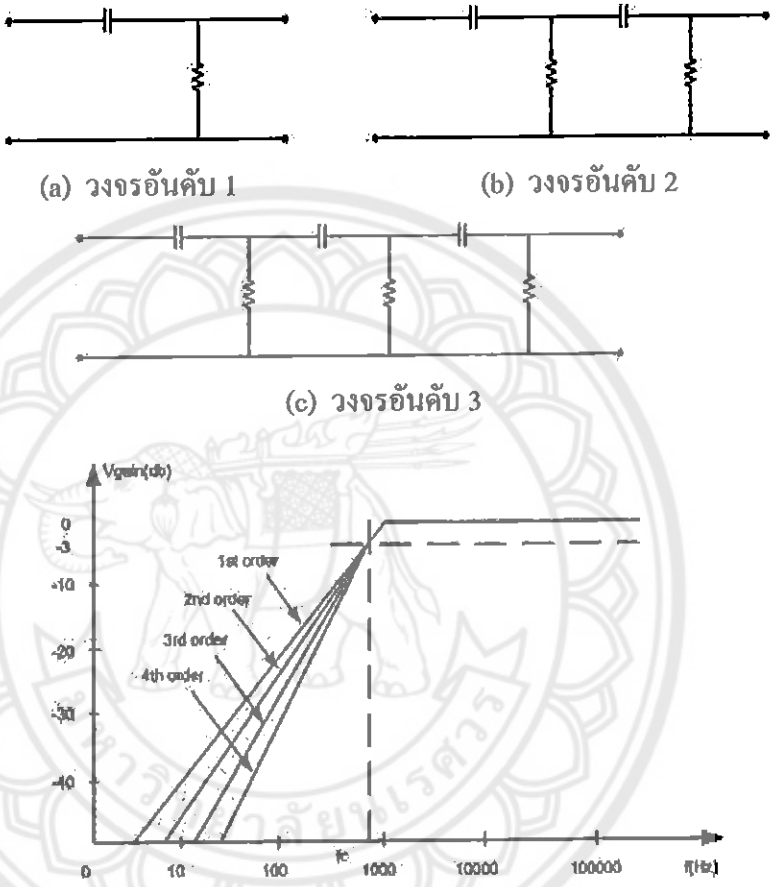
วิธีทำ

จากสมการที่ $R = \frac{1}{\omega_c C} = \frac{1}{2\pi f_c}$

จะได้ $R = \frac{1}{C\omega_c} = \frac{1}{6.2\%(2\times10^3)(5\times10^{-9})}$ = 15.9 กิโลวัตต์

ข. วงจรกรองสัญญาณความถี่สูง (High Pass Filter : HPF)

วงจรกรองสัญญาณความถี่สูงมีคุณสมบัติตรงข้ามกับวงจรกรองสัญญาณความถี่ต่ำ คือการตอบสนองต่อสัญญาณของวงจรจะยอมให้สัญญาณความถี่สูงผ่าน และลดทอนสัญญาณความถี่ต่ำ ซึ่งวงจรและกราฟตอบสนองอัตราขยายแรงดันความถี่แสดงในรูปที่ 2.8



รูปที่ 2.8 วงจรกรองสัญญาณความถี่สูงและกราฟแสดงผลการตอบสนองต่ออัตราขยายเชิงความถี่

จากรูปที่ 2.8 ตัวเก็บประจุต่อแบบอนุกรมกับวงจร และตัวต้านทานต่อแบบขนานกับวงจร เมื่อป้อนความถี่ต่ำกว่าเข้าไปในวงจร ตัวเก็บประจุจะมีค่าความต้านทานคาปาซิเตอร์สูงทำให้สัญญาณผ่านได้น้อย และทำให้สัญญาณที่ผ่านมาจากตัวเก็บประจุลงกราวด์ได้หมด แต่เมื่อความถี่สูงขึ้นตัวเก็บประจุจะมีค่าคาปาซิเตอร์ลดลงทำให้สัญญาณผ่านได้มากขึ้น และสัญญาณลงกราวด์น้อยลง เนื่องจากมีตัวต้านทานกั้นไว้ สัญญาณที่ออกไปยังสัญญาณขาออกจะมีมากขึ้นจนถึงระดับความแรงของสัญญาณประมาณ 70.7 เปอร์เซ็นต์ ของความแรงสูงสุด ซึ่งเรียกว่า ช่วงความถี่คัทออฟ เมื่อความถี่สูงกว่าความถี่นี้ตัวเก็บประจุจะยอมให้สัญญาณผ่านได้ และตัวต้านทานจะต้านสัญญาณไม่ให้ลงกราวด์ ความถี่จึงผ่านไปที่จุดสัญญาณขาออกได้ทั้งหมด

ขั้นตอนการคำนวณวงจรกรองความถี่สูงมีหลักการคำนวณเช่นเดียวกับวงจรกรองสัญญาณความถี่ต่ำ

ตัวอย่างที่ 3 จงคำนวณหาค่าความต้านทาน เมื่อกำหนดให้ตัวเก็บประจุมีค่าเท่ากับ 0.002 ไมโครฟารัด และความถี่คัตออฟมีค่าเท่ากับ 10 กิโลเฮิร์ต

วิธีทำ

จากสมการ $\omega_c = \frac{1}{RC} = 2\pi f_c$

$$R = \frac{1}{C\omega_c} = \frac{1}{6.2\%(10\times10^3)(0.002\times10^{-9})} = 8 \text{ กิโลเฮิร์ต}$$

ตัวอย่างที่ 4 ถ้ากำหนดให้ R = 22 kW และ C = 0.01 mF จงคำนวณหา ω_c และ f_c

วิธีทำ

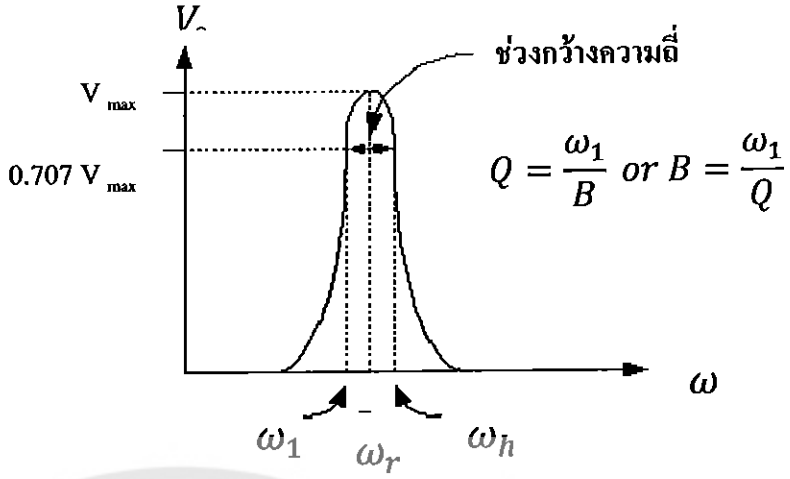
จากสมการ $\omega_c = \frac{1}{RC} = 2\pi f_c$

$$\omega_c = \frac{1}{(22\times10^3)\left(0.01\times10^{-6}\right)} = 4.45 \text{ กราดต่อวินาที}$$

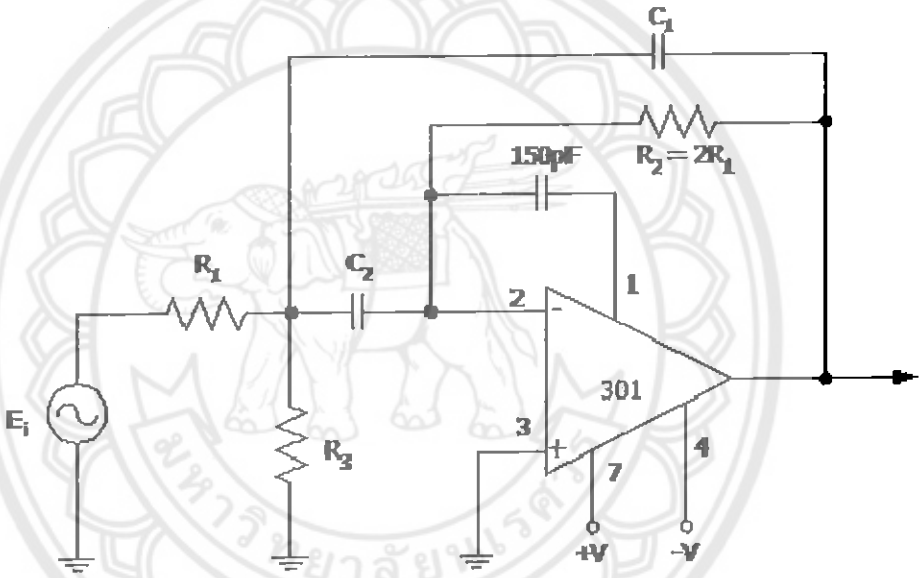
$$f_c = \frac{\omega_C}{2\pi} = \frac{4.54\times10^3}{6.2\%} = 724 \text{ กิโลเฮิร์ต}$$

ค. วงจรกรองสัญญาณช่วงความถี่ (Band Pass Filter: BPF)

เนื่องจากวงจรกรองสัญญาณช่วงความถี่มีลักษณะคล้ายกับการนำเอาวงจรกรองสัญญาณความถี่ต่ำ และความถี่สูงมาต่อร่วมกัน (Cascade) ดังนั้นวงจรกรองความถี่ผ่านเฉพาะช่วงจะยอมให้สัญญาณผ่านไปได้เฉพาะช่วงที่กำหนดเท่านั้น ส่วนความถี่ที่นอกเหนือจากที่กำหนดจะถูกลดทอนจนหมดไป วงจรกรองสัญญาณช่วงความถี่และกราฟแสดงผลตอบสนองอัตราขยายเชิงความถี่ (resonant frequency) แสดงในรูป 2.9



กราฟการตอบสนองความถี่ของวงจรกรองความถี่ผ่านเฉพาะช่วง



ลักษณะการต่อวงจรกรองความถี่ผ่านเฉพาะช่วง

รูปที่ 2.9 วงจรกรองสัญญาณช่วงความถี่และกราฟแสดงผลตอบสนองอัตราขยายเชิงความถี่

ค่าแรงดันเอาท์พุตสูงสุด V_{max} จะปรากฏขึ้นเฉพาะบางช่วงของความถี่เท่านั้น ซึ่งเรียกว่าความถี่เรโซแนนต์ (resonant frequency) แทนด้วย ω_r ตำแหน่งของความถี่ที่มีค่าแรงดันลดลงเป็น 0.707 V_{max} ซึ่งมีสองความถี่คือ ความถี่ที่สูงกว่า ω_r ค่าหนึ่ง และที่ต่ำกว่า ω_r อีกค่าหนึ่งเรียกความถี่ $V_0 = 0.707\ V_{max}$ ที่สูงกว่า ω_r นี้ว่าความถี่คัตออฟช่วงสูง (high cutoff frequency) แทนด้วย ω_r และความถี่ที่ต่ำกว่า ω_r ความถี่คัตออฟช่วงต่ำ (low cutoff frequency) แทนด้วย ω_1 จากความถี่ทั้งสองจะได้ช่วงกว้างความถี่ B เป็น

$$B = \omega_h - \omega_1 \tag{2.10}$$

วงจรกรองความถี่ผ่านเฉพาะช่วงแบ่งได้เป็น 2 ชนิดคือ ชนิดที่ยอมให้ความถี่ผ่านได้ในช่วงแคบ (narrow Band) และชนิดที่ยอมให้ผ่านได้ในช่วงกว้าง (wide band) ซึ่งถ้ามีช่วงกว้างความถี่น้อยกว่าหนึ่งในสิบของความถี่เรโซแนนต์ ($B < 0.1\ \omega_r$) เรียกว่าวงจรชนิดที่ยอมให้ผ่านได้ในช่วงแคบ ถ้ามีช่วงกว้างความถี่มากกว่าหนึ่งในสิบของความถี่เรโซแนนต์ ($B > 0.1\ \omega_r$) เรียกว่าวงจรชนิดที่ยอมให้ความถี่ผ่านได้ในช่วงกว้าง

วงจรแบบเลือกความถี่ผ่านนี้จะมีอัตราส่วนระหว่างความถี่เรโซแนนต์ และช่วงกว้างความถี่เรียกว่า ควอลิตี้แฟกเตอร์ (quality factor) แทนได้ด้วย Q ซึ่งเป็นตัวแสดงความสามารถในการเลือกความถี่ใช้งาน ยิ่งค่า Q มีค่าสูงมากขึ้นเท่าใด วงจรก็ยิ่งมีความสามารถในการเลือกความถี่ได้ดีมากขึ้นเท่านั้น ค่า Q แสดงได้ดังในสมการ

$$Q = \frac{\omega_r}{B} \tag{2.11}$$

หรือ

$$B = \frac{\omega_r}{Q}\ rad/s \tag{2.12}$$

B มีหน่วยเป็นเรเดียนต่อวินาที สำหรับวงจรในแบบที่ยอมให้ความถี่ผ่านได้ในช่วงแคบจะมีค่า Q สูงมากกว่า 10 และในวงจรผ่านแบบช่วงกว้างจะมีค่า Q น้อยกว่า 10

สรุป ช่วงความถี่แคบคือ มีค่า $B < 0.01$; $Q > 10$

ช่วงความถี่กว้างคือ มีค่า $B > 0.01$; $Q < 10$

วงจรกรองความถี่เฉพาะช่วงแคบผ่านในการออกแบบวงจรกำหนดค่าของช่วงกว้างความถี่ B ที่จะใช้งาน และคำนวณค่า Q ของวงจร โดยคำนวณ ได้จากสมการที่ (2.11) หรืออาจกำหนดค่า Q และคำนวณค่า B ก็ได้ หลังจากนั้นจึงจะเป็นการเลือกและคำนวณค่าของอุปกรณ์เพิ่มเติมอื่น ๆ ขั้นตอนในการพิจารณาดังต่อไปนี้ โดยจะกำหนดให้ $A_r = 1$ เลือก $C_1 = C_2 = C$ ให้มีเหมาะสมค่าที่ต่อจากนั้นก็คำนวณค่า R_1 R_2 และR_3 โดยได้จากสมการ

$$R_1 = \frac{R_2}{2} \tag{2.13}$$

$$R_2 = \frac{2}{BC} \tag{2.14}$$

$$R_3 = \frac{R_2}{(4Q^2-2)} = \frac{R_2}{4Q^2} \tag{2.15}$$

B ในสมการที่ (2.14) มีหน่วยเป็น เรเดียนต่อวินาที

ตัวอย่างที่ 5 จากการออกแบบวงจรกรองความถี่แบบผ่านเฉพาะช่วง ตามในรูปที่ 2.9 เมื่อกำหนดให้วงจรมี F_r = 1590 กิโลเฮิรตซ์ หรือ w_r=10 การ์ดต่อวินาที มี Q = 10 และ C_1=C_2=C=0.01 ไมโครฟารัด

วิธีทำ

หาค่า B จากสมการที่ (2.11)

$$B = \frac{10\times10^3}{10} = 1.0\,\frac{krad}{s} = 159 \text{ เฮิร์ต}$$

และหาค่าความต้านทานทั้งหมด โดยใช้สมการที่ (2.13) ถึงสมการที่ (2.15)

$$R_2 = \frac{2}{(2\times10^3)(0.001\times10^{-6})} = 200 \text{ กิโลวัตต์}$$

$$R_1 = \frac{200\,k}{2} = 100 \text{ กิโลวัตต์}$$

$$R_3 = \frac{200\times10^3}{4(100)} = 0.5 \text{ กิโลวัตต์}$$

ตัวอย่างที่ 6 จากในตัวอย่างที่ 1 ถ้ากำหนดให้วงจรมีช่วงกว้างความถี่ B เพิ่มขึ้นเป็น 2 กิโลวัตต์เรเดียน ต่อวินาที จงคำนวณหา

1. Q
2. R_2
3. R_1
4. R_3 (C_1 = C = 0.01 mF)

วิธีทำ

1. จากสมการที่ (2.11)

$$Q = \frac{10\times10^3}{2\times10^3} = 5 \text{ กิโลโอห์ม}$$

2. จากสมการที่ (2.13)

$$R_2 = \frac{2}{(2\times10^3)(0.01\times10^{-6})} = 100 \text{ กิโลโอห์ม}$$

3. จากสมการที่ (2.14)

$$Q = \frac{100k\Omega}{2} = 50 \text{ กิโลโอห์ม}$$

4. จากสมการที่ (2.15)

$$R_3 = \frac{100\times10^3}{4(15)} = 1 \text{ กิโลโอห์ม}$$

วงจรกรองความถี่ผ่านในช่วงกว้างสามารถใช้เป็นวงจรเดียวและวงกรองความถี่แบบผ่านในช่วงแคบได้ดังแสดงในรูปที่ 2.9 โดยเลือกให้ Q มีค่าน้อยกว่า 10 (Q<10) การคำนวณค่าของอุปกรณ์คล้ายกับวงจรกรองความถี่ผ่านในช่วงแคบ คือใช้สมการที่ (2.13) ถึงสมการที่ (2.15) ได้ (เฉพาะในกรณีที่ $4Q^2 > 2$ เท่านั้น)

ตัวอย่างที่ 7 จงออกแบบวงจรกรองความถี่แบบผ่านได้ในช่วงกว้าง ตามวงจรในรูปที่ 2.9 เมื่อกำหนดให้ f_c = 3,183 Hz หรือ ω_r = 20 krad/s, Q = 5 และ $C_1 = C_2 = 0.01$ mF

วิธีทำ

จากสมการที่ (2.11)

$$B = \frac{20\times10^3}{5} = 4 \text{ krad/s} = 636 \text{ เฮิร์ต}$$

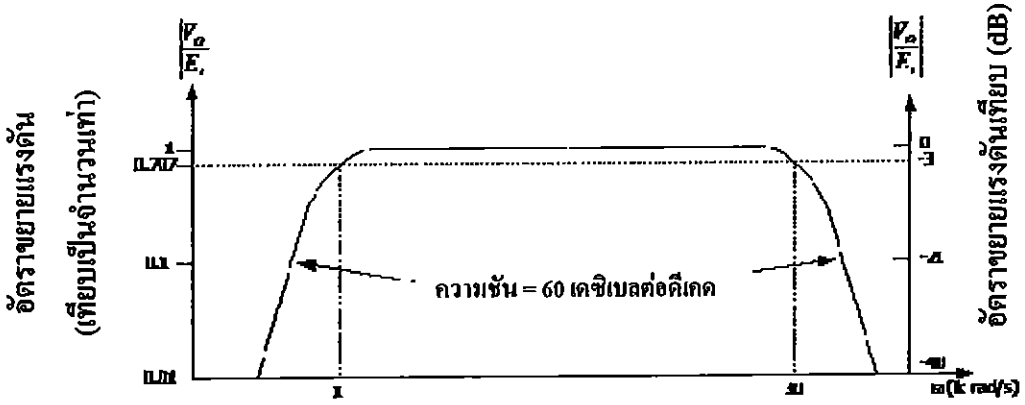
และจากสมการที่ (2.13) ถึงสมการที่ (2.15)

$$R_2 = \frac{2}{(4\times10^3)(0.01\times10^{-6})} = 50 \text{ กิโลวัตต์}$$

$$R_1 = \frac{\left(50k\Omega\right)}{2}$$

$$R_3 = \frac{50\times10^3}{4(25)} = 5 \text{ กิโลวัตต์}$$

วงจรชนิดนี้สามารถสร้างจากวงจรกรองความถี่ต่ำต่อร่วมกับวงจรกรองความถี่สูง เมื่อนำวงจรกรองความถี่ต่ำแบบบัตเตอร์เวิร์ทที่ให้ค่า –60 เดซิเบลต่อดีเคดมาต่อร่วมกับวงจรกรองความถี่สูงให้ผลตอบสนองทางความถี่ดังในรูปที่ 2.8 แม้ว่าวงจรใหม่ที่สร้างนี้จะต้องใช้ออป-แอมป์จำนวน 4 ตัว แต่มีข้อดีคือจะให้อัตราความชันเท่ากับ 60 เดซิเบลต่อดีเคดในดัตออฟช่วงความถี่ต่ำช่วงความถี่สูง และมีอัตราขยายเท่ากับ 1 ตลอดในความถี่ช่วงผ่าน

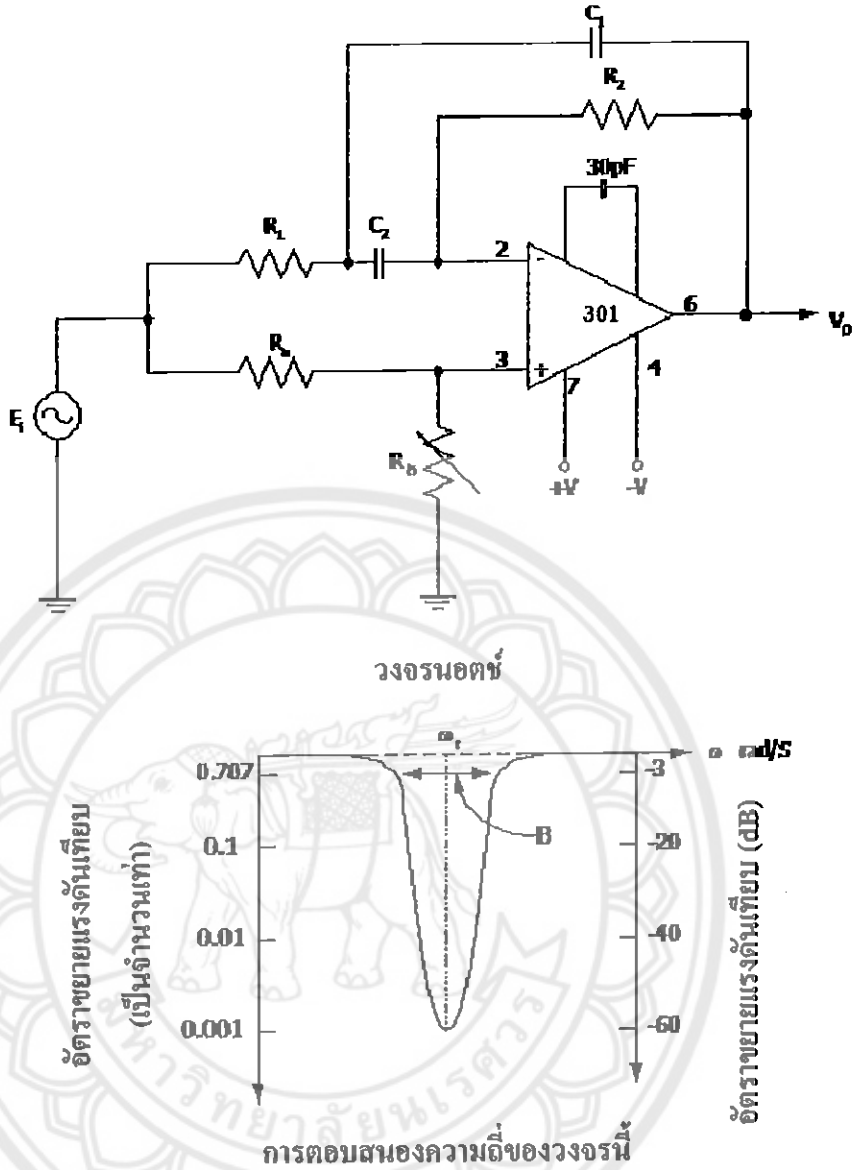


รูปที่ 2.10 กราฟแสดงการตอบสนองความถี่ของวงจรผสมที่เกิดจากการเอาวงจรกรองความถี่ต่ำแบบบัตเตอร์เวิร์ทที่ให้ค่า –60 เดซิเบลต่อดีเคด ต่อรวมกับวงจรวงจรกรองความถี่สูงผ่านแบบให้ค่า 60 เดซิเบลต่อดีเคด

การนำวงจรกรองความถี่ต่ำและความถี่สูงต่อรวมกัน เพื่อให้ได้วงจรกรองความถี่เฉพาะช่วง สามารถสลับตำแหน่งระหว่างวงจรทั้งสองได้ ซึ่งให้ผลการทำงานเหมือนกัน ทั้งนี้ต้องเลือกความถี่คัตออฟให้ถูกต้อง

ง. วงจรลดทอนสัญญาณช่วงความถี่ (Band Reject Filter: BRF)

วงจรลดทอนสัญญาณช่วงความถี่เป็นวงจรที่กำจัดความถี่บางช่วงออก เรียกว่า นอตช์ฟิลเตอร์ (notch filter) ซึ่งให้ผลตอบสนองทางความถี่ดังแสดงในรูปที่ 2.11 ซึ่งมีความถี่ช่วงผ่านสองช่วงคือทางซ้ายและขวา ส่วนความถี่ที่อยู่ตรงกลางถูกกำจัดไป เรียกว่าความถี่เรโซแนนต์ ω_r เช่นเดียวกับวงจรกรองความถี่ต่ำผ่านเฉพาะช่วง ได้แก่การกำจัดความถี่ 50 เฮิรตซ์หรือ 400 เฮิรตซ์ในการใช้งานมอเตอร์เจเนอเรเตอร์ เพื่อเป็นการลดสัญญาณรบกวนที่เกิดขึ้น การคำนวณและออกแบบวงจรทำเช่นเดียวกับวงจรกรองความถี่ผ่านเฉพาะช่วงคือ นำเอาช่วงกว้างความถี่ B และค่าควอลิตี้แฟก-เตอร์ Q มาคิดร่วมกับ ω_r ด้วย



รูปที่ 2.11 แสดงวงจรลดทอนสัญญาณช่วงความถี่และผลตอบสนองต่อความถี่

รายละเอียดการออกแบบวงจรมีดังนี้

1. กำหนดให้ $C_1 = C_2 = C$ โดยควรมีค่าอยู่ระหว่าง 100 พิโกฟารัดถึง 0.1 ไมโครฟารัด

2. คำนวณ R_2 จากสมการ

$$R_2 = \frac{2}{BC} \tag{2.16}$$

เมื่อ B มีหน่วยเป็นเรเดียนต่อวินาที

3. คำนวณ R_1 จากสมการ

$$R_1 = \frac{R_2}{4Q^2} \tag{2.17}$$

4. เลือก Ra ให้มีค่าเหมาะสม ควรมีค่าประมาณ 1 กิโลโอห์ม

5. คำนวณ Rb จากสมการ

$$R_b = 2Q^2 R_a \tag{2.18}$$

ตัวอย่างที่ 8 จงออกแบบวงจรนอตช์ฟิลเตอร์ดังในรูปที่ 2.11 ที่มี f_c = 400 เฮิร์ต และ Q=5 เมื่อกำหนดให้ $C_1 = C_2 = C = 0.01$ ไมโครฟารัด

วิธีทำ

$\omega_r = 2\pi f_c = (6.28)(400) = 2.51$ กราดต่อวินาที และจากสมการที่ (2.11)

จะได้ $B = \frac{25\times10^3}{5} = 500$ krad/s $= 79.6$ เฮิร์ต

จากสมการที่ (2.16)

$$R_2 = \frac{2}{(500)(0.01\times10^{-6})} = 400 \text{ กิโลโอห์ม}$$

จากสมการที่ (2.17)

$$R_1 = \frac{400k\Omega}{4(25)} = 4 \text{ กิโลโอห์ม}$$

เลือก R_a = 1 kW และสมการที่ (2.18) จะได้ R_b = 2(25)1 KW = 50 กิโลวัตต์

วงจรในรูปที่ 2.11 ถ้าต่อกราวด์ที่ขาอินพุทบวกของออป-แอมป์ สามารถเปลี่ยนวงจรให้เป็นวงจรกรองความถี่แบบผ่านเฉพาะช่วงคล้ายกับรูปที่ 2.10 ซึ่งแตกต่างตรงที่ไม่มี R_3 วงจรทั้งสองนี้มีค่าของอัตราขยายที่ความถี่เรโซแนนต์ ω_r เป็น $2Q^2$ การปรับค่าของ ω_r และ B ทำได้ง่าย เพียงปรับค่าความต้านทาน R_1 และ R_2 เท่านั้น

2.3 ไมโครคอนโทรลเลอร์ ATMEGA8

ATMEGA8 เป็นไมโครคอนโทรลเลอร์อนุกรม AVR ที่ผลิตโดย Atmel Corporation สหรัฐอเมริกา ที่เลือกใช้ในบอร์ดทดลอง Microcamp เป็นตัวถังแบบ DIP 28 ขา มีการจัดขาดังแสดงในรูปที่ 2.12 ซึ่งคุณสมบัติของ ATMEGA8 มีรายละเอียดดังต่อไปนี้

1. เป็นไมโครคอนโทรลเลอร์ 8 บิตในอนุกรม AVR มีสถาปัตยกรรมแบบแอดวานซ์อาร์ไอเอสซี (Advance RISC) มีความเร็วในการทำงานสูงโดยสามารถประมวลผล 1 คำสั่งในเวลา 1 สัญญาณนาฬิกา สามารถทำงานกับความถี่สัญญาณนาฬิกาสูงสุด 16 เมกกะเฮิร์ตจึงสามารถประมวลผลคำสั่งได้สูงถึง 16 ล้านคำสั่งต่อวินาที

2. มีหน่วยความจำ 3 แบบเพื่อรองรับการทำงานประกอบด้วย

- หน่วยความจำโปรแกรมแบบแฟรชความจุ 8 กิโลไบต์ซึ่งสามารถรักษาข้อมูลโปรแกรมไว้ได้แม้ไม่มีไฟเลี้ยง และสามารถป้องกันการอ่านได้ สามารถลบหรือเขียนข้อมูลใหม่ได้ 10,000 รอบ

- หน่วยความจำข้อมูลอีอีพรอมความจุ 512 ไบต์สามารถรักษาข้อมูลไว้ได้แม้ไม่มีไฟเลี้ยง สามารถลบหรือเขียนข้อมูลใหม่ได้ 10,000 รอบ

- หน่วยความจำข้อมูลแรมความจุ 1 กิโลไบต์ซึ่งใช้ในการประมวลผลหลักร่วมกับซีพียู

3. มีพอร์ตอินพุตเอาต์พุตอิสระที่สามารถโปรแกรมได้ 23 ขา แบ่งเป็น 3 กลุ่มคือ

- พอร์ต B ใช้งานได้สูงสุด 8 ขา (PB0 ถึง PB7) โดยมี 2 ขาเพื่อต่อกับคริสตอลเพื่อกำเนิดสัญญาณนาฬิกาให้แก่ไมโครคอนโทรลเลอร์

- พอร์ต C ขา (PC0 ถึง PC6) โดย PC0 ถึง PC5 สามารถใช้เป็นอินพุตอนาลอกได้

- พอร์ต D ขา (PD0 ถึง PD7)

4. มีตัวตั้งเวลา/ตัวนับ (Timer/Counter) ขนาด 8 บิต 2 ชุดที่แยกการทำงานอย่างอิสระ

5. มีตัวตั้งเวลา/ตัวนับ Timer/Counter ขนาด 16 บิต 1 ชุดรองรับการทำงานสมบูรณ์แบบทั้งในโหมดตั้งเวลา ตัวนับ เปรียบเทียบ (compare) และตรวจจับ (capture) สัญญาณ

6. มีโมดูลกำเนิดสัญญาณ PMW 3 ชุด

7. มีโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล (ADC) ความละเอียด 10 บิตจำนวน 6 ช่อง

8. มีอินพุตเปรียบเทียบสัญญาณอนาลอก 2 ช่อง

9. มีโมดูลสื่อสารอนุกรม 2 สาย รองรับการทำงานกับบัส I2C

10. มีโมดูลเชื่อมต่ออุปกรณ์อนุกรมหรือ SPI (Serial Peripheral Interface) ใช้สำหรับการโปรแกรมหน่วยความจำโปรแกรมแบบแฟรชในชิพโดยปกติแล้วมักสงวนไว้เพื่อการโปรแกรมแต่สามารถนำไปใช้เชื่อมต่ออุปกรณ์บนระบบบัส SPI ได้

11. มีโมดูลสื่อสารข้อมูลอนุกรม UART (Universal Asynchronous Receiver Transmitter)

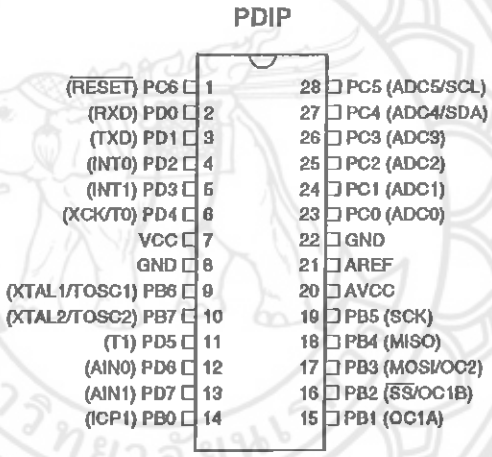
12. มีวอทช์ด็อกไทเมอร์เพื่อช่วยในการตรวจสอบการทำงานของระบบ

13. รองรับการอินเตอร์รัพต์ทั้งจากสัญญาณภายนอกและการทำงานของโมดูลต่างๆ ภายในไมโครคอนโทรลเลอร์มีระบบเพาเวอร์-ออนรีเซต

14. มีวงจรตรวจจับแรงดันไฟเลี้ยงผิดปกติหรือบราวเอาต์ที่สามารถโปรแกรมได้

15. ไฟเลี้ยง +4.5 ถึง +5.5V กระแส 3.6mA แต่ถ้าใช้รุ่น ATMEGA8L สามารถใช้ไฟเลี้ยงในย่านที่กว้างขึ้นจาก +2.7 ถึง +5.5 V แต่สัญญาณนาฬิกาต้องลดลงไม่เกิน 8 MHz

16. กำหนดการทำงานในโหมดหยุดทำงาน (sleep) ได้ 5 โหมดคือไอเดิล (Idle) โหมดลดสัญญาณรบกวนในวงจร ADC โหมดประหยัดพลังงาน (power-save) โหมดลดพลังงาน (power-down) และโหมดหยุดรอ (standby)



รูปที่ 2.12 การจัดขาของ ATMEGA8

หน้าที่การทำงานของขาสัญญาณทั้งหมดของ ATMEGA8

ตารางที่ 2.1 หน้าที่การทำงานของขาสัญญาณทั้งหมดของ ATMEGA8

ชื่อขา	ตำแหน่งขา	ชนิดของขา	รายละเอียดการทำงาน
Vcc	7	อินพุต	ขาต่อไฟเลี้ยงบวกตั้งแต่ 4.5 – 5.5 V
GND	8,22	อินพุต	ขาต่อกราวด์
AVcc	20	อินพุต	ขาต่อไฟเลี้ยง +5 V แก่โมดูล ADC ภายใน ATMEGA8
AREF	21	อินพุต	ขาต่อแรงดันอ้างอิงให้แก่โมดูล ADC ภายใน ATMEGA8
พอร์ต B เป็นพอร์ต 2 ทิศทาง			
PB0 ICP1	14	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตดิจิตอล PB0 - อินพุตโมดูลตรวจจับสัญญาณ ชุด1
PB1 OC1A	15	อินพุต/เอาต์พุต เอาต์พุต	- ขาพอร์ตดิจิตอล PB3 - เอาต์พุตวงจรเปรียบเทียบสัญญาณ PMW ช่อง 1A
PB2 OC1B SS	16	อินพุต/เอาต์พุต เอาต์พุต อินพุต	- ขาพอร์ตดิจิตอล PB3 - เอาต์พุตวงจรเปรียบเทียบสัญญาณ PMW ช่อง 1B - อินพุตเลือกอุปกรณ์สเลฟเมื่อทำงานเป็นอุปกรณ์สเลฟในระบบ SPI ใช้ในกระบวนการ SPI
PB3 OC2 MOSI	17	อินพุต/เอาต์พุต เอาต์พุต อินพุต/เอาต์พุต	- ขาพอร์ตดิจิตอล PB3 - เอาต์พุตวงจรเปรียบเทียบสัญญาณ PMW ช่อง 1B - อินพุตรับข้อมูลเมื่อทำงานเป็นอุปกรณ์สเลฟในระบบบัส SPI - เอาต์พุตส่งข้อมูลเมื่อทำงานเป็นอุปกรณ์มาสเตอร์ในระบบบัส SPI , ใช้ในกระบวนการ SPI

ตารางที่ 2.1 (ต่อ)

ขาพอร์ต B เป็นขาพอร์ต 2 ทิศทาง (ต่อ)			
ชื่อขา	ตำแหน่งขา	ชนิดของขา	รายละเอียดการทำงาน
PB4 MISO	18	อินพุต/เอาต์พุต อินพุต/เอาต์พุต	- ขาพอร์ตดิจิตอล PB4 - อินพุตรับข้อมูลเมื่อทำงานเป็นอุปกรณ์มาสเตอร์ในระบบบัส SPI , ใช้ในกระบวนการ SPI - เอาต์พุตส่งข้อมูลเมื่อทำงานเป็นอุปกรณ์สเลฟ
PB5 SCK	19	อินพุต/เอาต์พุต อินพุต/เอาต์พุต	- ขาพอร์ตดิจิตอล PB5 - อินพุตรับสัญญาณนาฬิกาเมื่อทำงานเป็นอุปกรณ์สเลฟในระบบบัส SPI - เอาต์พุตส่งสัญญาณนาฬิกาเมื่อทำงานเป็นอุปกรณ์มาสเตอร์ในระบบบัส SPI , ใช้ในกระบวนการ SPI
PB6 XTAL1 TOSC1	9	อินพุต/เอาต์พุต อินพุต อินพุต	- ขาพอร์ตดิจิตอล PB6 เมื่อเลือกทำงานกับสัญญาณนาฬิกาภายใน - อินพุตสัญญาณนาฬิกาภายนอก,ต่อกับคริสตอล - ไม่ใช้งานเมื่อเลือกทำงานกับสัญญาณนาฬิกาภายใน
PB7 XTAL2 TOSC2	10	อินพุต/เอาต์พุต อินพุต เอาต์พุต	- ขาพอร์ตดิจิตอล PB7 เมื่อเลือกทำงานกับสัญญาณนาฬิกาภายใน - ใช้ต่อกับคริสตอลหรือเซรามิกเรโซเนเตอร์ - เอาต์พุตสัญญาณนาฬิกาเมื่อใช้สัญญาณนาฬิกาภายใน
ขาพอร์ต C เป็นขาพอร์ต 2 ทิศทาง			
ชื่อขา	ตำแหน่งขา	ชนิดของขา	รายละเอียดการทำงาน
PC0 ADC0	23 23	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตดิจิตอล PC0 - อินพุตอนาลอกโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล ช่อง0
PC1 ADC1	24	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตดิจิตอล PC1 - อินพุตอนาลอกโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล ช่อง1

ตารางที่ 2.1 (ต่อ)

ชื่อขา	ตำแหน่งขา	ชนิดขา	รายละเอียดการทำงาน
พอร์ต C เป็นขาพอร์ต 2 ทิศทาง(ต่อ)			
PC2 ADC2	25	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตดิจิตอล PC2 - อินพุตอนาลอกโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล ช่อง2
PC3 ADC3	26	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตดิจิตอล PC3 - อินพุตอนาลอกโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล ช่อง3
PC4 ADC0 SDA	27	อินพุต/เอาต์พุต อินพุต เอาต์พุต	- ขาพอร์ตดิจิตอล PC4 - อินพุตอนาลอกโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล ช่อง 4 - ขาข้อมูลอนุกรมสำหรับระบบบัส 2 สาย
PC5 ADC5 SCL	28	อินพุต/เอาต์พุต อินพุต เอาต์พุต	- ขาพอร์ตดิจิตอล PC5 - อินพุตอนาลอกโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล ช่อง 5 - ขาเอาต์พุตสัญญาณนาฬิกาอนุกรมสำหรับระบบบัส 2 สาย
PC6 RESET	1	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตดิจิตอล PC6 - ขาอินพุตสัญญาณรีเซต
[illegible]			
PD0 RxD	23	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตดิจิตอล PD0 - อินพุตรับข้อมูลของโมดูลสื่อสารข้อมูลพอร์ตอนุกรมUSART
PD1 TxD	24	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตดิจิตอล PD1 - เอาต์พุตส่งข้อมูลของโมดูลสื่อสารข้อมูลพอร์ตอนุกรม USART

ตารางที่ 2.1 (ต่อ)

ขาพอร์ต D เป็นขาพอร์ต 2 ทิศทาง (ต่อ)			
ชื่อขา	ตำแหน่งขา	ชนิดของขา	รายละเอียดการทำงาน
PD2 INT0	25	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตดิจิตอล PD2 - อินพุตสัญญาณอินเตอร์รัพต์จากภายนอกช่อง 0
PD3 INT1	26	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตดิจิตอล PD3 - อินพุตสัญญาณอินเตอร์รัพต์จากภายนอกช่อง 1
PD4 XCK T0	27	อินพุต/เอาต์พุต อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตดิจิตอล PD4 - ขาสัญญาณภายนอกของโมดูลสื่อสารข้อมูลอนุกรม USART - อินพุตรับสัญญาณจากภายนอกสำหรับโมดูลไทเมอร์ 0
PD5 T1	11	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตดิจิตอล PD5 - อินพุตรับสัญญาณจากภายนอกสำหรับโมดูลไทเมอร์ 1
PD6 AIN0	12	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตดิจิตอล PD6 - อินพุตสำหรับวงจรเปรียบเทียบแรงดันอนาลอกช่อง 2
PD7 AIN1	13	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตดิจิตอล PD7 - อินพุตสำหรับวงจรเปรียบเทียบแรงดันอนาลอกช่อง 1

การทำงานภายในทั้งหมดของไมโครคอนโทรลเลอร์ ATMEGA8 หัวใจสำคัญคือ AVR ซีพียูที่ทรงประสิทธิภาพผนวกเข้ากับโมดูลเชื่อมต่ออุปกรณ์พิเศษที่มีอย่างสมบูรณ์ สำหรับไมโครคอนโทรลเลอร์ที่มีขาต่อใช้งาน 28 ขา ส่วนที่ต้องให้ความสนใจเป็นพิเศษคือขา PB6 และขา PB7 ซึ่งทำงานได้ทั้งขาของพอร์ตอินพุตและพอร์ตเอาต์พุตดิจิตอลและขาติดต่อวงจรกำเนิดสัญญาณนาฬิกาจากภายนอก หากเลือกใช้วงจรกำเนิดสัญญาณนาฬิกาภายในด้วยตัวไมโครคอนโทรลเลอร์จะสามารถกำหนดให้ขาพอร์ตทั้งสองนั้นเป็นพอร์ตอินพุต พอร์ตเอาท์พุตดิจิตอลได้ จึงมีพอร์ตเพิ่มขึ้นอีก 2 ขา แต่เพื่อให้การทำงานมีความเที่ยงตรงด้านจังหวะเวลาสูง โดยเฉพาะอย่างยิ่งหากมี

การเชื่อมต่อคอมพิวเตอร์เพื่อแลกเปลี่ยนข้อมูลระหว่างกันจึงมักจะเลือกใช้ขาทั้งสองนี้ในการต่อกับคริสตอลหรือเซรามิกโซเนเตอร์เพื่อรับสัญญาณนาฬิกามาจากภายนอกมากกว่า

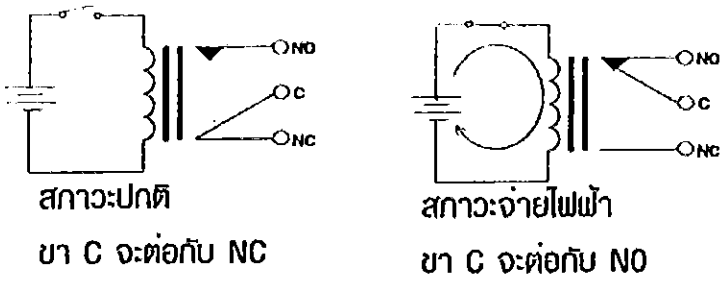
ATMEGA8 มีหน่วยความจำโปรแกรมแบบแฟรชที่สามารถโปรแกรมได้ด้วยกระบวนการโปรแกรมในวงจรหรือในระบบ (In-System Programming : ISP) ความจุ 8 กิโลไบต์ และสามารถอ่านข้อมูลในขณะที่เขียนได้ มีหน่วยความจำข้อมูลอีอีพรอมความจุ 512 ไบต์ และหน่วยความจำสแตติกแรม 1 กิโลไบต์ ซึ่งเป็นคุณสมบัติเด่นที่หายากในไมโครคอนโทรลเลอร์ในระดับเดียวกัน ส่งผลให้ ATMEGA8 สามารถรองรับโปรแกรมที่พัฒนาด้วยภาษา C ได้ดีพอสมควร นอกจากนี้ยังมีรีจีสเตอร์สำหรับใช้งานอีก 32 ตัว มีไทเมอร์/เคาน์เตอร์ที่ทำงานอิสระต่อกันอีก 3 ตัว สามารถกำเนิดสัญญาณ PWM 3 ช่อง ส่วนการจัดอินเตอร์รัพต์ทั้งจากภายในและภายนอก ส่วนติดต่ออุปกรณ์ระบบบัส 2 สาย มีโมดูลแปลงสัญญาณอนาล็อกเป็นดิจิตอล 6 ช่อง ความละเอียดในการแปลงสัญญาณ 10 บิต มีวอตช์ด็อกไทเมอร์ในตัว รวมทั้งวงจรตรวจจับไฟเลี้ยงผิดปกติหรือบราวเอาต์ (Brown – out detection) ที่สามารถกำหนดระดับแรงดันได้

2.4 รีเลย์ (Relay)

รีเลย์เป็นอุปกรณ์อิเล็กทรอนิกส์เชิงกล ชนิดหนึ่งซึ่งทำหน้าเป็นสวิตช์แต่รีเลย์นั้นจะถูกควบคุมด้วยกระแสไฟฟ้าดังแสดงในรูปที่ 2.13

รูปที่ 2.13 รีเลย์และสัญลักษณ์ของรีเลย์

การทำงานของรีเลย์ คือเมื่อมีกระแสไฟฟ้าไหลผ่านขดลวดจะทำให้ขดลวดเกิดสนามแม่เหล็กไปดึงแผ่นหน้าสัมผัสให้ดึงลงมาแตะหน้าสัมผัสอีกอันทำให้มีกระแสไหลผ่านหน้าสัมผัสไปได้ ดังแสดงในรูปที่ 2.14



รูปที่ 2.14 สภาวะการทำงานของรีเลย์

ขาของรีเลย์มีรายละเอียดดังต่อไปนี้

ขาจ่ายแรงดันใช้งาน ซึ่งจะมีอยู่ 2 ขา จากรูปที่ 2.14 จะเห็นสัญลักษณ์ขดลวดแสดงตำแหน่งขา coil หรือ ขาต่อแรงดันใช้งาน

ขาคอมมอน (C หรือ COM) จะเป็นขาต่อระหว่าง NO และ NC

ขาปกติเปิด (NO หรือ Normally opened) โดยปกติขานี้จะเปิดไว้และจะทำงานเมื่อป้อนแรงดันให้รีเลย์

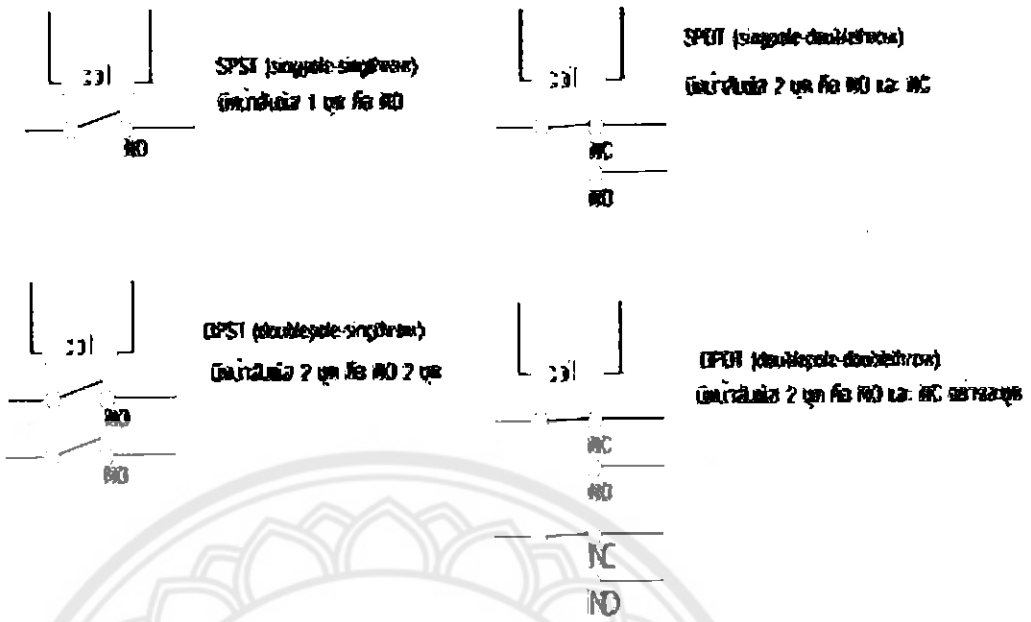
ขาปกติปิด (NC หรือ Normally closed) โดยปกติขานี้จะต่อกับขา C ในกรณีที่เราไม่ได้จ่ายแรงดันหน้าสัมผัสของ C และ NC จะต่อถึงกัน

ข้อคำนึงในการใช้งานรีเลย์

1. แรงดันใช้งานดูที่ตัวรีเลย์จะระบุค่า แรงดันใช้งานไว้ เช่น 12 VDC คือต้องใช้แรงดันที่ 12 VDC เท่านั้นหากใช้มากกว่านี้ขดลวดภายในตัวรีเลย์อาจจะขาดได้หรือหากใช้แรงดันต่ำกว่ามากจะทำให้รีเลย์ไม่ทำงาน ส่วนในการต่อวงจรนั้นสามารถต่อขั้วใดก็ได้เพราะตัวรีเลย์ไม่ระบุขั้วต่อไว้นอกจากชนิดพิเศษ

2. การใช้งานกระแสผ่านหน้าสัมผัส ซึ่งที่ตัวรีเลย์จะระบุไว้ เช่น 10A 220 AC คือหน้าสัมผัสของรีเลย์นั้นสามารถทนกระแสได้ 10 แอมแปร์ที่ 220 VAC แต่ควรใช้งานที่ระดับกระแสต่ำกว่านี้เพราะถ้ากระแสมากหน้าสัมผัสของรีเลย์จะเสียหายได้

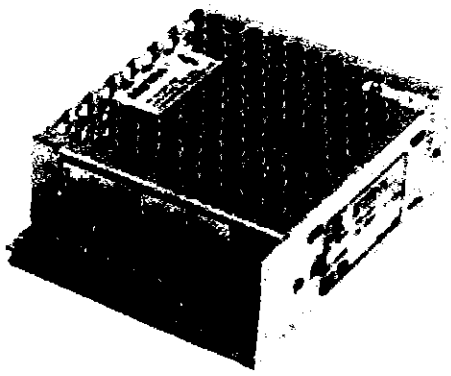
3. จำนวนหน้าสัมผัสการใช้งานควรดูว่ารีเลย์นั้นมีหน้าสัมผัสให้ใช้งานกี่อันและมีขั้วคอมมอนด้วยหรือไม่ รีเลย์จะมีหน้าสัมผัสและเรียกจำนวนหน้าสัมผัสดังรูปที่ 2.15



รูปที่ 2.15 หน้าสัมผัสและการเรียกจำนวนหน้าสัมผัส

2.5 สวิตชิ่งเพาเวอร์ซัพพลาย (Switching Power Supply)

สวิตชิ่งเพาเวอร์ซัพพลาย ดังแสดงในรูปที่ 2.16 เป็นแหล่งจ่ายไฟตรงคงค่าแรงดันแบบหนึ่ง และสามารถเปลี่ยนแรงดันจากไฟสลับโวลต์สูงให้เป็นแรงดันไฟ ตรงค่าต่ำ เพื่อใช้ในงานอิเลคทรอนิกส์ได้เช่นเดียวกับแหล่งจ่ายไฟเชิงเส้น (Linear Power Supply) ถึงแม้เพาเวอร์ซัพพลายทั้งสองแบบจะต้องมีการใช้หม้อแปลงในการลดทอนแรงดันสูงให้เป็นแรงดันต่ำเช่นเดียวกัน แต่สวิตชิ่งเพาเวอร์ซัพพลายจะต้องการใช้หม้อแปลงที่มีขนาดเล็ก และน้ำหนักน้อย เมื่อเทียบกับแหล่งจ่ายไฟเชิงเส้น อีกทั้งสวิตชิ่งเพาเวอร์ซัพพลายยังมีประสิทธิภาพสูงกว่า

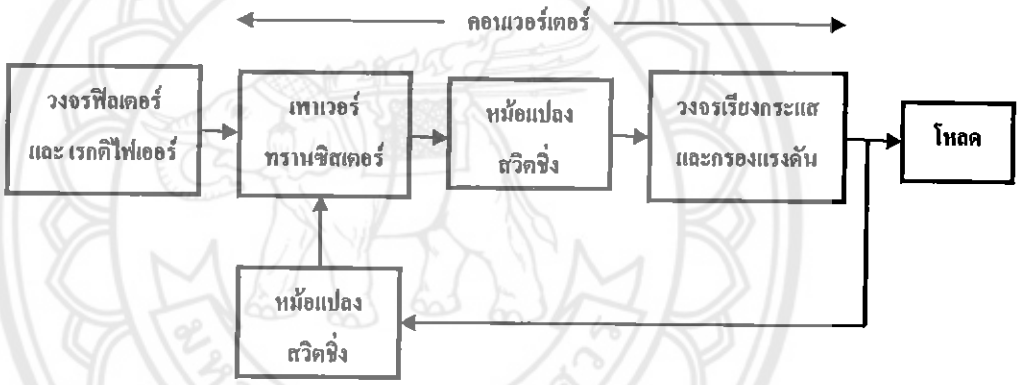


รูปที่ 2.16 สวิตชิ่งเพาเวอร์ซัพพลาย

สวิตชิ่งเพาเวอร์ซัพพลายจะมีเสถียรภาพในการทำงานที่ต่ำกว่า และก่อให้เกิดสัญญาณรบกวนได้สูงเมื่อเปรียบเทียบกับแหล่งจ่ายไฟเชิงเส้น รวมทั้งสวิตชิ่งเพาเวอร์ซัพพลายยังมีความซับซ้อนของวงตรมากกว่าและมีราคาสูง ที่กำลังงานต่ำๆ แหล่งจ่ายไฟเชิงเส้นจะประหยัดกว่าและให้ผลดีเท่าเทียมกัน ดังนั้นสวิตชิ่งเพาเวอร์ซัพพลายจึงนิยมใช้กันในงานที่ต้องการกำลังงานตั้งแต่ 20 วัตต์ขึ้นไปเท่านั้น

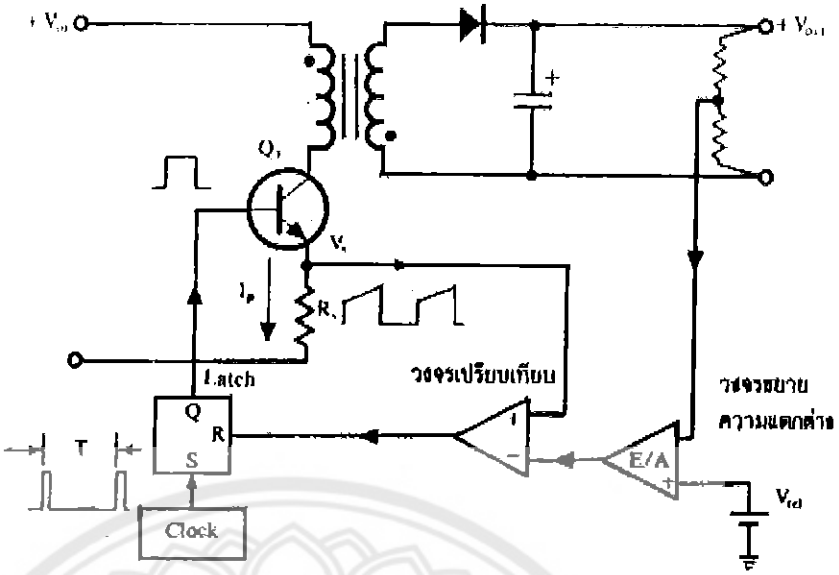
หลักการทำงานของสวิตชิ่งเพาเวอร์ซัพพลาย

สวิตชิ่งเพาเวอร์ซัพพลายโดยทั่วไปมีองค์ประกอบพื้นฐานที่คล้ายคลึงกัน และไม่ซับซ้อนมากนัก ดังแสดงในรูปที่ 2.17 หลักสำคัญของสวิตชิ่งเพาเวอร์ซัพพลายจะอยู่ที่คอนเวอร์เตอร์ เนื่องจากทำหน้าที่ทั้งลดทอนแรงดันและคงค่าแรงดันเอาต์พุตด้วย



รูปที่ 2.17 องค์ประกอบพื้นฐานของสวิตชิ่งเพาเวอร์ซัพพลาย

แรงดันไฟสลับค่าสูงจะผ่านเข้ามาทางวงจรฟิลเตอร์ เพื่อกรองสัญญาณรบกวนและแปลงเป็นไฟตรงค่าสูงด้วยวงจรเรกติไฟเออร์ เพาเวอร์ทรานซิสเตอร์จะทำงานเป็นเพาเวอร์คอนเวอร์เตอร์ โดยการตัดต่อแรงดันเป็นช่วงๆ ที่ความถี่ประมาณ 20-200 กิโลเฮิร์ตจากนั้นจะผ่านไปยังหม้อแปลงสวิตชิ่งเพื่อลดแรงดันลง เอาต์พุตของหม้อแปลงจะต่อกับวงจรเรียงกระแส และกรองแรงดันให้เรียบ การคงค่าแรงดันจะทำได้โดยการป้อนกลับค่าแรงดันที่เอาต์พุตกลับมายังวงจรควบคุม เพื่อควบคุมให้เพาเวอร์ทรานซิสเตอร์นำกระแสมากขึ้นหรือน้อยลงตามการเปลี่ยนแปลงของแรงดันที่เอาต์พุต ซึ่งจะมีผลทำให้แรงดันเอาต์พุตคงที่ ดังแสดงในรูปที่ 2.18



รูปที่ 2.18 วงจรสวิตชิ่งเพาเวอร์ซัพพลาย

บทที่ 3
วิธีการดำเนินงาน

3.1 ศึกษาและรวบรวมข้อมูลเกี่ยวกับโปรแกรมและตัวกรองความถี่

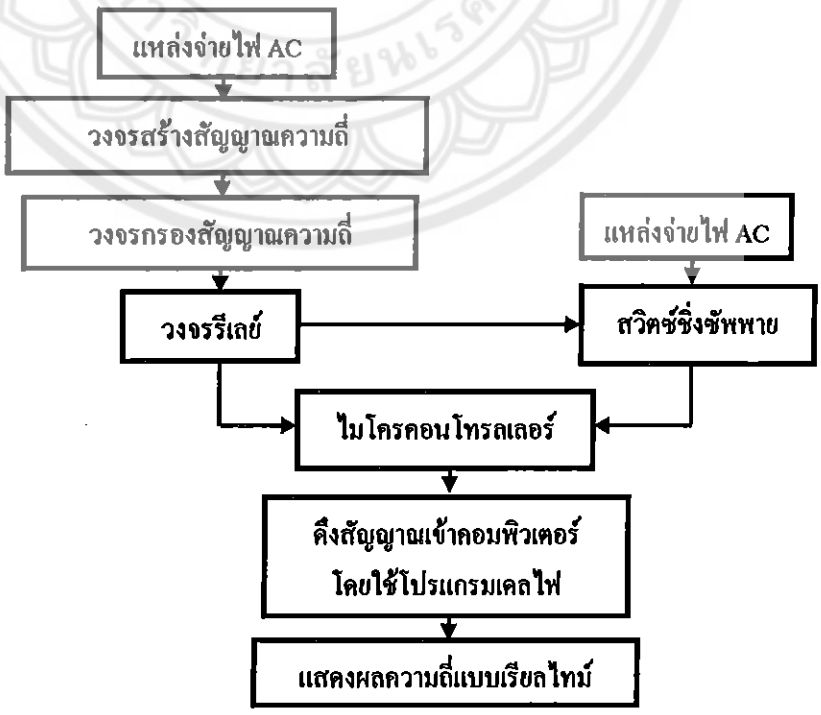
3.1.1 โปรแกรมที่ใช้ในการดำเนินงานได้แก่ โปรแกรมเดลไฟ (Delphi)โดยใช้ในการดึงข้อมูลจากไมโครคอนโทนเลอร์แสดงผลที่หน้าจอคอมพิวเตอร์ในรูปแบบเรียลไทม์ ซึ่งได้สัญญาณความถี่รูปซายน์ ซึ่งรายละเอียดของโปรแกรมแสดงอยู่ในภาคผนวก

3.1.2 ตัวกรองความถี่ที่ใช้ในการดำเนินงานมี 4 ชนิดได้แก่

1. ตัวกรองความถี่ต่ำผ่าน (low – pass filter : LPF)
2. ตัวกรองความถี่สูงผ่าน (high – pass filter : HPF)
3. ตัวกรองความถี่แถบผ่าน (band – pass filter : BPF)
4. ตัวกรองความถี่แถบหยุด (band – stop filter : BSF)

3.2 การออกแบบวงจรกรองสัญญาณความถี่แบบเรียลไทม์

ขั้นตอนการทำงานของวงจรกรองสัญญาณความถี่แบบเรียลไทม์สามารถแสดงได้ในรูปที่ 3.1

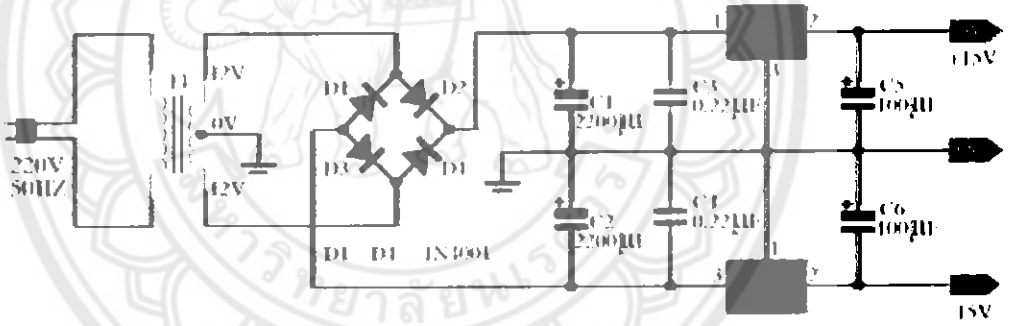


รูปที่ 3.1 ขั้นตอนการทำงานของวงจรกรองสัญญาณความถี่แบบเรียลไทม์

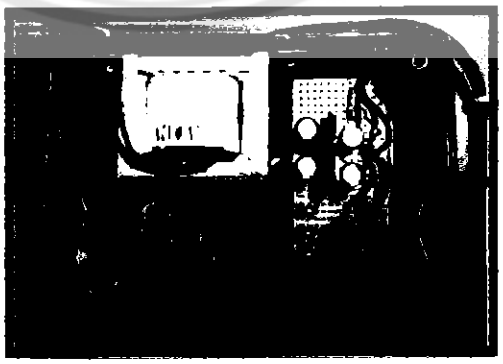
จากรูปที่ 3.1 ป้อนแหล่งจ่ายไฟ 15 โวลต์ไปยังวงจรสร้างสัญญาณความถี่โดยวงจรจะสร้างสัญญาณรูปซายน์มีค่าเท่ากับหนึ่งกิโลเฮิร์ต ป้อนไปยังวงจรกรองความถี่โดยจะมีรีเลย์และสวิตช์ซึ่งซัพพายเป็นวงจรควบคุมดึงสัญญาณเอาท์พุตและอินพุตเพื่อมาป้อนให้กับไมโครคอนโทรลเลอร์ หลังจากนั้นใช้โปรแกรมเคลไฟดึงสัญญาณเข้าคอมพิวเตอร์เพื่อแสดงผลความถี่ให้ออกมาอยู่ในรูปแบบเรียลไทม์

3.3 แหล่งจ่ายไฟกระแสสลับให้กับวงจร

แหล่งจ่ายไฟมีวงจรดังแสดงในรูปที่ 3.2 ซึ่งใช้หม้อแปลงเป็นแทบกลางโดยมีเอาท์พุตประมาณ +12โวลต์ –12โวลต์ จากนั้นส่งต่อมายังวงจรไดโอดแบบบริดจ์ทำการเรียงกระแสเต็มคลื่นส่งมายังตัวกรอง C1 C2 C3 C4 แล้วส่งไปยัง IC 7815 และ IC 7915 สุดท้ายใช้ C5 และ C6 กรองกระแสอีกครั้งก่อนนำมาต่อใช้งาน แล้วสร้างวงจรแหล่งจ่ายไฟดังแสดงในรูปที่ 3.3 มีกำลัง 15 โวลต์ เพื่อป้อนไปให้กับวงจรกรองความถี่



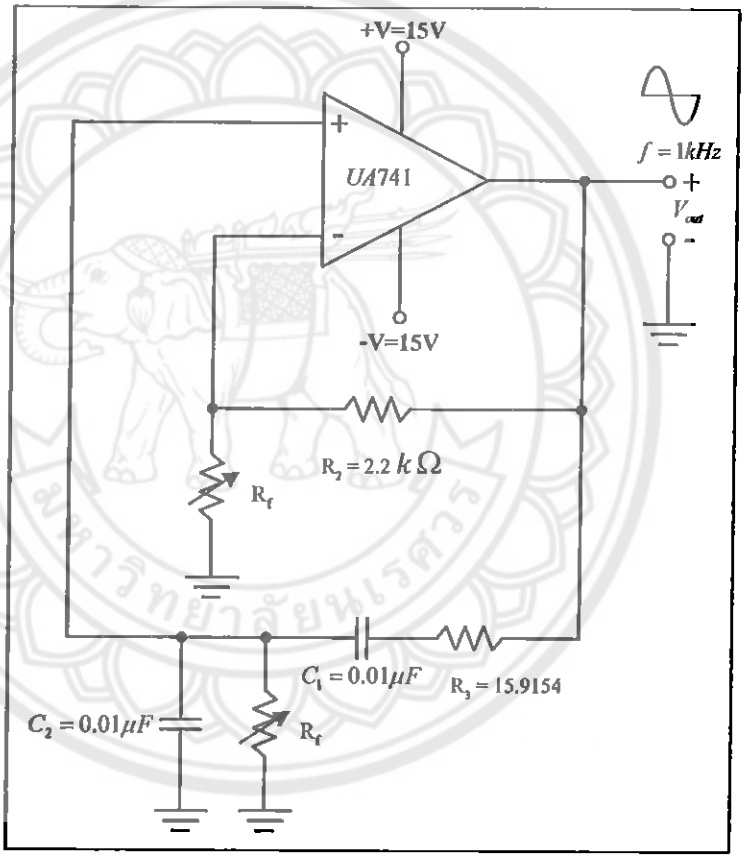
รูปที่ 3.2 แบบวงจรแหล่งจ่ายไฟ



รูปที่ 3.3 แหล่งจ่ายไฟกระแสสลับ

3.4 การสร้างวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์

วงจรผลิตความถี่ หรือ "วงจรเวนออสซิลเลเตอร์" เป็นวงจรไฟฟ้าที่ทำหน้าที่ผลิตสัญญาณไฟฟ้าที่มีขนาดเปลี่ยนแปลงตามเวลา ซึ่งลักษณะของสัญญาณไฟฟ้าที่สร้างขึ้นมาเรียกว่าสัญญาณรูปคลื่น ตัวอย่างเช่น สัญญาณรูปซายน์ สัญญาณรูปสี่เหลี่ยม สัญญาณรูปสามเหลี่ยม และสัญญาณรูปฟันเลื่อย ซึ่งการออกแบบวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์ออกแบบได้จากรูปที่ 3.4 วงจรผลิตความถี่แบบนี้จะใช้ ออป-แอมป์ ตัวเก็บประจุ และตัวต้านทานนำมาสร้างเป็นวงจรในรูปที่ 3.5 ซึ่งจะให้ผลตอบสนองทางเอาท์พุตออกมาเป็นสัญญาณรูปซายน์



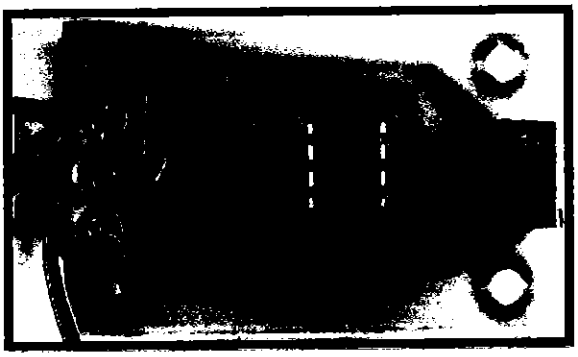
รูปที่ 3.4 แบบวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์

การคำนวณหาค่าตัวแปรในการออกแบบวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์เมื่อกำหนดความถี่เท่ากับหนึ่งกิโลเฮิร์ตคำนวณค่าตัวแปรจากสมการดังต่อไปนี้

$$R_3 = \frac{1}{2\pi f C_1} = \frac{1}{2\pi \times 1kHz \times 0.01\mu F} = \text{ 15.9154 กิโลโอห์ม}$$

$$A_V = 1 + \frac{R_2}{R_1} = 1 + \frac{2.2k\Omega}{1k\Omega} = 3.2$$

$$f = \frac{1}{2\pi R_3 C_1} = \frac{1}{6.28 \times 15.9154k\Omega \times 0.01\mu F} = \text{ 1 กิโลเฮิร์ต}$$



รูปที่ 3.5 วงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์

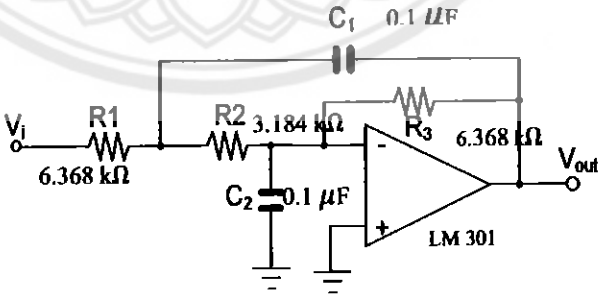
3.5 การสร้างวงจรกรองความถี่ต่ำผ่าน (low – pass filter : LPF)

การคำนวณหาค่าตัวแปรในการออกแบบวงจรกรองความถี่ต่ำผ่านได้โดยกำหนดสัญญาณที่เข้ามาเท่ากับ 1 กิโลเฮิร์ต และ กำหนดความถี่คัตออฟไว้ที่ 500 เฮิร์ตและเลือกใช้ตัวเก็บประจุ 0.1 ไมโครฟารัด คำนวณค่าความต้านทาน R จากสมการ

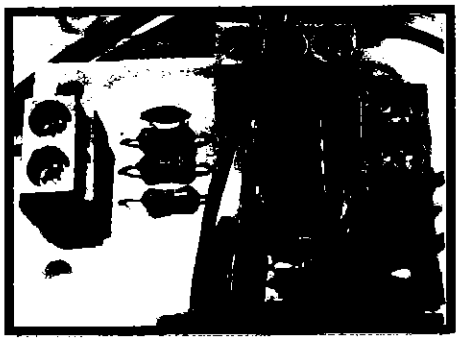
$$R_2 = R_3 = \frac{1}{2\pi f_c C} = \frac{1}{(6.28)\times(500)\times(0.01\times10^{-6})} = 3.184 \text{ กิโลโอห์ม}$$

$$R_3 = R_1 = 3.184 \times 2 = 6.368 \text{ กิโลโอห์ม}$$

จากนั้นออกแบบวงจรกรองความถี่ต่ำผ่านดังรูปที่ 3.6 แล้วสร้างวงจรกรองความถี่ต่ำผ่านได้ดังรูปที่ 3.7 เพื่อนำมาใช้กรองสัญญาณเวนบริดจ์ออสซิลเลเตอร์ที่มีความถี่ 1 กิโลเฮิร์ต



รูปที่ 3.6 แบบวงจรตัวกรองความถี่ต่ำผ่าน



รูปที่ 3.7 วงจรกรองความถี่ต่ำผ่าน

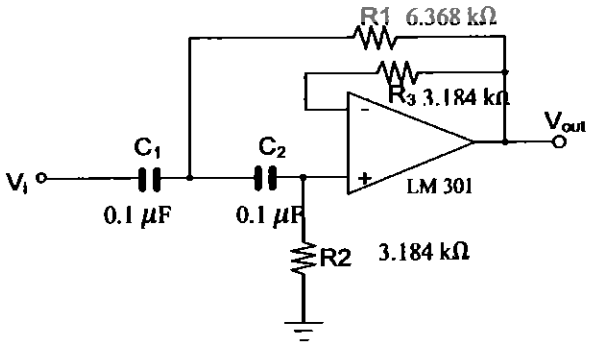
3.6 การสร้างวงจรกรองความถี่สูงผ่าน (high – pass filter : HPF)

การคำนวณหาค่าตัวแปรในการออกแบบวงจรกรองความถี่สูงผ่าน โดยกำหนดสัญญาณที่เข้ามาเท่ากับ 1 กิโลเฮิร์ต และกำหนดความถี่คัตออฟเท่ากับ 500 กิโลเฮิร์ตและเลือกใช้ตัวเก็บประจุ 0.1 ไมโครฟารัด คำนวณค่าความต้านทาน R จากสมการ

$$R_2 = R_3 = \frac{1}{2\pi f_c C} = \frac{1}{(6.28)\times(500)\times(0.01\times10^{-6})} = 3.184 \text{ กิโลโอห์ม}$$

$$R_1 = \frac{R_3}{2} = \frac{3.184\times10^{3}}{2} = 6.368 \text{ กิโลโอห์ม}$$

จากนั้นคำนินการออกแบบวงจรกรองความถี่สูงผ่านดังแสดงในรูปที่ 3.8 แล้วสร้างวงจรกรองความถี่สูงผ่านดังแสดงในรูปที่ 3.9 เพื่อใช้กรองสัญญาณเวนบริดจ์ออสซิลเลเตอร์ที่มีความถี่หนึ่งกิโลเฮิร์ต



รูปที่ 3.8 แบบวงจรกรองความถี่สูงผ่าน

รูปที่ 3.9 วงจรกรองความถี่สูงผ่าน

3.7 การสร้างวงจรกรองความถี่แถบผ่าน (band – pass filter : BPF)

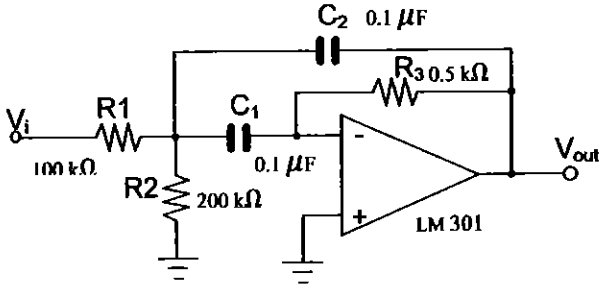
การคำนวณหาค่าตัวแปรในการออกแบบวงจรกรองสัญญาณช่วงความถี่แถบผ่านโดยกำหนดสัญญาณที่เข้ามาเท่ากับ 1 กิโลเฮิร์ต และกำหนดความถี่คัตออฟเท่ากับ 159 เฮิร์ตและ เลือกใช้ตัวเก็บประจุ 0.1 ไมโครฟารัด โดยกำหนดให้ B คือช่วงความถี่ 159 เฮิร์ตหรือ 1.0 กราดต่อวินาที คำนวณค่าความต้านทาน R จากสมการ

$$R_2 = \frac{2}{BC} = \frac{2}{(1\times10^3)(0.01\times10^{-6})} = 200 \text{ กิโลโอห์ม}$$

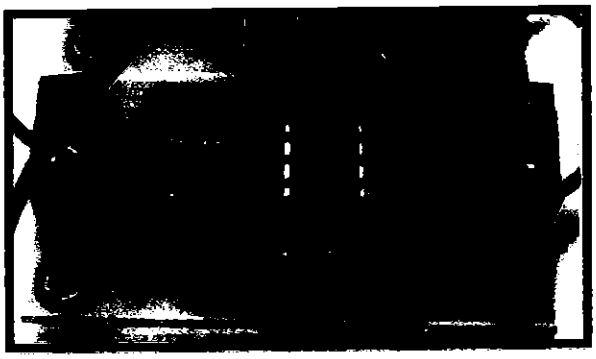
$$R_1 = \frac{R_2}{2} = \frac{(200\times10^3)}{2} = 100 \text{ กิโลโอห์ม}$$

$$R_3 = \frac{R_2}{4Q^2} = \frac{(200\times10^3)}{(4\times100)} = 0.5 \text{ กิโลโอห์ม}$$

จากนั้นทำการออกแบบวงจรกรองความถี่แถบผ่านดังแสดงในรูปที่ 3.10 เพื่อนำมาใช้กรองสัญญาณเวนบริดจ์ออสซิลเลเตอร์ที่มีความถี่ 1 กิโลเฮิร์ต แล้วสร้างวงจรกรองความถี่แถบผ่านดังแสดงในรูปที่ 3.11



รูปที่ 3.10 แบบวงจรกรองความถี่แถบผ่าน



รูปที่ 3.11 วงจรกรองความถี่แถบผ่าน

3.8 การสร้างวงจรกรองความถี่แถบหยุด (band – stop filter: BSF)

การคำนวณหาค่าตัวแปรสำหรับออกแบบวงจรกรองความถี่แถบหยุดเฉพาะช่วงแคบผ่านเมื่อกำหนดสัญญาณที่เข้ามาเท่ากับ 1 กิโลเฮิร์ต และกำหนดความถี่คัตออฟเท่ากับ 159 เฮิร์ตและเลือกใช้ตัวเก็บประจุ 0.1 ไมโครฟารัดโดยกำหนดให้ B คือช่วงความถี่ 159 เฮิร์ตหรือ 1.0 กราดต่อวินาที แล้วคำนวณค่าความต้านทาน R จากสมการ

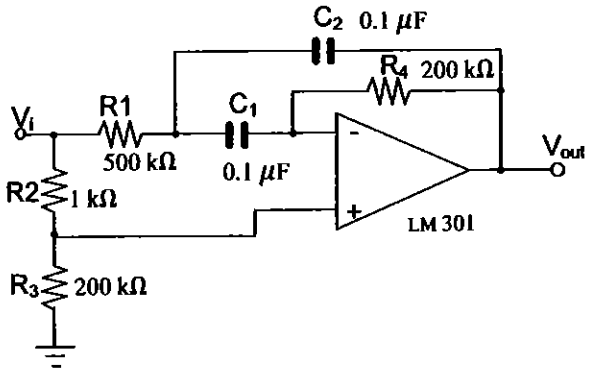
$$R_4 = \frac{2}{BC} = \frac{2}{(1\times10^3)(0.01\times10^{-6})} = 200 \text{ กิโลโอห์ม}$$

$$R_1 = \frac{R_4}{4Q^2} = \frac{(200\times10^3)}{(4\times100)} = 500 \text{ กิโลโอห์ม}$$

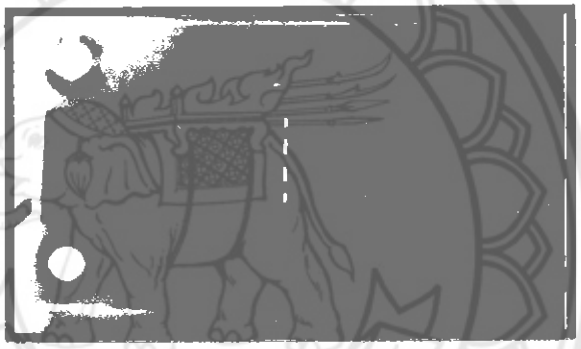
$$R_3 = 2Q^2R_2 = 200 \times (1 \times 10^3) = 200 \text{กิโลโอห์ม}$$

$$R_2 = 1 \text{ กิโลโอห์ม}$$

จากนั้นทำการออกแบบวงจรกรองความถี่แถบหยุดเฉพาะช่วงแคบผ่านดังแสดงในรูปที่ 3.12 เพื่อนำมาใช้กรองสัญญาณเวนบริดจ์ออสซิลเลเตอร์ที่มีความถี่ 1 กิโลเฮิร์ต แล้วสร้างวงจรกรองความถี่แถบหยุดเฉพาะช่วงแคบผ่านดังแสดงในรูปที่ 3.13



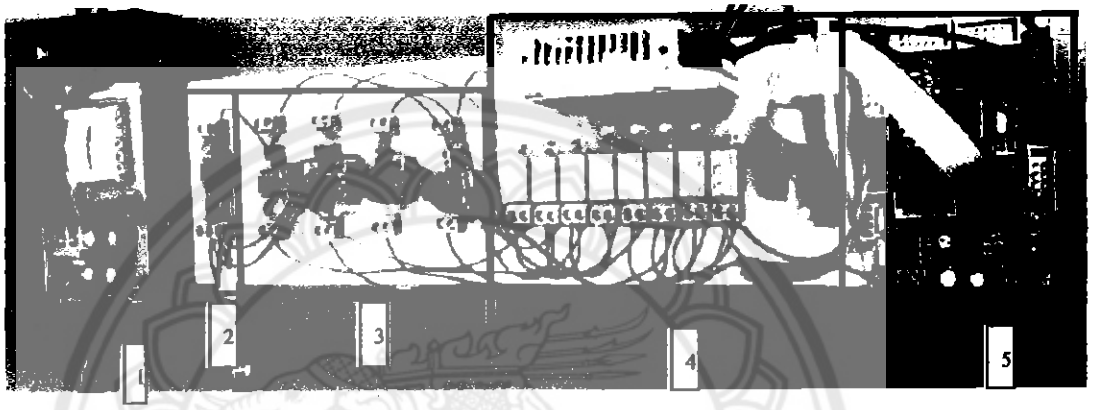
รูปที่ 3.12 แบบวงจรกรองความถี่แถบหยุดเฉพาะช่วงแคบผ่าน



รูปที่ 3.13 วงจรกรองความถี่แถบหยุดเฉพาะช่วงแคบผ่าน

3.9 การประกอบวงจรกรองความถี่พร้อมใช้งานแบบเรียลไทม์

จากการออกแบบวงจรผลิตสัญญาณเวนบริดจ์ออสซิลเลเตอร์ และวงจรกรองความถี่ 4 ชนิด นำมาต่อกับวงจรจ่ายไฟกระแสสลับ วงจรรีเลย์ วงจรสวิทช์ชิ่งซับพาย และวงจรไมโครคอนโทรลเลอร์ ได้เป็นวงจรกรองสัญญาณความถี่แบบเรียลไทม์ดังแสดงในรูปที่ 3.14



รูปที่ 3.14 วงจรกรองสัญญาณความถี่พร้อมใช้งานแบบเรียลไทม์

หมายเลข 1 คือ วงจรแหล่งจ่ายกระแสสลับป้อนไฟให้กับวงจรวีนบริดจ์ออสซิลเลเตอร์

หมายเลข 2 คือ วงจรเวนบริดจ์ออสซิลเลเตอร์ทำหน้าที่สร้างสัญญาณความถี่ ซึ่งกำหนดให้สร้างสัญญาณความถี่เท่ากับ 1 กิโลเฮิร์ต เพื่อป้อนให้กับวงจรกรองความถี่

หมายเลข 3 คือ วงจรกรองความถี่มี 4 ชนิดได้แก่ วงจรกรองความถี่ต่ำผ่าน วงจรกรองความถี่สูงผ่าน วงจรกรองความถี่แถบผ่าน และวงจรกรองความถี่แถบหยุดแบบช่วงแคบ ซึ่งวงจรกรองความถี่ทั้งสี่ชนิดนี้จะทำหน้าที่กรองสัญญาณความถี่ที่มาจากวงจรเวนบริดจ์ออสซิลเลเตอร์แล้วส่งสัญญาณให้กับรีเลย์

หมายเลข 4 คือ รีเลย์ทำหน้าที่ดึงสัญญาณจากวงจรกรองความถี่แล้วป้อนให้กับไมโครคอนโทนเลอร์

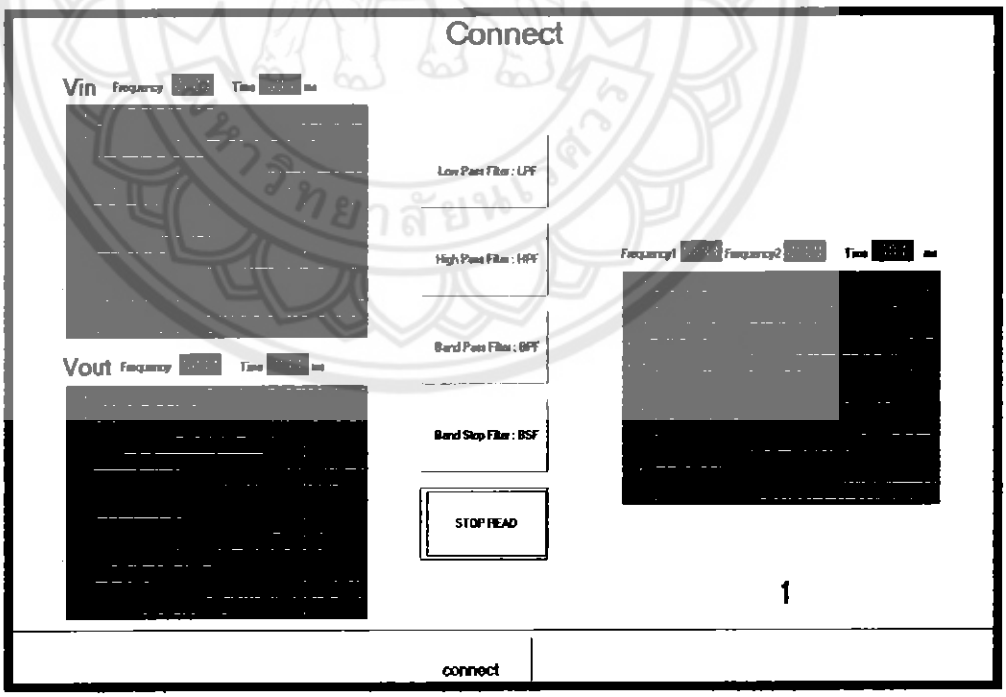
หมายเลข 5 คือ ไมโครคอนโทนเลอร์ เมื่อไมโครคอนโทรลเลอร์ได้รับสัญญาณจากรีเลย์แล้วจะส่งสัญญาณไปยังคอมพิวเตอร์เพื่อวิเคราะห์สัญญาณความถี่ต่อไป

บทที่ 4

ผลการทดลอง

4.1 หน้าต่างโปรแกรมเดลไฟ

ภายหลังการออกแบบวงจร และสร้างแต่ละวงจรตามแบบที่กำหนด ได้ทำการทดสอบวงจรกรองความถี่ทั้งสี่ชนิดได้แก่ วงจรกรองความถี่ต่ำผ่าน วงจรกรองความถี่สูงผ่าน วงจรกรองความถี่แถบผ่าน และวงจรกรองความถี่แถบหยุด จากนั้นนำแต่ละชิ้นส่วนมาประกอบเป็นวงจรกรองความถี่แบบเรียลไทม์ แล้วต่อกับเครื่องคอมพิวเตอร์ โดยใช้โปรแกรมเดลไฟ ในการแสดงสัญญาณความถี่จากวงจรกรองความถี่แบบเรียลไทม์ หน้าต่างของโปรแกรมเดลไฟแสดงดังแสดงในรูปที่ 4.1 ถ้ามีการเชื่อมต่อกับวงจรกรองความถี่แบบเรียลไทม์หน้าต่างโปรแกรมจะแสดงคำว่าคอนเนค (Connect) ถ้าไม่มีการเชื่อมต่อระหว่างวงจรกรองความถี่แบบเรียลไทม์และคอมพิวเตอร์หน้าต่างจะแสดงคำว่าดิสคอนเนค (Disconnect)



รูปที่ 4.1 หน้าต่างของโปรแกรมเดลไฟ

รายละเอียดของหน้าต่างโปรแกรมเดลไฟมีดังนี้

- Vin คือ สัญญาณอินพุต ซึ่งเกิดจากวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์

- Vout คือ สัญญาณที่ผ่านการกรองจากวงจรกรองความถี่ทั้งสี่ชนิดได้แก่ วงจรกรองความถี่ต่ำผ่าน วงจรกรองความถี่สูงผ่าน วงจรกรองความถี่แถบผ่าน และวงจรกรองความถี่แถบหยุดเฉพาะช่วงแถบผ่าน

- Connect คือ คำที่แสดงการเชื่อมต่อระหว่างวงจรกรองความถี่แบบเรียลไทม์และคอมพิวเตอร์

- Disconnect คือ คำที่แสดงการไม่เชื่อมต่อระหว่างวงจรกรองความถี่แบบเรียลไทม์และคอมพิวเตอร์

- Frequency1 คือ ค่าความถี่ตัดช่วงที่ 1

- Frequency2 คือ ค่าความถี่ตัดช่วงที่ 2

- Time คือ สัญญาณเคลื่อนที่เทียบกับเวลามีหน่วยเป็นมิลลิวินาที

- Low – pass filter: LPF คือ ปุ่มคลิกเพื่อใช้วงจรกรองความถี่ต่ำผ่านในการกรองสัญญาณความถี่จากวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์

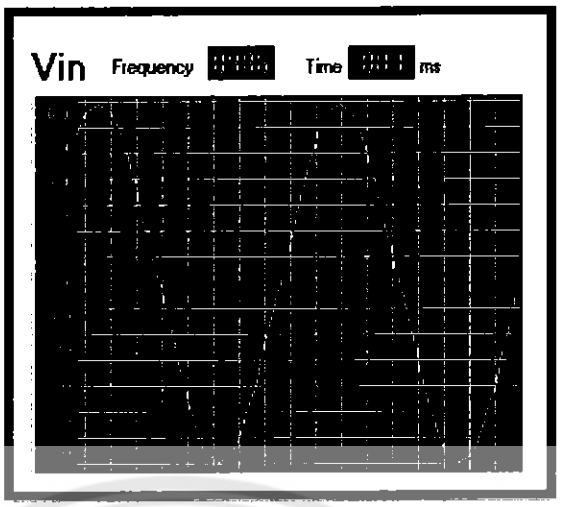
- High – pass filter: HPF คือ ปุ่มคลิกเพื่อใช้วงจรกรองความถี่สูงผ่านในการกรองสัญญาณความถี่จากวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์

- Band – pass filter: BPF คือ ปุ่มคลิกเพื่อใช้วงจรกรองความถี่แถบผ่านในการกรองสัญญาณความถี่จากวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์

- Band – stop filter: BSF คือ ปุ่มคลิกเพื่อใช้วงจรกรองความถี่แถบหยุดเฉพาะช่วงความถี่แถบผ่านในการกรองสัญญาณความถี่จากวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์

4.2 สัญญาณที่เกิดจากวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์

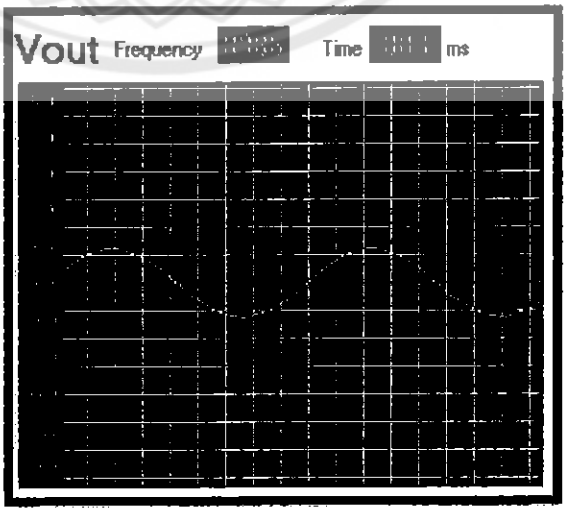
วงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์เป็นวงจรที่ใช้สำหรับสร้างสัญญาณความถี่ 1 กิโลเฮิร์ต เมื่อคอมพิวเตอร์ได้รับสัญญาณจากวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์แล้ว หน้าต่างของโปรแกรมเดลไฟจะแสดงสัญญาณรูปไซน์ ดังแสดงในรูปที่ 4.2 สัญญาณความถี่ที่เกิดจากวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์นี้จะถูกป้อนให้กับวงจรกรองความถี่ต่ำผ่าน วงจรกรองความถี่สูงผ่าน วงจรกรองความถี่แถบผ่าน และวงจรกรองความถี่แถบหยุดเฉพาะช่วงแถบผ่านต่อไป



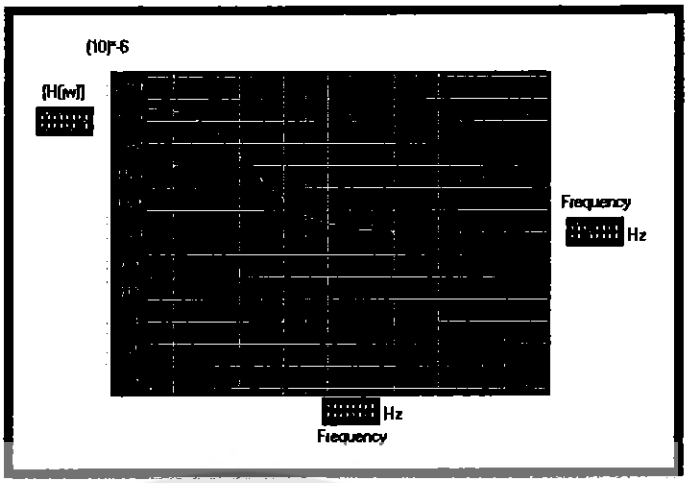
รูปที่ 4.2 สัญญาณจากวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์

4.3 สัญญาณที่เกิดจากวงจรกรองความถี่ต่ำผ่าน

วงจรกรองความถี่ต่ำผ่านได้รับสัญญาณความถี่ 1 กิโลเฮิร์ต ดังแสดงในรูปที่ 4.2 จากวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์ วงจรกรองความถี่ต่ำผ่านทำหน้าที่กรองสัญญาณความถี่โดยสัญญาณที่เกิดจากวงจรกรองความถี่ต่ำผ่านจะถูกส่งผ่านรีเลย์ และไมโครคอนโทรลเลอร์แล้วส่งข้อมูลสัญญาณความถี่ไปยังโปรแกรมเคลไฟ ซึ่งกราฟสัญญาณความถี่ที่เกิดจากวงจรผลิตความถี่ต่ำผ่านจะแสดงกราฟเป็นรูปซายน์ ดังแสดงในรูปที่ 4.3 จากนั้นใช้โปรแกรมเคลไฟวิเคราะห์สัญญาณความถี่ออกมาในรูปกราฟสัญญาณฟังก์ชั่นโครงข่ายเทียบกับความถี่ ดังแสดงในรูปที่ 4.4



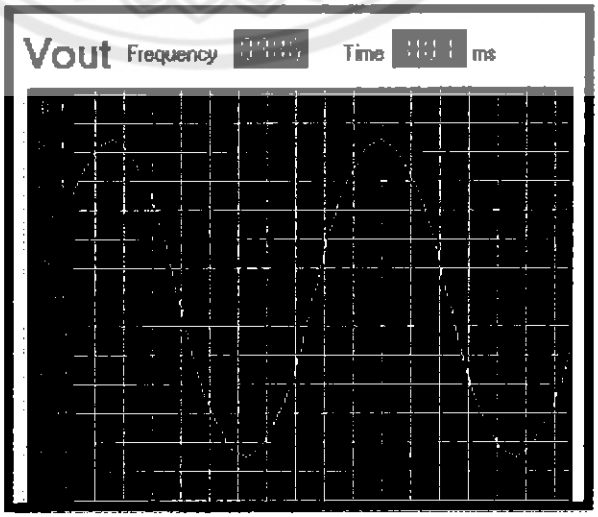
รูปที่ 4.3 สัญญาณที่เกิดจากวงจรกรองความถี่ต่ำผ่าน



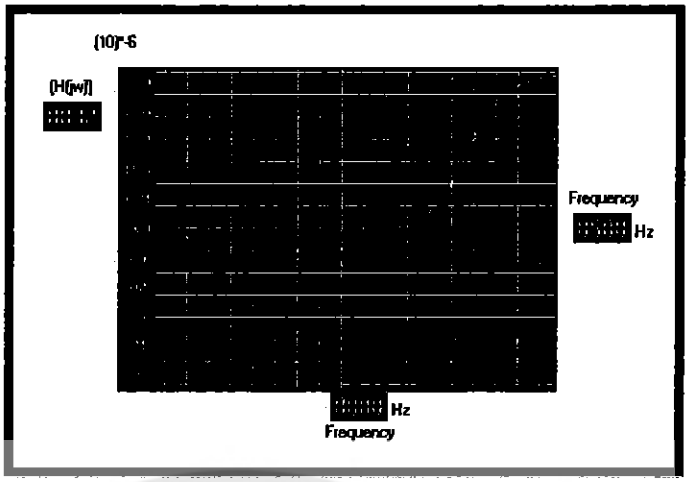
รูปที่ 4.4 กราฟแสดงการวิเคราะห์ฟังก์ชั่นโครงข่ายเทียบกับความถี่ของสัญญาณความถี่ที่มาจากวงจรกรองความถี่ต่ำผ่าน

4.4 สัญญาณที่เกิดจากวงจรกรองความถี่สูงผ่าน

วงจรกรองความถี่สูงผ่านได้รับสัญญาณความถี่ 1 กิโลเฮิร์ต ดังแสดงในรูปที่ 4.2 จากวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์ วงจรกรองความถี่สูงผ่านทำหน้าที่กรองสัญญาณความถี่แล้วส่งข้อมูลผ่านรีเลย์ และไมโครคอนโทรลเลอร์ ไปยังโปรแกรมเดลไฟ ซึ่งกราฟสัญญาณความถี่ที่เกิดจากวงจรผลิตความถี่สูงผ่านจะแสดงกราฟเป็นรูปไซน์ ดังแสดงในรูปที่ 4.5 จากนั้นโปรแกรมเดวไฟจะวิเคราะห์ค่าระหว่างแกน x คือ ความถี่ (Frequency) เทียบกับแกน y คือค่าฟังชั่นโครงข่าย หรือค่าโวลเทจเอาท์พุต (Voltage output) ดังแสดงในรูปที่ 4.6



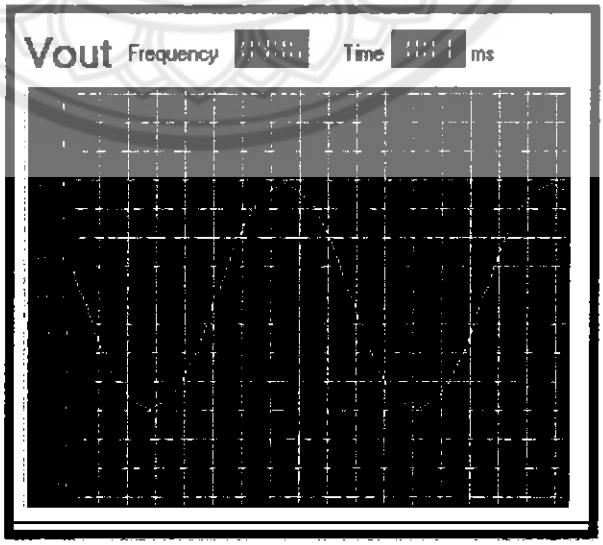
รูปที่ 4.5 สัญญาณที่เกิดจากวงจรกรองความถี่สูงผ่าน



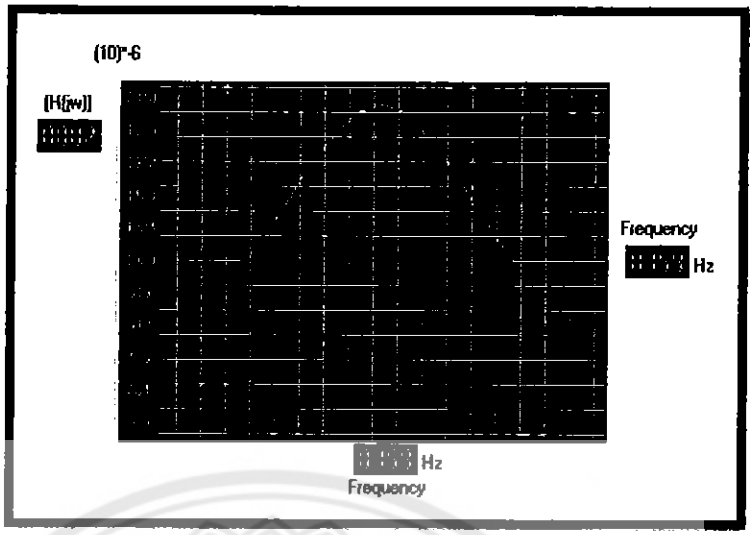
รูปที่ 4.6 กราฟแสดงการวิเคราะห์ฟังก์ชั่นโครงข่ายเทียบกับความถี่ของสัญญาณความถี่ที่มาจากวงจรกรองความถี่สูงผ่าน

4.5 สัญญาณที่เกิดจากวงจรกรองความถี่แถบผ่าน

วงจรกรองความถี่แถบผ่านได้รับสัญญาณความถี่ 1 กิโลเฮิร์ต ดังแสดงในรูปที่ 4.2 จากวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์ วงจรกรองความถี่แถบผ่านทำหน้าที่กรองสัญญาณความถี่แล้วส่งข้อมูลผ่านรีเลย์ และไมโครคอนโทรลเลอร์ ไปยังโปรแกรมเดลไฟ ซึ่งกราฟสัญญาณความถี่ที่เกิดจากวงจรผลิตความถี่แถบผ่านจะแสดงกราฟเป็นรูปไซน์ ดังแสดงในรูปที่ 4.7 จากนั้นโปรแกรมเดลไฟวิเคราะห์ค่าความถี่เทียบกับค่าฟังชั่นโครงข่ายได้กราฟแสดงดังแสดงในรูปที่ 4.8



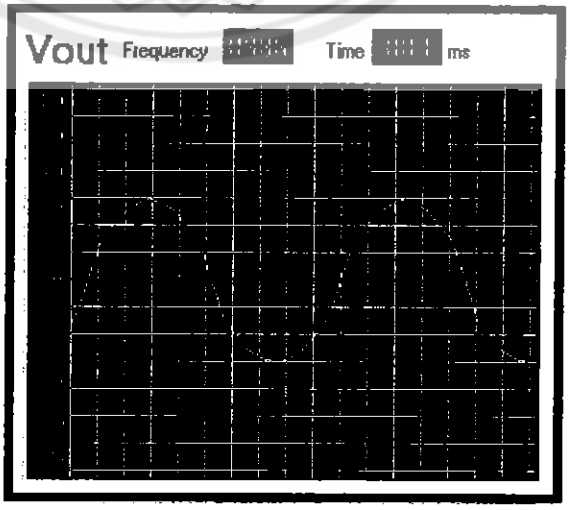
รูปที่ 4.7 สัญญาณที่เกิดจากวงจรกรองความถี่แถบผ่าน



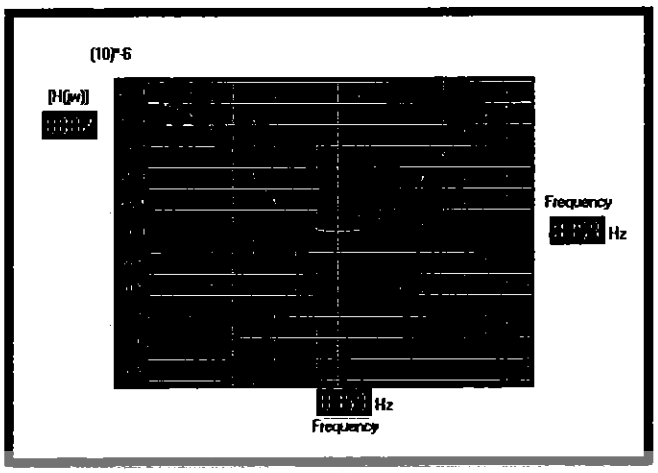
รูปที่ 4.8 กราฟแสดงการวิเคราะห์ฟังก์ชั่นโครงข่ายเทียบกับความถี่ของสัญญาณความถี่ที่มาจากวงจรกรองความถี่แถบผ่าน

4.6 สัญญาณที่เกิดจากวงจรกรองความถี่แถบหยุด

วงจรกรองความถี่แถบหยุดได้รับสัญญาณความถี่ 1 กิโลเฮิร์ต ดังแสดงในรูปที่ 4.2 จากวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์ วงจรกรองความถี่ต่ำผ่านทำหน้าที่กรองสัญญาณความถี่แล้วส่งข้อมูลไปยังโปรแกรมเคลไฟผ่านรีเลย์ และไมโครคอนโทรลเลอร์ ซึ่งกราฟสัญญาณความถี่ที่เกิดจากวงจรผลิตความถี่ต่ำผ่านจะแสดงกราฟเป็นรูปไซน์ ดังแสดงในรูปที่ 4.9 จากนั้นโปรแกรมเดวไฟวิเคราะห์ค่าความถี่เทียบกับค่าฟังชั่นโครงข่ายได้กราฟแสดงดังแสดงในรูปที่ 4.10



รูปที่ 4.9 สัญญาณวงจรกรองความถี่แถบหยุดเฉพาะช่วงแคบผ่าน



รูปที่ 4.10 กราฟแสดงการวิเคราะห์ฟังก์ชั่นโครงข่ายเทียบกับความถี่ของสัญญา**ณความถี่ที่มาจาก**วงจรกรองความถี่แถบหยุดเฉพาะช่วงแคบผ่าน

บทที่ 5
สรุปและอภิปรายผลการทดลอง

5.1 สรุปผลการดำเนินงาน

การออกแบบโปรแกรมและสร้างวงจรกรองสัญญาณความถี่แบบเรียลไทม์ เริ่มจากการสร้างวงจรแหล่งจ่ายกระแสสลับป้อนไฟให้กับวงจรสร้างความถี่แบบวีนบริดจ์ออสซิเลเตอร์ ซึ่งทำหน้าที่สร้างสัญญาณความถี่ที่กำหนดไว้เท่ากับ 1 กิโลเฮิร์ต จากนั้นวงจรสร้างสัญญาณความถี่แบบออสซิลเลเตอร์จะส่งสัญญาณที่สร้างขึ้นนี้ให้กับวงจรกรองความถี่สี่ชนิดได้แก่ วงจรกรองความถี่ต่ำผ่าน วงจรกรองความถี่สูงผ่าน วงจรกรองความถี่แถบผ่าน และวงจรกรองความถี่แถบหยุด วงจรกรองความถี่ทั้งสี่ชนิดนี้ทำหน้าที่กรองสัญญาณความถี่แล้วส่งสัญญาณให้ไมโครคอนโทรลเลอร์ โดยผ่านทางตัวรีเลย์ซึ่งทำหน้าที่สลับวงจรกรองความถี่ในการดึงสัญญาณให้กับไมโครคอนโทรลเลอร์ เมื่อไมโครคอนโทรลเลอร์ได้รับสัญญาณจากรีเลย์แล้วจะส่งสัญญาณไปยังคอมพิวเตอร์เข้าสู่โปรแกรมเคลไฟ ซึ่งแสดงสัญญาณความถี่ในรูปซายน์ เมื่อวิเคราะห์สัญญาณจากวงจรสร้างสัญญาณความถี่ 1 กิโลเฮิร์ตและวงจรกรองความถี่ ผลที่ได้จะแสดงรูปกราฟฟังก์ชั่นโครงข่ายเทียบกับความถี่แบบเรียลไทม์

5.2 ปัญหาและข้อเสนอแนะ

การออกแบบและสร้างวงจรกรองสัญญาณความถี่แบบเรียลไทม์ มีความคลาดเคลื่อนเกิดขึ้นเนื่องจากอุปกรณ์อิเล็กทรอนิกส์บางชิ้นไม่ได้มาตรฐาน ส่งผลให้ตัวโปรแกรมดึงสัญญาณออกมาผิดเพี้ยนจากค่าความเป็นจริงและทำให้ผลการทดลองมีความคลาดเคลื่อน ดังนั้นควรเลือกใช้อุปกรณ์อิเล็กทรอนิกส์ที่ได้มาตรฐานเพื่อลดความคลาดเคลื่อนที่จะเกิดขึ้นให้ได้มากที่สุด

5.3 ประเมินผล

1. มีความรู้เกี่ยวกับหลักการทำงานของโปรแกรมเคลไฟและวงจรกรองความถี่ทั้งสี่ชนิด ได้แก่ วงจรกรองความถี่ต่ำผ่าน วงจรกรองความถี่สูงผ่าน วงจรกรองความถี่แถบผ่าน และวงจรกรองความถี่แถบหยุด

2. สามารถนำหลักการทำงานของแต่ละวงจรมาสร้างวงจรกรองความถี่แบบเรียลไทม์ ซึ่งประกอบด้วยวงจรผลิตความถี่แบบเวนบริดจ์ออสซิลเลเตอร์ วงจรกรองความถี่ทั้งสี่ชนิด

3. สามารถใช้ไมโครคอนโทรลเลอร์เบอร์ ATMEGA 8 ควบคุมและดึงสัญญาณของวงจรได้
4. สามารถใช้โปรแกรมเดลไฟ (Delphi) ดึงสัญญาณออกมาในรูปแบบเรียลไทม์ได้

5.3 แนวทางในการพัฒนาโครงงานต่อไป

การสร้างวงจรสร้างสัญญาณความถี่ที่ 1 กิโลเฮิร์ตแบบเวนบริดจ์ออสซิลเลเตอร์ ส่งสัญญาณผ่านตัวกรองความถี่ 4 ชนิด สามารถเป็นแบบนำไปประยุกต์ในการสร้างวงจรผลิตความถี่ที่สามารถปรับค่าได้ ป้อนให้กับวงจรกรองความถี่ เพื่อลดทอนสัญญาณที่ไม่ต้องการให้มีอัตราส่วนสัญญาณข่าวสารต่อสัญญาณรบกวนเข้าใกล้อนันต์ให้มากที่สุด และสามารถปรับสภาพการกรองตามสัญญาณอ้างอิงที่ต้องการ ซึ่งจะต้องไม่แปรผันไปตามขนาด แต่จะแปรผันไปตามรูปแบบของสัญญาณอ้างอิงเท่านั้น

เอกสารอ้างอิง

[1] การออกแบบสร้าง และทดสอบวงจรกรองความถี่ http://www.wara.com/modules.php?name=News&file=article&sid=613 สืบค้นเมื่อ สิงหาคม 2555

[2] กนก กุสุมาลย์นุกูล .คู่มือการใช้งานโปรแกรม Borland Delphi 5 . พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : ซัสเซส มีเดีย. 2552

[3] ชนิดของวงจรกรองความถี่ http://www.hs8jyx.com/html/filter_circuit.html สืบค้นเมื่อ สิงหาคม 2555

[4] วงจรกรองความถี่ http://www.kmitl.ac.th/~kpteeraw/data_com/datacom_52/Filter.htm สืบค้นเมื่อ กรกฎาคม 2555

[5] วงจรกรองความถี่ต่ำผ่านควบคุมด้วยแรงดัน http://wara.com/article-694.html สืบค้นเมื่อ กันยายน 2555

[6] วงจรผลิตความถี่ http://wara.com/article-858.html สืบค้นเมื่อ กรกฎาคม 2555

[7] วงจรผลิตความถี่ http://www.whatgroupthai.com/whatelec/38/op-amp.html สืบค้นเมื่อ กรกฎาคม 2555

[8] นคร ภัคดีชาติ.กฤษฎา ใจเย็น.ชัยวัฒน์ ลิ้มพรจิตรวิไล. เรียน-เล่นไมโครคอนโทรลเลอร์ AVR กับ ATMEGA8.พิมพ์ครั้งที่ 1.กรุงเทพมหานคร: อินโนเวตีฟ เอ็กเพอริเมนต์ .2500.

[9] ธนันต์ ศรีสกุล. (2552). พื้นฐานการออกแบบวงจรอิเล็กทรอนิกส์. วิตตี้กรุ๊ป. กรุงเทพฯ

ภาคผนวก ก

Source code ของโปรแกรม AVR

Source code ของโปรแกรม AVR

```
#include <mega64.h>
#include <string.h>
#include <stdio.h>
#include <delay.h>
int data1[1500];
//int data2[1500];
long count=0;
long sck;
long t;

#define RXB8 1
#define TXB8 0
#define UPE 2
#define OVR 3
#define FE 4
#define UDRE 5
#define RXC 7

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<OVR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

// USART0 Receiver buffer
#define RX_BUFFER_SIZE0 8
char rx_buffer0[RX_BUFFER_SIZE0];

#if RX_BUFFER_SIZE0<256
unsigned char rx_wr_index0,rx_rd_index0,rx_counter0;
```

```
#else
unsigned int rx_wr_index0,rx_rd_index0,rx_counter0;
#endif

// This flag is set on USART0 Receiver buffer overflow
bit rx_buffer_overflow0;

// USART0 Receiver interrupt service routine
interrupt [USART0_RXC] void usart0_rx_isr(void)
{
char status,data;
status=UCSR0A;
data=UDR0;
if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
  {
  rx_buffer0[rx_wr_index0]=data;
  if (++rx_wr_index0 == RX_BUFFER_SIZE0) rx_wr_index0=0;
  if (++rx_counter0 == RX_BUFFER_SIZE0)
    {
    rx_counter0=0;
    rx_buffer_overflow0=1;
    };
  };
}

#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART0 Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
char data;
```

```
while (rx_counter0==0);
data=rx_buffer0[rx_rd_index0];
if (++rx_rd_index0 == RX_BUFFER_SIZE0) rx_rd_index0=0;
#asm("cli")
--rx_counter0;
#asm("sei")
return data;
}
#pragma used-
#endif

// USART0 Transmitter buffer
#define TX_BUFFER_SIZE0 8
char tx_buffer0[TX_BUFFER_SIZE0];

#if TX_BUFFER_SIZE0<256
unsigned char tx_wr_index0,tx_rd_index0,tx_counter0;
#else
unsigned int tx_wr_index0,tx_rd_index0,tx_counter0;
#endif

// USART0 Transmitter interrupt service routine
interrupt [USART0_TXC] void usart0_tx_isr(void)
{
if (tx_counter0)
  {
  --tx_counter0;
  UDR0=tx_buffer0[tx_rd_index0];
  if (++tx_rd_index0 == TX_BUFFER_SIZE0) tx_rd_index0=0;
  };
}
```

```
#ifndef _DEBUG_TERMINAL_IO_
// Write a character to the USART0 Transmitter buffer
#define _ALTERNATE_PUTCHAR_
#pragma used+
void putchar(char c)
{
while (tx_counter0 == TX_BUFFER_SIZE0);
#asm("cli")
if (tx_counter0 || ((UCSR0A & DATA_REGISTER_EMPTY)==0))
   {
   tx_buffer0[tx_wr_index0]=c;
   if (++tx_wr_index0 == TX_BUFFER_SIZE0) tx_wr_index0=0;
   ++tx_counter0;
   }
else
   UDR0=c;
#asm("sei")
}
#pragma used-
#endif

// Standard Input/Output functions
#include <stdio.h>

#define ADC_VREF_TYPE 0x40

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
ADMUX=adc_input|ADC_VREF_TYPE;
// Start the AD conversion
ADCSRA|=0x40;
```

```
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCW;

}
// Declare your global variables here

void main(void)
{
// Declare your local variables here
  char loop=0;
// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;
// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x00;

// Port E initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTE=0x00;
DDRE=0x00;

// Port F initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTF=0x00;
DDRF=0x00;

// Port G initialization
// Func4=In Func3=In Func2=In Func1=In Func0=In
// State4=T State3=T State2=T State1=T State0=T
PORTG=0x00;
DDRG=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
ASSR=0x00;
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
// Timer/Counter 1 initialization
```

```
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// OC1C output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
OCR1CH=0x00;
OCR1CL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
TCCR2=0x00;
```

```
TCNT2=0x00;
OCR2=0x00;

// Timer/Counter 3 initialization
// Clock source: System Clock
// Clock value: Timer 3 Stopped
// Mode: Normal top=FFFFh
// Noise Canceler: Off
// Input Capture on Falling Edge
// OC3A output: Discon.
// OC3B output: Discon.
// OC3C output: Discon.
// Timer 3 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR3A=0x00;
TCCR3B=0x00;
TCNT3H=0x00;
TCNT3L=0x00;
ICR3H=0x00;
ICR3L=0x00;
OCR3AH=0x00;
OCR3AL=0x00;
OCR3BH=0x00;
OCR3BL=0x00;
OCR3CH=0x00;
OCR3CL=0x00;

// External Interrupt(s) initialization
// INT0: Off
```

```
// INT1: Off
// INT2: Off
// INT3: Off
// INT4: Off
// INT5: Off
// INT6: Off
// INT7: Off
EICRA=0x00;
EICRB=0x00;
EIMSK=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;
ETIMSK=0x00;

// USART0 initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART0 Receiver: On
// USART0 Transmitter: On
// USART0 Mode: Asynchronous
// USART0 Baud rate: 115200
UCSR0A=0x00;
UCSR0B=0xD8;
UCSR0C=0x06;
UBRR0H=0x00;
UBRR0L=0x08;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;
```

```
// ADC initialization
// ADC Clock frequency: 1000.000 kHz
// ADC Voltage Reference: AREF pin
ADMUX=ADC_VREF_TYPE;
ADCSRA=0x84;

// ADC initialization
// ADC Clock frequency: 125.000 kHz
// ADC Voltage Reference: AREF pin
//ADMUX=ADC_VREF_TYPE;
//ADCSRA=0x8F;


// Global enable interrupts
#asm("sei")
printf("Test Rs232 For delphi7 \r\n");
while (1)
    {
    // Place your code here

    while(read_adc(0)<=20);
    while(read_adc(0)>=20)
    {
     data1[count] =  read_adc(0);
     count++;
    }
     putchar('$');
     for(t=0;t<count;t++)
     {
      if(data1[t] != 0 && data1[t] != 1024 ) printf("%i,",data1[t]);
      delay_ms(5);
```

```
        }
        putchar('!');
        count = 0;
        while(read_adc(0)>=20);

      };
}
```

ภาคผนวก ข

Source code ของโปรแกรม Borland Delphi

Source code ของโปรแกรม Borland Delphi

```
unit uscope;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, VrClasses, VrScope, StdCtrls, ComCtrls, VrScale, VrBorder,
  VrControls, VrDesign, VrLeds, VrBlotter, VrLcd, VrThreads, VrButtons,
  CPortCtl, CPort, DB, ADODB;

type
  TForm1 = class(TForm)
    VrBlotter1: TVrBlotter;
    VrUserLed1: TVrUserLed;
    VrBlotter2: TVrBlotter;
    VrShadowButton1: TVrShadowButton;
    VrShadowButton2: TVrShadowButton;
    Label3: TLabel;
    UpDown3: TUpDown;
    Label1: TLabel;
    UpDown1: TUpDown;
    Label2: TLabel;
    VrBlotter3: TVrBlotter;
    VrScope1: TVrScope;
    VrScale1: TVrScale;
    VrNum2: TVrNum;
    VrNum3: TVrNum;
    CheckBox1: TCheckBox;
    ComPort1: TComPort;
```

ComTerminal1: TComTerminal;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Memo1: TMemo;

VrBlotter4: TVrBlotter;

VrScale2: TVrScale;

ComTerminal2: TComTerminal;

Memo2: TMemo;

VrBlotter5: TVrBlotter;

VrScope3: TVrScope;

VrScale3: TVrScale;

ComTerminal3: TComTerminal;

Memo3: TMemo;

Label4: TLabel;

Label5: TLabel;

Button4: TButton;

Button5: TButton;

Button6: TButton;

Button7: TButton;

Label6: TLabel;

Label7: TLabel;

Label9: TLabel;

ADOConnection1: TADOConnection;

ADOQuery1: TADOQuery;

ADO_free: TADOQuery;

VrNum4: TVrNum;

Label8: TLabel;

VrNum5: TVrNum;

VrScope2: TVrScope;

Label10: TLabel;

VrNum7: TVrNum;

```
    ComTerminal4: TComTerminal;
    Button8: TButton;
    Timer1: TTimer;
    Label12: TLabel;
    Label13: TLabel;
    Label14: TLabel;
    VrNum1: TVrNum;
    Label15: TLabel;
    Label16: TLabel;
    VrNum6: TVrNum;
    VrNum9: TVrNum;
    Label18: TLabel;
    Label17: TLabel;
    Label19: TLabel;
    VrNum8: TVrNum;
    Label11: TLabel;
    Label20: TLabel;
    Label21: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure UpDown3Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown1Click(Sender: TObject; Button: TUDBtnType);
    procedure UpDown2Click(Sender: TObject; Button: TUDBtnType);
    procedure VrScope1NeedData(Sender: TObject; Channel: Integer;
      var Value: Integer);
    procedure CheckBox1Click(Sender: TObject);
    procedure CheckBox2Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure UpDown4Click(Sender: TObject; Button: TUDBtnType);
    procedure ComTerminal1Char(Sender: TObject; Ch: Char);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
```

```
    procedure VrScope2NeedData(Sender: TObject; Channel: Integer;
      var Value: Integer);
    procedure VrScope3NeedData(Sender: TObject; Channel: Integer;
      var Value: Integer);
    procedure Button5Click(Sender: TObject);
    procedure Button6Click(Sender: TObject);
    procedure Button7Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure ComTerminal4Char(Sender: TObject; Ch: Char);
    procedure Button8Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
    send,save_data,index,step1,step2,step3,i ,ii,iii,cp,start,loop,phase: integer;
    y: real;
    sinwave : array of integer;
    sinwave2 : array of integer;
    sinwave3 : array of integer;
    datarx :  string;
    menu,cnt_ok ,discon,cnt_e: Integer;

    N : integer;
    Wc : real;
    fs,fc : integer;
    Wc1,Wc2 : real;
    fc1,fc2 : integer;



end;
```

```
var
  Form1: TForm1;

implementation

uses Variants;

{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);
const
  PS: array[boolean] of PChar = ('START', 'STOP');
begin
  VrScope1.Active := not VrScope1.Active;
  VrUserLed1.Active := VrScope1.Active;
  VrShadowButton1.Caption := PS[VrScope1.Active];

 // ComPort1.ShowSetupDialog;
  // ComPort1.Connected:=true;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  // VrScope1.Clear;
end;

procedure TForm1.UpDown3Click(Sender: TObject; Button: TUDBtnType);
begin
  //VrNum2.Value := UpDown3.Position;
  //VrScope1.Interval := VrNum2.Value;
end;
```

```
procedure TForm1.UpDown1Click(Sender: TObject; Button: TUDBtnType);
begin
 // VrNum3.Value := UpDown1.Position;
 // VrScope1.GridSize := VrNum3.Value;
end;

procedure TForm1.UpDown2Click(Sender: TObject; Button: TUDBtnType);
begin
 // VrNum4.Value := UpDown2.Position;
 // VrScope1.Frequency := VrNum4.Value;
end;

procedure TForm1.VrScope1NeedData(Sender: TObject; Channel: Integer;
 var Value: Integer);
begin

   // if (phase=1) then
    begin
      Value := sinwave[i];   //512+
      i:=i+1;
      if (i>step1-1) then
      begin
        i:=0;
      //   phase:=2;
      end
    end;
   { if (phase=2) then
    begin
      Value := sinwave[i];   //512+
      i:=i+1;
      if (i>step1-1) then
```

```
        begin
          i:=0;
          phase:=1;
        end
      end;
}

 end;

procedure TForm1.CheckBox1Click(Sender: TObject);
begin
 vrScope1.Channels[0].Visible := CheckBox1.Checked;
end;

procedure TForm1.CheckBox2Click(Sender: TObject);
begin
 // vrScope1.Channels[1].Visible := CheckBox2.Checked;
end;

procedure TForm1.FormShow(Sender: TObject);
var  A : Integer;
begin
 vrScope1.Channels[1].Visible := false;
 vrScope2.Channels[1].Visible := false;
 vrScope3.Channels[1].Visible := false;

 SetLength(sinwave,1650);
 SetLength(sinwave2,1650);
 SetLength(sinwave3,1650);

 {
```

```
 sinwave[0]:=1;
 sinwave[1]:=2;
 sinwave[2]:=10;
 sinwave[3]:=15;
 sinwave[4]:=20;
 sinwave[5]:=25;
 sinwave[6]:=30;
 sinwave[7]:=35;
 sinwave[8]:=140;
 sinwave[9]:=150;
 sinwave[10]:=200;
 sinwave[11]:=250;
 sinwave[12]:=300;
 sinwave[13]:=350;
 sinwave[14]:=400;
 sinwave[15]:=425;
 sinwave[16]:=450;
 sinwave[17]:=500;
 sinwave[18]:=520;
 sinwave[19]:=550;

   }

// x = sin(y);
{
 for A:=0 to 20 do
 begin
    sinwave[A]:= trunc (sin( ( (A*(22/7))/180 )));
 end; .}
{
sinwave[0]:=1;
sinwave[1]:=10;
```

```
  sinwave[2]:=20;
  sinwave[3]:=35;
  sinwave[4]:=48;
  sinwave[5]:=50;
  sinwave[6]:=66;
  sinwave[7]:=78;
  sinwave[8]:=85;
  sinwave[9]:=90;

  sinwave[10]:=100;
  sinwave[11]:=120;
  sinwave[12]:=130;
  sinwave[13]:=135;
  sinwave[14]:=145;
  sinwave[15]:=165;
  sinwave[16]:=166;
  sinwave[17]:=167;
  sinwave[18]:=168;
  sinwave[19]:=169;
  }
end;

procedure TForm1.UpDown4Click(Sender: TObject; Button: TUDBtnType);
begin
  // ii:= UpDown4.Position;
  // VrNum1.Value :=ii;
end;

procedure TForm1.ComTerminal1Char(Sender: TObject; Ch: Char);
begin
  {
 if (save_data=1) then
```

```
begin

if (Ch='$') then
begin
    //VrScope1.Active:=false;

    datarx:='';
    index:=0;
    start:=0;

end
else  if (Ch='!') and (start=0) then
begin
    // VrScope1.Active:=true;
    // i:=strtoint(datarx);
    //  $ 112,545,656,885,121!

    start:=1;
    step := index;
    save_data:=0;
    VrScope1.Active:=true;
end
else   if (Ch=',') and (start=0) then
begin
    sinwave[index] := strtoint(datarx);

    memo1.Lines.Add(inttostr(sinwave[index]));
    inc(index);
    datarx:='';
end
else
begin
```

```
        if (Ch >='0') and (Ch <='9') and (start=0)  then
        datarx:=datarx+Ch;
    end;
    end;

    }
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
  ComPort1.ShowSetupDialog;
  ComPort1.Connected:=true;
  memo1.Clear;
  Timer1.Enabled :=true;
  //timer1.Enabled:=true;
end;

procedure TForm1.Button4Click(Sender: TObject);
begin

Wc :=N/2;
Wc := (2*(2.14) * fc)/fs;
VrScope3.Clear;
if ( discon =1) then
begin
   VrScope1.Active :=false;
   VrScope2.Active :=false;
   VrScope3.Active :=false;


   ADO_free.Close;
   ADO_free.sql.Clear;
```

```
ADO_free.SQL.Add('select * from t_lpf');
ADO_free.Open;
step1  := 0;

if (ADO_free.Fields[0].Value <> Null) then
begin
  i:=0;
  while not ADO_free.Eof do
  begin
       sinwave[i]:= ADO_free.Fields[1].Value;
       ADO_free.next;
       inc(i);
  end;
  step1  := i;
end;


ADO_free.Close;
ADO_free.sql.Clear;
ADO_free.SQL.Add('select * from t_lpf2');
ADO_free.Open;
step2  := 0;

if (ADO_free.Fields[0].Value <> Null) then
begin
  ii:=0;
  while not ADO_free.Eof do
  begin
       sinwave2[ii]:= ADO_free.Fields[1].Value;
       ADO_free.next;
       inc(ii);
  end;
```

```
    step2  := ii;
  end;


  ADO_free.Close;
  ADO_free.sql.Clear;
  ADO_free.SQL.Add('select * from t_lpf3');
  ADO_free.Open;
  step3  := 0;

  if  (ADO_free.Fields[0].Value <> Null) then
  begin
    iii:=0;
    while not ADO_free.Eof do
    begin
          sinwave3[iii]:=  ADO_free.Fields[1].Value;
          ADO_free.next;
          inc(iii);
    end;
    step3  := iii;
  end;



  VrScope1.Active :=true;
  VrScope2.Active :=true;
  VrScope3.Active :=true;
  i:=0;
  ii:=0;
  iii:=0;
```

```
    VrNum4.Value := 986;    VrNum1.Value := 1;
    VrNum5.Value := 986;    VrNum6.Value := 1;
    VrNum7.Value := 500;    VrNum8.Value := 0;
    VrNum9.Value := 4;

    ComPort1.WriteStr('A');
    menu:=1;
   end;
 end;

 procedure TForm1.VrScope2NeedData(Sender: TObject; Channel: Integer;
  var Value: Integer);
 begin
     Value := sinwave2[ii];  //512+
     ii:=ii+1;
     if (ii>step2-1) then
     begin
        ii:=0;
     end
end;

procedure TForm1.VrScope3NeedData(Sender: TObject; Channel: Integer;
 var Value: Integer);
begin
     Value := sinwave3[iii];  //512+
     iii:=iii+1;
     if (iii>step3-1) then
     begin
         iii:=0;

         VrScope3.Active :=false;
     end
```

```
end;

procedure TForm1.Button5Click(Sender: TObject);
begin




Wc :=N/2;
Wc := (2*(2.14) * fc)/fs;
VrScope3.Clear;


if ( discon =1) then
begin
  VrScope1.Active :=false;
  VrScope2.Active :=false;
  VrScope3.Active :=false;


  ADO_free.Close;
  ADO_free.sql.Clear;
  ADO_free.SQL.Add('select * from t_lpf');
  ADO_free.Open;
  step1  := 0;

  if  (ADO_free.Fields[0].Value <> Null) then
  begin
    i:=0;
    while not ADO_free.Eof do
    begin
         sinwave[i]:=  ADO_free.Fields[1].Value;
```

```
        ADO_free.next;
        inc(i);
    end;
    step1  := i;
end;


ADO_free.Close;
ADO_free.sql.Clear;
ADO_free.SQL.Add('select * from t_hpf2');
ADO_free.Open;
step2  := 0;

if  (ADO_free.Fields[0].Value <> Null) then
begin
   ii:=0;
   while not ADO_free.Eof do
   begin
        sinwave2[ii]:=  ADO_free.Fields[1].Value;
        ADO_free.next;
        inc(ii);
   end;
   step2  := ii;
end;


ADO_free.Close;
ADO_free.sql.Clear;
ADO_free.SQL.Add('select * from t_hpf3');
ADO_free.Open;
step3  := 0;
```

```
if (ADO_free.Fields[0].Value <> Null) then
begin
  iii:=0;
  while not ADO_free.Eof do
  begin
      sinwave3[iii]:= ADO_free.Fields[1].Value;
      ADO_free.next;
      inc(iii);
  end;
  step3  := iii;
end;


VrScope1.Active :=true;
VrScope2.Active :=true;
VrScope3.Active :=true;
i:=0;
ii:=0;
iii:=0;

VrNum4.Value := 986;    VrNum1.Value := 1;
VrNum5.Value := 986;    VrNum6.Value := 1;
VrNum7.Value := 500;    VrNum8.Value := 0;
VrNum9.Value := 12;

ComPort1.WriteStr('B');
menu:=2;
end;
end;
```

```
procedure TForm1.Button6Click(Sender: TObject);
begin

Wc1 := (2*(2.14) * fc1)/fs;
Wc2 := (2*(2.14) * fc2)/fs;
VrScope3.Clear;


if ( discon =1) then
begin
   VrScope1.Active :=false;
   VrScope2.Active :=false;
   VrScope3.Active :=false;


   ADO_free.Close;
   ADO_free.sql.Clear;
   ADO_free.SQL.Add('select * from t_lpf');
   ADO_free.Open;
   step1  := 0;


   if  (ADO_free.Fields[0].Value <> Null) then
   begin
      i:=0;
      while not ADO_free.Eof do
      begin
            sinwave[i]:=  ADO_free.Fields[1].Value;
            ADO_free.next;
            inc(i);
      end;
      step1  := i;
   end;
```

```
ADO_free.Close;
ADO_free.sql.Clear;
ADO_free.SQL.Add('select * from t_bpf2');
ADO_free.Open;
step2  := 0;

if (ADO_free.Fields[0].Value <> Null) then
begin
  ii:=0;
  while not ADO_free.Eof do
  begin
       sinwave2[ii]:= ADO_free.Fields[1].Value;
       ADO_free.next;
       inc(ii);
  end;
  step2  := ii;
end;


ADO_free.Close;
ADO_free.sql.Clear;
ADO_free.SQL.Add('select * from t_bpf3');
ADO_free.Open;
step3  := 0;

if (ADO_free.Fields[0].Value <> Null) then
begin
  iii:=0;
  while not ADO_free.Eof do
  begin
```

```
            sinwave3[iii]:=  ADO_free.Fields[1].Value;
            ADO_free.next;
            inc(iii);
        end;
        step3  := iii;
    end;


    VrScope1.Active :=true;
    VrScope2.Active :=true;
    VrScope3.Active :=true;
    i:=0;
    ii:=0;
    iii:=0;

    VrNum4.Value := 986;     VrNum6.Value := 1;
    VrNum5.Value := 986;     VrNum1.Value := 1;
    VrNum7.Value := 159;     VrNum8.Value := 159;
    VrNum9.Value := 2;


    ComPort1.WriteStr('C');
    menu:=3;
  end;

end;

procedure TForm1.Button7Click(Sender: TObject);
begin
Wc1 := (2*(2.14) * fc1)/fs;
```

```
Wc2 := (2*(2.14) * fc2)/fs;
VrScope3.Clear;

if ( discon =1) then
begin
   VrScope1.Active :=false;
   VrScope2.Active :=false;
   VrScope3.Active :=false;


   ADO_free.Close;
   ADO_free.sql.Clear;
   ADO_free.SQL.Add('select * from t_lpf');
   ADO_free.Open;
   step1  := 0;

   if (ADO_free.Fields[0].Value <> Null) then
   begin
     i:=0;
     while not ADO_free.Eof do
     begin
          sinwave[i]:=  ADO_free.Fields[1].Value;
          ADO_free.next;
          inc(i);
     end;
     step1  := i;
   end;


   ADO_free.Close;
   ADO_free.sql.Clear;
   ADO_free.SQL.Add('select * from t_bsf2');
```

```
ADO_free.Open;
step2  := 0;

if  (ADO_free.Fields[0].Value <> Null) then
begin
   ii:=0;
   while not ADO_free.Eof do
   begin
         sinwave2[ii]:=  ADO_free.Fields[1].Value;
         ADO_free.next;
         inc(ii);
   end;
   step2  := ii;
end;


ADO_free.Close;
ADO_free.sql.Clear;
ADO_free.SQL.Add('select * from t_bsf3');
ADO_free.Open;
step3  := 0;

if  (ADO_free.Fields[0].Value <> Null) then
begin
   iii:=0;
   while not ADO_free.Eof do
   begin
         sinwave3[iii]:=  ADO_free.Fields[1].Value;
         ADO_free.next;
         inc(iii);
   end;
   step3  := iii;
```

```
    end;



    VrScope1.Active :=true;
    VrScope2.Active :=true;
    VrScope3.Active :=true;
    i:=0;
    ii:=0;
    iii:=0;

     VrNum4.Value := 986;     VrNum1.Value := 1;
    VrNum5.Value := 986;      VrNum6.Value := 1;
    VrNum7.Value := 159;      VrNum8.Value := 159;
    VrNum9.Value := 2;


    ComPort1.WriteStr('D');
    menu:=4;
   end;

end;

procedure TForm1.Timer1Timer(Sender: TObject);
var   y : Integer;
begin
    inc(cnt_ok);
    label14.Caption := inttostr(cnt_ok);


    inc(cnt_e);
```

```
if  cnt_e = 4 then
begin
     if menu = 1 then Button4.Click;
     if menu = 2 then Button5.Click;
     if menu = 3 then Button6.Click;
     if menu = 4 then Button7.Click;

end;


if (cnt_ok = 3) then
begin
     discon :=0;
    // if ( discon = 1) then
    // begin
      Label12.Caption := 'Disconnect';
      for y:=0 to step1 do    sinwave[y]:=1024;
      for y:=0 to step2 do    sinwave2[y]:=1024;
      for y:=0 to step3 do    sinwave3[y]:=1024;

      Button8.Click;
     // discon:=0;
    // end;

end
else
begin
   if ((discon = 0) and (cnt_ok = 1)    )then
   begin
       discon:=1;
       Label12.Caption := 'Connect';
       if menu = 1 then Button4.Click;
```

```
            if menu = 2 then Button5.Click;
            if menu = 3 then Button6.Click;
            if menu = 4 then Button7.Click;
        end;
    end;


end;

procedure TForm1.ComTerminal4Char(Sender: TObject; Ch: Char);
var   y : Integer;
begin

 if (Ch = 'K') then
 begin
     cnt_ok:=0;
    // discon :=1;
 end;

 if (Ch = 'E') then
 begin
     cnt_e:=0;
     for y:=0 to step1 do    sinwave[y]:=1024;
     for y:=0 to step2 do    sinwave2[y]:=1024;
     for y:=0 to step3 do    sinwave3[y]:=1024;
 end;


end;

procedure TForm1.Button8Click(Sender: TObject);
var   y : Integer;
```

```
begin
  ComPort1.WriteStr('E');

   for y:=0 to step1 do    sinwave[y]:=1024;
   for y:=0 to step2 do    sinwave2[y]:=1024;
   for y:=0 to step3 do    sinwave3[y]:=1024;

  VrScope1.Clear;
  VrScope2.Clear;
  VrScope3.Clear;
   VrScope1.Active :=false;
  VrScope2.Active :=false;
  VrScope3.Active :=false;

   VrNum4.Value := 0;
   VrNum5.Value := 0;
   VrNum7.Value := 0;

end;
procedure TForm1.FormCreate(Sender: TObject);
begin
discon :=0;

fc:=1;
fs:=1;
fc1:=1;
fc2:=1;
end;

end.
```

ภาคผนวก ค

รายละเอียดของไอซีหมายเลข LM 301

National Semiconductor

September 1999

LM101A/LM201A/LM301A Operational Amplifiers

General Description

The LM101A series are general purpose operational amplifiers which feature improved performance over industry standards like the LM709. Advanced processing techniques make possible an order of magnitude reduction in input currents, and a redesign of the biasing circuitry reduces the temperature drift of input current. Improved specifications include:

- Offset voltage 3 mV maximum over temperature (LM101A/LM201A)
- Input current 100 nA maximum over temperature (LM101A/LM201A)
- Offset current 20 nA maximum over temperature (LM101A/LM201A)
- Guaranteed drift characteristics
- Offsets guaranteed over entire common mode and supply voltage ranges
- Slew rate of 10V/μs as a summing amplifier

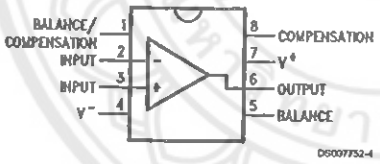
This amplifier offers many features which make its application nearly foolproof: overload protection on the input and output, no latch-up when the common mode range is exceeded, and freedom from oscillations and compensation with a single 30 pF capacitor. It has advantages over internally compensated amplifiers in that the frequency compensation can be tailored to the particular application. For example, in low frequency circuits it can be overcompensated for increased stability margin. Or the compensation can be optimized to give more than a factor of ten improvement in high frequency performance for most applications.

In addition, the device provides better accuracy and lower noise in high impedance circuitry. The low input currents also make it particularly well suited for long interval integrators or timers, sample and hold circuits and low frequency waveform generators. Further, replacing circuits where matched transistor pairs buffer the inputs of conventional IC op amps, it can give lower offset voltage and a drift at a lower cost.

The LM101A is guaranteed over a temperature range of −55°C to +125°C, the LM201A from −25°C to +85°C, and the LM301A from 0°C to +70°C.

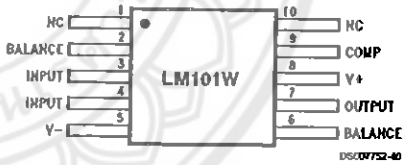
Connection Diagrams (Top View)

Dual-In-Line Package



Order Number LM101AJ, LM101J/883 (Note 1), LM201AN or LM301AN
See NS Package Number J08A or N08E

Ceramic Flatpack Package

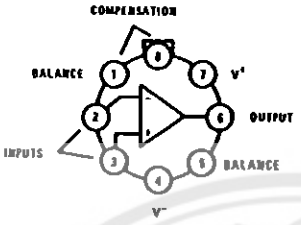


Order Number LM101AW/883 or LM101W/883
See NS Package Number W10A

© 1999 National Semiconductor Corporation

Connection Diagrams (Top View) (Continued)

Metal Can Package

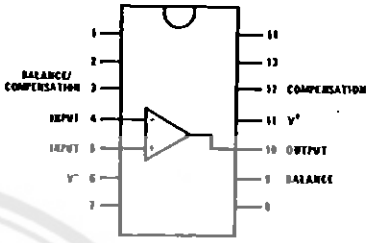


DS007752-2

Note: Pin 4 connected to case.

Order Number LM101AH, LM101AH/883 (Note 1), LM201AH or LM301AH
See NS Package Number H08C

Dual-In-Line Package



DS007752-3

Order Number LM101AJ-14/883 (Note 1)
See NS Package Number J14A

Note 1: Available per JM38510/10103.

Absolute Maximum Ratings (Note 2)

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

	LM101A/LM201A	LM301A
Supply Voltage	±22V	±18V
Differential Input Voltage	±30V	±30V
Input Voltage (Note 3)	±15V	±15V
Output Short Circuit Duration (Note 4)	Continuous	Continuous
Operating Ambient Temp. Range	−55°C to +125°C (LM101A) −25°C to +85°C (LM201A)	0°C to +70°C
T_J Max		
H-Package	150°C	100°C
N-Package	150°C	100°C
J-Package	150°C	100°C
Power Dissipation at T_A = 25°C		
H-Package (Still Air)	500 mW	300 mW
(400 LF/Min Air Flow)	1200 mW	700 mW
N-Package	900 mW	500 mW
J-Package	1000 mW	650 mW
Thermal Resistance (Typical) θ_{JA}		
H-Package (Still Air)	165°C/W	165°C/W
(400 LF/Min Air Flow)	67°C/W	67°C/W
N Package	135°C/W	135°C/W
J-Package	110°C/W	110°CmW
(Typical) θ_{JC}		
H-Package	25°C/W	25°C/W
Storage Temperature Range	−65°C to +150°C	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)		
Metal Can or Ceramic	300°C	300°C
Plastic	260°C	260°C
ESD Tolerance (Note 7)	2000V	2000V

Electrical Characteristics (Note 5)

$T_A = T_J$

Parameter	Conditions		LM101A/LM201A			LM301A			Units
			Min	Typ	Max	Min	Typ	Max	
Input Offset Voltage	T_A = 25°C, $R_S \le 50\ k\Omega$			0.7	2.0		2.0	7.5	mV
Input Offset Current	T_A = 25°C			1.5	10		3.0	50	nA
Input Bias Current	T_A = 25°C			30	75		70	250	nA
Input Resistance	T_A = 25°C		1.5	4.0		0.5	2.0		MΩ
Supply Current	T_A = 25°C	V_S = ±20V		1.8	3.0				mA
		V_S = ±15V					1.8	3.0	mA
Large Signal Voltage Gain	T_A = 25°C, V_S = ±15V V_{OUT} = ±10V, $R_L \ge 2\ k\Omega$		50	160		25	160		V/mV
Input Offset Voltage	$R_S \le 50\ k\Omega$				3.0			10	mV
Average Temperature Coefficient of Input Offset Voltage	$R_S \le 50\ k\Omega$			3.0	15		6.0	30	µV/°C
Input Offset Current					20			70	nA
Average Temperature Coefficient of Input Offset Current	25°C ≤ $T_A \le T_{MAX}$			0.01	0.1		0.01	0.3	nA/°C
	$T_{MIN} \le T_A \le$ 25°C			0.02	0.2		0.02	0.6	nA/°C
Input Bias Current					0.1			0.3	µA
Supply Current	$T_A = T_{MAX}$, V_S = ±20V			1.2	2.5				mA

Electrical Characteristics (Note 5) (Continued)

$T_A = T_J$

Parameter	Conditions		LM101A/LM201A			LM301A			Units
			Min	Typ	Max	Min	Typ	Max	
Large Signal Voltage Gain	$V_S = \pm 15V$, $V_{OUT} = \pm 10V$ $R_L \geq 2k$		25			15			V/mV
Output Voltage Swing	$V_S = \pm 15V$	$R_L = 10\ k\Omega$	±12	±14		±12	±14		V
		$R_L = 2\ k\Omega$	±10	±13		±10	±13		V
Input Voltage Range	$V_S = \pm 20V$		±15						V
	$V_S = \pm 15V$			+15, −13		±12	+15, −13		V
Common-Mode Rejection Ratio	$R_S \leq 50\ k\Omega$		80	96		70	90		dB
Supply Voltage Rejection Ratio	$R_S \leq 50\ k\Omega$		80	96		70	96		dB

Note 2: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur.Operating ratings indicate for which the device is functional, but do no guarantee specific performance limits. Electrical Characteristics state DC and AC electrical specifications under particular test conditions which guarantee specific limits. This assumes that the device is within the Operating Ratings. Specifications are not guaranteed for parameters where no limit is given, however, the typical value is a good indication of device performance.

Note 3: For supply voltages less than ±15V, the absolute maximum input voltage is equal to the supply voltage.

Note 4: Continuous short circuit is allowed for case temperatures to 125°C and ambient temperatures to 75°C for LM101A/LM201A, and 70°C and 55°C respectively for LM301A.

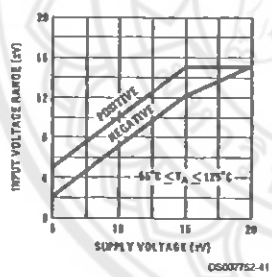
Note 5: Unless otherwise specified, these specifications apply for C1 = 30 pF, $\pm 5V \leq V_S \leq \pm 20V$ and $-55°C \leq T_A \leq +125°C$ (LM101A), $\pm 5V \leq V_S \leq \pm 20V$ and $-25°C \leq T_A \leq +85°C$ (LM201A), $\pm 5V \leq V_S \leq \pm 15V$ and $0°C \leq T_A \leq +70°C$ (LM301A).

Note 6: Refer to RETS101AX for LM101A military specifications and RETS101X for LM101 military specifications.

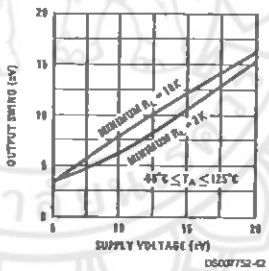
Note 7: Human body model, 100 pF discharged through 1.5 kΩ.

Guaranteed Performance Characteristics LM101A/LM201A

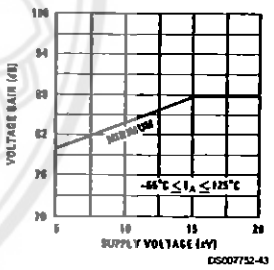
Input Voltage Range



Output Swing

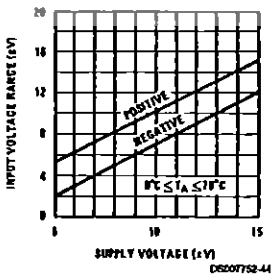


Voltage Gain

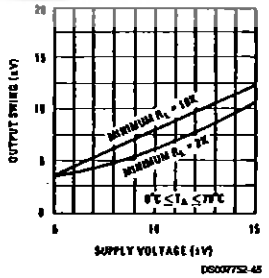


Guaranteed Performance Characteristics LM301A

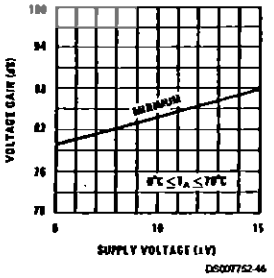
Input Voltage Range



Output Swing

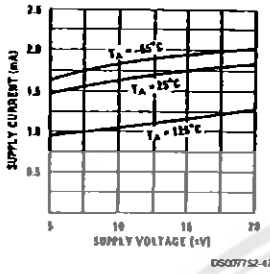


Voltage Gain

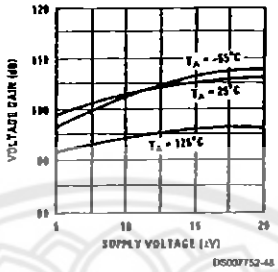


Typical Performance Characteristics

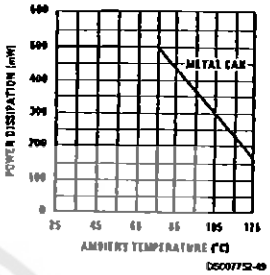
Supply Current



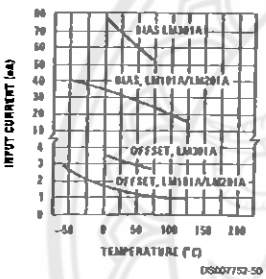
Voltage Gain



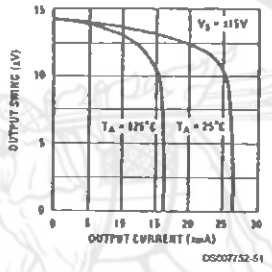
Maximum Power Dissipation



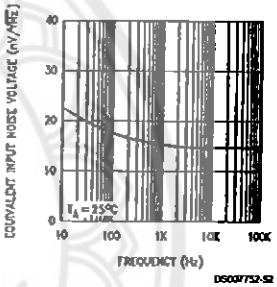
Input Current, LM101A/LM201A/LM301A



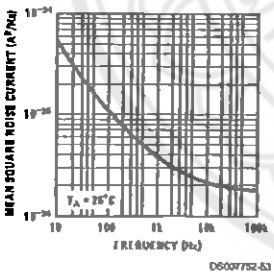
Current Limiting



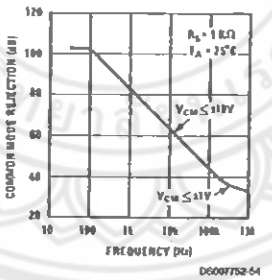
Input Noise Voltage



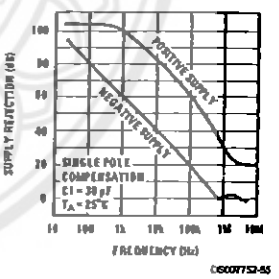
Input Noise Current



Common Mode Rejection

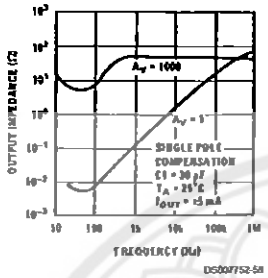


Power Supply Rejection



Typical Performance Characteristics (Continued)

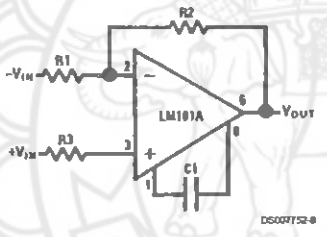
Closed Loop Output Impedance



DS007752-58

Typical Performance Characteristics for Various Compensation Circuits (Note 9)

Single Pole Compensation

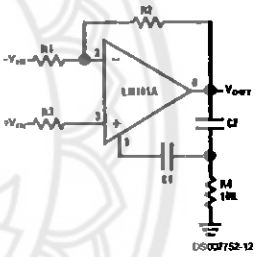


DS007752-9

$$C1 \geq \frac{R1\,C_S}{R1 + R2}$$

C_S= 30 pF

Two Pole Compensation

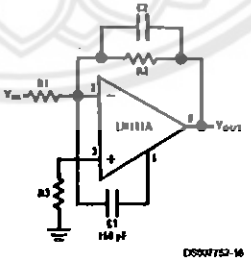


DS007752-12

$$C1 \geq \frac{R1\,C_S}{R1 + R2}$$

C_S= 30 pF
C2 = 10 C1

Feedforward Compensation



DS007752-10

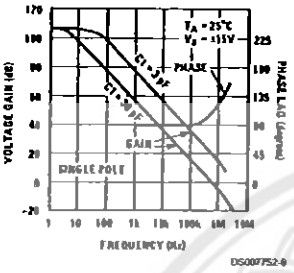
$$C2 = \frac{1}{2\pi f_o R2}$$

f_o= 3 MHz

Typical Performance Characteristics for Various Compensation Circuits

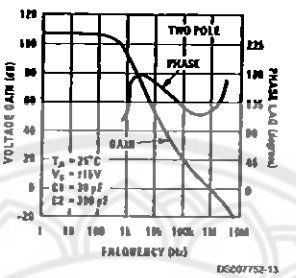
(Note 9) (Continued)

Open Loop Frequency Response



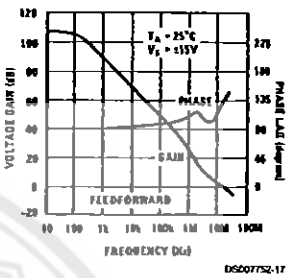
DS007752-9

Open Loop Frequency Response



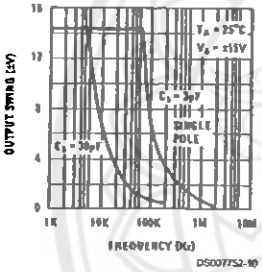
DS007752-13

Open Loop Frequency Response



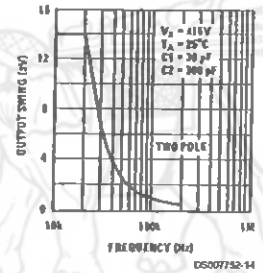
DS007752-17

Large Signal Frequency Response



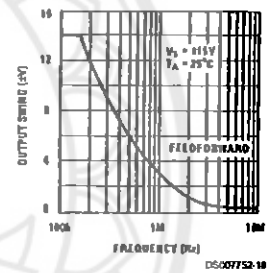
DS007752-10

Large Signal Frequency Response



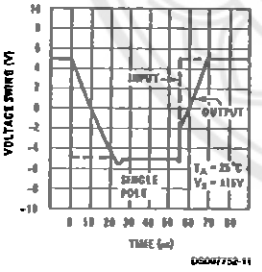
DS007752-14

Large Signal Frequency Response



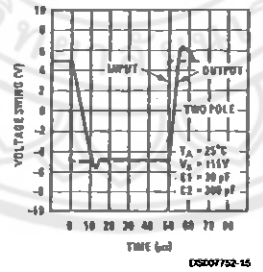
DS007752-18

Voltage Follower Pulse Response



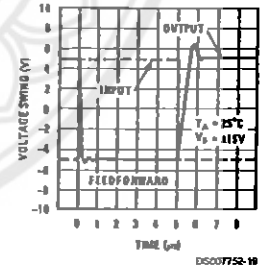
DS007752-11

Voltage Follower Pulse Response



DS007752-15

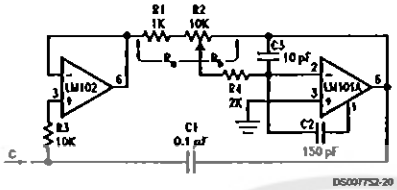
Inverter Pulse Response



DS007752-19

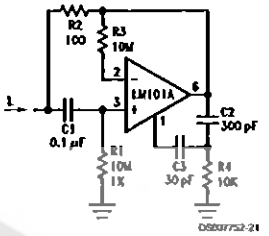
Typical Applications (Note 9)

Variable Capacitance Multiplier



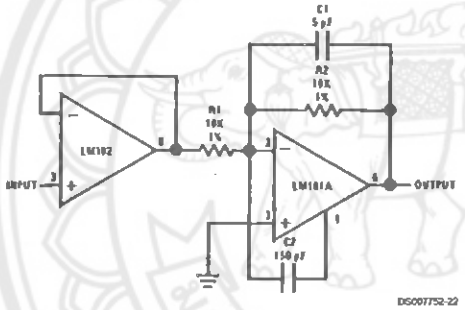
$$C = 1 + \frac{R_b}{R_a} C1$$

Simulated Inductor

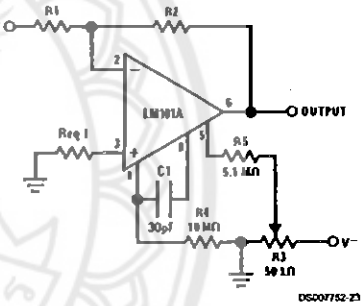


$L = R1\,R2\,C1$
$R_S = R2$
$R_P = R1$

Fast Inverting Amplifier with High Input Impedance

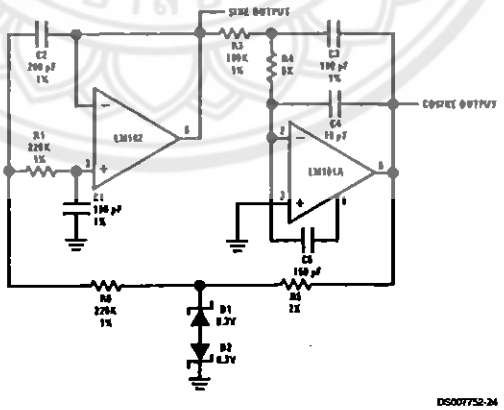


Inverting Amplifier with Balancing Circuit



1May be zero or equal to parallel combination of R1 and R2 for minimum offset.

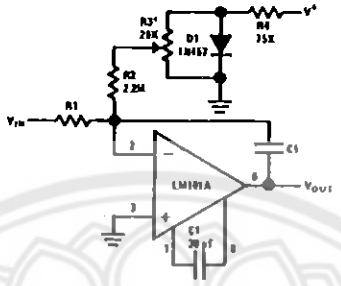
Sine Wave Oscillator



f_o = 10 kHz

Typical Applications (Note 9) (Continued)

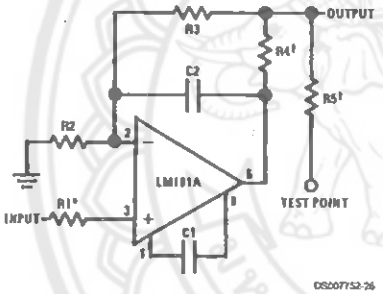
Integrator with Bias Current Compensation



*Adjust for zero integrator drift. Current drift typically 0.1 nA/°C over −55°C to +125°C temperature range.

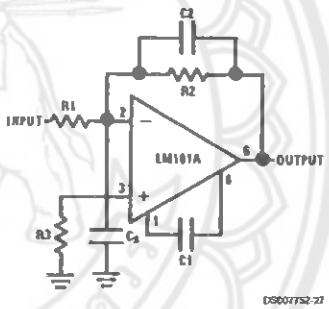
Application Hints (Note 9)

Protecting Against Gross Fault Conditions



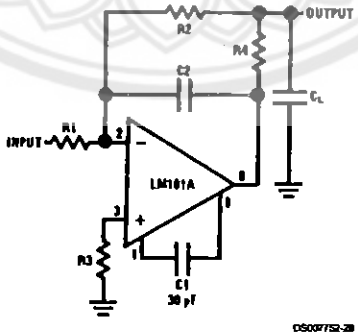
*Protects input
†Protects output
‡Protects output — not needed when R4 is used.

Compensating for Stray Input Capacitances or Large Feedback Resistor



$$C2 = \frac{R1\,C_S}{R2}$$

Isolating Large Capacitive Loads



Although the LM101A is designed for trouble free operation, experience has indicated that it is wise to observe certain precautions given below to protect the devices from abnormal operating conditions. It might be pointed out that the advice given here is applicable to practically any IC op amp, although the exact reason why may differ with different devices.

Application Hints (Note 9) (Continued)

When driving either input from a low-impedance source, a limiting resistor should be placed in series with the input lead to limit the peak instantaneous output current of the source to something less than 100 mA. This is especially important when the inputs go outside a piece of equipment where they could accidentally be connected to high voltage sources. Large capacitors on the input (greater than 0.1 μF) should be treated as a low source impedance and isolated with a resistor. Low impedance sources do not cause a problem unless their output voltage exceeds the supply voltage. However, the supplies go to zero when they are turned off, so the isolation is usually needed.

The output circuitry is protected against damage from shorts to ground. However, when the amplifier output is connected to a test point, it should be isolated by a limiting resistor, as test points frequently get shorted to bad places. Further, when the amplifer drives a load external to the equipment, it is also advisable to use some sort of limiting resistance to preclude mishaps.

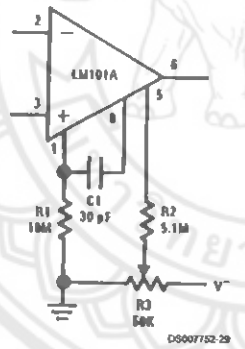
Precautions should be taken to insure that the power supplies for the integrated circuit never become reversed—even under transient conditions. With reverse voltages greater than 1V, the IC will conduct excessive current, fusing internal aluminum interconnects. If there is a possibility of this happening, clamp diodes with a high peak current rating should be installed on the supply lines. Reversal of the voltage between V^+ and V^- will always cause a problem, although reversals with respect to ground may also give difficulties in many circuits.

The minimum values given for the frequency compensation capacitor are stable only for source resistances less than 10 kΩ, stray capacitances on the summing junction less than 5 pF and capacitive loads smaller than 100 pF. If any of these conditions are not met, it becomes necessary to overcompensate the amplifier with a larger compensation capacitor. Alternately, lead capacitors can be used in the feedback network to negate the effect of stray capacitance and large feedback resistors or an RC network can be added to isolate capacitive loads.

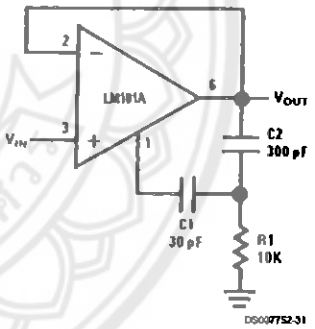
Although the LM101A is relatively unaffected by supply bypassing, this cannot be ignored altogether. Generally it is necessary to bypass the supplies to ground at least once on every circuit card, and more bypass points may be required if more than five amplifiers are used. When feed-forward compensation is employed, however, it is advisable to bypass the supply leads of each amplifier with low inductance capacitors because of the higher frequencies involved.

Typical Applications (Note 9)

Standard Compensation and Offset Balancing Circuit



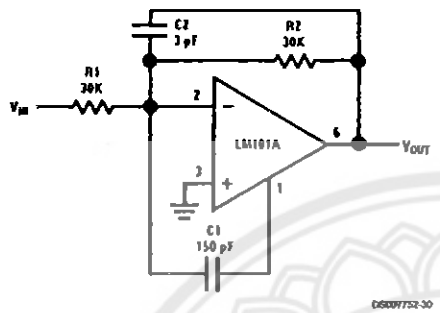
Fast Voltage Follower



Power Bandwidth: 15 kHz
Slew Rate: 1V/μs

Typical Applications (Note 9) (Continued)

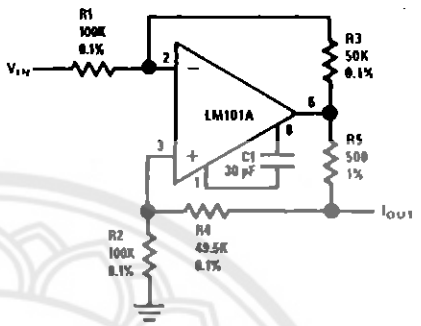
Fast Summing Amplifier



DS007752-30

Power Bandwidth: 250 kHz
Small Signal Bandwidth: 3.5 MHz
Slew Rate: 10V/μs

Bilateral Current Source

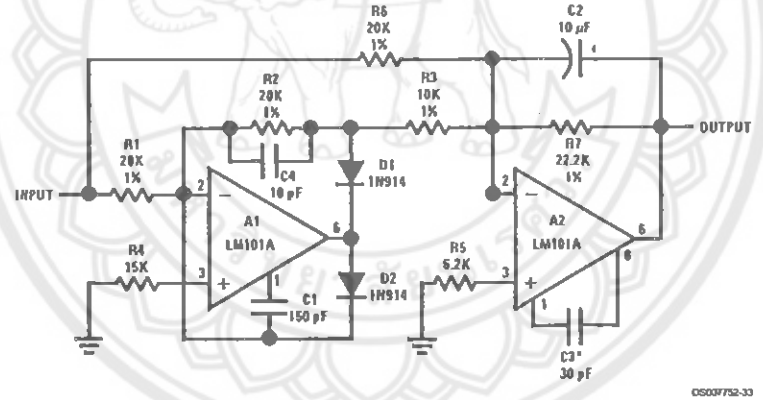


DS007752-32

$$I_{OUT} = \frac{R3\,V_{IN}}{R1\,R5}$$

R3 = R4 + R5
R1 = R2

Fast AC/DC Converter (Note 8)

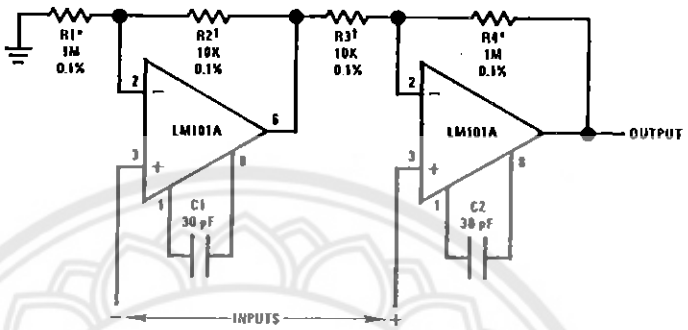


DS007752-33

Note 8: Feedforward compensation can be used to make a fast full wave rectifier without a filter.

Typical Applications (Note 9) (Continued)

Instrumentation Amplifier

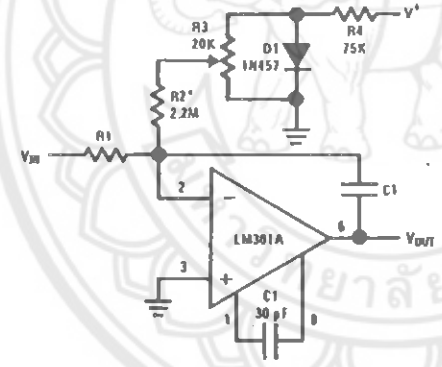


R1 = R4; R2 = R3

$$A_V = 1 + \frac{R1}{R2}$$

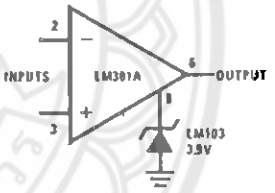
*,† Matching determines CMRR.

Integrator with Bias Current Compensation



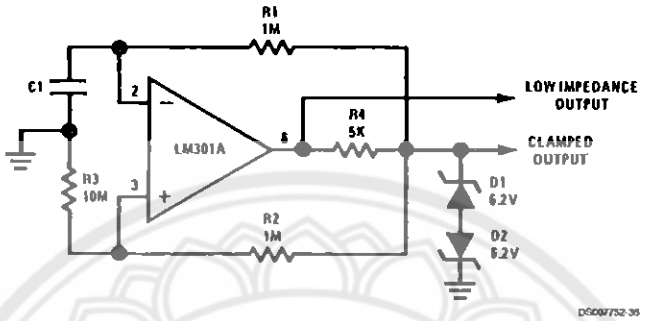
*Adjust for zero integrator drift. Current drift typically 0.1 nA/°C over 0°C to +70°C temperature range.

Voltage Comparator for Driving RTL Logic or High Current Driver

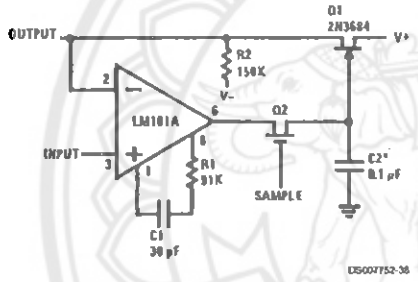


Typical Applications (Note 9) (Continued)

Low Frequency Square Wave Generator

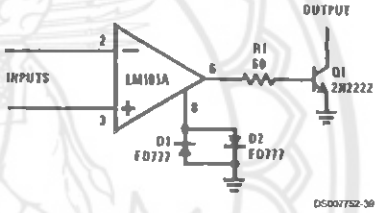


Low Drift Sample and Hold

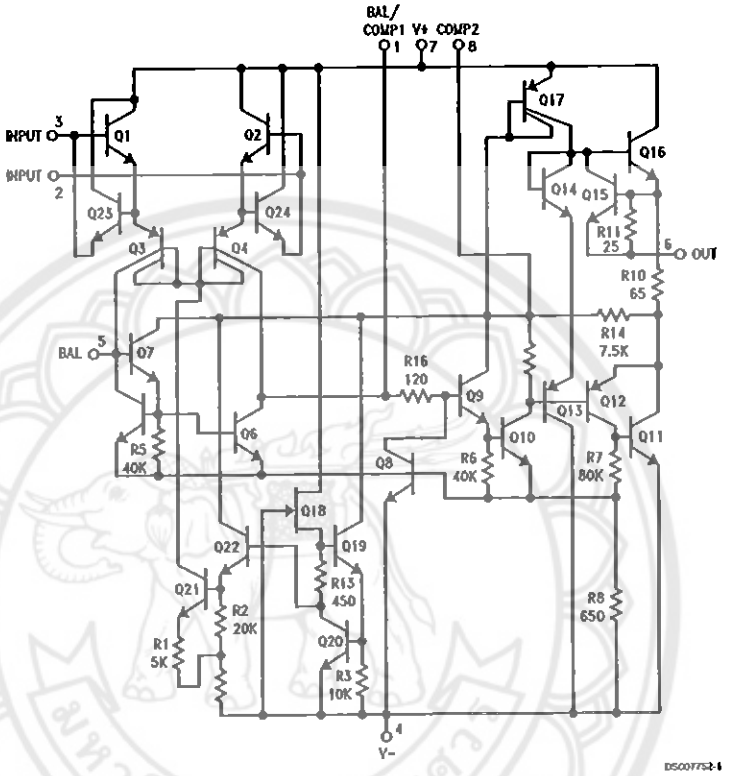


*Polycarbonate-dielectric capacitor

Voltage Comparator for Driving DTL or TTL Integrated Circuits

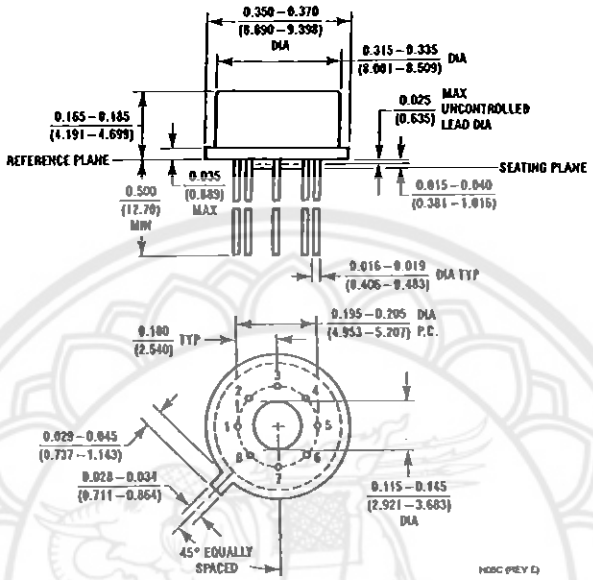


Schematic (Note 9)

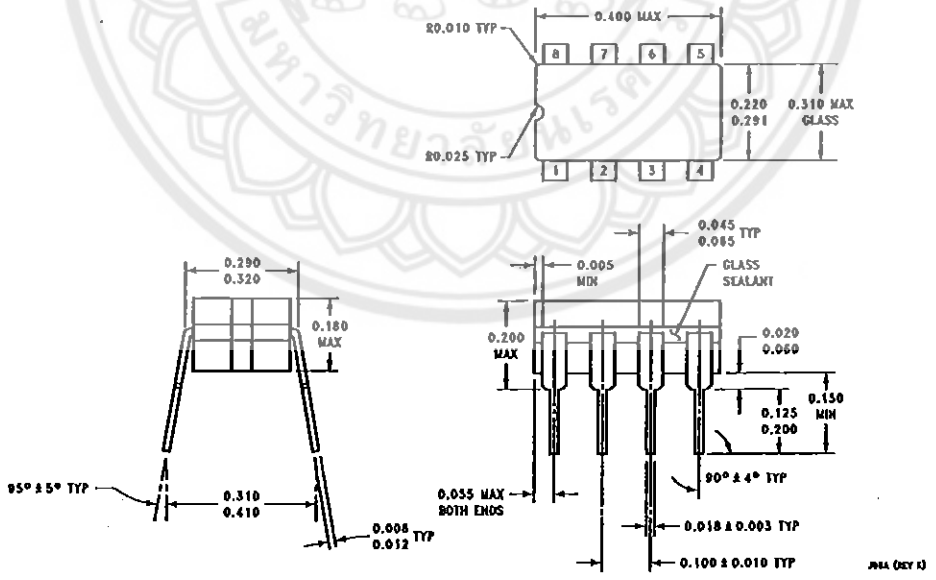


Note 9: Pin connections shown are for 8-pin packages.

Physical Dimensions Inches (millimeters) unless otherwise noted

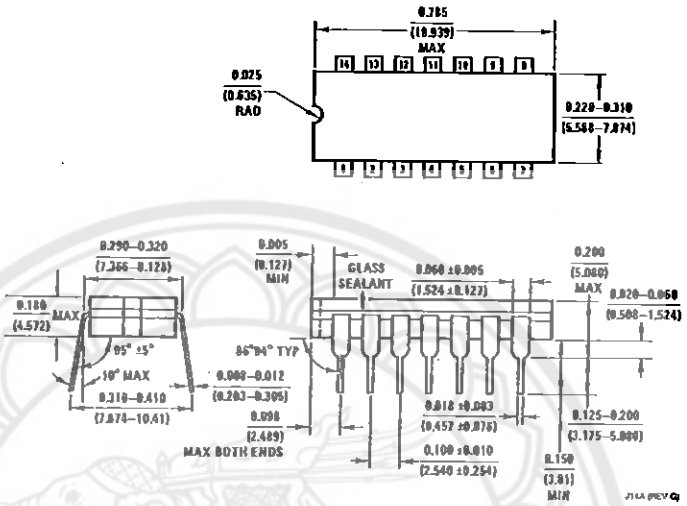


Metal Can Package (H)
Order Number LM101AH, LM101AH/883
LM201AH or LM301AH
NS Package Number H08C

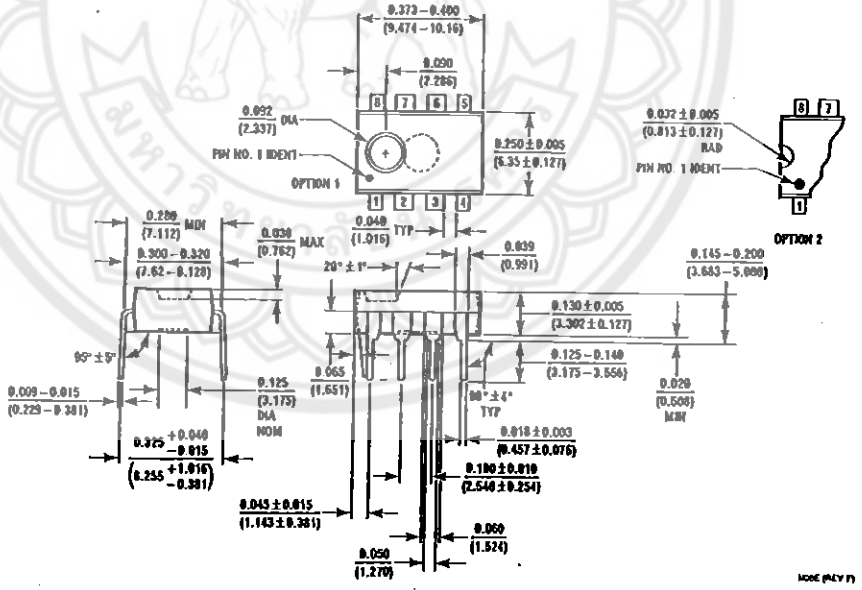


Ceramic Dual-In-Line Package (J)
Order Number LM101J/883 or LM101AJ
NS Package Number J08A

Physical Dimensions Inches (millimeters) unless otherwise noted (Continued)

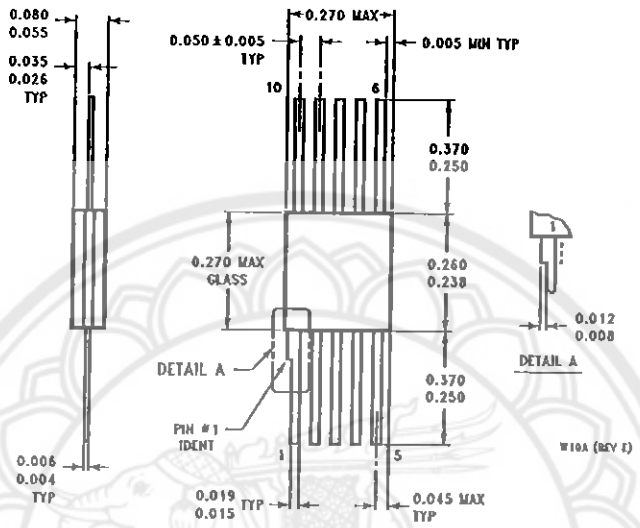


Ceramic Dual-In-Line Package (J)
Order Number LM101AJ-14/883
NS Package Number J14A



Molded Dual-In-Line Package (N)
Order Number LM201AN or LM301AN
NS Package Number N08E

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



Ceramic Flatpack Package (W)
Order Number LM101AW/883 or LM101W/883
NS Package Number W10A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
Americas
Tel: 1-800-272-9959
Fax: 1-800-737-7018
Email: support@nsc.com
www.national.com

National Semiconductor Europe
Fax: +49 (0) 1 80-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 1 80-530 85 85
English Tel: +49 (0) 1 80-532 78 32
Français Tel: +49 (0) 1 80-532 93 58
Italiano Tel: +49 (0) 1 80-534 16 80

National Semiconductor Asia Pacific Customer Response Group
Tel: 65-2544466
Fax: 65-2504466
Email: sea.support@nsc.com

National Semiconductor Japan Ltd.
Tel: 81-3-5639-7560
Fax: 81-3-5639-7507

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.