



เครื่องปรับแต่งเสียงด้วยคอมพิวเตอร์

PC BASE EQUALIZER



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ปีการศึกษา 2537

ปริญญาานิพนธ์ปีการศึกษา 2537

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง เครื่องปรับแต่งเสียงด้วยคอมพิวเตอร์

ผู้จัดทำ

1. นายอนันต์ สุทธิมาลา 34109479
2. นายอรรถพล พลธิชประเสริฐ 34109490



(Handwritten signature)

(ศ. ดร. วิมลภ สุระกำพลธร)

อาจารย์ที่ปรึกษา

เครื่องปรับแต่งเสียงควบคุมด้วยคอมพิวเตอร์

นาย อนันต์ สุทธิมาลา 34109479

นาย อรรถพล พณิชประเสริฐ 34109490

ศ.ดร. วัลลภ สุระกำพลธร อาจารย์ที่ปรึกษา
ปีการศึกษา 2537

บทคัดย่อ

เครื่องปรับแต่งเสียง มีความสำคัญต่อระบบอุปกรณ์เครื่องเสียงเป็นอันมาก เพราะใช้ปรับแต่งเสียงให้ได้ตามต้องการจึงเหมาะสมที่จะทำการพัฒนาให้เป็นระบบอัตโนมัติโดยใช้คอมพิวเตอร์เป็นตัวควบคุมซึ่งนอกจากสามารถที่จะปรับแต่งเสียงได้โดยอัตโนมัติแล้วยังสามารถที่จะเก็บข้อมูลและและนำข้อมูลมาใช้ได้ ซึ่งทำให้เกิดความสะดวกในการใช้งานอีกด้วย

PC BASE EQUALIZER

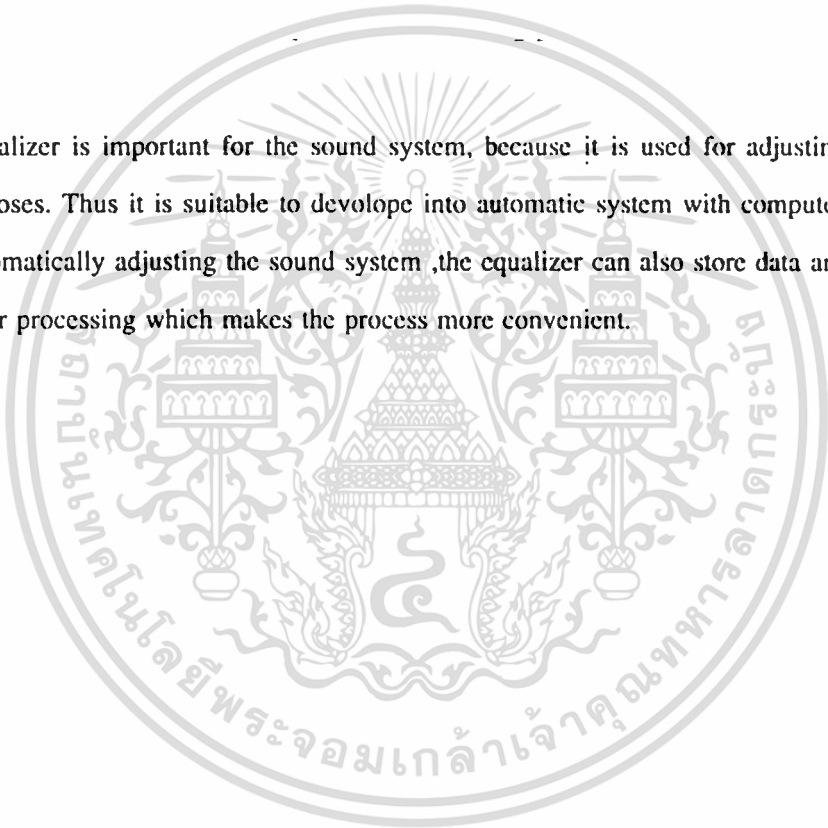
Anan Suthimala 34109479

Atthapol Panitprasuit 34109490

Professor Dr.Wanlop Surakumpoltron Advisor

Abstract

Equalizer is important for the sound system, because it is used for adjusting sound for special purposes. Thus it is suitable to develop into automatic system with computer controller. Besides automatically adjusting the sound system, the equalizer can also store data and return the data back for processing which makes the process more convenient.



สารบัญ

เนื้อเรื่อง	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีเบื้องต้นของฟิลเตอร์	2
คำจำกัดความต่างๆ ของฟิลเตอร์	2
การออกแบบแอกทีฟฟิลเตอร์	5
การเลือกอัตราขยายของออปแอมป์เพื่อกำหนดค่า Q	9
คุณสมบัติของฟิลเตอร์แบบต่างๆ	13
Bi-Quadratic filter	14
บทที่ 3 ชนิดของ อีควอไลเซอร์	16
กราฟฟิสิกอีควอไลเซอร์	17
พารามตริกอีควอไลเซอร์	18
พารากราฟฟิสิกอีควอไลเซอร์	19
ประโยชน์ของอีควอไลเซอร์	19
บทที่ 4 การออกแบบอีควอไลเซอร์	22
การเลือกช่วงความถี่	24
Band Selection	25
การเลือกค่า Q สูงสุด	26
การออกแบบค่านัดกเตอร์โดยใช้วงจรถ่ายเรเตอร์	26
บทที่ 5 การออกแบบและการสร้าง	30
ส่วนกาวด์ที่ติดต่อกับอีควอไลเซอร์	30
ส่วนที่ควบคุมอีควอไลเซอร์	33
ส่วนของอีควอไลเซอร์	38
บทที่ 6 การทดลอง	42
บทที่ 7 สรุปผลและวิจารณ์	50
ภาคผนวก	
กิตติกรรมประกาศ	
เอกสารอ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

เครื่องเล่นแผ่นหรือเครื่องเล่นเทป แอมป์หรือลำโพง เป็นเพียงอุปกรณ์ที่ช่วยให้ได้เสียงสมบูรณ์ มีคุณภาพตามที่เทปหรือแผ่นเสียงถูกบันทึกไว้ แม้ว่าแอมป์จะมีปุ่มปรับเสียง ทัม-แหลม แต่ก็เพียงเป็นการเพิ่มปรับเสียง ในขอบเขตที่ถูกจำกัดตามที่ออกแบบวงจรมา “ อีควอลไลเซอร์ “ จึงเป็นอุปกรณ์ที่มีบทบาทโดยตรงในการตอบสนองต่อความต้องการนี้

ในอดีต อีควอลไลเซอร์เป็นเพียงส่วนประกอบในการปรุงแต่งเสียงสำหรับมือในวงการดนตรีเท่านั้น สาเหตุสำคัญได้แก่ ราคาที่แพงเกินกว่าผู้เล่นแผ่นเสียงโดยทั่วไปจะซื้อได้ และขาดความรู้พื้นฐานในขีดความสามารถของอีควอลไลเซอร์ และคุณสมบัติของเสียงและการใช้อีควอลไลเซอร์ได้อย่างมีประสิทธิภาพยังต้องอาศัยประสบการณ์ และความชำนาญอย่างมาก ปัจจุบัน อีควอลไลเซอร์ได้รับการพัฒนาจนเป็นส่วนประกอบสำคัญชิ้นหนึ่งในชุดเครื่องเสียง โดยสามารถเลือกซื้อได้ในราคาที่ถูกลง แต่เจ้าของก็ยังใช้ไม่เป็นและขาดความรู้พื้นฐานเกี่ยวกับเสียงและตัวอีควอลไลเซอร์

อีควอลไลเซอร์ในโครงการนี้ เป็นการออกแบบตามท้องตลาด แต่การควบคุมนี้การใช้งานผ่านทางคอมพิวเตอร์ ซึ่งปัจจุบันมีราคาถูก อีกทั้งในโครงการนี้ได้ทำในตัวอีควอลไลเซอร์สามารถที่จะปรับค่าให้เหมาะกับระบบได้เอง โดยตัวอีควอลไลเซอร์นี้สามารถปรับความถี่ต่างๆ กัน ครอบคลุมทุกๆ ย่านความถี่ ตั้งแต่ 63 Hz จนถึง 16 kHz สำหรับการควบคุมค่าทางดิจิทัล จะทำงานในลักษณะ logic switching เพื่อป้อนเลขไบนารี 8 บิตเพื่อควบคุมค่าการ บูทส์ และ คัท

บทที่ 2

ทฤษฎีเบื้องต้นของฟิลเตอร์

ในวงจรอิควอไลเซอร์นั้น ฟิลเตอร์เป็นส่วนสำคัญอย่างยิ่งในเรื่องความถี่ ดังนั้นการเข้าใจเกี่ยวกับวงจรฟิลเตอร์จึงเป็นเรื่องที่สำคัญมาก ในปัจจุบันได้มีการออกแบบ ไอซีออปแอมป์มาใช้ในวงจรฟิลเตอร์อย่างแพร่หลาย ทำให้สามารถออกแบบภาคฟิลเตอร์ได้อย่างง่าย

2.1 คำจำกัดความต่างๆ ของฟิลเตอร์

วงจรฟิลเตอร์ก็คือวงจรที่ไม่ยอมให้สัญญาณที่ไม่ต้องการผ่านไปได้ออกเป็น 4 แบบคือ โดยทั่วไปแบ่ง

1. วงจรกรองความถี่สูง (High pass Filters : HPF)
2. วงจรกรองความถี่ต่ำ (Low pass Filter : LPF)
3. วงจรกรองความถี่เฉพาะ (Band pass Filter : BPF)
4. วงจรกรองทั้งความถี่เฉพาะ (Band-Eliminated Filter : BEF)

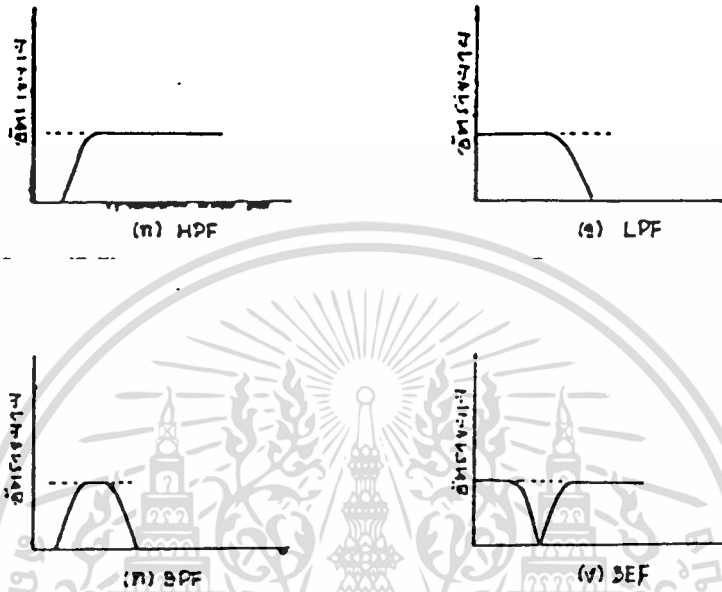
1. **วงจรกรองความถี่สูง** คือวงจรที่กรองสัญญาณความถี่ต่ำเอาไว้ โดยยอมให้สัญญาณที่มีความถี่สูงผ่านไป บางทีเรียกว่า “Low Cut Filter”

2. **วงจรกรองความถี่ต่ำ** คือวงจรที่ยอมให้สัญญาณความถี่ต่ำผ่านไปได้นั้น แล้วจะกรองเอาสัญญาณความถี่สูงไว้ บางทีเรียกว่า “High Cut Filter”

3. **วงจรกรองความถี่เฉพาะ** คือวงจรที่ให้สัญญาณความถี่เฉพาะบางความถี่ผ่านไปได้ ส่วนสัญญาณอื่นๆ ไม่ยอมให้ผ่าน

4. **วงจรกรองทั้งความถี่เฉพาะ** เป็นวงจรที่ทำหน้าที่กลับกันกับวงจรกรองความถี่เฉพาะ คือจะกรองความถี่ในบางช่วงเท่านั้น ส่วนช่วงอื่นจะยอมให้ผ่านไปทั้งหมด

คุณสมบัติของฟิลเตอร์แต่ละแบบทั้ง 4 แบบนี้ สามารถเขียนกราฟต่างๆ ไปเพื่ออธิบายการทำงานได้ดังรูปที่ 2.1



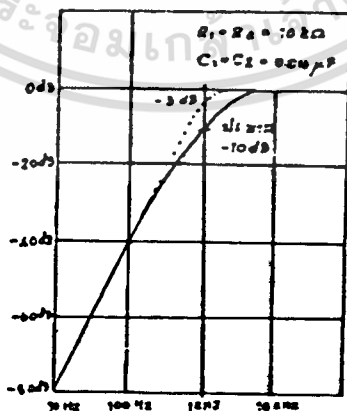
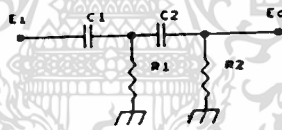
รูปที่ 2.1 คุณสมบัติของฟิลเตอร์แบบต่างๆ



รูปที่ 2.2 วงจรกรองความถี่สูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรมัลติเพล็กซ์พื้นฐานที่ 2.2 เป็นแบบอย่างง่าย โดยใช้ R และ C ในการกรองความถี่ จากความสัมพันธ์ของสัญญาณอินพุต และ เอาท์พุท จะได้ค่าเท่ากับ -3 dB ซึ่งเป็นค่าความถี่คัทออฟ โดยทั่วไปในวงจรมัลติเพล็กซ์ต่างๆ ไปถือค่า -6 dB เป็นเกณฑ์ ในกรณีที่ต้องการตัดสัญญาณให้มากกว่า -3 dB ดังตัวอย่างในรูปที่ 3 นั้นจะทำโดยเพิ่ม RC เข้าไปอีก 1 ชุดจะได้ค่าคัทออฟเป็น -12 dB/oct และในทำนองเดียวกัน ถ้าเพิ่ม RC เป็น 3 ชุด จะได้ค่าคัทออฟเป็น -18 dB/oct สรุปได้ว่า RC 1 ชุดจะให้ค่าเพิ่มขึ้น 6 dB แต่ปัญหามียุคคือ ในรูปที่ 2.3 นี้ ถ้าใช้ RC ที่มีค่าเท่ากัน 2 ชุดต่อกัน ค่าของคัทออฟจะมีค่าประมาณ -10 dB/oct สาเหตุที่เป็นเช่นนี้ เนื่องจากค่าของ RC ที่ใช้ไม่สามารถหาค่าที่แน่นอนได้จริงๆ ทุกประการ เราอาจจะแก้ไขโดยใช้ RLC มาเป็นวงจรมัลติเพล็กซ์ การออกแบบวงจรนี้จะให้คุณสมบัติของคัทออฟที่สามารถเปลี่ยนแปลงได้หรือที่เราเรียกว่า “แอกทีฟ ฟิเตอร์” นั้นเราสามารถใส่ทรานซิสเตอร์หรือออปแอมป์ มาช่วยในวงจรมัลติเพล็กซ์ ข้อได้เปรียบในวงจรมัลติเพล็กซ์นั้นมีอยู่หลายประการที่เห็นได้ชัดๆ ก็คือไม่ต้องใช้ L ในวงจรทำให้การกรองความถี่ทำได้ผลดีและสามารถลดขนาดของวงจรลงได้มาก การสร้างผ่านของการกรองสัญญาณกระทำได้ง่ายและให้ผลดีกว่า



รูปที่ 2.3 วงจร RC 2 ชุดต่อกัน

2.2 การออกแบบแอกทีฟฟิลเตอร์

ในรูปที่ 2.4 คือวงจรกรองความถี่ต่ำ e_1 และ e_2 ในวงจรนี้คือ

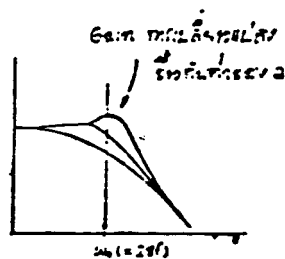
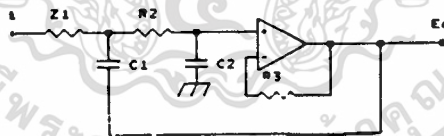
$$\frac{e_1}{e_2} = \frac{w_0^2}{s^2 + (w_0/Q)s + w_0^2}$$

ซึ่งในที่นี้ $s = jw$ (w คือ angular frequency = $2\pi f$) ในสมการค่า s จะเป็นค่ายกกำลังสอง. ในวงจร RC 2 ชุดต่อกันอยู่ Transfer function จึงเป็น Secound order เราจึงเรียกว่า secound order filter

Q คือค่า Quality factor หรืออีกนัยหนึ่ง Q ก็คือค่าแสดง damping ของความถี่คัทออฟนั่นเอง ซึ่งคุณสมบัติของความถี่คัทออฟ เมื่อพิจารณาจะพบว่าขึ้นอยู่กับค่า Q และ w_0 เป็นสำคัญ

w_0 คือค่าความถี่คัทออฟ ในที่นี้เราให้ค่า w และ w_0 ให้เท่ากันก่อน

ในรูปที่ 2.4 ค่าของ j ที่เขียนอยู่ด้วยแสดงถึงเฟสหน้าหน้า 90° ไม่ว่าจะเป็นการออกแบบฟิลเตอร์แบบใดก็ตามวงจรกรองความถี่ต่ำแบบ Secound order จะแสดงสมการนี้เสมอ ซึ่งถ้ากำหนดค่าของ w_0 (ความถี่คัทออฟ) และค่าของ Q แล้วก็นำมาคำนวณค่าของ RC ได้ทันที



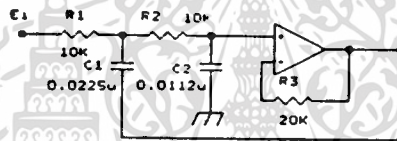
รูปที่ 2.4 Secound order LPF

ในรูปที่ 2.5 เป็นการแสดงตัวอย่างของวงจรกรองความถี่ต่ำของความถี่ 1 kHz โดยมีค่าความชันช่วงคัทออฟ -6 dB ให้ค่า $Q = 0.707$ ค่า $2f$ และ ω_0 เท่ากันที่ -6 dB เมื่อแทนค่าของ $Q = 0.707$ ลงในสมการ ω_0 จะมีค่าเท่ากับ $2\pi \cdot 1 \text{ kHz}$ สมมติเลือก $R_1 = R_2 = 10 \text{ kHz}$ จะได้

$$C = 0.0519 \text{ } \mu\text{F}$$

$$C_1 = 0.0225 \text{ } \mu\text{F}$$

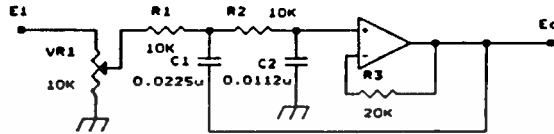
$$C_2 = 0.0112 \text{ } \mu\text{F}$$



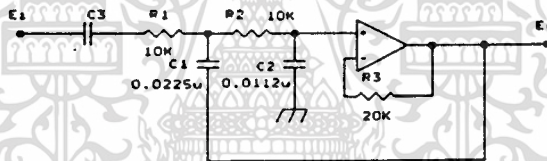
รูปที่ 2.5 การกำหนดค่าต่างๆ ของฟิลเตอร์

ค่าของ R_3 นี้จะเห็นว่าไม่มีความเกี่ยวข้องกับวงจรฟิลเตอร์เลย เป็นเพียงทำหน้าที่ไบอัสกระแสไม่ให้เกิดออฟเซตออกมา ในที่นี้กำหนดให้ $R_3 = R_1 + R_2 = 20 \text{ kHz}$ เราก็จะได้วงจรกรองความถี่ต่ำขนาด 1 kHz ข้อควรระวังสำหรับวงจรนี้คือ เอาท์พุทอิมพีแดนซ์ของช่วงแรกจะต้องต่ำ

ในรูปที่ 2.6 การเพิ่ม VR $10 \text{ k}\Omega$ เข้าไปทางด้านอินพุทจะทำให้ค่า R_1 สูงกว่า $10 \text{ k}\Omega$ ที่กำหนดโดยทันที และถ้าใช้ C ต่อก็คงเหมือนกัน จะทำให้กระแสไบอัสไม่สามารถไหลไปตลอดได้ และ ทำให้กระแสไบอัสอินพุทรวมกับไฟที่มาเลี้ยงออปแอมป์ หรือ กระแสไบอัสไหลไปยัง เพาเวอร์ซัพพลายเลย



(ก) ค่าของ R_1 มากกว่าที่คำนวณไว้



(ข) กระแสไบอัสเข้าจะไม่ไหลออก

รูปที่ 2.6 ข้อควรระวังสำหรับฟิลเตอร์

สำหรับวงจรกรองความถี่สูง ควรคำนึงถึงก็คือ R และ C ให้คิดขงะที่ 10 มีค่าอิมพีแดนซ์เท่ากัน สามารถเปลี่ยนตำแหน่งกันได้กล่าวคือ C_{1H} และ C_{2H} (ในรูปที่ 2.7) สำหรับวงจรกรองความถี่ต่ำที่ใช้ R 10kHz ถ้าใช้ C ต่อแทน R โดยสลับตำแหน่งกันจะได้ค่า $0.016 \mu F$ และ $0.016 \mu F$ ส่วนค่า R_{1H} , R_{2H} ซึ่งแทน C_1 และ C_2 ในวงจรเดิมก็จะได้ $7.07 k\Omega$ และ $14.14 k\Omega$ ตามลำดับ การต่อ R และ C แทนที่กันนี้จะเรียกว่า “การเปลี่ยนความถี่จากวงจรกรองความถี่ต่ำเป็นวงจรกรองความถี่สูง” โดย

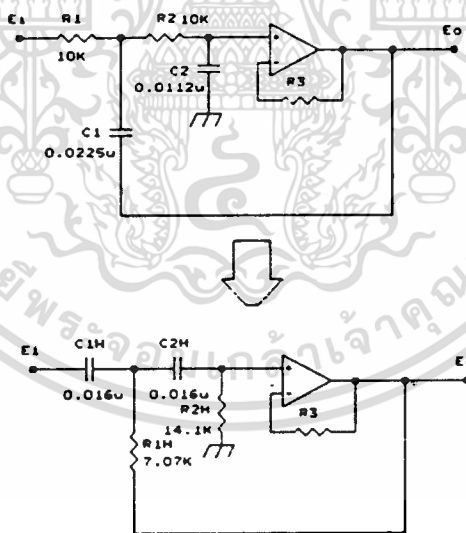
1. เปลี่ยนตำแหน่ง C กับ R
2. ที่ ω_0 มีอิมพีแดนซ์เท่ากัน

$$R_H = \frac{1}{\omega_0 * C_L}$$

$$C_H = \frac{1}{\omega_0 * R_L}$$

$R_H, C_H =$ HPF Constant

$R_L, C_L =$ LPF Constant



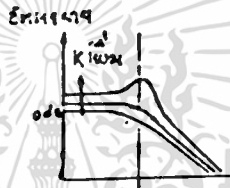
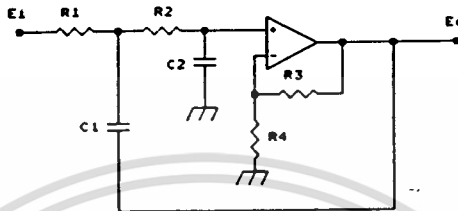
รูปที่ 2.7 การเปลี่ยนจาก LPF เป็น HPF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



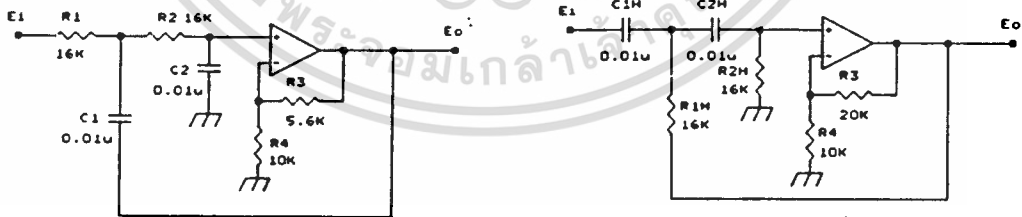
2.3 การเลือกค่าอัตราขยายของออปแอมป์เพื่อกำหนดค่า Q

ในวงจรที่กล่าวมานั้น กำหนดให้ออปแอมป์มีอัตราขยายเป็น 0 dB (ขยาย 1 เท่า) ทำให้การเปลี่ยนแปลงในวงจรขึ้นอยู่กับ C มากกว่าค่า Q ซึ่งไม่ถูกต้องตามความจริง



$$K = (R_1 + R_2) / R_1$$

รูปที่ 2.8 การใช้ RC เดียวกัน



(ก) LPF

(ข) HPF

รูปที่ 2.9 แอคทีฟฟิลเตอร์ (คัทออฟเท่ากับ 1 KHz)

ในรูปที่ 2.8 ออปแอมป์มีค่าอัตราขยายเท่ากับ K ค่า R และ C มีค่าเหมือนกับวงจรเดิม ตอนนี้ค่า Q จะเปลี่ยนไปจากเดิมซึ่งความสัมพันธ์ของ Q และอัตราขยายของออปแอมป์ (K) จะได้ว่าดังนี้

$$Q = \frac{1}{3 - K}$$

$$\text{หรือ } K = 3 - \frac{1}{Q}$$

ค่าของ K จะอยู่ระหว่าง 1 - 3 ส่วนค่าของ Q จะอยู่ในช่วง 0.5 ถึงอนันต์ (infinity) เปรียบเทียบช่วงเปลี่ยนแปลงของ K กับ Q แล้ว Q จะเปลี่ยนช่วงกว้างมาก ริงเป็นจุดเสี้ยวของวงจรนี้ ทำให้ต้องใช้ R ที่มีความผิดพลาดน้อยมาก ๆ ในวงจรถ้า $Q = 0.707$, $K = 1.58$ C จะมีค่าเท่ากันคือ $0.01 \mu\text{F}$ ซึ่งง่ายต่อการออกแบบและการสร้าง แต่ถ้า $K = 1$, $Q = 0.5$, f_0 จะมีค่า -6 dB สมมุติว่าลองออกแบบวงจรกรองความถี่ต่ำขนาด 1 kHz ใหม่ ในรูปที่ 2.9 โดยใช้ R ที่มีค่า 16 k

$$C = \frac{1}{2\pi * f_0 * R}$$

$$= 0.01 \mu\text{F}$$

$$K = 3 - \frac{1}{0.707}$$

$$= 1.58$$

$$\text{ใช้ } R_4 = 10 \text{ k}$$

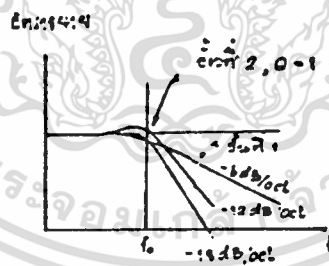
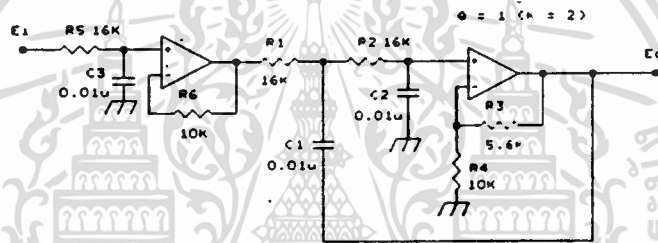
$$(10 \text{ k} + R_3) / 10 \text{ k} = 1.58$$

ค่า $R_3 = 5.8 \text{ k}$ แต่ค่า $5.8 \text{ k} + 5\%$ ไม่มี จึงต้องใช้ค่า 5.6 k แทน ทำให้ค่า $Q = 0.69$ สูงขึ้นกว่าเดิม แต่ก็ไม่มีผลมากนักในทางปฏิบัติ ที่นี้ถ้าลองเปลี่ยนจาก LPF เป็น HPF คดขยเปลี่ยนตำแหน่งของ R และ C จะให้ผลแตกต่างกันหลายๆ แบบ ริงขึ้นอยู่กับค่าของ Q นั้นเอง ดัง

ตัวอย่างต่อไปนี้จะวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

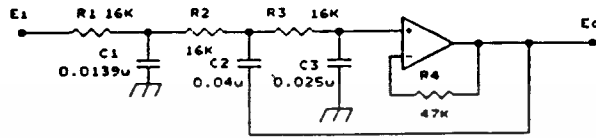
- 2 ชั้น : $Q1 = 0.707$
- 3 ชั้น : $Q1 = 1.000$
- 4 ชั้น : $Q1 = 0.541, Q2 = 1.306$
- 5 ชั้น : $Q1 = 0.618, Q2 = 1.618$
- 6 ชั้น : $Q1 = 0.517, Q2 = 0.707, Q3 = 1.932$
- 7 ชั้น : $Q1 = 0.556, Q2 = 0.802, Q3 = 2.247$
- 8 ชั้น : $Q1 = 0.510, Q2 = 0.601, Q3 = 0.900, Q4 = 2.563$

จะเห็นได้ว่า บางชั้น Q จะมีได้หลายค่า ถ้าเราออกแบบให้มีค่า Q เท่ากับที่แสดงแล้ว ก็จะได้ฟิลเตอร์ที่มีคุณสมบัติตามที่ต้องการ

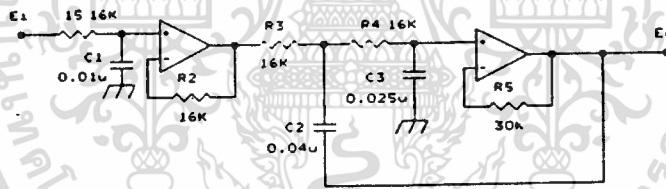


รูปที่ 2.10 ฟิลเตอร์แบบ 3rd order

รูปที่ 2.10 เป็นวงจร 3 ชั้น ชั้นที่ 1 เป็น RC พาสซีฟฟิลเตอร์ ไม่คิดค่าของ Q แยกที่ฟิลเตอร์เกิดจากชั้นที่ 2 ขึ้นไป ในชั้นที่ 1 จะได้ที่ -6 dB ชั้นที่ 2 ถ้าค่าของ Q ไม่เท่ากับ 1.0 จะไม่เป็น -6 dB จึงต้องให้ $Q = 1*(K + 2)$



(ก) คำนวณแบบรวม



(ข) คำนวณแบบแยก

รูปที่ 2.11 แสดงวงจรกรองความถี่ต่ำกำลัง 3

รูปที่ 2.11 แสดงถึง third order ของวงจรกรองความถี่ต่ำ แบบชนิดใช้โอปแอมป์ตัวเดียว (ในรูป ก.) และใช้โอปแอมป์ 2 ตัว (ในรูป ข.) ซึ่งในรูป ก. การใช้โอปแอมป์ตัวเดียวจะเห็นได้ว่า ค่า C ที่ใช้ต้องละเอียดมาก เนื่องจาก C มีผลต่อวงจรกันและกัน ส่วนในรูป ข. นั้น แบ่งเป็น 2 ตอน ตอนแรก ($f_0 = 1\text{kHz}$) ,LPF และตอนที่ 2 ($f_0 = 1\text{kHz}$), $Q = 1$, LPF ซึ่งออกแบบง่ายกว่าแบบแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 คุณสมบัติของฟิลเตอร์แบบต่างๆ

ในการออกแบบแอกทีฟฟิลเตอร์สามารถแบ่งชนิดของฟิลเตอร์ได้เป็น 3 ชนิด คือ บัตเตอร์เวิร์ทฟิลเตอร์ , เบสเซลฟิลเตอร์ และ เชฟบีเชฟฟิลเตอร์ โดยสมการ ของฟิลเตอร์ทั้ง 3 แบบจะอยู่ในรูป

$$A(s) = \frac{K}{H(s)}$$

เมื่อ $A(s)$ เป็นฟังก์ชันอัตราขยายของฟิลเตอร์ลเตอร์
 $H(s)$ เป็นฟังก์ชันของฟิลเตอร์แบบต่างๆ
 K เป็นค่าคงที่

บัตเตอร์เวิร์ทฟิลเตอร์

เป็นฟิลเตอร์ที่ช่วงความถี่ผ่านจะมีความเรียบของสัญญาณมากที่สุด และในช่วงความถี่หยุดจะเท่ากับ $6n$ dB/octave และสมการของบัตเตอร์เวิร์ทฟิลเตอร์นี้สามารถหาได้จากตารางที่ 1. โดยมีค่า K เป็น 1

n	H(s)
1	$S+1$
2	$S^2+1.41S+1$
3	$(S^2+S+1)(S+1)$
4	$(S^2+0.76537S+1)(S^2+1.84776S+1)$
5	$(S^2+0.61803S+1)(S^2+1.61803S+1)(S+1)$

ตารางที่ 1. แสดงฟังก์ชันของบัตเตอร์เวิร์ท เมื่อ n เป็น order ที่ต้องการ

เบสเสลฟิลเตอร์

เป็นฟิลเตอร์ที่มีผลตอบสนองต่อฟังก์ชันหนึ่งหน่วยดีที่สุดและมีการหน่วงเวลาในช่วงเปลี่ยนช่วงความถี่จากช่วงความถี่ผ่านเป็นช่วงความถี่หยุดน้อยที่สุด และสมการของเบสเสลฟิลเตอร์แสดงไว้ในตารางที่ 2.

n	H(S)	K
1	$S+1$	1
2	S^2+3S+1	3
3	$(S^2+3.67782S+6.45944)(S+2.32219)$	15
4	$(S^2+5.79242S+9.14013)(S^2+4.20758S+11.4878)$	105
5	$(S^2+6.7039S+14.2725)(S^2+4.64934S+18.15631)(S+3.64674)$	945

ตารางที่ 2. แสดงฟังก์ชันของ เบสเสลเมื่อ n เป็น order ที่ต้องการ

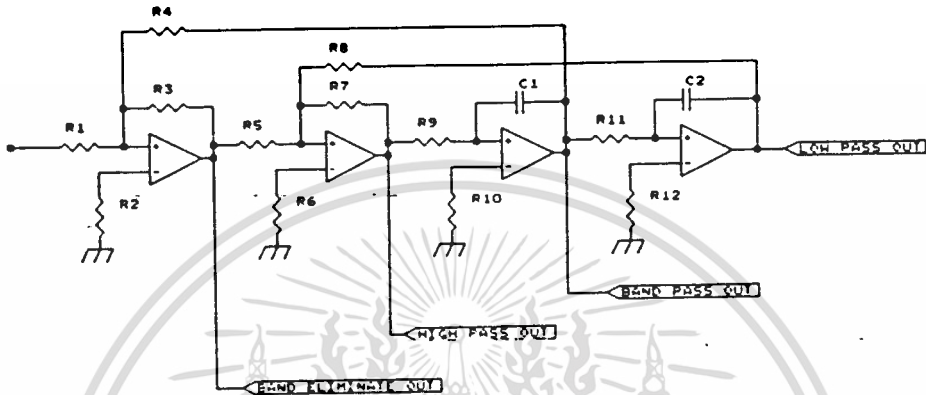
เชฟบีเชฟฟิลเตอร์

เป็นฟังก์ชันที่มีความชันในช่วงความถี่หยุดมากที่สุดเมื่อมี order เท่ากัน แต่ในช่วงความถี่ผ่านนั้นจะมี ripple เกิดขึ้น โดย n มีค่ามากขึ้น ripple ก็จะมีค่าเพิ่มขึ้นด้วย และสมการของเชฟบีเชฟฟิลเตอร์นี้จะขึ้นอยู่กับค่าอัตราขยายในช่วงความถี่ผ่าน ดังนั้นสมการจึงไม่ได้นำมาแสดง

Bi-Quadratic Filter (BQF)

เป็นวงจรที่มีคุณสมบัติพิเศษคือ สามารถให้คุณสมบัติของฟิลเตอร์ทั้ง 4 แบบในวงจรเดียวกัน ดังรูปที่ 2.12 อีกทั้งให้ค่า w_0 และ Q ของแต่ละส่วนไม่ขึ้นแก่กันและกัน ทำให้การคำนวณง่าย ในรูปที่ 2.12. นี้ กำหนดค่า $Q = R_3 / R_2$ และอัตราขยาย $= R_2 / R_3$ แต่ข้อจำกัดของ BQF นี้คือ ไม่สามารถใช้ HPF และ LPF พร้อมกับ BPF และ BEF ในเวลาเดียวกัน เนื่องจาก BPF นั้นค่าอัตราขยายขึ้นกับ Q

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 แสดงวงจร Bi-Quadratic Filter

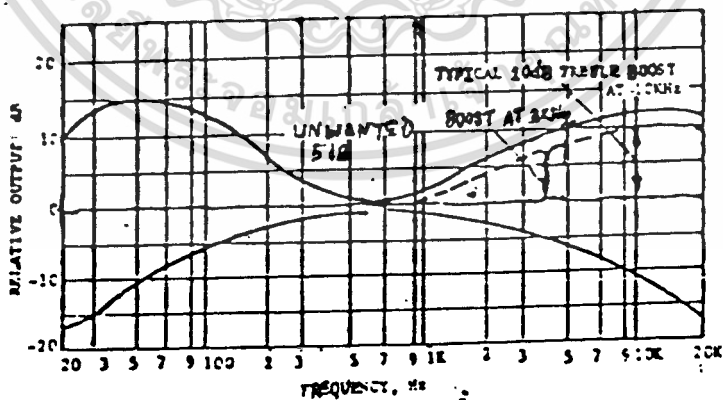
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ชนิดของอีควอไลเซอร์

ในระบบเครื่องเสียงอุปกรณ์ทุกชนิดมีความสมบูรณ์แล้ว เราไม่ต้องอาศัยโทนคอนโทรลเลย แต่เป็นการยากมากที่ระบบเครื่องเสียงจะสมบูรณ์ได้ ถึงแม้ระบบจะสมบูรณ์จริง แต่โอกาสที่จะพบกับการบันทึกแผ่นเสียงไม่ได้มาตรฐานก็ยังมีอยู่ ดังนั้นอย่างไรก็ตามก็ยังต้องใช้ โทนคอนโทรล อยู่ดี

ชนิดของ โทนคอนโทรล แบบง่ายๆ ที่พบกันอยู่ทั่วไป จะมีปุ่มปรับ หูม-แหลม หรือถ้าดีหน่อยก็อาจจะมีเสียงกลางมาด้วย นั่นคือ การบูสต์ (Boost) และ การคัท (Cut) มีผลกระทบกับความดังของความถี่ใกล้เคียงมากเกินไป เช่น ถ้ารู้สึกว่ลำโพงให้เสียงแหลมไม่ดี ท่านอาจจะบูสต์เสียงแหลมขึ้นมา ผลพลอยได้ที่นึกไม่ถึงคือ ท่านได้บูสต์เสียงกลางขึ้นมาด้วยอีกมากทีเดียว ดังตัวอย่างกราฟในรูปที่ 3.1



รูปที่ 3.1 แสดงกราฟตอบสนองความถี่ของ Tone Control

รูปที่ 3.1 ผลของการ บูสท์ และ คัท ความถี่ของ โทนคอนโทรล ทั่วไปจะเห็นว่าเสียงแหลม (ความถี่ประมาณ 10 kHz) จะถูกยกขึ้นมาได้ 10 dB แต่เสียงกลาง (ความถี่ประมาณ 4 kHz) จะถูกยกขึ้นมาถึง 5 dB อันนี้เองเป็นสิ่งที่ไม่ต้องการ แต่ถ้าใช้อีควอไลเซอร์ มาแทน โทนคอนโทรล ซึ่งตัวอีควอไลเซอร์สามารถชดเชยความถี่ได้ละเอียดคั่งนั้นจึงสามารถบูสท์ หรือ คัทความถี่ได้ในช่วงแคบๆ ทำให้สามารถปรับความสมดุลของเสียงได้ใกล้เคียงความต้องการมากขึ้น

อีควอไลเซอร์ในปัจจุบัน มี 3 แบบคือ

1. แบบกราฟฟิเคิลอีควอไลเซอร์ (Graphics Equalizer)
2. แบบพารามेटริกอีควอไลเซอร์ (Parametric Equalizer)
3. แบบพารากราฟฟิเคิลอีควอไลเซอร์ (Pargraphics Equalizer)

1. แบบกราฟฟิเคิลอีควอไลเซอร์ (Graphics Equalizer) จะมีความถี่ศูนย์กลาง (Center frequency) และ แถบความถี่ (Bandwidth) เป็นแบบตายตัว

กราฟฟิเคิลอีควอไลเซอร์ จะแบ่งสเปกตรัมของความถี่ออกตามจำนวนแบนด์ ซึ่งแต่ละแบนด์จะมีการบูสท์ และ คัท เฉพาะเป็นอิสระแก่กัน ในวงจรแบบ 5 แบนด์ หรือ 2 ออกเตป จะมีค่า Q ต่ำ โดย

$$Q = \frac{WD}{BW}$$

WD = ความถี่เชิงมุมตรงกลาง หรือ Enter Angular Frequency

BW = แบนด์วิดท์

โดยจะนับจากตำแหน่งที่อัตราขยายลดลง 3 dB ทั้งสองข้างของ WD จะเป็นพาสแบนด์ ($WD = 2f_0$) เมื่อค่า Q ต่ำความกว้างของแบนด์จะกว้าง กราฟฟิเคิลอีควอไลเซอร์โดยทั่วไปมักจะมีค่าตรงกลางที่ถูกกำหนดไว้คงที่แล้ว การตอบสนองความถี่ของกราฟฟิเคิลอีควอไลเซอร์จะขึ้นเป็นลูกๆ ตลอดเวลาที่อ่านความถี่ จึงได้ชื่อเรียกว่า Pedkind type Equalizer

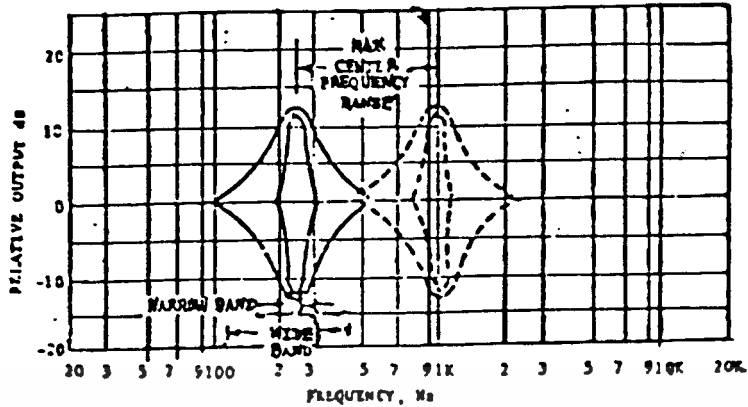
ความถี่ศูนย์กลางในตัวอิกวอไลเซอร์ หมายถึง ความถี่ของสัญญาณเสียงที่เราสามารถจะเร่งหรือลดระดับความดัง แถบความถี่ หมายถึง ช่วงห่างของความถี่ศูนย์กลางแต่ละตัว ในการกำหนดความกว้างของแถบความถี่ส่วนความถี่ศูนย์กลางนั้น ควรจะครอบคลุมช่วงความถี่เสียงที่มนุษย์สามารถได้ยิน คือ 20 Hz ถึง 20 kHz หรือมากกว่านี้เล็กน้อย

ออกเตปแบนด์ (Octave Band) หมายถึง ความกว้างของแถบความถี่ซึ่งทำให้ความถี่ศูนย์กลาง เป็น 2 เท่า (2^1) ของความถี่ศูนย์กลางที่ต่ำกว่าถัดไป ดังนั้น 2 ออกเตปแบนด์ จึงหมายถึง ความถี่ศูนย์กลางที่ต่ำกว่าตัวถัดไป นั่นคือตัวเลขหน้าคำว่า ออกเตปแบนด์ยิ่งมากก็ยิ่งทำให้ความถี่ศูนย์กลางยิ่งห่างกันมากขึ้น การปรับแต่งจึงไม่ละเอียดพอ ดังนั้นกราฟฟิควอไลเซอร์ ควรมีความถี่ ตั้งแต่ $1/3$ ออกเตปแบนด์ จนถึง $1/6$ ออกเตปแบนด์ จะเหมาะสมที่สุด

2. แบบพารามตริกอิควอไลเซอร์ (Parametric Equalizer) เป็นแบบที่เลือกความถี่ศูนย์กลางได้มาก และเปลี่ยนความถี่ให้ละเอียดมากขึ้น หรือ น้อยเพียงใดก็ได้ เท่าที่มีจำหน่ายมีตั้งแต่ 2 แบนด์ ถึง 4 แบนด์ ตัวเลขหน้าคำว่าแบนด์ คือ จำนวนแถบศูนย์กลางความถี่ที่เราจะเลือกปรับได้อย่างอิสระ โดยอาศัยปุ่มปรับเสียงทึบแหลมของชุดขยายเครื่องเสียงมาปรับความถี่ช่วงอื่นๆ ได้ด้วย ทำให้แก้ปัญหาความก้องกังวาน หรือลักษณะทับทางด้านเสียงของห้องฟังเพลงได้เป็นอย่างดี

พารามตริกอิควอไลเซอร์ จะต่างกับ กราฟฟิควอไลเซอร์ คือ ความถี่ตรงกลางและความกว้างของแบนด์ หรือค่า Q สามารถปรับหรือเปลี่ยนได้ ดังนั้นการที่เราที่เราจะบูสท์หรือคัทที่ความถี่เท่าไรเป็นตัวแปรอิสระแก่กัน จึงทำให้พารามตริกอิควอไลเซอร์ สามารถช่วยได้มากในเรื่องการปรับแต่งเสียง การแสดง หรือ การอัดเสียงทำมาสเตอร์เทปอีกด้วย

พารามตริกอิควอไลเซอร์ มักใช้ในระบบเสียงอาชีพมากกว่า ถึงแม้การแบ่งช่วงความถี่จะน้อยกว่าแบบกราฟฟิควอไลเซอร์ก็ตาม แต่ในแต่ละช่วงความถี่สามารถตั้งความถี่ในการ บูสท์หรือ คัท ได้ตามความพอใจได้ดังรูปที่ 3.2



รูปที่ 3.2 แสดงการตั้งความถี่และการเปลี่ยน Bandwidth Parametric Equalizer

ในความเป็นจริงแล้ว ระบบเสียงต้องการการชดเชยจุดบกพร่องของความถี่เสียงไม่ที่จุดตั้งนั้น พารามตริกอีควอลไลเซอร์จะให้ความสะดวกและตอบสนองความต้องการได้ตรงจุดมากกว่าแบบกราฟฟิคอีควอลไลเซอร์ นอกจากตั้งความถี่ได้แล้ว ยังสามารถกำหนดความกว้างของความถี่ที่กำลัง บูสท์ หรือ คัท ได้ แต่ถ้ากำหนด แบนด์วิดท์กว้าง ความถี่ใกล้เคียงก็จะถูก บูสท์ หรือ คัทตามไปด้วย ซึ่งการเลือกความกว้างนี้ขึ้นอยู่กับผู้ใช้ต้องการอย่างไร

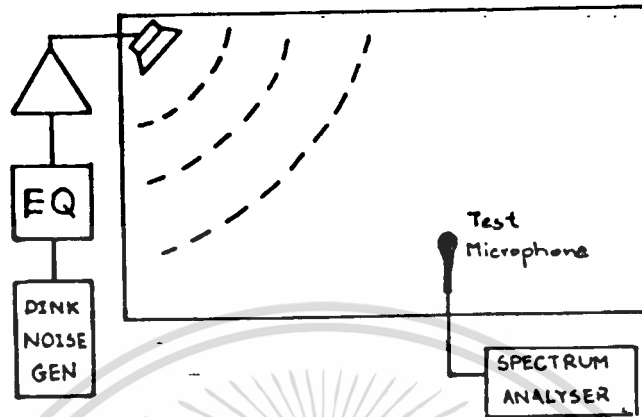
3. แบบพารากราฟฟิคอีควอลไลเซอร์ (Parametric Equalizer) เป็นการผสมระหว่างพารามตริกอีควอลไลเซอร์ กับ กราฟฟิคอีควอลไลเซอร์ คือเราสามารถเลือกความถี่ศูนย์กลาง และแถบความถี่ได้อย่างอิสระเช่นกัน แต่จะเป็นช่วงแคบๆ ไม่กว้างแบบพารามตริกอีควอลไลเซอร์ จึงนับว่าเป็นแบบที่เหมาะสมกับการใช้งานมากที่สุด

3.1 ประโยชน์ของอีควอลไลเซอร์

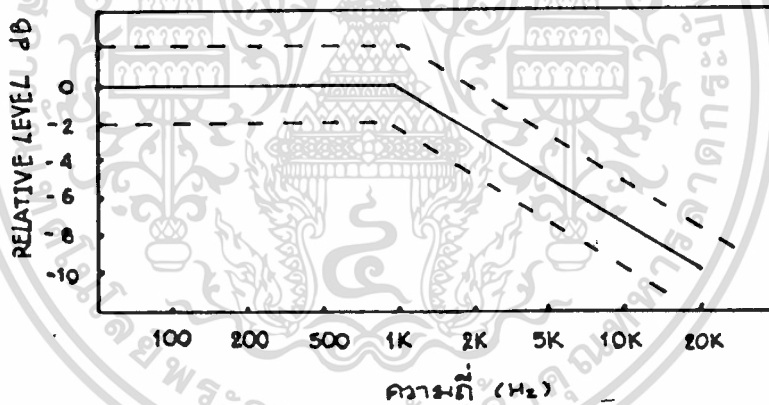
อีควอลไลเซอร์เป็นอุปกรณ์ที่ใช้แก้ไขข้อบกพร่องของเสียงซึ่งไม่ค่อยราบเรียบซึ่งเกิดจากลำโพง หรือ อะคูสติกของห้อง หรือภาคขยายเสียงซึ่งสามารถปรับแต่งได้โดย

(1) ใช้หูในการปรับแต่ง ซึ่งจะยึดเสียงที่ได้ยินจากหูฟังเป็นเกณฑ์ในการปรับซึ่งในความเป็นจริงเสียงที่ได้ยินจากหูฟังเป็นเสียงตรง (direct) แต่เสียงที่ได้ยินจากห้องเป็นเสียงตรงเสียงก้อง และเสียงสะท้อน ซึ่งจะทำให้เสียงที่ได้ยินแตกต่างกันออกไปตามลักษณะของห้องที่ใช้ในการฟัง ซึ่งนอกจากนี้ลำโพงและภาคขยาย ยังมีผลต่อเสียงที่ออกมาด้วยซึ่งขึ้นอยู่กับค่าผลตอบสนอง (response) ของลำโพงและภาคขยาย

(2) ใช้เครื่องมือช่วยในการปรับคังรูป



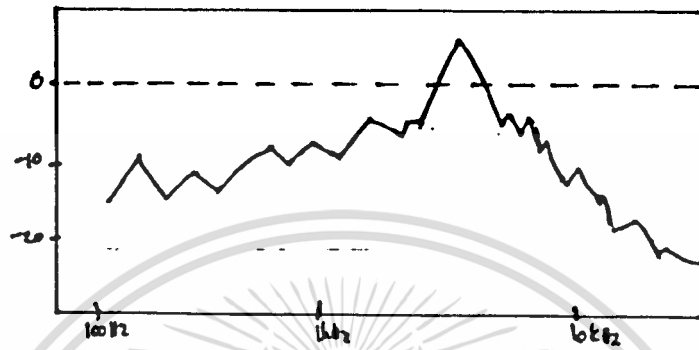
รูปที่ 3.3 แสดงวิธีการวัด



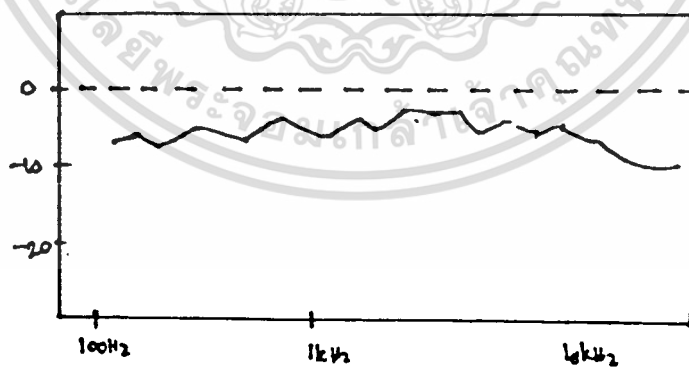
รูปที่ 3.4 แสดงเกณฑ์ในการปรับอีควอไลเซอร์

ซึ่งการปรับจะเริ่มจากมีแหล่งกำเนิดสัญญาณรบกวน (pink noise) ป้อนให้กับอีควอไลเซอร์และผ่านภาคขยายลำโพงซึ่งตั้งอยู่ในห้องและเราก็ใช้ไมค์เป็นตัวรับสัญญาณเสียงเหมือนหูคน ซึ่งจะมี spectrum analyzer เป็นตัววัดระดับสัญญาณเสียงออกมาที่ความถี่มูลฐาน(fundamental) ต่างๆ ซึ่งทำให้เห็นระดับสัญญาณชัดเจนแน่นอนมากกว่าการฟังโดยใช้หูในการปรับนี้เราจะปรับโดยให้ flat ที่ 0 dB แต่ในความเป็นจริงเกรทซ์ของเสียงที่ต้องการจะใช้เกณฑ์ตามรูป 2 โดยปรับให้ความถี่คัตออฟ(frequency cutoff) ที่ความถี่ระหว่าง 2 KHz ถึง 4 KHz จากการปรับจริงเราจะเห็นกราฟเป็นดังรูป 3.5 - 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 กราฟ response ก่อนปรับอควอไลเซอร์



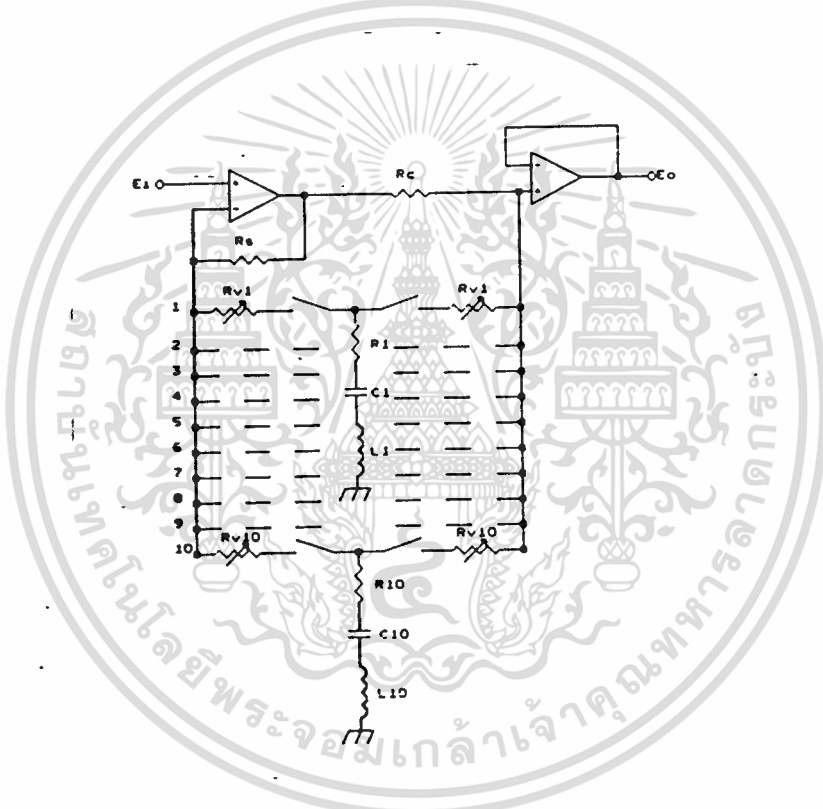
รูปที่ 3.6 กราฟ response หลังการปรับอควอไลเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การออกแบบอิกวอไลเซอร์

พื้นฐานอิกวอไลเซอร์แสดงไว้ในรูป 4.1 ที่มีออปแอมป์อยู่ ส่วนในทางเดินสัญญาณและมีสวิทช์ในการเลือกช่วง บูลท์ และ คัท ออปแอมป์ตัวแรกจะเป็นการ บูลท์ และออปแอมป์ตัวที่สองจะเป็นbufferไว้ คัทสำหรับความถี่เพียง ช่อง จะมีทั้ง บูลท์ และ คัทแต่จะไม่เลือกในเวลาเดียวกัน ดังนั้นจะแยกโดย switch exclusive ที่จูนวงจรว่าจะ บูลท์ หรือ คัท



รูปที่ 4.1 พื้นฐานของอิกวอไลเซอร์

$$\text{(Boost)} \quad GAIN(A_w) = \frac{R_v + R_{i1} + R_v}{R_v + R_0}$$

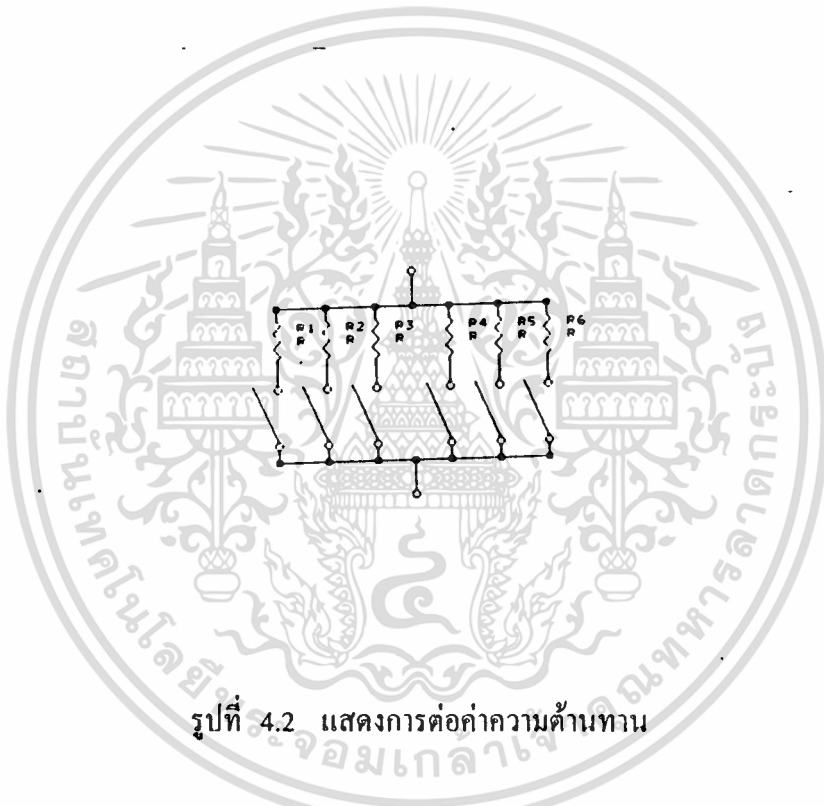
$$\text{(Cut)} \quad GAIN(A_w) = \frac{-R_v + R_0}{R_v + R_0 + R_c}$$

$$F_i = \frac{1}{2\pi * \sqrt{L_i * C_i}}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วน บูลท์ และ คัท จะถูกควบคุมโดยความต้านทานปรับค่าได้ $R_{v1} - R_{v10}$ ซึ่งได้ออกแบบตามอัตราส่วน R_x, R_c ให้ได้ 1 dB

ความต้านทานที่ปรับได้นี้แสดงในรูปที่ 1.2 ซึ่งมีความต้านทานที่ใช้ในการกำหนด GAIN อยู่ 6 ตัว ต่อขนานกันดังรูปซึ่งมีสวิตช์ใช้เลือก GAIN ดังรูป รายละเอียดในรูปที่ 4.2 โดยจะเลือกค่าความต้านทาน 6 ค่าที่เหมาะสมมาต่อรวมกันให้เลือกได้ 12 step ส่วน code การเรียงลำดับได้อธิบายไว้ในส่วนของโปรแกรม



รูปที่ 4.2 แสดงการต่อค่าความต้านทาน

คุณสมบัติทางความถี่และช่วงกว้างของความถี่ (Bandwidth) จะถูกตั้งโดย network ในช่วงรีโซแนนซ์ส่วนของ LC จะมีอิมพีแดนซ์ลดลงจนถึง 0 ในการ บูลท์ออปแอมป์จะถูกตั้ง GAIN ด้วยอัตราส่วน R_b และ $R_o + R_v$ ในทางตรงกันข้ามในส่วนคัทที่ออปแอมป์ เป็น Buffer ที่ประกอบด้วย R_c, R_d และ R_v ในวงจรจูน LC เมื่อปรากฏออกในรูปอิมพีแดนซ์สูงและ GAIN ลดลงเป็น 1

เนื่องจาก คุณสมบัติของอิกวอไลเซอร์หาได้โดย LC network โดยนักออกแบบจะค่อยปรับปรุงวงจรอิกวอไลเซอร์ให้เหมาะสมตามความต้องการอยู่เสมอ

4.1 การเลือกช่วงความถี่

ทางพื้นฐานของอิกวอไลเซอร์แบ่งเป็น 2 ชนิด

ชนิดที่ 1 คือ การเลือกช่องความถี่ไว้คงที่ตลอด

ชนิดที่ 2 คือ เป็นการปรับช่วงความถี่ ซึ่ง สามารถปรับปรุงให้ครอบคลุมไปยังช่องต่างๆ

ในการเลือกความถี่ก็แบ่งได้เป็น 2 วิธี คือ

1. ขยายแบนด์ให้สม่ำเสมอกับย่านความถี่ที่ต้องการ
2. เว้นแบนด์ให้ติดกับย่านความถี่ที่ควบคุม และเว้นให้กับความถี่อื่น

ตัวอย่างของแบนด์ที่ 2 นี้ คือ Modified ซึ่งจะเว้นแบนด์ไว้เยอะมาก สำหรับความถี่ต่ำซึ่งมีใช้งานกันมาก ส่วนช่วงที่เหลือจะเว้นไว้มากกว่าความกว้างที่ความถี่สูง (>500Hz) ในเวลาเดียวกัน ช่วงความถี่จะถูกเลือกวงจร Q และ ค่าการคำนวณของวงจร serie tune รวมถึงสูตรการใช้ ออกแบบอิกวอไลเซอร์

การเว้นช่วงความถี่ให้คงที่ คือ การวัดโดยทั่วไปในส่วนของ octave ซึ่ง octave จะควบคุมความถี่ ในอัตราส่วน 2:1 ในช่วงความถี่ระหว่าง 1000-2000Hz จำนวนของ octave ที่อยู่ภายในระหว่างความถี่ 2 ความถี่จะหาได้จากสมการ

$$\text{octave} = \log(F_2 / F_L / \log(2))$$

$$F_2 > F_L$$

อีกอย่างหนึ่งที่ใช้ในการออกแบบอิกวอไลเซอร์ซึ่งหาจากเสียงกลางความถี่ 2 ความถี่ คือ

$$\text{ความถี่กลาง} = \sqrt{\frac{F_2}{F_L}}$$

โดยเราอาจพิจารณา 2 ความถี่ 200 Hz และ 440 Hz จะอยู่ห่าง 1 octave ถ้าดูดีๆจะปรากฏเป็น 300 Hz อยู่ระหว่างความถี่ทั้งสอง แต่จะไกลกว่าที่หูเราจะรับได้ 311 Hz จะมีระยะเท่ากันจาก 220 Hz และ 440 Hz เพราะหูเราปรับได้ตลอดเวลา

4.2 Band selection

ในตัวอย่าง 7 band equalizer ซึ่งมีความถี่ 1 KHz เป็นความถี่กลางมี factor เท่ากับ 2.5 ถ้าในความถี่ 1 KHz มีซ้ำกันหลายช่วง จะแบ่งโดย 2.5 ถ้าความถี่อื่นจะกำหนดมา

สรุปว่าถ้าเราต้องการจะควบคุมย่านให้ชัดเจน จะต้องหาความถี่จากสูตร

$$\text{center band} = \sqrt{F_{\max} * F_{\min}}$$

และหา factor จาก

$$\text{factor} = \frac{F_{\max} * \sqrt{F_{\min}}}{2}$$

F_{\min} คือ ค่า -3dB ที่แบนด์ต่ำสุดเมื่อบูสท์เต็มที่แล้วแบนด์อื่นจะอยู่สภาวะ flat

F_{\max} มีค่า -3dB ที่ความถี่สูงสุด

และ จากสูตรความถี่รวมกันได้

$$F_n = \frac{F_{\max} * (2n - 1)}{2d} * \frac{F_{\min} + (2(d - n) + 1)}{2d}$$

เมื่อ n คือจำนวนแบนด์ (1 ถึง d)

อาจจะเป็นเรื่องเข้าใจผิดที่ช่องความถี่ของอิกวอลไลเซอร์เกี่ยวข้องกับความถี่ของการจัดซึ่งไม่ใช่เรื่องที่ต้องช่อง flat ความถี่ที่สมบูรณ์จะขึ้นอยู่กับ แบนด์ความถี่อิกวอลไลเซอร์ ส่วนมาก จะออกแบบให้มีการขยายได้กว้างออกไปเพื่อการพิจารณาที่ควบคุมได้ที่ความถี่ต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การเลือกค่า Q สูงสุด

ถ้าต้องการหาค่า Q สูงสุดของแต่ละแบนด์เมื่อเกิด full boost หรือ full cut และเช็คค่าของ $R_o + R_{vt}$ และ L_o ค่า Q_{max} จะเป็นฟังก์ชันของช่องความถี่ที่ต่อกัน

$$Q_{max} = \frac{\sqrt{F_2}}{\sqrt{F_3} - \sqrt{F_1}}$$

เมื่อ $Q_{f|max}$ คือ ค่า Q สูงสุดของ F_2 | ระหว่าง fullcut หรือ บูลสต์ และ F_3 | F_1 เป็นแบนด์ความถี่ที่อยู่ใกล้กันช่องสูงสุด และ ค่าสุดบนอิกวอลเซอร์จะมีเพียงแบนด์ที่เป็นแบนด์ประชิด

$$Q_{max} = \frac{\sqrt{F_1 * F_2}}{F_3 - F_1}$$

เมื่อ F_1 คือ มุมประชิดและในเทอมของ factor

$$Q_{max} = \frac{\sqrt{factor}}{factor - 1}$$

ในเทอมของ F_{min} และ F_{max}

$$Q_{max} = \frac{2d\sqrt{F_{max} * F_{min}}}{d\sqrt{F_{max} * (F_{min} - 1)}}$$

การหา Q_{max} อีกอย่างคือจากค่า L และ C ที่คำนวณจาก

$$L_n = 2270 * Q_{max} * \sqrt{w_n}$$

$$C_n = \sqrt{w_2 * L_n}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยค่า 2270 คือค่าความต้านทานต่ำสุดที่รวมเอา ความต้านทาน 680 ความต้านทานสวิทช์ และความต้าน R_0

4.4 การออกแบบค่า Inductor โดยใช้วงจร Gytrators

ในทางปฏิบัติวงจร LC network เราจะไม่สามารถจะ หาตัว L มาใช้ได้โดยธรรมดา เราจึงมีการใช้ วงจร Gytrators ซึ่งจะแสดงคุณสมบัติของ inductor ซึ่งวงจรจะประกอบด้วย ออปแอมป์ และตัวเก็บประจุจะหาง่ายกว่า inductor ซึ่งวงจร Gytrators สามารถเขียนเป็นสัญลักษณ์ของ พอร์ทได้ดังรูป



รูปที่ 4.3 แสดง Gytrator a) สัญลักษณ์ที่ใช้งาน

b) มีloadต่อที่ขั้ว 2' 2

จากรูปเราสามารถเขียน matrix ของวงจร two port

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} 0 & -K \\ K & 0 \end{bmatrix} * \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

จากสมการข้างบนเราหา Z_{in}

$$Z_{in} = \frac{K^2}{Z_1}$$

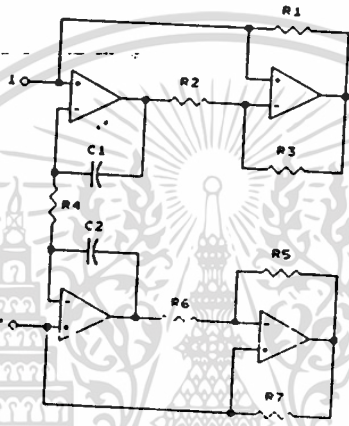
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราจะให้ Z_2 แทนตัวเก็บประจุ C อุปกรณ์ตัวอื่นคือความต้านทาน R

$$Z_{in} = sR^2C$$

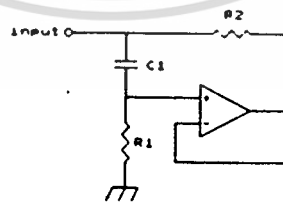
$$L = R^2C$$

เขียนวงจรโดยสมบูรณ์เวลาเอามาใช้งานได้ดังรูปที่ 4.5



รูป 4.5 แสดงการเอาวงจรไปใช้งาน

จากข้างบน เราจะเห็นว่า สามารถนำออปแอมป์และตัวเก็บประจุมาทำเป็น L ค่ามากได้ โดยเพิ่มค่า K แต่จากวงจรที่ใช้ทำอิกควอลไลเซอร์ใช้งานส่วนใหญ่เป็นดังรูป 4.6



รูปที่ 4.6 แสดง Gyrator ที่ใช้งานจริง

โดยค่าของ L หาได้จาก

$$L = R_1 * R_2 * C_1$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การออกแบบและการสร้าง

ในการออกแบบฮาร์ดแวร์ได้ออกแบบส่วนต่างๆ ที่ใช้เป็นตัวต่อต่อไปนี้

1. ส่วนการ์ดที่ใช้ติดต่อกับฮาร์ดแวร์
2. ส่วนที่ใช้ควบคุมฮาร์ดแวร์
3. ส่วนของฮาร์ดแวร์

ส่วนการ์ดที่ใช้ติดต่อกับฮาร์ดแวร์

บนสล็อตที่ใช้เสียบการ์ดต่างๆ ที่ใช้ในการอินเตอร์เฟสจะประกอบด้วยขาสัญญาณต่างๆ ที่ใช้ในการอินเตอร์เฟส ซึ่งขาสัญญาณนี้มีทั้งหมด 62 ขา ตำแหน่งขาต่างๆ ได้แสดงดังรูปที่ 5.1 และหน้าที่ของขาต่าง ๆ ที่ใช้งานมีดังนี้

- *Reset Drv* ขาสัญญาณนี้เป็นเอาต์พุต จะแอกทีฟในช่วงที่เราเริ่มจ่ายไฟใช้กับระบบ และยังคงแอกทีฟไปจนกระทั่งระบบต่างๆ ภายใน PC พร้อมทั้งจะทำงานได้ โดยทั่วไปสัญญาณนี้จะถูกนำมาใช้ในการรีเซ็ตวงจรอินเตอร์เฟส หรืออุปกรณ์ I/O ต่างๆ

- *A0 - A19* ขาสัญญาณนี้มีทั้งหมด 20 ขา เป็นเอาต์พุต ซึ่งใช้สำหรับกำหนดแอดเดรสของหน่วยความจำหรือ I/O ที่เราต้องการติดต่อกับ โดย A0 จะมีนัยสำคัญต่ำสุด A19 จะมีนัยสำคัญสูงสุด ซึ่งในขบวนการ DMA เขียนข้อมูลลงหน่วยความจำโดยตรงจะใช้ค่าแอดเดรสทั้งหมด 20 เส้น

สำหรับการอ้างตำแหน่งแอดเดรสของ I/O พอร์ต จะใช้สัญญาณเพียง 16 เส้นคือ A0 - A15 ซึ่งอ้างอิงได้เพียง 64K พอร์ตเท่านั้น แต่การใช้งานจริงนั้น I/O พอร์ตจะอ้างอิงเพียง 10 เส้น คือ A0 - A9 เท่านั้น

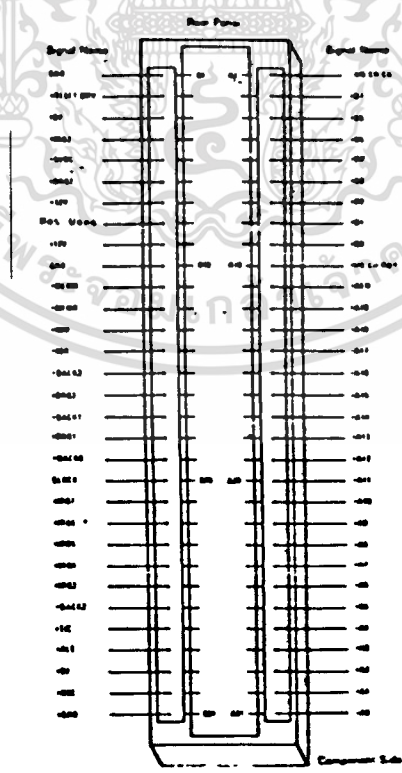
- *D0 - D7* ขาสัญญาณนี้จะแบบ Bi - direction ซึ่งจะต่อกับบัสข้อมูลของระบบ เพื่อทำหน้าที่ในการส่งผ่านข้อมูลระหว่าง I/O พอร์ต กับ PC โดยบิต D0 จะมีนัยสำคัญต่ำสุด D7 มีนัยสำคัญสูงสุด สำหรับในบัสไซเคิลของการเขียนข้อมูล หรือ อ่านลงบัส โดยขึ้นอยู่กับ IOW หรือ MEMW กับ IOR หรือ MEMR

- IOR ขาสัญญาณนี้เป็นเอาต์พุต แอคทีฟที่ลอจิก 0 ที่สร้างขึ้นโดย 8288 Bus Controller เพื่อใช้ในการแสดงว่าบัสไซเคิลของการอ่านข้อมูลจาก I/O พอร์ต เพื่อให้พอร์ท I/O ที่มีแอดเดรสตรงกับแอดเดรสบนบัสนั้นส่งข้อมูลลงบนบัส โดยข้อมูลจะต้องถูกส่งลงบนบัสข้อมูลก่อนขอบขาขึ้นของ IOR

- IOW ขาสัญญาณนี้เป็นเอาต์พุตแอคทีฟที่ลอจิก 0 ซึ่งถูกสร้างขึ้นโดย 8288 เพื่อใช้แสดงว่าบัสไซเคิลที่เกิดขึ้นเป็นบัสไซเคิลของการเขียนข้อมูลลงบนพอร์ท I/O เพื่อให้ I/O รับข้อมูลที่อยู่บนบัสไปเก็บไว้

- AEN ขาสัญญาณนี้เป็นเอาต์พุต ที่ใช้ในการแสดงว่าบัสไซเคิลที่เกิดขึ้นในช่วงเวลาที่สัญญาณ AEN แอคทีฟ เป็นบัสไซเคิลของขบวนการ DMA

สำหรับบนเมนบอร์ดของ PC จะใช้สัญญาณนี้ในการคิสเอเบิล 8288 Bus Controller และจะใช้คิสเอเบิล I/O พอร์ตต่างๆ ที่ไม่เกี่ยวข้องกับขบวนการ DMA ที่จำเป็นต้องทำเช่นนี้ก็เพราะในระหว่างขบวนการ DMA นั้น 8237A-5 จะส่งแอดเดรสของหน่วยความจำบนบัสแอดเดรสและจะทำให้สัญญาณ IOR หรือ IOW แอคทีฟด้วย ดังนั้นถ้าไม่ทำการคิสเอเบิลพอร์ท I/O ที่ไม่เกี่ยวข้องไว้ ก็จะทำให้พอร์ท I/O ที่มีแอดเดรสตรงกับค่าแอดเดรส บนบัสแอดเดรส นั้นทำการอ่านหรือส่งข้อมูลออกมาบนบัสข้อมูลทำให้เกิดความผิดพลาดเกิดขึ้นได้



รูปที่ 5.1 แสดงตำแหน่งของสัญญาณต่างๆ บนสล็อตของ PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานแอดเดรสสำหรับพอร์ต I/O ใน PC

พอร์ต I/O ที่ใช้งานอยู่ใน PC จะมี 1,024 พอร์ต ซึ่งถูกแบ่งออกเป็น 2 กลุ่ม กลุ่มละ 512 พอร์ต

- กลุ่มแรกเป็นกลุ่มของพอร์ต I/O ที่อยู่บนเมนบอร์ด PC จะมีแอดเดรส 0000H-01FFFH หรือ แอดเดรสที่ A9 เป็น 0 สำหรับแอดเดรสในกลุ่มนี้จะถูกใช้อย่างอิงแอดเดรสบนชิพซีพียูและอุปกรณ์ I/O บนเมนบอร์ดของ PC

- กลุ่มที่สองเป็นกลุ่มของพอร์ต I/O ที่ถูกใช้งานอยู่บนการ์ดที่ใช้เสียบบนสล็อตต่างๆ ของ PC สำหรับแอดเดรสเหล่านี้จะเริ่มต้นจากแอดเดรส 0200H-03FFFH ซึ่งก็คือ A9 เป็น 1

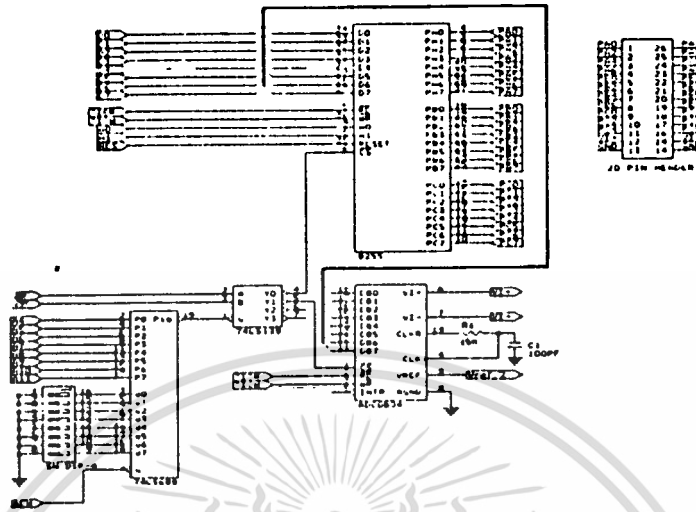
การทำงานพื้นฐาน

การดีโค้ดแอดเดรสของพอร์ตจะทำให้โดยใช้ 74LS688 เป็น IC เปรียบเทียบข้อมูล 8 บิต ในวงจรจะทำการเปรียบเทียบสัญญาณระหว่าง A4-A11 บิต PC กับข้อมูลที่กำหนดโดยดิฟสวิตช์ ถ้าหากข้อมูลทั้ง 2 มีค่าตรงกัน เอาท์พุทที่ขา 19 จะเป็น 0 แล้วไปอินเอาต์ 74LS139 ซึ่งจะถอดรหัส A2 กับ A3 ไว้เลือกในการติดต่อว่าจะติดต่อกับ 8255 หรือ ADC0804 ซึ่งถ้าเป็น 8255 จะทำการควบคุมการส่งข้อมูลออก ส่วน ADC0804 นั้นจะรับสัญญาณอนาล็อก มาทำการเปลี่ยนเป็นสัญญาณดิจิตอล แล้วส่งเข้าไปเก็บไว้ใน PC

ในขบวนการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอลจะใช้ IC ADC0804 เป็นชิพแปลงสัญญาณอนาล็อกเป็นดิจิตอลขนาด 8 บิต โดยตั้งโหมดการทำงานของ ADC ให้อยู่ในโหมด Free Run ซึ่งทำให้สัญญาณ RD, WR และ CS สามารถต่อเข้าควบคุมการทำงาน ส่วนแรงดันอ้างอิงจะเลือกกระหว่าง 2.5 โวลต์และ ค่าแรงดันอ้างอิง/2 จากภายนอก ซึ่งวงจรนี้ได้แสดงไว้ดังรูปที่ 5.2

การเลือกแอดเดรสของการ์ด

ตำแหน่งของพอร์ตบนการ์ดจะใช้ 74LS688 , 74LS139 และ ดิฟสวิตช์ ต่อร่วมกันเป็นวงจรเลือกแอดเดรส โดยดิฟสวิตช์จะเป็นตัวกำหนดหมายเลขแอดเดรสของพอร์ตตามความต้องการของผู้ใช้ จะใช้ตำแหน่งพอร์ตทั้งหมด 22 พอร์ต แต่ที่ใช้งานจริงมีดังนี้



รูปที่ 5.2 แสดงวงจรในส่วนของการ์ดที่ใช้ติดต่อกับอิกวอไลเซอร์

- 300H = พอร์ต PA ของ 8255
- 301H = พอร์ต PB ของ 8255
- 302H = พอร์ต PC ของ 8255
- 303H = พอร์ตควบคุมของ 8255
- 304H = พอร์ตของ ADC0804

ผู้ใช้สามารถกำหนดตำแหน่งของพอร์ตได้โดยกำหนดระดับลอจิกให้ตรงกับแอดเดรสที่ต้องการ โดยถ้าคิฟสวิทซ์ “ ON “ จะได้ระดับลอจิก (1) หาก “ OFF “ คิฟสวิทซ์! จะได้ระดับลอจิกเป็น 0

ส่วนที่ใช้ควบคุมอิกวอไลเซอร์

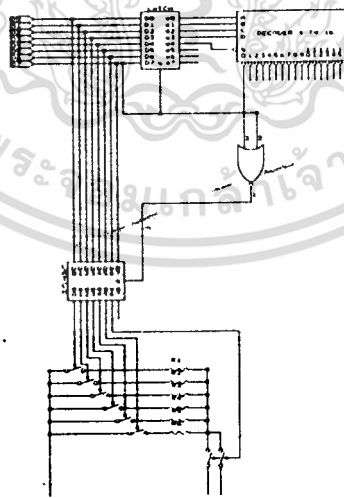
จากการใช้งานอิกวอไลเซอร์โดยทั่วไปจะทำการปรับโดยใช้มือในการปรับความดังของเสียงในเวลาที่ใช้งาน โดยการปรับความต้านทานที่ปรับค่าได้ ซึ่งการทำงานในลักษณะนี้เราไม่สามารถที่จะเก็บค่าที่ใช้ในการปรับในสถานที่ต่างๆ เมื่อเรานำเอาอิกวอไลเซอร์มาใช้งานในสถานที่อื่นๆ กัน ทำให้เราจะต้องใช้เวลาในการปรับใหม่ และการที่จะปรับความถี่ต่างๆ ในได้มาตรฐานนั้นจะใช้เวลามาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การปรับอิกวอไลเซอร์โดยใช้คนนั้นจะเสียเวลามาก ดังนั้น เมื่อนำเอาคอมพิวเตอร์มาควบคุมแทน ทำให้สามารถเก็บพารามิเตอร์ต่างๆ ไว้ได้ และสามารถที่จะนำมาใช้ใหม่ในสถานที่เดิมได้โดยไม่ต้องเสียเวลาปรับใหม่ แล้วยังสามารถปรับค่าพารามิเตอร์ต่างๆ ให้ได้มาตรฐานโดยใช้เวลาน้อยกว่าใช้คนมาปรับมาก ซึ่งในส่วนนี้เราจะต้องทำการสร้างส่วนสร้างสัญญาณ และส่วนวัดค่าพารามิเตอร์ต่างๆ เพื่อป้อนให้แก่คอมพิวเตอร์ทำการคำนวณ และ ปรับให้ได้ตามมาตรฐาน

ในการที่จะนำคอมพิวเตอร์มาติดต่อกับอิกวอไลเซอร์นั้นจะต้องสร้างการ์ดในการติดต่อ และในส่วนของพารามิเตอร์ต่างๆ ก็จะใช้ค่าดิจิทัลในการปรับ ซึ่งในอิกวอไลเซอร์ทั่วไปจะใช้ความดันฐานที่ปรับค่าได้ในการปรับความดัง ซึ่งเป็นค่าทางอนาลอก แต่เราสามารถควบคุมความดังโดยใช้คอมพิวเตอร์ได้ด้วยวงจรดังรูปที่ 5.3

จากรูปที่ 5.4 คือส่วนที่ใช้ควบคุมอัตราขยายของแต่ละแบนด์ โดยนำมาแสดงแค่ส่วนหนึ่ง เพื่อดูการทำงาน จะเห็นว่ามีส่วนสำคัญอยู่ 4 ส่วน คือ Latch , decoder 4 to 16 , Selecter L/R , select Switch



รูปที่ 5.4 แสดงส่วนของ Latch , Decoder, Switch L/R และ Selector Switch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 5.4 จะเห็นว่า เราจะมียังจร Latch 2 ชุด คือ Latch 1 และ Latch 2 ซึ่งใช้สำหรับพักข้อมูล 2 ชุด โดย Latch 1 จะใช้ในการพักข้อมูลที่ใช้ในการเลือกแบนด์ความถี่ และ Latch 2 จะใช้ในการพักข้อมูลในการปรับอัตราขยายของแต่ละแบนด์ โดยใช้ D7 เป็นตัวกำหนดว่าข้อมูลเป็นการเลือกแบนด์ หรือ เป็นการเลือกอัตราขยาย

เมื่อ D7 เป็น Low ข้อมูลนั้นเป็นการเลือกแบนด์ โดย D0 - D3 เป็นข้อมูลของการเลือกแบนด์ และ เมื่อ D7 เป็น High ข้อมูลก็จะเป็นการเลือกอัตราขยาย โดย D0 - D5 จะใช้เลือกอัตราขยาย และ D6 จะใช้เลือกว่าอัตราขยายนั้นเป็นการ บูสท์ หรือ คัท

Selector switch

ส่วนนี้เป็นส่วนสำคัญที่สุดเพราะเป็นการกำหนดอัตราขยาย และกำหนดว่าเป็นการ บูสท์ หรือ คัท ด้วย ซึ่งจะใช้โปรแกรมไปควบคุมสวิทช์ ในการเลือกระดับของความต้านทาน ทำให้เสมือนเป็นความต้านทานที่ปรับค่าได้ โดย Selector switch นี้ได้แสดงดังรูปที่ 3. ซึ่งจะเห็นว่า D1 - D5 สามารถควบคุมอัตราขยายได้ตั้งแต่ 1 dB จนถึง 12 dB และใช้ D6 ใช้การเลือกว่าเป็นการ บูสท์ หรือ คัท ส่วนตัวที่ใช้ทำหน้าที่เป็นสวิทช์คือ IC เบอร์ 4066

ตัวถอดรหัส(Decoder)

ตัวถอดรหัสที่ใช้จะเป็นแบบ เป็น ข้อมูลไบนารี 4 บิตเป็นข้อมูลแบบ 16 เส้น ซึ่งแสดงดังตารางที่ 1. โดยตัวถอดรหัสที่ใช้คือ IC เบอร์ 74154 เป็นตัวถอดรหัส

จากตารางที่ 1. จะเห็นว่าวงจรถอดรหัสจะทำงานได้ก็ต่อเมื่อ G1 และ G2 มีค่าเป็น Low เมื่อถอดรหัสออกมาเส้นที่ถูกเลือกจะมีสถานะเป็น Low นอกนั้นก็จะเป็น High แต่ในการใช้งานนี้เราใช้ตัวถอดรหัส 2 ชุดเพื่อใช้เลือก channel ซึ่งจะใช้ D4 เป็นตัวเลือก และในการใช้งานนี้เราใช้เพียง channel ละ 10 แบนด์ ซึ่งจะเห็นว่าเราสามารถที่จะขยายแบนด์ได้อีกถ้าต้องการ

A	B	C	D	QF	QE	QD	QC	QB	QA	Q9	Q8	Q7	Q6	Q5	Q4	Q3	Q2	Q1	Q0
L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H
L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H
L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
L	H	L	L	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H
L	H	L	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
L	H	H	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
H	L	L	L	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
H	L	L	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
H	L	H	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
H	H	L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
H	H	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
H	H	H	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H

ตารางที่ 1. แสดงการถอดรหัสของ 74LS154

ในตารางที่ 2. นั้นเป็นตารางที่แสดงถึงรูปแบบของข้อมูลที่เราต้องส่งจากคอมพิวเตอร์เพื่อที่จะมาควบคุมตัวอิกควอลไลเซอร์ จะเห็นว่าเราแบ่งข้อมูลออกเป็น 2 ส่วน

- ในส่วนแรกนั้นเป็นข้อมูลที่ใช้ในการเลือกแบนด์ความถี่ที่ใช้ โดยใช้ D4 เป็นเลือก Channel ว่าเป็นทางซ้าย หรือ ขวา และใช้ D0 - D3 เป็นตัวเลือกแบนด์ที่ใช้

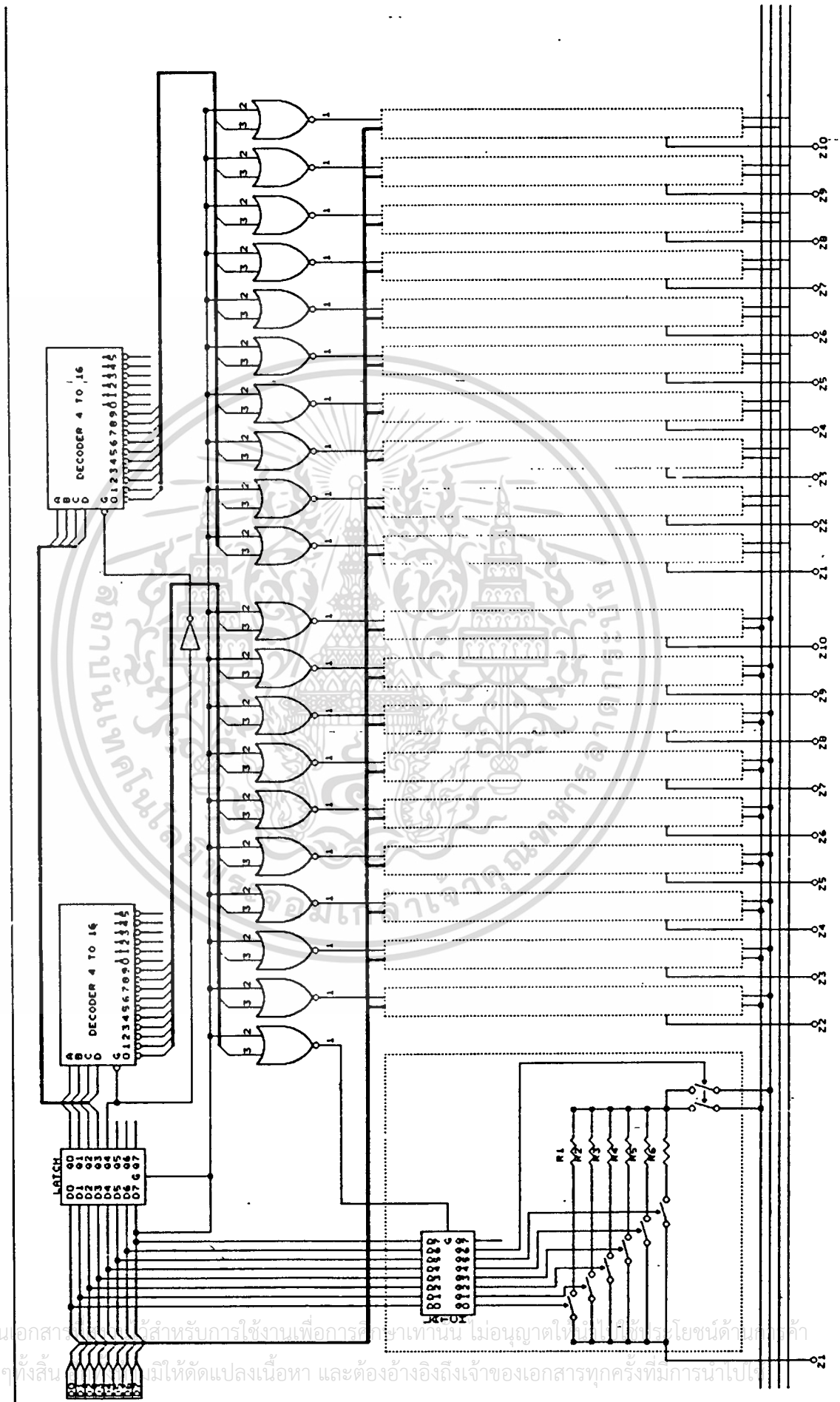
- ในส่วนที่ 2 นั้นเป็นข้อมูลที่ใช้ในการเลือกอัตราการขยาย โดยใช้ D0 - D5 เป็นตัวเลือกอัตราการขยาย และใช้ D6 เป็นตัวบอกว่าอัตราการขยายที่ใช้นั้นเป็นการ บูสท์ หรือ คัท

โดยข้อมูลทั้ง 2 ส่วนนี้จะใช้ D7 เป็นตัวชี้ว่าเป็นข้อมูลชุดแรก หรือ ชุดที่ 2 โดยถ้า D7 มีลอจิกเป็น 0 ข้อมูลนั้นจะเป็นข้อมูลชุดแรก และถ้า D7 มีค่าลอจิกเป็น 1 ข้อมูลนั้นจะเป็นข้อมูลชุดที่ 2

D7	D6	D5	D4	D3	D2	D1	D0	Select Right Channel
H	X	X	L	L	L	L	L	Select Band 0
H	X	X	L	L	L	L	H	Select Band 1
H	X	X	L	L	L	H	L	Select Band 2
H	X	X	L	L	L	H	H	Select Band 3
H	X	X	L	L	H	L	L	Select Band 4
H	X	X	L	L	H	L	H	Select Band 5
H	X	X	L	L	H	H	L	Select Band 6
H	X	X	L	L	H	H	H	Select Band 7
H	X	X	L	H	L	L	L	Select Band 8
H	X	X	L	H	L	L	H	Select Band 9
H	X	X	H	Select R			Select Left Channel	
L	X	L	L	L	L	L	L	FLAT
L	H	H	L	L	L	L	L	+ 1 dB BOOST
L	H	L	H	L	L	L	L	+ 2 dB BOOST
L	H	L	L	H	L	L	L	+ 3 dB BOOST
L	H	L	L	L	H	L	L	+ 4 dB BOOST
L	H	L	L	L	L	H	L	+ 5 dB BOOST
L	H	L	H	L	L	H	L	+ 6 dB BOOST
L	H	H	L	H	L	H	L	+ 7 dB BOOST
L	H	L	H	L	H	H	L	+ 8 dB BOOST
L	H	L	L	L	L	L	H	+ 9 dB BOOST
L	H	H	L	H	L	L	H	+ 10 dB BOOST
L	H	H	L	H	H	L	H	+ 11 dB BOOST
L	H	H	L	H	H	H	H	+ 12 dB BOOST
L	L	VALID ABOVE INPUT					-1 ถึง -12 dB CUT	

ตารางที่ 2. แสดงถึงลักษณะของข้อมูลที่ใช่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



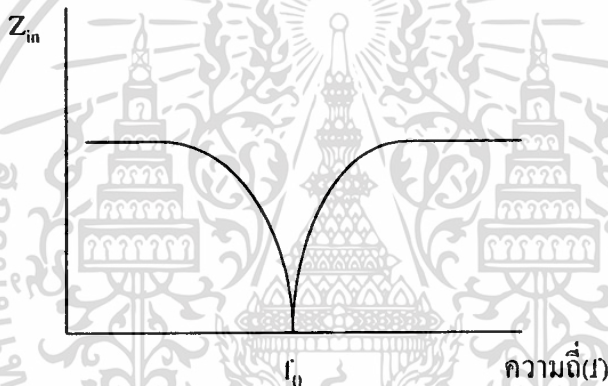
เอกสารนี้เป็นเอกสารสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต
 ไม่ว่ากรณีใดๆทั้งสิ้น มิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของอิกวอไลเซอร์

การออกแบบวงจร equalizer จะประกอบด้วยการออกแบบ วงจร Resonant ที่ความถี่ที่เราต้องการปรับซึ่งจะประกอบด้วย L และ C ต่ออนุกรม ซึ่งจะเห็นว่าจากวงจร resonant จะให้ค่า $Z_{in} = 0$ เมื่อ

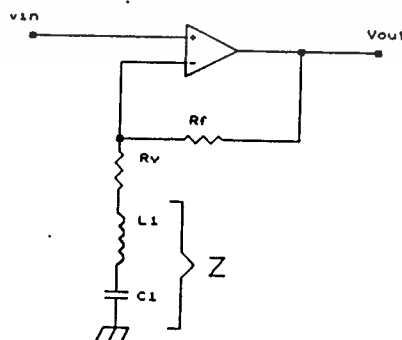
$$\omega_0 = \frac{1}{\sqrt{LC}}$$

ซึ่งจะมีค่ากราฟของ Z_{in} ดังรูป



รูปที่ 5.5 แสดงการตอบสนองความถี่ของวงจรเรโซแนนท์

การออกแบบส่วน boost ของวงจรจะใช้วงจรดังรูป



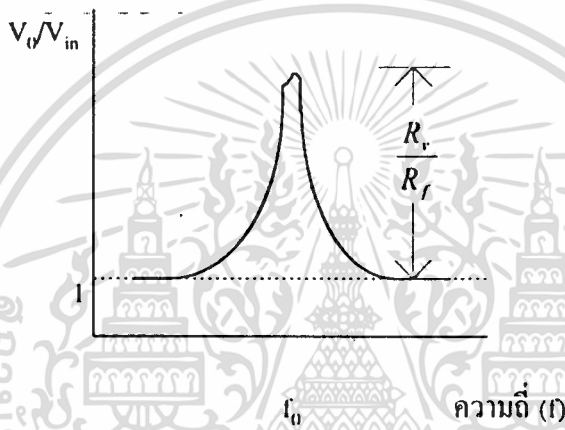
รูปที่ 5.6 แสดงวงจรในส่วนที่ใช้บูสต์ความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 5.6 จะได้ว่า

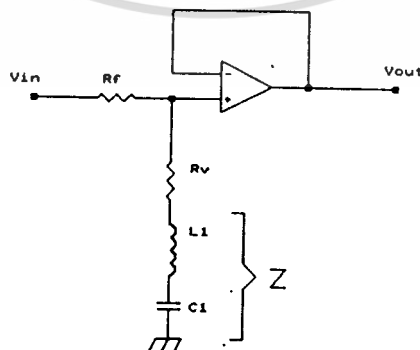
$$\frac{V_o}{V_{in}} = 1 + \frac{R_v}{R_f + Z}$$

จะเห็นว่า Gain ของวงจรขึ้นอยู่กับ $R_v/(R_f+Z)$ ซึ่งถ้าเราให้ $Z = 0$ จะให้ Gain ที่มากที่สุด ซึ่งจะมีค่า resonant ของระบบออกมาดังรูป



รูปที่ 5.7 แสดงผลตอบสนองของค่าความถี่ของวงจรในรูปที่ 5.6

การออกแบบส่วน Cut ของวงจร equalizer



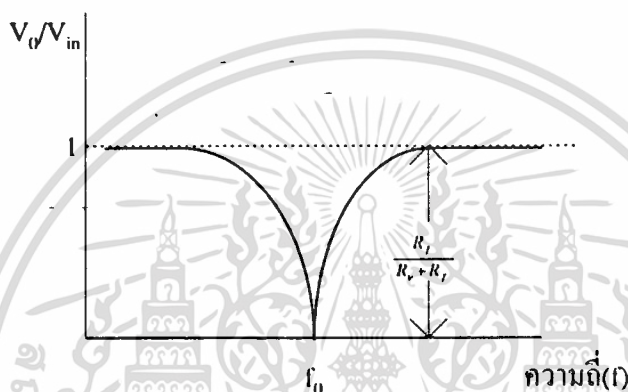
รูปที่ 5.8 แสดงวงจรที่ใช้ในการคัทความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 5.8 เราจะได้ว่า

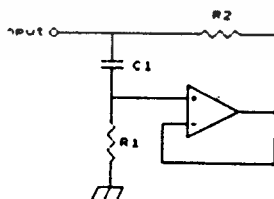
$$\frac{V_o}{V_{in}} = 1 - \left(\frac{R_f}{R_v + Z + R_f} \right)$$

จะเห็นว่าจากวงจรเราสามารถนำมาพล็อตกราฟตอบสนองความถี่ได้ดังรูป



รูปที่ 5.9 แสดงผลตอบสนองความถี่ของวงจรในรูปที่ 5.8

แต่ค่าของ L ที่ต้องการนั้นไม่สามารถหาได้หรือหาได้ยากมากดังนั้นจึงใช้วงจร Gyrator
เอามาใช้แทนตัว inductance ดังรูป



รูปที่ 5.10 แสดงวงจรไจเรเตอร์ที่ใช้แทน inductor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งค่า inductor หาได้จาก $L = R_1 * R_2 * C_1$ ซึ่งจากการคำนวณ เราสามารถหาค่า L กับ C ที่ความถี่ต่างๆ ได้ดังตารางที่ 3

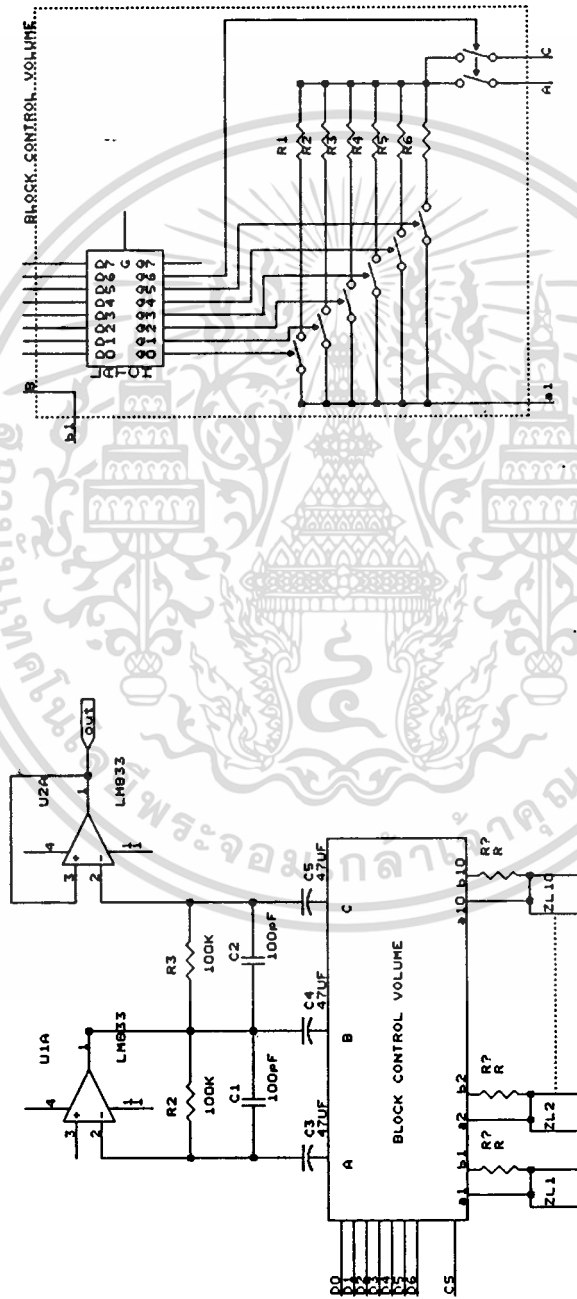
f0	L(H)	C	R1	R2	C1
32	16.49	1 μ 5	220K	1K	68n
64	8.036	820n	220K	1K	33n
125	4.16	390n	220K	1K	18n
250	1.84	220n	220K	1K	8n2
500	1.0	100n	220K	1K	3n9
1K	0.54	47n	220K	1K	2n2
2K	0.24	27n	220K	1K	1n
4K	0.13	12n	220K	1K	560p
8K	0.0582	6n8	220K	1K	240p
16K	0.03	3n3	220K	1K	150p

ตารางที่ 3 แสดงค่า R,L,C ที่ใช้ในวงจร

ขั้นตอนการทำงานของวงจร

การทำงานของวงจรจะประกอบด้วยส่วนของวงจรที่เป็น band pass filter โดยมี วงจร LC เป็นตัวกำหนดความถี่ศูนย์กลางของ band ซึ่งจากวงจรจะประกอบด้วย 10 band ซึ่งการที่จู่ปรับค่าเสียงเหล่านี้จะต้องมีการเลือก band ของความถี่ที่ใช้ในการปรับก่อนจากนั้นจึงทำการเลือกความถี่ต่างๆ -12 ถึง 12 db ซึ่งมีวิธีการ boost หรือการ cut นั้นได้อธิบายวิธีการทำงานไว้ข้างต้น ซึ่งการที่จะเลือกการ boost หรือ cut นั้นขึ้นอยู่กับว่าค่าของ R ที่ปรับต่ออยู่ที่ ขา A หรือ C ถ้าเป็นขา A จะเป็นขา C จะเป็นการ cut ดังรูปที่ 5.11

นอกจากเราจะเลือกการ boost หรือ cut โดยใช้ D_0 เ้าเราจะต้องตั้งว่า เราจะ boost หรือ cut ไปกี่ db โดยใช้ $D_0 - D_6$ ในการเลือก Analog Switch ที่มีค่าความต้านทานต่ออยู่ ซึ่งได้ทำการหาไว้แล้ว จะใช้ค่าความต้านทานที่ขั้ว Analog Switch



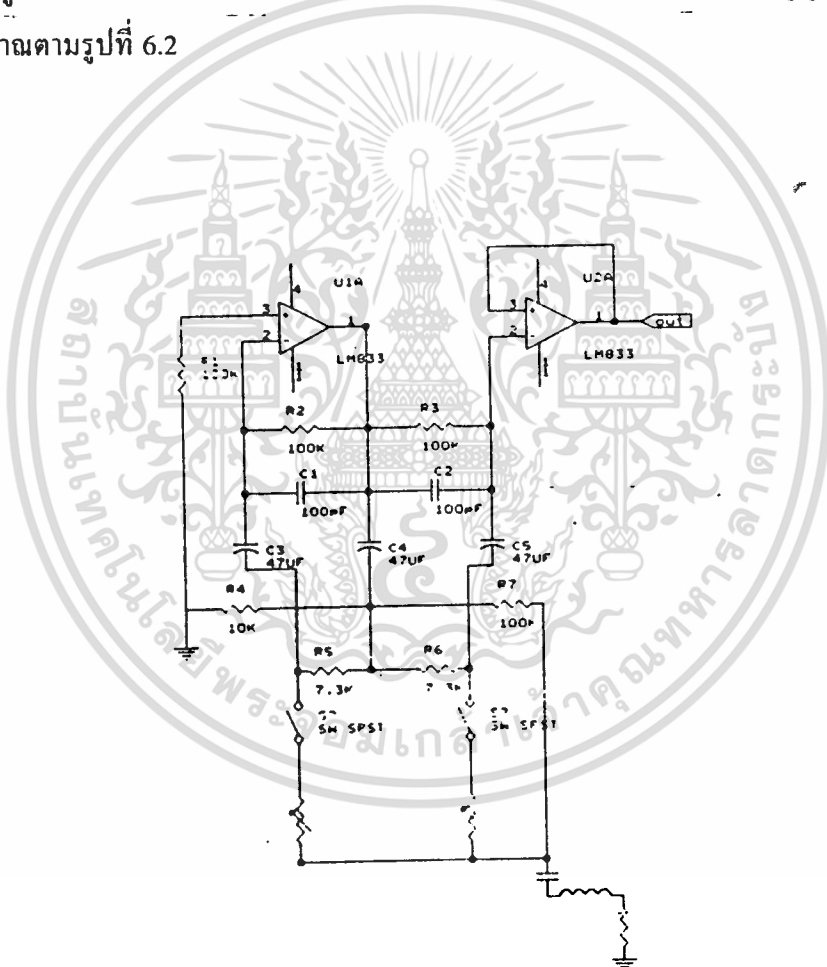
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การทดลอง

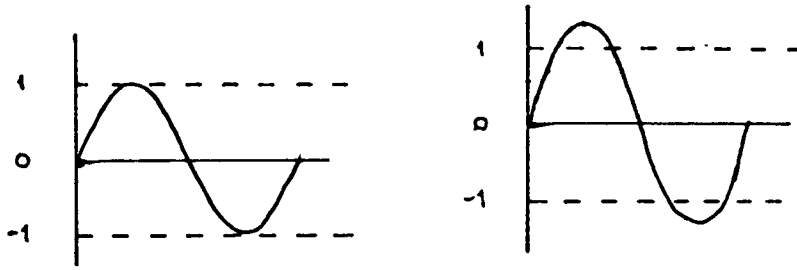
การทดลองการบวมและคัทของอิกวอไลเซอร์

ในการทดลองนี้ได้ทำการทดลองโดยป้อนค่าสัญญาณ sine wave ขนาด 1 V_{p-p} แล้วป้อนเข้าวงจรในรูปที่ 6.1 จากนั้นทำการเปิด-ปิดสวิตช์ A1 และ A2 ทีละตัวแล้ววัดสัญญาณที่เอาท์พุทจะได้สัญญาณตามรูปที่ 6.2



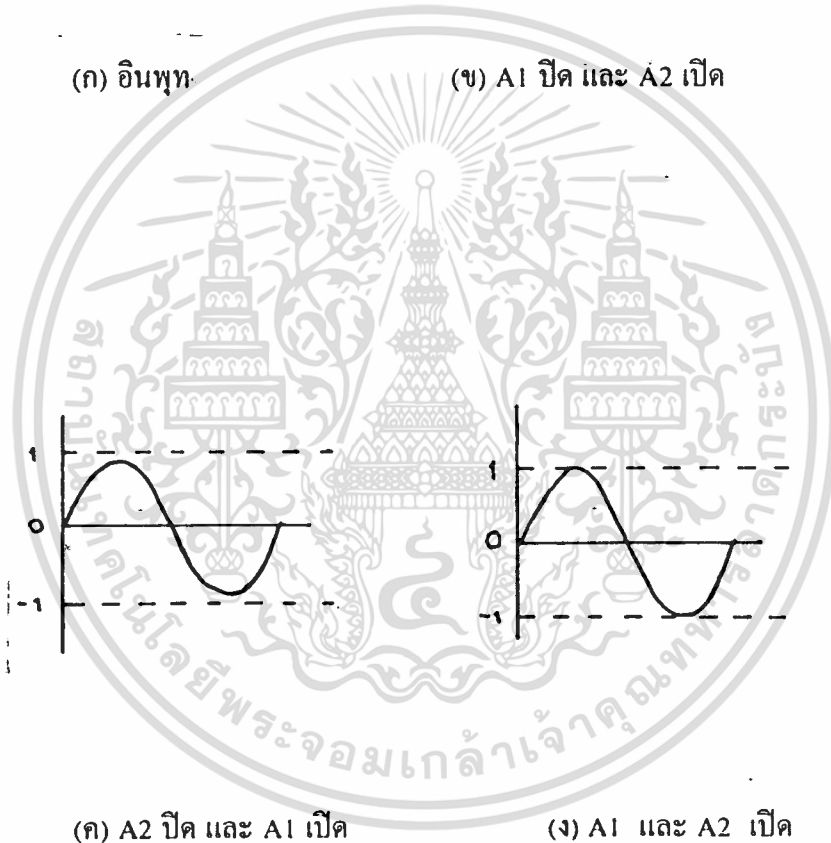
รูปที่ 6.1 แสดงวงจรพื้นฐานของอิกวอไลเซอร์ที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) อินพุต

(ข) A1 ปิด และ A2 เปิด



(ค) A2 ปิด และ A1 เปิด

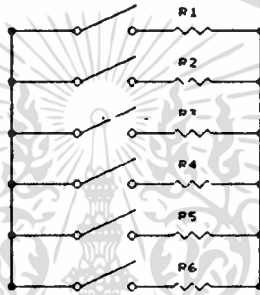
(ง) A1 และ A2 เปิด

รูปที่ 6.2 แสดงผลการทดลอง

จากสัญญาณที่ได้ตามรูปที่ 6.2 จะเห็นว่า เมื่อเราทำการปิดสวิตช์ A1 และ เปิดสวิตช์ A2 สัญญาณเอาต์พุตที่ได้จะมีขนาดใหญ่กว่าในสัญญาณอินพุต แสดงว่าสวิตช์ A1 เป็นสวิตช์ที่ใช้ในการบรูสต์สัญญาณ และ เมื่อเราทำการปิดสวิตช์ A2 และ เปิดสวิตช์ A1 สัญญาณที่ได้จะมีขนาดเล็กกว่าสัญญาณอินพุต แสดงว่าสวิตช์ A2 เป็นสวิตช์ที่ใช้ในการคัทสัญญาณ

การทดลองการปรับค่าอัตราขยาย

ในการทดลองหาค่าอัตราขยายนี้ทำโดยหาค่า R_f ของวงจร เมื่อใช้ขนาดลอคสวิตซ์ตัวเดียวค่าอัตราขยายที่เกิดขึ้นจะออกมาตามที่ต้องการแต่เมื่อนำเอาขนาดลอคสวิตซ์หลายๆ ตัวมาต่อกันดังรูปที่ 6.3



รูปที่ 6.3 แสดงการใช้สวิตซ์ในการเลือกอัตราขยาย

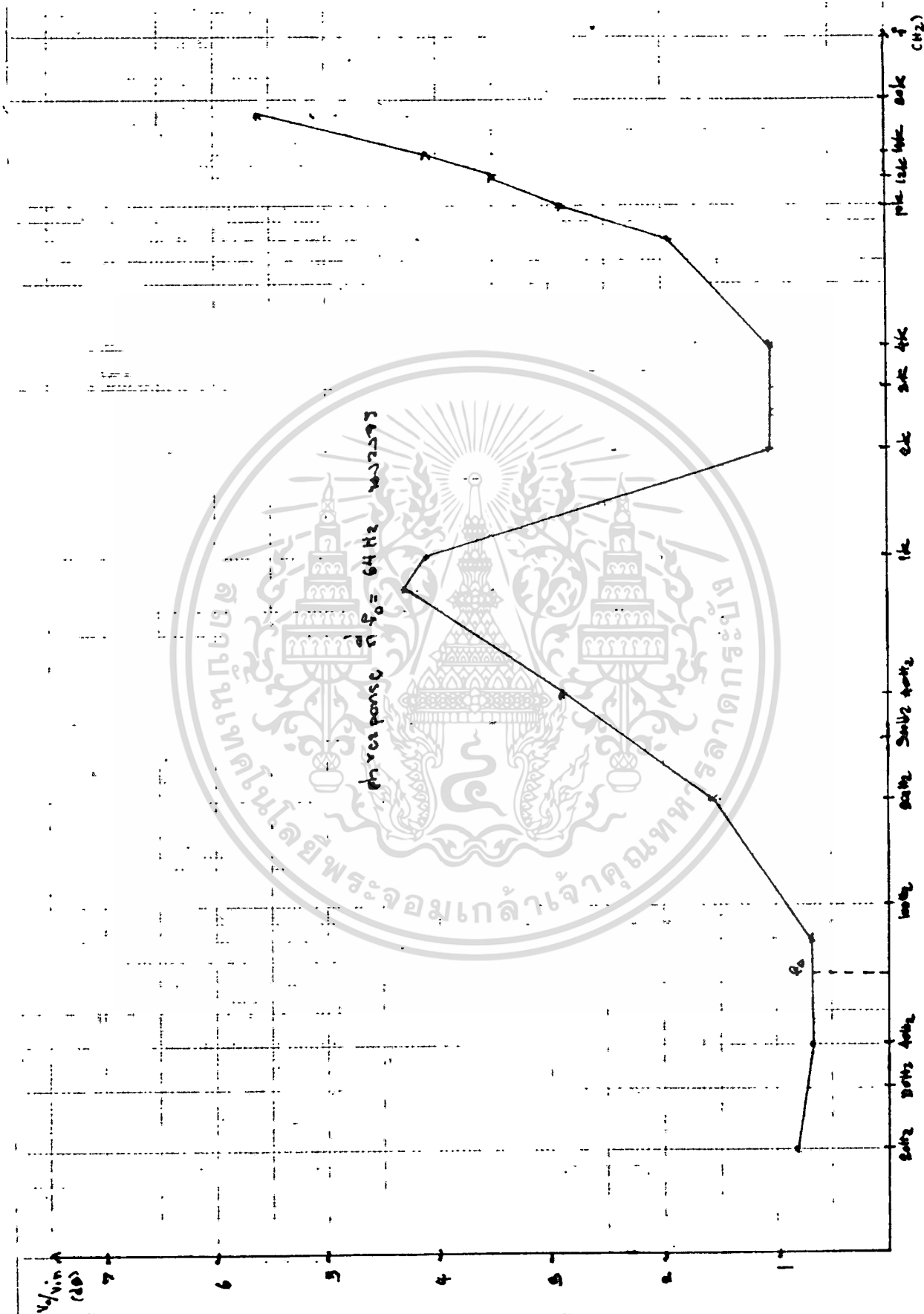
ปรากฏว่าอัตราขยายที่ได้ตรงตามความต้องการแต่เนื่องจากค่าความต้านทานของสวิตซ์แต่ละสวิตซ์มีค่าไม่เท่ากัน จึงเป็นผลทำให้อัตราขยายไม่เท่ากันและมีค่าน้อยลงด้วยและเมื่อเราทำการทดลองต่อกับเครื่องเสียง แล้วผ่านเครื่องขยายเสียง แล้วฟังเสียงคุณก็จะเห็นว่าเสียงออกมาตามแบนด์ต่างๆ นั้นออกมาตามที่เรปรับค่าไว้

และเมื่อนำเอาอิกควอลไลเซอร์ไปต่อกับคอมพิวเตอร์แล้วใช้คอมพิวเตอร์มาควบคุมการปรับอัตราขยายจะเห็นว่าสามารถที่จะควบคุมได้ตามที่เราต้องการ

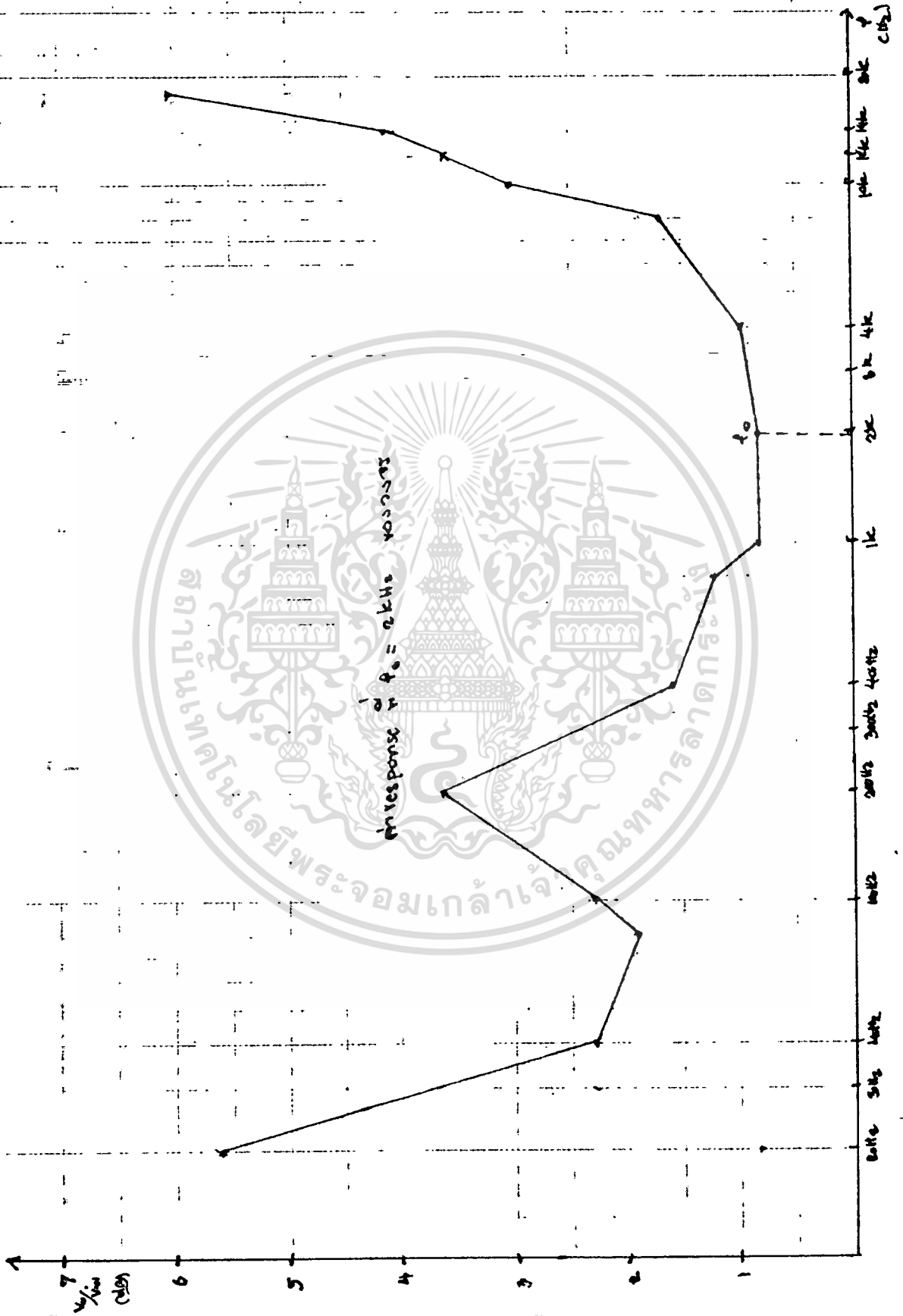
ความถี่(Hz)	V_o/V_i (dB) $f_o = 64$ Hz	V_o/V_i (dB) $f_o = 500$ Hz	V_o/V_i (dB) $f_o = 2K$ Hz	V_o/V_i (dB) $f_o = 8K$ Hz
20	0.83	5.6	5.6	1.14
40	0.67	2.3	2.3	0.4
80	0.67	2.3	1.94	0.4
100	0.83	2.34	2.3	0.83
200	1.58	0.83	3.64	1.58
400	2.9	0.83	1.58	2.9
800	4.3	0.83	1.21	3.5
1000	4.1	0.83	0.83	2.3
2000	1.06	0.83	0.83	0.98
4000	1.06	0.98	0.98	0.91
8000	1.94	1.21	1.7	1.58
10K	2.9	2.3	3.05	2.7
12K	3.5	2.92	4.1	4.7
14K	4.1	3.64	4.61	5.1
18K	5.6	4.9	6.02	5.6

ตารางแสดงค่าความตอบสนอง ของวงจร ฮีควอไลเซอร์ ที่ f_o ต่างๆ

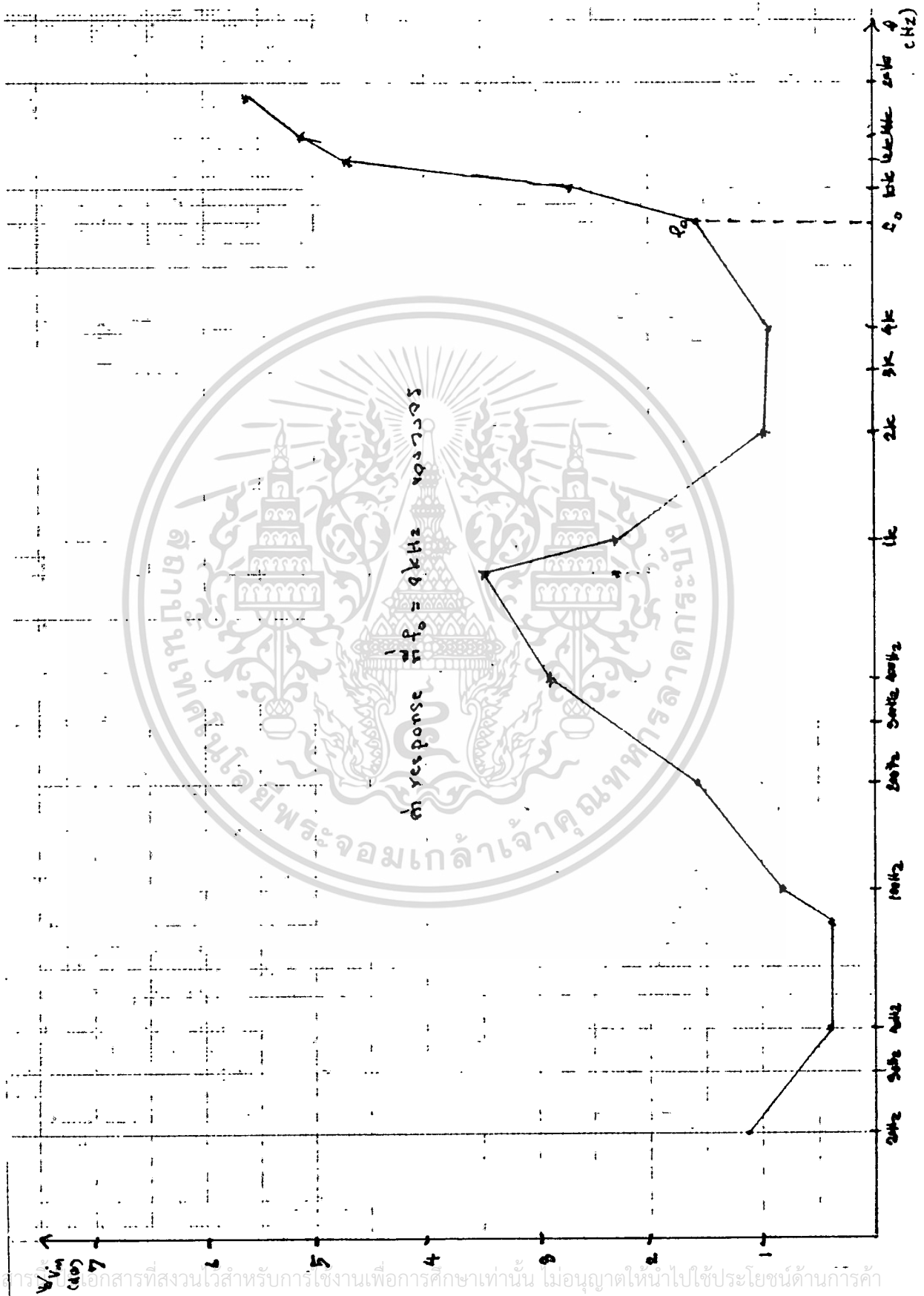
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

สรุปผลและวิจารณ์

การที่นำเอาคอมพิวเตอร์มาควบคุม อีควอไลเซอร์ ทำให้สามารถจัดเก็บค่าของการปรับความดัง ที่ แบนด์ ต่างๆโดยสามารถเรียกกลับมาใช้ได้ นอกจากนี้เราสามารถนำมาใช้กับระบบมัลติมีเดีย โดยติดตั้งเป็นชุดกับระบบเครื่องเสียง นอกจากนี้ยังสามารถนำมาใช้กับการควบคุมเสียงในหอประชุมการแสดง ซึ่งสามารถเอาคอมพิวเตอร์มาทำการวัดระดับเสียงที่ออกมาตามห้อง แล้วเอาค่าของระดับเสียงนั้นมาคำนวณเพื่อเอาค่าเหล่านั้นมาใช้ในการปรับทำให้สามารถใช้ ลำโพงได้อย่างมีประสิทธิภาพเพิ่มขึ้น ซึ่งจากการทดลองทำให้เห็นว่าเราสามารถปรับปรุงคุณภาพของเสียงที่ออกมาได้จริง แต่ก็ไม่สามารถให้คุณภาพเสียงออกมาดีเต็มที่ซึ่งเป็นปัญหาที่เกิดจากค่าความตอบสนองของลำโพงที่ใช้กับเครื่องขยายเสียง

จากผลการทดลองวัดค่าของระดับความดังของสัญญาณโดยทำการ เพิ่ม (Boost) หรือ ลด (Cut) ความดัง เราสามารถทำการ เพิ่ม หรือ ลด ได้ตามต้องการ แต่ทำการวัดค่าความตอบสนองของวงจรที่ความถี่พื้นฐาน(fundamental)ต่างๆ เราจะเห็นค่าความตอบสนองที่ได้มาไม่ตรงกับทฤษฎี ซึ่งเกิดจาก

- ความผิดพลาดของวิธีการวัด

- เกิดจากความผิดพลาดของตัววงจรที่ออกแบบมายังไม่สมบูรณ์เพราะตัวของวงจรที่เอามานั้นนำมาจากหนังสือหลายเล่มมาประกอบกัน ซึ่งยังไม่ได้คำนึงผลของวงจรที่นำมาต่อรวมกันจึงทำให้เกิดความผิดพลาดขึ้นในช่วงค่าความตอบสนองที่ความถี่พื้นฐานต่างๆที่วัดมา เพราะเวลาวัดจะวัดจากวงจรรวมที่ควบคุมด้วยดิจิตอล

จากผลการทดลองต่อคอมพิวเตอร์ ทดลองเขียนกราฟดูจะเห็นว่าค่าหลายช่วงความถี่มีความแตกต่างกันน้อยมาก บางช่วงก็สามารถแยกความแตกต่างกันได้ อาจเกิดจากสาเหตุที่ได้กล่าวมาแล้ว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings

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

Supply Voltage	$V_{CC}-V_{EE}$	36V
Differential Input Voltage (Note 1)	V_{ID}	$\pm 30V$
Input Voltage Range (Note 1)	V_{IC}	$\pm 15V$
Power Dissipation (Note 2)	P_D	500 mW
Operating Temperature Range	T_{OPR}	-40 - 85°C
Storage Temperature Range	T_{STG}	-60 - 150°C

Soldering Information

Dual-In-Line Package	260°C
Soldering (10 seconds)	
Small Outline Package	215°C
Vapor Phase (60 seconds)	
Infrared (15 seconds)	220°C

See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.

DC Electrical Characteristics ($T_A = 25^\circ\text{C}$, $V_S = \pm 15V$)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{OS}	Input Offset Voltage	$R_S = 10\Omega$		0.3	5	mV
I_{OS}	Input Offset Current			10	200	nA
I_B	Input Bias Current			500	1000	nA
A_v	Voltage Gain	$R_L = 2\text{ k}\Omega$, $V_O = \pm 10V$	90	110		dB
V_{Osw}	Output Voltage Swing	$R_L = 10\text{ k}\Omega$ $R_L = 2\text{ k}\Omega$	± 12 ± 10	± 13.5 ± 13.4		V V
V_{CM}	Input Common-Mode Range		± 12	± 14.0		V
CMRR	Common-Mode Rejection Ratio	$V_{IN} = \pm 12V$	80	100		dB
PSRR	Power Supply Rejection Ratio	$V_S = 15 \sim 5V$, $-15 \sim -5V$	80	100		dB
I_Q	Supply Current	$V_O = 0V$, Both Amps		5	8	mA

AC Electrical Characteristics ($T_A = 25^\circ\text{C}$, $V_S = \pm 15V$, $R_L = 2\text{ k}\Omega$)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
SR	Slew Rate	$R_L = 2\text{ k}\Omega$	5	7		V/ μs
GBW	Gain Bandwidth Product	$f = 100\text{ kHz}$	10	15		MHz

Design Electrical Characteristics ($T_A = 25^\circ\text{C}$, $V_S = \pm 15V$)

The following parameters are not tested or guaranteed

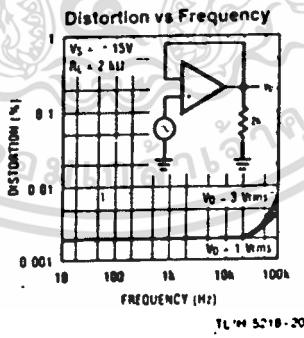
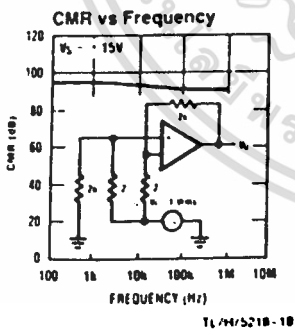
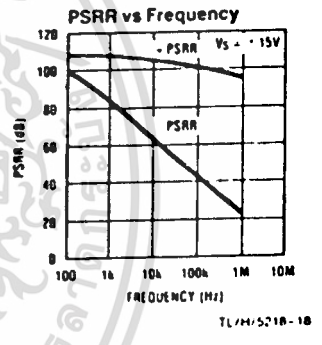
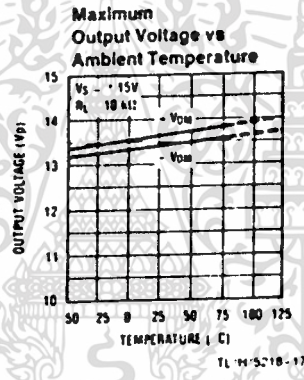
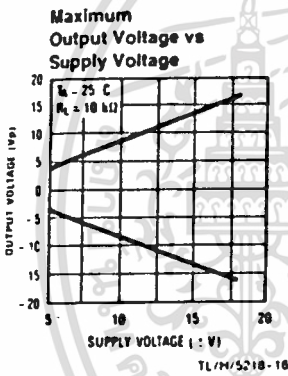
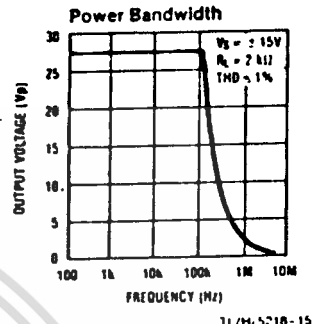
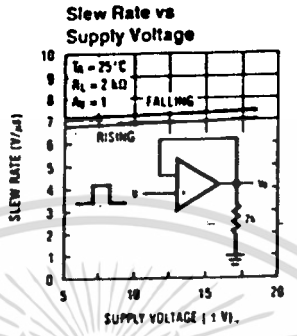
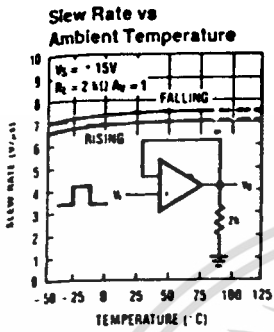
Symbol	Parameter	Conditions	Typ	Units
$\Delta V_{OS}/\Delta T$	Average Temperature Coefficient of Input Offset Voltage		2	$\mu\text{V}/^\circ\text{C}$
THD	Distortion	$R_L = 2\text{ k}\Omega$, $f = 20 \sim 20\text{ kHz}$ $V_{OUT} = 3\text{ Vrms}$, $A_v = 1$	0.002	%
e_n	Input Referred Noise Voltage	$R_S = 100\Omega$, $f = 1\text{ kHz}$	4.5	$\text{nV}/\sqrt{\text{Hz}}$
i_n	Input Referred Noise Current	$f = 1\text{ kHz}$	0.7	$\text{pA}/\sqrt{\text{Hz}}$
PBW	Power Bandwidth	$V_O = 27\text{ Vpp}$, $R_L = 2\text{ k}\Omega$, THD $\leq 1\%$	120	kHz
f_u	Unity Gain Frequency	Open Loop	9	MHz
ϕ_M	Phase Margin	Open Loop	60	deg
	Input Referred Cross Talk	$f = 20 \sim 20\text{ kHz}$	-120	dB

Note 1: If supply voltage is less than $\pm 15V$, it is equal to supply voltage

Note 2: This is the permissible value at $T_A = 85^\circ\text{C}$.

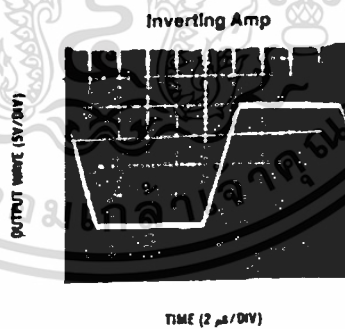
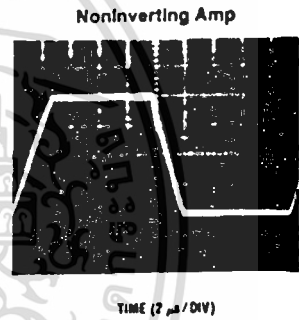
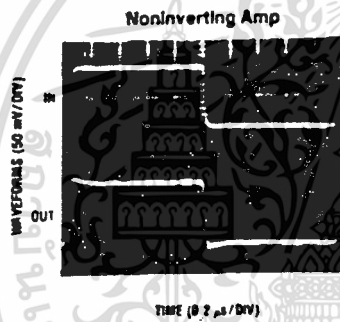
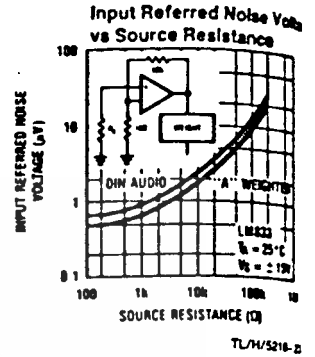
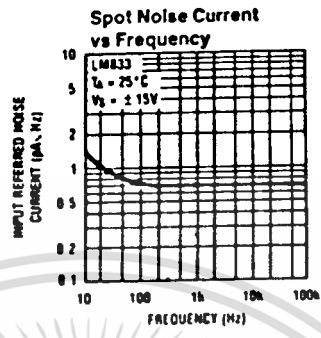
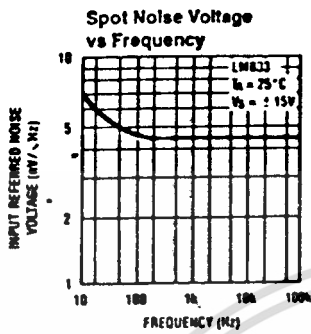
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Performance Characteristics (Continued)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Performance Characteristics (Continued)



Application Hints

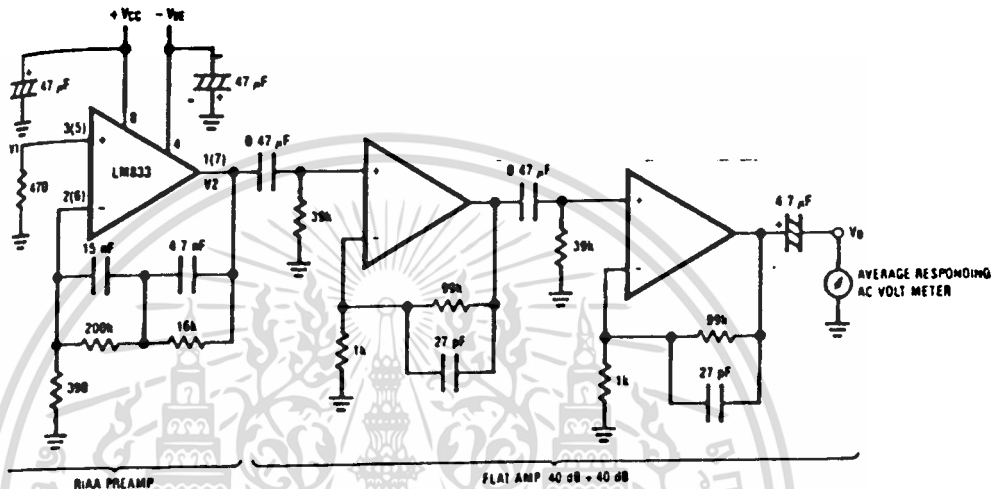
The LM833 is a high speed op amp with excellent phase margin and stability. Capacitive loads up to 50 pF will cause little change in the phase characteristics of the amplifiers and are therefore allowable.

Capacitive loads greater than 50 pF must be isolated from the output. The most straightforward way to do this is to put a resistor in series with the output. This resistor will also prevent excess power dissipation if the output is accidentally shorted.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Noise Measurement Circuit

Complete shielding is required to prevent induced pick up from external sources. Always check with oscilloscope for power line noise.

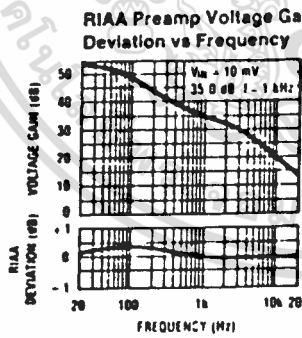


RIAA PREAMP

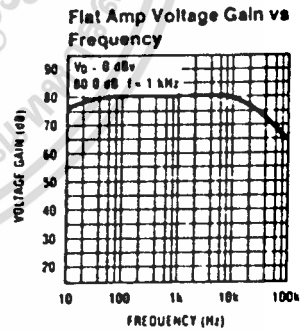
FLAT AMP 40 dB + 40 dB

Total Gain: 115 dB @ $f = 1 \text{ kHz}$
 Input Referred Noise Voltage: $e_n = V_0/560,000 \text{ (V)}$

TL/H/5218-2



TL/H 5218-2B

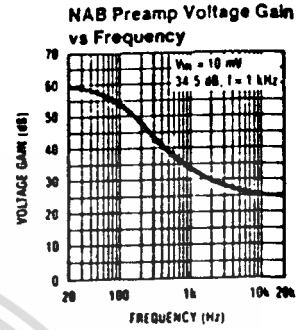
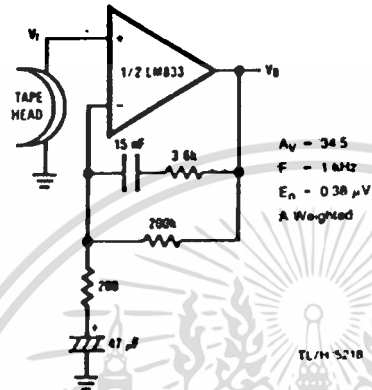


TL/H/5218-2C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

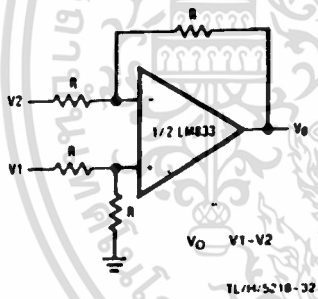
Typical Applications

NAB Preamp



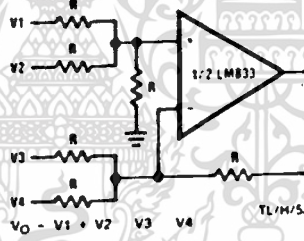
TL/M/5218-1

Balanced to Single Ended Converter



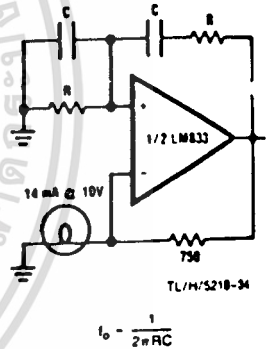
TL/M/5218-32

Adder/Subtractor



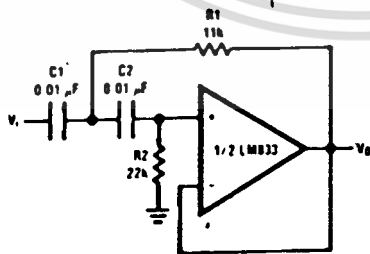
TL/M/5218-33

Sine Wave Oscillator



TL/M/5218-34

Second Order High Pass Filter (Butterworth)



TL/M/5218-35

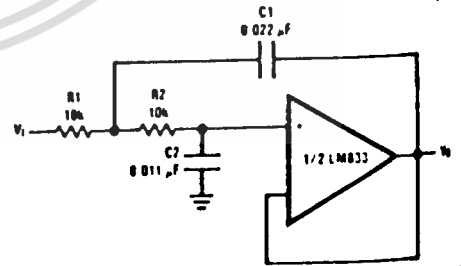
$R_1 = C_1 = C_2 = C$

$R_1 = \frac{R}{2\omega_0 C}$

$R_2 = 2 \cdot R_1$

Illustration $\omega_0 = 1 \text{ kHz}$

Second Order Low Pass Filter (Butterworth)



TL/M/5218-36

$R_1 = R_2 = R$

$C_1 = \frac{R}{\omega_0 R}$

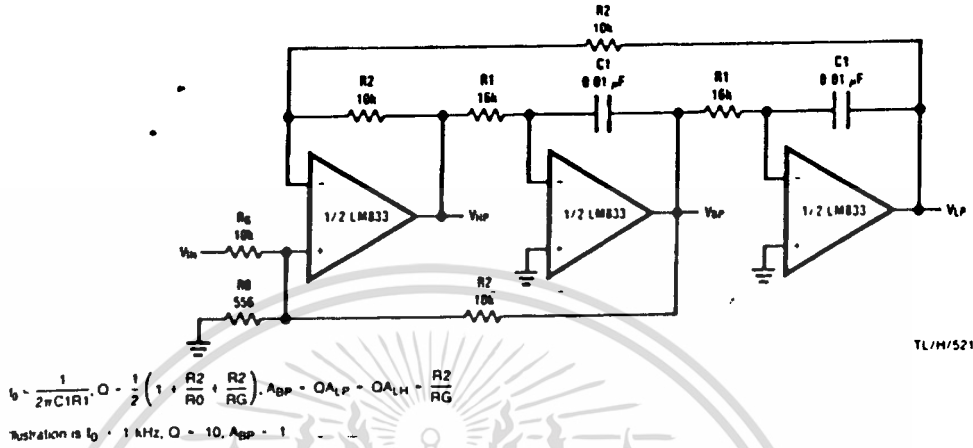
$C_2 = \frac{C_1}{2}$

Illustration $\omega_0 = 1 \text{ kHz}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

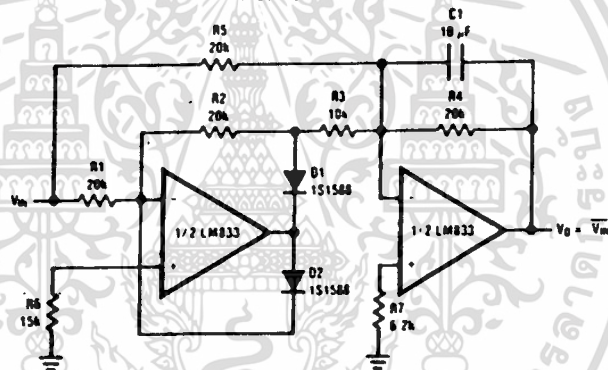
Typical Applications (Continued)

State Variable Filter



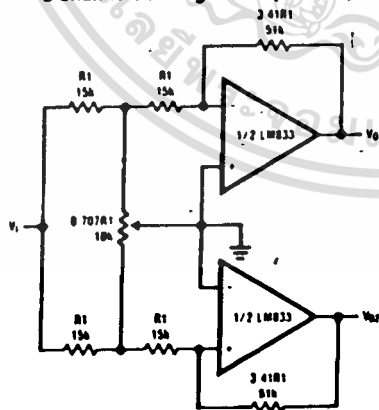
TL/H/5218-37

AC/DC Converter



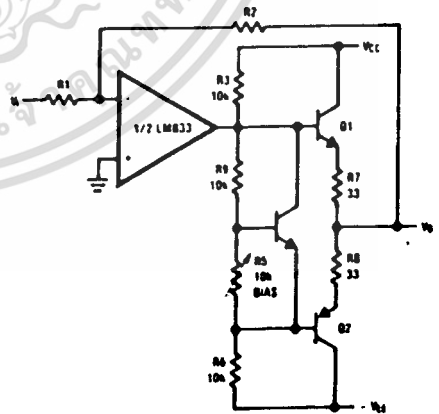
TL/H/5218-38

2 Channel Panning Circuit (Pan Pot)



TL/H/5218-39

Line Driver

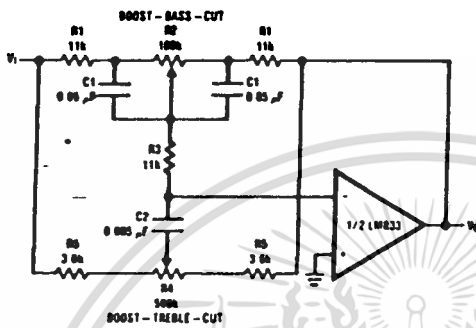


TL/H/5218-40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Application (Continued)

Tone Control



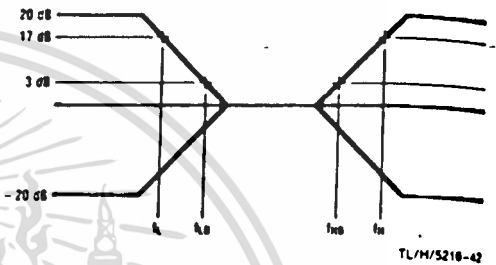
$$f_L = \frac{1}{2\pi R_2 C_1}, f_{LB} = \frac{1}{2\pi R_1 C_1}$$

$$f_H = \frac{1}{2\pi R_5 C_2}, f_{HB} = \frac{1}{2\pi(R_1 + R_5 + 2R_3)C_2}$$

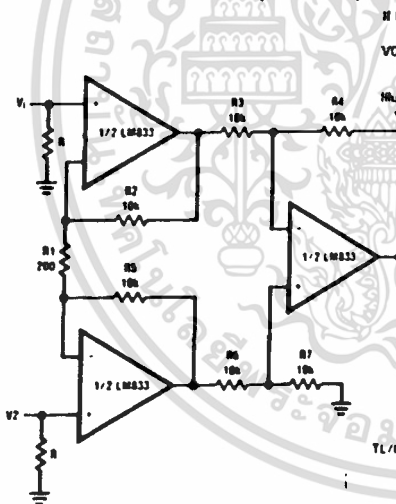
Illustration 1a:

$$f_L = 32 \text{ Hz}, f_{LB} = 320 \text{ Hz}$$

$$f_H = 11 \text{ kHz}, f_{HB} = 11 \text{ kHz}$$



Balanced Input Mic Amp

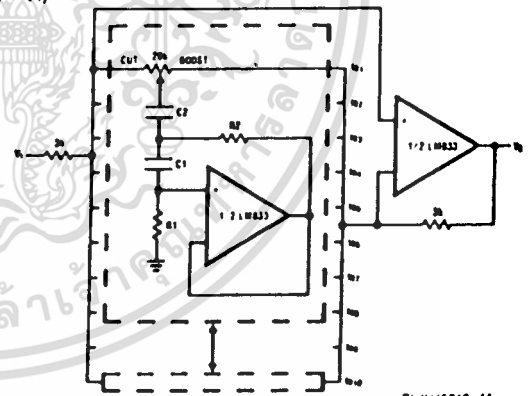


$$\text{If } R_2 = R_5, R_3 = R_6, R_4 = R_7$$

$$V_0 = \left(1 + \frac{2R_2}{R_1}\right) \frac{R_4}{R_3} (V_2 - V_1)$$

Illustration 1b:
 $V_0 = 101(V_2 - V_1)$

10 Band Graphic Equalizer



f _o (Hz)	C ₁	C ₂	R ₁	R ₂
32	0.12μF	4.7μF	75kΩ	500Ω
64	0.056μF	3.3μF	68kΩ	510Ω
125	0.033μF	1.5μF	62kΩ	510Ω
250	0.015μF	0.82μF	68kΩ	470Ω
500	8200pF	0.39μF	62kΩ	470Ω
1k	3900pF	0.22μF	68kΩ	470Ω
2k	2000pF	0.1μF	68kΩ	470Ω
4k	1100pF	0.056μF	62kΩ	470Ω
8k	510pF	0.022μF	68kΩ	510Ω
16k	330pF	0.012μF	51kΩ	510Ω

At volume of change = ± 12 dB
Q = 1.7

Reference: "AUDIO/RADIO HANDBOOK", National Semiconductor, 1980, Page 2-81

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADC0801, ADC0802, ADC0803, ADC0804, ADC0805



A to D, D to A

ADC0801, ADC0802, ADC0803, ADC0804, ADC0805 8-Bit μ P Compatible A/D Converters

General Description

The ADC0801, ADC0802, ADC0803, ADC0804 and ADC0805 are CMOS 8-bit successive approximation A/D converters which use a differential potentiometric ladder—similar to the 256R products. These converters are designed to allow operation with the NSC800 and JNS8080A derivative control bus, and TRI-STATE[®] output latches directly drive the data bus. These A/Ds appear like memory locations or I/O ports to the microprocessor and no interfacing logic is needed.

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

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

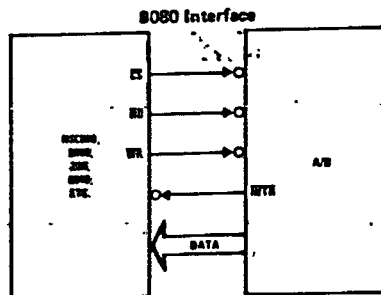
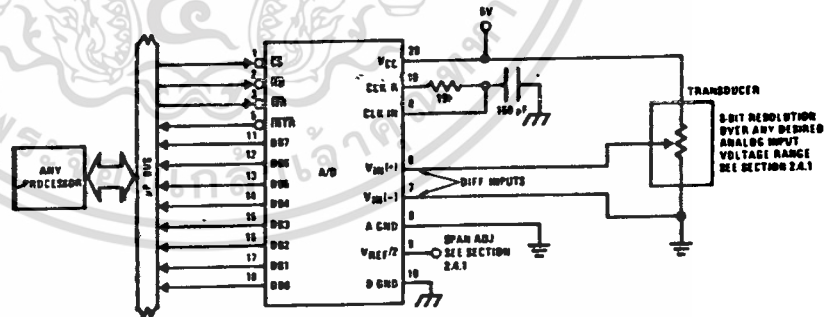
Features

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

Key Specifications

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

Typical Applications



PART NUMBER	FULL SCALE ADJUSTED	ERROR SPECIFICATION (INCLUDES FULL SCALE, ZERO ERROR, AND NON LINEARITY)	
		VREF/2 = 2.500 VDC (NO ADJUSTMENT)	VREF/2 = NO CONNECTION (NO ADJUSTMENT)
ADC0801	$\pm 1/4$ LSB	$\pm 1/2$ LSB	
ADC0802			
ADC0803	$\pm 1/2$ LSB	± 1 LSB	
ADC0804			
ADC0805			± 1 LSB

TRI-STATE[®] is a registered trademark of National Semiconductor Corp.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings (Notes 1 and 2)

Supply Voltage (VCC) (Note 3)	6.5V
Voltage	
Logic Control Inputs	-0.3V to +18V
At Other Input and Outputs	-0.3V to (VCC + 0.3V)
Storage Temperature Range	-65°C to +150°C
Package Dissipation at TA = 25°C	875 mW
Lead Temperature (Soldering, 10 seconds)	300°C

Operating Ratings (Notes 1 and 2)

Temperature Range	TMIN ≤ TA ≤ TMAX
ADC0801/02LD	-55°C ≤ TA ≤ +125°C
ADC0801/02/03/04LCD	-40°C ≤ TA ≤ +85°C
ADC0801/02/03/05LCN	-40°C ≤ TA ≤ +85°C
ADC0804LCN	0°C ≤ TA ≤ +70°C
Range of VCC	4.5 VDC to 6.3 VDC

Electrical Characteristics

The following specifications apply for VCC = 5 VDC, TMIN ≤ TA ≤ TMAX and fCLK = 640 kHz unless otherwise specified.

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
ADC0801: Total Adjusted Error (Note 8)	With Full Scale Adj. (See Section 2.5.2)			±1/4	LSB
ADC0802: Total Unadjusted Error (Note 8)	VREF/2 = 2.500 VDC			±1/2	LSB
ADC0803: Total Adjusted Error (Note 8)	With Full Scale Adj. (See Section 2.5.2)			±1/2	LSB
ADC0804: Total Unadjusted Error (Note 8)	VREF/2 = 2.500 VDC			±1	LSB
ADC0805: Total Unadjusted Error (Note 8)	VREF/2 - No Connection			±1	LSB
VREF/2 Input Resistance (Pin 8)	ADC0801/02/03/05 ADC0804 (Note 8)	2.5 1.0	8.0 1.3		kΩ kΩ
Analog Input Voltage Range	(Note 4) VIN(+) or VI(-)	Gnd-0.05		VCC+0.05	VDC
DC Common-Mode Error	Over Analog Input Voltage Range		±1/16	±1/8	LSB
Power Supply Sensitivity	VCC = 5 VDC ±10% Over Allowed VIN(+) and VI(-) Voltage Range (Note 4)		±1/16	±1/8	LSB

AC Electrical Characteristics

The following specifications apply for VCC = 5 VDC and TA = 25°C unless otherwise specified.

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Tc	Conversion Time fCLK = 640 kHz (Note 6)	103		114	μs
Tc	Conversion Time (Note 5, 6)	88		73	1/fCLK
fCLK	Clock Frequency VCC = 5V, (Note 5)	100	640	1460	kHz
fCLK	Clock Duty Cycle (Note 5)	40		60	%
CR	Conversion Rate in Free-Running Mode INTR tied to WR with CS = 0 VDC, fCLK = 640 kHz			8770	conv/s
tW(WR)L	Width of WR Input (Start Pulse Width) CS = 0 VDC (Note 7)	100			ns
tACC	Access Time (Delay from Falling Edge of RD to Output Data Valid) CL = 100 pF		135	200	ns
tTH, tOH	TRI-STATE Control (Delay from Rising Edge of RD to Hi-Z State) CL = 10 pF, RL = 10k (See TRI-STATE Test Circuits)		125	200	ns
tWI, tRI	Delay from Falling Edge of WR or RD to Reset of INTR		300	450	ns
CIN	Input Capacitance of Logic Control Inputs		5	7.5	pF
COUT	TRI-STATE Output Capacitance (Data Buffers)		5	7.5	pF

Electrical Characteristics

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

PARAMETER		CONDITIONS	MIN	TYP	MAX	UNITS
CONTROL INPUTS [Note: CLK IN (Pin 4) is the input of a Schmitt trigger circuit and is therefore specified separately]						
$V_{IN(1)}$	Logical "1" Input Voltage (Except Pin 4 CLK IN)	$V_{CC} = 5.25\text{ VDC}$	2.0		15	VDC
$V_{IN(0)}$	Logical "0" Input Voltage (Except Pin 4 CLK IN)	$V_{CC} = 4.75\text{ VDC}$			0.8	VDC
$I_{IN(1)}$	Logical "1" Input Current (All Inputs)	$V_{IN} = 5\text{ VDC}$		0.005	1	μADC
$I_{IN(0)}$	Logical "0" Input Current (All Inputs)	$V_{IN} = 0\text{ VDC}$	-1	-0.005		μADC
CLOCK IN AND CLOCK R						
V_{T+}	CLK IN (Pin 4) Positive Going Threshold Voltage		2.7	3.1	3.5	VDC
V_{T-}	CLK IN (Pin 4) Negative Going Threshold Voltage		1.5	1.8	2.1	VDC
V_H	CLK IN (Pin 4) Hysteresis (V_{T+}) - (V_{T-})		0.6	1.3	2.0	VDC
$V_{OUT(0)}$	Logical "0" CLK R Output Voltage	$I_O = 360\ \mu\text{A}$ $V_{CC} = 4.75\text{ VDC}$			0.4	VDC
$V_{OUT(1)}$	Logical "1" CLK R Output Voltage	$I_O = -360\ \mu\text{A}$ $V_{CC} = 4.75\text{ VDC}$	2.4			VDC
DATA OUTPUTS AND INTR						
$V_{OUT(0)}$	Logical "0" Output Voltage Data Outputs INTR Output	$I_{OUT} = 1.6\text{ mA}$, $V_{CC} = 4.75\text{ VDC}$ $I_{OUT} = 1.0\text{ mA}$, $V_{CC} = 4.75\text{ VDC}$			0.4 0.4	VDC VDC
$V_{OUT(1)}$	Logical "1" Output Voltage	$I_O = -360\ \mu\text{A}$, $V_{CC} = 4.75\text{ VDC}$	2.4			VDC
$V_{OUT(1)}$	Logical "1" Output Voltage	$I_O = -10\ \mu\text{A}$, $V_{CC} = 4.75\text{ VDC}$	4.5			VDC
I_{OUT}	TRI-STATE Disabled Output Leakage (All Data Buffers)	$V_{OUT} = 0\text{ VDC}$ $V_{OUT} = 5\text{ VDC}$	-3		3	μADC μADC
I_{SOURCE}		V_{OUT} Short to Gnd, $T_A = 25^\circ\text{C}$	4.5	6		mA
I_{SINK}		V_{OUT} Short to V_{CC} , $T_A = 25^\circ\text{C}$	9.0	16		mA
POWER SUPPLY						
I_{CC}	Supply Current (includes Ladder Current)	$f_{CLK} = 640\ \text{kHz}$, $V_{REF/2} = \text{NC}$, $T_A = 25^\circ\text{C}$ and $\overline{CS} = "1"$ ADC0801/02/03/05 ADC0804 (Note 9)		1.1 1.9	1.8 2.5	mA mA

Note 1: Absolute maximum ratings are those values beyond which the life of the device may be impaired.

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

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

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

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

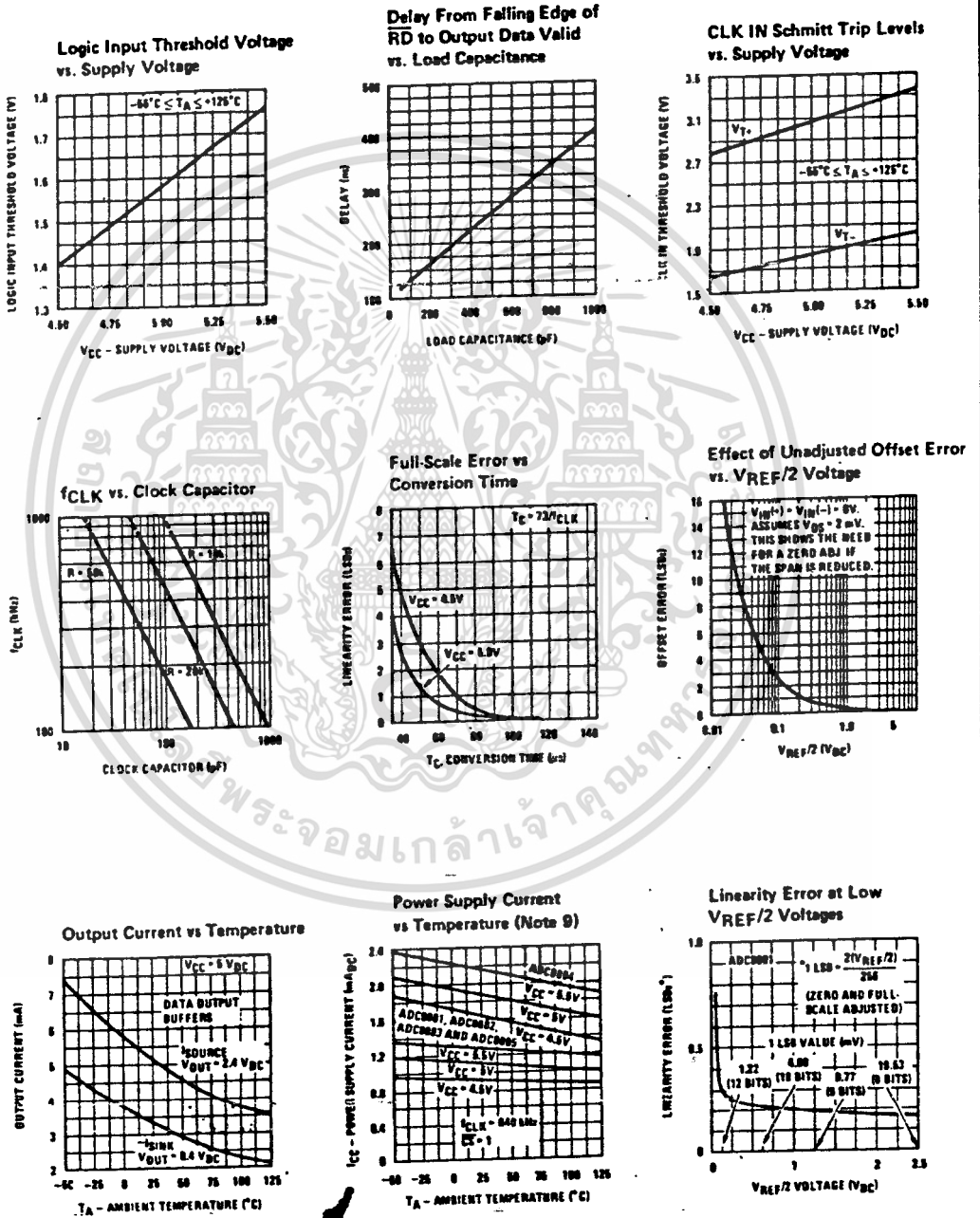
Note 6: \overline{CS} , an asynchronous start pulse, up to 8 clock periods may be required before the internal clock phases are proper to start the conversion. \overline{CS} start request is internally latched, see Figure 2 and section 2.0.

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

Note 8: None of these A/Ds requires a zero adjust (see section 2.5.1). To obtain zero code at other analog input voltages see section 2.5 and 2.6.

Note 9: For ADC0804LCD typical value of $V_{REF/2}$ input resistance is 8 k Ω and of I_{CC} is 1.1 mA.

Typical Performance Characteristics



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

```
#include <graphics.h>
#include <stdio.h>
#include <alloc.h>
#include <dos.h>
#include <bios.h>
#include <stdlib.h>
#include <stdarg.h>
```

```
#define ESC 0x1b
#define DOWN 0x50
#define UP 0x48
#define RIGHT 0x4d
#define LEFT 0x4b
#define F1 0x3b
#define F2 0x3c
#define F3 0x3d
```

```
struct Patfile{
    char patname[12];
    int patvalue[2][10];
};
```

```
union REGS r;
```

```
int x,y,readm;
```

```
int i,j,k;
```

```
int countm,count;
```

```
int Tab,ChTab.old;
```

```
int select[2][10] = {{0,0,0,0,0,0,0,0,0,0},{0,0,0,0,0,0,0,0,0,0}};
```

```
int oldselect[2][10] = {{0,0,0,0,0,0,0,0,0,0},{0,0,0,0,0,0,0,0,0,0}};
```

```
int startx,starty;
```

```
char *frequency[10] = {"32Hz","62Hz","125Hz","250Hz","500Hz","1KHz","2KHz","4KHz","8KHz","16KHz"};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char ch,*back;
FILE *pat;

struct Patfile *pattern[10];

void InitGraph(void);
void CloseGraph(void);
int Resetmouse(void);
void Showmouse(void);
void Hidemouse(void);
int Readmouse(void);
void Analyzer(void);
void Equalizer(void);
void Screen(void);
void Case(int x,int y);
void Box(int x,int y,int w,int l,int color);
void DrawUp(int x,int color,int color2);
void DrawDown(int x,int color,int color2);
void DrawSelectUp(int x,int y,int color1,int color2);
void DrawSelectDown(int x,int y,int color1,int color2);
void Load(void);
void Save(void);
void Auto(void);
int CheckX(int ax);
int CheckMenu(int ax);
int Checknum(int ay);
void CtrlKey(void);
void CtrlMouse(void);
void Outport(unsigned port,int side,int Ch);
void Wroteled(int ch,int side,int loud);
void Cleared(int ch,int side,int loud);
void Screen_Ana(void);
void Openwindow(int left,int top,int right,int bottom,int bkcolor);
void Closewindow(int left,int top);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int gprintf(int xloc,int yloc,char *fmt, ... );
void ScreenLS(char *menu,struct Patfile *N[10]);
char Curser(int x,int y,int bkcolor,int textcolor);
unsigned char FindMax(unsigned char t[5]);
void convert(unsigned char new[5],unsigned char old[5]);
void Exit(void);

void main(void)
{
    for(i=0;i<10;i++)
        pattern[i] = (struct Patfile *)malloc(sizeof(struct Patfile));

    if(Resetmouse()!=0){
        r.x.ax=0x0004;
        r.x.cx=0;
        r.x.dx=0;
        int86(0x33.&r,&r);
    }

    outportb(0x303,0x80);
    Equalizer();
}

void Equalizer(void)
{
    InitGraph();
    Screen();
    Showmouse();
    count=0;
    Tab=0;
    ChTab=0;
    old=0;
    do{
        if(bioskey(1)!= 0){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        CtrlKey():
        Outport(0x300,Tab.count);
    }
    else{
        CtrlMouse():
        Outport(0x300,ChTab.countm);
    }
}while(1):
}

void Screen(void)
{
    int a,c,xx,yy;
    setbkcolor(BLUE);
    setcolor(7);
    for(c=0;c<4;c++){
        line(0,24+c,getmaxx(),24+c);
        for(a=0;a<6;a++){
            line(0+c+(a*127),4.0+c+(a*127),24);
            Case(c,c);
        }
        gprintf((getmaxx()/2)-68,40,"EQUALIZER 10 BAND");
        gprintf((getmaxx()/2)-52,270,"LIFT CHANNEL");
        gprintf((getmaxx()/2)-52,225,"RIGHT CHANNEL");

        setcolor(WHITE);

        gprintf(28,10," F1-LOAD ");
        gprintf(155,10," F2-SAVE ");
        gprintf(282,10," F3-AUTO ");
        gprintf(409,10,"TAB-ANALY");
        gprintf(536,10,"ESC-QUIT!");
        for(c=0;c<10;c++){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DrawUp(c,WHITE,LIGHTGRAY);
DrawDown(c,WHITE,LIGHTGRAY);
DrawSelectUp(c,oldselect[0][c],BLACK,BLACK);
DrawSelectDown(c,oldselect[1][c],BLACK,BLACK);
DrawSelectUp(c,select[0][c],WHITE,LIGHTGRAY);
DrawSelectDown(c,select[1][c],WHITE,LIGHTGRAY);
gprintf(35+60*c,210 ,frequency[c]);
gprintf(35+60*c,285,frequency[c]);
Outport(0x300,0,c);
Outport(0x300,1,c);
}
for(xx=0;xx<10;xx++)
for(yy=0;yy<25;yy++){
if((yy==12)||(yy==0)||(yy==24)){
gprintf(22+(xx*60),320+(yy*4),"--");
gprintf(22+(xx*60),80+(yy*4),"--");
}
else{
gprintf(30+(xx*60),320+(yy*4),"--");
gprintf(30+(xx*60),80+(yy*4),"--");
}
}
}
for(xx=0;xx<10;xx++){
gprintf(15+(xx*60),312,"+12");
gprintf(15+(xx*60),425,"-12");
gprintf(12+(xx*60),367,"0");
gprintf(15+(xx*60),72,"+12");
gprintf(15+(xx*60),185,"-12");
gprintf(12+(xx*60),127,"0");
}
}
if(Tab==0){
DrawUp(count,WHITE,GREEN);
DrawSelectUp(count,select[Tab][count],WHITE,GREEN);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else{
    DrawDown(count,WHITE,GREEN);
    DrawSelectDown(count,select[Tab][count],WHITE,GREEN);
}
}

void Case(int x,int y)
{
    line(x,y,getmaxx()-x,y);
    line(x,y,x,getmaxy()-y);
    line(getmaxx()-x,y,getmaxx()-x,getmaxy()-y);
    line(x,getmaxy()-y,getmaxx()-x,getmaxy()-y);
}

void Box(int x,int y,int w,int l,int color)
{
    setcolor(color);
    line(x,y,x+w,y);
    line(x+w,y,x+w,y+l);
    line(x,y+l,x+w,y+l);
    line(x,y,x,y+l);
}

void DrawDown(int x,int color,int color2)
{
    Hidemouse();
    Box(40+(x*60),318,20,126,color);
    Box(41+(x*60),318,18,126,color);
    setfillstyle(SOLID_FILL
,color);
    bar(40+(x*60),298,60+(x*60),318);
    bar(40+(x*60),424,60+(x*60),444);
    setfillstyle(SOLID_FILL,color2);
    bar(42+(x*60),300,58+(x*60),316);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        bar(42+(x*60),426,58+(x*60),442);
        Showmouse();
    }

```

```

void DrawUp(int x,int color,int color2)

```

```

{
    Hidemouse();
    Box(40+(x*60),78,20,126,color);
    Box(41+(x*60),78,18,126,color);
    setfillstyle(SOLID_FILL,color);
    bar(40+(x*60),58,60+(x*60),78);
    bar(40+(x*60),184,60+(x*60),204);
    setfillstyle(SOLID_FILL,color2);
    bar(42+(x*60),60,58+(x*60),76);
    bar(42+(x*60),186,58+(x*60),202);
    Showmouse();
}

```

```

void DrawSelectUp(int x,int y,int color1,int color2)

```

```

{
    Hidemouse();
    setfillstyle(SOLID_FILL,color1);
    bar(42+(x*60),128-(y*4),58+(x*60),134-(y*4));
    setfillstyle(SOLID_FILL,color2);
    bar(43+(x*60),129-(y*4),57+(x*60),133-(y*4));
    Showmouse();
}

```

```

void DrawSelectDown(int x,int y,int color1,int color2)

```

```

{
    ;
    Hidemouse();
    setfillstyle(SOLID_FILL,color1);
    bar(42+(x*60),368-(y*4),58+(x*60),374-(y*4));
    setfillstyle(SOLID_FILL,color2);

```

```

bar(43+(x*60),369-(y*4),57+(x*60),373-(y*4));
Showmouse();
}

void CtrlKey(void)
{
    ch=getch();
    if(ch==0)
        ch=getch();

    if(ch==F1)
        Load();

    if(ch==F2)
        Save();

    if(ch==F3)
        Auto();

    if(ch==' '){
        if(Tab==1){
            ChTab=Tab=0;
            DrawUp(count,WHITE,GREEN);
            DrawDown(count,WHITE,LIGHTGRAY);
            DrawSelectDown(count,select[1][count],WHITE,LIGHTGRAY);
            DrawSelectUp(count,select[Tab][count],WHITE,GREEN);
        }
        else{
            ChTab=Tab=1;
            DrawDown(count,WHITE,GREEN);
            DrawUp(count,WHITE,LIGHTGRAY);
            DrawSelectUp(count,select[0][count],WHITE,LIGHTGRAY);
            DrawSelectDown(count,select[Tab][count],WHITE,GREEN);
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

if(ch==DOWN){
    select[Tab][count]--;
    if(select[Tab][count]<(-12))
        select[Tab][count]=(-12);
    if(Tab==1){
        DrawSelectDown(count,(select[Tab][count]+1),0,0);
        DrawSelectDown(count,select[Tab][count],WHITE,GREEN);
    }
    else{
        DrawSelectUp(count,(select[Tab][count]+1),0,0);
        DrawSelectUp(count,select[Tab][count],WHITE,GREEN);
    }
}

if(ch==UP){
    select[Tab][count]++;
    if(select[Tab][count] > 12)
        select[Tab][count]=12;
    if(Tab==1){
        DrawSelectDown(count,(select[Tab][count]-1),0,0);
        DrawSelectDown(count,select[Tab][count],WHITE,GREEN);
    }
    else{
        DrawSelectUp(count,(select[Tab][count]-1),0,0);
        DrawSelectUp(count,select[Tab][count],WHITE,GREEN);
    }
}
}
}

```

```

if(ch==RIGHT){

```

```

    if(Tab==0){

```

```

        count++;

```

```

        if(count>10)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        DrawSelectUp(9,select[Tab][9],WHITE,LIGHTGRAY);
    else
        DrawSelectUp(count-1,select[Tab][count-1],WHITE,LIGHTGRAY);
    if(count==10){
        count=0;
        DrawUp(count,WHITE,GREEN);
        DrawUp(9,WHITE,LIGHTGRAY);
        DrawSelectUp(count,select[Tab][count],WHITE,GREEN);
    }
    else{
        DrawUp(count,WHITE,GREEN);
        DrawUp((count-1),WHITE,LIGHTGRAY);
        DrawSelectUp(count,select[Tab][count],WHITE,GREEN);
    }
    else{
        count++;
        if(count>10)
            DrawSelectDown(9,select[Tab][9],WHITE,LIGHTGRAY);
        else
            DrawSelectDown(count-1,select[Tab][count-1],WHITE,LIGHTGRAY);
        if(count==10){
            count=0;
            DrawDown(count,WHITE,GREEN);
            DrawDown(9,WHITE,LIGHTGRAY);
            DrawSelectDown(count,select[Tab][count],WHITE,GREEN);
        }
    }
    else{
        DrawDown(count,WHITE,GREEN);
        DrawDown((count-1),WHITE,LIGHTGRAY);
        DrawSelectDown(count-1,select[Tab][count-1],WHITE,LIGHTGRAY);
        DrawSelectDown(count,select[Tab][count],WHITE,GREEN);
    }
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
old=count;
```

```
}
```

```
if(ch==LEFT){
```

```
    if(Tab==0){
```

```
        count--;
```

```
        if(count<0)
```

```
            DrawSelectUp(0,select[Tab][0],WHITE,LIGHTGRAY);
```

```
        else
```

```
            DrawSelectUp(count-1,select[Tab][count-1],WHITE,LIGHTGRAY);
```

```
        if(count==(-1)){
```

```
            count=9;
```

```
            DrawUp(count,WHITE,GREEN);
```

```
            DrawUp(0,WHITE,LIGHTGRAY);
```

```
            DrawSelectUp(count+1,select[Tab][count+1],WHITE,LIGHTGRAY);
```

```
            DrawSelectUp(count,select[Tab][count],WHITE,GREEN);
```

```
        }
```

```
        else{
```

```
            DrawUp(count,WHITE,GREEN);
```

```
            DrawUp((count+1),WHITE,LIGHTGRAY);
```

```
            DrawSelectUp(count+1,select[Tab][count+1],WHITE,LIGHTGRAY);
```

```
            DrawSelectUp(count,select[Tab][count],WHITE,GREEN);
```

```
        }
```

```
    }
```

```
else{
```

```
    count--;
```

```
    if(count<0)
```

```
        DrawSelectDown(count,select[Tab][0],WHITE,LIGHTGRAY);
```

```
    else
```

```
        DrawSelectDown(count-1,select[Tab][count-1],WHITE,LIGHTGRAY);
```

```
    if(count==(-1)){
```

```
        count=9;
```

```
        DrawDown(count,WHITE,GREEN);
```

```
        DrawDown(0,WHITE,LIGHTGRAY);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        DrawSelectDown(0,select[Tab][0],WHITE,LIGHTGRAY);
        DrawSelectDown(count,select[Tab][count],WHITE,GREEN);
    }
    else{
        DrawDown(count,WHITE,GREEN);
        DrawDown((count+1),WHITE,LIGHTGRAY);
        DrawSelectDown(count+1,select[Tab][count+1],
WHITE,LIGHTGRAY);
        DrawSelectDown(count,select[Tab][count],WHITE,GREEN);
    }
}
oid=count;
}
if(ch=='\n')
    Analyzer();
if(ch==ESC)
    Exit();
}
void CtrlMouse(void)
{
    int Check;

    readm=Readmouse();
    if(readm==0x0001){
        if(readm==1)
            while(Readmouse()==1);
        x = r.x.cx;
        y = r.x.dx;

        if(y>=4 && y<=24){
            Check=CheckMenu(x);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

switch(Check){
    case 0:
        Hidemouse();
        setfillstyle(SOLID_FILL,WHITE);
        bar(4,4,126,23);
        setcolor(BLUE);
        gprintf(28,10," F1-LOAD ");
        Showmouse();
        Load();
        break;
    case 1:
        Hidemouse();
        setfillstyle(SOLID_FILL,WHITE);
        bar(131,4,253,23);
        setcolor(BLUE);
        gprintf(155,10," F2-SAVE ");
        Showmouse();
        Save();
        break;
    case 2:
        Hidemouse();
        setfillstyle(SOLID_FILL,WHITE);
        bar(258,4,380,23);
        setcolor(BLUE);
        gprintf(282,10," F3-AUTO ");
        Showmouse();
        Auto();
        break;
    case 3:
        Analyzer();
        break;
    case 4:
        Exit();
}

```

1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

if(y>=78 && y<= 184){
    countm=CheckX(x);
    count=countm;
    if(countm!=(-1)){
        if(ChTab==1){
            DrawDown(old,WHITE,LIGHTGRAY);
            DrawSelectDown(old,select[ChTab][old],
WHITE,LIGHTGRAY);
        }
        DrawUp(old,WHITE,LIGHTGRAY);
        DrawSelectUp(old,select[0][old],WHITE,LIGHTGRAY);
        DrawSelectUp(countm,select[0][countm]-1,0,0);
        DrawUp(countm,WHITE,GREEN);
        DrawSelectUp(countm,select[0][countm],WHITE,GREEN);
        old=countm;
        Tab=ChTab=0;
    }
}

if(y>=318 && y<=424){
    countm=CheckX(x);
    count=countm;
    if(countm!=(-1)){
        if(ChTab==0){
            DrawUp(old,WHITE,LIGHTGRAY);
            DrawSelectUp(old,select[ChTab][old],WHITE,LIGHTGRAY)
        }
        DrawDown(old,WHITE,LIGHTGRAY);
        DrawSelectDown(old,select[1][old],WHITE,LIGHTGRAY);
        DrawSelectDown(countm,select[1][countm]-1,0,0);
        DrawDown(countm,WHITE,GREEN);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        DrawSelectDown(countm,select[1][countm],WHITE,GREEN);
        old=countm;
        Tab=ChTab=1;
    }
}

if(y>=58 && y<=78){
    countm=CheckX(x);
    if(countm!=(-1)){
        select[0][countm]++;
        count=countm;
        if(select[0][countm]>12)
            select[0][countm]=12;
        if(ChTab==1){
            DrawDown(old,WHITE,LIGHTGRAY);
            DrawSelectDown(old,select[ChTab][old],
WHITE,LIGHTGRAY);
        }
        DrawUp(old,WHITE,LIGHTGRAY);
        DrawSelectUp(old,select[0][old],WHITE,LIGHTGRAY);
        DrawSelectUp(countm,select[0][countm]-1,0,0);
        DrawUp(countm,WHITE,GREEN);
        DrawSelectUp(countm,select[0][countm],WHITE,GREEN);
        old=countm;
        Tab=ChTab=0;
    }
}
}

```

```

if(y>=184 && y<=204){
    countm=CheckX(x);
    if(countm!=(-1)){
        select[0][countm]--;
        count=countm;
        if(select[0][countm]<(-12))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

select[0][countm]=(-12);
if(ChTab==1){
    DrawDown(old,WHITE,LIGHTGRAY);
    DrawSelectDown(old,select[ChTab][old],
WHITE,LIGHTGRAY);
}
DrawUp(old,WHITE,LIGHTGRAY);
DrawSelectUp(old,select[0][old],WHITE,LIGHTGRAY);
DrawSelectUp(countm,select[0][countm]+1,0,0);
DrawUp(countm,WHITE,GREEN);
DrawSelectUp(countm,select[0][countm],WHITE,GREEN);
old=countm;
Tab=ChTab=0;
if(y>=298 && y<=318){
    countm=CheckX(x);
    if(countm!=(-1)){
        select[1][countm]++;
        count=countm;
        if(select[1][countm]>12)
            select[1][countm]=12;
        if(ChTab==0){
            DrawUp(old,WHITE,LIGHTGRAY);
            DrawSelectUp(old,select[ChTab][old],WHITE,LIGHTGRAY)
;
}
DrawDown(old,WHITE,LIGHTGRAY);
DrawSelectDown(old,select[1][old],WHITE,LIGHTGRAY);
DrawSelectDown(countm,select[1][countm]-1,0,0);
DrawDown(countm,WHITE,GREEN);
DrawSelectDown(countm,select[1][countm],WHITE,GREEN);
old=countm;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        .Tab=ChTab=1;
    }
}

if(y>=424 && y<=444){
    countm=CheckX(x);
    if(countm!=(-1)){
        select[1][countm]--;
        count=countm;
        if(select[1][countm]<(-12))
            select[1][countm]=(-12);
        if(ChTab==0){
            DrawUp(old,WHITE,LIGHTGRAY);
            DrawSelectUp(old,select[ChTab][old],WHITE,LIGHTGRAY)
        }
        DrawDown(old,WHITE,LIGHTGRAY);
        DrawSelectDown(old,select[1][old],WHITE,LIGHTGRAY);
        DrawSelectDown(countm,select[1][countm]+1,0,0);
        DrawDown(countm,WHITE,GREEN);
        DrawSelectDown(countm,select[1][countm],WHITE,GREEN);
        old=countm;
        Tab=ChTab=1;
    }
}

}

}

}

void Load(void)
{
    char key="";
    int numpat=0;
    int i,j;
    int addx,addy;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Openwindow((getmaxx()/2)-100,(getmaxy()/2)-200,(getmaxx()/2)+100,(getmaxy()/2)+200,
DARKGRAY);

setcolor(WHITE);

delay(500);

Hidemouse();

setfillstyle(SOLID_FILL,BLUE);

bar(4,4,126,23);

gprintf(28,10," F1-LOAD ");

Showmouse();

if((pat=fopen("pattern.sav","r+b"))==NULL){
    fprintf(stderr,"No file pattern.sav");
    exit(0);
}

for(i=0;i<10;i++){
    fread(pattern[i],sizeof(struct Patfile),1,pat);
    fclose(pat);
    ScreenLS("LOAD",pattern);
    settextjustify(CENTER_TEXT,TOP_TEXT);

do{
    if(bioskey(1)!=0){
        while(key != '\r'){
            key=getch();

            if(key==0)

                key=getch();

        }

        if(key==DOWN){
            numpat++;

            if(numpat>9){
                numpat=0;

            }

        }

    }

    Hidemouse();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Box((getmaxx()/2)-80,(getmaxy()/2)-155+
(numpat*35),160,20,WHITE);

setfillstyle(SOLID_FILL,BROWN);
bar((getmaxx()/2)-80,(getmaxy()/2)-155+
(numpat*35),(getmaxx()/2)+80,(getmaxy()/2)-135+(numpat*35));
gprintf(getmaxx()/2,(getmaxy()/2)-148+
(numpat*35),"%s",pattern[numpat]->patname);

Box((getmaxx()/2)-80,(getmaxy()/2)-155+(9*35),
160,20,WHITE);

setfillstyle(SOLID_FILL,LIGHTGRAY);
bar((getmaxx()/2)-80,(getmaxy()/2)-155+(9*35),
(getmaxx()/2)+80,(getmaxy()/2)-135+(9*35));
gprintf(getmaxx()/2,(getmaxy()/2)-148+(9*35),"%
s",pattern[9]->patname);

Showmouse();
}
else{
Hidemouse();
Box((getmaxx()/2)-80,(getmaxy()/2)-155+
(numpat*35),160,20,WHITE);

setfillstyle(SOLID_FILL,BROWN);
bar((getmaxx()/2)-80,(getmaxy()/2)-155+
(numpat*35),(getmaxx()/2)+80,(getmaxy()/2)-135+(numpat*35));
gprintf(getmaxx()/2,(getmaxy()/2)-148+
(numpat*35),"%s",pattern[numpat]->patname);

Box((getmaxx()/2)-80,(getmaxy()/2)-155+((numpat-
i)*35),160,20,WHITE);

setfillstyle(SOLID_FILL,LIGHTGRAY);
bar((getmaxx()/2)-80,(getmaxy()/2)-155+((numpat-
1)*35),(getmaxx()/2)+80,(getmaxy()/2)-135+((numpat-1)*35));
gprintf(getmaxx()/2,(getmaxy()/2)-148+((numpat-1)
*35),"%s",pattern[numpat-1]->patname);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

gprintf(getmaxx()/2,(getmaxy()/2)-148+
(numpat*35)."%s",pattern[numpat]->patname);

Box((getmaxx()/2)-80,(getmaxy()/2)-155+
((numpat+1)*35),160,20,WHITE);

setfillstyle(SOLID_FILL,LIGHTGRAY);
bar((getmaxx()/2)-80,(getmaxy()/2)-155+
((numpat+1)*35),(getmaxx()/2)+80,(getmaxy()/2)-135+((numpat+1)*35));
gprintf(getmaxx()/2,(getmaxy()/2)-148+((numpat+1)
*35)."%s",pattern[numpat+1]->patname);

Showmouse();
}
if(key=='r'){
for(i=0;i<2;i++){
for(j=0;j<10;j++){
oldselect[i][j]=select[i][j];
select[i][j] = pattern[numpat]->patvalue[i]
j];
}
}
else{
readm=Readmouse();
if(readm==0x0001){
if(readm==1)
while(Readmouse()==1);
addx = r.x.cx;
addy = r.x.dcx;

numpat=Checknum(addy);
if(numpat >= 0){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Hidemouse();
Box((getmaxx()/2)-80,(getmaxy()/2)-155+(numpat*35),
160,20,WHITE);

setfillstyle(SOLID_FILL,BROWN);
bar((getmaxx()/2)-80,(getmaxy()/2)-155+(numpat*35),
(getmaxx()/2)+80,(getmaxy()/2)-135+(numpat*35));
gprintf(getmaxx()/2,(getmaxy()/2)-148+(numpat*35),"%
s",pattern[numpat]->patname);

if(numpat > 0){
Box((getmaxx()/2)-80,(getmaxy()/2)-155,160,20,
WHITE);

setfillstyle(SOLID_FILL,LIGHTGRAY);
bar((getmaxx()/2)-80,(getmaxy()/2)-155,(getmaxx()
/2)+80,(getmaxy()/2)-135);
gprintf(getmaxx()/2,(getmaxy()/2)-148,"%s",pattern
[0]->patname);
}

Showmouse();
delay(500);
if((addx >= getmaxx()/2-80)&&(addx <= getmaxx()/2+80)){
for(i=0;i<2;i++){
for(j=0;j<10;j++){
oldselect[i][j]=select[i][j];
select[i][j] = pattern[numpat]->
patvalue[i][j];
}
}
}
}

}while((key !='\r')&&(readm!=1));
settextjustify(LEFT_TEXT,TOP_TEXT);
Closewindow((getmaxx()/2)-100,(getmaxy()/2)-200);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Screen();
}

void Save(void)
{
    char name[12];
    char key=",keyl=";
    int var=0,numpat=0;
    int oldtextcolor,oldx;
    int addx,addy;

    Openwindow((getmaxx()/2)-100,(getmaxy()/2)-200,(getmaxx()/2)+100,(getmaxy()/2)+200,
DARKGRAY);
    setcolor(WHITE);
    delay(500);
    Hidemouse();
    setfillstyle(SOLID_FILL,BLUE);
    bar(131,4,253,23);
    gprintf(155,10," F2-SAVE ");
    Showmouse();
    if((pat=fopen("pattern.sav","r+b"))==NULL){
        fprintf(stderr,"No file pattern.sav");
        exit(0);
    }
    for(i=0;i<10;i++)
        fread(pattern[i],sizeof(struct Patfile),1,pat);
    fclose(pat);
    ScreenLS("SAVE",pattern);
    setttextjustify(CENTER_TEXT.TOP_TEXT);
    do{
        if(bioskey(1)!=0){
            while(key != ESC){
                key=getch();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(key==0)
    key=getch();

if(key==DOWN){
    numpat++;
    if(numpat>9){
        numpat=0;

        Hidemouse();
        Box((getmaxx()/2)-80,(getmaxy()/2)-155+(
(numpat*35),160,20,WHITE);
        setfillstyle(SOLID_FILL,BROWN);
        bar((getmaxx()/2)-80,(getmaxy()/2)-155+(
(numpat*35),(getmaxx()/2)+80,(getmaxy()/2)-135+(numpat*35));
        gprintf(getmaxx()/2,(getmaxy()/2)-148+(
(numpat*35),"%s",pattern[numpat]->patname);
        Box((getmaxx()/2)-80,(getmaxy()/2)-155+(9*35),
160,20,WHITE);
        setfillstyle(SOLID_FILL,LIGHTGRAY);
        bar((getmaxx()/2)-80,(getmaxy()/2)-155+(9*35),
(getmaxx()/2)+80,(getmaxy()/2)-135+(9*35));
        gprintf(getmaxx()/2,(getmaxy()/2)-148+(9*35),"%
s",pattern[9]->patname);

        Showmouse();
    }
    else{
        Hidemouse();
        Box((getmaxx()/2)-80,(getmaxy()/2)-155+(
(numpat*35),160,20,WHITE);
        setfillstyle(SOLID_FILL,BROWN);
        bar((getmaxx()/2)-80,(getmaxy()/2)-155+(
(numpat*35),(getmaxx()/2)+80,(getmaxy()/2)-135+(numpat*35));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gprintf(getmaxx()/2,(getmaxy()/2)-148+
(numpat*35),"%s",pattern[numpat]->patname);

Box((getmaxx()/2)-80,(getmaxy()/2)-155+((numpat-
1)*35),160,20,WHITE);

setfillstyle(SOLID_FILL,LIGHTGRAY);
bar((getmaxx()/2)-80,(getmaxy()/2)-155+((numpat-
1)*35),(getmaxx()/2)+80,(getmaxy()/2)-135+((numpat-1)*35));
gprintf(getmaxx()/2,(getmaxy()/2)-148+((numpat-1)
*35),"%s",pattern[numpat-1]->patname);

Showmouse();
}
if(key==UP){
numpat--;
if(numpat<0){
numpat=9;

Hidemouse();
Box((getmaxx()/2)-80,(getmaxy()/2)-155+
(numpat*35),160,20,WHITE);
setfillstyle(SOLID_FILL,BROWN);
bar((getmaxx()/2)-80,(getmaxy()/2)-155+
(numpat*35),(getmaxx()/2)+80,(getmaxy()/2)-135+(numpat*35));
gprintf(getmaxx()/2,(getmaxy()/2)-148+
(numpat*35),"%s",pattern[numpat]->patname);

Box((getmaxx()/2)-80,(getmaxy()/2)-155,160,20,
WHITE);

setfillstyle(SOLID_FILL,LIGHTGRAY);
bar((getmaxx()/2)-80,(getmaxy()/2)-155,(getmaxx()
/2)+80,(getmaxy()/2)-135);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gprintf(getmaxx()/2,(getmaxy()/2)-148,"%s",pattern
[0]->patname);

Showmouse();
}
else{
Hidemouse();
Box((getmaxx()/2)-80,(getmaxy()/2)-155+
(numpat*35),160,20,WHITE);
setfillstyle(SOLID_FILL,BROWN);
bar((getmaxx()/2)-80,(getmaxy()/2)-155+
(numpat*35),(getmaxx()/2)+80,(getmaxy()/2)-135+(numpat*35));
gprintf(getmaxx()/2,(getmaxy()/2)-148+
(numpat*35),"%s",pattern[numpat]->patname);
Box((getmaxx()/2)-80,(getmaxy()/2)-155+
((numpat+1)*35),160,20,WHITE);
setfillstyle(SOLID_FILL,LIGHTGRAY);
bar((getmaxx()/2)-80,(getmaxy()/2)-155+
((numpat+1)*35),(getmaxx()/2)+80,(getmaxy()/2)-135+((numpat+1)*35));
gprintf(getmaxx()/2,(getmaxy()/2)-148+((numpat+1)
*35),"%s",pattern[numpat+1]->patname);
Showmouse();
}
}
}

```

```

if(key=='\r'){
startx=getmaxx()/2+(5*textwidth("W"));
starty=getmaxy()/2-148+(numpat*35);
key1=Curser(startx,starty,BROWN,WHITE);
setcolor(WHITE);
settextjustify(LEFT_TEXT,TOP_TEXT);-

```

```

if(key!='\r'){

```

```

key=ESC;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(i=0;i<2;i++)
    for(j=0;j<10;j++)
        pattern[numpat]->patvalue[i][j]=
select[i][j]:
    }
else{
    startx=getmaxx()/2-39;
    starty=getmaxy()/2-148+(numpat*35);
    oldx=startx;
    Hidemouse();
    Box((getmaxx()/2)-80,(getmaxy()/2)-155+
(numpat*35),160,20,WHITE);
    setfillstyle(SOLID_FILL,BROWN);
    bar((getmaxx()/2)-80,(getmaxy()/2)-155+
(numpat*35),(getmaxx()/2)+80,(getmaxy()/2)-135+(numpat*35));
    if((key1>=97)&&(key1<=122))
        key1 = key1-32;
    if((key1!='\b')&&(key1!='ESC')){
        gprintf(startx,starty,"%c",key1);
        startx += startx+textwidth("W");
        name[0]=key1;
        var=1;
    }
    Showmouse();
do{
    name[var]=Curser(startx,starty,BROWN,WHITE);
    setcolor(WHITE):
    if((name[var]>=97)&&(name[var]<=122))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

name[var] = name[var]-32;

if(((name[var]>=65)&&(name[var]<=90))||
(name[var]=='\r')||(name[var]=='\b')||(name[var]>=48)&&(name[var]<=57))||(name[var]==' ')){
    if((name[var]!='\r')&&(name
[var]!='\b')){

gprintf(startx,starty,"%c",name[var]);

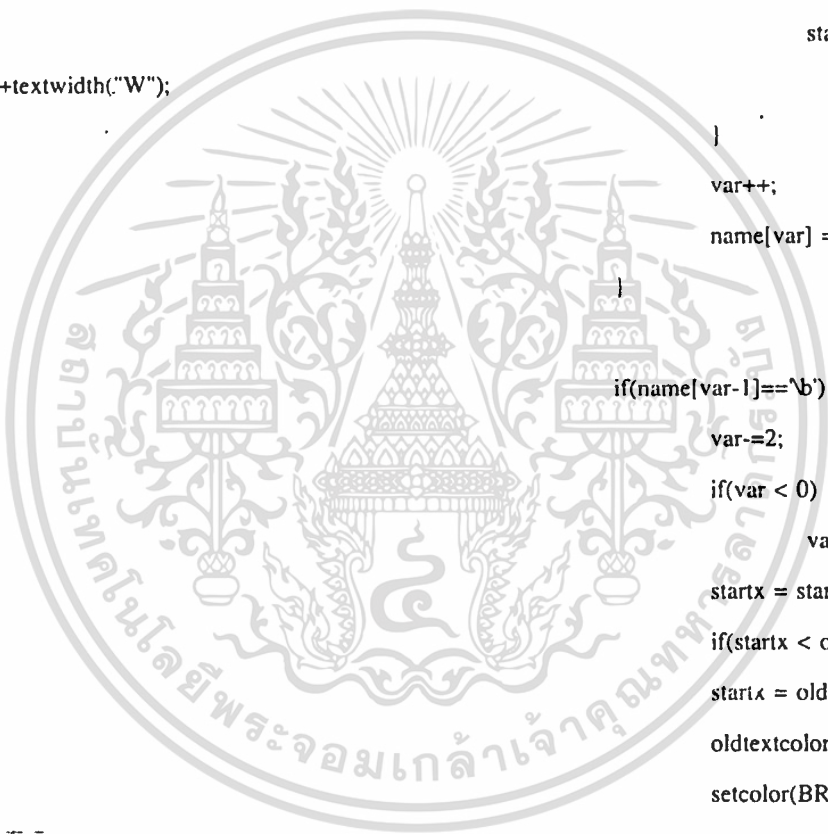
startx =
startx+textwidth("W");
}
var++;
name[var] = '\0';
if(name[var-1]=='\b'){
var-=2;
if(var < 0)
var = -1;
startx = startx-textwidth("W");
if(startx < oldx)
startx = oldx;
oldtextcolor=getcolor();
setcolor(BROWN);
gprintf(startx,starty, " ");
setcolor(oldtextcolor);
}

if(name[var-1]!='\r')
name[var-1]='\0';

}while((var!=10)&&(name[var-1]!='\0'));

key=ESC;

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

bar((getmaxx()/2)-80,(getmaxy()/2)-155+
(numpat*35),(getmaxx()/2)+80,(getmaxy()/2)-135+(numpat*35));
gprintf(getmaxx()/2,(getmaxy()/2)-148+
(numpat*35),"%s",pattern[numpat]->patname);

if(numpat != 0){
Box((getmaxx()/2)-80,(getmaxy()/2)-
155,160,20,WHITE);
setfillstyle(SOLID_FILL,LIGHTGRAY);
bar((getmaxx()/2)-80,(getmaxy()/2)-155,
(getmaxx()/2)+80,(getmaxy()/2)-135);
gprintf(getmaxx()/2,(getmaxy()/2)-148,"%
s",pattern[0]->patname);
}
Showmouse();
keyl=Curser(startx,starty,BROWN,WHITE);
setcolor(WHITE);
setttextjustify(LEFT_TEXT,TOP_TEXT);
if(keyl=='\r'){
key=ESC;
for(i=0;i<2;i++){
for(j=0;j<10;j++){
pattern[numpat]->
patvalue[i][j]=select[i][j];
}
else{
startx=getmaxx()/2-39;
starty=getmaxy()/2-148+(numpat*35);
oldx=startx;
Hidemouse();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Box((getmaxx()/2)-80,(getmaxy()/2)-155+
(numpat*35),160,20,WHITE);

setfillstyle(SOLID_FILL,BROWN);
bar((getmaxx()/2)-80,(getmaxy()/2)-155+
(numpat*35),(getmaxx()/2)+80,(getmaxy()/2)-135+(numpat*35));

```

```

if((key1>=97)&&(key1<=122))

```

```

    key1 = key1-32;

```

```

if((key1!='\b')&&(key1!=ESC)){

```

```

    gprintf(startx,starty,"%c",key1);

```

```

    startx = startx+textwidth("W");

```

```

    var=1;

```

```

    name[0]=key1;

```

```

}

```

```

    Showmouse();

```

```

do{

```

```

    name[var]=Curser(startx,starty,BROWN,WHITE);

```

```

    setcolor(WHITE);

```

```

    if((name[var]>=97)&&(name

```

```

[var]<=122))

```

```

        name[~var] = name[var]-

```

```

32;

```

```

    if(((name[var]>=65)&&(name

```

```

[var]<=90))||((name[var]=='\r')||((name[var]=='\b')||((name[var]>=48)&&(name[var]<=57))||((name[var]=='\n'))){

```

```

        if((name[var]!='\r')&&

```

```

;

```

```

(name[var]!='\b')){

```

```

            g p r i n t f

```

```

            (startx,starty,"%c",name[var]);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
startx+textwidth("W");
```

```
("W");
```

```
patname[j]=name[j];
```

```
patvalue[i][j]=select[i][j];
```

```
startx =
```

```
}
```

```
var++;
```

```
name[var] = '\0';
```

```
if(name[var-1]=='\b'){
```

```
var-=2;
```

```
if(var < 0)
```

```
var = -1;
```

```
startx = startx-textwidth
```

```
if(startx < oldx)
```

```
startx = oldx;
```

```
oldtextcolor=getcolor();
```

```
setcolor(BROWN);
```

```
gprintf(startx,starty, " ");
```

```
setcolor(oldtextcolor);
```

```
if(name[var-1]=='\r')
```

```
name[var-1]='\0';
```

```
}while((var!=10)&&(name[var-1]!='\0'));
```

```
for(i=0;i<2;i++)
```

```
for(j=0;j<10;j++){
```

```
pattern[numpat]->
```

```
pattern[numpat]->
```

```
name[var]='\0';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

key=ESC;
}
}
}
}while(key != ESC);
settextjustify(LEFT_TEXT, TOP_TEXT);
Closewindow((getmaxx()/2)-100, (getmaxy()/2)-200);
if((pat=fopen("pattern.sav", "r+b"))==NULL){
    fprintf(stderr, "No file pattern.sav");
    exit(0);
}
for(i=0; i<10; i++){
    fwrite(pattern[i], sizeof(struct Patfile), 1, pat);
}
fclose(pat);
}
void Auto(void)
{
    int s, c;
    unsigned char temp[5].Average;

    Openwindow((getmaxx()/2)-100, (getmaxy()/2)-15, (getmaxx()/2)+100, (getmaxy()/2)+15, 7);
    setcolor(WHITE);
    gprintf((getmaxx()/2)-72, (getmaxy()/2)-4, "!! Please wait !!");
    delay(500);
    Hidemouse();
    setfillstyle(SOLID_FILL, BLUE);
    bar(258, 4, 380, 23);
    gprintf(282, 10, " F3-AUTO ");
    Showmouse();
    delay(4500);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

do{
    for(i=0;i<10;i++){
        for(j=0;j<5;j++){
            outportb(0x301,i);
            outportb(0x304,0);
            delay(0.2);
            temp[j] = inportb(0x304);
        }

        for(j=0;j<5;j++){
            outportb(0x301,i);
            outportb(0x304,0);
            delay(0.2);
            temp[j] = inportb(0x304);
        }
    }while(0);
Closewindow(((getmaxx()/2)-100,((getmaxy()/2)-15));
}

void Analyzer(void)
{
    char ch;
    int s,c;
    unsigned char nled[2][10]={{12,13,10,11,12,15,18,17,12,9},{10,9,16,13,16,14,19,16,12,18}};
    unsigned char oled[2][10]={{15,16,14,13,18,18,22,21,15,13},{15,14,21,17,22,20,23,19,17,20}};
    unsigned char temp[5],MAX;
    CloseGraph();
    InitGraph();
    Screen_Ana();
    Showmouse();
}

```

```
do{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(bioskey(1)!= 0){
    ch=getch();
    if(ch=='\n'){
        CloseGraph();
        Equalizer();
    }
}
else{
    readm=Readmouse();
    if(readm==0x0001){
        CloseGraph();
        Equalizer();
    }
}
for(s=0;s<2;s++){
    for(c=0;c<10;c++){
        oled[s][c] = nled[s][c];
    }
    for(i=0;i<10;i++){
        for(j=0;j<5;j++){
            outputb(0x301,i);
            outputb(0x304,0);
            delay(0.2);
            temp[j] = inportb(0x304);
            convert(temp,temp);
            MAX = FindMax(temp);
        }
    }
    nled[0][i] = MAX;
}
for(j=0;j<5;j++){
    outputb(0x301,i);
    outputb(0x304,0);
    delay(0.2);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

temp[j] = inportb(0x304);
convert(temp.temp);
MAX = FindMax(temp);
}
nled[1][i] = MAX;
}

```

```

for(s=0;s<2;s++)
    for(c=0;c<10;c++)
        Writeled(c,s,nled[s][c]);
delay(500);
for(s=0;s<2;s++)
    for(c=0;c<10;c++)
        Clearled(c,s,nled[s][c]);
delay(1);
for(s=0;s<2;s++)
    for(c=0;c<10;c++)
        Writeled(c,s,oled[s][c]);
delay(500);
for(s=0;s<2;s++)
    for(c=0;c<10;c++)
        Clearled(c,s,oled[s][c]);
delay(1);
}while(1);
}

```

```

void Writeled(int ch,int side,int loud)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int lo;

Hidemouse();
if(loud<=15){
    setfillstyle(1,GREEN);
    for(lo=1;lo<=loud; lo++){
        if(side==0)
            bar(30+60*ch,254-8*lo,60+60*ch,258-8*lo);
        else
            bar(30+60*ch,467-8*lo,60+60*ch,471-8*lo);
    }
}
else{
    if(loud>15&&loud<22){
        setfillstyle(1,GREEN);
        for(lo=1;lo<=15; lo++){
            if(side==0)
                bar(30+60*ch,254-8*lo,60+60*ch,258-8*lo);
            else
                bar(30+60*ch,467-8*lo,60+60*ch,471-8*lo);
        }
        setfillstyle(1,YELLOW);
        for(lo=16;lo<=loud; lo++){
            if(side==0)
                bar(30+60*ch,254-8*lo,60+60*ch,258-8*lo);
            else
                bar(30+60*ch,467-8*lo,60+60*ch,471-8*lo);
        }
    }
}
else{
    setfillstyle(1,GREEN);
    for(lo=1;lo<=15; lo++){
        if(side==0)
            bar(30+60*ch,254-8*lo,60+60*ch,258-8*lo);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
    bar(30+60*ch,467-8*lo,60+60*ch,471-8*lo);
}
setfillstyle(1,YELLOW);
for(lo=16;lo<22; lo++){
    if(side==0)
        bar(30+60*ch,254-8*lo,60+60*ch,258-8*lo);
    else
        bar(30+60*ch,467-8*lo,60+60*ch,471-8*lo);
}
setfillstyle(1,RED);
for(lo=22;lo<=loud; lo++){
    if(side==0)
        bar(30+60*ch,254-8*lo,60+60*ch,258-8*lo);
    else
        bar(30+60*ch,467-8*lo,60+60*ch,471-8*lo);
}
Showmouse();
}

void Screen_Ana(void)
{
    Hidemouse();
    setbkcolor(BLUE);
    setcolor(7);
    for(j=0;j<4;j++){
        Case(j,j);
        line(0,24+j,getmaxx(),24+j);
    }
    setcolor(WHITE);
    gprintf(getmaxx()/2-32,10,"ANALYZER");
    for (i=0;i<10;i++){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for (j=0;j<25;j++){
    Box(29+60*i,245-8*j,32,6,WHITE);
    Box(29+60*i,458-8*j,32,6,WHITE);
}
setcolor(7);
gprintf(30+60*i,43 .frequency[i]);
gprintf(30+60*i,255.frequency[i]);

setcolor(WHITE);

gprintf(5+(i*60),265,"-12");
gprintf(5+(i*60),52,"-12");
gprintf(5+(i*60),458,"-12");
gprintf(5+(i*60),245,"-12");
gprintf(20+(i*60),362,"0");
gprintf(20+(i*60),149,"0");
}
Showmouse();
}

void Cleared(int ch,int side,int loud)
{
    int lo;

    Hidemouse();
    setfillstyle(1,BLACK);
    for(lo=1;lo<=loud; lo++){
        if(side==0)
            bar(30+60*ch,254-8*lo,60+60*ch,258-8*lo);
        else
            bar(30+60*ch,467-8*lo,60+60*ch,471-8*lo);
    }
    Showmouse();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void InitGraph(void)
{
    int graphdriver=DETECT,graphmode;
    int ErrorCode;

    ErrorCode=registerbgidriver(EGAVGA_driver);
    if(ErrorCode<0){
        printf("Graphics error %s\n",grapherrormsg(ErrorCode));
        Exit();
    }
    initgraph(&graphdriver,&graphmode,"");
    ErrorCode=graphresult();
    if(ErrorCode!=grOk){
        printf("Error while opening \n");
        Exit();
    }
}

void CloseGraph(void)
{
    closegraph();
}

int Resetmouse(void)
{
    r.x.ax=0x0000;
    int86(0x33,&r,&r);
    return(r.x.ax);
}

void Showmouse(void)
{
    r.x.ax=0x0001;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        int86(0x33,&r,&r);
    }

void Hidemouse(void)
{
    r.x.ax=0x0002;
    int86(0x33,&r,&r);
}

int Readmouse(void)
{
    r.x.ax=0x0003;
    int86(0x33,&r,&r);
    return (r.x.bx);
}

void Openwindow(int left,int top,int right,int bottom,int bkcolor)
{
    int size;

    size = imagesize(left,top,right,bottom);
    if((back=malloc(size)) != NULL){
        Hidemouse();
        getimage(left,top,right,bottom,back);
        setwritemode(COPY_PUT);
        setfillstyle(SOLID_FILL,bkcolor);
        bar(left,top,right,bottom);
        setcolor(WHITE);
        setlinestyle(SOLID_LINE,0,THICK_WIDTH);
        rectangle(left+1,top+1,right-1,bottom-1);
        Showmouse();
    }
}

```

```

void Closewindow(int left,int top)
{
    Hidemouse();
    if(back != NULL){
        putimage(left,top,back,COPY_PUT);
        free(back);
    }
    Showmouse();
    setlinestyle(SOLID_LINE,0,NORM_WIDTH);
}

```

```

int gprintf(int xloc,int yloc,char *fmt, ... )

```

```

{
    va_list argptr;
    char str[140];
    int cnt;

    va_start(argptr,format);
    cnt = vsprintf(str,fmt,argptr);
    outtextxy(xloc,yloc,str);
    va_end(argptr);
    return(cnt);
}

```

```

void ScreenLS(char *menu,struct Patfile *N[10])

```

```

{
    int num;

    Hidemouse();
    setttextjustify(CENTER_TEXT,TOP_TEXT);
    gprintf((getmaxx()/2),(getmaxy()/2)-185,"%s,PATTERN",menu);
    for(num=0;num<10;num++){
        N[num]->patname[10]='\0';
        Box((getmaxx()/2)-80,(getmaxy()/2)-155+(num*35),160,20,WHITE);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        setfillstyle(SOLID_FILL,LIGHTGRAY);
        bar((getmaxx()/2)-80,(getmaxy()/2)-155+(num*35),(getmaxx()/2)+80,(getmaxy()/2)-135+
(num*35));

        gprintf(getmaxx()/2,(getmaxy()/2)-148+(num*35),"%s",N[num]->patname);
    }

    Box((getmaxx()/2)-80,(getmaxy()/2)-155,160,20,WHITE);
    setfillstyle(SOLID_FILL,BROWN);
    bar((getmaxx()/2)-80,(getmaxy()/2)-155,(getmaxx()/2)+80,(getmaxy()/2)-135);
    gprintf(getmaxx()/2,(getmaxy()/2)-148,"%s",N[0]->patname);
    setttextjustify(LEFT_TEXT, TOP_TEXT);
    Showmouse();
}

char Curser(int x,int y,int bkcolor,int textcolor)
{
    char logic,keyp;

    do{
        setcolor(textcolor);
        gprintf(x,y,"_");
        delay(200);
        setcolor(bkcolor);
        gprintf(x,y,"_");
        delay(200);

        if(bioskey(1))
            logic=0;
        else
            logic=1;
    }while(logic);

    keyp=getch();
    return(keyp);
}

```

int CheckX(int ax)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if(ax >= 40 && ax <= 60) return 0;
    if(ax >= 100 && ax <= 120) return 1;
    if(ax >= 160 && ax <= 180) return 2;
    if(ax >= 220 && ax <= 240) return 3;
    if(ax >= 280 && ax <= 300) return 4;
    if(ax >= 340 && ax <= 360) return 5;
    if(ax >= 400 && ax <= 420) return 6;
    if(ax >= 460 && ax <= 480) return 7;
    if(ax >= 520 && ax <= 540) return 8;
    if(ax >= 580 && ax <= 600) return 9;
    return -1;
}

int CheckMenu(int ax)
{
    if(ax >= 4 && ax <= 126) return 0;
    if(ax >= 131 && ax <= 253) return 1;
    if(ax >= 258 && ax <= 380) return 2;
    if(ax >= 385 && ax <= 507) return 3;
    if(ax >= 512 && ax <= 634) return 4;
    return -1;
}

```

```

int Checknum(int ay)
{
    if((ay >= 85) && (ay <= 105)) return 0;
    if((ay >= 120) && (ay <= 140)) return 1;
    if((ay >= 155) && (ay <= 175)) return 2;
    if((ay >= 190) && (ay <= 210)) return 3;
    if((ay >= 225) && (ay <= 245)) return 4;
    if((ay >= 260) && (ay <= 280)) return 5;
    if((ay >= 295) && (ay <= 315)) return 6;
    if((ay >= 330) && (ay <= 350)) return 7;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if((ay >= 365)&&(ay <= 385))return 8;
        if((ay >= 400)&&(ay <= 420))return 9;
        return -1;
    }

void Exit(void)
{
    for(i=0;i<10;i++)
        free(pattern[i]);
    Hidemouse();
    CloseGraph();
    exit(0);
}

void Outport(unsigned port,int side,int Ch)
{
    int address;
    unsigned char channel;
    unsigned char vout[13]={0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};

    if(side==0)
        channel=(unsigned char)(Ch+1);
    else
        channel=(unsigned char)((Ch+1)%10);
    address=select[side][Ch];
    channel=channel%80;
    outportb(port,channel);
    outportb(port,(channel & 0x7f));
    delay(0.1);
    if(address<0){
        address=(-1)*address;
        vout[address]=vout[address]|0x40;
    }

    outportb(port,vout[address]);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

unsigned char FindMax(unsigned char t[5])
{
    unsigned char max;

    max = t[0];
    for(i=1;i<5;i++){
        if(max <= t[i])
            max = t[i];
    }
    return(max);
}

void convert(unsigned char new[5],unsigned char old[5])
{
    int a;
    for(a=0;a<5;a++){
        if(old[a] < 0x80)
            new[a] = !old[a];
    }
}

```

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สำเร็จได้ด้วยดี ด้วยได้รับความช่วยเหลือจากผู้มีพระคุณต่าง ๆ คณะผู้จัดทำจึงขอแสดงความขอบคุณทุกท่าน อันได้แก่ บิดามารดาที่ให้กำเนิด , ศ.ดร.วัลลภ สุระกำพลธ อาจารย์ที่ปรึกษา , อาจารย์ภาควิชาอิเล็กทรอนิกส์และรุ่นพี่ทุกท่านที่ให้คำแนะนำ และรุ่นน้องที่ให้ความช่วยเหลือ ปริญญานิพนธ์นี้ได้รับความช่วยเหลือจากท่านด้วยดี ทางคณะผู้จัดทำจึงขอขอบคุณมา ณ ที่นี้

ผู้จัดทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. ทีมงานอีทีที , “ET-DIO การ์ดอินพุทเอาต์พุตเนกประสงค์” ,วารสารเซมิคอนดักเตอร์อิเล็กทรอนิกส์ ฉบับที่ 141
1. ชานินทร์ ถาวรศาสนวงศ์ , ทินกร ตึก ,” การอินเทอร์เฟซ IBM PC ”,ฟิสิกส์เซ็นเตอร์การพิมพ์,หน้า 43-51
2. ธันวา ศรีประโมง, ” การเขียนโปรแกรมภาษาซี สำหรับวิศวกรรม “ มหาวิทยาลัยเทคโนโลยีมหานคร, 739 หน้า , 2537
3. Gobind Daryanani, ” Principles of Active Network Synthesis and Design” , John Wiley&Sons,495 p.,1976.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้