สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การสื่อสารสัญญาณคลื่นไฟฟ้าหัวใจผ่านโครงข่ายโทรศัพท์

ELECTROCARDIOGRAM COMMUNICATION VIA TELEPHONE

T050128

โดย

นายเกรียงไกร วระพันธุ์

นายเฉลิมศักดิ์ วงศ์โพธิ์ทอง

ปริญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....
เลขทะเบียน 50128
วัน,เดือน,ปี 21 เม.ย. 2547
b.....
i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารสัญญาณคลื่นไฟฟ้าหัวใจผ่านโครงข่ายโทรศัพท์

ELECTROCARDIOGRAM COMMUNICATION VIA TELEPHONE

โดย

นายเกรียงไกร วระพันธุ์ 42010027

นายเฉลิมศักดิ์ วงศ์โพธิ์ทอง 42010069

อาจารย์ที่ปรึกษา

รศ.ดร. สุวิพล สิทธิชีวภาค

ผศ. เกรียงไกร วงศ์โรจนภรณ์

ปริญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **การสื่อสารสัญญาณคลื่นไฟฟ้าหัวใจผ่านโครงข่ายโทรศัพท์**

ELETROCARDIOGRAM COMMUNICATION VIA TELEPHONE

ผู้จัดทำ

1. นาย เกรียงไกร วระพันธุ์ **42010027**
2. นาย เฉลิมศักดิ์ วงศ์โพธิ์ทอง **42010069**

..... อาจารย์ที่ปรึกษา

(รศ.ดร. สุวิพล สิทธิชีวภาค)

..... อาจารย์ที่ปรึกษา

(ผศ. เกรียงไกร วงศ์โรจนภรณ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารสัญญาณคลื่นไฟฟ้าหัวใจผ่านโครงข่ายโทรศัพท์

ELECTROCARDIOGRAM COMMUNICATION VIA TELEPHONE

โดย นายเกรียงไกร วระพันธุ์ 42010027

นายเฉลิมศักดิ์ วงศ์โพธิ์ทอง 42010069

อาจารย์ที่ปรึกษา รศ.ดร. สุวิพล สิทธิชีวภาค

ผศ. เกรียงไกร วงศ์โรจนภรณ์

บทคัดย่อ

โครงงานนี้นำเสนอการสื่อสารสัญญาณคลื่นหัวใจผ่านโครงข่ายโทรศัพท์ ซึ่งจะใช้อุปกรณ์ที่ทำหน้าที่ขยายสัญญาณ ซึ่งประกอบด้วย วงจรขยายความแตกต่าง วงจรกรองความถี่ และวงจรขยายสัญญาณ และใช้การแปลงสัญญาณอนาลอกเป็นดิจิตอล เพื่อแปลงสัญญาณนั้นให้เป็นข้อมูลดิจิตอล ส่วนภาคการแสดงผลและส่งสัญญาณผ่านโครงข่ายโทรศัพท์ จะใช้การเขียนโปรแกรมเป็นตัวควบคุม

ABSTRACT

This project, Electrocardiogram (ECG) Communication via Telephone, uses the ECG receiver which consists of a differential amplifier, low pass filter and amplifiers. An analog to digital converter card is used to convert the ECG signal to digital signal. Finally we write a program to control displaying and transferring the ECG signal between computers via the network.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทที่ 1 บทนำ 1

บทที่ 2 ทฤษฎีหรือหลักการ 2

2.1 คลื่นไฟฟ้าหัวใจ 2

2.2 หลักการของการวัดคลื่นไฟฟ้าหัวใจ 4

2.3 ออปเปอร์เรชันแนล แอมพลิฟายเออร์ 10

2.4 วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล 18

2.5 การรับส่งข้อมูลจากไมโครคอมพิวเตอร์ 20

2.6 การเขียนโปรแกรมให้ไมโครคอมพิวเตอร์ติดต่อกับพอร์ทสื่อสาร 23

2.7 วงจรตรวจจับกรอบสัญญาณ 31

บบที่ 3 การคำนวณและการสร้าง 33

3.1 บล็อคไดอะแกรม 33

3.2 เครื่องรับสัญญาณคลื่นไฟฟ้าหัวใจ 33

3.3 การออกแบบโปรแกรม 39

บทที่ 4 การทดลองและผลการทดลอง 43

4.1 วงจรดิฟเฟอร์เรนเชียลแอมพลิฟลายเออร์ 43

4.2 วงจรกรองความถี่ต่ำแบบนอทชฟิวเตอร์ 45

4.3 วงจรกรองความถี่ต่ำผ่าน 46

4.4 เอาท์พุทวงจรรวม 46

4.5 เอาท์พุทเมื่อใช้สัญญาณ ECG เป็นอินพุท 47

บทที่ 5 บทวิจารณ์และบทสรุป 53

ภาคผนวก

กิตติกรรมประกาศ

หนังสืออ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้า

รูปที่ 2.1รูปหัวใจและลักษณะของคลื่นหัวใจ 2

รูปที่ 2.2 องค์ประกอบต่างๆ ของคลื่นไฟฟ้าหัวใจ 3

รูปที่ 2.3 ความสัมพันธ์ของความต่างศักดาไฟฟ้าบวก ลบกับแรงดันเวลา ในคลื่นไฟฟ้าหัวใจ 4

รูปที่ 2.4 แสดงตำแหน่งการวัด Bipolar Standard Leads 5

รูปที่ 2.5 แสดงตำแหน่งการวัด Unipolar Limb Leads 5

รูปที่ 2.6 แสดงตำแหน่งการวัด Unipolar chest leads 6

รูปที่ 2.7 แสดง LEAD V1-V6 6

รูปที่ 2.8 แสดงตำแหน่งการวัดคลื่นไฟฟ้าหัวใจทั้ง12 LEAD มาตรฐาน 7

รูปที่ 2.9 แสดงสัญญาณของ LEAD I,LEAD II,LEAD III 8

รูปที่ 2.10 แสดงสัญญาณของ LEAD aVR,LEAD aVL,LEAD aVFและUnipolar chest leads 9

รูปที่ 2.11 วงจรขยายแบบกลับ 10

รูปที่ 2.12 วงจรขยายความแตกต่าง 11

รูปที่ 2.13 การหาค่าของ Common Mode Gain 12

รูปที่ 2.14 การหาค่าของ Differential Mode Gain 12

รูปที่ 2.15 การตอบสนองความถี่ของวงจรกรองความถี่ต่ำ 13

รูปที่ 2.16 วงจรกรองความถี่ต่ำ 14

รูปที่ 2.17 วงจรกรองความถี่ต่ำแบบ Sallen and Key 15

รูปที่ 2.18 การตอบสนองความถี่ของวงจรนอท์ชฟิลเตอร์ 16

รูปที่ 2.19 วงจรนอท์ชฟิลเตอร์พื้นฐาน 16

รูปที่ 2.20 แสดงตำแหน่งของโพล , ซีโร่ และการสูญเสียในย่านความถี่ของวงจรกรองความถี่แบบนอท์ชฟิวเตอร์ 17

รูปที่ 2.21 แสดงตำแหน่งของโพล,ซีโร่และการสูญเสีย ในย่านความถี่ต่างๆ 17

รูปที่ 2.22 แสดงบล็อกไดอะแกรม Successive approximation ADC Converter 18

รูปที่ 2.23 วิธีการ Successive approximation 19

รูปที่ 2.24 แสดงการแปลงแบบ Successive approximation ที่ใช้ในวงจร sample and hold 19

รูปที่ 2.25 การเชื่อมติดต่อระหว่างคอมพิวเตอร์ด้วยกันผ่านทางสายโทรศัพท์ 20

รูปที่ 2.26 แสดงการเชื่อมต่อสายสัญญาณระหว่างCPUของเครื่องคอมพิวเตอร์กับพอร์ตอนุกรม 20

รูปที่ 2.27 การเชื่อมต่อระหว่างอุปกรณ์ DTE กับไมโครคอนโทรลเลอร์ 22

รูปที่ 2.28 การทำงานของวงจร 31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้า

รูปที่ 3.1 บล็อกไดอะแกรมของโครงงาน 33

รูปที่ 3.2 บล็อกไดอะแกรมของเครื่องรับสัญญาณไฟฟ้าหัวใจ 33

รูปที่ 3.33 วงจร วงจร Differential Amp 34

รูปที่ 3.41 รูปวงจร Non-Inverting Amplifier 34

รูปที่ 3.42 วงจรกรองความถี่ต่ำแบบบนอท์ชฟิลเตอร์ 35

รูปที่ 3.5 วงจรกรองความถี่ต่ำแบบแอคทีฟ 36

รูปที่ 3.6 รูปฮาร์ดแวร์ด้านหน้า 37

รูปที่ 3.7 รูปฮาร์ดแวร์ด้านหลัง 37

รูปที่ 3.8 รูปแสดงภายในส่วนของฮาร์ดแวร์ 38

รูปที่ 3.9 แสดงการทำงานของโปรแกรม ECG ตัวส่ง 39

รูปที่ 3.10 แสดงการทำงานของโปรแกรม ECG ที่ Form Main 39

รูปที่ 3.11 แสดงการทำงานในขั้นตอนDial Up 40

รูปที่ 3.12 แสดงการทำงานในขั้นตอน การ Save 40

รูปที่ 3.13 แสดงการทำงานของโปรแกรม ECG ตัวรับ 41

รูปที่ 3.14 แสดงการทำงานของโปรแกรม MCS-51 42

รูปที่ 3.15 แสดงการเชื่อมต่อของโครงงาน 42

รูปที่ 4.1 การหาคอมมอนโหมดเกนที่ความถี่ 150 Hz 43

รูปที่ 4.2 การหาดิฟเฟอร์เรนเชียลโหมดเกนที่ความถี่ 100 Hz 44

รูปที่ 4.3 ผลตอบสนองทางความถี่ของวงจรกรองความถี่ต่ำแบบบนอท์ชฟิลเตอร์ 45

รูปที่ 4.4 ผลตอบสนองทางความถี่ของวงจรกรองความถี่ต่ำ 46

รูปที่ 4.5 เอาท์พุทรวมของวงจร 46

รูปที่ 4.6 แสดงการใช้งานเครื่อง The Doppler 47

รูปที่ 4.7 รูปคลื่นหัวใจที่ได้จากเครื่อง The Doppler 47

รูปที่ 4.8 รูปสัญญาณหัวใจได้จากเครื่อง The Doppler 48

รูปที่ 4.9 รูปสัญญาณหัวใจได้จากเครื่อง The Doppler ผ่านวงจรตรวจจับสัญญาณ 48

รูปที่ 4.10 แสดงรูปโปรแกรมด้านส่ง 49

รูปที่ 4.11 รูปแสดงโปรแกรมส่วนการ Dial 49

รูปที่ 4.12 รูปแสดงโปรแกรมส่วนการ Setup Modem ทางด้านส่ง 50

รูปที่ 4.13 รูปแสดงโปรแกรมส่วนการ Save ทางด้านส่ง 50

รูปที่ 4.14 แสดงรูปโปรแกรมด้านรับ 51

รูปที่ 4.15 รูปแสดงโปรแกรมส่วนการ Setup Modem ทางด้านรับ 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้า

รูปที่ 4.16 รูปแสดงผลที่ด้านส่ง 52
รูปที่ 4.17 รูปแสดงผลที่ด้านรับ 52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 4.1 ตารางค่าคอมมอนโหมดเกน 43
ตารางที่ 4.2 ตารางค่าดิฟเฟอร์เรนเชียลโหมดเกน 44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1
บทนำ

ความก้าวหน้าทางด้านโครงข่ายโทรศัพท์ทำให้เราสามารถประยุกต์ใช้งานด้านต่างๆได้มากขึ้น จนในปัจจุบันสามารถจะกล่าวได้ว่าโทรศัพท์ถือเป็น เครื่องใช้ชนิดหนึ่งที่จำเป็นในการดำเนินชีวิต โดยโครงงานนี้เป็นการส่งสัญญาณคลื่นไฟฟ้าหัวใจ จากคอมพิวเตอร์ด้านต้นทางไปยังคอมพิวเตอร์ด้านปลายทาง โดยโมเดมผ่านโครงข่ายโทรศัพท์ซึ่งปัจจุบันสามารถครอบคลุมเกือบทุกพื้นที่ โครงงานนี้น่าจะก่อประโยชน์ต่อวงการแพทย์ได้บ้าง โดยเฉพาะอย่างยิ่งเมื่อผู้ป่วยและแพทย์ที่เชี่ยวชาญอยู่ห่างไกลกัน ซึ่งอุปกรณ์ชิ้นนี้สามารถอำนวยความสะดวกความยุ่งยากลำบากนี้ได้ เป็นอุปกรณ์ที่ลงทุนไม่มาก มีขนาดพอเหมาะ ใช้งานได้ง่ายไม่ยุ่งยาก

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

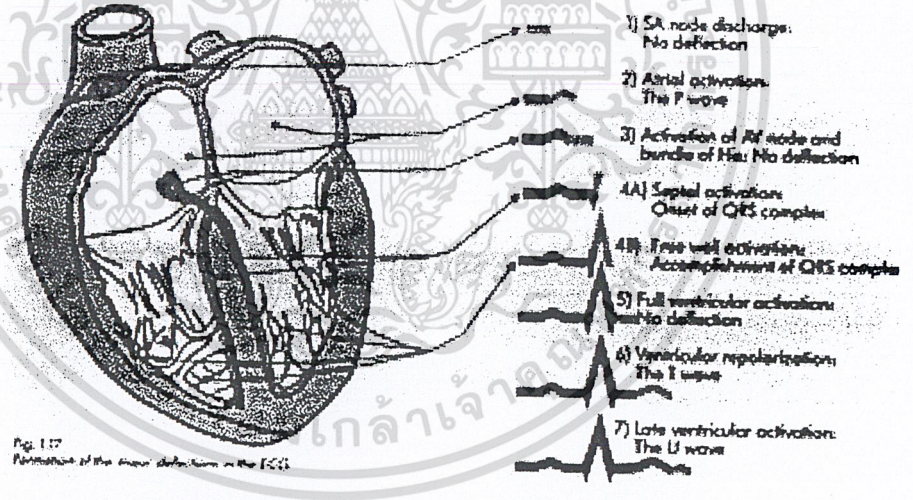
บทที่ 2
ทฤษฎีหรือหลักการ

2.1 คลื่นไฟฟ้าหัวใจ (Electrocardiogram - E.C.G.)

เซลล์ของหัวใจแบ่งได้ 3 ประเภทคือ

1. Pace maker cell เซลล์ที่ทำหน้าที่เป็นตัวก่อกำเนิด การเต้นของหัวใจ
2. Conducting cell เซลล์ที่เป็นตัวนำคลื่นไฟฟ้า
3. Myocardium เซลล์ที่มีหน้าที่บีบตัวเมื่อถูกกระตุ้น

Pace maker cell มีคุณสมบัติพิเศษ สามารถเกิดการเปลี่ยนแปลงศักดาไฟฟ้าในตัวเองได้เป็นรอบๆ โดยการเคลื่อนย้าย electrolyte คือ Na+ และ K- ผ่านผนังเซลล์ เริ่มต้นแต่ละรอบเป็นไซเคิล (cycle) ด้วยการที่ศักดาไฟฟ้าในเซลล์เพิ่มขึ้นอย่างทันทีทันใดเรียกว่า depolarization แล้วกลับลดไปที่ระดับเดิมใหม่เรียกว่า repolarization



รูปที่ 2.1 รูปหัวใจและลักษณะของคลื่นไฟฟ้าหัวใจ

การเปลี่ยนแปลงศักดาไฟฟ้านี้ จะถูกนำให้เกิดจากเซลล์หนึ่งไปยังอีกเซลล์หนึ่ง ต่อๆกันไปโดยผ่าน conducting cell ไปจนถึง myocardium ซึ่งจะหดตัว เมื่อเกิดการเปลี่ยนศักดาไฟฟ้าขึ้น การที่คลื่นของการเปลี่ยนศักดาไฟฟ้า ถูกเหนี่ยวนำให้เกิดต่อๆ กันไปในหัวใจนี้ ทำให้มีคลื่นไฟฟ้ารวมของหัวใจเกิดขึ้น และผ่านเซลล์ของร่างกายมายังผิวหนัง เมื่อเอาขั้วบวกหรือขั้วลบจาก Galvanometer ไฟฟ้าซึ่งเราเรียกว่า อิเล็คโทรด (electrode) ไปวางที่ผิวหนัง จึงสามารถรับคลื่นไฟฟ้านี้ได้ซึ่งเรียกว่า คลื่นไฟฟ้าหัวใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปแสวงหาผลประโยชน์ทางการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลื่นไฟฟ้าหัวใจ ที่เกิดขึ้นในแต่รอบ มีลักษณะเป็นคลื่นสูงๆ ต่ำๆ ซึ่งถูกเรียกชื่อ เพื่อความสะดวกดังนี้

P wave เป็นคลื่นแรกของแต่ละไซเคิลเกิดจากกล้ามเนื้อหัวใจห้องบนบีบตัว

QRS complex เกิดจากกล้ามเนื้อหัวใจห้องล่างบีบตัว

T wave เกิดจากกล้ามเนื้อหัวใจห้องล่างคลายตัว

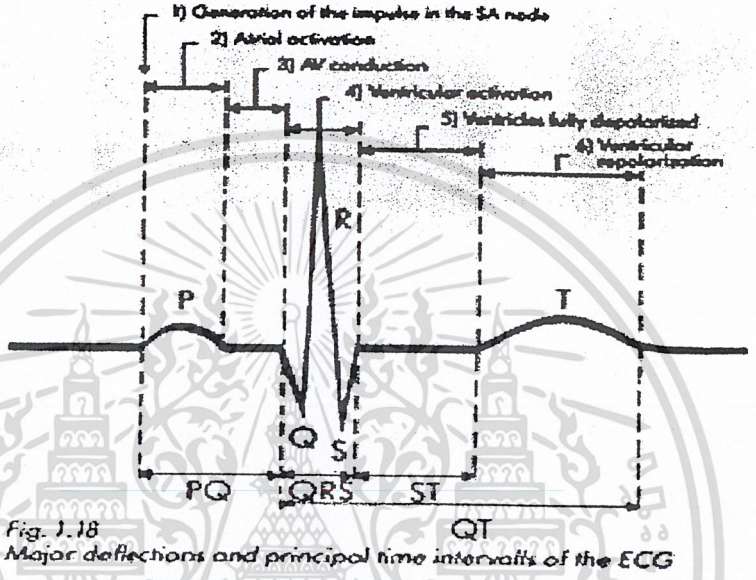


Fig. 1.18
Major deflections and principal time intervals of the ECG

รูปที่ 2.2 องค์ประกอบต่างๆ ของคลื่นไฟฟ้าหัวใจ

ช่วงเวลาต่าง ๆ ของคลื่นไฟฟ้าหัวใจ	ช่วงเวลาปกติ (วินาที)
ช่วงเวลาของ PR (P-R Interval)	0.12-0.20
ช่วงเวลาของ QT (Q-T Interval)	0.35-0.44
ช่วงเวลาของ ST (S-T Segment)	0.05-0.15
ช่วงเวลาของ P (P Wave Interval)	0.11
ช่วงเวลาของ QRS (QRS Interval)	0.09

ตารางที่ 2.1 แสดงช่วงเวลาต่าง ๆ ของคลื่นไฟฟ้าหัวใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 หลักการของการวัดคลื่นไฟฟ้าหัวใจ (Principle of electrocardiography)

เป็นการวัดความต่างศักดาไฟฟ้าที่เกิดจากคลื่นไฟฟ้าหัวใจ ระหว่างจุด 2 จุด บนผิวหนัง โดยอาศัยอุปกรณ์ต่างๆ ซึ่งประกอบด้วย

1. ขั้วไฟฟ้า (The electrode)
2. สายนำไฟฟ้า (The leads)
3. ภาคขยาย (The amplifier)
4. ภาคบันทึก (The recorder)
5. กระดาษบันทึก (The paper)

โดยอาศัยหลักการคือ

ถ้าความต่างศักดาไฟฟ้าเป็นบวก เส้นกราฟจะมีทิศทางขึ้นด้านบน

ถ้าความต่างศักดาไฟฟ้าเป็นลบ เส้นกราฟจะมีทิศทางลงล่าง

ถ้าความต่างศักดิ์เป็นศูนย์ กราฟจะเป็นเส้นตรง

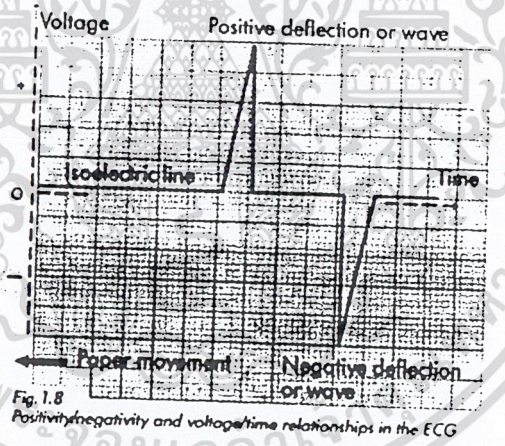


Fig. 1.8
Positivity/negativity and voltage/time relationships in the ECG

รูปที่ 2.3 ความสัมพันธ์ของความต่างศักดาไฟฟ้าบวก/ลบกับแรงดัน/เวลา ในคลื่นไฟฟ้าหัวใจ

การนับจากขณะเริ่มเกิด P wave ไปจนเริ่มเกิด QRS complex ซึ่งเรียกว่า PR interval ปกติไม่ควรนานกว่า 0.2 วินาที การเกิด QRS complex ตั้งแต่ต้นจนจบ ไม่ควรนานกว่า 0.12 วินาที ถ้านานกว่านั้นแสดงว่าผิดปกติ

การวัดคลื่นไฟฟ้าหัวใจ ประกอบด้วย lead มาตรฐาน 12 leads ได้แก่

-Bipolar Standard Leads จำนวน 3 leads : LEAD I, LEAD II, LEAD III

-Unipolar Limb Leads จำนวน 3 leads :LEAD aVR, LEAD aVL, LEAD aVF

-Unipolar Chest Leads จำนวน 3 leads :LEAD V1-V6

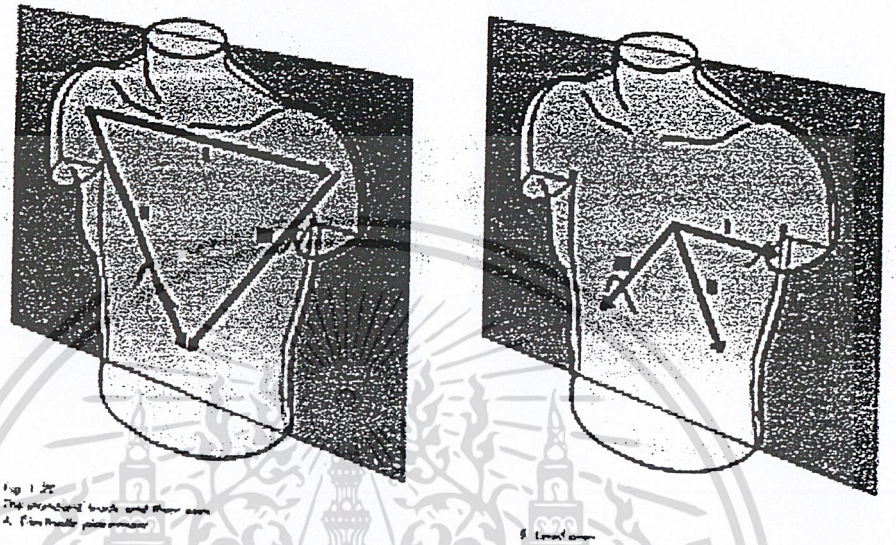
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bipolar Standard Leads

LEAD I = แขนซ้าย (ขั้วบวก) กับแขนขวา (ขั้วลบ)

LEAD II = ขาซ้าย (ขั้วบวก) กับแขนขวา (ขั้วลบ)

LEAD III = ขาซ้าย (ขั้วบวก) กับแขนซ้าย (ขั้วลบ)



รูปที่ 2.4 แสดงตำแหน่งการวัด Bipolar Standard Leads

Unipolar Limb Lead

aVR = แขนขวา

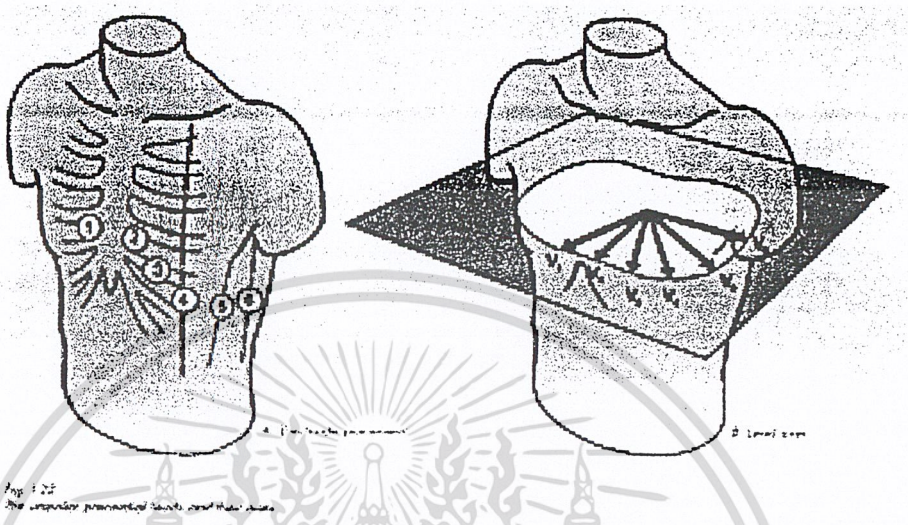
aVL = แขนซ้าย

aVF = ขาซ้าย

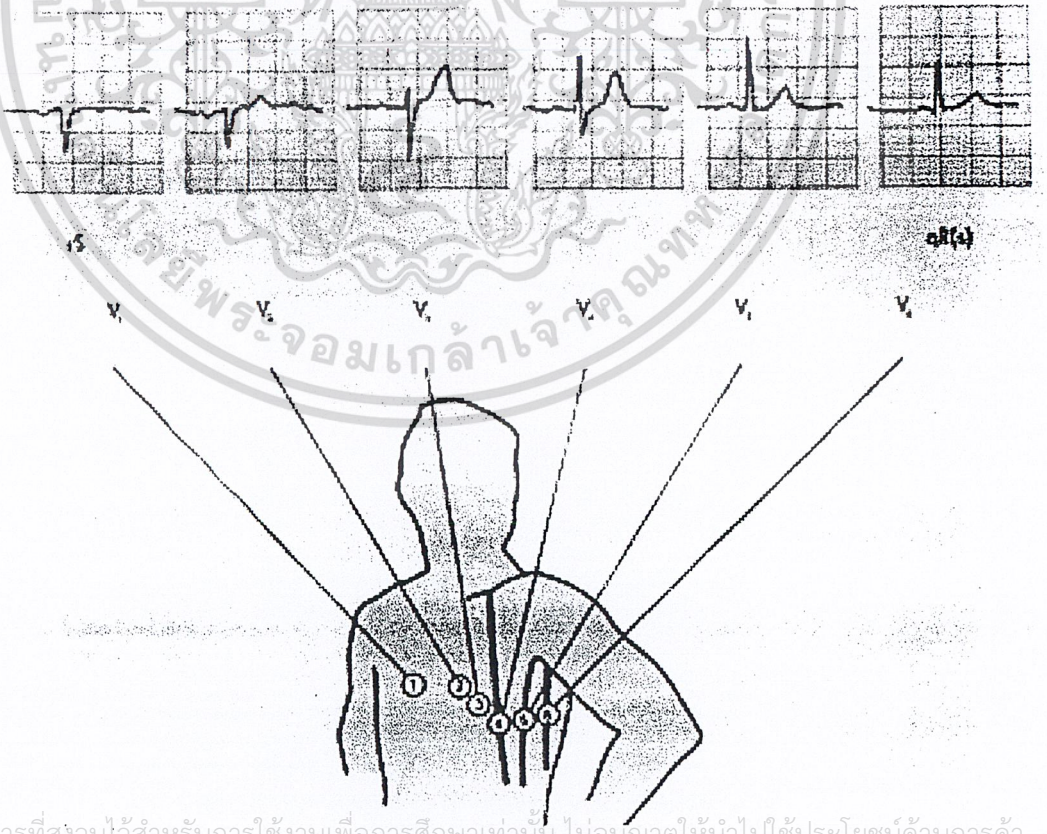
รูปที่ 2.5 แสดงตำแหน่งการวัด Unipolar Limb Leads

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Unipolar Chest Leads

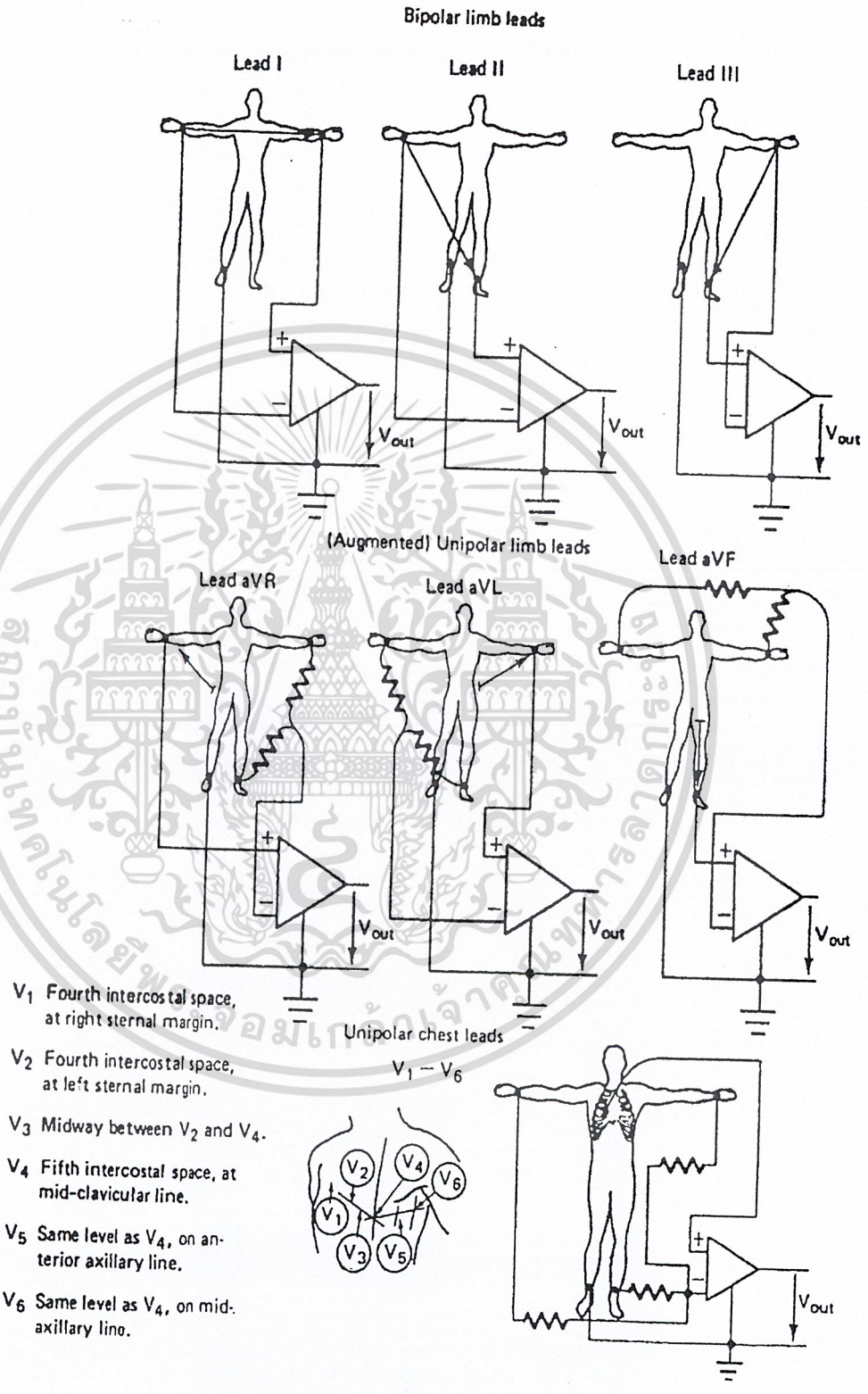


รูปที่ 2.6 แสดงตำแหน่งการวัด Unipolar Chest Leads



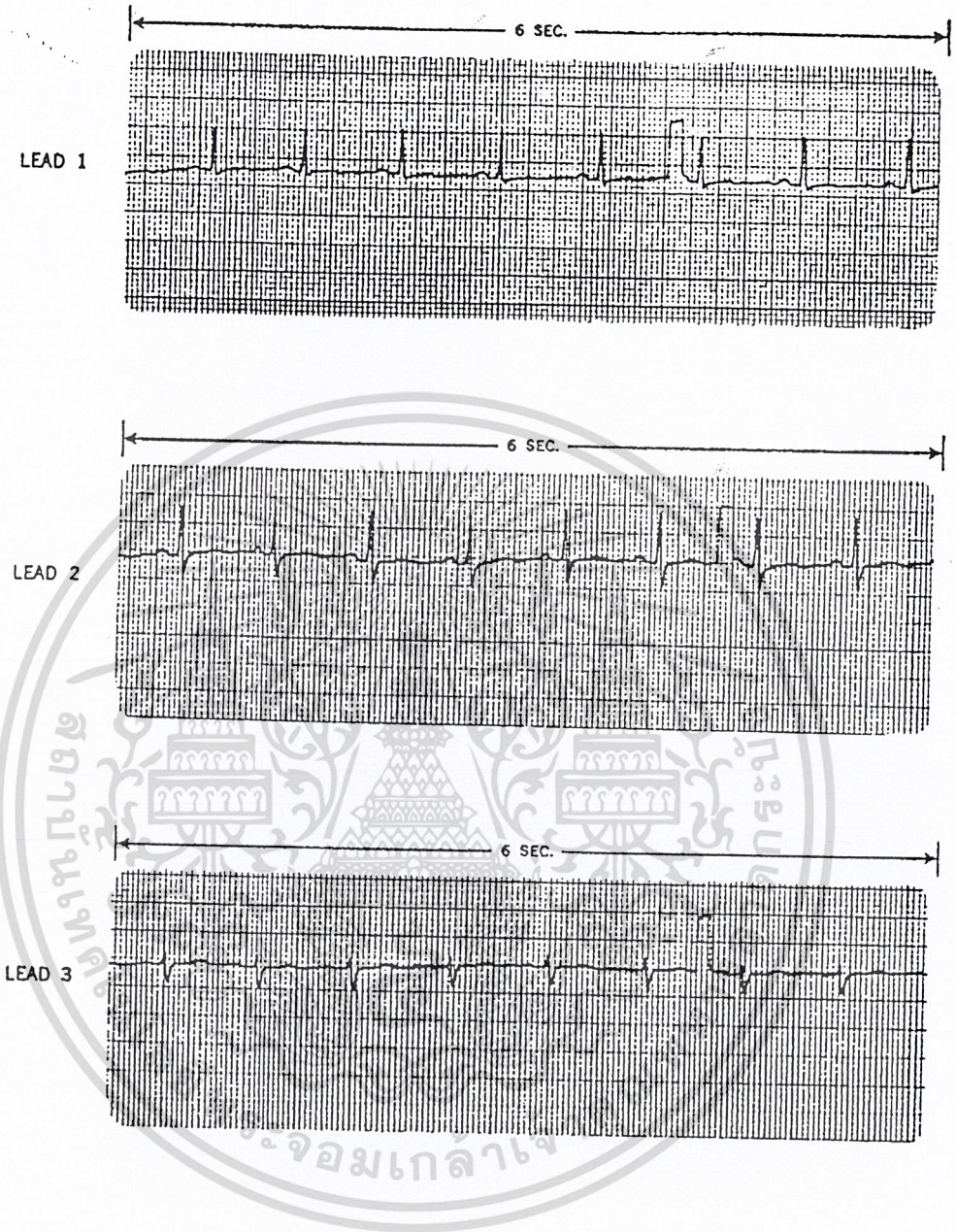
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.7 แสดง LEAD V1-V6



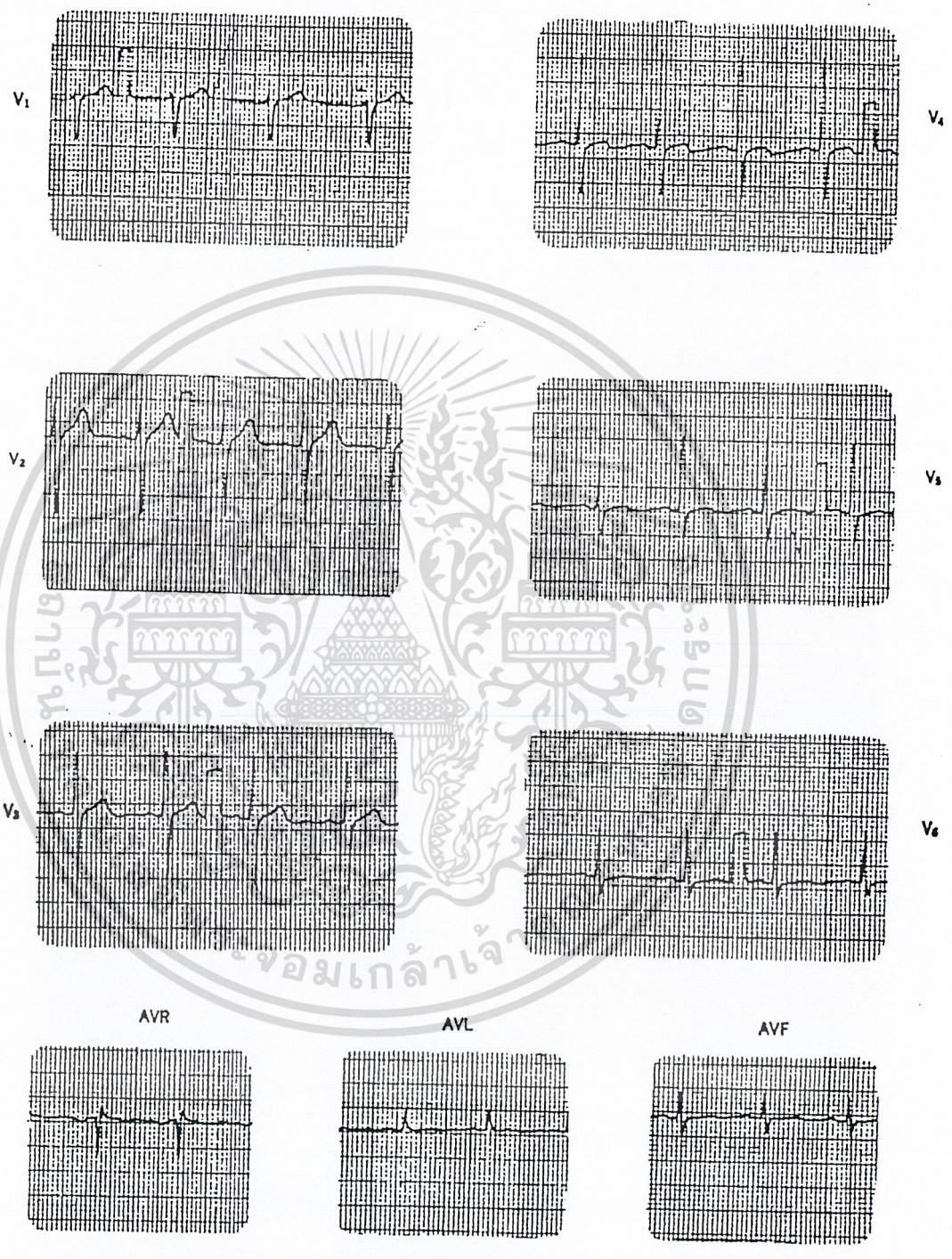
รูปที่ 2.8 แสดงตำแหน่งการวัดคลื่นไฟฟ้าหัวใจทั้ง 12 LEADS มาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 แสดงสัญญาณของ Lead I, Lead II และ Lead III

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 แสดงสัญญาณของ Lead aVR, Lead aVL, Lead aVF และ Unipolar chest leads

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ออปแอร์เรชันแนล แอมพลิฟายเออร์(Operational Amplifier)

ออปแอร์เรชันแนล แอมพลิฟายเออร์(Operational Amplifier) หรือที่เรียกกันสั้นๆ ว่าออปแอมป์(Op-Amp)

สำหรับวงจรการใช้งานของออปแอมป์มีอยู่มากมายหลายแบบ เช่น

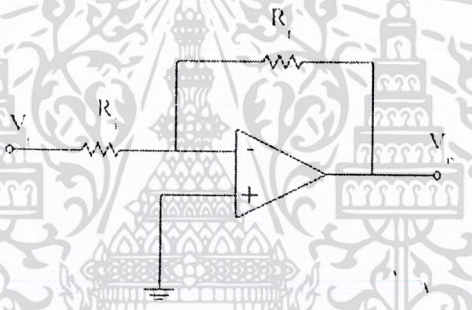
-วงจรขยายแบบกลับ (Inverting Amplifier)

-วงจรขยายความแตกต่าง (Differential Amplifier)

-วงจรกรองความความถี่ (Filter Amplifier)

2.3.1 วงจรขยายแบบกลับ

วงจรขยายแบบกลับ เป็นวงจรที่รับเอาสัญญาณเข้าทางขั้วอินเวอร์ติ้ง(Inverting terminal) และเอาท์พุทที่ได้จะเป็นสัญญาณที่มีเฟสต่างกัน 180 องศากับสัญญาณอินพุทโดยวงจรมีลักษณะเป็นดังรูปที่2.11



รูปที่ 2.11 วงจรขยายแบบกลับ

สัญญาณเอาท์พุทที่ได้จะมีค่ามากน้อย ขึ้นอยู่กับอัตราขยายแรงดัน(Voltage Gain)

$$\text{Voltage Gain } (A_v) = -R_f / R_i$$

เครื่องหมายลบแสดงถึงการกลับเฟสของวงจร และแรงดันเอาท์พุทสมารถคำนวณได้จากสูตร

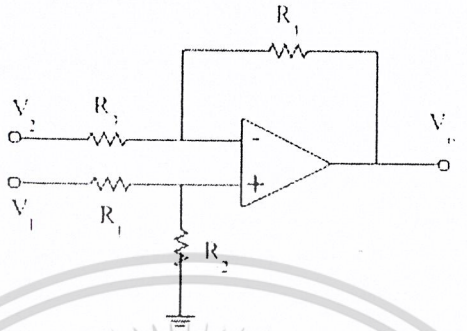
$$V_{out} = A_v V_i$$

$$= (-R_f/R_i)\, V_i$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 วงจรขยายความแตกต่าง(Differential Amplifier)

วงจรขยายความแตกต่าง เป็นวงจรขยายสัญญาณความแตกต่างที่เข้ามาทางขั้วอินเวอร์ติงและขั้วนอน-อินเวอรติง (Noninverting terminal) มีลักษณะพิเศษในการกำจัดสัญญาณรบกวนได้ดี



รูปที่ 2.12 วงจรขยายความแตกต่าง

โดยแรงดันเอาท์พุทหาได้ดังนี้

$$V_o = [(R_3+R_4)/R_3*(R_2)/(R_1+R_2)*V_1]-(R_4/R_3)*V_2$$

เมื่อ $R_1 = R_2 = R_3 = R_4$ อัตราขยายความแตกต่างจะเท่ากับ 1

ส่วนค่าอินพุทอิมพีแดนซ์จะเป็นดังนี้

อินพุทอิมพีแดนซ์ของขั้ว อินเวอร์ติง $= R_3$

อินพุทอิมพีแดนซ์ของขั้ว นอนอินเวอร์ติง $= R_1+R_2$

โดยเอาท์พุทจะเท่ากับ

$$V_o = (V^+-V^-)A$$

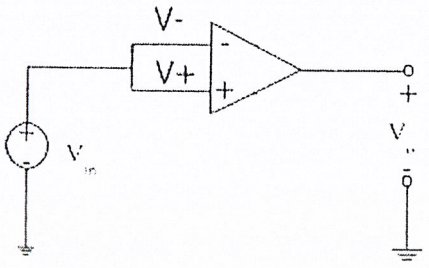
เมื่อ A เป็นค่า gain ของออปแอมป์ V^+เป็นแรงดันที่ขั้วบวก V^- เป็นแรงดันที่ขั้วลบ

ถ้าต่อแบบ Differential op amp ดังรูป 2.13 แรงดันที่เท่ากันป้อนเข้าที่ขั้วทั้งสอง แรงดันเอาท์พุทควรจะเป็นศูนย์ แต่จากสมการข้างบนนี้แสดงว่าของออปแอมป์อุดมคติมีความสมมาตร คือเกนจากขั้วลบไปยังเอาท์พุท, A^- มีค่าเท่ากับกับเกนจากขั้วบวกไปยังเอาท์พุท , A^+ ออปแอมป์ที่ใช้จริงจะไม่สมมาตรกัน นำไปสู่แรงดันเอาท์พุทในแบบคอมมอนโหมดดังนี้

$$V_o = (A^+-A^-)V_{in}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทอม(A^+-A^-) คือ คอมมอนโหมดเกน (Common Mode Gain) ในอุดมคติค่าคอมมอนเกนนี้จะเท่ากับศูนย์

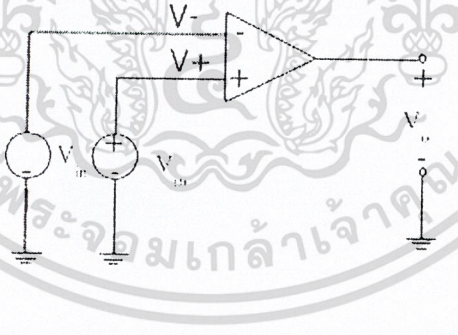


รูปที่ 2.13 การหาค่าของ Common Mode Gain

ต่อไปทำการพิจารณา Differential op-amp ทางอุดมคติดังรูป 2.14 จะเห็นได้ว่าแรงดันเอาท์พุทของออปแอมป์ทางอุดมคติมีค่าเป็น $2AV_{in}$ แต่ในความเป็นจริงแล้ว เอาท์พุทจะเท่ากับ

$$V_o=(A^+ + A^-)*V_{in} = 2[(A^+ + A^-)/2]*V_{in}$$

เทอม $(A^+ + A^-)/2$ จะเรียกว่า Differential Mode Gain



รูปที่ 2.14 การหาค่าของ Differential Mode Gain

ในการวัดการกดของสัญญาณคอมมอนโหมด ที่ไม่ต้องการเทียบกับสัญญาณดิฟเฟอร์เรนเชียลโหมดซึ่งเป็นสัญญาณที่ต้องการ จะกล่าวถึงในรูปCommond Mode Rejection Ratio (CMRR)

$$CMRR = (\text{Differrential Mode Gain})/(\text{Common Mode Gain})$$

$$= [(A^+ + A^-)/2] / [A^+ + A^-]$$

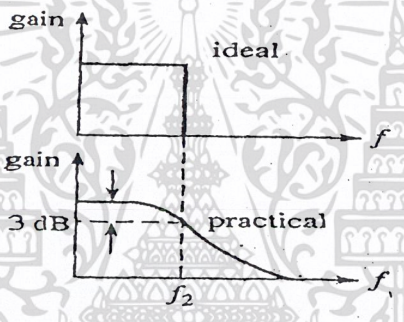
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 วงจรกรองความถี่ (Filter Circuit)

ในการใช้งานอิเล็กโทรนิค บางครั้งเราอาจต้องการให้สัญญาณบางความถี่ผ่านเข้ามาเท่านั้น โดยที่สัญญาณซึ่งที่มีความถี่นอกเหนือจากนี้จะถูกกำจัดออกไปซึ่งก็คือประโยชน์ของวงจรกรองความถี่ สำหรับคลื่น อี.ซี.จี. นั้นจะมีความถี่อยู่ในช่วง 0.1 - 200 เฮิร์ทซ จึงจำเป็นอย่างมากที่จะต้องใช้วงจรกรองความถี่ในการกำจัดสัญญาณที่ไม่ต้องการออกไป เพื่อให้สัญญาณที่ออกมามีความถูกต้องมากที่สุด

2.3.3.1 วงจรกรองความถี่ต่ำ (Low – Pass Filter)

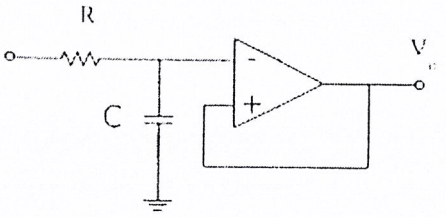
วงจรกรองความถี่ต่ำเป็นวงจรที่กันไม่ให้สัญญาณที่มีความถี่สูงกว่าความถี่คัทออฟ (Cut Off frequency) ผ่านเข้าไปในวงจรเลย แต่ในทางปฏิบัติไม่ได้เป็นในลักษณะนั้นแต่จะค่อยๆถูกลดอัตราขยายลงเรื่อยๆดังรูปที่ 2.15



รูปที่ 2.15 การตอบสนองความถี่ของวงจรกรองความถี่ต่ำ

วงจรกรองความถี่ต่ำนี้จะลดทอนขนาดของสัญญาณที่มีสูงกว่าความถี่คัทออฟซึ่งเรียกช่วงสัญญาณที่มีความถี่ต่ำกว่าความถี่คัทออฟว่า ช่วง Pass Band และเรียกช่วงที่มีความถี่สูงกว่าความถี่คัทออฟว่า ช่วง Stop Band จากรูปที่ 2.16 แสดงตัวอย่างของวงจรกรองความถี่ต่ำแบบพื้นฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 วงจรกรองความถี่ต่ำ

เมื่อมีสัญญาณอินพุทที่มีความถี่ต่ำ ค่ารีแอคแทนซ์ของตัวเก็บประจุ (X_c) จะมีค่าสูงทำให้แรงดันทั้งหมดจากอินพุทตกคร่อมตัวเก็บประจุ เป็นผลทำให้แรงดันเอาท์พุท Vo มีค่าประมาณเท่ากับแรงดันอินพุท Vin ด้วย หากสัญญาณอินพุทที่เข้ามามีความถี่สูง จะทำให้ค่ารีแอคแทนซ์ของตัวเก็บประจุ มีค่าต่ำเป็นผลให้ตัวเก็บประจุเกิดลัดวงจร ดังนั้น Vo จึงมีค่าต่ำด้วย สรุปได้ว่า ช่วงของสัญญาณที่มีค่าความถี่ต่ำจะผ่านไปปรากฏที่เอาท์พุท โดยที่สัญญาณที่มีความถี่สูงจะถูกกั้นเอาไว้ เราสามารถหาความถี่ F_c ได้จากสมการ

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

2.3.3.2 วงจรกรองความถี่ต่ำแบบแอคทีฟฟิลเตอร์ (Active Low – Pass Filter)

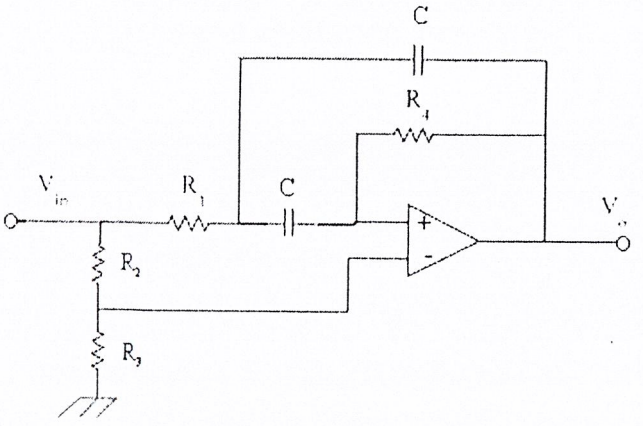
ในที่นี้จะใช้วงจรกรองความถี่แบบ Sallen and Key ดังแสดงในรูปที่ 2.18 โดย ทรานสเฟอร์ฟังก์ชั่นของวงจรแอคทีฟ (สมมติเป็นอุดมคติ) และให้อัตราการขยายของ ออปแอมป์มีค่า

$$T_v = \frac{k/R_1 R_2 C_1 C_2}{S^2 + S\left(\frac{1}{R_1C_1} + \frac{1}{R_2C_2} + \frac{1-k}{R_2C_2}\right) + \frac{1}{R_1R_2C_1C_2}}$$

เป็นอนันต์แล้วจะได้

$$k = 1 + \frac{r_2}{r_1}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 วงจรกรองความถี่ต่ำแบบ Sallen and Key

พิจารณาทรานสเฟอร์ฟังก์ชั่นของวงจรกรองความถี่ต่ำ

$$T_{LP} = \frac{K}{s^2 + \frac{\omega_p}{Q_p} s + \omega_p^2}$$

ซึ่งจะได้ว่า

$$\omega_p = \sqrt{\frac{1}{R_1 R_2 C_1 C_2}}$$

$$Q_p = \frac{\omega_p}{(bw)_p} = \frac{\sqrt{\frac{1}{R_1 R_2 C_1 C_2}}}{\frac{1}{R_1 C_1} + \frac{1}{R_2 C_2} + \frac{1-k}{R_2 C_2}}$$

$$K = \frac{k}{R_1 R_2 C_1 C_2}$$

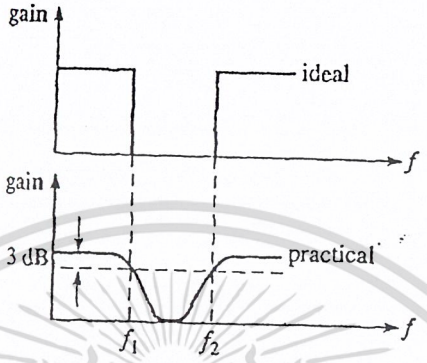
จากสมการข้างบนกำหนดให้ $C_1 = C_2 = 1$ $R_1 = R_2 = R$

ก็จะได้ $R_1 = R_2 = 1/\omega_p$ $k = 3 - 1/Q_p$

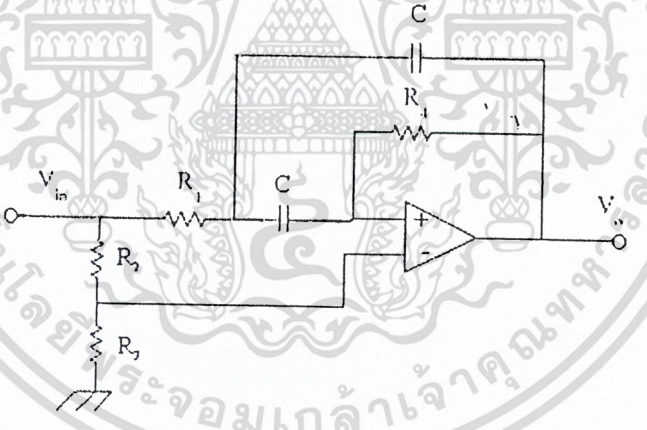
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3.3 วงจรนอท์ชฟิลเตอร์ (Notch Filter)

วงจรนอท์ชฟิลเตอร์นี้จะยอมให้ความถี่ทุกๆค่าผ่านยกเว้นความถี่ช่วงหนึ่งซึ่งได้กำหนดไว้ โดยทั่วไปวงจรนี้มักใช้เพื่อกันสัญญาณรบกวนที่พวกเราทราบค่าความถี่แล้ว ดังรูปที่ 2.18 แสดงการตอบสนองความถี่ของวงจรนอท์ชฟิลเตอร์



รูปที่2.18 การตอบสนองความถี่ของวงจรนอท์ชฟิลเตอร์



รูปที่ 2.19 วงจรนอท์ชฟิลเตอร์พื้นฐาน

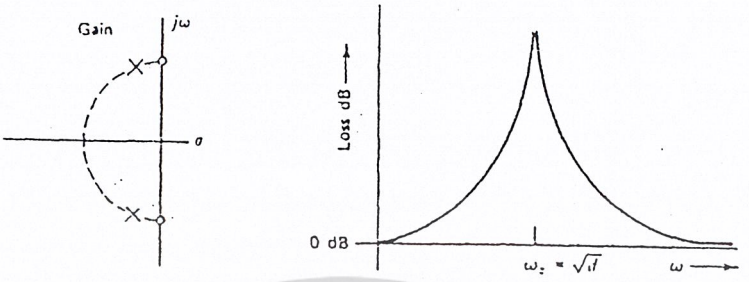
ทรานสเฟอร์ฟังก์ชั่นอันดับ 2 เท่ากับ

$$\frac{V_o}{V_{in}} = \frac{s^2+d}{s^2+as+d} = \frac{s^2+\omega_z}{s^2+\frac{\omega_p}{Q_p}s+\omega_p^2}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

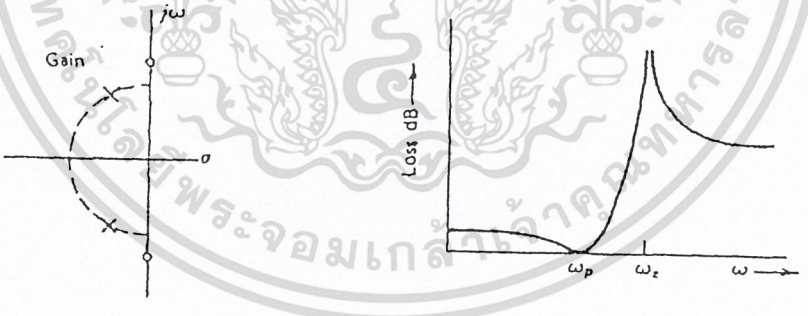
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

พิจารณาตำแหน่งของโพลและซีโร่ของสมการข้างบนเมื่อ $\omega_z = \omega_p$ จะพบว่าสมการมีโพลเชิงซ้อนอยู่ที่เส้นรอบวงและมีซีโร่เชิงซ้อนอยู่บนแกนจินตภาพตามรูปที่ 2.21



รูปที่ 2.20 แสดงตำแหน่งของโพล , ซีโร่ และการสูญเสียในย่านความถี่ของวงจรกรองความถี่แบบนอท์ชฟิวเตอร์

เมื่อ $\omega_z >> \omega_p$ จะมีลักษณะเป็นวงจรกรองความถี่ต่ำแบบนอท์ชฟิวเตอร์ (Low – Pass Notch Filter) ตามรูปที่ 2.21



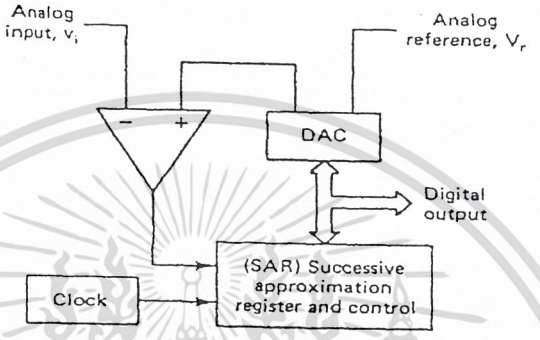
รูปที่ 2.21 แสดงตำแหน่งของโพล,ซีโร่และการสูญเสียในย่านความถี่ต่างๆ ของวงจรกรองความถี่ต่ำแบบนอท์ชฟิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50128

2.4 วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล (Analog to Digital : A/D Converter : ADC)

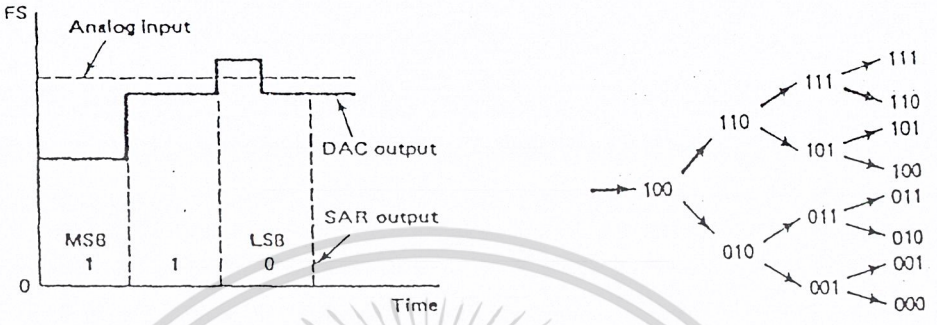
A/D Converter เป็นอุปกรณ์ชนิดหนึ่งซึ่งรับสัญญาณแรงดันอนาลอกเข้ามาเป็นอินพุท แล้วสร้างสัญญาณดิจิตอลที่ใช้แรงดันอนาลอกนั้นขึ้น โดยจะมีอยู่ 2 กระบวนการที่เกี่ยวข้องกับการทำข้อมูลอนาลอกให้เป็นดิจิตอล กระบวนการแรก คือ การแซมปลิ้ง (Sampling) ซึ่งเป็นการหาค่าแรง ดันอนาลอก ณ จุดที่ไม่ต่อเนื่องทางเวลา แรงดันที่ถูกแซมเปิ้ลแล้วต้องนำไปจัดระดับต่อไป ซึ่งเป็นกระบวนการที่สอง การจัดระดับคือการเลือกค่าดิจิตอลที่มีความยาวค่าหนึ่งที่จะใช้แทนแรงดันอนาลอกนั้น



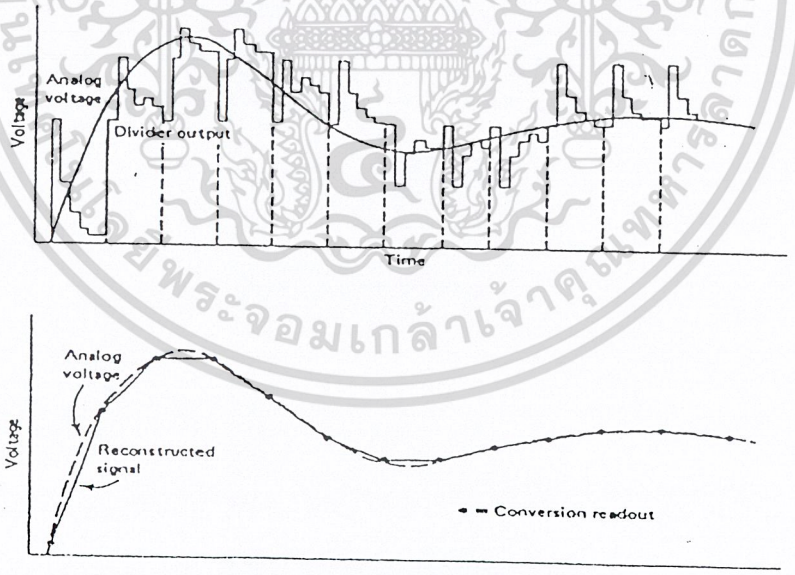
รูปที่ 2.22 แสดงบล็อกไดอะแกรมของ Successive approximation ADC Converter

ในรูปที่ 2.22 เป็นบล็อกไดอะแกรมของ Successive approximation ADC Converter โดยทั่วไปแล้ววิธีการใช้ในการแปลงข้อมูลอนาลอกเป็นดิจิตอลที่ใช้กันส่วนมาก คือวิธีการเปรียบเทียบค่าในรีจีสเตอร์กับแรงดันอินพุท ส่วนวิธีการที่จะปรับค่าสัญญาณดิจิตอลให้ตรงกับสัญญาณอินพุทนั้นมีหลายวิธีด้วยกันและวิธี Successive approximation นี้ก็เป็นเทคนิคที่ใช้บ่อยจากรูปที่ 2.22 จะเห็นว่าวงจรนี้มีวงจรแปลงสัญญาณดิจิตอลเป็นอนาลอก (Digital to Analog Converter : DAC) อยู่ภายในทำหน้าที่สร้างสัญญาณอนาลอกให้สอดคล้องกับสัญญาณที่ออกมาจาก Successive approximation Register (SAR) ซึ่งรีจีสเตอร์นี้ก็คือชุดของฟลิปฟลอปนั่นเอง โดยแต่ละบิทจะถูกตรวจสอบว่า การเพิ่มค่าในรีจีสเตอร์นั้นจะทำให้ค่าในรีจีสเตอร์มากกว่าค่าของสัญญาณอินพุทหรือไม่ ถ้ายังไม่เกิน บิทที่ถูกตรวจสอบนั้นก็จะมีค่าเป็น 1 แต่ถ้าเกินค่าสัญญาณอินพุท บิทนั้นก็จะมีค่าเป็น 0 โดยกระบวนการดังนี้ เริ่มจากบิทที่มีความสำคัญมากที่สุด(Most Significant Bit) และทำต่อไปเรื่อยๆจากบิทซ้ายไปขวา ข้อดีของกระบวนการนี้คือ ช่วงเวลาการแปลงสัญญาณจะคงที่ ไม่ขึ้นกับสัญญาณอินพุทอีกทั้งยังให้ผลตอบสนองที่ดีต่อการเปลี่ยนอย่างรวดเร็วของสัญญาณอินพุท โดยจะใช้ตัววงจรแซมปลิ้งแอนด์โฮลเป็นตัวอ่านค่าแรงดันที่เวลาเริ่มต้นของการเปลี่ยนแปลงและคงค่าไว้ระหว่างทำการเปลี่ยนแปลงซึ่งจะทำให้ผลลัพธ์ที่ได้มีค่าใกล้เคียงกับสัญญาณอินพุทมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.23 วิธีการ Successive approximation



รูปที่2.24 แสดงการแปลงแบบ Successive approximation ที่ใช้ในวงจร sample and hold

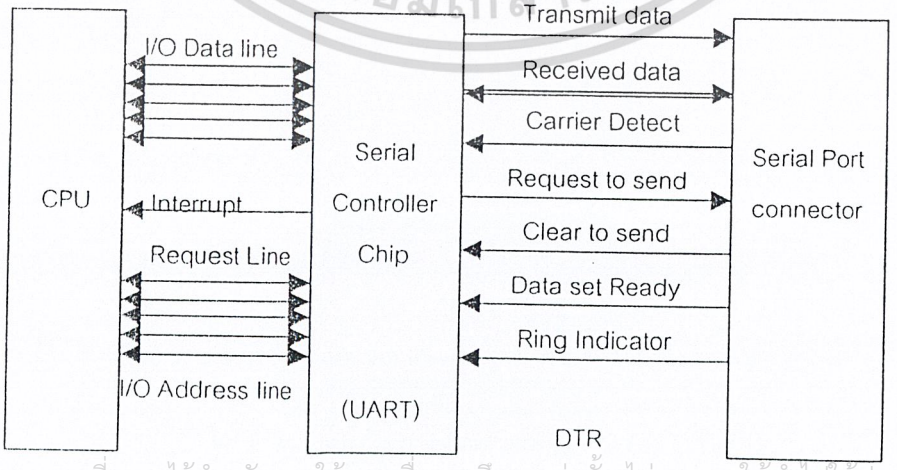
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 การรับส่งข้อมูลจากไมโครคอมพิวเตอร์

ข้อมูลจากไมโครคอมพิวเตอร์จะทำการส่งเป็นแบบอนุกรมผ่านทางพอร์ตอนุกรม ซึ่งใช้มาตรฐานแบบ RS-232 เป็นมาตรฐานที่ถูกกำหนด โดย EIA ซึ่งเป็นองค์กรอุตสาหกรรมอิเล็กทรอนิกส์ของอเมริกา โดยการเชื่อมต่อแบ่งออกเป็น 2 ส่วนคือ DTE (Data Terminal Equipment) และ DCE (Data Communication Equipment) ซึ่งโดยปกติ DTE จะต้องต่อเข้ากับ DCE เช่น การต่อเครื่องคอมพิวเตอร์ (อุปกรณ์ DTE) เข้ากับโมเด็ม (อุปกรณ์ DEC) เป็นต้น โดย DTE มักจะแทนแหล่งกำเนิดข้อมูลแหล่งแรก และ/หรืออุปกรณ์ที่เป็นแหล่งรับข้อมูลแหล่งสุดท้าย ส่วน DCE เป็นอุปกรณ์ที่ทำให้การสื่อสารข้อมูลระหว่างแหล่งกำเนิดกับตัวรับข้อมูลที่ปลายทางทำได้สะดวกขึ้น ส่วนการเชื่อมโยงระหว่าง DTE และ DCE นั้นใช้มาตรฐาน RS-232 ตัวอย่างเช่นการเชื่อมติดต่อระหว่างคอมพิวเตอร์ด้วยกันผ่านทางสายโทรศัพท์

รูปที่ 2.25 การเชื่อมติดต่อระหว่างคอมพิวเตอร์ด้วยกันผ่านทางสายโทรศัพท์

พอร์ตอนุกรม RS-232 จะเป็นพอร์ตของเครื่องคอมพิวเตอร์ที่มีขาต่อ (Connector) ทั้งประเภท 9 และ 25 ขา นั้นคือ Com1 และ Com2 นั้นเอง ในความเป็นจริงพอร์ตอนุกรมไม่ได้ถูกควบคุมโดยตรงจากหน่วยประมวลผลกลาง (CPU) แต่จะถูกจัดการโดย ชิป UART (Universal Asynchronous Receiver/Transmitter) สำหรับการเชื่อมต่อสายสัญญาณต่างๆ ระหว่างหน่วยประมวลผลกลาง (CPU) ของเมนบอร์ดของเครื่องคอมพิวเตอร์กับพอร์ตอนุกรมนั้น จะต้องกระทำผ่านทางชิป UART ซึ่งจะมีวิธีการเชื่อมต่อดังในรูปที่ 2.26



รูปที่ 2.26 แสดงการเชื่อมต่อสายสัญญาณระหว่างCPUของเครื่องคอมพิวเตอร์กับพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยสายสัญญาณแต่ละเส้นมีความหมายดังต่อไปนี้

1. Transmit Data (TD)

เป็นวงจรที่สร้างสัญญาณ Transmitted Data ซึ่งถูกส่งจาก DTE ไปยัง DCE ซึ่งสัญญาณที่ส่งออกมาจะเป็นข้อมูลที่ต้องการส่ง

2. Received Data (RD)

เป็นวงจรที่สร้างสัญญาณ Received Data ซึ่งถูกส่งจาก DEC ไปยัง DTE ซึ่งเป็นสัญญาณที่ส่งออกมาจะเป็นข้อมูลที่ CPU จะรับเข้าไป

3. Data Terminal Ready (DTR)

สัญญาณ DTR จะถูกส่งจาก DTE ไปยัง DCE เพื่อเป็นการแจ้งความพร้องให้อุปกรณ์ปลายทางทราบว่าจะมีการส่งข้อมูลไปยังอุปกรณ์ปลายทาง

4. Carrier Detect (CD)

สัญญาณ CD จะถูกส่งจาก DCE ไปยัง DTE เพื่อเป็นการแจ้งว่า อุปกรณ์ปลายทางกำลังได้รับสัญญาณที่พร้อมสำหรับการติดต่อสื่อสาร หรือกำลังติดต่อกับอุปกรณ์ตัวอื่น

5. Request To Send (RTS)

สัญญาณ RTS จะถูกส่งจาก DTE ไปยัง DCE โดยเมื่อสัญญาณ RTS อยู่ในสถานะ on หมายถึง เครื่องคอมพิวเตอร์พร้อมที่จะรับข้อมูล ถ้าหาก RTS อยู่ใน สถานะ off หมายถึง เครื่องคอมพิวเตอร์ไม่พร้อมที่จะรับข้อมูล

6. Clear To Send (CTS)

สัญญาณ CTS จะถูกส่งจาก DCE ไปยัง DTE โดยเมื่อสัญญาณ CTS อยู่ในสถานะ on ก็ต่อเมื่ออุปกรณ์ปลายทางพร้อมที่รับข้อมูล ถ้าหากสัญญาณ CTS อยู่ในสถานะ off หมายถึง อุปกรณ์ปลายทางไม่พร้อมที่จะรับข้อมูล

7. Data Set Ready (DSR)

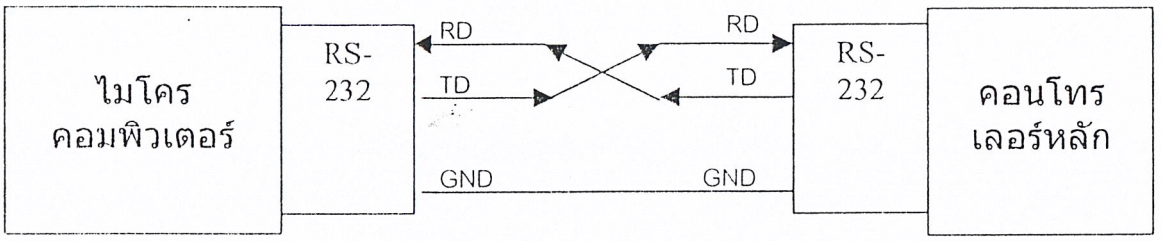
สัญญาณ DSR จะถูกส่งจากDCE ไปยัง DTE เพื่อแจ้งความพร้อมในการสื่อสารจากอุปกรณ์ปลายทางให้กับเครื่องคอมพิวเตอร์ทราบ

8. Ring Indicator (RI)

สัญญาณ RI จะถูกส่งจาก DCE ไปยัง DTE เพื่อเป็นการแจ้งให้เครื่องคอมพิวเตอร์ทราบว่าอุปกรณ์ปลายทางได้รับสัญญาณกริ่ง (ring signal) จากอุปกรณ์ปลายทางตัวอื่น

สำหรับโครงงานนี้ไมโครคอมพิวเตอร์เป็น DTE และ คอนโทรลเลอร์หลัก เป็นอุปกรณ์ DTE เช่นเดียวกัน ในกรณีที่เป็นอุปกรณ์ DTE ทั้ง 2 ตัว จะต้องทำการสลับเส้นสัญญาณระหว่าง RD และ TD เพื่อให้สามารถเชื่อมต่อกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.27 การเชื่อมต่อระหว่างอุปกรณ์ DTE กับไมโครคอนโทรลเลอร์

จะเห็นได้ว่าการสื่อสารระหว่างไมโครคอมพิวเตอรกับคอนโทรลเลอร์หลักจะเป็นการสื่อสารแบบ Full duplex หมายความจะสามารถทำการส่งและรับข้อมูลได้ในเวลาเดียวกันนั่นเองในการรับและส่งข้อมูลนั้นจะใช้มาตราฐาน RS-232 (Recommended Standard Number 232 , Revision C.) ซึ่งสัญญาณที่กล่าวมาข้างต้นทั้งหมดนั้นจะเชื่อมต่อกันไปที่ขาต่อแบบ 9 ขา ซึ่งนั้นก็คือ พอร์ต Com1 และ Com2 นั้นเอง

2.5.1 มาตรฐาน RS-232

โดยทั่ว ๆ ไป ข้อมูลจะถูกแทนด้วยเลขฐานสอง ระดับแรงดัน 5 โวลท์จะแทนลอจิก 1 และระดับแรงดัน 0 โวลท์ จะแทนลอจิก 0 การแทนลอจิกด้วยระดับแรงดันเหล่านี้ เรียนว่าการแทนระดับสัญญาณของอุปกรณ์ TTL (Transistor-Transistor Logic) ซึ่งการแทนระดับสัญญาณแบบนี้ถูกใช้กันทั่ว ๆ ไป ระดับแรงดันที่ใช้แทนลอจิก 0 และ 1 นั้นถูกกำหนดเป็นพิสัย (range) แรงดันที่ส่งจากอุปกรณ์ที่เป็นตัวส่งจะต้องจ่ายแรงดันที่มีระดับสัญญาณต่ำสุดเท่ากัน 2.4 โวลท์ ในการส่ง ลอจิก 1 แต่ในภาครับจะถือระดับสัญญาณต่ำสุดที่แทนลอจิก 1 คือ 2.0 โวลท์สาเหตุเนื่องมาจากการคำนึกถึงการสูญเสียระหว่างตัวส่งและตัวรับนั่นเอง เช่นเดียวกันในการแทนลอจิก 0 ตัวส่งจะส่งระดับสัญญาณสูงสุดไม่เกิน 0.4 โวลท์ ในการแทนลอจิก 0 และตัวรับจะถือระดับสัญญาณสูงสุดที่เป็นลอจิก 0 ไม่เกิน 0.8 โวลท์ ในการส่งข้อมูลระหว่างอุปกรณ์ชิ้นหนึ่งไปยังอีกชิ้นหนึ่งในเครื่องคอมพิวเตอร์ แต่ในการส่งและรับข้อมูลระหว่างอุปกรณ์ภายนอกเครื่องคอมพิวเตอร์ เทคนิคที่ใช้ในการสื่อสารข้อมูลภายในเครื่องคอมพิวเตอร์นั้นอาจนำมาใช้ได้ซึ่งถ้าเราทำการสื่อสารข้อมูลด้วยระดับสัญญาณในระยะมากกว่า 2-3 ฟุตอาจเกิดปัญหาบางประการขึ้นได้เนื่องจาก

1. ระดับสัญญาณ TTL มักถูกเหนี่ยวนำจากสัญญาณรบกวนจากภายนอกได้ง่าย
2. การสูญเสียในสายส่งทำให้ระดับแรงดันของสัญญาณที่ส่งออกไปลดลงซึ่งมีผลกระทบที่สำคัญต่อระดับแรงดัน 0-5 โวลท์ของ TTL เพราะว่าการสูญเสียระดับเพียง 2-3 โวลท์ทำให้ลอจิกต่าง ๆ ที่ได้รับผิดพลาดไป

สาเหตุดังกล่าวทำให้เกิดมาตรฐาน RS –232 ขึ้นมา โดยใช้ลอจิกลบ (negative logic) แทนระดับแรงดันต่าง ๆ คือวิธีการเปรียบเทียบระดับแรงดันแบบหนึ่ง ถ้าระดับ แรงดันหนึ่งมีค่าเป็นลบมากกว่าจะเป็นลอจิก "สูง" ดังนี้ 1 = -V ส่วน 0 = +V นอกจากนี้ช่วงของระดับแรงดันระหว่าง –3 ถึง +3 โวลท์ จะเป็นช่วงการเปลี่ยนแปลงลอจิก จึงไม่มีการระบุสถานะของสัญญาณในช่วงนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 รูปแบบของข้อมูล

ในการสื่อสารระหว่างอุปกรณ์ 2 ตัวนั้น จะต้องมีความเข้าในถึงรูปแบบข้อมูลที่จะติดต่อระหว่างกันให้ตรงกัน โดยข้อมูลในการสื่อสารจะประกอบด้วยบิทข้อมูลที่มีความหมายดังต่อไปนี้

1. บิตข้อมูล (Data bit) เป็นบิตที่เป็นข้อมูลที่สื่อสารกันและถูกส่งมาในลักษณะที่เป็นสายข้อมูลโดยเริ่มต้นส่งจากบิตต่ำสุดไปยังบิตสูงสุด สายข้อมูลชนิดบิตนี้จะมีความยาวตั้งแต่ 5 ถึง 8 บิต

2. บิตเริ่มต้นข้อมูล (Start bit) ในการส่งข้อมูล เราจะต้องมีวิธีการบอกอุปกรณ์ปลายทางให้ทราบจุดเริ่มต้นของข้อมูล ดังนั้นก่อนหน้าข้อมูลทุกๆข้อมูลจะต้องถูกนำหน้าด้วยบิตเริ่มต้นข้อมูล (Start bit) เสมอ

3. บิตสิ้นสุดข้อมูล (Stop bit) ในการส่งข้อมูล ในกรณีที่บิตเริ่มต้อนข้อมูลเกิดการสูญหายในระหว่างการส่งอุปกรณ์ปลายทางก็จะไม่สามารถทราบถึงจุดสิ้นสุดของสายข้อมูลได้เลย นอกเสียจากจะตรวจพบบิตเริ่มต้นข้อมูลใหม่อีกครั้ง ดังนั้นจึงมีการเพิ่มบิตสิ้นสุดข้อมูล ต่อท้ายทุกๆ ข้อมูลของแต่ละตัวอักษรเพื่อแจ้งการสิ้นสุดของสายข้อมูล

4. บิตพาริตี้ (Parity bit) เนื่องจากในการสื่อสารนั้นสามารถเกิดสัญญาณรบกวนได้ง่ายเป็นไปได้มากที่สถานะของแต่ละบิตของข้อมูลที่ถูกส่งจะมีการเปลี่ยนแปลง เช่น จากบิต 0 เป็น บิต 1 ด้วยเหตุนี้จึงมีการคิดค้น

การตรวจสอบความถูกต้องของสถานะของบิตข้อมูล ซึ่งเรียกกันว่า การตรวจสอบค่าพาริตี้ (parity check) ดังนั้นจึงมีการเพิ่มบิตพาริตี้ขึ้นมา เพื่อตรวจสอบสถานะของผลบวกของบิตที่เป็น 1 ของสายบิตข้อมูล ซึ่งมีหลักการคำนวณค่าของบิตพาริตี้มีดังนี้

1. ถ้าหากจำนานบิตที่มีค่าเท่ากับ 1 ของสายบิตข้อมูลมีค่าเป็นเลขคู่ (Even number) บิตพาริตี้จะมีค่าเท่ากับ 0

2. ถ้าหากจำนานบิตที่มีค่าเท่ากับ 1 ของสายบิตข้อมูลมีค่าเป็นเลขคี่ (Odd number) บิตพาริตี้จะมีค่าเท่ากับ 1

ในโครงงานนี้จะทำการแบบทั่ว ๆ ไป คือ ไม่มีพาริตี้บิต เพราะเป็นการง่ายที่สุดในการสื่อสารอีกทั้งคอนโทรลเลอร์หลักก็ไม่ได้อยู่ไกลจากไมโครคอมพิวเตอร์มากนัก

2.6 การเขียนโปรแกรมให้ไมโครคอมพิวเตอร์ติดต่อกับพอร์ตสื่อสาร

ในการเขียนโปรแกรม Visual Basic เพื่อสื่อสารข้อมูลนั้นทำได้โดยใช้คอนโทรล MSComm (Communications) ซึ่งเป็นคอนโทรลตัวหนึ่งที่ช่วยในการติดต่อกับพอร์ตอนุกรม (Serial port) ซึ่งทำให้สามารถรับ - ส่งข้อมูลผ่านทางพอร์ตอนุกรมได้ด้วยคอนโทรลตัวนี้ เช่นการติดต่อผ่าน โมเด็ม (Modem) หรือติดต่อโดยตรงกับบอร์ดอิเล็กทรอนิกส์ เป็นต้น ซึ่งคอนโทรล MSComm ที่มากับ Visual Basic จะเป็นคอนโทรลที่ทำงานโดยมีการตอนสนองต่อเหตุการณ์แบบ event - driven นั่นคือ คอนโทรลจะทำหน้าที่ตรวจสอบการเกิดขึ้นหรือจากการร้องขอให้เกิดเหตุการณ์ต่างๆ กับพอร์ตอนุกรมโดยอัตโนมัติ ในความเป็นจริงแล้วคอนโทรล MSComm ไม่ได้ทำหน้าที่ติดต่อกับพอร์ตอนุกรมโดยตรง แต่จะทำหน้าที่เรียกใช้ฟังก์ชั่นวินโดวส์ API ซึ่งวินโดวส์จะทำการส่งหรือรับข้อมูลผ่านทางพอร์ตอนุกรมโดยอาศัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไดรเวอร์ Comm.drv อีกทอดหนึ่ง ทุกครั้งที่มีการเรียกใช้คอนโทรล MSComm ก็หมายถึงเรียกใช้ฟังก์ชันวินโดวส์ API ซึ่งจะถูกตีความอีกทอดหนึ่งโดยไดรเวอร์ Comm.drv จากนั้นก็จะส่งผ่านข้อมูลที่ถูกจัดรูปแบบตามมาตราฐานการสื่อสารให้กับดีไวซ์ไดรเวอร์อีกทอดหนึ่ง ในการตั้งค่าความเร็วและรูปแบบของข้อมูลสามารถสั่งงานได้โดยใช้ทำสั่ง MSComm ซึ่งสามารถควบคุมการปิดเปิดพอร์ต และสามารถเลือกได้ว่าต้องการให้สื่อสารผ่านทางพอร์ตไหน เช่น Com1 , Com2 , Com3 หรือ Com4 เป็นต้น แต่สำหรับโครงงานนี้จะขอเลือกใช้พอร์ต Com1 หรือ Com2 เท่านั้นเนื่องจากเป็น พอร์ตที่มีการต่อออกมาภายนอกผ่านทาง RS-232 แล้วซึ่งมีลักษณะเป็นแบบพอร์ตขาต่อ 9 ขา สามารถนำไปใช้งานได้เลย ในการเขียนโปรแกรมเพื่อทำการกำหนดพอร์ต และ ข้อมูลได้ดังนี้

MSComm1.CommPort = 1	หมายถึง ใช้การสื่อสารที่พอร์ต Com1
MSComm1.Settings = "2400,N,8,1"	หมายถึง รูปแบบและความเร็วของข้อมูล
MSComm1.PortOpen = True	หมายถึง ให้ทำการเปิดพอร์ต

ในการสื่อสารระหว่างคอนโทรลเลอร์หลักกับไมโครคอมพิวเตอร์ จะทำการสื่อสารผ่านทาง RS-232 ซึ่งในการสื่อสารนี้คอนโทรลเลอร์หลักจะต้องทำการรับรู้รูปแบบบิตของข้อมูล และอัตราความเร็วในการรับ-และส่งข้อมูลกับไมโครคอมพิวเตอร์ ซึ่งในไมโครคอมพิวเตอร์สามารถตั้งค่าความเร็วและรูปแบบของข้อมูลได้ด้วยการเขียนโปรแกรม Visual Basic โดยใช้คำสั่ง MSComm เช่นเดียวกัน ใน MSC-51 ก็จะเป็นที่จะต้องทำการตั้งคุณลักษณะในการส่งให้ตรงกัน

การสื่อสารข้อมูลแบบอนุกรมใน MCS-51

พอร์ตสื่อสารข้อมูลแบบอนุกรมที่มีใน MCS-51 สามารถทำงานได้ในแบบ Full duplex สามารถรับและส่งข้อมูลได้พร้อมๆกัน โดยในการรับข้อมูลจะมีการบัฟเฟอร์ข้อมูลให้ด้วย จึงทำให้ MCS-51 สามารถกำหนดการรับข้อมูลไบต์ที่สองซึ่งถูกส่งตามเข้ามาก่อนที่ไบต์แรกที่ได้รับเข้ามาจะถูกอ่านจากรีจิสเตอร์ใช้งานเฉพาะที่ใช้สำหรับรับข้อมูล (Receive Register) เพื่อนำไปเก็บไว้ในหน่วยความจำต่อไป (หากไบต์แรกยังไม่ถูกอ่านเมื่อได้รับไบต์ที่สองเรียบร้อยแล้วข้อมูลจะหายไปหนึ่งไบต์)

พอร์ตสื่อสารข้อมูลแบบอนุกรมใน MCS-51 ประกอบด้วยรีจิสเตอร์ขนาด 8 บิตจำนวนสองตัว แต่ละตัวมีชื่อเรียกตามหน้าที่ดังนี้คือ

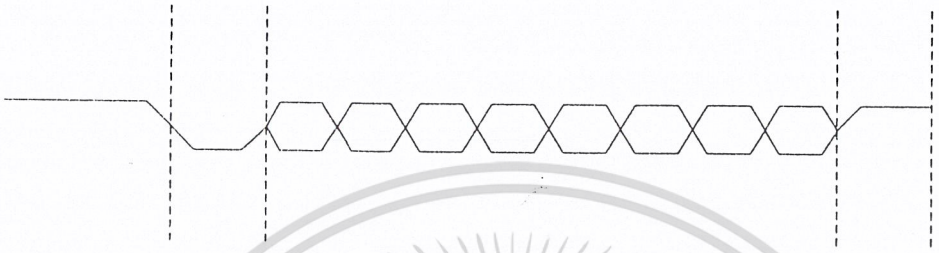
- รีจิสเตอร์สำหรับใช้รับข้อมูลที่ส่งเข้ามาจากภายนอก
- รีจิสเตอร์สำหรับส่งข้อมูล (Transmit register) ใช้ส่งข้อมูลจาก MCS-51 ออกไปภายนอก

รีจิสเตอร์ทั้งสองมีตำแหน่งเดียวกันในรีจิสเตอร์ใช้งานเฉพาะ คือ ตรงกับตำแหน่งของรีจิสเตอร์ใช้เฉพาะ SBUF (ตำแหน่ง 99H) การเข้าถึงข้อมูลในรีจิสเตอร์แต่ละตัว MCS-51 จะทราบเองว่าผู้ใช้ต้องการติดต่อกับรีจิสเตอร์ตัวใดโดยตรวจสอยจากรหัสคำสั่ง ทั้งนี้เพราะในการเขียนโปรแกรมเพื่อเขียนข้อมูลไปไว้ในรีจิสเตอร์ใช้งานเฉพาะ (SBUF) หมายถึงการโหลดข้อมูลไปที่ริจิสเตอร์สำหรับส่งข้อมูลเพื่อทำการส่งข้อมูลออกไปภายนอก ส่วนการอ่านข้อมูลจากรีจิสเตอร์ใช้งานเฉพาะ(SBUF) จะหมายถึงนำค่าที่รับเข้ามาได้จากภายนอกที่เก็บไว้ในรีจิสเตอร์สำหรับรับข้อมูลมาใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การsetserialportให้กับMcontroler

ในการทำหนดรูปแบบของข้อมูลเพื่อใช้ในการติดต่อกับ Computer นั้น โดยส่วนใหญ่แล้วจะใช้รูปแบบคือ 8 บิทข้อมูล 1 บิตหยุด(stop bit) ไม่มีparity check ในโครงงานนี้จะรับส่งข้อมูลด้วยความเร็ว 28800 บิต/วินาที เพราะฉะนั้น จะรับส่งด้วยความเร็ว 2880 byte/s เพราะ หนึ่งbyte จะประกอบไปด้วย 10 บิตข้อมูลที่ส่งให้



วิธีการทำให้MCS-51 สามารถรับส่งข้อมูลที่ความเร็ว 28800bps นั้นจะต้องใช้counter ที่มีอยู่ในตัว 8051 เป็นตัวกำหนดจังหวะการรับส่งข้อมูล ซึ่งสามารถทำมีกำหนดให้register ของ 8051 มีค่าดังนี้

1 register SCON

มีขนาด 8 บิต

- SM0 SCON.7
- SM1 SCON.6
- SM2 SCOM.5 เป็นบิตที่ให้ควบคุมการทำงานของฮาร์ดแวร์ที่จะทำให้บิต RI เป็น 1 หรือไม่ในกรณีที่ SM2 เป็น 0 ค่าในบิต RI ก็จะเป็นไปตามกลไกของมันเอง(จะอธิบายภายหลัง) ในโหมด 2 และ 3 ซึ่งปกติแล้ว RI จะเป็น 1 เมื่อมีบิตที่ 9 เข้ามา แต่ถ้า SM2 เป็น 1 แล้ว RI จะเป็น 1 ก็ต่อเมื่อข้อมูลบิทที่ 9 มีค่า เป็น 1 และจะเป็น 0 เมื่อข้อมูลบิตที่ 9 เป็น 0
- REN SCON.4 เป็นบิตที่จะใช้กำหนดให้ทำการรับข้อมูลเข้ามาจากทางพอร์ทอนุกรม หรือไม่ ถ้าบิตนี้เป็น 1 ก็จะรับข้อมูลเข้ามา แต่ถ้าเป็น 0 ก็จะไม่รับข้อมูลที่ขา RXD เข้ามา การให้บิตนี้เป็น 1 หรือ 0 ทำได้โดยใช้คำสั่ง setb หรือ clr bit ที่ตำแหน่งของบิท 9cH
- TB8 SCON.3 ในการส่งข้อมูลในโหมด 2,3 จะใช้บิตนี้สำหรับเก็บบิตที่ 9 ส่วนในโหมดอื่น บิตนี้ไม่ใช้
- RB8 SCON.2 เมื่อมีการรับข้อมูลในโหมด2,3 จะใช้บิตนี้เป็นที่เก็บบิตที่ 9 ส่วนในโหมดอื่นบิตนี้ไม่ใช้
- TI SCON.1 ค่าในบิต TI จะถูกกำหนดให้เป็น 1 ด้วยฮาร์ดแวร์ โดยในการส่งข้อมูลแบบอนุกรมโหมด 0 บิตนี้จะเป็น 1 เพื่อจะบอกว่าการส่งข้อมูลในรีจิสเตอร์ SBUF ออกไปทางพอร์ทอนุกรมครบทั้ง 8 บิต แต่ถ้าเป็นการส่งในโหมดอื่นจะทำให้ข้อมูลในบิตนี้เป็น 1 เมื่อเริ่มมีการส่ง STOP bit ข้อมูลบิตนี้สามารถ Clear เป็น 0 ได้ด้วยคำสั่ง Clr Bit โดยที่ค่าตำแหน่งของบิตนี้ เท่ากับ 99h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิทนี้ยังมีประโยชน์เพื่อบอกว่าการส่งข้อมูลจาก SBUF ออกไปพอร์ทอนุกรมนั้นสิ้นสุดแล้วพร้อมที่จะให้โปรแกรมเขียนข้อมูลลงไปยัง SBUF สำหรับส่งออกต่อไปได้

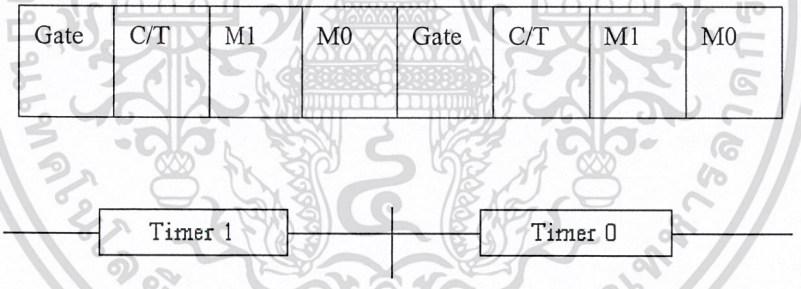
- RI SCON.0 บิตนี้จะถูกกำหนดโดยฮาร์ดแวร์ ให้มีค่าเป็น 0 หรือ 1 โดยที่ในการรับข้อมูลโหมด 0 นั้น บิต RI จะมีค่าเป็น 1 เมื่อมีข้อมูลเข้ามาครบทั้ง 8 บิต ส่วนในโหมดอื่น บิต RI จะเป็น 1 ก็ต่อเมื่อข้อมูลเข้ามาถึงเวลาครึ่งหนึ่งของ Stop Bit (ยกเว้นบางกรณีดังที่ได้กล่าวมาแล้วในเรื่อง SM2) บิตนี้สามารถ clear ให้มีค่าเป็น 0 ได้ด้วยคำสั่ง Clr Bit โดยค่าตำแหน่งของบิตมีค่าเท่ากับ 98h บิตนี้มีประโยชน์ให้รู้ว่าข้อมูลได้เข้ามาอยู่ใน SBUF ครบแล้ว

Register TMOD

TMOD Timer / Counter mode register

ตำแหน่งหน่วยความจำภายในเท่ากับ 89H

TMOD เป็นรีจิสเตอร์ขนาด 8 บิต ที่มีหน้าที่ควบคุมการทำงานของ Timer 0 และ Timer 1 แต่ละบิต ในรีจิสตอร์นี้มีความหมายเฉพาะดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูป MO เป็นชื่อของบิต 0 และ GATE ทางซ้ายสุดเป็นชื่อของบิท 7 รีจิสเตอร์นี้ แบ่งข้อมูลออกเป็น 2 ชุด ชุดละ 4 บิต คือ บิต 0-3 ใช้สำหรับควบคุมการทำงานของ Timer 0 และบิต 4-7 ใช้ควบคุมการทำงานของ Timer 1 หน้าที่ในการควบคุม Timer ของแต่ละบิตที่มีชื่อเดียวกันจะเหมือนกัน

GATE เป็นบิตที่ใช้ควบคุมให้ Timer ทำงานหรือไม่ ถ้าบิตนี้ของ Timer ถูกตั้งเป็น 1 จะทำให้ Timer ทำงานก็ต่อเมื่อที่ขา INTx ที่มีสภาวะลอจิกเป็น 1 และบิต TRx ในรีจิสเตอร์ TCON เป็น 1 ด้วย

C/T บิตนี้ในใช้สำหรับเลือกการทำงานของ Timer ว่าจะใช้เป็น Timer หรือ Counter ถ้าบิตนี้เป็น 1 ก็หมายความว่าเลือกการทำงานเป็น Counter ซึ่งจะนับจำนวนไซเคิลของสัญญาณที่เข้ามาทางขา Tx

M1,M0 เป็น 2 บิตที่ใช้ร่วมกันเพื่อเลือกโหมดการทำงานของ Timer การทำงานโหมด 0,1 และ 2 ของ Timer 0 จะเหมือนกับ Timer 1 แต่ในโหมด 3 การทำงานของทั้งสองจะต่างกัน ค่าใน M1 และ M0 จะเลือกโหมดการทำงานดังนี้

M1	M0	การทำงาน
0	0	โหมด 0 รีจิสเตอร์ THx และ TLx ทำตัวเป็นตัวนับ 13 บิต ค่าจากการนับ 8 บิทบนมาจาก 8 บิตของ THx และอีก 5 บิตล่างมาจากค่า 5 บิตล่างของรีจิสเตอร์ TLx โดยที่ 3 บิตบนของ TLx จะไม่ต้องสนใจเลย
0	1	โหมด 1 รีจิสเตอร์ THx และ TLx ทำตัวเป็นตัวนับ 16 บิตค่าจากการนับ 8 บิทบนอยู่ในรีจิสเตอร์ THx และค่าจากการนับ 8 บิตล่างอยู่ในรีจิสเตอร์ TLx
1	0	โหมด 2 ในการนับของรีจิสเตอร์ TLx ขนาด 8 บิตเมื่อนับถึงค่าสูงสุดคือ FFH เมื่อทำการนับต่อไปจะเกิดการ Overflow แล้วก็จะ "Reload" เอาข้อมูลจาก THx เข้าไปยัง TLx เพื่อเป็นค่าเริ่มต้นในการนับครั้งต่อไป
1	1	โหมด 3 การทำงานของ Timer 0 และ Timer 1 จะต่างกันดังที่จะกล่าวต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TCON Timer Control Register ตำแหน่งหน่วยความจำภายในเท่ากับ 088H

รีจิสเตอร์ขนาด 8 บิตนี้ใช้ควบคุมการทำงานและบอกสภาวะของ Timer 0 และ Timer 1 แต่บิตของรีจิสเตอร์นี้จะทำงานต่างกันดังรูป

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1	TCON.7	Timer 1 overflowflag.Set by hardware when the Timer/Counter 1 overflows.Cleared by hardware as processor vectors to the interrupt service routine.
TR1	TOCN.6	Timer 1 run control bit. Set/cleared by software to turn Timer/Counter 1 ONOFF.
TF0	TOCN.5	Timer 0 overflowflag.Set by hardware when the Timer/Counter 0 overflows.Cleared by hardware as processor vectors to the service routine.
TR0	TOCN.4	Timer 0 run control bit.Set/cleared by software to turn Timer/Count 0 ON/OFF.
IE1	TOCN.3	External Interrupt 1 edge flag. Set by hardware when External interrupt edge is detected. Cleared by hardware when interrupt is processed.
IT1	TCON.2	Interrupt 1 type control bit. Set/cleared by software to specify failing edge/flow level triggered External Interrupt.
IE0	TCON.1	External Interrupt 0 edge flag. Set by hardware when External Interrupt edge detected. Cleared by hardware when interrupt is processed.
IT0	TCON.0	Interrupt 0 type control bit. Set/cleared by software to specify failing edge/flow level triggered External Interrupt

ในรูป T0 เป็นชื่อของบิต 0 และ TF เป็นชื่อของบิต 7 ในรีจิสเตอร์ TCON แต่ละบิตมีหน้าที่การทำงานดังนี้

IT0 Interrupt 0 เป็นบิตที่จะใช้กำหนดวิธีการขัดจังหวะโปรแกรม อันเนื่องจากสภาวะของสัญญาณที่เข้ามาทางขา INT0

ถ้า IT0 เป็น 1 จะเกิดการขัดจังหวะโปรแกรม ถ้าสภาวะของสัญญาณที่ขา INT0 เปลี่ยนจาก 1 เป็น 0

ถ้า IT0 เป็น 0 จะเกิดการขัดจังหวะโปรแกรม ถ้าสภาวะขาของสัญญาณที่ขา INT0 เป็น 0

IE0 บิตนี้จะเป็น 1 ถ้าสัญญาณที่เข้ามาทางขา INT0 มีสภาวะลอจิกของสัญญาณ ตามที่กำหนดในบิท IT0 แล้วทำให้เกิดการขัดจังหวะโปรแกรม เมื่อเกิดการกระโดด ไปทำงานยังโปรแกรมตอบสนองการขัดจังหวะแล้ว จะทำให้บิตนี้กลับเป็น 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IT1 Interrupt 1 เป็นบิทที่จะใช้กำหนดวิธีการขัดจังหวะโปรแกรมอันเนื่องจากสภาวะของสัญญาณที่เข้ามาทางขา INT1

ถ้า IT1 เป็น 1 จะเกิดการขัดจังหวะโปรแกรม ถ้าสภาวะของสัญญาณที่ขา INT1 เปลี่ยนจาก 1 เป็น 0

ถ้า IT1 เป็น 0 จะเกิดการขัดจังหวะโปรแกรม ถ้าสภาวะขาของสัญญาณที่ขา INT1เป็น 0

IE1 บิตนี้จะเป็น 1 ถ้ามีสัญญาณเข้ามาทางขา INT1 มีสภาวะลอจิกของสัญญาณ ตามที่กำหนดในบิต IT1 แล้วทำให้เกิดการขัดจังหวะโปรแกรมเหมือนกันบิต IT0 ที่ทำงานกับสัญญาณ INT0

TR0 Timer 0 Run Control Bit บิตนี้ถ้าเป็น 0 Timer 0 ไม่ทำการนับสัญญาณไม่ว่ากรณีใด ๆ ทั้งสิ้น แต่ถ้าบิตนี้เป็น 1 จะทำให้ Timer 0 ทำงาน โดยขึ้นกับสัญญาณ GATE,INT0 ข้อมูลในบิตนี้จะสามารถ Set เป็น 1 หรือ Clear เป็น 0

TF0 Timer 0 Overflow flag บิตนี้จะเป็น 1 เมื่อการนับของรีจิสเตอร์ใน Timer 0 (TL0 หรือ THO ขึ้นกับโหมดของการทำงาน) เกิด Overflow ขึ้น คือเอาการนับเพิ่มไปจนถึงค่าสูงสุดแล้วนับต่อไปทำให้ค่านับกลับมาเริ่มต้นใหม่ที่ 0 หรือค่า Reload เมื่อ 8051 กระโดดไปทำงานที่โปรแกรมตอบสนอง การขัดจังหวะจะทำให้บิตนี้กลับเป็น 0

TR1 Timer 1 Run Control Bit การทำงานจะเหมือนกับการทำงานของบิท TR0 แต่บิทนี้จะทำงานกับ Timer 1

TF1 Timer 1 Overflow Flag บิตนี้เหมือนกับบิต TF0 ต่างตันที่ขึ้นกับการทำงานของ Timer 1

4 บิตแรกที่กล่าวมานั้นจะเกี่ยวข้องกับการขัดจังหวะ (Interrupt) ส่วน 4 บิตหลังนั้นได้กล่าวมาแล้วอย่างละเอียดในเรื่องโหมดการทำงานของ Timer

ในขณะที่ Timer ทำงานในโหมดของ Timer นั้น รีจิสเตอร์ที่ทำหน้าที่เป็นตัวนับจะมีค่าเพิ่มขึ้น 1 ทุกๆ 1 ไซเคิลของเครื่อง ซึ่งเท่ากับ 12 คาบของสัญญาณจากออสซิลเลเตอร์ ในกรณีที่ Timer ทำงานเป็น Counter เพื่อนับจำนวนไซเคิลของสัญญาณที่เข้ามาทางขา T0 หรือ T1 รีจิสเตอร์จะเพิ่มค่าไป 1 เมื่อมีการเปลี่ยนสถานะของสัญญาณที่ขาดังกล่าวจาก 1 เป็น 0 โดยวงจรภายใน 8051 จะตรวจสอบสถานะของสัญญาณที่ขาดังกล่าวในช่วงเวลาเฟส 2 ของ State5 (S5P2) ในทุกๆ 1 ไซเคิลของเครื่อง เช่นในเวลา S5P2 ครั้งหนึ่งพบว่าสัญญาณที่ขา T0 มีสภาวะลอจิกเป็น 1 และในเวลา S5P2 ของไซเคิลของเครื่องถัดมาพบว่าสัญญาณที่ขา T0 มีสภาวะลอจิกเป็น 0 ก็จะทำให้ค่าในรีจิสเตอร์ตัวนับเพิ่มค่าไป 1 แต่ละไซเคิลของเครื่องจะเท่ากับ 12 ไซเคิลของสัญญาณจากออสซิลเลเตอร์ ดังนั้นสัญญาณที่จะนับได้จะต้องเป็น 1 อย่างน้อยให้ถูกจับได้ใน 1 ไซเคิลของเครื่องและ เป็น 0 อย่างน้อยก็ต้องให้ถูกตรวจจับได้ใน 1 ไซเคิลของเครื่องเช่นกัน ทำให้ความถี่สูงสุดของสัญญาณที่ขา T0 หรือ T1 ที่จะนับได้ต้องไม่เกิน 1/24 เท่าของความถี่สัญญาณจากออสซิลเลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การโปรแกรมเพื่อใช้งาน Modem

การโปรแกรมเพื่อใช้งาน Modem นั้น โดยปกติ การโปรแกรมสั่งงาน เพื่อควบคุม Modem นั้น จะใช้ AT Command เพื่อสั่งงาน ซึ่งตอนจบคำสั่งการสั่งงานจะต้องส่ง ค่า 13H ด้วย โดยในproject นี้ ใช้คำสั่งต่างๆ โดยมีความหมายดังนี้

ATX4 ให้modem แสดงผลลัพท์ต่อคำสั่งที่ได้รับ

ATLn โดย n มีค่าตั้งแต่ 0-3 เป็นการตั้งค่า speaker ให้ Modem

ATMn โดย n มีค่าตั้งแต่ 0-3 เป็นค่าให้ speaker ดัง ถึงช่วงไหน

0 ไม่ดัง

1 ดังจนกว่า connect ได้

2 ดังตลอด

3 ดังเมื่อมีการรับส่งข้อมูล

ATP โหมด pluse

ATT โหมด Tone

ATDn n มีค่า 0-9 เป็นคำสั่งในการโทรออก

ATE1S0=1 สั่งให้ Modem รอรับสาย

ATH=0 วางสาย

โดยmodem จะแจ้งเหตุการณ์ต่างๆให้โปรแกรมดังนี้

CONNECT เมื่อmodem มีการConnect สำเร็จ

OK เมื่อmodem ทำตามคำสั่งที่ได้รับเรียบร้อย

RING เมื่อmodem ได้รับสัญญาณ ring tone (มีการโทรเข้ามา)

BUSY สายไม่ว่าง

NO DIAL TONE ไม่มีสัญญาณ

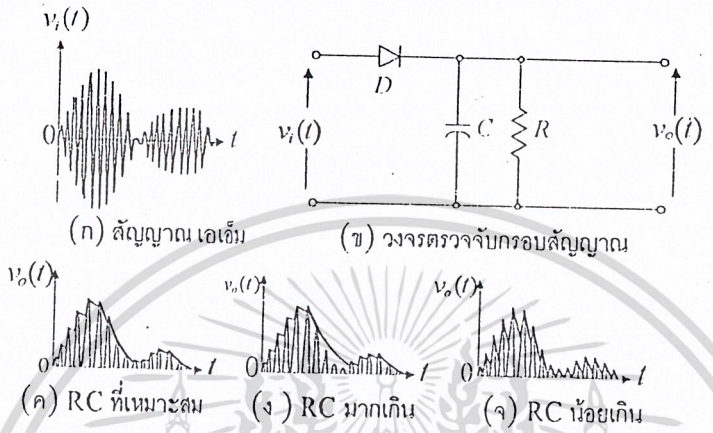
เราสามารถทราบได้ว่า สายโทรศัพท์ ขาด หรือ มีการวางสายได้ จากการเปลี่ยนแปลงสัญญาณ CDHOLDING จากcomport ที่ต่อกับmodem ได้ โดยจะเป็น 1 เมื่อมีการconnect อยู่ จะเป็น 0 เมื่อมีการวางสาย ส่วนcomport ที่ใช้ติดต่อกับ8051 นั้นให้ 28800bps ทำงานในโหมด 1 เพราะฉะนั้นจะส่งได้เร็วสุด 2880 byte/sec แต่ A to D นั้นจะต้องให้เวลาในการแปลงค่าประมาณ 8 Clock โดย Clock มีความถี่ที่640 kHZ

เพราะฉะนั้นเวลาส่วนใหญ่ที่เป็น เวลาของการส่งข้อมูล ถ้าสัญญาณ sine ที่คาบเวลาจะต้องให้ การแซมปลิ้ง 16 ค่า อัตราความถี่สูงสุดที่ได้จะอยู่ที่ 180 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 วงจรตรวจจับกรอบสัญญาณ

วงจรตรวจจับกรอบสัญญาณ คือวงจรที่ให้เอาท์พุตมีลักษณะเป็นไปตามกรอบ หรือแนวทางเดินของยอดคลื่นของสัญญาณอินพุต วงจรตรวจจับกรอบสัญญาณแบบง่ายจะเป็นวงจรชนิดไม่เชิงเส้น ดังแสดงในรูป 2.28 การทำงานของวงจรนี้สามารถอธิบายได้ดังนี้ คือ เมื่อมีสัญญาณ $\phi_{Am}(t)$ ตามแสดงในรูป 2.28 ก



รูปที่ 2.28 การทำงานของวงจร

เข้ามาสู่วงจรในตอนแรกนั้น ขณะที่สัญญาณมีค่าเพิ่มขึ้นในทางบวก ไดโอด D จะนำกระแส ทำให้เกิดการประจุตัวเก็บประจุ C ให้มีให้มีแรงดันเอาท์พุตเกิดขึ้นคร่อม C ตามคุณการเพิ่มของสัญญาณอินพุท จนกระทั่งถึงค่าพีก (peak) หรือยอดของสัญญาณ แต่ต่อจากนี้เมื่ออินพุต $\phi_{Am}(t)$ ลดลง ก็จะทำให้ค่าแรงดันไฟฟ้าทางอาโนดของ D ต่ำกว่าคาโทด ในสถานะการเช่นนี้ไดโอด D จะหยุดนำกระแสและกั้นไม่ให้ C คายประจุผ่านตัวมันไปได้ดังนั้นการคายประจุของ C จึงมีทางเดียว คือ คายประจุผ่าน R ไปอย่างช้าๆ จนกระทั่งถึงเวลาที่ $\phi_{Am}(t)$ จะมีค่ากลับเป็นบวก และมีค่าเพิ่มขึ้นมากกว่าแรงดันไฟฟ้าคร่อม C ไดโอด D จึงจะทำกระแสอีก และจะเกิดการประจุของ C ทำให้เอาท์พุตเป็นไปตามค่าของสัญญาณ $\phi_{Am}(t)$ อีกไปจนถึงค่ายอดของคลื่นพาห์ถัดมา แล้วพฤติกรรมการทำงานก็จะเกิดซ้ำในลักษณะเดิมอีก กล่าวคือเกิดการคายประจุของ C ผ่าน R อีกครั้งหนึ่งพฤติกรรมเช่นนี้จะเกิดซ้ำแล้วซ้ำอีกเรื่อยไป ดังนั้น ถ้าเราเลือกค่าคงตัวเชิงเวลา (time constant) RC ที่ใช้ให้เหมาะสม ก็จะได้เอาท์พุตของสัญญาณเป็นไปตามเส้นทึบดังแสดงในรูป 2.28 ค แต่ถ้าค่าคงตัวเชิงเวลาของวงจรตรวจจับกรอบของสัญญาณมีค่ามากเมื่อเทียบกับค่าคาบเวลาที่สั้นที่สุดของสัญญาณที่ประกรอบอยู่ใน m(t) แล้วก็จะทำให้เอาท์พุตของวงจรเกิดลดตามยอดคลื่นของ $\phi_{Am}(t)$ ไม่ทัน และจะทำให้เกิดการผิดเพี้ยนของสัญญาณเอาท์พุต ดังแสดงในรูป 2.28 ง และในกรณีที่เกิดมีการใช้ RC มีค่าน้อยมาก กล่าวคือค่าผลคูณของ RC มีค่าน้อย เมื่อเทียบกับคาบเวลาของคลื่นพาห์ก็จะทำให้เกิดการกระเพื่อม(ripple) บนสัญญาณเอาท์พุทมากดังแสดงในรูป 2.28ก อย่างไรก็ดี ค่าความถี่ของคลื่นพาห์ที่ใช้ในทางปฏิบัติ จะมีสูงกว่าความถี่ของสัญญาณที่เป็นส่วนประกอบของ m(t) มาก หรือกล่าวอีกในหนึ่งได้ว่า m(t) จะเปลี่ยนแปลงช้ามากเมื่อเทียบกับคลื่นพาห์ที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จึงทำให้การเลือกค่าคงตัวของวงจร RC ทำได้อย่างเหมาะสมง่าย กล่าวคือจะทำให้การกระเพื่อมบนเอาท์พุตของวงจรมีค่าน้อยมากได้จนไม่ต้องคำนึงถึงในทางปฏิบัติ เงื่อนไขสำหรับการเลือกค่า RC ที่เหมาะสม สามารถสรุปได้ในรูปของอสมการดังต่อไปนี้คือ

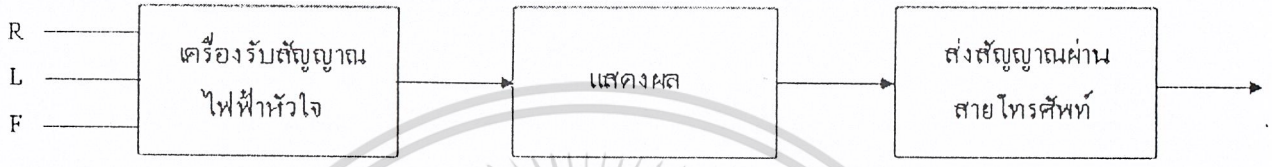
$$\frac{1}{f_c} << RC << \frac{1}{f_m}$$

โดยในที่นี้ f_c และ f_m คือ ค่าความถี่คลื่นพาห์ และค่าสูงที่สุดที่ประกอบอยู่ในสัญญาณข่าวสาร $m(t)$ ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การคำนวณและการสร้าง

3.1 บล็อกไดอะแกรม



รูปที่ 3.1 บล็อกไดอแกรมของโครงงาน

รูปที่ 3.2 บล็อกไดอะแกรมของเครื่องรับสัญญาณไฟฟ้าหัวใจ

3.2 เครื่องรับสัญญาณไฟฟ้าหัวใจ

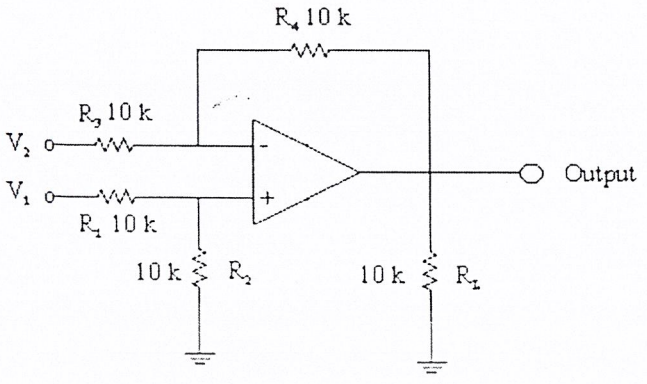
ในภาคนี้ประกอบด้วยวงจรต่างๆดังต่อไปนี้

3.2.1 ส่วนที่ทำหน้าที่ขยายคลื่นไฟฟ้าหัวใจ

วงจรดิฟเฟอร์เรนเชียล แอมพลิฟายเออร์ (Differential Amp.)

มีลักษณะการทำงาน คือเป็นตัวขยายความแตกต่างของสัญญานที่แตกต่างกันและจะทำการReject สัญญาณที่มีความเหมือนกัน (Common)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 วงจร Differential Amp

โดยที่แรงดันเอาท์พุทหาได้จาก

$$V_o = \left[\left(\frac{R_3 + R_4}{} \right) x \left(\frac{R_2}{R_1 + R_2} \right) x V_1 \right] - \left(\frac{R_4}{R_3} x V_2 \right)$$

เมื่อ $R_1 = R_2 = R_3 = R_4$ อัตราขยายความแตกต่าง เท่ากับ 1

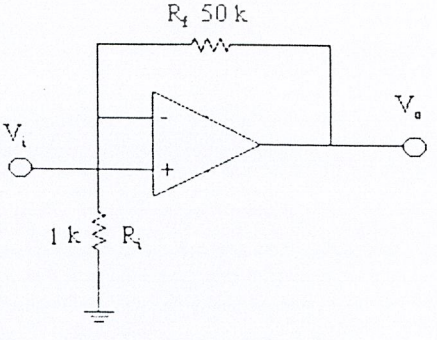
เอาท์พุทจะเท่ากับ

$$V_o = (V^+ - V^-) A$$

$$V_o = (V^+ - V^-) (1)$$

เมื่อ A เป็นอัตราขยาย V^+ เป็นแรงดันที่ขั้วบวก V^- เป็นแรงดันที่ขั้วลบ

วงจร Non-Inverting Amplifier



รูปที่ 3.41 รูปวงจร Non-Inverting Amplifier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

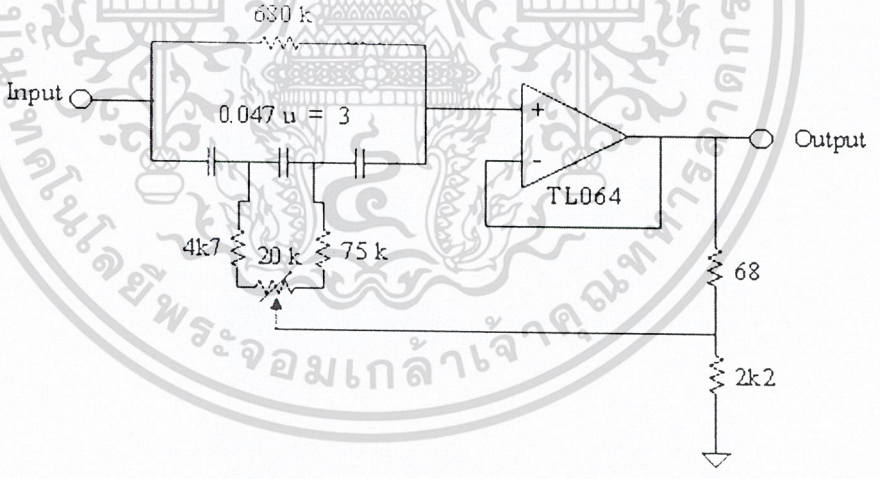
$$\text{Voltage Gain } (A_v) = R_f / R_i$$

$$= 50 \text{ เท่า}$$

วงจรขยายในภาคแรกไม่ควรให้มีอัตราขยายมากเกินไป เพราะว่าถ้าเกิดมีศักดาไฟฟ้าออฟเซ็ทเกิดขึ้น ทำให้สัญญานออกที่เอาท์พุทลอยขึ้นหรือต่ำไปจากระดับศูนย์มาก บางครั้งอาจอิ่มตัว อยู่ที่ค่าศักดาไฟฟ้าของแหล่งจ่ายไฟทำให้วงจรไม่สามารถทำงานได้

3.2.2 วงจรกรองความถี่ต่ำแบบนอท์ชฟิวเตอร์ (Low – Pass Notch Filter)

ถึงแม้ว่าวงจรขยายในภาคแรกจะเป็นวงจรขยายความแตกต่างซึ่งมีคุณสมบัติในการกำจัดสัญญาณรบกวนได้ก็ตาม แต่ถ้าเกิดความไม่สมดุลของวงจรขึ้นมา สัญญาณรบกวนซึ่งส่วนใหญ่มีความถี่ 50 เฮิร์ทซ ก็สามารถผ่านไปได้ แต่เนื่องจากคลื่นไฟฟ้าหัวใจมีความถี่ต่ำอยู่ในช่วง 0.5 - 200 เฮิร์ทซ ถ้าเราใช้วงจรกรองความถี่ต่ำแบบธรรมดาที่ยอมให้ความถี่ต่ำกว่า 50 เฮิร์ทซผ่านไปได้ก็จะทำให้สัญญาณไฟฟ้าหัวใจส่วนที่สูงกว่า 50 เฮิร์ทซถูกกำจัดออกไปดังนั้น เราจึงต้องใช้วงจรกรองความถี่ต่ำแบบ Low – Pass Notch Filter ซึ่งจะยอมให้สัญญาณความถี่สูงและต่ำกว่า 50 เฮิร์ทซผ่านไปได้ส่วนสัญญาณรบกวน 50 เฮิร์ทซ จะถูกกำจัดออกไปวงจรแสดงดังรูปที่ 3.42

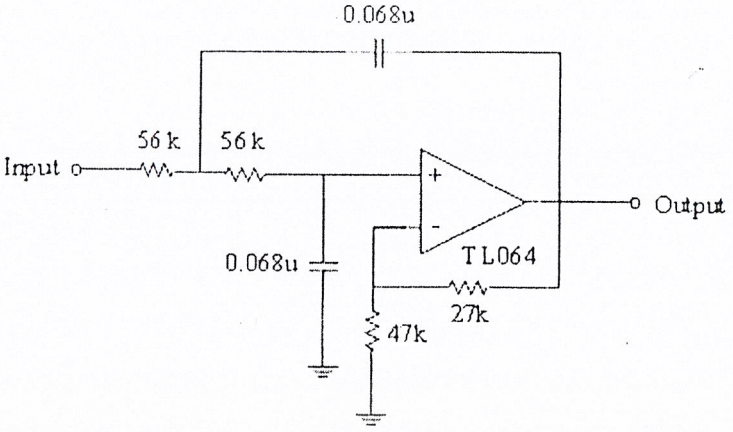


รูปที่ 3.42 วงจรกรองความถี่ต่ำแบบนอท์ชฟิลเตอร์

3.2.3 วงจรกรองความถี่ต่ำแบบแอคทีฟ (Active Low – Pass Filter)

เป็นวงจรกรองความถี่ต่ำที่ยอมให้ความถี่มในช่วงความถี่ของสัญญาณคลื่นไฟฟ้าหัวใจผ่านได้ ดังแสดงในรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 วงจรกรองความถี่ต่ำแบบแอคทีฟ

วงจรนี้เป็น Active Low – Pass Filter แบบ Sallen and Key Circuit ซึ่งจากหัวข้อ 2.3.3.2 จะได้ว่า ความถี่คัทออฟ (Cutoff Frequency or Passband Edge Frequency) หรือ

$$(f)_p = \frac{1}{\sqrt{(56k \times 0.068u)^2}} = 262 \text{ เฮิร์ทซ}$$

และ

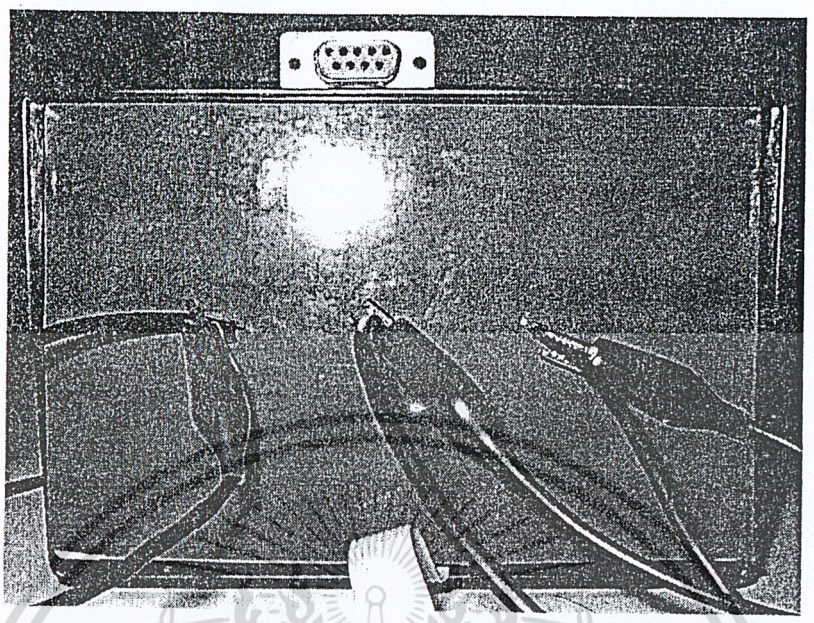
$$k = 1 + \frac{27\,k}{47\,k} = 1.5745$$

ดังนั้น

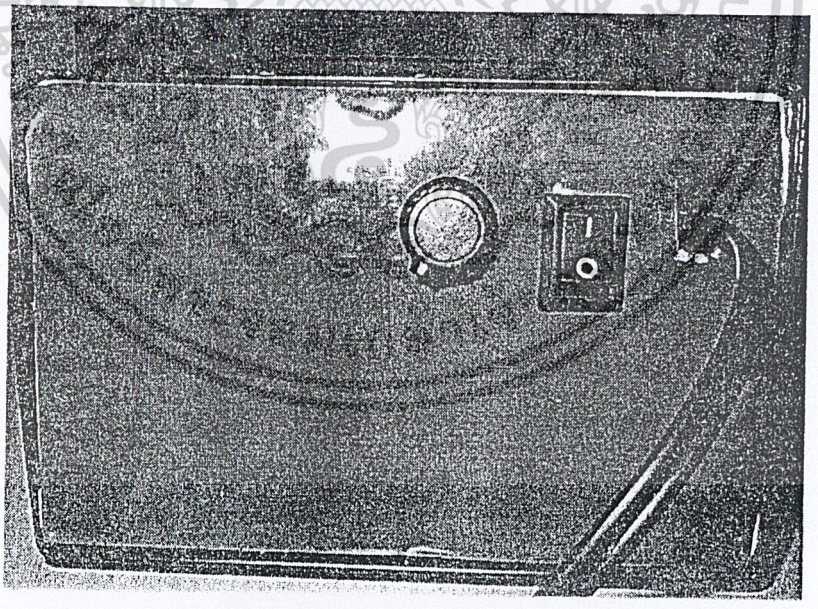
$$\text{Pole Q } (Q_p) = \frac{262}{\frac{1}{56k \times 0.068u} + \frac{1}{56k \times 0.068u} + \frac{1-1.5745}{56k \times 0.068u}}$$

$$\approx 0.7$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

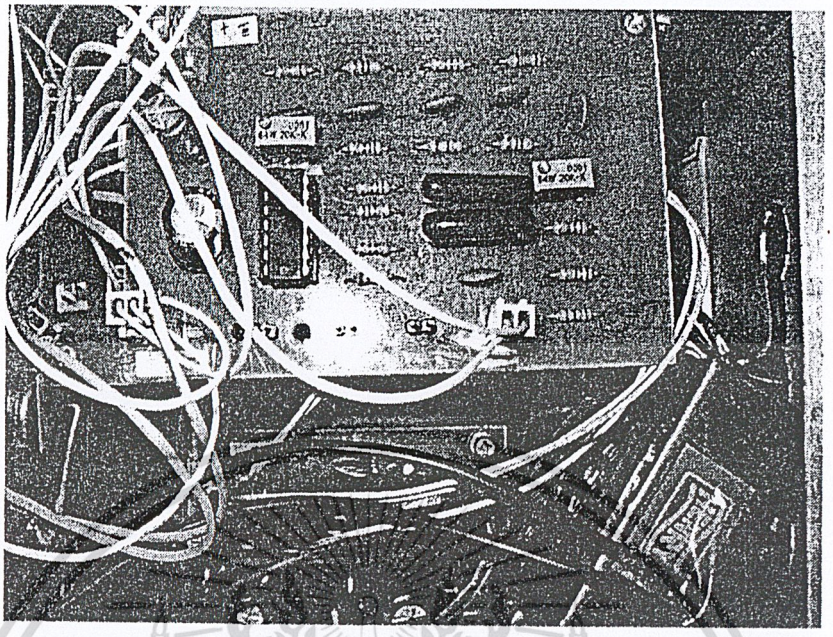


รูปที่ 3.6 รูปฮาร์ดแวร์ด้านหน้า



รูปที่ 3.7 รูปฮาร์ดแวร์ด้านหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

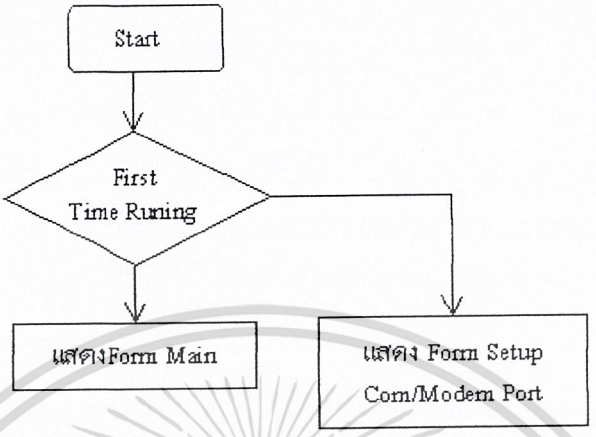


รูปที่ 3.8 รูปแสดงภายในส่วนของฮาร์ดแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

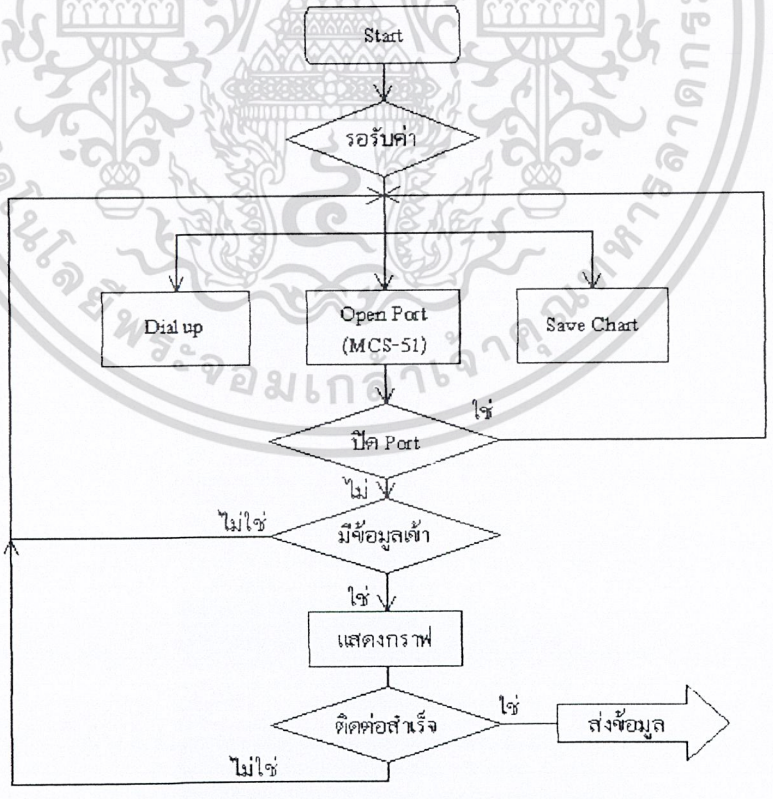
3.3 การออกแบบโปรแกรม

การทำงานของโปรแกรม ECG ตัวส่ง



รูปที่ 3.9 แสดงการทำงานของโปรแกรม ECG ตัวส่ง

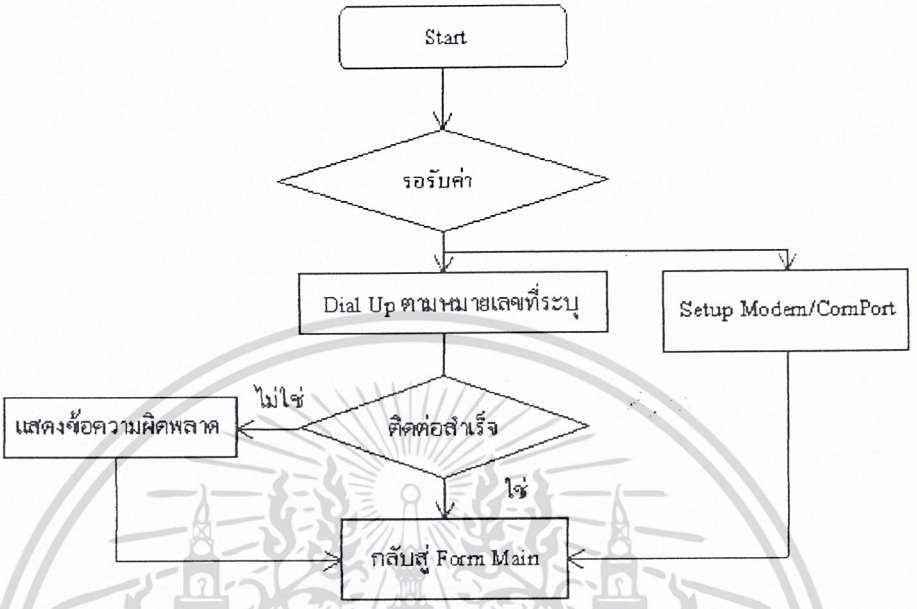
ที่ Form Main



รูปที่ 3.10 แสดงการทำงานของโปรแกรม ECG ที่ Form Main

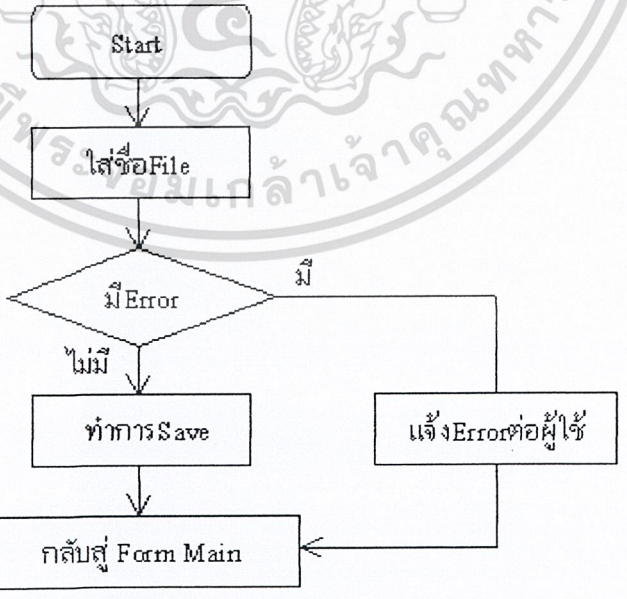
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานในขั้นตอนDial Up



รูปที่ 3.11 แสดงการทำงานในขั้นตอนDial Up

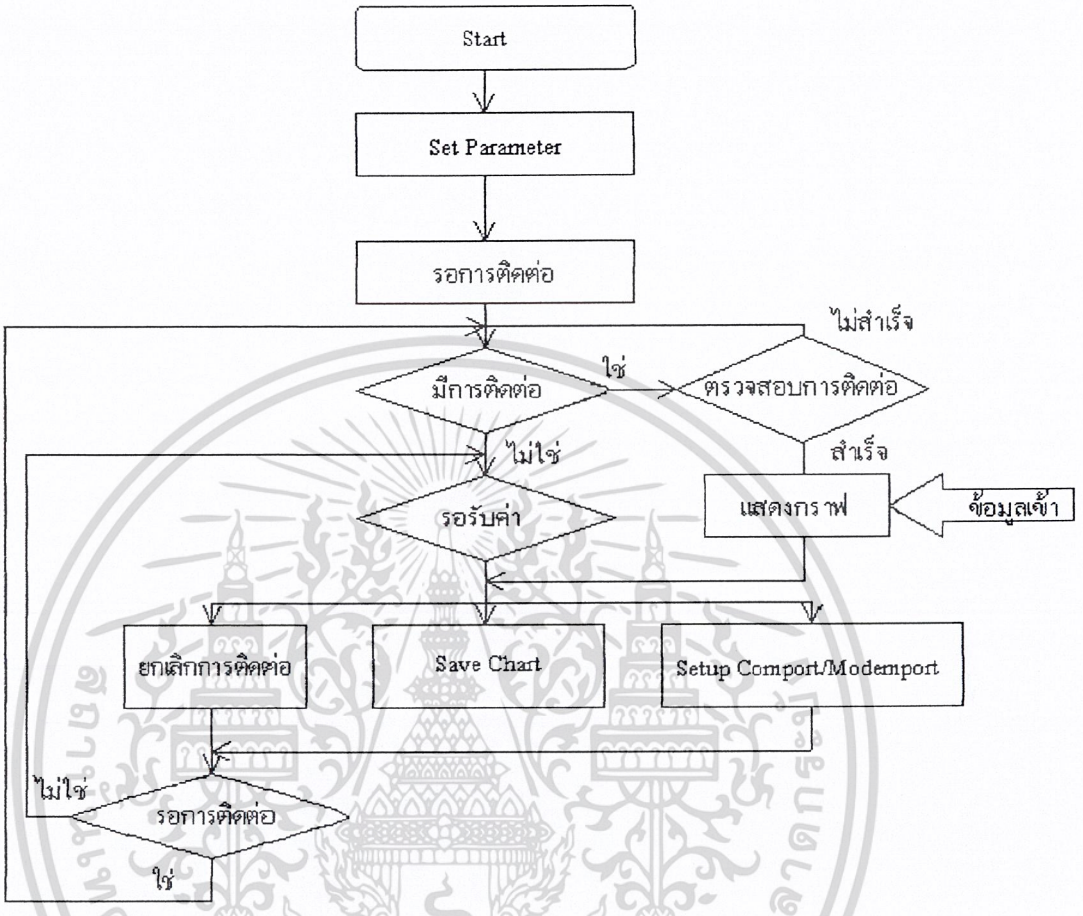
ขั้นตอนการ Save



รูปที่ 3.12 แสดงการทำงานในขั้นตอน การ Save

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

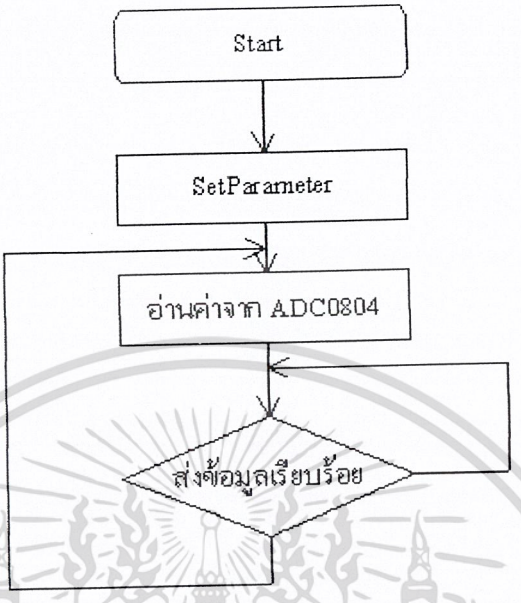
โปรแกรมECG ตัวรับ



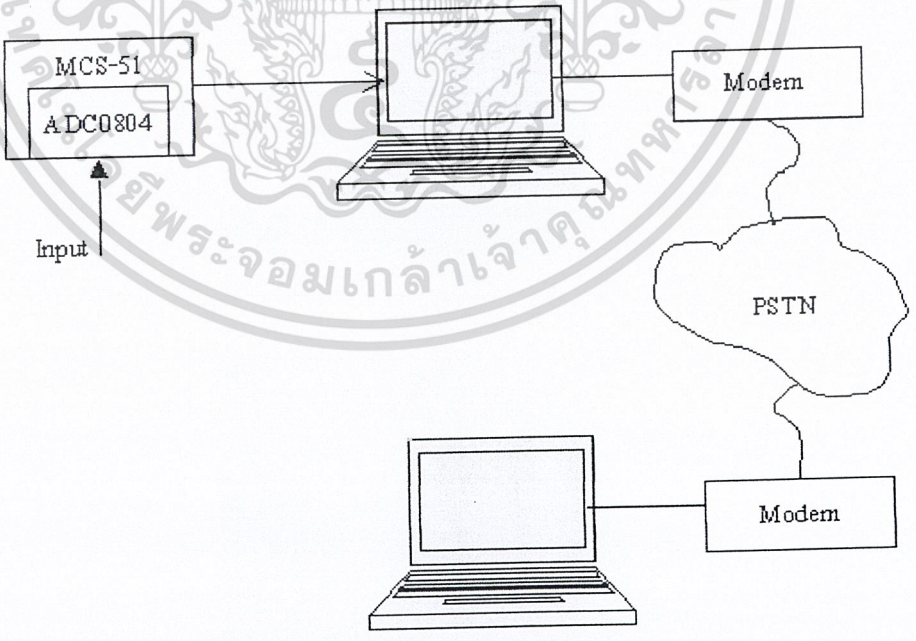
รูปที่ 3.13 แสดงการทำงานของโปรแกรม ECG ตัวรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MCS-51



รูปที่ 3.14 แสดงการทำงานของโปรแกรม MCS-51



รูปที่ 3.15 แสดงการเชื่อมต่อของโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

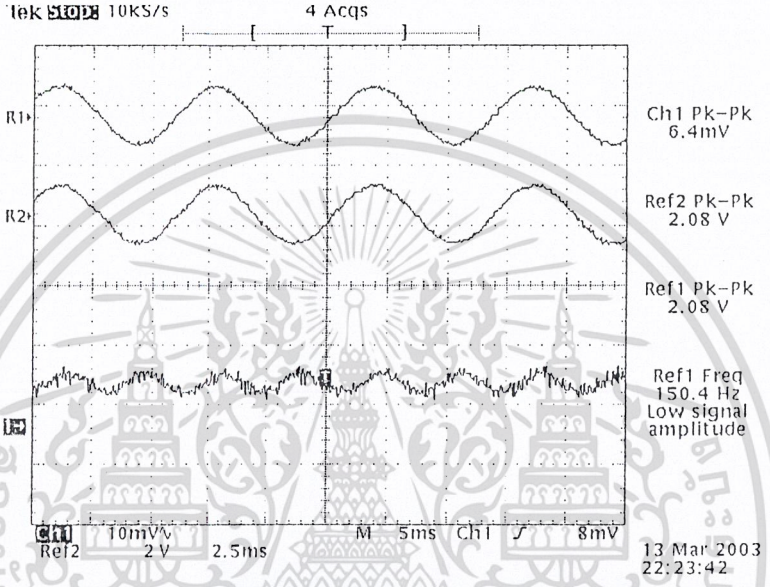
บทที่ 4

การทดลองและผลการทดลอง

4.1 วงจรดิฟเฟอร์เรนเชียล แอมพลิฟายเออร์

4.1.1 การหา คอมมอนโหมดเกน

R1 R2 คือสัญญาณอินพุต 2 V_{pp} ที่เข้าขาบวก ลบ ออปแอมป์



รูปที่ 4.1 การหาคอมมอนโหมดเกนที่ความถี่ 150 Hz

ความถี่	เอาท์พุต
50 Hz	6 mV
100 Hz	6.2 mV
150 Hz	6.4 mV
200 Hz	6.8 mV
300 Hz	6.6 mV
500 Hz	6.8 mV
1000 Hz	7.0 mV

ตารางที่ 4.1 ตารางค่า คอมมอนโหมดเกน

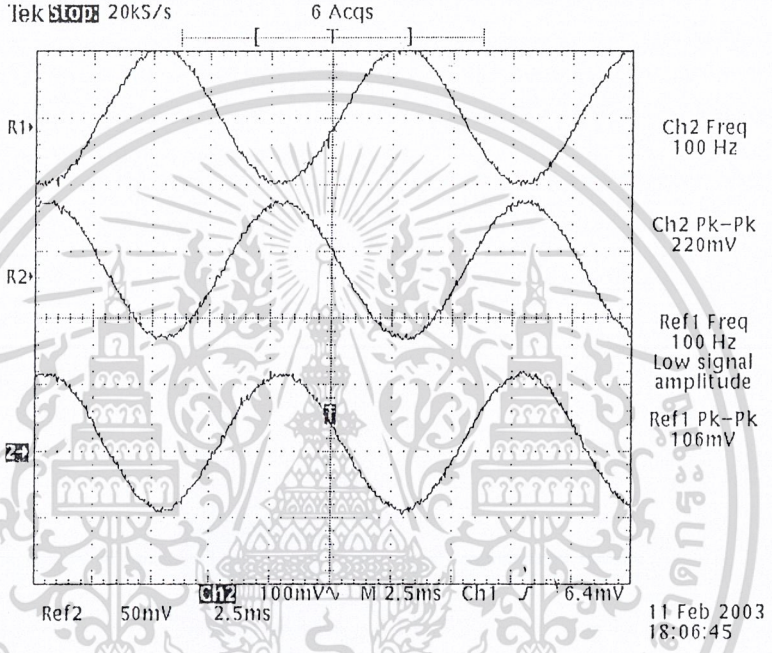
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าคอมมอนโหมดเกนเฉลี่ยเท่ากับ

$$A^{+} - A^{-} = V_o / V_i$$
$$= 6.62 \text{ mV}$$

4.1.2 การหาดิฟเฟอร์เรนเชียลโหมดเกน

R1 R2 คือสัญญาณอินพุต 100 mV_{pp} แต่กลับเฟสกัน เข้าขาบวก ลบ ออปแอมป์



รูปที่ 4.2 การหาดิฟเฟอร์เรนเชียลโหมดเกนที่ความถี่ 100 Hz

ความถี่	เอาท์พุต
50 Hz	202 mV
100 Hz	204 mV
150 Hz	206 mV
200 Hz	212 mV
300 Hz	204 mV
500 Hz	206 mV
1000 Hz	206 mV

ตารางที่ 4.2 ตารางค่า ดิฟเฟอร์เรนเชียลโหมดเกน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าดิฟเฟอร์เรนเชียลเกนเฉลี่ยเท่ากับ

$$\frac{A^{+} - A^{-}}{2} = V_o / 2V_I$$

$$= 1$$

ในการวงจรการออกแบบวงจรกำหนดให้ เกน = 1

4.13 การหาCommond Mode Rejection Ratio (CMRR)

$$\text{ค่า } CMRR = 20\log \frac{\text{differential Mode Gain}}{\text{Commond Mode Gain}}$$

$$= 45.12 \text{ db}$$

4.2 วงจรกรองความถี่ต่ำแบบนอท์ชฟิลเตอร์

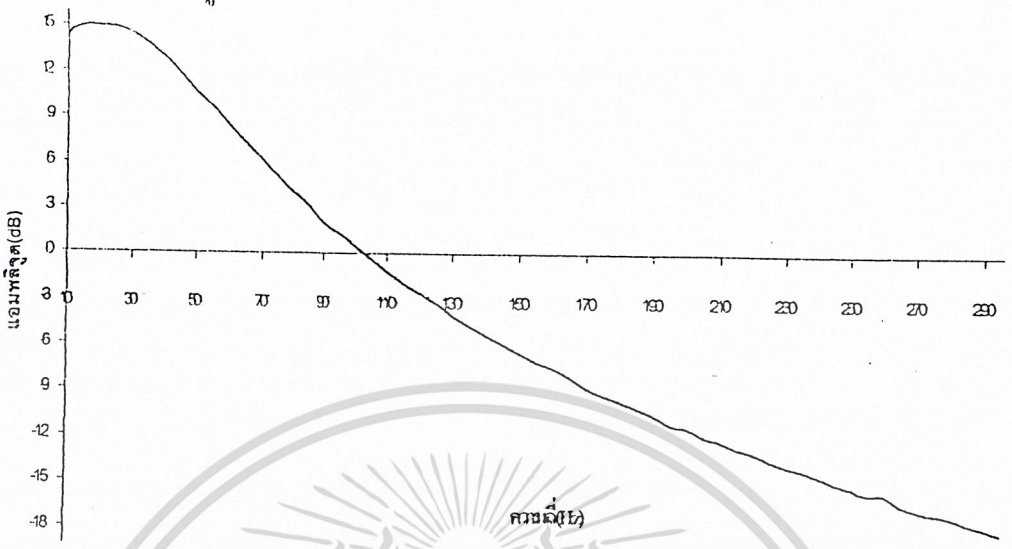
ทำการป้อนสัญญาณไซน์เป็นอินพุทเข้าวงจรด้วยขนาดคงที่ค่าหนึ่ง จาก10 เฮิรท์ไปถึง 200 เฮิรท์ ได้ผลการทดลองดังรูปกราฟ

รูปที่ 4.3 ผลตอบสนองทางความถี่ของวงจรกรองความถี่ต่ำแบบนอท์ชฟิลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 วงจรกรองความถี่ต่ำแบบแอคทีฟ

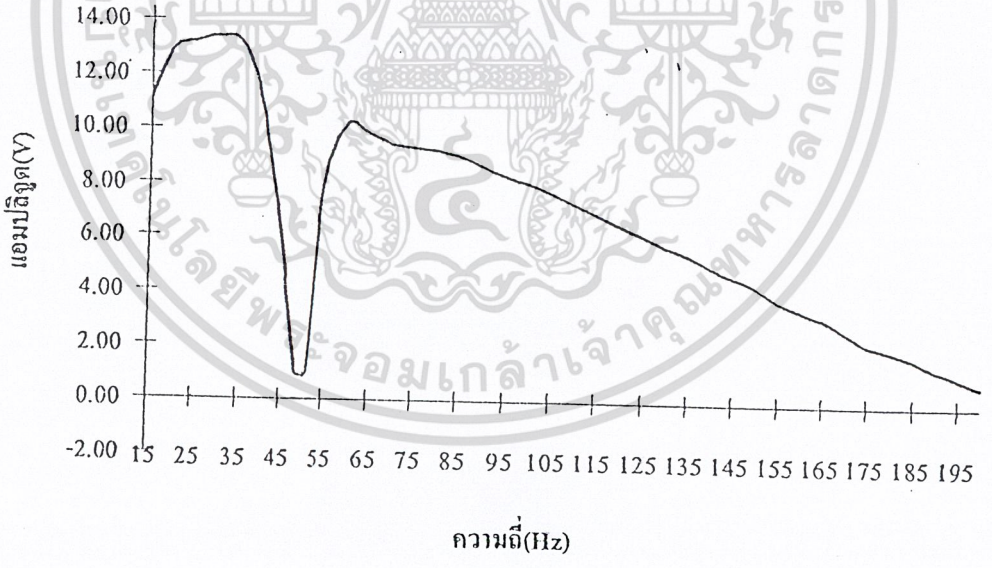
ทำการป้อนสัญญาณไซน์เป็นอินพุทเข้าวงจรด้วยขนาดคงที่ค่าหนึ่ง จาก10 เฮิรท์ไปถึง 300 เฮิรท์ ได้ผลการทดลองดังรูปกราฟ



รูปที่ 4.4 ผลตอบสนองทางความถี่ของวงจรกรองความถี่ต่ำ

4.4 เอาท์พุทรวมของวงจร

เมื่อทำการวัดเอาท์พุทรวมของวงจร ได้ผลการทดลองเป็นดังรูปกราฟ

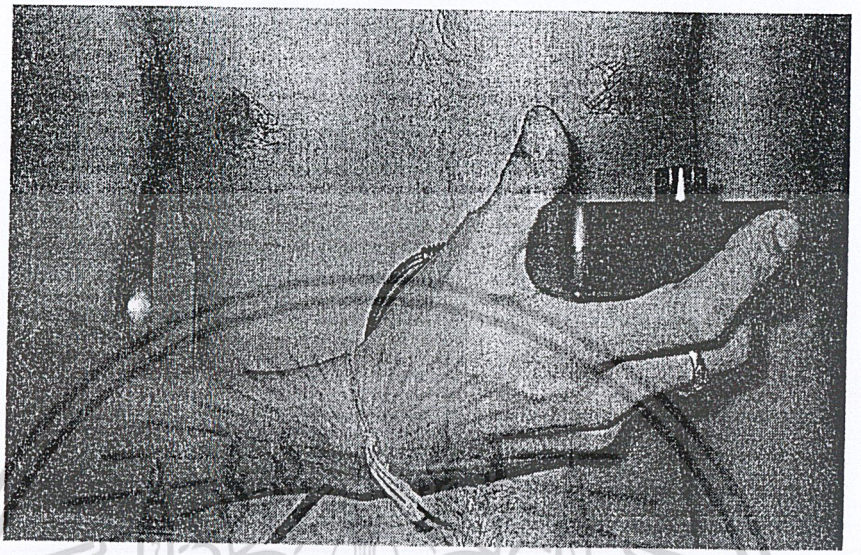


รูปที่ 4.5 เอาท์พุทรวมของวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

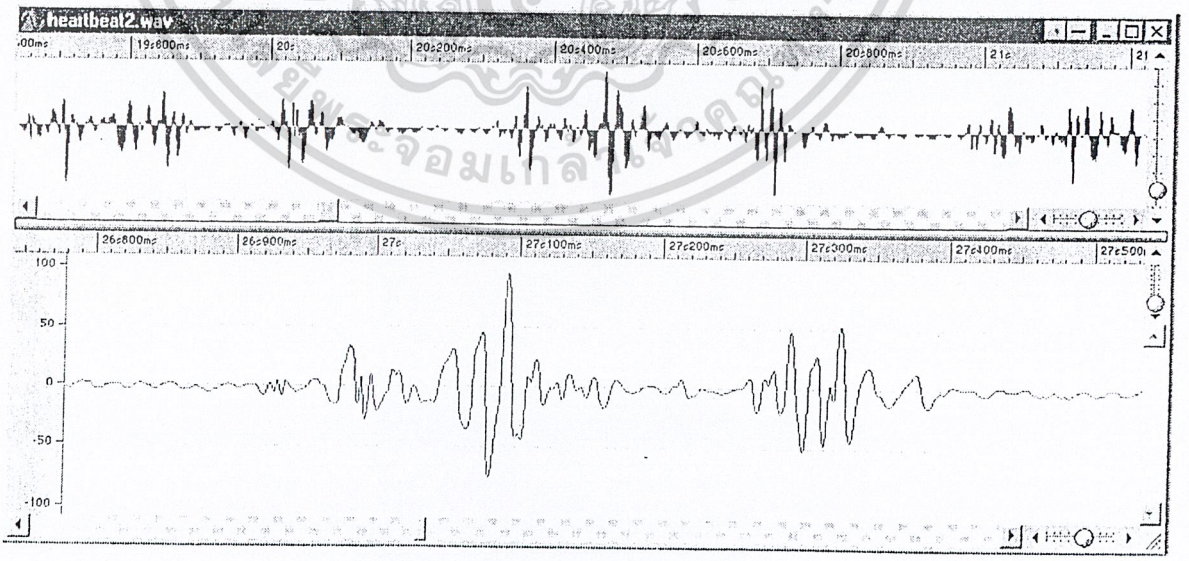
4.5 เอาท์พุทเมื่อใช้สัญญาณ ECG เป็นอินพุท

ทำการวัดสัญญาณจากร่างกายโดยใช้เครื่อง The Doppler แทนแผ่นอิเล็คโทรด โดยแนบกับลำตัว ตามรูป



รูป 4.6 แสดงการใช้งานเครื่อง The Doppler

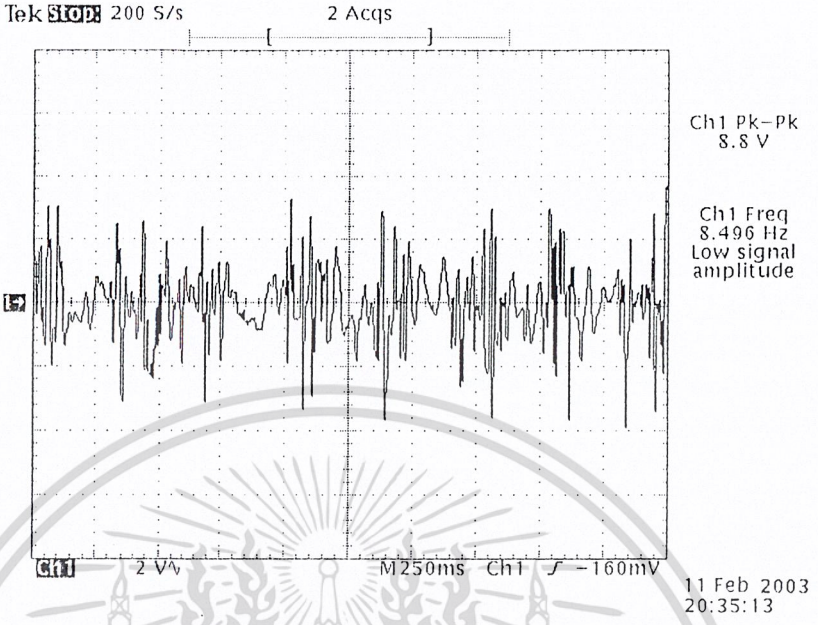
คลื่นสัญญาณหัวใจที่ได้จากเครื่อง The Doppler โดยใช้โปรแกรมเวฟแลบ



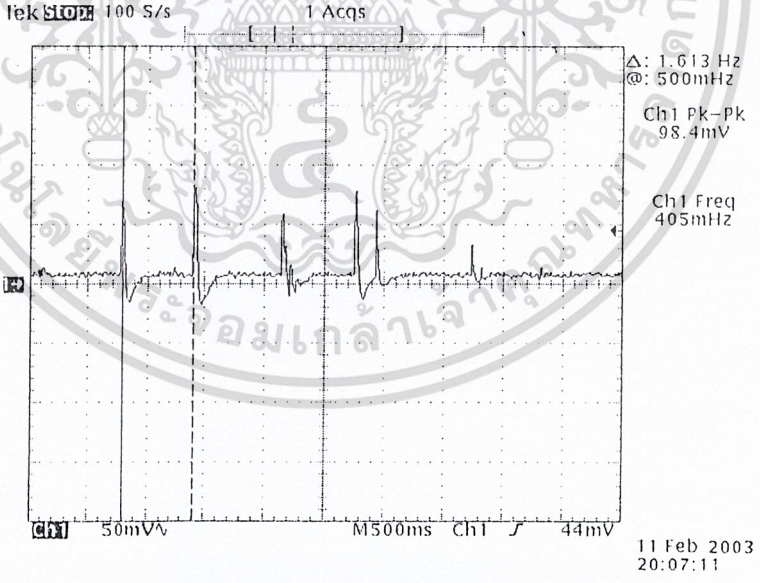
รูปที่ 4.7 รูปคลื่นหัวใจที่ได้จากเครื่อง The Doppler

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปคลื่นหัวใจที่ได้จากเครื่อง The Doppler โดยใช้ สโคปจับ



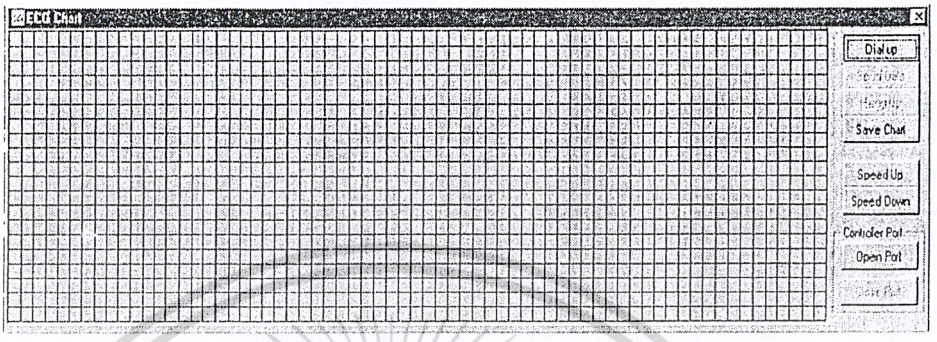
รูปที่ 4.8 รูปสัญญาณหัวใจได้จากเครื่อง The Doppler



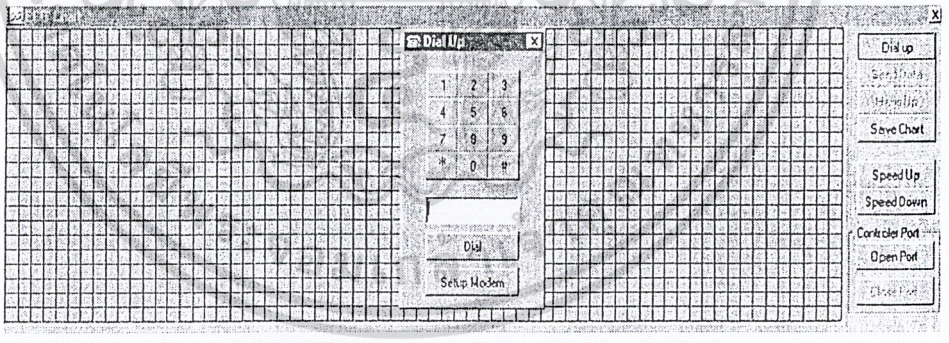
รูปที่ 4.9 รูปสัญญาณหัวใจได้จากเครื่อง The Doppler ผ่านวงจรตรวจจับสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทำงานของโปรแกรมแต่ละขั้นตอน

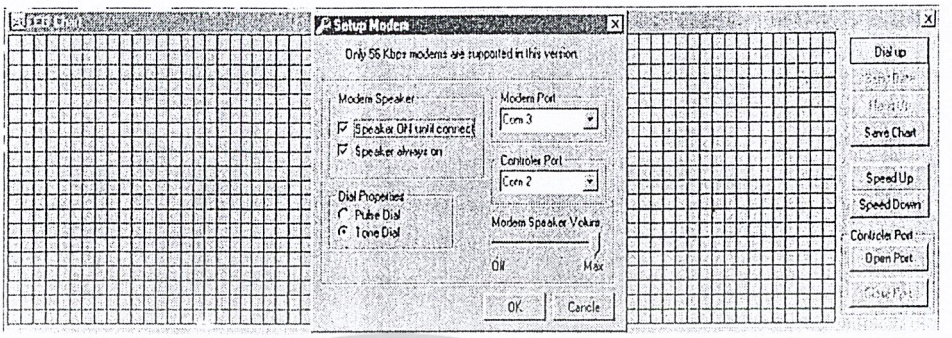


รูปที่ 4.10 แสดงรูปโปรแกรมด้านส่ง

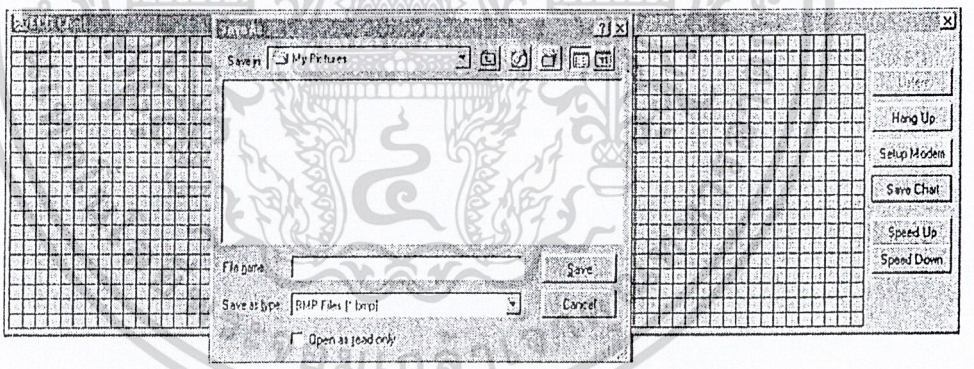


รูปที่ 4.11 รูปแสดงโปรแกรมส่วนการ Dial

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

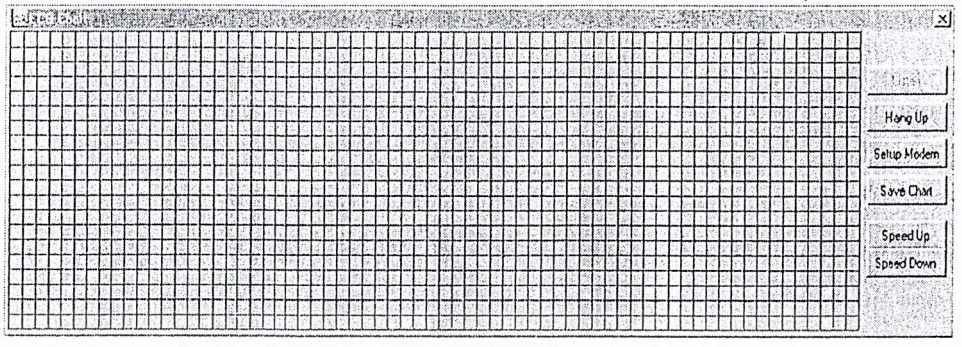


รูปที่ 4.12 รูปแสดงโปรแกรมส่วนการ Setup Modem ทางด้านส่ง

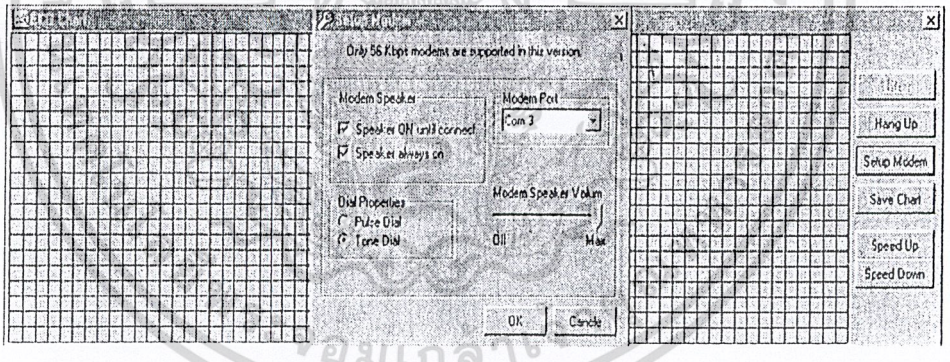


รูปที่ 4.13 รูปแสดงโปรแกรมส่วนการ Save ทางด้านส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



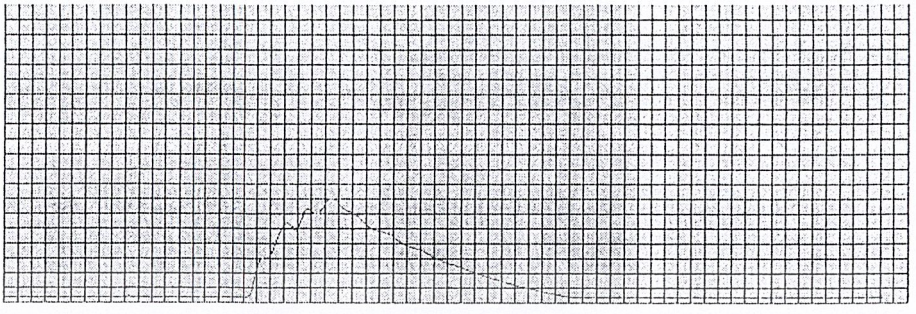
รูปที่ 4.14 แสดงรูปโปรแกรมด้านรับ



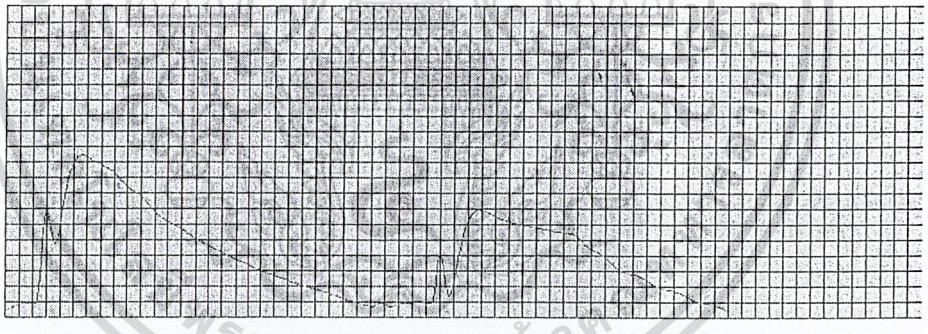
รูปที่ 4.15 รูปแสดงโปรแกรมส่วนการ Setup Modem ทางด้านรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทำงานแสดงในโปรแกรมภาคส่งและภาครับ



รูปที่ 4.16 รูปแสดงผลที่ด้านส่ง



รูปที่ 4.17 รูปแสดงผลที่ด้านรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุป

จะพบได้ว่าโปรเจคนี้ มีประโยชน์ทางด้านการแพทย์โดยอำนวยความสะดวกในด้านระยะทาง ประหยัดเวลาในการรักษา ซึ่งจะทำการส่งสัญญาณคลื่นไฟฟ้าหัวใจระหว่างคอมพิวเตอร์ ผ่านโครงข่ายโทรศัพท์ ควบคุมโดยโปรแกรม ทำให้ไม่ต้องเผชิญปัญหาผู้ป่วยและแพทย์ผู้วินิจฉัยอยู่ห่างไกลกันหรือไม่สะดวกจะไปพบแพทย์ได้

ปัญหาที่ได้จากการทดลองจริงพบว่ามีข้อบกพร่องอุปสรรคบางประการ เช่นแผ่นอิเลคโทรดที่ใช้ในการทดลองไม่มีขายทั่วไป และราคาแพงหายาก ทำให้มีข้อจำกัดในการทดลอง แนวทางแก้ไขปัญหาคือ การใช้เครื่อง The Doppler และวงจรดีเทคสัญญาณ ตรวจจับสัญญาณออกมา และปัญหาสัญญาณรบกวนที่มากเกินไป ทำให้รูปสัญญาณ ECG ผิดเพี้ยนไป

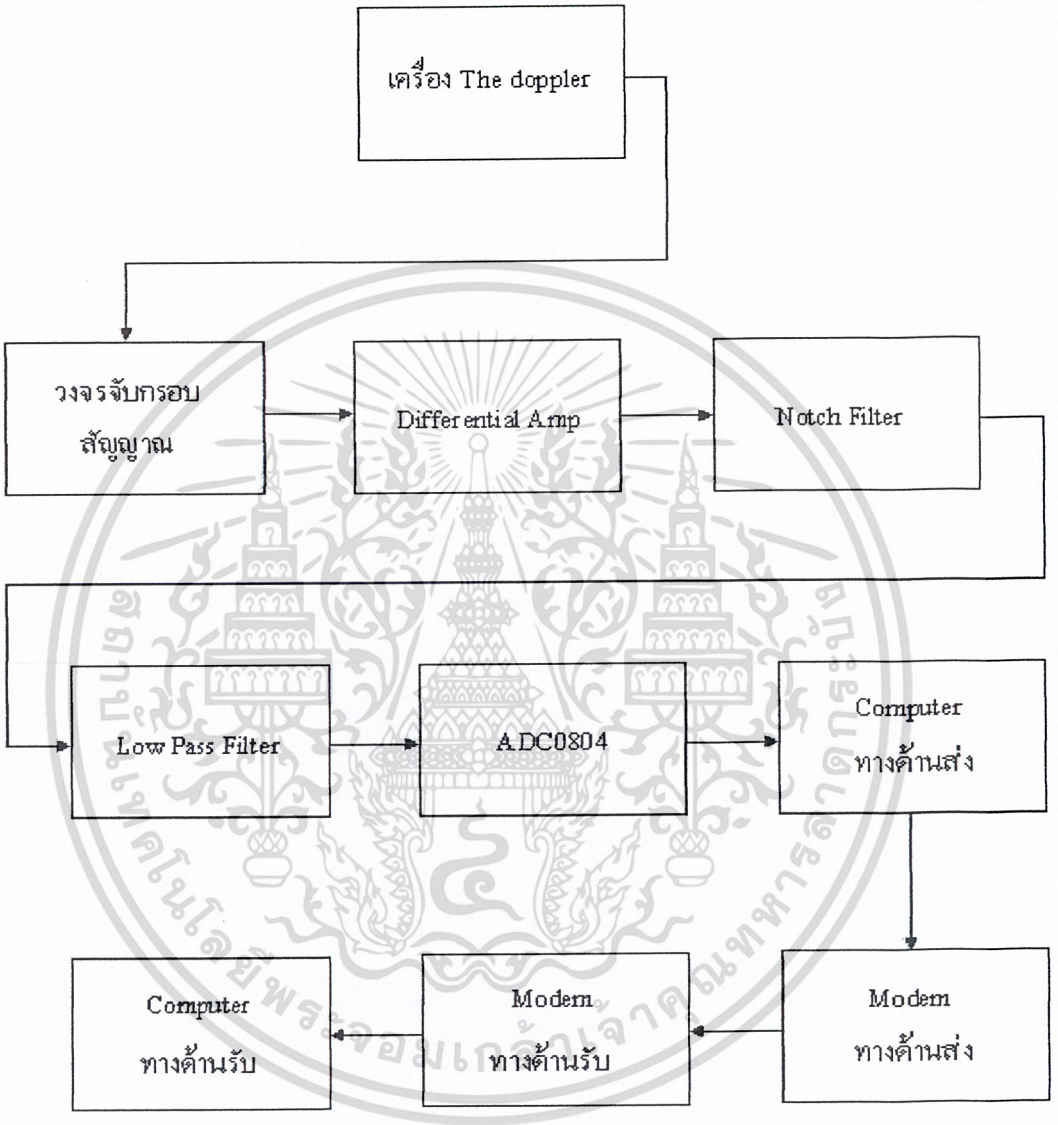
ในส่วนของโปรแกรมที่ใช้รับส่งข้อมูลนั้นใช้การส่งแบบ real time ซึ่งจะขึ้นอยู่กับความสามารถของเครื่องคอมพิวเตอร์ที่ใช้ด้วยว่ามีความสามารถในการส่งและประมวลผลมากน้อยแตกต่างกันออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

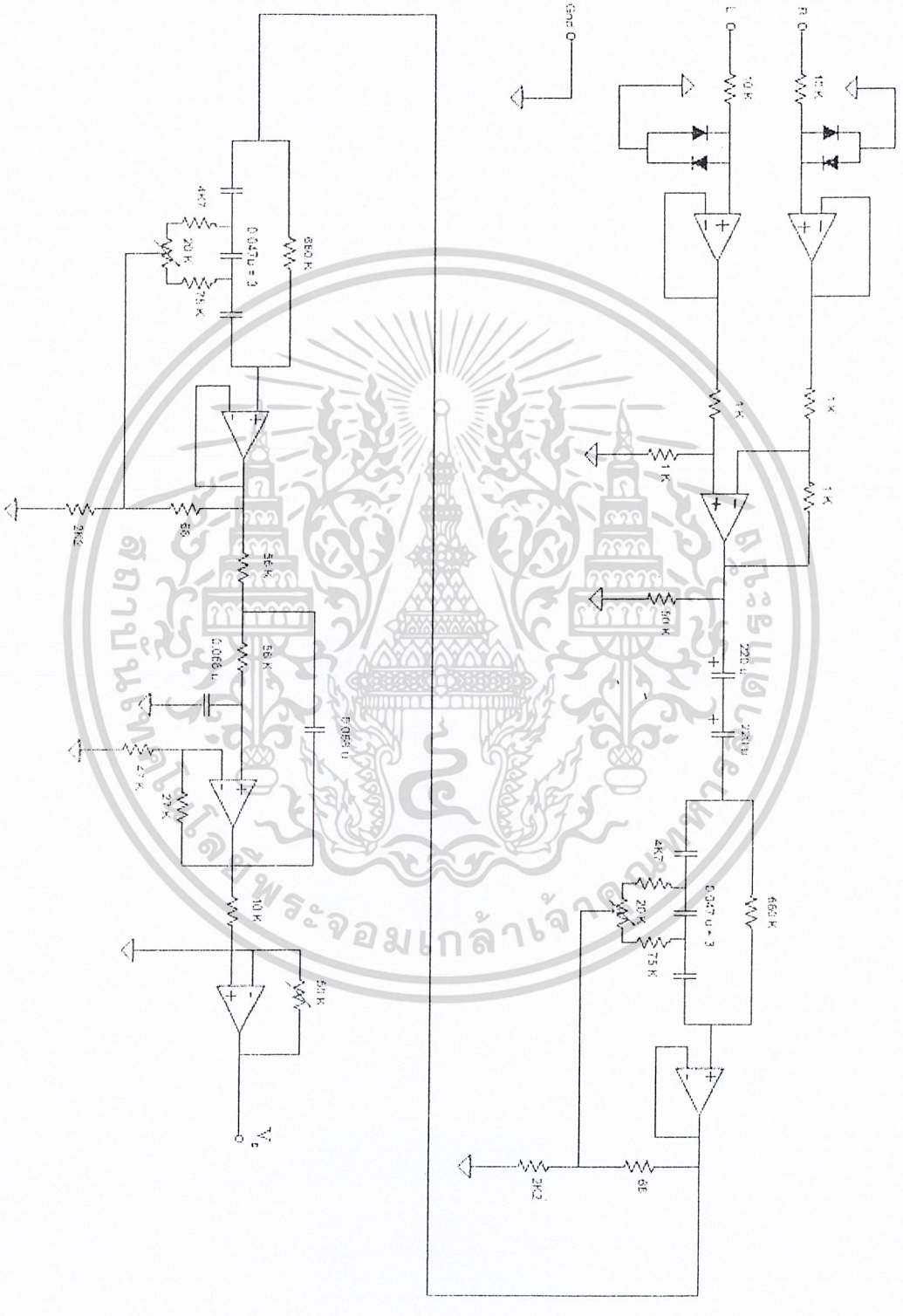
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บล็อกไดอะแกรมรวมของโครงงาน

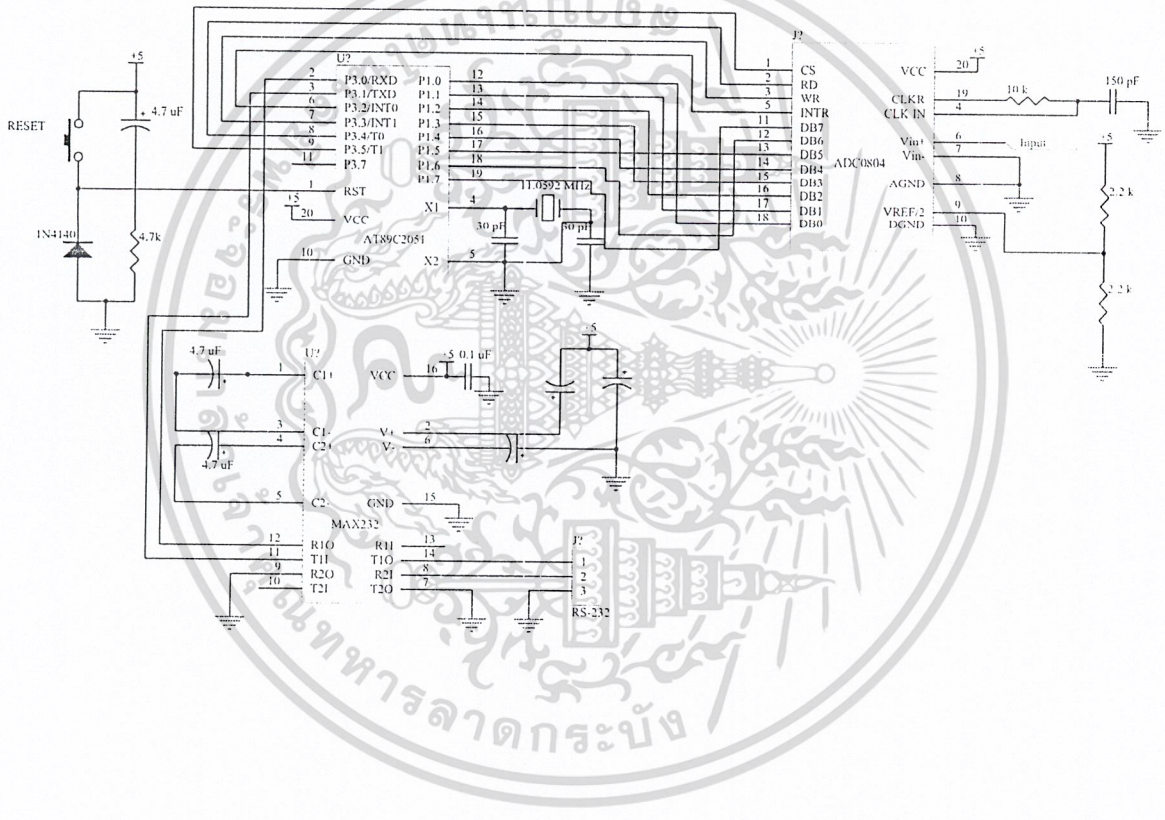


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปวงจรรวม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Title

Size B | Number | Revision

Date 19-Mar-2003 | Sheet of

ส่วนของโปรแกรม

ในที่นี้เราจะเขียนโปรแกรมด้วย MS Visual Basic

โปรแกรมฝั่งเซิร์ฟเวอร์

```
Option Explicit
Public m_frmgost As frmgost
Private Const SRCCOPY = &HCC0020 ' (DWORD) dest = source
Private Const PS_SOLID = 0
Private Const pWidth = 871 ' Width of picture box in pixels.
Private Const pHeight = 260   ' Height of picture box in pixels.
Private Const pGrid = 12.75  ' Distance between grid lines.
Dim step As Integer
Private Const pHeightHalf = pHeight \ 2
Dim counter As Long  ' Number of data points logged so far. Used to
sync grid.
Dim oldY As Long     ' Contains the previous y coordinate.
'Dim hDCh As Long, hPenB As Long, hPenC As Long
Dim SaveChart As Boolean
Public i As Integer
Public x1, y1, x2, y2, z As Single
Dim filechart As String
Dim testcounter As Integer
Public msg As String
Private Sub cmdClose_Click()
If frmgost.ComPort.PortOpen = True Then
    frmgost.ComPort.PortOpen = False
    frmChart.cmdClose.Enabled = False
    frmChart.cmdOpenport.Enabled = True
    frmChart.CmdSendData.Enabled = False
End If
End Sub
Private Sub cmdDial_Click()
        frmDial.Show
End Sub
Private Sub CmdHangup_Click()
If frmgost.ComModem.PortOpen = True Then
    frmgost.ComModem.Output = "ATH" & vbCrLf
    frmgost.ComModem.PortOpen = False
    frmgost.Connected = False
    frmChart.cmdDial.Enabled = True
    frmChart.CmdHangup = False
    Sent = False
End If
End Sub
Private Sub cmdOpenport_Click()
If frmgost.ComPort.PortOpen = False Then
    frmgost.ComPort.PortOpen = True
    frmChart.cmdClose.Enabled = True
    frmChart.cmdOpenport.Enabled = False
    If frmgost.Connected = True Then
        frmChart.CmdSendData.Enabled = True
    End If
End If
End Sub
Private Sub cmdsave_Click()
On Error Resume Next
SaveChart = True
CmdDialog.CancelError = False
'CmdDialog.flags = cdlOFNHideReadOnly
CmdDialog.Filter = "BMP Files (*.bmp)|*.bmp"
```

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น รวมทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CmdDialog.ShowSave
If Err = cdlCancel Then
    SaveChart = False
    Exit Sub
End If
    filechart = CmdDialog.FileName
End Sub
Private Sub CmdSendData_Click()
        Sent = True
End Sub
Private Sub CmdSUp_Click()
    step = step + 1
End Sub
Private Sub CmdSDown_Click()
If step = 1 Then
    Exit Sub
End If
step = step - 1
End Sub
Private Sub Form_Load()
Dim hBmp As Long
    Dim a As Integer
    step = 1
    Me.Move (VB.Screen.Width - Me.Width) / 2, (VB.Screen.Height -
Me.Height) / 2, (Me.Width), (Me.Height)
    Me.Show
    Picture1.ScaleMode = 3
    Picture1.Left = 0
    Picture1.Top = 0
    counter = 0
 'Plot horizontal grid lines.
    For a = pGrid To pHeight Step pGrid
        Picture1.Line (0, a)-(pWidth, a)
   Next
' Plot vertical grid lines.
    For a = pGrid - (counter Mod pGrid) To _
                    pWidth Step pGrid
        Picture1.Line (a, 0)-(a, pHeight)
    Next
    oldY = pHeightHalf
    Load frmgost
    frmgost.ScaleMode = 3
End Sub
Private Sub Form_Unload(Cancel As Integer)
If frmgost.ComPort.PortOpen = True Then
frmgost.ComModem.Output = "ATH0" & vbCrLf
    frmgost.ComModem.Output = "ATZ" & vbCrLf
    frmgost.ComPort.PortOpen = False
End If
End
End Sub
Public Sub ShowG(ByVal datain As Integer)
On Error Resume Next
Dim tempi As Integer
    Dim xx1, xx2 As Single
    Dim yy1, yy2 As Single
'If Sent = True Then
'    If frmgost.Connected = True Then
'        'msg = "<m><mg>" & i & "<mg><x>" & Chr$(13)
        frmgost.ComModem.Output = datain & Chr$(13)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'            'List1.AddItem msg
'
'    End If
'Else
    tempi = 255 - datain
    i = tempi
    xx1 = x1
    yy1 = y1
    xx2 = x2
    y2 = i  'Scale Y ¤èÒÂÔè§ÁÒ¡ÃÒ¿ÂÔè§ÊÙ§
    yy2 = y2
    'If Sent = True Then
    '    If frmgost.Connected = True Then
    '            frmgost.ComModem.Output = i & Chr$(13)
    '    End If
    '
     'End If

    frmChart.Picture1.Line (xx1, yy1)-(xx2, yy2), vbRed  ' Show line
    If x2 >= 871 Then
        x2 = -1
        If SaveChart = True Then
            Picture1.Picture = Picture1.Image
            If filechart <> "" Then
                SavePicture Picture1.Picture, filechart
                clearchart
            End If
            SaveChart = False
        End If
        frmChart.Picture1.Cls
        Call MakeGrid '¶éÒ¡ÃÒ¿ÇÔè§ä»¶Ö§¢Í°¢ÇÒÁ×ÍPicture1¨ÐClea
Picture1
        'clearchart
    End If
     x1 = x2
    y1 = y2
    x2 = x2 + step '¤èÒÃ¹á¹Çá¡¹ X ¤èÒÂÔè§ÁÒ¡Step ÂÔè§ËèÒ§
    'frmgost.ComPort.Output = "1"
'End If
'Text1.Text = y2
End Sub
Private Sub MakeGrid()
Dim a As Integer
clearchart
'Plot horizontal grid lines.
    For a = pGrid To pHeight Step pGrid
        Picture1.Line (0, a)-(pWidth, a)
   Next
' Plot vertical grid lines.
    For a = pGrid - (counter Mod pGrid) To _
                    pWidth Step pGrid
        Picture1.Line (a, 0)-(a, pHeight)
    Next
End Sub
Private Sub clearchart()
   Dim Xcounter As Integer
    Dim Ycounter As Integer
Dim tempH As Long
Dim tempW As Long
Dim CLSprompt As Integer
Picture1.Cls ' .Cls alone does not work for clearing picture.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    tempH = Picture1.Height 'it only undid recent changes so resized
the pic
    tempW = Picture1.Width  'and set all pixels to white then set
back to original size
    Picture1.Height = 1
    Picture1.Width = 1
    'If SaveChart = True Then
    '    Picture1.Picture = Picture1.Image
    '    Picture1.Picture = LoadPicture(App.Path & "\chart.bmp")
    'End If
    Picture1.Refresh
    Picture1.Height = tempH
    Picture1.Width = tempW
End Sub
Option Explicit
Private Sub ccmdDial_Click()
If frmgost.ComModem.PortOpen = False Then
    frmgost.ComModem.PortOpen = True
End If
frmgost.ComModem.Output = "ATD" & txtNum.Text & vbCrLf
End Sub
Private Sub cmdNum0_Click()
txtNum.Text = txtNum.Text + "0"
End Sub
Private Sub cmdNum1_Click()
txtNum.Text = txtNum.Text + "1"
End Sub
Private Sub cmdNum2_Click()
txtNum.Text = txtNum.Text + "2"
End Sub
Private Sub cmdNum3_Click()
txtNum.Text = txtNum.Text + "3"
End Sub
Private Sub cmdNum4_Click()
txtNum.Text = txtNum.Text + "4"
End Sub
Private Sub cmdNum5_Click()
txtNum.Text = txtNum.Text + "5"
End Sub
Private Sub cmdNum6_Click()
txtNum.Text = txtNum.Text + "6"
End Sub
Private Sub cmdNum7_Click()
txtNum.Text = txtNum.Text + "7"
End Sub
Private Sub cmdNum8_Click()
txtNum.Text = txtNum.Text + "8"
End Sub
Private Sub cmdNum9_Click()
txtNum.Text = txtNum.Text + "9"
End Sub
Private Sub CmdsetModem_Click()
frmOption.Show
End Sub
Private Sub cmdshap_Click()
txtNum.Text = txtNum.Text + "#"
End Sub
Private Sub cmdstart_Click()
txtNum.Text = txtNum.Text + "*"
End Sub
Private Sub Form_Load()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'frmChart.Hide
        Me.Move (VB.Screen.Width - Me.Width) / 2, (VB.Screen.Height -
Me.Height) / 2, (Me.Width), (Me.Height)
End Sub
Private Sub Form_Terminate()
       frmChart.Show
End Sub
Private Sub Form_Unload(Cancel As Integer)
     frmChart.Show
End Sub
Option Explicit
Public UseString As String
Public Connected As Boolean
Public m_frmchart As frmChart
Public Buffer As String
Private Sub commodem_OnComm()
'Occurs when there's a change in the modem signal
Dim InBuff As String
Select Case ComModem.CommEvent
    'Case comEventBreak : A Break was received from the line : not
required
    'The next three cases are timeout values, no longer detected in
Win32 systems
    'Case comEventCDTO
    'Case comEventCTSTO
    'Case comEventDSRTO
    'Case comEventFrame

    Case comEventOverrun ' Data Lost during transfer
        MsgBox "Some Data was lost during transfer."
    Case comEventRxOver  ' Receive buffer overflow.
        'taken care in another sub in form2
    Case comEventRxParity   ' Parity Error.
        MsgBox "Parity Error !"
    Case comEventTxFull  ' Transmit buffer full.
        'taken care in another module in form2
    Case comEvCD   ' Change in the CD line.
    'Used to detect connection state :
    'Though it's taken care by using ATX4 modem command, it's
required
    'for AT non compatible modems
If ComModem.CDHolding = True Then
            Connected = True
            'If frmgost.ComPort.PortOpen = True Then
                frmChart.CmdSendData.Enabled = True
            'End If
            frmChart.cmdDial.Enabled = False
            frmChart.CmdHangup.Enabled = True
            Unload frmDial
       Else
            Connected = False
            frmChart.CmdSendData.Enabled = True
            frmChart.cmdDial.Enabled = True
            frmChart.CmdHangup.Enabled = False
    End If
      Case comEvReceive 'Received RThreshold no. of chars.
         'Form2.inflow.Top = 130 'Update gui
         InBuff = StrConv(ComModem.Input, vbUnicode) 'Use strconv to
         'convert the binary data got from modem
         Call ScanCom(InBuff)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      Case comEvSend 'There are SThreshold number of characters in
the
                     'transmit buffer.(this should never occur as
there's form2, but, in case)
         Do While ComModem.OutBufferCount > 0
            DoEvents
         Loop
   End Select
End Sub
Private Sub ComPort_OnComm()
                Dim EVMsg$
                Dim ERMsg$
                Dim X As Integer 'position being read from instring
                Dim Y As String
                Dim buffer2 As String
                'Dim m_frmchart As frmChart
                ' Branch according to the CommEvent property.
                Select Case ComPort.CommEvent
                    ' Event messages.
                    Case comEvReceive
                        'Set m_frmchart = frmChart
                        Buffer = StrConv(ComPort.Input, vbUnicode)
                        If Sent = True Then
                            'Buffer = Buffer & Chr(13)
                            ComModem.Output = Buffer
                        End If
                        'm_frmchart.i = Asc(StrConv(ComPort.Input,
vbUnicode))
                        'ShowDataChart (Mid(Buffer, 1, (Len(Buffer) -
1)))
                        ShowDataChart Buffer
                    Case comEvSend
                    Case comEvCTS
                        EVMsg$ = "Change in CTS Detected"
                    Case comEvDSR
                        EVMsg$ = "Change in DSR Detected"
                    Case comEvCD
                        EVMsg$ = "Change in CD Detected"
                    Case comEvRing
                        EVMsg$ = "The Phone is Ringing"
                    Case comEvEOF
EVMsg$ = "End of File Detected"
' Error messages.
                    Case comBreak
                        ERMsg$ = "Break Received"
                    Case comCDTO
                        ERMsg$ = "Carrier Detect Timeout"
                    Case comCTSTO
                        ERMsg$ = "CTS Timeout"
                    Case comDCB
                        ERMsg$ = "Error retrieving DCB"
                    Case comDSRTO
                        ERMsg$ = "DSR Timeout"
                    Case comFrame
                        ERMsg$ = "Framing Error"
                    Case comOverrun
                        ERMsg$ = "Overrun Error"
                    Case comRxOver
                        ERMsg$ = "Receive Buffer Overflow"
                    Case comRxParity
                        ERMsg$ = "Parity Error"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    Case comTxFull
                        ERMsg$ = "Transmit Buffer Full"
                    Case Else
                        ERMsg$ = "Unknown error or event"
                End Select
End Sub
Private Sub Form_Load()
Dim regvaluellong As Long
Dim regvaluestr As String
                If ComPort.PortOpen = False Then
                          ComPort.InputLen = 0
                          ComPort.RThreshold = 250
                          ComPort.SThreshold = 0
                          ComPort.Settings = "28800,n,8,1"
                          regvaluestr = modRegistry.QueryValueStr
(REG_ROOT_PATH, "Controler Port")
                          If regvaluestr = "" Then
                            MsgBox "Please Check Your ComPort And
ModemPort", vbCritical, "Error"

                            frmOption.Show
                          Exit Sub
                          End If
                          ComPort.CommPort = regvaluestr
                End If
    If ComModem.PortOpen = False Then
                          regvaluestr = modRegistry.QueryValueStr
(REG_ROOT_PATH, "Modem Port") 'communication port
                          If regvaluestr = "" Then
                            MsgBox "Please Check Your ComPort And
ModemPort", vbCritical, "Error"
                            frmOption.Show
                            Exit Sub
                          End If
                          ComModem.CommPort = regvaluestr
                          ComModem.DTREnable = True 'Enable Data
Terminal Ready line
ComModem.Handshaking = comRTS 'type of modem handshaking (Request To
Send)
                          ComModem.InBufferSize = 2048
                          ComModem.InputLen = 0 'reads all characters
from input buffer
                          ComModem.InputMode = comInputModeBinary
'Data is sent in Binary format
                          ComModem.NullDiscard = False 'dont discard
chr$(0)
                          ComModem.OutBufferSize = 2048
                          ComModem.ParityReplace = "?" 'Replace
unrecognised chars with "?"
                          ComModem.RThreshold = 1 'no. of characters
to receive from com port at a time
                          ComModem.RTSEnable = True 'enable Request To
Send line
                          ComModem.Settings = "38400,n,8,1" 'baud
rate, parity, data bit, stop bit
                          ComModem.SThreshold = 0 'no. of characters
to send from com port at a time

                                    'RThreshold and Sthreshold must be at
least 1 so that data
                'can be received/sent
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
                '***MODEM COMMANDS***
                'These commands work if your modem accepts AT
commands (most modems do)
                ComModem.PortOpen = True
                frmgost.ComModem.Output = "ATZ" & vbCrLf
                ComModem.PortOpen = False


                ComModem.PortOpen = True
                frmgost.ComModem.Output = "ATX4" & vbCrLf
                'modem can report result codes "OK","CONNECT
#####","RING",
                '"NO CARRIER","NO ANSWER","BUSY","ERROR","NO
DIALTONE"
                '(maximum no. of reports)
                ComModem.PortOpen = False

                ComModem.PortOpen = True
'                'modem enables both V.42 bis and MNP COMPRESSION
                frmgost.ComModem.Output = "AT%C3" & vbCrLf
'                '  "AT%Cn" is used where n is variable 0-3
'                '  0 = Disable data COMPRESSION
'                '  1 = Enable MNP 5 data COMPRESSION
'                '  2 = Enable V.42 bis data COMPRESSION
'                '  3 = Enable both V.42 bis and MNP 5 data
COMPRESSION
                ComModem.PortOpen = False

                ComModem.PortOpen = True
                frmgost.ComModem.Output = "AT+MS=56,1" & vbCrLf
'                'modem enables K56flex modulation in preferance to
any other (V.90) modulation
'                'K56flex is a new modem technology that enables
connection at blistering rates
'                'upto 56 Kbps even over standard phone lines -
bridging the gap between current
'                'analog transmission rates and fully digital comm.
such as ISDN'
'                ' I've not checked form7 because there is no harm in
sending the command even
'                ' if it is not accepted : better to use K56flex if
available.
                ComModem.PortOpen = False


'
'                '*************
'                'The above three commands allow faster communication
'                'over the standard phone line
'                '*************
'
                ComModem.PortOpen = True
                frmgost.ComModem.Output = "ATL" &
modRegistry.QueryValueLong(REG_ROOT_PATH, "Modem Volume") & vbCrLf
                '  "ATLn" controls modem speaker volume where n is
variable 0-3
                '  0= Speaker off
                '  1= Speaker volume low
                '  2= Speaker volume medium
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                '  3= Speaker volume high
                ComModem.PortOpen = False

'                'modem speaker on until connect
'                '  "ATMn" where n is variable 0-3
'                '  0= Speaker always off
'                '  1= Speaker on until connect
'                '  2= Speaker is always on
'                '  3= Speaker always on only during answer and data
transmission
                ComModem.PortOpen = True
                frmgost.ComModem.Output = "ATM" &
modRegistry.QueryValueLong(REG_ROOT_PATH, "Modem Speaker") & vbCrLf
                ComModem.PortOpen = False

                If modRegistry.QueryValueLong(REG_ROOT_PATH, "Dial
Properties") = 1 Then
                'modem pulse dial
                ComModem.PortOpen = True
                frmgost.ComModem.Output = "ATP" & vbCrLf
                ComModem.PortOpen = False
                ElseIf modRegistry.QueryValueLong(REG_ROOT_PATH,
"Dial Properties") = 2 Then
                'modem tone dial
                ComModem.PortOpen = True
                frmgost.ComModem.Output = "ATT" & vbCrLf
                ComModem.PortOpen = False
                End If
     End If
                Exit Sub '
End Sub

Private Sub ScanCom(ByVal instring As String)
' This routine will work on data (instring) received from the Com
Port.
Dim X As Integer 'position being read from instring
Dim Y As String 'data in position being read (X)

If Connected = False Then
' Not connected...so data may be modem responce.
' Work on data (instring) received between vbcrlf's:
    For X = 1 To (Len(instring) + 1)
        Y = Mid$(instring, X, 1)
        If Y = Chr$(13) Then
            Y = ""
            '"MessageHandler" is where to do the processing on
            ' the packet after it is fully parsed here:
            Call MessageHandler(UseString)
            UseString = ""
        End If
        UseString = UseString & Y
    Next X
End If

End Sub
Private Sub MessageHandler(InBuff As String)

'***Rest are modem responses***

If InStr(InBuff, "CONNECT") Then
```

เอกสารนี้มีไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    'if command "ATX4" has been sent to modem, inbuff will be
"CONNECT #####" where ##### = speed
    'connected to remote comp
    Connected = True
    frmChart.CmdSendData.Enabled = True
    'Form2.t1 = Time 'Time of call
    'eg. MsgBox "Connected at" + Mid(InBuff, 8, Len(InBuff) - 7) + "
bps"
    InBuff = ""
End If

If InStr(InBuff, "OK") Then
'occurs when a command sent to modem is accepted
    InBuff = ""
End If

If InStr(InBuff, "RING") Then
'Occurs when the phone is ringing
    ComModem.Output = "ATA"
    'Accept call
    InBuff = ""
End If


If InStr(InBuff, "NO CARRIER") Then
    MsgBox "There is no carrier.  Please make sure that all your
phone cables are properly connected."
End If

If InStr(InBuff, "ERROR") Then
'also occurs when you're transferring data to yourself : without
being connected
'Sounds surprising? If youre not connected to a remote computer,
'data sent to modem bounces back to your own comp. since RThreshold=1
.
'Cool huh? :-) Thats the way i tested M-Chat .
    InBuff = ""
End If

If InStr(InBuff, "BUSY") Then
    MsgBox "The number is busy.  Please try again later."
    InBuff = ""

End If

If InStr(InBuff, "NO DIAL TONE") Then
'only if ATX4 is supported
    MsgBox "There is no dialtone.  Please check your modem
connections."
    InBuff = ""
    'If Form6.Visible = True Then Unload Form6: Dialer.Show
End If

If InStr(InBuff, "NO ANSWER") Then
'only if ATX4 is supported
    MsgBox "There was no answer from the remote modem / MultiComm is
not running at the other end. Please try again later."
    InBuff = ""
End If
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub ShowDataChart(ByVal Data As String)
Dim X As Integer 'position being read from instring
Dim Y As String 'data in position being read (X)
Dim D As Integer
 Set m_frmchart = frmChart

For X = 1 To Len(Data)
        Y = Mid$(Data, X, 1)
        D = Asc(Y)
        m_frmchart.ShowG (D)
Next X

End Sub
Option Explicit

Public chkOptionvalue As Long
Public optDialvalue As Long

Private Sub chkSpAl_Click()
If chkSpAl.Value = vbChecked Then
    chkOptionvalue = chkOptionvalue + 2
Else
    chkOptionvalue = chkOptionvalue - 2
End If
End Sub

Private Sub chkSpOn_Click()
If chkSpOn.Value = vbChecked Then
    chkOptionvalue = chkOptionvalue + 1
Else
    chkOptionvalue = chkOptionvalue - 1
End If
End Sub

Private Sub cmdOpt_Can_Click()
Dim check As Boolean
Dim temp As Long
Dim Regvalstr1 As String
Dim Regvalstr2 As String
    chkOptionvalue = 0
    check = modRegistry.QueryValueBool(REG_ROOT_PATH, "Not First")
    If check = False Then
        temp = MsgBox("Please Set your Modem and Portcontrol then
Click ok", vbCritical, "ECG")
        Exit Sub
    End If
    If (Mid(cboCport.Text, 4) = "") Or (Mid(cboMport.Text, 4) = "")
Then
        MsgBox "Please Check Your ComPort And ModemPort", vbCritical,
"Error"
        Exit Sub
    End If
    Regvalstr1 = modRegistry.QueryValueStr(REG_ROOT_PATH, "Modem
Port")
    Regvalstr2 = modRegistry.QueryValueStr(REG_ROOT_PATH, "Controler
Port")
    If Regvalstr1 = "" Or Regvalstr2 = "" Then
        MsgBox "Please Check Your ComPort And ModemPort and Click
Ok", vbCritical, "Error"
        Exit Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    End If
    Unload Me
End Sub

Private Sub cmdOpt_OK_Click()
Dim res As Long
    If (Mid(cboCport.Text, 4) = "") Or (Mid(cboMport.Text, 4) = "")
Then
        MsgBox "Please Check Your ComPort And ModemPort", vbCritical,
"Error"
        Exit Sub
    End If
    Call modRegistry.SetValueStr(REG_ROOT_PATH, "Modem Port", Mid
(cboMport.Text, 4))
    Call modRegistry.SetValueStr(REG_ROOT_PATH, "Controler Port", Mid
(cboCport.Text, 4))
    Call modRegistry.SetValueLong(REG_ROOT_PATH, "Modem Speaker",
chkOptionvalue)
    Call modRegistry.SetValueLong(REG_ROOT_PATH, "Dial Properties",
optDialvalue)
    Call modRegistry.SetValueLong(REG_ROOT_PATH, "Modem Volume",
SldVolume.Value)
    Call modRegistry.SetValueBool(REG_ROOT_PATH, "Not First", True)
    chkOptionvalue = 0
    Unload frmgost
    MsgBox "ECG reset Modem and Comport. Please Open Port again"

    'If frmgost.ComPort.PortOpen = False Then

        frmgost.ComPort.CommPort = Mid(cboCport.Text, 4)
    'End If
    'If frmgost.ComModem.PortOpen = False Then
        frmgost.ComModem.CommPort = Mid(cboMport.Text, 4)
    'End If
    Unload Me
Load frmgost

    frmChart.Show

    frmChart.cmdClose.Enabled = False
    frmChart.cmdOpenport.Enabled = True
    frmChart.CmdSendData.Enabled = False
End Sub

Private Sub Form_Load()
    Me.Move (VB.Screen.Width - Me.Width) / 2, _
                            (VB.Screen.Height - Me.Height) / 2, _
                             (Me.Width), _
                             (Me.Height)
    Call LoadPropertySettings
End Sub
Sub LoadPropertySettings()
Dim i As Integer
Dim Regvallong As Long
Dim Regvalstr As String
For i = 1 To 4
    cboMport.AddItem "Com" & (Str$(i))
Next i
For i = 1 To 4
    cboCport.AddItem "Com" & (Str$(i))
Next i
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Regvalstr = modRegistry.QueryValueStr(REG_ROOT_PATH, "Modem Port")
cboMport.Text = "Com" & Regvalstr
Regvalstr = modRegistry.QueryValueStr(REG_ROOT_PATH, "Controler
Port")
cboCport.Text = "Com" & Regvalstr
 Regvallong = modRegistry.QueryValueLong(REG_ROOT_PATH, "Modem
Speaker")
Select Case Regvallong
        Case 0
                chkSpAl.Value = vbUnchecked
                chkSpOn.Value = vbUnchecked
        Case 1
                chkSpAl.Value = vbUnchecked
                chkSpOn.Value = vbChecked
        Case 2
                chkSpAl.Value = vbChecked
                chkSpOn.Value = vbUnchecked
        Case 3
                chkSpAl.Value = vbChecked
                chkSpOn.Value = vbChecked
        Case Else
End Select
Regvallong = modRegistry.QueryValueLong(REG_ROOT_PATH, "Dial
Properties")
If Regvallong = 1 Then
    optPulse = True
    optDialvalue = 1
Else
    optTone = True
    optDialvalue = 2
End If
Regvallong = modRegistry.QueryValueLong(REG_ROOT_PATH, "Modem
volume")
SldVolume.Value = Regvallong
End Sub
Private Sub optPulse_Click()
    optDialvalue = 1
End Sub

Private Sub optTone_Click()
    optDialvalue = 2
End Sub
Option Explicit
Public Sent As Boolean

Sub main()
Dim check As Boolean

check = modRegistry.QueryValueBool(REG_ROOT_PATH, "Not First", False)
If check = False Then
    frmOption.Show
    Exit Sub
End If
    frmChart.Show
    'Load frmgost
End Sub
Option Explicit

Const MODULE_NAME = "modRegistry"
Global Const REG_ROOT_PATH = "HKLM\SOFTWARE\ProjectECG"
Global gError As Boolean
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Function QueryValueStr(RegPath As String, ValueName As String,
Optional DefaultValue As String) As String
   On Error GoTo ErrResume
   Dim reg As CRegistry
   Dim r As Long

   Set reg = New CRegistry
   r = reg.CreateKey(RegPath)
   If r Then
      QueryValueStr = reg.QueryValue(ValueName, DefaultValue)
   Else
      QueryValueStr = DefaultValue
   End If

   Exit Function
ErrResume:
   'Call LogError(MODULE_NAME, "QueryValueStr")
   Resume Next
End Function

Function QueryValueLong(RegPath As String, ValueName As String,
Optional ByVal DefaultValue As Long) As Long
   On Error GoTo ErrResume
   Dim reg As CRegistry
   Dim r As Long

   Set reg = New CRegistry
   r = reg.CreateKey(RegPath)
   If r Then
      QueryValueLong = reg.QueryValueLong(ValueName, DefaultValue)
   Else
      QueryValueLong = DefaultValue
   End If
Exit Function
ErrResume:
   'Call LogError(MODULE_NAME, "QueryValueLong")
   Resume Next
End Function
Function QueryValueBool(RegPath As String, ValueName As String,
Optional ByVal DefaultValue As Boolean) As Boolean
   On Error GoTo ErrResume
   Dim reg As CRegistry
   Dim s As String
   Dim r As Long
   Set reg = New CRegistry
   r = reg.CreateKey(RegPath)
   If r Then
      If DefaultValue Then
         s = LCase$(reg.QueryValue(ValueName, "yes"))
      Else
         s = LCase$(reg.QueryValue(ValueName, "no"))
      End If
      If s = "yes" Then
         QueryValueBool = True
      ElseIf s = "no" Then
         QueryValueBool = False
      Else
         QueryValueBool = DefaultValue
   End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Else
       QueryValueBool = DefaultValue
    End If
  Exit Function
 ErrResume:
    'Call LogError(MODULE_NAME, "QueryValueBool")
    Resume Next
 End Function
 Function SetValueStr(RegPath As String, ValueName As String, Value As
 String) As Boolean
    On Error GoTo ErrResume
    Dim reg As CRegistry
    Dim r As Long
 Set reg = New CRegistry
    r = reg.CreateKey(RegPath)
    If r Then
       SetValueStr = reg.SetValue(ValueName, Value)
    End If
 Exit Function
 ErrResume:
    'Call LogError(MODULE_NAME, "SetValueStr")
    Resume Next
 End Function

 Function SetValueLong(RegPath As String, ValueName As String, ByVal
 Value As Long) As Boolean
    On Error GoTo ErrResume
    Dim reg As CRegistry
    Dim r As Long

    Set reg = New CRegistry
    r = reg.CreateKey(RegPath)
    If r Then
       SetValueLong = reg.SetValueLong(ValueName, Value)
    End If
    Exit Function
 ErrResume:
    'Call LogError(MODULE_NAME, "SetValueLong")
    Resume Next
 End Function

 Function SetValueBool(RegPath As String, ValueName As String, ByVal
 Value As Boolean) As Boolean
    On Error GoTo ErrResume
    Dim reg As CRegistry
    Dim r As Long

    Set reg = New CRegistry
    r = reg.CreateKey(RegPath)
    If r Then
       If Value Then
          SetValueBool = reg.SetValue(ValueName, "yes")
       Else
          SetValueBool = reg.SetValue(ValueName, "no")
       End If
 End If
  Exit Function
 ErrResume:
    'Call LogError(MODULE_NAME, "SetValueBool")
 Resume Next
 End Function
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมฝั่งไคลแอนท์

```
Option Explicit
Public m_frmgost As frmgost
Private Const SRCCOPY = &HCC0020 ' (DWORD) dest = source
Private Const PS_SOLID = 0
Private Const pWidth = 871     ' Width of picture box in pixels.
Private Const pHeight = 260    ' Height of picture box in pixels.
Private Const pGrid = 12.75      ' Distance between grid lines.
Private Const pHeightHalf = pHeight \ 2
Dim counter As Long  ' Number of data points logged so far. Used to
                     ' sync grid.
Dim oldY As Long     ' Contains the previous y coordinate.
Dim hDCh As Long, hPenB As Long, hPenC As Long
Dim Sent, Savechart As Boolean
Public i As Integer
Dim step As Integer
Public x1, y1, X2, Y2 As Single
Dim filechart As String
Private Sub cmdClose_Click()
If frmgost.ComPort.PortOpen = True Then
    frmgost.ComPort.PortOpen = False
    'frmChart.cmdClose.Enabled = False
    'frmChart.cmdOpenport.Enabled = True
    'frmChart.CmdSendData.Enabled = False
End If
End Sub
Private Sub cmdDial_Click()
        frmDial.Show
End Sub
Private Sub CmdHangup_Click()
If frmgost.ComModem.PortOpen = True Then
    'frmgost.ComModem.Output = "ATH" & vbCrLf
    frmgost.ComModem.PortOpen = False
    frmgost.Connected = False
    Me.cmdlisten.Enabled = True
    frmChart.cmdlisten.Enabled = True
    frmChart.CmdHangup.Enabled = False
End If
End Sub
Private Sub cmdOpenport_Click()
'MsgBox "Open Port"
If frmgost.ComPort.PortOpen = False Then
    frmgost.ComPort.PortOpen = True
    'frmChart.cmdClose.Enabled = True
    'frmChart.cmdOpenport.Enabled = False
    'If frmgost.Connected = True Then
        'frmChart.CmdSendData.Enabled = True
    'End If
End If
End Sub
Private Sub cmdsave_Click()

On Error Resume Next
Savechart = True
CmdDialog.CancelError = False
'CmdDialog.flags = cdlOFNHideReadOnly
CmdDialog.Filter = "BMP Files (*.bmp)|*.bmp"
```

เอกสารนี้จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CmdDialog.ShowSave
If Err = cdlCancel Then
    Savechart = False
    Exit Sub
End If
    filechart = CmdDialog.FileName
End Sub
Private Sub cmdSdown_Click()
If step = 1 Then
    Exit Sub
End If
step = step - 1
End Sub
Private Sub CmdSendData_Click()
        Sent = True
End Sub
Private Sub cmdlisten_Click()
If frmgost.ComModem.PortOpen = False Then
    frmgost.ComModem.PortOpen = True
End If
frmgost.ComModem.DTREnable = True 'Enables waiting for call
frmgost.ComModem.Output = "ATE1S0=1" & vbCrLf 'Wait for call
frmChart.CmdHangup.Enabled = True
frmChart.cmdlisten.Enabled = False
End Sub

Private Sub cmdSetModem_Click()
    frmOption.Show
End Sub
Private Sub cmdSup_Click()
step = step + 1
End Sub
Private Sub Form_Load()
Dim hBmp As Long
    Dim a As Integer
    Me.Move (VB.Screen.Width - Me.Width) / 2, (VB.Screen.Height -
Me.Height) / 2, (Me.Width), (Me.Height)
    Me.Show
    Picture1.ScaleMode = 3
    Picture1.Left = 0
    Picture1.Top = 0
    frmgost.ScaleMode = 3
    counter = 0
 'Plot horizontal grid lines.
    For a = pGrid To pHeight Step pGrid
        Picture1.Line (0, a)-(pWidth, a)
    Next
' Plot vertical grid lines.
    For a = pGrid - (counter Mod pGrid) To _
                    pWidth Step pGrid
        Picture1.Line (a, 0)-(a, pHeight)
    Next
    oldY = pHeightHalf
    Load frmgost
    If frmgost.ComModem.PortOpen = False Then
        frmgost.ComModem.PortOpen = True
    End If
    frmgost.ComModem.DTREnable = True 'Enables waiting for call
    frmgost.ComModem.Output = "ATE1S0=1" & vbCrLf 'Wait for call
    step = 1
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Form_Unload(Cancel As Integer)
If frmgost.ComPort.PortOpen = True Then
    frmgost.ComModem.Output = "ATH0" & vbCrLf
    frmgost.ComModem.Output = "ATZ" & vbCrLf
frmgost.ComPort.PortOpen = False
End If
End
End Sub
Public Sub ShowG(ByVal datain As Integer)
'Dim tempi As Integer
On Error Resume Next
'Dim tempi As Integer
    Dim xx1, xx2 As Single
    Dim yy1, yy2 As Single
xx1 = x1
 yy1 = y1
 xx2 = X2
 Y2 = 255 - datain 'Scale Y
 yy2 = Y2
frmChart.Picture1.Line (xx1, yy1)-(xx2, yy2), vbRed  ' Show line
If X2 >= 871 Then 'frmChart.Picture1.Width Then
    X2 = -1
    If Savechart = True Then
        Picture1.Picture = Picture1.Image
        If filechart <> "" Then
            SavePicture Picture1.Picture, filechart
        End If
        Savechart = False
    End If
    frmChart.Picture1.Cls
    Call MakeGrid
End If
'List2.AddItem Str(xx1) & "," & Str(xx2) & "," & Str(Y2) & "," & Str
(yy2)
 x1 = X2
y1 = Y2
X2 = X2 + step
End Sub
Private Sub MakeGrid()
Dim a As Integer
clearchart
  'Plot horizontal grid lines.
    For a = pGrid To pHeight Step pGrid
        Picture1.Line (0, a)-(pWidth, a)
    Next
' Plot vertical grid lines.
    For a = pGrid - (counter Mod pGrid) To _
                    pWidth Step pGrid
        Picture1.Line (a, 0)-(a, pHeight)
    Next
End Sub
Private Sub clearchart()
   Dim Xcounter As Integer
    Dim Ycounter As Integer
Dim tempH As Long
Dim tempW As Long
Dim CLSprompt As Integer
    Picture1.Cls ' .Cls alone does not work for clearing picture.
    tempH = Picture1.Height 'it only undid recent changes so resized
the pic
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    tempW = Picture1.Width  'and set all pixels to white then set
back to original size
    Picture1.Height = 1
    Picture1.Width = 1
    Picture1.Picture = Picture1.Image
    Picture1.Refresh
    Picture1.Height = tempH
    Picture1.Width = tempW
End Sub
Option Explicit
Private Sub ccmdDial_Click()
If frmgost.ComModem.PortOpen = False Then
    frmgost.ComModem.PortOpen = True
End If
frmgost.ComModem.Output = "ATD" & txtNum.Text & vbCrLf
End Sub

Private Sub cmdNum0_Click()
txtNum.Text = txtNum.Text + "0"
End Sub

Private Sub cmdNum1_Click()
txtNum.Text = txtNum.Text + "1"
End Sub

Private Sub cmdNum2_Click()
txtNum.Text = txtNum.Text + "2"
End Sub
Private Sub cmdNum3_Click()
txtNum.Text = txtNum.Text + "3"
End Sub
Private Sub cmdNum4_Click()
txtNum.Text = txtNum.Text + "4"
End Sub
Private Sub cmdNum5_Click()
txtNum.Text = txtNum.Text + "5"
End Sub
Private Sub cmdNum6_Click()
txtNum.Text = txtNum.Text + "6"
End Sub
Private Sub cmdNum7_Click()
txtNum.Text = txtNum.Text + "7"
End Sub
Private Sub cmdNum8_Click()
txtNum.Text = txtNum.Text + "8"
End Sub
Private Sub cmdNum9_Click()
txtNum.Text = txtNum.Text + "9"
End Sub
Private Sub cmdSetModem_Click()
frmOption.Show
End Sub
Private Sub cmdshap_Click()
txtNum.Text = txtNum.Text + "#"
End Sub
Private Sub cmdstart_Click()
txtNum.Text = txtNum.Text + "*"
End Sub
Private Sub Form_Load()
'frmChart.Hide
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Me.Move (VB.Screen.Width - Me.Width) / 2, (VB.Screen.Height -
Me.Height) / 2, (Me.Width), (Me.Height)
End Sub
Private Sub Form_Terminate()
       frmChart.Show
End Sub
Private Sub Form_Unload(Cancel As Integer)
     frmChart.Show
End Sub
Option Explicit
Public UseString As String
Public Connected As Boolean
Public m_frmchart As frmChart
Private Sub commodem_OnComm()
'Occurs when there's a change in the modem signal
Dim InBuff As String
Select Case ComModem.CommEvent
    'Case comEventBreak : A Break was received from the line : not
required
    'The next three cases are timeout values, no longer detected in
Win32 systems
    'Case comEventCDTO
    'Case comEventCTSTO
    'Case comEventDSRTO
    'Case comEventFrame

    Case comEventOverrun ' Data Lost during transfer
        MsgBox "Some Data was lost during transfer."
    Case comEventRxOver  ' Receive buffer overflow.
        'taken care in another sub in form2
    Case comEventRxParity   ' Parity Error.
        MsgBox "Parity Error !"
    Case comEventTxFull  ' Transmit buffer full.
        'taken care in another module in form2
    Case comEvCD    ' Change in the CD line.
    'Used to detect connection state :
    'Though it's taken care by using ATX4 modem command, it's
required
    'for AT non compatible modems
        If ComModem.CDHolding = True Then

            Connected = True
            frmChart.CmdHangup.Enabled = True
            frmChart.cmdlisten.Enabled = False
       Else
            Connected = False
            frmChart.CmdHangup.Enabled = False
            frmChart.cmdlisten.Enabled = True

      Case comEvReceive 'Received RThreshold no. of chars.
         InBuff = StrConv(ComModem.Input, vbUnicode) 'Use strconv to

         'convert the binary data got from modem
         'datafrm.BytesReceived = datafrm.BytesReceived + Len(InBuff)
         Call ScanCom(InBuff)
      Case comEvSend 'There are SThreshold number of characters in
the
                     'transmit buffer.(this should never occur as
there's form2, but, in case)
          Do While ComModem.OutBufferCount > 0
             DoEvents Loop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Select
End Sub
Private Sub Form_Load()
Dim regvaluellong As Long
Dim regvaluestr As String

    '              regvaluestr = modRegistry.QueryValueStr
(REG_ROOT_PATH, "Controler Port")
     '             ComPort.CommPort = regvaluestr
                  regvaluestr = modRegistry.QueryValueStr
(REG_ROOT_PATH, "Modem Port") 'communication port
                          If regvaluestr = "" Then
                             MsgBox "Please Check Your ModemPort",
vbCritical, "Error"
                             frmOption.Show
                             Exit Sub
                           End If
                  ComModem.CommPort = regvaluestr
                  ComModem.DTREnable = True 'Enable Data Terminal Ready
line
                  ComModem.Handshaking = comRTS 'type of modem
handshaking (Request To Send)
                  ComModem.InBufferSize = 2048
                  ComModem.InputLen = 0 'reads all characters from
input buffer
                  ComModem.InputMode = comInputModeBinary 'Data is sent
in Binary format
                  ComModem.NullDiscard = False 'dont discard chr$(0)
                  ComModem.OutBufferSize = 2048
                  ComModem.ParityReplace = "?" 'Replace unrecognised
chars with "?"
                  ComModem.RThreshold = 350 'no. of characters to
receive from com port at a time
                  ComModem.RTSEnable = True 'enable Request To Send
line
                  ComModem.Settings = "38400,n,8,1" 'baud rate, parity,
data bit, stop bit
                  ComModem.SThreshold = 0 'no. of characters to send
from com port at a time
                  'RThreshold and Sthreshold must be at least 1 so that
data
                  'can be received/sent

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
                  '***MODEM COMMANDS***
                  'These commands work if your modem accepts AT
commands (most modems do)

                  ComModem.PortOpen = True
                  frmgost.ComModem.Output = "ATX4" & vbCrLf
                  'modem can report result codes "OK","CONNECT
#####","RING",
                  '"NO CARRIER","NO ANSWER","BUSY","ERROR","NO
DIALTONE"
                  '(maximum no. of reports)
                  ComModem.PortOpen = False
'
'                 If Form7.Check4.Value = 1 Then
                  ComModem.PortOpen = True
'                 'modem enables both V.42 bis and MNP COMPRESSION
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ทำการแก้ไขดัดแปลงเนื้อหาและห้ามนำไปใช้

```
                    frmgost.ComModem.Output = "AT%C3" & vbCrLf
'                    '   "AT%Cn" is used where n is variable 0-3
'                    '   0 = Disable data COMPRESSION
'                    '   1 = Enable MNP 5 data COMPRESSION
'                    '   2 = Enable V.42 bis data COMPRESSION
'                    '   3 = Enable both V.42 bis and MNP 5 data
COMPRESSION
                    ComModem.PortOpen = False
'                    End If
'
                    ComModem.PortOpen = True
                    frmgost.ComModem.Output = "AT+MS=56,1" & vbCrLf
'                    'modem enables K56flex modulation in preferance to
any other (V.90) modulation
'                    'K56flex is a new modem technology that enables
connection at blistering rates
'                    'upto 56 Kbps even over standard phone lines -
bridging the gap between current
'                    'analog transmission rates and fully digital comm.
such as ISDN
'
'                    ' I've not checked form7 because there is no harm in
sending the command even
'                    ' if it is not accepted : better to use K56flex if
available.
                    ComModem.PortOpen = False

'
'                    '*************
'                    'The above three commands allow faster communication
'                    'over the standard phone line
'                    '*************
'
                    ComModem.PortOpen = True
                    frmgost.ComModem.Output = "ATL" &
modRegistry.QueryValueLong(REG_ROOT_PATH, "Modem Volume") & vbCrLf
                    '   "ATLn" controls modem speaker volume where n is
variable 0-3
                    '   0= Speaker off
                    '   1= Speaker volume low
                    '   2= Speaker volume medium
                    '   3= Speaker volume high
                   ComModem.PortOpen = False
'                    If Form7.Check1.Value = 1 Then
'                    'modem speaker on until connect
'                    '   "ATMn" where n is variable 0-3
'                    '   0= Speaker always off
'                    '   1= Speaker on until connect
'                    '   2= Speaker is always on
'                    '   3= Speaker always on only during answer and data
transmission
                   ComModem.PortOpen = True
                   frmgost.ComModem.Output = "ATM" &
modRegistry.QueryValueLong(REG_ROOT_PATH, "Modem Speaker") & vbCrLf
                   ComModem.PortOpen = False
'                    End If
'
'                    If Form7.Check2.Value = 1 Then
'                    'modem speaker is always on
                    ComModem.PortOpen = True
                    Form1.ComModem.Output = "ATM2" & vbCrLf
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'                   ComModem.PortOpen = False
'                   End If
'
                    If modRegistry.QueryValueLong(REG_ROOT_PATH, "Dial
Properties") = 1 Then
                    'modem pulse dial
                    ComModem.PortOpen = True
                    frmgost.ComModem.Output = "ATP" & vbCrLf
                    ComModem.PortOpen = False
                    ElseIf modRegistry.QueryValueLong(REG_ROOT_PATH,
"Dial Properties") = 2 Then
                    'modem tone dial
                    ComModem.PortOpen = True
                    frmgost.ComModem.Output = "ATT" & vbCrLf
                    ComModem.PortOpen = False
                    End If
                    Exit Sub '
End Sub

Private Sub ScanCom(ByVal instring As String)
' This routine will work on data (instring) received from the Com
Port.
Dim X As Integer 'position being read from instring
Dim Y As String 'data in position being read (X)
' Not connected...so data may be modem responce.
' Work on data (instring) received between vbcrlf's:
Dim X2 As Integer 'position being read from instring
Dim Y2 As String 'data in position being read (X)
Dim D As Integer
If Connected = False Then
     For X = 1 To (Len(instring) + 1)
          Y = Mid$(instring, X, 1)
          If Y = Chr$(13) Then
               Y = ""
               'frmChart.List1.AddItem UseString
               'If Connected = True Then
               ' For X2 = 1 To Len(UseString)
               '     frmChart.ShowG Asc(Mid(UseString, X2, 1))
               ' Next X2
                'frmChart.ShowG Val(UseString)
               'Else
                Call MessageHandler(UseString)
               'End If

               '"MessageHandler" is where to do the processing on
               ' the packet after it is fully parsed here:

              UseString = ""
          End If
          UseString = UseString & Y
       Next X
Else
    For X2 = 1 To Len(instring)
        frmChart.ShowG Asc(Mid(instring, X2, 1))
    Next X2

End If
End Sub
Private Sub MessageHandler(InBuff As String)
Dim temp As String
'***Rest are modem responses***
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If InStr(InBuff, "CONNECT") Then
    'if command "ATX4" has been sent to modem, inbuff will be
"CONNECT #####" where ##### = speed
    'connected to remote comp
    Connected = True
    'Form2.t1 = Time 'Time of call
    'eg. MsgBox "Connected at" + Mid(InBuff, 8, Len(InBuff) - 7) + "
bps"
    InBuff = ""
    'Form2.Show 'control panel
    'If Form6.Visible = True Then Form2.Timer1.Enabled = True:
Form6.Hide 'start timer for connected time
   ' If Form4.Visible = True Then Form4.Timer2.Enabled = False:
Form4.Hide 'stop animation
End If

If InStr(InBuff, "OK") Then
'occurs when a command sent to modem is accepted
    InBuff = ""
End If

If InStr(InBuff, "RING") Then
'Occurs when the phone is ringing

    'ComModem.Output = "ATA"
    'Accept call
    InBuff = ""
End If
If InStr(InBuff, "NO CARRIER") Then
    MsgBox "There is no carrier.  Please make sure that all your
phone cables are properly connected."
End If

If InStr(InBuff, "ERROR") Then
'also occurs when you're transferring data to yourself : without
being connected
'Sounds surprising? If youre not connected to a remote computer,
'data sent to modem bounces back to your own comp. since RThreshold=1
.
'Cool huh? :-) Thats the way i tested M-Chat .
    InBuff = ""
End If

If InStr(InBuff, "BUSY") Then
    MsgBox "The number is busy.  Please try again later."
    InBuff = ""
   ' If Form6.Visible = True Then Unload Form6: Dialer.Show
End If

If InStr(InBuff, "NO DIAL TONE") Then
'only if ATX4 is supported
    MsgBox "There is no dialtone.  Please check your modem
connections."
    InBuff = ""
    'If Form6.Visible = True Then Unload Form6: Dialer.Show
End If

If InStr(InBuff, "NO ANSWER") Then
'only if ATX4 is supported
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MsgBox "There was no answer from the remote modem / MultiComm is not
running at the other end. Please try again later."
    InBuff = ""
End If

End Sub
Option Explicit

Public chkOptionvalue As Long
Public optDialvalue As Long

Private Sub chkSpAl_Click()
If chkSpAl.Value = vbChecked Then
    chkOptionvalue = chkOptionvalue + 2
Else
    chkOptionvalue = chkOptionvalue - 2
End If
End Sub

Private Sub chkSpOn_Click()
If chkSpOn.Value = vbChecked Then
    chkOptionvalue = chkOptionvalue + 1
Else
    chkOptionvalue = chkOptionvalue - 1
End If
End Sub

Private Sub cmdOpt_Can_Click()
Dim check As Boolean
Dim temp As Long
Dim Regvalstr1 As String
    chkOptionvalue = 0
    check = modRegistry.QueryValueBool(REG_ROOT_PATH, "Not First")
    If check = False Then
        temp = MsgBox("Please Set your Modem then Click ok",
vbCritical, "ECG")
        Exit Sub
    End If
    If (Mid(cboMport.Text, 4) = "") Then
        MsgBox "Please Check Your ModemPort", vbCritical, "Error"
        Exit Sub
    End If
    Regvalstr1 = modRegistry.QueryValueStr(REG_ROOT_PATH, "Modem
Port")
    If Regvalstr1 = "" Then
        MsgBox "Please Set your Modem then Click ok", vbCritical,
"Error"
        Exit Sub
    End If
    Unload Me
End Sub

Private Sub cmdOpt_OK_Click()
Dim res As Long
    If Mid(cboMport.Text, 4) = "" Then
        MsgBox "Please Check Your ComPort And ModemPort", vbCritical,
"Error"
        Exit Sub
    End If
    Call modRegistry.SetValueStr(REG_ROOT_PATH, "Modem Port", Mid
(cboMport.Text, 4))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Call modRegistry.SetValueLong(REG_ROOT_PATH, "Modem Speaker",
chkOptionvalue)
    Call modRegistry.SetValueLong(REG_ROOT_PATH, "Dial Properties",
optDialvalue)
    Call modRegistry.SetValueLong(REG_ROOT_PATH, "Modem Volume",
SldVolume.Value)
    Call modRegistry.SetValueBool(REG_ROOT_PATH, "Not First", True)
    Unload frmgost
        MsgBox "ECG reset Modem and Comport. Please Open Port again"


        frmgost.ComModem.CommPort = Mid(cboMport.Text, 4)
    Unload Me
    Load frmgost
    frmChart.Show
    frmChart.cmdlisten.Enabled = True
    frmChart.CmdHangup.Enabled = False

End Sub

Private Sub Form_Load()
    Me.Move (VB.Screen.Width - Me.Width) / 2, _
                        (VB.Screen.Height - Me.Height) / 2, _
                        (Me.Width), _
                        (Me.Height)

    Call LoadPropertySettings
End Sub
Sub LoadPropertySettings()
Dim i As Integer
Dim Regvallong As Long
Dim Regvalstr As String
For i = 1 To 4
    cboMport.AddItem "Com" & (Str$(i))
Next i
For i = 1 To 4
    cboCport.AddItem "Com" & (Str$(i))
Next i
Regvalstr = modRegistry.QueryValueStr(REG_ROOT_PATH, "Modem Port")
cboMport.Text = "Com" & Regvalstr
Regvalstr = modRegistry.QueryValueStr(REG_ROOT_PATH, "Controler
Port")
cboCport.Text = "Com" & Regvalstr
 Regvallong = modRegistry.QueryValueLong(REG_ROOT_PATH, "Modem
Speaker")
Select Case Regvallong
        Case 0
                chkSpAl.Value = vbUnchecked
                chkSpOn.Value = vbUnchecked
        Case 1
                chkSpAl.Value = vbUnchecked
                chkSpOn.Value = vbChecked
        Case 2
                chkSpAl.Value = vbChecked
                chkSpOn.Value = vbUnchecked
        Case 3
                chkSpAl.Value = vbChecked
                chkSpOn.Value = vbChecked
        Case Else
End Select
Regvallong = modRegistry.QueryValueLong(REG_ROOT_PATH, "Dial
Properties")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If Regvallong = 1 Then
    optPulse = True
    optDialvalue = 1
Else
    optTone = True
    optDialvalue = 2
End If
Regvallong = modRegistry.QueryValueLong(REG_ROOT_PATH, "Modem
volume")
SldVolume.Value = Regvallong
End Sub



Private Sub optPulse_Click()
    optDialvalue = 1
End Sub

Private Sub optTone_Click()
    optDialvalue = 2
End Sub
Option Explicit

Sub main()
Dim check As Boolean

check = modRegistry.QueryValueBool(REG_ROOT_PATH, "Not First", False)
If check = False Then
    frmOption.Show
    Exit Sub
End If
    frmChart.Show
    'Load frmgost
End Sub
Option Explicit

Const MODULE_NAME = "modRegistry"
Global Const REG_ROOT_PATH = "HKLM\SOFTWARE\ProjectECG"
Global gError As Boolean

Function QueryValueStr(RegPath As String, ValueName As String,
Optional DefaultValue As String) As String
   On Error GoTo ErrResume
   Dim reg As CRegistry
   Dim r As Long

   Set reg = New CRegistry
   r = reg.CreateKey(RegPath)
   If r Then
      QueryValueStr = reg.QueryValue(ValueName, DefaultValue)
   Else
      QueryValueStr = DefaultValue
   End If

   Exit Function
ErrResume:
   'Call LogError(MODULE_NAME, "QueryValueStr")
   Resume Next
End Function
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

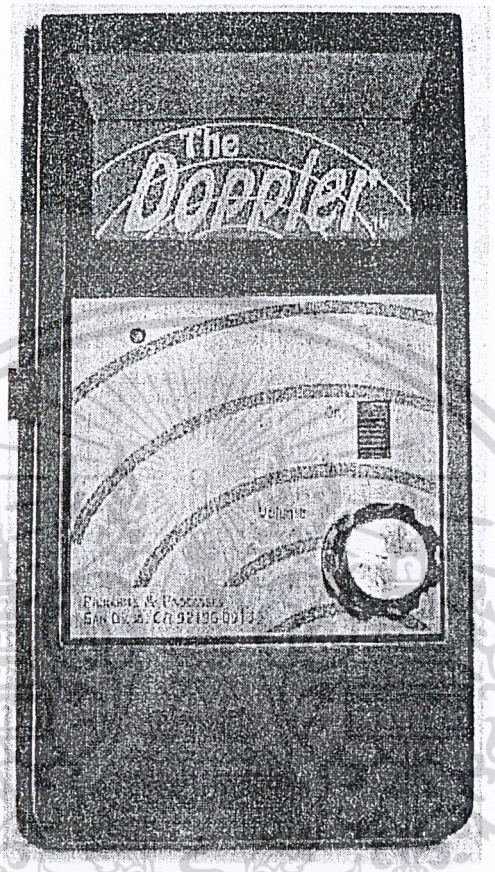
```
Function QueryValueLong(RegPath As String, ValueName As String,
Optional ByVal DefaultValue As Long) As Long
   On Error GoTo ErrResume
   Dim reg As CRegistry
   Dim r As Long

   Set reg = New CRegistry
   r = reg.CreateKey(RegPath)
   If r Then
      QueryValueLong = reg.QueryValueLong(ValueName, DefaultValue)
   Else
      QueryValueLong = DefaultValue
   End If

   Exit Function
ErrResume:
   'Call LogError(MODULE_NAME, "QueryValueLong")
   Resume Next
End Function

Function QueryValueBool(RegPath As String, ValueName As String,
Optional ByVal DefaultValue As Boolean) As Boolean
   On Error GoTo ErrResume
   Dim reg As CRegistry
   Dim s As String
   Dim r As Long

   Set reg = New CRegistry
   r = reg.CreateKey(RegPath)
   If r Then
      If DefaultValue Then
         s = LCase$(reg.QueryValue(ValueName, "yes"))
      Else
         s = LCase$(reg.QueryValue(ValueName, "no"))
      End If
      If s = "yes" Then
         QueryValueBool = True
      ElseIf s = "no" Then
         QueryValueBool = False
      Else
         QueryValueBool = DefaultValue
      End If
   Else
      QueryValueBool = DefaultValue
   End If

   Exit Function
ErrResume:
   'Call LogError(MODULE_NAME, "QueryValueBool")
   Resume Next
End Function

Function SetValueStr(RegPath As String, ValueName As String, Value As
String) As Boolean
   On Error GoTo ErrResume
   Dim reg As CRegistry
   Dim r As Long

   Set reg = New CRegistry
   r = reg.CreateKey(RegPath)
   If r Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SetValueStr = reg.SetValue(ValueName, Value)
    End If

    Exit Function
ErrResume:
    'Call LogError(MODULE_NAME, "SetValueStr")
    Resume Next
End Function

Function SetValueLong(RegPath As String, ValueName As String, ByVal
Value As Long) As Boolean
    On Error GoTo ErrResume
    Dim reg As CRegistry
    Dim r As Long

    Set reg = New CRegistry
    r = reg.CreateKey(RegPath)
    If r Then
        SetValueLong = reg.SetValueLong(ValueName, Value)
    End If

    Exit Function
ErrResume:
    'Call LogError(MODULE_NAME, "SetValueLong")
    Resume Next
End Function

Function SetValueBool(RegPath As String, ValueName As String, ByVal
Value As Boolean) As Boolean
    On Error GoTo ErrResume
    Dim reg As CRegistry
    Dim r As Long

    Set reg = New CRegistry
    r = reg.CreateKey(RegPath)
    If r Then
        If Value Then
            SetValueBool = reg.SetValue(ValueName, "yes")
        Else
            SetValueBool = reg.SetValue(ValueName, "no")
        End If
    End If

    Exit Function
ErrResume:
    'Call LogError(MODULE_NAME, "SetValueBool")
    Resume Next
End Function
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่อง The Doppler



สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Order this document by TL062/D

MOTOROLA

TL062
TL064

Low Power JFET Input Operational Amplifiers

These JFET input operational amplifiers are designed for low power applications. They feature high input impedance, low input bias current and low input offset current. Advanced design techniques allow for higher slew rates, gain bandwidth products and output swing.

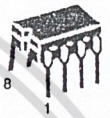
The commercial and vehicular devices are available in Plastic dual in–line and SOIC packages.

- Low Supply Current: 200 µA/Amplifier
- Low Input Bias Current: 5.0 pA
- High Gain Bandwidth: 2.0 MHz
- High Slew Rate: 6.0 V/µs
- High Input Impedance: 10^{12} Ω
- Large Output Voltage Swing: ±14 V
- Output Short Circuit Protection

LOW POWER JFET INPUT OPERATIONAL AMPLIFIERS

SEMICONDUCTOR TECHNICAL DATA

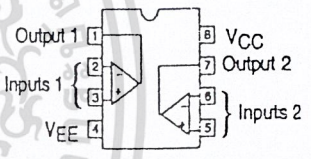
DUAL



P SUFFIX
PLASTIC PACKAGE
CASE 626

D SUFFIX
PLASTIC PACKAGE
CASE 751
(SO–8)

PIN CONNECTIONS



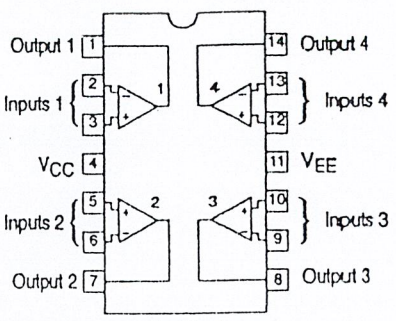
(Top View)

QUAD

N SUFFIX
PLASTIC PACKAGE
CASE 646

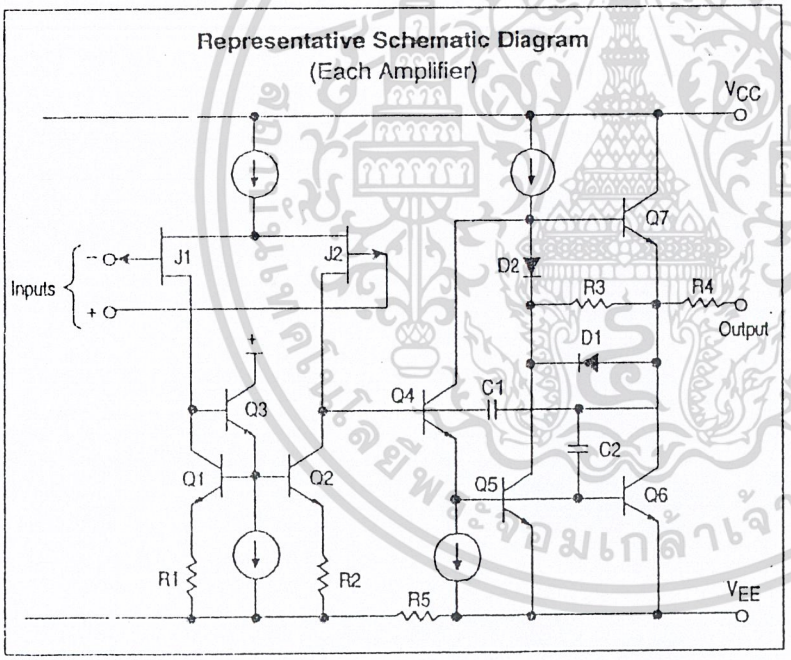
D SUFFIX
PLASTIC PACKAGE
CASE 751A
(SO–14)

PIN CONNECTIONS



(Top View)

Representative Schematic Diagram
(Each Amplifier)



ORDERING INFORMATION

Op Amp Function	Device	Operating Temperature Range	Package
Dual	TL062CD, ACD TL062CP, ACP	T_A = 0° to +70°C	SO–8 Plastic DIP
	TL062VD TL062VP	T_A = –40° to +85°C	SO–8 Plastic DIP
Quad	TL064CD, ACD TL064CN, ACN	T_A = 0° to +70°C	SO–14 Plastic DIP
	TL064VD TL064VN	T_A = –40° to +85°C	SO–14 Plastic DIP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

© Motorola, Inc. 1996

Figure 7. Open Loop Voltage Gain and Phase versus Frequency

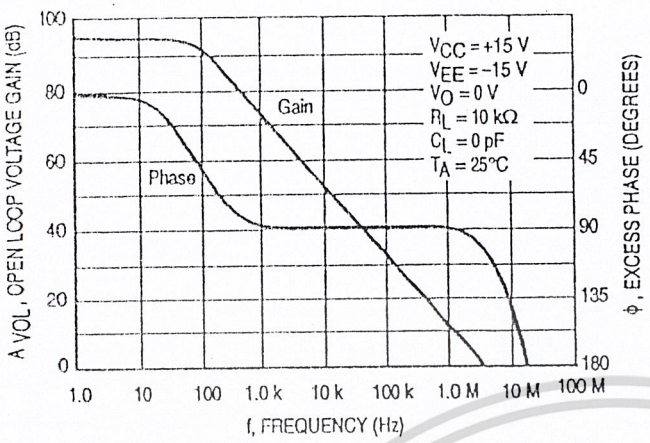


Figure 8. Supply Current per Amplifier versus Supply Voltage

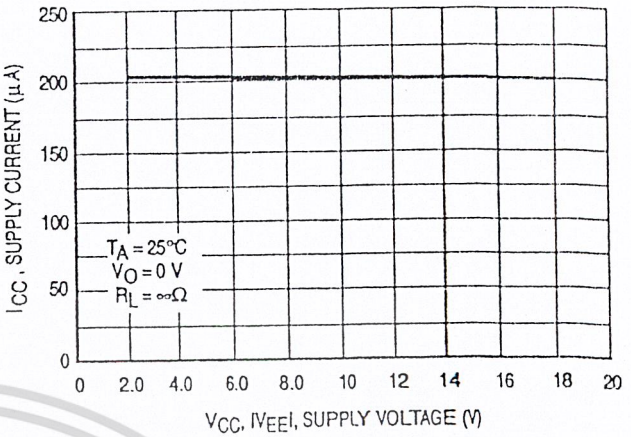


Figure 9. Supply Current per Amplifier versus Temperature

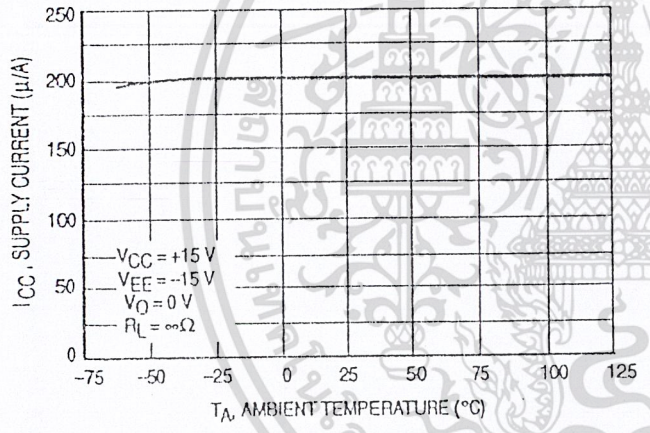


Figure 10. Total Power Dissipation versus Temperature

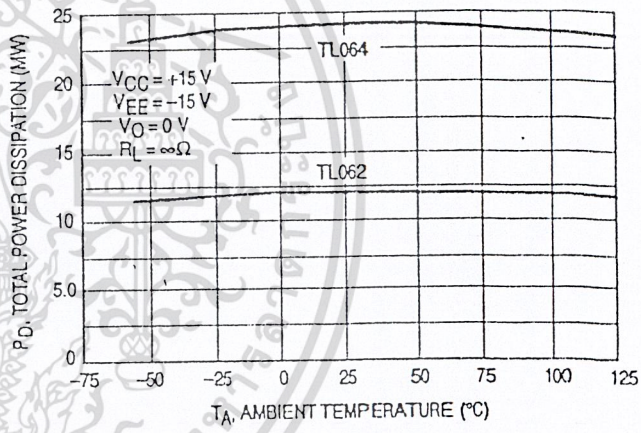


Figure 11. Common Mode Rejection versus Temperature

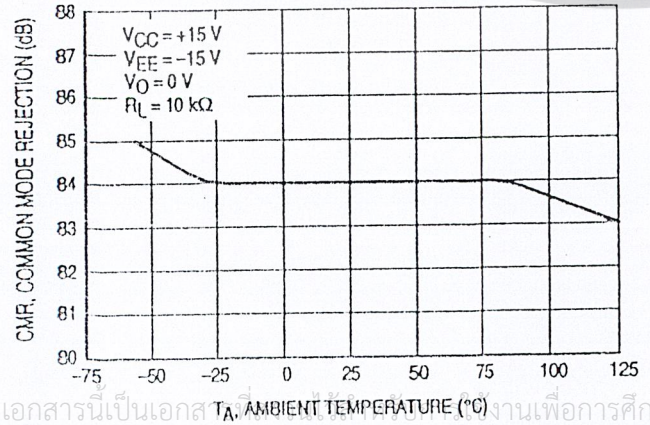
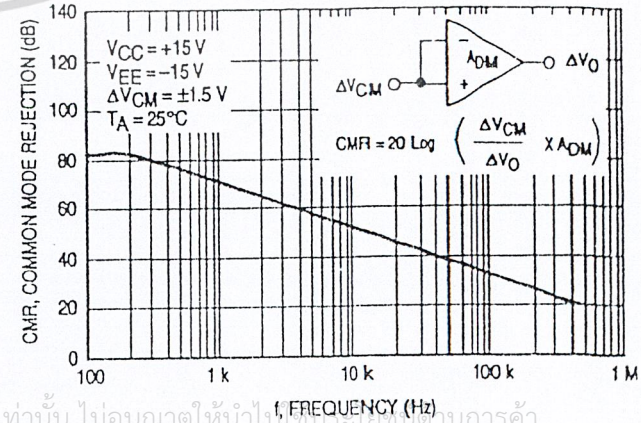


Figure 12. Common Mode Rejection versus Frequency



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 19. AC Amplifier

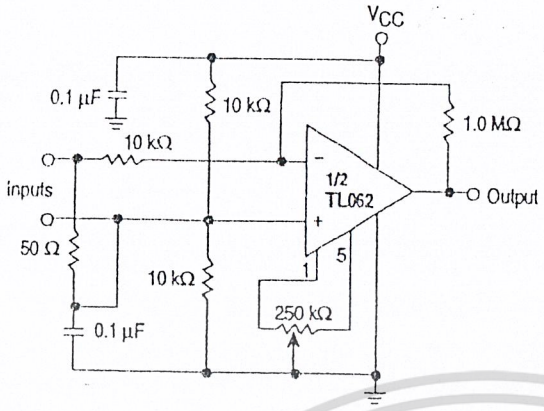


Figure 20. High–Q Notch Filter

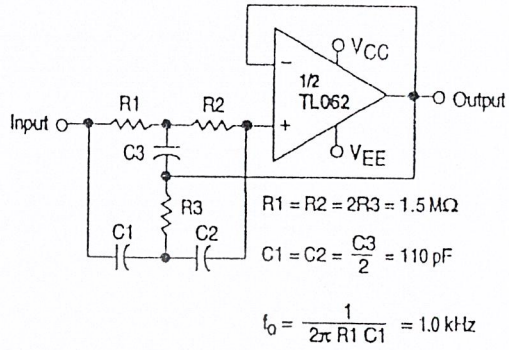


Figure 21. Instrumentation Amplifier

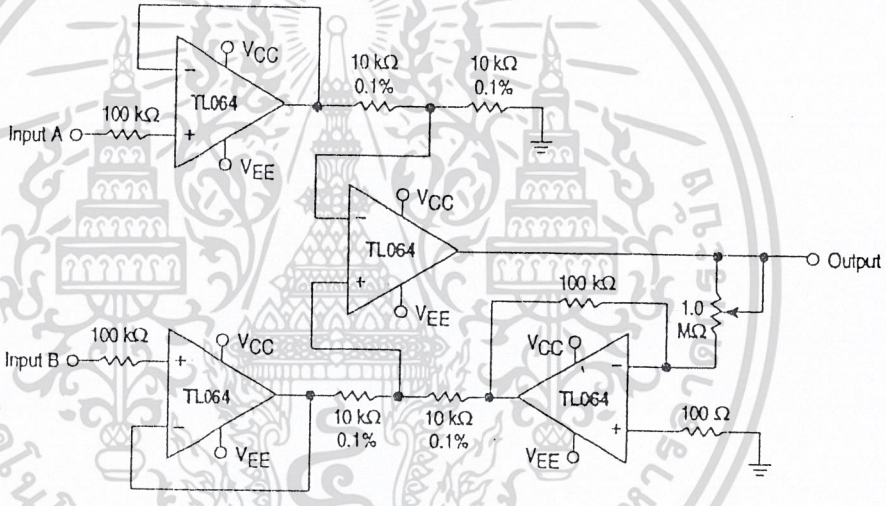


Figure 22. 0.5 Hz Square–Wave Oscillator

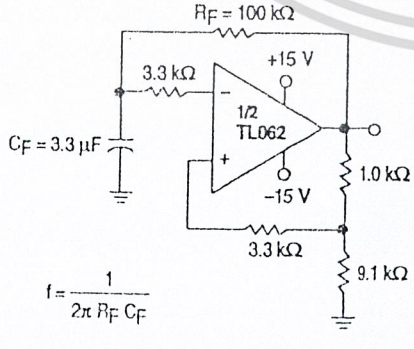
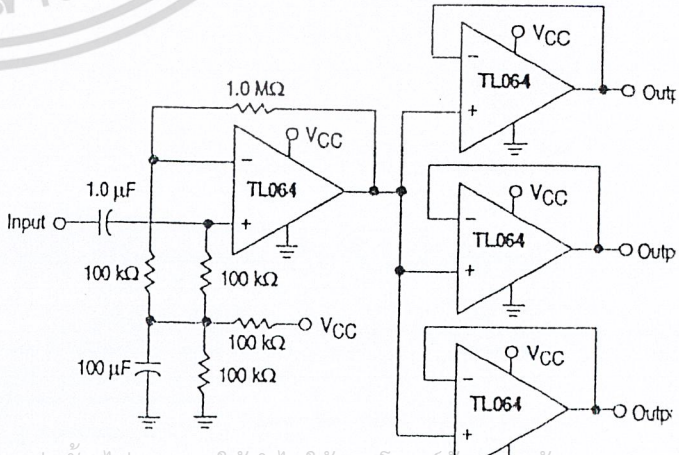


Figure 23. Audio Distribution Amplifier



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สามารถสำเร็จลุล่วงได้ด้วย คำปรึกษา คำแนะนำ และคำช่วยเหลือของอาจารย์ที่ปรึกษาคือ รศ.ดร. สุวิพล สิทธิชีวภาค และ ผศ. เกรียงไกร วงศ์โรจนภรณ์ รวมถึงพี่ปริญญาโทที่คอยให้คำชี้แนะในเรื่องอุปกรณ์ และเพื่อนๆ ทุกคนที่ให้ความช่วยเหลือตลอดมา จึงขอขอบพระคุณมา ณ. โอกาสนี้

นาย เกรียงไกร วระพันธุ์

นาย เฉลิมศักดิ์ วงศ์โพธิ์ทอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

วิวัฒน์ กิรานนท์, “วิสวกรรมการสื่อสาร” สถาบันเทคโนโลยี พระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2540

ภาควิชาโทรคมนาคม ,“Telecommunication Lab I” สถาบันเทคโนโลยี พระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2543

ฉัทหวุฒิ พืชผล,"Visual Basic6" กรุงเทพ : โปรวิชัน ,2542

ทวีชัย ภูรีทิพย์,"ไขปัญหา RS232" ซีเอ็ดยูเคชั่น,2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้