

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องวัดคลื่นไฟฟ้าหัวใจผ่านเครือข่ายคอมพิวเตอร์
Electrocardiogram via Computer Network



โดย

นางสาวพินนภา แก้วขาว

นายวิศาล จูติธรรมจรรยา

เลขหมู่.....
เลขทะเบียน..... **62723**
วัน,เดือน,ปี..... **21 ส.ค. 2549**

b..... **41721010**
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องวัดคลื่นไฟฟ้าหัวใจผ่านเครือข่ายคอมพิวเตอร์

Electrocardiogram via Computer Network



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

ผ่านการตรวจชิ้นงานแล้ว

(ลงชื่อ).....ผู้ตรวจ

ผ่านการตรวจรูปเล่มแล้ว

(ลงชื่อ).....ผู้ตรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านอื่นๆ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2548

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

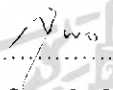
เรื่อง เครื่องวัดคลื่นไฟฟ้าหัวใจผ่านเครือข่ายคอมพิวเตอร์

Electrocardiogram via Computer Network

ผู้จัดทำ

1. นางสาวพินนภา แก้วขาว 46015063
2. นายวิศาล ฐิติธรรมจริยา 46015069


..... อาจารย์ที่ปรึกษา
(ผศ. เกียรติงไ.เร วงศ์โรจนกรณ์)


..... อาจารย์ที่ปรึกษา
(รศ.ดร. สุวิทย์ อธิธิธรรม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องวัดคลื่นไฟฟ้าหัวใจผ่านเครือข่ายคอมพิวเตอร์

Electrocardiogram via Computer Network

โดย นางสาวพินนภา แก้วขาว 46015063

นายวิศาล จูติธรรมจริยา 46015069

อาจารย์ที่ปรึกษา ผศ. เกรียงไกร วงศ์โรจนภรณ์

รศ.ดร. สุวิพล สีทธิชีวภาค

บทคัดย่อ

โครงการนี้เป็นการนำเอาเทคโนโลยีทางด้านคอมพิวเตอร์มาผสมผสานกับทางการแพทย์เพื่อทำให้แพทย์ผู้วินิจฉัยสามารถวินิจฉัยคลื่นไฟฟ้าหัวใจของผู้ป่วยที่อยู่ห่างไกล พร้อมกันนี้ยังทำให้แพทย์สามารถวินิจฉัยคลื่นไฟฟ้าหัวใจของผู้ป่วยหลายคนไปพร้อมกันได้ โดยโครงการนี้ใช้ไมโครโปรเซสเซอร์ควบคุมการวัดคลื่นไฟฟ้าหัวใจและส่งสัญญาณผ่านระบบเครือข่ายมายังคอมพิวเตอร์หลักแล้วคอมพิวเตอร์หลักจะนำเสนอสัญญาณคลื่นไฟฟ้าหัวใจที่ได้รับ ไปประมวลผลต่อ

ABSTRACT

This project presents a Electrocardiogram (ECG) via Computer Network by using microprocessor controls ECG and transmit the data of ECG to a main computer. Then the data will be processed by the main computer. It is convenient for the doctor to examine more than one patient at the same time.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณ ผศ.เกรียงไกร วงศ์โรจนภรณ์ รศ.ดร.สุวิพล สิริชีวะภาค ซึ่งเป็นอาจารย์ที่ปรึกษา ที่คอยให้คำปรึกษา และแนะนำ ให้การดำเนินโครงการเป็นไปได้อย่างดี ขอขอบคุณ คุณเคน มุละสิวะ ที่เอื้อเฟื้ออุปกรณ์ที่ใช้ในการทดลอง

ขอขอบพระคุณ พ่อและแม่ที่คอยให้กำลังใจ อบรมสั่งสอน ตลอดมาจนถึงทุกวันนี้

นางสาวพินณา แก้วขาว

นายวิศาล ฐิติธรรมจริยา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่ 1 บทนำ	หน้า
1.1 แนวคิดและที่มาของโครงการ	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของโครงการ	2
1.4 ผลที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 ทฤษฎีเกี่ยวกับหัวใจ	3
2.2 อิเล็กทรอนิกส์	12
2.3 ความต้องการ โดยเฉพาะของเครื่อง อี.ซี.จี.	17
2.4 ทฤษฎีเกี่ยวกับเครือข่ายคอมพิวเตอร์	24
2.5 ทฤษฎีเกี่ยวกับ Rabbit Core RCM2200	31
บทที่ 3 การออกแบบวงจร	34
3.1 การออกแบบสามารถแบ่งออกเป็น 2 ส่วนใหญ่	34
3.2 การออกแบบและการสร้างภาคขยายคลื่นไฟฟ้าหัวใจ	34
บทที่ 4 การทดลองและผลการทดลอง	43
4.1 ทดสอบการทำงานวงจรอินสตรูเมนต์แอมพลิไฟเออร์	44
4.2 ทดสอบการทำงานวงจรปรับศูนย์อัตโนมัติ (Auto-Zero Circuit)	48
4.3 ทดสอบการทำงานวงจรกรองความถี่แบบไม่ให้ความถี่ผ่านเฉพาะช่วง (Notch Filter)	51
4.4 ทดสอบการทำงานวงจรกรองความถี่ต่ำผ่าน (Low Pass Filter)	54
4.5 ทดสอบการทำงานวงจรพัลส์ – วิตซ์ มอดูเลเตอร์ (Pulse – Width Modulator)	57
4.6 ทดสอบการทำงานวงจรส่งผ่านสัญญาณด้วยแสง (Opto Isolator)	59
4.7 ทดสอบการทำงานวงจรดีมอดูเลเตอร์ (Demodulator)	60
4.8 ทดสอบการทำงานวงจรขยายและปรับระดับสัญญาณ (Amplifier and Clamp Circuit)	63
4.9 ทดสอบการทำงานวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล	64
4.10 ผลการทดลองที่ได้จากการส่งข้อมูลผ่านเครือข่ายคอมพิวเตอร์	65
บทที่ 5 บทวิจารณ์และบทสรุป	67
ภาคผนวก	
กิตติกรรมประกาศ	
เอกสารอ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 1.1 การเชื่อมต่อใช้งาน RCM2200 ในระบบเครือข่าย	1
รูปที่ 2.1 หัวใจ	3
รูปที่ 2.2 หัวใจแนวตัดจากบนลงล่าง	3
รูปที่ 2.3 ระบบไฟฟ้าหัวใจ	4
รูปที่ 2.4 แสดงส่วนต่างๆ ของระบบการนำไฟฟ้าหัวใจ	5
รูปที่ 2.5 แสดงคลื่นไฟฟ้าหัวใจของคนปกติ	6
รูปที่ 2.6 เซลล์กล้ามเนื้อหัวใจในระยะพัก (the resting or polarized state)	7
รูปที่ 2.7 แสดง Upward deflection คลื่นจะขึ้นบนถ้าการกระจายของคลื่นไฟฟ้า พุ่งตรงไปหา electrode	8
รูปที่ 2.8 Downward deflection คลื่นจะลงล่างถ้าการกระจายของคลื่นไฟฟ้าพุ่งออกจาก electrode	8
รูปที่ 2.9 แสดงขบวนการของ depolarization	8
รูปที่ 2.10 แสดงขบวนการของ repolarization	9
รูปที่ 2.11 วิธีการวัดแบบ Standard Limb Lead	10
รูปที่ 2.12 วิธีการวัดแบบ Augment Limb Lead	11
รูปที่ 2.13 วิธีการวัดคลื่นไฟฟ้าหัวใจแบบ Unipolar Chest Lead	11
รูปที่ 2.14 ตำแหน่งการติดตั้งขั้ววัดแบบ Unipolar Chest Lead	11
รูปที่ 2.15 ตำแหน่งการติดขั้ววัดหัวใจของวิธีการวัดคลื่นไฟฟ้าเพื่อการมอนิเตอร์	12
รูปที่ 2.16 แสดงโครงสร้างผิวหนังและวงจรสมมูลของผิวหนังที่ติดอิเล็กโทรด	13
รูปที่ 2.17 ความต้านทานที่เปลี่ยนตามความถี่	14
รูปที่ 2.18 แสดงคุณสมบัติของอิเล็กโทรดที่เป็นหลักไร้สนิมซึ่งอิมพีแดนซ์ขึ้นกับค่าความเก็บประจุ และความเข้มของกระแสไฟฟ้า	15
รูปที่ 2.19 อิเล็กโทรดแผ่นที่ทำด้วยโฟม (ชนิดใช้แล้วทิ้งเลย) สำหรับเครื่องอี.ซี.จี. ใช้ปิดผิวหนังติดแขน-ขา	16
รูปที่ 2.20 แสดงความผิดพลาดของรูปคลื่น อี.ซี.จี. ซึ่งมีผลมาจากความผิดเพี้ยนของความถี่	18
รูปที่ 2.21 แสดงความผิดพลาดของคลื่น อี.ซี.จี. ซึ่งมีผลจากการผิดเพี้ยนที่เกิดจากแอมพลิไฟเออร์	19
รูปที่ 2.22 แสดงการต่อสายดินของเครื่อง อี.ซี.จี. ร่วมกับอุปกรณ์ไฟฟ้าอื่นๆ	20
รูปที่ 2.23 ผลของสิ่งรบกวนที่เกิดจากแรงดันไฟฟ้าแรงสูงระยะสั้น	21
รูปที่ 2.24 การลดสัญญาณรบกวนโดยการพันสายเป็นเกลียว	22
รูปที่ 2.25 ระบบเครือข่ายคอมพิวเตอร์	24
รูปที่ 2.26 เครือข่ายแบบดาว	25
รูปที่ 2.27 เครือข่ายแบบวงแหวน	25
รูปที่ 2.28 เครือข่ายแบบบัส	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.29 เครือข่ายท้องถิ่น (Local Area Network หรือ LAN)	26
รูปที่ 2.30 เครือข่ายระดับเมือง (Metropolitan Area Network หรือ MAN)	26
รูปที่ 2.31 เครือข่ายระดับประเทศ (Wide Area Network หรือ WAN)	26
รูปที่ 2.32 ระบบเครือข่าย TCP/IP	28
รูปที่ 2.33 การตีความหมายของ DNS	29
รูปที่ 2.34 UDP Header	30
รูปที่ 2.35 Pseudo Header	30
รูปที่ 2.36 บอร์ดวงจร Rabbit Core RCM2200	31
รูปที่ 2.37 อินพุตและเอาต์พุตของ Rabbit Core RCM2200	32
รูปที่ 2.38 การเชื่อมต่อ Board RCM2200 กับระบบคอมพิวเตอร์	32
รูปที่ 3.1 บล็อกไดอะแกรมของภาคขยายคลื่นหัวใจ	34
รูปที่ 3.2 วงจรอินสตรูเมนแอมพลิไฟเออร์	36
รูปที่ 3.3 วงจรปรับศูนย์อัตโนมัติ	37
รูปที่ 3.4 วงจรกรองความถี่แบบกั้นไม่ให้ความถี่ผ่านเฉพาะช่วง	38
รูปที่ 3.5 วงจรกรองความถี่ต่ำผ่าน	38
รูปที่ 3.6 วงจรพัลส์-วิดท์ มอดูเลเตอร์	39
รูปที่ 3.7 แสดงการเปลี่ยนความกว้างพัลส์โดยการมอดูเลตแบบ PWM	39
รูปที่ 3.8 วงจรส่งผ่านสัญญาณด้วยแสง	40
รูปที่ 3.9 วงจรตีมอดูเลเตอร์	41
รูปที่ 3.10 วงจรขยายและปรับระดับสัญญาณ ไฟฟ้าหัวใจ	42
รูปที่ 3.11 วงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัล	42
รูปที่ 4.1 สัญญาณและสเปกตรัมของคลื่นไฟฟ้าหัวใจจากเครื่องซิมูเลเตอร์	43
รูปที่ 4.2 แสดงวงจรอินสตรูเมนแอมพลิไฟเออร์ในการทดสอบ	44
รูปที่ 4.3 กราฟแสดงผลตอบสนองทางความถี่ของวงจรอินสตรูเมนแอมพลิไฟเออร์	46
รูปที่ 4.4 ผลที่ได้จากเครื่องซิมูเลเตอร์ของวงจรอินสตรูเมนแอมพลิไฟเออร์	47
รูปที่ 4.5 ผลที่วัดได้จากการวัดจริงของวงจรอินสตรูเมนแอมพลิไฟเออร์	47
รูปที่ 4.6 กราฟแสดงผลตอบสนองทางความถี่ของวงจรปรับศูนย์อัตโนมัติ	49
รูปที่ 4.7 ผลที่ได้จากเครื่องซิมูเลเตอร์ของวงจรปรับศูนย์อัตโนมัติ	50
รูปที่ 4.8 ผลที่วัดได้จากการวัดจริงของวงจรปรับศูนย์อัตโนมัติ	50
รูปที่ 4.9 กราฟแสดงผลตอบสนองทางความถี่ของวงจรกรองความถี่แบบ กั้นไม่ให้ความถี่ผ่านเฉพาะช่วง	52
รูปที่ 4.10 ผลที่ได้จากเครื่องซิมูเลเตอร์ของวงจรกรองความถี่แบบ กั้นไม่ให้ความถี่ผ่านเฉพาะช่วง	53
รูปที่ 4.11 ผลที่ได้จากการวัดจริงของวงจรกรองความถี่แบบ	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กันไม่ให้ความถี่ผ่านเฉพาะช่วง

รูปที่ 4.12 กราฟแสดงผลตอบสนองทางความถี่ของวงจรกรองความถี่ต่ำผ่าน	55
รูปที่ 4.13 ผลที่ได้จากเครื่องซิมูเลเตอร์ของวงจรกรองความถี่ต่ำผ่าน	56
รูปที่ 4.14 ผลที่ได้จากการวัดจริงของวงจรกรองความถี่ต่ำผ่าน	56
รูปที่ 4.15 แสดงสัญญาณคลื่นไฟฟ้าหัวใจและสัญญาณสามเหลี่ยม	57
รูปที่ 4.16 ผลที่ได้จากเครื่องซิมูเลเตอร์ของวงจรพัลส์-วิคท์ มอดูเลเตอร์	58
รูปที่ 4.17 ผลที่ได้จากการวัดจริงของวงจรพัลส์-วิคท์ มอดูเลเตอร์	58
รูปที่ 4.18 ผลที่ได้จากเครื่องซิมูเลเตอร์ของวงจรส่งผ่านสัญญาณด้วยแสง	59
รูปที่ 4.19 ผลที่ได้จากการวัดจริงของวงจรส่งผ่านสัญญาณด้วยแสง	59
รูปที่ 4.20 ผลที่ได้จากเครื่องซิมูเลเตอร์ของวงจรอินทรีเกรเตอร์	60
รูปที่ 4.21 ผลที่ได้จากการวัดจริงของวงจรอินทรีเกรเตอร์	60
รูปที่ 4.22 กราฟแสดงผลตอบสนองทางความถี่ของวงจรกรองความถี่ต่ำผ่าน	61
รูปที่ 4.23 ผลที่ได้จากเครื่องซิมูเลเตอร์ของวงจรกรองความถี่ต่ำผ่าน	62
รูปที่ 4.24 ผลที่ได้จากการวัดจริงของวงจรกรองความถี่ต่ำผ่าน	62
รูปที่ 4.25 ผลที่ได้จากเครื่องซิมูเลเตอร์ของวงจรขยายและปรับระดับสัญญาณ	63
รูปที่ 4.26 ผลที่ได้จากการวัดจริงของวงจรขยายและปรับระดับสัญญาณ	63
รูปที่ 4.27 แสดงผลสัญญาณคลื่นไฟฟ้าหัวใจจากเครื่องซิมูเลเตอร์	65
รูปที่ 4.28 แสดงผลสัญญาณคลื่นไฟฟ้าหัวใจจากการวัดจริง	66

สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงช่วงของ UDP Header	30
ตารางที่ 4.1 ผลการทดลองการวัดแรงดันทางเอาต์พุทของวงจรอินสตรูเมนแอมป์ลิไฟเออร์	45
ตารางที่ 4.2 แสดงผลตอบสนองทางความถี่ของวงจรปรับศูนย์กลางอัตโนมัติ	48
ตารางที่ 4.3 แสดงผลตอบสนองทางความถี่ของวงจรกรองความถี่ แบบกั้นไม่ให้ความถี่ผ่านเฉพาะช่วง	51
ตารางที่ 4.4 แสดงผลตอบสนองทางความถี่ของวงจรกรองความถี่ต่ำผ่าน	54
ตารางที่ 4.5 แสดงผลตอบสนองทางความถี่ของวงจรกรองความถี่ต่ำผ่าน	61
ตารางที่ 4.6 ผลที่ได้จากการทดลองวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล	64



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

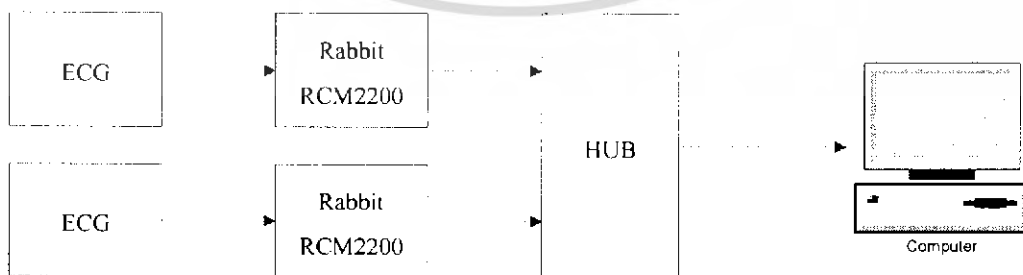
บทนำ

ในอดีตเครื่องวัดคลื่นสัญญาณไฟฟ้าหัวใจมีลักษณะเป็นเครื่องมือที่แพทย์จะต้องเดินมาวัดที่ห้องผู้ป่วย และบันทึกผลลงบนม้วนกระดาษซึ่งเป็นการยุ่งยากถ้าแพทย์อยู่ในที่ไกลๆ และการเก็บผลเป็นกระดาษอาจทำให้โอกาสที่ข้อมูลจะเสียหายมีเพิ่มมากขึ้นอีกด้วย

ดังนั้นในปัจจุบันนี้ซึ่งเป็นยุคของข้อมูลข่าวสาร ได้มีการคิดค้นและพัฒนาเทคโนโลยีสารสนเทศขึ้นอย่างมากมาย ทางผู้จัดทำจึงคิดที่จะนำเทคโนโลยีนี้มาประยุกต์ใช้ในทางการแพทย์ โดยการส่งข้อมูลสัญญาณไฟฟ้าหัวใจที่วัดได้จากผู้ป่วยในห้องผู้ป่วย เข้าคอมพิวเตอร์แล้วส่งผ่านข้อมูลไปยังเครื่องคอมพิวเตอร์ในห้องพักของแพทย์ ซึ่งจะทำให้แพทย์สามารถมองเห็นสัญญาณไฟฟ้าหัวใจของผู้ป่วยได้ โดยที่ไม่ต้องเดินไปตรวจที่ห้องผู้ป่วย ซึ่งจะอำนวยความสะดวกแก่แพทย์ในการดูแลผู้ป่วยเป็นอย่างมาก อีกทั้งการที่สามารถเก็บค่าลงในไฟล์และอ่านค่าจากไฟล์คอมพิวเตอร์ได้ยังเป็นการช่วยลดปริมาณการใช้กระดาษในการทำงานและเป็นการเก็บรักษาข้อมูลที่ดีอีกด้วย

1.1 แนวคิดและที่มาของโครงการ

ปัจจุบันไมโครโปรเซสเซอร์ (Microprocessor) มีการพัฒนาไปมากทำให้มีความสามารถในด้านต่างๆเพิ่มมากขึ้น ซึ่งในงานอุตสาหกรรมไมโครโปรเซสเซอร์ได้มีการถูกใช้ในระบบควบคุมอัตโนมัติ และกึ่งอัตโนมัติ รวมไปถึงสิ่งอำนวยความสะดวกในชีวิตประจำวัน เนื่องจากคุณสมบัติของไมโครโปรเซสเซอร์มีขนาดเล็ก และมีความสามารถในการประมวลผลข้อมูลสูง จึงได้นำไมโครโปรเซสเซอร์มาประยุกต์ใช้ในโครงการนี้ซึ่งไมโครโปรเซสเซอร์ที่นำมาใช้คือ RCM2200 (Rabbit Core RCM2200 C-Programmable Module with Ethernet) ได้มีการพัฒนาการรับส่งข้อมูลที่รวดเร็วผ่านโปรโตคอล TCP/IP และ UDP จึงทำให้ RCM2200 มีความสามารถในการสื่อสารผ่านเครือข่ายคอมพิวเตอร์ ในโครงการนี้ได้นำเอา RCM2200 มาใช้ในการส่งสัญญาณคลื่นไฟฟ้าหัวใจ ECG (Electrocardiogram) ผ่านเครือข่ายคอมพิวเตอร์ดังรูป



รูปที่ 1.1 การเชื่อมต่อใช้งาน RCM2200 ในระบบเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์

เพื่อศึกษาและเรียนรู้การใช้งานไมโคร โปรเซสเซอร์ RCM2200 ในการรับส่งข้อมูลผ่านเครือข่าย และนำไปประยุกต์ใช้งาน โดยใช้ไมโคร โปรเซสเซอร์ RCM2200 เป็นตัวกลางระหว่างเครื่องคอมพิวเตอร์ ที่ใช้อ่านข้อมูลกับเครื่องวัดคลื่นไฟฟ้าหัวใจ

1.3 ขอบเขตของโครงการ

นำไมโครโปรเซสเซอร์ RCM2200 มาเป็นตัวเชื่อมต่อ และส่งข้อมูลระหว่างเครื่องวัดคลื่นไฟฟ้าหัวใจกับเครื่องคอมพิวเตอร์หลักที่ใช้อ่านสัญญาณคลื่นไฟฟ้าหัวใจ โดยสัญญาณคลื่นไฟฟ้าหัวใจที่ได้รับมาจากเครื่องวัดคลื่นไฟฟ้าหัวใจ 2 เครื่อง เพื่อนำมาแสดงผลบนหน้าจอคอมพิวเตอร์หลัก

1.4 ผลที่คาดว่าจะได้รับ

1. สามารถส่งสัญญาณคลื่นไฟฟ้าหัวใจผ่านระบบเครือข่ายได้
2. สามารถแสดงผลสัญญาณคลื่นไฟฟ้าหัวใจ ที่มาจากเครื่องวัดคลื่นไฟฟ้าหัวใจ 2 เครื่อง บนหน้าจอคอมพิวเตอร์หลักได้
3. สามารถจะพัฒนาการรับส่งข้อมูลให้ดีขึ้นในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

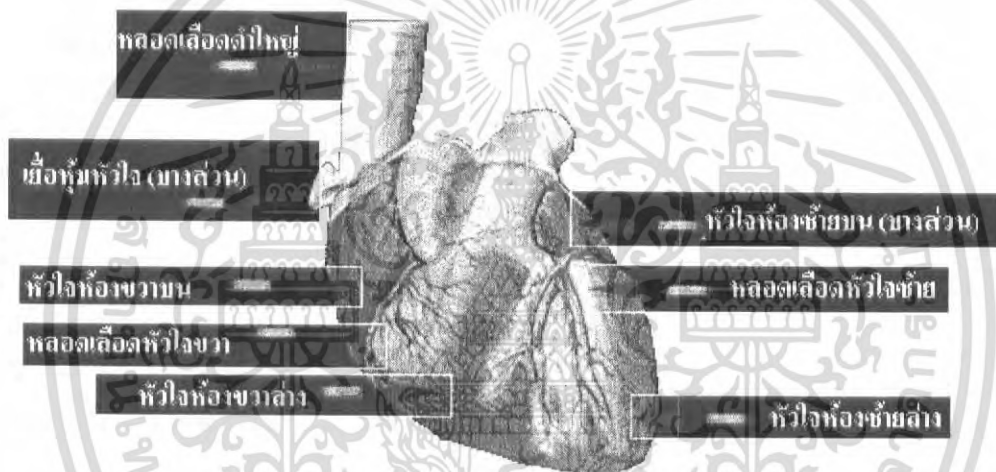
ทฤษฎีและหลักการ

2.1 ทฤษฎีเกี่ยวกับหัวใจ

หัวใจเป็นอวัยวะที่มีความสำคัญและแข็งแรงมาก มีโครงสร้างภายนอกที่ไม่ซับซ้อนมาก แต่ในส่วนของการทำงานแล้ว จะซับซ้อนมาก การทำงานของแต่ละระบบในหัวใจต้องมีความสัมพันธ์กัน หัวใจจึงจะทำงานได้อย่างปกติ

หัวใจเป็นอวัยวะที่มีความมหัศจรรย์มาก กลุ่มเซลล์มีการก่อสร้างหัวใจขึ้นตั้งแต่ยังเป็นตัวอ่อนในครรภ์มารดา หัวใจเริ่มเต้นแล้ว และจะเต้นตลอดไป ไม่มีวันหยุดตราบเท่าที่ยังมีชีวิต หัวใจไม่เคยมีวันพัก จึงเป็นอวัยวะที่มีความแข็งแรงมาก

สามารถแบ่งหัวใจออกตามลักษณะ(กายวิภาค) และตามหน้าที่ได้ ดังนี้



รูปที่ 2.1 หัวใจ



รูปที่ 2.2 หัวใจแนวตัดจากบนลงล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เยื่อหุ้มหัวใจ เป็นเยื่อบางๆ ใสๆ ห่อหุ้มหัวใจไว้

หลอดเลือดหัวใจ จะอยู่บริเวณภายนอกหัวใจ (เยื่อหุ้มหัวใจ) ส่งแขนงเล็กๆลงไปเลี้ยงกล้ามเนื้อหัวใจ หลอดเลือดหัวใจมีเส้นใหญ่ๆ อยู่ 2 เส้น คือ ขวา (right coronary artery) เลี้ยงหัวใจด้านขวา และซ้าย (left coronary artery) เลี้ยงหัวใจด้านซ้ายเป็นส่วนใหญ่ ด้านซ้ายจะแตกแขนงใหญ่ๆ 2 แขนง คือ left anterior descending artery และ left circumflex artery ซึ่งจะมีแขนงเล็กๆ อีกมากมาย

กล้ามเนื้อหัวใจ เป็นส่วนที่ทำหน้าที่ในการบีบตัวไล่เลือดไปเลี้ยงส่วนต่างๆของร่างกาย และขยายตัวเพื่อรับเลือดกลับเข้าสู่หัวใจ จึงเป็นส่วนที่มีความสำคัญอย่างมาก หากกล้ามเนื้อหัวใจบีบตัวหรือคลายตัวผิดปกติแล้ว ก็จะทำให้เกิดปัญหาต่างๆตามมา ซึ่งส่วน มากอาจไม่สามารถแก้ไขให้กลับเป็นปกติได้ การทำงานของกล้ามเนื้อหัวใจต้องอาศัยพลังงานที่ได้จากสารอาหารที่ถูกนำมา โดยหลอดเลือดหัวใจ

ลิ้นหัวใจ และ ผนังกันห้องหัวใจ หัวใจคนเรามี 4 ห้องแบ่ง ซ้าย-ขวา ซึ่งถูกกั้นโดยผนังของกล้ามเนื้อหัวใจ และแบ่งห้อง บน-ล่าง เลือดระหว่างห้องซ้าย-ขวาก็ไม่ปะปนกัน ลิ้นหัวใจทำหน้าที่ให้เลือดไหลผ่านและไม่ไหลย้อนกลับ



รูปที่ 2.3 ระบบไฟฟ้าหัวใจ

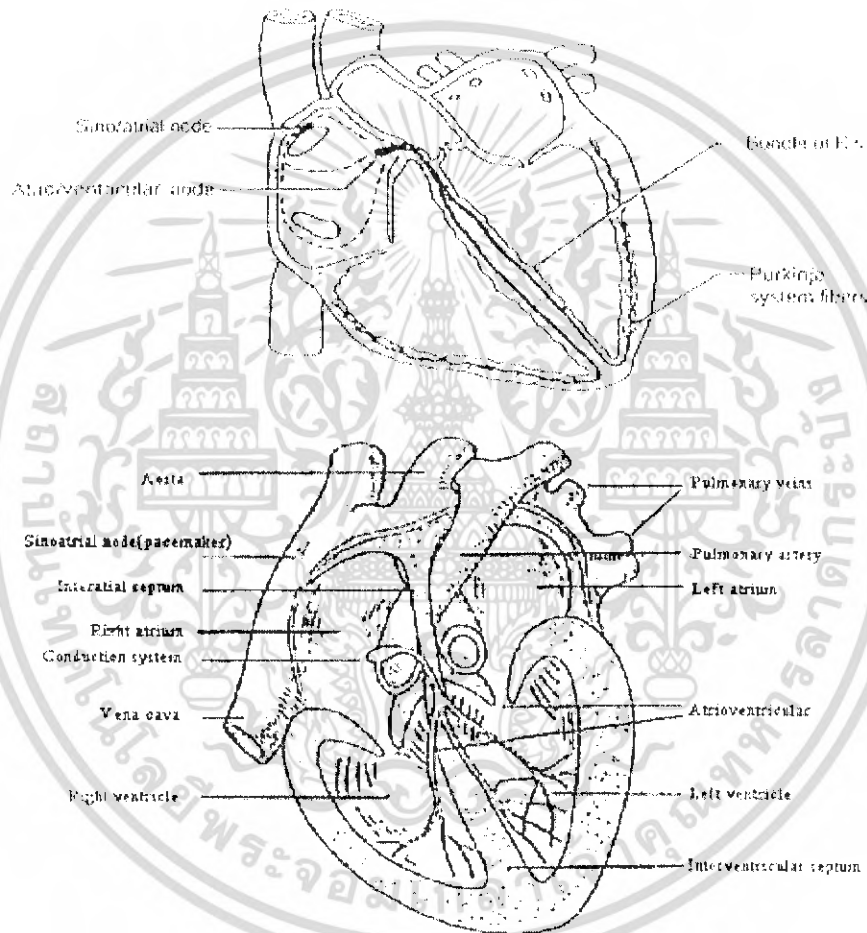
ระบบไฟฟ้าหัวใจ การที่หัวใจสามารถเต้นได้นั้น เนื่องจากหัวใจส่งกระแสไฟฟ้าที่สร้างขึ้นเอง จากหัวใจห้องขวาบนมายัง หัวใจห้องซ้ายบนและห้องล่าง เมื่อไฟฟ้าผ่านไปจะกระตุ้นให้กล้ามเนื้อหัวใจเกิดการบีบตัวไล่เลือด เลือดจึงไหลอย่างมีระเบียบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 การตรวจคลื่นไฟฟ้าหัวใจ

คลื่นไฟฟ้าหัวใจ หรือ กราฟหัวใจ หรือภาษาอังกฤษเรียกว่า **Electrocardiogram** ใช้คำย่อว่า **ECG** หรือ **EKG** ก็คือการตรวจจับกระแสไฟฟ้าที่ออกมาจากหัวใจ เมื่อเรานำเอาตัวจับสัญญาณไฟฟ้า (electrode) มาวางไว้ที่หน้าอก โคนหลังหัวใจ เราก็สามารถบันทึกไฟฟ้าที่ออกมาจากหัวใจนี้ได้

ระบบการนำไฟฟ้าของหัวใจ (Conducting System of the Heart) ประกอบด้วย sinoatrial node, atrioventricular node, common AV bundle (bundle of His), right และ left bundle branch และ peripheral หรือ Purkinje fiber นอกจากนี้ยังมีทางติดต่อระหว่าง node และระหว่างเอเดรียม (internodal และ interatrial pathway) ด้วย ซึ่งเชื่อว่ามีส่วนสำคัญใน paroxysmal supraventricular tachycardia



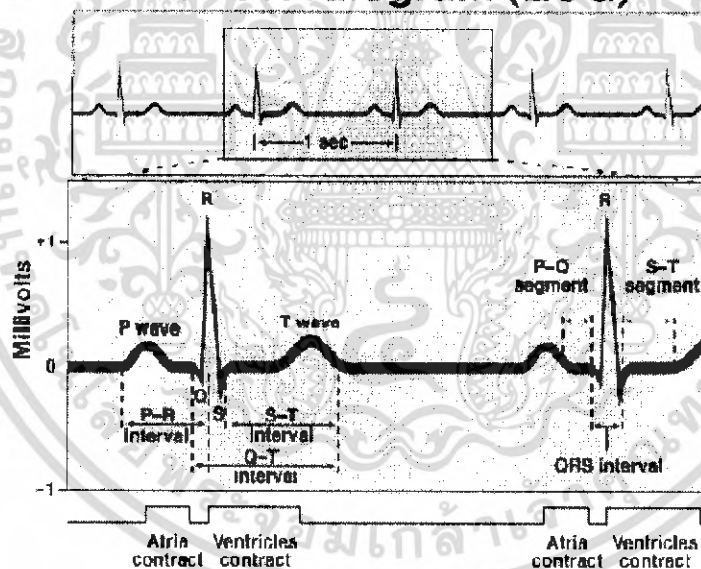
รูปที่ 2.4 แสดงส่วนต่างๆ ของระบบการนำไฟฟ้าหัวใจ

คลื่นหัวใจมีแหล่งกำเนิดเป็นเป็นสัญญาณอิมพัลส์ที่ บริเวณ SA node (Sinoatrial node) ในบริเวณผนังหัวใจซีกบนขวา (right atrium) และการกระจายพร้อมทั้งรีโพลาไรซ์ไปทั่วร่างกาย เมื่อวัดโดยการต่อขั้วไฟฟ้าเข้าแบบ ดิฟเฟอเรนเชียล เข้ากับผิวหนังบริเวณหน้าอก แขน ขา ได้ลูกคลื่น 1 คาบเวลา คล้ายกับรูปที่แสดงในรูปที่ 2.5 ในคนปกติ คลื่นไฟฟ้าหัวใจจะประกอบด้วยคลื่น P, QRS, T และ U ซึ่งลักษณะการมีอยู่ของคลื่นองค์ประกอบเหล่านี้ จะเป็นข้อมูลที่สำคัญในการวิเคราะห์การทำงานของหัวใจและความคิดปรกติ เมื่อเกิด อิมพัลส์และเริ่มกระจายไปบริเวณซีกบนของหัวใจ จะทำให้เกิดการบีบตัวของกล้ามเนื้อด้านบนของหัวใจโลหิตในหัวใจห้องบนด้านขวาหรือด้านซ้ายนั้นจะถูกบีบลงไป เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้านล่างของหัวใจอิมพัลส์นี้จะถูกแพร่กระจายผ่านเนื้อเยื่อตัวนำมายังผิวหนังจะทำให้เกิดสัญญาณที่เรียกว่า P Wave ในบริเวณ atrioventricular node จะมีการหน่วงเวลาของการกระตุ้นอิมพัลส์เพื่อให้เลือดสามารถถ่ายโอนจากด้านบนของหัวใจไปด้านล่างของหัวใจเสร็จสมบูรณ์ การหน่วงเวลานี้เป็นส่วนหลักของ P-R Interval node ของรูปคลื่น ECG อิมพัลส์ที่เกิดจากการกระจายไปด้านล่างของหัวใจ ด้วยการคลายตัวของกล้ามเนื้อด้านล่างหัวใจเมื่อขบวนการเสร็จสมบูรณ์จะมีคาบเวลาเกิดซ้ำขึ้นอีก และจะผลิตรูปคลื่น ECG ออกมาอีกครั้ง

จะเห็นว่าส่วนผลิตรูปคลื่น ECG จะนำข้อมูลทางพยาธิสภาพของหัวใจ มาสู่แพทย์เพื่อวินิจฉัยอาการผิดปกติ ตัวอย่างเช่น R-R Interval สามารถบอกถึงอัตราการเต้นของหัวใจ (Cardiac Rhythm) ภายใต้อิทธิพลของระบบประสาทอัตโนมัติ ความไม่เสถียรของ Cardiac Rhythm สามารถบ่งถึงการเต้นที่ผิดปกติ ขนาดและช่วงเวลาของ P และ QRS บ่งชี้ถึงสภาพกล้ามเนื้อหัวใจ การลดทอนขนาดสัญญาณ อาจบ่งบอก การทำงานของกล้ามเนื้อหัวใจบริเวณที่เกี่ยวข้อง ขนาดเพิ่มอาจบอกความผิดปกติของหัวใจโต นอกจากนี้การหน่วงเวลานานเกินไปในจุด atrioventricular เป็นการบ่งบอกถึงการปิดกั้นทั้งหมดหรือบางส่วน ของ อิมพัลส์ จากการขาดช่วงการซิงโครไนซ์ระหว่าง P-Wave และ QRS Complex อาการผิดปกติสามารถรักษาได้ทางการใช้ยาและสังเกตผลการรักษาจากรูปคลื่น ECG

Electrocardiogram (ECG)



รูปที่ 2.5 แสดงคลื่นไฟฟ้าหัวใจของคนปกติ

บนรูปคลื่น ECG จะมีสัญญาณไฟฟ้าร่างกายชนิดอื่นๆ ปะปนมาด้วยเช่นสัญญาณ ECG จากสมอง สัญญาณ EMG จากกล้ามเนื้อ รวมทั้ง Motion Artifact ใดๆก็ได้ทางเทคนิคการแยกสัญญาณ ECG ออกมาจากสัญญาณเหล่านี้ได้ไม่ยากเนื่องจากคลื่นไฟฟ้าหัวใจมีความถี่สเปกตรัมในช่วง 3-40 Hz อุปกรณ์แสดงรูปคลื่น ECG ทางการแพทย์จะมีแบนวิดของการตอบสนองความถี่สำหรับการประยุกต์การใช้งานที่แตกต่างกันแบบที่ใช้งานในคลินิกที่ใช้สำหรับบันทึกมาตรฐาน 12 Lead ECG คือ 0.05-100 Hz ในกรณีที่คนไข้มีอาการทรุดหนัก แบนวิดของเครื่องวัดจะกำหนดไว้ที่ 0.5-50 Hz สำหรับการวัดของอัตรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเดินของหัวใจที่ใช้ทดสอบ QRS Complex จะตัดสัญญาณรบกวน P-Wave และ T-Wave ออกจาก ECG

2.1.2 คำศัพท์ต่างๆที่ควรทำความเข้าใจ

Electrocardiography เป็นการบันทึกการเปลี่ยนแปลงของศักดาไฟฟ้า (potential) ที่ผิวของร่างกายซึ่งเกิดจาก depolarization ของกล้ามเนื้อหัวใจ

Polarized state เป็นภาวะที่เซลล์กล้ามเนื้อหัวใจอยู่ในระยะพัก (resting state) จะคงไว้ซึ่งประจุลบภายในเซลล์และประจุบวกภายนอกเซลล์ ความแตกต่างของศักดาไฟฟ้าภายในเซลล์เมื่อเปรียบเทียบกับภายนอกเซลล์ คือ resting membrane potential

Depolarization เมื่อเซลล์กล้ามเนื้อหัวใจถูกกระตุ้น โซเดียม ไอออนจากภายนอกเซลล์จะเคลื่อนผ่านเข้าไปในเซลล์ ทำให้มีการเปลี่ยนแปลงของประจุไฟฟ้าภายในเซลล์กลายเป็นบวก ภายนอกเซลล์กลายเป็นลบสนามไฟฟ้าจะเกิดตรงบริเวณระหว่างส่วนที่ depolarize กับส่วนที่ polarize ของกล้ามเนื้อหัวใจ

P wave ตรงกับ atrial depolarization

QRS complex ตรงกับ ventricular depolarization

Repolarization ไอออนเคลื่อนที่อย่างช้าๆข้ามผ่านเมมเบรนเพื่อให้เซลล์กลับเข้าสู่สภาวะ polarize จุดเริ่มต้นของ repolarization เกิดขึ้นขณะที่โพแทสเซียมไอออนเคลื่อนออกจากเซลล์กล้ามเนื้อหัวใจหลังจาก repolarization แล้ว โพแทสเซียมและโซเดียมจะกลับเข้าสู่ตำแหน่งเดิมเพื่อคงไว้ซึ่งความเข้มข้นเดิมและเข้าสู่สภาวะ polarize

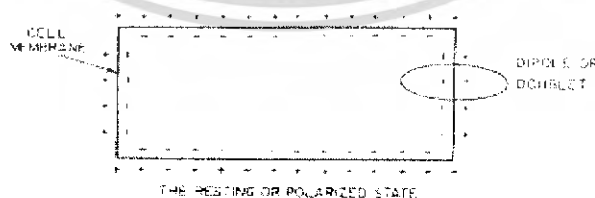
T wave ตรงกับ ventricular repolarization เกิดตามหลัง QRS complex

ST segment คือ isoelectric line นับจากจุดสิ้นสุดของ QRS complex ไปยังจุดเริ่มต้นของ T wave

Deflection หมายถึง wave หรือ complex ใดก็ตามที่เกิดในภาพคลื่นไฟฟ้าหัวใจ (electrocardiogram) ถ้าอยู่เหนือ base line ถือเป็นบวก อยู่ใต้ base line ถือเป็นลบ

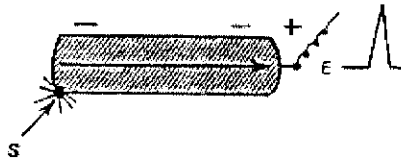
Upward deflection เกิดเมื่อทิศทางของ depolarization พุ่งเข้าหา electrode

Downward deflection เกิดเมื่อทิศทางของ depolarization พุ่งออกจาก electrode

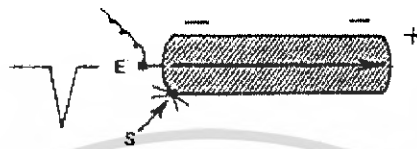


รูปที่ 2.6 เซลล์กล้ามเนื้อหัวใจในระยะพัก (the resting or polarized state)

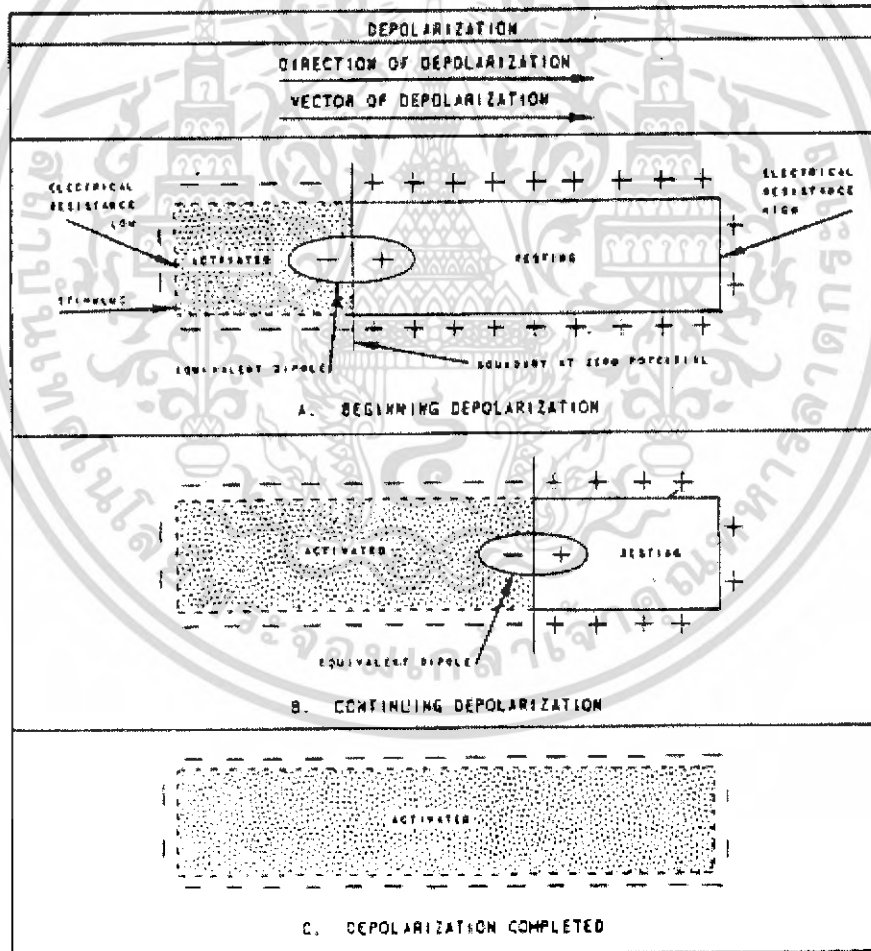
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดง Upward deflection คลื่นจะขึ้นบนถ้าการกระจายของคลื่นไฟฟ้าพุ่งตรงไปหา electrode

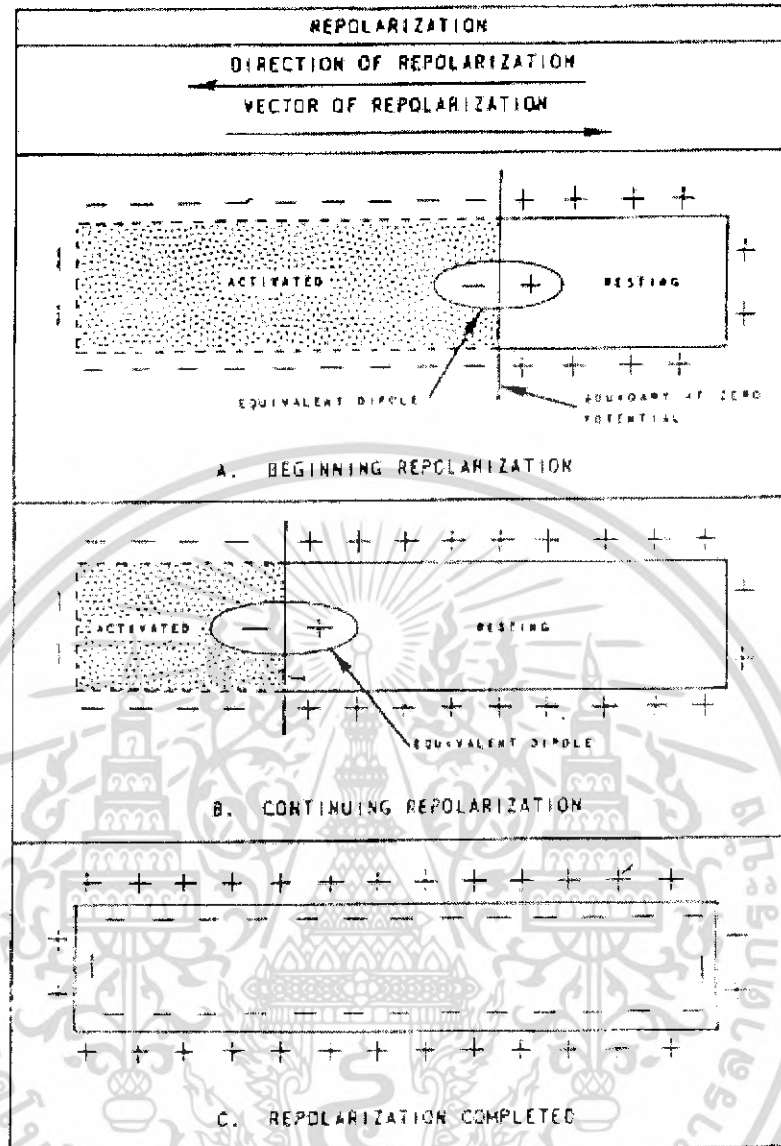


รูปที่ 2.8 Downward deflection คลื่นจะลงล่างถ้าการกระจายของคลื่นไฟฟ้าพุ่งออกจาก electrode



รูปที่ 2.9 แสดงขบวนการของ depolarization

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 แสดงขบวนการของ repolarization

2.1.3 การวัดคลื่นไฟฟ้าหัวใจ

การวัดคลื่นไฟฟ้าหัวใจสามารถทำได้ 2 รูปแบบคือการวัดแบบเวกเตอร์คาร์ดิโอกราฟ (Vector cardiograph) และการวัดแบบอิเล็กทรอนิกส์คาร์ดิโอกราฟ (Electrocardiograph) ซึ่งสามารถอธิบายได้ดังนี้

การวัดเวกเตอร์แบบคาร์ดิโอกราฟ คือการวัดการเปลี่ยนแปลงขนาดของเวกเตอร์ ของความต่างศักย์ที่เกิดขึ้น บนแกนหนึ่งเทียบกับอีกแกนหนึ่ง โดยพิจารณาจาก 3 แกนที่ตั้งฉากกันมีอยู่ด้วยกัน 3 ระนาบ คือ ระนาบที่มองทางด้านหน้า ด้านซ้าย และด้านบน วิธีนี้ต้องใช้แพทย์ผู้เชี่ยวชาญการวินิจฉัย

การวัดแบบอิเล็กทรอนิกส์คาร์ดิโอกราฟ คือการวัดการเปลี่ยนแปลงขนาดของเวกเตอร์ของ ความต่างศักย์ที่เกิดขึ้นในแนวแกนใดๆเทียบกับเวลา สามารถวินิจฉัยได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

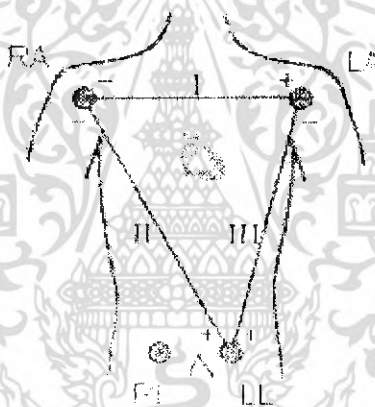
การวัดคลื่นหัวใจแบบอิเล็กทรอนิกส์โทรคาร์ดิโอกราฟมีเพื่อการวินิจฉัยระบบการทำงานของหัวใจสามารถ แบ่งตามวัตถุประสงค์ของการวัดได้ 2 ประเภทคือ เพื่อการวินิจฉัยคนไข้ข้างเตียงแบบมาตรฐาน (Standard Clinical ECG) และการวัดเพื่อการมอนิเตอร์ (Monitoring ECG)

2.1.3.1 การวัดเพื่อวินิจฉัยคนไข้ข้างเตียงแบบมาตรฐาน

การวัดเพื่อวินิจฉัยคนไข้ข้างเตียงแบบมาตรฐานนั้น เป็นการวัดคลื่นไฟฟ้าหัวใจโดยตำแหน่งที่เป็นการวัดได้กำหนดไว้เป็นมาตรฐานแล้ว วิธีการวัดเพื่อวินิจฉัยคนไข้ข้างเตียงสามารถแบ่งออกได้เป็น 3 วิธีคือ วิธีการวัดแบบ Standard Limb Lead วิธีการวัดแบบ Augment Limb Lead และวิธีการวัดแบบ Unipolar Chest Lead ซึ่งสามารถอธิบายได้ดังนี้

วิธีการวัดแบบ Standard Limb Lead

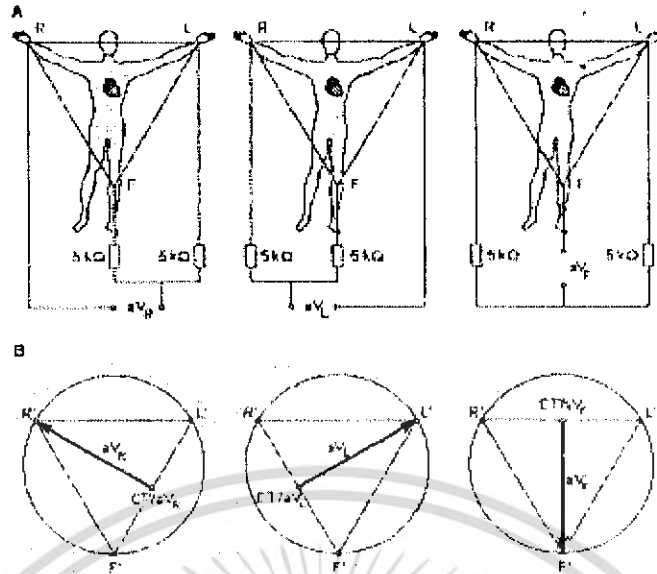
วิธีการวัดแบบ Standard Limb Lead เป็นการวัดสัญญาณคลื่นไฟฟ้าหัวใจประกอบไปด้วย Lead I, II และ III ดังรูปที่ 2.11 ซึ่งสามารถทำการวัดคลื่นไฟฟ้าหัวใจแบบ Standard Limb Lead ทั้ง Lead I, II และ III โดยการติดขั้ววัดของวงจรรขยายค่าความแตกต่าง



รูปที่ 2.11 วิธีการวัดแบบ Standard Limb Lead

วิธีการวัดแบบ Augment Limb Lead

วิธีการวัดแบบ Augment Limb Lead เป็นวิธีการวัดคลื่นหัวใจที่ประกอบด้วย Lead aVR, Lead aVL และ Lead aVF ดังรูปที่ 2.12 สำหรับวัดสัญญาณไฟฟ้าหัวใจแบบ Augment Limb Lead จะมีตัวต้านทานค่า $R/2$ ต่อที่ขั้วบวกของวงจรรขยายค่าความแตกต่าง มีไว้เพื่อสมดุลของความต้านทานของอินพุทของวงจรรขยายค่าความแตกต่าง



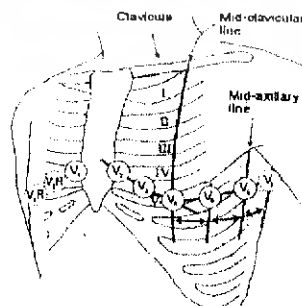
รูปที่ 2.12 วิธีการวัดแบบ Augment Limb Lead

วิธีการวัดแบบ Unipolar Chest Lead

วิธีการวัดแบบ Unipolar Chest Lead เป็นการวัดขนาดสัญญาณไฟฟ้าหัวใจระหว่างตำแหน่งใดจุดบนหน้าอก (ขั้ววัดบนอก) เทียบกับค่าเฉลี่ยของความต่างศักย์ของตำแหน่ง RA, LA และ LL วิธีการวัดในรูปที่ 2.13 วิธีนี้ประกอบด้วย 6 ลีด คือ Lead V1 ถึง V6 คือการกำหนดตำแหน่งของขั้วบนอกอยู่ในบริเวณต่างๆบริเวณหน้าอก 6 ตำแหน่ง แสดงในรูปที่ 2.14



รูปที่ 2.13 วิธีการวัดคลื่นหัวใจไฟฟ้าแบบ Unipolar Chest Lead



รูปที่ 2.14 ตำแหน่งการติดตั้งขั้ววัดบนอกวิธีการวัดแบบ Unipolar Chest Lead

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3.2 การวัดเพื่อการมอนิเตอร์

การวัดแบบมอนิเตอร์ใช้วัดสัญญาณไฟฟ้าหัวใจในขณะที่มีการเคลื่อนย้ายผู้ป่วยเพื่อตรวจสอบจังหวะและอัตราการเต้นของหัวใจ ตำแหน่งที่วัดควรเป็นตำแหน่งที่ให้สัญญาณ R แรงมาก เพื่อให้อัตราการเต้นของหัวใจต่อสัญญาณรบกวน (Signal to noise Ratio : S/N) มีค่าสูง ตำแหน่งของการวัดมอนิเตอร์แสดงได้ดังรูปที่ 2.15 โดยการติดขั้วบวกไว้ที่ตำแหน่ง V1 ของ Unipolar Chest Lead (ตำแหน่งหมายเลข 1) และติดขั้วลบไว้ใกล้ไหล่ซ้าย (ตำแหน่งหมายเลข 2) และติดขั้ววัดใช้อ้างอิงที่ตำแหน่งใดๆ บริเวณหน้าอก (ตำแหน่งหมายเลข 3) ลักษณะของคลื่นหัวใจที่วัดได้จะใกล้เคียงกับ V1 ของ Unipolar Chest Lead ซึ่งเป็นสัญญาณที่ใช้ในการคำนวณอัตราการเต้นของหัวใจ



รูปที่ 2.15 ตำแหน่งการติดขั้ววัดหัวใจของวิธีการวัดคลื่นไฟฟ้าเพื่อการมอนิเตอร์

2.2 อิเล็กโตรด

การจะวัดแรงดันไฟฟ้าและกระแสไฟฟ้าบนร่างกายจะต้องมีตัวเชื่อม นั่นก็คือ อิเล็กโตรด ซึ่งจะทำงานเสมือนเป็นทรานสดิวเซอร์ เพราะในร่างกายมีการนำกระแสด้วย ไอออน แต่ในเครื่องวัดจะนำกระแสด้วยอิเล็กตรอน ดังนั้น อิเล็กโตรดต้องทำหน้าที่เปลี่ยน Ionic current ให้เป็น Electronic current

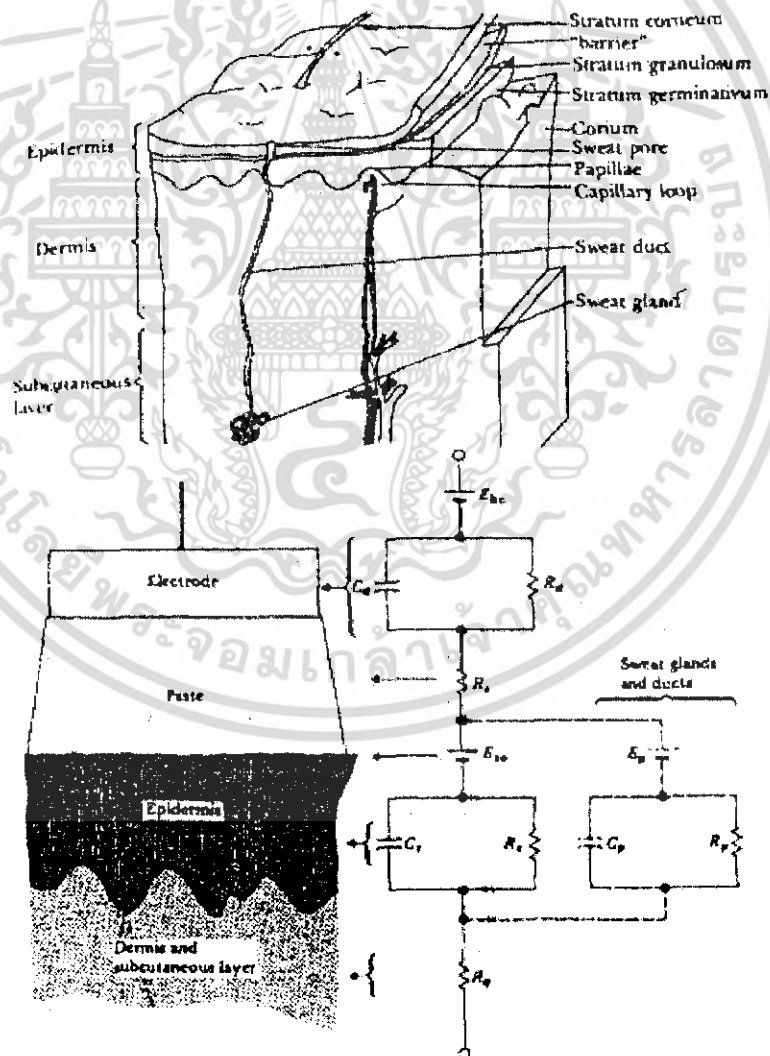
ไต่อะแกรมของพื้นหน้าระหว่างอิเล็กโตรดและอิเล็กโตรไลต์แสดงไว้ในรูปที่ 2.16 กระแสไฟฟ้าจะข้ามจากอิเล็กโตรดไปยัง อิเล็กโตรไลต์ จะประกอบด้วย

1. อิเล็กตรอนที่เคลื่อนไปในทางทิศตรงข้ามกับกระแสอิเล็กโตรด
2. แคทไอออนเคลื่อนไปในทางทิศเดียวกับกระแสไฟฟ้า
3. แอนไอออนเคลื่อนไปในทางทิศตรงข้ามกับกระแสไฟฟ้าในอิเล็กโตรไลต์

สำหรับประจุที่ข้ามพื้นหน้านั้น เนื่องจากไม่มีอิเล็กตรอนอิสระในอิเล็กโตรไลต์และไม่มีแคทไอออนและแอนไอออนอิสระในอิเล็กโตรดด้วยจึงต้องมีการเปลี่ยนแปลงเกิดขึ้น เพื่อเป็นการถ่ายทอดประจุระหว่างพาหะทั้งสอง ดังนั้นอิเล็กโตรไลต์ที่ห่อหุ้มด้วยโลหะจะมีศักย์ไฟฟ้าแตกต่างไปเรียกว่า ศักย์ไฟฟ้าครึ่งเซลล์ (Half - cell potential) แต่เราไม่สามารถวัดศักย์ไฟฟ้าครึ่งเซลล์ของอิเล็กโตรดได้ จึงต้องใช้อิเล็กโตรดอีกอันในการเปรียบเทียบแรงดันไฟฟ้า

ศักย์ไฟฟ้าครึ่งเซลล์ของอิเล็กโทรดที่กล่าวมาเป็นภาวะที่ไม่มีกระแสไฟฟ้าไหล ถ้ามีกระแสไหล แรงดันไฟฟ้าที่วัดได้จะต้องเปลี่ยนแปลงไป ความแตกต่างนี้เป็นผลมาจาก polarization ของอิเล็กโทรด ความต่างแรงดันไฟฟ้าที่เกิดขึ้นเมื่อเทียบกับสภาวะสมดุลนั้นเรียกว่า over voltage มีกลไกที่เกี่ยวข้องปรากฏการณ์นี้ 3 ส่วน คือ

1. **Ohmic Over voltage** เป็นผลมาจากความต้านทานของ อิเล็กโทรไลต์ เมื่อกระแสไฟฟ้าผ่าน อิเล็กโทรดทั้งสองอัน เมื่อความต้านทานระหว่างอิเล็กโทรดเปลี่ยนแปลงตามกระแสไฟฟ้า ศักย์ไฟฟ้าที่เกินทางต้านไอห้ม (Ohmic over voltage) ก็จะไม่มีความสัมพันธ์เป็นเส้นตรง กับกระแสไฟฟ้า ตามกฎของโอห์ม
2. **Concentration Over voltage** เกิดจากความเข้มข้นที่เป็นผลมาจากการเปลี่ยนแปลงการกระจายของไอออนในอิเล็กโทรไลต์
3. **Activation Over voltage** เป็นผลมาจากการถ่ายทอดประจุของปฏิกิริยาการเติมและการลดออกซิเจน ไม่สามารถเปลี่ยนกลับได้ทั้งหมด



รูปที่ 2.16 แสดงโครงสร้างผิวหนังและวงจรมูลยของผิวหนังที่ติดอิเล็กโทรด เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 อิเล็กโทรดที่ปลาไรซ์และอิเล็กโทรดที่ไม่ปลาไรซ์

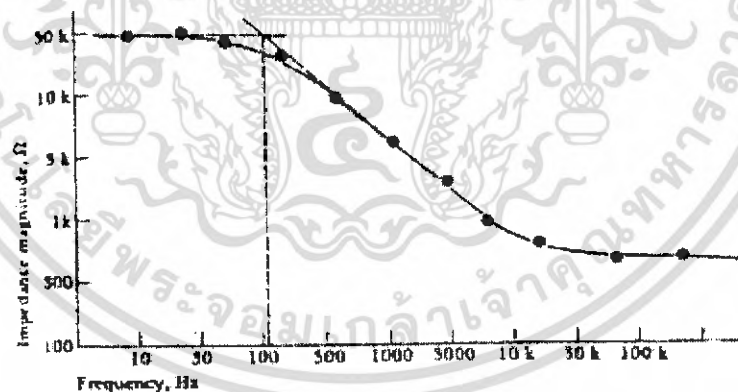
ตามทฤษฎีสามารถแบ่งอิเล็กโทรดได้เป็น 2 ชนิด

1. อิเล็กโทรดที่ปลาไรซ์ได้ อิเล็กโทรดนี้เมื่อผ่านกระแสไฟฟ้าเข้าไปจะไม่มีประจุข้ามพื้นหน้าของอิเล็กโทรดและอิเล็กโทรไลต์จะทำงานเสมือนว่าเป็นคาปาซิเตอร์
2. อิเล็กโทรดที่ไม่ปลาไรซ์ เมื่อมีกระแสผ่านจะสามารถผ่านพื้นหน้าได้อย่างเสรีโดยไม่สูญเสียพลังงาน ดังนั้นจึงไม่ทำให้เกิด Over voltage แต่เราไม่สามารถสร้างอิเล็กโทรดที่ปลาไรซ์ได้หรืออิเล็กโทรดที่ไม่ปลาไรซ์ได้อย่างสมบูรณ์

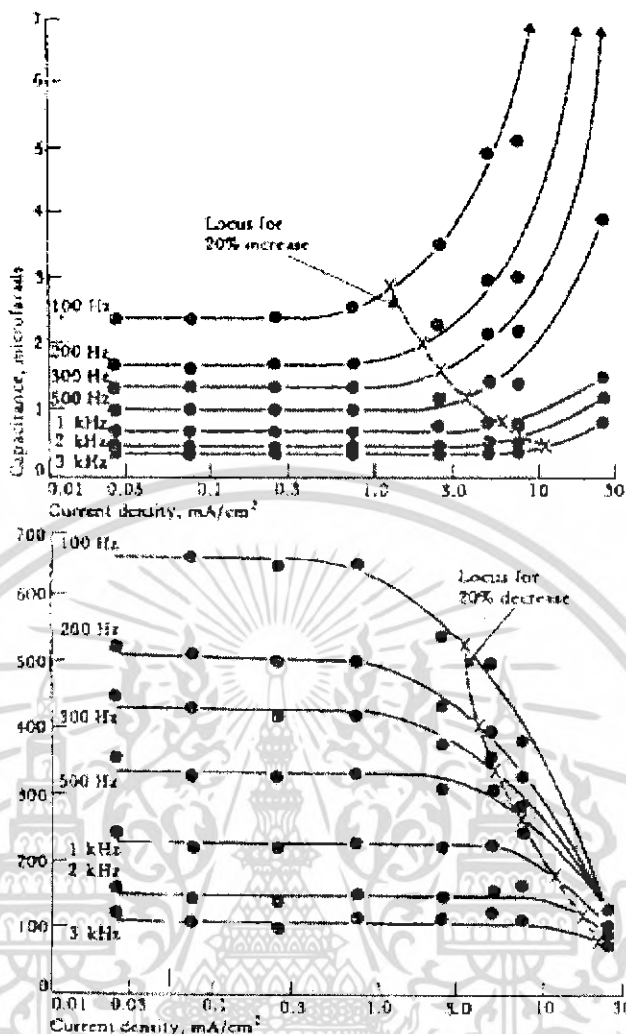
2.2.2 คุณสมบัติของอิเล็กโทรด

เราพบว่าคุณสมบัติของกระแสและแรงดันของอิเล็กโทรดนั้นไม่เป็นเส้นตรงเพราะอิเล็กโทรดมีคุณสมบัติเฉพาะคือ ความไวต่อกระแสที่ผ่านลงไป ในอิเล็กโทรด ถ้ามีความเข้มข้นกระแส มาก คุณสมบัติจะแตกต่างไปจากเดิม และคุณสมบัติของอิเล็กโทรดยังขึ้นอยู่กับรูปคลื่นไฟฟ้า ถ้าเป็นรูปไซน์ก็ยังคงขึ้นกับความถี่ด้วย เพราะอิเล็กโทรดเหมือนมีตัวต้านทานและตัวเก็บประจุอยู่ เมื่ออิเล็กโทรด มาติดผิวหนัง เราต้องพิจารณาพื้นหน้าระหว่างอิเล็กโทรด, อิเล็กโทรไลต์ และผิวหนังด้วย

เราจะใช้ครีมอิเล็กโทรไลต์ที่ประกอบด้วยเกลือไรต์ไอออน ทาก่อนจะติดอิเล็กโทรด เป็นตัวประสาน บังคับหนึ่งที่มีผลต่อคุณสมบัติทางไฟฟ้าของผิวหนังคือการวัด galvanic skin reflex (GSR) เพราะเกี่ยวกับเหงื่อและท่อของต่อมเหงื่อ ซึ่งจะมี โซเดียม โปแตสเซียมและเกลือไรต์ไอออนหลั่งออกมาจากต่อมเหงื่อ ทำให้เกิดความต่างศักย์ระหว่างรูของท่อต่อมเหงื่อกับผิวหนังแต่ส่วนประกอบที่กล่าวมาสามารถติดไปได้ กับอิเล็กโทรดที่ใช้วัดทางชีววิทยารวมค่า ที่ไม่เกี่ยวกับการวัดทางผิวหนัง



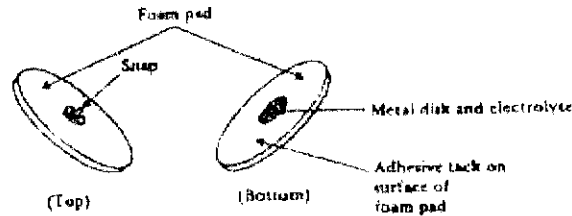
รูปที่ 2.17 ความต้านทานที่เปลี่ยนตามความถี่



รูปที่ 2.18 แสดงคุณสมบัติของอิเล็กโทรดที่เป็นเหล็กไร้สนิมซึ่งสัมพันธ์กับค่าความถี่ ประจุ และความเข้มของกระแสไฟฟ้า

2.2.3 อิเล็กโทรดแบบแผ่นที่ทำด้วยโลหะ

อิเล็กโทรดที่ใช้บ่อยในการรับศักย์ไฟฟ้าในทางชีววิทยาคืออิเล็กโทรดแผ่นที่ทำด้วยโลหะ โดยการนำแผ่นโลหะมาสัมผัสกับผิวหนัง และมักใช้ครีมอิเล็กโทรไลต์เชื่อมระหว่างกลางเพื่อทำให้มีการสัมผัสที่ดียิ่งขึ้น รูปที่ 2.19 แสดงอิเล็กโทรดแผ่นโลหะแบบต่างๆ เป็นอิเล็กโทรดที่ใช้ได้รวดเร็วและใช้แล้วทิ้ง ทั้งนี้เพื่อประหยัดเวลาและบุคลากรทางด้านนี้ อิเล็กโทรดนี้ประกอบด้วยพลาสติกที่ทำเป็นโฟม และมีแผ่นเงินติดอยู่ข้างหนึ่ง แผ่นเงินนี้อาจเคลือบด้วยซิลเวอร์คลอไรด์ ในการใช้อิเล็กโทรดนี้ ผู้ใช้เพียงแต่ทำความสะอาดผิวหนัง เปิดห่ออิเล็กโทรดดึงกระดาษที่ปิดอยู่ทิ้งไปแล้วกดอิเล็กโทรดลงบนผิวหนังทันที



รูปที่ 2.19 อิเล็กโทรดแผ่นที่ทำด้วยโฟม (ชนิดใช้แล้วทิ้งเลย) สำหรับเครื่องอี.ซี.จี. ใช้ปิดผิวหนังติดแขนขา

2.2.4 ข้อแนะนำในการใช้อิเล็กโทรดในทางปฏิบัติ

ในการใช้อิเล็กโทรดโลหะสำหรับวัดแรงดันไฟฟ้าหรือกระตุ้นก็ดี จะต้องนึกถึงข้อปฏิบัติ 5 ประการดังต่อไปนี้คือ

1. ในการสร้างอิเล็กโทรดรวมทั้งสายไฟที่นำมาต่อ โดยเฉพาะส่วนที่จะต้องสัมผัสกับเนื้อเยื่อของร่างกาย ควรเป็นวัสดุชนิดเดียวกันตลอด เมื่อใช้วัสดุอย่างใดอย่างหนึ่งในการเชื่อมก็ควรจะใช้ขนาดหุ้มไว้ ไม่ให้สัมผัสกับเนื้อเยื่อหรืออิเล็กโทรไลต์ของร่างกาย โลหะต่างชนิดกันไม่ควรนำมาใช้ให้สัมผัสกันเพราะจะมีแรงดันไฟฟ้าครึ่งเซลล์ต่างกัน นอกจากนั้นเมื่อสัมผัสกับอิเล็กโทรไลต์ก็จะทำให้เกิดปฏิกิริยาทางเคมีไฟฟ้าเกิดขึ้น เป็นผลให้มี polarization เพิ่มขึ้น และมักทำให้อิเล็กโทรดอันหนึ่งถูกกัดกร่อนไป ปัจจุบันนี้ทำให้แรงดันไฟฟ้าครึ่งเซลล์มีเสถียรภาพน้อย ทำให้เพิ่มการรบกวนทางไฟฟ้าของอิเล็กโทรดได้

2. เมื่อใช้อิเล็กโทรดคู่ใดคู่หนึ่งสำหรับวัดแรงดันไฟฟ้าร่างกาย ควรใช้อิเล็กโทรดที่ทำด้วยวัสดุอย่างเดียวกัน เนื่องจากแรงดันไฟฟ้าครึ่งเซลล์ที่เกิดขึ้นมีค่าเท่ากัน ดังนั้นแรงดันไฟฟ้า ดี ซี ที่ป้อนเข้าไปยังอินพุทของแอมพลิไฟเออร์จะได้มีค่าน้อยมาก อันเป็นการทำให้การอิมิตัวของแอมพลิไฟเออร์ไม่เกิดขึ้น โดยเฉพาะเมื่อแอมพลิไฟเออร์ที่ใช้เป็นชนิด ดี ซี และมีกำลังขยายสูง

3. อิเล็กโทรดที่ใช้ติดบนผิวหนังมักจะหลุดได้ง่าย อย่างไรก็ตาม ปัญหาจะไม่เกิดขึ้นถ้าอิเล็กโทรดได้รับการออกแบบที่ดี เส้นลวดที่ต่อออกมาจากอิเล็กโทรดควรจะอ่อนตัวได้มาก แต่ก็ต้องแข็งแรง จุดต่อของเส้นลวดที่เข้าไปยังแผ่นอิเล็กโทรดมักหลุดได้ง่าย เนื่องมาจากการโค้งงอของเส้นลวดที่มีอยู่เรื่อยๆ จะคำนึงถึงข้อนี้ด้วยเสมอในการออกแบบ

4. อิเล็กโทรดมักถูกใช้ในสภาวะแวดล้อมที่มีความชื้นสูงมาก ฉนวนของอิเล็กโทรดเหล่านี้มักทำด้วยวัสดุพวกโพลีเมอร์ ซึ่งสามารถดูดน้ำได้เมื่อใช้ไปนานๆ

5. การใช้แอมพลิไฟเออร์ที่มีอินพุทสูงมากๆ ทำให้การบันทึกไฟฟ้าได้ผลดี ถ้าอิมพีแดนซ์ของแอมพลิไฟเออร์มีค่าไม่สูงพอ นอกจากจะได้สัญญาณที่มีความสูงลดลงแล้ว ยังมีรูปร่างผิดเพี้ยนไปอีกด้วย

2.3 ความต้องการโดยเฉพาะของเครื่อง อี.ซี.จี.

คณะกรรมการการตรวจ อี.ซี.จี. ของสมาคมโรคหัวใจแห่ง U.S.A. ได้ให้ข้อเสนอแนะสำหรับเครื่อง อี.ซี.จี. มาตรฐานที่ใช้เขียนโดยตรงบนกระดาษ Pipberger ค.ศ 1975 ข้อเสนอแนะดังกล่าวมีดังต่อไปนี้

1. Linearity and distortion ความแม่นยำและความผิดเพี้ยนเป็นคุณสมบัติที่สำคัญของเครื่อง อี.ซี.จี. การเบี่ยงเบนไปจากความต้องการมีค่าน้อยกว่า 5 % เมื่อทำการบันทึกบนกระดาษด้วยความสูง 5 มม. และ 50 มม. ความต้องการนี้ครอบคลุมคลื่นไฟฟ้าที่ความถี่ระหว่าง 0.05-100 เฮิรท์

2. Input Range เครื่อง อี.ซี.จี. ต้องสามารถขยายสัญญาณได้ในช่วงกว้าง คือ ช่วงที่มีความสูงได้ถึง 10 มิลลิโวลท์

3. Input Impedance and current อินพุทอิมพีแดนซ์ระหว่างอิเล็กโตรดกับพื้นดิน ควรมีค่าน้อยกว่า 5 เมกกะโห์ม ในระหว่างการวัดนั้น อิเล็กโตรดทุกอันควรจะทำการต่อลงดินให้หมด เครื่อง อี.ซี.จี. ไม่ควรให้มีกระแสไฟฟ้ามากกว่า 10 ไมโครแอมแปร์ไหลผ่านผู้ป่วย

4. Central Terminal วงจรรีซิสเตนซ์ที่ต้องการเพื่อทำให้เกิด Central Terminal ไม่ควรจะทำให้มีการผิดเพี้ยนของสัญญาณเพิ่มขึ้นจากที่ได้กล่าวในข้อที่ 1 หรือเพิ่มขึ้น 2% เมื่อรวมกับความต้องการทางด้านอินพุทอิมพีแดนซ์แล้ว ค่ารีซิสเตนซ์ที่น้อยที่สุดควรมีค่า 3.3 เมกกะโห์ม

5. Gain เครื่อง อี.ซี.จี. ควรปรับกำลังขยายได้ 3 ค่า คือ 5, 10 และ 20 มม./มิลลิโวลท์

6. Frequency Response การตอบสนองของเครื่อง อี.ซี.จี. ในช่วงความถี่ตั้งแต่ 0.14 ถึง 25 เฮิรท์ ควรเท่ากันจะดีไม่ควรต่างกันเกิน 0.5 dB สำหรับสัญญาณที่มีความสูงน้อยกว่าหรือเท่ากับ 5 มม. บนกระดาษบันทึกเมื่อใช้ความถี่ 25 เฮิรท์ แล้วการตอบสนองต่อสัญญาณรูปไซน์จะลดลงมากกว่า 3 dB

7. Common Mode Rejection Ratio (CMRR) ในทุกตำแหน่งของสวิทช์ที่ใช้เลือกสายต่อเมื่อปรับตั้งกำลังขยายเครื่องบันทึกไว้ที่ 10 มม./มิลลิโวลท์ และสายต่ออิเล็กโตรดทั้งหมดเข้ากับจุดรวมของสายอิเล็กโตรดทั้งหมดจะนำไปต่ออนุกรมกับคาปาซิเตนซ์ 22 พิโคฟารัด ผลที่ได้นั้นไม่ควรจะมีการเคลื่อนที่ของเข็มมากกว่า 20 มม. ข้อกำหนดเฉพาะนี้ยังคงใช้ได้เมื่อต่อรีซิสเตนซ์ 100 กิโลโห์ม แบบอนุกรมกับสายอินพุท

8. Calibration เครื่อง อี.ซี.จี. ควรจะมีแรงดันไฟฟ้ามากกว่า 1.0 มิลลิโวลท์ สำหรับปรับเปรียบเทียบค่ากำลังขยายของเครื่อง

9. Chart Speed คือความเร็วของการกระดาษบันทึกมาตรฐานควรเป็น 25 มม./วินาที นอกจากนั้นควรมีความเร็วสูงกว่าด้วย คือ ความเร็ว 50 มม./วินาที ความแม่นยำของความเร็วควรมีค่า 2%

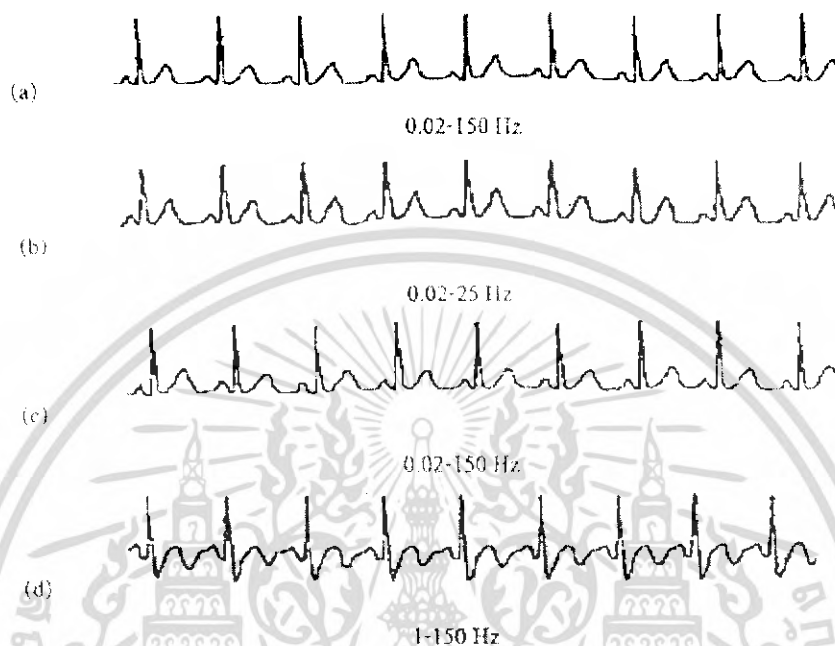
10. Output เอาท์พุทอิมพีแดนซ์ควรมีค่าน้อยกว่า 100 โห์ม เอาท์พุทสูงสุดควรมีค่า 1 โวลท์

11. Event Marker อุปกรณ์ที่ทำเครื่องหมายด้วยมือควรมีไว้ในเครื่อง อี.ซี.จี. สำหรับให้ผู้ใช้เครื่องทำเครื่องหมายเมื่อทำการบันทึกคลื่นไฟฟ้า อี.ซี.จี.

2.3.1 ปัญหาที่พบบ่อยในการออกแบบและใช้เครื่อง อี.ซี.จี.

2.3.1.1 ความผิดเพี้ยนทางด้านความถี่

เมื่อเครื่อง อี.ซี.จี. มีการตอบสนองความถี่ที่เปลี่ยนแปลงไป คือ ไม่ได้มาตรฐานที่กล่าวไว้ข้างต้น จะมีความผิดเพี้ยนของคลื่น อี.ซี.จี. ดังแสดงให้เห็นในรูปที่ 2.20



รูปที่ 2.20a แสดงคลื่น อี.ซี.จี. ปกติ จากเครื่องมือซึ่งมีการตอบสนองความถี่กว้างกว่า 0.05-100 เฮิรท์

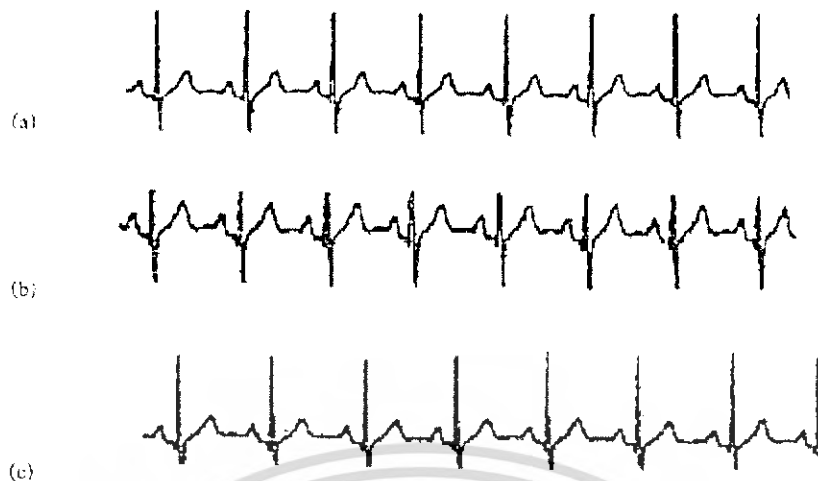
รูปที่ 2.20b แสดงคลื่น อี.ซี.จี. ที่บันทึกได้ด้วยเครื่องมือที่มีการตอบสนองความถี่จาก 0.02-25 เฮิรท์ ความผิดปรกตินี้เรียกว่า High-Frequency Distortion รูปคลื่นแต่เดิมจะเป็นนุ่มแหลม เมื่อความผิดเพี้ยนเกิดขึ้นจะมีขนาดกลมและความสูงยังคงด้วย

รูปที่ 2.20c แสดงคลื่น อี.ซี.จี. จากต้นตอเดียวกัน แต่บันทึกด้วยเครื่องมือที่มีการตอบสนองความถี่ 1- 150 เฮิรท์ สังเกตความผิดเพี้ยนที่พื้นฐานเครื่อง อี.ซี.จี. เส้นพื้นฐานจะไม่อยู่ในแนวราบ สังเกตว่าคลื่นซึ่งแต่เดิมมียอดเดียวจะกลายเป็น 2 ยอด ความผิดปรกติเช่นนี้จะเรียกว่า Low-Frequency Distortion

รูปที่ 2.20 แสดงความผิดพลาดของรูปคลื่น อี.ซี.จี. ซึ่งมีผลมาจากความผิดเพี้ยนของความถี่

2.3.1.2 การอ้อมตัวหรือความผิดเพี้ยนที่ตัดการทำงาน (Saturation or Cut off Distortion)

Offset Voltage ที่อิเล็กทรอนิกส์หรือการปรับแอมพลิไฟเออร์ไม่ถูกต้องในเครื่อง อี.ซี.จี. อาจทำให้เกิดการอ้อมตัวหรือความผิดเพี้ยนที่ตัดการทำงาน ซึ่งทำให้รูปร่างของคลื่น อี.ซี.จี. เปลี่ยนไปได้มาก รูปที่ 2.21a แสดงคลื่น อี.ซี.จี. ปกติ ส่วนรูปที่ 2.21b แสดงคลื่นไฟฟ้าที่ผิดเพี้ยนไปจากการเปลี่ยนระดับไปสู่การอ้อมตัว ทำให้ยอดคลื่น อี.ซี.จี. ถูกตัดออกไป เนื่องจากแอมพลิไฟเออร์ไม่สามารถมีเกาท์พุทเกินแรงดันไฟฟ้าที่อ้อมตัวได้



รูปที่ 2.21c แสดงการเปลี่ยนแปลงทำนองเดียวกัน เมื่อส่วนล่างของคลื่น อี.ซี.จี. ถูกตัดออกไป ทั้งนี้เป็นผล จากการอิมิตัวทางด้านลบของแอมพลิไฟเออร์ สังเกตว่าในกรณีเส้นพื้นฐานจะเรียกดี ขอดคลื่น P และ T อาจยังคงเห็นได้จากภาพบันทึกอาจต่ำกว่าระดับที่ถูกตัด ถ้าเป็นเช่นนี้ จะเห็นแต่คลื่น R อย่างเดียว

รูปที่ 2.21 แสดงความผิดพลาดของคลื่น อี.ซี.จี. ซึ่งมีผลจากการผิดเพี้ยนที่เกิดจากแอมพลิไฟเออร์

2.3.1.3 วงจรสายดิน (Ground Loops)

ผู้ป่วยที่กำลังถูกบันทึก อี.ซี.จี. เช่นในมอนิเตอร์การทำงานของหัวใจเป็นต้น ย่อมได้รับการต่ออุปกรณ์ไฟฟ้าอย่างอื่นด้วยอุปกรณ์ไฟฟ้าแต่ละอย่างจะมีการต่อสายดินของตนเอง อาจจะโดยอาศัยร่วมไปกับสายไฟ หรือในบางกรณีนั้นจะต่อโดยผ่านไปกับสายดินใหญ่ไปยังจุดสายดินในห้อง

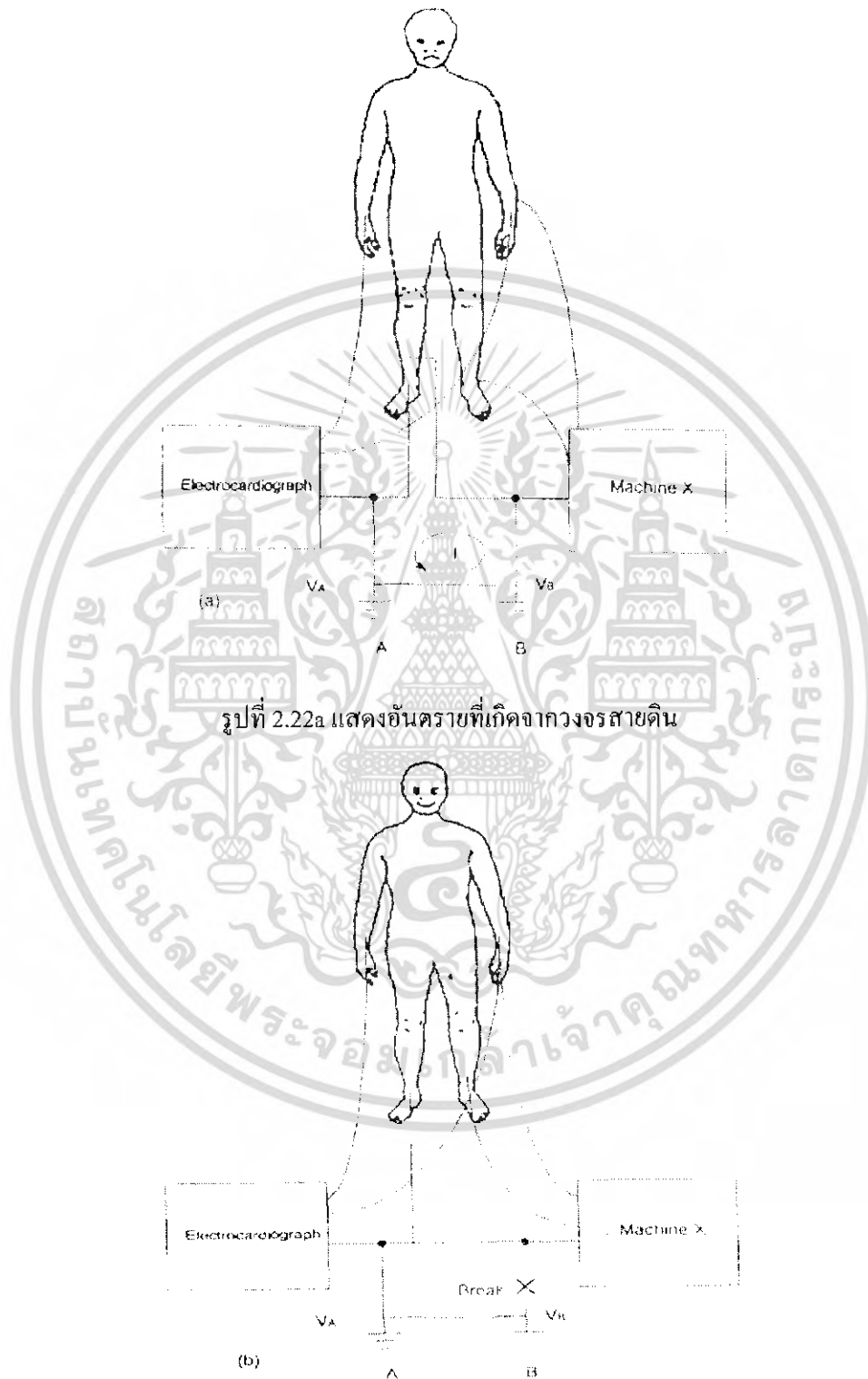
วงจรสายดินอาจเกิดขึ้นในภาวะที่แสดงในรูปที่ 2.22a ในรูปตัวนี้ผู้ป่วยได้ต่อกับกับเครื่อง 2 เครื่อง คือ เครื่อง อี.ซี.จี. และเครื่องอื่น แต่ละเครื่องมีสายดินต่อกับตัวของผู้ป่วย เครื่อง อี.ซี.จี. ถูกต่อลงดิน โดยผ่านทางปลั๊กของสายไฟที่เรียกว่าสายดิน A ส่วนเครื่องอีกเครื่องหนึ่งนั้นก็มีสายดิน โดยผ่านทางสายไฟเช่นเดียวกัน แต่มีปลั๊กอยู่อีกที่อีกแห่งหนึ่งซึ่งมีสายดินคนละเส้นเรียกว่า เป็นสายดิน B ถ้าสายดิน B มีศักย์สูงกว่า สายดิน A จะมีกระแสไฟฟ้าไหลจากสายดิน B ผ่านไปยัง ตัวเครื่องและผู้ป่วย และผ่านอิเล็กโตรดของเครื่อง อี.ซี.จี. ไปลงที่จุด A กระแสไฟฟ้าดังกล่าวนอกจากจะมีปัญหาทางด้านความปลอดภัยแล้วยังทำให้แรงดันไฟฟ้าในตัวผู้ป่วยสูงกว่าของสายดิน ซึ่งต่อกับกับตัวเครื่องตัวอย่างแสดงให้เห็นเป็นดังรูปที่ 2.22a ผู้ป่วยจะมีแรงดันไฟฟ้าอยู่ระหว่างสายดิน B และ สายดิน A จึงทำให้เกิด Common Mode Voltage บนเครื่อง อี.ซี.จี. ซึ่งจะทำให้ CMRR ต่ำ ดังนั้น จึงเป็นการเพิ่มปริมาณของสัญญาณรบกวน

ทางเดินของกระแสไฟฟ้าระหว่างสายดินทั้งสองดังแสดงในรูปที่ 2.24a นั้นเรียกว่า Ground Loop ปรากฏการณ์เช่นนี้ควรพยายามหลีกเลี่ยงไม่ให้เกิดขึ้นในระบบของอุปกรณ์ทางการแพทย์

รูปที่ 2.22b เป็นภาวะที่ควรระวังจะทำให้เกิดขึ้นในรูปนี้เครื่องทั้งสองมีสายดินที่ต่อมายังจุดเดียวกันจึงไม่มีวงจรสายดินเกิดขึ้นแรงดันไฟฟ้าสายดินของเครื่อง อี.ซี.จี. และอีกเครื่องหนึ่งควรจะอยู่ในระดับเดียวกัน ในกรณีเช่นนี้จึงไม่ควรมีแรงดันไฟฟ้าตกคร่อมสายดินอันจะทำให้เครื่องมีแรงดันไฟฟ้าสูงกว่า จุดสายดิน ถึงแม้ว่าผู้ป่วยจะมีสายดินที่ต่อกับกับเครื่องเพียงเครื่องเดียว แต่ก็ไม่มีกระแสไฟฟ้าใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรสายดินไหลผ่านตัวของผู้ป่วยตามปกติแล้วเครื่องทั้งสองจะมีแรงดันไฟฟ้าของสายดินเท่ากัน ดังนั้นจึงไม่มีกระแสไฟฟ้าในวงจรสายดินไหนผ่านตัวของผู้ป่วย ถึงแม้ว่าผู้ป่วยจะไปสัมผัสกับสายดินของเครื่องก็ตาม



รูปที่ 2.22b แสดงอันตรายที่เกิดจากรอยต่อที่ขาด

รูปที่ 2.22 แสดงผลของการต่อสายดินของเครื่อง อี.ซี.จี.กับอุปกรณ์ไฟฟ้าอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.4 สายต่อที่ขาด (Open Lead Wire)

มีหลายครั้งที่สายต่อที่อิเล็กทรอนิกส์ของเครื่อง อี.ซี.จี. เส้นใดเส้นหนึ่งขาดไป แรงดันไฟฟ้าที่สูงสามารถเหนี่ยวนำในสายต่อที่ขาดนี้ได้เป็นผลทำให้ปากที่ใช้บันทึกมีการเคลื่อนไหวไปตามคลื่นความถี่ของคลื่นไฟฟ้าเหนี่ยวนำนั้น ภาวะเช่นนี้อาจเกิดขึ้นได้แม้สายไม่ขาด เนื่องจากอิเล็กทรอนิกส์ไม่ดีกับตัวผู้ป่วย

2.3.1.5 สิ่งที่รบกวนเกิดจากศักย์แรงสูงระยะสั้น

ในบางภาวะที่กำลังตรวจวัด อี.ซี.จี. จะต้องทำการดีฟิบริเลชัน ในกรณีเช่นนี้จะต้องป้อนคลื่นแรงดันไฟฟ้าสูง และกระแสจำนวนมากผ่านตัวผู้ป่วย ดังนั้นจึงเกิดแรงดันไฟฟ้าแรงสูงคร่อมอิเล็กทรอนิกส์ของเครื่อง อี.ซี.จี. แรงดันไฟฟ้าเหล่านี้มีค่าสูงกว่าสัญญาณ อี.ซี.จี. มาก แรงดันไฟฟ้านี้จะทำให้เข็มของเครื่อง อี.ซี.จี. เปลี่ยนระดับไปมาก ดังแสดงในรูปที่ 2.23 ทั้งนี้เนื่องจากมีการอ้อมตัวของแอมพลิไฟเออร์ ในเครื่อง อี.ซี.จี. คลื่นไฟฟ้านี้มีจำนวนมากพอที่จะทำให้มีการเก็บประจุไฟฟ้าของ คาปาซิเตอร์ที่อยู่ในแอมพลิไฟเออร์ แล้วหลังจากนั้นจึงค่อยๆ สดกลับเข้าสู่พื้นฐานใหม่ แต่ใช้เวลานาน ซึ่งต้องขึ้นอยู่กับความถี่ของแอมพลิไฟเออร์ ต้นตออีกอย่างของสิ่งรบกวนในทำนองเดียวกันนี้คือ การเคลื่อนไหวของอิเล็กทรอนิกส์ ซึ่งทำให้มีการเปลี่ยนแปลงของแรงดันไฟฟ้ามากกว่าแรงดันไฟฟ้าจากเครื่อง อี.ซี.จี



รูปที่ 2.23 ผลของสิ่งรบกวนที่เกิดจากแรงดันไฟฟ้าแรงสูงระยะสั้น

- การเปลี่ยนแปลงในระยะสั้น
- กลับคืนสู่สภาพปกติ
- สภาพการเปลี่ยนแปลงเช่นเดียวกัน แต่ลดกำลังขยายของ first-order Recover ของระบบ

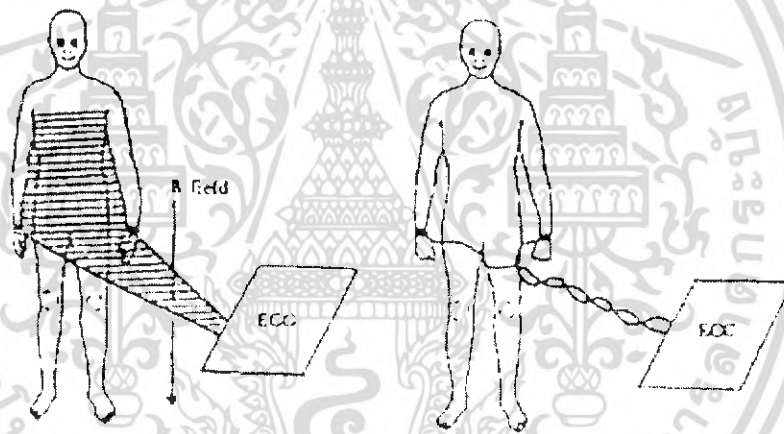
2.3.1.6 การรบกวนจากเครื่องไฟฟ้า

ต้นตอสำคัญของการรบกวนเมื่อทำการบันทึกหรือมอนิเตอร์คลื่นไฟฟ้า อี.ซี.จี. คือ การรบกวนที่เกิดจากระบบไฟฟ้ากำลังระบบไฟฟ้านั้นนอกจากจะทำหน้าที่จ่ายพลังงานไฟฟ้าให้กับเครื่อง อี.ซี.จี. แล้วยังต้องจ่ายไฟฟ้าให้กับอุปกรณ์ภายในห้องเดียวกันอีกด้วย การรบกวนดังกล่าวอาจเกิดขึ้นด้วยกลไกที่แตกต่างกันอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สนามไฟฟ้าที่ควมระหว่างเครื่องจ่ายไฟกับเครื่อง อี.ซี.จี. หรือตัวผู้ป่วยเป็นผลจากสนามไฟฟ้ารอบๆสายไฟหรือรอบๆเครื่องใช้ไฟฟ้า สนามไฟฟ้าเหล่านี้อาจเกิดขึ้นได้แม้แต่ยังไม่ได้ปล่อยพลังงานไฟฟ้าเข้าสู่เครื่อง การรบกวนจึงเป็นเสมือนหนึ่งมีคาปาซิเตอร์ต่ออยู่กับสายไฟที่จ่ายพลังงานไฟฟ้าต้นตออย่างอื่นของการรบกวนที่เกิดจากสายไฟจ่ายพลังงานคือ การเหนี่ยวนำแม่เหล็กกระแสไฟฟ้าที่ไหลผ่านจะทำให้เกิดการเหนี่ยวนำสนามไฟฟ้ารอบสายตัวนำสนามแม่เหล็ก อาจเกิดจาก ทรานส์ฟอร์เมอร์ หรือ บาลาสของหลอดไฟฟลูออเรสเซนต์ ถ้าสนามแม่เหล็กผ่านเข้าไปในขดลวดที่สร้างขึ้นโดยเครื่อง อี.ซี.จี. สายต่อและตัวผู้ป่วย ดังแสดงไว้ในรูปที่ 2.24 แรงดันไฟฟ้าจะเกิดไฟฟ้าเหนี่ยวนำในวงจรนี้ แรงดันไฟฟ้านี้จะได้สัดส่วนกับความเข้มของสนามแม่เหล็กและพื้นที่ของขดลวดรอบเดียวที่เกิดขึ้นด้วย สามารถลดการรบกวนนี้ได้โดย

1. การลดสนามแม่เหล็กโดยการหุ้มกำบัง
2. การหลีกเลี่ยงสนามแม่เหล็กโดยการวางเครื่องอี.ซี.จี.และสายต่อให้ห่างจากสนามแม่เหล็ก
3. การลดพื้นที่ของขดลวดรอบเดียว โดยการบิดสายต่อให้พันเป็นเกลียว



รูปที่ 2.24 การลดสัญญาณรบกวนโดยการพันสายเป็นเกลียว

2.3.1.7 การรบกวนทางไฟฟ้าจากต้นตออื่นๆ

การรบกวนทางไฟฟ้าจากต้นตออื่นสามารถมีผลต่อการบันทึกคลื่นไฟฟ้า อี.ซี.จี. ได้การรบกวนจากแม่เหล็กไฟฟ้าซึ่งมาจากสถานีวิทยุที่อยู่ใกล้หรือสถานีโทรทัศน์หรือสถานีเรดาร์เครื่อง อี.ซี.จี. สามารถรับคลื่นแม่เหล็กไฟฟ้าเหล่านี้ได้และ เเรดิไฟเออร์โดยเครื่อง บางทีคลื่นไฟฟ้าอาจเปลี่ยนแปลงได้โดยพื้นที่หน้าตัดระหว่างอิเล็กโตรดและอิเล็กโตรไลต์สายต่อและตัวผู้ป่วยทำหน้าที่เป็นเสาอากาศ

การรบกวนทางแม่เหล็กไฟฟ้าอาจเกิดขึ้นโดยเครื่องกำเนิดไฟฟ้าที่มีความถี่สูงซึ่งอยู่ในโรงพยาบาลเอง เครื่องตัวจีด้วยไฟฟ้าและเครื่องไดอะเรอมีซ์มักเป็นต้นตอที่สำคัญ การแผ่รังสีของแม่เหล็กไฟฟ้าอาจเกิดจากเครื่องเอ็กซ์เรย์หรือสวิตซ์หรือรีเลย์ของเครื่องใช้ไฟฟ้าขนาดใหญ่ของโรงพยาบาล แม้แต่การกระพริบของหลอดไฟฟลูออเรสเซนต์ ก็ยังสามารถทำให้เกิดการรบกวนได้เช่นกัน

การรบกวนทางแม่เหล็กไฟฟ้าสามารถทำให้ลดน้อยลง โดยการต่อคาปาซิเตอร์ขนาด เล็กให้ขนานกับเครื่อง อี.ซี.จี. ซึ่งรีแอคแตนซ์ของเครื่องนี้มีค่าสูงมาก ที่ต่อต้านคลื่นความถี่ในช่วงของ คลื่น อี.ซี.จี. ดังนั้นจึงไม่ทำให้อินพุทอิมพีแดนซ์ของเครื่อง อี.ซี.จี. ลดลง อย่างไรก็ตาม ความถี่ของ คลื่นวิทยุนี้ค่าของรีแอคแตนซ์มีค่าต่ำพอที่จะทำให้คลื่นรบกวนทางแม่เหล็กไฟฟ้าลัดวงจร ไปโดยไม่เข้าไปทางอินพุทของแอมพลิไฟเออร์

ยังมีการรบกวนทางไฟฟ้าซึ่งมีต้นตอทางร่างกายเอง การรบกวนนี้เกิดจากกล้ามเนื้อลาย ที่อยู่ระหว่างอิเล็กโตรด เมื่อใดก็ตามที่กล้ามเนื้อเหล่านี้หดตัวจะทำให้เกิดคลื่นไฟฟ้า อี เอ็ม จี และป้อนเข้าไปทำการรบกวนในเครื่อง อี.ซี.จี. ได้

2.3.2 การป้องกันคลื่นไฟฟ้าช่วงสั้นที่มารบกวน (Transient Protection)

วงจรที่ใช้แยก ซึ่งได้ออกแบบโดยทั่วไปนั้นจะใช้ป้องกันผู้ป่วยจากอันตราย ที่เกิดจากไฟฟ้า ชุด ที่เกิดขึ้นระหว่างตัวผู้ป่วยกับตัวเครื่องและเครื่องใช้ไฟฟ้าที่อยู่ใกล้เคียง ในบางกรณีจากต้นตออื่นที่ ผ่านตัวผู้ป่วยเข้าไปในเครื่อง อี.ซี.จี. อาจทำให้เครื่องได้รับอันตรายจนเสียหายได้ เช่น เครื่องตัดจี้ด้วย ไฟฟ้าที่ใช้ในห้องผ่าตัด ในขณะที่ใช้เครื่องตัดจี้ด้วยไฟฟ้านั้น ถ้าสายดินที่ต่อกับตัวเครื่องหลุดหรือขาดไป จะมีแรงดันไฟฟ้าในตัวผู้ป่วยเมื่อเทียบเท่าสายดินสูงมาก แรงดันไฟฟ้าที่สูงนี้จะเข้าไปในเครื่อง อี.ซี.จี. หรือ เครื่องมอนิเตอร์หัวใจ ถ้ามีจำนวนมากพอ ก็สามารถทำให้เกิดอันตรายต่อวงจรอิเล็กทรอนิกส์ได้ นอกจากนี้ยังทำให้เกิดคลื่นไฟฟ้าที่ไม่ต้องการในภาพบันทึก ดังรูปที่ 2.23

ตามอุดมคติแล้วเครื่อง อี.ซี.จี. ควรจะได้รับการออกแบบเพื่อป้องกันอันตรายดังกล่าวแต่อย่างไร ก็คืออาจไม่มีระบบป้องกันหรือป้องกันได้ไม่หมด วงจรที่ใช้ป้องกัน จะมีอุปกรณ์จำกัดแรงดันไฟฟ้าสอง ปลายที่ต่ออยู่ระหว่างอิเล็กโตรดแต่ละอันของผู้ป่วยกับสายดิน

การต่อไดโอดสองอันให้กลับขั้วกันเมื่อแรงดันไฟฟ้าที่ถึงระดับ 700 มิลลิโวลต์ ไดโอดตัวใดตัว หนึ่งจะทำงานจึงช่วยจำกัดแรงดันไฟฟ้าให้อยู่ที่ระดับ 700 มิลลิโวลต์ ได้ อย่างไรก็ตาม การจำกัดไม่ได้ เกิดขึ้นทันทีทันใด ที่ระดับ 700 มิลลิโวลต์ แต่จะค่อยๆเกิดขึ้นที่ระดับ 300 มิลลิโวลต์ ดังนั้น สัญญาณที่ถูก บันทึกจึงอาจมีความผิดเพี้ยนได้

2.3.3 การลด Common Mode และการรบกวนอย่างอื่น

ดังที่ได้กล่าวในตอนต้นแล้วว่า Common Mode Voltage สามารถทำให้เกิดการรบกวนใน แอมพลิไฟเออร์ ซึ่งใช้ในทางชีววิทยาได้มาก ถึงแม้ว่า แอมพลิไฟเออร์ที่มี High Common Mode Rejection Ratio ที่ สามารถทำการลดผลที่เกิดขึ้นจาก Common Mode Voltage ได้ วิธีที่ดีกว่าในการแก้ปัญหา คือ การกำจัดต้นตอของแรงดันไฟฟ้ารบกวนนั้น ที่จะกล่าวถึงต่อไปนี้ จะกล่าวถึงต้นตอของสิ่งรบกวนอื่นๆ พร้อมทั้งการค้นหาก็จะทำให้ลดน้อยลง

2.3.3.1 การรบกวนจากสนามไฟฟ้าและสนามแม่เหล็ก

เราสามารถลดการรบกวนเหล่านี้ โดยการกำจัดต้นตอด้วยวิธีหุ้มกำบัง (Shielding) การ หุ้มกำบังไฟฟ้าสถิตนั้น โดยการต่อวัสดุที่นำไฟฟ้าได้ กันอยู่ระหว่างสนามไฟฟ้าและระบบที่จะทำการ ตรวจวัดและต่อลงดินด้วย โรงพยาบาลส่วนมากจะมีห้องที่ได้หุ้มกำบังไว้สำหรับการตรวจ อี.ซี.จี. โดยการ ฝังวัสดุที่นำไฟฟ้าได้ เช่น มุ้งลวดในฝาผนังของห้องดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหุ้มกำบังดังกล่าวจะไม่ได้ผลในการป้องกันสนามแม่เหล็ก นอกจากนี้ โลหะที่ใช้ นั้นจะเป็นชนิด high permeability เช่น การใช้แผ่นเหล็กไร้สนิม เป็นต้น หรืออาจกล่าวอีกนัยหนึ่งได้ว่า แผ่นกั้นนั้นต้องเป็นทั้งตัวนำแม่เหล็กและตัวนำไฟฟ้าที่ดีด้วย การลดพื้นที่ระหว่างดิฟเฟอเรนเชียลอินพุท กับแอมพลิไฟเออร์ ในกรณีที่เป็นสัญญาณดิฟเฟอเรนเชียลและระหว่างอินพุทสายดิน ในกรณีที่เป็น สัญญาณ Common Mode กระทำได้ง่าย โดยการบิดสายต่อให้เป็นเกลียวก็สามารถลดการรบกวนได้มาก

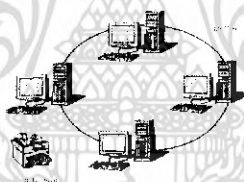
2.3.3.2 การกำจัดวงจรสายดิน (Elimination of Ground Loop)

วงจรสายดินทำให้เกิด Common Mode Voltage จากการไหลของกระแสไฟฟ้าของวงจร ที่ไหลลงดิน ในการป้องกันการรบกวนนี้จะต้อง ออกแบบระบบสายดิน ให้มีทางเดินทางเดียวระหว่าง อุปกรณ์แต่ละอย่างกับดิน วงจรสายดินควรมีรูปร่างคล้ายดาว ที่มีจุดลงดินที่กึ่งกลาง และรัศมีกระจายออกไปสู่อุปกรณ์ต่างๆ

2.4 ทฤษฎีเกี่ยวกับระบบเครือข่ายคอมพิวเตอร์

2.4.1 ความหมายและความสำคัญของ ระบบเครือข่ายคอมพิวเตอร์

ระบบเครือข่ายคอมพิวเตอร์ (Computer Network) หมายถึง การเชื่อมต่อคอมพิวเตอร์ตั้งแต่ 2 เครื่องขึ้นไปเข้าด้วยกันด้วยสายเคเบิลหรือสื่ออื่นๆทำให้คอมพิวเตอร์สามารถรับส่งข้อมูลแก่กันและกันได้



รูปที่ 2.25 ระบบเครือข่ายคอมพิวเตอร์

ในกรณีที่เป็นการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์หลายๆ เครื่องเข้ากับเครื่องคอมพิวเตอร์ขนาดใหญ่ที่เป็นศูนย์กลาง เราเรียกคอมพิวเตอร์ที่เป็นศูนย์กลางนี้ว่า โฮสต์ (Host) และเรียกคอมพิวเตอร์ขนาดเล็กที่เข้ามาเชื่อมต่อว่า ไคลเอนต์ (Client) ระบบเครือข่าย (Network) จะเชื่อมโยงคอมพิวเตอร์เข้าด้วยกันเพื่อการติดต่อสื่อสาร เราสามารถส่งข้อมูลภายในอาคาร หรือข้ามระหว่างเมืองไปจนถึงอีกซีกหนึ่งของโลก ซึ่งข้อมูลต่างๆ อาจเป็นทั้งข้อความ รูปภาพ เสียง ก่อให้เกิดความสะดวก รวดเร็วแก่ผู้ใช้ ซึ่งความสามารถเหล่านี้ทำให้เครือข่ายคอมพิวเตอร์มีความสำคัญ และจำเป็นต่อการใช้งานในแวดวงต่างๆ

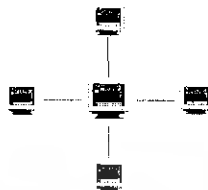
2.4.2 ลักษณะการเชื่อมต่อ ของระบบเครือข่ายคอมพิวเตอร์

จุดปลายทางของการรับ-ส่งข้อมูล เราเรียกว่า โหนด(Node) ซึ่งโหนดนี้อาจเป็น คอมพิวเตอร์ เครื่องพิมพ์ ATM หรือเครื่องรับโทรศัพท์ ซึ่งแล้วแต่วัตถุประสงค์ของการใช้งาน ซึ่งการที่จะทำให้แต่ละ โหนด ติดต่อรับ-ส่งข้อมูลถึงกันได้ นั้น ต้องมีการเชื่อมต่อที่เป็นระบบ ในระบบเครือข่ายคอมพิวเตอร์นี้ เราสามารถแบ่งลักษณะของการเชื่อมโยงออกเป็น 3 ลักษณะ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เครือข่ายแบบดาว (Star Network)

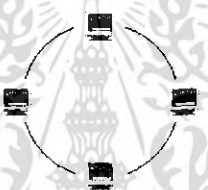
เครือข่ายแบบนี้จะมีคอมพิวเตอร์คอมพิวเตอร์หลักที่เป็น โฮสต์ (Host) ต่อสายสื่อสารกับคอมพิวเตอร์ย่อยที่เป็นไคลเอนต์ (Client) คอมพิวเตอร์ที่เป็นไคลเอนต์แต่ละเครื่องไม่สามารถติดต่อกันได้โดยตรง การติดต่อจะต้องผ่านคอมพิวเตอร์โฮสต์ที่เป็นศูนย์กลาง



รูปที่ 2.26 เครือข่ายแบบดาว

2. เครือข่ายแบบวงแหวน (Ring Network)

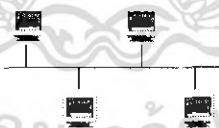
เครือข่ายแบบนี้จะมีการติดต่อสื่อสารเป็นแบบวงแหวนโดยที่ไม่มีคอมพิวเตอร์หลัก คอมพิวเตอร์แต่ละเครื่องในเครือข่ายสามารถติดต่อกันได้โดยตรง



รูปที่ 2.27 เครือข่ายแบบวงแหวน

3. เครือข่ายแบบบัส (Bus Network)

เครือข่ายแบบนี้จะมีการเชื่อมต่อคอมพิวเตอร์บนสายเคเบิล ซึ่งเรียกว่าบัส คอมพิวเตอร์เครื่องหนึ่งๆ สามารถส่งถ่ายข้อมูลได้เป็นอิสระ โดยข้อมูลจะวิ่งผ่านอุปกรณ์ต่างๆ บนสายเคเบิลจนกว่าจะถึงจุดที่ระบุไว้ (Address)



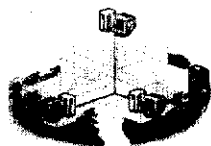
รูปที่ 2.28 เครือข่ายแบบบัส

2.4.3 ประเภทของระบบเครือข่ายคอมพิวเตอร์

ประเภทของระบบเครือข่ายคอมพิวเตอร์แบ่งออกเป็น 3 ประเภท คือ

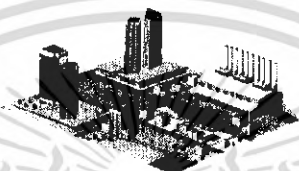
1. เครือข่ายท้องถิ่น (Local Area Network หรือ LAN) เป็นเครือข่ายระยะใกล้ ใช้กันอยู่ในบริเวณไม่กว้างนัก อาจอยู่ในองค์กรเดียวกัน หรืออาคารที่ใกล้กัน เช่น ภายในสำนักงาน ภายในโรงเรียน หรือมหาวิทยาลัย ระบบเครือข่ายท้องถิ่นจะช่วยให้ติดต่อกันได้สะดวก ช่วยลดต้นทุน และเพิ่มประสิทธิภาพในการใช้งานอุปกรณ์ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.29 เครือข่ายท้องถิ่น (Local Area Network หรือ LAN)

2. เครือข่ายระดับเมือง (Metropolitan Area Network หรือ MAN) เป็นเครือข่ายขนาดกลาง ใช้ภายในเมือง หรือจังหวัดที่ใกล้เคียงกัน เช่น ระบบเคเบิลทีวีที่มีสมาชิกตามบ้านทั่วไปที่เราดูกันอยู่ทุกวันนี้ จัดเป็นระบบเครือข่ายแบบ MAN



รูปที่ 2.30 เครือข่ายระดับเมือง (Metropolitan Area Network หรือ MAN)

3. เครือข่ายระดับประเทศ (Wide Area Network หรือ WAN) เป็นระบบเครือข่ายขนาดใหญ่ ใช้ติดตั้งบริเวณกว้าง มีสถานีหรือจุดเชื่อมต่อมากมาย มากกว่า 1 แสนจุด ใช้สื่อกลางหลายชนิด เช่น ระบบคลื่นวิทยุ ไมโครเวฟ หรือดาวเทียม



รูปที่ 2.31 เครือข่ายระดับประเทศ (Wide Area Network หรือ WAN)

2.4.4 ช่องทางการสื่อสารในระบบเครือข่ายคอมพิวเตอร์

ช่องทางการสื่อสาร (Communication Channel) หมายถึงสื่อ (Medium) ที่เป็นตัวกลางและอนุญาตให้ข้อมูล/สารสนเทศผ่านจากจุดส่งถึงผู้รับในระบบเครือข่ายคอมพิวเตอร์ หรือระหว่างคอมพิวเตอร์ในระบบเครือข่ายหนึ่งไปยังอีกเครือข่ายหนึ่ง ปริมาณของข้อมูลที่ช่องทางการสื่อสารสามารถนำไปได้นั้น เรียกว่า ความจุของช่องทางการสื่อสาร หรือ แบนด์วิดท์ (Bandwidth) ซึ่งนับเป็นจำนวนบิต (Bits) ต่อ 1 วินาที (bits per second : bps) สื่อที่ทำหน้าที่เป็นช่องทางการสื่อสาร ประกอบไปด้วย สายโทรศัพท์ (Telephone Line) เป็นช่องทางการสื่อสารในระบบเครือข่ายคอมพิวเตอร์ที่รู้จักและใช้กันอย่างแพร่หลาย ประกอบด้วยลวดทองแดงหุ้มด้วยฉนวน 2 เส้นพันบิดเป็นเกลียว เป็นสายสื่อสารที่ใช้ได้ทั้งในบ้านและในองค์กรธุรกิจ ซึ่งโดยทั่วไปองค์กรโทรศัพท์ฯ จะเป็นผู้รับผิดชอบในการให้บริการสื่อสารข้อมูลผ่านสื่อกลางชนิดนี้ บริการดังกล่าวได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Voice-grade Service** หมายถึง การสื่อสารข้อมูลในรูปของสัญญาณอนาล็อก (Analog) บนสายโทรศัพท์ โดยมีโมเด็มเป็นเครื่องแปลงสัญญาณ มีแบนด์วิดท์เท่ากับ 56 K bps โดยประมาณ
- **ISDN (Integrated Services Digital Network)** เป็นระบบเครือข่ายที่มีความเร็วและความจุของช่องสื่อสารสูงถึงประมาณ 128Kbps และยังสามารถแยกช่องสื่อสารเดียวกันออกเป็นช่องสื่อสารเสียง และช่องสื่อสารสัญญาณอิเล็กทรอนิกส์
- **Two-megabit Service** เป็นเทคโนโลยีใหม่ล่าสุดที่มีความเร็ว 2 M bps (2,000,000 bits per second) โดยผ่านโมเด็ม สามารถรับข้อมูลที่อยู่ในรูปของภาพเคลื่อนไหวในระบบวีดิทัศน์ รวมทั้งกราฟิกความเร็วสูง และการเข้าถึงสารสนเทศแบบ on line real-time ของผู้ใช้งาน จุดต่างๆ ในระบบเครือข่าย

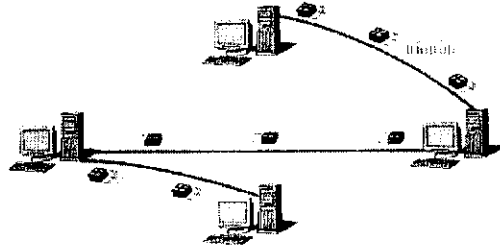
สายโคแอกเชียล (Coaxial Cable) หรือที่รู้จักในนามของสายโทรทัศน์ (Cable Television) ประกอบด้วยลวดทองแดงหลายเส้นหุ้มด้วยฉนวนกันน้ำ จัดเป็นสายสื่อสารที่มีความเร็วในการส่งสัญญาณสูง มีการรบกวนต่ำ นิยมใช้เป็นช่องสัญญาณอนาล็อกผ่านทะเล มหาสมุทร และใช้เป็นช่องสัญญาณในระบบเครือข่ายแบบ LAN มีความจุประมาณ 100 M bps ซึ่งจัดได้ว่าเป็นช่องสื่อสารที่มีความจุสูงมาก

สายใยแก้ว (Fiber Optic Cable) ประกอบด้วย หลอดหรือเส้นไฟเบอร์ขนาดเล็กจิ๋วทำเส้นผสมมนุษย์ ภายในกลวงเพื่อให้แสงเลเซอร์วิ่งผ่าน เป็นสายสื่อสารที่มีความจุของช่องสื่อสารนับเป็นล้านล้านบิตต่อวินาที (Gbps) เนื่องจากใช้แสงในการนำส่งข้อมูลแทนการใช้สัญญาณไฟฟ้า จึงทำให้มีความเร็วในการนำส่งข้อมูลมากกว่าช่องทางการสื่อสารทุกชนิด

สัญญาณไมโครเวฟ (Microwave Signals หรือ Radio Signals) เป็นช่องทางการสื่อสารไร้สายความเร็วสูง (High Speed Wireless) ส่งข้อมูลจากผู้ส่งไปยังผู้รับโดยอาศัยสัญญาณไมโครเวฟหรือสัญญาณวิทยุ โดยสัญญาณจะวิ่งเป็นเส้นตรง จึงต้องมีสถานีรับ-ส่งเป็นระยะๆ จากจุดส่งถึงจุดรับสถานีขยายสัญญาณจึงมักตั้งอยู่บนที่สูงเพื่อไม่ให้มีสิ่งกีดขวางขณะส่งสัญญาณไปในอากาศ

2.4.5 รูปแบบระบบเครือข่าย TCP/IP

การที่เครื่องคอมพิวเตอร์ที่ถูกเชื่อมโยงกันไว้ในระบบจะสามารถติดต่อสื่อสารกันได้นั้น จำเป็นจะต้องมีภาษาสื่อสารที่เรียกว่า โพรโตคอล (Protocol) เช่นเดียวกับคนเราที่ต้องมีภาษาพูดเพื่อให้สื่อสารเข้าใจกันได้ ในระบบอินเทอร์เน็ต จะใช้ภาษาสื่อสารมาตรฐานที่ชื่อว่า TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นภาษาหลัก ดังนั้นหากเครื่องคอมพิวเตอร์ไม่ว่าจะเป็นเครื่องระดับไมโครคอมพิวเตอร์ มินิคอมพิวเตอร์ หรือเมนเฟรมคอมพิวเตอร์ ก็สามารถเชื่อมโยงเข้าสู่อินเทอร์เน็ตได้



รูปที่ 2.32 ระบบเครือข่าย TCP/IP

ในการส่งข้อมูลผ่านทาง TCP/IP นั้น TCP/IP จะทำการแบ่งข้อมูลนั้นๆ ออกเป็นส่วนย่อยๆ ซึ่งเรียกว่า แพ็กเก็ต (Packet) โดยแต่ละส่วนจะถูกเพิ่มข้อมูลบอกตำแหน่งต้นทาง และปลายทางที่จะส่งไว้ให้ จากนั้นแพ็กเก็ตเหล่านี้จะถูกส่งกระจายผ่านไปยังเส้นทางต่างๆ ที่เชื่อมโยงกันในระบบตามเส้นทางที่สามารถส่งถึงปลายทางได้ โดยแต่ละแพ็กเก็ตไม่จำเป็นต้องเรียงลำดับหรือไปตามเส้นทางเดียวกัน ซึ่งในระบบจะมีอุปกรณ์ที่เรียกว่า เราเตอร์ (Router) จะเป็นตัวคอยจัดหาเส้นทางที่ดีที่สุดให้กับทุกแพ็กเก็ต เมื่อไปถึงผู้รับที่ปลายทางแล้ว แพ็กเก็ตจะมารวมกันเป็นข้อความยาวๆ เหมือนเดิม แต่ถ้าแพ็กเก็ตใดขาดหายหรือตกหล่น คอมพิวเตอร์ก็จะตรวจสอบ และส่งแพ็กเก็ตมาใหม่ จนข้อมูลครบเหมือนเดิม

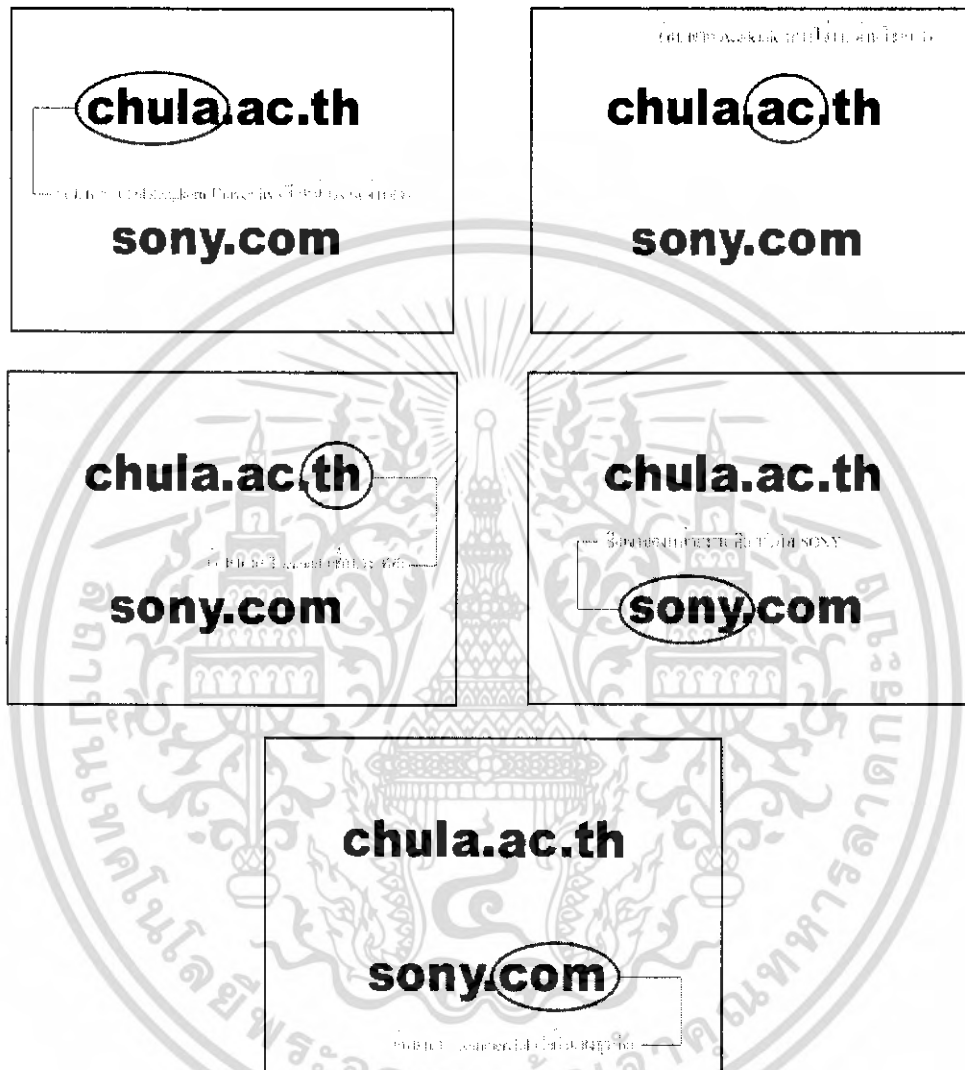
2.4.6 IP Address และ DNS

หมายเลขประจำเครื่อง หรือที่อยู่ (Address) ของผู้ใช้อินเทอร์เน็ตแต่ละคนจะมีที่อยู่ประจำเครื่อง ซึ่งกำหนดเรียกตัวเลขระบุตำแหน่ง เช่น 202.44.202.222 , 201.44.202.3 หรือ 203.146.7.200 เป็นต้น แต่ระบบหมายเลขมีข้อบกพร่องคือ จำยากและไม่ได้สื่อความหมายให้ผู้ใช้งานทั่วไปทราบ ดังนั้น จึงมีผู้คิดระบบตั้งชื่อให้ง่ายขึ้น เรียกว่า ระบบชื่อของเครื่อง (Domain Name System-DNS) DNS จะเปลี่ยนตัวเลขให้เป็นคำที่อ่านแล้วเข้าใจและจำได้ง่าย เช่น chula.ac.th , moc.go.th หรือ microsoft.com เป็นต้น การกำหนดDNSจะเรียงลำดับความสำคัญของชื่อจากขวาไปซ้ายโดยมีจุดคั่นซึ่งมีหลักการดังต่อไปนี้

- ชื่อทางขวาสุดจะบอกชื่อประเทศเช่น
 - th=ประเทศไทย
 - uk = ประเทศอังกฤษ
- ชื่อถัดมาจากชื่อประเทศจะบอกลักษณะของหน่วยงาน แบ่งออกเป็น 5 กลุ่ม คือ
 - ac หมายถึง Academic สถาบันการศึกษา
 - co หมายถึง Commercial ภาคองค์กร ภาคเอกชน
 - go หมายถึง Government หน่วยงานราชการ
 - or หมายถึง Organization องค์กรที่ไม่แสวงหาผลกำไร
 - net หมายถึง Network องค์กรที่ให้บริการระบบเครือข่าย
- นอกจากนี้ในส่วนทางขวาสุดอาจไม่แบ่งตามลักษณะของสองข้อที่ผ่านมา แต่ใช้เพียงแค่คำย่อคำเดียว โดยไม่ต้องแยกออกเป็นชื่อประเทศ และลักษณะหน่วยงาน ซึ่งได้แก่
 - com หมายถึง Commercial ใช้ในธุรกิจ บริษัท ห้างร้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- edu หมายถึง Education ใช้ในสถานการศึกษา
- gov หมายถึง Government ใช้ในหน่วยงานราชการ
- net หมายถึง Network ใช้ในหน่วยงานที่เป็นเครือข่าย
- ทางซ้ายสุดจะเป็นชื่อหน่วยงานที่เป็นเจ้าของ Address นั้นๆ



รูปที่ 2.33 การตีความหมายของ DNS

2.4.7 UDP (User Datagram Protocol)

UDP เป็นโปรโตคอลที่ถูกออกแบบมาให้ทำหน้าที่รับส่งข้อมูลโดยมีขั้นตอนการทำงานไม่ซับซ้อนและทำงานได้รวดเร็วแต่มีจุดด้อยคือมีความน่าเชื่อถือ (unreliable) และเป็นการสื่อสารแบบไม่ต่อเนื่อง (connectionless) โปรโตคอล UDP ทำงานในชั้น Transport Layer ซึ่งจะต้องพึ่งพาโปรโตคอล IP ในการรับส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UDP Header

16-bit Source port	16-bit destination port
length	checksum
data	

รูปที่ 2.34 UDP Header

ตำแหน่ง	ชื่อ	อธิบาย
บิต 0-15	Source port number	หมายเลขพอร์ตต้นทางที่ส่งค่าตัวแปรนี้ มีความยาว 16 บิต
บิต 16-31	Destination port number	หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับค่าตัวแปรนี้ มีความยาว 16 บิตเช่นกัน
บิต 32-47	UDP length	หมายความว่า ค่าที่น้อยที่สุดในฟิลด์นี้คือ 8 ซึ่งเป็นขนาดของ Header
บิต 48-63	Checksum	เป็นตัวตรวจสอบความถูกต้องของ UDP datagram และจะนำข้อมูลบางส่วนใน IP Header มาคำนวณด้วย

ตารางที่ 2.1 แสดงช่วงของ UDP Header

UDP Checksum

Checksum เป็นเลข 16 บิตถูกคำนวณด้วยวิธี one's complement โดยนำ Pseudo Header และ ข้อมูลทั้งหมดใน UDP Datagram มาคำนวณ

Pseudo Header เป็นข้อมูลที่อยู่ในส่วนของ IP Header ประกอบด้วยฟิลด์ Source IP address , Destination IP Address , Zero , Protocol , UDP length ดังแสดงในรูปด้านล่าง

32-bit source IP address		
32-bit destination IP address		
zero	8-bit protocol(17 for UDP)	16 bit length

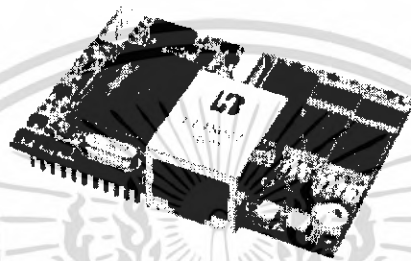
รูปที่ 2.35 Pseudo Header

หากค่า Checksum ที่คำนวณออกมาเป็น 0 ค่า checksum จะถูกเซตเป็น 1 ทั้งหมดแทน (มีค่าเท่ากับในระบบ 1's complement) ทั้งนี้เพราะในบางแอปพลิเคชันที่ไม่ต้องการตรวจสอบค่า checksum ในระดับ UDP จะเซตค่านี้เป็น 0 (disable checksum)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ทฤษฎีเกี่ยวกับ Rabbit Core RCM2200

ลักษณะโครงสร้างของ Rabbit Core RCM2200 เป็นชิ้นส่วนไมโครโปรเซสเซอร์ที่ถูกออกแบบให้มีความสามารถที่หลากหลาย มี Port Ethernet สำหรับ Lan และ Internet ระบบวงจรที่ใช้ใน Rabbit Core RCM2200 ถูกออกแบบให้ง่ายต่อการใช้งาน มีความเร็วในการทำงานอยู่ที่ 22.1 MHz หน่วยความจำที่ใช้เป็นแบบ Static Ram และ Flash Memory สัญญาณนาฬิกาที่ใช้งานในวงจรที่ 2 แหล่ง (Oscillator และ time – Keeping) ใช้แหล่งจ่ายไฟ +5V ในการทำงานของบอร์ดนอกจากนี้ยังสามารถติดต่อกับอุปกรณ์ภายนอก โดยผ่านทาง Port ที่จัดสรรไว้ผ่านคอนเน็คเตอร์ J4 กับ J5



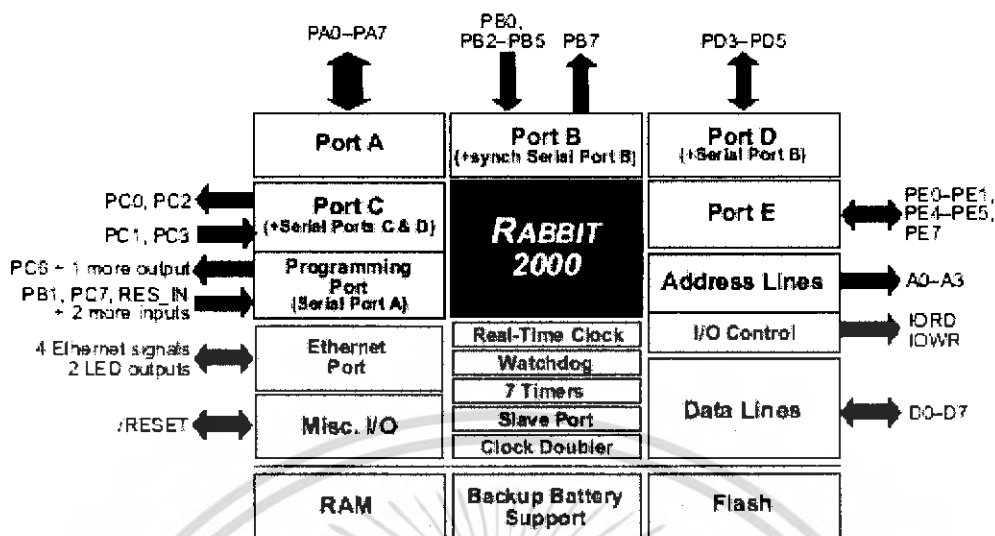
รูปที่ 2.36 บอร์ดวงจร Rabbit Core RCM2200

2.5.1 ลักษณะที่สำคัญและความสามารถของ Rabbit Core RCM2200

- ขนาดเล็ก 1.60" x 2.30" x 0.86" หรือ (41 mm x 58mm x 22mm)
- ไมโครโปรเซสเซอร์ Rabbit 2000 ใช้ความเร็วในการทำงานที่ 22.1 MHz
- มีขาที่เป็น Input / Output ทั้งหมด 26 ขา เป็นแบบ Parallel มีโหมดการทำงานหลาย
- มี Data Bus 8 เส้น (D0-D7)
- มี Address Bus 4 เส้น (A0-A3)
- มี Flash Memory ขนาด 256 Kbytes และ Static Ram ขนาด 128 Kbytes
- มีสัญญาณนาฬิกาให้จังหวะการทำงานใช้หัวต่อสัญญาณแบบ RJ-45 เป็น Ethernet Port Serial เป็นชนิด CMOS ซึ่งใช้อัตราบอรรถแบบ Asynchronous ได้ความเร็วสูงสุดถึง 3345,600 bps อัตราบอรรถที่ใช้แบบ Synchronous ได้ความเร็วสูงสุดถึง 138,240 bps

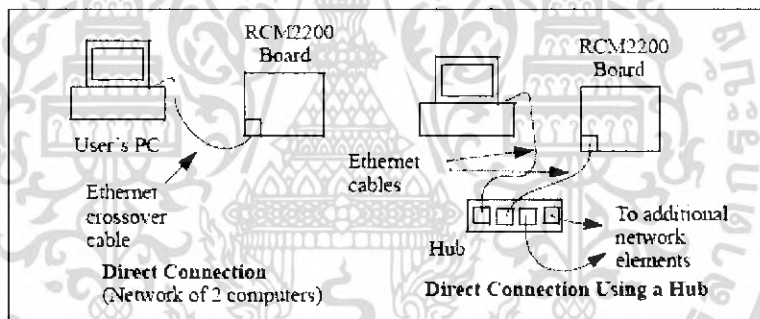
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 ส่วนประกอบอุปกรณ์และอุปกรณ์ย่อยของระบบ Rabbit Core RCM2200



รูปที่ 2.37 อินพุตและเอาต์พุตของ Rabbit Core RCM2200

2.5.3 การเชื่อมต่อ RCM2200



รูปที่ 2.38 การเชื่อมต่อ Board RCM2200 กับระบบคอมพิวเตอร์

RCM2200 ออกแบบให้เชื่อมต่อกับ Motherboard เพื่อให้มีประสิทธิภาพในการใช้งาน โดยมีจุดเชื่อมต่อกับอุปกรณ์ภายนอก คือ Headers 26 pin 2 Header ในบอร์ด RCM2200 จะประกอบไปด้วย 10Base-T Ethernet Port, 256Kbytes Flash Memory และ 128 Kbytes Static RAM

2.5.4 การสื่อสารแบบอนุกรม (Serial Communication)

ในบอร์ด Rabbit Core RCM2200 จะไม่มี RS - 232 หรือ RS - 485 ที่เป็นตัวรับและส่งโดยตรงบนบอร์ด อย่างไรก็ตาม RS - 232 หรือ RS - 485 ส่วนที่ใช้ติดต่ออาจจะนำมาต่อบนบอร์ด Rabbit Core RCM2200 เป็นต้นว่าบอร์ดที่มีชิพการรับส่ง ที่สนับสนุนมาตรฐานการรับส่ง RS - 232

2.5.5 โปรแกรม Dynamic C

Dynamic C คือ ระบบการพัฒนารวมของการเขียนซอฟต์แวร์ สร้างมาจากระบบคอมพิวเตอร์ของ IBM ซึ่งออกแบบให้เข้ากับคอมพิวเตอร์ทั่วไปให้ใช้งานได้โปรแกรม Dynamic C นั้นมีการใช้ในเครื่องขายตั้งแต่ปี 1989 ถูกออกแบบสำหรับการเขียนโปรแกรมที่ฝังระบบ และมีความสามารถตรวจสอบในตัวของมันเองอย่างรวดเร็ว สำหรับไมโครโปรเซสเซอร์ Rabbit 2000 ที่ใช้งานกันทั่วไป สามารถติดต่อกันโดยสายแพ 10 สาย ที่พอร์ต C โปรแกรมพื้นฐานของระบบ มีข้อมูลประมาณ 1,000 bytes ที่ใช้ในการจัดเตรียม Debugging และการติดต่อข้อมูลต่างๆ Dynamic C ต้องการไบออส(BIOS) เพื่อใช้ในการตรวจสอบโปรแกรมเพื่อที่จะให้ใช้งานได้สะดวกถ้าผู้ใช้หยุดการ Run โปรแกรมและใช้โปรแกรมใหม่ BIOS ก็จะเริ่มการทำงานใหม่ตลอด Dynamic C ออกแบบให้เข้ากับภาษา Assembly หรือใช้ร่วมกับโปรแกรมภาษาซี การ Interrupt อาจเกิดขึ้นได้กับการเขียนในภาษา Dynamic C หรือภาษา Assembly ซึ่งซอฟต์แวร์ Dynamic C เป็นซอฟต์แวร์ที่มีประสิทธิภาพในการเขียนโปรแกรมเพื่อควบคุมการทำงานของไมโครโปรเซสเซอร์ Rabbit2000 โดยซอฟต์แวร์มีโครงสร้างคล้ายคลึงกับภาษา C ลักษณะของตัวซอฟต์แวร์โปรแกรม Dynamic C จะสนับสนุน TCP/IP โดยจะมี Libraries หลักที่สนับสนุนคือ DCRTCP.LIB, DNS.LIB, IP.LIB, TCP.LIB, UDP.LIB, NET.LIB ส่วนในการติดต่อชั้นเครือข่ายของโปรโตคอล TCP/IP จะมี Libraries ARP.LIB กับ ICMP.LIB

ผู้ใช้โปรแกรม Dynamic C มีตัวเลือกในการที่จะพัฒนาซอฟต์แวร์ภาษาเขียน ใน flash memory ขนาด 256 Kbytes หรือใน Static Ram ขนาด 128 Kbytes ผลการทำงานในหน่วยความจำคือการบันทึกข้อมูลสามารถบันทึกได้ถึง 100,000 ครั้งของการเขียน

ข้อเสียของการใช้ Flash Memory เมื่อมีการดับกโปรแกรมเพื่อขัดจังหวะการทำงาน จะทำให้ Interrupt เกิดข้อผิดพลาด การทำงานของโปรแกรมก็จะหยุดตามไปด้วย

บทที่ 3

การออกแบบวงจร

3.1 การออกแบบสามารถแบ่งออกเป็น 2 ส่วนใหญ่คือ

1. เครื่องวัดคลื่นไฟฟ้าหัวใจ (ECG : Electrocardiograph)

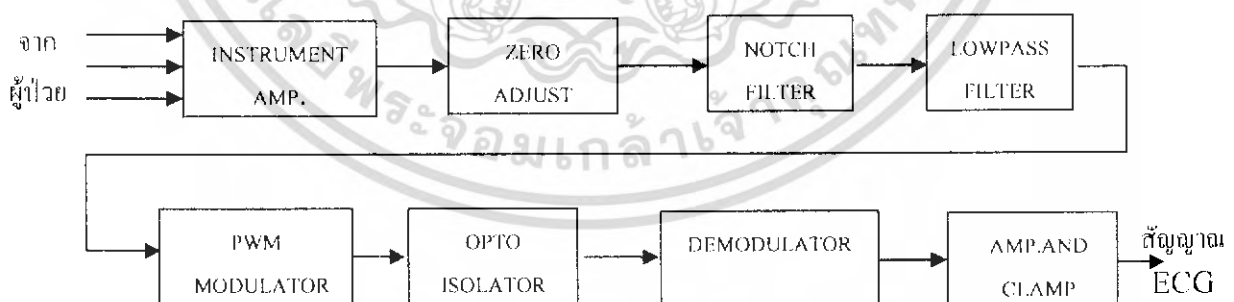
เครื่อง อี.ซี.จี. เป็นเครื่องมือที่ใช้ในการวัดสัญญาณคลื่นไฟฟ้าหัวใจ ซึ่งสัญญาณคลื่นไฟฟ้าหัวใจจะมีขนาดสัญญาณน้อยมาก ซึ่งประมาณ 1 มิลลิโวลต์ (mV) โดยรับสัญญาณจากอิเล็กโทรดที่ติดที่ผิวหนัง ซึ่งจะต้องมีส่วนขยายสัญญาณ (Instrument Amp) และจะต้องมีการป้องกันแรงดันไฟฟ้าออฟเซตที่เป็นไฟฟ้ากระแสตรงมารบกวนอีกโดยใช่วงจรปรับศูนย์ (Auto Zero Circuit) และยังมีส่วนที่เราต้องการกำจัดออกไปคือสัญญาณความถี่ 50 Hz ซึ่งต้องกำจัดออกไปโดยใช่วงจรกรองความถี่แบบกั้นไม่ให้ความถี่ผ่านเฉพาะช่วง (Notch Filter) ซึ่งจะยอมให้ความถี่สูงกว่าและต่ำกว่า 50 Hz ผ่านไปด้วย จากนั้นส่งสัญญาณดังกล่าวผ่านวงจรกรองความถี่ต่ำผ่านก่อนที่จะนำไปมอดูเลตเป็นสัญญาณ PWM ส่งไปยังวงจรแยกกราวด์เพื่อป้องกันกระแสไฟฟ้าที่จะเป็นอันตรายต่อผู้ป่วย จากนั้นจะนำสัญญาณที่ได้ไปทำการ ดีมอดูเลต เพื่อให้ได้สัญญาณคลื่นไฟฟ้าหัวใจตามเดิม

2. RABBIT

Rabbit 2000 นี้ได้พัฒนามาจากไมโครคอนโทรลเลอร์ Z180 ซึ่ง Rabbit สามารถในการอินเตอร์เฟซเข้ากับเครือข่ายอินเทอร์เน็ตซึ่งการเขียนโปรแกรมเพื่อใช้งานบน Rabbit โดยใช้ไดนามิก C ในการเขียน

3.2 การออกแบบวงจรและการสร้างภาคขยายคลื่นไฟฟ้าหัวใจ

การออกแบบภาคขยายคลื่นหัวใจไฟฟ้าเราสามารถออกแบบเป็นบล็อกไดอะแกรมได้ดังนี้



รูปที่ 3.1 บล็อกไดอะแกรมของเครื่องวัดคลื่นไฟฟ้าหัวใจ(ECG)

จากบล็อกไดอะแกรมที่เห็น เรานำมาออกแบบโดยแยกแต่ละส่วนออกจากกันก่อนเพื่อง่ายต่อการออกแบบ หลังจากนั้นนำมาเชื่อมต่อกัน แล้วทำการปรับแต่งเพื่อให้คุณลักษณะที่สีของสัญญาณ ตามที่เราต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 วงจรอินสตรูเมนแอมพลิไฟเออร์ (Instrument Amplifier)

วงจรอินสตรูเมนแอมพลิไฟเออร์ (Instrument Amplifier) ดังแสดงในรูปที่ 3.2 เป็นวงจรแรกที่ขยายคลื่นไฟฟ้าหัวใจที่มีขนาดของสัญญาณน้อยมากเพียงประมาณ 1 mV โดยการรับสัญญาณจากอิเล็กโทรดที่ติดบนผิวหนัง ซึ่งมีค่าความต้านทานสูง และมีสัญญาณรบกวนจากไฟฟ้ากระแสสลับที่มีความถี่ 50 Hz ปนมาด้วย ดังนั้นวงจรที่จะนำมาใช้ต้องมีคุณสมบัติพิเศษดังต่อไปนี้

ก. อินพุตอิมพีแดนซ์สูงมาก เมื่อเทียบกับความต้านทานของผิวหนัง เพื่อป้องกันการเสียดสีของวงจรและการรบกวนสัญญาณที่ป้อนเข้าสู่อินพุต การเสียดสีของวงจรจะมีผลเสียต่อวงจรขยายคือสัญญาณรบกวนที่เข้ามาในลักษณะคอมมอนโหมด (Common Mode Signal) ไม่สามารถกำจัดออกไปได้ และยังทำให้เกิดแรงดันไฟฟ้าออฟเซต (Offset Voltage) ซึ่งจะถูขยายให้มีขนาดมากขึ้นที่เอาต์พุตถ้าแรงดันไฟฟ้าออฟเซตมีค่ามากจะทำให้วงจรขยายอิมตัวมีแรงดันไฟฟ้าเอาต์พุต อยู่ที่ค่าเกือบเท่ากับแรงดันไฟฟ้าของแหล่งจ่ายไฟด้านใดด้านหนึ่ง (บวก หรือ ลบ) และวงจรไม่สามารถทำงานได้

ข. ค่า CMRR (Common Mode Rejection Ratio) สูง ค่า CMRR เป็นคุณสมบัติอย่างหนึ่งของวงจรขยายความแตกต่างที่สามารถกำจัดสัญญาณรบกวนได้ คุณสมบัติอันนี้ก็คือการมีอัตราขยายของสัญญาณดิฟเฟอเรนเชียลโหมด (Differential Mode Signal) สูงและมีอัตราขยายของสัญญาณคอมมอนโหมดต่ำ ทั้งนี้เนื่องจากสัญญาณที่ต้องการขยาย (ECG) จะเข้าไปที่อินพุตในลักษณะสัญญาณดิฟเฟอเรนเชียลโหมด

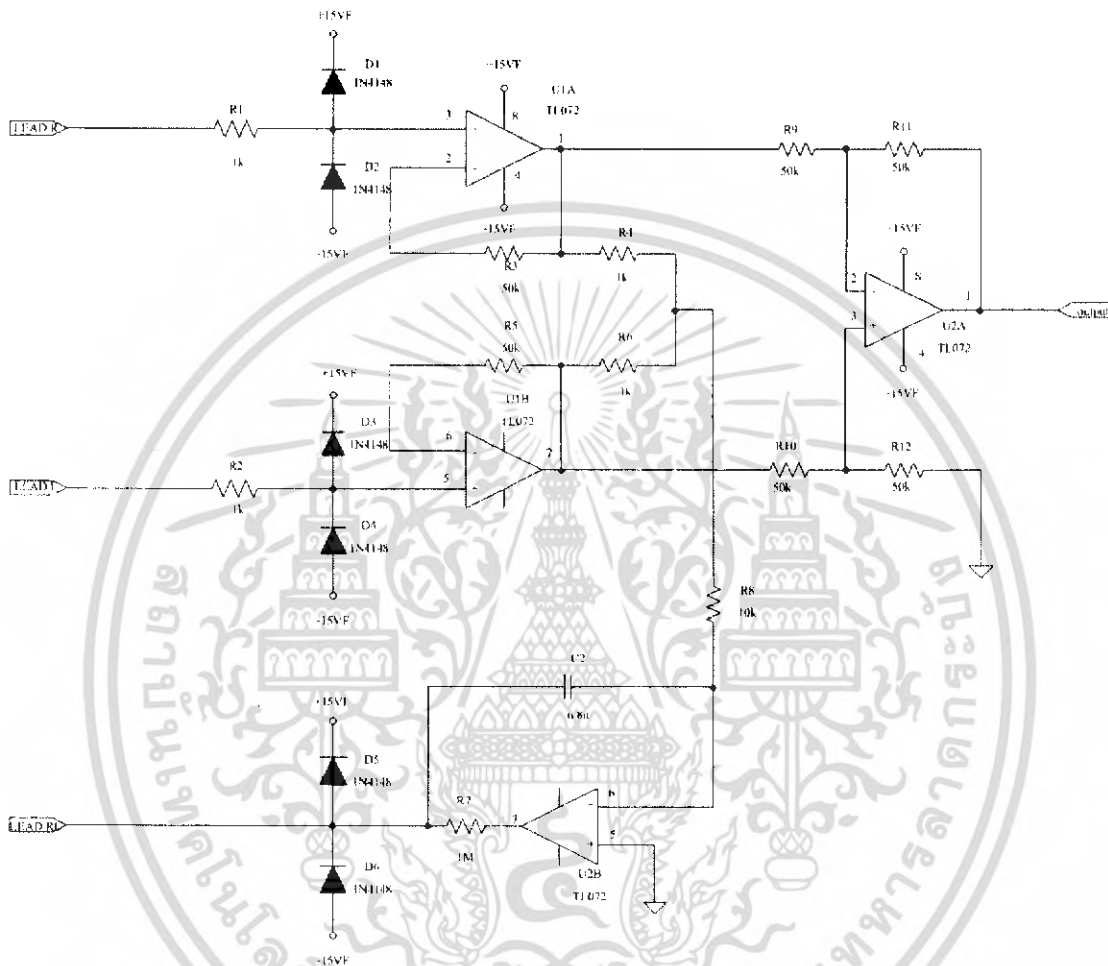
วงจรขยายในภาคแรกไม่ควรให้มีอัตราขยายมากเกินไป เพราะว่า ถ้าหากเกิดแรงดันไฟฟ้าออฟเซตที่อินพุต ไม่ว่าจะสาเหตุใดก็ตามจะทำให้เกิดสัญญาณที่เอาต์พุตมีแรงดันลอยจากศูนย์มาก ซึ่งถ้าหากมากจนใกล้เคียงกับแรงดันไฟฟ้าของแหล่งจ่ายไฟ จะทำให้เกิดการอิมตัวของสัญญาณจนวงจรไม่สามารถทำงานได้

จากวงจรในรูปที่ 3.2 จะประกอบด้วยวงจรอินสตรูเมนแอมพลิไฟเออร์ และวงจรขับขาขวา โดยวงจรอินสตรูเมนแอมพลิไฟเออร์ ประกอบด้วย U1A , U1B และ U2A ส่วน U2B เป็นวงจรขับขาขวาวงจรอินสตรูเมนแอมพลิไฟเออร์จะทำหน้าที่ขยายสัญญาณความแตกต่าง ซึ่งอัตราขยายของวงจรอินสตรูเมนแอมพลิไฟเออร์มีค่าเท่ากับ

$$\begin{aligned} A_v &= \frac{V_o}{V_{LEADR} - V_{LEADL}} = \frac{R_{11}}{R_8} \left(1 + \frac{2R_3}{R_4 + R_5} \right) \\ &= \frac{50k}{50k} \left(1 + \frac{2(50k)}{1k + 1k} \right) \\ &= 51 \text{ เท่า} \end{aligned}$$

∴ อัตราขยายแรงดันไฟฟ้าทั้งหมดของวงจรขยายความแตกต่างเท่ากับ 51 เท่า

วงจรขยายในภาคแรกๆ ไม่ควรมีอัตราขยายให้มากเกินไป เพราะจะทำให้เกิดมีเอาต์พุตลอยขึ้นไปหรือต่ำลงจากระดับศูนย์ ซึ่งบางครั้งอาจทำให้อิมิตวอย์ที่ค่าใกล้กับแรงดันแหล่งจ่ายไฟ ทำให้วงจรสามารถทำงานได้ ส่วนวงจรขั้วขาจะช่วยลดแรงดันโหมคร่วม ซึ่งจะมีผลให้สัญญาณรบกวนลดลง



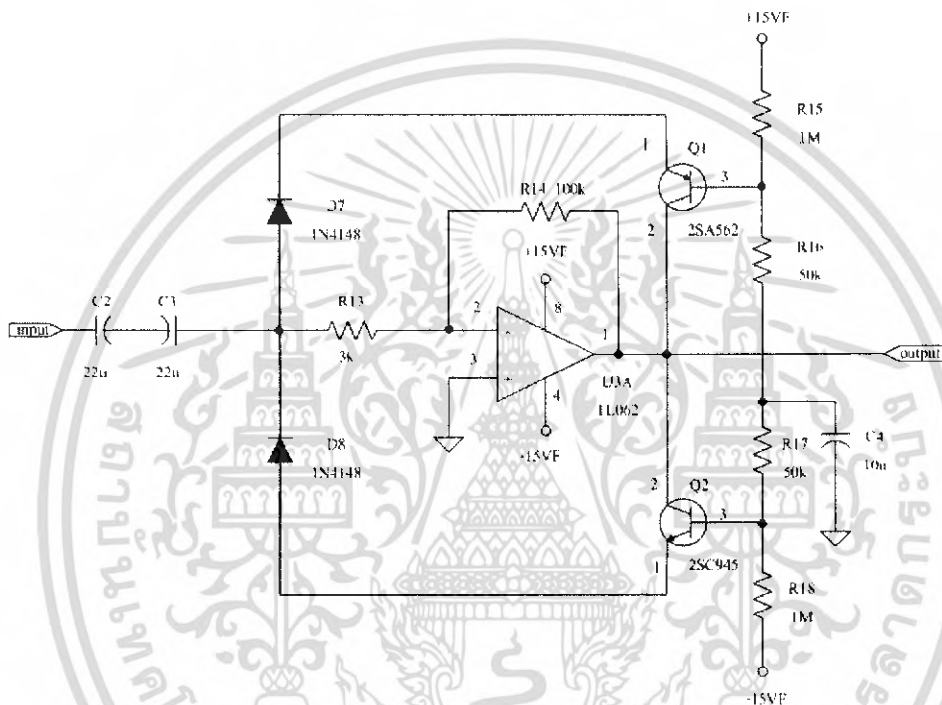
รูปที่ 3.2 วงจรอินสตรูเมนแอมพลิไฟเออร์

3.2.2 วงจรปรับศูนย์อัตโนมัติ (Auto Zero Circuit)

เมื่อสัญญาณคลื่นไฟฟ้าหัวใจผ่านวงจรขยายแตกต่างกัน มักจะมีแรงดันไฟฟ้าออฟเซตถูกขยายมาด้วย ซึ่งแรงดันนี้เป็นแรงดันไฟฟ้ากระแสตรง ซึ่งสามารถแยกออกจากสัญญาณคลื่นไฟฟ้าหัวใจได้ โดยการให้สัญญาณผ่านตัวเก็บประจุ โดยการต่ออนุกรมกัน เนื่องจากสัญญาณคลื่นไฟฟ้าหัวใจมีความถี่ต่ำอยู่ในช่วง 0.5-100 Hz ดังนั้นตัวเก็บประจุที่ใส่ต้องมีค่ามาก ทั้งนี้เพื่อให้สัญญาณในช่วงความถี่ต่ำผ่านไปได้ และอีกปัญหาที่ตามมาคือ การเกิดสัญญาณรบกวนที่อเล็กโตรด ถ้าหากว่าคนไข้ได้มีการขยับตัวจะทำให้ความต้านทานที่ผิวหนังของคนไข้เปลี่ยนแปลง จะเกิดเป็นแรงดันไฟฟ้าออฟเซตที่ทางออกของวงจรขยาย ความแตกต่าง สัญญาณคลื่นไฟฟ้าหัวใจที่ผ่านตัวเก็บประจุจะลอยออกห่างจากระดับศูนย์และจะกลับเข้าสู่ศูนย์ช้ามากยังมีค่าความจุของตัวเก็บประจุมีค่ามากเพียงใด ก็จะทำให้เวลาในการกลับเข้าสู่ระดับศูนย์ช้ามากขึ้น เป็นผลเนื่องมาจากกระบวนการชาร์จและดิสชาร์จของตัวเก็บประจุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการของวงจรปรับศูนย์คือจะช่วยลดค่าเวลาคงที่ (Time Constant) ให้เหลือน้อยมาก เมื่อเกิดแรงดันไฟฟ้าออฟเซตขึ้น ถึงระดับที่กำหนดไว้ ในขณะที่ยังไม่เกิดแรงดันไฟฟ้าออฟเซตค่าเวลาคงที่จะมีค่าเท่ากับผลคูณของ R_{13} กับ C_3 เท่ากับ $3k \times 22\mu = 66msec$ และเมื่อมีไฟฟ้าออฟเซตประมาณ $+0.9V$ หรือ $-0.9V$ (เท่ากับ แรงดันไฟฟ้าคร่อมไดโอดกับ V_{BE} ของทรานซิสเตอร์) โดยทรานซิสเตอร์จะทำหน้าที่เป็นทางผ่านของสัญญาณ ซึ่งจะทำให้ค่าเวลาคงที่ลดลงเหลือเพียงเท่ากับผลคูณของ C_3 กับค่าความต้านทานภายในของทรานซิสเตอร์ ซึ่งมีค่าน้อย ส่วนออปแอมป์จะทำหน้าที่เป็นวงจรขยายแบบกลับเฟส (Inverting Amplifier)



รูปที่ 3.3 วงจรปรับศูนย์อัตโนมัติ

จากวงจรปรับศูนย์อัตโนมัติ (Auto Zero Circuit) จะมีอัตราขยายแรงดันไฟฟ้า ดังนี้

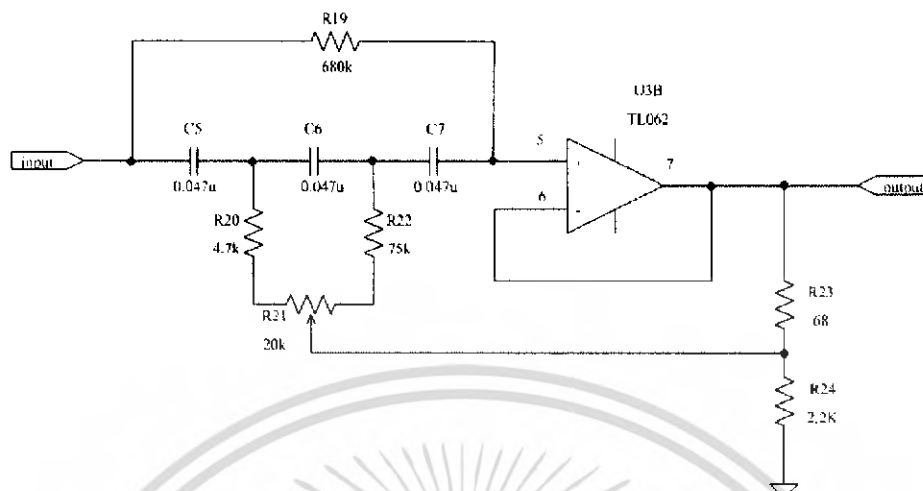
$$\begin{aligned} \text{จากสมการ} \quad A &= \frac{R_{14}}{R_{13}} \quad (\text{เป็นวงจรขยายแรงดันแบบกลับเฟส Inverting Amplifier}) \\ &= 100k/3k \\ &= 33.33 \end{aligned}$$

ดังนั้น วงจรนี้จะมีอัตราขยายแบบกลับเฟส 33.33 เท่า

3.2.3 วงจรกรองความถี่แบบกั้นไม่ให้ความถี่ผ่านเฉพาะช่วง (Notch Filter)

ถึงแม้ว่าวงจรขยายในภาคแรกจะเป็นวงจรขยายความแตกต่าง ซึ่งมีคุณสมบัติในการกำจัดสัญญาณรบกวนได้ก็ตาม แต่ถ้าเกิดความไม่สมดุลของวงจรขึ้นมาสัญญาณรบกวนซึ่งส่วนใหญ่เป็นความถี่ 50 Hz ก็สามารถผ่านไปได้ แต่เนื่องจากคลื่นไฟฟ้าหัวใจมีความถี่ต่ำอยู่ในช่วง 0.5 ถึง 100 Hz ถ้าเราใช้วงจรกรองความถี่ต่ำ (Low pass filter) แบบธรรมดาที่ยอมให้ความถี่ต่ำกว่า 50 Hz ผ่านไปได้ก็จะทำให้ออกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

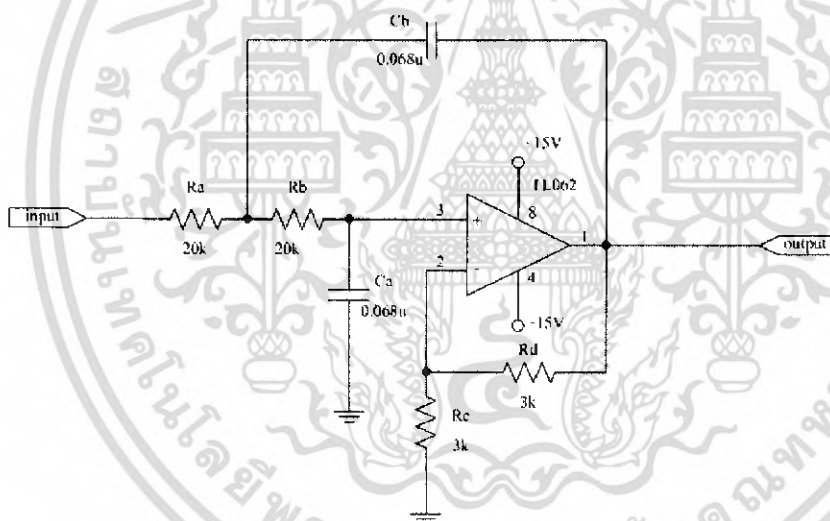
สัญญาณไฟฟ้าหัวใจส่วนที่มีความถี่สูงและต่ำกว่า 50 Hz ผ่านไปได้ ส่วนสัญญาณรบกวนความถี่ 50 Hz จะถูกกำจัดออกไป วงจรแสดงดังรูปที่ 3.4



รูปที่ 3.4 วงจรกรองความถี่แบบกันไม่ให้ความถี่ผ่านเฉพาะช่วง

3.2.4 วงจรกรองความถี่ต่ำผ่าน (Low Pass Filter)

ทำหน้าที่กำจัดสัญญาณสูงกว่า 117 Hz ก่อนที่จะนำไป มอดูเลตในภาคถัดไป



รูปที่ 3.5 วงจรกรองความถี่ต่ำผ่าน

จากรูปที่ 3.5 เป็นวงจรกรองความถี่ต่ำผ่านจะมีความถี่ คัทออฟ (Cut off frequency) อยู่ที่ประมาณ 117 Hz คำนวณได้จาก

$$\begin{aligned}
 f_c &= \frac{1}{2\pi\sqrt{R_a \times R_b \times C_a \times C_b}} \\
 &= \frac{1}{2\pi\sqrt{20k \times 20k \times 0.068\mu \times 0.068\mu}} \\
 &= 117 \text{ Hz}
 \end{aligned}$$

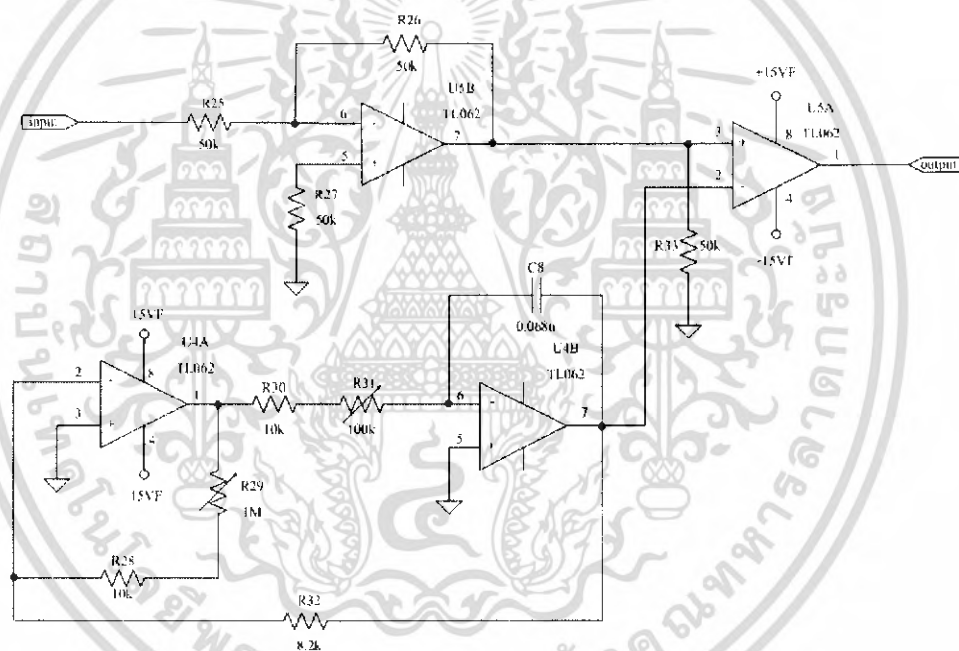
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และสามารถคำนวณอัตราขยายของวงจรได้จาก

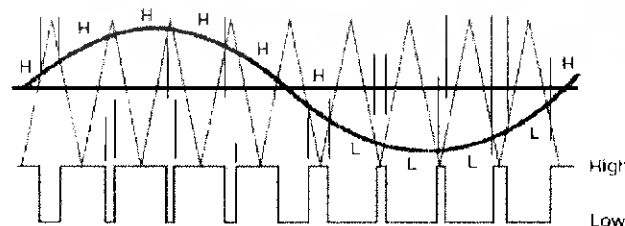
$$A_v = 1 + \left(\frac{R_d}{R_c} \right) = 1 + \left(\frac{3k}{3k} \right) \\ = 2 \text{ เท่า}$$

3.2.5 วงจรพัลส์-วิดท์ มอดูเลเตอร์ (Pulse-Width Modulator)

จากรูปที่ 3.6 U4A และ U4B ทำหน้าที่สร้างสัญญาณสามเหลี่ยม (Triangle Signal) เพื่อเป็นสัญญาณเปรียบเทียบ (Reference Signal) เข้าที่ขาอินพุตกลับเฟส (Inverting Input) ของ U5A ซึ่งทำหน้าที่เปรียบเทียบสัญญาณ (Comparator) โดยสัญญาณที่ออกจากวงจรรอกความถี่แบบกันไม่ให้ความถี่เฉพาะ ช่วงจะถูกป้อนเข้าวงจรขยายแบบกลับเฟสก่อนนำเข้าสู่ขาอินพุตไม่กลับเฟส (Non-Inverting Input) U5A สัญญาณที่ออกเอาท์พุทของ U5A จะมีลักษณะเป็นพัลส์ (Pulse Width) จะแปรเปลี่ยนตามสัญญาณที่เข้ามาที่ขาอินพุต ดังแสดงได้ดังรูปที่ 3.7



รูปที่ 3.6 วงจรพัลส์-วิดท์ มอดูเลเตอร์



รูปที่ 3.7 แสดงการเปลี่ยนความกว้างพัลส์โดยการมอดูเลตแบบ PWM

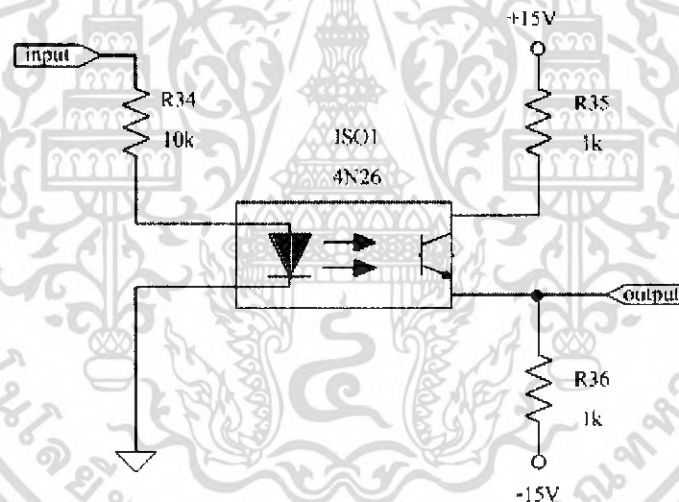
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหตุผลที่ต้องทำการมอดูเลตเป็นสัญญาณ PWM เพราะว่าเราจะต้องส่งสัญญาณไปไบอัสตัวส่งสัญญาณแสงซึ่งเป็นไดโอดให้ทำงานอยู่ในช่วงลิเนียร์ ซึ่งการปรับกระแสไบอัสให้อยู่ในช่วงลิเนียร์นี้กระทำได้ยากและตัวส่งสัญญาณแสงแต่ละตัวก็มีคุณสมบัติแตกต่างกันไป เราจึงทำการปรับไบอัสทุกครั้งที่มีการเปลี่ยนตัวส่งสัญญาณ (แม้ว่าตัวส่งสัญญาณจะเป็นเบอร์เดียวกันก็ตาม) และอาจจะต้องมีการเปลี่ยนแปลงค่าอุปกรณ์ของวงจรถัดไปด้วย

ดังนั้นเราจึงแก้ปัญหาดังกล่าวโดยการมอดูเลตสัญญาณก่อนจะส่งผ่านไปยังตัวส่งสัญญาณแสง โดยเราจะใช้วงจร พัลส์-วิดท์ มอดูเลเตอร์ (Pulse-Width Modulator) ดังแสดงใน รูปที่ 3.6

3.2.6 วงจรส่งผ่านสัญญาณด้วยแสง (Opto Isolator)

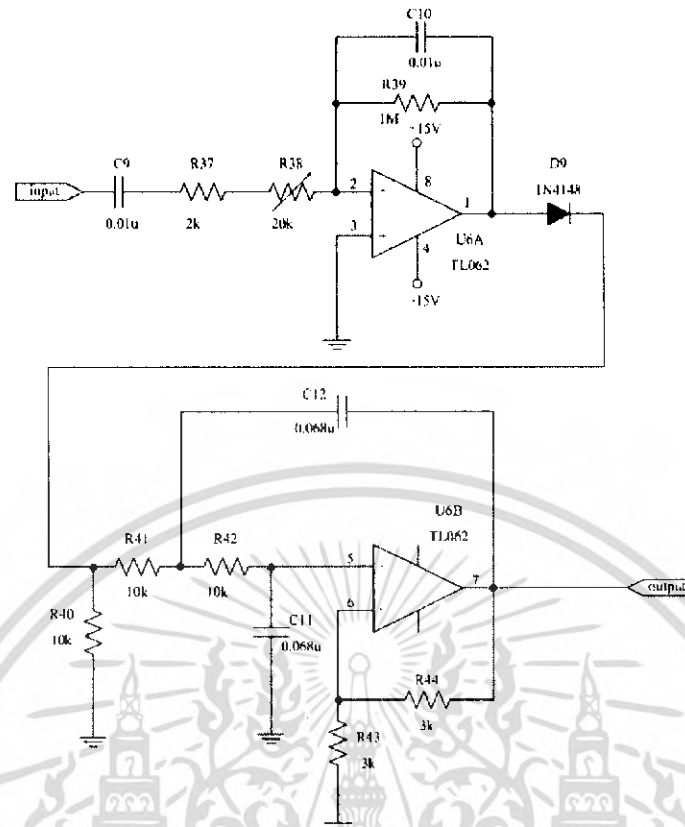
เพื่อทำการแยกจุดดินระหว่างวงจรส่วนหน้าที่สัมผัสร่างกายกับวงจรถัดไป เพื่อป้องกันกระแสรั่วไหลจากเครื่องไปทำอันตรายต่อคนไข้ได้ จากวงจรในรูปที่ 3.8 ความต้านทาน R35 และ R36 จะเป็นตัวป้องกันกระแสไบอัสให้กับตัวส่ง การตั้งไบอัสทำได้โดยการเปลี่ยนตัวความต้านทาน R35 และ R36 จนกระทั่งแรงดันไฟฟ้าที่คอลเลกเตอร์ของตัวรับสัญญาณแสงมีค่าประมาณครึ่งหนึ่งของแรงดันไฟฟ้าจากแหล่งจ่ายไฟ +V



รูปที่ 3.8 วงจรส่งผ่านสัญญาณด้วยแสง

3.2.7 วงจรดีมอดูเลเตอร์ (Demodulator)

หลังจากที่เราได้รับสัญญาณจากตัวรับสัญญาณแสงซึ่งเป็นสัญญาณพัลส์แล้วจะต้องแปลงสัญญาณดังกล่าวให้เป็นสัญญาณคลื่นไฟฟ้าหัวใจตามเดิม โดยการใช้วงจรดีมอดูเลเตอร์ ซึ่งวงจรที่ทำหน้าที่ดังกล่าวคือวงจรอินทิเกรเตอร์ (Integrator) และวงจรรองความถี่ต่ำที่ยอมให้ความถี่ในช่วงความถี่ของสัญญาณคลื่นไฟฟ้าหัวใจผ่านได้ดังแสดงในรูปที่ 3.9



รูปที่ 3.9 วงจรตีมอดูเลเตอร์

จากวงจรกรองความถี่ต่ำผ่านจะมีความถี่คัทออฟ (Cut off frequency) อยู่ที่ประมาณ 117 Hz
คำนวณได้จาก

$$f_c = \frac{1}{2\pi\sqrt{R_{41} \times R_{42} \times C_{11} \times C_{12}}}$$

$$= \frac{1}{2\pi\sqrt{20k \times 20k \times 0.068\mu \times 0.068\mu}}$$

$$= 117 \text{ Hz}$$

และสามารถคำนวณอัตราขยายของวงจรได้จาก

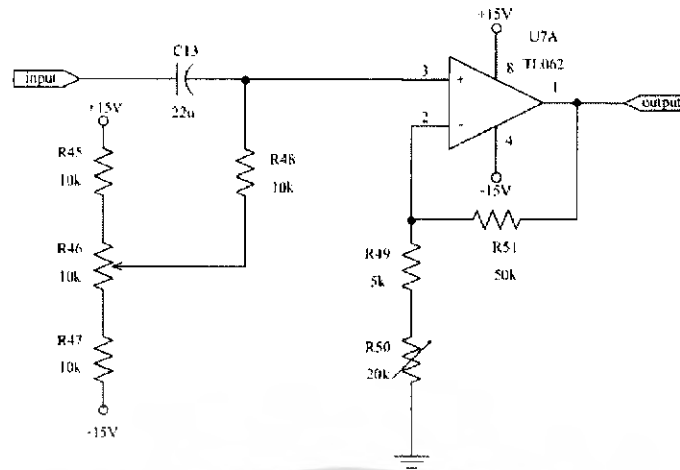
$$A_v = 1 + \left(\frac{R_{44}}{R_{43}}\right) = 1 + \left(\frac{3k}{3k}\right)$$

$$= 2 \text{ เท่า}$$

3.2.8 วงจรขยายและปรับระดับสัญญาณ (Amplifier and Clamp Circuit)

ทำหน้าที่ขยายสัญญาณไฟฟ้าหัวใจให้มีแรงดันเอาต์พุต 5 Vp-p และยกระดับสัญญาณอินพุตที่มีแรงดันต่ำกว่าศูนย์ให้มีแรงดันมากกว่าศูนย์เพื่อให้สัญญาณเหมาะสมที่จะป้อนให้แก่วงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัล

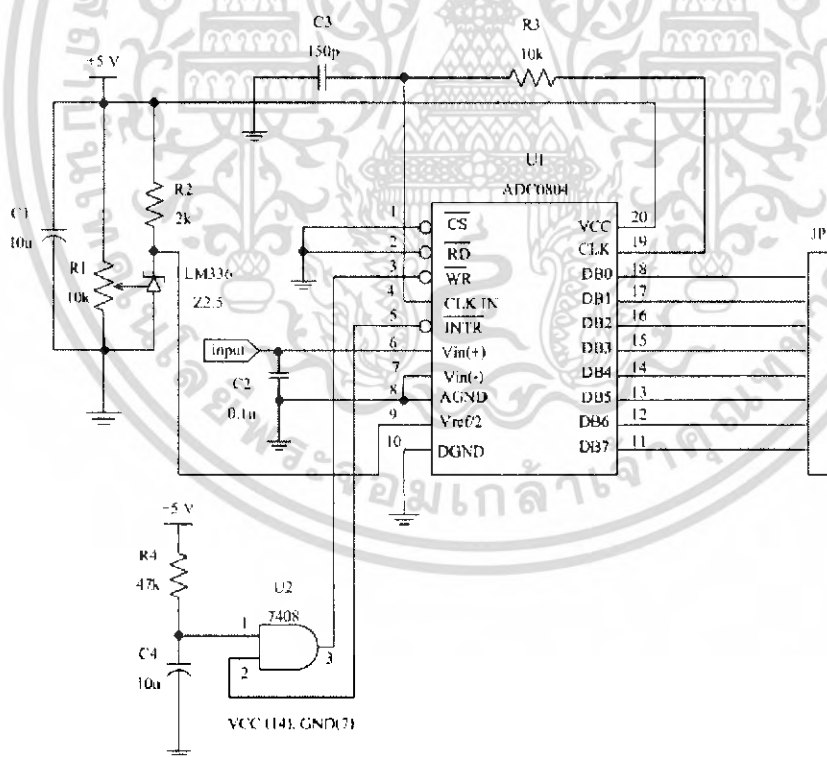
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 วงจรขยายและปรับระดับสัญญาณไฟฟ้าหัวใจ

3.2.9 วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล (Analog to Digital Converter)

ทำหน้าที่แปลงสัญญาณหัวใจที่เป็นสัญญาณอนาล็อกให้เป็นสัญญาณดิจิทัลโดยใช้ไอซีเบอร์ ADC0804 ซึ่งให้สัญญาณดิจิทัลขนาด 8 บิตส่งไปยังไมโครโปรเซสเซอร์ RCM 2200 โดยสัญญาณอนาล็อกที่ป้อนให้กับวงจรมันต้องปรับระดับแรงดันให้อยู่ในช่วง 0-5V โดยระดับแรงดันดังกล่าวสามารถปรับได้จากวงจรขยายและปรับระดับสัญญาณก่อนที่จะนำมาป้อนให้กับวงจร



รูปที่ 3.11 วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

เนื่องจากแบนวิดท์ของคลื่นไฟฟ้าหัวใจที่ได้จาก เครื่องซิมูเลเตอร์อยู่ในช่วงความถี่ประมาณ 0.5 – 100 Hz ดังแสดงในรูปที่ 4.1 จึงทำการทดสอบการทำงานของวงจรในบทที่3 โดยพิจารณาช่วงความถี่จาก เครื่องซิมูเลเตอร์เป็นเกณฑ์ในการทดลอง



รูปที่ 4.1 a สัญญาณคลื่นไฟฟ้าหัวใจ

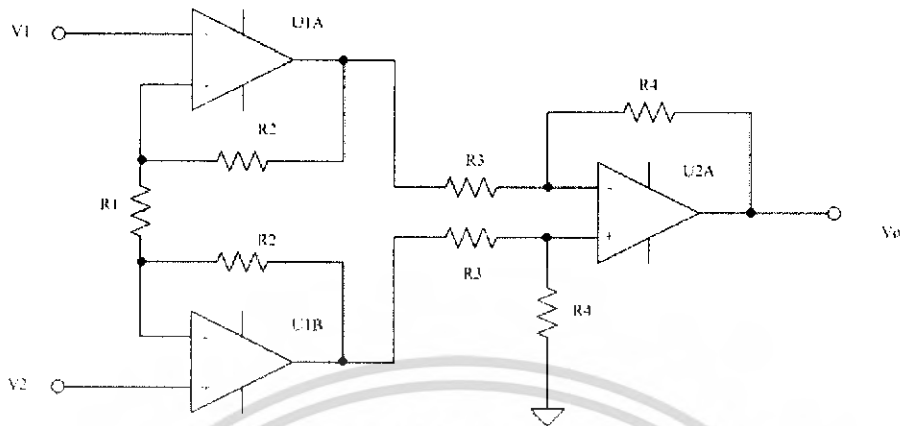


รูปที่ 4.1 b แสดงสเปกตรัมของคลื่นไฟฟ้าหัวใจ

รูปที่ 4.1 สัญญาณและสเปกตรัมของคลื่นไฟฟ้าหัวใจจากเครื่องซิมูเลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 ทดสอบการทำงานวงจรอินสตรูเมนแอมพลิไฟเออร์



รูปที่ 4.2 แสดงวงจรอินสตรูเมนแอมพลิไฟเออร์ในการทดสอบ

ทำการทดสอบโดยการหาผลตอบสนองทางความถี่ โดยป้อนสัญญาณไซน์ แรงดัน 100 mV_{p-p} ที่ความถี่ต่างๆ เข้าที่อินพุต V_1 และ ต่อ V_2 ลงกราวด์ วัดแรงดันเอาต์พุตว่าเป็นไปตามทฤษฎีหรือไม่ ต่อไปทำการสลับจุดการป้อนสัญญาณ คือต่อ V_1 ลงกราวด์ และป้อนสัญญาณเข้าที่ V_2 แทน วัดแรงดันเอาต์พุตว่าเป็นไปตามทฤษฎีหรือไม่เช่นเดียวกัน

โดยทางทฤษฎีหาแรงดันเอาต์พุตได้จากสูตร

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

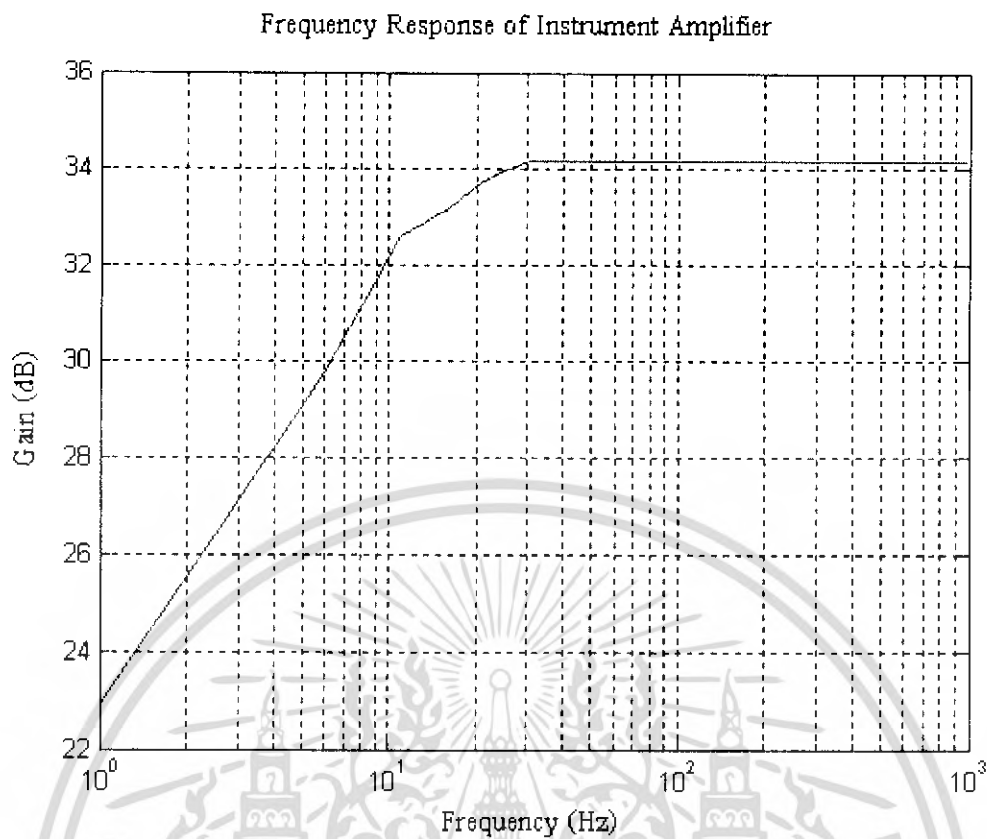
เมื่อ $R_4 = R_3 = R_2 = 50\text{ k}\Omega$ และ $R_1 = 2\text{ k}\Omega$

จากการทดลองจะเห็นว่าสัญญาณเอาต์พุตที่ได้ในกรณีแรกคือป้อนสัญญาณเข้าที่ V_1 และ V_2 ต่อลงกราวด์จะกลับเฟสกับสัญญาณอินพุต ส่วนในกรณีหลัง คือป้อนสัญญาณเข้าที่ V_2 และ V_1 ต่อลงกราวด์ จะได้สัญญาณเอาต์พุตที่มีเฟสตรงกับสัญญาณอินพุตซึ่งตรงตามทฤษฎี ส่วนผลการวัดแรงดันเอาต์พุตแสดงดังตารางที่ 4.1

Frequency (Hz)	$V_{out} (V_{p-p})$	Gain= $20\log\frac{V_{out}}{V_{in}}$
1	1.40	22.92
2	1.48	23.40
5	3.08	29.77
10	4.28	32.63
20	4.88	33.76
30	5.12	34.18
40	5.12	34.18
50	5.12	34.18
60	5.12	34.18
70	5.12	34.18
80	5.12	34.18
90	5.12	34.18
100	5.12	34.18
200	5.12	34.18
400	5.12	34.18
600	5.12	34.18
800	5.12	34.18
1000	5.12	34.18

ตารางที่ 4.1 ผลการทดลองการวัดแรงดันทางเอาต์พุตของวงจรอินสตรูเมนแอมพลิไฟเออร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

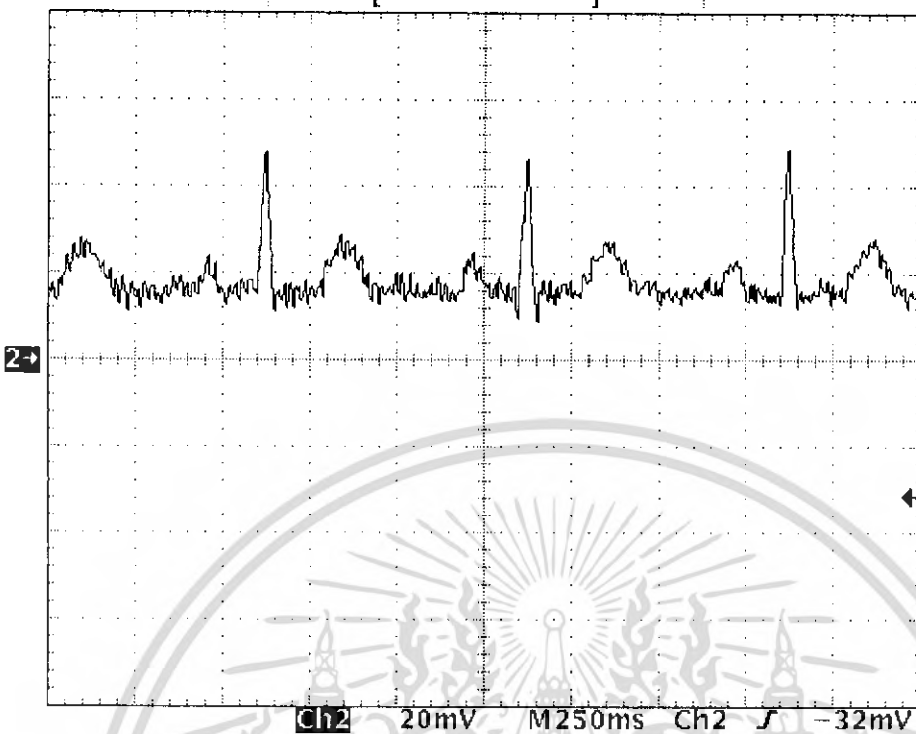


รูปที่ 4.3 กราฟแสดงผลตอบสนองทางความถี่ของวงจรอินสตรูเมนแอมพลิไฟเออร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Tek Stop: 200 S/s

7 Acqs



Ch2 Freq
1.332 Hz
Low signal
amplitude

Ch2 Pk-Pk
44.8mV

2→

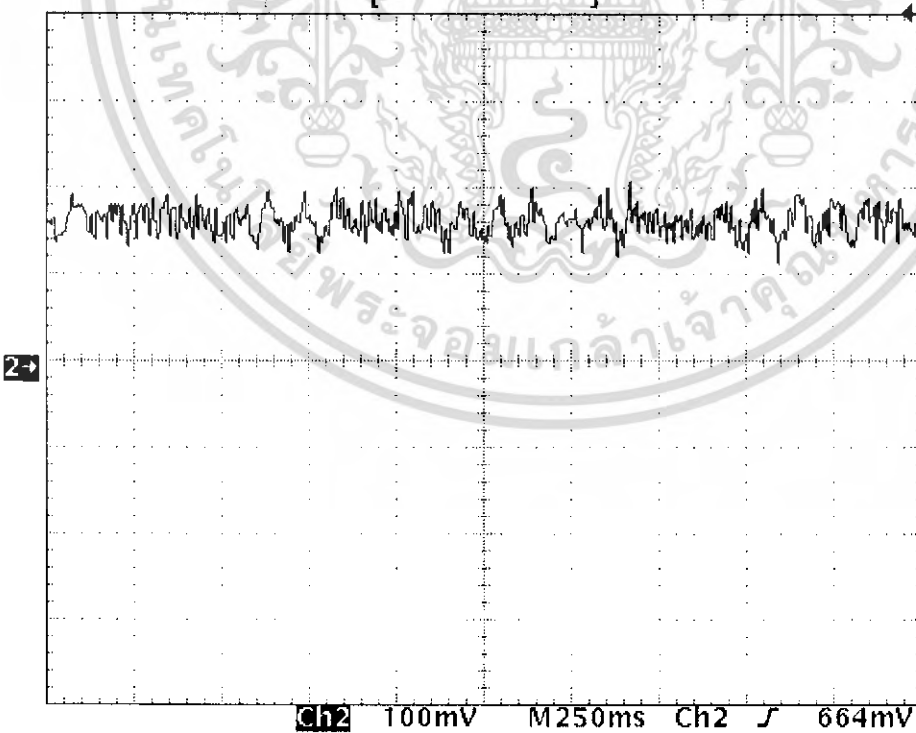
ch2 20mV M250ms Ch2 32mV

18 Mar 2006
19:26:56

รูปที่ 4.4 ผลที่วัดได้จากเครื่องจิมมูเลเตอร์ของวงจรอินสตรูเมนแอมพลิไฟเออร์

Tek Stop: 200 S/s

1 Acqs



Ch2 Freq
 ∞ Hz
No ref
crossing

Ch2 Pk-Pk
96mV

2→

ch2 100mV M250ms Ch2 664mV

24 Jan 2006
23:19:38

รูปที่ 4.5 ผลที่วัดได้จากการวัดจริงของวงจรอินสตรูเมนแอมพลิไฟเออร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

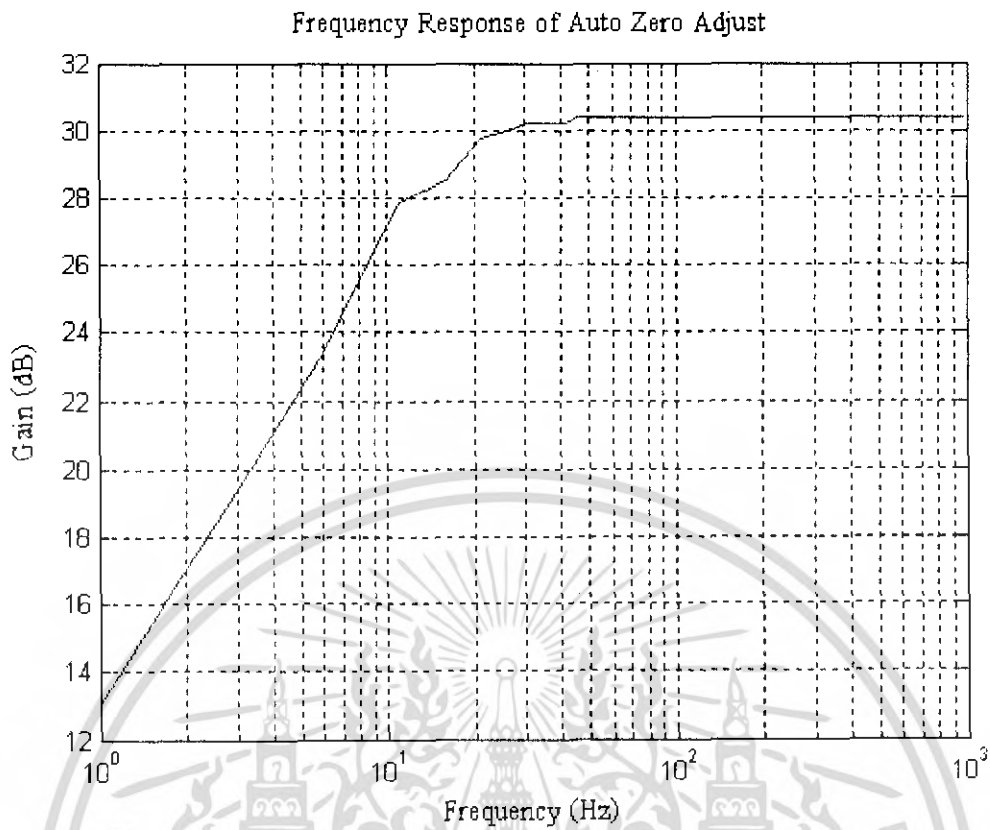
4.2 ทดสอบการทำงานวงจรปรับศูนย์อัตโนมัติ (Auto-Zero Circuit)

ทำการทดสอบโดยการนำสัญญาณคลื่นไซน์ขนาด 100 mVp-p ความถี่ 100 Hz ที่มีแรงดันไฟฟ้าออฟเซตปนอยู่ด้วยเข้าสู่วงจรปรับศูนย์อัตโนมัติ โดยวงจรมีอัตราขยาย 33.33 เท่า จะเห็นได้ว่ารูปสัญญาณเกิดการลอยตัวออกจากระดับแรงดันอ้างอิงไป แต่เมื่อวัดทางด้านเอาต์พุตปรากฏว่ารูปสัญญาณกลับมามีระดับแรงดันอ้างอิง

Frequency (Hz)	Vout (Vp-p)	Gain=20log $\frac{V_{out}}{V_{in}}$
1	0.45	13.06
2	0.48	13.62
5	1.48	23.40
10	2.48	27.88
20	3.08	29.77
30	3.24	30.21
40	3.24	30.21
50	3.28	30.31
60	3.32	30.42
70	3.32	30.42
80	3.32	30.42
90	3.32	30.42
100	3.32	30.42
200	3.32	30.42
400	3.32	30.42
600	3.32	30.42
800	3.32	30.42
1000	3.32	30.42

ตารางที่ 4.2 แสดงผลตอบสนองทางความถี่ของวงจรปรับศูนย์อัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

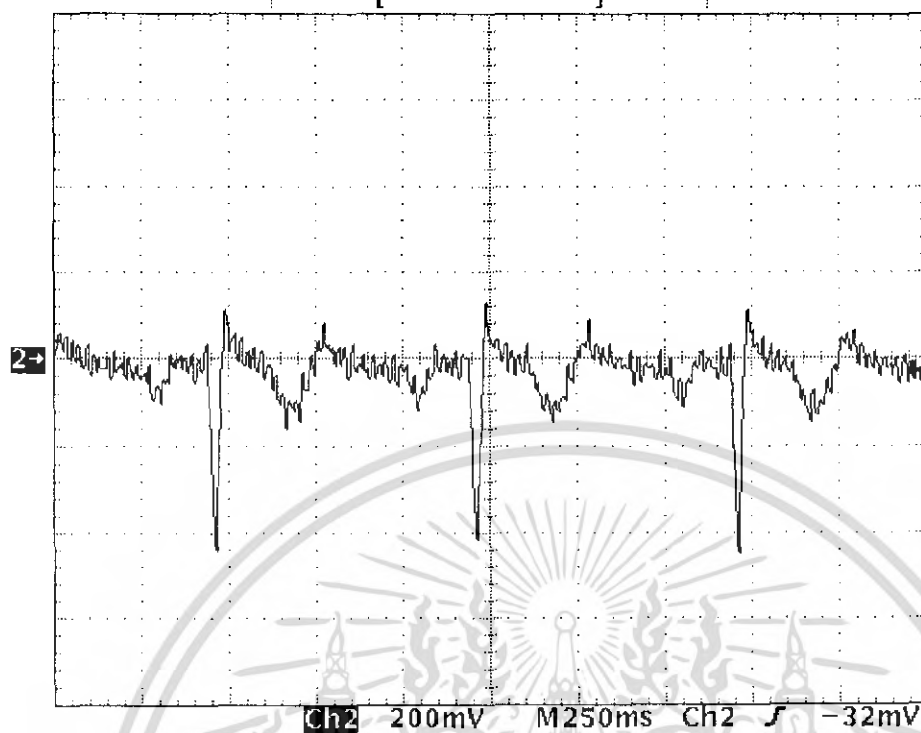


รูปที่ 4.6 กราฟแสดงผลตอบสนองทางความถี่ของวงจรปรับศูนย์อัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Tek Stop: 200 S/s

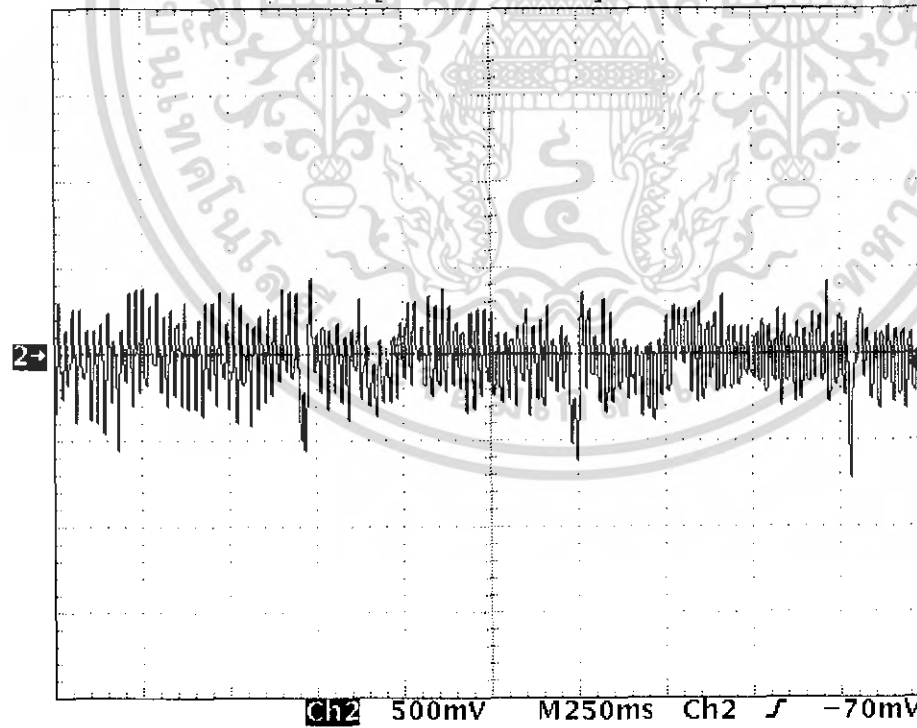
15 Acqs

18 Mar 2006
19:29:17

รูปที่ 4.7 ผลที่วัดได้จากเครื่องซิมูเลเตอร์ของวงจรปรับศูนย์อัตโนมัติ

Tek Stop: 200 S/s

3 Acqs

18 Mar 2006
21:02:23

รูปที่ 4.8 ผลที่วัดได้จากการวัดจริงของวงจรปรับศูนย์อัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ทดสอบการทำงานวงจรกรองความถี่แบบกั้นไม่ให้ความถี่ผ่านเฉพาะช่วง (Notch Filter)

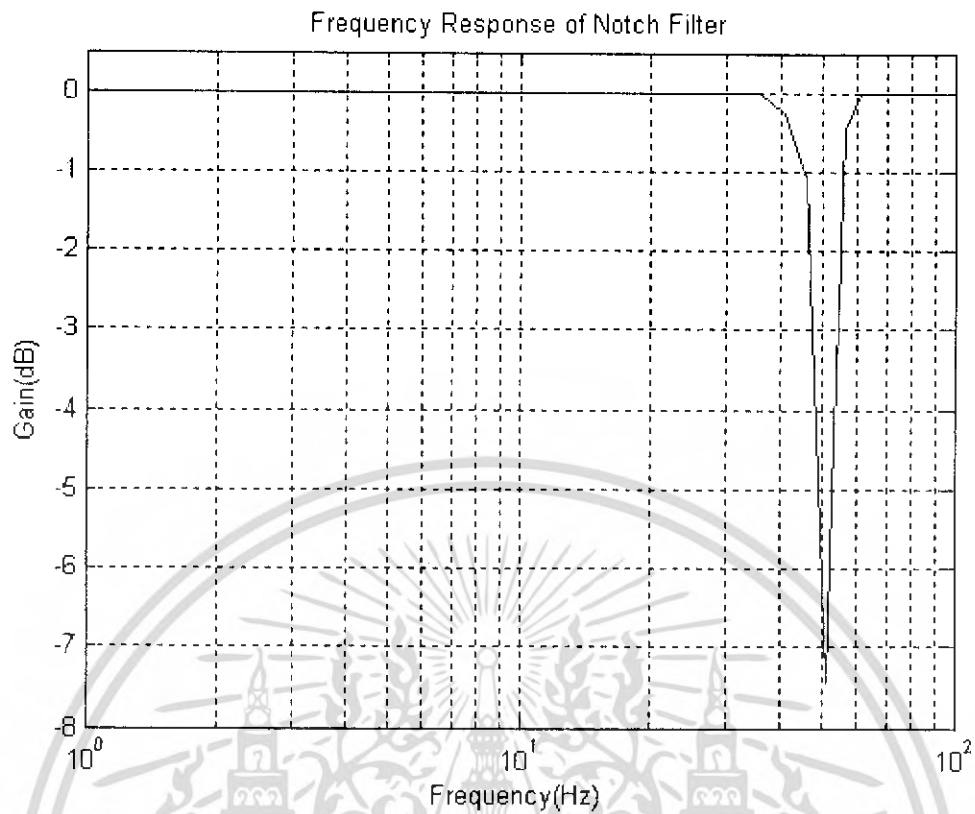
ทดสอบโดยการหาผลตอบสนองทางความถี่ต่างๆ โดยมีความถี่ศูนย์กลางที่ 50 Hz ในการทดลองทำได้ โดยป้อนสัญญาณคลื่นไซน์ขนาด 2 V วัดค่าแรงดันเอาต์พุตที่ได้ จากวงจรไปใส่ในตารางที่ 4.3 โดยเปลี่ยนความถี่ตามตารางที่ 4.3 ซึ่งนำผลที่ได้จากการทดลองจริงที่ได้ไปพล็อตกราฟ

ความถี่ (Hz)	เอาต์พุต (Vp-p)	20log Vo/Vin (dB)
5	2	0
10	2	0
15	2	0
20	2	0
25	2	0
30	2	0
35	2	0
40	1.88	-0.27
45	1.56	-1.08
50	0.36	-7.44
55	1.8	-0.46
60	2	0
65	2	0
70	2	0
80	2	0
90	2	0
100	2	0

ตารางที่ 4.3 แสดงผลตอบสนองทางความถี่ของวงจรกรองความถี่แบบกั้นไม่ให้ความถี่ผ่านเฉพาะช่วง จากตารางที่ 4.3 ที่ความถี่ 50 Hz สามารถหาอัตราขยายได้ดังนี้

$$\begin{aligned}
 \text{Gain}(dB) &= 20\log \frac{V_{out}}{V_{in}} \\
 &= 20\log \frac{0.36}{2} \\
 &= -7.44dB
 \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

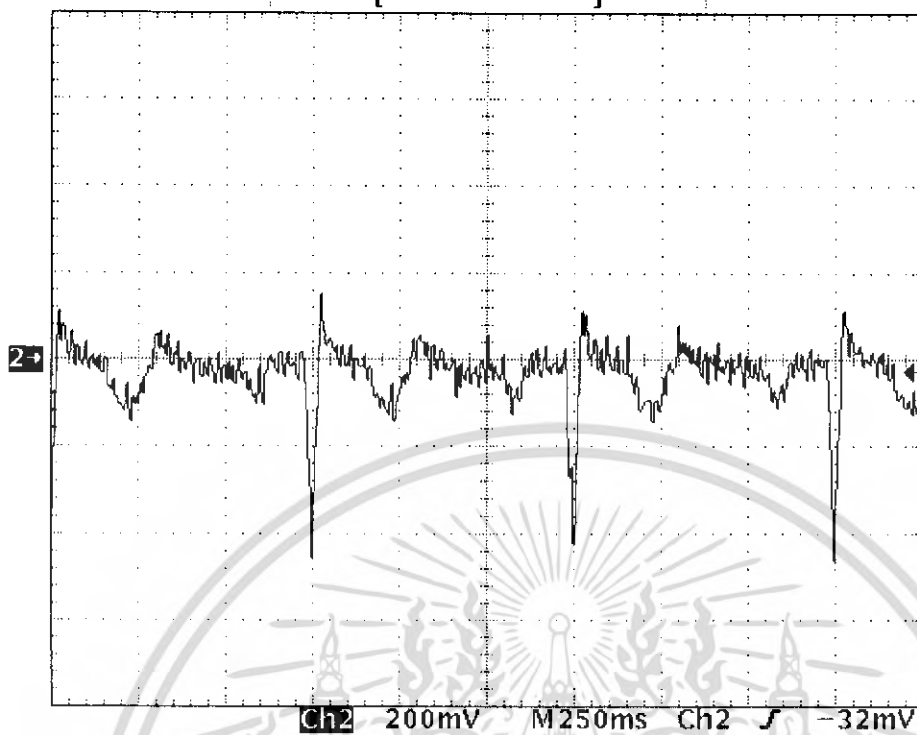


รูปที่ 4.9 กราฟแสดงผลตอบสนองทางความถี่ของวงจรกรองความถี่แบบกั้นไม่ให้ความถี่ผ่านเฉพาะช่วง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Tek Stop: 200 S/s

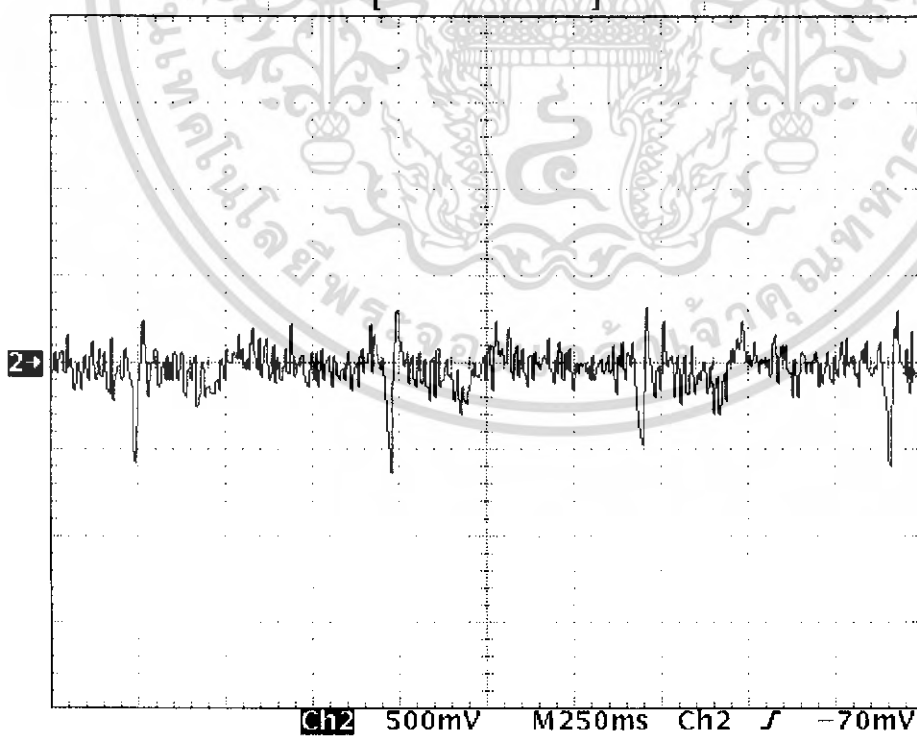
6 Acqs

18 Mar 2006
19:30:44

รูปที่ 4.10 ผลที่วัดได้จากเครื่องขมูเลเตอร์ของวงจรรองความถี่แบบกันไม่ให้ความถี่ผ่านเฉพาะช่วง

Tek Stop: 200 S/s

5 Acqs

18 Mar 2006
21:04:00

รูปที่ 4.11 ผลที่วัดได้จากกรวัดจริงของวงจรรองความถี่แบบกันไม่ให้ความถี่ผ่านเฉพาะช่วง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

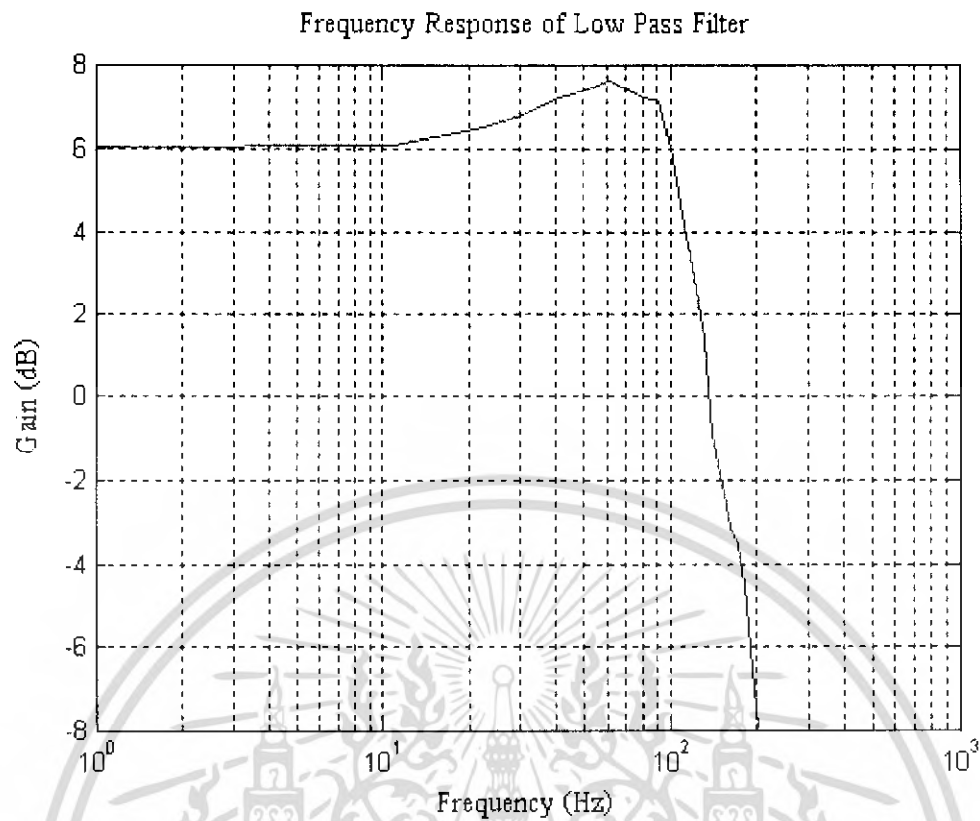
4.4 ทดสอบการทำงานวงจรกรองความถี่ต่ำผ่าน (Low Pass Filter)

การทดสอบวงจรกรองความถี่ต่ำผ่าน ที่มี Frequency cutoff = 117 Hz และมีอัตราขยาย 2 เท่าโดยป้อนสัญญาณอินพุตคลื่นไซน์ขนาด 1 V แสดงผลการทดสอบดังตารางที่ 4.4

Frequency (Hz)	Vout (Vp-p)	Gain=20log $\frac{V_{out}}{V_{in}}$
1	2	6.02
2	2	6.02
5	2	6.02
20	2.1	6.44
40	2.3	7.23
60	2.4	7.60
80	2.3	7.23
100	2	6.02
120	1.4	2.9
140	1.1	0.82
160	0.9	-0.92
180	0.7	-3.09
200	0.6	-4.44
220	0.5	-6.02
240	0.4	-7.96

ตารางที่ 4.4 แสดงผลตอบสนองทางความถี่ของวงจรกรองความถี่ต่ำผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

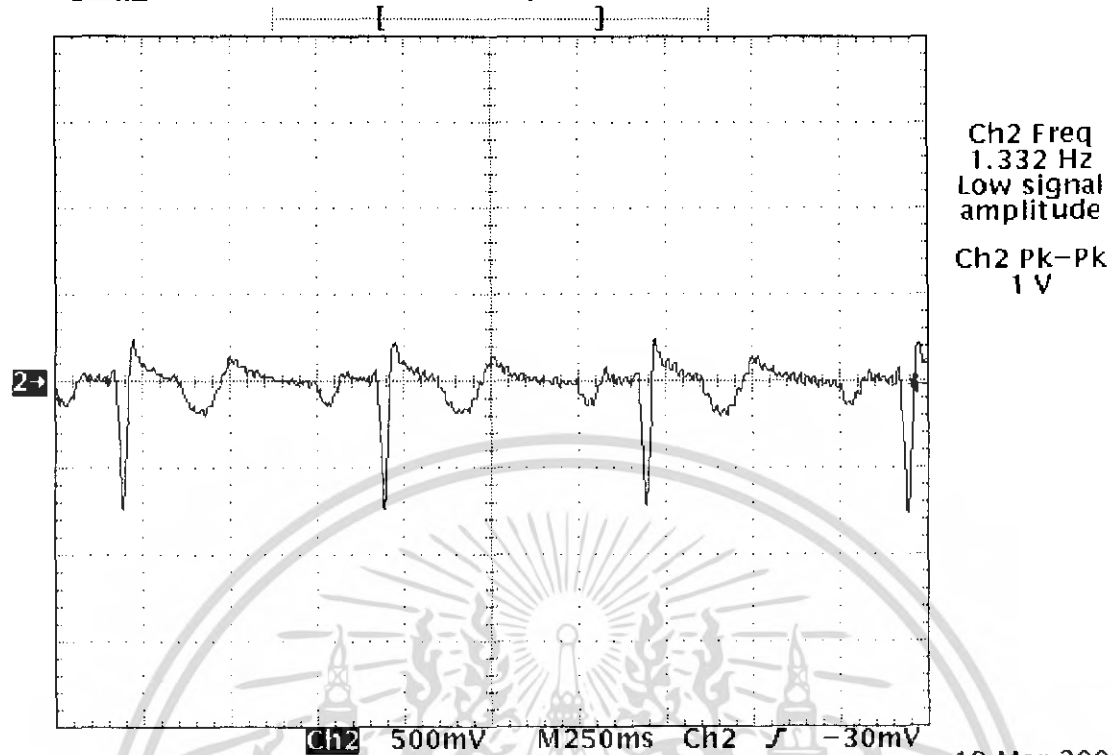


รูปที่ 4.12 กราฟแสดงผลตอบสนองทางความถี่ของวงจรกรองความถี่ต่ำผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Tek Stop: 200 S/s

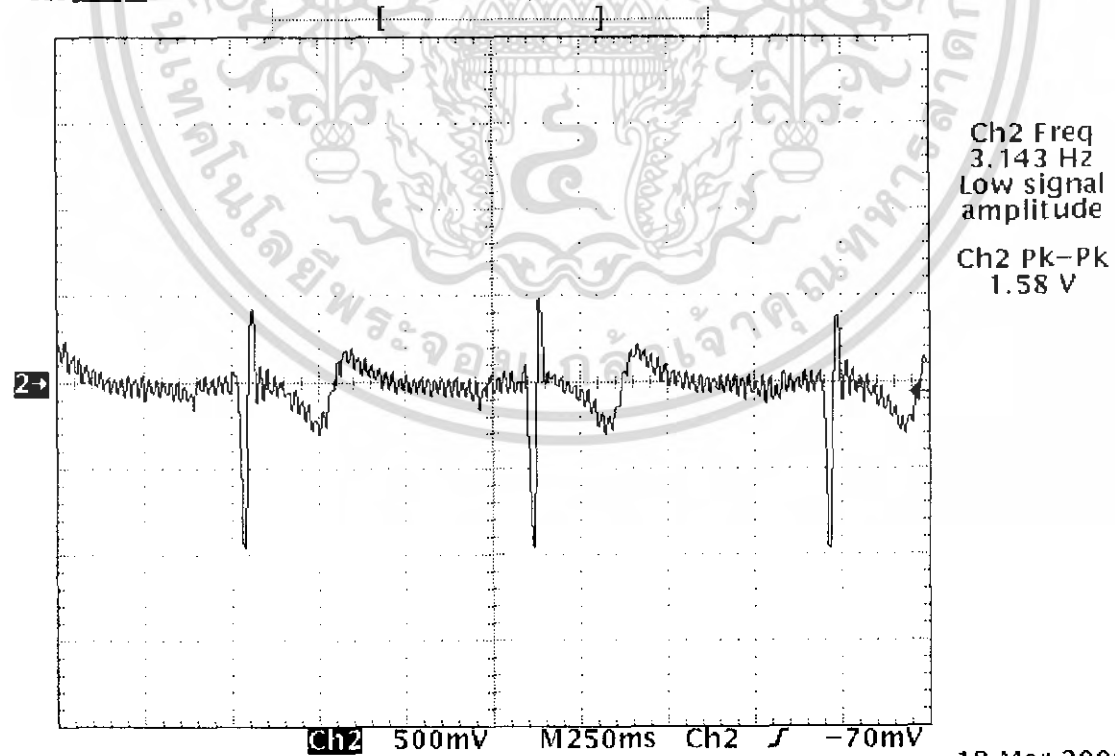
3 Acqs

18 Mar 2006
19:31:53

รูปที่ 4.13 ผลที่วัดได้จากเครื่องซินมูเลเตอร์ของวงจรกรองความถี่ต่ำผ่าน

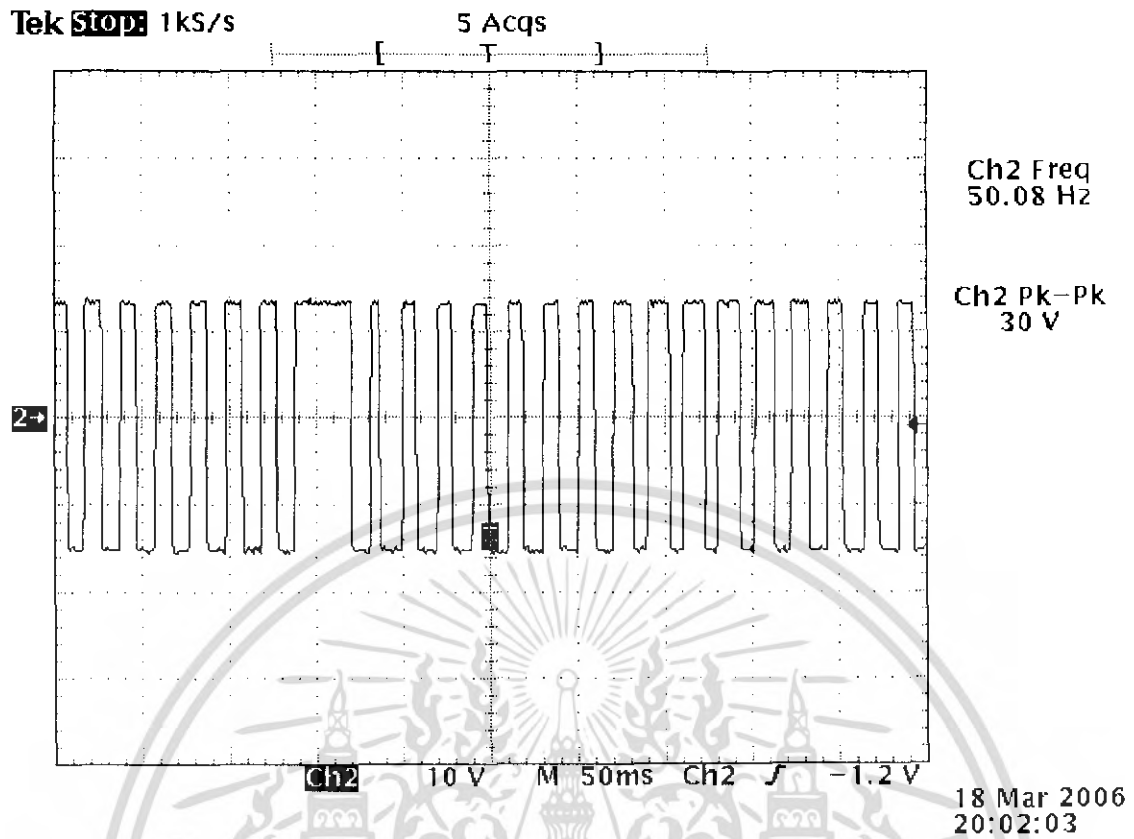
Tek Stop: 200 S/s

2 Acqs

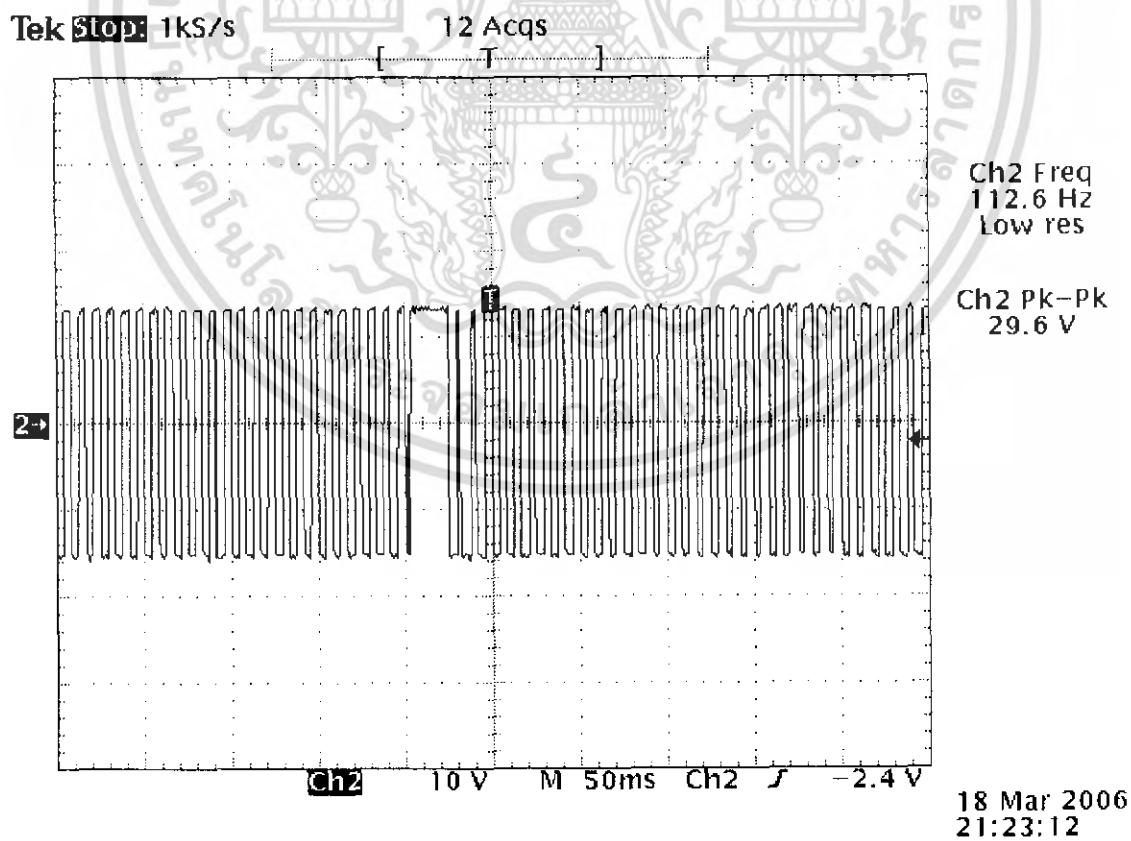
18 Mar 2006
21:05:10

รูปที่ 4.14 ผลที่วัดได้จากการวัดจริงของวงจรกรองความถี่ต่ำผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 ผลที่วัดได้จากเครื่องซิมูเลเตอร์ของวงจรพัลส์-วิดท์ มอดูเลเตอร์



รูปที่ 4.17 ผลที่วัดได้จากการวัดจริงของวงจรพัลส์-วิดท์ มอดูเลเตอร์

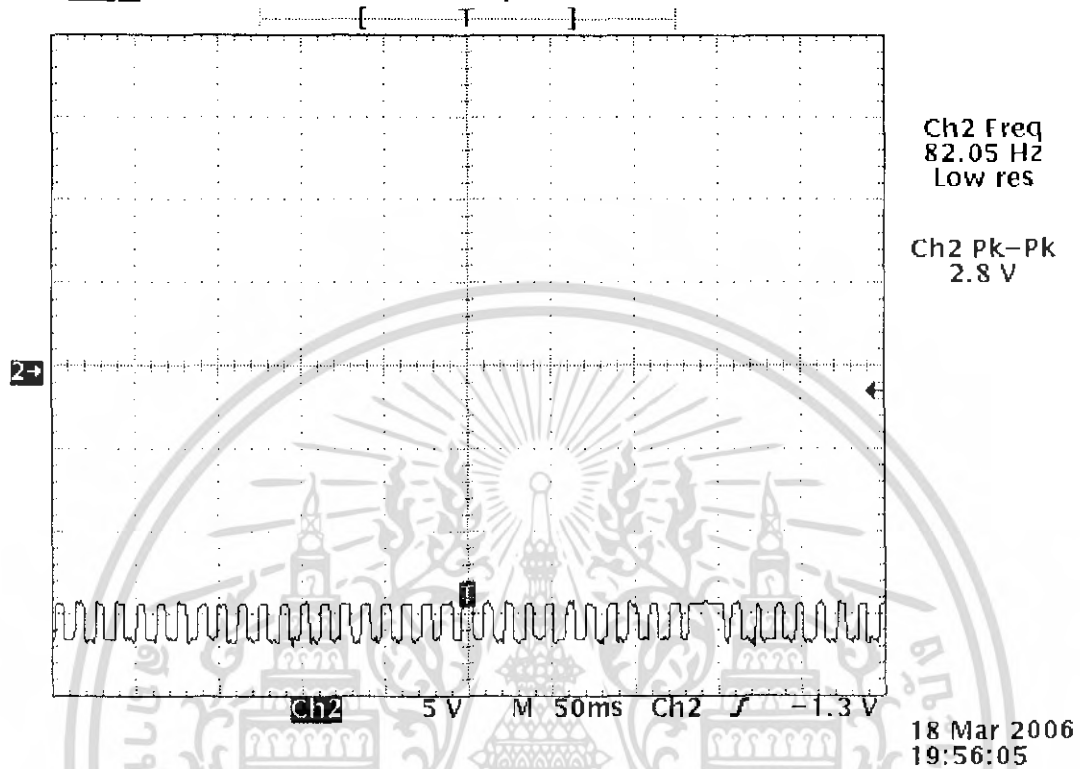
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 ทดสอบการทำงานวงจรส่งผ่านสัญญาณด้วยแสง (Opto Isolator)

เมื่อสัญญาณ PWM ป้อนเข้าไปที่อินพุตของวงจรส่งผ่านสัญญาณด้วยแสงเอาท์พุทที่วัดได้จากขา 4 ของ ISO1 4N26 จะได้สัญญาณในช่วงลบ ไม่กลับเฟส

Tek **Stop** 1ks/s

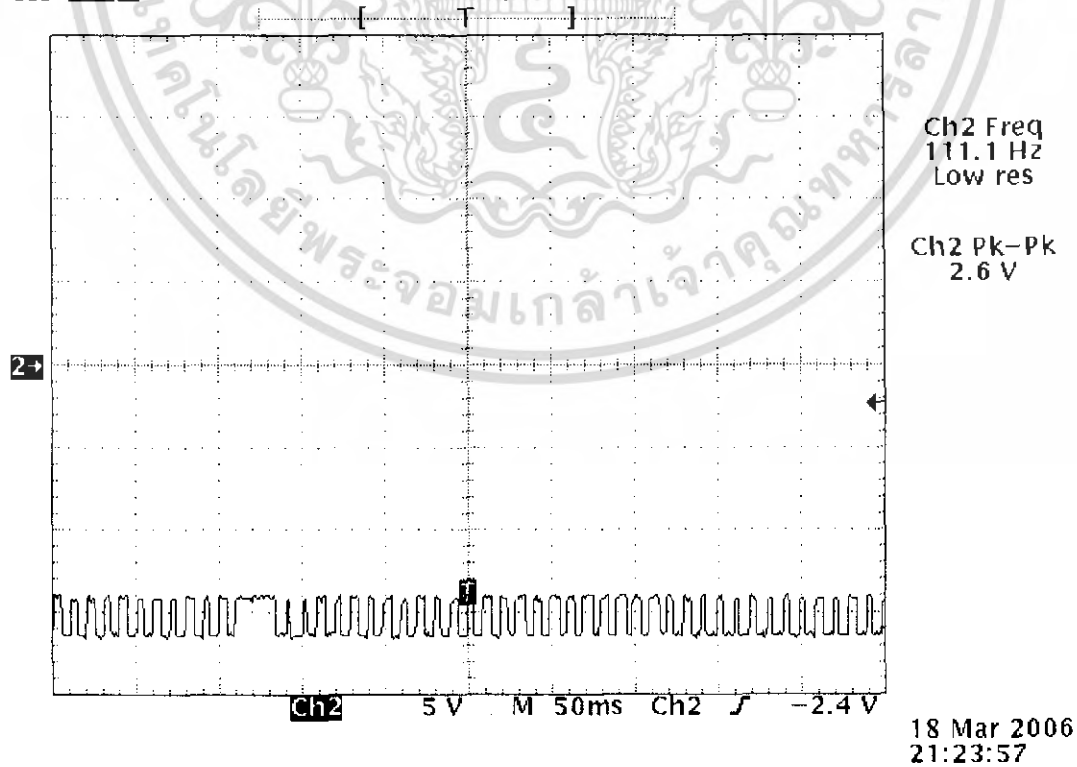
6 Acqs



รูปที่ 4.18 ผลที่วัดได้จากเครื่องซิมูเลเตอร์ของวงจรส่งผ่านสัญญาณด้วยแสง

Tek **Stop** 1ks/s

5 Acqs



รูปที่ 4.19 ผลที่วัดได้จากการวัดจริงของวงจรส่งผ่านสัญญาณด้วยแสง

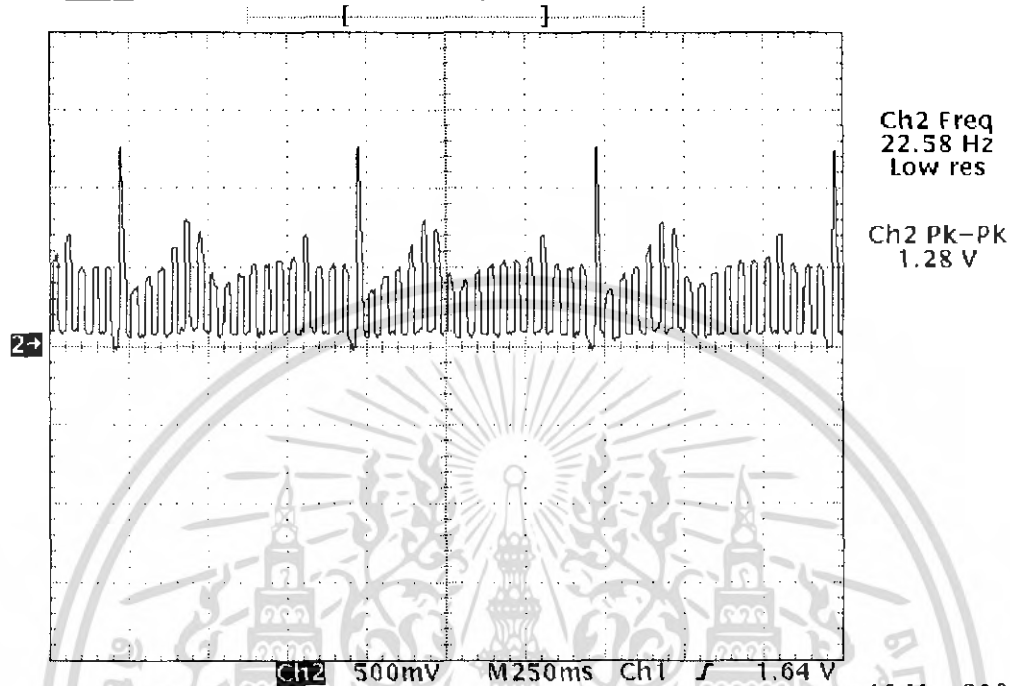
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 ทดสอบการทำงานวงจรดีมอดูเลเตอร์ (Demodulator)

วงจรดีมอดูเลเตอร์ประกอบด้วย 2 ส่วนคือ วงจรอินทรีเกรเตอร์ และวงจรกรองความถี่ต่ำผ่าน เพื่อนำสัญญาณคลื่นไฟฟ้าหัวใจที่ถูก มอดูเลตกลับคืนมา

Tek **STOP** 200 S/s

3 Acqs

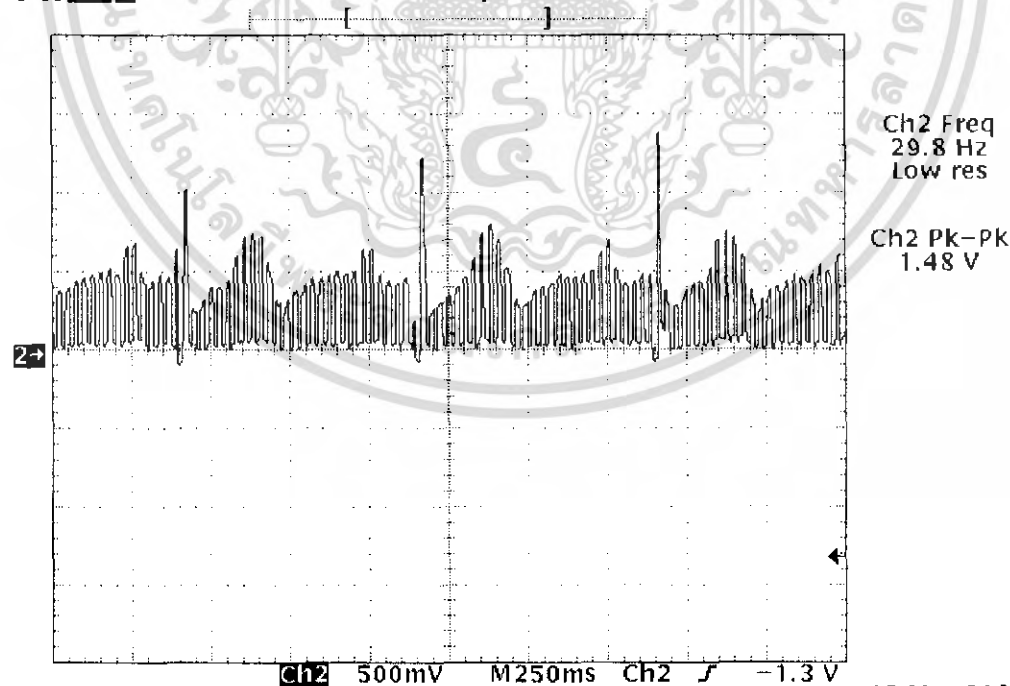


16 Mar 2006
00:42:53

รูปที่ 4.20 ผลที่วัดได้จากเครื่องซิมูเลเตอร์ของวงจรอินทรีเกรเตอร์

Tek **STOP** 200 S/s

6 Acqs



18 Mar 2006
19:57:35

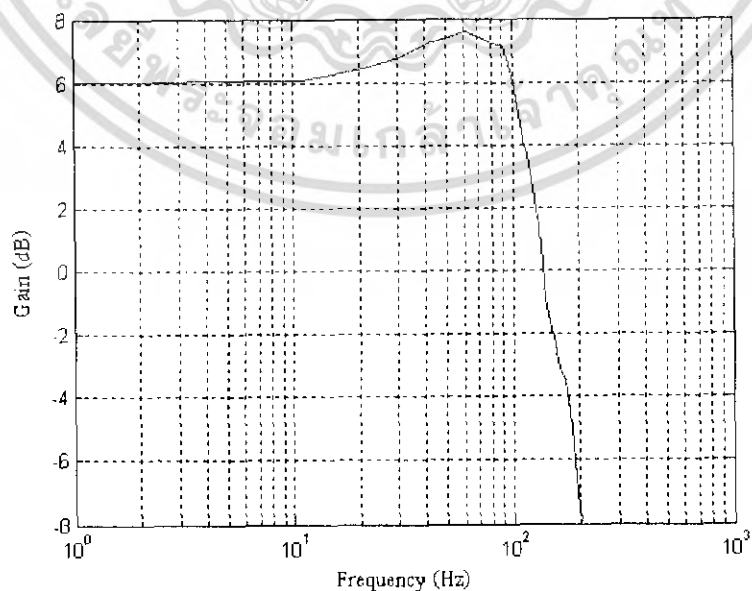
รูปที่ 4.21 ผลที่วัดได้จากการวัดจริงของวงจรอินทรีเกรเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทดสอบวงจรกรองความถี่ต่ำผ่าน ที่มี Frequency cutoff = 117 Hz และมีอัตราขยาย 2 เท่า โดยป้อนสัญญาณอินพุตคลื่นไซน์ขนาด 1 V แสดงผลการทดลองดังตารางที่ 4.4

Frequency (Hz)	Vout (Vp-p)	Gain=20log $\frac{V_{out}}{V_{in}}$
1	2	6.02
2	2	6.02
5	2	6.02
20	2.1	6.44
40	2.3	7.23
60	2.4	7.60
80	2.3	7.23
100	2	6.02
120	1.4	2.9
140	1.1	0.82
160	0.9	-0.92
180	0.7	-3.09
200	0.6	-4.44
220	0.5	-6.02
240	0.4	-7.96

ตารางที่ 4.5 แสดงผลตอบสนองทางความถี่ของวงจรกรองความถี่ต่ำผ่าน
Frequency Response of Low Pass Filter

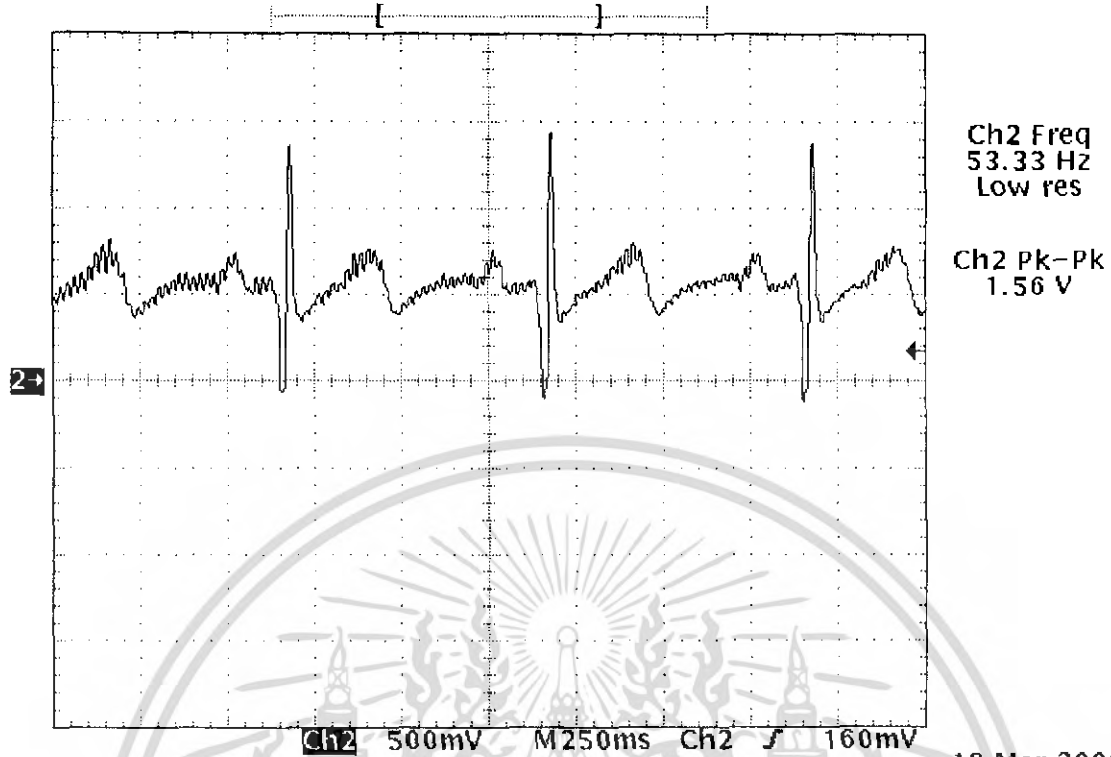


รูปที่ 4.22 กราฟแสดงผลตอบสนองทางความถี่ของวงจรกรองความถี่ต่ำผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Tek Stop: 200 S/s

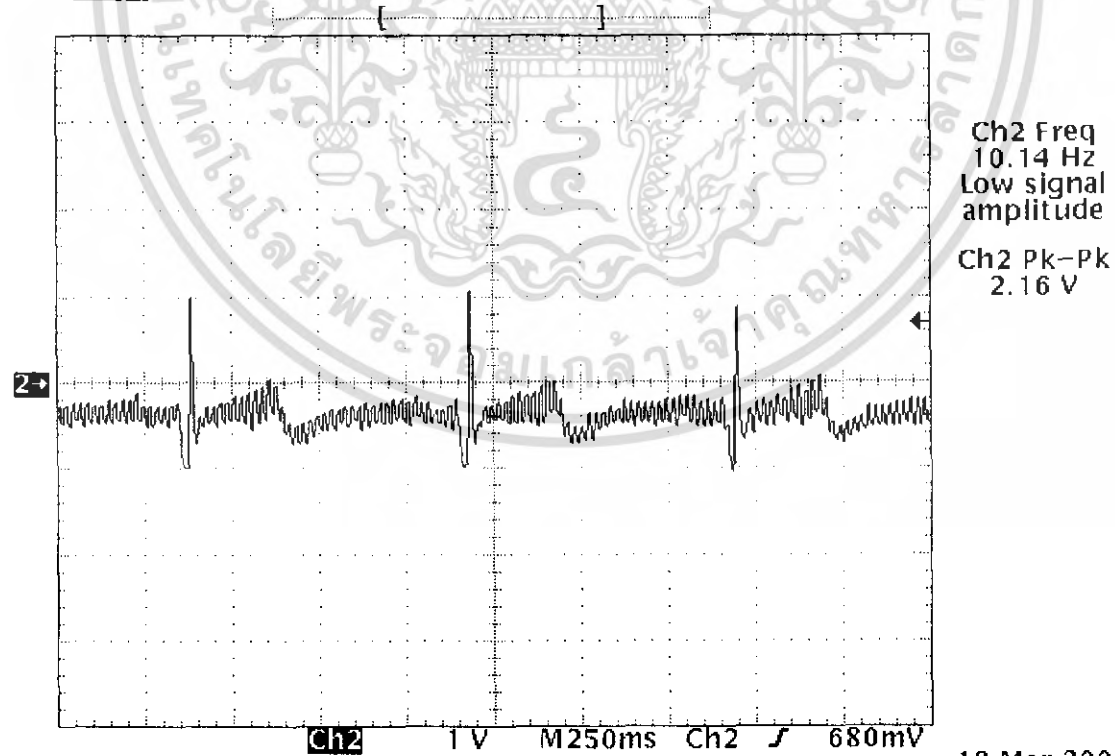
3 Acqs

18 Mar 2006
19:59:23

รูปที่ 4.23 ผลที่วัดได้จากเครื่องซิมูเลเตอร์ของวงจรกรองความถี่ต่ำผ่าน

Tek Stop: 200 S/s

1 Acqs

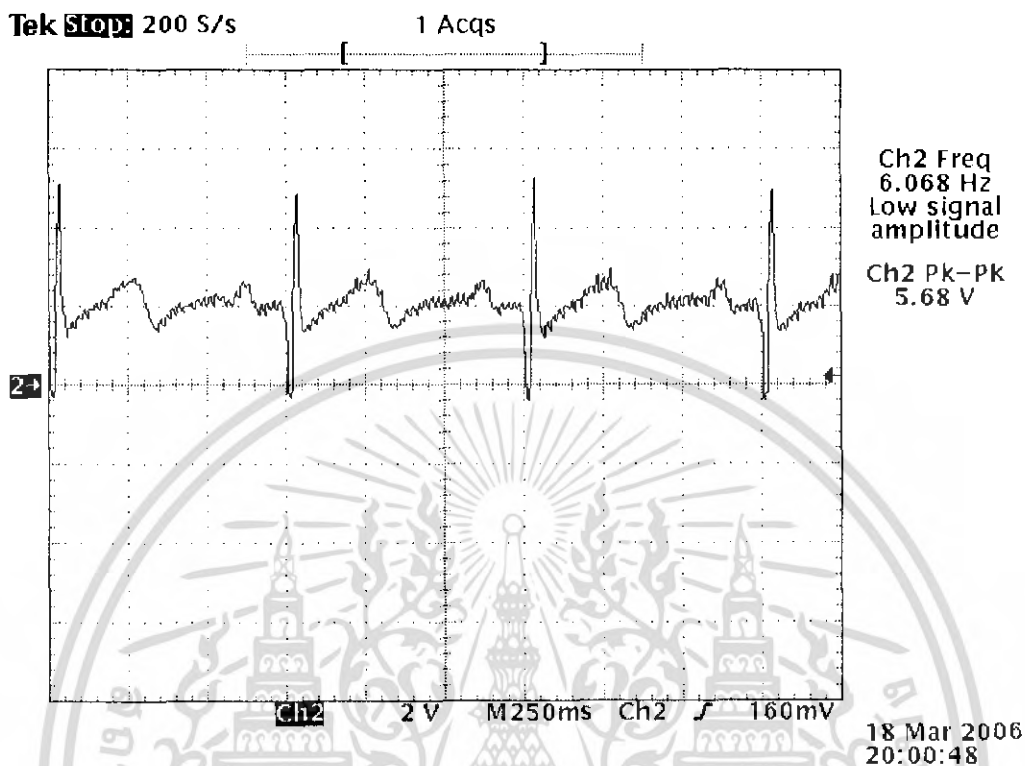
18 Mar 2006
21:25:25

รูปที่ 4.24 ผลที่วัดได้จากการวัดจริงของวงจรกรองความถี่ต่ำผ่าน

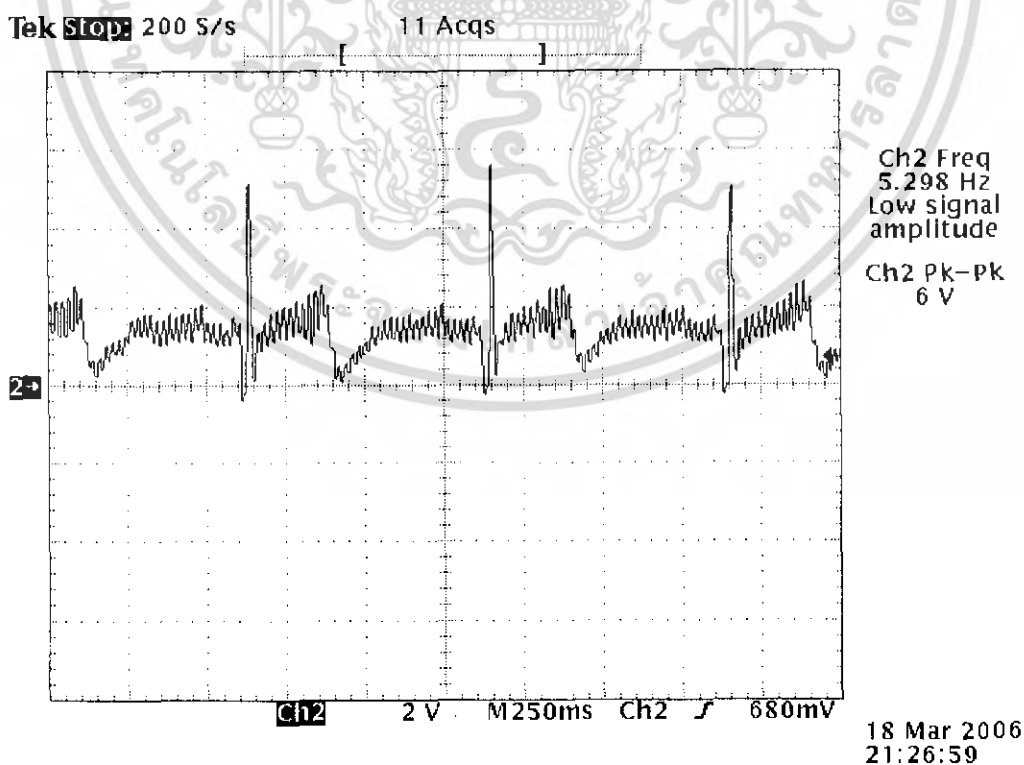
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 ทดสอบการทำงานวงจรขยายและปรับระดับสัญญาณ (Amplifier and Clamp Circuit)

เราจะปรับค่าแรงดันของสัญญาณคลื่นไฟฟ้าหัวใจให้มีขนาด 5V และยกระดับสัญญาณให้อยู่เฉพาะซี่กบวกเท่านั้น



รูปที่ 4.25 ผลที่วัดได้จากเครื่องซิมูเลเตอร์ของวงจรขยายและปรับระดับสัญญาณ



รูปที่ 4.26 ผลที่วัดได้จากการวัดจริงของวงจรขยายและปรับระดับสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.9 ทดสอบการทำงานของวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล

ทำการทดสอบโดยการป้อนแรงดันไฟตรงที่ระดับแรงดันต่างๆ แล้ววัดสัญญาณที่ขาเอาต์พุตทั้ง 8 ขา ซึ่งเราจะแทนแรงดันเอาต์พุตที่ระดับสูงด้วยรหัส 1 และแทนแรงดันเอาต์พุตที่ระดับต่ำด้วยรหัส 0 ผลที่ได้แสดงดังตารางที่ 4.6

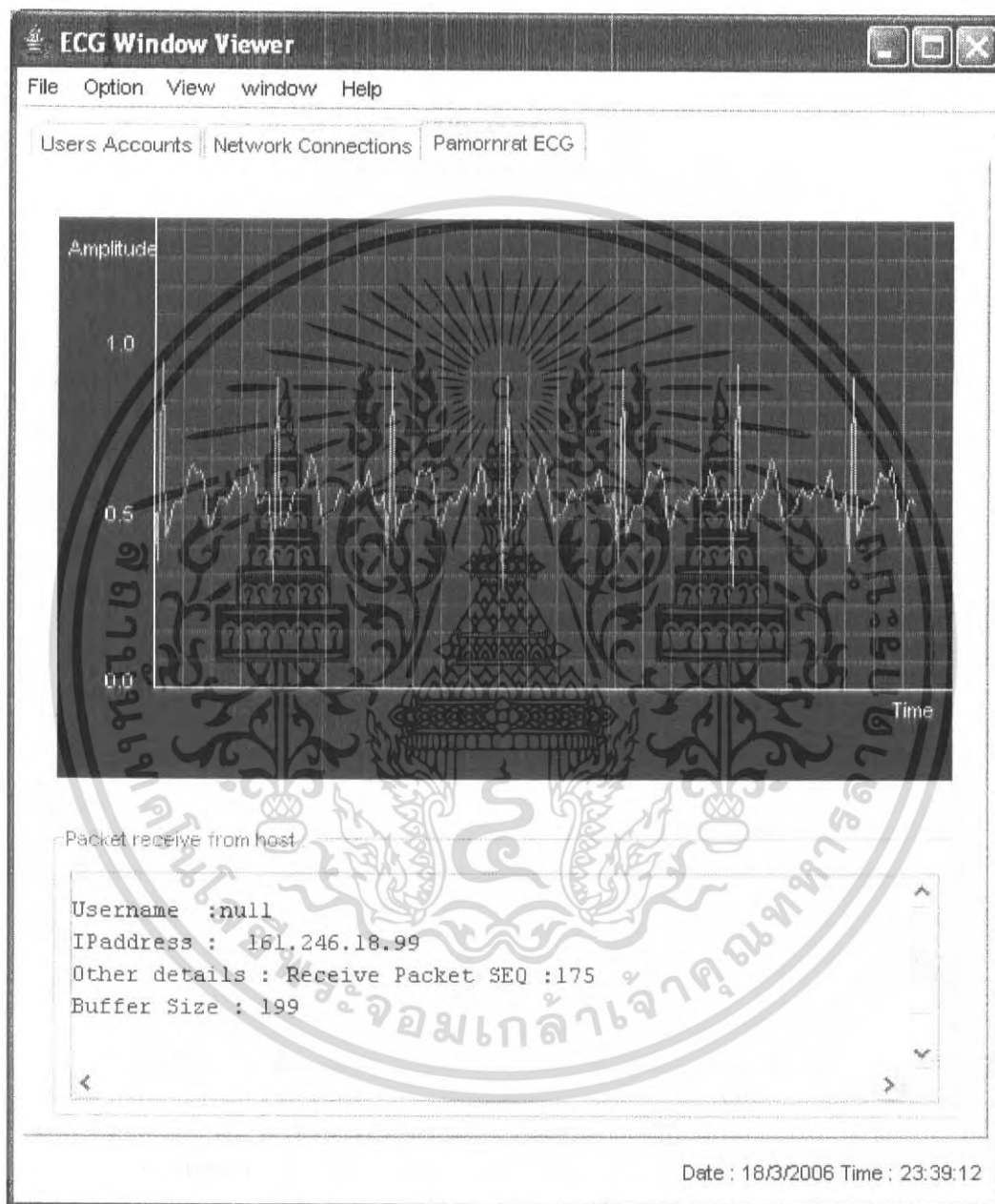
ระดับแรงดันไฟตรง (Volt)	รหัสที่ได้จากขาเอาต์พุต
0	0000 0000
0.5	0010 0101
1	0011 1011
1.5	0101 1000
2	0111 0010
2.5	1000 1110
3	1010 1001
3.5	1100 1000
4	1110 0110
4.5	1111 1000
5	1111 1111

ตารางที่ 4.6 ผลที่ได้จากการทดลองวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

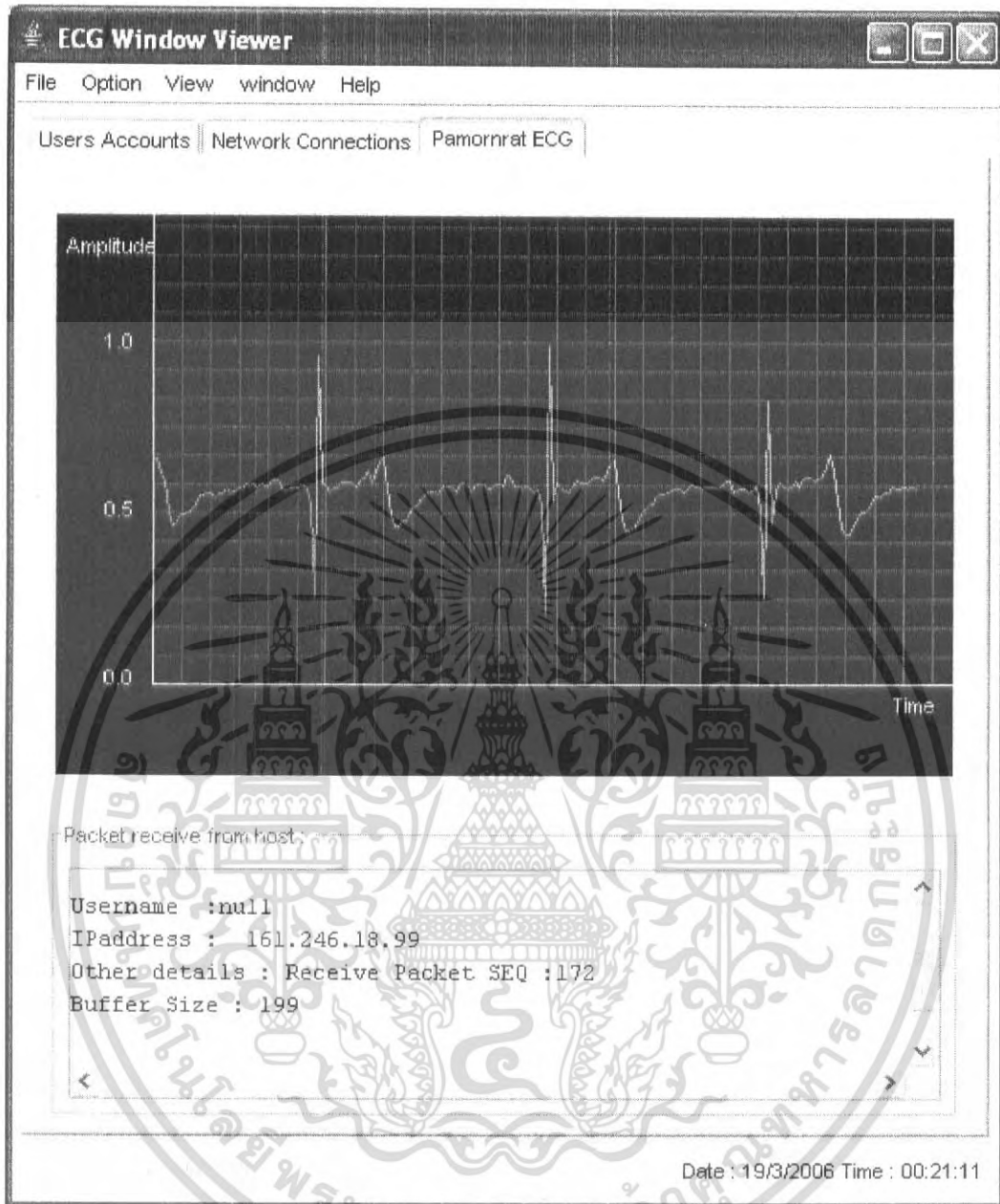
4.10 ผลการทดลองที่ได้จากการส่งข้อมูลผ่านเครือข่ายคอมพิวเตอร์

โปรแกรมแสดงผลการรับข้อมูล สัญญาณคลื่นไฟฟ้าหัวใจจากไมโครโปรเซสเซอร์ RCM 2200 ผ่านเครือข่ายคอมพิวเตอร์



รูปที่ 4.27 แสดงผลสัญญาณคลื่นไฟฟ้าหัวใจจากเครื่องซิมูเลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.28 แสดงผลสัญญาณคลื่นไฟฟ้าหัวใจจากการวัดจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 บทวิจารณ์และบทสรุป

การที่ได้ทำการทดลองวงจรเครื่องวัดคลื่นไฟฟ้าหัวใจสามารถสรุปแยกออกเป็นวงจรต่างๆ ได้ดังนี้

- วงจรอินสตรูเมนแอมพลิไฟเออร์ ได้เลือกใช้ออปแอมป์ที่มีอินพุทอิมพีแดนซ์สูงและมีสัญญาณรบกวน ต่ำมาสร้างเป็นวงจรทดลองหาค่าผลตอบสนองทางความถี่ในช่วง 0.5-250 Hz จะเห็นว่าผลตอบสนองทางความถี่ในช่วง 0.5-30 Hz ให้ผลตอบสนองที่ต่ำกว่าทางทฤษฎีอยู่ ส่วนช่วงความถี่ที่มากกว่านี้จะได้อัตราขยายที่คงที่ตรงตามทฤษฎี และจากการ วัดจริงสัญญาณที่ได้จะมีขนาดเล็กและมีสัญญาณรบกวนอยู่
- วงจรปรับศูนย์อัตโนมัติ จากการทดลองป้อนสัญญาณที่มีแรงดันออฟเซตเข้าไปที่วงจรสัญญาณเอาต์พุทที่ได้ จะปราศจากแรงดันออฟเซตและมีการขยายสัญญาณแบบกลับเฟสผลตอบสนองทางความถี่ในช่วง 0.5-30 Hz ค่าต่ำกว่าทฤษฎีเช่นกัน ส่วนช่วงความถี่ที่มากกว่านี้จะได้อัตราการขยายที่คงที่ตรงตามทฤษฎี จากการวัดจริงสัญญาณที่ได้มีขนาดใหญ่ขึ้นและยังมีสัญญาณรบกวนอยู่
- วงจรกรองความถี่แบบกั้นไมให้ความถี่ผ่านเฉพาะช่วง โดยจะกันความถี่ 50 Hz ซึ่งเป็นสัญญาณรบกวนที่เกิดมาจากแรงดันไฟฟ้ากระแสสลับที่ใช้งานตามบ้านเพื่อไม่ให้ไปรบกวนสัญญาณคลื่นไฟฟ้าหัวใจ จากการวัดจริงจะเห็นการเปลี่ยนแปลงของสัญญาณ ได้ชัดเจนขึ้น
- วงจรกรองความถี่ต่ำผ่าน จุดประสงค์ในการกรองความถี่ต่ำผ่าน เพื่อกำจัดสัญญาณความถี่สูงที่ไม่ได้อยู่ในช่วงความถี่ที่ต้องการคือ 0-100 Hz ทิ้งไป
- วงจรพัลส์-วิดท์ มอดูเลเตอร์ จุดประสงค์ในการมอดูเลต คือ เพื่อให้การไบอัสของไดโอดในวงจรส่งผ่านสัญญาณด้วยแสงทำงานอยู่ในช่วงลิเนียร์เพื่อความถูกต้องของสัญญาณคลื่นไฟฟ้าหัวใจ
- วงจรส่งผ่านสัญญาณด้วยแสง เป็นการแยกกราวด์ออกเป็น 2 ส่วนซึ่งส่วนหนึ่งจะใช้ในส่วนสัมผัสกับตัวของผู้ป่วย และอีกส่วนหนึ่งเป็นกราวด์ของตัวเครื่อง จากการทดลองป้อนสัญญาณ PWM เข้าวงจร ปรากฏว่าสัญญาณที่วัดได้จะได้สัญญาณในซีกลบ และเฟสตรงกันกับสัญญาณ PWM
- วงจรดีมอดูเลเตอร์ เป็นการนำสัญญาณคลื่นไฟฟ้ากลับมาจากการมอดูเลต ประกอบด้วยวงจรอินทริเกรเตอร์และวงจรกรองความถี่ต่ำ จากการทดลองเมื่อป้อนสัญญาณที่ได้จากวงจรส่งผ่านสัญญาณด้วยแสงเข้าวงจร ปรากฏว่า สัญญาณที่วัดได้มีความถี่เดียวกับสัญญาณที่นำไปมอดูเลต จากการวัดจริงสัญญาณที่ได้มีความคมชัดพอสมควร เพราะยังมีสัญญาณรบกวนอยู่
- วงจรขยายและปรับระดับสัญญาณ สามารถปรับอัตราขยายสัญญาณและปรับระดับสัญญาณให้อยู่ในช่วงบวกเพื่อให้เหมาะสำหรับการเป็นสัญญาณอินพุทของวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล
- วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล จะแปลงสัญญาณคลื่นไฟฟ้าหัวใจที่เป็นสัญญาณอนาล็อกให้กลายเป็นสัญญาณดิจิทัลเพื่อส่งไปยัง ไมโคร โปรเซสเซอร์ RCM 2200 และ ไมโคร โปรเซสเซอร์ RCM 2200 จะส่งข้อมูลไปประมวลผลยังคอมพิวเตอร์หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาในการทดลอง

- ผลตอบสนองทางความถี่ในช่วง 0.5-30 Hz ของวงจรอี.ซี.จี. ยังให้ผลตอบสนองยังไม่ดีเท่าที่ควร
- จากสัญญาณที่วัดได้ยังไม่สามารถกำจัดสัญญาณได้หมด
- ขาดทักษะและความชำนาญในเรื่องของการออกแบบ การปฏิบัติงาน การเขียนโปรแกรม รวมถึงความรู้ด้านอุปกรณ์การแพทย์ จึงทำให้เกิดความล่าช้าในการทำงาน

แนวทางการดำเนินงาน

- ปรับปรุงวงจรให้ผลตอบสนองทางความถี่ในช่วง 0.5-30 Hz ให้ดีขึ้น
- กำจัดสัญญาณรบกวนให้น้อยลงกว่าเดิม
- ปรับปรุงซอฟต์แวร์ให้มีความสามารถในการแสดงผลข้อมูลให้มากยิ่งขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

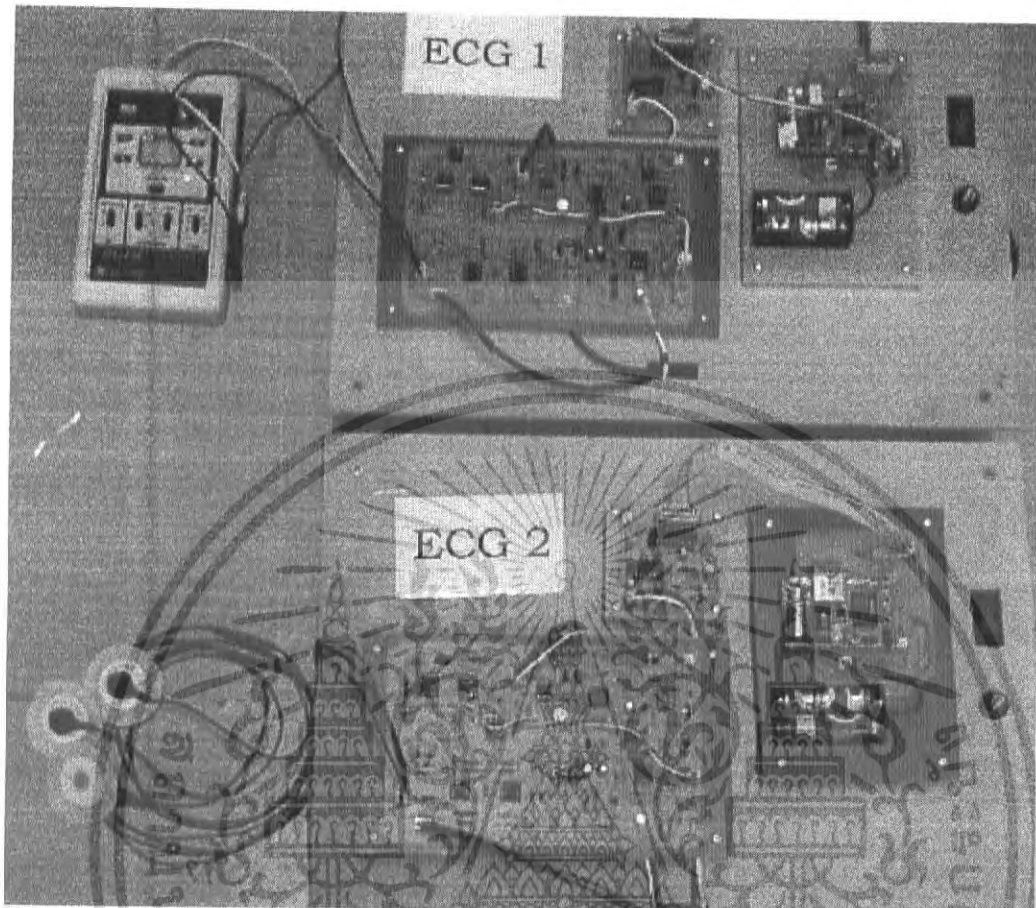
เอกสารอ้างอิง

- สมรศรี ดาวฉาย ."อุปกรณ์การแพทย์สำหรับผู้ป่วยหนัก".ภาควิชาสรีรวิทยา คณะแพทยศาสตร์ ศิริราช
พยาบาล กรุงเทพมหานคร,2539
- เกษม วัฒนชัย ."การแปรผลคลื่นไฟฟ้าหัวใจและความสำคัญทางคลินิก".สำนักพิมพ์พัฒนาศึกษา,2532
- อัจฉรา เชนฤทธิ์พิทักษ์ ."การพยาบาลผู้ป่วยภาวะวิกฤติในระบบหัวใจและหลอดเลือด".บริษัทลิฟวิง
ทรานส์มีเดีย จำกัด,2540
- นายแพทย์ชูศักดิ์ เวชแพศย์ ."อิเล็กทรอนิกส์ทางการแพทย์".ภาควิชาสรีรวิทยา คณะแพทยศาสตร์ศิริราช
พยาบาล มหาวิทยาลัยมหิดล .ชวนพิมพ์,2526
- เรืองไกร รังสีพล ."เจาะระบบ TCP/IP".บริษัท โปรวิชั่น จำกัด,2544
- สัตยุทธิ์ สว่างวรรณ และ ศิวพงษ์ ตั้งสุจริต ."เครือข่ายคอมพิวเตอร์".บริษัทเพียร์สันเอดดูเคชั่นอินโคไช
น่า จำกัด,2542
- <http://www.cybered.co.th/warnuts/wbi/wbi3/web/intro.htm#>
- <http://www.thaiheartweb.com/EKG.htm>

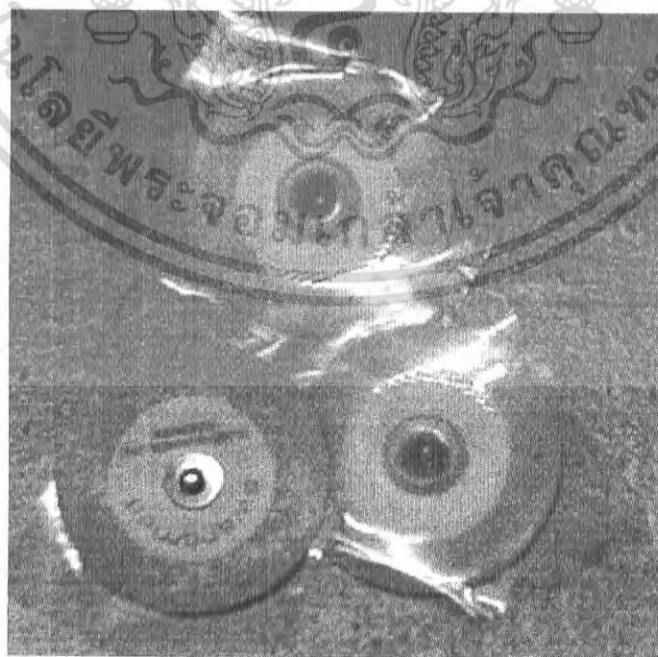
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

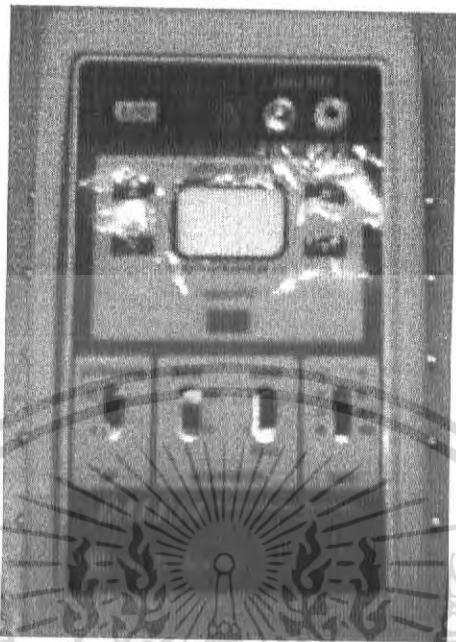


แสดงอุปกรณ์ที่ใช้ในการทดสอบคลื่นไฟฟ้าหัวใจ

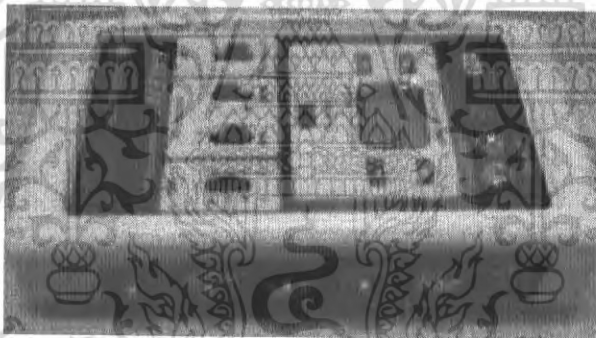


แผ่นอิเล็กทรอนิกส์ที่ใช้ในการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



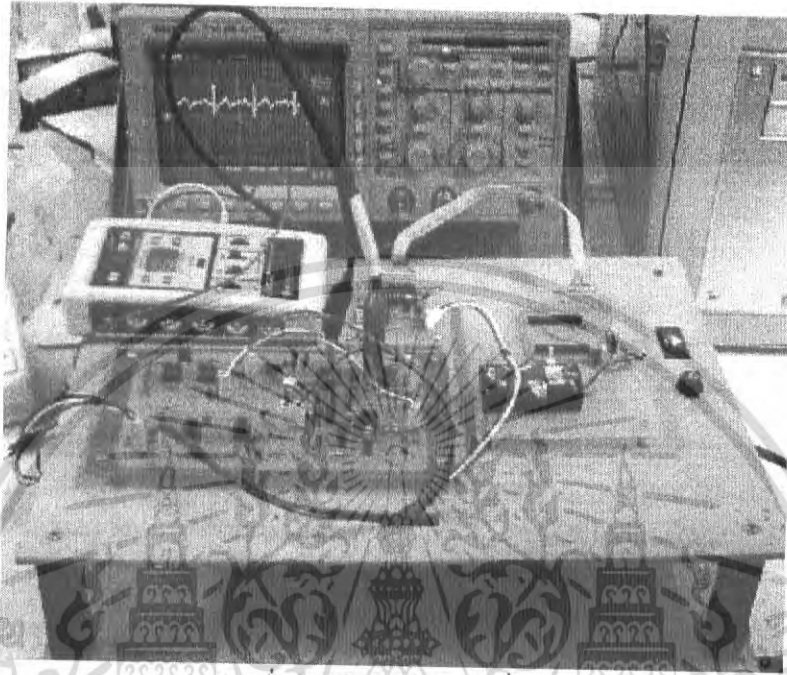
ด้านบน



ด้านข้าง

เครื่องซีมมูลเตอร์สัญญาณไฟฟ้าหัวใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

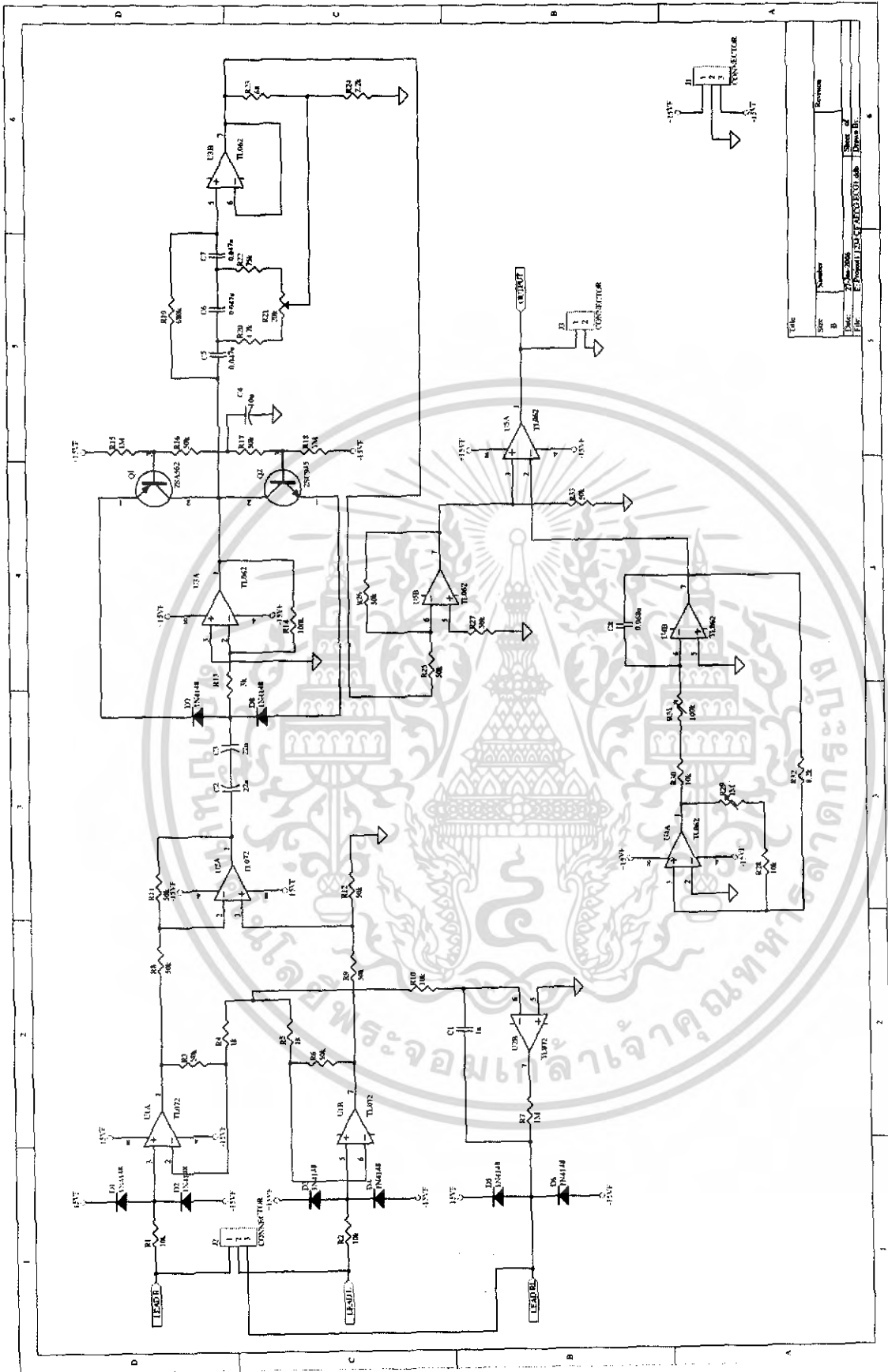


แสดงการวัดคลื่นไฟฟ้าหัวใจด้วยเครื่องซิมมูลเตอร์



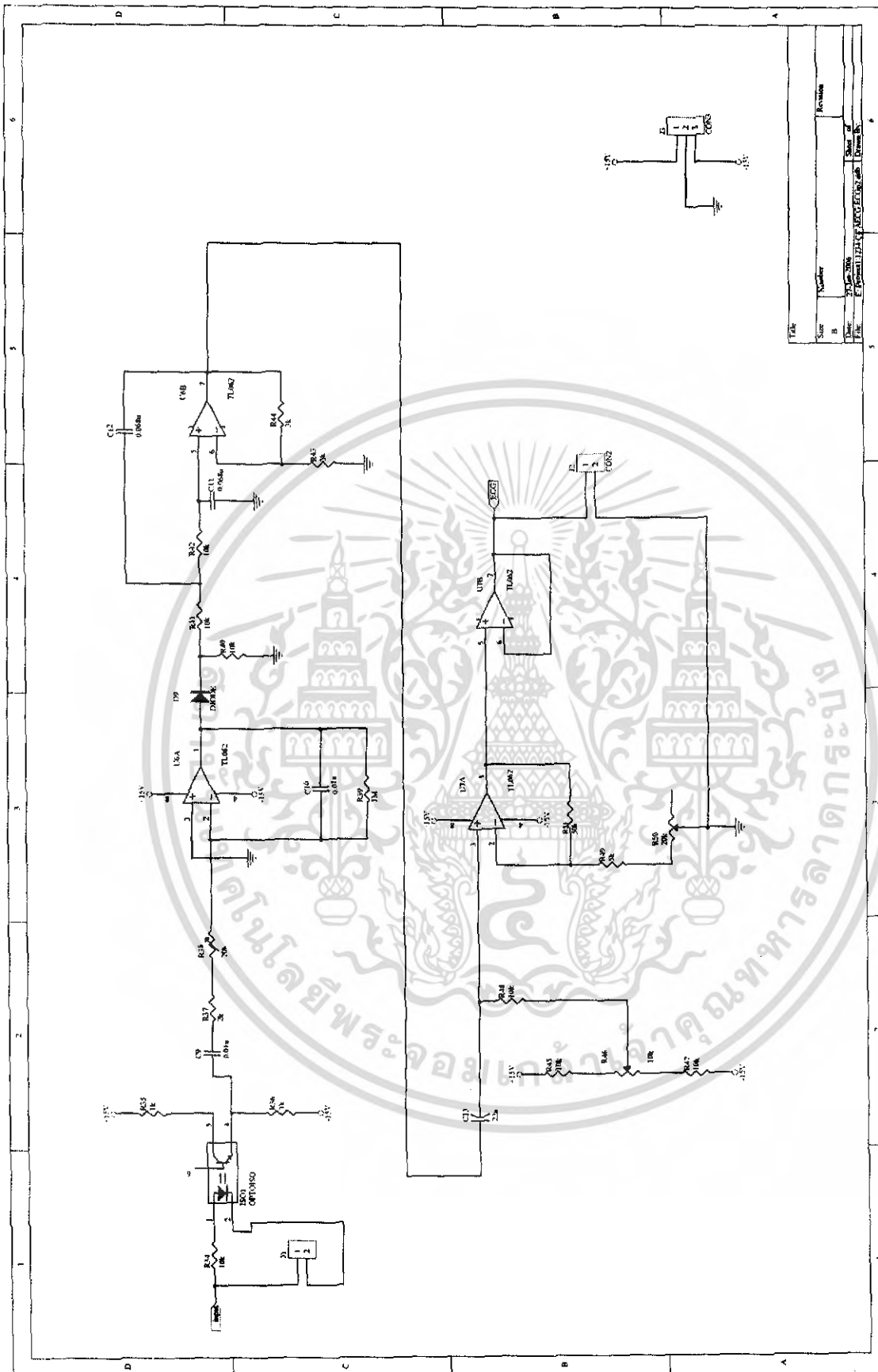
แสดงการวัดคลื่นไฟฟ้าหัวใจจากคน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



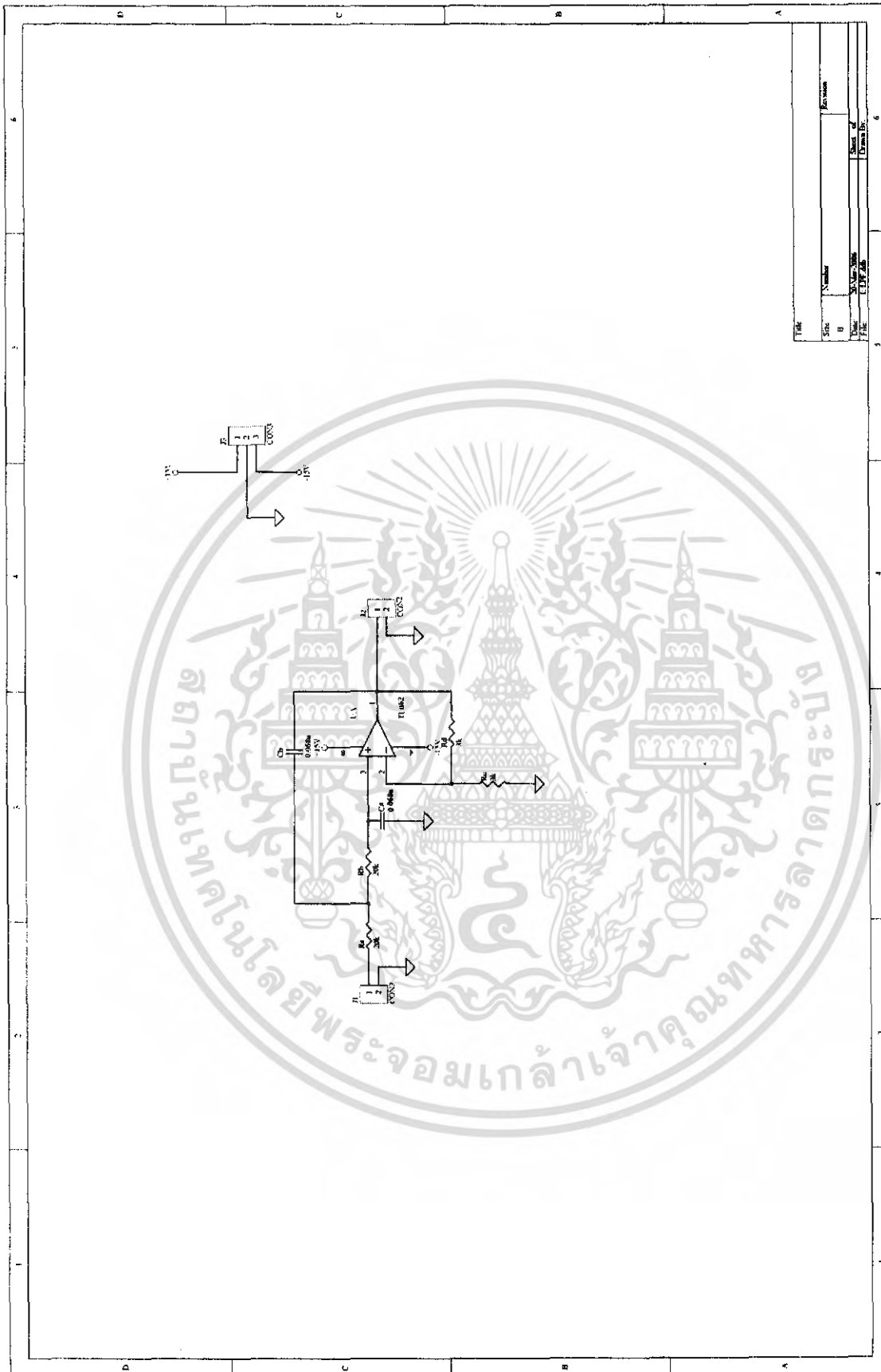
รูปแสดงวงจรเครื่อง ECG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงวงจรเครื่อง ECG (ต่อ)

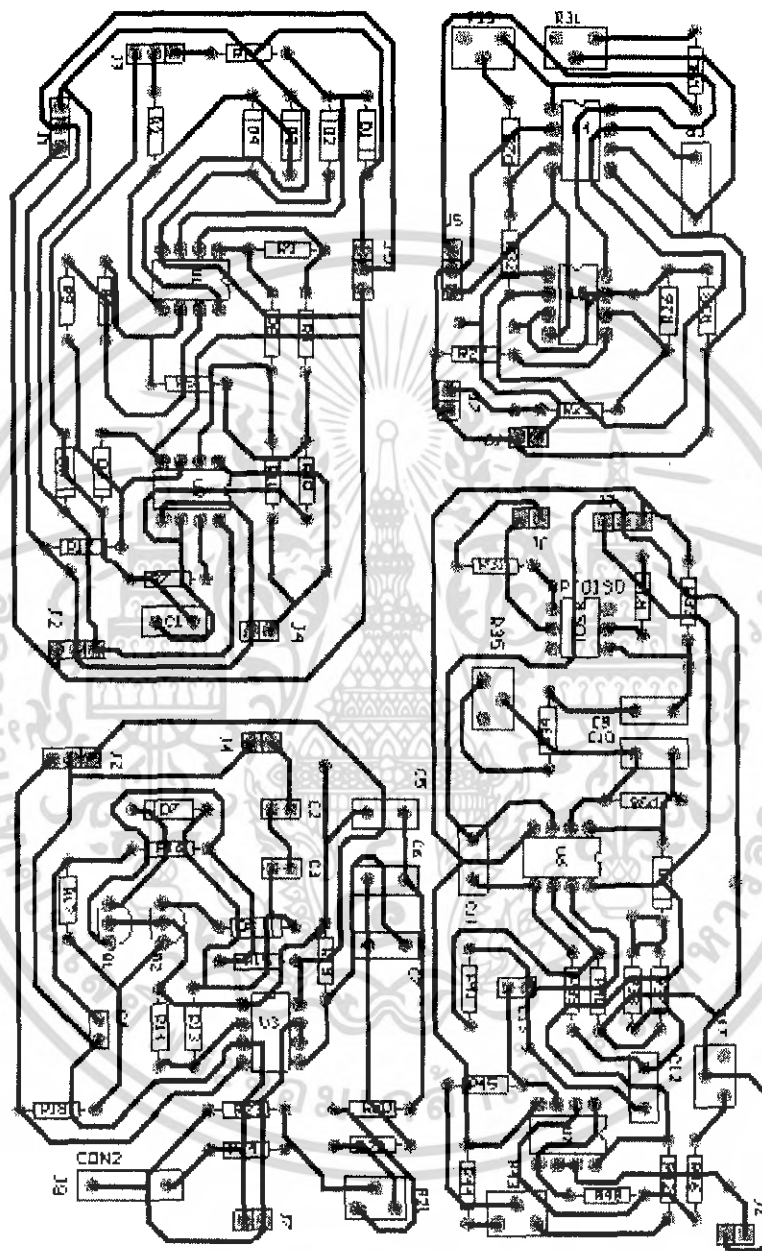
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



File	
Size	Number
B	Revision
Year	Start of
File	Drawn by

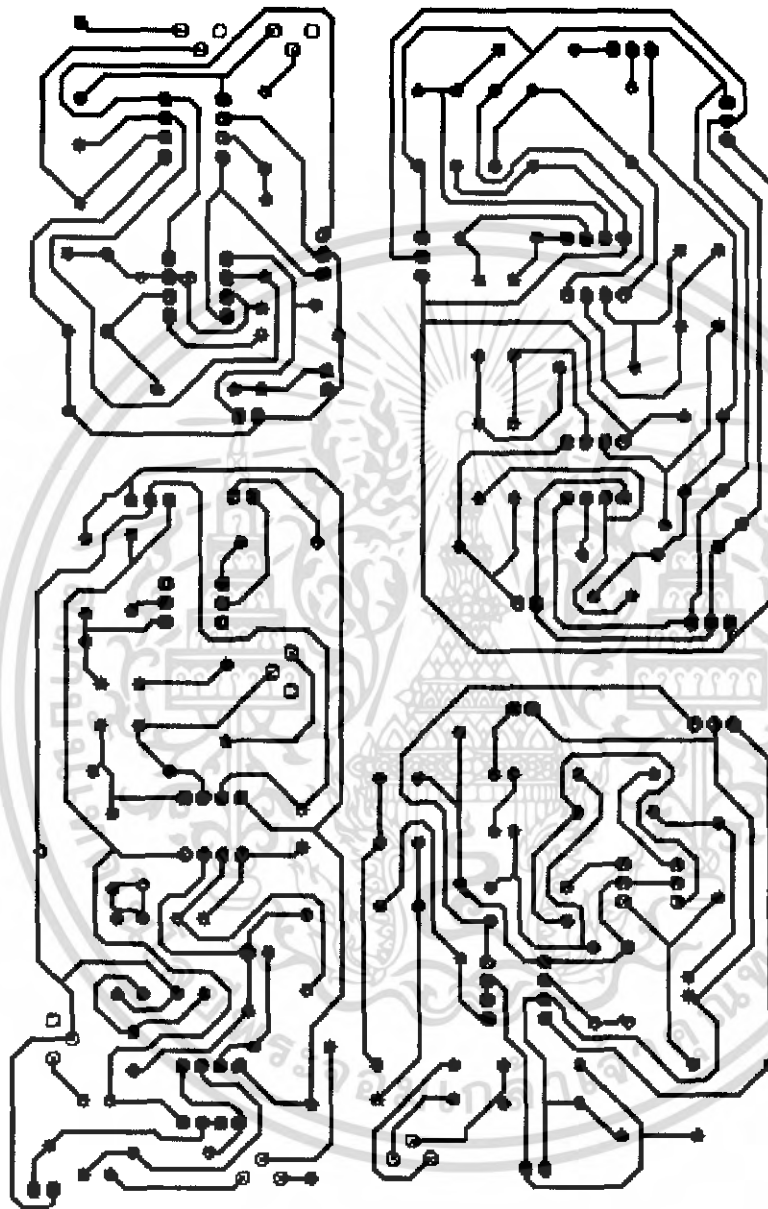
รูปแสดงวงจรความถี่ต่ำผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



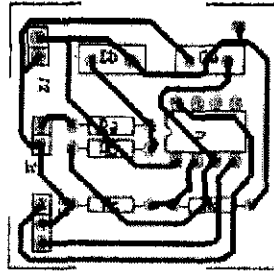
รูปลายวงจรเครื่อง ECG ด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

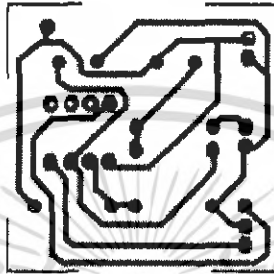


รูปลายวงจรเครื่อง ECG ด้านล่าง

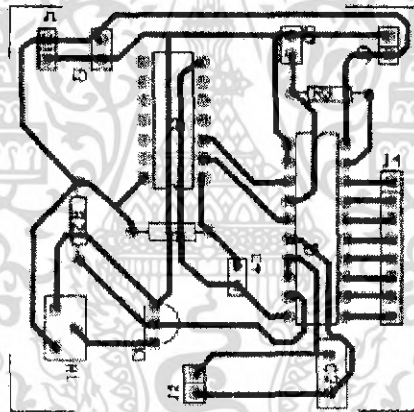
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



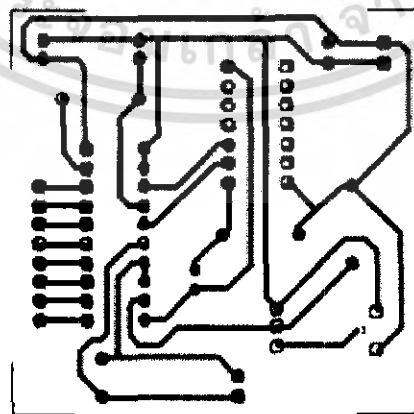
รูปลายวงจรรองความถี่ต่ำผ่านด้านบน



รูปลายวงจรรองความถี่ต่ำผ่านด้านล่าง

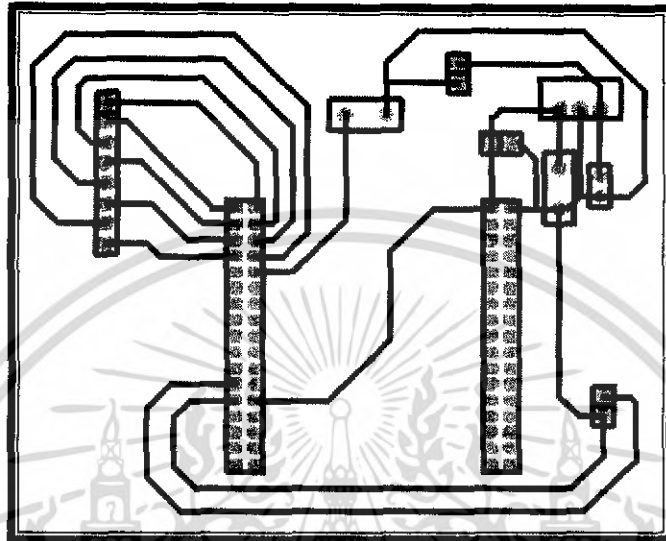


รูปลายวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลด้านบน

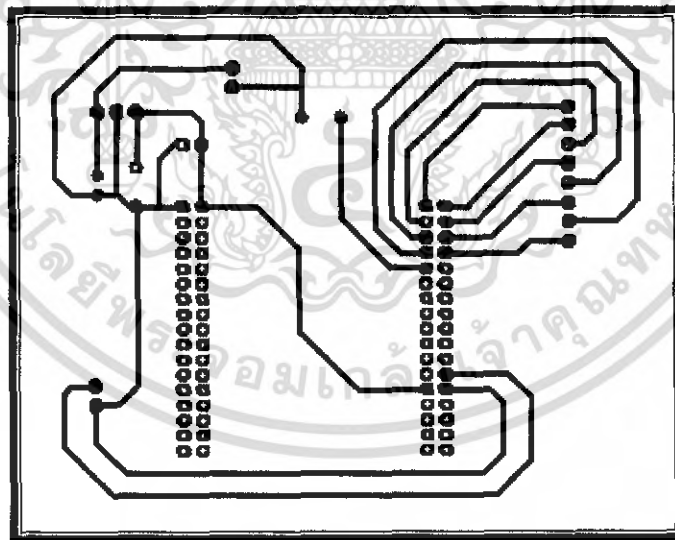


รูปลายวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปลายวงจร บอร์ด Rabbit ด้านบน



รูปลายวงจร บอร์ด Rabbit ด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



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.

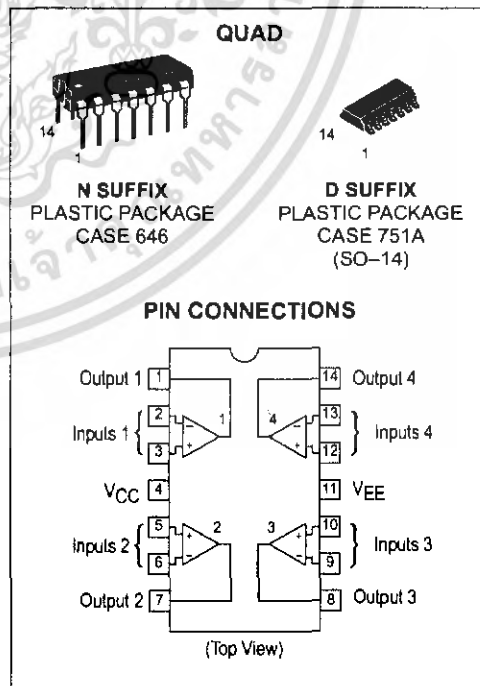
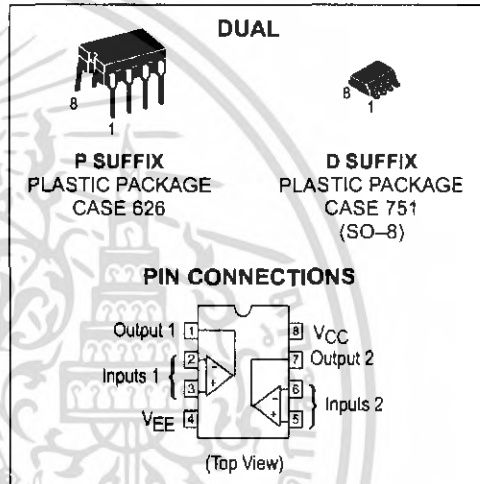
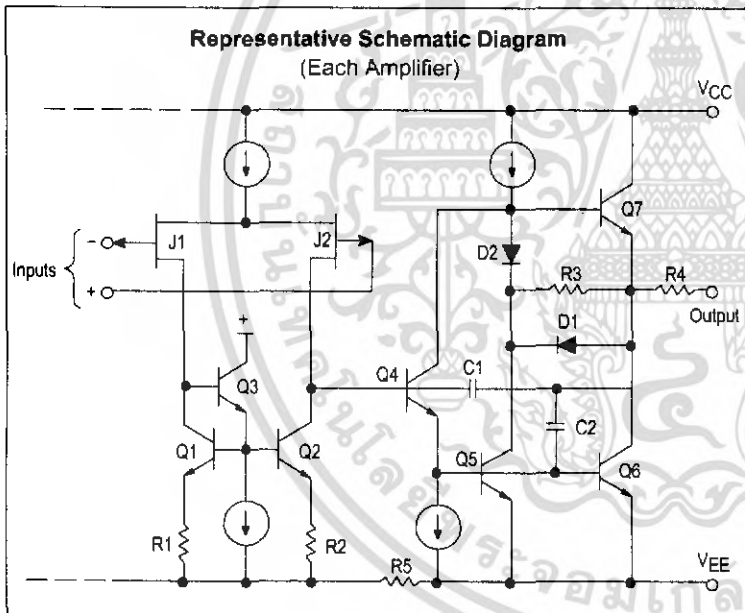
The commercial and vehicular devices are available in Plastic dual in-line and SOIC packages.

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

TL062 TL064

LOW POWER JFET INPUT OPERATIONAL AMPLIFIERS

SEMICONDUCTOR TECHNICAL DATA



ORDERING INFORMATION

Op Amp Function	Device	Operating Temperature Range	Package
Dual	TL062CD, ACD TL062CP, ACP	$T_A = 0^\circ \text{ to } +70^\circ\text{C}$	SO-8 Plastic DIP
	TL062VD TL062VP	$T_A = -40^\circ \text{ to } +85^\circ\text{C}$	SO-8 Plastic DIP
Quad	TL064CD, ACD TL064CN, ACN	$T_A = 0^\circ \text{ to } +70^\circ\text{C}$	SO-14 Plastic DIP
	TL064VD TL064VN	$T_A = -40^\circ \text{ to } +85^\circ\text{C}$	SO-14 Plastic DIP

TL062 TL064

DC ELECTRICAL CHARACTERISTICS ($V_{CC} = +15\text{ V}$, $V_{EE} = -15\text{ V}$, $T_A = T_{low}$ to T_{high} [Note 4], unless otherwise noted.)

Characteristics	Symbol	TL062V			TL064V			Unit
		Min	Typ	Max	Min	Typ	Max	
Input Offset Voltage ($R_S = 50\ \Omega$, $V_O = 0\text{ V}$) $T_A = 25^\circ\text{C}$ $T_A = T_{low}$ to T_{high}	V_{IO}	— —	3.0 —	6.0 9.0	— —	3.0 —	9.0 15	mV
Average Temperature Coefficient for Offset Voltage ($R_S = 50\ \Omega$, $V_O = 0\text{ V}$)	$\Delta V_{IO}/\Delta T$	—	10	—	—	10	—	$\mu\text{V}/^\circ\text{C}$
Input Offset Current ($V_{CM} = 0\text{ V}$, $V_O = 0\text{ V}$) $T_A = 25^\circ\text{C}$ $T_A = T_{low}$ to T_{high}	I_{IO}	— —	5.0 —	100 20	— —	5.0 —	100 20	pA nA
Input Bias Current ($V_{CM} = 0\text{ V}$, $V_O = 0\text{ V}$) $T_A = 25^\circ\text{C}$ $T_A = T_{low}$ to T_{high}	I_{IB}	— —	30 —	200 50	— —	30 —	200 50	pA nA
Input Common Mode Voltage Range ($T_A = 25^\circ\text{C}$)	V_{ICR}	— -11.5	+14.5 -12.0	+11.5 —	— -11.5	+14.5 -12.0	+11.5 —	V
Large Signal Voltage Gain ($R_L = 10\text{ k}\Omega$, $V_O = \pm 10\text{ V}$) $T_A = 25^\circ\text{C}$ $T_A = T_{low}$ to T_{high}	A_{VOL}	4.0 4.0	58 —	— —	4.0 4.0	58 —	— —	V/mV
Output Voltage Swing ($R_L = 10\text{ k}\Omega$, $V_{ID} = 1.0\text{ V}$) $T_A = 25^\circ\text{C}$ $T_A = T_{low}$ to T_{high}	V_{O+} V_{O-} V_{O+} V_{O-}	+10 — +10 —	+14 -14 — —	— -10 — -10	+10 — +10 —	+14 -14 — —	— -10 — -10	V
Common Mode Rejection ($R_S = 50\ \Omega$, $V_{CM} = V_{ICR\ min}$, $V_O = 0$, $T_A = 25^\circ\text{C}$)	CMR	80	84	—	80	84	—	dB
Power Supply Rejection ($R_S = 50\ \Omega$, $V_{CM} = 0\text{ V}$, $V_O = 0$, $T_A = 25^\circ\text{C}$)	PSR	80	86	—	80	86	—	dB
Power Supply Current (each amplifier) (No Load, $V_O = 0\text{ V}$, $T_A = 25^\circ\text{C}$)	I_D	—	200	250	—	200	250	μA
Total Power Dissipation (each amplifier) (No Load, $V_O = 0\text{ V}$, $T_A = 25^\circ\text{C}$)	PD	—	6.0	7.5	—	6.0	7.5	mW

NOTE: 4. $T_{low} = -40^\circ\text{C}$ $T_{high} = +85^\circ\text{C}$ for TL062, 4V

AC ELECTRICAL CHARACTERISTICS ($V_{CC} = +15\text{ V}$, $V_{EE} = -15\text{ V}$, $T_A = +25^\circ\text{C}$, unless otherwise noted.)

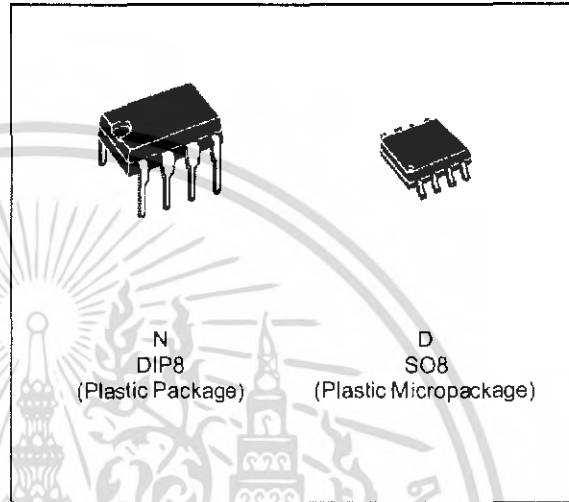
Characteristics	Symbol	Min	Typ	Max	Unit
Slew Rate ($V_{in} = -10\text{ V}$ to $+10\text{ V}$, $R_L = 10\text{ k}\Omega$, $C_L = 100\text{ pF}$, $A_V = +1.0$)	SR	2.0	6.0	—	V/ μs
Rise Time ($V_{in} = 20\text{ mV}$, $R_L = 10\text{ k}\Omega$, $C_L = 100\text{ pF}$, $A_V = +1.0$)	t_r	—	0.1	—	μs
Overshoot ($V_{in} = 20\text{ mV}$, $R_L = 10\text{ k}\Omega$, $C_L = 100\text{ pF}$, $A_V = +1.0$)	OS	—	10	—	%
Settling Time ($V_{CC} = +15\text{ V}$, $V_{EE} = -15\text{ V}$, $A_V = -1.0$, $R_L = 10\text{ k}\Omega$, $V_O = 0\text{ V}$ to $+10\text{ V}$ step)	t_s	— —	1.6 2.2	— —	μs
Gain Bandwidth Product ($f = 200\text{ kHz}$)	GBW	—	2.0	—	MHz
Equivalent Input Noise ($R_S = 100\ \Omega$, $f = 1.0\text{ kHz}$)	e_n	—	47	—	nV/ $\sqrt{\text{Hz}}$
Input Resistance	R_i	—	10^{12}	—	Ω
Channel Separation ($f = 10\text{ kHz}$)	CS	—	120	—	dB



TL072 TL072A - TL072B

LOW NOISE J-FET DUAL OPERATIONAL AMPLIFIERS

- WIDE COMMON-MODE (UP TO V_{cc}^+) AND DIFFERENTIAL VOLTAGE RANGE
- LOW INPUT BIAS AND OFFSET CURRENT
- LOW NOISE $e_n = 15\text{nV}/\text{Hz}$ (typ)
- OUTPUT SHORT-CIRCUIT PROTECTION
- HIGH INPUT IMPEDANCE J-FET INPUT STAGE
- LOW HARMONIC DISTORTION : 0.01% (typ)
- INTERNAL FREQUENCY COMPENSATION
- LATCH UP FREE OPERATION
- HIGH SLEW RATE : $16\text{V}/\mu\text{s}$ (typ)



DESCRIPTION

The TL072, TL072A and TL072B are high speed J-FET input dual operational amplifiers incorporating well matched, high voltage J-FET and bipolar transistors in a monolithic integrated circuit.

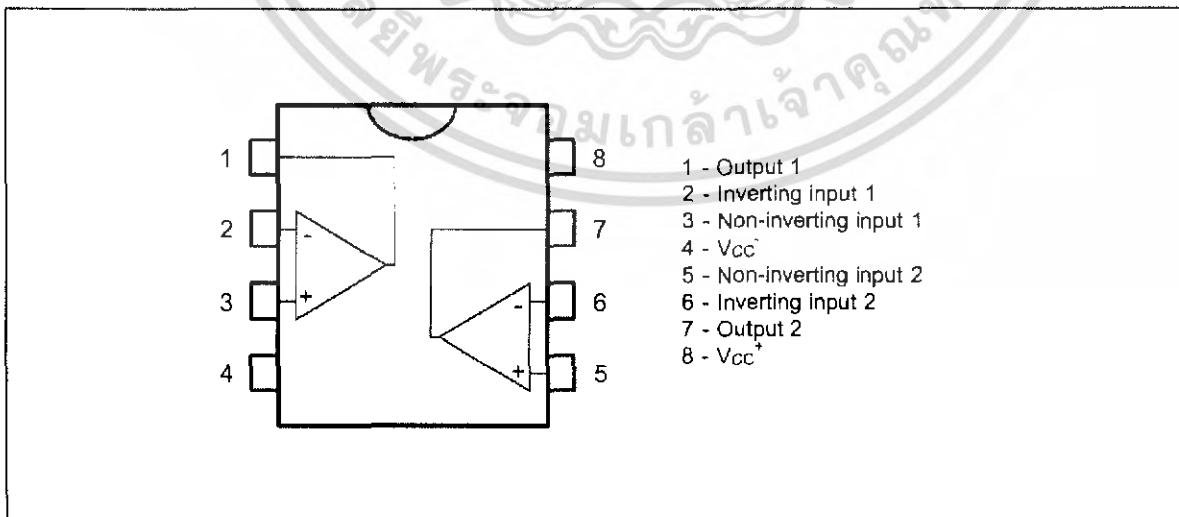
The devices feature high slew rates, low input bias and offset current, and low offset voltage temperature coefficient.

ORDER CODES

Part Number	Temperature Range	Package	
		N	D
TL072M/AM/BM	-55°C, +125°C	•	•
TL072I/AI/BI	-40°C, +105°C	•	•
TL072C/AC/BC	0°C, +70°C	•	•

Example : TL072CN

PIN CONNECTIONS (top view)



ELECTRICAL CHARACTERISTICS

V_{CC} = ±15V, T_{amb} = 25°C (unless otherwise specified)

Symbol	Parameter	TL072I,M,AC,AI, AM,BC,BI,BM			TL072C			Unit
		Min.	Typ.	Max.	Min.	Typ.	Max.	
V _{io}	Input Offset Voltage (R _s = 50Ω) T _{amb} = 25°C T _{min.} ≤ T _{amb} ≤ T _{max.}		3 3 1	10 6 3 13 7 5		3	10	mV
DV _{io}	Input Offset Voltage Drift		10			10		μV/°C
I _{io}	Input Offset Current * T _{amb} = 25°C T _{min.} ≤ T _{amb} ≤ T _{max.}		5	100 4		5	100 10	pA nA
I _{ib}	Input Bias Current * T _{amb} = 25°C T _{min.} ≤ T _{amb} ≤ T _{max.}		20	200 20		20	200 20	pA nA
A _{vd}	Large Signal Voltage Gain (R _L = 2kΩ, V _O = ±10V) T _{amb} = 25°C T _{min.} ≤ T _{amb} ≤ T _{max.}	50 25	200		25 15	200		V/mV
SVR	Supply Voltage Rejection Ratio (R _s = 50Ω) T _{amb} = 25°C T _{min.} ≤ T _{amb} ≤ T _{max.}	80 80	86		70 70	86		dB
I _{cc}	Supply Current, per Amp, no Load T _{amb} = 25°C T _{min.} ≤ T _{amb} ≤ T _{max.}		1.4	2.5 2.5		1.4	2.5 2.5	mA
V _{icm}	Input Common Mode Voltage Range	±11	+15 -12		±11	+15 -12		V
CMR	Common Mode Rejection Ratio (R _s = 50Ω) T _{amb} = 25°C T _{min.} ≤ T _{amb} ≤ T _{max.}	80 80	86		70 70	86		dB
I _{os}	Output Short-circuit Current T _{amb} = 25°C T _{min.} ≤ T _{amb} ≤ T _{max.}	10 10	40	60 60	10 10	40	60 60	mA
±V _{OPP}	Output Voltage Swing T _{amb} = 25°C T _{min.} ≤ T _{amb} ≤ T _{max.}		10 12 10 12	12 13.5	10 12 10 12	12 13.5		V
SR	Slew Rate (V _{in} = 10V, R _L = 2kΩ, C _L = 100pF, T _{amb} = 25°C, unity gain)	8	16		8	16		V/μs
t _r	Rise Time (V _{in} = 20mV, R _L = 2kΩ, C _L = 100pF, T _{amb} = 25°C, unity gain)		0.1			0.1		μs
K _{OV}	Overshoot (V _{in} = 20mV, R _L = 2kΩ, C _L = 100pF, T _{amb} = 25°C, unity gain)		10			10		%
GBP	Gain Bandwidth Product (f = 100kHz, T _{amb} = 25°C, V _{in} = 10mV, R _L = 2kΩ, C _L = 100pF)	2.5	4		2.5	4		MHz
R _i	Input Resistance		10 ¹²			10 ¹²		Ω
THD	Total Harmonic Distortion (f = 1kHz, A _v = 20dB, R _L = 2kΩ, C _L = 100pF, T _{amb} = 25°C, V _O = 2V _{PP})		0.01			0.01		%
e _n	Equivalent Input Noise Voltage (f = 1kHz, R _s = 100Ω)		15			15		nV √Hz
∅ _m	Phase Margin		45			45		Degrees
V _{O1} /V _{O2}	Channel Separation (A _v = 100)		120			120		dB

* The input bias currents are junction leakage currents which approximately double for every 10°C increase in the junction temperature.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม Dynamic C

```
#define MY_IP_ADDRESS    "161.246.18.99"
#define MY_NETMASK      "255.255.255.0"
#define MY_GATEWAY      "161.246.18.1"
#define MY_NAMESERVER   "161.246.18.4"
#define MAX_UDP_SOCKET_BUFFERS 1
#define LOCAL_PORT      54321
#define REMOTE_IP       "161.246.18.94"
//#define REMOTE_IP     "255.255.255.255" /*broadcast*/
/* the destination port to send to */
#define REMOTE_PORT     54321
/*****
 * End of configuration section *
*****/
#memmap xmem
#use "dcrtcp.lib"
udp_socket sock;
unsigned long sequence;
unsigned long reLevel , plotLevel;
int length , retval;
char buf[128];
char tempBUFF[4];
int UpdateADC()
{
    #GLOBAL_INIT
    {
        sequence = 0;
    }
    /* fill the packet with sampling data (plotLevel) */
    sequence = RdPortI(PADR);
    reLevel = (sequence * 1.0);
    plotLevel = 250 - reLevel;
    sprintf(buf, "%ld", plotLevel); // format string to Buffer
    strcat(buf, "/");
    return 1;
}
int fillPACKET()
{
    sequence = RdPortI(PADR);
    reLevel = (sequence * 1.0);
    plotLevel = 250 - reLevel;
    sprintf(tempBUFF, "%ld" , plotLevel);
    strcat(buf, tempBUFF);
    strcat(buf, "/");
    return 1;
}
int filleOS()
{
    sequence = RdPortI(PADR);
    reLevel = (sequence * 1.0);
    plotLevel = 250 - reLevel;
    sprintf(tempBUFF , "%ld" , plotLevel);
    strcat(buf , tempBUFF);
    strcat(buf, "/EOS");
    return 1;
}
main()
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม Java

```
import javax.swing.*;
import javax.swing.event.*;
import javax.swing.border.*;
import javax.swing.plaf.basic.*;
import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.applet.*;
import java.net.*;
import java.util.GregorianCalendar;
import java.util.Vector;
import java.awt.image.*;

class MyProject extends JFrame
{
    public static void main(String[] args) throws Exception
    {
        JFrame f = new JFrame("ECG Window Viewer");

        try {
            UIManager.setLookAndFeel(
"com.sun.java.swing.plaf.windows.WindowsLookAndFeel" );
        } catch (Exception e) {}
        //Add MenuBar to Frame

        JMenuBar mb = new JMenuBar();
        JMenu m1 = new JMenu("File");
        JMenu m2 = new JMenu("Option");
        JMenu m3 = new JMenu("View");
        JMenu m4 = new JMenu("window");
        JMenu m5 = new JMenu("Help");
        mb.add(m1);
        mb.add(m2);
        mb.add(m3);
        mb.add(m4);
        mb.add(m5);
        f.setJMenuBar(mb);
        //TitledBorder border = new TitledBorder(new
SoftBevelBorder(BevelBorder.RAISED),"title");
        //border.setTitlePosition(border.BELOW_TOP);
        Border border = new
BasicBorders.MenuBarBorder(Color.white,Color.gray);
        mb.setBorder(border);

        //End Add MenuBar

        ECG_Swing app = new ECG_Swing();

        Container c = f.getContentPane();
        c.add(app);
        f.setSize(520,630);
        f.show();

        app.CreateDatagramSocket();

        System.out.println("Recieve packet :"+app.hostAddr);
        f.addWindowListener(
            new WindowAdapter() {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        public void windowClosing(WindowEvent e)
        {
            System.exit(0);
        }
    }
};
app.receivePackets();
}

class ECG_Swing extends JPanel implements
ActionListener//,CaretListener
{
    JTabbedPane tabbedPane;
    GridLayout grid;
    JPanel contents,textContainer,UserAccount,NetworkConnect;
    static DefaultTableModel UAdetailModel,NCdetailModel;
    static int nowindex = 0;
    Object Default_User[][] = {
        {" Pamornrat
Piyavanichskul","161.246.18.99",""},
        {" Padorn
Juntapimol","161.246.18.98",""},
        {" Purich
Nakkanong","161.246.18.97",""},
        {" Pornsuda
Vongsushoto","161.246.18.96",""},
        {" Mana
Saatung","161.246.18.95",""},
        {" Wuttipol
Aukkaranimol","161.246.18.94",""}
    };

    JTable DetailTable,StatusTable;
    JMenuBar mb;
    JMenu m1,m2,m3,m4,m5;
    JLabel label1,label2;
    JButton
UAXitButton, ShowButton, RemoveButton, AddButton, NCExitButton, NCCheckBu
tton, NCAddButton, NCClearButton, NCResetButton;
    JTextField
AddUserName, IP_Byte1, IP_Byte2, IP_Byte3, IP_Byte4, IPCheck_B1, IPCheck_B2
, IPCheck_B3, IPCheck_B4;
    static JTextField UAindexShow;
    int UAnumRow,UAheightRow,NCnumRow,NCheightRow;
    int index = 6;
    static DatagramSocket socket;
    static DatagramPacket packet_rx;
    static String hostAddr,dataReceive;
    static int numUsebuff;
    byte DataReceive[];
    static Vector buffer1,buffer2,buffer3,buffer4,buffer5;
    ECG1_Container ECGContent1;
    ECG2_Container ECGContent2;
    ECG3_Container ECGContent3;
    ECG4_Container ECGContent4;
    ECG5_Container ECGContent5;
    int checkpoint = 0;
    static int Count1,Count2,Count3,Count4,Count5;
    int data_length;
    String temp_receive;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Vector temp_buffer;

ECG_Swing()
{
    setPreferredSize(new Dimension(525,575));
    setLayout(new BorderLayout());
    tabbedPane = new JTabbedPane();

    tabbedPane.setBorder(BorderFactory.createEmptyBorder(5,5,0,5));

//UserAccount tabbed

    UserAccount = new JPanel();
    UAnumRow = 13;
    UAheightRow = 25;
    Object UAheaderNames[] = {"User Name","IP
Address","Status"};
    UAdetailModel = new
DefaultTableModel(UAheaderNames,UAnumRow);
    StatusTable = new JTable(UAdetailModel);
    JTableHeader UAheader = new JTableHeader();
    UAheader.setResizingAllowed(true);
    UAheader.setTable(StatusTable);
    StatusTable.setFont(new Font("Arial",0,11));

    StatusTable.setPreferredScrollableViewportSize(new
Dimension(475,13*20));
    StatusTable.setShowGrid(false);
    StatusTable.setRowHeight(UAheightRow);

    StatusTable.setSelectionMode(ListSelectionModel.SINGLE_SELECTIO
N);

    StatusTable.setEnabled(false);
    StatusTable.setForeground(Color.blue);
    JScrollPane UAscPane = new
JScrollPane(StatusTable,ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEE
DED,ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS);
    JPanel UAtableContainer = new JPanel();

    UAtableContainer.setBorder(BorderFactory.createEmptyBorder(6,5,
0,5));

    UAtableContainer.add(UAscPane);
    UserAccount.add(UAtableContainer);

    UAindexShow = new JTextField(2);
    UAExitButton = new JButton("Exit");
    UAExitButton.setForeground(Color.red);
    ShowButton = new JButton("Show ECG");
    Color bcolor = new Color(35,200,35);
    ShowButton.setForeground(bcolor);
    RemoveButton = new JButton("Remove Tab");
    RemoveButton.setForeground(Color.red);
    AddButton = new JButton("Add");
    AddButton.setForeground(bcolor);
    JPanel Select_RemoveContainer = new JPanel();
    JPanel UAbuttonContainer = new JPanel();
    FlowLayout UAbuttonLayout = new
FlowLayout(FlowLayout.LEFT,10,0);
    UAbuttonContainer.setLayout(UAbuttonLayout);
    JLabel UAblank = new JLabel("
");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        //Border UAborder1 = new
SoftBevelBorder(BevelBorder.RAISED);
        TitledBorder UAborder2 = new
TitledBorder("Select index for Show ECG or Add User and Exit");
        UAborder2.setTitleColor(Color.blue);
        //CompoundBorder UAborder3 = new
CompoundBorder(UAborder2,UAborder1);
        Select_RemoveContainer.setLayout(new
FlowLayout(FlowLayout.RIGHT,13,13));
        Select_RemoveContainer.setBorder(UAborder2);
        Select_RemoveContainer.add(UAindexShow);
        Select_RemoveContainer.add(AddButton);
        Select_RemoveContainer.add(ShowButton);
        Select_RemoveContainer.add(RemoveButton);
        Select_RemoveContainer.add(UAblank);
        Select_RemoveContainer.add(UAExitButton);

        UAbuttonContainer.add(Select_RemoveContainer);
        UserAccount.add(UAbuttonContainer);

        UAExitButton.addActionListener(new
ActionListener() {
        e) {
                public void actionPerformed(ActionEvent
                System.exit(0);
                }
        });
//
        tabbedPane.addTab("Users Accounts",UserAccount);

//Network Connection tabbed
        NetworkConnect = new JPanel();
        NCnumRow = 7;
        NCheightRow = 25;
        Object NCheaderNames[] = {"User Name", "IP
Address", "Status"};
        NCdetailModel = new
DefaultTableModel(NCheaderNames,NCnumRow);
        NCdetailModel.insertRow(0,Default_User[0]);
        NCdetailModel.insertRow(1,Default_User[1]);
        NCdetailModel.insertRow(2,Default_User[2]);
        NCdetailModel.insertRow(3,Default_User[3]);
        NCdetailModel.insertRow(4,Default_User[4]);
        NCdetailModel.insertRow(5,Default_User[5]);
        DetailTable = new JTable(NCdetailModel);
        DetailTable.setFont(new Font("Arial",0,11));
        JTableHeader NChheader = new JTableHeader();
        NChheader.setResizingAllowed(true);

        NChheader.setBorder(BorderFactory.createEmptyBorder(5,0,5,0));
        NChheader.setTable(DetailTable);
        DetailTable.setFont(new Font("Arial",0,11));

        DetailTable.setPreferredScrollableViewportSize(new
Dimension(475,13*20));
        DetailTable.setShowGrid(false);
        DetailTable.setRowHeight(NCheightRow);
        DetailTable.setEnabled(false);
        DetailTable.setRowSelectionAllowed(true);
        DetailTable.setForeground(Color.blue);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        JScrollPane NCscPane = new
JScrollPane (DetailTable,ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEE
DED,ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS);
        JPanel NctableContainer = new JPanel();

        NctableContainer.setBorder (BorderFactory.createEmptyBorder (6,5,
0,5));

        NctableContainer.add (NCscPane);

//

        JPanel AddContainer = new JPanel();
        JPanel IPContainer = new JPanel();
        JPanel AddUsernameContainer = new JPanel();
        NCAddButton = new JButton ("Add");
        NCClearButton = new JButton ("Remove");
        //Border border1 = new
SoftBevelBorder (BevelBorder.RAISED);
        TitledBorder border2 = new
TitledBorder ("Enter UserName and IPAddress then click Add or
Remove");

        border2.setTitleColor (Color.blue);
        //CompoundBorder border3 = new
CompoundBorder (border2,border1);
        AddContainer.setBorder (border2);
        AddUserName = new JTextField (15);
        AddUsernameContainer.add (AddUserName);
        IP_Byte1 = new JTextField (3);
        IP_Byte2 = new JTextField (3);
        IP_Byte3 = new JTextField (3);
        IP_Byte4 = new JTextField (3);
        JLabel dot = new JLabel (".");
        JLabel dot1 = new JLabel (".");
        JLabel dot2 = new JLabel (".");
        dot.setFont (new Font ("Arial",0,18));
        dot1.setFont (new Font ("Arial",0,18));
        dot2.setFont (new Font ("Arial",0,18));
        AddContainer.setLayout (new
FlowLayout (FlowLayout.RIGHT,13,5));
        IPContainer.setLayout (new
FlowLayout (FlowLayout.RIGHT,0,5));
        IPContainer.add (IP_Byte1);
        IPContainer.add (dot);
        IPContainer.add (IP_Byte2);
        IPContainer.add (dot1);
        IPContainer.add (IP_Byte3);
        IPContainer.add (dot2);
        IPContainer.add (IP_Byte4);
        AddContainer.add (IPContainer);
        AddContainer.add (AddUsernameContainer);
        JPanel AddContainer1 = new JPanel();
        AddContainer1.setLayout (new
FlowLayout (FlowLayout.RIGHT,0,10));
        JLabel or = new JLabel ("/");
        or.setFont (new Font ("Arial",0,16));
        AddContainer1.add (NCAddButton);
        AddContainer1.add (or);
        AddContainer1.add (NCClearButton);
        AddContainer.add (AddContainer1);

        JPanel CheckContainer = new JPanel();
        JPanel IPContainer1 = new JPanel();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        NCCheckButton = new JButton("Check");
        IPCheck_B1 = new JTextField(3);
        IPCheck_B2 = new JTextField(3);
        IPCheck_B3 = new JTextField(3);
        IPCheck_B4 = new JTextField(3);
        JLabel dot3 = new JLabel(".");
        JLabel dot4 = new JLabel(".");
        JLabel dot5 = new JLabel(".");
        dot3.setFont(new Font("Arial", 0, 18));
        dot4.setFont(new Font("Arial", 0, 18));
        dot5.setFont(new Font("Arial", 0, 18));
        CheckContainer.setLayout(new
FlowLayout(FlowLayout.RIGHT, 13, 5));
        IPContainer1.setLayout(new
FlowLayout(FlowLayout.RIGHT, 0, 7));
        //Border border4 = new
SoftBevelBorder(BevelBorder.RAISED);
        TitledBorder border5 = new
TitledBorder("Enter IP Address Check for Status of User");
        border5.setTitleColor(Color.blue);
        //CompoundBorder border6 = new
CompoundBorder(border5, border4);
        CheckContainer.setBorder(border5);
        IPContainer1.add(IPCheck_B1);
        IPContainer1.add(dot3);
        IPContainer1.add(IPCheck_B2);
        IPContainer1.add(dot4);
        IPContainer1.add(IPCheck_B3);
        IPContainer1.add(dot5);
        IPContainer1.add(IPCheck_B4);
        JPanel CheckContainer1 = new JPanel();
        CheckContainer1.setLayout(new
FlowLayout(FlowLayout.RIGHT, 13, 5));
        CheckContainer.add(IPContainer1);
        CheckContainer.add(NCCheckButton);
        CheckContainer1.add(CheckContainer);
        CheckContainer1.add(new JLabel("
"));
        JPanel CtResetandExit = new JPanel();
        CtResetandExit.setBorder(BorderFactory.createEmptyBorder(15, 0, 0
, 5));
        NCResetButton = new JButton("Reset");
        NCExitButton = new JButton("Exit");
        NCExitButton.setForeground(Color.red);
        CtResetandExit.add(NCResetButton);
        CtResetandExit.add(new JLabel(""));
        CtResetandExit.add(NCExitButton);
        CtResetandExit.add(new JLabel("
"));
        CheckContainer1.add(CtResetandExit);

        NetworkConnect.add(NCtableContainer);
        NetworkConnect.add(CheckContainer1);
        NetworkConnect.add(AddContainer);

        NCExitButton.addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent
e) {
                System.exit(0);
            }
        })
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        });

        tabbedPane.addTab("Network
Connections",NetworkConnect);

//End Network Connection Tabbed

        textContainer = new JPanel();
        textContainer.setLayout(new
FlowLayout(FlowLayout.LEFT));
        labell1 = new JLabel();
        labell1.setHorizontalAlignment(JLabel.LEFT);
        labell1.setText("ECG Window Starting...
");

        labell1.setForeground(Color.orange);
        labell1.setFont(new Font("Arial",0,12));
        textContainer.add(labell1);
        textContainer.add(new DateTime());
        add(tabbedPane, BorderLayout.CENTER);
        add(textContainer, BorderLayout.SOUTH);

//Create ECG object and new buffer
temp_buffer = new Vector();
buffer1 = new Vector();
buffer2 = new Vector();
buffer3 = new Vector();
buffer4 = new Vector();
buffer5 = new Vector();
buffer1.removeAllElements();
buffer2.removeAllElements();
buffer3.removeAllElements();
buffer4.removeAllElements();
buffer5.removeAllElements();
ECGContent1 = new ECG1_Container();
ECGContent2 = new ECG2_Container();
ECGContent3 = new ECG3_Container();
ECGContent4 = new ECG4_Container();
ECGContent5 = new ECG5_Container();

//Event Handling

NCAddButton.addActionListener(this);
NCClearButton.addActionListener(this);
AddUserName.addActionListener(this);
NCResetButton.addActionListener(this);
NCCheckButton.addActionListener(this);
ShowButton.addActionListener(this);
RemoveButton.addActionListener(this);
AddButton.addActionListener(this);
UAindexShow.addActionListener(this);
}

/***** Method for manage DetailTable
*****/

//Method Reset All TextField
void resetTextField()
{
    AddUserName.setText("");
    IP_Byte1.setText("");
    IP_Byte2.setText("");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        IP_Byte3.setText("");
        IP_Byte4.setText("");
    }
    //End Method Reset All TextField

    //Method AddData (NCAddButton Listener)
    int addData(int index1)
    {
        if ( (""+(AddUserName.getText())).equals("")) ||
        (""+(IP_Byte1.getText())).equals("")) ||
        (""+(IP_Byte2.getText())).equals("")) ||
        (""+(IP_Byte3.getText())).equals("")) ||
        (""+(IP_Byte4.getText())).equals("")) )
            JOptionPane.showMessageDialog(null, "Please
enter Username and IP Address", "Error", JOptionPane.ERROR_MESSAGE);

        else if ( (AddUserName.getText()) != "" ) &&
        (Integer.parseInt(IP_Byte1.getText()) >= 0) &&
        (Integer.parseInt(IP_Byte2.getText()) >= 0) &&
        (Integer.parseInt(IP_Byte3.getText()) >= 0) &&
        (Integer.parseInt(IP_Byte4.getText()) >= 0) )
        {
            if ( (Integer.parseInt(IP_Byte1.getText()) >
255) || (Integer.parseInt(IP_Byte2.getText()) > 255) ||
(Integer.parseInt(IP_Byte3.getText()) > 255) ||
(Integer.parseInt(IP_Byte4.getText()) > 254) )
                JOptionPane.showMessageDialog(null, "Incorrect IP
Address", "Error", JOptionPane.ERROR_MESSAGE);
            else
            {
                //
                String All_IPbyte, temp_IPbyte1,
temp_IPbyte2, temp_IPbyte3, temp_IPbyte4, X1, X2, X3, X4;
                temp_IPbyte1 = "";
                temp_IPbyte2 = "";
                temp_IPbyte3 = "";
                temp_IPbyte4 = "";
                X1 = new
String(IP_Byte1.getText());
                X2 = new
String(IP_Byte2.getText());
                X3 = new
String(IP_Byte3.getText());
                X4 = new
String(IP_Byte4.getText());
                if ( Integer.parseInt(X1) == 0 )
                    temp_IPbyte1 = "0";
                else
                {
                    int lX1 = 0;
                    for (int lX1 = 0 ;
lX1<X1.length() ; lX1++)
                    {
                        if (X1.charAt(lX1) != '0')
                        {
                            lX1 = lX1;
                            break;
                        }
                    }
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; i++)
+ X1.charAt(i);
for ( int i = lX1 ; i<X1.length()
{
temp_IPbyte1 = temp_IPbyte1
}
}
if ( Integer.parseInt(X2) == 0 )
temp_IPbyte2 = "0";
else
{
int lX2 = 0;
for (int lX2 = 0 ;
lX2<X2.length() ; lX2++)
{
if (X2.charAt(lX2) != '0')
{
lX2 = lX2;
break;
}
}
for ( int i = lX2 ; i<X2.length()
{
temp_IPbyte2 = temp_IPbyte2
}
}
if ( Integer.parseInt(X3) == 0 )
temp_IPbyte3 = "0";
else
{
int lX3 = 0;
for (int lX3 = 0 ;
lX3<X3.length() ; lX3++)
{
if (X3.charAt(lX3) != '0')
{
lX3 = lX3;
break;
}
}
for (int i = lX3 ; i<X3.length()
; i++)
{
temp_IPbyte3 = temp_IPbyte3
}
}
+ X3.charAt(i);
}
if ( Integer.parseInt(X4) == 0 )
temp_IPbyte4 = "0";
else
{
int lX4 = 0;
for (int lX4 = 0 ;
lX4<X4.length() ; lX4++)
{
if (X4.charAt(lX4) != '0')

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {
            lx4 = lX4;
            break;
        }
    }
    for ( int i = lx4 ; i<X4.length()
; i++)
    {
        temp_IPbyte4 = temp_IPbyte4
+ X4.charAt(i);
    }

    int check = -1;
    for (int i=0 ; i<index1 ; i++)
    {
        if (
((""+NCdetailModel.getValueAt(i,0)).equals(""+(AddUserName.getText()
)) || (
(""+NCdetailModel.getValueAt(i,1)).equals(""+(IP_Byte1.getText()+"."+
(IP_Byte2.getText()+"."+(IP_Byte3.getText()+"."+(IP_Byte4.getText(
))) ) )
        check = i;
    }
    if (check == -1)
    {
        NCdetailModel.setValueAt("
"+AddUserName.getText(),index1,0);
        //NCdetailModel.setValueAt(
(""+(IP_Byte1.getText()+"."+(IP_Byte2.getText()+"."+(IP_Byte3.getTe
xt()+"."+(IP_Byte4.getText()))),index1,1);
        NCdetailModel.setValueAt(
(temp_IPbyte1+"."+temp_IPbyte2+"."+temp_IPbyte3+"."+temp_IPbyte4),ind
ex1,1);
        index1 = ++index1;
    }
    else
    {
        JOptionPane.showMessageDialog(null,"This Username or IP Address
has already use at index
:"+ (check+1), "Error", JOptionPane.ERROR_MESSAGE);
    }
}

return index1;
}
//End Method AddData (NCAddButton Listener)

//Method Remove (NCClearButton)
int findSelectIndex()
{
    int k = 0;
    while (!((""+NCdetailModel.getValueAt(k,0)).equals("
"+(AddUserName.getText())) )
    {
        k = ++k;
    }
    return k;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void removeData(int index1,String tempName1)
{
    int check = -1;
    for ( int i = 0 ; i<UAdetailModel.getRowCount() ;
i++ )
    {
        if (
(""+UAdetailModel.getValueAt(i,0)).equals(tempName1) )
        {
            check = i;
            break;
        }
    }
    if ( check != -1 )
    {
        Object initData1[] = {"","",""};
        UAdetailModel.removeRow(check);
        UAdetailModel.addRow(initData1);
    }

    Object initData2[] = {"","",""};
    NCdetailModel.removeRow(index1);
    NCdetailModel.addRow(initData2);
}
//End Method Remove (NCClearButton)

//Method Reset All in Table
void resetAll()
{
    Object initData1[] = {"","",""};
    for (int b = 0 ; b<13 ; b++)
    {
        NCdetailModel.removeRow(0);
        NCdetailModel.addRow(initData1);
        UAdetailModel.removeRow(0);
        UAdetailModel.addRow(initData1);
    }
}
//End Method Reset All in Table

/***** End Method for manage DetailTable
*****/

/***** Method for manage StatusTable
*****/

int AddUser_showECG(int indexmax,int indexnow)
{
    indexmax = indexmax;
    if ( Integer.parseInt(UAindexShow.getText()) == 0 )
        JOptionPane.showMessageDialog(null,"Index
must more than 0","Error",JOptionPane.ERROR_MESSAGE);
    else
    {
        if ( (""+UAindexShow.getText()).equals("")
)

        JOptionPane.showMessageDialog(null,"Please Enter table's index
","Error",JOptionPane.ERROR_MESSAGE);
        else if (
(Integer.parseInt(UAindexShow.getText())-1 >= 0) )

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {
            if (
(Integer.parseInt(UAindexShow.getText())-1 >= indexmax))

                JOptionPane.showMessageDialog(null,"This index don't have
User", "Errir", JOptionPane.ERROR_MESSAGE);

                else
                {
                    int enterIndex =
Integer.parseInt(UAindexShow.getText())-1;
                    int check = -1;
                    for ( int i = 0 ;
i<UAdetailModel.getRowCount() ; i++ )
                    {
                        if (
(""+NCdetailModel.getValueAt(enterIndex,0)).equals(""+UAdetailModel.g
etValueAt(i,0)) )
                        {
                            check = i;
                            break;
                        }
                        if ( check != -1 )
                            JOptionPane.showMessageDialog(null,"This User has already use
at index :"+(check+1),"Error",JOptionPane.ERROR_MESSAGE);
                            else
                            {
                                UAdetailModel.setValueAt(
NCdetailModel.getValueAt(enterIndex,0),indexnow,0 );
                                UAdetailModel.setValueAt(
NCdetailModel.getValueAt(enterIndex,1),indexnow,1);
                                indexnow = ++ indexnow;
                            }
                        }
                    }
                    return indexnow;
                }
            }

/***** End Method for manage StatusTable
*****/

/***** Method about Network
*****/

//Create Datagram Socket
public void CreateDatagramSocket()
{
    try
    {
        socket = new DatagramSocket( 54321 );
    }
    catch( SocketException socketException )
    {
        socketException.printStackTrace();
        System.exit( 1 );
    }
}

//End Create Datagram Socket

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//Receiving a UDP packet
public void receivePackets() throws Exception
{
    while (true)
    {
        byte data_rx[] = new byte[250];
        DatagramPacket packet_rx = new
DatagramPacket(data_rx, data_rx.length);
        socket.receive(packet_rx);
        InetAddress hostInetAddress =
packet_rx.getAddress();
        hostAddr = ""+hostInetAddress.getHostAddress();
        dataReceive = new
String(packet_rx.getData(), 0, packet_rx.getLength()-1);
        //int data_Rx =
Integer.parseInt(dataReceive);
        System.out.println("HostAddr :
"+hostAddr+"Data Receive : "+dataReceive);

        temp_buffer.removeAllElements();
        data_length = dataReceive.length();
        System.out.println("Data length :
"+data_length); // need
        //System.out.println("data receive :
"+dataReceive.charAt(39)+dataReceive.charAt(40)+dataReceive.charAt(41
)+dataReceive.charAt(42));
        for ( int i = 0 ; i < data_length ; i++ )
        {
            if (
{"("/") .equals(""+dataReceive.charAt(i))} && {
{"(E") .equals(""+dataReceive.charAt(i+1))} } && {
{"(O") .equals(""+dataReceive.charAt(i+2))} } && {
{"(S") .equals(""+dataReceive.charAt(i+3))} } )
            {
                temp_buffer.addElement(
temp_receive );
                temp_receive = "";
                break;
            }
            else if (
!{"("/") .equals(""+dataReceive.charAt(i))} )
                temp_receive = temp_receive +
dataReceive.charAt(i);
            else if {
{"("/") .equals(""+dataReceive.charAt(i))} )
            {
                temp_buffer.addElement(
temp_receive );
                temp_receive = "";
            }
        }
        System.out.println("Data Receive :
"+temp_buffer.size()+" size");
        System.out.println("Data :
"+temp_buffer.elementAt(0)+" "+temp_buffer.elementAt(1)+"
"+temp_buffer.elementAt(2)+" "+temp_buffer.elementAt(3)+"
"+temp_buffer.elementAt(4)+" "+temp_buffer.elementAt(5)+"
"+temp_buffer.elementAt(6)+" "+temp_buffer.elementAt(7)+"
"+temp_buffer.elementAt(8)+" "+temp_buffer.elementAt(9));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if ( !(buffer1.isEmpty()) )
        {
            if (
hostAddr.equals(""+(buffer1.elementAt(0))) )
            {
                if ( buffer1.size() <= 200 )
                {
                    for ( int i = 0 ;
i<temp_buffer.size() ; i++ )
                        buffer1.addElement(
(temp_buffer.elementAt(i)) );
                }
                else
                {
                    for ( int i = 0 ;
i<temp_buffer.size() ; i++ )
                    {
                        buffer1.removeElementAt(1);
                        buffer1.addElement(
(temp_buffer.elementAt(i)) );
                    }
                    Count1 = buffer1.size();
                    ECGContent1.Text_rxl.setText("Username
:"+ECGContent1.getNameformModell+" \n"+"IPAddress :
"+(ECG_Swing.buffer1.elementAt(0))+" \n"+"Other details : Receive
Packet_SEQ :"+ECG_Swing.buffer1.lastElement()+" \n"+"Buffer Size :
"+ECGContent1.count1);
                    for ( int i = 0 ; i <
buffer1.size()-2 ; i++ )
                    {
                        ECGContent1.temps1 =
""+(ECG_Swing.buffer1.elementAt((i+1)));
                        ECGContent1.tempi1 =
Integer.parseInt(ECGContent1.temps1);
                        ECGContent1.Amp_axis1[i] = ECGContent1.tempi1;
                        ECGContent1.count1 =
Count1-2;
                    }
                    ECGContent1.plot();
                }
            }
        }

        if ( !(buffer2.isEmpty()) )
        {
            if (
hostAddr.equals(""+(buffer2.elementAt(0))) )
            {
                if ( buffer2.size() <= 200 )
                {
                    for ( int i = 0 ;
i<temp_buffer.size() ; i++ )
                        buffer2.addElement(
(temp_buffer.elementAt(i)) );
                }
                else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {
            for ( int i = 0 ;
i<temp_buffer.size() ; i++ )
            {
                buffer2.removeElementAt(1);
                buffer2.addElement(
(temp_buffer.elementAt(i)) );
            }
            Count2 = buffer2.size();

            ECGContent2.Text_rx2.setText("Username
:"+ECGContent2.getNameformModel2+" \n"+"IPaddress :
"+(ECG_Swing.buffer2.elementAt(0))+" \n"+"Other details : Receive
Packet_SEQ :"+ECG_Swing.buffer2.lastElement()+" \n"+"Buffer Size :
"+ECGContent2.count2);
            for ( int i = 0 ; i <
buffer2.size()-2 ; i++ )
            {
                ECGContent2.temps2 =
""+(ECG_Swing.buffer2.elementAt((i+1)));
                ECGContent2.tempi2 =
Integer.parseInt(ECGContent2.temps2);
                ECGContent2.Amp_axis2[i] = ECGContent2.tempi2;
                ECGContent2.count2 =
Count2-2;
            }
            ECGContent2.plot();
        }
        if ( !(buffer3.isEmpty()) )
        {
            if (
hostAddr.equals(""+(buffer3.elementAt(0))) )
            {
                if ( buffer3.size() <= 200 )
                {
                    for ( int i = 0 ;
i<temp_buffer.size() ; i++ )
                    buffer3.addElement(
(temp_buffer.elementAt(i)) );
                }
                else
                {
                    for ( int i = 0 ;
i<temp_buffer.size() ; i++ )
                    {
                        buffer3.removeElementAt(1);
                        buffer3.addElement(
(temp_buffer.elementAt(i)) );
                    }
                }
                Count3 = buffer3.size();

                ECGContent3.Text_rx3.setText("Username
:"+ECGContent3.getNameformModel3+" \n"+"IPaddress :

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

"+(ECG_Swing.buffer3.elementAt(0))+" \n"+"Other details : Receive
Packet_SEQ :"+ECG_Swing.buffer3.lastElement()+" \n"+"Buffer Size :
"+ECGContent3.count3);
                                for ( int i = 0 ; i <
buffer3.size()-2 ; i++ )
                                {
                                                                ECGContent3.temps3 =
""+(ECG_Swing.buffer3.elementAt((i+1)));
                                                                ECGContent3.tempi3 =
Integer.parseInt(ECGContent3.temps3);
                                ECGContent3.Amp_axis3[i] = ECGContent3.tempi3;
                                                                ECGContent3.count3 =
Count3-2;
                                }
                                ECGContent3.plot();
                                }
                                if ( !(buffer4.isEmpty()) )
                                {
                                        if (
hostAddr.equals(""+(buffer4.elementAt(0))) )
                                        {
                                                if ( buffer4.size() <= 200 )
                                                {
                                                        for ( int i = 0 ;
i<temp_buffer.size() ; i++ )
                                                                buffer4.addElement(
(temp_buffer.elementAt(i)) );
                                                }
                                                else
                                                {
                                                        for ( int i = 0 ;
i<temp_buffer.size() ; i++ )
                                                                {
                                                                        buffer4.removeElementAt(i);
                                                                }
                                                                buffer4.addElement(
(temp_buffer.elementAt(i)) );
                                                                }
                                                        }
                                Count4 = buffer4.size();

                                ECGContent4.Text_rx4.setText("Username
:"+ECGContent4.getNameFormModel4+" \n"+"IPaddress :
"+(ECG_Swing.buffer4.elementAt(0))+" \n"+"Other details : Receive
Packet_SEQ :"+ECG_Swing.buffer4.lastElement()+" \n"+"Buffer Size :
"+ECGContent4.count4);
                                for ( int i = 0 ; i <
buffer2.size()-2 ; i++ )
                                {
                                                                ECGContent4.temps4 =
""+(ECG_Swing.buffer4.elementAt((i+1)));
                                                                ECGContent4.tempi4 =
Integer.parseInt(ECGContent4.temps4);
                                ECGContent4.Amp_axis4[i] = ECGContent4.tempi4;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Count4-2;
ECGContent4.count4 =
}
ECGContent4.plot();
}
}
if ( !(buffer5.isEmpty()) )
{
    if (
hostAddr.equals(""+(buffer5.elementAt(0))) )
    {
        if ( buffer5.size() <= 200 )
        {
            for ( int i = 0 ;
i<temp_buffer.size() ; i++ )
                buffer5.addElement(
(temp_buffer.elementAt(i)) );
        }
        else
        {
            for ( int i = 0 ;
i<temp_buffer.size() ; i++ )
            {
                buffer5.removeElementAt(1);
                buffer5.addElement(
(temp_buffer.elementAt(i)) );
            }
            Count5 = buffer5.size();
            ECGContent5.Text_rx5.setText("Username
:"+ECGContent5.getNameformModel5+" \n"+"IPaddress :
"+(ECG_Swing.buffer5.elementAt(0))+" \n"+"Other details : Receive
Packet_SEQ :"+ECG_Swing.buffer5.lastElement()+" \n"+"Buffer Size :
"+ECGContent5.count5);
            for ( int i = 0 ; i <
buffer5.size()-2 ; i++ )
            {
                ECGContent5.temps5 =
""+(ECG_Swing.buffer5.elementAt((i+1)));
                ECGContent5.tempi5 =
Integer.parseInt (ECGContent5.temps5);
                ECGContent5.Amp_axis5[i] = ECGContent5.tempi5;
                ECGContent5.count5 =
Count5-2;
            }
            ECGContent5.plot();
        }
    }
}
}
}
//}
//***** End method about Network
*****

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//***** Create new ECG Window
*****

//Find Username
String FindUsername(int user_select_index)
{
    String Username_at_selectIndex =
""+UAdetailModel.getValueAt( (user_select_index-1),0);
    String temp_username = "";
    for ( int i = 2 ;
i<Username_at_selectIndex.length()-1 ; i++ )
    {
        if (
!(""+Username_at_selectIndex.charAt(i)).equals(" ") )
            temp_username =
temp_username+Username_at_selectIndex.charAt(i);
        else
            break;
    }
    Username_at_selectIndex = temp_username;
    return Username_at_selectIndex;
}
//End Find Username

//Find Address
String FindAddr(int user_select_Addr)
{
    String Addr_at_selectIndex =
""+UAdetailModel.getValueAt((user_select_Addr-1),1);
    return Addr_at_selectIndex;
}
//End find Address

//Find unuse buffer for new User
public int FindBuffcanUse()
{
    int i = 1;
    int checkEmpty = 0;
    if ( !buffer1.isEmpty() )
        checkEmpty = i;
    else if ( !buffer2.isEmpty() )
        checkEmpty = i+1;

    else if ( !buffer3.isEmpty() )
        checkEmpty = i+2;

    else if ( !buffer4.isEmpty() )
        checkEmpty = i+3;

    else if ( !buffer5.isEmpty() )
        checkEmpty = i+4;

    return checkEmpty;
}
//End Find unuse buffer for new User

//Create new Tabbed
public void Create_new_Tabbed(String Username,String
Address,int getCreateIndex,int BuffUse)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        System.out.println("Username : "+Username+"
\n"+"IPAddress : "+Address+" \n"+"SelectIndex : "+getCreateIndex+"
\n"+"Buffer use : "+BuffUse);
        if ( getCreateIndex != 0 && getCreateIndex <=
nowindex )
        {
            int check = -1;
            for ( int i = 0 ;
i<UAdetailModel.getRowCount() ; i++ )
            {
                if (
("Running".equals(""+(UAdetailModel.getValueAt((getCreateIndex-
1),2)))) )
                {
                    check = i;
                    break;
                }
            }
            if ( check != -1 )
                JOptionPane.showMessageDialog(null,"The
user you specified already
exists", "Error", JOptionPane.ERROR_MESSAGE);
            else
            {
                switch (BuffUse)
                {
                    case 1:
tabbedPane.addTab(""+Username+" ECG",ECGContent1);
                UAdetailModel.setValueAt("Running", Integer.parseInt(UAindexShow
.getText())-1,2);
                buffer1.removeAllElements();
                buffer1.add(0,Address);
                break;
                    case 2:
tabbedPane.addTab(""+Username+" ECG",ECGContent2);
                UAdetailModel.setValueAt("Running", Integer.parseInt(UAindexShow
.getText())-1,2);
                buffer2.removeAllElements();
                buffer2.add(0,Address);
                break;
                    case 3:
tabbedPane.addTab(""+Username+" ECG",ECGContent3);
                UAdetailModel.setValueAt("Running", Integer.parseInt(UAindexShow
.getText())-1,2);
                buffer3.removeAllElements();
                buffer3.add(0,Address);
                break;
                    case 4:
tabbedPane.addTab(""+Username+" ECG",ECGContent4);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        UAdetailModel.setValueAt("Running", Integer.parseInt(UAindexShow
.getText())-1,2);

        buffer4.removeAllElements();

        buffer4.add(0,Address);

                                                break;
                case 5:
tabbedPane.addTab(""+Username+" ECG",ECGContent5);

        UAdetailModel.setValueAt("Running", Integer.parseInt(UAindexShow
.getText())-1,2);

        buffer5.removeAllElements();

        buffer5.add(0,Address);

                                                break;
    }
}
}
else
    JOptionPane.showMessageDialog(null,"Index
don't have User","Error",JOptionPane.ERROR_MESSAGE);
}
//End Create new Tabbed
//Remove Tabbed
public int FindBuff(String Address)
{
    int i = 1;
    int checkEmpty = 0;
    if ( !buffer1.isEmpty() )
    {
        if (
Address.equals(""+(buffer1.elementAt(0))) )
        {
            checkEmpty = i;
            System.out.println("Find buff
:"+checkEmpty);
        }
    }
    if ( !buffer2.isEmpty() )
    {
        if (
Address.equals(""+(buffer2.elementAt(0))) )
        {
            checkEmpty = i+1;
            System.out.println("Find buff
:"+checkEmpty);
        }
    }
    if ( !buffer3.isEmpty() )
    {
        if (
Address.equals(""+(buffer3.elementAt(0))) )

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {
            checkEmpty = i+2;
            System.out.println("Find buff
:"+checkEmpty);
        }
    }
    if ( !buffer4.isEmpty() )
    {
        if (
Address.equals(""+(buffer4.elementAt(0))) )
        {
            checkEmpty = i+3;
            System.out.println("Find buff
:"+checkEmpty);
        }
    }
    if ( !buffer5.isEmpty() )
    {
        if (
Address.equals(""+(buffer5.elementAt(0))) )
        {
            checkEmpty = i+4;
            System.out.println("Find buff
:"+checkEmpty);
        }
    }
    return checkEmpty;
}
public void Remove_Tabbed(int getCreateIndex,String
Username,int BuffUseR)
{
    switch (BuffUseR)
    {
        case 1: buffer1.removeAllElements();
                break;
        case 2: buffer2.removeAllElements();
                break;
        case 3: buffer3.removeAllElements();
                break;
        case 4: buffer4.removeAllElements();
                break;
        case 5: buffer5.removeAllElements();
                break;
    }
    String tempName = ""+Username+" ECG";
    for ( int i = 2 ; i<tabbedPane.getTabCount() ; i++
)
    {
        if (
tempName.equals(""+(tabbedPane.getTitleAt(i))) )
        {
            tabbedPane.remove(i);
            break;
        }
    }
    UAdetailModel.setValueAt("",getCreateIndex-1,2);
    System.out.println("buff1 size :"+buffer1.size());
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        System.out.println("buff2 size :"+buffer2.size());
        System.out.println("buff3 size :"+buffer3.size());
        System.out.println("buff4 size :"+buffer4.size());
        System.out.println("buff5 size :"+buffer5.size());
    }

//***** End Method Create new Window
*****

//***** implement ActionListener
*****

//implement ActionListener
public void actionPerformed(ActionEvent e)
{
    if (e.getSource() == NCAddButton)
    {
        int Index = addData(index);
        index = Index;
        resetTextField();
    }
    if ( e.getSource() == AddUserName)
        NCAddButton.doClick();
    if (e.getSource() == NCClearButton)
    {
        int tempIndex1 = findSelectIndex();
        String tempName =
""+NCdetailModel.getValueAt(tempIndex1,0);
        removeData(tempIndex1,tempName);
        index = --index;
        resetTextField();
    }
    if (e.getSource() == NCRResetButton)
    {
        resetAll();
        index = 0;
        nowindex = 0;
    }
    if (e.getSource() == NCCheckButton)
    {
    }

    if (e.getSource() == ShowButton)
    {
        int getIndex_Selected =
Integer.parseInt(UAindexShow.getText());
        int buffUse = FindBuffcanUse();
        String getUsername_Selected=
FindUsername(getIndex_Selected);
        String getAddr_Selected =
FindAddr(getIndex_Selected);

        Create_new_Tabbed(getUsername_Selected,getAddr_Selected,getInde
x_Selected,buffUse);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (e.getSource() == RemoveButton)
        {
            int getIndex_Selected =
Integer.parseInt(UAindexShow.getText());
            String getAddr_Selected =
FindAddr(getIndex_Selected);
            int buffUser = FindBuff(getAddr_Selected);
            String getUsername_Selected =
FindUsername(getIndex_Selected);
            if ( tabbedPane.getTabCount() > 2 )

Remove_Tabbed(getIndex_Selected, getUsername_Selected, buffUser);
        }

        if (e.getSource() == AddButton)
        {
            int Nowindex =
AddUser_showECG(index, nowindex);
            nowindex = Nowindex;
            UAindexShow.setText("");
        }

        if (e.getSource() == UAindexShow)
            AddButton.doClick();
    }
}
//*****End implement ActionListener*****
}
class DateTime extends JPanel implements Runnable
{
    Thread runner;
    GregorianCalendar g;
    String now = "Date";
    JLabel date;

    DateTime()
    {
        date = new JLabel();
        add(date);
        runner = new Thread(this);
        runner.start();
    }

    public void run()
    {
        while(true) {
            try
            {
                g = new GregorianCalendar();
                now = "Date : " + g.get(g.DAY_OF_MONTH) + "/"
+ (g.get(g.MONTH) + 1) + "/" + g.get(g.YEAR);
                String hour = "" + g.get(g.HOUR_OF_DAY);
                if(hour.length() == 1)hour = "0" + hour;
                String minute = "" + g.get(g.MINUTE);
                if(minute.length() == 1)minute = "0" +
minute;

                String second = "" + g.get(g.SECOND);
                if(second.length() == 1) second = "0" +
second;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        now = now + " Time : " + hour + ":" + minute
+ ":" + second;
        date.setText(now);
        runner.sleep(1000);
    }

    catch (Exception error)
    {
    }
}

}

class ECG1_Container extends JPanel
{
    static JTextArea Text_rx1;
    int [] x_axis1;
    static Vector Paint_Buffer1;
    static String getNameformModell, IPaddrformModell, temps1;
    static int temp1, count1;
    static int [] Amp_axis1;
    static int check1;

    ECG1_Container()
    {
        setLayout(new BorderLayout());
        JPanel RxContainer = new JPanel();
        JPanel RxPanel = new JPanel();
        Text_rx1 = new JTextArea(6,62);
        JScrollPane TextPane = new
JScrollPane(Text_rx1,ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED
,ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS);

        Text_rx1.setBorder(BorderFactory.createEmptyBorder(10,0,10,0));
        TitledBorder Text_rx_bord = new TitledBorder("Packet
receive from host : ");
        Text_rx_bord.setTitleColor(Color.magenta);
        RxPanel.setBorder(Text_rx_bord);
        RxPanel.add(TextPane);
        RxContainer.add(RxPanel);

        RxContainer.setBorder(BorderFactory.createEmptyBorder(18,0,0,0)
);

        add(RxContainer, BorderLayout.SOUTH);
        JPanel Empty = new JPanel();
        JPanel Empty1 = new JPanel();
        JPanel Empty2 = new JPanel();

        Empty.setBorder(BorderFactory.createEmptyBorder(10,0,9,0));
        Empty1.setBorder(BorderFactory.createEmptyBorder(10,2,9,2));
        Empty2.setBorder(BorderFactory.createEmptyBorder(10,2,9,3));
        add(Empty, BorderLayout.NORTH);
        add(Empty1, BorderLayout.WEST);
        add(Empty2, BorderLayout.EAST);
        setBackground(Color.black);

        x_axis1 = new int[208];
        int runx = 65;
        for ( int i = 0 ; i < 208 ; i++ )

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {
            x_axis1[i] = runx;
            runx = runx+2;          // for set x_axis1 => x_axis
1[65] - x_axis 1[480]
        }

        Amp_axis1 = new int[208];

        Paint_Buffer1 = new Vector();

    }

    public void paintComponent(Graphics g)
    {
        super.paintComponent(g);
        Color Rim = new Color(200,200,200);
        g.setColor(Rim);
        g.drawRect(14,29,467,303);

        g.setColor(Color.yellow);
        g.drawLine(65,30,65,275);
        g.drawLine(65,275,480,275);

        g.drawString("1.0",40,100);
        g.drawString("0.5",40,188);
        g.drawString("0.0",40,276);
        g.setFont(new Font("Time New Roman",0,11));
        g.drawString("Time",450,290);
        g.drawString("Amplitude",20,50);

        Color backLine = new Color(0,125,0);
        g.setColor(backLine);
        int x1,x2,y1,y2;
        for ( int x = 65 ; x<480 ; x = x+15)
        {
            y1 = 30;
            y2 = 275;
            g.drawLine(x,y1,x,y2);
        }
        for ( int y = 275 ; y>30 ; y = y-15)
        {
            x1 = 65;
            x2 = 480;
            g.drawLine(x1,y,x2,y);
        }

        g.setColor(Color.yellow);
        g.drawLine(65,30,65,275);
        g.drawLine(65,275,480,275);

        g.setColor(Color.green);
        g.drawPolyline(x_axis1,Amp_axis1,count1);
    }

    public void plot()
    {
        repaint();
    }

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

class ECG2_Container extends JPanel
{
    static JTextArea Text_rx2;
    static int [] x_axis2;
    static Vector Paint_Buffer2;
    static String getNameformModel2, IPaddrformModel2, temps2;
    static int tempi2, count2;
    static int [] Amp_axis2;
    static int checkc2;

    ECG2_Container()
    {
        setLayout(new BorderLayout());
        JPanel RxContainer = new JPanel();
        JPanel RxPanel = new JPanel();
        Text_rx2 = new JTextArea(6,62);
        JScrollPane TextPane = new
JScrollPane(Text_rx2,ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED
,ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS);

        Text_rx2.setBorder(BorderFactory.createEmptyBorder(10,0,10,0));
        TitledBorder Text_rx_bord = new TitledBorder("Packet
receive from host : ");
        Text_rx_bord.setTitleColor(Color.magenta);
        RxPanel.setBorder(Text_rx_bord);
        RxPanel.add(TextPane);
        RxContainer.add(RxPanel);

        RxContainer.setBorder(BorderFactory.createEmptyBorder(18,0,0,0)
);
        add(RxContainer, BorderLayout.SOUTH);
        JPanel Empty = new JPanel();
        JPanel Empty1 = new JPanel();
        JPanel Empty2 = new JPanel();

        Empty.setBorder(BorderFactory.createEmptyBorder(10,0,9,0));
        Empty1.setBorder(BorderFactory.createEmptyBorder(10,2,9,2));
        Empty2.setBorder(BorderFactory.createEmptyBorder(10,2,9,3));
        add(Empty, BorderLayout.NORTH);
        add(Empty1, BorderLayout.WEST);
        add(Empty2, BorderLayout.EAST);
        setBackground(Color.black);

        x_axis2 = new int[208];
        int runx = 65;
        for ( int i = 0 ; i < 208 ; i++ )
        {
            x_axis2[i] = runx;
            runx = runx+2;
        }

        Amp_axis2 = new int[220];

        Paint_Buffer2 = new Vector();
    }

    public void paintComponent(Graphics g)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

super.paintComponent(g);
Color Rim = new Color(200,200,200);
g.setColor(Rim);
g.drawRect(14,29,467,303);

g.setColor(Color.yellow);
g.drawLine(65,30,65,275);
g.drawLine(65,275,480,275);

g.drawString("1.0",40,100);
g.drawString("0.5",40,188);
g.drawString("0.0",40,276);
g.setFont(new Font("Time New Roman",0,11));
g.drawString("Time",450,290);
g.drawString("Amplitude",20,50);

Color backLine = new Color(0,125,0);
g.setColor(backLine);
int x1,x2,y1,y2;
for ( int x = 65 ; x<480 ; x = x+15)
{
    y1 = 30;
    y2 = 275;
    g.drawLine(x,y1,x,y2);
}
for ( int y = 275 ; y>30 ; y = y-15)
{
    x1 = 65;
    x2 = 480;
    g.drawLine(x1,y,x2,y);
}

g.setColor(Color.yellow);
g.drawLine(65,30,65,275);
g.drawLine(65,275,480,275);

g.setColor(Color.green);
g.drawPolyline(x_axis2,Amp_axis2,count2);
}

public void plot()
{
    repaint();
}

}

class ECG3_Container extends JPanel
{
    static JTextArea Text_rx3;
    static int [] x_axis3;
    static Vector Paint_Buffer3;
    static String getNameformModel3,IPaddrformModel3,temps3;
    static int tempi3,ccunt3;
    static int [] Amp_axis3;
    static int checkc3;

    ECG3_Container()
    {
        setLayout(new BorderLayout());
        JPanel RxContainer = new JPanel();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        JPanel RxPanel = new JPanel();
        Text_rx3 = new JTextArea(6,62);
        JScrollPane TextPane = new
JScrollPane(Text_rx3,ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED
,ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS);

        Text_rx3.setBorder(BorderFactory.createEmptyBorder(10,0,10,0));
        TitledBorder Text_rx_bord = new TitledBorder("Packet
receive from host : ");
        Text_rx_bord.setTitleColor(Color.magenta);
        RxPanel.setBorder(Text_rx_bord);
        RxPanel.add(TextPane);
        RxContainer.add(RxPanel);

    RxContainer.setBorder(BorderFactory.createEmptyBorder(18,0,0,0)
);

        add(RxContainer, BorderLayout.SOUTH);
        JPanel Empty = new JPanel();
        JPanel Empty1 = new JPanel();
        JPanel Empty2 = new JPanel();

        Empty.setBorder(BorderFactory.createEmptyBorder(10,0,9,0));
        Empty1.setBorder(BorderFactory.createEmptyBorder(10,2,9,2));
        Empty2.setBorder(BorderFactory.createEmptyBorder(10,2,9,3));
        add(Empty, BorderLayout.NORTH);
        add(Empty1, BorderLayout.WEST);
        add(Empty2, BorderLayout.EAST);
        setBackground(Color.black);

        x_axis3 = new int[208];
        int runx = 65;
        for ( int i = 0 ; i < 208 ; i++ )
        {
            x_axis3[i] = runx;
            runx = runx+2;
        }

        Amp_axis3 = new int[220];
        Paint_Buffer3 = new Vector();
    }

    public void paintComponent(Graphics g)
    {
        super.paintComponent(g);
        Color Rim = new Color(200,200,200);
        g.setColor(Rim);
        g.drawRect(14,29,467,303);

        g.setColor(Color.yellow);
        g.drawLine(65,30,65,275);
        g.drawLine(65,275,480,275);

        g.drawString("1.0",40,100);
        g.drawString("0.5",40,188);
        g.drawString("0.0",40,276);
        g.setFont(new Font("Time New Roman",0,11));
        g.drawString("Time",450,290);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

g.drawString("Amplitude",20,50);

Color backLine = new Color(0,125,0);
g.setColor(backLine);
int x1,x2,y1,y2;
for ( int x = 65 ; x<480 ; x = x+15)
{
    y1 = 30;
    y2 = 275;
    g.drawLine(x,y1,x,y2);
}
for ( int y = 275 ; y>30 ; y = y-15)
{
    x1 = 65;
    x2 = 480;
    g.drawLine(x1,y,x2,y);
}

g.setColor(Color.yellow);
g.drawLine(65,30,65,275);
g.drawLine(65,275,480,275);

g.setColor(Color.green);
g.drawPolyline(x_axis3,Amp_axis3,count3);
}

public void plot()
{
    repaint();
}
}

class ECG4_Container extends JPanel
{
    static JTextArea Text_rx4;
    int [] x_axis4;
    static Vector Paint_Buffer4;
    static String getNameformModel4,IPaddrformModel4,temp4;
    static int tempi4,count4;
    static int [] Amp_axis4;
    static int checkc4;

    ECG4_Container()
    {
        setLayout(new BorderLayout());
        JPanel RxContainer = new JPanel();
        JPanel RxPanel = new JPanel();
        Text_rx4 = new JTextArea(6,62);
        JScrollPane TextPane = new
JScrollPane(Text_rx4,ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED
,ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS);

        Text_rx4.setBorder(BorderFactory.createEmptyBorder(10,0,10,0));
        TitledBorder Text_rx_bord = new TitledBorder("Packet
receive from host : ");
        Text_rx_bord.setTitleColor(Color.magenta);
        RxPanel.setBorder(Text_rx_bord);
        RxPanel.add(TextPane);
        RxContainer.add(RxPanel);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RxContainer.setBorder(BorderFactory.createEmptyBorder(18,0,0,0)
);
    add(RxContainer, BorderLayout.SOUTH);
    JPanel Empty = new JPanel();
    JPanel Empty1 = new JPanel();
    JPanel Empty2 = new JPanel();

Empty.setBorder(BorderFactory.createEmptyBorder(10,0,9,0));

Empty1.setBorder(BorderFactory.createEmptyBorder(10,2,9,2));

Empty2.setBorder(BorderFactory.createEmptyBorder(10,2,9,3));
    add(Empty, BorderLayout.NORTH);
    add(Empty1, BorderLayout.WEST);
    add(Empty2, BorderLayout.EAST);
    setBackground(Color.black);

    x_axis4 = new int[208];
    int runx = 65;
    for ( int i = 0 ; i < 208 ; i++ )
    {
        x_axis4[i] = runx;
        runx = runx+2;
    }

    Amp_axis4 = new int[220];
    Paint_Buffer4 = new Vector();
}

public void paintComponent(Graphics g)
{
    super.paintComponent(g);
    Color Rim = new Color(200,200,200);
    g.setColor(Rim);
    g.drawRect(14,29,467,303);

    g.setColor(Color.yellow);
    g.drawLine(65,30,65,275);
    g.drawLine(65,275,480,275);

    g.drawString("1.0",40,100);
    g.drawString("0.5",40,188);
    g.drawString("0.0",40,276);
    g.setFont(new Font("Time New Roman",0,11));
    g.drawString("Time",450,290);
    g.drawString("Amplitude",20,50);

    Color backLine = new Color(0,125,0);
    g.setColor(backLine);
    int x1,x2,y1,y2;
    for ( int x = 65 ; x<480 ; x = x+15)
    {
        y1 = 30;
        y2 = 275;
        g.drawLine(x,y1,x,y2);
    }
    for ( int y = 275 ; y>30 ; y = y-15)
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        x1 = 65;
        x2 = 480;
        g.drawLine(x1, y, x2, y);
    }

    g.setColor(Color.yellow);
    g.drawLine(65, 30, 65, 275);
    g.drawLine(65, 275, 480, 275);

    g.setColor(Color.green);
    g.drawPolyline(x_axis4, Amp_axis4, count4);
}

public void plot()
{
    repaint();
}
}

class ECG5_Container extends JPanel
{
    static JTextArea Text_rx5;
    int [] x_axis5;
    static Vector Paint_Buffer5;
    static String getNameformModel5, IPAddrformModel5, temps5;
    static int tempi5, count5;
    static int [] Amp_axis5;
    static int checkc5;

    ECG5_Container()
    {
        setLayout(new BorderLayout());
        JPanel RxContainer = new JPanel();
        JPanel RxPanel = new JPanel();
        Text_rx5 = new JTextArea(6, 62);
        JScrollPane TextPane = new
JScrollPane(Text_rx5, ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED
, ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS);

        Text_rx5.setBorder(BorderFactory.createEmptyBorder(10, 0, 10, 0));
        TitledBorder Text_rx_bord = new TitledBorder("Packet
receive from host : ");
        Text_rx_bord.setTitleColor(Color.magenta);
        RxPanel.setBorder(Text_rx_bord);
        RxPanel.add(TextPane);
        RxContainer.add(RxPanel);

        RxContainer.setBorder(BorderFactory.createEmptyBorder(18, 0, 0, 0)
);

        add(RxContainer, BorderLayout.SOUTH);
        JPanel Empty = new JPanel();
        JPanel Empty1 = new JPanel();
        JPanel Empty2 = new JPanel();

        Empty.setBorder(BorderFactory.createEmptyBorder(10, 0, 9, 0));
        Empty1.setBorder(BorderFactory.createEmptyBorder(10, 2, 9, 2));
        Empty2.setBorder(BorderFactory.createEmptyBorder(10, 2, 9, 3));
        add(Empty, BorderLayout.NORTH);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

add(Empty1, BorderLayout.WEST);
add(Empty2, BorderLayout.EAST);
setBackground(Color.black);

x_axis5 = new int[208];
int runx = 65;
for ( int i = 0 ; i < 208 ; i++ ) ,
{
    x_axis5[i] = runx;
    runx = runx+2;
}

Amp_axis5 = new int[208];

Paint_Buffer5 = new Vector();

}

public void paintComponent(Graphics g)
{
    super.paintComponent(g);
    Color Rim = new Color(200,200,200);
    g.setColor(Rim);
    g.drawRect(14,29,467,303);
    g.setColor(Color.yellow);
    g.drawLine(65,30,65,275);
    g.drawLine(65,275,480,275);
    g.drawString("1.0",40,100);
    g.drawString("0.5",40,188);
    g.drawString("0.0",40,276);
    g.setFont(new Font("Time New Roman",0,11));
    g.drawString("Time",450,290);
    g.drawString("Amplitude",20,50);
    Color backLine = new Color(0,125,0);
    g.setColor(backLine);
    int x1,x2,y1,y2;
    for ( int x = 65 ; x<480 ; x = x+15)
    {
        y1 = 30;
        y2 = 275;
        g.drawLine(x,y1,x,y2);
    }
    for ( int y = 275 ; y>30 ; y = y-15)
    {
        x1 = 65;
        x2 = 480;
        g.drawLine(x1,y,x2,y);
    }
    g.setColor(Color.yellow);
    g.drawLine(65,30,65,275);
    g.drawLine(65,275,480,275);
    g.setColor(Color.green);
    g.drawPolyline(x_axis5,Amp_axis5,count5);
}

public void plot()
{
    repaint();
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้