การประยุกต์ใช้งาน IEEE-488

APPLICATION IEEE-488

โดย

นาย ไตรรงค์ ตั้งตระกูล

นาย ประเมต มิตรชอบ

นาย โยธี อิฐรัตน์

วัน เดือน ปี 1 ส.ค. 2540

เลขทะเบียน 037097

เลขเรียกหนังสือ T 38140 ๓๗๒ก

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิชาอิเล็กทรอนิคส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประยุกต์ใช้งาน IEEE-488

APPLICATION IEEE-488

โดย

นาย ไตรรงค์ ตั้งตระกูล รหัส 36013110

นาย ประเมต มิตรชอบ รหัส 36013112

นาย โยธี อิฐรัตน์ รหัส 36013119

อาจารย์ที่ปรึกษา

รศ.ดร. มนัส สังวรศิลป์

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิชาอิเล็กทรอนิคส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2538

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารสาดกระบัง

เรื่อง การประยุกต์ใช้งาน IEEE-488

ผู้จัดทำ

นาย ไตรรงค์ ตั้งตระกูล รหัส 36013110

นาย ประเมต มิตรชอบ รหัส 36013112

นาย โยธี อิฐรัตน์ รหัส 36013119

..... อาจารย์ที่ปรึกษา

(ผศ.สุวรรณชัย .....)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

การประยุกต์ใช้งาน IEEE-488

APPLICATION IEEE-488

นาย ไตรรงค์ ตั้งตระกูล รหัส 36013110

นาย ประเมต มิตรชอบ รหัส 36013112

นาย โยธี อิฐรัตน์ รหัส 36013119

โครงงานนี้รับการตรวจสอบแล้ว พร้อมที่จะทำการสอบได้

.....

(.....)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประยุกต์ใช้งาน IEEE-488 (GPIB)

นายไตรรงค์ ตั้งตระกูล รหัส 36013110
นายประเมต มิตรชอบ รหัส 36013112
นายโยธี อิฐรัตน์ รหัส 36013119
รศ.ดร. มนัส สังวรศิลป์ อาจารย์ที่ปรึกษา

บทคัดย่อ

ปริญญานิพนธ์ นี้ เป็นการประยุกต์ใช้งาน IEEE-488 (GPIB) ซึ่งเป็นระบบเชื่อมต่อมาตรฐาน กล่าวถึง การควบคุมอุปกรณ์เครื่องมือวัดต่าง ๆ โดยการนำคอมพิวเตอร์มาใช้เป็นตัวควบคุมระบบ โครงงานนี้แบ่งออกเป็น 2 ส่วน ส่วนแรก คือ ส่วนแสดงผล และควบคุมอุปกรณ์เครื่องมือวัด โดยสร้างเป็นเมนูในการใช้งานอันได้แก่ WRITE ( ) ใช้ในการบันทึกสัญญาณที่ต้องการจะเก็บ READ ( ) ใช้ในการอ่านสัญญาณจากการบันทึก SET VOLT ( ) ใช้เซ็ตขนาดของสัญญาณที่จอมอนิเตอร์ SET GEN ( ) ใช้ในการควบคุม GENERATOR ส่วนที่สอง เป็นส่วนประยุกต์ใช้งาน คือ FRE-RES ( ) ใช้เพื่อหาผลตอบสนองความถี่ของวงจร เช่นความถี่คัทออฟ อัตราขยายวงจร ชนิดของวงจร ความชัน (SLOPE) ของวงจรที่จุดคัทออฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APLICATION IEEE-488

Trirong Tangtakun

Pramate Mitchob

Yotee Itrat

Manut Sungwaonsin Avisor

1995

ABSTRACT

This thesis is the application of IEEE-488 (GPIB) which is the standard for controlling the measurement equipment by computer. This thesis comprises two parts.

1. Display and control part : consists of the following functions

WRITE() = Recording the required signal

READ() = Reading the signal from the memory

SET VOLT() = Setting the signal amplitude

SET GEN() = Controlling the generator

2. Application part : consists of the following functions

FRE-RES() = Finding the response of the circuit,for instance, cut-off frequency of the amplifier circuit, type of the circuit and slope of the circuit at the cut-off point.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง		หน้า
บทที่ 1	บทนำ	1
บทที่ 2	ประวัติการพัฒนา IEEE-488	2
	2.1 โครงสร้างของ IEEE-488	2
	2.2 อุปกรณ์ที่มี GPIB ทำหน้าที่	2
	2.3 ขีดจำกัด IEEE-488	3
	2.4 รายละเอียดเกี่ยวกับ IEEE-488	3
	2.41 บัสข้อมูล	4
	2.42 สายสัญญาณข้อมูล	4
	2.43 สายแฮนด์เชค	4
	2.5 ความหมายของสัญญาณต่าง ๆ	4
	2.51 กลุ่มสัญญาณข้อมูล	4
	2.52 กลุ่มสัญญาณควบคุม	4
	2.53 กลุ่มสัญญาณการรับส่งข้อมูล	5
	2.6 การเชื่อมต่ออุปกรณ์ต่าง ๆ	6
	2.61 การเชื่อมต่อแบบต่อเนื่อง	6
	2.62 การเชื่อมต่อแบบกระจาย	6
	2.7. คำสั่งใช้งาน GPIB	8
	2.71 คำสั่งเจาะจงจุดหมาย	8
	2.72 กลุ่มสัญญาณครอบคลุม	8
	2.73 กลุ่มคำสั่งอุปกรณ์ตัวรับ	8
	2.74 กลุ่มคำสั่งอุปกรณ์ตัวส่ง	9
	2.75 กลุ่มคำสั่งรอง	9
บทที่ 3	CODE การทำงานของอุปกรณ์	11
	3.1 คำสั่งใช้งาน	12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

เรื่อง		หน้า
บทที่ 4	แอคทีฟ ฟิลเตอร์	15
	4.1 ข้อดีของ แอคทีฟ ฟิลเตอร์	16
	4.11 แอคทีฟ ฟิลเตอร์ แบ่งตามการทำงาน	16
	4.2 โลพาส ฟิลเตอร์ อันดับ 1	17
	4.3 โลพาส ฟิลเตอร์ อันดับ 2	17
	4.4 ไฮพาส ฟิลเตอร์ อันดับ 2	18
	4.5 ไฮพาส ฟิลเตอร์ อันดับ 3	19
บทที่ 5	การออกแบบอุปรณ์เครื่องมือวัดพื้นฐาน	24
	5.1 ส่วนแสดงผลและควบคุม	24
	5.11 ส่วนแสดงผล	24
	5.12 ส่วนควบคุม	24
	5.2 ส่วนประยุกต์ใช้งาน	25
บทที่ 6	การติดต่ออุปกรณ์	29
	6.1 ฟังค์ชั่นการติดต่อ hm 1007	29
	6.2 ฟังค์ชั่นการติดต่อ hm 8130	30
	6.3 ฟังค์ชั่น PLOT WAVE FROM	32
	6.4 ฟังค์ชั่น เขียน	33
	6.5 ฟังค์ชั่น อ่าน	34
	6.6 ฟังค์ชั่น ความถี่เรสปอร์น	34
	6.7 ฟังค์ชั่น เซตโวลท์	37
	6.8 ฟังค์ชั่น เซ็ต เจนเนอร์เรเตอร์	38
	6.9 ฟังค์ชั่น อันดับ	39
บทที่ 7	สรุปผล และวิจารณ์	41
	7.1 ส่วนแสดงผล และควบคุม	41
	7.2 การประยุกต์ใช้งาน	43
	7.3 โปรแกรมและประยุกต์ใช้งาน	54
หนังสืออ้างอิง		79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

การต่อระบบมาตรฐานในอดีตระบบอุตสาหกรรม ใช้เครื่องมือและอุปกรณ์ไม่มากชิ้นซึ่งสามารถต่อเชื่อมระบบเข้าด้วยกันได้ง่ายแต่ใน ปัจจุบันนี้อุตสาหกรรมได้เจริญไปอย่างรวดเร็วจึงทำให้ระบบอุตสาหกรรมมีขนาดใหญ่ และซับซ้อนมาก ซึ่งในโครงงานนี้ได้นำคอมพิวเตอร์ (COMPUTER) ใช้ทำงานเป็นตัวควบคุม โดยแบ่งเป็น 2 ส่วน ส่วนแรกเป็นการแสดงผล ส่วนที่สองเป็นการประยุกต์ ส่วนแสดงผลก็คือการนำสัญญาณที่แสดงผลบนจอสโคป มาแสดงผลบนจอมอนิเตอร์ (MONITER) ของคอมพิวเตอร์ส่วนการประยุกต์มี 4 ส่วน ซึ่งใช้ในการประยุกต์ใช้งานโดยมีเมนนูให้เลือก 4 ตัวคือ

1.1 WRITE ใช้บันทึกสัญญาณที่ต้องการจะเก็บ

1.2 READ ใช้ในการอ่านสัญญาณที่ได้จากการบันทึก

1.3 SET VOLTAGE ใช้เซ็ตขนาดสัญญาณที่มอนิเตอร์

1.4 SET GEN ใช้ในการควบคุม เจนเนอร์เรเตอร์ (GENERATER)

ส่วนที่ สอง คือส่วนประยุกต์ใช้งาน ได้สร้างเป็น คือ (FRE-RES) ใช้เพื่อการหาผลตอบสนอง ความถี่ ของวงจรหรือระบบ โดยจะแสดงค่า ความถี่ คัทออฟ , อัตราขยาย , ความชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ประวัติการพัฒนา IEEE-488

ดังที่กล่าวมาข้างต้นว่าการเชื่อมต่อทางอุตสาหกรรมในอดีต เป็นไปด้วยความยากลำบากและเสียค่าใช้จ่ายมากดังนั้นบริษัทผู้ผลิตเครื่องมือวัดต่างๆ ในประเทศสหรัฐอเมริกาจึงร่วมกัน จัดหาระบบเชื่อมต่อมาตรฐานขึ้นมาซึ่งประเทศเยอรมันก็มีการพัฒนาระบบเชื่อมต่อมาตรฐานเช่นกัน โดยความร่วมมือของ IEC (Internationnal Electrotechnical Commmission) จนกระทั่งในปี 1972สหรัฐอเมริกาโดยการนำของ IEEE (Institute Of Electrical And Electronic Enginners) จึงได้มีการประชุมเพื่อวางแผนพิจารณาระบบเชื่อมต่อมาตรฐานร่วมกัน

บริษัทฮิวเล็ตแพกการ์ดผู้ผลิตเครื่องมือวัดรายใหญ่ของอเมริกา ได้ทำการพัฒนาระบบเชื่อมต่อมาตรฐานอยู่ก่อนแล้วชื่อว่า HPIB (Hewwlett Packard Tnterface Bus) จึงได้เสนอโครงงานให้แก่ IEEE เพื่อพิจาราณาและได้การยอมรับในปี 1975 โดย IEEE จัดให้เป็นมาตรฐานลำดับที่ 488 ดังนั้นจึงได้ชื่อว่า IEEE-STD 488-1975 ซึ่งต่อมาได้พัฒนาเป็น IEEE 488-1978 หรือเป็นที่นิยมเรียกกันว่า GPIB (Gennerral Purpose Interface Bus)

2.1 โครงสร้างของ IEEE-488

ในระบบพื้นฐานของ GPIB จะประกอบด้วยอุปกรณ์พื้นฐานคือ ผู้ส่ง (Talker) ผู้รับ (Listenner) และ ผู้ควบคุม (Controllor)

- ผู้ส่ง ทำหน้าที่ส่งข้อมูลโดยในระบบสามารถมี ผู้ส่ง ได้หลายตัวแต่จะมีตัวเดียวเท่านั้นที่กำลังทำงานอยู่
- ผู้รับ ทำหน้าที่เป็นตัวรับข้อมูลโดยในระบบเดียวกันสามารถมี ผู้รับ ได้หลายตัวเช่นกัน แต่ ผู้รับสามารถทำได้ครั้งหลาย ๆ ตัวได้
- ผู้ควบคุม ทำหน้าที่ควบคุมอุปกรณ์ต่างๆในระบบโดยจะกำหนดให้ ผู้ส่ง ทำการส่งข้อมูล หรือกำหนดให้ ผู้รับ รับข้อมูล

2.2 อุปกรณ์ที่มี GPIB นั้นสามารถแบ่งหน้าที่ได้ดังนี้

- ทำหน้าที่เป็น ผู้ส่ง เท่านั้นเช่น เครื่องมือวัดเป็นต้น
- ทำหน้าที่เป็น ผู้รับ เท่านั้นเช่น เครื่องพิมพ์ (printer) เครื่องบันทึก (Recordder)
- ทำหน้าที่เป็นทั้ง ผู้ส่ง และ ผู้รับ เช่น คอมพิวเตอร์ เครื่องมือวัด
- ทำหน้าที่เป็น ผู้ส่ง ผู้รับ และ ผู้ควบคุม ในตัวเดียวกันเช่นคอมพิวเตอร์ควบคุมระบบ

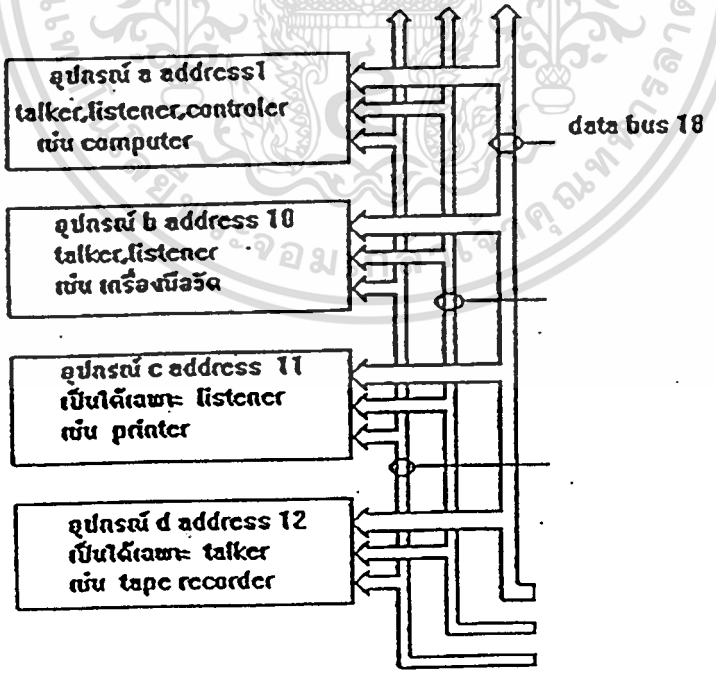
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ขีดจำกัดของ IEEE-488

- จำนวนอุปกรณ์ของระบบ (ผู้ส่ง , ผู้รับ , ผู้ควบคุม) ที่ต่อกับสายสัญญาณ 1 เส้นจะต้องไม่ไม่เกิน 15 เครื่อง
- สายเคเบิลที่ต่อระหว่างอุปกรณ์จะต้องยาวไม่เกิน 4 เมตร และความยาวของสายรวมในระบบจะต้องไม่เกิน 20 เมตร
- ความยาวในการส่งข้อมูลจะต้องไม่เกิน 1 Mb/Sec (1ล้านไบต์ต่อหนึ่งวินาที)
- จะต้องมีการจ่ายไฟให้กับอุปกรณ์มากกว่าครึ่งหนึ่งของระบบ

2.4 รายละเอียดเกี่ยวกับ IEEE-488

ลักษณะทางกายภาพของ IEEE-488 นั้นคือเป็นสายสัญญาณแบบ 24 เส้น ขนานกันและมีขั้วอยู่ทางปลายทางทั้งสองของสาย เพื่อต่อกับอุปกรณ์หรือต่อกันเพื่อให้สายสัญญาณมีความยาวเพิ่มขึ้นจำนวนสายสัญญาณ 24 เส้นนี้ มีเพียง 16 เส้นเท่านั้นที่ทำหน้าที่นำสัญญาณ ส่วนที่เหลืออีก 8 เส้นทำหน้าที่กราวด์ (GROUND) ชีลด์ (SHIELD) โดยจำนวนสายที่นำสัญญาณ 16 เส้นยังแบ่งได้เป็น 3 ประเภทตามรูปที่ 2.1 คือ



รูปที่ 2.1 บัสข้อมูล (DATA BUS) จำนวน 8 สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.41 บัสข้อมูล (Data Bus) จำนวน 8 สาย

DI012-DI08

2.42 สายสัญญาณควบคุม (Controller) จำนวน 5 สาย คือ

IFC (Interface Clear)

ATN (Attention)

SRQ (Service Request)

REN (Remote Enable)

EOI (Not Data Accepted)

2.43 สายแฮนด์เชค (HAND SHAKE) 3 สาย คือ

DAV (Data Vaild)

NRFD (Not Ready For Data)

NDAC (Not Data Accepted)

2.5 ความหมายของสัญญาณต่าง ๆ ภายใน IEEE-488

ดังที่กล่าวมาแล้วว่าสายสัญญาณ ต่าง ๆ ใน GPIB ได้แบ่งออกเป็น 3 กลุ่ม ในหัวข้อนี้จะอธิบายได้ดังนี้

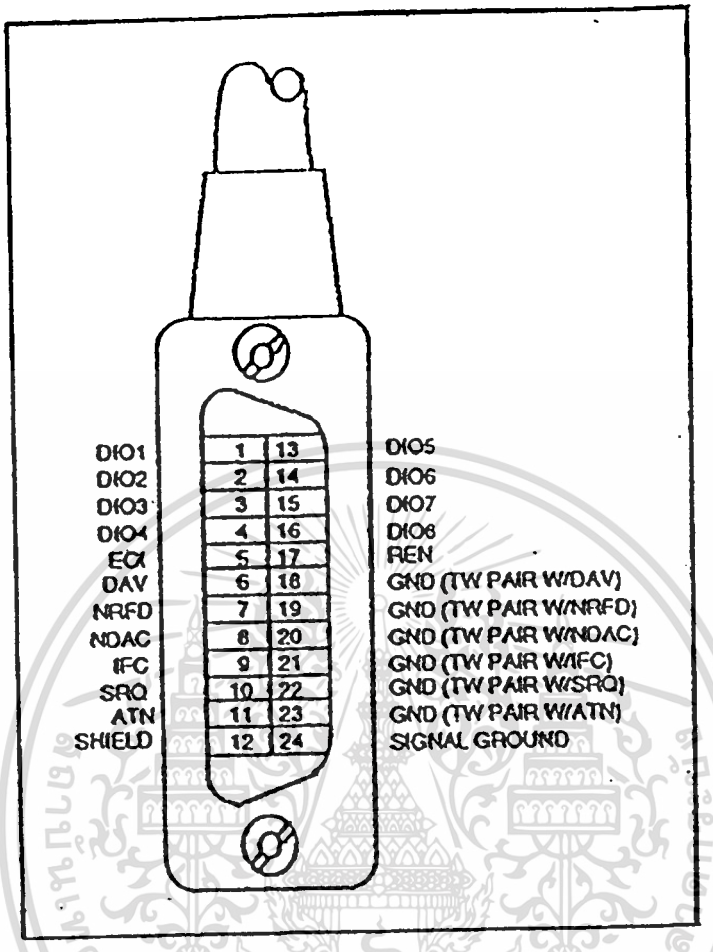
2.51 กลุ่มสัญญาณข้อมูล

DI01-DI08 (Data Input/Output) สัณฐาญทั้ง 8 เส้นนี้ทำหน้าที่เป็นทางผ่านข้อมูลของระบบ

2.52 กลุ่มสัญญาณควบคุมการเชื่อมต่อ (interface)

- IFC (Interface Clear) เป็นสัญญาณรีเซ็ตหรือเคลียร์ระบบ กำเนิดได้โดยตัวควบคุมเท่านั้นเมื่ออุปกรณ์ในบัสได้รับสัญญาณเคลียร์นี้ จะกลับคืนสภาวะเริ่มต้นใหม่ซึ่งเป็น สภาวะเริ่มก่อนการกำหนดฟังค์ชั่น เหมือนแรกเปิด (sw)
- ATN (Attention) เป็นสัญญาณที่ถูกส่งโดยอุปกรณ์ ที่เป็นตัวควบคุมเช่นเดียวกันใช้ในการให้อุปกรณ์ ทุกตัวในระบบเตรียมพร้อมเพื่อรอคำสั่งต่อไป
- SRQ (Service Request) เป็นสัญญาณที่ถูกส่งจากอุปกรณ์ต่าง ๆ เพื่อเป็นการบอกแก่ระบบว่าขณะนี้อุปกรณ์ดังกล่าวต้องการติดต่อจากตัวควบคุม
- REN (Remote Enable) สัญญาณนี้เป็นสัญญาณที่ถูกส่งจากตัวควบคุมเพียงตัวเดียวเท่านั้นเพื่อใช้สั่งให้อุปกรณ์ต่าง ๆ เปลี่ยนจากโหมดที่ใช้งานปกติมาเป็นการควบคุมโดยตัวควบคุมแทน
- EOI (End Or Identify) เป็นสัญญาณที่ถูกส่งได้ โดยอุปกรณ์ที่เป็นตัวควบคุมหรืออุปกรณ์ ที่เป็นตัวส่ง ก็ได้ ใช้สำหรับแสดงว่าข่าวสารที่ส่งเป็นชุดนั้นได้เสร็จสิ้นลงแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดง ขั้วของ GPIB และการจัดขาอุปกรณ์ต่าง ๆ

2.53 กลุ่มสัญญาณการรับส่งข้อมูล

- DAV (Data Valid) เมื่อสัญญาณนี้ถูกดึงเป็นลอจิก " low" โดยอุปกรณ์ที่เป็นตัวส่งเป็นการขณะนี้ตัวส่งได้ทำการส่งข้อมูลลงไปที่สายข้อมูลเรียบร้อยแล้ว
- NRFD (Nor Read For Data) เมื่อสัญญาณนี้มีลอจิกเป็น " low" จะเป็นการแสดงว่าขณะ นี้ระบบบัสยังไม่พร้อมท่จะรับข้อมูล เนื่องจากอุปกรณ์ในระบบยังพร้อมไม่หมดทุกตัว ซึ่งสัญญาณเส้นนี้จะไม่เป็น"HI" จนอุปกรณ์ทุกตัวให้ลอจิกเป็น "HI" ครบถ้วนแล้วสัญญาณนี้มีประโยชน์ในกรณีที่อุปกรณ์ระบบมีความเร็วที่แตกต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

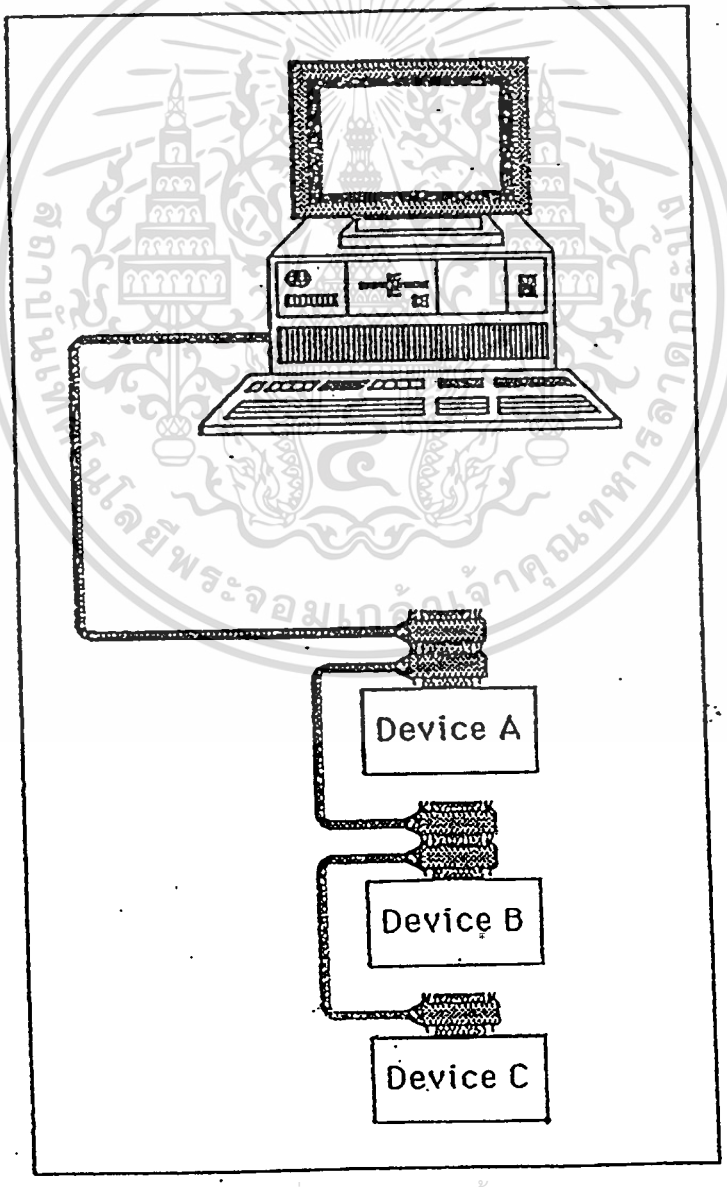
- NDAC (Not Data Accepted) สัญญาณเส้นนี้เป็นสัญญาณที่ถูกควบคุมโดยอุปกรณ์ที่เป็นตัวรับ โดยสัญญาณนี้จะมีลอจิกเป็น " HI " ในขณะอุปกรณ์ที่เป็นตัวกำลังเก็บ ข้อมูลจากสายข้อมูล และจะเป็น " HI " เมื่ออุปกรณ์นั้นได้ทำการอ่านเสร็จเรียบร้อยแล้วโดยสัญญาณลอจิกที่ใช้ใน(D1-D8) ของ IEEE-488 มีลักษณะเป็น คอมพลีเมนต์ทั้งหมดคือ " 1 " เท่ากับ (LOW) และ " 0 " เท่ากับ (HI)

2.6 การเชื่อมต่ออุปกรณ์ต่าง ๆ ในระบบ IEEE-488 BUS

สำหรับการเชื่อมต่อระหว่างอุปกรณ์ต่าง ๆ ในระบบ IEEE-488 นั้นมีอยู่ 2 วิธี คือ

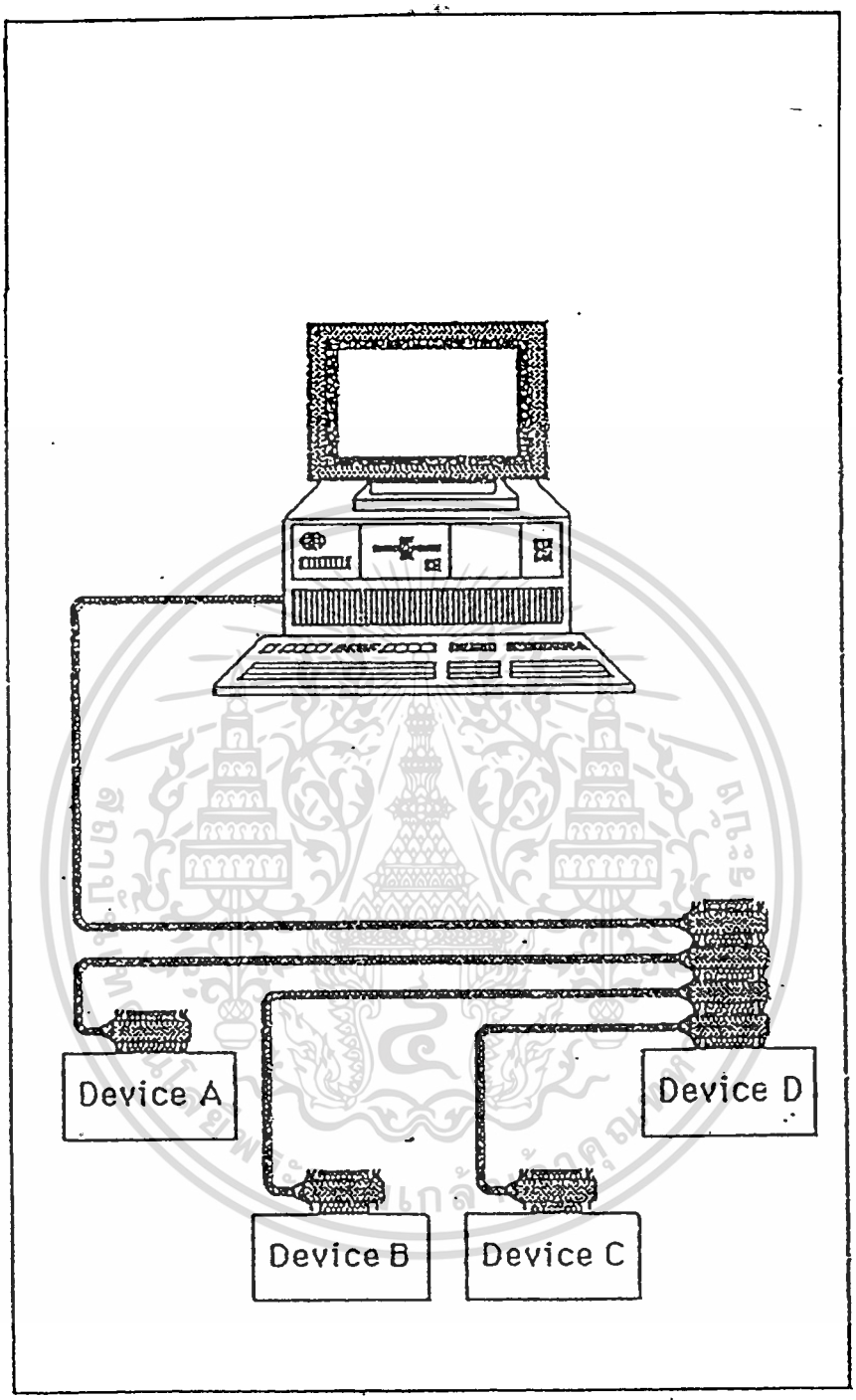
2.61 การเชื่อมแบบเรียงต่อเนื่อง (Daisy Chain Configuration)

2.62 การเชื่อมแบบกระจาย (Star Configuration)



รูปที่ 2.3 การเชื่อมต่อแบบเรียงต่อเนื่องกัน (Daisy Chain Configuration)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 การเชื่อมต่อแบบกระจาย (Star Configuration)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 คำสั่งใช้งานของ GPIB

การสั่งการต่าง ๆ เพื่อกำหนดหน้าที่การทำงานและกำหนดฟังค์ชั่น กำหนดช่องการวัด โหมดการวัดหรืออื่น ๆ แก่เครื่องวัดที่ต่ออยู่เหล่านี้ จะเป็นตัวกำหนดโดยการส่งรหัสคำสั่งไปที่อุปกรณ์โดยผ่าน DI1-DI6 รหัสคำสั่งนี้จะถูกส่งไปในช่วงที่สายสัญญาณ ATN เป็น LOW คำสั่งเพื่อกำหนดหน้าที่การทำงานต่าง ๆ ตามมาตรฐานของ GPIB มีอยู่ด้วยกัน 128 คำสั่งดังนี้ โดยแบ่งเป็น 5 กลุ่มคำสั่ง

รหัสที่ใช้ในระบบ GPIBบัสนั้นใช้ร่วมกันทั้งรหัสข้อมูลและรหัสคำสั่งนั้นคือ รหัสเดียวกันมีความหมายได้ 2 อย่างคือ ATN เป็น LOW จะหมายถึงรหัสคำสั่งแต่ถ้า ATN เป็น HIGH รหัสนี้จะแทนที่เป็นASCIIแทนซึ่งในตารางนี้ได้แบ่งความหมายออกเป็น 2 คอร์ลั่ม

2.71 คำสั่งเจาะจงจุดหมาย (addressed command group)เป็นคำสั่งที่เจาะจงไปยังอุปกรณ์ที่เป็นตัวส่งหรือกำหนดไว้ล่วงหน้าและคำสั่งนี้ประกอบด้วย

- GTL (got to local) สั่งให้อุปกรณ์กลับคืนสู่สภาพการควบคุมปกติด้วยมือ
- SDC (selected device clear) สั่งให้อุปกรณ์เคลียร์ตัวเองสู่สภาพเริ่มต้นใหม่
- PPC (paralled poll configure) เป็นคำสั่งที่สำหรับจัดสายของการกระทำขบวนการตรวจสอบสภาพอุปกรณ์ด้วยวิธีการขนานโดยใช้กับกลุ่มสำรอง
- GET (group excute trigger) ใช้ส่งเริ่มต้นคำสั่งอุปกรณ์ทีละหลายตัว
- TCT (take control) เป็นการกำหนดให้อุปกรณ์ตัวส่งทำหน้าที่เป็นตัวควบคุม

2.72 กลุ่มคำสั่งครอบคุม (universal command group) เป็นคำสั่งที่ส่งไปยังอุปกรณ์ทุกตัวที่อยู่ในบัสประกอบด้วย

- LLO (local lockout) เป็นการสั่งให้อุปกรณ์ล๊อกอยู่ที่อุปกรณ์ควบคุมโดยปุ่มปรับที่หน้าปัดตามปกติ
- DCL (device clear) สั่งให้อุปกรณ์ทุกตัวกลับไปสู่สภาวะเริ่มต้น
- PPC (parallel poll unconfigure) ใช้ยกเลิกขนวนการตรวจสอบสภาพแบบขนานทั้งหมด
- SPC (serial pollenabel) เปลี่ยนโหมดการตรวจสอบสภาพเป็นแบบอนุกรมในโหมดนี้จะเป็นการส่งสถานะของเครื่องแทนการส่งข้อมูล
- SPD (serial poll disable) ยกเลิกโหมดการตรวจสอบแบบอนุกรม

2.73 กลุ่มคำสั่งกำหนดอุปกรณ์ตัวรับ (listener address group) เป็นคำสั่งสำหรับกำหนดให้อุปกรณ์เป็นตัวรับตามรหัสหมายเลขจาก 0-30 และมีคำสั่ง UNT (untalker) สำหรับยกเลิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.74 กลุ่มคำสั่งอุปกรณ์ตัวส่ง (talker address group) สำหรับกำหนดให้อุปกรณ์เป็นตัวส่งตามรหัสหมายเลข 0-30 และมีคำสั่ง UNT (unlisten) สำหรับยกเลิกเช่นกัน

2.75 กลุ่มคำสั่งรอง (second commmand group) เป็นคำสั่งที่กำหนดรายละเอียดย่อมของอุปกรณ์แต่ละตัวที่ต่ออยู่กับระบบให้มีการทำงานอย่างไรตามจุดประสงค์ใช้งานของเครื่องมือ แช่นเดียวกับการปรับปุ่มต่าง ๆ ด้วยมือนั่นเอง คำสั่งรองนี้จะตามหลังคำสั่งหลักคือ จะใช้หลักจากอุปกรณ์ต่าง ๆ ถูกกำหนดวางตัวในระบบเรียบร้อยแล้ว

คำสั่งที่กล่าวไป ซึ่งใช้ในการกำหนดสภาวะการทำงานของอุปกรณ์แต่ละสภาวะที่กำหนดไปนั้นเป็นอย่างไรและมีจุดประสงค์เพื่ออะไรดังต่อไปนี้

Device clear / Interface Clear

- Device clear ใช้ในการให้อุปกรณ์ที่ต่ออยู่ในบัสกลับไปอยู่สภาวะเริ่มต้นยังไม่มีการกำหนดฟังค์ชั่นใด ๆ สภาวะเริ่มต้นนี้จะแตกต่างกันไปแล้วแต่ว่าอุปกรณ์นั้นออกแบบไว้อย่างไรมี 2 แบบคือ เคลียร์หมดทุก ตัวที่ต่ออยู่ (DCL) กับเคลียร์เจาะจงอุปกรณ์ใดอุปกรณ์หนึ่งแต่ว่าในการเคลียร์อุปกรณ์ตัวใดตัวหนึ่ง ให้อยู่ในสภาวะเริ่มต้นนั้นไม่ได้หมายความว่า Interface funtion ของ GPIB จะถูกเคลียร์อุปกรณ์ให้ไปอยู่ในสภาวะเริ่มต้นด้วยแต่อย่างใด interface funtion คือสภาพการ อินเตอร์เฟส ที่ได้กำหนดไว้ในระบบประกอบด้วยฟังค์ชั่นต่าง ๆ

Remote / Local

- remote เป็นการกำหนดให้อุปกรณ์ที่อยู่ในระบบ เช่น เครื่องมือวัดให้อยู่ในการควบคุมของอุปกรณ์ตัวตัวอื่นแทนซึ่งปุ่มปรับต่าง ๆ บนหน้าปัดเครื่องจะไม่มีผลต่อการทำงานส่วน local เป็นการควบคุมการทำงานของเครื่องมือวัดด้วยปุ่มปรับบนหน้าปัดปกติ การใช้ remote มีประโยชน์ในแง่ที่ขณะที่ตัวควบคุม เช่น คอมพิวเตอร์กำลังติดต่ออุปกรณ์ตัวนั้นอยู่หากไม่มีการตัดการควบคุม โดยปุ่มปรับบนจอหน้าปัดออกถ้ามีใครมาปรับแต่งก็จะทำให้การทำงานผิดพลาดไปได้การทำงานของ GPIB ใน remote and local มี 4 ลักษณะดังนี้
- LOCS ก็คือ local นั้นเอง เป็นสภาพการควบคุมที่ปุ่มตามปกติที่อยู่ในสภาพนี้ตอนที่เปิดเครื่องหรือ REN เป็น HIGH หรือเมื่อได้รับคำสั่ง GTL
- REMS คือ remote หมายถึงการตัดการควบคุมโดยปุ่มหน้าปัดออกจะเกิดขึ้นเมื่อ REN เป็น LOW และจะถูกล๊อกไว้เว้นแต่ว่าสวิตซ์ local ที่ตัวอุปกรณ์จะถูกเปลี่ยนไปตำแหน่งอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- RWLS เป็น remote ที่ถูกล็อกเอาไว้เช่นกันแต่การตัดการควบคุมตรงสวิตช์ที่ตัวอุปกรณ์ออกไปสภาพ remote เป็น RWLS จึงมีความสำคัญสูงกว่า REMS อย่างไรก็ตามยังถูกยกเลิกไว้ด้วยคำสั่ง
- LWLS มีสภาพเช่นเดียวกับ local แต่จะแตกต่างกันตรงสภาพ local โดย LWLS นี้เมื่อได้รับคำสั่งกำหนดอุปกรณ์ตัวรับจะเปลี่ยนไปอยู่ในสภาพแบบล็อกหรือ RWLS ทันทีในการที่จะมาที่สภาพ LWLS นี้ได้มี 2 กรณีคือ เมื่ออยู่ในสภาพ local ธรรมดา (locs) แล้วได้รับคำสั่ง LLO หรืออยู่ใน RWLS แล้วได้รับคำสั่ง GTL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

CODE การทำงานของอุปกรณ์ในระบบ

ในการใช้งาน CARD IEEE-488 ในการควบคุมอุปกรณ์อันได้แก่ PROGRAMMABLE FUNCTION GENERATOR (HM) OSCILLOSCOPE (HM 1007) PROGRAMMABLE MULTIMETER (HM 8112-2) POWER SUPLY (HM1007) ในการติดต่อกับอุปกรณ์ผ่าน CARD จะมีรูปแบบ COMMAND CODE โดยใช้ภาษาซีในการติดต่อผ่าน NI-488 จะมี COMMAND CODE รูปที่ 3.1 และ รูปที่ 3.2

Call Syntax	Description
ibbna (ud,bname)	Change access board of device
ibcac (ud,v)	Become Active Controller
ibclr (ud)	Clear specified device
ibcmd (ud,cmd,cnt)	Send commands from string
ibcmda (ud,cmd,cnt)	Send commands asynch. from string
ibconfig (ud,option,value)	Configure the software
ud = ibdev (bd_index, pad,sad,tmo,eot,eos)	Open and initialize an unused device when the device name is unknown
ibdma (ud,v)	Enable/disable DMA
ibeos (ud,v)	Change/disable EOS mode (write)
ibeot (ud,v)	Enable/disable END message
ibevent (ud,&event)	Return the next event
ud = ibfind (udname)	Open device and return unit descriptor
ibgts (ud,v)	Go from Active Controller to Standby
ibist (ud,v)	Set/clear ind. status bit for Parallel Polls
iblines (board,lines)	Get status of GPIB lines
ibln (pad,sad,listen)	Check for presence of device on bus.
ibloc (ud)	Go to Local
ibonl (ud,v)	Place device or board online/offline
ibpad (ud,v)	Change Primary Address
ibpct (ud)	Pass Control
ibppc (ud,v)	Parallel Poll Configure
ibrd (ud,rd,cnt)	Read data to string
ibrda (ud,rd,cnt)	Read data asynchronously to string

รูปที่ 3.1 แสดงตาราง โค๊ดการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Call Syntax	Description
ibrdf (ud,flname)	Read data to file
ibrpp (ud,&ppr)	Conduct a Parallel Poll
ibrsc (ud,v)	Request/release System Control
ibrsp (ud,&spr)	Return serial poll byte
ibrsv (ud,v)	Request service, set/change serial poll
ibsad (ud,v)	Change/disable Secondary Address
ibsic (ud)	Send Interface Clear for 100 µsec
ibsre (ud,v)	Set/clear Remote Enable line
ibsrq (func)	Register an SRQ "interrupt routine"
ibstop (ud)	Abort asynchronous operation
ibtmo (ud,v)	Change/disable time limit
ibtrap (mask,mode)	Configure Applications Monitor
ibtrg (ud)	Trigger selected device
ibwait (ud,mask)	Wait for selected event
ibwrt (ud,wrt,cnt)	Write data from string
ibwrta (ud,wrt,cnt)	Write data asynchronously from string
ibwrtf (ud,flname)	Write data from file

รูปที่ 3.2 ตารางแสดง โค๊ดการทำงาน

3.1 คำสั่งที่ใช้งานกับ ฟังค์ชั่น เจนเนอร์เรเตอร์

เมื่อต่อฟังค์ชั่น เจนเนอร์เรเตอร์ และคอมพิวเตอร์โดยใช้บัสมาตรฐาน IEEE-488 คำสั่งคอมพิวเตอร์สามารถควบคุมการทำงานของฟังค์ชั่นเจนเนอร์เรเตอร์ ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Command With Data

SIN	Sinewave signal function
TR	Triangle signal function
SQR	Squarewave signal function
PLS	Pulse signal function
RMP	Sawtooth positive-going
RMN	Sawtooth negative-going
ARB	Arbitrary function
SW1/0	Sweep mode on/off
CTM	Continous mode
GTM	Gated mode
TRM	Triggered mode
OT1/0	Output signal on/off
OF1/0	Offset on/off
DFR	Display signal frequency
DST	Display start frequency
DSP	Display stop frequency
DWT	Display pluse time
DSW	Display pulse width
DAM	Display out amplitude
DOF	Display offset voltage
RMO	Disable remote status
LK1	Enable local-inhibit status
LK2	Disable local-inhibit status
TRG	Trigger a signal perriod (sweep off) or a complete sweep (sweep on)
CLR	Reset the function gennerrator and returns all setting to their default
ARC	Delete alll arbitary data and reset the internal arbitary counter to zero

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ARE Terminate the arbitary edito

Avialable commmand

FRQ : (Data)	Set frequency to	(.....) Hz
STT ; (Data)	Set start frequency to	(.....)Hz
STP ; (Data)	Set stop frequency to	(.....)Hz
SWT : (Data)	Set sweep time to	(.....)Hz
WDT ; (Data)	Set pulse width to	(.....)Hz

Command with whole-number value

STO=(Data)	Store values	(0.....8)
RCL=(Dta)	Read values	(0.....9)

Reset commands

FRQ	Frequency
STT	Start Frequency
STP	Stop frequency
SWT	Sweep Frequency
WDT	Pulse Width
AMP	Output Voltage
OFS	Offset
ARD	Arbitary Data
ID	Device ID
VER	Equipment verion
STA	Device Status

คำสั่งที่ใช้กับ สโคป

DIG (channal code)	Extraction of stored data from the scope memmory with channal selection
GET (channal code)	Actiation of the reset trigger function at the without channal selection
STA	Query for oscilloscope channal setting
TEA	Tex tranfer to the interface with the signal-data to the device

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ACTIVE FILTER

Active Filter ทำหน้าที่กรองความถี่สัญญาณ เป็นวงจรที่ประกอบด้วยภาคขยายเช่น ทราซิสเตอร์ (Transistor) หรือ ไอซี (IC) และ network เลือกความถี่ (Frequency Selective network) จำพวก ความต้านทาน (Resistor) กับ คาปาซิสเตอร์ (Capacitor) วงจรให้ผ่านได้เฉพาะสัญญาณในช่วงความถี่ที่กำหนด ขณะเดียวกันจะขวางกั้น (block) หรือลดทอน สัญญาณนอกเหนือช่วงความถี่ดังกล่าวมิให้ปรากฏที่ output

กล่าวโดยทั่วไป ฟิลเตอร์ แบ่งออกเป็นหลายรูปแบบ

1. ฟิลเตอร์ ชนิด อนาล็อก หรือ ชนิด ดิจิตอล
2. ฟิลเตอร์ ประเภท พาสซีพ หรือ แอคทีฟ
3. ฟิลเตอร์ ย่านความถี่เสียง (Audio Frequency) หรือย่านวิทยุ (Radio Frequency)

อนาล็อก ฟิลเตอร์ ออกแบบมาเพื่อใช้กับสัญญาณ์ อนาล็อก ส่วน ดิจิตอล ใช้งานกับสัญญาณ อนาล็อก โดยอาศัยเทคนิคทาง ดิจิตอล มาช่วย ถ้ากำนึงถึงชิ้นส่วน พาสซีพ ได้แก่ตัวตัวต้านทาน, คาปาซิสเตอร์ และขดลวด ส่วนแอคทีพ ฟิลเตอร์ ประกอบด้าย ตัวขยายสัญญาณจำพวกทรานซิสเตอร์ หรือ ไอซี ในรูป ออฟแอมป์ และ ตัวต้านทาน และ คาปาซิสเตอร์ ทำงานร่วมกัน ตัวต้านทาน, คาปาซิสเตอร์ และ ขดลวด ถือว่าเป็นชิ้นส่วนประเภท พาสซีพ การจะเลือกใช้ชิ้นส่วนชนิดใดนั้น ขึ้นกับย่านความถี่สัญญาณที่ต้องการให้วงจร ฟิลเตอร์ ทำงาน ยกตัวอย่าง RC Filterใช้กับย่านความถี่เสียง (Audio Frequency) หรือ ใช้ในย่านความถี่ต่ำ (low frequency)

4.1 ข้อดีของ Active Filter ที่มีเหนือกว่า Passive Filtor

ข้อดีที่เหนือกว่ามีดังนี้

- การปรับอัตราการขยาย และปรับความถี่เป็นอิสระต่อกัน เราสามารถจัดค่าอัตราการขยายของ ออฟแอมป์ ชดเชยกับอัตราลดทอนสัญญาณของวงจร RC หรือเพิ่มอัตราการขยายของวงจรทั้งหมด ด้วยการจัดค่าชิ้นส่วนอุปกรณ์ในส่วนของวงจรขยาย ออปแอมป์ และการปรับ เปลี่ยนความถี่ก็อยู่ที่ค่า RC เท่านั้น
- ไม่มีปัญหา โหลดดิ่ง จากการที่ ออฟแอมป์มีคุณสมบัติของอินพุท อิพีแดนซ์สูงและเอ้าท์พุท อิมพีแดนซ์ ต่ำ วงจร แอคทีฟ อาศัย ออฟแอมป์ จึงไม่เกิดปัญหา โหลดดิ่งกับเอ้าท์พุท และ อินพุท ของวงจร ณ จุดที่นำ แอคทีฟ ฟิลเตอร์ เข้าไปต่อ

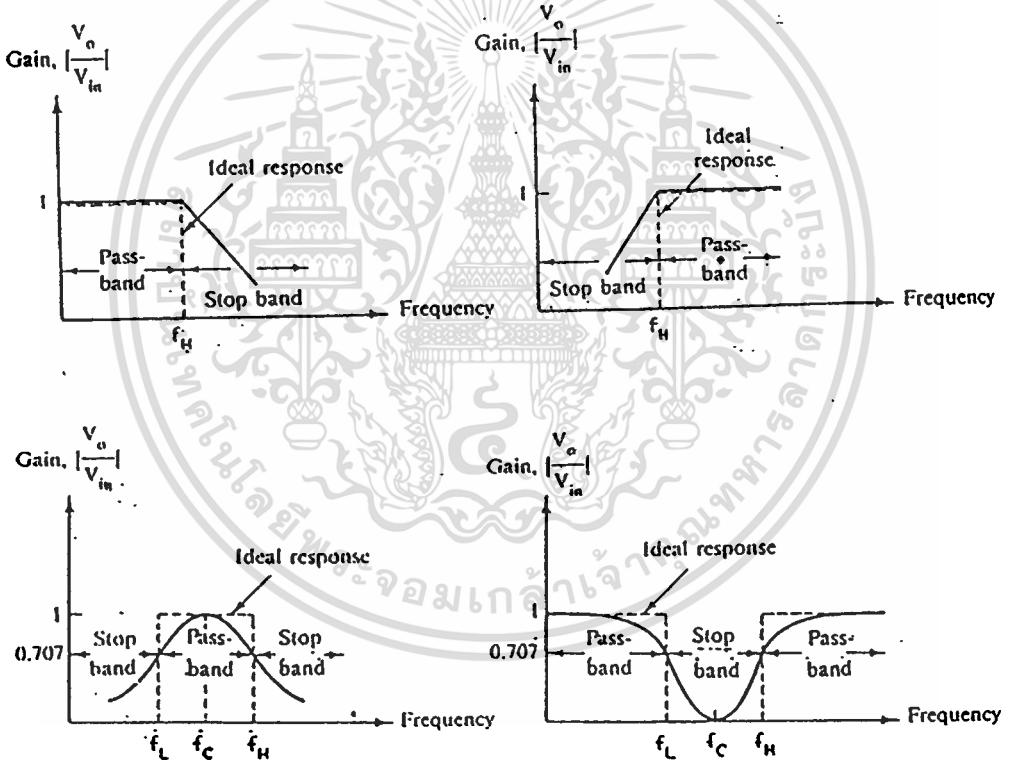
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

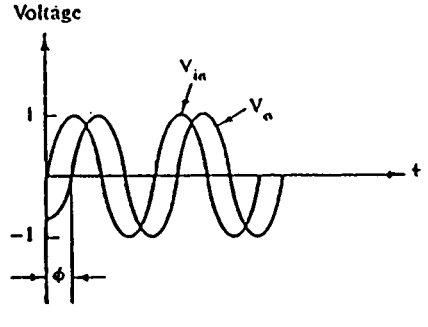
- ราคาถูก แอคทีฟ มีราคาถูกกว่า พาสซีพ เพราะไม่ต้องอาศัย ขดลวด ที่มีราคาแพง และยังใช้ออฟแอมป์ ซึ่งในปัจจุบันราคาในท้องตลาดถูกมาก

4.11 Active Filter ตามลักษณะการทำงาน

เราแบ่ง Active Filter ตามลักษณะการทำงานเป็น 5 ชนิด

- Low Pass Filter (LPF)
- High Pass Filter (HPF)
- Band Pass Filter (BPF)
- Band Reject หรือ Band Stop Filter (BSF)
- All Pass Filter





รูปที่ 4.1 แสดงคุณสมบัติของ แอคทีฟ ฟิลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เ ...อร์ ชนิด อันดับ

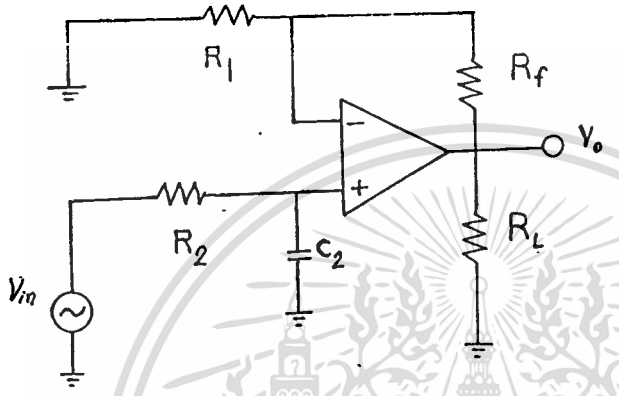
Vo/Vin = Af/1+j(F/Fh)

เมื่อ Vo/Vin = อัตราการขยายของวงจร

Af = 1+Rf/R1 = แบนพาสของวงจร

f = ความถี่ของสัญญาณอินพุท

Fh = 1/(2ΠR2C2)

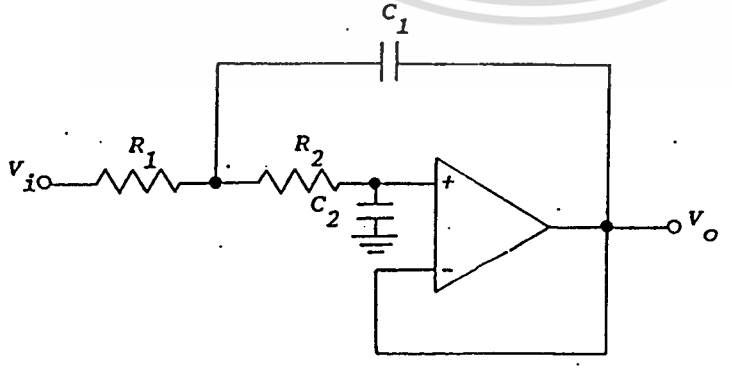


รูปที่ 4.2 แสดง วงจร บัสเตอร์เวิร์ต โลพาส อันดับ หนึ่ง

การออกแบบ

1. เลือกค่า Fh
2. เลือกค่า C2 = 10/Fh
3. คำนวณ R2 = 1/(2ΠFhC2)
4. คำนวณ Rf , R1 ตามค่าพาสแบนเกณฑ์

4.3 วงจร โลพาสฟิลเตอร์ ชนิด order 2



รูปที่ 4.3 แสดงวงจร โลพาส อันดับ สอง

ปกติการออกแบบวงจรฟิลเตอร์จะใช้คอมพิวเตอร์ช่วย เพื่อให้มีความรวดเร็วและค่าที่ได้ถูกต้องโดยเฉพาะอย่างยิ่งการคำนวณทรานเฟอร์ฟังค์ชั่น ซึ่งต้องการตัวเลขที่ละเอียดพอสมควร และสามารถคำนวณได้ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการออกแบบ

1. เลือกค่า $C''1$ และ $C''2$
2. แทนค่า Fcp ในสูตรคำนวณหาค่า $C''1$, $C''2$

$$C'1 = C''1/2\Pi Fcp \quad , \quad C'2 = C''2/2\Pi Fcp$$

3. เลือกค่า $R = R1 = R2$ ที่ทำให้ C1 , C2 ทำงานได้

$$C1 = C'1/R \quad , \quad C2 = C'2/R$$

การออกแบบวงจรกรองความถี่ต่ำอันดับที่ 2

ต้องการ Fcp = 1000Hz มียอดความถี่ประมาณ 3 db (บัสเตอร์เวิร์ท) ขนาด c = 0.01μF

ขั้นตอนการออกแบบ

1. จากรูป ตาราง ที่ จะได้ $C''1 = 1.414$, $C''2 = 0.7071$
2. $C'1 = C''2/2\Pi Fcp = 1.414/2\times3.14\times1000 = 2.25\times10^{-4}$

$C'2 = C''2/2\Pi Fcp = 0.7071/2\times3.14\times1000 = 1.125\times10^{-4}$

3. ค่า C2 จะเป็นค่าคาปาซิเตอร์ที่สูงสุด = 0.01 μF

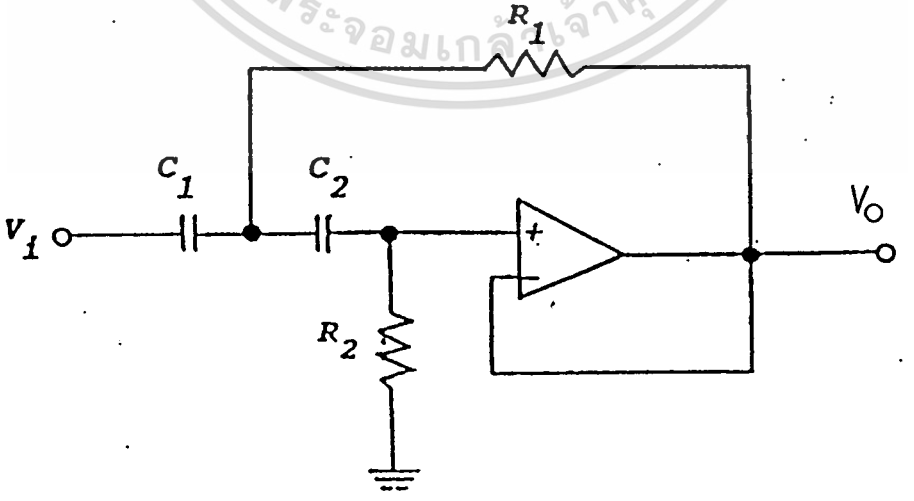
เลือกค่า $R = R1 = R2 = 10800\ \Omega$

$C1 = C'1/R = 2.25\times10 = 0.02\ \mu F$

$C2 = 0.01\ \mu F$

4.4 วงจรกรองความถี่สูง อันดับ สอง

วงจรจะไม่ตอบสนองต่อสัญญาณไฟตรงหรือสัญญาณความถี่ต่ำ มีเกณฑ์เท่ากับ 1 ตั้งแต่ความถี่ Fcp ถึงย่านที่เกณฑ์ของออฟแอมป์ลดต่ำลงกว่า 1 โดยเลือกค่า R1 , R2 , C1 , C2ที่เหมาะสม



รูปที่ 4.4 แสดง วงกรองความถี่สูง อันดับ สอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการออกแบบ

1. เลือกค่า $R'1$, $R'2$
2. คำนวณหาค่า C จากค่า Fcp

$$C = 1/2\Pi Fcp$$

3. เลือกค่า K ซึ่งแทนค่าสูตรแล้วทำให้ได้ค่า C1 , C2 ที่หาซื้อได้

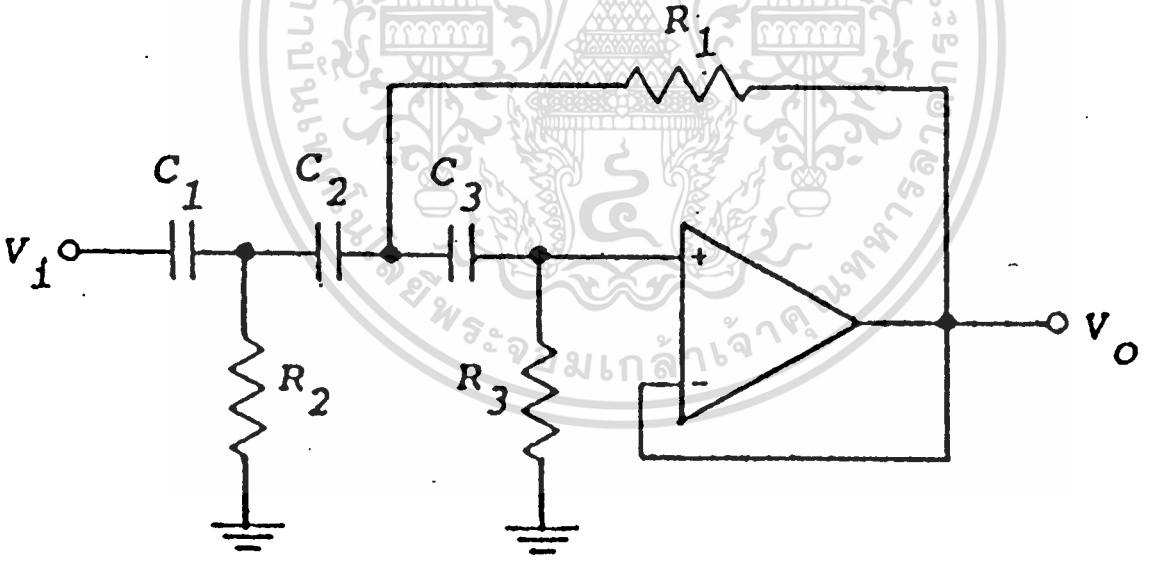
$$C1 = C2 = C/K$$

4. คำนวณค่าความต้านทาน

$$R1 = KR'1 , R2 = KR'2$$

4.4 วงจรกรองสัญญาณความถี่สูง อันดับ สาม

การทำงานของวงจรจะไม่ตอบสนองต่อสัญญาณความถี่ต่ำและมีเกณฑ์เป็น 1 ตลอดย่านความถี่ Fcp จนความถี่ที่เกณฑ์ของวงจรลดลงจาก 1 โดยการเลือกค่าที่เหมาะสมทั้ง 6 ตัว



รูปที่ 4.5 แสดง วงจรกรองความถี่สูง อันดับ สาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
037097

ขั้นตอนการออกแบบ

1. เลือกค่า $R'1$, $R'2$,$R'3$
2. หาค่า C โดยแทนค่า Fcp ที่กำหนด

$$C = 1/2\Pi Fcp$$

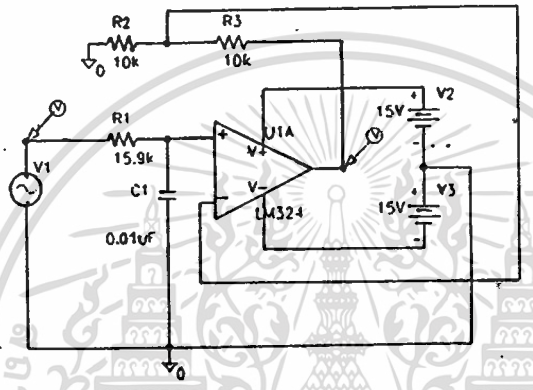
3. เลือกค่าคงที่ K แทนค่าในสูตร

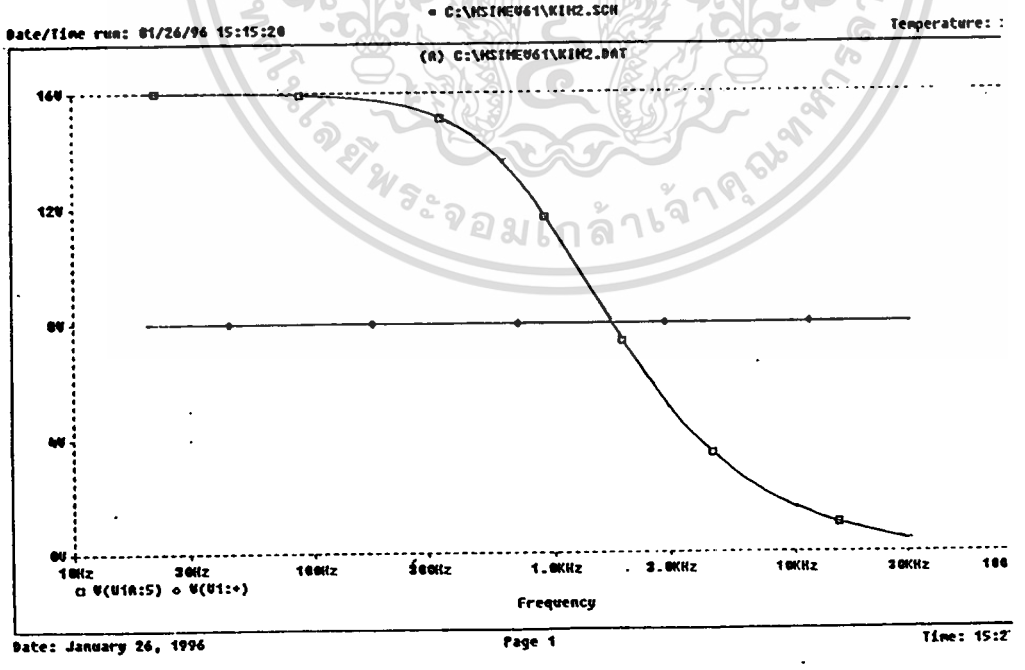
$$C1 = C2 = C3 + C/K$$

4. คำนวณหาค่าความต้านทาน

$$R1 = KR'1 \quad , \quad R2 = KR'2 \ , \ R3 = KR'3$$

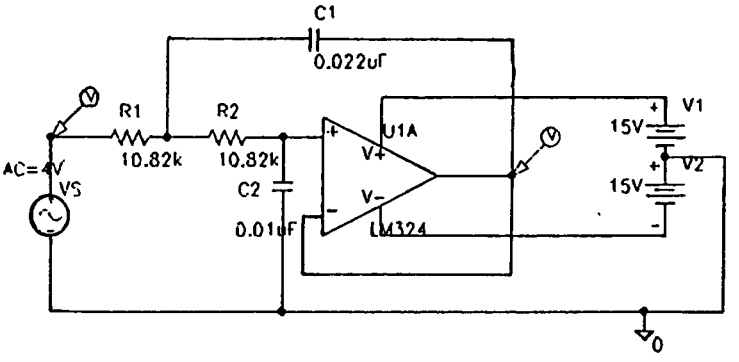
5. กำหนดย่านความถี่ที่ Av มากกว่าหรือเท่ากับ 100 โดยใช้ข้อมูลออฟแอมป์ประกอบ

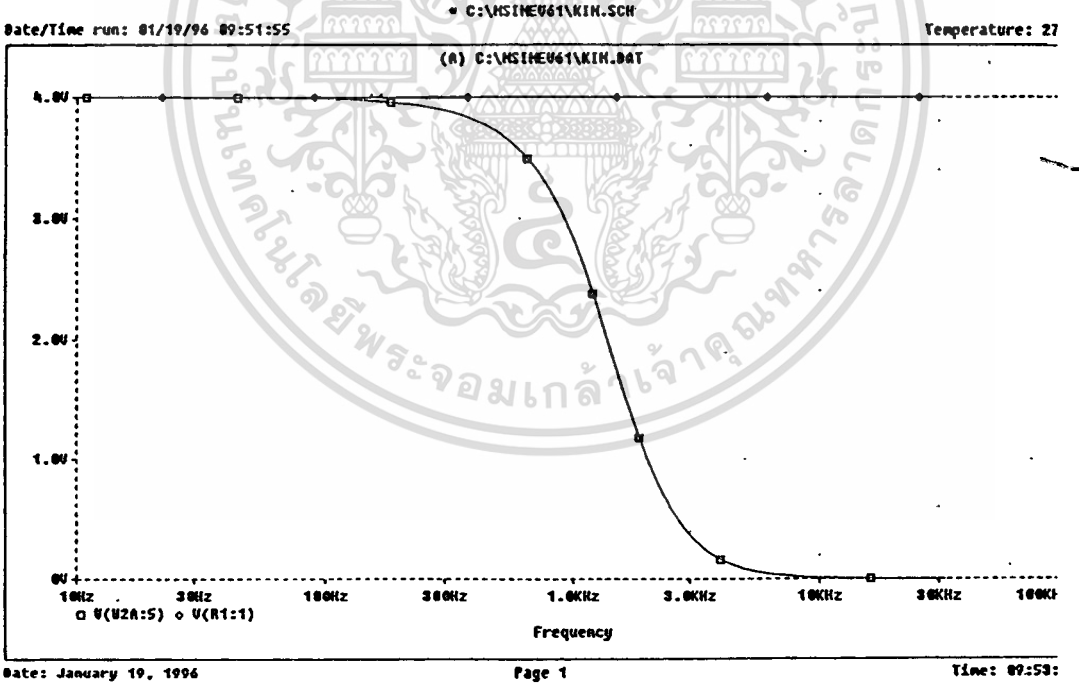




รูปที่ 4.6 วงจร โลพาส ฟิลเตอร์ อันดับ หนึ่ง และ กราฟ จากการทดลอง PSPICE

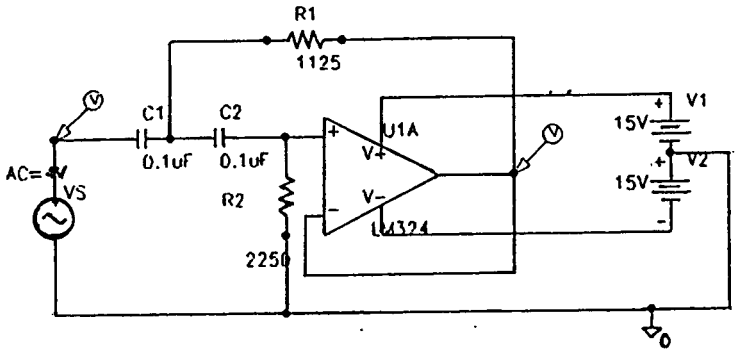
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

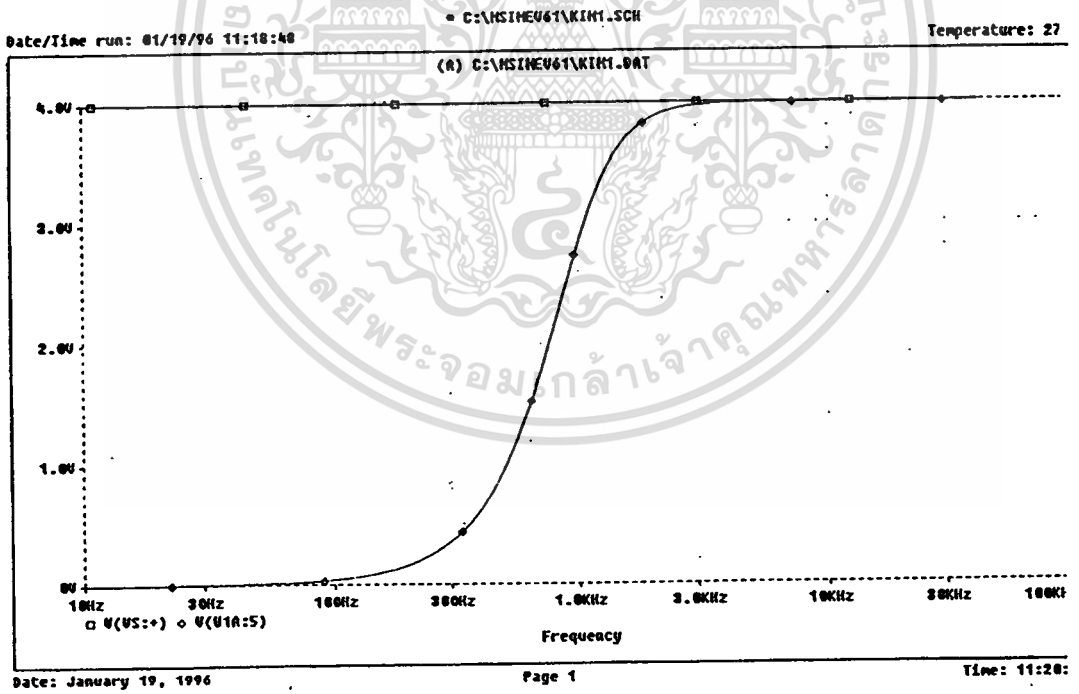




รูปที่ 4.7 วงจร โลพาส ฟิลเตอร์ อันดับ สอง และ กราฟ จากการทดลอง PSPICE

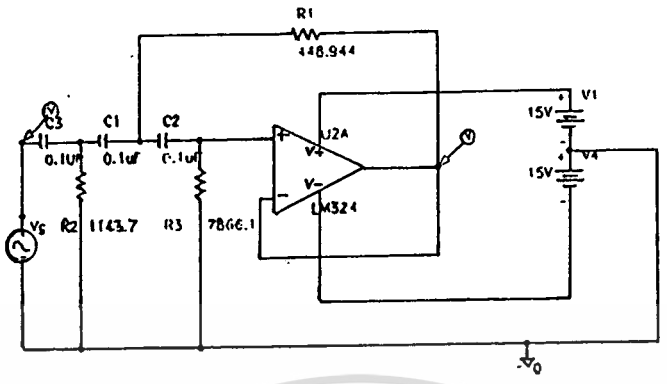
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

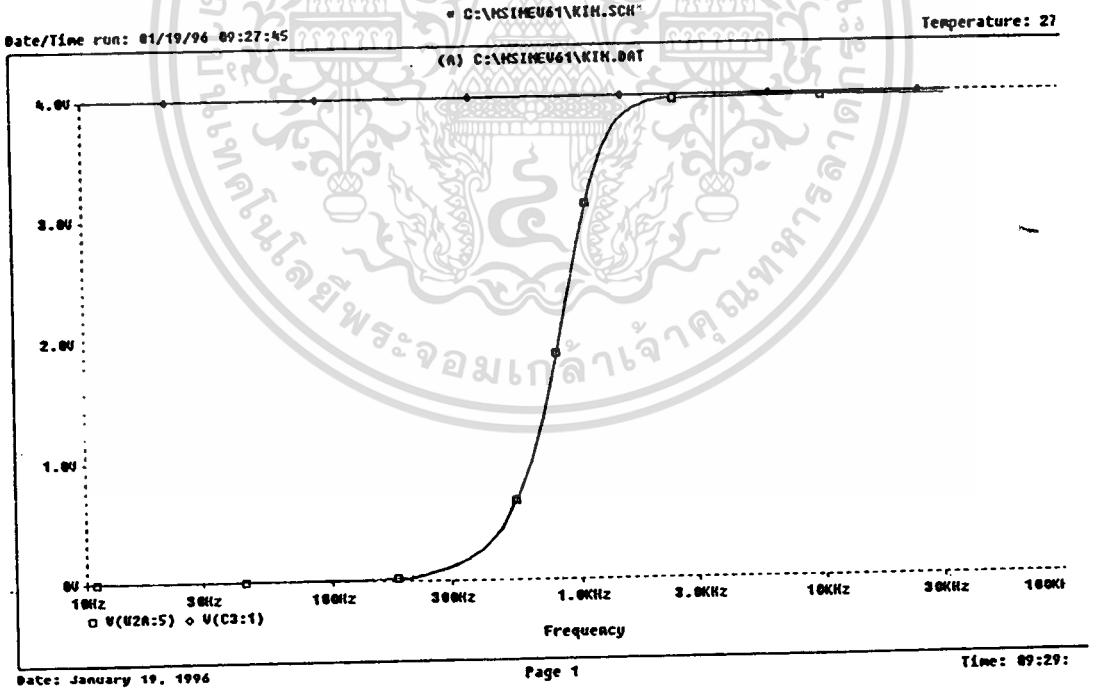




รูปที่ 4.8 วงจร ไฮพาส ฟิลเตอร์ อันดับ สอง และ กราฟ จากการทดลอง PSPICE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูปที่ 4.9 วงจร ไฮพาส ฟิลเตอร์ อันดับ สาม และ กราฟ จากการทดลอง PSPICE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การออกแบบอุปกรณ์เครื่องมือวัดพื้นฐานโดยใช้ IEEE-488 (GPIB)

โครงงานนี้ แบ่งการทำงานเป็น 2 ส่วน คือ

- ส่วนแสดงผลและควบคุม
- ส่วนประยุกต์ใช้งาน

5.1 ส่วนแสดงผลและควบคุม

5.11 ส่วนแสดงผลจะเป็นการติดต่อระหว่าง คอมพิวเตอร์ เจนเนอร์เรเตอร์ และ สโคป โดย นำสัญญาณที่ปรากฏบนจอ สโคป นำมาแสดงผลที่จอ มอนิเตอร์ ของคอมพิวเตอร์

รูปที่ 5.1 แสดงการติดต่อของอุปกรณ์ในส่วนแสดงผลและควบคุม

เมื่อการติดต่อเครื่องมือวัดอันได้แก่ PROGRAMMABLE FUNCTION GENNERATOR HM8130 และ OSCILLOSCOPE HM 1007 เป็นผลสำเร็จในขั้นต่อไปของส่วนแสดงผลก็ คือ อ่านสัญญาณ (ข้อมูล) จาก หน่วยความจำ ของ สโคป มาเก็บไว้ในตัวแปรแบบ CHARโดยในหน่วยความจำของ สโคป จะมี 4K byte โดยแบ่งเป็น แชลแนล ละ 2 Kbyte โดย CH1จะมีค่าระหว่าง 0-2048 byte CH2 จะมีค่าระหว่าง 2049-4096byte ซึ่งในที่นี้จะอ่าน ข้อมูล เข้ามาเก็บไว้ในตัวแปร 4K byte หรือทั้ง 2 แชลแนล โดยข้อมูล ที่อ่านเข้าจะเป็นภาษาเครื่องมีค่าอยู่ระหว่าง 0-256 ค่า เมื่อได้รับข้อมูล แล้วต่อจะเป็นการ PLOT WAVE FROM โดยเข้าไปสู่โหมด กราฟฟิคส์ และใช้ฟังค์ชั่น

PLOT WAVE FROM ( );

5.12 การควบคุมได้สร้างเป็น เมนู ให้เลือก 4 เมนู คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- WRITE ( ) เป็น เมนู ใช้เมื่อต้องการจะเก็บสัญญาณที่ปรากฏบน มอนิเตอร์ ลง ฮาร์ดดิส ของ คอมพิวเตอร์ โดยจะเก็บเป็น ไฟล์ ข้อมูล และ เมนูนี้จะสามารถจะเก็บรูปผลตอบสนองความถี่ ซึ่งจะกล่าวภายหลัง

- READ ( ) เป็นเมนู ใช้ควบคู่กับ เมนู เขียน ( ) คือสัญญาณที่จะเก็บโดย เมนู เขียน ( ) สามารถอ่านกลับมาโดย เมนู อ่าน ( ) โดย เมนู อ่าน ( ) ยังแบ่งออกเป็น 2 ชนิด คือ

1 READ DATA คือ การอ่านสัญญาณที่เก็บเป็น ไฟล์ อยู่ใน ฮาร์ดดิส แบ่งออก 2แบบ

1.1 READ CONTINUOUS คือ อ่านข้อมูลตั้งแต่ต้น ไฟล์ จนจบ ไฟล์

1.2 READ SEARCH คือ สามารถเลือกรูปที่ต้องการอ่านจาก ไฟล์ โดยไม่ต้องอ่านตั้งต้น ไฟล์ จะเสียเวลาในกรณีที่ ไฟล์ มีขนาดใหญ่

2. READ FRE. ( )

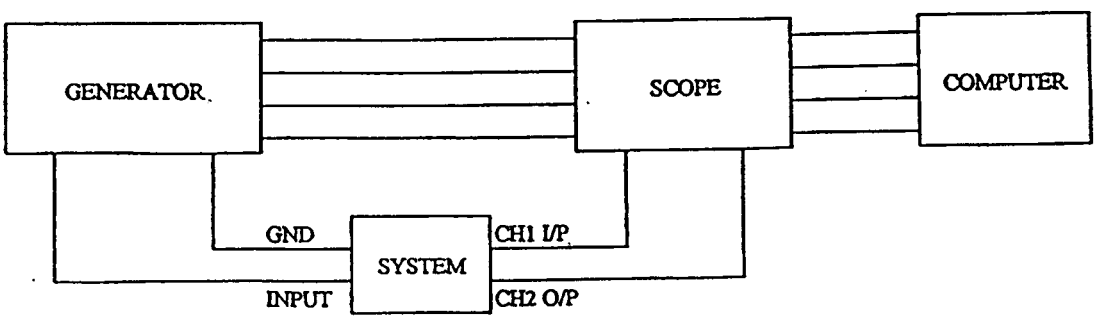
คือการอ่านรูปผลตอบสนองความถี่ ที่ เก็บโดย เมนู WRITE ( )

- SET VOLT ( ) ใช้เมื่อต้องการ เซ็ต ขนาดของสัญญาณที่ปรากฏบนจอ มอนิเตอร์ เมื่อสัญญาณนั้นมีขนาดผิดกว่าความเป็นจริง

- SET GENNERATOR เป็น เมนู ใช้เมื่อต้องการ เซ็ต สัญญาณ ; ขนาด (AMPLITUDE) ความถี่ (FREQUENCY) ของ เจนเนอร์เรเตอร์เพื่อให้ได้ตามต้องการ .

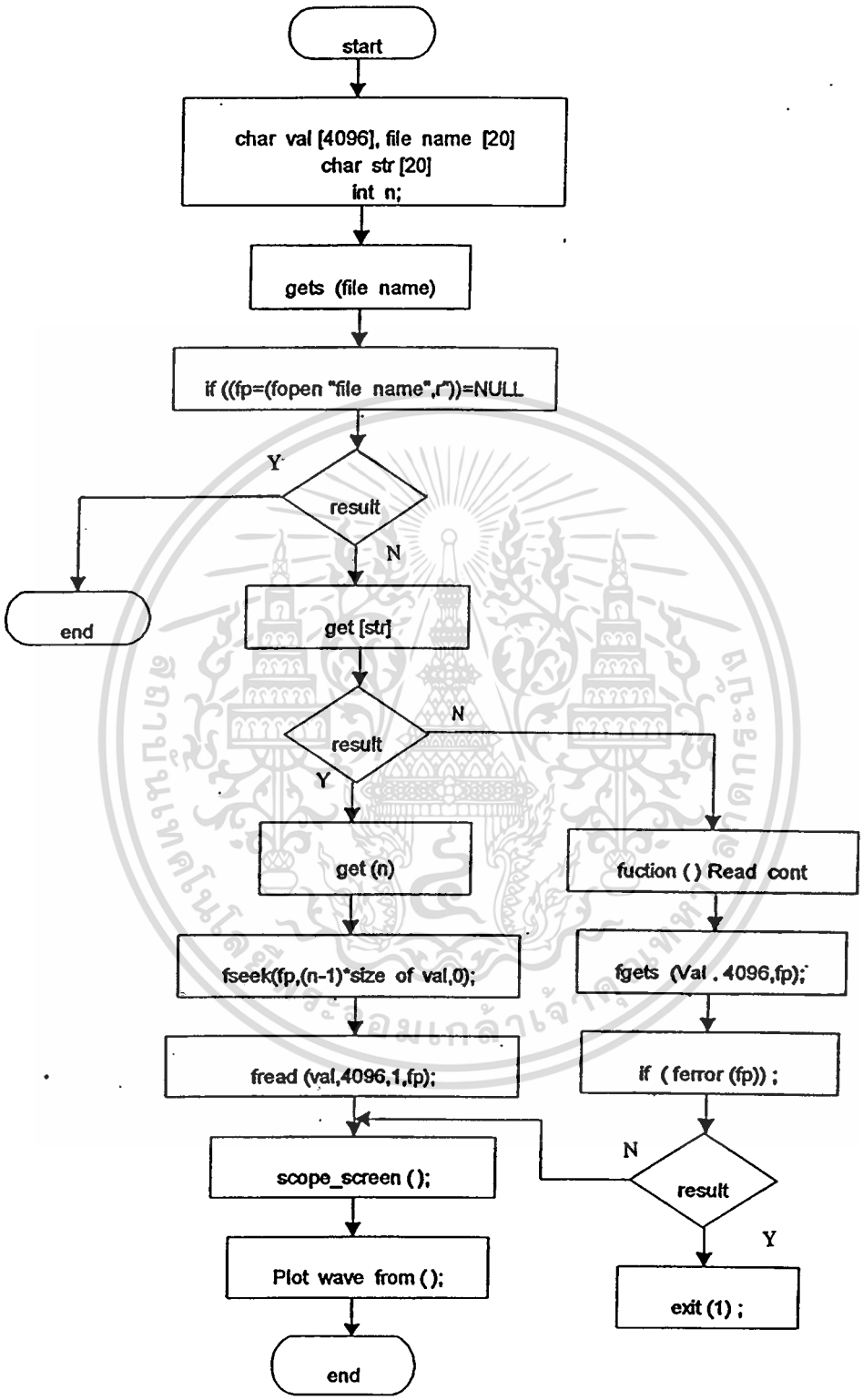
5.2 ส่วนประยุกต์ใช้งาน

โดยได้สร้างเป็น เมนู คือ ความถี่ เรสปอร์น เป็น เมนู ใช้เพื่อต้องการหาผลตอบสนองความถี่ของวงจรโดยสามารถหาค่าความถี่ คัทออฟ อัตราขยายของวงจร (GAIN) ชนิดของวงจร (LOW PASS , HIGH PASS) , ความชัน ของวงจรที่จุด คัทออฟ และนำ ค่าเหล่านี้มา PLOT GRAPH เป็นผลของการตอบสนองความถี่ โดยแกนนอนเป็นแกนของความถี่ เป็น LOG แกนตั้งเป็น อัตราขยายของวงจร (GAIN) ของ



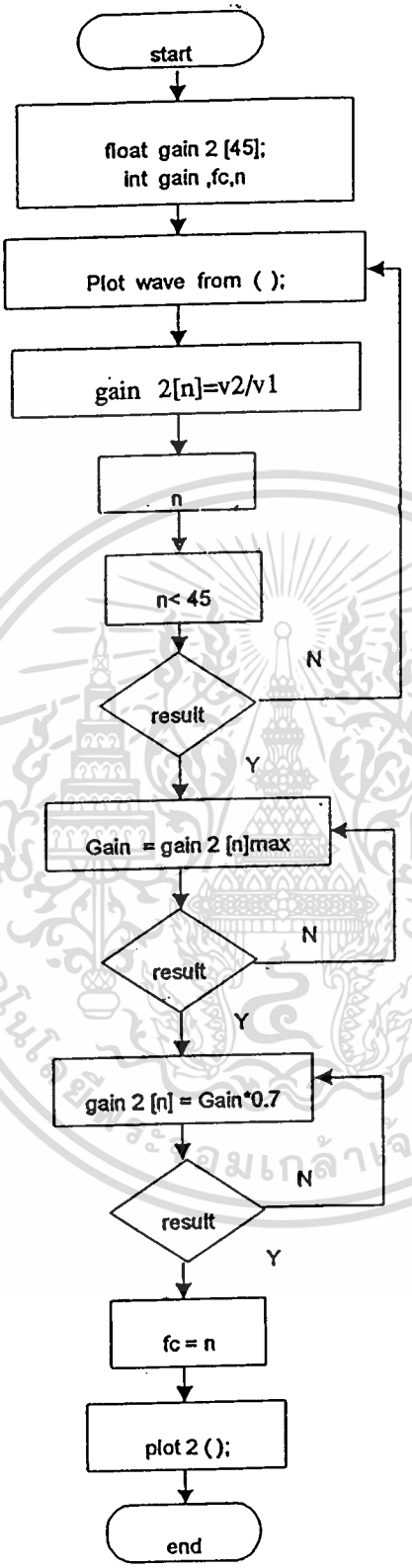
รูปที่ 5.2 แสดงการติดต่อระหว่างอุปกรณ์ในการหา ความถี่ เรสปอร์น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



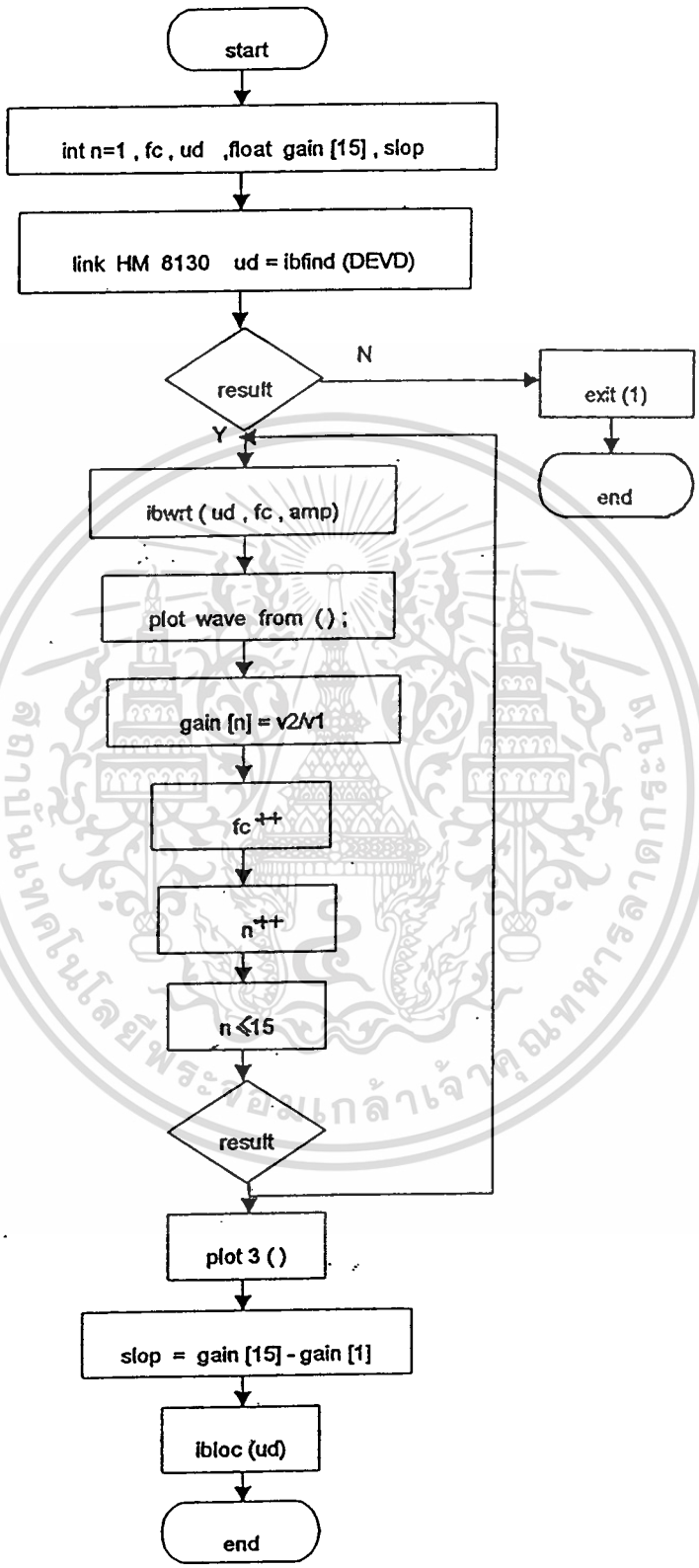
รูปที่ 5.3 แสดง flow chart ของ ฟังค์ชั่นอ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4 แสดง flow chart ฟังค์ชั่น ความถี่ เรสปอร์น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 แสดง flow chart ของ find order ( ) ;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การติดตั้งอุปกรณ์

ในโครงงานนี้ใช้ card-488 ในการควบคุมเครื่องมือวัดต่าง ๆ ได้แก่ programmable function gennerater hm 8130 oscilloscope hm 1007 programmable multimeter hm 8112-2 และ power supply hm 8142

ซึ่งในโครงงานนี้ จะควบคุมเครื่องมือวัด 2 ตัว คือ oscilloscope hm 1007 และ programmable function gennerater hm 8130

6.1 โดยที่ฟังค์ชั่นที่ใช้ในการติดต่อ กับ oscilloscope hm 1007 โดยเขียนเป็นฟังค์ชั่นดังต่อไปนี้

```
                int  HM  1007 (void)
{
        int  ud ;
        char   val [4096] ;
                ud = ibfind ("DEC C") ;
         if (ud<0)
                exit (1) ;
        else
{
        ibwrt (ud) , " DIG 3" , 5) ;
        ibed (ud , val , 4096) ;
        delay [1000]
        }
```

การทำงาน

1. ติดต่อกับ hm 1007 โดยใช้ cmmand code

```
ud = ibfind ("devc") ;
```

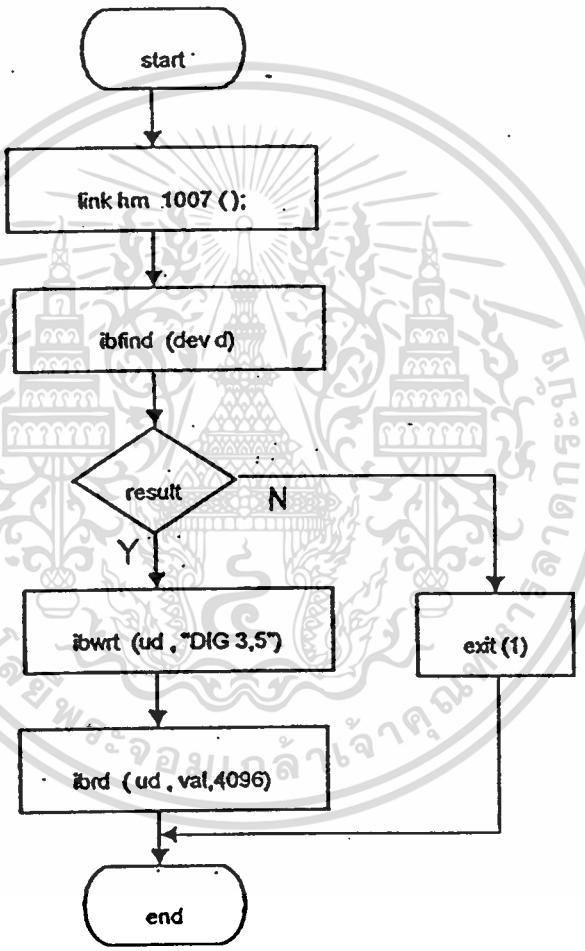
2. ตรวจสอบการติดต่อ ถ้าผลการติดต่อสำเร็จ จะ link กับ HM 1007 โดยใช้ command code

```
ibwrt ( ud , "DIG 3, 5")
ibrd  (ud , val , 4096 )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าผลการติดต่อไม่สำเร็จก็จะจบการทำงาน

3. จบการทำงาน



รูปที่ 6.1 แสดง flow chart การติดต่อ hm 1007

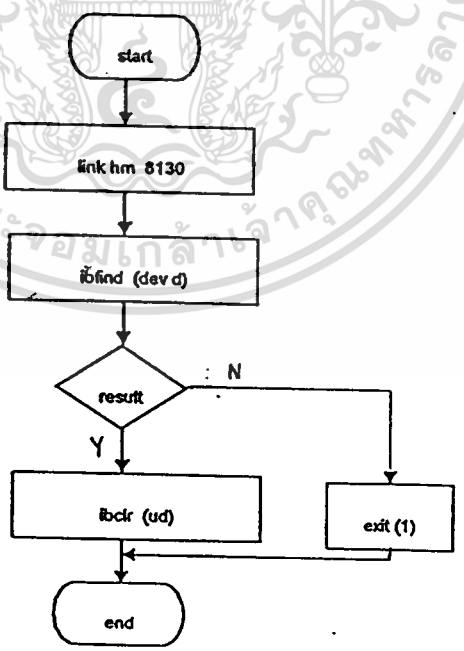
6.2. ฟังค์ชั่น ที่ใช้ในการติดต่อกับ programmable function gennerater hm 8130 โดยเขียนเป็น ฟังค์ชั่นดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int   Hm   8130 (void)
{
        int   ud ;
              ud = ibfind  (" dev d")
              if ( ud <0) ;
              exit (1) ;
              else   {
                    ibclr (ud) ;
              }
}
```

การทำงาน

1. ติดต่อ hm 8130 โดยใช้ command code ud = ibfind ("dev d")
2. ตรวจสอบผลการติดต่อ ถ้าเป็ยผลสำเร็จก็จะ ลิงค์กับ hm 8130 โดยใช้ command code ibc (ud) ถ้าผลการติดต่อไม่สำเร็จก็จะหยุดการทำงาน
3. จบการทำงานของฟังค์ชั่น



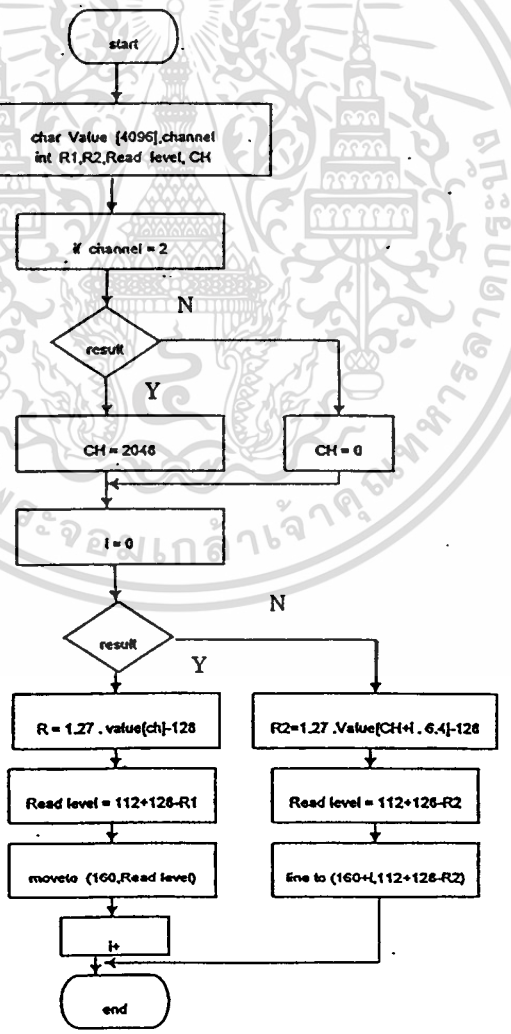
รูปที่ 6.2 แสดง flow chart แสดงการติดต่อกับ hm 8130

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 ฟังค์ชั่น plot wave from (int plot color , int channal , * vale)

การทำงาน

1. ตรวจสอบ แชลแนล = 2 หรือไม่ ถ้า แชลแนล = 2 จะกำหนด ch = 0 ถ้า แชลแนลมีค่าไม่ = 2 จะกำหนด ch = 2048
2. ตรวจสอบ I = 0 หรือไม่จะกำหนดให้ r1 = 1.27 value [ch] - 128 และ readlevel = 112+128-r1 และ move to เป็นการ plot สัญญาณจุดแรก
3. ถ้า I ไม่เท่ากับ 0 จะกำหนด r2 = 1.2 value [ch+i6.4]-128 และ readlevel = 112-r2 ซึ่งเป็นการ sampling ค่าจาก 4096 ค่าเป็น 320 ค่าและ line to เป็นการ พล๊อตสัญญาณ



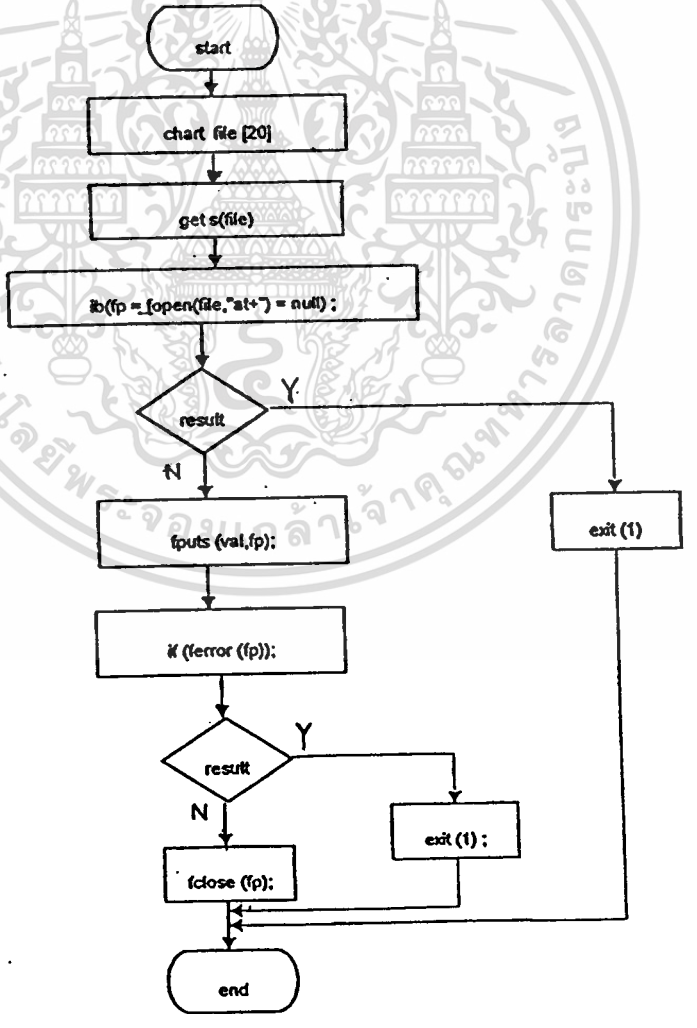
รูปที่ 6.3 แสดง foow chart ของฟังค์ชั่น plot wave from

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 ฟังค์ชั่น write ( )

การทำงาน

- ประกาศตัวแปร char file [20] เพื่อจะเปิดไฟล์โดยการเปิดแบบเอาข้อมูลต่อท้ายไฟล์เดิม
- ตรวจสอบผลการเปิด ไฟล์ ถ้าเป็นผลสำเร็จ จะเขียนข้อมูลในค่าตัวแปร สตริง val [4096] ลงในไฟล์ ซึ่งมี fp เป็นไฟล์ pointerถ้าผลการเปิดไฟล์ไม่เป็นผลสำเร็จก็จะจบการทำงาน
- ตรวจก่อนผลในการเขียนข้อมูลลงไฟล์ถ้ามีข้อผิดผลาดในการเขียนข้อมูลลงในไฟล์จะลบการทำงาน
- ปิดไฟล์จะยกเลิกการติดต่อระหว่างไฟล์และเขียนข้อที่เหลืออยู่ในบัฟเฟอร์ลงในไฟล์
- จบการทำงาน



รูปที่ 6.4 แสดง flow chart ของ ฟังค์ชั่นเขียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5 ฟังค์ชั่น READ

การทำงาน

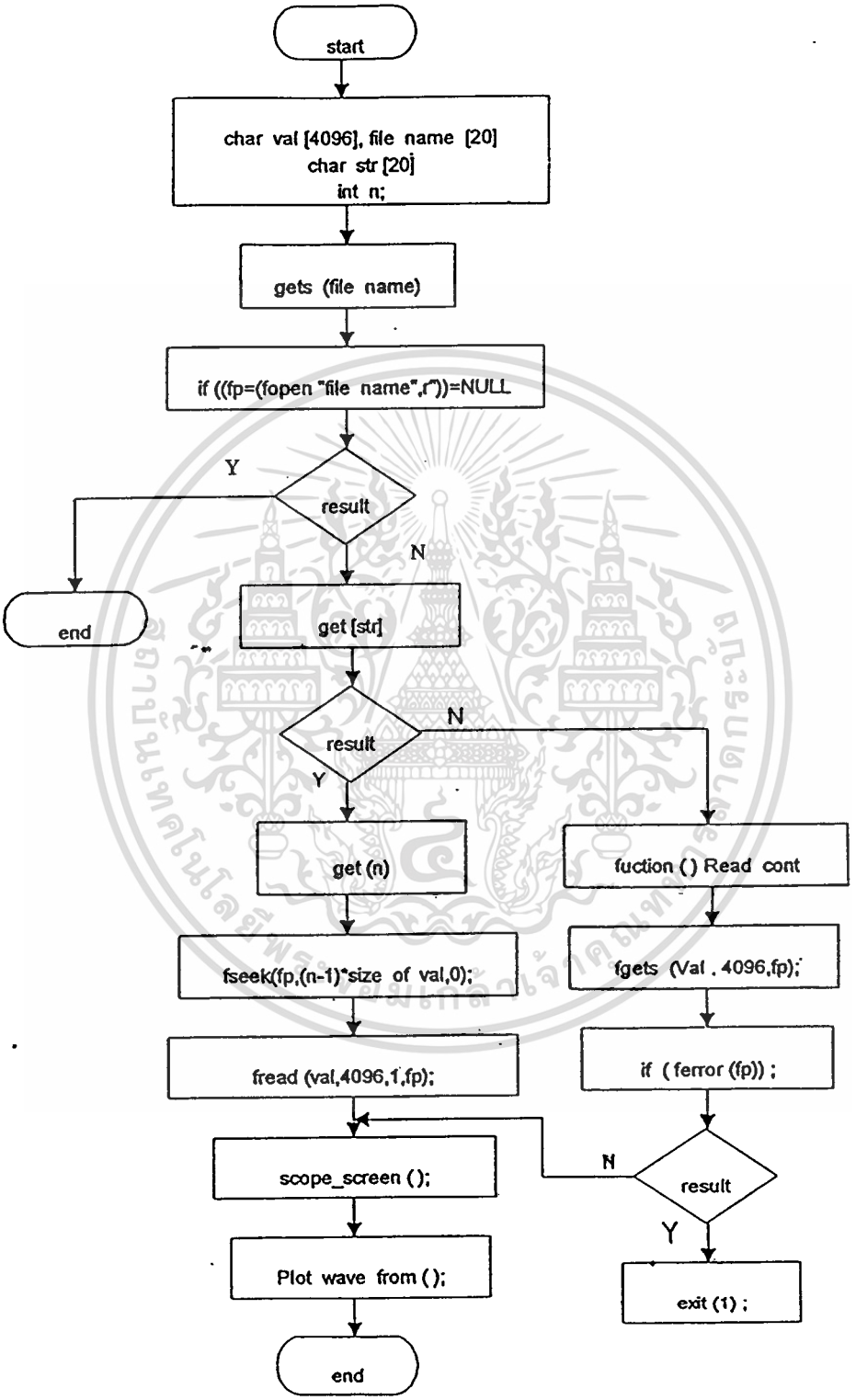
- ประกาศตัวแปร ไฟล์ NAME เพื่อเก็บค่าชื่อ ไฟล์ ที่ต้องการจะอ่าน char (4096) เพื่อเก็บค่าสัญญาณที่อ่านจาก ฮาร์ดดิส ส่วน char str (20) เพื่อเก็บค่าที่เลือกว่าอ่านแบบใด (CONTINUTE หรือ SEARCH) int n ใช้เก็บรูปที่จะอ่าน
- ตรวจสอบการเปิด ไฟล์ ถ้าเป็นผลสำเร็จทำต่อไป ถ้าเป็นเท็จก็จบการทำงาน
- ตรวจสอบผลจาก char sto (20) ว่าเป็นการอ่านแบบใด ถ้าเป็นการอ่านแบบต่อเนื่องก็จะไปที่ ฟังค์ชั่น read cont แต่ถ้าเป็นการอ่านเพื่อค้นหาสัญญาณที่ต้องการ จะทำการเลื่อนตัวแสดงตำแหน่งใน ไฟล์ซึ่งมี FP เป็นไฟล์ pointer โดยใช้คำสั่ง FSEEK ซึ่งจะเลื่อนไป (n-1) SIZE OF VAL
- อ่านค่าข้อมูลจาก ไฟล์ โดยใช้คำสั่ง FREAD อ่านค่าเข้ามาเก็บไว้ที่ตัวแปร VAL จำนวน 4096ไบต์
- เรียก ฟังค์ชั่น ( ) SCOPE-SCREEN ( ) ;
- เรียก ฟังค์ชั่น ( ) PLOT WAVEFROM ( ) ;
- จบการทำงาน

6.6 ฟังค์ชั่น FREQUENCY RESPONE ( ) ;

การทำงาน

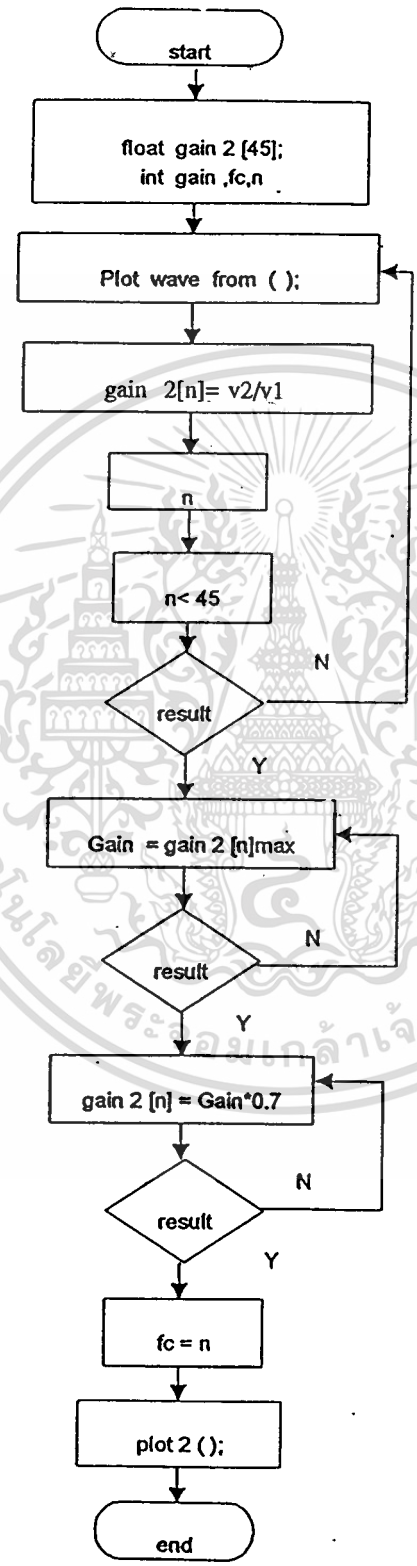
- ก่อนจะเริ่มการทำงาน ฟังค์ชั่น ( ) ; นี้ได้มีการติดต่อกับ hm 8130 (เจนเนอร์เรเตอร์) โดยสั่ง เจนเนอร์เรเตอร์ เพิ่มค่าความถี่เป็น ล๊อค จาก 10 Hz - 10 Khz เพื่อป้อนเข้า อินพุท ของวงจรเพื่อหาคุณสมบัติของวงจรต่าง ๆ
- ประกาศตัวแปร FLOT GAINZ (45) เพื่อเก็บค่า โวลท์ อิน ของวงจรที่ความถี่ต่าง ๆ INT GAIN เพื่อเก็บค่า GAIN MAX ; INT FC เพื่อเก็บค่าความถี่คัทออฟ INT nเพื่อเก็บความถี่
- เรียก ฟังค์ชั่น ( ) ; plot wave from เพื่อหาค่า V1 , V2 ของวงจรที่ต้องการทราบคุณสมบัติของวงจร
- หาค่า เกณฑ์ (n) ที่ n ต่าง ๆ จนกระทั่ง n น้อยกว่าหรือเท่ากับ 45 หรือเก็บค่าเกณฑ์ (n) ที่ค่าความถี่ต่าง ๆ
- หาค่า เกณฑ์ (n) ที่มีค่ามากที่สุด และหาค่า เกณฑ์ที่ทำให้ เกณฑ์ (n) มีค่าเท่ากับ GAIN x 0.7 และกำหนดให้ fc = n
- เรียกฟังค์ชั่น plotz ( ) ; เพื่อนำค่าต่าง ๆ ของวงจร เช่น ค่าความถี่คัทออฟ ค่า เกณฑ์
- จบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.5 แสดง flow chart ของฟังค์ชั่น อ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



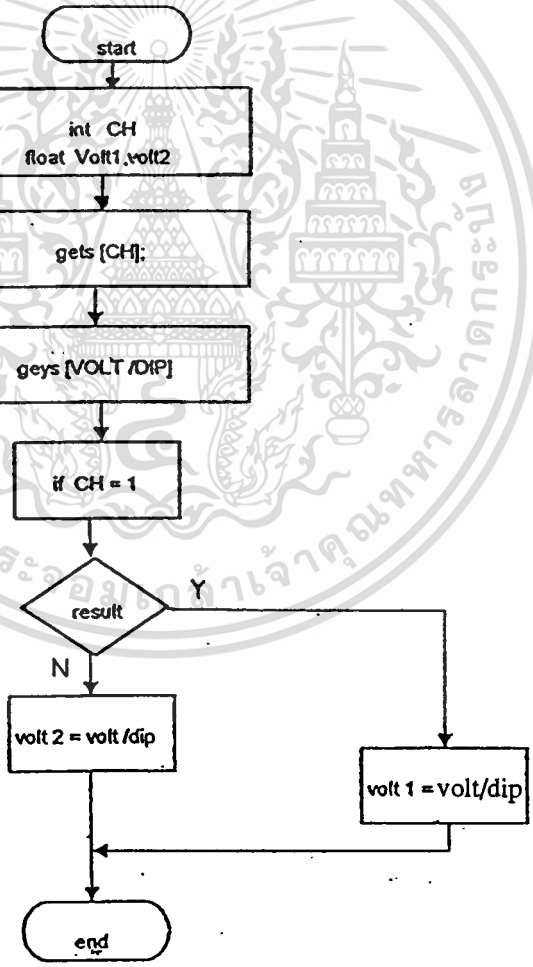
รูปที่ 6.6 แสดง flow chart ฟังค์ชั่นความถี่ เรสปอร์น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.7 ฟังค์ชั่น SET VOLT ( ) ;

การทำงาน

- ประกาศตัวแปร int CH เมื่อเก็บค่า แชลแนลที่ต้องการจะ เซ็ต และ float volt 1 , volt2 เมื่อเก็บค่า voltdip
- รับคำ แชลแนล และค่า VOLTDIP
- ตรวจสอบว่า CH = 1 หรือไม่ถ้า CH=1 จะให้ volt 1 = VOLTDIP และถ้า CH ค่าไม่เท่ากับ 1 ก้จะทำให้ volt 2 = VOLTDIP
- จบการทำงาน



รูปที่ 6.7 แสดง flow chart ของ ฟังค์ชั่น เซ็ต โวลท์

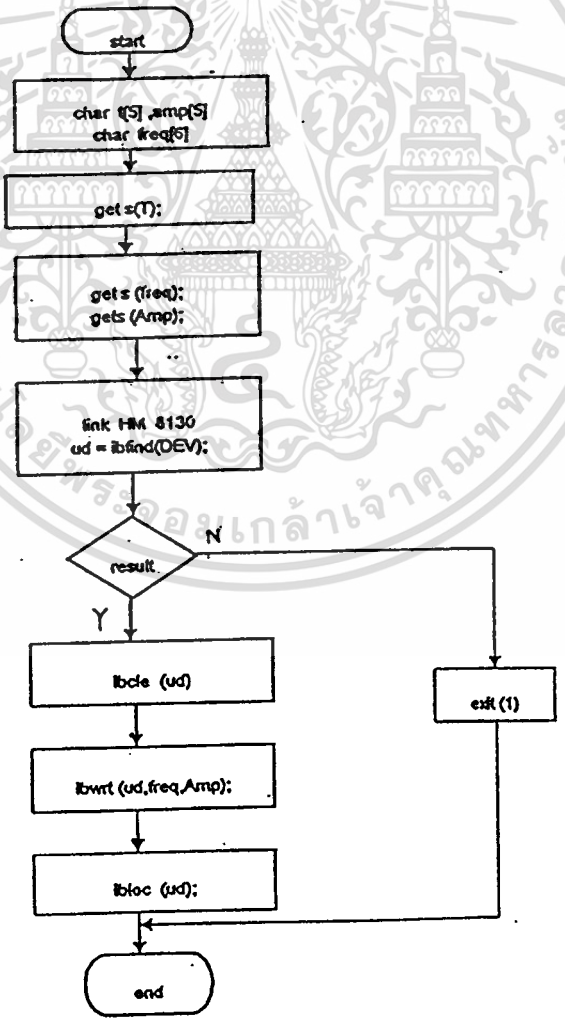
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.8 ฟังค์ชั่น เซ็ต เจนเนอร์เรเตอร์

การทำงาน

- ประกาศตัวแปร CHAR T [5] เมื่อเก็บรูปร่างของสัญญาณ เช่น SIN , RAM , Sqr ,Char Amp [5] เก็บขนาดของสัญญาณfreq [5] เก็บค่าความถี่ของสัญญาณ
- รับค่า ขนาดของสัญญาณ , ความถี่ และรูปร่างของสัญญาณ
- ติดต่อ hm 8130 (เจนเนอร์เรเตอร์) โดยใช้ คอมมานโค๊ต

  Ud = ibfind ("bcv D") ;
- ตรวจสอบผลการติดต่อ ถ้าผลสำเร็จจะทำการ เซ็ต เจนเนอร์เรเตอร์ตามค่าต่าง ๆ ที่กำหนด
- เซ็ต เจนเนอร์เรเตอร์ กลับสู่สภาวะปกติ
- จบการทำงาน



รูปที่ 6.8 แสดง flow chart ของ ฟังค์ชั่น เซ็ต เจนเนอร์เรเตอร์

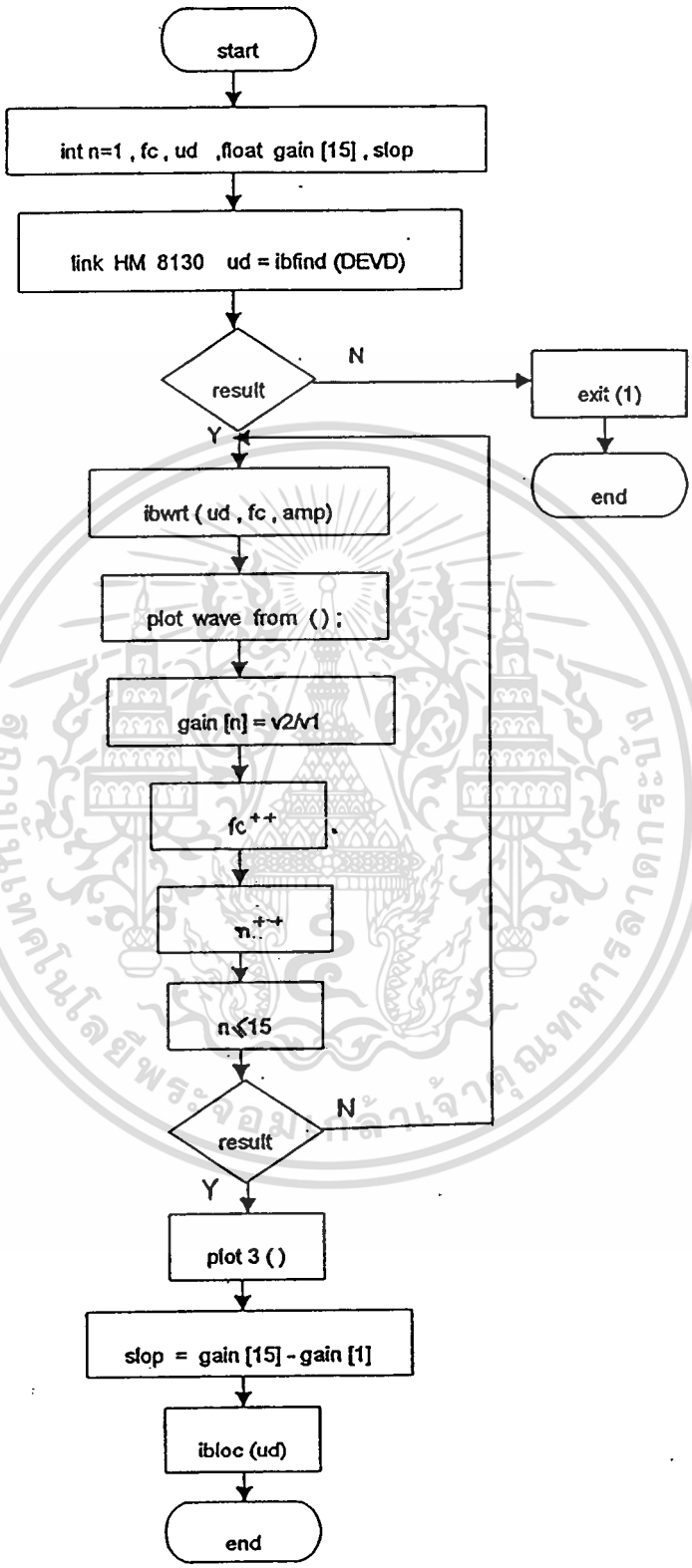
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.9 การทำงาน ของ ฟังค์ชั่น ORDER ( ) ;

การทำงาน

1. ประกาศตัวแปร int ud เพื่อเก็บค่าควบคุมอุปกรณ์ (เจนเนอร์เรเตอร์) int ud เพื่อเก็บค่าความถี่คัทออฟ Flote gain (15) เพื่อเก็บค่าอัตราการขยาย flote slop เพื่อเก็บค่าความชัน
2. ติดต่อกับอุปกรณ์เครื่องมือวัด เจนเนอร์เรเตอร์ ตรวจสอบผลการติดต่อ ถ้าผลการติดต่อสำเร็จก็จะลิงค์กับ hm 8130 ถ้าผลการติดต่อไม่สำเร็จก็จะหยุดการทำงาน
3. ใช้ คอมมานโค๊ต ibwrt (ud , fc , amp) เพื่อสั่งให้เจนเนอร์เรเตอร์ผลิตความถี่ที่ต้องการ
4. เรียก ฟังค์ชั่น plot wave from เพื่อหาค่า V1 , V2 และเก็บค่าผลหาร V1/V2ไว้ที่ตัวแปร gain (n)
5. เพิ่มค่าความคัทออฟ และค่า n
6. ตรวจสอบค่า n มีค่าน้อยกว่าหรือเท่ากับ 15 หรือไม่ ถ้าผลลัพธ์เป็นเท็จก็จะวนกลับไปที่คอมมานโค๊ต ibwrt (ud , fc Amp) เพื่อให้ได้ค่า gain (n) 15 ค่า ถ้าผลลัพธ์เป็นจริงก็จะเรียกฟังค์ชั่น plot 3 ( ) ; เพื่อ plot grap linear ระหว่างอัตราการขยายกับความถี่
7. เก็บค่า เกณฑ์ (15) - ไว้ที่ สโลป สั่งให้ เจนเนอร์เรเตอร์กลับสู่สภาวะปกติ โดยใช้คอมมานโค๊ต คือ ibloc (ud)
8. จบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.9 แสดง flow chart ของ find order ( ) ;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
สรุปผล และวิจารณ์

จากการทดลองใช้งาน Card IEEE-488 ในการควบคุมอุปกรณ์เครื่องมือวัดต่าง ๆ อันได้แก่ Programmable function generator (HM 8130), Osciloscupe (HM 1007) ; Programable multimeter (HM 8112-2), Power Supply (HM 8142) ผ่าน Card NI-488 โดยใช้ Program ในการควบคุม เขียนจากภาษาซี

โดย โปรแกรม ที่ใช้ในการควบคุม แบ่งการทำงานออกเป็น 2 ส่วน คือ

1. ส่วนแสดงผล และควบคุม
2. ส่วนประยุกต์ใช้งาน

7.1 ส่วนแสดงผล และควบคุม เมื่อมีการใช้ โปรแกรม นี้ จะเป็นการเข้าสู่ส่วนแสดงผลและควบคุม ซึ่งส่วนแสดงผลส่วนแรก จะแสดงการติดต่อระหว่างอุปกรณ์เครื่องมือวัด ได้แก่ คอมพิวเตอร์ สโคป เจนเนอร์เรเตอร์ ดังรูปที่ 7.1 แสดงส่วนแสดงผลส่วนแรกในการติดต่อ โดยใช้ เม้า (mouse) ในการเลือกว่า ต้องการติดต่อหรือไม่ ถ้าเลือก ออก(Exit) จะเป็นการออกจาก โปรแกรม นี้ หรือจบการทำงาน แต่ถ้าเลือกลิ้งค์ออฟ(LINK OFF) จะเป็นการควบคุมโดยให้อุปกรณ์ ทั้ง 3 สามารถติดต่อสื่อสารกันได้ดัง รูปที่ 7.2 แสดงการติดต่อของ คอมพิวเตอร์ สโคป เจนเนอร์เรเตอร์ เป็นผลสำเร็จ

เมื่อการติดต่อเป็นผลสำเร็จ จะเข้าสู่ส่วนแสดงผลส่วนที่ สอง ซึ่งส่วนแสดงผลในส่วนนี้ จะเป็นการนำสัญญาณที่ปรากฏบนจอ สโคป มาแสดงผลบนจอ มอนิเตอร์ ซึ่งเป็นการอ่านข้อมูล จาก หน่วยความจำ ของ สโคป 4K byte ดังรูปที่ 7.3 และที่บนด้านซ้ายของจอ จะแสดง เมนนู โดยใช้ เม้า ในการควบคุม เมื่อมีการกด เม้า ด้านซ้าย บริเวณนี้จะปรากฏว่ามี เมนนู ย่อย อีก 7 เมนนู ซึ่งใน เมนนู ย่อยนี้ แบ่งออกเป็น 2 ส่วน คือ ส่วนควบคุม และประยุกต์ใช้งาน ดังรูปที่ 7.4 แสดง เมนนู

โดยในที่นี้ จะเริ่มการใช้งานในส่วนควบคุม คือ เมนนู เขียน โดยการเลื่อน เม้ามายังบริเวณขอบเขตของ เมนนูเขียน แล้วคิก เม้า ด้านซ้าย จะเป็นการใช้งาน เมนนูนี้ ดังแสดงดังรูปที่ 7.5 แสดงการใช้งาน เมนนูเขียน ซึ่ง เมนนู นี้ใช้ในการจัดเก็บรูปสัญญาณที่ปรากฏบนจอ มอนิเตอร์ ลงใน ฮาร์ดดิส ของ คอมพิวเตอร์ โดยจะเก็บเป็น ไฟล์ ข้อมูล ดังแสดง ดังรูปที่ 7.6 แสดงการจัดเก็บรูปสัญญาณเป็น ไฟล์ โดยใส่ชื่อของ ไฟล์ ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนควบคุมในส่วนที่ 2 คือ เมนนู อ่าน ในการใช้งาน เมนนู นี้ โดยเลื่อน เม้ามาคิก บริเวณขอบเขตของ เมนนู อ่าน ดังรูปที่ 7.7 แสดงการใช้งาน เมนนูอ่านซึ่ง เมนนู นี้ จะใช้งานร่วมกับ เมนนู เขียน คือ สัญญาณที่จัดเก็บโดย เมนนู เขียน สามารถอ่านกลับมา โดย เมนนู อ่าน ซึ่งจะต้องบอกชื่อ ไฟล์ ข้อมูลที่ต้องการจะแสดงรูปของสัญญาณ เมนนู อ่าน ยังแบ่งวิธีการอ่านข้อมูลเป็น 2 แบบ คือ ข้อมูล และ ความถี่เรสปอร์น ดังแสดงดังรูปที่ 7.8 แสดงชนิดของข้อมูลที่ต้องการจะอ่านโดยที่การอ่านข้อมูลแบบ ข้อมูล เป็นการอ่านข้อมูลที่เป็นรูปสัญญาณที่ปรากฏบนจอ สโคป ส่วนการอ่านแบบ ความถี่เรสปอร์น เป็นการอ่านข้อมูลที่เป็นรูปผลตอบสนองของความถี่ โดยการจัดเก็บของ เมนนู ความถี่เรสปอร์นซึ่งจะกล่าวภายหลัง โดยการใช้ เม้า ในการเลือก เมื่อเลือกการอ่านข้อมูลแบบ ข้อมูล ยังแบ่งย่อยออกเป็นอีก 2 ประเภท คือ คอล (CONT) และ เซอร์ส (SEARE) ดังแสดง ดังรูปที่ 7.9 แสดงประเภทของการอ่านข้อมูลแบบ ข้อมูล ในการเลือกประเภทแบบการอ่าน โดยใช้ เม้า ในการเลือกกดปุ่มบนด้านขวาของ เม้า

การอ่าน ข้อมูล แบบคอล เป็นการอ่านข้อมูล หรือรูปสัญญาณแบบต่อเนื่องตั้งแต่ต้น ไฟล์ ข้อมูล จนจบ ไฟล์ ข้อมูล ส่วนการอ่าน ข้อมูล แบบ เซอร์ส เป็นการอ่านข้อมูลหรือรูปสัญญาณแบบเจาะจง ว่าต้องการอ่าน ไฟล์ ข้อมูล รูปที่เท่าไร ใน ไฟล์ ข้อมูลโดยใส่เป็นลำดับ ของรูปสัญญาณ

ส่วนควบคุมส่วนที่ 3 คือ เมนนู เซ็ต โวลท์ในการใช้งาน เมนนู โดยการเลื่อน เม้ามาคิกบริเวณขอบเขตของเมนนู เซ็ต โวลท์ ดังแสดง ดังรูปที่ 7.10 แสดงการใช้งาน เมนนู เซ็ต โวลท์ ในจอแสดงรูปสัญญาณบนจอ มอนิเตอร์ มุมด้านบนซ้ายจะแสดงค่าขนาดของสัญญาณ โวลท์เดจ ทั้ง 2 แชลแนน และล่างด้านขวา จะแสดงจำนวน ไฟล์ ข้อมูลที่บันทึก เขียน และ จำนวนไฟล์ ข้อมูลที่อ่าน ซึ่ง เมนนู เซ็ต โวลท์ จะใช้ในการ เซ็ต โวลท์เดจ ให้ถูกต้องตามค่าที่อ่านเข้ามา โดยการใช้งานจะเริ่มที่ แชลแนล ที่ต้องการจะ เซ็ต (แชลแนล 1 หรือ แชลแนล 2) และขนาดของ Vol / Dip ที่ถูกต้อง

ส่วนควบคุมส่วนที่ 4 คือ เมนนู เซ็ต เจนเนอร์เรเตอร์ ในการใช้งาน เมนนู นี้โดยการเลื่อนเม้ามาคิกบริเวณขอบเขตของ เมนนูเซ็ต เจนเนอร์เรเตอร์ ดังแสดง ดังรูปที่ 7.11 แสดงการใช้งาน เมนนู เซ็ต เจนเนอเรเตอร์ ซึ่งใช้ในการควบคุม เจนเนอร์เรเตอร์ โดยจะเริ่ม SET รูปสัญญาณที่ต้องการ (RAM,TRI, SIN,SQR,PLS) ดังแสดง ดังรูปที่ 7.12 แสดงการ เซ็ต รูปสัญญาณที่ต้องการเมื่อเลือกรูปสัญญาณที่ต้องการโดยการใช้ เม้า แล้ว ต่อไปจะเป็นการ เซ็ต แอมปริจูดของสัญญาณ และความถี่ ของสัญญาณตามความต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนควบคุมส่วนที่ 5 คือ เมนนู ยกเลิก (CANCEL) การใช้งาน เมนนู นี้ โดยใช้ เม้าในการควบคุม ซึ่ง เมนนู ยกเลิก จะใช้ในการยกเลิกการใช้ เมนนู ย่อยทั้งหมด หรือกลับสู่ส่วนแสดงผลส่วนที่ สอง

ส่วนควบคุมส่วนที่ 6 คือ เมนนู ออก (EXIT) ซึ่ง เมนนู นี้จะใช้ในการยกเลิกการทำงานทั้งหมดของ โปรแกรม หรือ จบการทำงานของ โปรแกรม

7.2. ส่วนประยุกต์ใช้งาน ในส่วนประยุกต์ใช้งาน คือ เมนนู ความถี่ เรสปอร์น การใช้งาน เมนนู นี้โดยเลื่อน เม้า มาคิกบริเวณขอบเขตของ เม้า ดังแสดงดังรูปที่ 7.13 แสดงการใช้งาน เมนนู ความถี่ เรสปอร์น ซึ่ง เมนนู นี้ใช้ในการหาผลตอบสนองของความถี่ เรสปอร์นของวงจร หรือ ระบบที่ต้องการหา ซึ่งสามารถหาค่า ความถี่คัทออฟ อัตราขยายของวงจร ชนิดของวงจร โรพาส , ไฮพาส ความชัน ของวงจร

ในขั้นต้นของการใช้งาน เมนนู นี้ จะต้องต่อระบบ หรือวงจร ดังแสดง ดังรูปที่ 2 ซึ่งจะแสดงการติดต่อระหว่างอุปกรณ์ในการหา ความถี่ เรสปอร์น ในบทการออกแบบอุปกรณ์และเครื่องมือวัดพื้นฐานโดยใช้ IEE-488 ซึ่ง แชลแนล 1 จะต้องเป็น อินพุท และ แชลแนล 2 เป็น เอ้าท์พุทถ้าต่อสลับกันผลที่ได้ทั้งหมดจะผิดพลาด

การใช้งาน เมนนู นี้ จะเริ่มที่ เมนนู ย่อย โดยใน เมนนู ย่อย แบ่งออกเป็น 2 ส่วน คือแอคทีฟ และ พาสซีพ ดังแสดง ดังรูปที่ 7.14 รูปแสดง เมนนู ย่อยของ เมนนูความถี่ เรสปอร์นซึ่งเป็นการเลือกว่า ต้องการทดสอบของวงจร หรือระบบชนิดใด ระหว่างอุปกรณ์ประเภท แอคทีฟหรือ พาสซีฟ เพราะถ้าเป็นอุปกรณ์ประเภท แอคทีฟ ในการ พล๊อต รูปกราฟผลตอบสนองของความถี่จะเกิดการล้น หรือ พ้น ขอบจอ มอนิเตอร์ ดังนั้น การเลือกในที่นี้ ก็คือการจัดสเกล (scal)นั่นเอง เมื่อเลือกประเภทของวงจร หรือระบบแล้วในขั้นตอนต่อไป จะต้องปรับ TIME/DIV ของความชันไว้ที่ค่ามากที่สุด เพราะค่าที่เริ่ม พล๊อต ค่าแรกคือ 10 Hz ซึ่งเป็นค่าความถี่ต่ำ โดยใช้แกนนอนเป็นแกนของความถี่ โดยเพิ่มค่าของความถี่เป็น ล๊อค โดยเริ่มตั้งแต่ค่า 10Hz-1MHzจะเห็นว่าจะต้องปรับค่า TIME/DIV ของ สโคปเพื่อให้ค่า โวลท์เตจ ที่อ่านจาก สโคป มีความสมบูรณ์ หรือถูกต้องตลอดเวลา และแกนตั้ง คือ อัตราขยายของวงจรเป็นแบบ ลีเนียส์(linear) ซึ่งในที่นี้จะเห็นได้ว่าค่าที่อ่านจาก สโคป ถ้ามีการผิดเพี้ยน ค่าที่ได้จากการ พล๊อต กราฟ เซมิล๊อค ก็จะผิดพลาดด้วย เมื่อมีการอ่านค่าจาก สโคป เป็นที่เรียบร้อยแล้ว ต่อไปก็จะนำค่าเหล่านี้มา พล๊อต กราฟ ดังแสดง ดังรูปที่ 7.15 แสดงรูปผลตอบสนองของความถี่ของวงจร หรือระบบ ซึ่งในรูปที่7.15 เป็นผลตอบสนองของระบบเชิงตัวเลข ดิจิตอล ซึ่งมี ความชัน อนันต์ แต่ถ้าใช้ในการทดสอบผลตอบสนองความถี่ของวงจร ลิเนียส์ หรือ ฟิลเตอร์ จะเกิดมี ความชัน ให้เห็น และบนด้านขวาของรูปที่ 7.15 จะแสดงข้อความ PLEASE ENTER FOR REPORT SYSTEM เพื่อจะแสดงค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่าง ๆ ของระบบดังรูปที่ 7.16 แสดงค่าผลตอบสนองความถี่ อันได้แก่ ชนิดของวงจร โรพาส , ไฮพาส อัตราขยายของวงจร ความถี่คัทออฟ ค่า โวลย์เตจ อินพุท และ โวลท์เตจที่จุดคัทออฟ ออฟของวงจร

จากรูปที่ 7.16 จะปรากฏหน้าต่างในการเลือกโดยใช้ เม้า คือ YOU WANT SAVE? คือ ต้องการจะจัดเก็บรูปผลตอบสนองของความถี่นี้หรือไม่ โดยจะจัดเก็บเป็น ไฟล์ ข้อมูล เช่นเดียวกับ เมนนู เขียน เมื่อทำการเลือกว่าจะจัดเก็บรูปผลตอบสนองของความถี่แล้ว ก็จะปรากฏหน้าต่างอีกหน้าต่างหนึ่ง คือ YOU WANT TO FIND SLOP ? ดังแสดงดังรูปที่ 7.17 รูปแสดงผลการเลือกว่าต้องการหา ความชัน ของวงจรหรือระบบหรือไม่ เพราะผลในการหา ความชัน จากรูปผลตอบสนองของความถี่ แบบ เซมิ-ล็อค ค่า ความชัน หรือความชัน มีค่าคลาดเคลื่อนสูง โดยเฉพาะในวงจร หรือ ฟิลเตอร์ ที่มีค่า อันดับ สูงกว่า 2 อันดับ

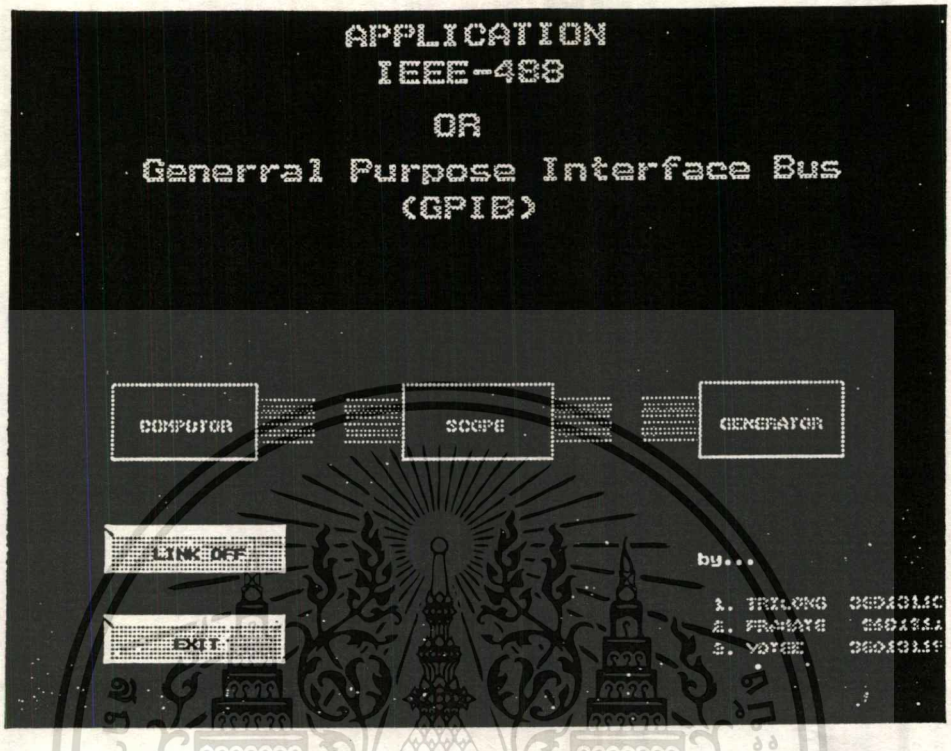
โดยในการหาค่า ความชัน นี้จะเริ่ม พล็อต ค่าตั้งแต่ค่าความถี่คัทออฟ ไปอีก10 ค่าของความถี่ โดยเพิ่มค่าความถี่ แบบลิเนียส์ เพื่อความละเอียดในการหาค่าของ ความชัน ดังแสดงดังรูปที่ 7.18 รูปแสดง ความชัน ของระบบ หรือวงจร และบนด้านขวาของรูปที่ 7.18 จะแสดงข้อความ PLEASE ENTER FOR REPORT SYSTEM ซึ่งจะเป็นการนำค่าที่ได้จากการพล็อต กราฟ แบบ ลิเนียส์ ไปหาค่าคุณสมบัติต่าง ๆ ของวงจร หรือระบบร่วมทั้งความชันของระบบด้วยโดยจะบอกค่าในหน่วยของ dB/OCT หรือ ค่าลดทอนในหน่วยของ dB ต่อ 2 เท่าของความถี่คัทออฟ

จากการทดลองใช้ card NI-488 ในการควบคุมอุปกรณ์เครื่องมือวัดโดยใช้มาตรฐาน IEEE-488 มีจุดเด่น คือ ความไวในการติดต่อสื่อสาร ระหว่างอุปกรณ์เครื่องมือวัด คือสามารถส่งข้อมูลได้ถึง 1 ล้าน byte ต่อวินาที

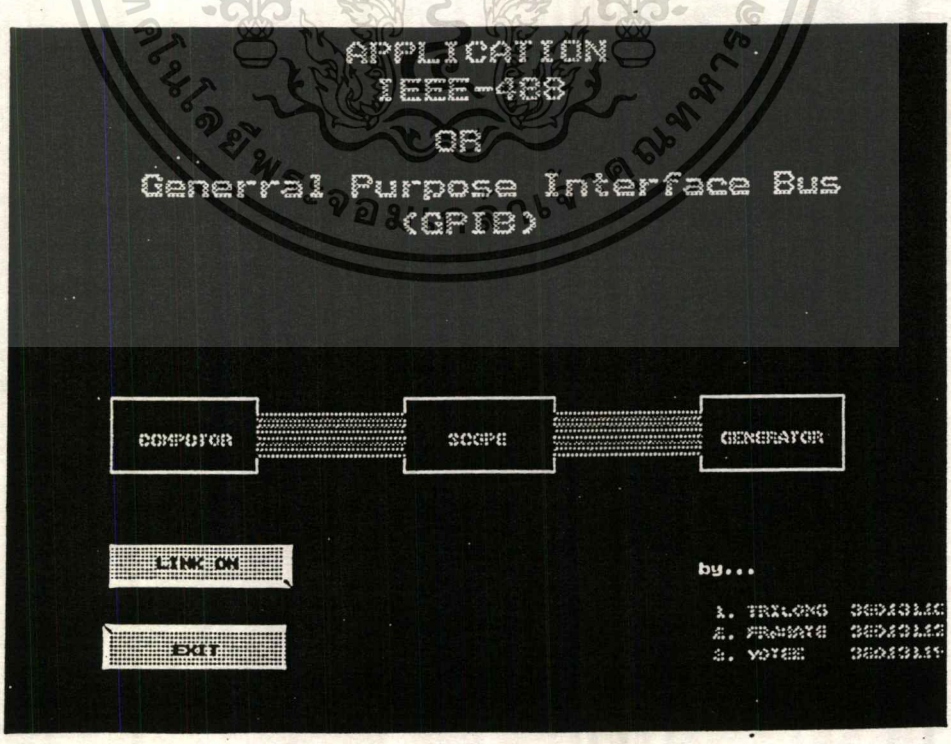
ปัญหาที่พบในการทดลอง คือ ขีดความสามารถของเครื่องมือวัด คอมพิวเตอร์ และตัวผู้ทดลอง ขีดความสามารถของอุปกรณ์ เครื่องมือวัด อันได้แก่ สโคป ไม่สามารถควบคุมฟังค์ชั่นได้หมด ทุก ฟังชั่นด้วย คอมพิวเตอร์ เช่น การปรับค่า VOLT/DIV และ TIME/DIV และ คอมพิวเตอร์ ซึ่งมีการทำงานที่ล่าช้าในการอ่านข้อมูลครั้งละมาก ๆ รวมทั้งผู้ทดลองซึ่งไม่เคยมีความรู้ด้าน คอมพิวเตอร์ และประสบการณ์ด้านการเขียน โปรแกรม ด้วยภาษามาก่อน จึงทำให้โปรแกรม ที่ใช้มีความซับซ้อน เช่นในการใช้ เม้า ควบคุมการใช้งานเมนนู จะไม่สามารถตอบสนองได้ทันทีทันใด เนื่องจากมีการสร้าง ลูป (loop) มากเกินความจำเป็น

แนวทางในการพัฒนา โดยเพิ่มขีดความสามารถของโปรแกรม โดยการเพิ่ม เมนนูในการควบคุม และ เมนนู ในการประยุกต์ใช้งาน อุปกรณ์เครื่องมือวัด และปรับปรุงในส่วนของโปรแกรมเพื่อที่จะให้มีการทำงานที่รวดเร็วมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

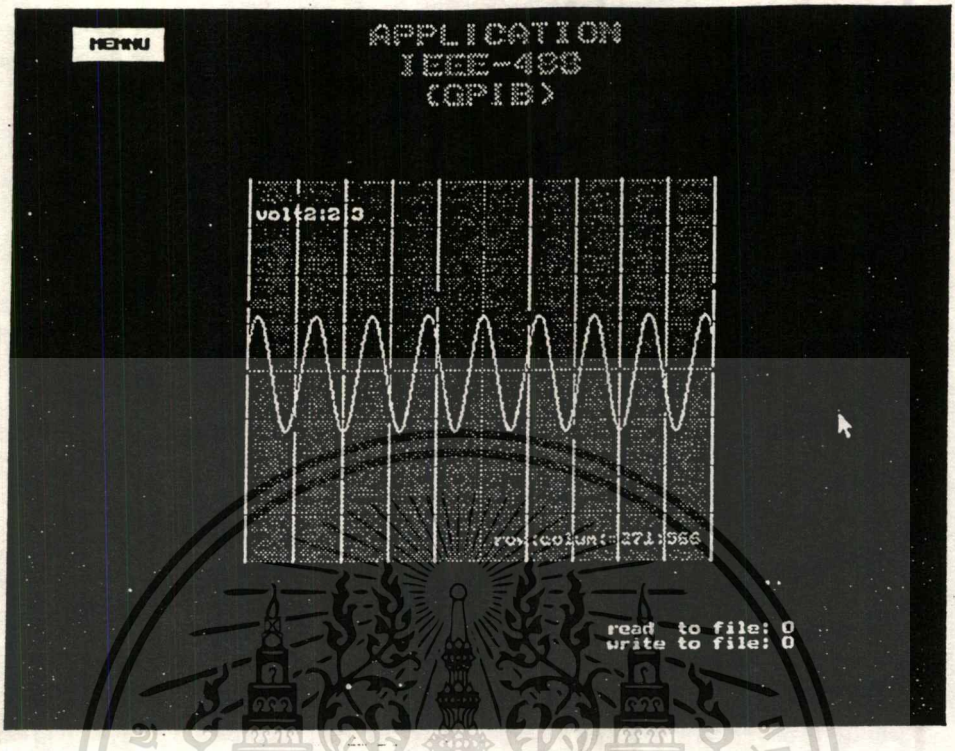


รูปที่ 7.1 แสดงส่วนแสดงผลส่วนแรก ในการติดต่อ

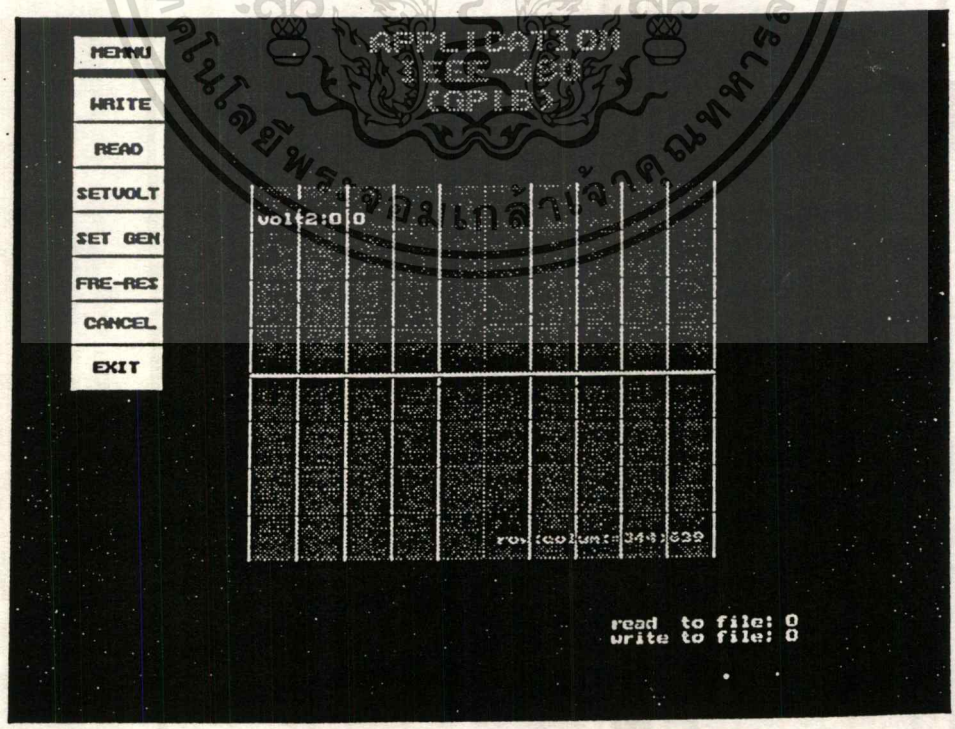


รูปที่ 7.2 แสดงการติดต่ออุปกรณ์เป็นผลสำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

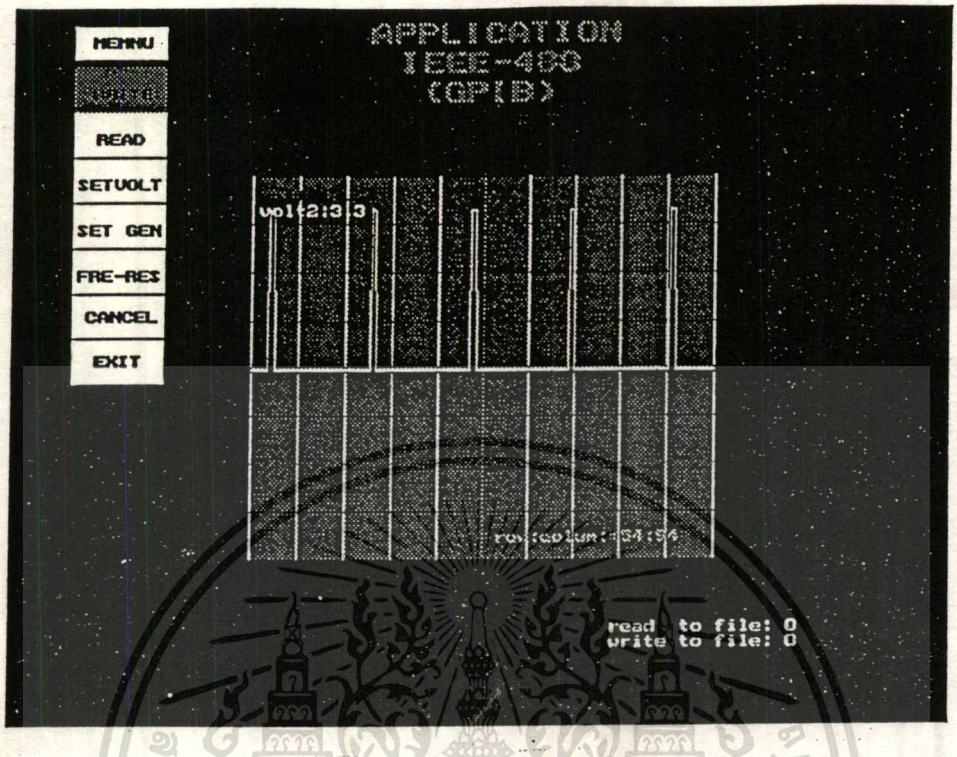


รูปที่ 7.3 ส่วนแสดงผลที่ 2 จะเป็นการเอารูปสัญญาณที่ปรากฏบนจอสโคปมาแสดงบนมอนิเตอร์

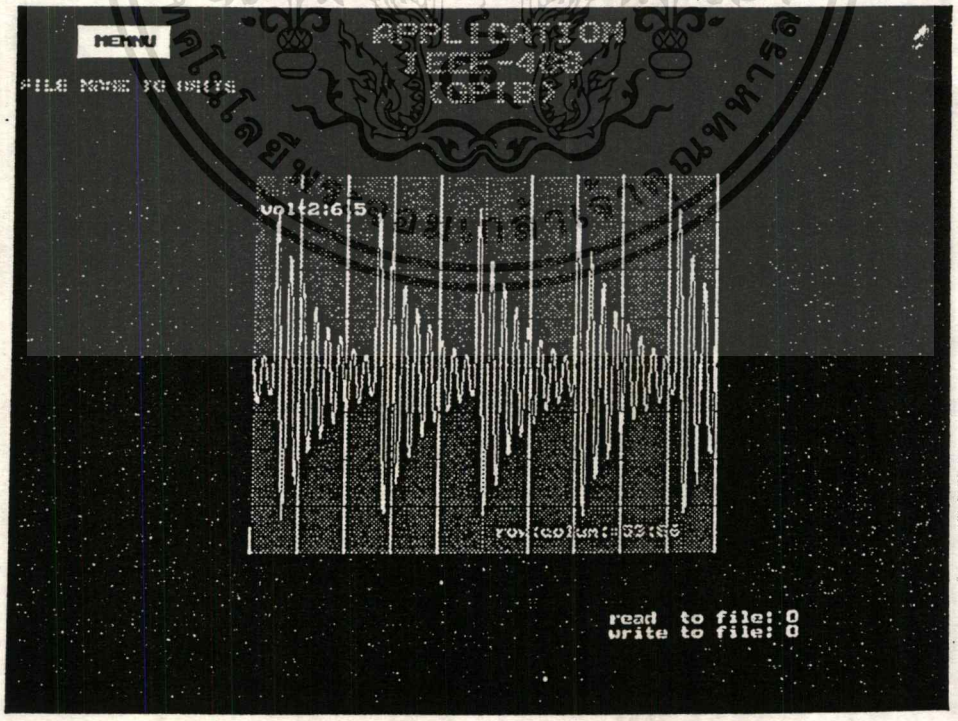


รูปที่ 7.4 ส่วนแสดงส่วนแสดงเมนนูในการควบคุมและประยุกต์ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

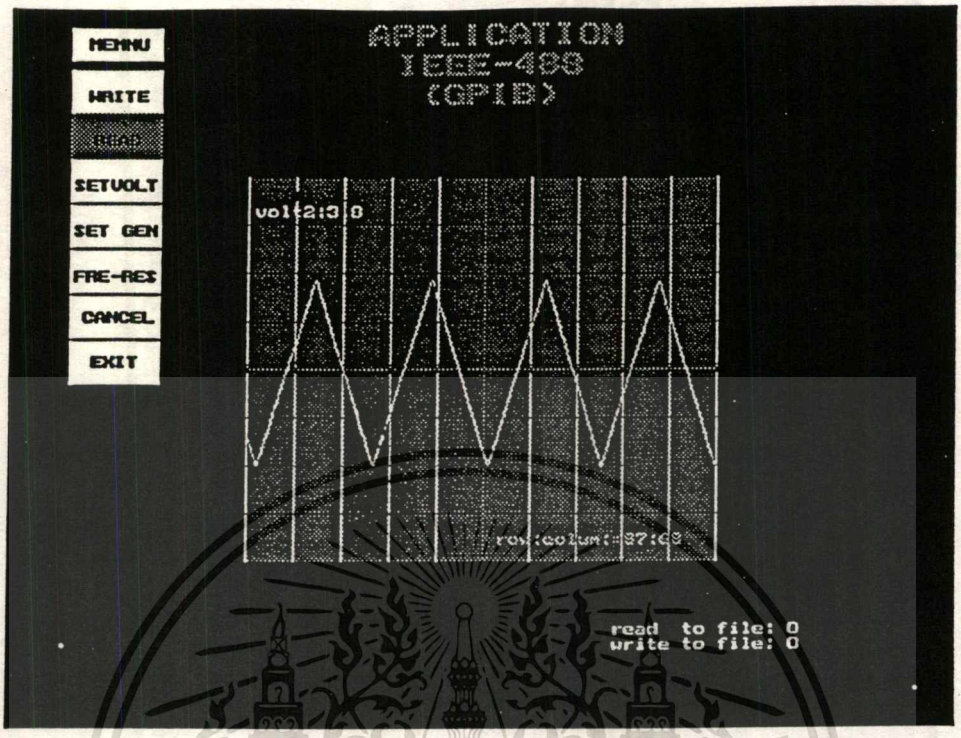


รูปที่ 7.5 แสดงการใช้งาน เมนนู เขียน

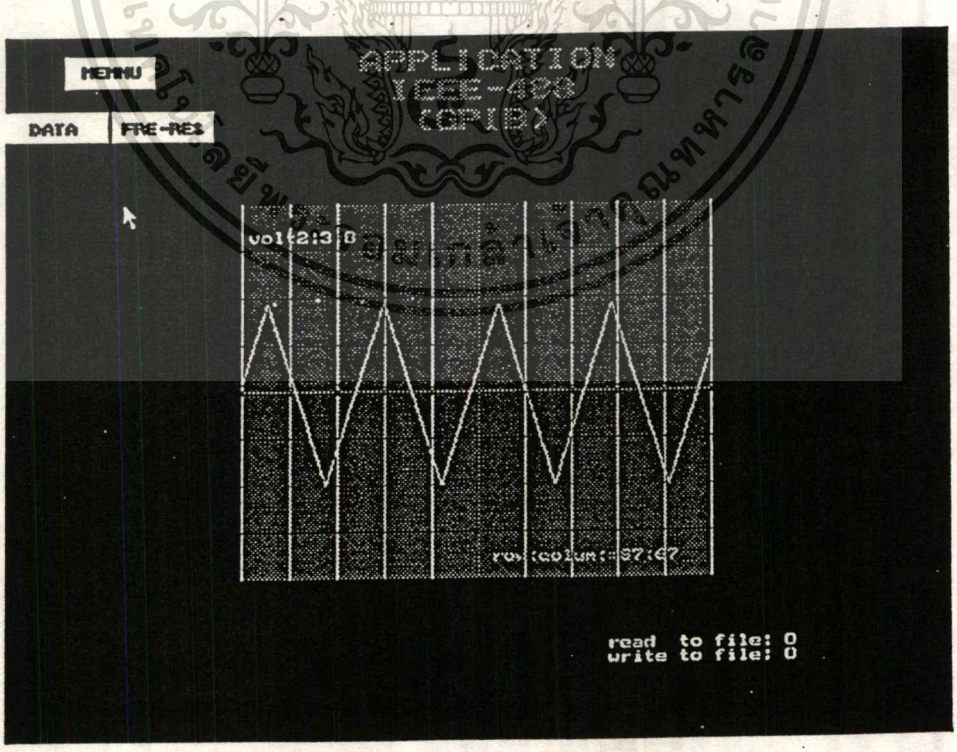


รูปที่ 7.6 แสดงการจัดเก็บรูปสัญญาณเป็น ไฟร์ ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

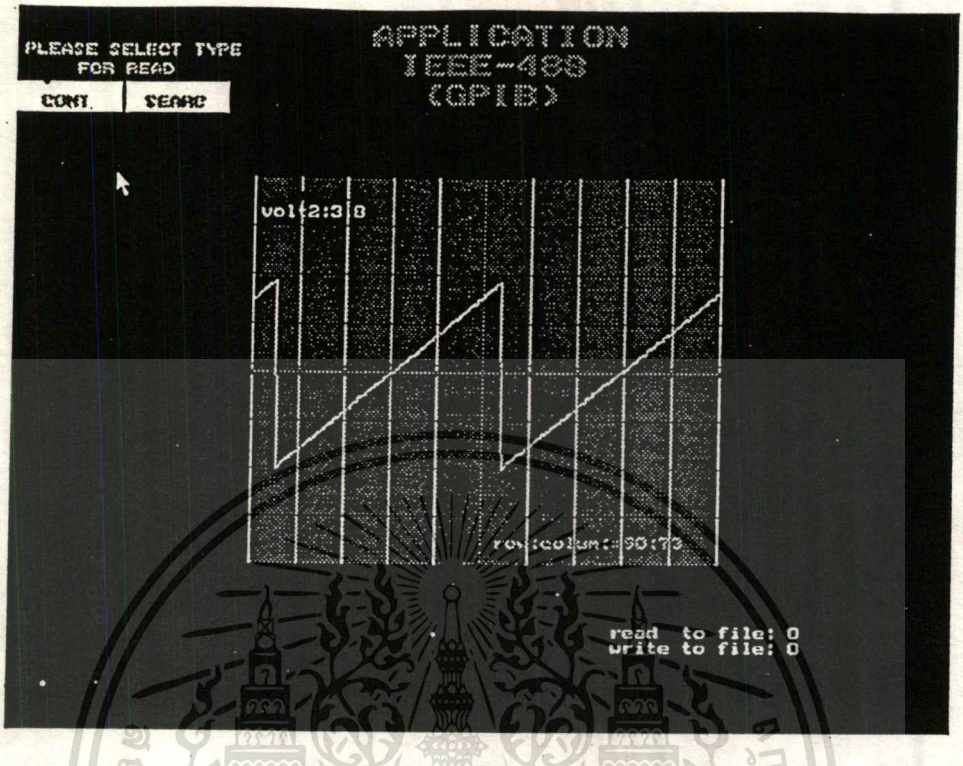


รูปที่ 7.7 แสดง การใช้งาน เมนู อ่าน

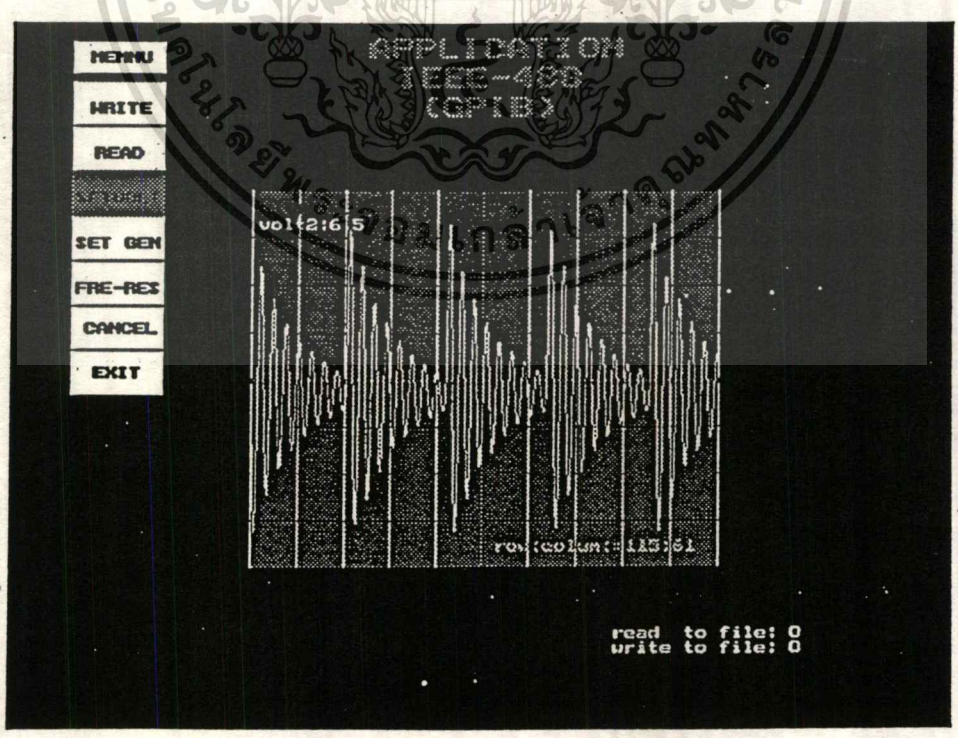


รูปที่ 7.8 แสดง ชนิดของข้อมูลที่ต้องการอ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

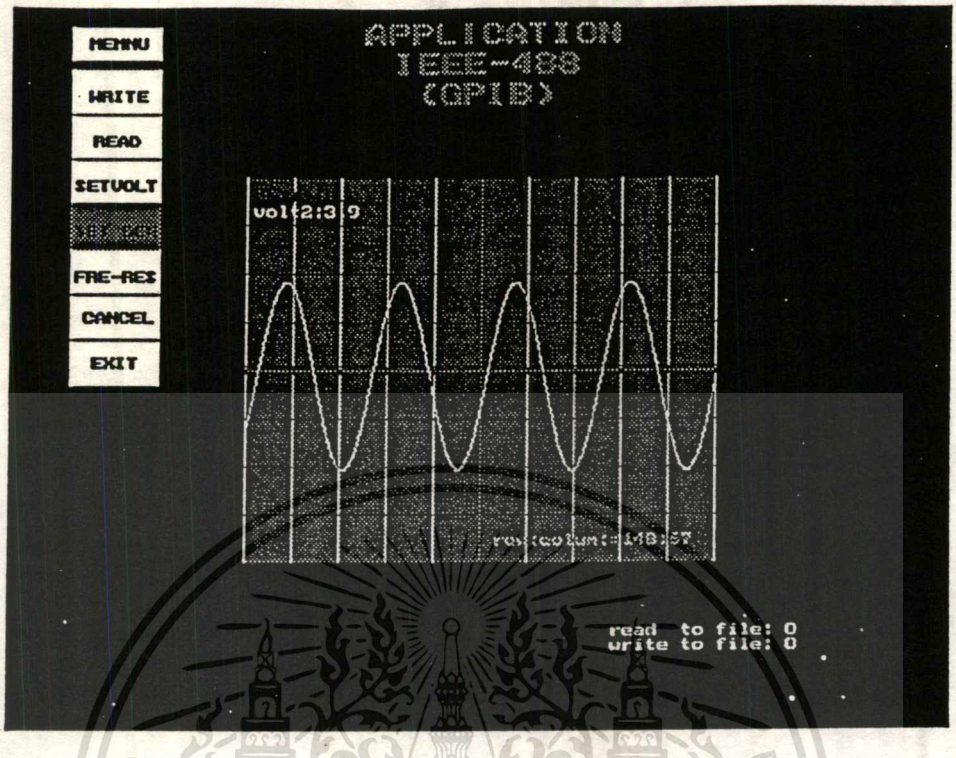


รูปที่ 7.9 แสดงประเภทย่อยของการอ่านข้อมูล

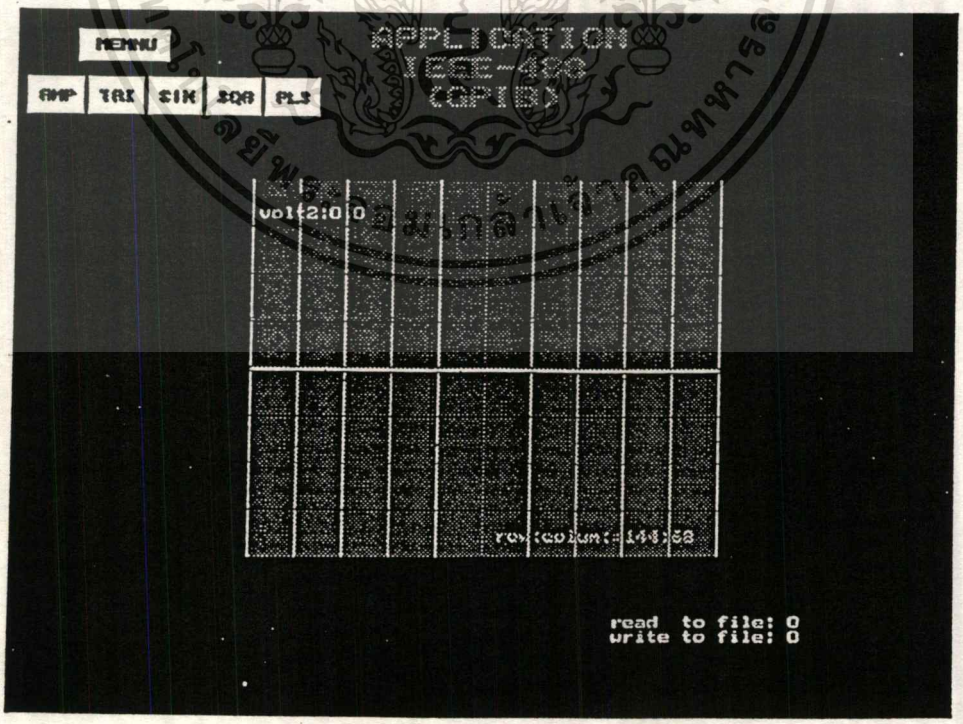


รูปที่ 7.10 แสดงการใช้งาน เมนู เซ็ต โวลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

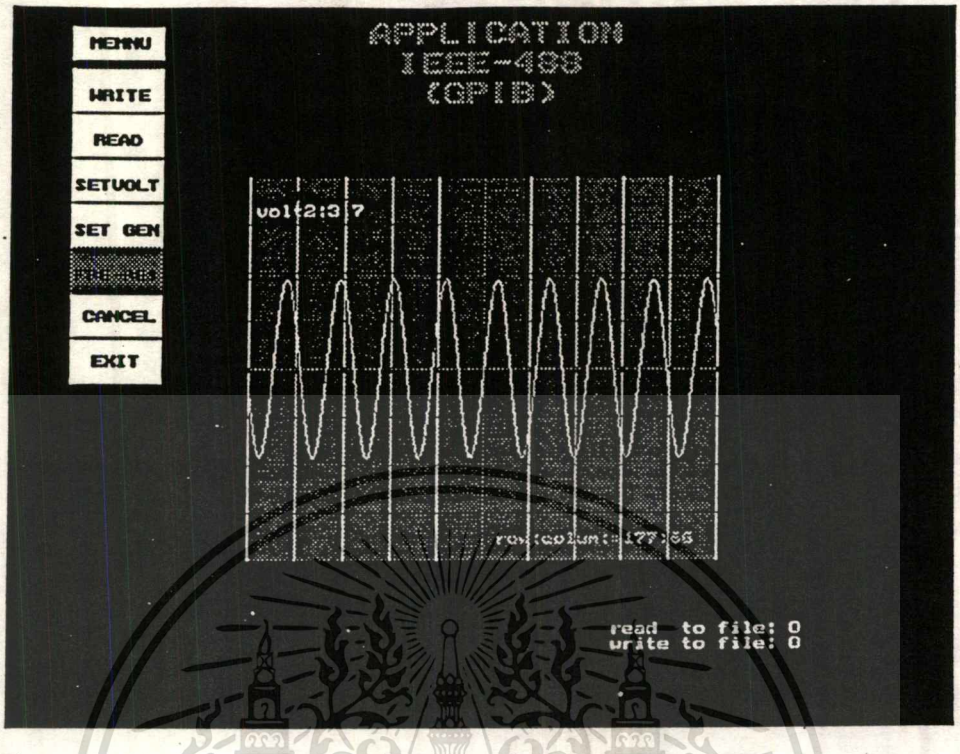


รูปที่ 7.11 แสดงการใช้งาน เมนู เซ็ต เจนเนอร์เรเตอร์

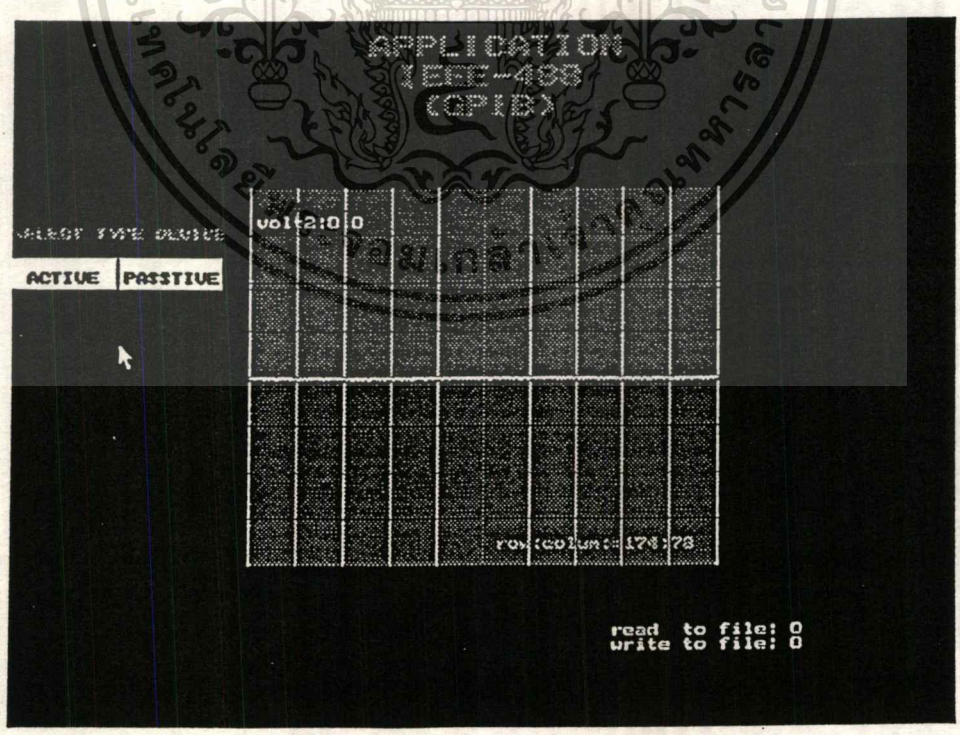


รูปที่ 7.12 แสดงการเซ็ต รูปสัญญาณที่ต้องการ ของ เจนเนอร์เรเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

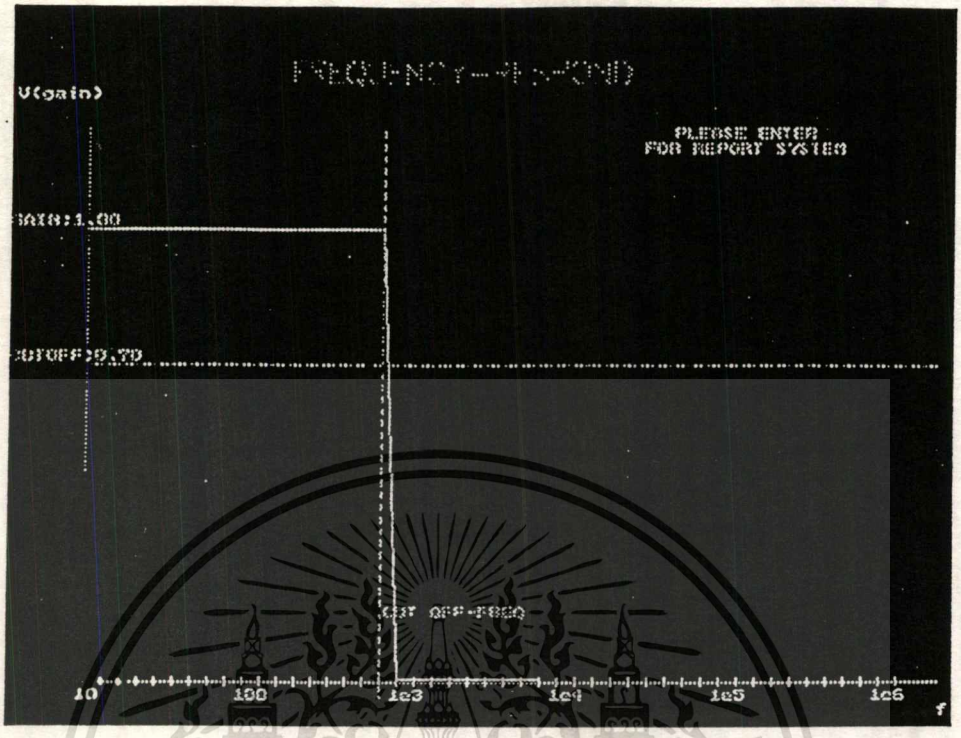


รูปที่ 7.13 แสดงการใช้งาน เมนนู ความถี่ เรสปอร์น

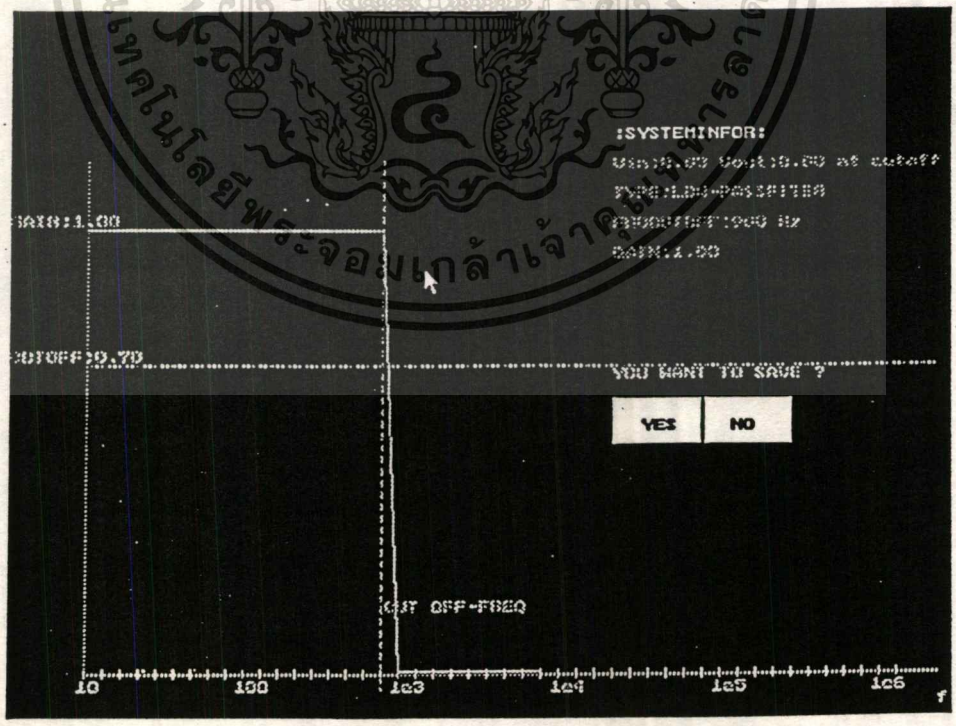


รูปที่ 7.14 แสดงเมนนูย่อย ของ เมนนู ความถี่ เรสปอร์น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

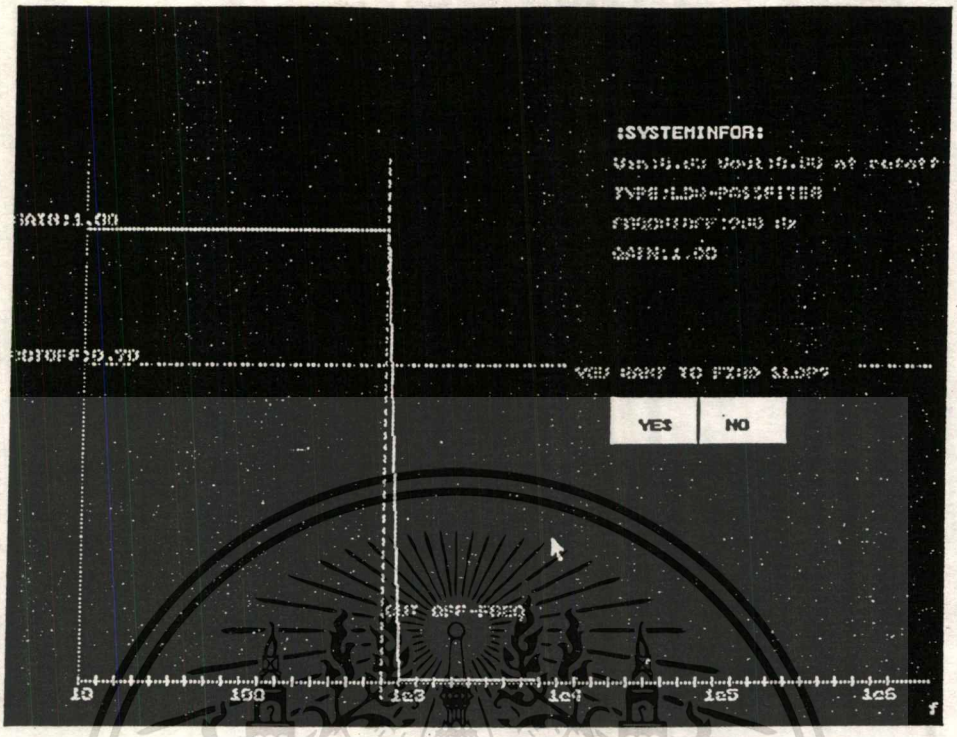


รูปที่ 7.15 แสดงรูปการตอบสนองของความถี่ของวงจร หรือ ระบบ

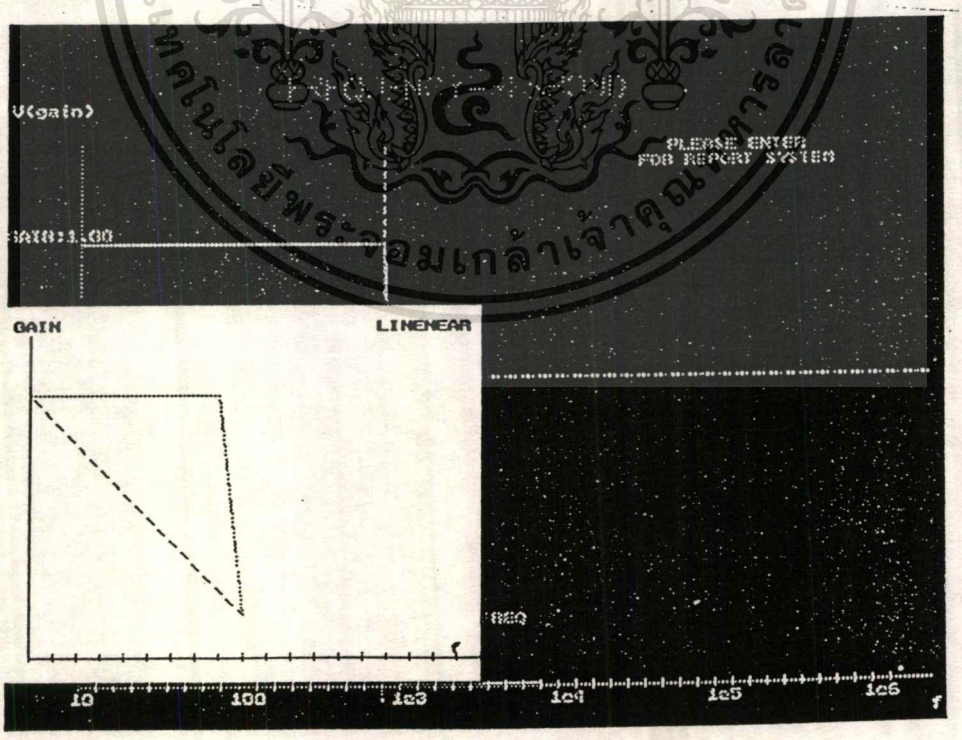


รูปที่ 7.16 แสดงค่าผลตอบสนองของความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.17 แสดงผลการเลือกว่าต้องการความชันของวงจรหรือไม่



รูปที่ 7.18 แสดงความชันของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3 โปรแกรมควบคุมและประยุกต์ใช้งาน

```
#include <math.h>
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#include <dos.h>
#include <string.h>
#include <stdlib.h>
#include <alloc.h>
#include <ctype.h>

char val[4096],wrt[50];
char smouse[50],SLOP[30];
char file[20];
char cc[10],ii[20],j[15],yy[40];
char a[10],b[10],c[10],dd[10],e[10],f[10];
char x[30],y[30],g[10],aa[15],bb[30],hh[30];
char *p;
int m,k,CH,nn,T,t,mm,num,SS,HIG,Devid;
int ud,mouse,row,m1,m2,colum,right,left;
float volt1=1,volt2=1,gain[45],gain2[20];
float MAX,W,G;
unsigned float v1,v2,size;
int drive=DETECT,gmode;

int plotwaveform(int PlotColor,int Channal,char *value);
void mouseinformation(int *right,int *left,int *row,int *colum);
void start(void);
void close(void);
void scope_screen(void);
void screen(void);
void mouseinitial(void);
void mousehind(void);
void mouseshow(void);
void cursoron(void);
void cursoroff(void);
void memnu(void);
void cancel(void);
void Write(void);
void Write2(void);
void Read(void);
void Read2(void);
void Read3(void);
void Read4(void);
void Setvolt(void);
void showmemnu(void);
void SetGen(void);
void fre_res(void);
void fre_res2(void);
void findmaxnitude(void);
void findmaxnitude2(void);
void findmaxnitude3(void);
void find_cutoff1(void);
void find_cutoff2(void);
void findlinear1(void);
void findlinear2(void);
void plot(void);
void plot2(void);
void plot3L(void);
void plot3H(void);
void plot4(void);
void plot5(void);
void plot6(void);

void main(void)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
                        ud=ibfind("DEVC ");
                        initgraph(&drive,&gmode,"");
                        mouseinitial();
                        mouseshow();
                        start();
                        showmemnu();
                        scope_screen();
                        mouseinformation(&right,&left,&row,&colum);
while(!((40<=colum)&&(colum<=100)&&(220<=row)&&(row<=250)&&(left))) {
                        mouseinformation(&right,&left,&row,&colum);
                        sprintf(smouse,"row:colum:=%d:%d",row,colum);

if((40<=colum)&&(colum<=100)&&(13<=row)&&(row<=35)&&(left))   memnu();

if(!((40<=colum)&&(colum<=100)&&(190<=row)&&(row<=220)&&(left)))  {
                        ibwrt(ud,"DIG 3",5);
                        ibrd(ud,val,4096);
                        putimage(160,112,p,COPY_PUT);
                        plotwaveform(16,1,val);
                        plotwaveform(15,2,val);
        }

if((40<=colum)&&(colum<=100)&&(40<=row)&&(row<=70)&&(left)) Write();

if((40<=colum)&&(colum<=100)&&(70<=row)&&(row<=100)&&(left))  Read();

if((40<=colum)&&(colum<=100)&&(100<=row)&&(row<=130)&&(left)) Setvolt();

if((40<=colum)&&(colum<=100)&&(130<=row)&&(row<=160)&&(left)) SetGen();

if((40<=colum)&&(colum<=100)&&(160<=row)&&(row<=190)&&(left)) fre_res();

if((40<=colum)&&(colum<=100)&&(190<=row)&&(row<=220)&&(left)) cancel();

        }
                        free((void *)p);
                        closegraph();
                        printf("\007");
 }

void start()
{
                        mousehind();
                        settextstyle(0,0,2);
                        setcolor(3);
                        outtextxy(230,10,"APPLICATION");
                        outtextxy(250,35,"IEEE-488");
                        outtextxy(290,70,"OR");
                        outtextxy(90,100,"Generral Purpose Interface Bus");
                        outtextxy(265,125,"(GPIB)");
                        setfillstyle(2,9);
                        bar(170,260,270,290);
                        bar(370,260,470,290);
                        setcolor(15);
                        rectangle(70,250,170,300);
                        rectangle(270,250,370,300);
                        rectangle(470,250,570,300);
                        setfillstyle(1,0);
                        bar(210,250,230,300);
                        bar(410,250,430,300);
                        settextstyle(1,0,1);
                        setcolor(2);
                        outtextxy(90,275,"COMPUTOR");
                        outtextxy(300,275,"SCOPE");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        outtextxy(485,275,"GENERATOR");
                        settextstyle(0,0,1);
                        setcolor(15);
                        outtextxy(470,365,"by...");
                        setcolor(4);
                        outtextxy(480,410,"2. TRIRONG  36013110");
                        outtextxy(480,395,"1. PRAMATE  36013109");
                        outtextxy(480,425,"3. YOTEE    36013119");
                        setfillstyle(1,8);
                        bar(70,350,190,375);
                        bar(70,410,190,435);
                        setfillstyle(1,7);
                        bar(66,344,70,375);
                        bar(66,344,190,350);
                        bar(66,404,70,435);
                        bar(66,404,190,410);
                        setcolor(0);
                        line(65,345,70,350);
                        line(65,405,70,410);
                        outtextxy(100,360,"LINK OFF");
                        outtextxy(115,420,"EXIT");
                        mouseshow();
                        do{
                        mouseinformation(&right,&left,&row,&colum);
if((70<=colum)&&(colum<=190)&&(350<=row)&&(row<=370)&&(left))
        {
                        mousehind();
                        delay(500);
                        setfillstyle(1,0);
                        bar(50,320,200,380);
                        setfillstyle(1,8);
                        bar(70,350,190,375);
                        setfillstyle(1,7);
                        bar(70,375,195,380);
                        bar(190,350,195,380);
                        setcolor(0);
                        line(190,375,195,380);
                        outtextxy(103,360,"LINK ON");
                        delay(700);
                        setfillstyle(2,4);
                        bar(170,260,270,290);
                        bar(370,260,470,290);
                        delay(3000);
                        clearviewport();
                                                                }
if((70<=colum)&&(colum<=190)&&(410<=row)&&(row<=435)&&(left))
        {
                        mousehind();
                        delay(500);
                        setfillstyle(1,0);
                        bar(50,400,200,445);
                        setfillstyle(1,8);
                        bar(70,410,190,435);
                        setfillstyle(1,7);
                        bar(70,435,195,440);
                        bar(190,410,195,440);
                        setcolor(0);
                        line(190,435,195,440);
                        outtextxy(115,420,"EXIT");
                        delay(2000);
                        printf("\007");
                        clearviewport();
                        closegraph();
                        cursoron();
                        exit(1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                                                    }
        }
while(!((70<=colum)&&(colum<=190)&&(350<=row)&&(row<=435)&&(left))) ;
if(left)  left=0;
                        mouseshow();
                        settextstyle(0,0,0);
                                                                    }
int plotwaveform(int PlotColor,int Channal,char *value)
{
    int i,R1,R2;
    int ReadLevel;
    int ch;
    char s[20];
    int Max[2],Min[2];
    float v;
                        setcolor(PlotColor);
                        if(Channal==2)      ch=2048;
                        else                ch=0;
                        for(i=0;i<=320;i++)     {
                        if(i==0){
                        R1=(int)(1.27*(double)(value[ch]-128));
                        ReadLevel=112+128-R1;
                        if((ReadLevel)<(112))  ReadLevel=112;
                        if((ReadLevel)>(112+256)) ReadLevel=112+256;
                        moveto(160,(ReadLevel));
                        Max[Channal]=ReadLevel;
                        Min[Channal]=ReadLevel;
}
          else

      {
                        R2=(int)(1.27*(double)(value[ch+(int)i*6.39]-128));
                        ReadLevel=112+128-R2;
                        if((ReadLevel)<(112))  ReadLevel=112;
                        if((ReadLevel)>(112+256)) ReadLevel=112+256;
                        if((ReadLevel>(112))&&((ReadLevel)<(112+256)))
         {
                        lineto(160+i,112+128-R2);
                        if(ReadLevel<=Min[Channal]) Min[Channal]=ReadLevel;
                        if(ReadLevel>=Max[Channal]) Max[Channal]=ReadLevel;
         }
      }
}
                        if(Max[Channal]!=Min[Channal]) {
if(Channal==1)    {
                        v=((Max[Channal]-Min[Channal])/(32.4))*(volt1);
                        v1=v;     }
if(Channal==2)     {
                        v=((Max[Channal]-Min[Channal])/(32.4))*(volt2);
                        v2=v;
                            }
                                                                    }
                        if(Max[Channal]==Min[Channal])
                        v=0.00;
                        sprintf(y,"write to file: %d ",m);  m=0;
                        sprintf(s,"volt%d:%2.1f",Channal,v);
                        sprintf(x,"read  to file: %d ",k);
                        if(Channal==1)  outtextxy(165,122,s);
                        if(Channal==2) outtextxy(165,132,s);
                        setcolor(10);
                        outtextxy(330,350,smouse);
                        setcolor(14);
                        outtextxy(410,410,x);
                        outtextxy(410,420,y);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
void scope_screen(void)
{
  int i;
  int x=160,y=112;
  char a[12],b[15];
                        mousehind();
                        setcolor(9);
                        setfillstyle(1,9);
                        bar(x,y,x+320,y+256);
                        setcolor(7);
                        for(i=0;i<=10;i++)
                        line(x+i*32,y,x+i*32,y+256);
                        for(i=0;i<=50;i++)
  {
                        setcolor(BLACK);
                        line((x+i*6.4),(y+127),(x+i*6.4),(y+130));
  }
                        for(i=0;i<=8;i++)
                        line(x,y+i*32,x+320,y+i*32);
                        for(i=0;i<=50;i++)
  {
                        setcolor(BLACK);
                        line((x+157),(y+(i*6.4)),(x+162),(y+(i*6.4)));
  }
                        setcolor(RED);
                        line(x+160,y,x+160,y+256);
                        line(x,y+128,x+320,y+128);
                        setfillstyle(1,0);
                        bar(360,400,600,470);
                        size=imagesize(160,112,480,368);
                        p=(char *)malloc(size);
                        getimage(160,112,480,368,p);
                        sprintf(a,"APPLICATION");
                        sprintf(b,"IEEE-488");
                        sprintf(c,"(GPIB)");
                        settextstyle(0,0,2);
                        setcolor(4);
                        outtextxy
                        outtextxy
                        outtextxy
                        settextsty
                        mouseshow
 }
 void screen(void)
{
  int i;
  int x=160,y=112;
  char a[12],b[15];
                        mousehind();
                        setcolor(9);
                        setfillstyle(1,9);
                        bar(x,y,x+320,y+256);
                        setcolor(7);
                        for(i=0;i<=10;i++)
                        line(x+i*32,y,x+i*32,y+256);
                        for(i=0;i<=50;i++)
  {
                        setcolor(BLACK);
                        line((x+i*6.4),(y+127),(x+i*6.4),(y+130));
  }
                        for(i=0;i<=8;i++)
                        line(x,y+i*32,x+320,y+i*32);
                        for(i=0;i<=50;i++)
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        setcolor(BLACK);
                        line((x+157),(y+(i*6.4)),(x+162),(y+(i*6.4)));
   }
                        setcolor(RED);
                        line(x+160,y,x+160,y+256);
                        line(x,y+128,x+320,y+128);
                        setfillstyle(1,0);
                        bar(360,400,600,470);
                        sprintf(a,"APPLICATION");
                        sprintf(b,"IEEE-488");
                        sprintf(c,"(GPIB)");
                        settextstyle(0,0,2);
                        setcolor(4);
                        outtextxy(240,10,a);
                        outtextxy(260,30,b);
                        outtextxy(275,50,c);
                        settextstyle(0,0,0);
                        mouseshow();
  }
 void mouseinitial(void)
 {
                        union REGS regs;
                        regs.x.ax=0;
                        int86(0x33,&regs,&regs);
 }
 void mouseshow(void)
 {
                        union REGS regs;
                        regs.x.ax=1;
                        int86(0x33,&regs,&regs);
 }
void mouseinformation(int *right,int *left,int *row,int *colum)
{
                        union REGS regs;
                        regs.x.ax=3;
                        int86(0x33,&regs,&regs);
                        switch(regs.x.bx){
                        case 1: *left=1;
                        break;
                        case 2: *right=2;
                        break;
                        case 3: *right=3;
                                *left=3;
                }
                        *row=regs.x.dx;
                        *colum=regs.x.cx;

}
void mousehind(void)
{
                        union REGS
                        regs.x.ax=2;
                        int86
}
void cursoroff(void)
{
                        union REGS regs;
                        regs.h.ah=1;
                        regs.x.cx=0x2000;
                        int86(0x10,&regs,&regs);
}
void cursoron(void)
{
                        union REGS regs;
                        regs.h.ah=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        regs.h.ch=6;
                        regs.h.ch=7;
                        int86(0x10,&regs,&regs);
}
void Write(void)
{
char cc[30];

            FILE    *fp;
                        free((void *)p);
                        mousehind();
                        setfillstyle(SOLID_FILL,4);
                        bar(40,40,100,70);
                        setcolor(1);
                        outtextxy(52,55,a);
                        delay(300);
                        setfillstyle(SOLID_FILL,0);
                        bar(40,40,220,70);
                        setfillstyle(SOLID_FILL,0);
                        bar(40,40,160,350);
                        sprintf(cc,"FILE MANE TO WRITE");
                        setcolor(5);
                        outtextxy(3,50,cc);
                        gotoxy(21,4);
                        gets(file);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,37,240,70);
                        if((fp=fopen(file,"a+"))==NULL) {
                        free((void *)p);
                        printf("error in open file  \n ");
                        exit(1);        }
                        fputs(val,fp);
                        m++;
                        if(ferror(fp))      {
                        printf(" error in write  \n ");
                        exit(1);            }
                        memnu();
                        mouseshow();
                        scope_screen();
                        left=0;
                        fclose(fp);
}
void memnu(void)
{
   int x=30,i;
                        mousehind();
                        setfillstyle(SOLID_FILL,15);
                        bar(40,40,100,250);
                        for(i=0;i<=6;i++)
   {
                        setcolor(1);
                        line(40,(40+i*x),100,(40+i*x));
   }
                        sprintf(a,"WRITE");
                        sprintf(b,"READ");
                        sprintf(c,"SETVOLT");
                        sprintf(dd,"SET GEN");
                        sprintf(e,"FRE-RES");
                        sprintf(f,"CANCEL");
                        sprintf(g,"EXIT");
                        setcolor(0);
                        outtextxy(52,55,a);
                        outtextxy(55,85,b);
                        outtextxy(43,115,c);
                        outtextxy(43,145,dd);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        outtextxy(43,175,e);
                        outtextxy(49,202,f);
                        outtextxy(55,232,g);
                        mouseshow();
                        left=0;

        }
void cancel(void)
  {

                        mousehind();
                        setfillstyle(SOLID_FILL,4);
                        bar(40,190,100,220);
                        setcolor(1);
                        outtextxy(49,202,f);
                        delay(300);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,0,150,250);
                        showmemnu();
                        mouseshow();
                        left=0;
  }
void Setvolt(void)
  {
  float voltage;
  char cc[20],ww[20],hh[20],nn[2];
                        mousehind();
                        setfillstyle(SOLID_FILL,4);
                        bar(40,100,100,130);
                        setcolor(1);
                        outtextxy(43,115,c);
                        delay(300);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,35,140,250);
                        sprintf(hh,"SET CH:" );
                        setcolor(5);
                        outtextxy(3,50,hh);
                        gotoxy(10,4);
                        gets(nn);
                        CH=atoi(nn);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,35,140,200);
                        sprintf(cc,"VOLT/DIV:" );
                        setcolor(5);
                        outtextxy(3,50,cc);
                        gotoxy(13,4);
                        gets(ww);
                        voltage=atof(ww);
                        if(CH==1) volt1=voltage;
                        if(CH==2) volt2=voltage;
                        setfillstyle(SOLID_FILL,0);
                        bar(0,37,240,70);
                        memnu();
                        left=0;
                        mouseshow();
    }
void showmemnu(void)
  {
    char s[10];
                        cursoroff();
                        sprintf(s,"MEMNU");
                        setfillstyle(SOLID_FILL,15);
                        bar(40,13,100,35);
                        setcolor(BLACK);
                        line(40,35,100,35);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        outtextxy(52,20,s);
                        left=0;
                        mouseinitial();
                        mouseshow();
                    }
 void Read(void)
{
   char gg[10],zz[30];

                        free((void *)p);
                        mousehind();
                        setfillstyle(SOLID_FILL,4);
                        bar(40,70,100,100);
                        setcolor(1);
                        outtextxy(55,85,b);
                        delay(300);
                        left=0;
                        setfillstyle(SOLID_FILL,0);
                        bar(0,35,140,250);
                        sprintf(zz,"FILE NAME TO READ ");
                        setcolor(5);
                        outtextxy(3,50,zz);
                        gotoxy(20,4);
                        gets(file);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,37,240,70);
                        setfillstyle(SOLID_FILL,15);
                        bar(0,50,140,70);
                        setcolor(0);
                        line(70,50,70,70);
                        sprintf(aa,"DATA");
                        sprintf(bb,"FRE-RES");
                        setcolor(5);
                        outtextxy(17,58,aa);
                        outtextxy(79,58,bb);
 do{
                        mouseinformation(&right,&left,&row,&colum);
                        mouseshow();
 if((0<=colum)&&(colum<=70)&&(50<=row)&&(row<=70)&&(left))
        {
                        mousehind();
                        delay(300);
                        Read2();
                        left=0;
                                                                }
 if((70<=colum)&&(colum<=140)&&(50<=row)&&(row<=70)&&(left))
        {
                        mousehind();
                        Read3();
                        left=0;
                                                                }
        }
 while(!((0<=colum)&&(colum<=140)&&(50<=row)&&(row<=70)&&(right))) ;
                        if(left)  left=0;
                                  right=0;
                        showmemnu();
                        scope_screen();

        }

void Read2(void)
{
   char yo[4096],str10[10],st[20];
   char str7[25],str8[10],str9[10];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                FILE *fp;
                      free((void *)p);
                      left=0;
                      setfillstyle(SOLID_FILL,4);
                      bar(0,50,70,70);
                      setcolor(1);
                      outtextxy(17,58,aa);
                      delay(500);
                      setfillstyle(SOLID_FILL,0);
                      bar(0,0,140,250);
                      sprintf(str7,"PLEASE SELECT TYPE");
                      sprintf(str8,"FOR READ");
                      sprintf(str9,"CONT");
                      sprintf(str10,"SEARC");
                      setfillstyle(SOLID_FILL,15);
                      bar(0,50,140,70);
                      setcolor(0);
                      line(70,50,70,70);
                      setcolor(3);
                      outtextxy(5,25,str7);
                      outtextxy(40,38,str8);
                      outtextxy(17,60,str9);
                      outtextxy(85,60,str10);
               do{
                      mouseinformation(&right,&left,&row,&colum);
                      mouseshow();
if((0<=colum)&&(colum<=70)&&(50<=row)&&(row<=70)&&(right))
        {
                      mousehind();
                      sprintf(st,"ERORR FILE OPEN");
                      setfillstyle(SOLID_FILL,4);
                      bar(0,50,70,70);
                      setcolor(0);
                      outtextxy(17,60,str9);
                      delay(500);
                      setfillstyle(SOLID_FILL,0);
                      bar(0,0,150,250);
                      if((fp=fopen(file,"r"))==NULL)  {
                      free((void *)p);
                      setfillstyle(SOLID_FILL,0);
                      bar(0,0,150,250);
                      setcolor(3);
                      outtextxy(10,50,st);
                      gets(a);
                      setfillstyle(SOLID_FILL,0);
                      bar(0,0,150,250);
                      break;              }

                      while(!feof(fp))  {
                      fgets(yo,4096,fp);
                      k++;
                      if(ferror(fp)) {
                      printf("error file read \n");
                      exit(1);        }
                      screen();
                      plotwaveform(16,1,yo);
                      plotwaveform(15,2,yo);
                      gets(a);
                                      }
                      fclose(fp);
                      left=0;
                      k=0;
                                                                  }
if((70<=colum)&&(colum<=140)&&(50<=row)&&(row<=70)&&(right))
     {
```

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        mousehind();
                        setfillstyle(SOLID_FILL,4);
                        bar(70,50,140,70);
                        setcolor(0);
                        outtextxy(85,60,str10);
                        delay(500);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,0,150,250);
                        left=0;
                        Read4();
                        left=0;
                                                                    }
        }
while(!((0<=colum)&&(colum<=160)&&(50<=row)&&(row<=100)&&(right))) ;
                        left=0;
        }
void Read3(void)
{
   char str3[15],str5[20];

                        FILE *fp;
                        sprintf(str3,"PLEASE ENTER");
                        sprintf(str5,"FOR REPORT SYSTEM");
                        setfillstyle(SOLID_FILL,4);
                        bar(70,50,140,70);
                        setcolor(1);
                        outtextxy(79,58,bb);
                        delay(500);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,35,140,250);
                        if((fp=fopen(file,"r"))==NULL)  {
                        free((void *)p);
                        printf("eroor file open\n");
                        exit(1);                        }
                        fread(gain,sizeof gain,1,fp);
                        if(ferror(fp)) {
                        printf("error file read \n");
                        exit(1);        }
                        findmaxnitude3();
                        if(gain[4]>gain[25])    find_cutoff1();
                        if(gain[4]<gain[25])    find_cutoff2();
                        HIG=300;
                        plot();
                        setcolor(2);
                        outtextxy(450,80,str3);
                        outtextxy(430,90,str5);
                        gets(a);
                        plot4();
                        setfillstyle(SOLID_FILL,0);
                        bar(0,0,600,100);
                        plot5();
                        gets(a);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,0,650,550);
                        fclose(fp);
                        right=1;
                        left=0;
                        k=0;

   }
void Read4(void)
{
   char yo[4096],str14[30],str16[20];
   char str11[25],str12[15],str13[5];
   int NUM1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        FILE *fp;
                        free((void *)p);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,0,140,250);
                        sprintf(str11,"PLEASE SELECT SIGNAL");
                        sprintf(str12,"FOR NUMBER.....");
                        sprintf(str14,"..END OF FILE..");
                        sprintf(str16,"ERORR FOR READ DATD");
                        setcolor(3);
                        outtextxy(5,25,str11);
                        outtextxy(13,38,str12);
                        gotoxy(14,3);
                        gets(str13);
                        NUM1=atoi(str13);
                        left=0;
                        setfillstyle(SOLID_FILL,0);
                        bar(0,0,200,100);
                        if((fp=fopen(file,"r"))==NULL)  {
                        free((void *)p);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,0,150,250);
                        setcolor(15);
                        outtextxy(18,50,"ERORR FILE OPEN");
                        outtextxy(0,60,"PLEASE ENTER FOR EXIT");
                        gets(a);
                        free((void *)p);
                        exit(1);                              }
                        fseek(fp,(long)((NUM1-1)*sizeof yo),0);
                        fgets(yo,4096,fp);
        if(feof(fp)) {
                        setcolor(15);
                        outtextxy(13,50,str14);
                        outtextxy(0,65,str16);
                        gets(a);
                        free((void *)p);
                        exit(1);
                              }
                        k++;
                        screen();
                        plotwaveform1(  yo);
                        plotwaveform2(  yo);
                        gets(a);
                        fclose(fp);
                        left=0;
                        k=0;

}
void  SetGen(void)
{

  char amp[5],frq[20],oo,nn;
  char s[10],aa[30],bb[30],wrt[70];
  char a[5],b[5],c[5],d[5],e[5],T[5];
  float A,F;
  int i,x=40,ud;
                        mousehind();
                        free((void *)p);
                        setfillstyle(SOLID_FILL,4);
                        bar(40,130,100,160);
                        setcolor(1);
                        outtextxy(43,145,dd);
                        delay(300);
                        left=0;
                        setfillstyle(SOLID_FILL,0);
                        bar(0,35,150,250);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        delay(100);
                        left=0;
                        setfillstyle(SOLID_FILL,0);
                        bar(0,35,240,100);
                        mouseshow();
                        sprintf(aa,"PUT AMPLITUE:" );
                        sprintf(bb,"PUT FREQUENCY:" );
                        setcolor(5);
                        outtextxy(3,50,aa);
                        gotoxy(16,4);
                        gets(amp);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,37,240,70);
                        outtextxy(3,50,bb);
                        gotoxy(16,4);
                        gets(frq);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,37,240,70);
                        ud=ibfind("DEVD");
                        ibclr(ud);
                        A=atof(amp);
                        F=atof(frq);
                        sprintf(wrt,"FRQ:%f AMP:%f %s ",F,A;T);
                        ibwrt(ud,wrt,strlen(wrt));
                        ibwrt(ud,"ot1 ",4);
                        ibloc(ud);
                        scope_screen();

 }
void fre_res(void)
{
        int ud1,n=1,i;
        char a[2],b[30],ccc[10],eee[10],fff[30];
        char str3[15],str4[15],str5[20];
        float d;
                        free((void *)p);
                        left=0;
                        mousehind();
                        delay(500);
                        setfillstyle(SOLID_FILL,4);
                        bar(40,160,100,190);
                        setcolor(1);
                        outtextxy(43,175,e);
                        delay(350);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,0,100,250);
                        sprintf(ccc,"ACTIVE");
                        sprintf(eee,"PASSTIVE");
                        sprintf(fff,"SELECT TYPE DEVICE");
                        sprintf(str3,"PLEASE ENTER");
                        sprintf(str4,"AND SELECT T/D");
                        sprintf(str5,"FOR REPORT SYSTEM");
                        setfillstyle(SOLID_FILL,15);
                        bar(0,160,140,180);
                        line(70,160,70,180);
                        setcolor(6);
                        outtextxy(10,170,ccc);
                        outtextxy(75,170,eee);
                        outtextxy(0,140,fff);

                do{     mouseinformation(&right,&left,&row,&colum);
                        mouseshow();
 if((0<=colum)&&(colum<=70)&&(160<=row)&&(row<=180)&&(left))  {
                        mousehind();
                        setfillstyle(SOLID_FILL,4);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        delay(100);
                        left=0;
                        setfillstyle(SOLID_FILL,0);
                        bar(0,35,240,100);
                        mouseshow();
                        sprintf(aa,"PUT AMPLITUE:" );
                        sprintf(bb,"PUT FREQUENCY:" );
                        setcolor(5);
                        outtextxy(3,50,aa);
                        gotoxy(16,4);
                        gets(amp);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,37,240,70);
                        outtextxy(3,50,bb);
                        gotoxy(16,4);
                        gets(frq);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,37,240,70);
                        ud=ibfind("DEVD");
                        ibclr(ud);
                        A=atof(amp);
                        F=atof(frq);
                        sprintf(wrt,"FRQ:%f AMP:%f %s ",F,A,T);
                        ibwrt(ud,wrt,strlen(wrt));
                        ibwrt(ud,"ot1 ",4);
                        ibloc(ud);
                        scope_screen();

 }
void fre_res(void)
{
        int ud1,n=1,i;
        char a[2],b[30],ccc[10],eee[10],fff[30];
        char str3[15],str4[15],str5[20];
        float d;
                        free((void *)p);
                        left=0;
                        mousehind();
                        delay(500);
                        setfillstyle(SOLID_FILL,4);
                        bar(40,160,100,190);
                        setcolor(1);
                        outtextxy(43,175,e);
                        delay(350);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,0,100,250);
                        sprintf(ccc,"ACTIVE");
                        sprintf(eee,"PASSTIVE");
                        sprintf(fff,"SELECT TYPE DEVICE");
                        sprintf(str3,"PLEASE ENTER");
                        sprintf(str4,"AND SELECT T/D");
                        sprintf(str5,"FOR REPORT SYSTEM");
                        setfillstyle(SOLID_FILL,15);
                        bar(0,160,140,180);
                        line(70,160,70,180);
                        setcolor(6);
                        outtextxy(10,170,ccc);
                        outtextxy(75,170,eee);
                        outtextxy(0,140,fff);

                do{     mouseinformation(&right,&left,&row,&colum);
                        mouseshow();
if((0<=colum)&&(colum<=70)&&(160<=row)&&(row<=180)&&(left))   {
                        mousehind();
                        setfillstyle(SOLID_FILL,4);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        bar(0,160,70,180);
                        setcolor(0);
                        outtextxy(10,170,ccc);
                        delay(300);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,100,150,200);
                        HIG=100;

if((70<=colum)&&(colum<=140)&&(160<=row)&&(row<=180)&&(left)){
                        mousehind();
                        setfillstyle(SOLID_FILL,4);
                        bar(70,160,140,180);
                        setcolor(0);
                        outtextxy(75,170,eee);
                        delay(300);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,100,150,200);
                        HIG=300;
                                                              }

while(!((0<=colum)&&(colum<=140)&&(160<=row)&&(row<=180)&&(left))) ;
              if(left)   left=0;
              delay(300);

                        ud1=ibfind("DEVD");
                        ibclr(ud1);

        while(n<=3){
                        setcolor(2);
                        outtextxy(10,50,str3);
                        outtextxy(5,60,str4);
                        gets(a);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,0,150,100);
        for(i=1;i<=9;i++){
                        d=(i*pow(10,n));
                        sprintf(wrt,"FRQ:%f AMP:5 ",d);
                        ibwrt(ud1,wrt,strlen(wrt));
                        ibwrt(ud1,"ot1 ",4);
                        ibwrt(ud,"DIG 3",5);
                        ibrd(ud,val,4096);
                        screen();
                        plotwaveform(16,1,val);
                        plotwaveform(15,2,val);
                        findmaxnitude();
        }
                        n++;
        }
                        ibloc(ud1);
                        if(gain[4]>gain[25])    find_cutoff1();
                        if(gain[4]<gain[25])    find_cutoff2();
                        plot();
                        setcolor(2);
                        outtextxy(450,80,str3);
                        outtextxy(430,90,str5);
                        gets(a);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,0,600,100);
                        plot4();
                        plot5();
                        Write2();
                        fre_res2();
                        mousehind();
                        setfillstyle(SOLID_FILL,0);
                        bar(0,0,650,550);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        left=0;
                        mm=0;
                        m2=0;
                        nn=0;
                        m1=0;
                        W=0;
                        scope_screen();
                        mouseshow();
                        showmemnu();
                    }
void plot(void)
{
   char c[10],b[10],d[15],e[20];
   char z[8][15];
   int i,x=12,a,y=108;
   int h=100,n,m=12;
   float A1,A2,M,N,E,I,vo;

                        mousehind();
                        setfillstyle(SOLID_FILL,0);
                        bar(0,0,650,450);
                        setcolor(4);
                        line(50,80,50,450);
                        line(50,450,630,450);
                for(i=0;i<=45;i++){
                        setcolor(15);
                        line(62+(i*x),447,62+(i*x),452);  }
                for(a=0;a<=5;a++) {
                        setcolor(3);
                        line(50+(a*y),447,50+(a*y),452);  }
                        sprintf(z[0],"10");
                        sprintf(z[1],"100");
                        sprintf(z[2],"1e3");
                        sprintf(z[3],"1e4");
                        sprintf(z[4],"1e5");
                        sprintf(z[5],"1e6");
                        sprintf(z[6],"f");
                        sprintf(z[7],"V(gain)");
                        outtextxy(45,455,z[0]);
                        outtextxy(153,455,z[1]);
                        outtextxy(261,455,z[2]);
                        outtextxy(369,455,z[3]);
                        outtextxy(477,455,z[4]);
                        outtextxy(585,455,z[5]);
                        outtextxy(630,465,z[6]);
                        outtextxy(3,55,z[7]);

      for(n=3;n<=26;n++) {
         if(n==3){
                        setcolor(14);
                        A1=(float)((HIG*gain[n])-128);
                        M=h+220-A1;
                        moveto(50,(h+220-A1));
                                  }
         else {
                        A1=(float)((HIG*gain[n])-128);
      /*                k=h+220-A1;
                        if(k<=M)  M=k;    */
                        I=(50+(T*m));
      if(num!=1)        delay(250);
                        lineto(50+(m*n),(h+220-A1));
                    }

         }
                        E=(float)((HIG*G)-128);          /*---350---*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        N=h+220-E;
                        setlinestyle(DOTTED_LINE,0,1);
                        setcolor(2);
                        line(50,N,630,N);
                        setcolor(2);
                        sprintf(c,"GAIN:%2.2f",W);
                        outtextxy(0,(M-9),c);
                        sprintf(b,"CUTOFF:%2.2f",G);
                        outtextxy(0,(N-9),b);
                        setlinestyle(DASHED_LINE,0,1);
                        setcolor(5);
                        line(I,80,I,460);
                        sprintf(d,"CUT OFF-FREQ");
                        setcolor(5);
                        outtextxy(I+3,400,d);
                        setcolor(1);
                        sprintf(e,"FREQUENCY-RESPOND");
                        settextstyle(3,0,3);
                        outtextxy(190,30,e);
                        settextstyle(0,0,0);
                        setlinestyle(SOLID_LINE,0,1);
                        num=0;
    }
void  plot2(void)
{
   float slop;

                        slop=SS/Devid;      /*--- 16---,25-----*/

    if((slop<6)&&(slop>=1))   sprintf(SLOP,"SLOPE   6 Db/OCT    1ST");
    if((slop<9)&&(slop>=6.1)) sprintf(SLOP,"SLOPE  12 Db/OCT    2ST");
    if((slop<11)&&(slop>=9))  sprintf(SLOP,"SLOPE  18 Db/OCT    3ST");
    if(slop>=12)              sprintf(SLOP,"SLOPE  >18 Db/OCT  >4ST");

}
void plot3L(void)
{
 int x=0,y=190,i,n,m,k=460,hig;
 char a[15],b[5],c[5];
 float A1,M,M1;

                        hig=HIG/2;
                        num=1;
                        plot();
                        delay(1000);
                        setfillstyle(1,7);
                        for(i=0;i<=32;i++)    {
                        delay(400);
                        bar(x,y,i*10,y+(i*8));     }
                        sprintf(a,"LINENEAR");
                        sprintf(b,"GAIN");
                        sprintf(c,"f");
                        setcolor(16);
                        line(16,210,16,430);
                        line(16,430,360,430);
                        outtextxy(250,200,a);
                        outtextxy(5,200,b);
                        outtextxy(300,420,c);

                        for(i=1;i<=20;i++)          {
                        setcolor(4);
                        line(i*16,427,i*16,432);
                                                    }
                  for(n=1;n<=10;n++) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if(n==1) {
                        setcolor(11);
                        A1=(float)((hig*gain2[n])-150);
                        M1=250-A1;
                        /*  if(k<=M)  M1=k;          */
                        moveto(16,M1);   }
          else {
                        A1=(float)((hig*gain2[n])-150);
                        M=250-A1;

                        if(k<=M)  M=k;
                        delay(250);
                        lineto((16*n),M);
                   }
                }
                        SS=M-M1;
                        setlinestyle(DASHED_LINE,0,1);
                        setcolor(16);
                        lineto(16,M1);
                        setlinestyle(SOLID_LINE,0,1);

 }
void plot3H(void)
{
 int x=0,y=190,i,n,m,k=460,hig;
 char a[15],b[5],c[5];
 float A1,M,M1;
                        num=1;
                        hig=HIG/2;
                        plot();
                        delay(1000);
                        setfillstyle(1,7);
                        for(i=0;i<=32;i++)   {
                        delay(400);
                        bar(x,y,i*10,y+(i*8));    }
                        sprintf(a,"LINENEAR");
                        sprintf(b,"GAIN");
                        sprintf(c,"f");
                        setcolor(16);
                        line(16,210,16,430);
                        line(16,430,360,430);
                        outtextxy(250,200,a);
                        outtextxy(5,200,b);
                        outtextxy(300,420,c);

                        for(i=1;i<=20;i++)          {
                        setcolor(4);
                        line(i*16,427,i*16,432);
                        setcolor(16);

               for(n=1;n<=10;n++) {
                if(n==1) {
                        setcolor(11);
                        A1=(float)((hig*gain2[n])-150);
                        M1=250-A1;
                       /*   if(k<=M)  M1=k;          */
                        moveto(16,M1);   }
          else {
                        A1=(float)((hig*gain2[n])-150);
                        M=250-A1;

                        if(k<=M)  M=k;
                        delay(250);
                        lineto((16+(16*n)),M);
                   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
                SS=M-M1;
                        setlinestyle(DASHED_LINE,0,1);
                        setcolor(16);
                        lineto(16,M1);
                        setlinestyle(SOLID_LINE,0,1);

 }
void plot4(void)
{
float vo;

                switch(t)  {
                case 1:  strcpy(f,"10 ");
                                MAX=10;                 break;
                case 2:  strcpy(f,"20 ");
                                MAX=10;                 break;
                case 3:  strcpy(f,"30 ");
                                MAX=10;                 break;
                case 4:  strcpy(f,"40 ");
                                MAX=10;                 break;
                case 5:  strcpy(f,"50 ");
                                MAX=10;                 break;
                case 6:  strcpy(f,"60 ");
                                MAX=10;                 break;
                case 7:  strcpy(f,"70 ");
                                MAX=10;                 break;
                case 8:  strcpy(f,"80 ");
                                MAX=10;                 break;
                case 9:  strcpy(f,"90 ");
                                MAX=10;                 break;
                case 10: strcpy(f,"100 ");
                                MAX=100;                break;
                case 11: strcpy(f,"200 ");
                                MAX=100;                break;
                case 12: strcpy(f,"300 ");
                                MAX=100;                break;
                case 13: strcpy(f,"400 ");
                                MAX=100;                break;
                case 14: strcpy(f,"500 ");
                                MAX=100;                break;
                case 15: strcpy(f,"600 ");
                                MAX=100;                break;
                case 16: strcpy(f,"700 ");
                                MAX=100;                break;
                case 17: strcpy(f,"800 ");
                                MAX=100;                break;
                case 18: strcpy(f,"900 ");
                                MAX=100;                break;
                case 19: strcpy(f,"1000 ");
                                MAX=1000;               break;
                case 20: strcpy(f,"2000 ");
                                MAX=1000;               break;
                case 21: strcpy(f,"3000 ");
                                MAX=1000;               break;
                case 22: strcpy(f,"4000 ");
                                MAX=1000;               break;
                case 23: strcpy(f,"5000 ");
                                MAX=1000;               break;
                case 24: strcpy(f,"6000 ");
                                MAX=1000;               break;
                case 25: strcpy(f,"7000 ");
                                MAX=1000;               break;
                case 26: strcpy(f,"8000 ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    sprintf(j,":SYSTEMINFOR:");
                    outtextxy(410,80,j);
                    setcolor(4);
                    sprintf(ii,"TYPE:%s",hh);
                    sprintf(g,"FRQCUTOFF:%sHz",f);
                    sprintf(cc,"GAIN:%2.2f",W);
                    outtextxy(410,100,yy);
                    delay(1000);
                    outtextxy(410,120,SLOP);
                    delay(1000);
                    outtextxy(410,140,ii);
                    delay(1000);
                    outtextxy(410,160,g);
                    delay(1000);
                    outtextxy(410,180,cc);

  }
void find_cutoff1(void)
{
    int i;
    float L;

                    strcpy(hh,"LOW-PASSFITER ");
                    L=gain[nn]*.7;
                    for(i=3;i<=35;i++)   {

                    if((gain[i]<L))    break;
                                       }
                    t=i;
                    T=i-1;
}
void find_cutoff2(void)
{
    int i;
    float L;

                    strcpy(hh,"HIGHT-PASSFITER ");
                    L=gain[nn]*.7;
                    for(i=3;i<=35;i++)   {

                    if((gain[i]>L))   break;

                                       }
                    t=i;
                    T=i-1;
}

void findmaxnitude(void)
{
                    if(mm==3)     W=v2/v1;

                    if(mm>=3) {
                    gain[mm]=v2/v1;
                    if(gain[mm]>=W)      {
                                              W=gain[mm];
                                              nn=mm;
                                         }
               }
                    mm++;
                    G=W*.7;
}

void findmaxnitude2(void)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        gain2[m1]=v2/v1;
                        m1++;

}

void findmaxnitude3(void)
{
                        W=gain[3];
                        for(m2=3;m2<=35;m2++) {
                        if(gain[m2]>=W)      {
                        W=gain[m2];
                        nn=m2;
                                             }
                }
                        G=W*.7;

}

void findlinear1(void)
 {
 int i,n,m,ud1,nnn,N;
 float D;
 char aaa[25],bbb[15];

                        sprintf(aaa,"PLEASE ENTER FOR LINEAR");
                        sprintf(bbb,"AND SELEC T/D");
                        setcolor(15);
                        outtextxy(410,230,aaa);
                        outtextxy(420,250,bbb);
                        gets(aaa);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,0,650,550);
                        ud1=ibfind("DEVD");
                        ibclr(ud1);
                        N=atoi(f);
        for(nnn=0;nnn<=10;nnn++)      {

           if(nnn==0)  D=N;

        else    {
                        D=N*nnn;
                        sprintf(wrt,"FRQ:%f
                        ibwrt(ud1,wrt,strle
                        ibwrt(ud1,"ot1 ",4),
                        ibwrt(ud,"DIG 3",5);
                        ibrd(ud,val,4096);
                        screen();
                        plotwaveform(16,1,val);
                        plotwaveform(15,2,val);
                        findmaxnitude2();
                                                              }
                                                                 }
                        setfillstyle(SOLID_FILL,0);
                        bar(0,0,650,550);
                        plot3L();
                        if(HIG==100)    Devid=8;
                        if(HIG==300)    Devid=25;
                        plot2();


}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 void findlinear2(void)
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
```

```
                        gain2[m1]=v2/v1;
                        m1++;

}

void findmaxnitude3(void)
{
                        W=gain[3];
                        for(m2=3;m2<=35;m2++) {
                        if(gain[m2]>=W)      {
                        W=gain[m2];
                        nn=m2;
                                             }
                }
                        G=W*.7;

}

void findlinear1(void)
 {
 int i,n,m,ud1,nnn,N;
 float D;
 char aaa[25],bbb[15];

                        sprintf(aaa,"PLEASE ENTER FOR LINEAR");
                        sprintf(bbb,"AND SELEC T/D");
                        setcolor(15);
                        outtextxy(410,230,aaa);
                        outtextxy(420,250,bbb);
                        gets(aaa);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,0,650,550);
                        ud1=ibfind("DEVD");
                        ibclr(ud1);
                        N=atoi(f);
        for(nnn=0;nnn<=10;nnn++)      {

           if(nnn==0)  D=N;

        else     {
                        D=N*nnn;
                        sprintf(wrt,"FRQ:%f AMP:5 ",D);
                        ibwrt(ud1,wrt,strlen(wrt));
                        ibwrt(ud1,"ot1 ",4);
                        ibwrt(ud,"DIG 3",5);
                        ibrd(ud,val,4096);
                        screen();
                        plotwaveform(16,1,val);
                        plotwaveform(15,2,val);
                        findmaxnitude2();
                                                              }
                                                                 }

                        setfillstyle(SOLID_FILL,0);
                        bar(0,0,650,550);
                        plot3L();
                        if(HIG==100)    Devid=8;
                        if(HIG==300)    Devid=25;
                        plot2();


}

 void findlinear2(void)
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
int i,n,m,ud1,cc;
float  N;
char aaa[25],bbb[15];

                        sprintf(aaa,"PLEASE ENTER FOR LINEAR");
                        sprintf(bbb,"AND SELEC T/D");
                        setcolor(15);
                        outtextxy(410,230,aaa);
                        outtextxy(420,250,bbb);
                        gets(aaa);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,0,650,550);
                        ud1=ibfind("DEVD");
                        ibclr(ud1);
                        N=atof(f);

      for(cc=0;cc<=10;cc++)      {

      if(cc==0)  {
                        N=N+MAX;
                        sprintf(wrt,"FRQ:%f AMP:5 ",N);
                        ibwrt(ud1,wrt,strlen(wrt));
                        ibwrt(ud1,"ot1 ",4);
                        ibwrt(ud,"DIG 3",5);
                        ibrd(ud,val,4096);
                        screen();
                        plotwaveform(16,1,val);
                        plotwaveform(15,2,val);
                        findmaxnitude2();

                                                          }
       else    {
                              N=N-MAX;
                              if(N==0)                {
                              N=MAX;
                              MAX=MAX/10;
                              N=N-MAX;
                                                      }
          if(MAX==1)    break;

                        sprintf(wrt,"FRQ:%f AMP:5 ",N);
                        ibwrt(ud1,wrt,strlen(wrt));
                        ibwrt(ud1,"ot1 ",4);
                        ibwrt(ud,"DIG 3",5);
                        ibrd(ud,val,4096);
                        screen();
                        plotwaveform(16,1,val);
                        plotwaveform(15,2,val);
                        findmaxnitude2();
                                                  }
                                                      }

                        setfillstyle(SOLID_FILL,0);
                        bar(0,0,650,550);
                        plot3H();
                        if(HIG==100)    Devid=5.1;
                        if(HIG==300)    Devid=15.3;
                        plot2();
}
void Write2(void)
{
   int n;
   float f;
   char str[20],yes[5],no[5];
   char c1[30];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        FILE   *fp;
                        sprintf(str,"YOU WANT TO SAVE ?");
                        sprintf(yes,"YES");
                        sprintf(no,"NO");
                        setcolor(11);
                        outtextxy(410,240,str);
                        setfillstyle(SOLID_FILL,15);
                        bar(410,260,530,290);
                        setcolor(0);
                        line(470,260,470,290);
                        outtextxy(430,275,yes);
                        outtextxy(490,275,no);
                        left=0;
  do{   mouseinformation(&right,&left,&row,&colum);
                    mouseshow();

if((410<=colum)&&(colum<=470)&&(260<=row)&&(row<=290)&&(left))  {
                        mousehind();
                        setfillstyle(SOLID_FILL,4);
                        bar(410,260,470,290);
                        setcolor(0);
                        outtextxy(430,275,yes);
                        delay(300);
                        setfillstyle(SOLID_FILL,0);
                        bar(410,220,540,300);
                        sprintf(c1,"FILE MANE TO WRITE");
                        setcolor(5);
                        outtextxy(430,275,c1);
                        gotoxy(60,20);
                        gets(file);
                        setfillstyle(SOLID_FILL,0);
                        bar(380,220,580,400);
                        if((fp=fopen(file,"w"))==NULL) {
                        printf("error in open file  \n ");
                        exit(1);        }
                        fwrite(gain,sizeof gain,1,fp);
                        m++;
                        if(ferror(fp))     {
                        printf(" error in write  \n ");
                        exit(1);         }
                        fclose(fp);


if((470<=colum)&&(colum<=530)&&(260<=row)&&(row<=290)&&(left))   {
                        mousehind();
                        setfillstyle(SOLID_FILL,4);
                        bar(470,260,530,290);
                        setcolor(0);
                        outtextxy(490,275,no);
                        delay(300);
                        setfillstyle(SOLID_FILL,0);
                        bar(380,220,580,400);
                        break;
                                                                  }

while(!((410<=colum)&&(colum<=530)&&(260<=row)&&(row<=290)&&(left))) ;
                        if(left)  left=0;
                        delay(300);

   }
void fre_res2(void)
{
  char str2[25],yes[5],no[5];
  char str3[15],str5[20];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        delay(1000);
                        sprintf(str2,"YOU WANT TO FIND SLOP?");
                        sprintf(yes,"YES");
                        sprintf(no,"NO");
                        sprintf(str3,"PLEASE ENTER");
                        sprintf(str5,"FOR REPORT SYSTEM");
                        setcolor(6);
                        outtextxy(385,240,str2);
                        setfillstyle(SOLID_FILL,15);
                        bar(410,260,530,290);
                        setcolor(0);
                        line(470,260,470,290);
                        outtextxy(430,275,yes);
                        outtextxy(490,275,no);
                        left=0;
  do{   mouseinformation(&right,&left,&row,&colum);
                        mouseshow();

if((410<=colum)&&(colum<=470)&&(260<=row)&&(row<=290)&&(left))  {
                        mousehind();
                        setfillstyle(SOLID_FILL,4);
                        bar(410,260,470,290);
                        setcolor(0);
                        outtextxy(430,275,yes);
                        delay(500);
                        setfillstyle(SOLID_FILL,0);
                        bar(385,220,580,350);
                        if(gain[4]>gain[25])          findlinear1();
                        if(gain[4]<gain[25])          findlinear2();
                setcolor(2);
                        outtextxy(450,80,str3);
                        outtextxy(430,90,str5);
                        gets(a);
                        setfillstyle(SOLID_FILL,0);
                        bar(0,0,600,100);
                        plot6();
                        gets(a);


if((470<=colum)&&(colum<=530)&&(260<=row)&&(row<=290)&&(left))   {
                        mousehind();
                        setfillstyle(SOLID_FILL,4);
                        bar(470,260,530,290);
                        setcolor(0);
                        outtextxy(490,275,no);
                        delay(500);
                        setfillstyle(SOLID_FILL,0);
                        bar(385,220,580,350);
                        break;


while(!((410<=colum)&&(colum<=530)&&(260<=row)&&(row<=290)&&(lef
                        if(left)  left=0;
                        delay(300);

  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. เอกสารประกอบการเรียน สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
2. การเขียนโปรแกรม Turbo C สำหรับวิศวกรรม อ. ธันวา สรีประมง
3. คู่มือการเขียนโปรแกรม ภาษา C ดร. วิทยา เรื่องพรวิสุทธ์
4. การเขียนโปรแกรม Turbo C อ. มนตรี พจนาถลาวัลย์
5. โครงงาน Application IEEE-488 คุณ สาร อาชา
6. National Instrument NI - 488.2 Softwear Referent for MS-DOS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้