สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การเขียนโปรแกรมควบคุมเครื่องมือวัดผ่านระบบบัส IEEE-488

CONTROL MEASURING EQUIPMENT USING IEEE-488 STANDARD BUS (GPIB)

T036946

โดย
นางสาวธนาภรณ์ รัตนเมธาวี
นางสาวนิษารัตน์ เทิดสิทธิกุล

เลขหมู่.....
เลขทะเบียน 36946
วัน, เดือน, ปี 29 ส.ค. 2543

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนโปรแกรมควบคุมเครื่องมือวัดผ่านระบบบัส IEEE-488

CONTROL MEASURING EQUIPMENTS USING IEEE – 488 STANDARD BUS (GPIB)

โดย

นางสาวธนาภรณ์ รัตนเมธาวี

นางสาวนิษารัตน์ เทิดสิทธิกุล

อาจารย์ที่ปรึกษา

รศ. ดร. มนัส สังวรศิลป์

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2542

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การเขียนโปรแกรมควบคุมเครื่องมือวัดผ่านระบบบัส IEEE-488

ผู้จัดทำ

1. นางสาวธนาภรณ์ รัตนเมธาวี รหัสประจำตัว 39014220
2. นางสาวนิษารัตน์ เทิดสิทธิกุล รหัสประจำตัว 39014271

.....อาจารย์ที่ปรึกษา

(รศ.ดร. มนัส สังวรศิลป์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนโปรแกรมควบคุมเครื่องมือวัดผ่านระบบบัส IEEE-488

CONTROL MEASURING EQUIPMENTS USING IEEE-488 STANDARD BUS (GIPB)

นางสาวธนาภรณ์ รัตนเมธาวี รหัสนักศึกษา 39014220

นางสาวนิษารัตน์ เทิดสิทธิกุล รหัสนักศึกษา 39014271

โครงงานได้รับการตรวจสอบแล้ว พร้อมที่จะทำการสอบได้

.....

(รศ.ดร. มนัส สังวรศิลป์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนโปรแกรมควบคุมเครื่องมือวัดผ่านระบบบัส IEEE – 488

นางสาวธนาภรณ์ รัตนเมธาวี
นางสาวนิษารัตน์ เทิดสิทธิกุล
รศ.ดร.มนัส สังวรศิลป์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2542

บทคัดย่อ

โครงงานนี้จะเป็นการพัฒนาโปรแกรมเพื่อควบคุมเครื่องมือวัดทางอิเล็กทรอนิกส์โดยผ่านระบบบัสมาตรฐานแบบ IEEE 488.2 หรือ GPIB (GENERAL PURPOSE INTERFACE BUS) ที่ใช้กันเป็นส่วนใหญ่ในงานอุตสาหกรรมทั่วไป โดยเขียนโปรแกรมด้วยภาษา Borland Delphi4 เพื่อควบคุมเครื่องมือวัดต่างๆ ดังนี้

1.) Programmable Function Generator รุ่น HM8130 ของ บริษัท HAMEG

2.) Programmable Power Supply รุ่น HM8142 ของบริษัท HAMEG

3.) Digital Oscilloscope 200 MHz รุ่น TDS 360 ของบริษัท TEKTRONIX

สำหรับใน Digital Oscilloscope นั้นได้มีการประยุกต์ในส่วนของ การนำสัญญาณต่าง ๆ มาวิเคราะห์หาสเปคตรัมความถี่ โดยวิธี FFT (Fast Fourier Transform) ซึ่งสามารถนำผลการวิเคราะห์นี้ไปใช้ประโยชน์ในด้านอื่น ๆ ต่อไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONTROL MEASURING EQUIPMENTS USING IEEE – 488 STANDARD BUS (GPIB)

Miss Tanaporn Rattanametawee

Miss Nisarat Therdsittikul

Assoc.Prof.Dr.Manas Sungworasil (Advisor)

1999

Abstract

This project purposes the program developed for controling GPIB bus equipment.The program as developed on Borland Delphi4 for controling some measuring equipment as followed,

1) Programmable Function Generator (HM8130 from HAMEG Company)
2) Programmable Power Supply (HM8142 from HAMAG Company)
3) Digital Oscilloscope 200 MHz (TDS360 from TEKTRONIX)

This equipments are the standard measuring equipment and generally used in industrial. In addition, we purpose an application of FFT spectrum analyser.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนำ

การควบคุมเครื่องมือวัดต่างๆ ในทางอุตสาหกรรมนั้น สามารถที่จะทำการควบคุมได้โดยผ่านทางสายข้อมูลหรือบัส (Bus) ซึ่งบัสที่ใช้ในการควบคุมนี้มีอยู่ด้วยกันหลายแบบ จึงมีระบบบัสที่ใช้เป็นมาตรฐานเดียวกัน และที่นิยมใช้ คือ ระบบบัสมาตรฐาน IEEE-488 หรือ GPIB (General Purpose Interfacing Bus) ซึ่งเกิดขึ้นโดยความร่วมมือของบริษัทผู้ผลิตเครื่องมือวัดในอเมริกา การเชื่อมต่อเครื่องมือวัดกับตัวควบคุมระบบของระบบมาตรฐาน GPIB นี้จะใช้ไมโครคอมพิวเตอร์เป็นตัวควบคุมเนื่องจากสะดวกและใช้งานง่าย ในการเชื่อมต่อระหว่างเครื่องมือวัดกับตัวควบคุมก็จะต้องมีการ์ดอินเตอร์เฟสเป็นตัวกลางส่งผ่านข้อมูล กล่าวคือ เราจะต้องควบคุมเครื่องมือวัดโดยการบันทึกโปรแกรมเพื่อส่งคำสั่งควบคุมให้ไมโครคอมพิวเตอร์ไปควบคุมเครื่องมือวัดต่างๆ โดยผ่านการ์ดอินเตอร์เฟส คณะผู้จัดทำหวังว่า รายงานฉบับนี้จะเป็นประโยชน์ในการศึกษาการเขียนโปรแกรม ภาษา Borland Delphi 4 ผ่านการ์ดอินเตอร์เฟส GPIB ไปควบคุมเครื่องมือวัด ตลอดจนเป็นแนวทางในการใช้โปรแกรมภาษาอื่นๆ เพื่อควบคุมเครื่องมือวัดได้ต่อไป

ผู้จัดทำ

ธนาภรณ์ รัตนเมธาวี

(นางสาวธนาภรณ์ รัตนเมธาวี)

นิษารัตน์ เทิดสิทธิกุล

(นางสาวนิษารัตน์ เทิดสิทธิกุล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
คำนำ	III
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
บทที่ 2 IEEE-488 (GPIB)	2
2.1 GPIB คืออะไร	2
2.1.1 การอ้างอิงตำแหน่งข้อมูล	2
2.1.2 ขีดจำกัดของการต่อ GPIB	2
2.2 โครงสร้างของ GPIB	3
2.3 รายละเอียดเกี่ยวกับ GPIB	3
2.4 ความหมายของสัญญาณต่างๆภายใน GPIB	4
2.5 การเชื่อมต่ออุปกรณ์ต่างๆในระบบ IEEE-488	5
บทที่ 3 การใช้งานโปรแกรม NI-488	7
3.1 โปรแกรม NI-488 (NI-488.2) สำหรับ window95	7
3.2 การใช้โปรแกรม NI-488.2M กับ 32 bit Delphi Application	8
บทที่ 4 โครงสร้างและฟังก์ชันคำสั่งในการทำงานของอุปกรณ์เครื่องมือวัด	13
บทที่ 5 การแปลงฟูริเยอร์ (Fourier Transform)	20
5.1 ฟังก์ชันต่อเนื่อง	20
5.2 การแปลงฟูริเยอร์เต็มหน่วย	20
5.3 หลักการเบื้องต้นของ FFT (Fast Fourier Transform)	22
5.3.1 FFT ชนิดลดทอนทางเวลา	22
5.3.2 FFT ชนิดลดทอนทางความถี่	23
บทที่ 6 การทดลองและการประยุกต์ใช้งาน	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1 การทดลองสั่งงาน Programmable Function Generator 24
6.2 การทดลองสั่งงาน Programmable Power Supply 26
6.3 การทดลองสั่งงาน Digital Oscilloscope 29
บทที่ 7 สรุปและวิจารณ์ผลการทดลอง 36
ภาคผนวก
กิตติกรรมประกาศ
เอกสารอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

		หน้า
รูปที่ 2.1	การเชื่อมต่อแบบเรียงต่อเนื่องกัน (Daisy Chain Configuration)	6
รูปที่ 2.2	การเชื่อมต่อแบบกระจาย (Star Configuration)	6
รูปที่ 3.1	การเชื่อมต่อระหว่าง NI – 488 กับ วินโดว์95	7
รูปที่ 4.1	แสดงอุปกรณ์ที่ต่ออยู่ในระบบที่ทำการทดลอง	19
รูปที่ 6.1	แสดงอุปกรณ์ที่ต่ออยู่ในระบบที่ทำการทดลอง	24
รูปที่ 6.2	แสดงสถานะปกติของหน้าจอฟังก์ชันเจนเนอเรเตอร์ที่ได้จำลองไว้บนหน้าต่างวินโดว์	25
รูปที่ 6.3	แสดงการเลือกโหมดตามที่กำหนดไว้	25
รูปที่ 6.4	แสดงการเลือกปุ่มออฟเซ็ท และ สถานะ	26
รูปที่ 6.5	แสดงสถานะปกติของหน้าจอเพาว์เวอร์ซัพพลายที่จำลองบนหน้าต่างวินโดว์	26
รูปที่ 6.6	แสดงการตั้งค่า V1 และ I1	27
รูปที่ 6.7	แสดงค่าแรงดันและกระแสที่ได้จากการเลือกปุ่ม Tracking	27
รูปที่ 6.8	แสดงลักษณะสัญญาณ Channel1 และ FFT ที่ปรากฏบนหน้าปัทม์ของสโคป	29
รูปที่ 6.9	แสดงลักษณะสัญญาณจาก Channel1 และ FFT ที่อ่านจากสโคป	30
รูปที่ 6.10	แสดงลักษณะสัญญาณจาก Channel1 และ FFT ที่แปลงเอง	30
รูปที่ 6.11	แสดงลักษณะของสัญญาณ Channel2 และ FFT ที่ปรากฏบนหน้าปัทม์	31
รูปที่ 6.12	แสดงลักษณะสัญญาณจาก Channel2 และ FFT ที่อ่านจากสโคป	31
รูปที่ 6.13	แสดงลักษณะสัญญาณจาก Channel2 และ FFT ที่แปลงเอง	32
รูปที่ 6.14	แสดงลักษณะสัญญาณที่วัดจากวงจรตัวอย่างและสัญญาณ CH1+CH2 ที่ได้จากสโคป	33
รูปที่ 6.15	แสดงลักษณะสัญญาณที่วัดจากวงจรตัวอย่างและสัญญาณ CH1+CH2 ที่ทดลองได้	33
รูปที่ 6.16	แสดงลักษณะสัญญาณที่วัดจากวงจรตัวอย่างและสัญญาณ CH1+CH2 ที่ได้จากสโคป	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.17 แสดงลักษณะสัญญาณที่ได้จากวงจรตัวอย่าง และสัญญาณ CH2-CH1 ที่ทดลองได้ 34
รูปที่ 6.18 แสดงลักษณะของสัญญาณ FFT ของสัญญาณเสียงที่ตำแหน่ง 2000 35
รูปที่ 6.19 แสดงลักษณะของสัญญาณ FFT ของสัญญาณเสียงที่ตำแหน่ง 1500 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

		หน้า
ตารางที่ 2.1	แสดงตำแหน่งบัสของ GPIB	2
ตารางที่ 6.1	แสดงการควบคุม Power Supply เมื่อมีการต่อโหลด โดยจ่ายแรงดัน 10 V. จำกัดกระแสที่ 0.05 A.	28
ตารางที่ 6.2	แสดงการควบคุม Power Supply เมื่อมีการต่อโหลด โดยจ่ายแรงดัน 20 V. จำกัดกระแสที่ 0.1 A.	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ปัจจุบัน ในโรงงานอุตสาหกรรมและหน่วยงานส่วนใหญ่ มักใช้คอมพิวเตอร์ในการควบคุมการทำงานของเครื่องมือและเครื่องจักร ซึ่งมีความสะดวก และมีความแม่นยำสูง ดังนั้นในงานด้านอิเล็กทรอนิกส์ จึงได้พัฒนาให้สามารถที่จะควบคุมเครื่องมือทางอิเล็กทรอนิกส์ผ่านทางคอมพิวเตอร์ได้เช่นกัน การที่จะทำให้คอมพิวเตอร์สามารถติดต่อกับเครื่องมืออิเล็กทรอนิกส์ได้นั้น สามารถทำได้โดยการเชื่อมต่อ (interface) คอมพิวเตอร์กับเครื่องมืออิเล็กทรอนิกส์ ผ่านทางระบบบัส และระบบบัสที่นิยมใช้กันเป็นมาตรฐาน คือ ระบบ IEEE-488 (GPIB : GENERAL PURPOSE INTERFACE BUS)

โดยปกติการ์ด GPIB นี้ สามารถติดต่อกับเครื่องมือวัดได้ โดยผ่านทางดอส (DOS)ซึ่งไม่สะดวกในการใช้งาน เนื่องจาก มีการใช้คำสั่งที่ซับซ้อนและยุ่งยาก จึงได้มีการพัฒนาให้สามารถประยุกต์ใช้งานได้สะดวกยิ่งขึ้น โดยนำมาใช้งานบนหน้าต่างวินโดว์ (Window) ซึ่งมีฟังก์ชั่นที่ครอบคลุมการใช้งานได้มากกว่าดอส และได้ใช้โปรแกรมภาษา Borland Delphi 4 ออกแบบเขียนโปรแกรมควบคุมเครื่องมือวัด ซึ่งโปรแกรมภาษา Borland Delphi 4 นี้ ประกอบด้วย 2 ส่วน คือ ส่วนของโปรแกรม (Source code) และส่วนของฟอร์ม (Form) ซึ่งเป็นส่วนที่จำลองหน้าจอของเครื่องมือวัด ในการใช้งานสามารถควบคุมการทำงานของเครื่องมือวัดผ่านทางส่วนของฟอร์มนี้ได้เลย ทำให้สะดวกและง่ายต่อการใช้งาน สำหรับโครงงานนี้ได้ทำการควบคุมเครื่องมือวัดทางอิเล็กทรอนิกส์ 3 ชนิด คือ Programmable Function Generator, Programmable Power Supply และ Digital Oscilloscope ซึ่งเป็นอุปกรณ์ที่ใช้งานโดยทั่วไปในงานด้านอิเล็กทรอนิกส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

IEEE-488 (GPIB)

2.1 GPIB คืออะไร

GPIB (General Purpose Interface Bus) หรืออีกอย่างหนึ่งว่า IEEE Standard 488 เป็นสายบัส (Bus) ที่เชื่อมระหว่างคอมพิวเตอร์ กับอุปกรณ์ภายนอก เช่น Power Supply (เครื่องกำเนิดไฟฟ้ากระแสตรง) , Multimeter (เครื่องวัดแรงดัน,วัดกระแส,ความต้านทาน) ,Function Generator (เครื่องกำเนิดกระแสสลับ) ซึ่งบัสที่ใช้ต่อเป็นบัสแบบขนาน (parallel) และใช้การส่งแบบขนาน 8 บิท (8-bit Communications) ซึ่งมีความเร็วในการส่งข้อมูล เท่ากับ 1 Mbytes/Sec ทั้งนี้และทั้งนั้นก็ขึ้นอยู่กับความเร็วของ ไมโครโปรเซสเซอร์ (Microprocessor) และอื่นๆ

2.1.1 การอ้างอิงตำแหน่งข้อมูล GPIB (GPIB Addressing)

อุปกรณ์ (Divice) หรือ บอร์ด (Board) จะกำหนดตำแหน่ง GPIB (GPIB Addressing) อย่างแน่นอน และตำแหน่ง GPIB ประกอบไปด้วย

1. ตำแหน่งปฐมภูมิ (Primary Address) จะมีค่าระหว่าง 0-30 ซึ่งทำหน้าที่ อ้างอิงตำแหน่งของข้อมูล ในการป้อน หรือรับข้อมูล

2. ตำแหน่งทุติยภูมิ (Secondary Address) จะมีค่าระหว่าง 60-7E(ฐาน 16) เมื่อตำแหน่งปฐมภูมิถูกเรียกใช้ แล้วตัวควบคุม หรือคอมพิวเตอร์ จะส่งตำแหน่งปฐมภูมิของการเขียนหรือ อ่านข้อมูล แล้วตามด้วยตำแหน่งทุติยภูมิ ของอุปกรณ์

ตารางที่ 2.1 ตำแหน่งบัสของ GPIB

Bit Position	7	6	5	4	3	2	1
ความหมาย	0	TA	LA	ตำแหน่งปฐมภูมิของ GPIB(0-30)			

2.1.2 ข้อจำกัดของการต่อ GPIB

1. อุปกรณ์แต่ละตัวจะอยู่ห่างกันได้ อย่างมากที่สุดคือ 4 เมตร ในการใช้สายบัส GPIB ตัวเดียวกัน

2. สายเคเบิล (bus) ที่ต่อกับคอมพิวเตอร์ยาวได้มากที่สุดเท่ากับ 20 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สายบัส GPIB หนึ่งสายใช้กับอุปกรณ์ ได้เท่ากับ 15 เครื่อง และอุปกรณ์จะต้องอยู่ในสถานะเปิดเครื่อง อยู่อย่างน้อย 2-3 เครื่อง

และถ้าต้องการให้อุปกรณ์ทำงานได้เร็ว ต้องมีข้อแม้เพิ่มดังนี้

1. อุปกรณ์ทุกตัวต้องอยู่ในสถานะเปิดเครื่องทั้งหมด
2. สายเคเบิลควรจะมีขนาดที่สั้นที่สุดเท่าที่จะทำได้
3. ควรต่ออุปกรณ์ที่มีความต้านทาน (Load) ที่เท่ากัน ต่อ 1 เมตรของสายเคเบิล

2.2 โครงสร้างของ GPIB

ในระบบพื้นฐานของ GPIB จะประกอบด้วยอุปกรณ์ คือ ผู้ส่ง(Talker) , ผู้รับ(Listener) และผู้ควบคุม (Controller)

- Talker ทำหน้าที่ส่งข้อมูล โดยในระบบสามารถมี Talker ได้หลายตัว แต่จะมีเพียงตัวเดียวเท่านั้นที่กำลังทำงานอยู่
- Listener ทำหน้าที่เป็นตัวรับข้อมูล โดยในระบบเดียวกันสามารถมี Listener ได้หลายตัวเช่นเดียวกัน แต่ Listener สามารถทำงานได้ครั้งละหลายๆตัวได้
- Controller ทำหน้าที่ควบคุมอุปกรณ์ต่างๆในระบบ โดยจะกำหนดให้ Talker ทำการส่งข้อมูล หรือกำหนดให้ Listener ทำการรับข้อมูล

อุปกรณ์ที่มีนั้น สามารถแบ่งตามหน้าที่ได้ดังนี้

1. ทำหน้าที่เป็น Talker เท่านั้น เช่น เครื่องมือวัด เป็นต้น
2. ทำหน้าที่เป็น Listener เท่านั้น เช่น เครื่องพิมพ์(Printer) , เครื่องบันทึก(Recorder) เป็นต้น
3. ทำหน้าที่เป็นทั้ง Talker และ Listener เช่น คอมพิวเตอร์ , เครื่องมือวัดที่สามารถควบคุมได้จากภายนอก เป็นต้น
4. ทำหน้าที่เป็น Talker Listener และ Controller ในตัวเดียวกัน เช่น คอมพิวเตอร์ที่ทำหน้าที่ควบคุมระบบ

2.3 รายละเอียดเกี่ยวกับ GPIB

GPIB นั้น เป็นสายสัญญาณแบบ 24 เส้น ขนานกันและมีขั้วอยู่ทางตอนปลายทั้งสองของสายเพื่อต่อกับอุปกรณ์ หรือต่อกันเพื่อให้สายสัญญาณมีความยาวเพิ่มขึ้น จำนวนสายสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

24 เส้นนี้ มีเพียง 16 เส้นเท่านั้นที่ทำหน้าที่นำสัญญาณ ส่วนที่เหลืออีก 8 เส้น ทำหน้าที่กราวน์ (Ground) ชีลด์(Shield) โดยจำนวนสายที่นำสัญญาณ 16 เส้น ยังแบ่งได้เป็น 3 ประเภท

2.3.1 บัสข้อมูล (Data Bus) จำนวน 8 สาย

DIO1 – DIO8

2.3.2 สายสัญญาณควบคุม (Controller) จำนวน 5 สาย คือ

IFC (Interfacing Clear)

ATN (Attenuation)

SRQ (Service Request)

REN (Remote Enable)

EOI (End Or Identify)

2.3.3 สายแฮนด์เช็ค (Hand Check) จำนวน 3 สาย คือ

DAV (Data Valid)

NRFD (Not Ready For Data)

NDAC (Not Data Accepted)

2.4 ความหมายของสัญญาณต่างๆภายใน GPIB

ดังที่กล่าวมาแล้วว่าสายสัญญาณต่างๆ ใน GPIB ได้แบ่งออกเป็น 3 กลุ่ม ในหัวข้อนี้จะอธิบายได้ดังนี้

2.4.1 กลุ่มสัญญาณข้อมูล

DIO1 – DIO8 (Data Input / Output) สัญญาณทั้ง 8 เส้นนี้ ทำหน้าที่เป็นทางผ่านข้อมูลของระบบ

2.4.2 กลุ่มสัญญาณควบคุมการเชื่อมต่อ (Interfacing)

- IFC (Interface Clear) เป็นสัญญาณรีเซ็ท หรือเคลียร์ระบบ กำเนิดได้โดยตัวควบคุมเท่านั้น

- ATN (Attenuation) เป็นสัญญาณที่ถูกส่งโดยอุปกรณ์ ที่เป็นตัวควบคุมเช่นเดียวกันใช้ในการให้อุปกรณ์ทุกตัวในระบบเตรียมพร้อมเพื่อรอคำสั่งจากตัวควบคุม

- SRQ (Service Request) เป็นสัญญาณที่ถูกส่งจากอุปกรณ์ต่างๆเพื่อเป็นการบอกแก่ระบบว่าขณะนี้อุปกรณ์ดังกล่าวต้องการติดต่อจากตัวควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- REN (Remote Enable) สัญญาณนี้เป็นสัญญาณที่ถูกส่งจากตัวควบคุมเพียงตัวเดียวเท่านั้นเพื่อใช้สั่งให้อุปกรณ์ต่างๆเปลี่ยนจากโหมดใช้งานปกติมาเป็นการควบคุมโดยตัวควบคุมแทน

- EOI (End Or Identify) เป็นสัญญาณที่ถูกส่งได้โดยอุปกรณ์ที่เป็นตัวควบคุม หรืออุปกรณ์ที่เป็นตัวส่งก็ได้ ใช้สำหรับแสดงว่าข่าวสารที่ส่งเป็นชุดนั้นได้เสร็จสิ้นลงแล้ว

2.4.3 กลุ่มสัญญาณการรับส่งข้อมูล

- DAV (Data Valid) เมื่อสัญญาณนี้ถูกดึงเป็นลอจิก "low" โดยอุปกรณ์ที่เป็นตัวส่ง เป็นการแสดงว่าขณะนี้ตัวส่งได้ทำการส่งข้อมูลลงไปที่สายข้อมูลเรียบร้อยแล้ว

- NRFD (Nor Read For Data) เมื่อสัญญาณนี้มีลอจิกเป็น "low" จะเป็นการแสดงว่าขณะนี้ระบบบัสยังไม่พร้อมที่จะรับข้อมูล เนื่องจากอุปกรณ์ในระบบยังพร้อมไม่หมดทุกตัวซึ่งสัญญาณเส้นนี้จะไม่เป็น "HI" จนอุปกรณ์ทุกตัวให้ลอจิกเป็น "HI" ครบถ้วนแล้วสัญญาณนี้มีประโยชน์ในกรณีที่อุปกรณ์ระบบมีความเร็วที่แตกต่างกัน

- NDAC (Not Data Accepted) สัญญาณที่ถูกควบคุมโดยอุปกรณ์ที่เป็นตัวรับ โดยสัญญาณนี้จะมีลอจิกเป็น "HI" ในขณะอุปกรณ์ที่เป็นตัวกำลังเก็บข้อมูลจากสายข้อมูลและจะเป็น "HI" เมื่ออุปกรณ์นั้นได้ทำการอ่านเสร็จเรียบร้อยแล้ว โดยสัญญาณลอจิกที่ใช้ใน(DI1 - DI8) ของ IEEE-488 มีลักษณะเป็นคอมพลีเมนต์ทั้งหมดคือ "1" เท่ากับ (LOW) และ "0" เท่ากับ (HI)

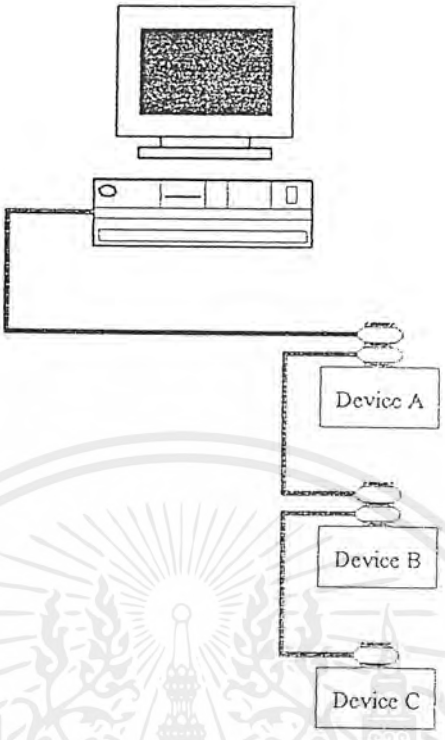
2.5 การเชื่อมต่ออุปกรณ์ต่างๆในระบบ IEEE-488 BUS

สำหรับการเชื่อมต่อระหว่างอุปกรณ์ต่างๆในระบบ IEEE-488 นั้น มีอยู่ 2 วิธี คือ

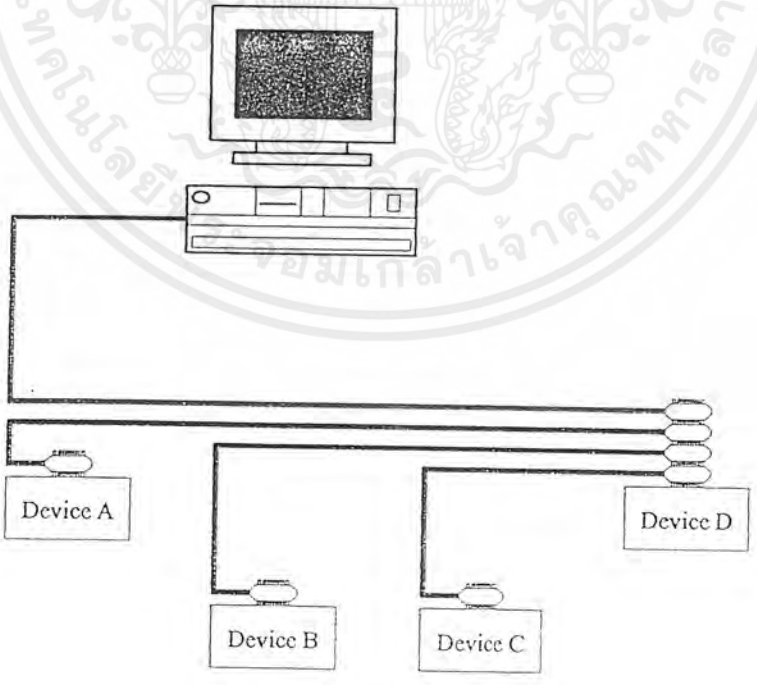
1. การเชื่อมต่อแบบเรียงต่อเนื่องกัน (Daisy Chain Configuration)
2. การเชื่อมต่อแบบกระจาย (Star Configuration)

ซึ่งได้แสดงลักษณะการเชื่อมต่อทั้ง 2 แบบไว้เพื่อดูความแตกต่างให้ชัดเจนขึ้น ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 การเชื่อมต่อแบบเรียงต่อเนื่องกัน(Daisy Chain Configuration)



รูปที่ 2.4 การเชื่อมต่อแบบกระจาย (Star Configuration)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การใช้งานโปรแกรม NI-488

3.1 โปรแกรม NI-488 (NI-488.2) สำหรับ วินโดว์ 95 (Window 95)

โปรแกรมที่ใช้ในการติดต่อระหว่างวินโดว์ 95 กับ NI-488 (หรือ NI-488.2) และโปรแกรมที่ช่วยในการติดต่อ มีอยู่ด้วยกันหลายโปรแกรม ดังนี้

1. ไฟล์ข้อมูล readme.txt ซึ่งประกอบไปด้วยข้อมูลข่าวสารที่เกี่ยวกับ NI-488 (NI-488.2) และข้อมูลอื่นๆด้วย

2. Win32 dynamic link library . gpib-32.dll เป็นโปรแกรมที่ใช้ในการติดต่อระหว่างโปรแกรมวินโดว์กับโปรแกรม NI-488 . NI-488.2

3. Win32 Interactive Control ซึ่งใช้ในการติดต่อระหว่างอุปกรณ์ GPIB โดยใช้ NI-488

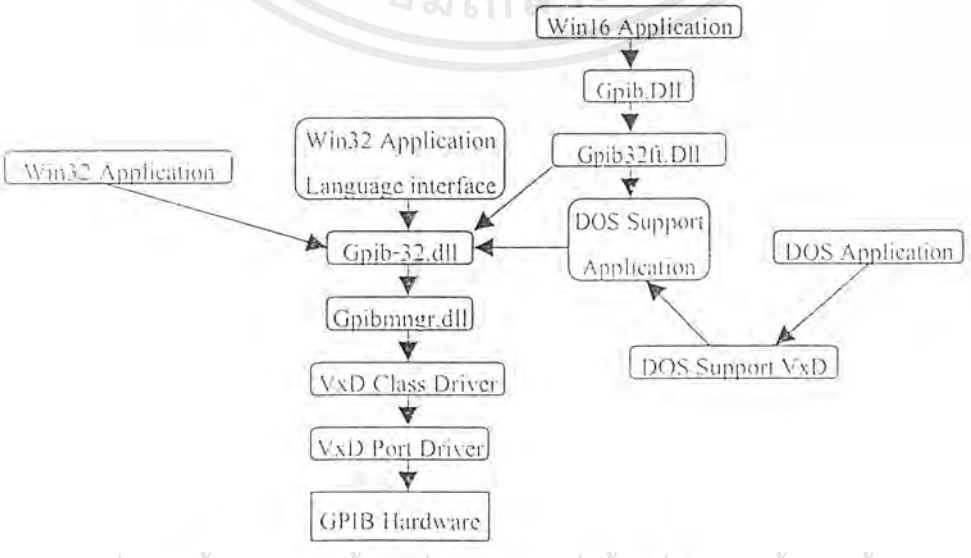
4. GPIB Spy เป็นโปรแกรมประยุกต์ที่ใช้กับโปรแกรมการแสดงผล

5. 16-bit Windows dynamic link library . gpib.dll

6. 32-bit Windows dynamic link librart . gpib32ft.dll เป็นโปรแกรมที่ช่วย gpib.dll เพื่อที่จะเป็นโปรแกรม gpib-32.dll เพื่อที่จะใช้กับโปรแกรม gpib-32.dll

7. Virtual Device Driver (VxD) . gpibdosk.vxd เป็นโปรแกรมที่ใช้ติดต่อระหว่างดอส (DOS) กับ NI-488 แต่จำเป็นที่จะต้องมีโปรแกรมช่วยนั้นก็คือ gpibdos.exe เพื่อที่จะเป็นโปรแกรม gpib-32.dll เพื่อที่จะใช้กับโปรแกรม gpib-32.dll

3.1.1 การเชื่อมต่อระหว่าง NI-488(NI-488.2) กับวินโดว์ 95



รูปที่ 3.1 การเชื่อมต่อระหว่าง NI-488 กับวินโดว์95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้โปรแกรม NI-488 กับวินโดว์ 95 มีข้อปฏิบัติดังนี้

โปรแกรม Win32 Application (วินโดว์ 95) จะใช้กับโปรแกรม gpib-32.dll ได้เลย

โปรแกรม Win16 Application (วินโดว์ 3.1) จะต้องมีโปรแกรม gpib.dll และโปรแกรม gpib.dll เพื่อที่จะติดต่อกับ gpib-32.dll

โปรแกรม DOS Application จะต้องมีโปรแกรม DOS Support VxD และโปรแกรม DOS Support Application

3.1.2 เปรียบเทียบ NI-488 กับ NI-488.2

โดยปกติแล้วการส่ง , รับข้อมูล หรือการควบคุมการทำงานต่างๆของอุปกรณ์ เราสามารถที่จะใช้โปรแกรม NI-488(NI-488 Functions) ก็เพียงพอแล้วสำหรับการทำงาน แต่เราใช้โปรแกรม NI-488(NI-488 Functions) ก็ต่อเมื่อเราจะควบคุมการทำงานของอุปกรณ์หลายๆชิ้นและหลายบอร์ดอินเตอร์เฟสได้ ซึ่งใน NI-488 สามารถใช้กับอุปกรณ์ได้หลายๆเครื่อง (ไม่เกิน 15 เครื่อง) เหมือนกัน แต่ต้องใช้ บอร์ดอินเตอร์เฟสเพียงบอร์ดเดียว

หมายเหตุ ในชิ้นงานนี้ใช้บอร์ดอินเตอร์เฟสเพียงบอร์ดเดียว ดังนั้นจึงใช้คำสั่ง NI-488.2 ไม่ใช้คำสั่งแบบ NI-488

3.2 การใช้โปรแกรม NI-488.2M กับ 32-bit Delphi Application

โปรแกรม Borland Delphi เป็นโปรแกรมที่สามารถใช้สร้าง Application บน Windows ได้อย่างง่ายและรวดเร็ว โดยถูกพัฒนามาจากภาษา Pascal Delphi สามารถที่จะเรียกใช้ Dynamic Link Librariea(DLLs) ซึ่งทำให้เป็นเรื่องง่ายที่เราจะนำเอาโปรแกรม Delphi มาใช้ติดต่อกับอุปกรณ์ GPIB โดยผ่านโปรแกรม NI-488.2M ซึ่งต่อไปนี้จะเป็นวิธีการเรียกใช้ NI-488.2M DLL โดยโปรแกรม Delphi

การเขียนโปรแกรมบน NI-488 หรือ NI-488.2

NI-488 และ NI-488.2 เป็น Software Driver ของการ์ด GPIB-PCII/IIA ซึ่งสามารถนำมาใช้งานกับ Delphi ได้ ทั้ง 2 โปรแกรมแตกต่างกันเพียงแต่รูปแบบของ function และ procedure ที่มีให้เรียกใช้งานอยู่ในไฟล์ .DLL ดังจะได้แสดงในตัวอย่างการใช้งานของโปรแกรมทั้งสองในหัวข้อต่อไป

32-bit Delphi Application สามารถที่จะเชื่อมต่อกับ NI-488.2M dynamics link library (gpib-32.dll) โดยการเรียกใช้ gpib-32.dll ดังนี้

3.2.1 การสร้างตัวแปรที่ใช้ชี้ตำแหน่งของ function และใช้บ่งบอกถึงสถานะ (Export Pointers to Status Variables and Function Name)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

32-bit GPIB DLL จะให้ค่าชี้ใช้บอกตำแหน่งของ NI-488.2 function subroutines และตัวแปรแบบโคบอล การชี้บอกตำแหน่งของตัวแปรแบบโคบอลของ NI-488.2 (ibsta ,iberr ,ibcntl) สามารถที่จะทำได้โดยการสร้างตัวแปรขึ้นมารับค่า

```
var user_ibsta ;
    user_iberr ;
    user_ibcnt : integer ;
```

การเรียกใช้ฟังก์ชันอย่างเช่น ibbna และ ibwrt หรือฟังก์ชันทั้งหมดของ NI-488.2 และ subroutine ทั้งหมด จะถูกเรียกใช้โดยการนำค่าออกมาจาก 32 bit GPIB DLL ซึ่งก็หมายความว่า เราสามารถที่จะเรียกใช้งานฟังก์ชันเหล่านี้ในการควบคุมอุปกรณ์ โดยเราจำเป็นจะต้องรู้ตำแหน่งของฟังก์ชันที่เราจะเรียกใช้ ทำได้โดยการใช้คำสั่ง GetProcAddress ทำการผ่านชื่อของฟังก์ชันมายังตัวแปรและเราจะใช้ตัวแปรนี้เป็นตัวรับค่าต่างๆที่เราจะใช้ควบคุมอุปกรณ์ตามการทำงานของแต่ละฟังก์ชัน คำอธิบายการทำงานของแต่ละฟังก์ชันสามารถดูเพิ่มได้จากคู่มือ NI-488.2M Function Reference Manual For Win32 (down load ได้ที่ ftp.natinst.support.com)

ฟังก์ชันต่างๆที่เราใช้งานจำเป็นต้องมีการป้อนค่าต่างๆให้แก่ฟังก์ชันเหล่านี้ เช่น ฟังก์ชัน ibbna ต้องป้อนชื่อของการ์ด (การตั้งชื่อการ์ดดูได้จากคู่มือการติดตั้งชื่อ Getting Started with Your GPIB-PCII/IIA and The GPIB Software For Window95) ฟังก์ชัน ibfind ต้องการชื่อการ์ดและอุปกรณ์ และฟังก์ชัน ibrd กับ ibwrt ต้องใช้ชื่อของไฟล์ แต่จากการที่ Windows NT สามารถเข้าใจได้ทั้งแบบ normal (8 บิต) และ unicode (16 บิต) character ซึ่ง gpib-32.dll จะส่งค่าออกมาทั้ง 2 แบบ แต่ใน Windows95 เราไม่สามารถใช้ character ขนาด 16 บิตได้ ดังนั้นสำหรับ Windows95 เราจะใช้ฟังก์ชัน 8 บิตASCII เช่น ibbnaA , ibrdfA , ibwrtfA

ในการเพิ่มตัวชี้ถึงตัวแปรที่บอกสถานะของอุปกรณ์และการโหลด gpib-32.dll เราต้องทำการกำหนดชนิดข้อมูลสำหรับฟังก์ชันที่เราต้องการเรียกใช้ ชนิดของข้อมูลของแต่ละฟังก์ชันที่ส่งผ่านโดย gpib-32.dll สามารถดูได้จากคู่มือชุดเดียวกับที่อธิบายการทำงานของแต่ละฟังก์ชัน และสามารถหาข้อมูลเพิ่มเติมได้จาก Win32 SDK

3.2.2 รายการต่างๆที่ต้องกำหนดใน Application (Items to Include)

ใน Win32 Delphi Application ขั้นตอนแรกที่เราต้องทำคือ ใช้ type เพื่อกำหนดตัวแปรแบบโคบอลสำหรับ GPIB ตัวแปร GPIB นี้ จะใช้กำหนดฟังก์ชันที่จะส่งค่ากลับมาในรูปของจำนวนเต็ม ตามด้วยการเรียกใช้แบบ directive , stdcall ตัวอย่างข้างล่างนี้เป็นวิธีการใช้ type กำหนดตัวแปรแบบโคบอลของ GPIB

```
Tibsta  = function : integer : stdcall ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Tinerr = function : integer : stdcall ;

Tibcntl = function : integer : stdcall ;

ในส่วน var ของโปรแกรม AddrIIbXXX จะถูกกำหนดโดยใช้ตำแหน่งที่ได้จากตัวแปรแบบโคบอล และถูกชี้โดย PibXXX ตามตัวอย่างดังต่อไปนี้

AddrIbsta : Tibsta ;

AddrIberr : Tiberr ;

AddrIbcntl : Tincntl ;

Pibsta : integer ;

Piberr : integer ;

Pibcntl : Longint ;

NI-488 จะถูกเรียกใช้โดยถูกกำหนดให้อยู่ในรูปของฟังก์ชันและNI-488.2 จะถูกเรียกใช้โดยกำหนดให้อยู่ในรูปของ Procedure ในส่วนที่ใช้กำหนดการเรียกใช้งานจะต้องตามด้วยรายการของตัวแปรและพารามิเตอร์ต่างๆ ของฟังก์ชันที่เรียกใช้ โดยที่ส่วนประกอบสุดท้ายจะเป็นการเรียกแบบ directive (stdcall) การเพิ่มส่วน TYPE เพื่อเพิ่มรูปแบบข้อมูลที่จะใช้กับฟังก์ชันของ GPIB ใน Pascal ยูนิต สามารถดูรูปแบบได้ตามตัวอย่างที่ได้ให้ไว้ในส่วนท้ายของบทนี้

ต่อไปนี้เป็นตัวอย่างของการกำหนด ฟังก์ชัน ibdev ในส่วนของ Type

Tibdev = function (ud : integer ;

pad : integer ;

sad : integer ;

tmo : integer ;

eot : integer ;

eos : integer) : integer ; stdcall ;

ตัวอย่างการกำหนด SendIFC โพรซีเดอร์ในส่วนของ Type

TSendIFC = Procedure (boardID : integer) ; stdcall ;

ตัวอย่างการเรียกใช้ โดยกำหนดในส่วนของการสร้างตัวแปร Var

ibdev : Tibdev ;

SendIFC : TSendIFC ;

นอกจากนี้เรายังต้องกำหนดค่า handle สำหรับ GPIB ไลบรารีไว้ในส่วนของ Var ตามตัวอย่างต่อไปนี้

Gpib32Lib : Thandle ;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของ Const ซึ่งเป็นส่วนที่ใช้ประกาศค่าคงที่ในโปรแกรมที่เขียนขึ้น เราสามารถที่จะกำหนดค่าที่ใช้เป็นตัวบอกสถานะของระบบและการเกิด Error ขึ้นในการติดต่อ ซึ่งค่าต่างๆ นี้จะมีค่าตามที่ถูกกำหนดไว้ในคู่มืออ้างอิงของNI-488 ค่าคงที่เหล่านี้สามารถใช้ในโปรแกรมของเราเพื่อช่วยในการตรวจสอบ โดยนำมาเปรียบเทียบกับค่าที่อยู่ใน ibsta และ iberr ซึ่งค่าคงที่ต่างๆ นี้ได้แสดงไว้ในภาคผนวกท้ายเล่ม

3.2.3 การโหลด GPIB-32.DLL Library

ในโปรแกรม Win32 ของเรา เราต้องโหลด GPIB Library (gpib-32.dll) Code ที่แสดงต่อไปนี้ จะแสดงวิธีการโดยการเรียกใช้ฟังก์ชัน LoadLibrary และตามด้วยการตรวจสอบข้อผิดพลาด

```
GPIB32Lib  := LoadLibrary ('GPIB-32.DLL);
If Gpib32Lib = 0 Then
        Begin
                (*Report an error here.*)
        end;
```

3.2.4 การหาตำแหน่ง (Getting the Address)

หลังจากที่มีการกำหนดค่าคงที่และมีการสร้างรูปแบบข้อมูลในส่วนของType แล้วต่อไปก็คือการหาตำแหน่งของตัวแปรโคบอล ที่ใช้บอกสถานะและฟังก์ชันที่เราจำเป็นต้องใช้ในโปรแกรมของเรา Code ที่จะแสดงต่อไปนี้เป็นตัวอย่างที่แสดงวิธีการเอาตำแหน่งของตัวแปรมาใช้

```
@AddrIbsta  := GetProcAddress (Gpib32Lib, 'user_ibsta');
@AddrIberr  := GetProcAddress (Gpib32Lib, 'user_iberr');
@AddrIbcntl := GetProcAddress (Gpib32Lib, 'user_ibcnt');

Pibsta  := @AddrIbsta;
Piberr  := @AddrIberr;
Pibcntl := @AddrIbcntl;

(*NI-488 Calls)
@ibdev      := GetProcAddress(Gpib32Lib, 'Gpib32Lib, 'ibdev');
@ibonl      := GetProcAddress(Gpib32Lib, 'Gpib32Lib, 'iboml');
(*NI-488.2 Calls)
@SendIFC    := GetProcAddress(Gpib32Lib, 'Gpib32Lib, 'SendIFC');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าฟังก์ชัน GetProcAddress เกิดข้อผิดพลาด มันจะส่งค่า NIL กลับมาที่พอยเตอร์(Pointer) Code ต่อไปนี้จะแสดงวิธีการตรวจสอบว่า ฟังก์ชัน GetProcAddress เกิดข้อผิดพลาดหรือไม่

```
If (@Addrlbsta    = NIL) Or
   (@Addrlberr    = NIL) Or
   (@Addrlbscntl  = NIL) Or
   (@ibdev        = NIL) Or
   (@ibonl        = NIL) Then
      begin
            (*Report an Error.*)
            FreeLibrary (Gpib32Lib); (*Free the library.*)
      end;
```

3.2.5 การติดต่อกับอุปกรณ์ GPIB และตัวแปร (Accessing the GPIB Calls and Variables)

ชุดคำสั่งต่อไปนี้ จะแสดงวิธีการเรียกใช้ function และ procedure รวมทั้งตัวแปรแสดงสถานะภายในโปรแกรมที่เราเขียนขึ้น

```
dev := ibdev (0,1,0,13,1,0);
If (Pibsta^ and  ERR) <> 0 Then
      Gpiberr('ibdev Error');

SendIFC (0);
If (Pibsta^ and ERR) <> 0 Then
      Gpiberr ('SendIFC Eror');
```

<u>การยกเลิกการใช้ Library (Freeing the Library)</u>

ก่อนที่จะออกจากโปรแกรมเราจะต้องทำการหารยกเลิกการติดต่อกับgpib-32.dll ก่อน เพื่อให้โปรแกรมอื่นสามารถเรียกใช้ gpib-32.dll ได้ ด้วยชุดคำสั่งต่อไปนี้

```
FreeLibrary (Gpib32Lib);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
โครงสร้างและฟังก์ชันคำสั่งในการทำงานของอุปกรณ์เครื่องมือวัด

โครงสร้างของระบบเครื่องมือวัดที่ทำการศึกษาและฟังก์ชันคำสั่งในการทำงานของอุปกรณ์ที่ต่ออยู่บนระบบ โดยระบบเครื่องมือวัดนี้ประกอบด้วยตัวควบคุมระบบซึ่งใช้ไมโครคอมพิวเตอร์ทำหน้าที่ดังกล่าว และอุปกรณ์เครื่องมือพื้นฐานทางด้านอิเล็กทรอนิกส์ 3 อุปกรณ์ คือ

1. Power Supply
2. Function Generator
3. Digital Oscilloscope

สำหรับการเชื่อมต่ออุปกรณ์ในระบบนั้น เป็นส่วนประกอบที่สำคัญอย่างหนึ่งที่จะช่วยให้ระบบมีประสิทธิภาพในการทำงานที่ดี

อุปกรณ์ภายในระบบ

1. ตัวควบคุมระบบ

ตัวควบคุมระบบตามมาตรฐาน IEEE-488 เป็นหัวใจสำคัญของระบบ เนื่องจากจะเป็นตัวควบคุมการติดต่อสื่อสารระหว่างตัวรับ,ตัวส่งและตัวควบคุมได้อย่างถูกต้องและรวดเร็วซึ่งในระบบที่ทำการศึกษานั้น ได้ใช้ไมโครคอมพิวเตอร์เป็นตัวควบคุม ซึ่งมีข้อดีคือสะดวกในการอินเตอร์เฟส เนื่องจากมีสล็อตไว้ให้เลือกใช้อยู่แล้ว การพัฒนาโปรแกรมทำได้สะดวกรวดเร็ว และยังสามารถบันทึกข้อมูลที่ได้จากการวัดในระบบเก็บเอาไว้ได้ เพื่อที่จะนำค่าที่บันทึกไว้ไปใช้งานต่อไป

2. อุปกรณ์ที่ต่ออยู่ในระบบ

ระบบเครื่องมือวัดตามมาตรฐาน IEEE-488 ที่ทำการศึกษานั้น นอกจากจะมีตัวควบคุมระบบแล้ว ยังมีอุปกรณ์ที่ต่ออยู่ในระบบซึ่งได้แก่อุปกรณ์ที่เป็นเครื่องมือพื้นฐานทางอิเล็กทรอนิกส์ที่ใช้สำหรับการทดลองทางอิเล็กทรอนิกส์ โดยอุปกรณ์เหล่านี้นอกจากจะสามารถควบคุมได้ผ่านทางหน้าปัทม์แล้ว ยังสามารถควบคุมโดยผ่านทางระบบบัสตามมาตรฐาน IEEE-488 หรือ GPIB ได้ด้วย ซึ่งในระบบจะประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. **Programmable Power Supply** ของบริษัท HAMEG รุ่น HM8142 ทำหน้าที่เป็นแหล่งจ่ายไฟฟ้ากระแสตรง เพื่อใช้เป็นไฟเลี้ยงให้แก่วงจรที่จะทำการทดลอง โดยมีแหล่งจ่ายไฟอยู่ 2 ตัว และมีคำสั่งควบคุมฟังก์ชันในการทำงานผ่านทางระบบบัสแบบ IEEE-488 คือ

1.1 คำสั่งเกี่ยวกับการรีโมท

- RM1 และ RM0
- LK1 และ LK0
- MX1 และ MX0

1.2 คำสั่งเกี่ยวกับศักดาไฟฟ้า

- SU1 และ SU2
- SI1 และ SI2
- MU1 และ MU2
- TRU

1.3 คำสั่งเกี่ยวกับกระแสไฟฟ้า

- RU1 และ RU2
- RI1 และ RI2
- MI1 และ MI2
- TRI

1.4 คำสั่งเกี่ยวกับสถานะและตัวอุปกรณ์

- STA
- SR0 และ SR1
- VER
- ID?
- OP1 และ OP0

2. **Programmable Function Generator** ของบริษัท HAMEG รุ่น HM 8130 ทำหน้าที่สร้างสัญญาณรูปแบบต่างๆ ป้อนให้แก่วงจรที่ทำการทดลอง และมีคำสั่งควบคุมฟังก์ชันในการทำงานผ่านระบบบัสแบบ IEEE-488 โดยสามารถแบ่งได้เป็นกลุ่มคำสั่ง คือ

<u>คำสั่งที่ไม่ต้องมีข้อมูลประกอบ</u>

2.1 คำสั่งเกี่ยวกับการรีโมท

- RM1 และ RM0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- LK1 และ LK0
- MX1 และ MX0

2.2 คำสั่งเกี่ยวกับการเลือกลักษณะสัญญาณ

- สัญญาณ SINE
- สัญญาณ รูปสามเหลี่ยม
- สัญญาณ รูปสี่เหลี่ยม
- สัญญาณ รูปพัลส์
- สัญญาณ รูปฟันเลื่อย
- สัญญาณ รูปแบบ Arbitery

2.3 คำสั่งเกี่ยวกับการเลือกโหมดการทำงาน

- โหมดการสร้างสัญญาณแบบต่อเนื่อง (CONT)
- โหมดการสร้างสัญญาณแบบ GATED
- โหมดการสร้างสัญญาณแบบ TRIGGER

2.4 คำสั่งเกี่ยวกับเอาท์พุทและออฟเซ็ท

- OT1 และ OT0
- OF1 และ OF0
- DOF

2.5 คำสั่งเกี่ยวกับการกวาดสัญญาณ

- SW1 และ SW0
- DSW
- TRG

2.6 คำสั่งเกี่ยวกับความถี่

- DFR
- DST
- DSP

2.7 คำสั่งเกี่ยวกับการยกเลิก

- Clear
- ARC
- ARE

2.8 คำสั่งเกี่ยวกับความกว้างของพัลส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- DWT

คำสั่งที่ประกอบด้วยข้อมูลที่เป็นทศนิยม

คำสั่งทั้งหมดที่ประกอบด้วยข้อมูลแบบทศนิยมจะประกอบด้วยตัวอักษร 3 ตัวแล้วตามด้วย " : " ความยาวของข้อมูลที่มากที่สุดจะเป็นเลขจำนวน 5 หลักรวมกับจุดทศนิยมด้วย โดยรูปแบบของคำสั่งนั้นจะมีค่าเลขยกกำลังหรือไม่ก็ได้ จะมีทศนิยมหรือไม่มีก็ได้ แต่จะมีหน่วยเป็นโวลท์ , เฮิร์ท และวินาที แต่จะไม่มีการส่งค่าหน่วยลงไปบนบัส ข้อมูลสามารถมีเครื่องหมายนำหน้าได้ถ้าจำเป็น แต่สัญญาณที่เป็นบวกไม่ต้องมีเครื่องหมายนำหน้าก็ได้ ระหว่างเครื่องหมายกับค่าที่กำหนดจะต้องติดกันห้ามเว้นวรรค

ตัวอย่าง FRQ:1000 มีค่าเท่ากับ FRQ:1000.0

FRQ:1E3 มีค่าเท่ากับ FRQ:1E+3

FRQ:1.0000E+3 มีค่าเท่ากับ FRQ:10E+2

FRQ:0.0001E7 มีค่าเท่ากับ FRQ:10000E-1

2.9 คำสั่งเกี่ยวกับความถี่

- FRQ
- STT
- STP

2.10 คำสั่งเกี่ยวกับการกวาดสัญญาณ

- SWT

2.11 คำสั่งเกี่ยวกับความกว้างของพัลส์

- WDT

2.12 คำสั่งเกี่ยวกับแอมปลิจูด

- AMP

2.13 คำสั่งเกี่ยวกับการออฟเซ็ท

- OFS

คำสั่งร้องขอ

คำสั่งเหล่านี้จะทำการเก็บค่าสตริงของข้อมูลที่สามารถอ่านออกมาได้ทันทีที่เครื่องถูกกำหนดให้เป็นตัวส่ง โดยจะมีคำสั่งต่างๆดังนี้

2.14 คำสั่งเกี่ยวกับความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

- FRQ?
- STT?
- STP?

2.15 คำสั่งเกี่ยวกับการกวาดสัญญาณ

- SWT?

2.16 คำสั่งเกี่ยวกับความกว้างของพัลส์

- WDT?

2.17 คำสั่งเกี่ยวกับแอมปลิจูด

- AMP?

2.18 คำสั่งเกี่ยวกับการออฟเซ็ท

- OFS?

2.19 คำสั่งเกี่ยวกับสถานะและตัวเครื่อง

- STA?
- ID?
- VER

2.20 คำสั่งเกี่ยวกับข้อมูลของสัญญาณแบบ arbitery

- ARD?

ค่าเอาท์พุทจะอยู่ในรูปแบบเลขทศนิยมกับเลขยกกำลัง สตริงข้อมูลแต่ละค่าจะขึ้นต้นด้วยคำสั่งของตัวมันเอง เช่น

"FRQ:1.2345E+3"

"OFS:-3.0E+0"

"WDT:45.6E-6"

สำหรับการส่งคำสั่งควบคุมเครื่อง สามารถที่จะส่งคำสั่งไปควบคุมพร้อมกันหลายฟังก์ชันโดยสามารถส่งคำสั่งเหล่านั้นไปพร้อมกันเป็นข้อมูลชุดเดียวกันได้ เช่น

"FRQ:12.34E+3 TRI OT1 AMP:10"

Frequency 12.3 K Hz; Triangle; Output on; Voltage amplitude 10 V

3. **Digital Oscilloscope** ของบริษัท Tektronix รุ่น TDS360 ทำหน้าที่ในการวัดสัญญาณที่มีค่าความถี่ได้ถึง 200 MHz โดย Oscilloscope นี้ สามารถที่จะทำการบันทึกสัญญาณที่ได้จากการวัดมาเก็บไว้ในหน่วยความจำของเครื่อง และสามารถที่จะอ่านค่าของสัญญาณที่บันทึกเก็บไว้ผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

36946

ระบบบัสแบบ IEEE-488 ได้ และสามารถที่จะควบคุมช่วงเวลาต่อช่อง (Time per Division) และจำนวนช่องต่อโวลท์ (Volt per Division) ได้ สำหรับคำสั่งที่ใช้ควบคุมการทำงานผ่านระบบบัส IEEE-488 มี ดังนี้ คือ

3.1 คำสั่งเกี่ยวกับการควบคุมการ Coupling

- CH<X> : COUPLING {AC/DC/GND}

3.2 คำสั่งเกี่ยวกับการควบคุมช่องต่อโวลท์ (Volt per Division)

- CH<X> : VOLTS <NR3>

3.3 คำสั่งเกี่ยวกับการควบคุมเวลาต่อช่อง (Time per Division)

- HORizontal:Main:Scale <NR3>

3.4 คำสั่งเคลียร์

- CLEARMenu

3.5 คำสั่งแปลงคลื่นแปลงคลื่นสัญญาณเป็นข้อมูลตัวเลข

- CURVE?

3.6 คำสั่งเกี่ยวกับการอ่านข้อมูล

- DATA : START <NR1>
- DATA : STOP <NR1>
- DATA : SOURCE <wfm>[<comma><wfm>]...
- DATA : ENCdg {ASCII/RIbinary/SRIbinary/SRPbinary}
- DATA : WIDTH <NR1>
- WFMPre:<wfm>

3.7 คำสั่งเกี่ยวกับฟังก์ชั่นคณิตศาสตร์

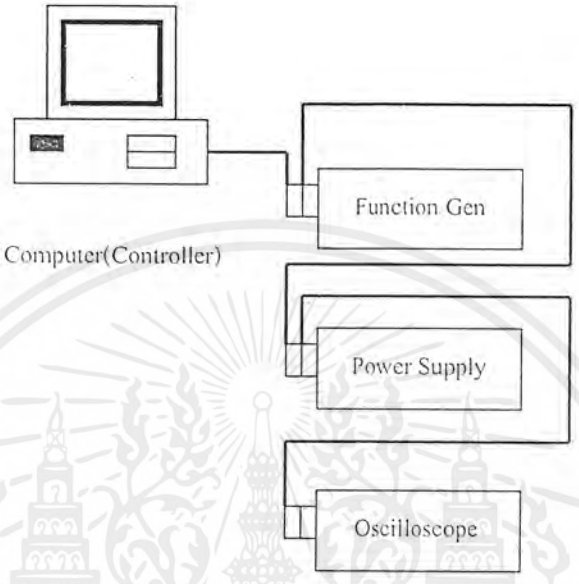
- MATH1:DEFINE{<Qstring>}

<u>การเชื่อมต่ออุปกรณ์ในระบบ</u>

สำหรับการต่อเชื่อมอุปกรณ์ในระบบเครื่องมือวัดที่ทำการศึกษานั้น ได้ทำการต่ออุปกรณ์ตามรูปแบบมาตรฐาน IEEE-488 โดยทำการต่อแบบการต่อต่อเนื่องกัน(Daisy ChainConfiguration) ดังแสดงในภาพที่ 2 เนื่องจากมีข้อดีคือ ขั้วต่อของสายสัญญาณที่ต่อเข้ากับอุปกรณ์ในระบบไม่ได้ต่อรวมกันไว้ที่จุดจุดเดียว แต่จะเรียงต่อเนื่องกันไปทำให้น้ำหนักของขั้วต่อสัญญาณซึ่งมีขนาดใหญ่และมีน้ำหนักมากจะกระจายไปอยู่ทุกๆตัวอุปกรณ์ ในขณะที่การต่อแบบกระจาย (Star Configuration) เป็นการต่อที่ขั้วต่อของสายสัญญาณ จะต่อรวมเข้าไว้ที่จุดใดจุดหนึ่ง ทำให้น้ำหนัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของขั้วสายสัญญาณไปรวมอยู่ที่จุดใดจุดหนึ่ง ซึ่งจะทำให้ขั้วต่อสายสัญญาณที่อุปกรณ์ตัวนั้นอาจเกิดการชำรุดเสียหายได้ง่าย ดังแสดงในรูปที่ 4.1



รูปที่ 4.1 แสดงอุปกรณ์ที่ต่ออยู่ในระบบที่ทำการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การแปลงฟูริเยอร์ (Fourier Transform)

ปัจจุบันอุปกรณ์คอมพิวเตอร์ได้เข้ามามีบทบาทในชีวิตประจำวันมากขึ้นเรื่อยๆ ทำให้มีการศึกษาและพัฒนาการประมวลผลเชิงเลขกันมากขึ้น เพื่อนำไปใช้กับงานต่างๆ เช่น การประมวลสัญญาณเสียง (speech), สัญญาณภาพ (image) ซึ่งในขบวนการประมวลผลของสัญญาณ (Signal Processing) นั้น จะทำการแปลงสัญญาณจากเดิมที่อยู่ในเทอมของเวลา (Time Domain) ให้อยู่ในเทอมของความถี่ (Frequency Domain) เพื่อประโยชน์ในการประยุกต์ใช้ต่อไป เรามีวิธีการแปลงจากเทอมของเวลาไปเป็นเทอมของความถี่ เรียกว่า การแปลงฟูริเยอร์เต็มหน่วย (Discrete Fourier Transform หรือ DFT)

5.1 ฟังก์ชันต่อเนื่อง (Continuous Function)

สำหรับการแปลงฟังก์ชัน f(t) ของฟังก์ชันต่อเนื่อง จะสามารถเขียนการแปลงฟูริเยอร์เป็นฟังก์ชัน F(f) ในรูปสมการได้ว่า

$$F(f) = \int_{-\infty}^{\infty} f(t)e^{-j2\pi ft}\,dt \qquad \text{.....(5.1)}$$

และมีการแปลงผกผัน (Inverse Transform) ในรูปสมการเป็น

$$f(t) = \int_{-\infty}^{\infty} F(f)e^{j2\pi ft}\,df \qquad \text{.....(5.2)}$$

เมื่อ j มีนิยามคือ $j^2 = -1$ และ e นิยามเป็น

$$e^{j\phi} = \cos(\phi) + j\sin(\phi) \qquad \text{.....(5.3)}$$

โดยที่ F(f) หมายถึง ฟังก์ชันโดเมนความถี่ และ f(t) หมายถึง ฟังก์ชันโดเมนเวลา

แต่เนื่องจากการใช้คอมพิวเตอร์ จะเป็นการวนรอบเพื่อหาคำตอบ ซึ่งใช้เวลานาน และไม่แน่ใจว่าจะได้คำตอบหรือไม่ ในทางกลับกัน หากเราสามารถมีการแปลงแบบใหม่ที่การคำนวณใช้อนุกรมจำกัดให้ค่าใกล้เคียงกับการแปลงเดิม แต่ใช้เวลาในการคำนวณลดลง ซึ่งเรียกว่า การแปลงฟูริเยอร์เต็มหน่วย (Discrete Fourier Transform , DFT)

5.2 การแปลงฟูริเยอร์เต็มหน่วย (Discrete)

พิจารณาอนุกรมเชิงซ้อน (complex series) x(k) ที่มีความยาว N จุด

$$x_0, x_1, x_2, x_3, x \ldots x_k, x_{N-1} \qquad \text{.....(5.4)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ x เป็นจำนวนเชิงซ้อน (complex number)

$$x_i = x_{real} + jx_{image} \qquad \text{.....(5.5)}$$

นอกจากนี้ เรายังสามารถสมมติค่าลำดับอยู่ในช่วง 0 ถึง N-1 เพื่อขยายคาบ N จุด นั่นคือได้ $x_k = x_{k-N}$ สำหรับทุกๆค่าของ k ในการแปลงอนุกรมฟูริเยอร์ (Fourier Series) จะเขียนเป็น X(k) โดยมี N จุด และจะสามารถเขียนสมการการแปลงโดยตรงได้เป็น

$$X(n) = \frac{1}{N}\sum_{k=0}^{N-1} x(k)e^{-jk2\pi k2\pi} \quad \text{สำหรับ } n= 0..N-1 \qquad \text{.....(5.6)}$$

และเขียนสมการการแปลงผกผันได้เป็น

$$x(n) = \sum_{k=0}^{N-1} X(k)e^{jk2\pi k2\pi} \quad \text{สำหรับ } n= 0..N-1 \qquad \text{.....(5.7)}$$

ถึงแม้ว่าเราจะสามารถอธิบายฟังก์ชันนี้ได้ในรูปของ complex series แต่ค่า real part ก็สามารถแสดงได้โดยการตั้งให้ค่า imaginary part เป็นศูนย์ และโดยทั่วไปการแปลงจากเทอม Frequency Domain จะสร้างอยู่ในรูปฟังก์ชันเชิงซ้อน (Complex Function) ซึ่งจะมีค่า magnitude และ phase ดังนี้

$$\text{magnitude} = \left\|X(n)\right\| = (x_{real} * x_{real} + x_{image} * x_{image})^{0.5} \qquad \text{.....(5.8)}$$

$$\text{phase} = \tan^{-1}\left(\frac{x_{image}}{x_{real}}\right) \qquad \text{.....(5.9)}$$

การแปลงฟาสท์ฟูริเยอร์ (Fast Fourier Teansform , FFT)

ปกติแล้วการแปลง DFT สำหรับสัญญาณที่มีความยาว N ลำดับหรือจุดนั้น คอมพิวเตอร์จะต้องทำการคำนวณจำนวนเชิงซ้อนถึง N× N ครั้ง และบวกจำนวนเชิงซ้อนอีก N(N-1) ครั้ง ซึ่งคอมพิวเตอร์ทั่วไปแล้วไม่มีคำสั่งภาษาเครื่องที่ใช้คูณตัวเลข จึงจำเป็นต้องเขียนโปรแกรมย่อย หรือเพิ่มวงจรคูณโดยเฉพาะเข้าไป ส่วนการบวกเลขนั้น คอมพิวเตอร์สามารถทำได้ง่ายและรวดเร็ว นั่นคือเราสามารถกล่าวได้ว่า ความเร็วในการคำนวณ DFT นั้นขึ้นอยู่กับความเร็วและจำนวนครั้งในการคูณตัวเลขเป็นสำคัญ ถ้าเราสามารถทำให้จำนวนครั้งในการคูณน้อยลง ก็จะทำให้สามารถคำนวณ DFT ได้เร็วขึ้น และวิธีที่สามารถทำให้จำนวนครั้งในการคูณตัวเลขน้อยลงก็คือ การคำนวณแบบ FFT (Fast Fourier Transform)

การแปลงฟาสท์ฟูริเยอร์(FFT) นี้ จะทำให้การคำนวณ DFT ใช้การคูณจำนวนเชิงซ้อนเพียง $N \log_2 N$ ครั้งเท่านั้น หรือก็คือจำนวนครั้งในการคูณตัวเลขลดลงไปถึง $N/(\log_2 N)$ เท่า เราจะได้ว่า การแปลงฟาสท์ฟูริเยอร์เป็นเพียงวิธีการ หรือลำดับการในการคำนวณที่ช่วยให้การคำนวณ DFT ซึ่งเป็นการแปลงฟูริเยอร์ ได้เร็วขึ้นเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัจจุบันนี้ได้มีการคิดค้น ดัดแปลง และเสนอผลงานเกี่ยวกับ FFT มากมายหลายแบบ ซึ่งแต่ละแบบก็จะมีข้อดีข้อเสียแตกต่างกันไป โดยทั่วไปแล้วเราอาจแบ่งออกเป็น 2 ชนิดใหญ่ๆ คือ ชนิดลดทอนทางเวลา(Decimation inTime , DIT) และชนิดลดทอนทางความถี่(Decimation in Frequency)

5.3 หลักการเบื้องต้นของ FFT

ในการแปลง FFT ต้องใช้ DFT แทนการแปลงฟูริเยอร์แบบต่อเนื่อง โดยใช้ฟังก์ชันวินโดว์ลดข้อมูลแบบไม่ต่อเนื่องทั้งหมดให้เหลือจำนวน N ตัว(N คือ ช่วงเวลาหรือขนาดของวินโดว์ที่ใช้ในการแปลง FFT) ข่าวสารต่างๆจะไม่สูญหายถ้าการแปลงนั้นมีการสุ่มมาด้วยความถี่ที่สูงพอ (ระยะห่างของ N) และ วินโดว์ w(n) ไม่มีจุดสุ่มที่เป็นศูนย์ตลอดช่วง N ค่าของ N จะมีผลต่อการวิเคราะห์คือ ถ้า N ต่ำเกินไป จะทำให้ความละเอียดในโดเมนความถี่จะหยาบมาก เพราะจะทำให้ได้ผลดีในโดเมนเวลา เนื่องจากการเฉลี่ยถูกทำเฉพาะในช่วงสั้นๆเท่านั้น และถ้า N มีขนาดใหญ่เกินไป ผลของความละเอียดในโดเมนเวลาจะแย่ แต่ทำให้โดเมนความถี่มีความละเอียดสูง

การวิเคราะห์ฟูริเยอร์เป็นเทคนิคที่ใช้ในการหาความสัมพันธ์ระหว่าง ฟังก์ชันโดเมนเวลากับฟังก์ชันโดเมนความถี่ โดยรูปการแปลงฟูริเยอร์สำหรับฟังก์ชันต่อเนื่อง แสดงดังสมการที่ 5.1 และ 5.2

ถ้าเราทำการแปลงรูปคลื่นในลักษณะที่โดเมนเวลาเป็นฟังก์ชันแบบไม่ต่อเนื่อง และมีช่วงจำกัด ลักษณะนี้เรียกว่า การแปลงฟูริเยอร์เต็มหน่วย (Discrete Fourier Transform) ซึ่งสามารถเขียนสมการการแปลงได้ตามสมการที่ 5.6 และ 5.7 ดังได้กล่าวไว้ข้างต้น

5.3.1 FFT ชนิดลดทอนทางเวลา (Decimation-in-Time , DIT)

เป็นการจัดแบ่งกลุ่มลำดับสัญญาณในโดเมนเวลา x(m) ที่มีขนาด N จุด ออกเป็นสองลำดับสัญญาณที่มีความยาว N/2 จุดเท่ากัน ทั้งสองลำดับนี้เรียกว่า ลำดับสัญญาณคู่และลำดับสัญญาณคี่ โดยที่ลำดับสัญญาณคู่เกิดจากการเอาลำดับในตำแหน่งเป็นเลขคู่มาเรียงกัน ที่เหลือก็เป็นลำดับสัญญาณคี่ ดังนั้น ถ้าเรานิยามให้ $x_E(m)$ เป็นลำดับคู่ และ $x_O(m)$ เป็นลำดับคี่ จะได้ว่า

$$x_E(m) = x(2m) \;;\; m = 0,1,\ldots,(N/2)-1 \qquad \ldots\ldots(5.10)$$

$$x_O(m) = x(2m+1) \;;\; m = 0,1,\ldots,(N/2)-1 \qquad \ldots\ldots(5.11)$$

ด้วยการจัดแบ่งเช่นนี้ ถ้าให้ W_N แทนค่า W ของลำดับยาว N จุด ทำให้การคำนวณการแปลง DFT ของลำดับสัญญาณ x(m) ที่ยาว N จุด เขียนใหม่ได้เป็น

$$X(k) = \sum_{m=0}^{(N/2)-1} x_E(m)(W_{N/2})^{mk} + (W_N)^k \sum_{m=0}^{(N/2)-1} x_O(m)(W_{N/2})^{mk} \quad \ldots(5.12)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$X(k) = X_1(k) + (W_N)^k X_2(k) \quad \text{.....(5.13)}$$

เมื่อเขียนให้พจน์ $$(W_N)^k = (e^{j2\pi 2\pi})^2 = e^{j2\pi 2\pi/N} = W_{N/2} \quad \text{.....(5.14)}$$

5.3.2 FFT ชนิดลดทอนทางความถี่ (Decimation-in-Frequency , DIF)

หลักการสำคัญของ DIF คล้ายคลึงกับ DIT โดยที่ DIF ทำการแบ่งลำดับโดเมนเวลา x(m) ออกเป็นสองส่วนเท่าๆกัน โดยการแบ่งครึ่งซึ่งทำโดย ถ้าให้ $x_E(m)$ และ $x_O(m)$ แทนลำดับสัญญาณที่ได้จากการแบ่งครึ่งนี้

$$x_E(m) = x(m) \quad ; \ m = 0,1,\ldots,(N/2)-1 \quad \text{.....(5.15)}$$

$$x_O(m) = x(m + N/2) \quad ; \ m = 0,1,\ldots,(N/2)-1 \quad \text{.....(5.16)}$$

เพราะฉะนั้นการคำนวณ DFT ขนาด N จุดของ x(n) สามารถเขียนแยกได้เป็น 2 ส่วน คือ

$$X(k) = \sum_{m=0}^{(N/2)-1} x_E(m)(W_{N/2})^{mk} + (W_N)^{Nk/2} \sum_{m=0}^{(N/2)-1} x_O(m)(W_N)^{mk} \quad \text{.....(5.17)}$$

จะได้แต่ละพจน์เป็น

$$X(k) = \sum_{m=0}^{(N/2)-1} [x_E(m) + (-1)^k * x_O(m + N/2)](W_N)^{mk} \quad \text{.....(5.18)}$$

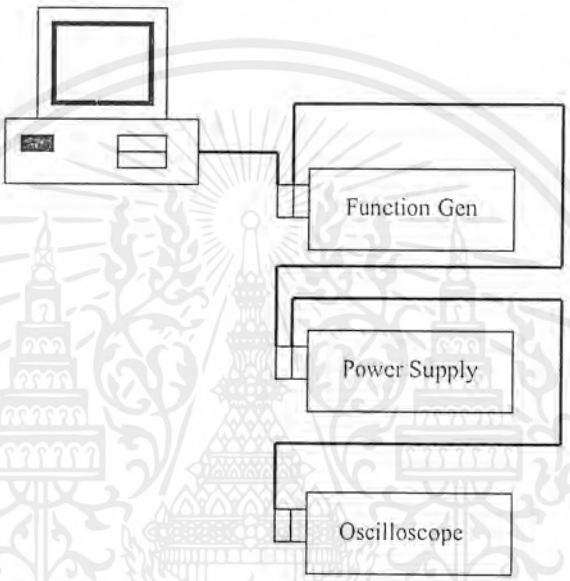
ข้อแตกต่างที่สำคัญระหว่างทั้งสองวิธีการคำนวณ คือ

1. DIT ข้อมูล x(n) ต้องทำการสลับตำแหน่งกัน แต่ X(k) เรียงตามธรรมชาติ ส่วน DIF จะกลับตรงกันข้าม คือ x(n) เรียงตามธรรมชาติ และ X(k) ถูกเรียงสลับแบบผันกลับบิต
2. หน่วยผีเสื้อของ DIF ต่างออกไปจากของ DIT โดย DIF ได้เอาลำดับ $x_E(m)$ และ $x_O(m)$ มาบวกและลบกันก่อนแล้วจึงทำการคูณด้วยจำนวนเชิงซ้อน $(W_N)^k$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การทดลองและการประยุกต์ใช้งาน

ในการทดลองบทนี้เป็นการทดลองเพื่อทดสอบว่า สามารถควบคุมชุดเครื่องมือวัด GPIB ได้จากเครื่องคอมพิวเตอร์ ซึ่งสำหรับระบบที่ใช้ในการทดลองจะเป็นดังรูปที่ 6.1



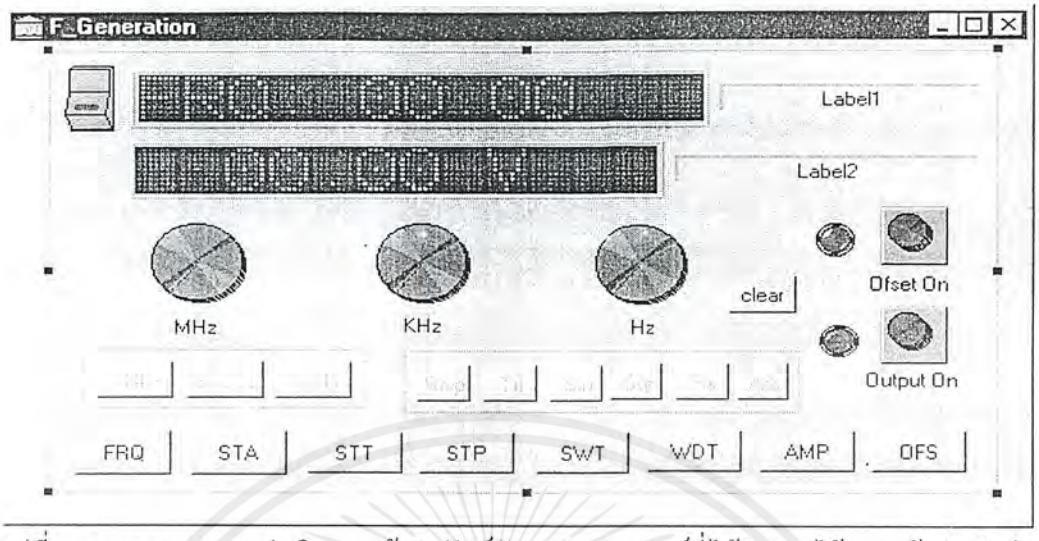
รูปที่ 6.1 แสดงอุปกรณ์ที่ต่ออยู่ในระบบที่ทำการทดลอง

การทดลองจะเป็นการทดสอบการทำงานผ่านการ์ด GPIB โดยควบคุมทางจอคอมพิวเตอร์ และเป็นการทดลองสั่งงานอุปกรณ์ทีละตัว

6.1 การทดลองสั่งงาน Programmable Function Generator (HM8130)

ในการทดลองสั่งงาน Programmable Function Generator นี้ จะเป็นการทดลองว่า ค่าความถี่ , แอมปลิจูด และ สัญญาณชนิดต่าง ๆ ที่ตั้งโดยผู้ใช้คอมพิวเตอร์ จะมีค่าของข้อมูลที่ปรากฏบนหน้าจอคอมพิวเตอร์เท่ากันกับที่ปรากฏบนหน้าจอของเครื่องมือวัด โดยได้แสดงสถานะปกติของหน้าจอฟังก์ชันเจนเนอเรเตอร์ที่ได้จำลองไว้บนหน้าต่างวินโดว์ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

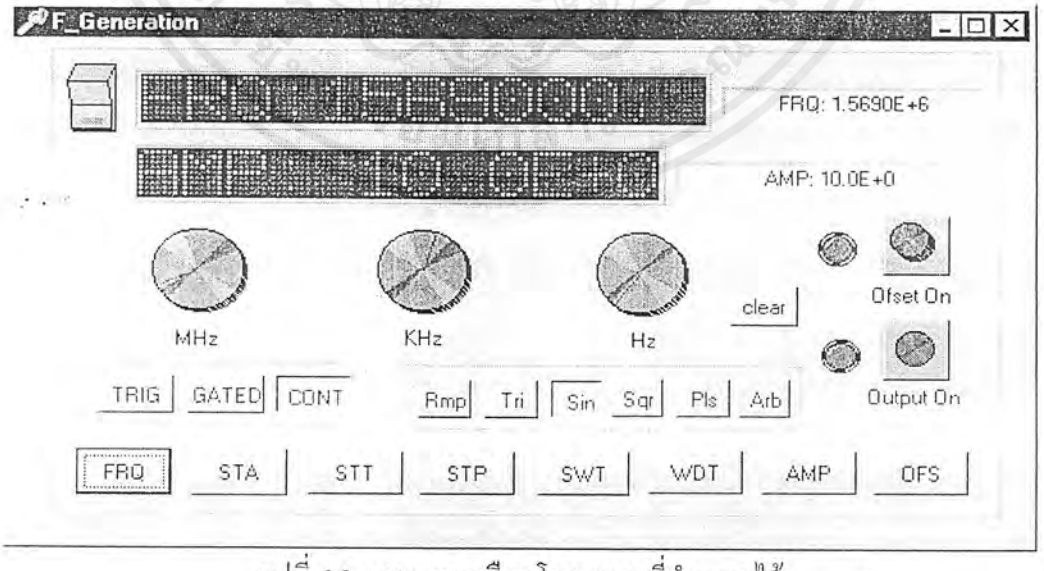


รูปที่ 6.2 แสดงสถานะปกติของหน้าจอฟังก์ชันเจนเนอเรเตอร์ที่ได้จำลองไว้บนหน้าต่างวินโดว์

และได้แสดงตัวอย่างการควบคุมและตั้งค่าเครื่องมือวัด ดังนี้

6.1.1 แสดงการเลือกโหมดรีโมท และตั้งค่าต่าง ๆ ไว้ คือ

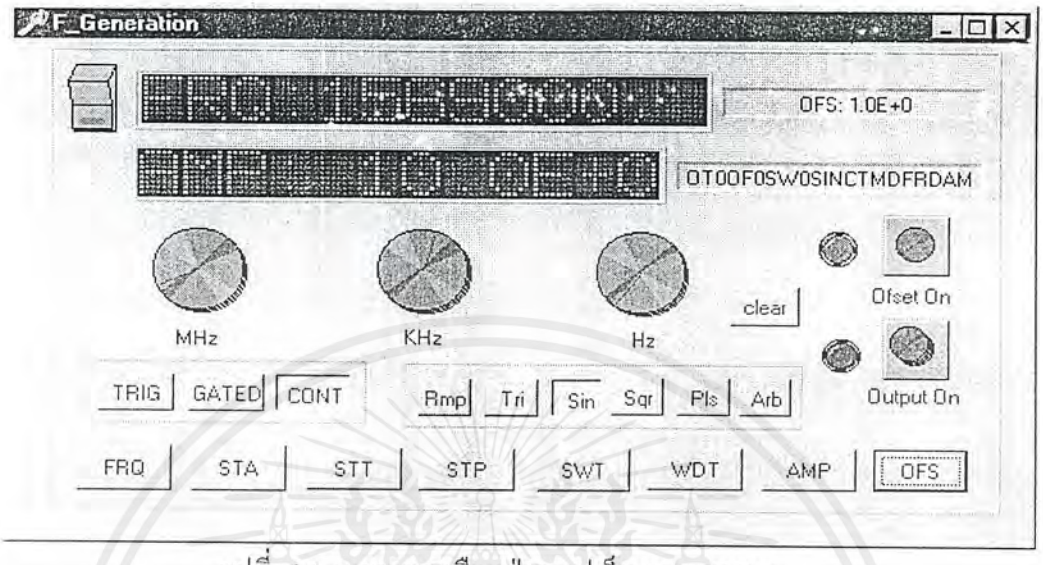
- ค่าความถี่ เท่ากับ 1.569 MHz
- แอมปลิจูด เท่ากับ 10 Volt
- ชนิดของสัญญาณ คือ สัญญาณซายน์
- เลือกโหมด CONT
- เลือกโหมดเอาท์พุท ให้ เปิด



รูปที่ 6.3 แสดงการเลือกโหมดตามที่กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

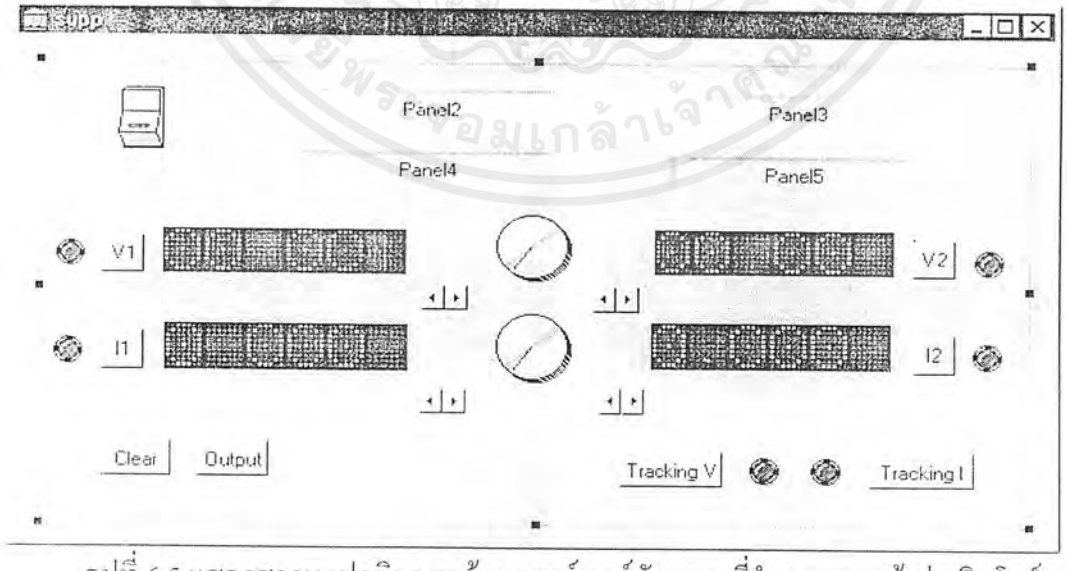
6.1.2 แสดงการเลือกปุ่มออฟเซ็ท และ แสดงสถานะปัจจุบันของฟังก์ชันเจนฯ



รูปที่ 6.4 แสดงการเลือกปุ่มออฟเซ็ท และ สถานะ

6.2 การทดลองสั่งงาน Programmable Power Supply (HM8142)

ในการทดลอง Programmable Power Supply นี้ จะเป็นการทดลองว่า ค่าแรงดันและค่ากระแสที่ตั้งไว้โดยคอมพิวเตอร์ของผู้ใช้ จะมีค่าของข้อมูลที่ปรากฏบนหน้าจอคอมพิวเตอร์เท่ากันกับที่ปรากฏบนหน้าจอของเครื่องมือวัด โดยได้แสดงสถานะปกติของหน้าจอฟังก์ชันเจนเนอเรเตอร์ที่ได้จำลองไว้บนหน้าต่างวินโดว์ ดังนี้

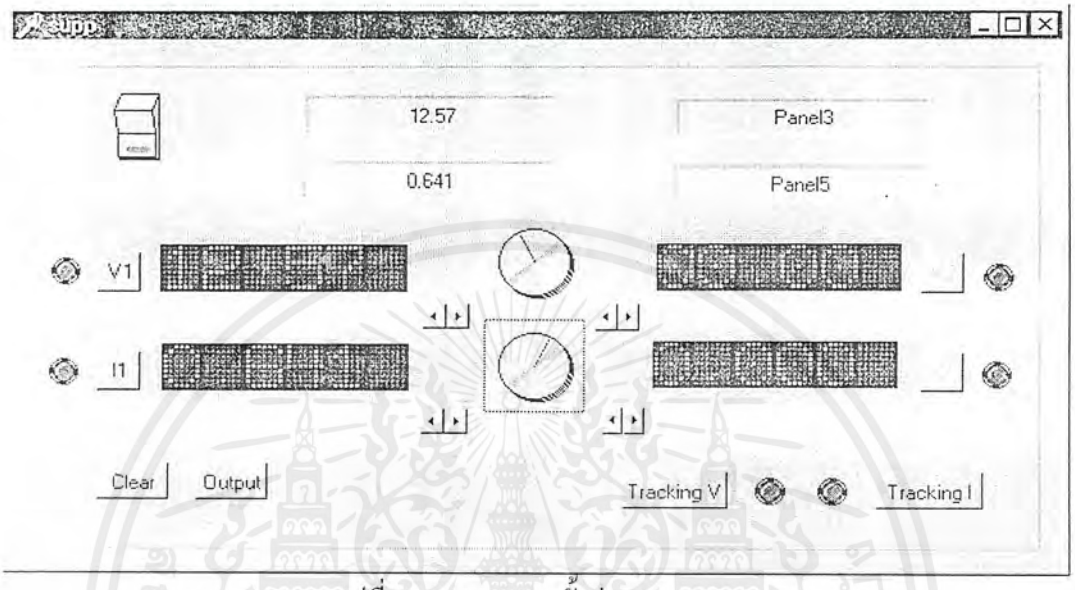


รูปที่ 6.5 แสดงสถานะปกติของหน้าจอเพาว์เวอร์ซัพพลายที่จำลองบนหน้าต่างวินโดว์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

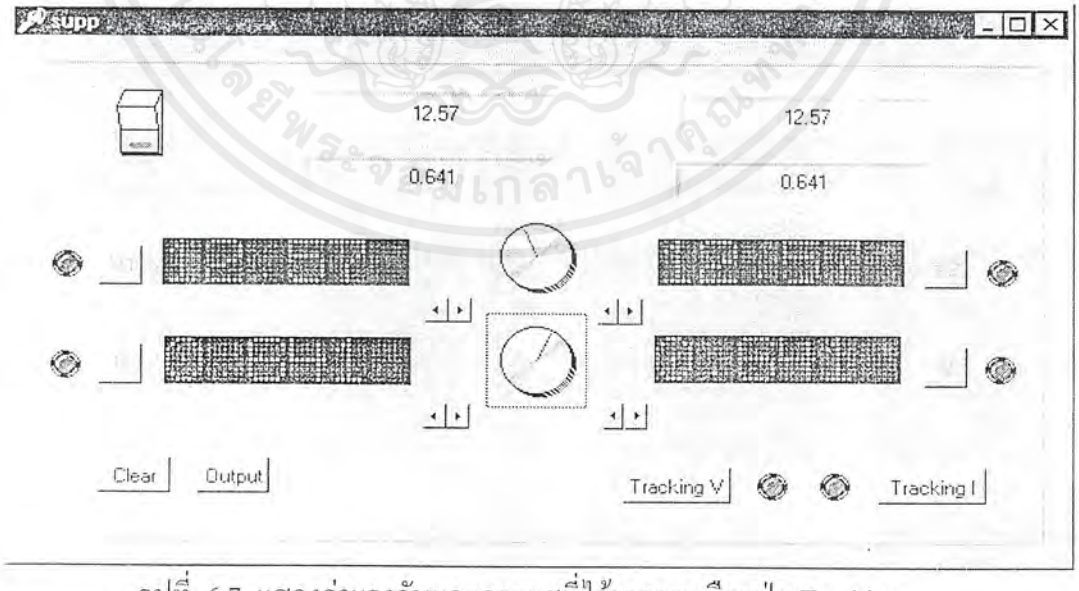
และได้แสดงตัวอย่างการควบคุมและตั้งค่าเครื่องมือวัด ดังนี้

6.2.1 การทดลองตั้งทีละช่องแรงดัน และทีละช่องกระแส โดยตั้งค่าแรงดัน V1 เท่ากับ 12.57 Volt และกระแส I1 เท่ากับ 0.641 A.



รูปที่ 6.6 แสดงการตั้งค่า V1 และ I1

6.2.2 การทดลองตั้งค่าแรงดัน และกระแสทั้งสองช่องเท่ากัน ที่แรงดันเท่ากับ 12.57 Volt และกระแสเท่ากับ 0.641 A. ทำได้โดยเลือกปุ่ม Tracking



รูปที่ 6.7 แสดงค่าแรงดันและกระแสที่ได้จากการเลือกปุ่ม Tracking

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.3 ทดลองต่อโหลด และทำการจ่ายแรงดันและจำกัดค่ากระแสจาก Programmable Power Supply ได้ผลดังนี้

ตารางที่ 6.1 จ่ายแรงดัน 10 V. และจำกัดกระแสไว้ที่ 0.05 A.

แรงดันที่จ่ายให้โหลด (V.)	ความต้านทาน (โหลด) (Ω)	กระแสที่ผ่านโหลด (A.)	แรงดันที่คร่อมโหลด (V.)
10	50	0.05	2.5
10	100	0.05	5
10	200	0.05	10
10	500	0.02	10
10	1000	0.01	10

ตารางที่ 6.2 จ่ายแรงดัน 20 V. และจำกัดกระแสไว้ที่ 0.1 A.

แรงดันที่จ่ายให้โหลด (V.)	ความต้านทาน (โหลด) (Ω)	กระแสที่ผ่านโหลด (A.)	แรงดันที่คร่อมโหลด (V.)
20	50	0.1	5
20	100	0.1	10
20	200	0.1	20
20	500	0.04	20
20	1000	0.02	20

จากการทดลองพบว่า เราสามารถที่จะควบคุม Programmable Power Supply ให้จ่ายแรงดันและจำกัดค่ากระแสได้ ดังจะเห็นได้จากตารางที่ 1 และตารางที่ 2 คือ ค่าแรงดันที่คร่อมโหลดจะน้อยกว่าค่าแรงดันที่จ่ายให้โหลด ทั้งนี้เนื่องจากค่ากระแสที่ผ่านโหลดจะมีค่าเกินค่ากระแสที่จำกัดไว้ไม่ได้ นั่นคือ เราสามารถควบคุม Programmable Power Supply ให้ทำงานได้จริง โดยควบคุมผ่านทางคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 การทดลองสั่งงาน Digital Oscilloscope (Tektronix TDS360)

ในการทดลองสั่งงาน Programmable Oscilloscope นี้จะเป็นการทดลองนำข้อมูลจากเครื่อง Oscilloscope และข้อมูลที่ได้จากการอ่านโดยใช้โปรแกรมคอมพิวเตอร์ที่ควบคุมตัวOscilloscope แล้วแสดงบนหน้าจอ Oscilloscope ที่จำลองไว้บนหน้าต่างวินโดว์ มาเปรียบเทียบว่าตรงกันหรือไม่

ในโครงงานนี้ ได้ทดลองรับค่าฟังก์ชันต่างๆ ดังนี้

1. Time per Division
2. Volt per Division
3. Channel Coupling
4. Select Channel
5. ฟังก์ชันคณิตศาสตร์ (Math)

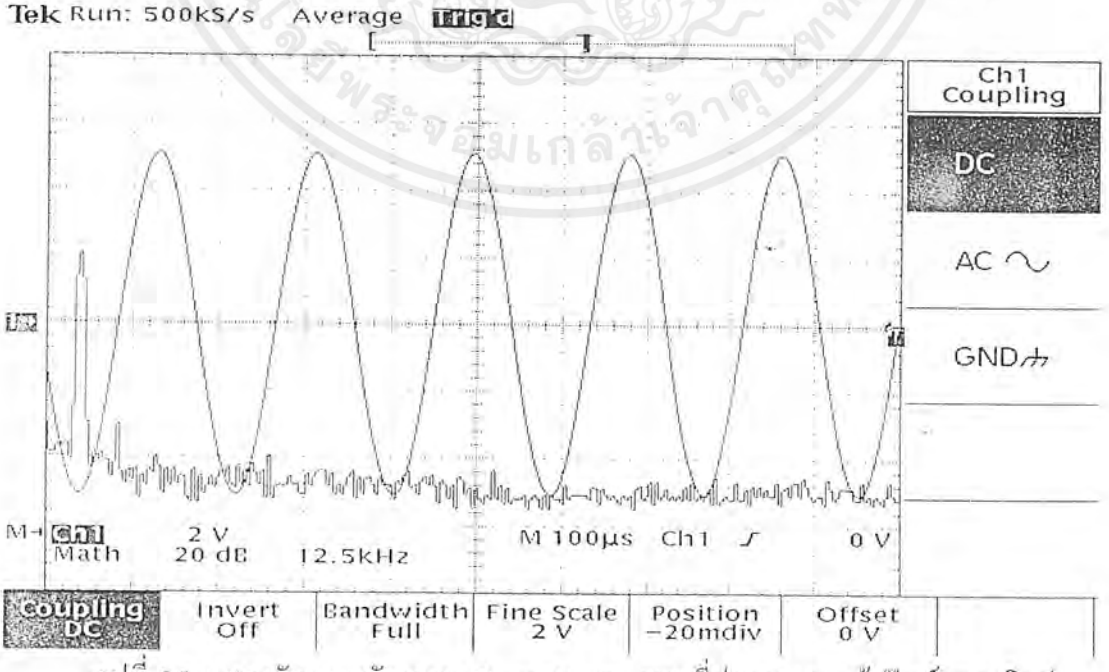
โดยในฟังก์ชัน Fast Fourier Transform (FFT) ซึ่งเป็นหนึ่งในฟังก์ชันคณิตศาสตร์ ได้ทำการรับข้อมูล 2 วิธี คือ รับข้อมูลที่ทำการแปลง FFT จากเครื่อง Oscilloscope เรียบร้อยแล้ว และ รับข้อมูลจากสัญญาณบนเครื่อง Oscilloscope มาทำการแปลง FFT เอง

6.3.1 การทดลองตั้งค่า Time/Div, Volt/Div, สัญญาณซายน์จาก channel1 ที่ความถี่หนึ่ง และสัญญาณ FFT พร้อมทั้งเปรียบผลที่ได้จากการทดลองกับผลที่ได้จากออสซิลโลสโคป

6.3.1.1 Volt/Div = 2 V. Time/Div = 100 us.

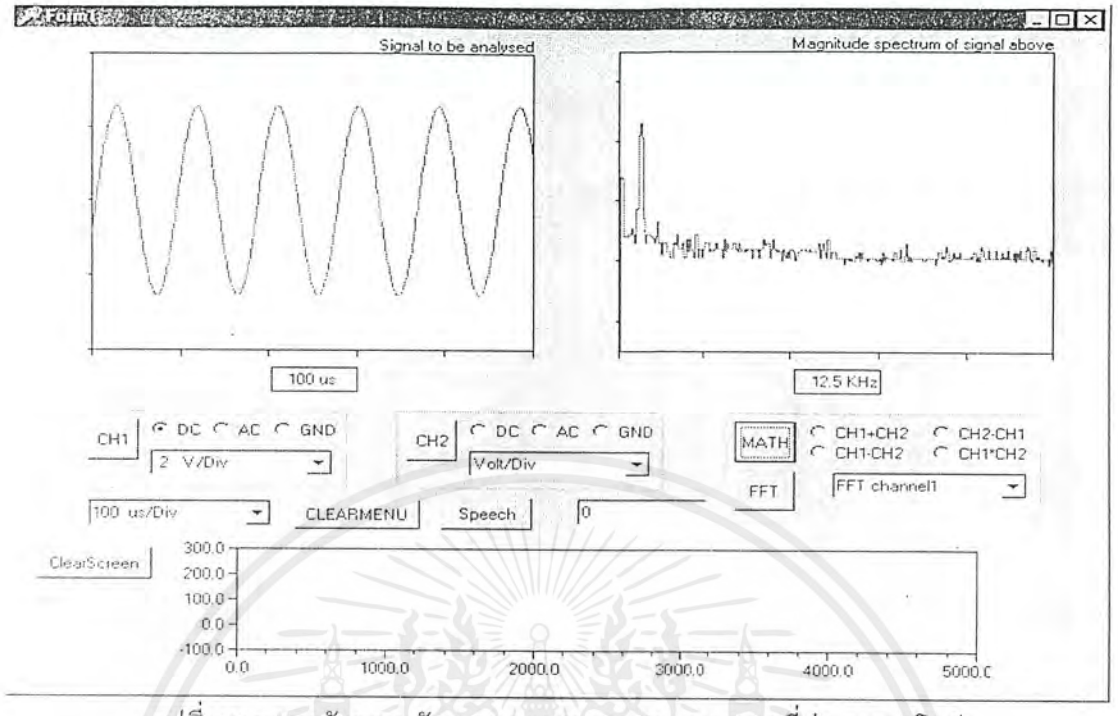
ความถี่ 5.47 KHz เลือก DC Coupling

สัญญาณ FFT ที่อ่านมาจากการแปลงโดยเครื่อง Oscilloscope



รูปที่ 6.8 แสดงลักษณะสัญญาณ Channel1 และ FFT ที่ปรากฏบนหน้าปัทม์ของสโคป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

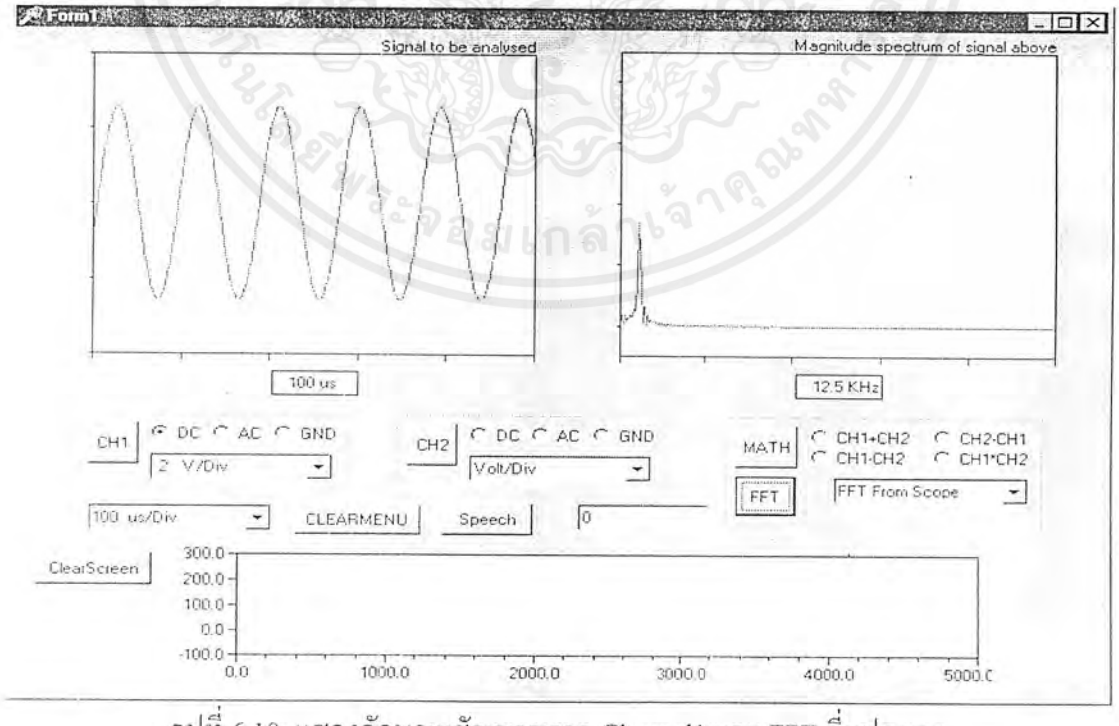


รูปที่ 6.9 แสดงลักษณะสัญญาณจาก Channel1 และ FFT ที่อ่านจากสโคป

6.3.1.2 Volt/Div = 2 V. Time/Div = 100 us.

ความถี่ 5.47 KHz. DC Coupling

สัญญาณ FFT ที่ได้จากการอ่านข้อมูลใน Channel1 มาแปลงเอง



รูปที่ 6.10 แสดงลักษณะสัญญาณจาก Channel1 และ FFT ที่แปลงเอง

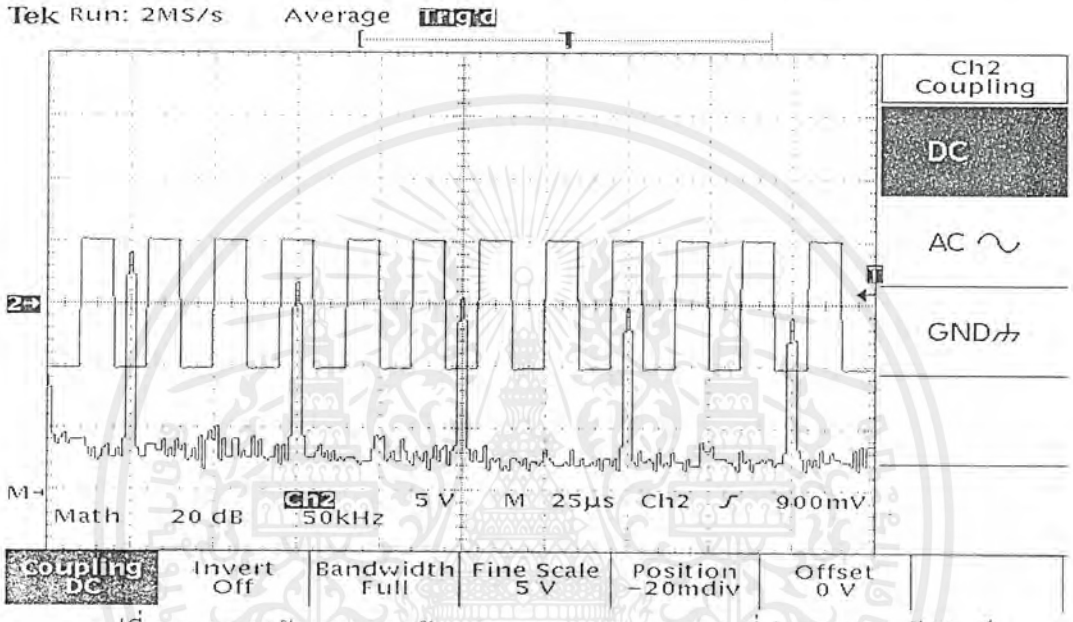
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.2 การทดลองตั้งค่า Time/Div, Volt/Div, สัญญาณสี่เหลี่ยมจาก channel2 ที่ความถี่หนึ่ง และสัญญาณ FFT พร้อมทั้งเปรียบเทียบผลที่ได้จากการทดลองกับผลที่ได้จากออสซิลโลสโคป

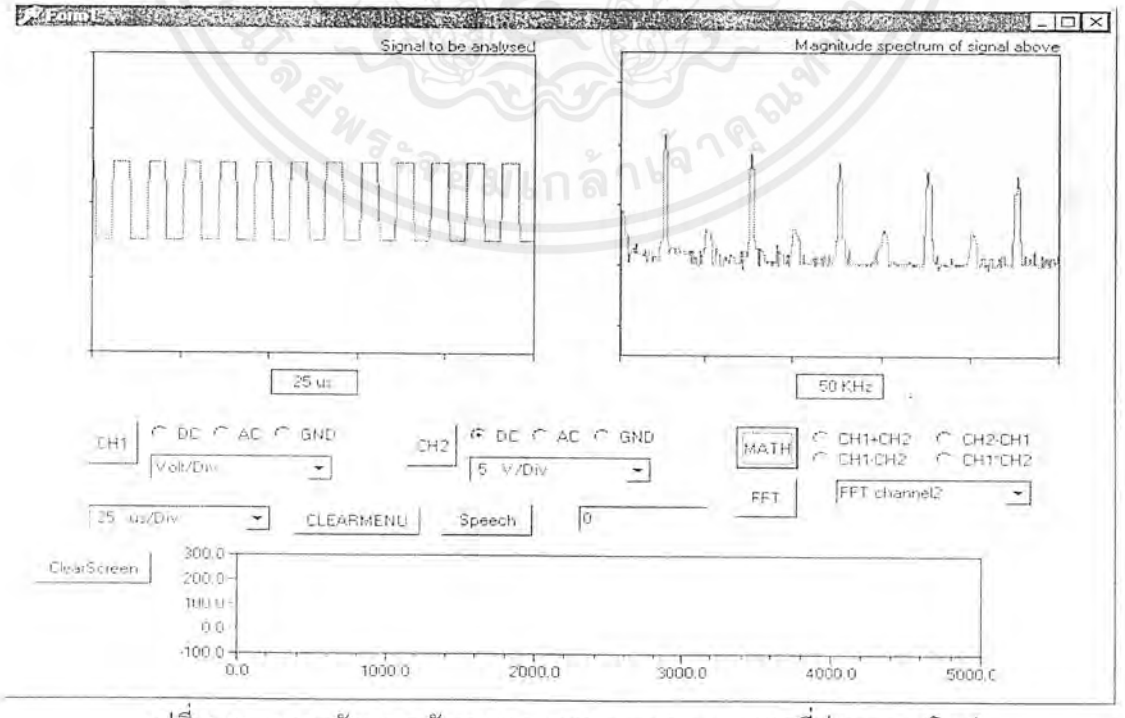
6.3.2.1 Volt/Div = 5 V. Time/Div = 25 us.

ความถี่ 50 KHz เลือก DC Coupling

สัญญาณ FFT ที่อ่านมาจากการแปลงโดยเครื่อง Oscilloscope



รูปที่ 6.11 แสดงลักษณะของสัญญาณ Channel2 และ FFT ที่ปรากฏบนหน้าปัทม์



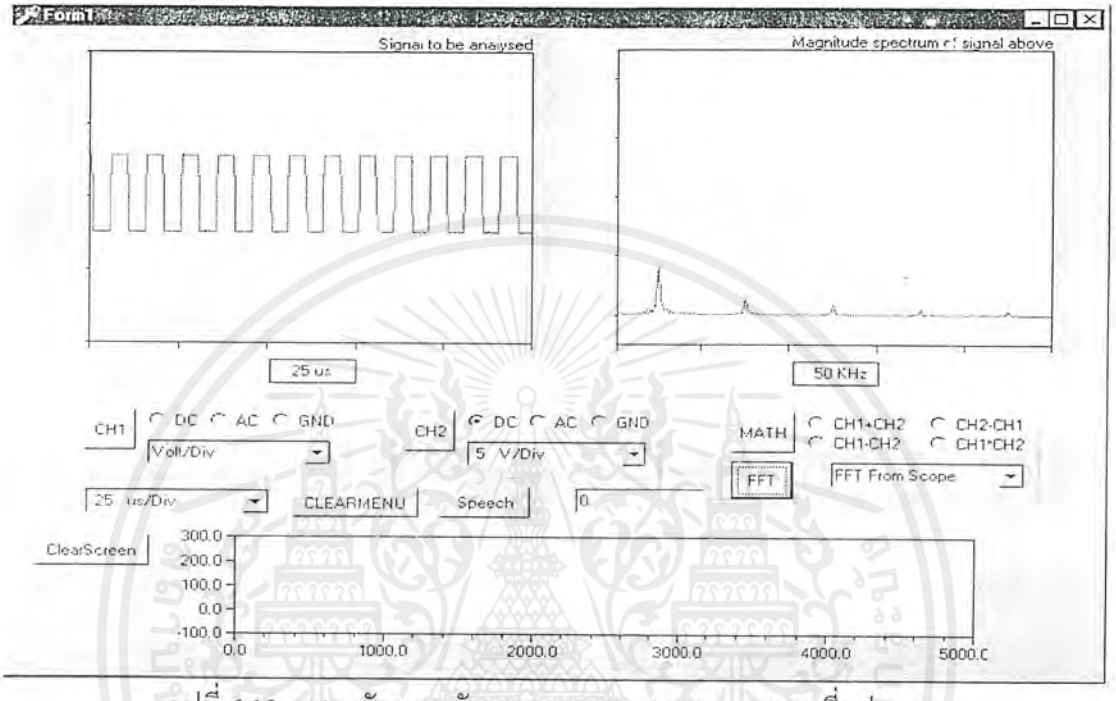
รูปที่ 6.12 แสดงลักษณะสัญญาณจาก Channel2 และ FFT ที่อ่านจากสโคป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.2.2 Volt/Div = 5 V. Time/Div = 25 us.

ความถี่ 50 KHz เลือก DC Coupling

สัญญาณ FFT ที่จากการอ่านข้อมูลใน Channel2 มาแปลงเอง



รูปที่ 6.13 แสดงลักษณะสัญญาณจาก Channel2 และ FFT ที่แปลงเอง

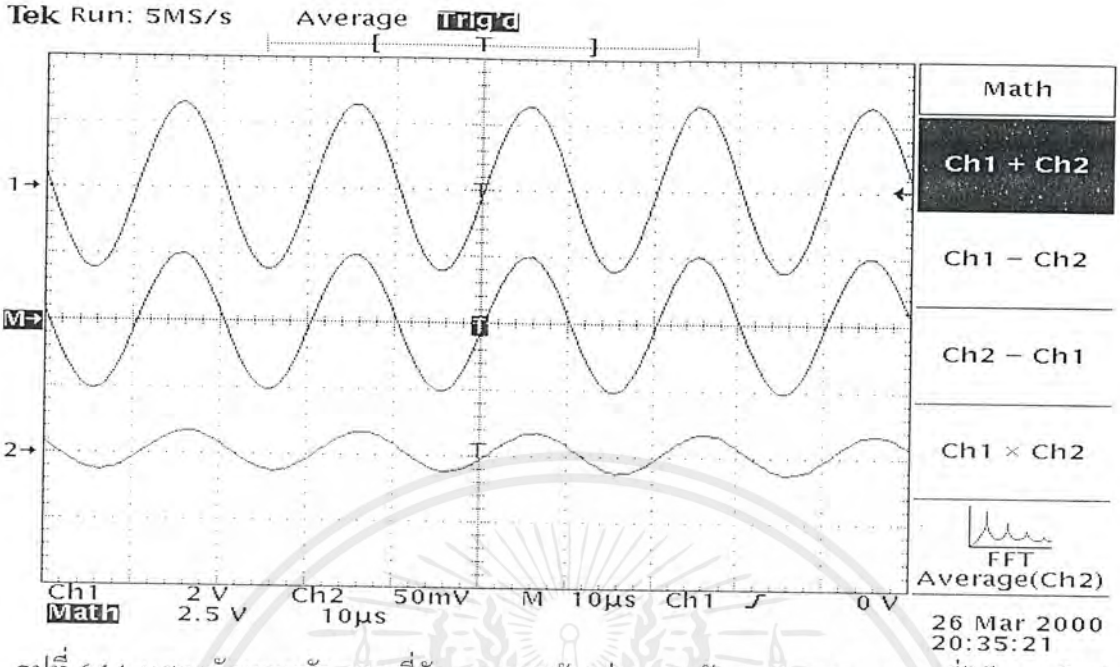
6.3.3 การทดลองเลือกฟังก์ชันทางคณิตศาสตร์ (ปุ่ม MATH) โดยการนำวงจรขยายแบบนอนอินเวอร์ติ้ง (Non-Inverting Amplifier) มาวัดสัญญาณ Input และ Output แล้วนำมากระทำการทางคณิตศาสตร์ ให้ Channel1 วัดสัญญาณ Input และ Channel2 วัดสัญญาณ Output พร้อมทั้งนำผลที่ได้จากการทดลองมาเปรียบเทียบกับผลที่ได้จากสโคป

6.3.3.1 ป้อนสัญญาณซายน์ (sine wave) , Time/Div = 10 us.

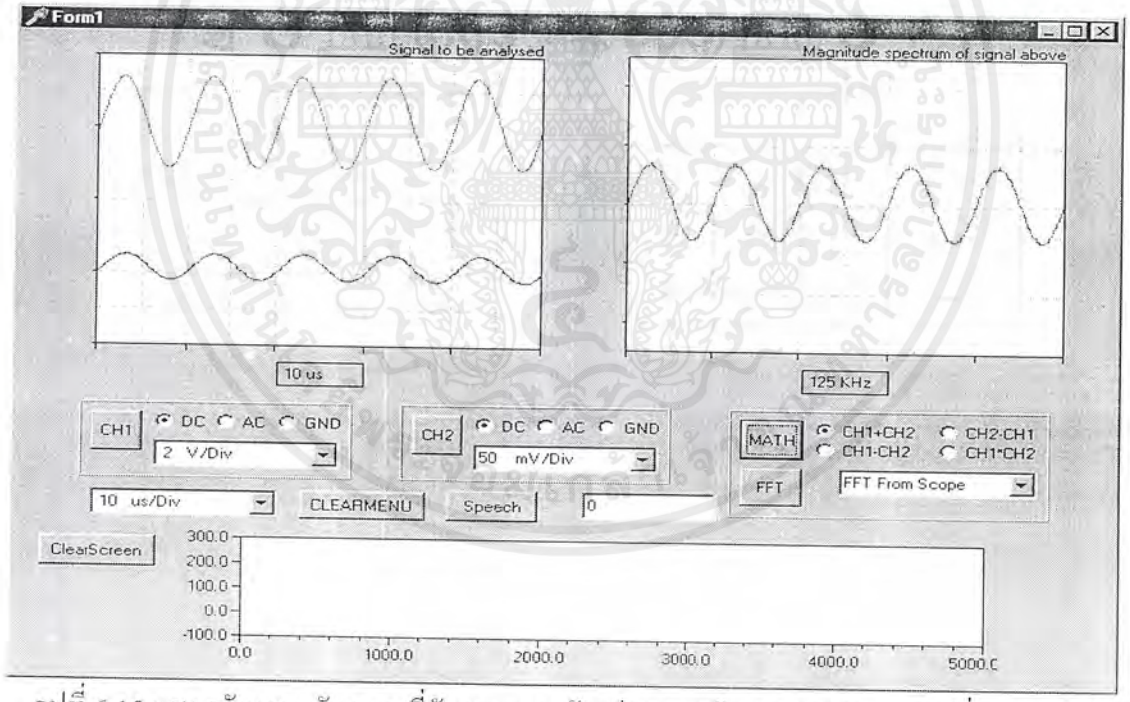
Channel1 Volt/Div = 2 V. , Channel2 Volt/Div = 50 mV.

เลือกฟังก์ชันทางคณิตศาสตร์ CH1 + CH2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.14 แสดงลักษณะสัญญาณที่วัดจากวงจรตัวอย่าง และสัญญาณ CH1 + CH2 ที่ได้จากสโคป



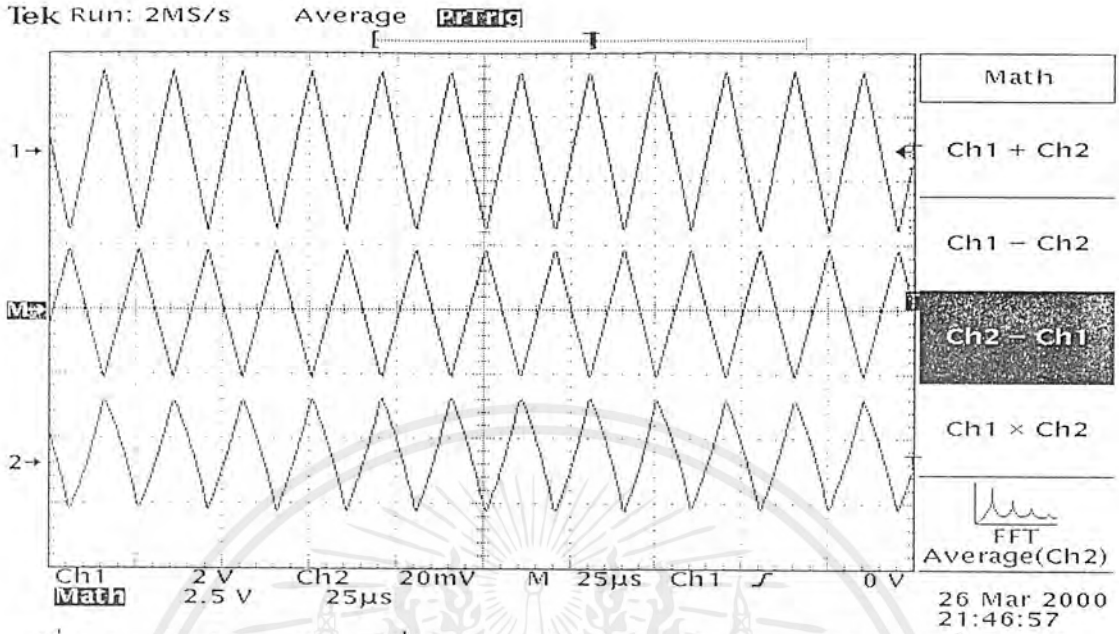
รูปที่ 6.15 แสดงลักษณะสัญญาณที่วัดจากวงจรตัวอย่าง และสัญญาณ CH1 + CH2 ที่ทดลองได้

6.3.3.2 ป้อนสัญญาณสามเหลี่ยม (Triangle wave) , Time/Div = 25 us.

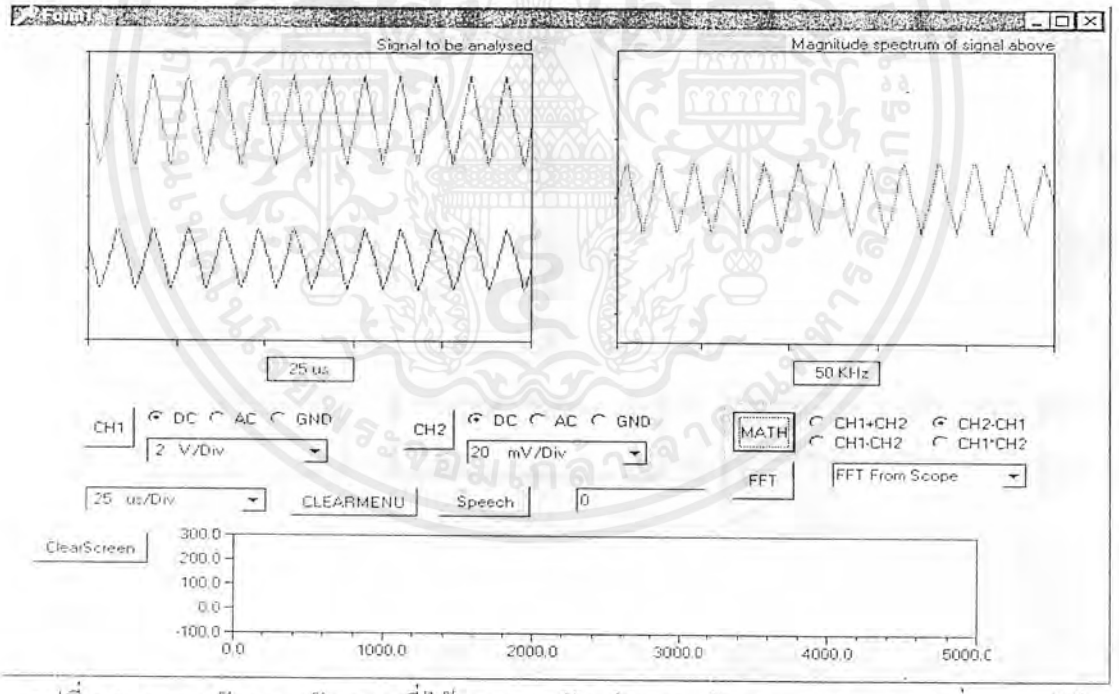
Channel1 Volt/Div = 2 V. , Channel2 Volt/Div = 20 mV.

เลือกฟังก์ชันทางคณิตศาสตร์ CH2 – CH1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.16 แสดงลักษณะสัญญาณที่วัดจากวงจรตัวอย่าง และสัญญาณ CH12 - CH2 ที่ได้จากสโคป



รูปที่ 6.17 แสดงลักษณะสัญญาณที่ได้จากวงจรตัวอย่าง และสัญญาณ CH2 – CH1 ที่ทดลองได้

6.3.4 การทดลองทำ FFT จากสัญญาณเสียงที่ได้อัดไว้แล้ว ซึ่งจะอยู่ในรูปของไฟล์ *.wav โดยจะสามารถกำหนดตำแหน่งที่ต้องการทำ FFT ได้ตามต้องการ

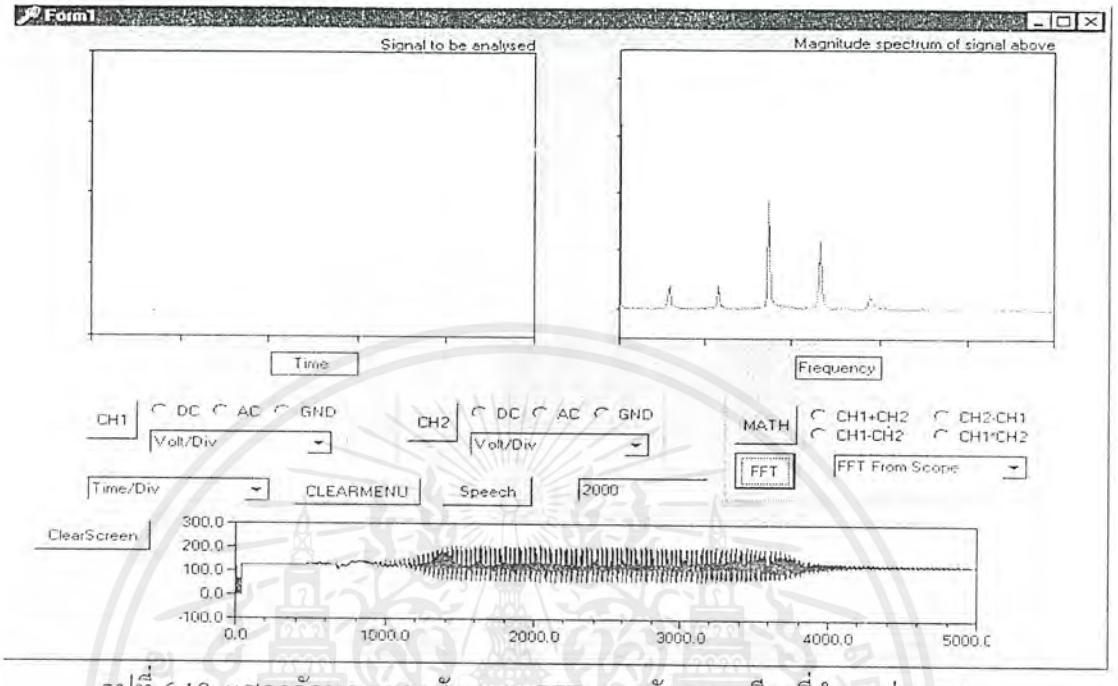
6.3.4.1 เลือกเปิดไฟล์ *.wav ไฟล์หนึ่ง (เสียงคำว่า "สอง")

แสดงการพล็อตสัญญาณ FFT ที่ตำแหน่ง 2000

(สามารถเลือกพล็อตที่ตำแหน่งใดก็ได้ โดยดูลักษณะของสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประกอบการเลือกตำแหน่ง)

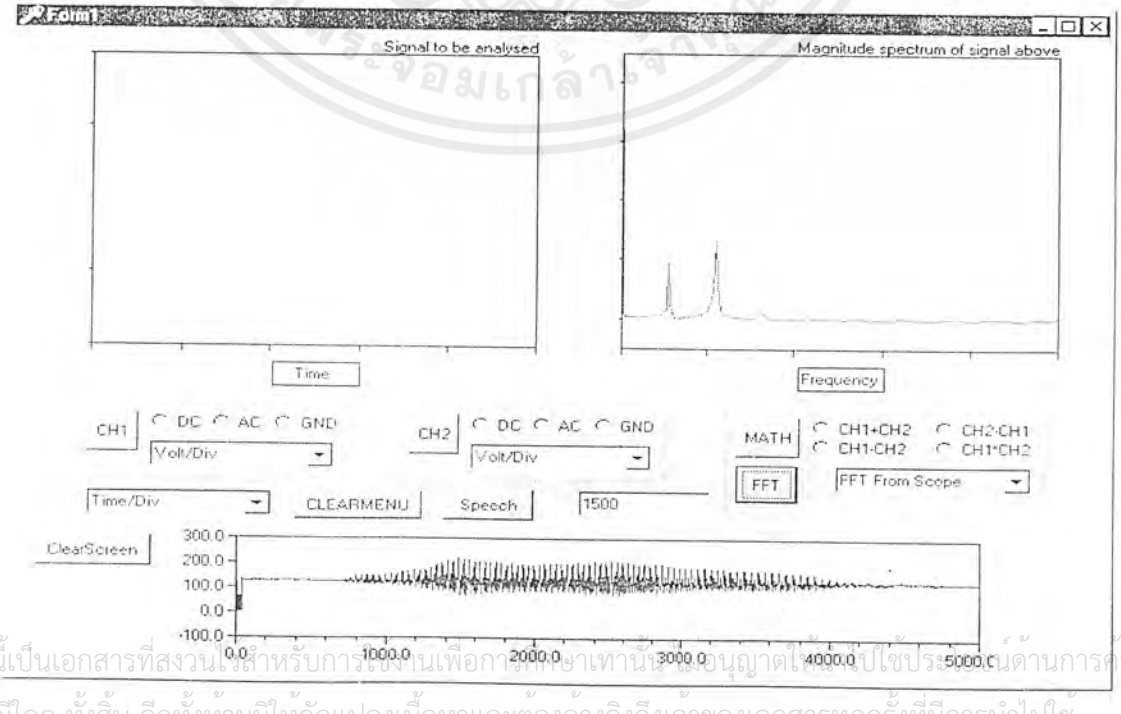


รูปที่ 6.18 แสดงลักษณะของสัญญาณ FFT ของสัญญาณเสียงที่ตำแหน่ง 2000

6.3.4.2 เลือกเปิดไฟล์ *.wav ไฟล์หนึ่ง (เสียงคำว่า "วิว")

แสดงการพล็อตสัญญาณ FFT ที่ตำแหน่ง 1500

(สามารถเลือกพล็อตที่ตำแหน่งใดก็ได้ โดยดูลักษณะของสัญญาณประกอบการเลือกตำแหน่ง)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.19 แสดงลักษณะของสัญญาณ FFT ของสัญญาณเสียงที่ตำแหน่ง 1500

บทที่ 7
สรุปและวิจารณ์ผลการทดลอง

ในการทดสอบเพื่อควบคุมอุปกรณ์ทางอิเล็กทรอนิกส์ ผ่านทาง IEEE-488 (GPIB) ได้แบ่งออกเป็น 2 ส่วน ใหญ่ ๆ คือ

- ส่วนของการควบคุมอุปกรณ์ทางอิเล็กทรอนิกส์
- ส่วนของการนำไปประยุกต์ใช้งาน

พบว่า ในส่วนของการควบคุมอุปกรณ์ ซึ่งได้แก่ Programmable Function Generator, Programmable Power Supply และ Digital Oscilloscope นั้น เราสามารถสั่งงานอุปกรณ์ผ่านทางคอมพิวเตอร์ และรับข้อมูลจากอุปกรณ์เข้ามายังคอมพิวเตอร์ได้ โดยในส่วนของ Programmable Function Generator และProgrammable Power Supply ผลที่ได้จากการทดลองมีความถูกต้อง แม่นยำดี แต่ในส่วนของ Digital Oscilloscope นั้น สามารถสั่งงานอุปกรณ์ได้ผลที่มีความถูกต้อง แม่นยำดีเช่นเดียวกัน ปัญหาที่พบในการทดลองคือ การรับข้อมูลจากสโคปเข้ามายังคอมพิวเตอร์ เนื่องจากระบบไม่เป็น Real time ทำให้การอ่านสัญญาณจากสโคปมาแสดงในคอมพิวเตอร์ จะเกิดดีเลย์(Delay)ขึ้น

ในส่วนของการนำไปประยุกต์ใช้งาน เป็นการวิเคราะห์หาสเปคตรัมของสัญญาณโดยวิธี FFT (Fast Fourier Transform) ซึ่งสัญญาณที่ได้จากการวิเคราะห์สามารถนำไปใช้ในการประมวลผลต่อไปได้ ในการทดลองเราได้นำสัญญาณที่อ่านได้จากสโคป และสัญญาณเสียงที่อัดเป็นไฟล์ไว้แล้ว เข้ามาเพื่อทำการวิเคราะห์ สเปคตรัมโดยวิธี FFT พบว่า ค่าสเปคตรัมที่ได้จากการวิเคราะห์มีความถูกต้อง โดยสามารถเปรียบเทียบผลที่ได้จากการวิเคราะห์โดยคอมพิวเตอร์ กับผลที่ได้จากการวิเคราะห์โดยสโคป ซึ่งจะได้ผลเหมือนกัน

งานวิจัยนี้สามารถนำไปพัฒนาโปรแกรมควบคุมระบบ เพื่อให้ง่ายและสะดวกต่อการใช้งาน โดยอาจนำไปใช้ในการวัดคุณสมบัติของอุปกรณ์อิเล็กทรอนิกส์ หรืออาจนำไปพัฒนาควบคุมอุปกรณ์ผ่านทางอินเตอร์เน็ตได้ รวมทั้งอาจจะนำไปประยุกต์ใช้ในการวิเคราะห์หาสเปคตรัมของสัญญาณ เพื่อนำข้อมูลที่ได้ไปใช้งานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันในการติดต่อกับ IEEE-488.2M

1. **Tibclr = Function (ud : integer) : integer ; stdcall ;**

- Tibclr คือ การเคลียร์สถานะของอุปกรณ์
- Ud คือ การบอกตำแหน่งชนิดของอุปกรณ์

2. **Tibdev = Function (ud : integer ;**
pad : integer ;
sad : integer ;
tmo : integer ;
eot : integer ;
eos : integer) ; integer ; stdcall ;

- Tibdev คือ การติดต่อและกำหนดค่าเริ่มต้นให้กับอุปกรณ์
- Ud คือ การบอกตำแหน่งชนิดของอุปกรณ์
- Pad คือ การบอกตำแหน่งหลัก GPIB ของอุปกรณ์
- Sad คือ การบอกตำแหน่งสำรอง GPIB ของอุปกรณ์
- Eot คือ การบอกตำแหน่งสิ้นสุดในการติดต่อ
- Eos คือ การบอกตำแหน่งข้อความสุดท้าย

3. **Tibrd = Function (ud : integer ;**
var rdbuf ;
cnt : longint) : integer ; stdcall ;

- Tibrd คือ การอ่านข้อมูลจากอุปกรณ์เข้ามาเก็บไว้ในหน่วยความจำ
- Ud คือ การบอกตำแหน่งชนิดของอุปกรณ์
- Rdbuf คือ ชื่อของหน่วยความจำที่ใช้เก็บอุปกรณ์
- Cnt คือ การกำหนดขนาดของการอ่านข้อมูล

4. **Tibtrg = Function (ud : integer) : integer ; stdcall ;**

- Tibtrg คือ การเลือกชนิดของอุปกรณ์
- Ud คือ การบอกตำแหน่งชนิดของอุปกรณ์

5. **Tibwrt = Function (ud : integer ;**
var wrtbuf ;
cnt :longint) : integer ; stdcall;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Tibwrtคือ การส่งข้อมูลจากผู้ใช้ไปให้อุปกรณ์
- Ud คือ การบอกตำแหน่งชนิดของอุปกรณ์
- Wrtbuf คือ ชื่อของหน่วยความจำที่ใช้ในการส่งข้อมูล
- Cnt คือ การกำหนดขนาดของการส่งข้อมูล

6. **Tibwait = Function (ud : integer;**

mask : integer) : integer ; stdcall ;

- Tibwait คือ การรอเวลาในการติดต่อหรือการดำเนินงานของ GPIB
- Ud คือ การบอกตำแหน่งชนิดของอุปกรณ์
- Mask คือ บิตแสดงสถานะของ GPIB ที่บอกถึงเหตุการณ์ในการรอคอยในการติดต่อกับ GPIB

7. **TdevClear = procedure(boardID : integer ;**

address : word) ; stdcall ;

- TdevClear คือ การเคลียร์สถานะของอุปกรณ์
- BoardID คือ การบอกตำแหน่งของบอร์ด ที่จะทำการติดต่อ
- Address คือ ตำแหน่งของอุปกรณ์ที่ต้องการเคลียร์

8. **TfindLstn = procedure(boardID : integer ;**

padlist : AddrList ;

resultlist : AddrList ;

limit : integer) ; stdcall ;

- TfindLstn คือ การหาการตอบรับของอุปกรณ์บนบัส GPIB
- BoardID คือ การบอกตำแหน่งของบอร์ด GPIB
- Padlist คือ รายการของตำแหน่งหลัก
- Resultlist คือ ผลลัพธ์ที่ได้จากการตรวจสอบทั้งหมด จะเก็บข้อมูลในรูปแบบของ array
- Limit คือ จำนวนที่ต้องการเก็บไว้ใน resultlist

9. **Tibonl = Function (ud : integer ;**

v : integer) : integer ; stdcall ;

- Tibonl คือ การแสดงถึงอุปกรณ์ หรือ GPIB ว่าทำงานหรือไม่ทำงาน
- Ud คือ การแสดงถึงบอร์ดหรืออุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- V คือ ถ้ามีค่าเป็น "1"จะเป็นการบอกว่าบอร์ดหรืออุปกรณ์นั้นทำงาน และถ้ามีค่าเป็น "0" คือไม่ทำงาน

10. **TreadStatusByte = procedure (boardId : integer ;**
address : word ;
var result : integer) ; stdcall ;

- TreadStatusByte คือ การอ่านสถานะของอุปกรณ์
- BoardID คือ การบอกตำแหน่งการติดต่อ GPIB
- Address คือ การบอกตำแหน่งของอุปกรณ์
- Result คือ การกำหนดตำแหน่งในการเก็บข้อมูลในรูปของไบท์

11. **TReceive = procedure (boardID : integer ;**
address : word ;
var buffer ;
cnt : longint ;
termination : integer) ; stdcall ;

- TReceive คือ การอ่านข้อมูลจากอุปกรณ์
- BoardID คือ การบอกตำแหน่งการติดต่อ GPIB
- Address คือ การบอกตำแหน่งของอุปกรณ์
- Buffer คือ ชื่อของตำแหน่งที่ใช้ในการเก็บข้อมูล
- Cnt คือ ความยาวในการอ่านข้อมูล

12. **Tsend = procedure (boardID : integer ;**
address : word ;
var buffer ;
datacnt : longint ;
eotmode : integer) ; stdcall ;

- Tsend คือ การส่งข้อมูลไปให้อุปกรณ์
- BoardID คือ การบอกตำแหน่งการติดต่อ GPIB
- Address คือ การบอกตำแหน่งของอุปกรณ์
- Buffer คือ ชื่อของตำแหน่งที่ใช้ในการส่งข้อมูล
- Datacnt คือ ความยาวในการส่งข้อมูล
- Eotmode คือ การบอกตำแหน่งสุดท้ายในการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. **TSendIFC = procedure (boardID : integer) ; stdcall ;**

- TSendIFC คือ การรีเซ็ท GPIB โดยการส่งสัญญาณ interfacing เคลียร์
- BoardID คือ การบอกตำแหน่งการติดต่อ

14. **TWaitSRQ = procedure (boardID : integer ;**

var result : integer) ; stdcall ;

- TWaitSRQ คือ การคอยจนกระทั่งอุปกรณ์ยืนยันพร้อมที่จะขอบริการจากGPIB
- BoardID คือ การบอกตำแหน่งการติดต่อ GPIB
- Result คือ สถานะของผลลัพธ์จะเป็น "1" เมื่อมีการยืนยัน และจะเป็น "0" เมื่อไม่มีการยืนยัน

ค่าคงที่ที่ต้องประกาศในการเขียนโปรแกรม

```
CONST
(* GPIB status bit definitions. *)
ERR     = $8000;    (*Error detected                    *)
TIMO    = $4000;    (*Timeout                           *)
ENDgpib = $2000;    (*EOI or EOS detected               *)
SRQI    = $1000;    (*SRQ detected by CIC               *)
RQS     = $800;     (*Device needs service              *)
SPOLL   = $400;     (*Board has been serially polled    *)
EVENT   = $200;     (*An event has occurred             *)
CMPL    = $100;     (*I/O completed                     *)
LOK     = $80;      (*Local lockout state               *)
REM     = $40;      (*Remote state                      *)
CIC     = $20;      (*Controller-in-charge              *)
ATN     = $10;      (*Attention asserted                *)
TACS    = $8;       (*Talker active                     *)
LACS    = $4;       (*Listener active                   *)
DTAS    = $2;       (*Device trigger state              *)
DCAS    = $1        (*Device clear state                *)

(* Error messages returned in global variable IBERR:*)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
EDVR  = 0;      (*System error                          *)
ECIC  = 1;      (*Function requires GPIB board to CIC   *)
ENOL  = 2;      (*Write function detected no Listeners  *)
EADR  = 3;      (*Interfac board not addressed correctly *)
EARG  = 4;      (*Invalid argument to function call     *)
ESAC  = 5;      (*Function require GPIB board to be SAC *)
EABO  = 6;      (*I/O operation aborted                 *)
ENEB  = 7;      (*Non-existent interface board          *)
EDMA = 8;       (*Error performing DMA                  *)
EOIP  = 9;      (*I/O operation started before previous *)
                (* operation complete *)
ECAP  = 11;     (*No capability for intened operation   *)
EFSO  = 12;     (*File system operation error           *)
EBUS  = 14;     (*Command error during device call      *)
ESTB  = 15;     (*Serial poll status byte lost          *)
ESRQ = 16;      (* SRQ remains asserted                  *)
ETAB = 20;      (*The return buffer is full             *)
```

PROGRAMMABLE POWER SUPPLY

<u>คำสั่งควบคุมฟังก์ชันในการทำงานผ่านทางระบบบัสแบบ IEEE-488 คือ</u>

คำสั่ง = RM1 และ RM0

รูปแบบ : RM1

ฟังก์ชัน : สั่งให้ Power Supply ทำงานในโหมดรีโมท โดยไม่สามารถควบคุมผ่านทางหน้าปัทม์ได้ แต่สามารถควบคุมได้โดยผ่านทางระบบบัสแบบ IEEE-488 เท่านั้น (เราสามารถออกจากโหมดรีโมทได้โดยการกดปุ่ม LOCAL ที่หน้าปัทม์ของเครื่อง หรือใช้การส่งคำสั่ง RM0)

รูปแบบ : RM0

ฟังก์ชัน : สั่งให้ยกเลิกโหมดรีโมทให้เครื่องกลับสู่สถานะLOCAL ซึ่งก็คือควบคุมการทำงานของเครื่องทางหน้าปัทม์เครื่อง

<u>ข้อสังเกต</u> คำสั่ง RM0 จะมีผลต่อคำสั่ง LK1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง = LK1 และ LK0

รูปแบบ : LK1

ฟังก์ชัน : สั่งให้เครื่องเข้าสู่โหมดรีโมท โดยไม่สามารถกดปุ่ม LOCAL เพื่อที่จะให้เครื่องกลับเข้าสู่โหมด LOCAL

รูปแบบ : LK0

ฟังก์ชัน : สั่งให้เครื่องออกจากโหมดรีโมทขณะที่ไม่สามารถกดปุ่ม LOCAL ได้ให้กดปุ่ม LOCAL ได้เพื่อให้ควบคุมเครื่องผ่านทางหน้าปัทม์ของเครื่องได้

ข้อสังเกต โหมดรีโมทที่ไม่สามารถกดปุ่ม LOCAL ได้นี้ จะถูกยกเลิกได้โดยคำสั่ง RM0

คำสั่ง = MX1 และ MX0

รูปแบบ : MX1

ฟังก์ชัน : สั่งให้เปลี่ยนจากโหมดรีโมทไปเป็นโหมดแบบผสม ในโหมดนี้สามารถที่จะควบคุมการทำงานของเครื่องได้ทั้งสองทางคือ ผ่านทางระบบบัสแบบ IEEE-488 และควบคุมผ่านทางหน้าปัทม์

รูปแบบ : MX0

ฟังก์ชัน : ยกเลิกโหมดแบบผสมและกลับเข้าสู่สถานะปกติที่ควบคุม โดยผ่านทางระบบบัสแบบ IEEE-488

คำสั่ง = SU1 และ SU2

รูปแบบ : SU1 (หรือ SU2) W.mV.mV เช่น SU1 01.34

ฟังก์ชัน : ตั้งค่าศักดาไฟฟ้าให้แหล่งจ่ายศักดาไฟฟ้าตัวที่ 1 และ 2

คำสั่ง = SI1 และ SI2

รูปแบบ : SI1 (หรือ SI2) W.mA.mA เช่น SI1 0.123

ฟังก์ชัน : ตั้งค่ากระแสไฟฟ้าให้แหล่งจ่ายกระแสไฟฟ้าตัวที่ 1 และ 2

คำสั่ง = RU1 RU2

รูปแบบ : RU1 (หรือ RU2)

ฟังก์ชัน : อ่านค่าศักดาไฟฟ้าที่ส่งกลับโดยเครื่อง จะเป็นค่าศักดาไฟฟ้าที่ถูกกำหนดค่าโดยคำสั่งกำหนดค่า (SET)

ข้อสังเกต ใช้คำสั่ง MUX เพื่อที่จะถามว่าค่าศักดาไฟฟ้าจริงที่วัดได้จากเอาท์พุท

คำสั่ง = RI1 หรือ RI2

รูปแบบ : RI1 หรือ RI2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน : อ่านค่ากระแสไฟฟ้าที่ส่งกลับโดยเครื่อง จะเป็นค่ากระแสไฟฟ้าที่ถูกกำหนดค่าโดยคำสั่งจำกัดค่ากระแสไฟฟ้า (SI1 หรือ SI2)

ตัวอย่างค่าที่ส่งกลับ เช่น I1_1.000A หรือ I2_0.01A

ข้อสังเกต ใช้คำสั่ง MIX เพื่อที่จะถามว่าค่ากระแสไฟฟ้าจริงที่วัดได้จากเอาท์พุท

คำสั่ง = MU1 หรือ MU2

รูปแบบ : MU1 หรือ MU2

ฟังก์ชัน : ค่าศักดาไฟฟ้าที่ส่งกลับโดยเครื่อง ซึ่งเป็นค่าที่วัดได้จากเอาท์พุทจริง

ตัวอย่างค่าที่ส่งกลับ เช่น U1:12.34V หรือ U2:12.34V

ข้อสังเกต คำสั่ง RUX ใช้เพื่อถามถึงค่าศักดาไฟฟ้าที่ทำการกำหนด

คำสั่ง = MI1 หรือ MI2

รูปแบบ : MI1 หรือ MI2

ฟังก์ชัน : ค่ากระแสไฟฟ้าที่ส่งกลับโดยเครื่อง ซึ่งเป็นค่าที่วัดได้จากเอาท์พุทจริง

ตัวอย่างค่าที่ส่งกลับ เช่น I1: +1.000A หรือ I2: -0.12A

ข้อสังเกต คำสั่ง RIX ใช้เพื่อถามถึงค่ากระแสไฟฟ้าที่ทำการจำกัดเอาไว้ ถ้าเอาท์พุทถูกปิดไว้ค่าที่ส่งกลับจะเป็น I1=_1.000A

คำสั่ง = TRU

รูปแบบ : TRU

ฟังก์ชัน : กำหนดค่าศักดาไฟฟ้าของแหล่งจ่ายศักดาไฟฟ้าตัวที่ 1 และ 2 ให้มีค่าเท่ากันตามต้องการ

คำสั่ง = TRI

รูปแบบ : TRI

ฟังก์ชัน : กำหนดค่ากระแสไฟฟ้าของแหล่งจ่ายกระแสไฟฟ้าตัวที่ 1 และ 2 ให้มีค่าเท่ากันตามต้องการ

คำสั่ง = SR1 และ SR0

รูปแบบ : SR1

ฟังก์ชัน : ทำการ enable โหมด Service Request โดยไม่มีการเปลี่ยนแปลงใดๆ ในสถานะของอุปกรณ์ที่จะมีผลต่อสัญญาณ SRQ (Service Request)

รูปแบบ : SR0

ฟังก์ชัน : ทำการยกเลิกโหมด Service Request

คำสั่ง = STA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบ : STA

ฟังก์ชัน : สั่งให้เครื่องส่งค่าที่เป็น string ที่เก็บสถานะปัจจุบันของเครื่องมาให้

ตัวอย่างค่าที่ส่งกลับ เช่น OP1/0SR1/0ER0/1CV1/CC1/CV2/RM0/1

OP0 สวิทช์เอาท์พุทปิด

OP1 สวิทช์เอาท์พุทเปิด

SR0 เมื่อสัญญาณ service request (SRQ) ถูกทำให้ enable แสดงว่าไม่มีการเปลี่ยนแปลงสถานะของเครื่องมือ

SR1 แสดงสถานะของอุปกรณ์ที่เปลี่ยนแปลง (CV ไปเป็น CC หรือ OP1 ไปเป็น OP0 เป็นต้น) (จะทำงานเฉพาะเมื่อ SRQ นั้น enable หรือมีค่าเป็น 1 , ดูคำสั่ง SR1)

ER0 ไม่มีความผิดพลาด

ER1 ความร้อนสูงเกิน

CV1 แหล่งจ่ายศักดาไฟฟ้าตัวที่ 1 ทำการจ่ายศักดาไฟฟ้าคงที่

CC1 แหล่งจ่ายกระแสไฟฟ้าตัวที่ 1 ทำการจ่ายกระแสไฟฟ้าคงที่

CV2 แหล่งจ่ายศักดาไฟฟ้าตัวที่ 2 ทำการจ่ายศักดาไฟฟ้าคงที่

CC2 แหล่งจ่ายกระแสไฟฟ้าตัวที่ 2 ทำการจ่ายกระแสไฟฟ้าคงที่

RM0 อุปกรณ์ไม่อยู่ในโหมดรีโมท

RM1 อุปกรณ์อยู่ในโหมดรีโมท

คำสั่ง = OP1 และ OP0

รูปแบบ : OP1

ฟังก์ชัน : สวิทช์เอาท์พุทเปิด

รูปแบบ : OP0

ฟังก์ชัน : สวิทช์เอาท์พุทปิด

คำสั่ง = CLR (clear)

รูปแบบ : CLR

ฟังก์ชัน : ยกเลิกฟังก์ชันทั้งหมดของเครื่อง และทำการเริ่มต้นใหม่ที่สถานะศูนย์ (zero status) ในโหมดรีโมท คีร์บอร์ดไม่สามารถใช้งานได้และสวิทช์เอาท์พุทปิด รวมทั้งศักดาไฟฟ้าและกระแสไฟฟ้าจะถูกตั้งค่าให้เป็นศูนย์

คำสั่ง = VER

รูปแบบ : VER

ฟังก์ชัน : แสดงเวอร์ชั่นของ Software และ Hardware ของเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างค่าที่ส่งกลับ เช่น sw Vx.xhw Vx.xxxxxxxx HAMEG/PARIS KRP&VM

คำสั่ง = ID?

รูปแบบ : ID?

ฟังก์ชัน : แสดงถึงชื่อรุ่นของเครื่อง

ตัวอย่างค่าที่ส่งกลับ เช่น HM8142-1

PROGRAMMABLE FUNCTION GENERATOR

<u>คำสั่งควบคุมฟังก์ชันในการทำงานผ่านระบบบัสแบบ IEEE-488 คือ</u>

<u>คำสั่งที่ไม่ต้องมีข้อมูลประกอบ</u>

คำสั่ง = SIN

รูปแบบ : SIN

ฟังก์ชัน : ฟังก์ชันการสร้างสัญญาณคลื่นรูปซายน์

คำสั่ง = TRI

รูปแบบ : TRI

ฟังก์ชัน : ฟังก์ชันการสร้างสัญญาณคลื่นรูปสามเหลี่ยม

คำสั่ง = SQR

รูปแบบ : SQR

ฟังก์ชัน : ฟังก์ชันการสร้างสัญญาณคลื่นรูปสี่เหลี่ยม

คำสั่ง = PLS

รูปแบบ : PLS

ฟังก์ชัน : ฟังก์ชันการสร้างสัญญาณคลื่นรูปพัลซ์

คำสั่ง = RMS

รูปแบบ : RMS

ฟังก์ชัน : ฟังก์ชันการสร้างสัญญาณคลื่นรูปฟันเลื่อย ที่มีสัญญาณขาขึ้นเป็นบวก

คำสั่ง = RMN

รูปแบบ : RMN

ฟังก์ชัน : ฟังก์ชันการสร้างสัญญาณคลื่นรูปฟันเลื่อย ที่มีสัญญาณขาขึ้นเป็นลบ

คำสั่ง = ARB

รูปแบบ : ARB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน : ฟังก์ชันการสร้างสัญญาณคลื่นรูปแบบ arbitary

คำสั่ง = SW1 และ SW0

รูปแบบ : SW1 หรือ SW0

ฟังก์ชัน : การเลือกการกวาดสัญญาณเปิด หรือปิด

คำสั่ง = CTM

รูปแบบ : CTM

ฟังก์ชัน : การเลือกโหมดการสร้างสัญญาณแบบต่อเนื่อง

คำสั่ง = GTM

รูปแบบ : GTM

ฟังก์ชัน : การเลือกโหมดการสร้างสัญญาณแบบ GATED

คำสั่ง = TRM

รูปแบบ : TRM

ฟังก์ชัน : การเลือกโหมดการสร้างสัญญาณแบบ TRIGGER

คำสั่ง = OT1 และ OT0

รูปแบบ : OT1 หรือ OT0

ฟังก์ชัน การเลือกให้เอาท์พุทเปิด หรือปิด

คำสั่ง = OF1 และ OF0

รูปแบบ : OF1 หรือ OF0

ฟังก์ชัน การเลือกให้ค่าออฟเซ็ทเปิด หรือปิด

คำสั่ง = DFR

รูปแบบ : DFR

ฟังก์ชัน : ให้แสดงความถี่ของสัญญาณ

คำสั่ง = DST

รูปแบบ : DST

ฟังก์ชัน : ให้แสดงความถี่เริ่มต้นของสัญญาณ

คำสั่ง = DSP

รูปแบบ : DSP

ฟังก์ชัน : ให้แสดงความถี่สุดท้ายของสัญญาณ

คำสั่ง = DWT

รูปแบบ : DWT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน : ให้แสดงความกว้างของสัญญาณพัลซ์

คำสั่ง = DSW

รูปแบบ : DSW

ฟังก์ชัน : ให้แสดงเวลาในกวาดสัญญาณ

คำสั่ง = DAM

รูปแบบ : DAM

ฟังก์ชัน : ให้แสดงค่าแอมปลิจูดของเอาท์พุท

คำสั่ง = DOF

รูปแบบ : DOF

ฟังก์ชัน : ให้แสดงค่าออฟเซ็ท

คำสั่ง = RM0

รูปแบบ : RM0

ฟังก์ชัน : ยกเลิกโหมดรีโมท แล้วกลับสู่การควบคุมการทำงานผ่านทางหน้าปัทม์ของเครื่อง เงื่อนไขนี้สามารถทำได้โดยการกดปุ่ม LOCAL ก็ได้

คำสั่ง = RM1

รูปแบบ : RM1

ฟังก์ชัน : การตั้งให้เครื่องอยู่ในโหมดรีโมท และสามารถยกเลิกได้โดยการกดปุ่ม LOCAL เช่นกัน

คำสั่ง = LK1

รูปแบบ : LK1

ฟังก์ชัน : ยกเลิกปุ่ม LOCAL ชั่วคราว โดยที่สถานะนี้การควบคุมการทำงานของเครื่องทำได้โดยผ่านทางระบบบัสเท่านั้น และการกลับสู่สถานะ LOCAL ไม่สามารถทำได้โดยการกดปุ่ม LOCAL

คำสั่ง = LK0

รูปแบบ : LK0

ฟังก์ชัน : ยกเลิกสถานะการยกเลิกปุ่ม LOCAL ชั่วคราว ทำให้สามารถกดปุ่ม LOCAL เพื่อที่จะกลับสู่สถานะ LOCAL ได้ ทำให้สามารถควบคุมการทำงานของเครื่องผ่านทางหน้าปัทม์ได้อีกครั้ง

คำสั่ง = TRG

รูปแบบ : TRG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน : TRIG คาบของสัญญาณ(ปิดการกวาดสัญญาณ) หรือการกวาดสัญญาณที่สมบูรณ์ (การกวาดสัญญาณเปิด)

คำสั่ง = CLR

รูปแบบ : CLR

ฟังก์ชัน : รีเซ็ทเครื่องและกลับสู่สถานะเริ่มต้น คำสั่ง CLR เหมือนกับคำสั่ง SDC ตามมาตรฐานของ IEEE-488

คำสั่ง = ARC

รูปแบบ : ARC

ฟังก์ชัน : ลบข้อมูลที่เป็นการสร้างสัญญาณแบบใดๆ และรีเซ็ทตัวนับภายในให้มีค่าเป็นศูนย์

คำสั่ง = ARE

รูปแบบ : ARE

ฟังก์ชัน : ยกเลิกการเขียนข้อมูลที่ใช้สร้างสัญญาณแบบ arbitary

<u>คำสั่งที่ประกอบด้วยข้อมูลที่เป็นเลขทศนิยม</u>

คำสั่ง = FRQ

รูปแบบ : FRQ:< ข้อมูล >

ฟังก์ชัน : กำหนดค่าความถี่ที่ต้องการ < > Hz

คำสั่ง = STT

รูปแบบ : STT:< ข้อมูล >

ฟังก์ชัน : กำหนดค่าความถี่เริ่มต้น < > Hz

คำสั่ง = STP

รูปแบบ : STP:< ข้อมูล >

ฟังก์ชัน : กำหนดค่าความถี่สุดท้าย < > Hz

คำสั่ง = SWT

รูปแบบ : SWT:< ข้อมูล >

ฟังก์ชัน : กำหนดค่าเวลาในการกวาดสัญญาณ < > s

คำสั่ง = WDT

รูปแบบ : WDT:< ข้อมูล >

ฟังก์ชัน : กำหนดค่าความกว้างของสัญญาณพัลซ์ < > s

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง = AMP

รูปแบบ : AMP:< ข้อมูล >

ฟังก์ชัน : กำหนดค่าแอมปลิจูดของสัญญาณ < > V

คำสั่ง = OFS

รูปแบบ : OFS:< ข้อมูล >

ฟังก์ชัน : กำหนดค่าออฟเซ็ทของสัญญาณ < > V

หมายเหตุ ค่าความถี่และเวลากำหนดได้มากที่สุด 5 หลัก และค่าศักดาไฟฟ้ากำหนดได้มากที่สุด 3 หลัก

ค่าแอมปลิจูดสามารถกำหนดได้ 2 วิธี ถ้าเป็นค่าที่ไม่มีเครื่องหมายจะถือว่าเป็นค่าศักดาไฟฟ้าแบบ peak to peak (ในกรณีของสัญญาณรูปพัลส์นั้น ศักดาไฟฟ้าสูงสุดจะมีค่าเท่ากับครึ่งหนึ่งของค่านี้) ถ้าเป็นค่าที่มีเครื่องหมายนำหน้าจะถือว่าเป็นค่าศักดาไฟฟ้าสูงสุด

คำสั่งร้องขอ

คำสั่ง = FRQ?

รูปแบบ : FRQ?

ฟังก์ชัน : แสดงความถี่ของสัญญาณ

คำสั่ง = STT?

รูปแบบ : STT?

ฟังก์ชัน : แสดงความถี่เริ่มต้นของสัญญาณ

คำสั่ง = STP?

รูปแบบ : STP?

ฟังก์ชัน : แสดงความถี่สุดท้ายของสัญญาณ

คำสั่ง = SWT?

รูปแบบ : SWT?

ฟังก์ชัน : แสดงเวลาในการกวาดสัญญาณ

คำสั่ง = WDT?

รูปแบบ : WDT?

ฟังก์ชัน : แสดงความกว้างของสัญญาณพัลซ์

คำสั่ง = AMP?

รูปแบบ : AMP?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน : แสดงค่าแอมปลิจูดของศักดาไฟฟ้าเอาท์พุท

คำสั่ง = OFS?

รูปแบบ : OFS?

ฟังก์ชัน : แสดงค่าออฟเซ็ท

คำสั่ง = ARD?

รูปแบบ : ARD?

ฟังก์ชัน : แสดงข้อมูลของสัญญาณแบบ arbitary

คำสั่ง = ID?

รูปแบบ : ID?

ฟังก์ชัน : แสดงหมายเลขรุ่นของเครื่อง

คำสั่ง = VER

รูปแบบ : VER

ฟังก์ชัน : แสดงเวอร์ชั่นของ Software และ Hardware ของเครื่อง

คำสั่ง = STA?

รูปแบบ : STA?

ฟังก์ชัน : แสดงสถานะของเครื่อง สตริงข้อมูลที่บอกสถานะของเครื่องอ่านได้ทั้งหมด 21 ตัวอักษร มีความหมายดังนี้

OT0OF0SW0SINCTMDFRDAM

- OT0 สวิทช์เอาท์พุทปิด (ถ้าเป็น OT1 จะเปิด)
- OF0 การเซ็ทค่าออฟเซ็ทปิด (ถ้าเป็น OF1 จะเปิด)
- SW0 การกวาดสัญญาณปิด (ถ้าเป็น SW1 จะเปิด)
- SIN รูปแบบของสัญญาณซายน์ (เปลี่ยนเป็นสัญญาณรูปอื่นๆได้)
- CTM โหมดการทำงานซึ่งสร้างสัญญาณแบบต่อเนื่อง (เปลี่ยนโหมดการทำงานได้)
- DFR แสดงรายละเอียดความถี่ (เปลี่ยนรายละเอียดอย่างอื่นได้)
- DAM แสดงรายละเอียดแอมปลิจูด (เปลี่ยนรายละเอียดอย่างอื่นได้)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAMMABLE OSCILLOSCOPE

<u>คำสั่งควบคุมฟังก์ชันในการทำงานผ่านระบบบัสแบบ IEEE-488 คือ</u>

Digital Oscilloscope ของบริษัท Tektronix รุ่น TDS360 ทำหน้าที่ในการวัดสัญญาณที่มีค่าความถี่ได้ถึง 200 MHz โดย Oscilloscope นี้ สามารถที่จะทำการบันทึกสัญญาณที่ได้จากการวัดมาเก็บไว้ในหน่วยความจำของเครื่อง และสามารถที่จะอ่านค่าของสัญญาณที่บันทึกเก็บไว้ผ่านระบบบัสแบบ IEEE-488 ได้ และสามารถที่จะควบคุมช่วงเวลาต่อช่อง (Time per Division) และจำนวนช่องต่อโวล์ท (Volt per Division) ได้ สำหรับคำสั่งที่ใช้ควบคุมการทำงานผ่านระบบบัส IEEE-488 มี ดังนี้ คือ

คำสั่ง = CH<X> : COUPLING {AC/DC/GND}

รูปแบบ : CH1:COUPLING AC

ฟังก์ชั่น : เลือกให้ Channel 1 เป็นสัญญาณ AC Coupling

คำสั่ง = CH<X> : VOLTS <NR3>

รูปแบบ : CH1:VOLTS 1.0E-1

ฟังก์ชั่น : ตั้งให้ Channel 1 มีค่าแรงดันเท่ากับ 100 mV/D

คำสั่ง = HORizontal:Main:Scale <NR3>

รูปแบบ : HORizontal:Main:Scale 2E-6

ฟังก์ชั่น : ตั้งค่า Scale ให้เป็น 2 us/D

คำสั่ง = CLEARMenu

รูปแบบ : CLEARMenu

ฟังก์ชั่น : เคลียร์ตัวเลือกต่าง ๆ ในเมนูบนหน้าจอ

คำสั่ง = CURVE?

รูปแบบ : CURVE?

ฟังก์ชั่น : แปลงคลื่นสัญญาณเป็นข้อมูลตัวเลข

คำสั่ง = DATA : START <NR1>

รูปแบบ : DATA : START <10>

ฟังก์ชั่น : ให้เริ่มอ่านข้อมูลของคลื่นสัญญาณตำแหน่งที่ 10

คำสั่ง = DATA : STOP <NR1>

รูปแบบ : DATA : STOP <150>

ฟังก์ชั่น : ให้หยุดอ่านข้อมูลของคลื่นสัญญาณตำแหน่งที่ 150

คำสั่ง = DATA : SOURCE <wfm>[<comma><wfm>]...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบ : DATA : SOURCE CH1

ฟังก์ชั่น : อ่านสัญญาณจากข้อมูล channel 1

คำสั่ง = DATA : ENCdg {ASCII/RIbinary/SRIbinary/SRPbinary}

รูปแบบ : ENCdg RPbinary

ฟังก์ชั่น : ส่งค่าสัญญาณข้อมูลออกมาเป็นไบนารี (0 – 255)

คำสั่ง = DATA : WIDTH <NR1>

รูปแบบ : DATA : WIDTH 1

ฟังก์ชั่น : ตั้งค่าความกว้างข้อมูลที่จะแปลงเป็นข้อมูลตัวเลขให้มีค่าความกว้าง 1 ไบต์ ต่อ 1 ตำแหน่งข้อมูล

คำสั่ง = WFMPre:<wfm>

รูปแบบ : WFMPre:CH1?

ฟังก์ชั่น : อ่านข้อมูลจาก channel 1 ก่อนที่จะถูกแปลงเป็นข้อมูลตัวเลข

คำสั่ง = MATH1:DEFINE{<Qstring>}

รูปแบบ : MATH1:DEFINE "CH1 + CH2"

ฟังก์ชั่น : เลือกฟังก์ชั่นคณิตศาสตร์ channel 1 + channel 2

รูปแบบ : MATH1:DEFINE "CH1 - CH2"

ฟังก์ชั่น : เลือกฟังก์ชั่นคณิตศาสตร์ channel 1 - channel 2

รูปแบบ : MATH1:DEFINE "CH2 – CH1"

ฟังก์ชั่น : เลือกฟังก์ชั่นคณิตศาสตร์ channel 2 - channel 1

รูปแบบ : MATH1:DEFINE "CH1 * CH2"

ฟังก์ชั่น : เลือกฟังก์ชั่นคณิตศาสตร์ channel 1 * channel 2

รูปแบบ : MATH1:DEFINE "FFT(CH1)"

ฟังก์ชั่น : เลือกฟังก์ชั่นคณิตศาสตร์ Fast Fourier transform ของ channel 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FLOW CHART

การทำงานของฟังก์ชันเจนเนอร์เรเตอร์

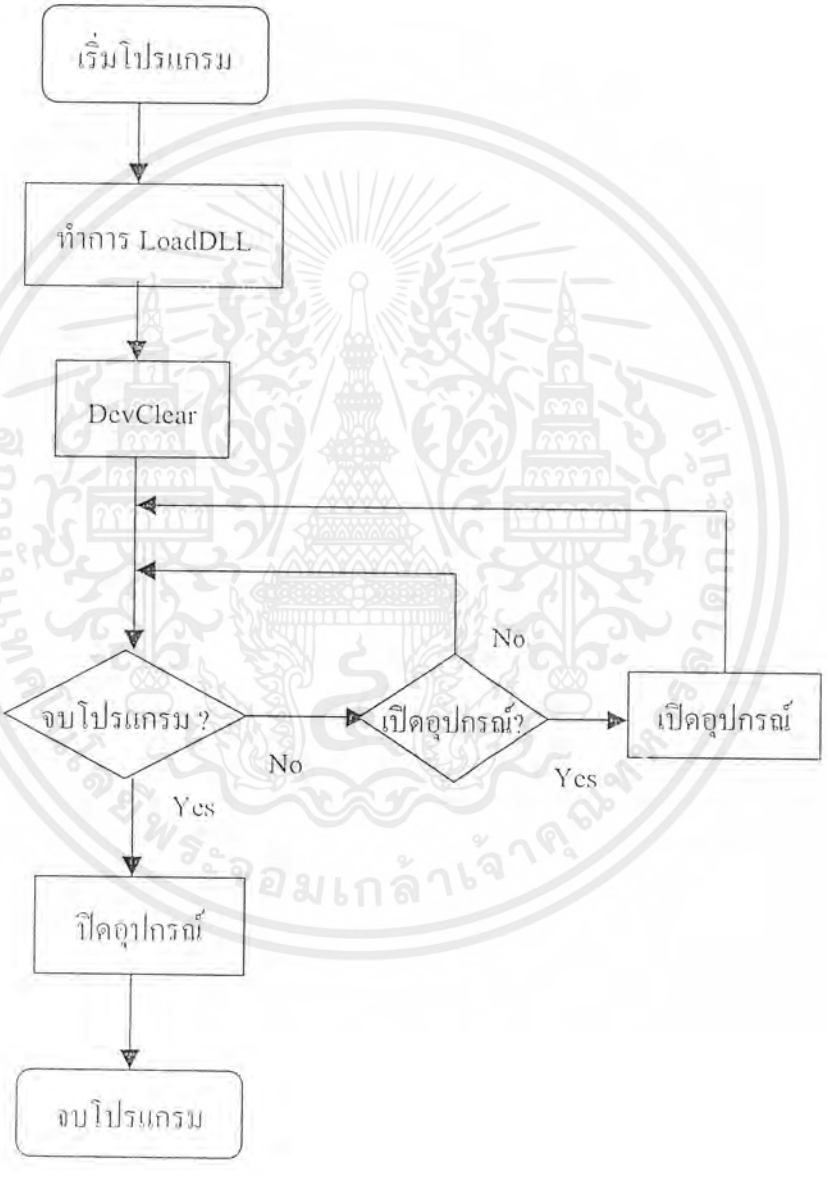
เพาเวอร์ซัพพลาย

และออสซิลโลสโคป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมหลัก

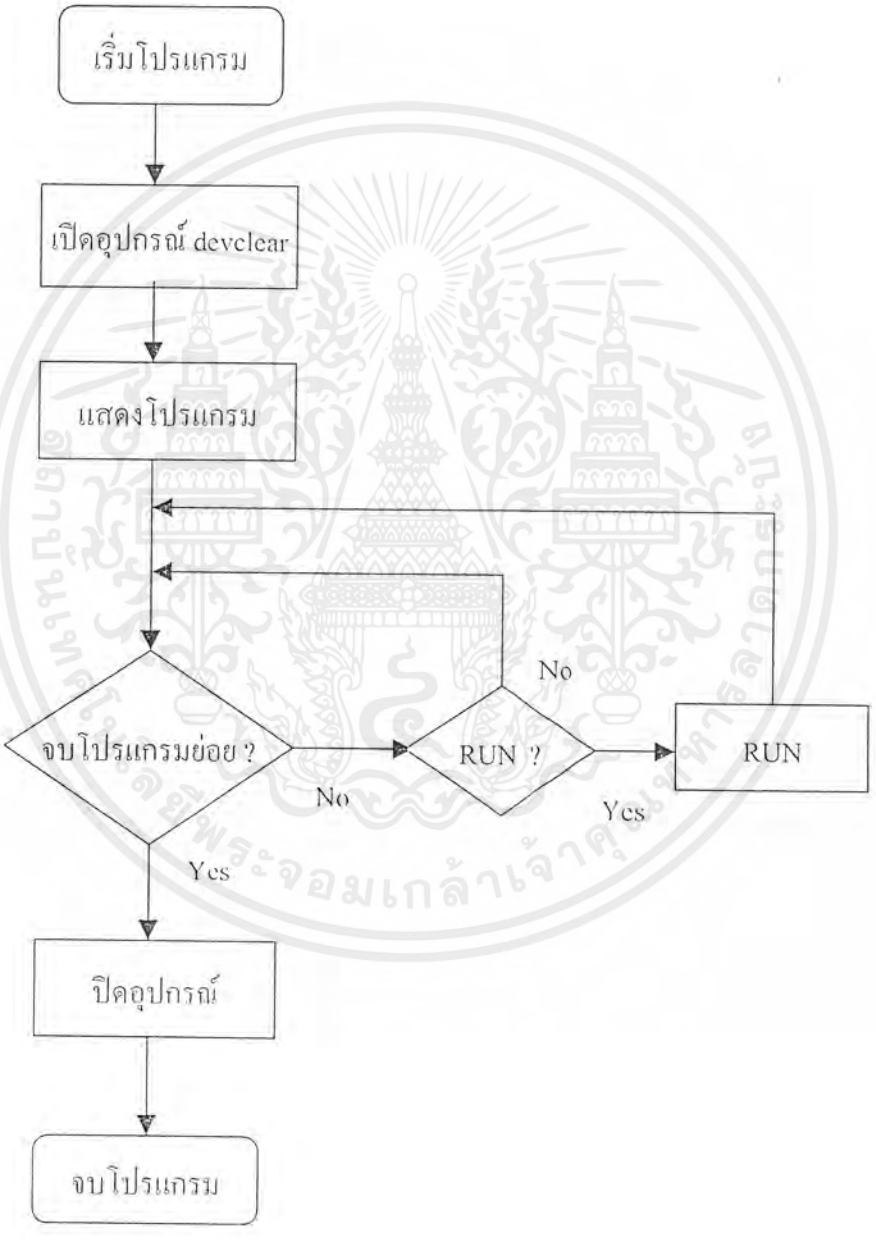
Power Supply, Function Generator และ Oscilloscope



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

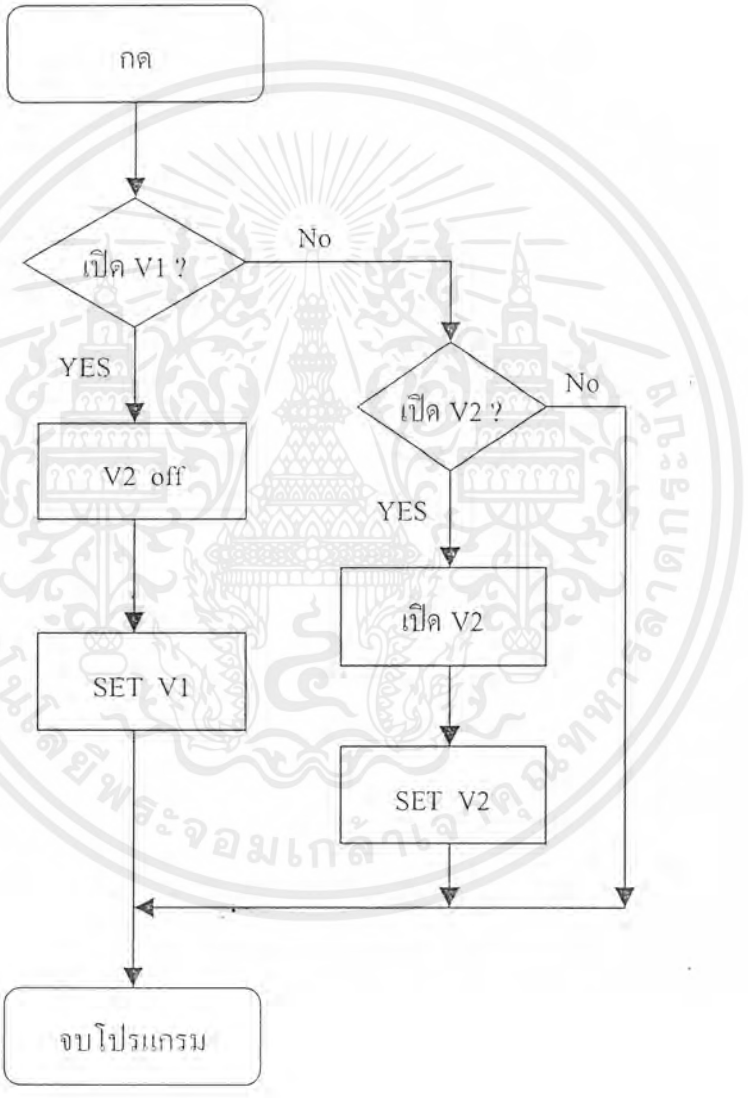
Flow Chart การทำงาน

Power Supply และ Function Generator



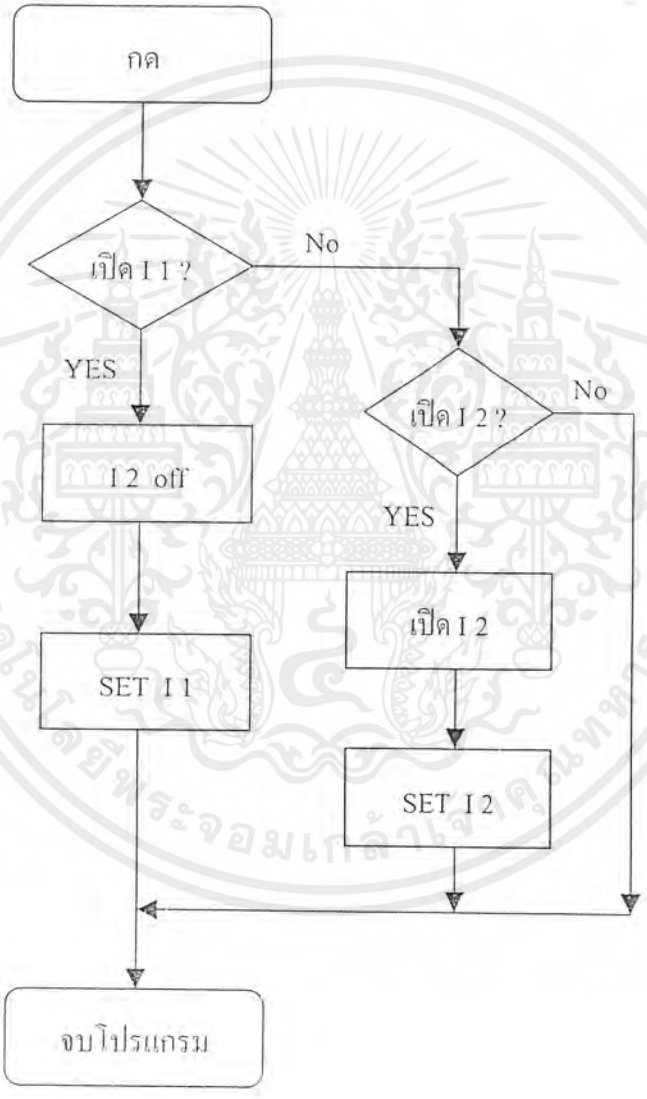
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของปุ่มกำหนดแรงดัน ของ Power Supply



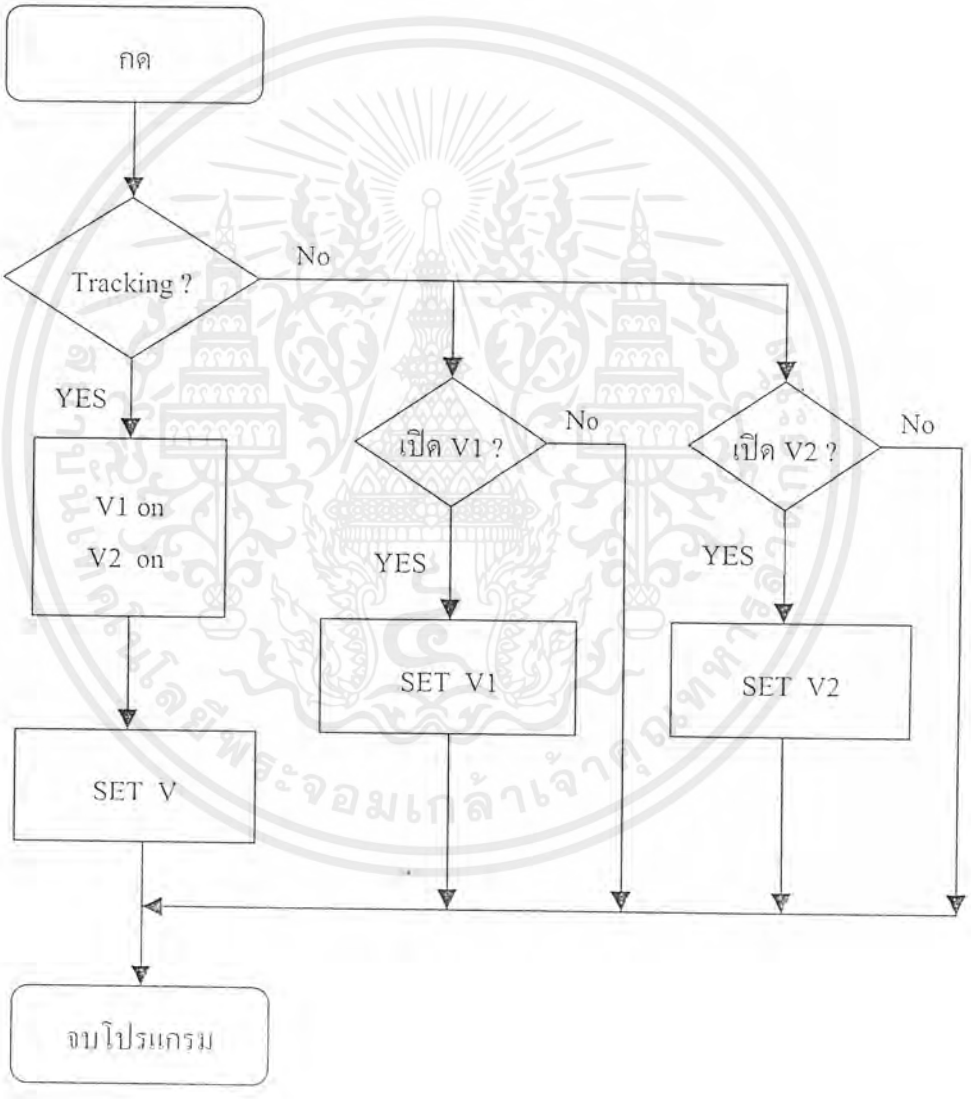
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของปุ่มกำหนดกระแสของ Power Supply



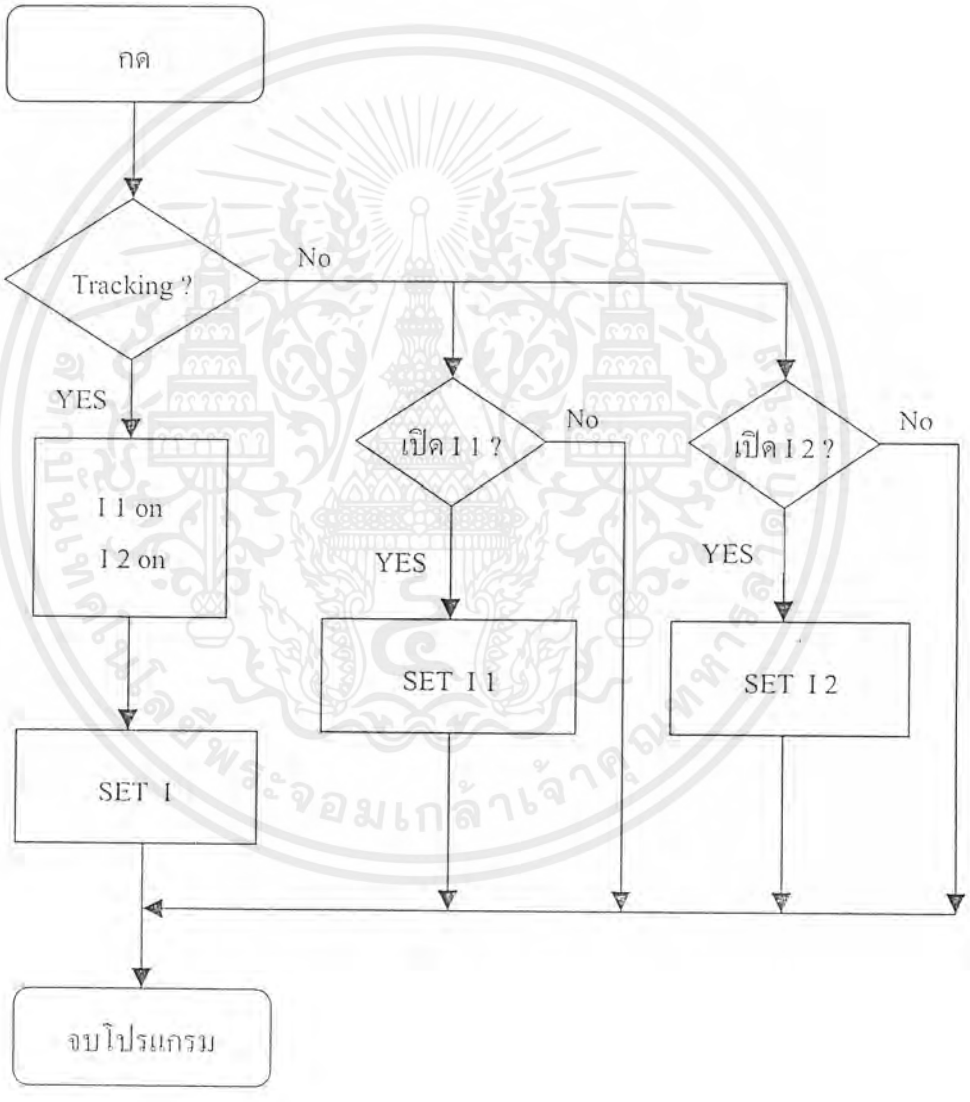
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของปุ่ม Tracking Voltage ของ Power Supply



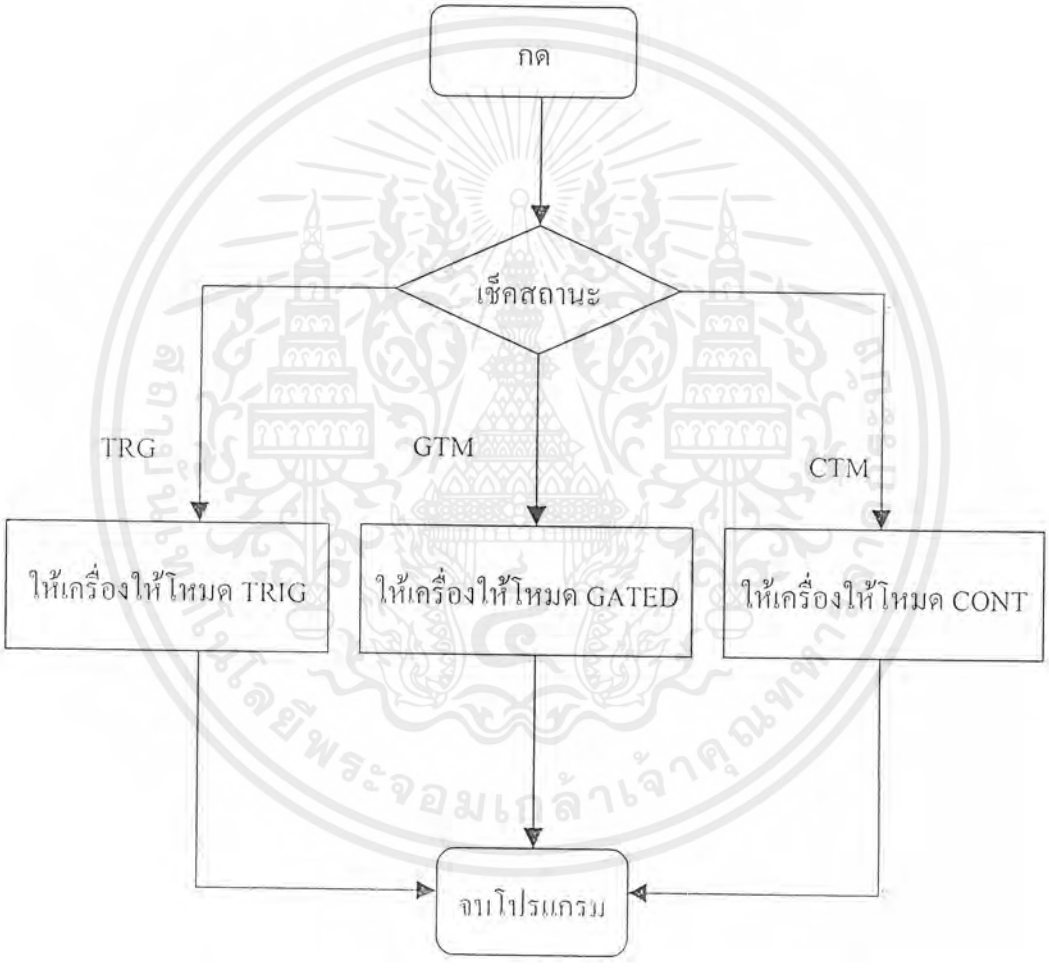
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของปุ่ม Tracking Current ของ Power Supply



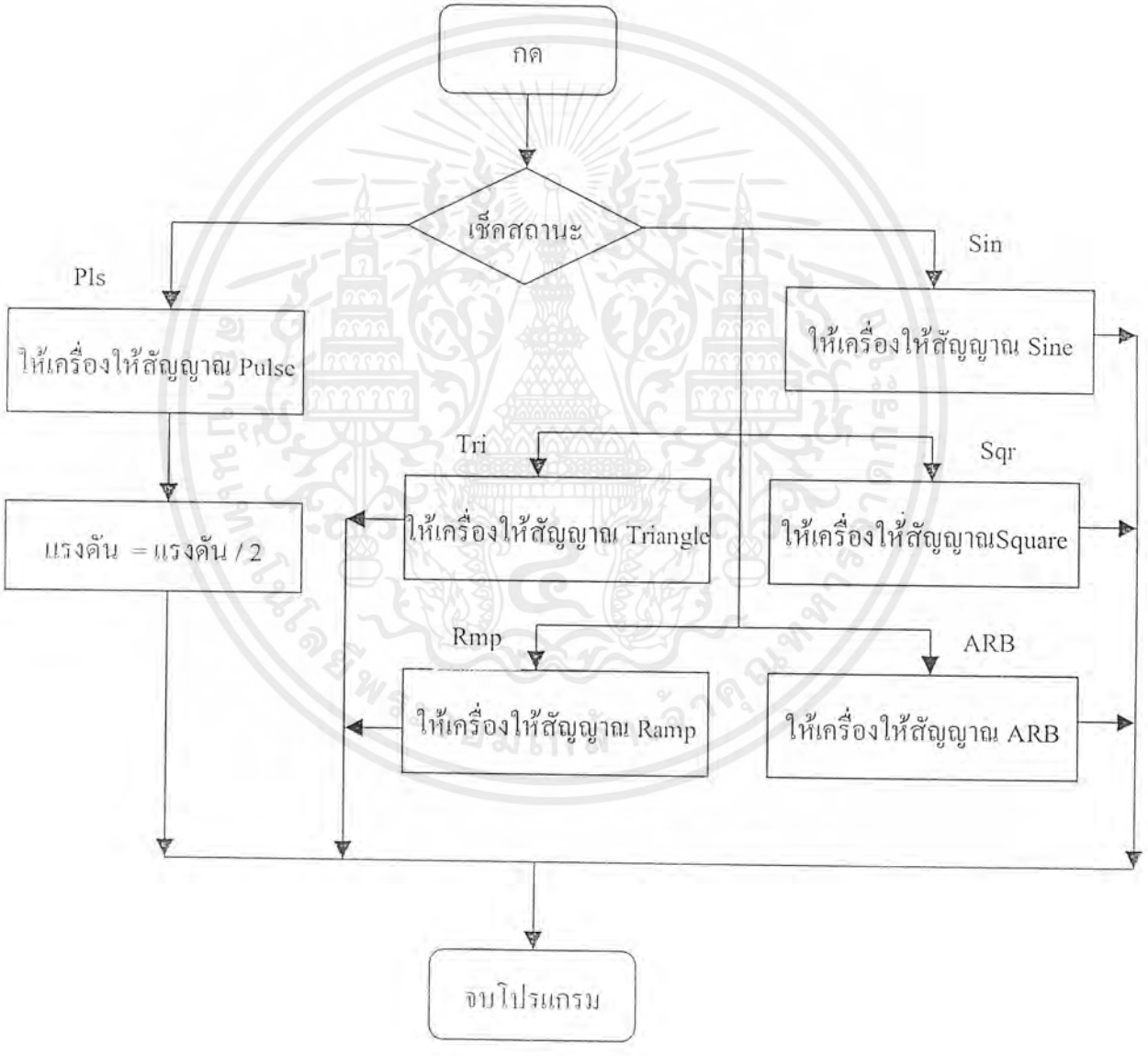
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปุ่มเลือกโหมดของ Function Generator



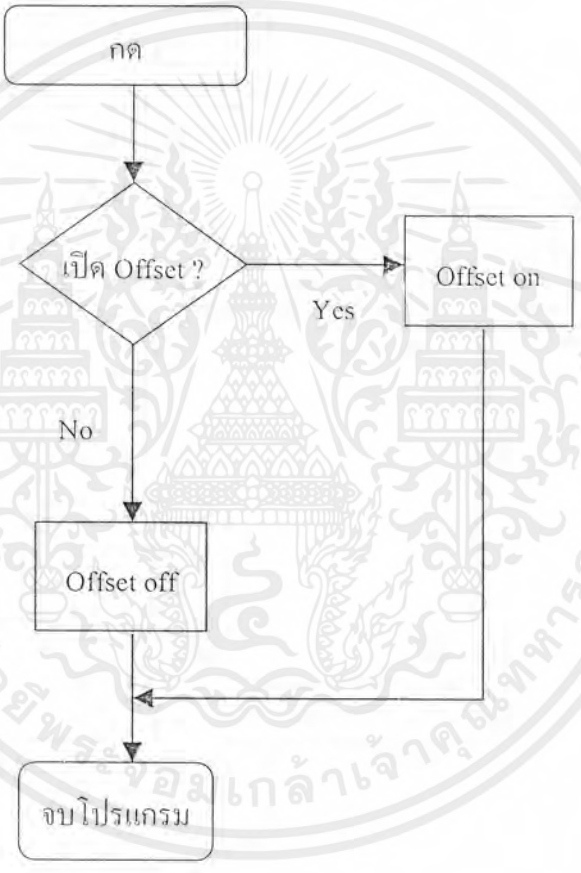
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปุ่มเลือกสัญญาณ Function ของ Function Generator



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

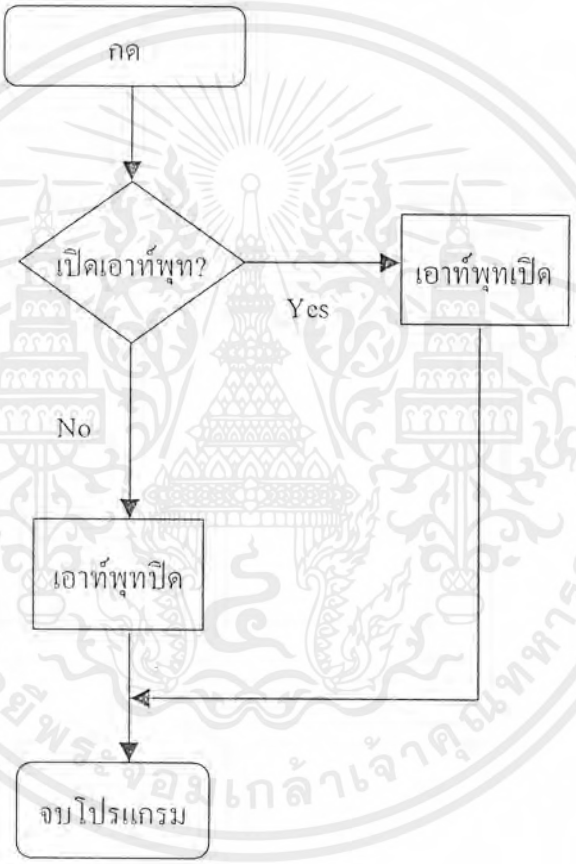
ปุ่ม Offset on ของ Function Generator



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปุ่มเอาท์พุท Output on

Power Supply และ Function Generator



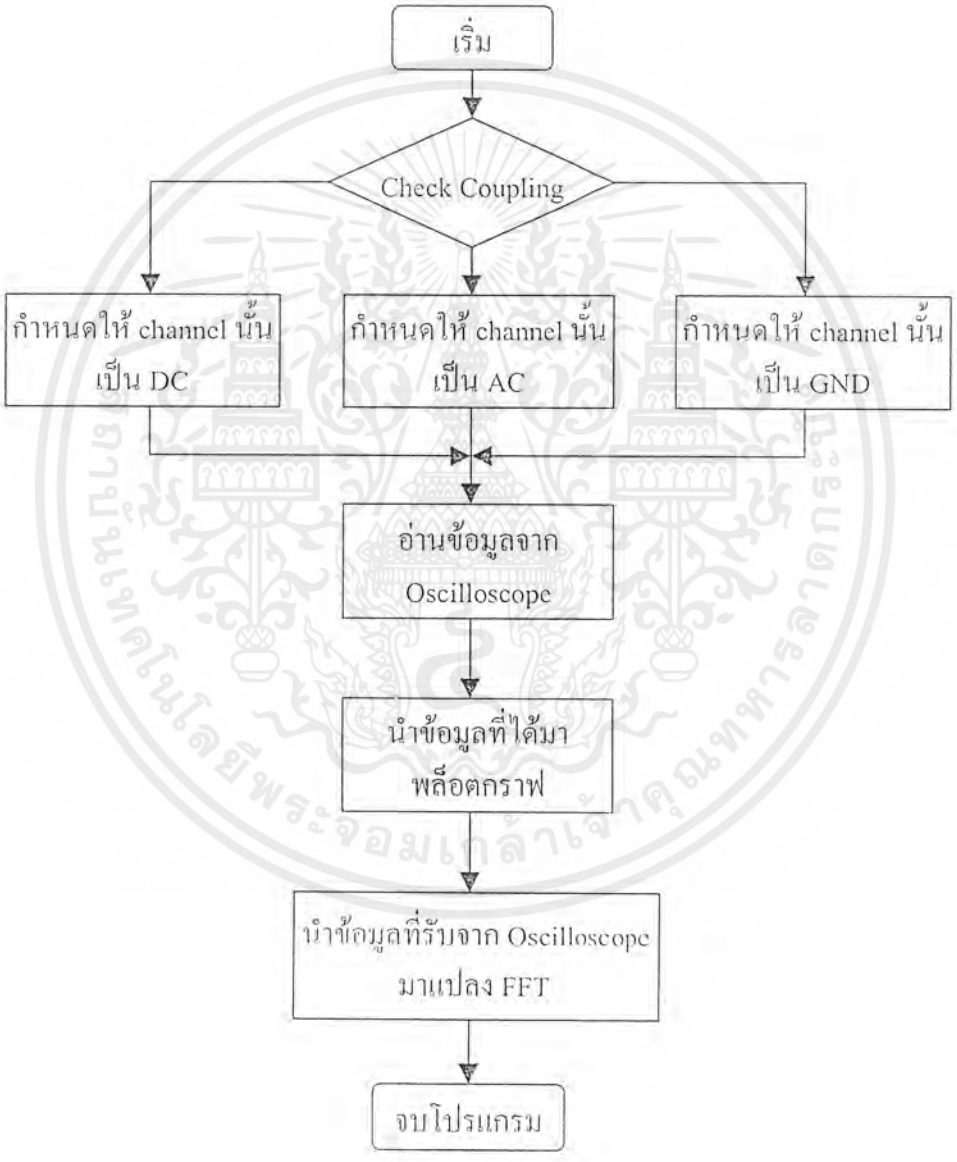
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปุ่ม Clear

Power Supply และ Function Generator

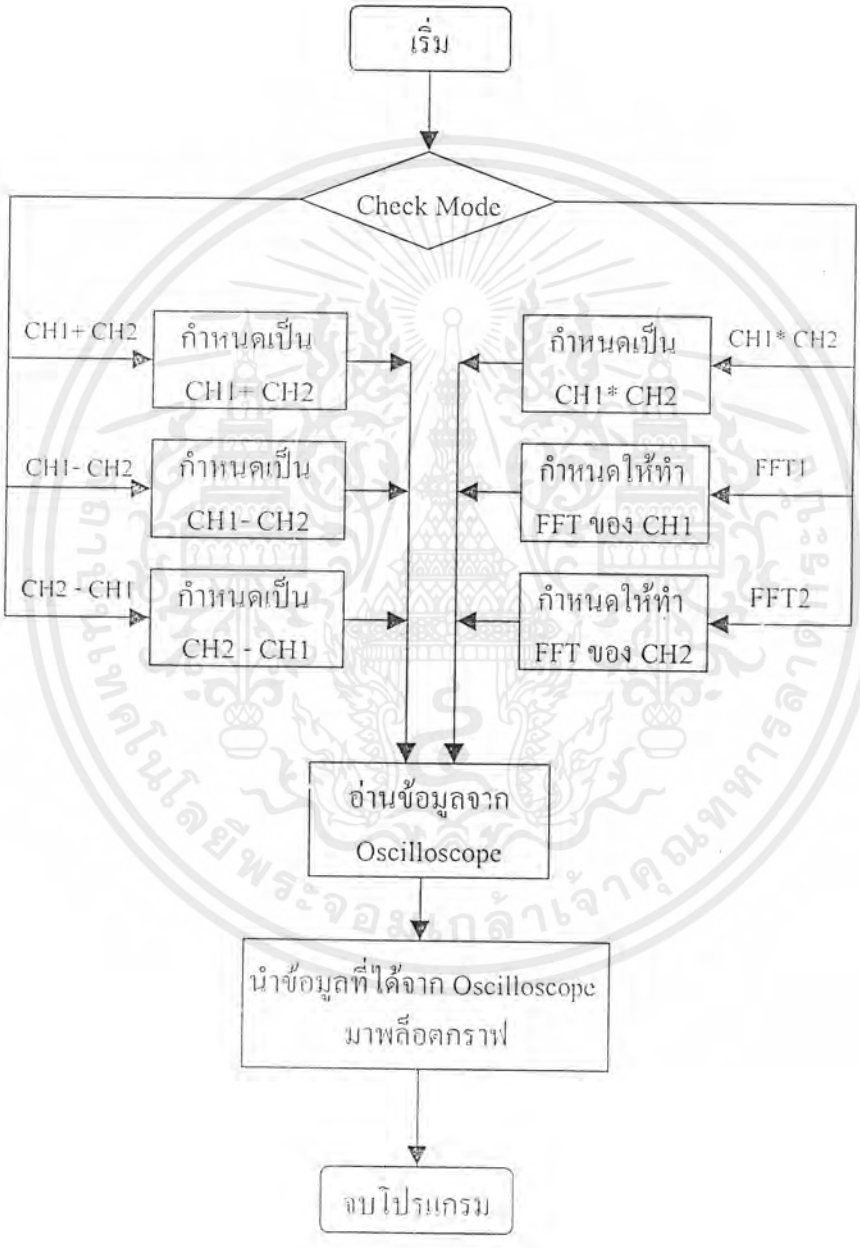
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปุ่ม Channel 1 และ Channel 2 ของ Oscilloscope



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปุ่ม MATH ของ Oscilloscope

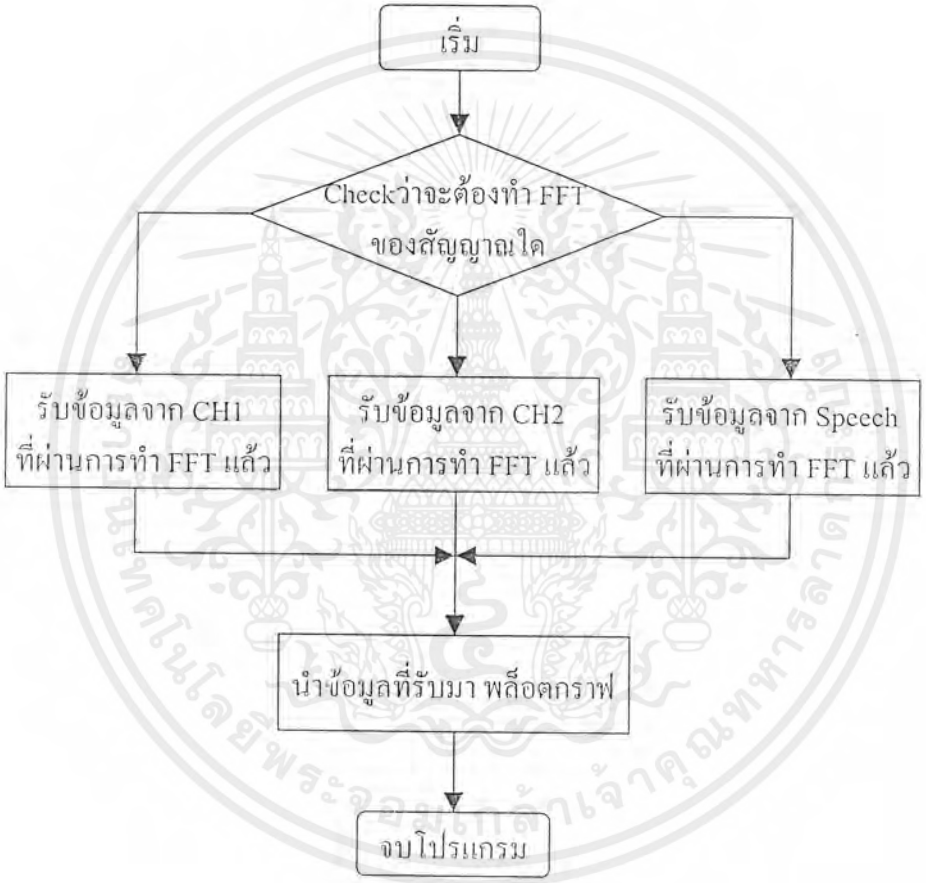


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปุ่มสัญญาณเสียง SPEECH ของ Oscilloscope

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FLOW CHART ของขบวนการ FFT ของ Oscilloscope

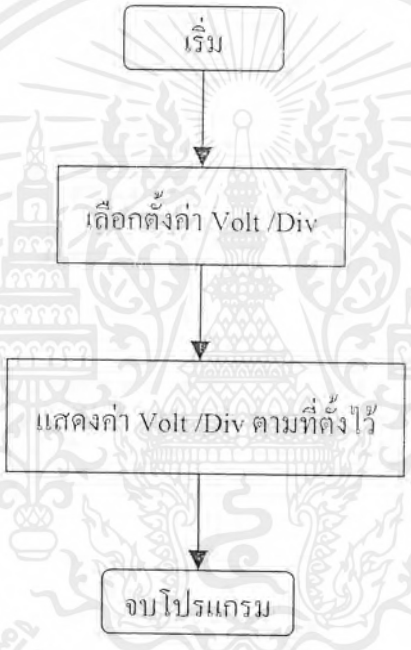


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตั้งค่า TIME / DIV ของ Oscilloscope

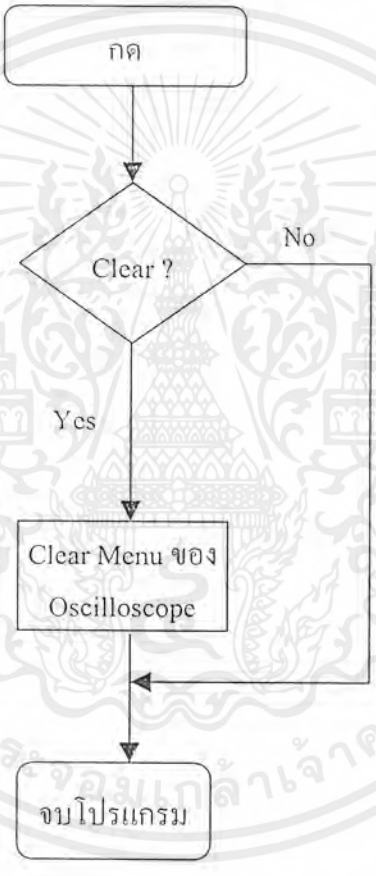
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตั้งค่า VOLT / DIV ของ Oscilloscope



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปุ่ม ClearMenu ของ Oscilloscope



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุมเครื่องมือวัด
Power Supply, Function Generator และ Oscilloscope
ผ่านระบบบัส IEEE-488 (GPIB)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมส่วนที่ใช้เชื่อมต่อระหว่าง GPIB กับไมโครคอมพิวเตอร์

```
Const
 (* GPIB status bit definitions  *)
 ERR    = $8000;  (* Error detectd  *)
 TIMO   = $4000;  (* Timeout       *)
 EEND   = $2000;  (* EOI or EOS detected  *)
 SRQI   = $1000;  (* SRQ detected by CIC  *)
 RQS    = $800;   (* Device needs service *)
 SPOLL  = $400;   (* Board has been serially polled  *)
 EVENT  = $200;   (* An event has occurred  *)
 CMPL   = $100;   (* I/O completed  *)
 LOK    = $80;    (* Local Lockout state  *)
 REM    = $40;    (* Remote state  *)
 CIC    = $20;    (* Controller-in-charge  *)
 ATN    = $10;    (* Attention asserted  *)
 TACS   = $8;     (* Talker active  *)
 LACS   = $4;     (* Listener active  *)
 DTAS   = $2;     (* Device trigger state  *)
 DCAS   = $1;     (* Device clear state  *)
(* Error messages returned in global variable IBERR:  *)
 EDVR   = 0;      (* System error  *)
 ECIC   = 1;      (* Function requires GPIB board to be CIC  *)
 ENOL   = 2;      (* Write function detected no Listeners  *)
 EADR   = 3;      (* Interface board not addressed correctly  *)
 EARG   = 4;      (* Invalid argument to function call  *)
 ESAC   = 5;      (* Function requires GPIB board to be SAC  *)
 EABO   = 6;      (* I/O operation aborted  *)
 ENEB   = 7;      (* Non-exsistent interface board  *)
 EDMA   = 8;      (* ERRor performing DMA  *)
 EOIP   = 10;     (* I/O operation started before previous  *)
          (* operation completed  *)
 ECAP   = 11;     (* No capability for intended operation  *)
 EFSO   = 12;     (* File system operation error  *)
 EBUS   = 14;     (* Command error during device call  *)
 ESTB   = 15;     (* Serial poll status byte lost  *)
 ESRQ   = 16;     (* SRQ remains asserted  *)
 ETAB   = 20;     (* The return buffer is full *)
 ELCK   = 21;     (* Address or board is locked  *)
(* Values for the 488.2 Send command.  *)
 NULLend = $00;   (* Do nothing at the end of a transfer  *)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
NLend    = $01;    (* Send NL with EOI after a transfer *)
DABend   = $02;    (* Send EOI with the last DAB *)
STOPend  = $100;   (* Value used by the 488.2 Receive command *)
MAVbit   = $10;    (* Position of the Massage Available bit *)
T10ms    = 7;      (* Timeout of 10 milliseconds(ideal) *)
T30ms    = 8;      (* Timeout of 30 milliseconds(ideal) *)
T100ms   = 9;      (* Timeout of 100 milliseconds(ideal) *)
T300ms   = 10;     (* Timeout of 300 milliseconds(ideal) *)
T1s      = 11;     (* Timeout of 1 second(ideal) *)
T3s      = 12;     (* Timeout of 3 seconds(ideal) *)
T10s     = 13;     (* Timeout of 10 seconds(ideal *)
T30s     = 14;     (* Timeout of 30 seconds(ideal) *)
T100s    = 15;     (* Timeout of 100 seconds(ideal) *)
T300s    = 16;     (* Timeout of 300 seconds(ideal) *)
T1000s   = 17;     (* Timeout of 1000 seconds(ideal) *)
Type
 AddrList = array[0..31] of SmallInt;
Tibsta   = function :integer;stdcall;
Tiberr   = function :integer;stdcall;
Tibcntl  = function :longint;stdcall;
Tibclr   = function (ud:integer):integer;stdcall;
Tibdev   = function (ud:integer;
          pad:integer;
          sad:integer;
          tmo:integer;
          eot:integer;
          eos:integer):integer;stdcall;
Tibloc   = function(ud:integer):integer;stdcall;
Tibrd    = function(ud:integer;
         var rdbuf;
          cnt :longint):integer;stdcall;
Tibtrg   = function(ud:integer):integer;stdcall;
Tibwrt   = function(ud:integer;
        var wrtbuf;
        cnt:longint):integer;stdcall;
TDevClear = procedure(boardID:integer;
          address:word);stdcall;
TFindLstn = procedure(boardID:integer;
          padlist: AddrList;
          resultlist: AddrList;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        limit: integer);stdcall;
  TEnableRemote = procedure(boardID:integer;
          address:word);stdcall;
 (* Tibfind = function(udname:string):integer;stdcall;*)
  Tibonl = function(ud:integer;
      v: integer):integer;stdcall;
  Tibwait = function(ud:integer;
      mask:integer):integer;stdcall;
  TRevRespMsg = procedure(boardID:integer;
        var buffer;
        cnt: longint;
        termination:integer);stdcall;
  TReceiveSetup = procedure(boardID:integer;
          address:word);stdcall;
  TReceive = procedure(boardID: integer;
      address: word;
      var buffer;
      cnt: longint;
      termination:integer);stdcall;
  TReadStatusByte = procedure(boardID: integer;
          address:word;
          var result:integer);stdcall;
  Tsend = procedure(boardID: integer;
     address:word;
     var buffer;
     datacnt:longint;
     eotmode:integer);stdcall;
  TSendSetup = procedure(boardID:integer;
        address: word);stdcall;
  TSendDataBytes = procedure(boardID:integer;
          var buffer;
          datacount:longint;
          eotmode:integer);stdcall;
  TSendIFC = procedure(boardID:integer);stdcall;
  TWaitSRQ = procedure(boardID:integer;
      var result:integer);stdcall;
  TEnableLocal = procedure(boardID:integer;
        address:word);stdcall;
Var
  Func_G: TFunc_G;        { Supply Tsupply }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  AddrIbsta : Tibsta;
  AddrIberr : Tiberr;
  AddrIbcntl: Tibcntl;
(* Pointers to the NI 488.2 GPIB global status variables *)
  Pibsta   : ^integer;
  Piberr   : ^integer;
  Pibcntl  : ^Longint;
(* Declaration for the Handle for the GPIB library *)
  Gpib32Lib :THandle;
(* Declarations for the NI-488.2 GPIB calls *)
(*  ibfind      : Tibfind;*)
  ibclr       : Tibclr;
  ibdev       : Tibdev;
  ibrd        : Tibrd;
  ibtrg       : Tibtrg;
  ibwait      : Tibwait;
  ibwrt       : Tibwrt;
  DevClear    : TDevClear;
  ibloc       : Tibloc;
  FindLstn    : TFindLstn;
  ReadStatusByte : TReadStatusByte;
  Receive     : TReceive;
  ReceiveSetup  : TReceiveSetup;
  RcvRespMsg    : TRcvRespMsg;
  SendDataBytes : TSendDataBytes;
  SendSetup     : TSendSetup;
  Send        : TSend;
  SendIFC     : TSendIFC;
  WaitSRQ     : TWaitSRQ;
  EnableRemote  : TEnableRemote;
  EnableLocal   : TEnableLocal;
(* Declaration for the NI-488 GPIB call *)
  ibonl       : Tibonl;
  WriteBuff:Packed array[0..32] of char;
  ReadBuff:Packed array[0..32] of char;
  F_temp:integer;
implementation
{$R *.DFM}

procedure loadDLL;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
var str : string;
begin
(* Load the GPIB-32.DLL library using the Load Library function *)
Gpib32Lib := LoadLibrary('GPIB-32.DLL');
If Gpib32Lib = 0 then
  begin
      str := 'LoadLibrary FAILED!';
      MessageDlg(str,mtError,[mbOK],0);
      halt;
  end;

(* Get the addresses of the GPIB Global Variables. *)
@AddrIbsta  := GetProcAddress(Gpib32Lib,'user_ibsta');
@AddrIberr  := GetProcAddress(Gpib32Lib,'user_iberr');
@AddrIbcntl := GetProcAddress(Gpib32Lib,'user_ibcntl');
(* Get the Addresses of the functions needed for this application *)
(* @ibfind      := GetProcAddress(Gpib32Lib,'ibfind'); *)
@ibclr       := GetProcAddress(Gpib32Lib,'ibclr');
@ibdev       := GetProcAddress(Gpib32Lib,'ibdev');
@ibrd        := GetProcAddress(Gpib32Lib,'ibrd');
@ibtrg       := GetProcAddress(Gpib32Lib,'ibtrg');
@ibwait      := GetProcAddress(Gpib32Lib,'ibwait');
@ibwrt       := GetProcAddress(Gpib32Lib,'ibwrt');
@DevClear    := GetProcAddress(Gpib32Lib,'DevClear');
@ibloc       := GetProcAddress(Gpib32Lib,'ibloc');
@FindLstn    := GetProcAddress(Gpib32Lib,'FindLstn');
@ibonl       := GetProcAddress(Gpib32Lib,'ibonl');
@ReadStatusByte := GetProcAddress(Gpib32Lib,'ReadStatusByte');
@RcvRespMsg  := GetProcAddress(Gpib32Lib,'RcvRespMsg');
@ReceiveSetup  := GetProcAddress(Gpib32Lib,'ReceiveSetup');
@Receive     := GetProcAddress(Gpib32Lib,'Receive');
@SendDataBytes  := GetProcAddress(Gpib32Lib,'SendDataBytes');
@Send        := GetProcAddress(Gpib32Lib,'Send');
@SendSetup   := GetProcAddress(Gpib32Lib,'SendSetup');
@SendIFC     := GetProcAddress(Gpib32Lib,'SendIFC');
@WaitSRQ     := GetProcAddress(Gpib32Lib,'WaitSRQ');
@EnableRemote  := GetProcAddress(Gpib32Lib,'EnableRemote');
@EnableLocal   := GetProcAddress(Gpib32Lib,'EnableLocal');
(* Initialize GPIB global pointers to point to address location *)
Pibsta  := @AddrIbsta;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Piberr  := @Addrlberr;
  Pibentl := @Addrlbentl;

if
  (@ibdev       = NIL)   or
  (@ibclr       = NIL)   or
  (* (@ibfind      = NIL) or *)
  (@ibrd        = NIL)   or
  (@ibtrg       = NIL)   or
  (@ibwait      = NIL)   or
  (@ibwrt         = NIL)  or
  (@AddrlbSta     = NIL)  or
  (@Addrlberr     = NIL)  or
  (@Addrlbentl    = NIL)  or
  (@DevClear      = NIL)  or
  (@FindLstn      = NIL)  or
  (@ibloc         = NIL)  or
  (@ibonl         = NIL)  or
  (@ReadStatusByte = NIL)  or
  (@RcvRespMsg    = NIL)  or
  (@ReceiveSetup  = NIL)  or
  (@Receive       = NIL)  or
  (@SendDataBytes = NIL)  or
  (@SendSetup     = NIL)  or
  (@Send          = NIL)  or
  (@SendIFC       = NIL)  or
  (@WaitSRQ       = NIL)  or
  (@EnableRemote  = NIL)  or
  (@EnableLocal   = NIL)  Then

  begin
       str := 'GetProcAddress FAILED!';
       MessageDlg(str,mtError,[mbOK],0);
       (* Free the GPIB library *)
       FreeLibrary(Gpib32Lib);
       halt;
  end;
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Unit supply;
Interface
Uses
        Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,ExtCtrls, Buttons, ComCtrls, StdCtrls,
        zPanel, zLed, Zseg, zonzoff;
Type
          Tsupply = class(TForm)
          Panel1: TPanel;
          Panel2: TPanel;
          Panel3: TPanel;
          Panel4: TPanel;
          Panel5: TPanel;
          OnOff1: TzeLightOnOff;
          dis1: TzeDotLabel;
          dis2: TzeDotLabel;
          dis3: TzeDotLabel;
          dis4: TzeDotLabel;
          zLed1: TzLed;
          zLed2: TzLed;
          zLed3: TzLed;
          zLed4: TzLed;
          zKnob1: TzKnob;
          zKnob2: TzKnob;
          Button1: TButton;
          Button2: TButton;
          Button3: TButton;
          Button4: TButton;
          UpDown1: TUpDown;
          UpDown2: TUpDown;
          UpDown3: TUpDown;
          UpDown4: TUpDown;
          SpeedButton1: TSpeedButton;
          SpeedButton2: TSpeedButton;
          SpeedButton3: TSpeedButton;
          SpeedButton4: TSpeedButton;
          zLed5: TzLed;
          zLed6: TzLed;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure FormCreate(Sender: TObject);
procedure OnOff1Off(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    procedure OnOff1On(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure zKnob1Change(Sender: TObject);
    procedure UpDown1Changing(Sender: TObject; var AllowChange:   Boolean);
    procedure UpDown2Changing(Sender: TObject; var AllowChange: Boolean);
    procedure SpeedButton3Click(Sender: TObject);
    procedure zKnob2Change(Sender: TObject);
    procedure UpDown3Changing(Sender: TObject; var AllowChange: Boolean);
    procedure UpDown4Changing(Sender: TObject; var AllowChange: Boolean);
    procedure SpeedButton4Click(Sender: TObject);
  private
              { Private declarations }
  public
              { Public declarations }
              procedure SendDV1(DataSend:integer);
              procedure SendDV2(DataSend:integer);
              procedure SendDI1(DataSend:integer);
              procedure SendDI2(DataSend:integer);
              procedure SendV(DataSend:integer);
              procedure SendI(DataSend:integer);
  end;
    {   Const
        ใส่ค่า Const ตามที่แสดงไว้ข้างต้นเพื่อเชื่อมต่อระหว่าง
        ไมโครคอมพิวเตอร์กับการ์ด GPIB      }
*****************************************************************************
  procedure Tsupply.SendDV1(DataSend:integer);
*****************************************************************************
var   n:integer;
      data,a1,a2,a3,a4,d:string;
      d_temp:extended;
begin
      writebuff:='SU1 ';
      d_temp:=DataSend/100;
      data:=floattostr(d_temp);
      panel2.Caption:=data;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for n:=0 to length(data) do
        begin
            d:=copy(data,n+1,1);
            if d='0' then writebuff[4+n]:='0';
            if d='1' then writebuff[4+n]:='1';
            if d='2' then writebuff[4+n]:='2';
            if d='3' then writebuff[4+n]:='3';
            if d='4' then writebuff[4+n]:='4';
            if d='5' then writebuff[4+n]:='5';
            if d='6' then writebuff[4+n]:='6';
            if d='7' then writebuff[4+n]:='7';
            if d='8' then writebuff[4+n]:='8';
            if d='9' then writebuff[4+n]:='9';
            if d='.' then writebuff[4+n]:='.';
        end;
    writebuff[4+n]:=';';
    send(0,8,writebuff,length(writebuff),1);
    writebuff:='MU1;';
    send(0,8,writebuff,length(writebuff),1);
    receive(0,8,readbuff,length(readbuff),1);
    a1:=readbuff[3];
    a2:=readbuff[4];
    a3:=readbuff[6];
    a4:=readbuff[7];
    dis1.Caption:=a1+a2+'.'+a3+a4;
end;
*****************************************************************************
procedure Tsupply.SendDV2(DataSend:integer);
*****************************************************************************
var   n:integer;
      data,a1,a2,a3,a4,d:string;
      d_temp:extended;
begin
      writebuff:='SU2 ';
      d_temp:=DataSend/100;
      data:=floattostr(d_temp);
      panel3.Caption:=data;
      for n:=0 to length(data) do
          begin
              d:=copy(data,n+1,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if d='0' then writebuff[4+n]:='0';
            if d='1' then writebuff[4+n]:='1';
            if d='2' then writebuff[4+n]:='2';
            if d='3' then writebuff[4+n]:='3';
            if d='4' then writebuff[4+n]:='4';
            if d='5' then writebuff[4+n]:='5';
            if d='6' then writebuff[4+n]:='6';
            if d='7' then writebuff[4+n]:='7';
            if d='8' then writebuff[4+n]:='8';
            if d='9' then writebuff[4+n]:='9';
            if d='.' then writebuff[4+n]:='.';
        end;
    writebuff[4+n]:=';';
    send(0,8,writebuff,length(writebuff),1);
    writebuff:='MU2;';
    send(0,8,writebuff,length(writebuff),1);
    receive(0,8,readbuff,length(readbuff),1);
    a1:=readbuff[3];
    a2:=readbuff[4];
    a3:=readbuff[6];
    a4:=readbuff[7];
    dis2.Caption:=a1+a2+'.'+a3+a4;
end;
*****************************************************************************
procedure Tsupply.SendV(DataSend:integer);
*****************************************************************************
var
    n:integer;
    data,a1,a2,a3,a4,d:string;
    d_temp:extended;
begin
    writebuff:='TRU ';
    d_temp:=DataSend/100;
    data:=floattostr(d_temp);
    panel2.Caption:=data;
    panel3.Caption:=data;
    for n:=0 to length(data) do
        begin
            d:=copy(data,n+1,1);
            if d='0' then writebuff[4+n]:='0';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if d='1' then writebuff[4+n]:='1';
            if d ='2' then writebuff[4+n]:='2';
            if d='3' then writebuff[4+n]:='3';
            if d='4' then writebuff[4+n]:='4';
            if d='5' then writebuff[4+n]:='5';
            if d='6' then writebuff[4+n]:='6';
            if d='7' then writebuff[4+n]:='7';
            if d='8' then writebuff[4+n]:='8';
            if d='9' then writebuff[4+n]:='9';
            if d='.' then writebuff[4+n]:='.';
        end;
    writebuff[4+n]:='.';
    send(0,8,writebuff,length(writebuff),1);
    writebuff:='MU2:';
    send(0,8,writebuff,length(writebuff),1);
    receive(0,8,readbuff,length(readbuff),1);
    a1:=readbuff[3];
    a2:=readbuff[4];
    a3:=readbuff[6];
    a4:=readbuff[7];
    dis1.Caption:=a1+a2+'.'+a3+a4;
    dis2.Caption:=a1+a2+'.'+a3+a4;
end;
*****************************************************************************
procedure Tsupply.SendDH1(DataSend:integer);
*****************************************************************************
var   n:integer;
    data,a1,a2,a3,a4,d,c2:string;
    d_temp,c1:extended;
begin
    writebuff:='SH1:';
    d_temp:=DataSend/1000;
    data:=floattostr(d_temp);
    panel4.Caption:=data;
    for n:=0 to length(data) do
        begin
            d:=copy(data,n+1,1);
            if d='0' then writebuff[4+n]:='0';
            if d='1' then writebuff[4+n]:='1';
            if d='2' then writebuff[4+n]:='2';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if d='3' then writebuff[4+n]:='3';
                if d='4' then writebuff[4+n]:='4';
                if d='5' then writebuff[4+n]:='5';
                if d='6' then writebuff[4+n]:='6';
                if d='7' then writebuff[4+n]:='7';
                if d='8' then writebuff[4+n]:='8';
                if d='9' then writebuff[4+n]:='9';
                if d='.' then writebuff[4+n]:='.';
            end;
        writebuff[4+n]:=',';
        send(0,8,writebuff,length(writebuff),1);
        writebuff:='R11,';
        send(0,8,writebuff,length(writebuff),1);
        receive(0,8,readbuff,length(readbuff),1);
        a1:=readbuff[8];
        a2:=readbuff[4];
        a3:=readbuff[6];
        a4:=readbuff[7];
        dis3.Caption:=a2+','+a3+a4+a1;
end;
*****************************************************************************
procedure Tsupply.SendDI2(DataSend:integer);
*****************************************************************************
var   n:integer;
      data,a1,a2,a3,a4,d,c2:string;
      d_temp,c1:extended;
begin
      writebuff:='SI2 ';
      d_temp:=DataSend/100;
      data:=floattostr(d_temp);
      panel5.Caption:=data;
      for n:=0 to length(data) do
          begin
              d:=copy(data,n+1,1);
              if d='0' then writebuff[4+n]:='0';
              if d='1' then writebuff[4+n]:='1';
              if d='2' then writebuff[4+n]:='2';
              if d='3' then writebuff[4+n]:='3';
              if d='4' then writebuff[4+n]:='4';
              if d='5' then writebuff[4+n]:='5';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if d='6' then writebuff[4+n]:='6';
            if d='7' then writebuff[4+n]:='7';
            if d='8' then writebuff[4+n]:='8';
            if d='9' then writebuff[4+n]:='9';
            if d='.' then writebuff[4+n]:='.';
        end;
    writebuff[4+n]:=';';
    send(0,8,writebuff,length(writebuff),1);
    writebuff:='R12;';
    send(0,8,writebuff,length(writebuff),1);
    receive(0,8,readbuff,length(readbuff),1);
    a1:=readbuff[8];
    a2:=readbuff[4];
    a3:=readbuff[6];
    a4:=readbuff[7];
    dis4.Caption:=a2+'.'+a3+a4+a1;
end;
*****************************************************************************
procedure Tsupply.Send1(DataSend:integer);
*****************************************************************************
var   n:integer;
    data,a1,a2,a3,a4,d:string;
    d_temp:extended;
begin
    writebuff:='TR1';
    d_temp:=DataSend/1000;
    data:=floattostr(d_temp);
    panel4.Caption:=data;
    panel5.Caption:=data;
    for   n:=0 to length(data) do
      begin
          d:=copy(data,n+1,1);
          if d='0' then writebuff[4+n]:='0';
          if d='1' then writebuff[4+n]:='1';
          if d='2' then writebuff[4+n]:='2';
          if d='3' then writebuff[4+n]:='3';
          if d='4' then writebuff[4+n]:='4';
          if d='5' then writebuff[4+n]:='5';
          if d='6' then writebuff[4+n]:='6';
          if d='7' then writebuff[4+n]:='7';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if d='8' then writebuff[4+n]:='8';
            if d='9' then writebuff[4+n]:='9';
            if d='.' then writebuff[4+n]:='.';
        end;
    writebuff[4+n]:=';';
    send(0,8,writebuff,length(writebuff),1);
    writebuff:='R11;';
    send(0,8,writebuff,length(writebuff),1);
    receive(0,8,readbuff,length(readbuff),1);
    a1:=readbuff[8];
    a2:=readbuff[4];
    a3:=readbuff[6];
    a4:=readbuff[7];
    dis3.Caption:=a2+'.'+a3+a4+a1;
    dis4.Caption:=a2+'.'+a3+a4+a1;
end;
*****************************************************************************
procedure Tsupply.FormClose(Sender: TObject; var Action: TCloseAction);
*****************************************************************************
begin
   FreeLibrary(GPIB32LIB);
end;
*****************************************************************************
procedure Tsupply.FormCreate(Sender: TObject);
*****************************************************************************
begin
    LoadDLL;
    SendIFC(0);
end;
*****************************************************************************
procedure Tsupply.OnOff1Off(Sender: TObject);
*****************************************************************************
begin
    writebuff:='RM0;';
    send(0,8,writebuff,8,1);
    button1.Enabled:=false;
    button2.Enabled:=false;
    button3.Enabled:=false;
    button4.Enabled:=false;
    zknob1.Enabled:=false;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        zknob2.Enabled:=false;
        UpDown1.Enabled:=false;
        UpDown2.Enabled:=false;
        UpDown3.Enabled:=false;
        UpDown4.Enabled:=false;
        zLed1.Enabled:=false;
        zLed2.Enabled:=false;
        zLed3.Enabled:=false;
        zLed4.Enabled:=false;
        zLed5.Enabled:=false;
        zLed6.Enabled:=false;
end;
*****************************************************************************
procedure Tsupply.OnOffTOn(Sender: TObject);
*****************************************************************************
begin
        DevClear(0,8);
        button1.Enabled:=true;
        button2.Enabled:=true;
        button3.Enabled:=true;
        button4.Enabled:=true;
        zknob1.Enabled:=true;
        zknob2.Enabled:=true;
        UpDown1.Enabled:=true;
        UpDown2.Enabled:=true;
        UpDown3.Enabled:=true;
        UpDown4.Enabled:=true;
end;
*****************************************************************************
procedure Tsupply.Button1Click(Sender: TObject);
*****************************************************************************
begin
      if button1.Tag=0 then
        begin
                zLed1.Enabled:=true;
                button1.Tag:=1;
                button2.Tag:=0;
                button2.Enabled:=false;
        end
      else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        begin
                zLed1.Enabled:=false;
                button1.Tag:=0;
                button2.Enabled:=true;
        end;
end;
*****************************************************************************
procedure Tsupply.Button2Click(Sender: TObject);
*****************************************************************************
begin
    if button2.Tag=0 then
        begin
                zLed2.Enabled:=true;
                button2.Tag:=1;
                button1.Tag:=0;
                button1.Enabled:=false;
        end
    else
        begin
                zLed2.Enabled:=false;
                button2.Tag:=0;
                button1.Enabled:=true;
        end;
end;
*****************************************************************************
procedure Tsupply.Button3Click(Sender: TObject);
*****************************************************************************
begin
    if button3.Tag=0 then
        begin
                zLed3.Enabled:=true;
                button3.Tag:=1;
                button4.Tag:=0;
                button4.Enabled:=false;
        end
    else
        begin
                zLed3.Enabled:=false;
                button3.Tag:=0;
                button4.Enabled:=true;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        end;
end;
*****************************************************************************
procedure Tsupply.Button4Click(Sender: TObject);
*****************************************************************************
begin
    if button4.Tag=0 then
      begin
            zLed4.Enabled:=true;
            button4.Tag:=1;
            button3.Tag:=0;
            button3.Enabled:=false;
      end
    else
      begin
            zLed4.Enabled:=false;
            button4.Tag:=0;
            button3.Enabled:=true;
      end;
end;
*****************************************************************************
procedure Tsupply.SpeedButton2Click(Sender: TObject);
*****************************************************************************
begin
    if SpeedButton2.Tag=0 then
      begin
            writebuff:='OP1;';
            send(0,8,writebuff[1],length(writebuff),1);
            SpeedButton2.Tag:=1;
      end
    else
      begin
            writebuff:='OP0;';
            send(0,8,writebuff[1],length(writebuff),1);
            SpeedButton2.Tag:=0;
      end;
end;
*****************************************************************************
procedure Tsupply.SpeedButton1Click(Sender: TObject);
*****************************************************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
    writebuff:='clr;';
    send(0,8,writebuff,length(writebuff),1);
    zknob1.Value:=0;
    zknob2.Value:=0;
    dis1.Caption:='00.00';
    dis2.Caption:='00.00';
    dis3.Caption:='0.000';
    dis4.Caption:='0.000';
    panel2.Caption:='00.00';
    panel3.Caption:='0.000';
    panel4.Caption:='0.000';
    panel5.Caption:='0.000';
    UpDown1.Position:=0;
    UpDown2.Position:=0;
    UpDown3.Position:=0;
    UpDown4.Position:=0;
end;
*****************************************************************************
procedure Tsupply.zKnob1Change(Sender: TObject);
*****************************************************************************
begin
    UpDown1.Position:=zknob1.Value;
    i_Data:=(UpDown1.Position*10)+UpDown2.Position;
    if i_data < 3000 then
      begin
          if button1.Tag = 1 then SendDV1(i_Data);
          if button2.tag = 1 then SendDV2(i_Data);
          if SpeedButton3.tag = 1 then SendV(i_data);
      end
    else
      begin
          i_data := 3000;
          if button1.tag = 1 then SendDV1(i_data);
          if button2.tag = 1 then SendDV2(i_data);
          if SpeedButton3.tag = 1 then sendV(i_data);
      end;
end;
*****************************************************************************
procedure Tsupply.UpDown1Changing(Sender: TObject;
*****************************************************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 var AllowChange: Boolean);
 begin
      zknob1.Value:=UpDown1.Position;
      i_Data:=(UpDown1.Position*10)+UpDown2.Position;
      if i_data < 3000 then
        begin
              if button1.Tag = 1 then SendDV1(i_Data);
              if button2.tag = 1 then SendDV2(i_Data);
              if SpeedButton3.tag = 1 then SendV(i_data);
        end
      else
        begin
              i_data := 3000;
              if button1.tag = 1 then SendDV1(i_data);
              if button2.tag = 1 then SendDV2(i_data);
              if SpeedButton3.tag = 1 then sendV(i_data);
        end;
end;
*****************************************************************************
procedure Tsupply.UpDown2Changing(Sender: TObject;
*****************************************************************************
var  AllowChange: Boolean;
begin
      i_Data:=(UpDown1.Position*10)+UpDown2.Position;
      if i_data < 3000 then
        begin
              if button1.Tag = 1 then SendDV1(i_Data);
              if button2.tag = 1 then SendDV2(i_Data);
              if SpeedButton3.tag = 1 then SendV(i_data);
        end
      else
        begin
              i_data := 3000;
              if button1.tag = 1 then SendDV1(i_data);
              if button2.tag = 1 then SendDV2(i_data);
              if SpeedButton3.tag = 1 then sendV(i_data);
        end;
end;
*****************************************************************************
procedure Tsupply.SpeedButton3Click(Sender: TObject);
*****************************************************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
    if SpeedButton3.Tag = 0 then
      begin
            zled5.Enabled := true;
            SpeedButton3.tag := 1;
            button1.Enabled:=false;
            button2.Enabled:=false;
            button1.tag:=1;
            button2.tag:=1;
            i_Data:=(UpDown1.Position*10)+UpDown2.Position;
            if i_data < 3000 then sendV(i_data);
      end
    else
      begin
            zled5.Enabled := false;
            speedButton3.tag := 0;
            button1.Enabled:=true;
            button2.Enabled:=true;
            button1.tag:=0;
            button2.tag:=0;
      end;
end;
*****************************************************************************
procedure Tsupply.zKnob2Change(Sender: TObject);
*****************************************************************************
begin
    UpDown3.Position:=zknob2.Value;
    i_Data:=(UpDown3.Position*100)+UpDown4.Position;
    if i_data < 1000 then
      begin
            if button3.Tag = 1 then SendDI1(i_Data);
            if button4.Tag = 1 then SendDI2(i_Data);
            if SpeedButton4.Tag = 1 then SendI(i_data);
      end
    else
      begin
            i_data := 1000;
            if button3.Tag = 1 then SendDI1(i_data);
            if button4.Tag = 1 then SendDI2(i_data);
            if SpeedButton4.Tag = 1 then sendI(i_data);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        end;
end;
*****************************************************************************
procedure Tsupply.UpDown3Changing(Sender: TObject;
*****************************************************************************
var  AllowChange: Boolean;
begin
     zknob2.Value:=UpDown3.Position;
     i_Data:=(UpDown3.Position*100)+UpDown4.Position;
     if i_data < 1000 then
       begin
            if button3.Tag = 1 then SendDI1(i_Data);
            if button4.Tag = 1 then SendDI2(i_Data);
            if SpeedButton4.Tag = 1 then SendI(i_data);
       end
     else
       begin
            I_data := 1000;
            if button3.Tag = 1 then SendDI1(i_data);
            if button4.Tag = 1 then SendDI2(i_data);
            if SpeedButton4.Tag = 1 then sendI(i_data);
       end;
end;
*****************************************************************************
procedure Tsupply.UpDown4Changing(Sender: TObject;
*****************************************************************************
var  AllowChange: Boolean;
begin
    i_Data:=(UpDown3.Position*100)+UpDown4.Position;
    if i_data < 1000 then
      begin
            if button3.Tag = 1 then SendDI1(i_Data);
            if button4.Tag = 1 then SendDI2(i_Data);
            if SpeedButton4.Tag = 1 then SendI(i_data);
      end
    else
      begin
            I_data := 1000;
            if button3.Tag = 1 then SendDI1(i_data);
            if button4.Tag = 1 then SendDI2(i_data);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if SpeedButton4.Tag = 1 then sendI(i_data);
        end ;
end;
*****************************************************************************
procedure Tstupply.SpeedButton4Click(Sender: TObject);
*****************************************************************************
begin
    if SpeedButton4.Tag = 0 then
      begin
                zled6.Enabled := true;
                SpeedButton4.tag := 1;
                button3.Enabled:=false;
                button4.Enabled:=false;
                button3.tag:=1;
                button4.tag:=1;
                i_Data:=(UpDown3.Position*10)+UpDown4.Position;
                if i_data < 1000 then sendI(i_data);
      end
    else
      begin
                zled6.Enabled := false;
                speedButton4.tag := 0;
                button3.Enabled:=true;
                button4.Enabled:=true;
                button3.tag:=0;
                button4.tag:=0;
      end;
end;
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Unit Generator;
Interface
Uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, zLed, Buttons, StdCtrls, zonzoff, zPanel,
    Zseg, ExtCtrls;
Type
    TFunc_G = class(TForm)
      Panel1: TPanel;
      Panel4: TPanel;
      Panel5: TPanel;
      Panel6: TPanel;
      Panel7: TPanel;
      dis1: TzcDotLabel;
      dis2: TzcDotLabel;
      MHz: TzKnob;
      KHz: TzKnob;
      Hz: TzKnob;
      Off: TzcSingleOnOff;
      Outp: TzcSingleOnOff;
      Button1: TButton;
      Button2: TButton;
      Button3: TButton;
      Button4: TButton;
      Button5: TButton;
      Button6: TButton;
      Button7: TButton;
      Button8: TButton;
      Label1: TLabel;
      Label2: TLabel;
      Label3: TLabel;
      Label4: TLabel;
      Label5: TLabel;
      Led1: TzLed;
      Led2: TzLed;
      GroupBox1: TGroupBox;
      SpeedButton1: TSpeedButton;
      SpeedButton2: TSpeedButton;
      SpeedButton3: TSpeedButton;
      GroupBox2: TGroupBox;
      SpeedButton4: TSpeedButton;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    SpeedButton5: TSpeedButton;
    SpeedButton6: TSpeedButton;
    SpeedButton7: TSpeedButton;
    SpeedButton8: TSpeedButton;
    SpeedButton9: TSpeedButton;
    zcLight(OnOff1: TzcLight(OnOff;
  procedure FormCreate(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure MHzChange(Sender: TObject);
  procedure KHzChange(Sender: TObject);
  procedure HzChange(Sender: TObject);
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure Button3Click(Sender: TObject);
  procedure Button4Click(Sender: TObject);
  procedure Button5Click(Sender: TObject);
  procedure Button6Click(Sender: TObject);
  procedure Button7Click(Sender: TObject);
  procedure Button8Click(Sender: TObject);
  procedure OffOff(Sender: TObject);
  procedure OffOn(Sender: TObject);
  procedure OutpOff(Sender: TObject);
  procedure OutpOn(Sender: TObject);
  procedure OffClick(Sender: TObject);
  procedure OutpClick(Sender: TObject);
  procedure SpeedButton1Click(Sender: TObject);
  procedure SpeedButton2Click(Sender: TObject);
  procedure SpeedButton3Click(Sender: TObject);
  procedure SpeedButton4Click(Sender: TObject);
  procedure SpeedButton5Click(Sender: TObject);
  procedure SpeedButton6Click(Sender: TObject);
  procedure SpeedButton7Click(Sender: TObject);
  procedure SpeedButton8Click(Sender: TObject);
  procedure SpeedButton9Click(Sender: TObject);
  procedure zcLight(OnOffTOff(Sender: TObject);
  procedure zcLight(OnOffTOn(Sender: TObject);
private
  { Private declarations }
public
  procedure Freq_Ch;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    { Public declarations }
  end;
{   Const
    ใส่ค่า Const ตามที่แสดงไว้ข้างต้นเพื่อเชื่อมต่อระหว่าง
    ไมโครคอมพิวเตอร์กับการ์ด GPIB          }
*****************************************************************************
procedure TFunc_G.Freq_Ch;
*****************************************************************************
var   F_data,a:string;
      n:integer;
begin
      writebuff:='FRQ:';
      F_temp:=(MHz.Value*1000000)+(KHz.Value*1000)+Hz.Value;
      F_data:=inttostr(F_temp);
      for n:=1 to Length(F_data) do
         begin
              a:= copy(F_data,n,1);
              If a='0' then writebuff[n+3]:='0';
              If a='1' then writebuff[n+3]:='1';
              If a='2' then writebuff[n+3]:='2';
              If a='3' then writebuff[n+3]:='3';
              If a='4' then writebuff[n+3]:='4';
              If a='5' then writebuff[n+3]:='5';
              If a='6' then writebuff[n+3]:='6';
              If a='7' then writebuff[n+3]:='7';
              If a='8' then writebuff[n+3]:='8';
              If a='9' then writebuff[n+3]:='9';
         end;
      writebuff[n+3]:=';';
      send(0,12,writebuff,length(writebuff),1);
      if (Ibsta and err)<> 0 then
        begin
      MessageDlg('Device error',mtError,[mboK],0);
                release;
        end;
     dis1.Caption:=writebuff;
     panel6.Caption:=readbuff;
end;
*****************************************************************************
procedure TFunc_G.FormCreate(Sender: TObject);
*****************************************************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
    LoadDll;
    SendIFC(0);
    F_temp:=0;
end;
*****************************************************************************
procedure TFunc_G.FormClose(Sender: TObject; var Action: TCloseAction);
*****************************************************************************
begin
    FreeLibrary(Gpib32Lib);
end;
*****************************************************************************
procedure TFunc_G.MHzChange(Sender: TObject);
*****************************************************************************
begin
    Freq_Ch;
end;
*****************************************************************************
procedure TFunc_G.KHzChange(Sender: TObject);
*****************************************************************************
begin
    Freq_Ch;
end;
*****************************************************************************
procedure TFunc_G.HzChange(Sender: TObject);
*****************************************************************************
begin
    Freq_ch;
end;

*****************************************************************************
procedure TFunc_G.Button1Click(Sender: TObject);
*****************************************************************************
begin
    writebuff:='FRQ?;';
    send(0,12,writebuff[1],length(writebuff),1);
    receive(0,12,readbuff[1],length(readbuff),1);
    panel6.Caption:=readbuff;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
*****************************************************************************
procedure TFunc_G.Button2Click(Sender: TObject);
*****************************************************************************
begin
      writebuff:='STA?;';
      send(0,12,writebuff,length(writebuff),1);
      receive(0,12,readbuff,length(readbuff),1);
      panel7.Caption:=readbuff;
end;
*****************************************************************************
procedure TFunc_G.Button3Click(Sender: TObject);
*****************************************************************************
begin
      writebuff:='STT?;';
      send(0,12,writebuff,length(writebuff),1);
      receive(0,12,readbuff,length(readbuff),1);
      panel7.Caption:=readbuff;
end;
*****************************************************************************
procedure TFunc_G.Button4Click(Sender: TObject);
*****************************************************************************
begin
      writebuff:='STP?;';
      send(0,12,writebuff,length(writebuff),1);
      receive(0,12,readbuff,length(readbuff),1);
      panel7.Caption:=readbuff;
end;
*****************************************************************************
procedure TFunc_G.Button5Click(Sender: TObject);
*****************************************************************************
begin
      writebuff:='SWT?;';
      send(0,12,writebuff,length(writebuff),1);
      receive(0,12,readbuff,length(readbuff),1);
      panel7.Caption:=readbuff;
end;
*****************************************************************************
procedure TFunc_G.Button6Click(Sender: TObject);
*****************************************************************************
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    writebuff:='WDT?;';
    send(0,12,writebuff,length(writebuff),1);
    receive(0,12,readbuff,length(readbuff),1);
    panel7.Caption:=readbuff;
end;
*****************************************************************************
procedure TFunc_G.Button7Click(Sender: TObject);
*****************************************************************************
begin
    writebuff:='AMP?;';
    send(0,12,writebuff,length(writebuff),1);
    receive(0,12,readbuff,length(readbuff),1);
    dis2.Caption:=readbuff;
    panel7.Caption:=readbuff;
end;
*****************************************************************************
procedure TFunc_G.Button8Click(Sender: TObject);
*****************************************************************************
begin
    writebuff:='OFS?;';
    send(0,12,writebuff,length(writebuff),1);
    receive(0,12,readbuff,length(readbuff),1);
    panel6.Caption:=readbuff;
end;
*****************************************************************************
procedure TFunc_G.OffOff(Sender: TObject);
*****************************************************************************
begin
    DevClear(0,12);
end;
*****************************************************************************
procedure TFunc_G.OffOn(Sender: TObject);
*****************************************************************************
begin
    writebuff:='OF1;';
    send(0,12,writebuff,length(writebuff),1);
end;
*****************************************************************************
procedure TFunc_G.OutpOff(Sender: TObject);
*****************************************************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
    DevClear(0,12);
end;
*****************************************************************************
procedure TFunc_G.OutpOn(Sender: TObject);
*****************************************************************************
begin
    writebuff:='OT1;';
    send(0,12,writebuff,length(writebuff),1);
end;
*****************************************************************************
procedure TFunc_G.OffClick(Sender: TObject);
*****************************************************************************
begin
    If Off.Tag=0 then
      begin
          Led1.Blink:=True;
          Off.Tag:=1;
      end
    else
      begin
          Led1.Blink:=false;
          Off.Tag:=0;
      end;
end;
*****************************************************************************
procedure TFunc_G.OutpClick(Sender: TObject);
*****************************************************************************
begin
    If Outp.Tag=0 then
      begin
          Led2.Blink:=True;
          Outp.Tag:=1;
      end
    else
      begin
          Led2.Blink:=false;
          Outp.Tag:=0;
      end;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
*****************************************************************************
procedure TFunc_G.SpeedButton1Click(Sender: TObject);
*****************************************************************************
begin
      writebuff:='TRM;';
      send(0,12,writebuff,length(writebuff),1);
end;
*****************************************************************************
procedure TFunc_G.SpeedButton2Click(Sender: TObject);
*****************************************************************************
begin
      writebuff:='GTM;';
      send(0,12,writebuff,length(writebuff),1);
end;
*****************************************************************************
procedure TFunc_G.SpeedButton3Click(Sender: TObject);
*****************************************************************************
begin
      writebuff:='CTM;';
      send(0,12,writebuff,length(writebuff),1);
end;
*****************************************************************************
procedure TFunc_G.SpeedButton4Click(Sender: TObject);
*****************************************************************************
begin
      writebuff:='RMP;';
      send(0,12,writebuff,length(writebuff),1);
end;
*****************************************************************************
procedure TFunc_G.SpeedButton5Click(Sender: TObject);
*****************************************************************************
begin
      writebuff:='TRI;';
      send(0,12,writebuff,length(writebuff),1);
end;
*****************************************************************************
procedure TFunc_G.SpeedButton6Click(Sender: TObject);
*****************************************************************************
begin
      writebuff:='SIN;';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    send(0,12,writebuff,length(writebuff),1);
end;
*****************************************************************************
procedure TFunc_G.SpeedButton7Click(Sender: TObject);
*****************************************************************************
begin
    writebuff:='SQR;';
    send(0,12,writebuff,length(writebuff),1);
end;
*****************************************************************************
procedure TFunc_G.SpeedButton8Click(Sender: TObject);
*****************************************************************************
begin
    writebuff:='PLS;';
    send(0,12,writebuff,length(writebuff),1);
end;
*****************************************************************************
procedure TFunc_G.SpeedButton9Click(Sender: TObject);
*****************************************************************************
begin
    writebuff:='ARB;';
    send(0,12,writebuff,length(writebuff),1);
end;
*****************************************************************************
procedure TFunc_G.zcLightOnOffClick(Sender: TObject);
*****************************************************************************
begin
    writebuff:='RM0;';
    send(0,12,writebuff,length(writebuff),1);
    MHz.Enabled:=false;
    KHz.Enabled:=false;
    Hz.Enabled:=false;
    speedbutton1.Enabled:=false;
    speedbutton2.Enabled:=false;
    speedbutton3.Enabled:=false;
    speedbutton4.Enabled:=false;
    speedbutton5.Enabled:=false;
    speedbutton6.Enabled:=false;
    speedbutton7.Enabled:=false;
    speedbutton8.Enabled:=false;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    speedbutton9.Enabled:=false;
    MHz.Value:=0;
    KHz.Value:=0;
    Hz.Value:=0;
    dis1.Caption:='  00.00 Hz';
    dis2.Caption:='  00.00 V';
end;
*****************************************************************************
procedure TFunc_G.zcLightOnOffTOn(Sender: TObject);
*****************************************************************************
begin
    devclear(0,12);
    MHz.Enabled:=True;
    KHz.Enabled:=True;
    Hz.Enabled:=True;
    speedbutton1.Enabled:=True;
    speedbutton2.Enabled:=True;
    speedbutton3.Enabled:=True;
    speedbutton4.Enabled:=True;
    speedbutton5.Enabled:=True;
    speedbutton6.Enabled:=True;
    speedbutton7.Enabled:=True;
    speedbutton8.Enabled:=True;
    speedbutton9.Enabled:=True;
end;

end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Unit scope;

Interface

Uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,

TeEngine, Series, ExtCtrls, TeeProcs, Chart, StdCtrls, TeeFunci,

Buttons, Fourier,Math, RCiart, MPlayer;

Type

TForm1 = class(TForm)

ComboBox2: TComboBox;

Button5: TButton;

Panel1: TPanel;

Button1: TButton;

ComboBox1: TComboBox;

RadioButton1: TRadioButton;

RadioButton2: TRadioButton;

RadioButton3: TRadioButton;

Panel2: TPanel;

Button2: TButton;

ComboBox3: TComboBox;

RadioButton4: TRadioButton;

RadioButton5: TRadioButton;

RadioButton6: TRadioButton;

Panel3: TPanel;

Button3: TButton;

RadioButton7: TRadioButton;

RadioButton8: TRadioButton;

RadioButton9: TRadioButton;

RadioButton10: TRadioButton;

ComboBox4: TComboBox;

FFT1: TFastFourier;

Button4: TButton;

RChart2: TRChart;

RChart1: TRChart;

Panel4: TPanel;

Panel5: TPanel;

Panel6: TPanel;

Edit1: TEdit;

Edit2: TEdit;

OpenDialog1: TOpenDialog;

SaveDialog1: TSaveDialog;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    RChart3: TRChart;
    Button8: TButton;
    Edit3: TEdit;
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure FormCreate(Sender: TObject);
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure Button3Click(Sender: TObject);
  procedure RadioButton7Click(Sender: TObject);
  procedure RadioButton9Click(Sender: TObject);
  procedure RadioButton8Click(Sender: TObject);
  procedure RadioButton10Click(Sender: TObject);
  procedure RadioButton1Click(Sender: TObject);
  procedure RadioButton2Click(Sender: TObject);
  procedure RadioButton3Click(Sender: TObject);
  procedure RadioButton4Click(Sender: TObject);
  procedure RadioButton5Click(Sender: TObject);
  procedure RadioButton6Click(Sender: TObject);
  procedure Button5Click(Sender: TObject);
  procedure ComboBox1Change(Sender: TObject);
  procedure ComboBox3Change(Sender: TObject);
  procedure ComboBox2Change(Sender: TObject);
  procedure ComboBox4Change(Sender: TObject);
  procedure Button4Click(Sender: TObject);
  procedure Button7Click(Sender: TObject);
  procedure Button6Click(Sender: TObject);
  procedure Button8Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
{ Const
  ใส่ค่า Const ตามที่แสดงไว้ข้างล่างเพื่อเชื่อมต่อระหว่าง
  ไมโครคอมพิวเตอร์กับการ์ด GPIB          }
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  ibsic(Gpib32.dll);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
*****************************************************************************
procedure TForm1.FormCreate(Sender: TObject);
*****************************************************************************
begin
    LoadDll;
    SendIFC(0);
end;
*****************************************************************************
procedure TForm1.Button1Click(Sender: TObject);
*****************************************************************************
var   i,j:integer;
      y:packed array[0..2048] of double;
begin
    writebuff:='select:ch1 on';
    send(0,1,writebuff,length(writebuff),1);
    writebuff:='data:source ch1';
    send(0,1,writebuff,length(writebuff),1);
    writebuff:='data:encdg rpbinary';
    send(0,1,writebuff,length(writebuff),1);
    writebuff:='data:width 1';
    send(0,1,writebuff,length(writebuff),1);
    writebuff:='data:start 1';
    send(0,1,writebuff,length(writebuff),1);
    writebuff:='data:stop 1000';
    send(0,1,writebuff,length(writebuff),1);
    writebuff:='wfmpre:ch1?';
    send(0,1,writebuff,length(writebuff),1);
    writebuff:='curve?';
    send(0,1,writebuff,length(writebuff),1);
    sleep(1000);
    receive(0,1,rdbuff,length(rdbuff),1);
    FFT1.Clear;
    FFT1.ClearImag;
    Rchart1.DataColor:=clred;
    Rchart1.MoveTo(0,0);
          for i:=13 to 600 do
              begin
                  Rchart1.DrawTo(i-13,rdbuff[i]-125);
                  FFT1.RealSpec[i]:=rdbuff[i]-125;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                end;
        Rchart1.ShowGraf;
        fft1.Transform;
end;
*****************************************************************************
procedure TForm1.Button2Click(Sender: TObject);
*****************************************************************************
var i,j:integer;
    y:packed array[0..2048] of double;
begin
        writebuff:='select:ch2 on';
        send(0,1,writebuff,length(writebuff),1);
        writebuff:='data:source ch2';
        send(0,1,writebuff,length(writebuff),1);
        writebuff:='data:encdg rpbinary';
        send(0,1,writebuff,length(writebuff),1);
        writebuff:='data:width 1';
        send(0,1,writebuff,length(writebuff),1);
        writebuff:='data:start 1';
        send(0,1,writebuff,length(writebuff),1);
        writebuff:='data:stop 1000';
      send(0,1,writebuff,length(writebuff),1);
      writebuff:='wfmpre:ch2?';
      send(0,1,writebuff,length(writebuff),1);
      writebuff:='curve?';
      send(0,1,writebuff,length(writebuff),1);
      sleep(1000);
      receive(0,1,rdbuff,length(rdbuff),1);
      FFT1.Clear;
      FFT1.ClearImag;
      Rchart1.DataColor:=clred;
      Rchart1.MoveTo(0,0);
            for i:=13 to fft1.Spectrumsize-24 do
              begin
                    Rchart1.DrawTo(i-13,rdbuff[i]-125);
                  FFT1.RealSpec[i]:=rdbuff[i]-125;
              end;
      Rchart1.ShowGraf;
      FFT1.Transform;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
*****************************************************************************
procedure TForm1.Button3Click(Sender: TObject);
*****************************************************************************
var  i:integer;
begin
      writebuff:='select:math on';
      send(0,1,writebuff,length(writebuff),1);
      writebuff:='data:source math1';
      send(0,1,writebuff,length(writebuff),1);
      writebuff:='data:encdg rpbinary';
      send(0,1,writebuff,length(writebuff),1);
      writebuff:='data:width 1';
      send(0,1,writebuff,length(writebuff),1);
      writebuff:='data:start 1';
      send(0,1,writebuff,length(writebuff),1);
      writebuff:='data:stop 1000';
      send(0,1,writebuff,length(writebuff),1);
      writebuff:='wfmpre:math1?';
      send(0,1,writebuff,length(writebuff),1);
      writebuff:='curve?';
      send(0,1,writebuff,length(writebuff),1);
      sleep(1000);
      receive(0,1,rdbuff,length(rdbuff),1);
      Rchart2.ClearGraf;        {ClearScreen}
      Rchart2.DataColor:=clGreen;
      Rchart2.MoveTo(0,0);
            for i:=0 to 1000 do
              begin
                  Rchart2.DrawTo(i div 2,(rdbuff[i+10])-95);
              end;
      Rchart2.ShowGraf;
end;
*****************************************************************************
procedure TForm1.RadioButton7Click(Sender: TObject);
*****************************************************************************
begin
      writebuff:='math1:define "ch1+ch2"';
      send(0,1,writebuff,length(writebuff),1);
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
*****************************************************************************
procedure TForm1.Button6Click(Sender: TObject);
*****************************************************************************
begin
    RChart1.ClearGraf;
end;
*****************************************************************************
procedure TForm1.RadioButton9Click(Sender: TObject);
*****************************************************************************
begin
      writebuff:='math1:define "ch2-ch1"';
      send(0,1,writebuff,length(writebuff),1);
end;
*****************************************************************************
procedure TForm1.RadioButton8Click(Sender: TObject);
*****************************************************************************
begin
      writebuff:='math1:define "ch1-ch2"';
      send(0,1,writebuff,length(writebuff),1);
end;
*****************************************************************************
procedure TForm1.RadioButton10Click(Sender: TObject);
*****************************************************************************
begin
      writebuff:='math1:define "ch1*ch2"';
      send(0,1,writebuff,length(writebuff),1);
end;
*****************************************************************************
procedure TForm1.RadioButton1Click(Sender: TObject);
*****************************************************************************
begin
      writebuff:='ch1:coupling dc';
      send(0,1,writebuff,length(writebuff),1);
end;
*****************************************************************************
procedure TForm1.RadioButton2Click(Sender: TObject);
*****************************************************************************
begin
      writebuff:='ch1:coupling ac';
      send(0,1,writebuff,length(writebuff),1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;

*****************************************************************************
procedure TForm1.RadioButton3Click(Sender: TObject);
*****************************************************************************
begin
      writebuff:='ch1:coupling gnd';
      send(0,1,writebuff,length(writebuff),1);
end;
*****************************************************************************
procedure TForm1.RadioButton4Click(Sender: TObject);
*****************************************************************************
begin
      writebuff:='ch2:coupling dc';
      send(0,1,writebuff,length(writebuff),1);
end;
*****************************************************************************
procedure TForm1.RadioButton5Click(Sender: TObject);
*****************************************************************************
begin
      writebuff:='ch2:coupling ac';
      send(0,1,writebuff,length(writebuff),1);
end;
*****************************************************************************
procedure TForm1.RadioButton6Click(Sender: TObject);
*****************************************************************************
begin
      writebuff:='ch2:coupling gnd';
      send(0,1,writebuff,length(writebuff),1);
end;
*****************************************************************************
procedure TForm1.Button5Click(Sender: TObject);
*****************************************************************************
begin
      writebuff:='ClearMenu';
      send(0,1,writebuff,length(writebuff),1);
end;
*****************************************************************************
procedure TForm1.ComboBox1Change(Sender: TObject);
*****************************************************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
    if combobox1.Text='10  V/Div' then
      begin
            writebuff:='ch1:scale 1e1';
          send(0,1,writebuff,length(writebuff),1);
      end;
    if combobox1.Text=' 5   V/Div' then
      begin
            writebuff:='ch1:scale 5e0';
            send(0,1,writebuff,length(writebuff),1);
      end;
    if combobox1.Text=' 2   V/Div' then
      begin
          writebuff:='ch1:scale 2e0';
            send(0,1,writebuff,length(writebuff),1);
      end;
    if combobox1.Text=' 1   V/Div' then
      begin
           writebuff:='ch1:scale 1e0';
          send(0,1,writebuff,length(writebuff),1);
      end;
    if combobox1.Text='500  mV/Div' then
      begin
          writebuff:='ch1:scale 5e-1';
          send(0,1,writebuff,length(writebuff),1);
      end;
    if combobox1.Text='200  mV/Div' then
      begin
          writebuff:='ch1:scale 2e-1';
          send(0,1,writebuff,length(writebuff),1);
      end;
    if combobox1.Text='100  mV/Div' then
      begin
          writebuff:='ch1:scale 1e-1';
          send(0,1,writebuff,length(writebuff),1);
      end;
    if combobox1.Text='50   mV/Div' then
      begin
          writebuff:='ch1:scale 5e-2';
          send(0,1,writebuff,length(writebuff),1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    end;
  if combobox1.Text='20   mV/Div' then
    begin
        writebuff:='ch1:scale 2e-2';
        send(0,1,writebuff,length(writebuff),1);
    end;
  if combobox1.Text='10   mV/Div' then
    begin
        writebuff:='ch1:scale 1e-2';
        send(0,1,writebuff,length(writebuff),1);
    end;
  if combobox1.Text='  5   mV/Div' then
    begin
        writebuff:='ch1:scale 5e-3';
        send(0,1,writebuff,length(writebuff),1);
    end;
  if combobox1.Text='  2   mV/Div' then
    begin
        writebuff:='ch1:scale 2e-3';
        send(0,1,writebuff,length(writebuff),1);
    end;
end;
*****************************************************************************
procedure TForm1.ComboBox3Change(Sender: TObject);
*****************************************************************************
begin
  if combobox3.Text='10  V/Div' then
    begin
        writebuff:='ch2:scale 1e1';
        send(0,1,writebuff,length(writebuff),1);
    end;
  if combobox3.Text='  5  V/Div' then
    begin
        writebuff:='ch2:scale 5e0';
        send(0,1,writebuff,length(writebuff),1);
    end;
  if combobox3.Text='  2  V/Div' then
    begin
        writebuff:='ch2:scale 2e0';
        send(0,1,writebuff,length(writebuff),1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  end;
if combobox3.Text=' 1   V/Div' then
  begin
       writebuff:='ch2:scale 1e0';
       send(0,1,writebuff,length(writebuff),1);
  end;
if combobox3.Text='500  mV/Div' then
  begin
       writebuff:='ch2:scale 5e-1';
       send(0,1,writebuff,length(writebuff),1);
  end;
if combobox3.Text='200  mV/Div' then
  begin
       writebuff:='ch2:scale 2e-1';
       send(0,1,writebuff,length(writebuff),1);
  end;
if combobox3.Text='100  mV/Div' then
  begin
       writebuff:='ch2:scale 1e-1';
       send(0,1,writebuff,length(writebuff),1);
  end;
if combobox3.Text='50   mV/Div' then
  begin
       writebuff:='ch2:scale 5e-2';
       send(0,1,writebuff,length(writebuff),1);
  end;
if combobox3.Text='20   mV/Div' then
  begin
       writebuff:='ch2:scale 2e-2';
       send(0,1,writebuff,length(writebuff),1);
  end;
if combobox3.Text='10   mV/Div' then
  begin
       writebuff:='ch2:scale 1e-2';
       send(0,1,writebuff,length(writebuff),1);
  end;
if combobox3.Text=' 5   mV/Div' then
  begin
       writebuff:='ch2:scale 5e-3';
       send(0,1,writebuff,length(writebuff),1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    end;
  if combobox3.Text=' 2   mV/Div' then
    begin
        writebuff:='ch2:scale 2e-3';
        send(0,1,writebuff,length(writebuff),1);
    end;
end;
*****************************************************************************
procedure TForm1.ComboBox2Change(Sender: TObject);
*****************************************************************************
begin
  if combobox2.Text='2.5  ms/Div' then
    begin
        writebuff:='Horizontal:Main:scale 2.5e-3';
        send(0,1,writebuff,length(writebuff),1);
        edit1.Text:=' 2.5 ms ';
        edit2.text:=' 500 Hz ';
    end;
  if combobox2.Text=' 1   ms/Div' then
    begin
        writebuff:='Horizontal:Main:scale 1e-3';
        send(0,1,writebuff,length(writebuff),1);
        edit1.Text:=' 1 ms ';
        edit2.text:=' 1.25 KHz ';
    end;
  if combobox2.Text=' 10  ns/Div' then
    begin
        writebuff:='Horizontal:Main:scale 1e-8';
        send(0,1,writebuff,length(writebuff),1);
        edit1.Text:=' 10 ns ';
        edit2.text:=' 125 MHz ';
    end;
  if combobox2.Text=' 25  ns/Div' then
    begin
        writebuff:='Horizontal:Main:scale 2.5e-8';
        send(0,1,writebuff,length(writebuff),1);
        edit1.Text:=' 25 ns ';
        edit2.text:=' 50 MHz ';
    end;
  if combobox2.Text=' 50  ns/Div' then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  begin
      writebuff:='Horizontal:Main:scale 5e-8';
      send(0,1,writebuff,length(writebuff),1);
      edit1.Text:=' 50 ns ';
      edit2.text:=' 25 MHz ';
  end;
if combobox2.Text='100  ns/Div' then
  begin
      writebuff:='Horizontal:Main:scale 1e-7';
      send(0,1,writebuff,length(writebuff),1);
      edit1.Text:=' 100 ns ';
      edit2.text:=' 12.5 MHz ';
  end;
if combobox2.Text='250  ns/Div' then
  begin
      writebuff:='Horizontal:Main:scale 2.5e-7';
      send(0,1,writebuff,length(writebuff),1);
      edit1.Text:=' 250 ns ';
      edit2.text:=' 5 MHz ';
  end;
if combobox2.Text='500  ns/Div' then
  begin
      writebuff:='Horizontal:Main:scale 5e-7';
      send(0,1,writebuff,length(writebuff),1);
      edit1.Text:=' 500 ns ';
      edit2.text:=' 2.5 MHz ';
  end;
if combobox2.Text='  1    us/Div' then
  begin
      writebuff:='Horizontal:Main:scale 1e-6';
      send(0,1,writebuff,length(writebuff),1);
      edit1.Text:=' 1 us ';
      edit2.text:=' 1.25 MHz ';
  end;
if combobox2.Text='2.5   us/Div' then
  begin
      writebuff:='Horizontal:Main:scale 2.5e-6';
      send(0,1,writebuff,length(writebuff),1);
      edit1.Text:=' 2.5 us ';
      edit2.text:=' 500 KHz ';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    end;
  if combobox2.Text=' 5   us/Div' then
    begin
          writebuff:='Horizontal:Main:scale 5e-6';
          send(0,1,writebuff,length(writebuff),1);
          edit1.Text:=' 5 us ';
          edit2.text:=' 250 KHz ';
    end;
  if combobox2.Text=' 10   us/Div' then
    begin
          writebuff:='Horizontal:Main:scale 1e-5';
          send(0,1,writebuff,length(writebuff),1);
          edit1.Text:=' 10 us ';
          edit2.text:=' 125 KHz ';
    end;
  if combobox2.Text=' 25   us/Div' then
    begin
          writebuff:='Horizontal:Main:scale 2.5e-5';
          send(0,1,writebuff,length(writebuff),1);
          Edit1.Text:='   25 us';
          Edit2.Text:='   50 KHz';
    end;
  if combobox2.Text=' 50   us/Div' then
    begin
          writebuff:='Horizontal:Main:scale 5e-5';
          send(0,1,writebuff,length(writebuff),1);
          Edit1.Text:='   50 us';
          Edit2.Text:='   25 KHz';
    end;
  if combobox2.Text='100  us/Div' then
    begin
          writebuff:='Horizontal:Main:scale 1e-4';
          send(0,1,writebuff,length(writebuff),1);
          Edit1.Text:='  100 us';
          Edit2.Text:='  12.5 KHz';
    end;
  if combobox2.Text='250  us/Div' then
    begin
          writebuff:='Horizontal:Main:scale 2.5e-4';
          send(0,1,writebuff,length(writebuff),1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            edit1.Text:=' 250 us ';
            edit2.text:=' 5 KHz ';
        end;
    if combobox2.Text='500  us/Div' then
        begin
            writebuff:='Horizontal:Main:scale 5e-4';
            send(0,1,writebuff,length(writebuff),1);
            edit1.Text:=' 500 us ';
            edit2.text:=' 2.5 KHz ';
        end;
end;
*****************************************************************************
procedure TForm1.ComboBox4Change(Sender: TObject);
*****************************************************************************
begin
    if combobox4.Text='FFT channel1' then
        begin
            writebuff:='math:define "fft(ch1)"';
            send(0,1,writebuff,length(writebuff),1);
        end
    else
        begin
            writebuff:='math:define "fft(ch2)"';
            send(0,1,writebuff,length(writebuff),1);
        end;
end;

*****************************************************************************
procedure TForm1.Button4Click(Sender: TObject);
*****************************************************************************
var    i:integer;
begin
    Rchart2.ClearGraf;        {ClearScreen}
    Rchart2.DataColor:=clred;
    Rchart2.MoveTo(fft1.freqofline(0,0.001),0);
    for i:=0 to (fft1.spectrumsize div 2) do
        begin
            Rchart2.DrawTo(fft1.freqofline(i,0.001),fft1.Magnitude[i]);
        end;
    Rchart2.ShowGraf;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
*****************************************************************************
procedure TForm1.Button7Click(Sender: TObject);
*****************************************************************************
var   F:textFile;
      S:String;
begin
     if OpenDialog1.Execute then
       begin
            AssignFile(F,Opendialog1.Filename);
            Reset(F);
              Readln(f,S);
              CloseFile(F);
       end;
end;
*****************************************************************************
procedure TForm1.Button8Click(Sender: TObject);
*****************************************************************************
var   f:file of byte;
     op01:array of byte;
     size:longint;
     s,a:string;
     i,n,pos:integer;
begin
     if OpenDialog1.Execute then
       begin
            AssignFile(f,openDialog1.Filename);
             Reset(f);
             Size:=FileSize(f);
             S:= Inttostr(size);
             SetLength(op01,size);
             Rchart3.ClearGraf;
             Rchart3.Datacolor:=clBlue;
             Rchart3.moveto(0,0);
             for i:= 5 to size do
             begin
                  read(f,op01[i]);
                  Rchart3.drawto(i-5,op01[j]);
               end;
            Rchart3.Showgraf;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            a:=edit3.Text;
            pos:=strtoint(edit3.text);
            for i:= 0 to fft1.SpectrumSize do
              begin
                    fft1.RealSpec[i]:=op01[i+pos];
              end;
            FFT1.Transform;
        end;
end;
end.
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี โดยได้รับความช่วยเหลือและสนับสนุนจากบุคคลหลายท่าน ผู้จัดทำขอขอบคุณเป็นอย่างสูง กล่าวคือ

อาจารย์ที่ปรึกษา รส.ดร. มนัส สังวรศิลป์ ผู้ชี้แนะและคอยกระตุ้นให้ผลงานนี้สำเร็จ

พี่เปี๊ยก พี่อั๋น พี่เกษม พี่นนท์ พี่เทพ ผู้ให้คำแนะนำและคำปรึกษาต่างๆ ตลอดจนช่วยแก้ปัญหาเกี่ยวกับความผิดพลาดต่างๆของโปรแกรมที่เกิดขึ้น

บิดา มารดา ผู้ให้กำลังใจเสมอ

เพื่อนๆ ชั้น 4C ที่ให้คำแนะนำในการเขียนโปรแกรม

ธนาภรณ์ รัตนเมธาวี

(นางสาวธนาภรณ์ รัตนเมธาวี)

นิษาวัฒน์ เทิดสิทธิกุล

(นางสาวนิษารัตน์ เทิดสิทธิกุล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. กนก เจนจิระพงศ์เวช, "GPIB (IEEE- 488) บัสอินเตอร์เฟส มาตรฐานและการใช้" , วารสารเซมิคอนดักเตอร์, ฉบับที่ 78 (พฤษภาคม,สิงหาคม,กันยายน 2530) .
2. ธานินทร์ ถาวรศาสนวงศ์ และ ทินกร ดุ๊ก , "การอินเตอร์เฟส IBM PC" , กรุงเทพฯ ฟิสิกส์เซ็นเตอร์ การพิมพ์ , ม.ป.ม.
3. นพดล มณีรัตน์, "การออกแบบและสร้างระบบการส่งข้อมูลแบบขนาน สำหรับการควบคุมเครื่องมือวัดผ่านระบบบัสมาตรฐาน" วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต, คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2540.
4. วัลลภ สุระกำพลธร, "การประมวลผลสัญญาณเชิงเลข" ,บริษัทไดนาพริ้นท์ จำกัด , 89-113 ,2533.
5. NI – 488.2 M Function Reference Manual for Win32 ,National Instrument Corporation , 1996.
6. NI – 488.2 M Function Reference Manual for Windows 95 and window NT , National Instrument Corporation , 1996.
7. L. Rantanen , "Using NI – 488.2 M Software with 32 – bit Delphi Application" , Application Note 085 , National Instrument.
8. Programmer Manual, "TDS340, TDS360 & TDS380 Digital Real – Time Oscilloscope", Tektronix Instrument Corporation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้