สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การควบคุมเครื่องมือวัดผ่านระบบ GPIB (IEEE-488)

INSTRUMENT CONTROLLING VIA GPIB (IEEE-488)

T061506

โดย

นาย จุมภฏ ปลีคงธุ รหัส 44010080

นางสาว สุภาพร ทองจันทร์ รหัส 44010546

อาจารย์ที่ปรึกษา

รศ.ดร. มนัส สังวรศิลป์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน..... 61506
วัน,เดือน,ปี..... 18 ก.ค. 2549

.b.....
.i.....

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายงาน ปีการศึกษา 2547

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การควบคุมเครื่องมือวัดผ่านระบบ GPIB (IEEE-488)

ผู้จัดทำ

1. นาย จุมภฏ ปลีคงธุ รหัส 44010080
2. นางสาว สุภาพร ทองจันทร์ รหัส 44010546

.....อาจารย์ที่ปรึกษา

(รศ.ดร. มนัส สังวรศิลป์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมเครื่องมือวัดผ่านระบบ GPIB (IEEE-488)

INSTRUMENT CONTROLLING VIA GPIB (IEEE-488)

นาย จุมภฏ ปลีคงธุ 44010080

นางสาว สุภาพร ทองจันทร์ 44010546

โครงงานนี้ได้รับการตรวจสอบแล้ว พร้อมที่จะทำการสอบได้

.....

(รศ.ดร. มนัส สังวรศิลป์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมเครื่องมือวัดผ่านระบบ GPIB (IEEE-488)

นายจุมภฏ ปลีคงธุ
นางสาวสุภาพร ทองจันทร์
รศ.ดร.มนัส สังวรศิลป์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2547

บทคัดย่อ

โครงงานนี้จะเป็นการพัฒนาโปรแกรมเพื่อควบคุมเครื่องมือวัดทางอิเล็กทรอนิกส์ โดยผ่านระบบบัสมาตรฐานแบบ IEEE-488 หรือที่เรียกว่า GPIB (GENERAL PURPOSE INTERFACE BUS) ที่ใช้กันเป็นส่วนใหญ่ในงานอุตสาหกรรมทั่วไป โดยโปรแกรมที่เราได้ใช้บนคอมพิวเตอร์นั้นเราได้ใช้โปรแกรม Microsoft Visual Basic 6.0 เพื่อควบคุมเครื่องมือวัดต่างๆ ดังนี้

1) Programmable Power Supply รุ่น HM-8142 ของบริษัท HAMEG

2) Programmable Function Generator รุ่น HM-8130 ของบริษัท HAMEG

3) Storage Analog/Digital Oscilloscope รุ่น HM 1007 ของบริษัท HAMEG

แต่สำหรับ Oscilloscope นั้นไม่สามารถคอลโทรลได้เพียงแต่นำภาพที่แสดงบนจอภาพมาแสดงบนคอมพิวเตอร์เท่านั้น อย่างไรก็ตาม โครงงานชิ้นนี้ยังสามารถสั่งงานโดยผ่านระบบเน็ตเวิร์คได้อีกด้วย ซึ่งจะทำให้การใช้งานมีความสะดวกเพิ่มมากขึ้นอีกมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ทางคณะผู้จัดทำรายงานขอกราบขอบพระคุณเป็นอย่างสูงต่อ รองศาสตราจารย์ ดร. มนัส สังวรศิลป์ อาจารย์ที่ปรึกษา ซึ่งให้คำแนะนำในการจัดทำโครงงาน และ ปริญญานิพนธ์ฉบับนี้จนเสร็จสมบูรณ์ และขอขอบคุณพี่ๆ เพื่อนๆ ทุกคน ที่กรุณาได้ให้คำแนะนำในด้านต่างๆ และคอยเป็นกำลังใจในทุกๆด้าน ซึ่งทำให้ปริญญานิพนธ์ฉบับนี้เสร็จสมบูรณ์ยิ่งขึ้น

.....

(นาย จุมภฏ ปลีคงธุ)

(นางสาว สุภาพร ทองจันทร์)

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INSTRUMENT CONTROLLING VIA GPIB (IEEE-488)

Mr. Jumpod Plekhongthu

Miss Supaporn Tongchan

Assoc.Prof.Dr. Manas Sungworasil(Advisor)

2004

Abstract

This project presents a program developed for controlling IEEE-488 standard bus or GPIB (GENERAL PURPOSE INTERFACE BUS). It used widespreadly in the industry now. The program on PC is developed on Microsoft Visual Basic 6.0 for controlling some instruments as followed

1) Programmable Power Supply (HM8142 from HAMEG Company)
2) Programmable Function Generator (HM8130 from HAMEG Company)
3) Storage Analog/Digital Oscilloscope (HM1007 from HAMEG Company)

However the oscilloscope can be control of only captured the traces showing on PC . This project applies the advantages of network system ,we can control this project from the PC connected in our network . It make more comfortably to use this project.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
	บทคัดย่อ	I
	Abstract	II
	กิตติกรรมประกาศ	III
	สารบัญ	IV
บทที่1	บทนำ	1
บทที่ 2	IEEE-488 (GPIB)	4
	2.1 โครงสร้างของ IEEE-488	4
	2.2 ขีดจำกัดของ IEEE – 488	4
	2.3 รายละเอียดเกี่ยวกับ IEEE-488	5
	2.4 ความหมายของสัญญาณต่างๆภายใน IEEE –488	7
	2.5 ขบวนการแฮนด์เช็ค (Handshake Procedure)	8
	2.6 การเชื่อมต่ออุปกรณ์ต่างๆในระบบบัสมาตรฐาน IEEE-488 (GPIB)	12
บทที่ 3	คุณสมบัติโดยทั่วไปของระบบบัส USB	14
	3.1 คุณสมบัติเด่นของระบบบัส USB	14
	3.2 การส่งถ่ายข้อมูลบนระบบบัส USB	15
	3.3 องค์ประกอบทางด้านซอฟท์แวร์	16
	3.4 องค์ประกอบทางด้านฮาร์ดแวร์	17
	3.5 รูปแบบการส่งถ่ายข้อมูลบนระบบบัส USB	18
	3.6 ดิสคริปเตอร์ : ความหมาย , ชนิดและรูปแบบการทำงาน	19
	3.7 การจัดการกับอุปกรณ์บนบัส USB ที่มีความเร็วต่างกัน	21
	3.8 การส่งสัญญาณในบัสUSB	21
	3.9 กระบวนการกำหนดการทำงานของอุปกรณ์	22
	3.10 ระบบบัส USB 2.0	22
บทที่ 4	ระบบเครือข่าย (Network)	24
	4.1 ระบบ TCP/IP	24
	4.2 Server & Client	25
บทที่ 5	คำสั่งในการควบคุมอุปกรณ์เครื่องมือวัด	27
	5.1 HAMEG Programmable Power Supply รุ่น HM8142	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
5.2 HAMEG Programmable Function Generator รุ่น HM8130	31
5.3 HAMEG Storage Analog/Digital Oscilloscope รุ่น HM 1007	39
บทที่ 6 การออกแบบส่วนการอินเตอร์เฟสของระบบ	42
6.1 Microcontroller	42
6.2 USB Converter	44
6.3 RS-232	45
6.4 Regulator	45
บทที่ 7การทำงานภายในไมโครคอลโทรลเลอร์	48
7.1 ขบวนการติดต่อไปยังอุปกรณ์เครื่องมือวัด	48
7.2 รูปแบบของชุดคำสั่งที่คอมพิวเตอร์ส่งมายังบอร์ดอินเตอร์เฟส GPIB	53
7.3 โปรแกรมการทำงานของไมโครคอลโทรลเลอร์	53
บทที่ 8 การทดลองและการประยุกต์ใช้งาน	55
8.1 การทดลองควบคุม Power Supply รุ่น HM8142	57
8.2 การทดลองควบคุม Function Generator รุ่น HM8130	62
8.3 การทดลองควบคุม Analog/Digital Oscilloscope รุ่น HM1007	67
บทที่ 9 สรุปและข้อเสนอแนะในการนำไปพัฒนาต่อ	72
เอกสารอ้างอิง	
ภาคผนวก	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 1.1 Block Diagrams ของโครงงานโดยรวม	3
รูปที่ 2.1 แสดงตำแหน่งของขาสัญญาณตามมาตรฐาน IEEE-488(GPIB)	5
รูปที่ 2.2 แสดงองค์ประกอบของสัญญาณในระบบ	6
รูปที่ 2.3 แสดงความสัมพันธ์ของสัญญาณควบคุมการรับส่งข้อมูล	8
รูปที่ 2.4 แสดงแผนผังเวลาของขบวนการแฮนด์เช็ค	9
รูปที่ 2.5 แสดงขบวนการแฮนด์เช็คเมื่อใช้ไมโครคอมพิวเตอร์ เป็นตัวควบคุมและตัวส่ง	10
รูปที่ 2.6 แสดงขบวนการแฮนด์เช็คเมื่อใช้ไมโครคอมพิวเตอร์ เป็นตัวควบคุมและตัวรับ	11
รูปที่ 2.7 รูปแสดงการต่อแบบเรียงต่อเนื่องกัน (Daisy Chain Configuration)	12
รูปที่ 2.8 รูปแสดงการต่อแบบกระจาย (Star Configuration)	13
รูปที่ 6.1 แสดงการเชื่อมต่อระหว่างวงจรส่วนไมโครคอลโทรลเลอร์ และ พอร์ต GPIB	43
รูปที่ 6.2 แสดงวงจรในส่วนของ USB Convertor	44
รูปที่ 6.3 แสดงวงจรในส่วนของ RS-232	45
รูปที่ 6.4 วงจรในส่วนของ Regulator	46
รูปที่ 6.5 Block Diagrams ของวงจรรวม	46
รูปที่ 6.6 วงจรรวมของบอร์ดไมโครคอนโทรลเลอร์	47
รูปที่ 7.1 แสดงผังเวลาเมื่อตัวควบคุมติดต่อไปยังภาคส่ง	49
รูปที่ 7. 2 แสดงผังเวลาเมื่อตัวควบคุมติดต่อไปยังภาครับ	50
รูปที่ 7.3 โฟล์วชาร์ตแสดงขบวนการแฮนด์เช็คในการส่งรหัสคำสั่งมาตรฐาน	51
รูปที่ 7.4 โฟล์วชาร์ตแสดงขบวนการแฮนด์เช็คในการส่งชุดข้อมูลคำสั่ง	52
รูปที่ 7.5 โฟล์วชาร์ตแสดงการทำงานของไมโครคอนโทรลเลอร์	54
รูปที่ 8.1รูปหน้าตา INTERFACE ของโปรแกรม SERVER.EXE	55
รูปที่ 8.2 รูปหน้าตา INTERFACE ของโปรแกรม CLIENT.EXE	56
รูปที่ 8.3 รูปแสดงวิธีการเลือกอุปกรณ์เพื่อจะทำการควบคุม	57
รูปที่ 8.4 สวิตช์ที่ใช้เซ็ท GPIB Address ของ Power Supply	57
รูปที่ 8.5 หน้าตาโปรแกรมส่วนของ Power Supply ก่อนการกดปุ่ม Control	58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 8.6 Power Supply หลังจากที่อยู่ในโหมดรีโมต (สังเกตบริเวณที่วงไว้) 58

รูปที่ 8.7 หน้าตาของโปรแกรมส่วน Power Supply หลังจากกด Control แล้ว 59

รูปที่ 8.8 แสดงผลการทำงานของโปรแกรมเมื่อมีการสั่งให้ตรวจสอบ 60
สถานะของแรงดันซึ่งจะแสดงผลที่ตำแหน่งของ Status/Display ด้านบน

รูปที่ 8.9 แสดงผลการทำงานของโปรแกรมเมื่อมีการสั่งให้ตั้งค่าแรงดัน 61

รูปที่ 8.10 ผลที่ได้หลังที่ได้สั่งงานจากโปรแกรมในรูปที่ 8.9 61

รูปที่ 8.11 สวิตช์ที่ใช้ เซ็ท GPIB Address ของ Function Generator 62

รูปที่ 8.12 หน้าตาโปรแกรมส่วนของ Function Generator ก่อนการกดปุ่ม Connect 62

รูปที่ 8.13 รูปของ Function Generator เมื่ออยู่ในโหมดของการรีโหมด 63
(สังเกตบริเวณที่วงไว้)

รูปที่ 8.14 หน้าตาของโปรแกรมส่วน Function Generator หลังจากกด Connect แล้ว 63

รูปที่ 8.15 แสดงการสั่งงานโปรแกรมและผลที่ได้ที่ตัว Function Gennerator 64
หลังจากที่ทำการสั่งงานจากโปรแกรมแล้วโดยสั่งงานให้เป็น
คลื่น Sine ที่ความถี่ 3 kHz ค่า Amplitude 10V

รูปที่ 8.16 แสดงการสั่งงานโปรแกรมและผลที่ได้ที่ตัว Function Gennerator 65
หลังจากที่ทำการสั่งงานจากโปรแกรมแล้วโดยสั่งงานให้เป็นคลื่น
สี่เหลี่ยม ที่ความถี่ 3 kHz ค่า Amplitude 10V

รูปที่ 8.17 แสดงการสั่งงานโปรแกรมและผลที่ได้ที่ตัว Function Gennerator 66
หลังจากที่ทำการสั่งงานจากโปรแกรมแล้วโดยสั่งงานให้เป็นคลื่น
สามเหลี่ยม ที่ความถี่ 1 kHz ค่า Amplitude 10V

รูปที่ 8.18 กดปุ่ม STOR ซึ่งอยู่ ณ ที่วงไว้ในรูปเพื่อให้ไฟสีเหลืองติด 67

รูปที่ 8.19 สวิตช์ใช้สำหรับ เซ็ทค่า GPIB Address ของ Oscilloscope 67

รูปที่ 8.20 หน้าตาของโปรแกรมส่วนของ Oscilloscope ก่อนที่จะกดปุ่ม connect 67

รูปที่ 8.21 เมื่อนำ CH1 ของ Oscilloscope ไปวัดไฟตรง 15V โดยปรับ Vol/div =5 68

รูปที่ 8.22 เมื่อนำ CH1 ของ Oscilloscope ไปวัดไฟตรง 15V โดยปรับ Vol/div =5 68
และ เมื่อนำ CH2 ของ Oscilloscope ไปวัดไฟตรง 10V โดยปรับ
Vol/div =5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่ 8.23 เมื่อนำ CH1 ของ Oscilloscope ไปวัดสัญญาณ sine 20V Peak to Peak มีความถี่=1KHz โดยปรับ Vol/div = 5 และ Time/div = 0.5 ms 69

รูปที่ 8.24 เมื่อนำCH1 ของ Oscilloscope ไปวัดสัญญาณสามเหลี่ยม 20V Peak to Peak มีความถี่=1KHz โดยปรับ Vol/div = 5 และ Time/div = 0.5 ms 69

รูปที่ 8.25 เมื่อนำ CH1 ของ Oscilloscope ไปวัดสัญญาณ sine 10V Peak to Peak มีความถี่=1KHz และมี Offset อยู่ที่ 5V โดยปรับ Vol/div = 5 และ Time/div = 0.5 ms 70

รูปที่ 8.26 เมื่อนำ CH1,CH2 ของ Oscilloscope ไปวัดสัญญาณ sine 10V Peak to Peak มีความถี่=3KHz และมี Offset อยู่ที่ 2.5V โดยปรับ Vol/div = 2 และ Time/div = 50 us โดย ได้กดปุ่ม Inverse CH2 ไว้ด้วย 70

รูปที่ 8.27 เมื่อนำ CH1,CH2 ของ Oscilloscope ไปวัดสัญญาณ สี่เหลี่ยม 8V Peak to Peak มีความถี่=3KHz และมี Offset อยู่ที่ 0V โดยปรับ Vol/div = 2 และ Time/div = 50 us โดย ได้กดปุ่ม Inverse CH2 ไว้ด้วย 71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

การควบคุมด้วย IEEE-488(GPIB)

ระบบอุตสาหกรรมในอดีตมักใช้อุปกรณ์ไม่มากชิ้น การเชื่อมต่อระหว่างอุปกรณ์ก็สามารถทำได้โดยง่าย ต่อมาการอุตสาหกรรมได้เจริญรุดหน้าไปอย่างรวดเร็วดังนั้นระบบอุตสาหกรรมจึงมีขนาดใหญ่และซับซ้อนขึ้นจึงจำเป็นที่จะต้องเชื่อมต่อระหว่างอุปกรณ์หลายๆ ชิ้นจะต้องเสียค่าใช้จ่ายจำนวนมากไปด้วย หากจะมีการเชื่อมต่อระหว่างอุปกรณ์หลายๆ ชิ้นจะต้องเสียค่าใช้จ่ายจำนวนมาก ดังนั้นจึงมีการคิดค้นระบบเชื่อมต่อ ซึ่งเป็นมาตรฐานขึ้นมา นั่นคือ IEEE-488(GPIB) นั่นเอง

โดยในคอมพิวเตอร์ทั่วๆไปนั้นจะไม่มี พอร์ต IEEE-488 ให้ดังนั้น เราจึงได้สร้างอุปกรณ์ขึ้นมา เพื่อทำการแปลง ข้อมูล จากข้อมูลอนุกรมที่ส่งออกมาจากคอมพิวเตอร์เป็นข้อมูลที่เป็นไปตามมาตรฐานของระบบบัส IEEE-488 และเนื่องจากมีการสั่งงานบนคอมพิวเตอร์ ดังนั้น จึงได้มีการพัฒนาโปรแกรมบนวินโดว์ (Window) ขึ้นเพื่อความสะดวกในการใช้งาน สำหรับ โปรแกรมที่เราสร้างขึ้นเพื่อใช้งานบนวินโดว์นั้นเราได้สร้างขึ้นโดยใช้ โปรแกรม วิชวล เบสิก 6.0 (Visual Basic 6.0) ซึ่งเป็นโปรแกรมที่ทำงานในลักษณะของGUI(Graphic User Interface) และ อุปกรณ์เครื่องมือวัดที่นำมาใช้ในโครงงานนี้คือ Programmable Power Supply รุ่น HM-8142 , Programmable Function Generator รุ่น HM-8130 , Programmable Oscilloscope รุ่น HM-1007

ประวัติการพัฒนา IEEE-488

ดังที่กล่าวแล้วข้างต้นว่า การเชื่อมต่อทางอุตสาหกรรมในอดีตเป็นไปด้วยความยากลำบากและเสียค่าใช้จ่ายมากดังนั้นบริษัทผู้ผลิตเครื่องมือวัดต่างๆ ในประเทศสหรัฐอเมริกา จึงร่วมกันจัดหาระบบเชื่อมต่อมาตรฐานขึ้นมาซึ่งในประเทศเยอรมันก็มีการพัฒนาระบบเชื่อมต่อ มาตรฐานเช่นกัน โดยความร่วมมือของ IEC(International Electrotechnical Commission) จนกระทั่งในปี 1972 สหรัฐอเมริกาโดยการนำของ IEEE (Institute of electrical and Electronical Engineers : IEEE) จึงได้มีการประชุมเพื่อวางแผนพิจารณาระบบ เชื่อมต่อมาตรฐานร่วมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บริษัทฮิวเล็กแพกการ์ด ผู้ผลิตเครื่องมือวัดรายใหญ่ในอเมริกา ได้ทำการพัฒนาระบบเชื่อมต่อมาตรฐานอยู่ก่อนแล้วชื่อว่า HPIB (Hewlett Packard Interface Bus) จึงได้นำเสนอโครงการ ให้ IEEE เพื่อพิจารณา และได้รับการยอมรับในปี 1975 โดย IEEE จัดให้เป็นมาตรฐานลำดับที่ 488 ดังนั้น จึงได้ชื่อว่า IEEE-Std 488-1975 ซึ่งต่อมาได้มีการปรับปรุงเป็น IEEE-Std 488-1987 หรือที่นิยมเรียกกันว่า GPIB (General Purpose Interface Bus)

โครงงานโดยรวมของระบบ

แบ่งออกเป็น 4ส่วนด้วยกันดังนี้

1.ส่วน Client (ลูกข่าย)เป็นส่วนที่ทำหน้าที่สั่งการเครื่องมือวัดโดยผ่านทางเครือข่ายไปยัง Server(แม่ข่าย)

2. ส่วน Server(แม่ข่าย)ทำหน้าที่รับข้อมูลคำสั่งจากจากClient(ลูกข่าย)แล้วส่งต่อไปยังบอร์ดไมโครคอนโทรลเลอร์หรือรับข้อมูลที่มาจากเครื่องมือวัดแล้วส่งต่อไปยัง Client (ลูกข่าย)

3.บอร์ดไมโครคอนโทรลเลอร์ ทำหน้าที่ควบคุมการรับ-ส่งข้อมูล

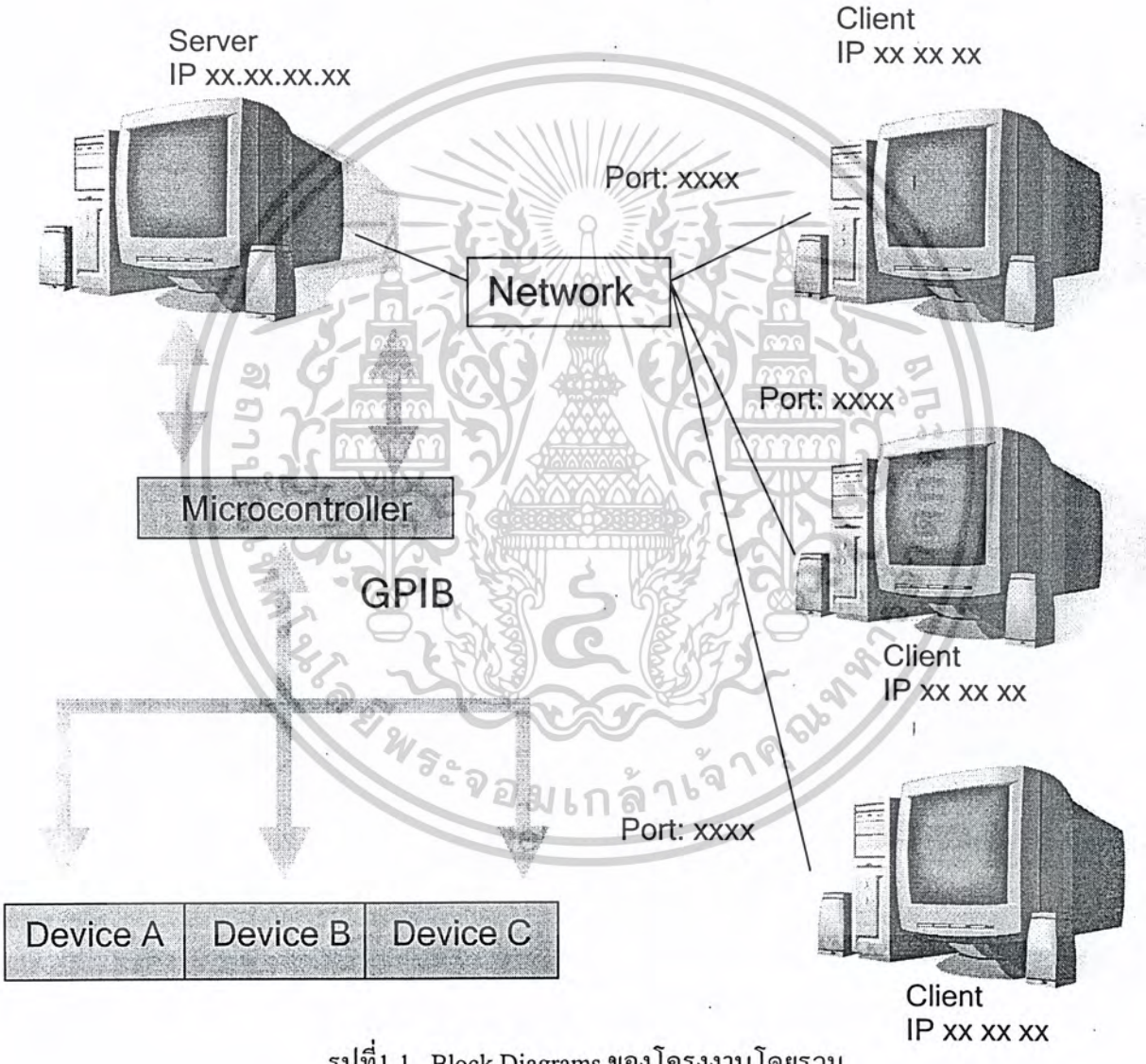
4. เครื่องมือวัดซึ่งได้แก่

1) Programmable Power Supply รุ่น HM-8142 ของบริษัท HAMEG

2) Programmable Function Generator รุ่น HM-8130 ของบริษัท HAMEG

3) Storage Analog/Digital Oscilloscope รุ่น HM 1007 ของบริษัท HAMEG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่1.1 Block Diagrams ของโครงงานโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

IEEE-488(GPIB)

2.1 โครงสร้างของ IEEE-488

ในระบบพื้นฐานของ GPIB จะประกอบด้วยอุปกรณ์ คือผู้ส่ง(Talker), ผู้รับ (Listener) และผู้ควบคุม (controller)

- Talker ทำหน้าที่ส่งข้อมูล โดยในระบบสามารถมี Talker ได้หลายตัวแต่จะมีเพียงตัวเดียวเท่านั้น ที่กำลังทำงานอยู่

- Listener ทำหน้าที่เป็นตัวรับข้อมูล โดยในระบบเดียวกันสามารถมี Listener ได้หลายตัวเช่นเดียวกัน แต่ Listener สามารถทำงานได้ครั้งละหลายๆตัวได้

- Controller ทำหน้าที่ควบคุมอุปกรณ์ต่างๆในระบบ โดยจะกำหนดให้ Talker ทำการส่งข้อมูล หรือ กำหนดให้ Listener ทำการรับข้อมูล

อุปกรณ์ที่มี GPIB นั้นสามารถแบ่งตามหน้าที่ได้ดังนี้

1.ทำหน้าที่เป็น Talker เท่านั้น เช่น เครื่องมือวัด เป็นต้น

2.ทำหน้าที่เป็น Listener เท่านั้น เช่น เครื่องพิมพ์ (Printer), เครื่องบันทึก(Recorder) เป็นต้น

3.ทำหน้าที่เป็น Talker และ Listener เช่น คอมพิวเตอร์,เครื่องมือวัดที่สามารถควบคุมได้จากภายนอก เป็นต้น

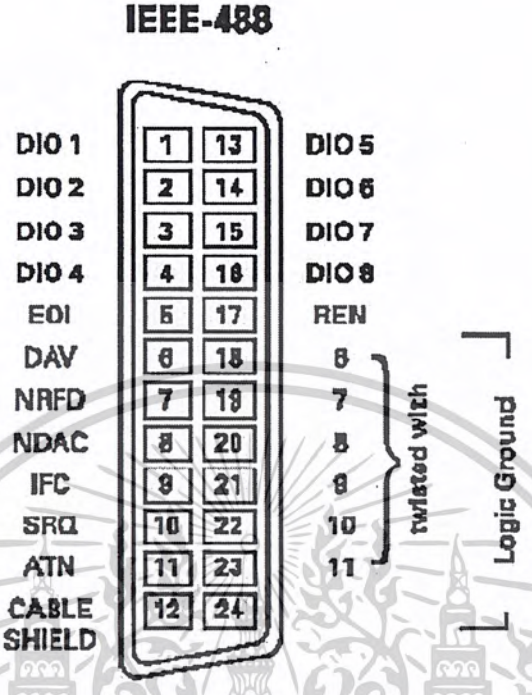
4. ทำหน้าที่เป็น Talker Listener และ Controller ในตัวเดียวกัน เช่น คอมพิวเตอร์ที่ทำหน้าที่ควบคุมระบบ

2.2 ขีดจำกัดของ IEEE – 488

1.จำนวนอุปกรณ์ในระบบ (Talker, Listener, Controller) ที่ต่อกับสายสัญญาณ 1 เส้น จะต้องไม่เกิน 15 เครื่อง

2.สายเคเบิ้ลที่ต่อระหว่างอุปกรณ์ จะต้องยาวไม่เกิน 4 เมตรและความยาวรวมของสายเคเบิ้ลในระบบจะต้องไม่เกิน 20 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 แสดงตำแหน่งของขาสัญญาณตามมาตรฐาน IEEE-488(GPIB)

2.3 รายละเอียดเกี่ยวกับ IEEE-488

รายละเอียดเกี่ยวกับ IEEE-488 นั่นคือเป็นสายสัญญาณแบบ 24 เส้นขนานกันและมีขั้วต่ออยู่ทางปลายทั้งสองของสาย เพื่อต่อกับอุปกรณ์ หรือต่อกันเพื่อให้สายสัญญาณมีความยาวเพิ่มขึ้น ในจำนวนสาย 24 เส้นนี้ มีเพียง 16 เส้นเท่านั้น ที่ทำหน้าที่นำสัญญาณ ส่วนที่เหลืออีก 8 เส้น ทำหน้าที่กราวนด์ (ground) และ (shield)

โดยจำนวนสายที่ใช้นำสัญญาณ 16 เส้นนั้นยังแบ่งได้เป็น 3 ประเภท

1.บัสข้อมูล(Data) จำนวน 8 สาย คือ

-DI01-DI08

2.สายสัญญาณควบคุม(Control Line) จำนวน 5 สาย คือ

-IEF (Interface Clear)

-ATN (Attention)

-SRQ (Service Request)

-REN (Remote Enable)

-EOI (End Or Identify)

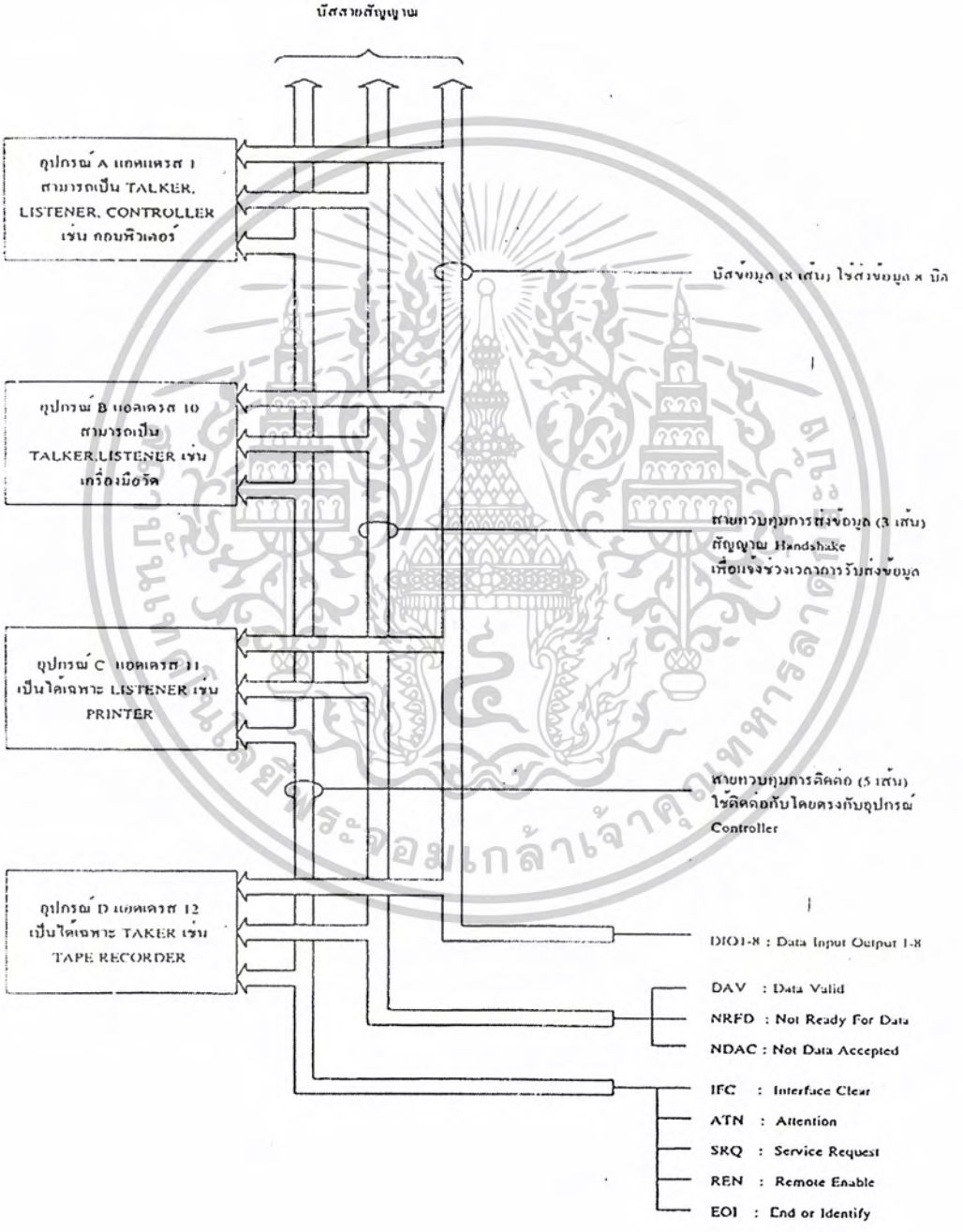
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.สายแฮนด์เชค(HAND SHAKE) 3 สาย คือ

-DAV (Data Valid)

-NAFD (Not Ready for Data)

-NDAC (Not Data Accepted)



รูปที่2.2 แสดงองค์ประกอบของสัญญาณในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ความหมายของสัญญาณต่างๆภายใน IEEE –488

ดังที่ได้กล่าวมาแล้วว่าสายสัญญาณต่างๆภายใน GPIB ได้แบ่งออกเป็น 3 กลุ่มในหัวข้อนี้จะอธิบายความหมายของสัญญาณต่างๆ ดังนี้

กลุ่มสัญญาณข้อมูล

DI01-DI08 สายสัญญาณทั้ง 8 เส้นนั้นทำหน้าที่เป็นทางผ่านของข้อมูลระบบ

กลุ่มสัญญาณควบคุมการเชื่อมต่อ (Interface)

1. IFC (Interface Clear) เป็นสัญญาณรีเซตหรือเคลียร์ระบบกำเนิดได้โดยควบคุมเท่านั้น เมื่ออุปกรณ์ในบัสได้รับสัญญาณเคลียร์นี้ จะกลับคืนสู่สภาวะเริ่มต้นใหม่ ซึ่งเป็นสภาวะแรกเริ่มก่อนการกำหนดฟังก์ชันเหมือนแรกเปิดสวิตซ์

2. ATN (Attention)เป็นสัญญาณที่ถูกส่งโดยอุปกรณ์ที่เป็นตัวควบคุมเช่นเดียวกันใช้ในการสั่งให้ อุปกรณ์ทุกตัวในระบบเตรียมพร้อมเพื่อรอรับคำสั่งต่อไป

3. SRQ (Service Request) เป็นสัญญาณที่ถูกส่งจากอุปกรณ์ เพื่อเป็นการบอกแก่ระบบว่าขณะนี้ อุปกณ์ดังกล่าวต้องการติดต่อจากตัวควบคุม

4. REN (Remote Enable) เป็นสัญญาณที่ถูกส่งมาจากตัวควบคุมเพียงเดียวเท่านั้นเพื่อใช้สั่งอุปกรณ์ต่างๆเปลี่ยนจากโหมดที่ใช้งานปกติมาเป็นการควบคุมโดยตัว ควบคุมแทน

5. EOI (End or Identify) เป็นสัญญาณที่ถูกส่งได้โดยอุปกรณ์ที่เป็นตัวควบคุม(Controller) หรืออุปกรณ์ที่เป็นตัวส่ง(Talker)ก็ได้ใช้สำหรับแสดงว่าข่าวสารที่ส่งเป็นชุดนั้นได้ เสร็จสิ้นลงแล้ว

กลุ่มสัญญาณควบคุมการรับ-ส่งข้อมูล

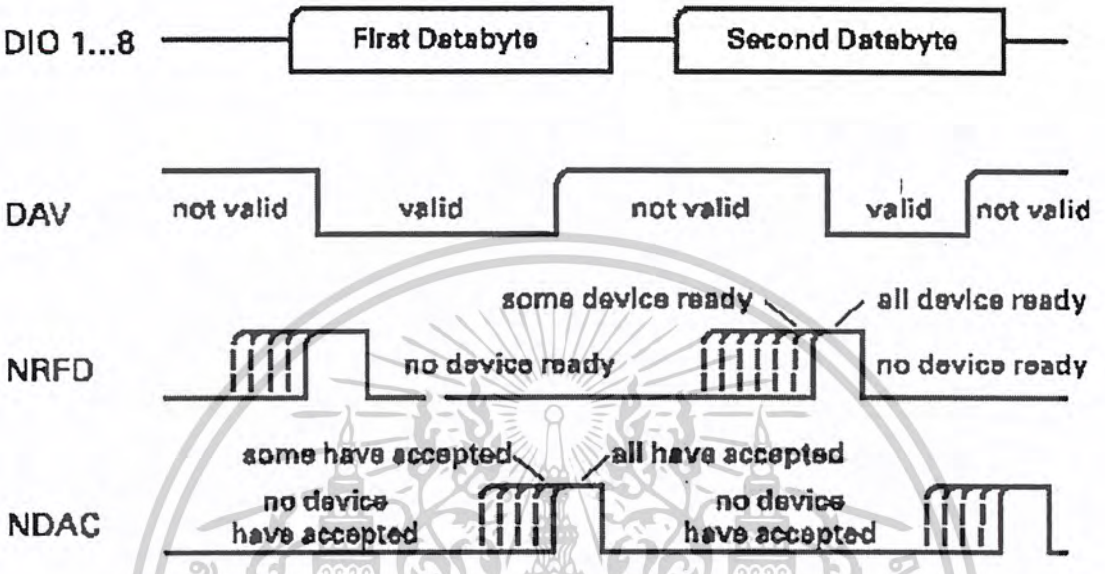
1. DAV (Data Valid) เมื่อสัญญาณนี้ถูกดึงเป็นลอจิก “Low” โดยอุปกรณ์ที่เป็นตัวควบคุม (Talker) เป็นการแจ้งแก่ระบบบัสว่า ขณะนี้ตัวส่ง ได้ทำการส่งข้อมูลลงไปที่สายสัญญาณข้อมูลเรียบร้อยแล้ว

2. NRFD (Not Ready ForData) เมื่อสัญญาณนี้มีลอจิก“Low”จะเป็นการแสดงว่าในขณะนี้ระบบยังไม่พร้อมที่จะรับข้อมูล เนื่องจากอุปกรณ์ในระบบยังพร้อมไม่หมดทุกตัว ซึ่งสัญญาณเส้นนี้จะไม่มี “High” จนกว่าอุปกรณ์ทุกตัวในลอจิกที่เป็น “High” ครบถ้วนแล้ว สัญญาณนี้มีประโยชน์ในกรณีที่อุปกรณ์ในระบบมีความเร็วต่างกัน

3. NDAC (Not Data Accepted) สัญญาณนี้เป็นสัญญาณที่ถูกควบคุมโดยอุปกรณ์ที่เป็นตัวรับ (Listener) โดยสัญญาณนี้จะมีลอจิกเป็น “Low” ในขณะที่อุปกรณ์ที่เป็นตัวรับกำลังเก็บข้อมูลจากสายข้อมูล(Data Bus)และจะเป็น “High” เมื่ออุปกรณ์นั้นได้ทำการอ่านข้อมูลเสร็จเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยสัญญาณลอจิกที่ใช้ใน DATA BUS (D1-D8)ของ IEEE-488 นี้มีลักษณะ เป็นคอมพลีเมนต์ ทั้งหมด คือ "1 เท่ากับ "Low" และ"0 เท่ากับ "High"ซึ่งตรงข้ามกับในวงจรที่เราคุ้นเคย



รูปที่ 2.3 แสดงความสัมพันธ์ของสัญญาณควบคุมการรับส่งข้อมูล

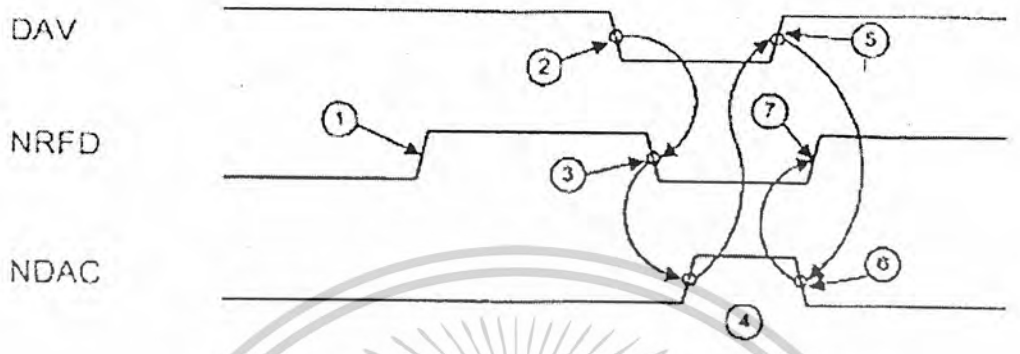
2.5 ขบวนการแฮนด์เช็ค (Handshake Procedure)

ในการสื่อสารระหว่างภายในระบบ GPIB นั้นจะเป็นการสื่อสารแบบอะซิงโครนัส คือเมื่อมีการรับส่งข้อมูลระหว่างตัวส่งและตัวรับ ตัวส่งจะต้องแจ้งให้ตัวรับทราบว่าตัวส่งได้ส่งข้อมูลลงไปบนบัสแล้ว และให้ตัวรับทำการเก็บข้อมูลได้ เมื่อตัวรับทำการเก็บข้อมูลเสร็จแล้ว ก็จะต้องแจ้งแก่ตัวส่งให้ทราบว่า ได้รับข้อมูลที่ส่งมาเรียบร้อยแล้ว เพื่อที่ตัวส่งจะได้ทำการหยุดส่งข้อมูลหรือทำการส่งข้อมูลชุดใหม่ลงไปบนบัส กระบวนการเหล่านี้จะเกิดทุกครั้งที่มีการรับส่งข้อมูลในระบบ ซึ่งกระบวนการเหล่านี้ถูกเรียกว่า ขบวนการแฮนด์เช็ค (Handshake Procedure)

ในการพิจารณาถึง ขบวนการแฮนด์เช็คนั้น จะทำการพิจารณาถึงระบบที่ไม่ซับซ้อนนัก เพื่อที่จะทำความเข้าใจได้โดยง่าย ในการสื่อสารระหว่างตัวส่งและตัวรับนั้น จะมีสายสัญญาณควบคุมการรับส่งข้อมูลอยู่ 3 สัญญาณ คือ NRFD, NDAC, DAV โดยสัญญาณ DAV จะเป็นสัญญาณที่ถูกควบคุมโดยตัวส่ง ส่วนสัญญาณ NRFD, NDAC นั้นเป็นสัญญาณที่จะชี้ให้เห็นว่าตัวรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พร้อมที่จะรับข้อมูลที่ส่งลงมาบนบัสของระบบหรือไม่ สำหรับขั้นตอนของขบวนการแฮนด์เช็ค สามารถแสดงได้ดังในภาพที่ 2.4



รูปที่ 2.4 แสดงแผนผังเวลาของขบวนการแฮนด์เช็ค

ขบวนการแฮนด์เช็คจะเริ่มขึ้นหลังจากที่ตัวควบคุมทำการบอกให้ระบบทราบว่าอุปกรณ์ตัวไหนทำหน้าที่เป็นตัวรับหรือตัวส่ง เมื่อตัวรับทราบแล้วก็จะส่งสัญญาณ NRFD ให้เป็น High (สถานะที่ 1) เพื่อบอกให้ตัวส่งทราบว่าตัวรับพร้อมที่จะรับข้อมูลแล้ว และตัวส่งก็จะทำการส่งข้อมูลลงไปบนบัสข้อมูล DIO1-DIO8 และจะทำการรออยู่ชั่วขณะหนึ่งแล้วตัวส่งสัญญาณ DAVให้เป็น Low เพื่อแจ้งให้ทราบว่าขณะนี้ตัวส่งได้ข้อมูลลงบนบัสข้อมูลแล้ว (สถานะที่ 2) เมื่อตัวรับทราบว่ามีข้อมูลอยู่บนบัสข้อมูลก็จะส่งสัญญาณ NRFD ให้มีค่าเป็น Low เมื่อตัวรับพร้อมที่จะรับข้อมูล (สถานะที่ 3) หลังจากตัวรับได้รับข้อมูลไปเก็บไว้ในบัฟเฟอร์เรียบร้อยแล้วก็จะส่งสัญญาณ NDAC ให้มีค่าเป็น High เพื่อแจ้งให้ทราบว่าตัวรับได้รับข้อมูลเรียบร้อยแล้ว (สถานะที่ 4) เมื่อตัวส่งได้รับสัญญาณ NDAC ที่เป็น High ตัวส่งก็จะทำการส่งสัญญาณ DAV ให้เป็น High เพื่อแจ้งให้ตัวรับไม่ต้องทำการเก็บข้อมูลนั้นอีก (สถานะที่ 5) เมื่อตัวรับได้รับสัญญาณ DAV ที่มีค่าเป็น High ก็จะส่งสัญญาณ NDAC ให้เป็น Low (สถานะที่ 6) ทำให้ข้อมูลในบัสถูกกำจัดออกไป หลังจากนั้นตัวรับก็จะส่งสัญญาณ NRFD ให้เป็น High (สถานะที่ 7) เพื่อบอกให้ทราบว่าตัวรับนั้นพร้อมที่จะรับข้อมูลชุดต่อไปที่จะถูกส่งเข้ามาในบัส เป็นอันเสร็จสิ้นขบวนการแฮนด์เช็ค

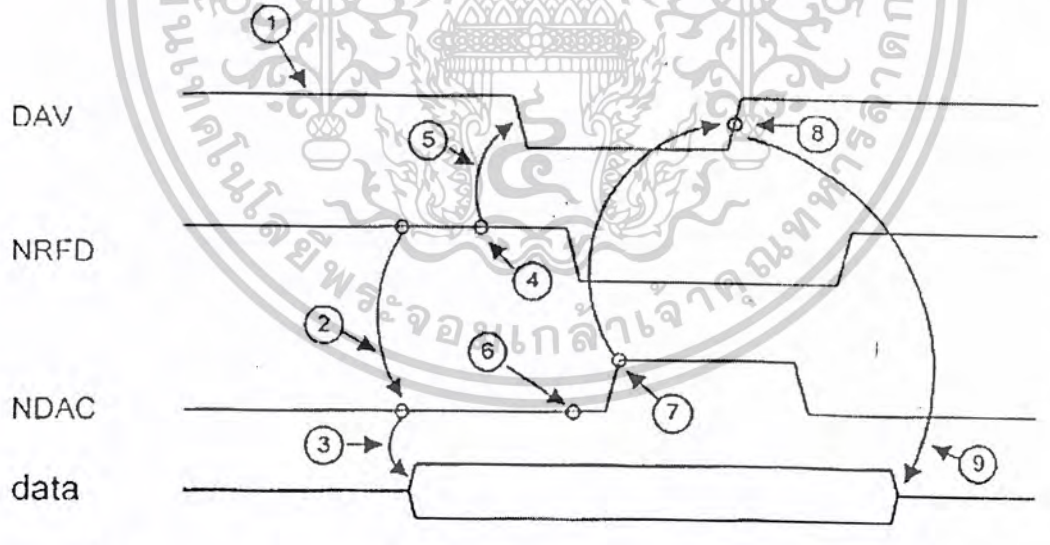
- ขบวนการแฮนด์เช็คเมื่อใช้ไมโครคอมพิวเตอร์เป็นตัวควบคุมและตัวส่ง

ขบวนการแฮนด์เช็คเมื่อใช้ไมโครคอมพิวเตอร์เป็นตัวควบคุมและตัวส่งสามารถแสดงด้วยแผงผังเวลา ดังภาพที่ 2.5 ซึ่งจะทำให้ง่ายต่อการเข้าใจ ในขั้นตอนการส่งข้อมูล

โดยขบวนการดังกล่าว จะเกิดขึ้นหลังจากการกำหนดอุปกรณ์ในระบบแล้ว ขบวนการแฮนด์เช็คจะเริ่มขึ้น เมื่อตัวควบคุมส่งสัญญาณ DAV ให้เป็น High (สถานะที่ 1) ซึ่งตัวควบคุมได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซ็ทให้สัญญาณ DAV ให้มีค่าเป็น High อยู่ก่อนแล้ว หลังจากนั้นตัวควบคุมจะทำการตรวจสอบสัญญาณ NRFD และ NDAC ว่ามีค่าเป็น High ทั้งคู่หรือไม่ (สถานะที่ 2) ถ้าสัญญาณทั้งสองเป็น High ทั้งคู่ แสดงว่าอุปกรณ์ไม่พร้อมที่จะทำงาน ขบวนการแฮนด์เช็คก็จะถูกยกเลิกไป แต่ถ้าที่สถานะที่ 2 หากสัญญาณใดสัญญาณหนึ่งเป็น Low ตัวควบคุมจะทำการส่งข้อมูลลงในบัสข้อมูล (สถานะที่ 3) หลังจากนั้นตัวควบคุมจะทำการตรวจสอบสัญญาณ NRFD ว่าเป็น High หรือไม่ (สถานะที่ 4) ถ้าสัญญาณ NRFD เป็น High ตัวควบคุมก็จะส่งสัญญาณ DAV ให้มีค่าเป็น Low เพื่อบอกให้ตัวรับทราบว่ามีข้อมูลอยู่ในบัสข้อมูล (สถานะที่ 5) แต่ถ้าที่สถานะที่ 4 สัญญาณ NRFD มีค่าเป็น Low แสดงว่าขบวนการแฮนด์เช็คเกิดความผิดพลาดขึ้น จะต้องทำการเริ่มต้นใหม่ จากสถานะที่ 5 ตัวควบคุมจะรอเวลาให้ตัวรับทำการเก็บข้อมูล เมื่อถึงเวลาที่กำหนดตัวควบคุมจะทำการตรวจสอบสัญญาณว่าสัญญาณ NDAC ถูกเปลี่ยนให้เป็น High ในเวลาที่กำหนดหรือไม่(สถานะที่6) ถ้าตัวรับไม่ได้รับข้อมูลในเวลาที่กำหนด ขบวนการแฮนด์เช็คจะถูกยกเลิก แต่ถ้าตัวรับได้รับข้อมูลภายในเวลาที่กำหนดตัวรับจะทำการเปลี่ยนสัญญาณ NDAC ให้เป็น High (สถานะที่ 7) ตัวควบคุมก็จะทำการตอบสนองโดยการเปลี่ยนสัญญาณ DAV ให้เป็น High (สถานะที่ 8) และตัวควบคุมก็จะทำการลบข้อมูลที่อยู่ในบัสข้อมูลออกไป(สถานะที่ 9) เป็นการเสร็จสิ้นขบวนการแฮนด์เช็คดังกล่าว



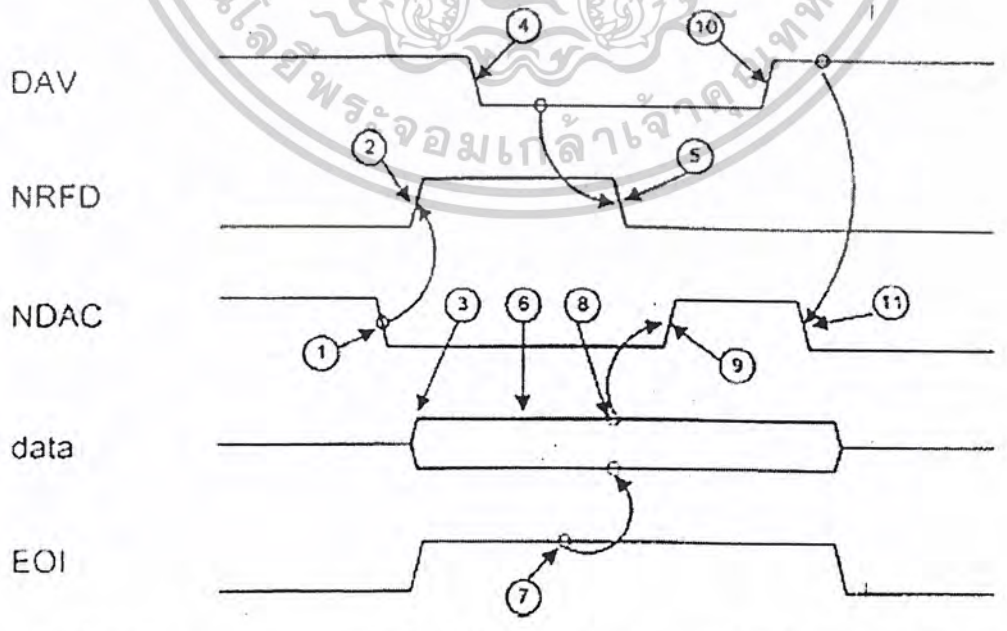
รูปที่ 2.5 แสดงขบวนการแฮนด์เช็คเมื่อใช้ไมโครคอมพิวเตอร์เป็นตัวควบคุมและตัวส่ง

- ขบวนการแฮนด์เช็คเมื่อใช้ไมโครคอมพิวเตอร์เป็นตัวควบคุมและตัวรับ

ขบวนการแฮนด์เช็คเริ่มขึ้นโดยตัวควบคุมรับรู้ว่าตัวส่งจะทำการส่งข้อมูล ตัวควบคุมจะส่งสัญญาณ NDAV ให้มีค่าเป็น Low เพื่อบอกให้ทราบว่าตัวควบคุมยังไม่ได้รับข้อมูล (สถานะที่ 1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อ จากนั้นตัวควบคุมจะส่งสัญญาณ NRFD ให้เป็น High เพื่อบอกให้ตัวส่งทราบว่าตัวควบคุมพร้อมที่จะรับข้อมูลแล้ว (สถานะที่ 2) ตัวส่งจะทราบได้ทันทีว่าขณะนี้สามารถที่จะส่งข้อมูลลงบนบัสข้อมูลได้แล้ว (สถานะที่ 3) จากการที่สัญญาณ NRFD มีลอจิกเป็น High และสัญญาณ NDAC มีลอจิกเป็น Low ขั้นต่อไปตัวควบคุมจะทำการรอเวลาให้ตัวส่งทำการส่งข้อมูลลงบนบัสข้อมูลให้เสร็จ และจะทำการตรวจสอบด้วยว่า เกินเวลาที่กำหนดหรือไม่ หากเกินเวลาที่กำหนดก็จะทำการออกจากขบวน การแฮนด์เช็ค หากยังไม่เกินก็จะรอจนหมดเวลา หรือ จนกว่าสัญญาณ DAV จะเป็น Low (สถานะที่ 4) เมื่อสัญญาณ DAV มีลอจิกเป็น Low ตัวควบคุมก็จะตอบรับโดยการทำให้สัญญาณ NRFD มีค่า เป็น Low (สถานะที่ 5) เพื่อบอกให้ตัวส่งทราบว่าตัวควบคุมพร้อมที่จะเริ่มทำการเก็บข้อมูลแล้ว (สถานะที่ 6) หลังจากนั้นตัวควบคุมจะทำการตรวจสอบสัญญาณ EOI ว่าข้อมูลที่ตัวส่ง ทำการส่งมานั้นหมดหรือยังโดยตัวส่งจะทำการเปลี่ยน สัญญาณ EOI ให้มีลอจิกเป็น Low เมื่อข้อมูลไบต์สุดท้าย ถูกส่งลงไปบนบัสข้อมูล(สถานะที่7) หากสัญญาณ EOI ยังเป็น High ตัวควบคุมจะทำการเก็บข้อมูล ในบัสข้อมูลต่อไป(สถานะที่ 8)และเมื่อตัวควบคุมได้ทำการเก็บข้อมูลเสร็จเรียบร้อยแล้ว ตัวควบคุมจะเปลี่ยนลอจิกของสัญญาณ NDAC ให้เป็น High (สถานะที่9) เมื่อตัวควบคุมทำสัญญาณ NDAC มีลอจิกเป็น High แล้ว ตัวส่งจะทำการลบข้อมูลบนบัสออกโดยการเปลี่ยนสัญญาณ DAV ให้มีค่าเป็น High (สถานะที่ 10) ตัวควบคุมก็จะทำการนำข้อมูลที่ได้ไปใช้งานและเปลี่ยนสัญญาณ NDAC ให้มีค่าเป็น Low (สถานะที่ 11)เป็นการสิ้นสุดขบวนการแฮนด์เช็ค ขบวนการแฮนด์เช็คแบบนี้สามารถ เขียนแทนด้วยแผนผังเวลา ดังภาพที่ 2.6



รูปที่ 2.6 แสดงขบวนการแฮนด์เช็คเมื่อใช้ไมโครคอมพิวเตอร์เป็นตัวควบคุมและตัวรับ

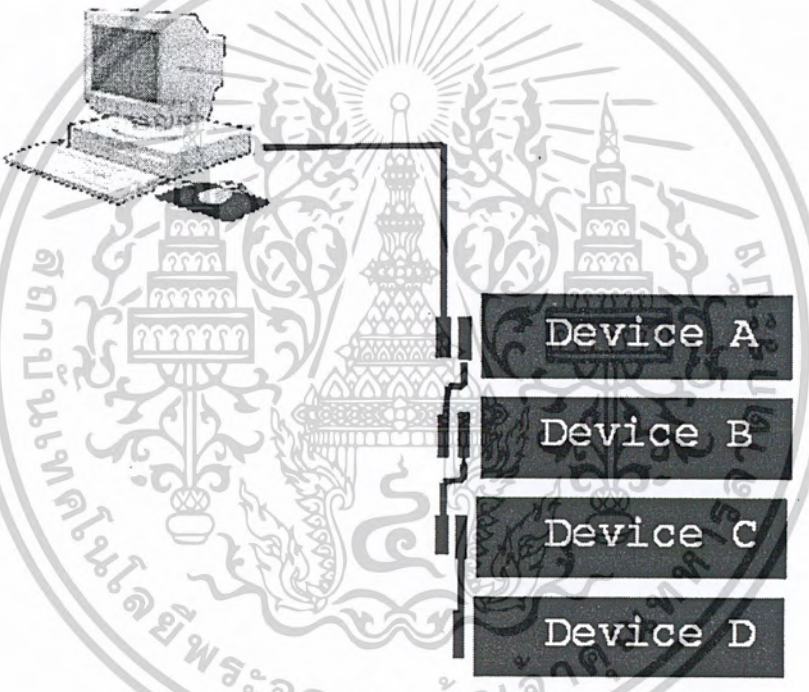
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 การเชื่อมต่ออุปกรณ์ต่างๆในระบบบัสมาตรฐาน IEEE-488 (GPIB)

สำหรับการเชื่อมต่อระหว่างอุปกรณ์ต่างๆในระบบ IEEE-488 นั้นมีอยู่ 2วิธีคือ

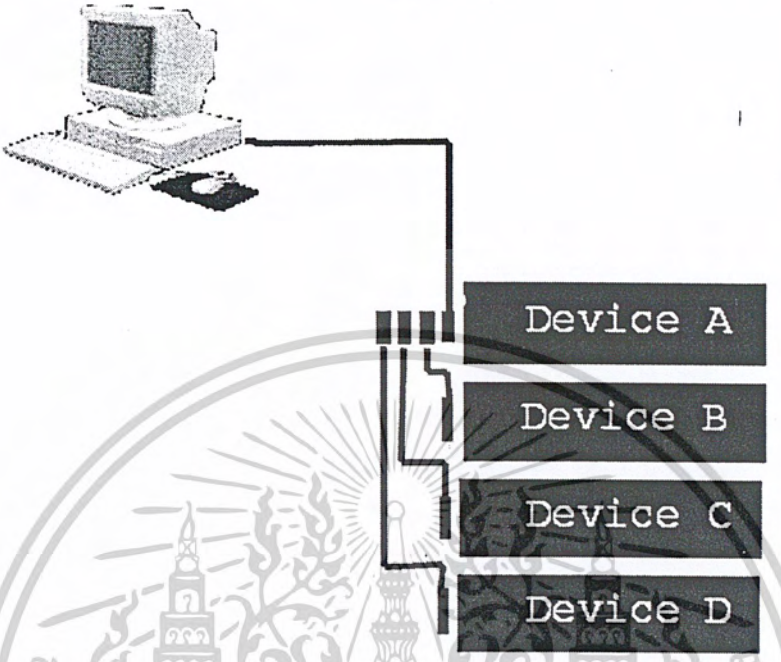
1. การเชื่อมต่อแบบเรียงต่อเนื่องกัน (Daisy Chain Configuration)
2. การเชื่อมต่อแบบกระจาย (Star Configuration)

ซึ่งได้แสดงลักษณะการเชื่อมต่อทั้ง 2 แบบไว้เพื่อดูความแตกต่างให้ชัดเจนขึ้นดังนี้



รูปที่ 2.7 รูปแสดงการต่อแบบเรียงต่อเนื่องกัน (Daisy Chain Configuration)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 รูปแสดงการต่อแบบกระจาย (Star Configuration)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

คุณสมบัติโดยทั่วไปของระบบบัส USB

3.1 คุณสมบัติเด่นของระบบบัส USB

ระบบบัส USB (Universal Serial Bus) นั้น เป็นระบบบัสที่มีความยืดหยุ่นสูง ปราศจากข้อจำกัดและการขัดขวางของการอินเตอร์เฟสทางด้านฮาร์ดแวร์ ซึ่งการตั้งค่าการทำงานต่างๆ จะถูกกระทำโดยระบบปฏิบัติการโดยอัตโนมัติ นอกจากนี้การเชื่อมต่ออุปกรณ์ USB เข้ากับระบบนั้น สามารถทำได้ทั้งในขณะที่เครื่องคอมพิวเตอร์ยังคงทำงานอยู่ ซึ่งกล่าวได้ว่าเป็นรูปแบบของปลั๊กแอนด์เพลย์ (Plug and Play) อย่างแท้จริง อีกทั้งความเร็วในการส่งถ่ายข้อมูลใน USB 2.0 นั้นก็มีความเร็วสูงกว่าการส่งถ่ายข้อมูลแบบขนานและอนุกรมแต่เดิมเป็นอย่างมาก และในการเพิ่มจำนวนของพอร์ตนั้นก็สามารถทำได้โดยง่ายเพียงนำ USB Hub มาต่อพ่วงเข้ากับระบบเท่านั้น

<u>คุณสมบัติเด่นในระบบบัส USB</u> ได้แก่

- สามารถนำอุปกรณ์ I/O มาต่อพ่วงเข้าสู่ระบบได้ในขณะที่เครื่องคอมพิวเตอร์ยังคงทำงานอยู่ได้ (Hot-Pluggable)
- ง่ายต่อการใช้งาน เนื่องจากเครื่องคอมพิวเตอร์จะมีความสามารถในการทำความรู้จักและจดจำอุปกรณ์ต่างๆที่นำมาต่อพ่วงในระบบโดยไดร์ฟเวอร์ที่เหมาะสม และในการตั้งค่าต่างๆ ก็เป็นไปโดยอัตโนมัติ
- ใช้คอนเน็คเตอร์ในการเชื่อมต่อเพียงชนิดเดียวจึงลดความสับสนในการเชื่อมต่อ
- มีประสิทธิภาพในการส่งถ่ายข้อมูลสูง โดยเป็นไปตามมาตรฐานดังนี้

  มาตรฐาน USB 1.0/1.1 มีระดับความเร็วในการส่งถ่ายข้อมูล 2 ระดับได้แก่ ที่ระดับความเร็วต่ำ (Low Speed) เท่ากับ 1.5 Mbit/Sec และที่ระดับความเร็วเต็มที่ (Full Speed) เท่ากับ 12 Mbit/Sec

  มาตรฐาน USB 2.0 จะมีการเพิ่มเติมระดับความเร็วในการส่งถ่ายขึ้นอีก 1 ระดับ ได้แก่ ที่ระดับความเร็วสูง (High Speed) มีความเร็วในการส่งถ่ายข้อมูลเท่ากับ 480 Mbit/sec
- ไม่เกิดการขัดแย้งกันของการเข้าใช้ทรัพยากรของระบบ (IRQ : Interrupt Request) ซึ่งเป็นการแก้ปัญ หาทางด้านข้อจำกัดของจำนวนอุปกรณ์ที่มาเชื่อมต่อ
- สามารถต่ออุปกรณ์ภายในระบบได้สูงสุด 127 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สายเคเบิลของ USB นั้น จะมีสายของแหล่งจ่ายกำลังงานรวมอยู่ภายใน และสามารถนำมาใช้งานได้

- มีการจัดการกับระบบพลังงานที่ชาญฉลาด ซึ่งกำลังงานบนบัสจะถูกลดระดับลงเมื่อไม่ได้มีการใช้งานเป็นระยะเวลาหนึ่ง

- มีความสามารถในการตรวจสอบและแก้ไขข้อผิดพลาดของข้อมูลโดยอัตโนมัติ

3.2 การส่งถ่ายข้อมูลบนระบบบัส USB

USB แป็นการสื่อสารข้อมูลในรูปแบบอนุกรมรูปแบบหนึ่ง ซึ่งอุปกรณ์ทุกๆ ตัวจะต้องส่งสัญญาณ รวมกันไปในสายส่งสัญญาณเพียงคู่เดียว ดังนั้นอุปกรณ์ทุกๆ ตัวที่เชื่อมต่อกับบัสจะต้องส่งข้อมูล เรียงลำดับกันไปเพื่อไม่ให้เกิดการชนกันของข้อมูล และเนื่องจาก USB เป็นระบบบัสที่ใช้สายส่งสัญญาณเพียงคู่เดียว ทำให้ในช่วงเวลาหนึ่งๆ จะมีข้อมูลวิ่งไปได้เพียงทิศทางเดียวเท่านั้น ไม่สามารถเกิดการรับและส่งข้อมูลไปในเวลาเดียวกันได้หรือที่เรียกกันว่าการส่งข้อมูลแบบฮาล์ฟดูเพล็กซ์ (half duplex) โดยจังหวะการรับส่งข้อมูลของระบบบัส USB ทั้งหมดจะถูกควบคุมจากโฮสต์ (host) ซึ่งก็คือเครื่องคอมพิวเตอร์ที่เป็นจุดรวมของอุปกรณ์ทุกตัวที่เชื่อมต่ออยู่นั่นเอง ดังนั้นจึงไม่สามารถเชื่อมต่ออยู่นั่นเอง ดังนั้นจึงไม่สามารถเชื่อมต่อเครื่องคอมพิวเตอร์ 2 เครื่อง ให้รับหรือส่งข้อมูลถึงกันได้โดยตรง เพราะถ้าคอมพิวเตอร์ทั้งสองเครื่องทำหน้าที่เป็นโฮสต์ทั้งคู่จะเกิดการชนกันของข้อมูลภายในบัส เนื่องจากแต่ละเครื่องก็จะพยายามกำหนดจังหวะในการรับส่งของตัวเองขึ้นมา ดังนั้นจะเชื่อมต่อคอมพิวเตอร์ 2 เครื่องเข้าด้วยกันผ่าน USB จะต้องมีอุปกรณ์ที่เป็นตัวกลางเพื่อซิงโครไนซ์ตัวเองเข้ากันกับโฮสต์ทั้งสองให้ได้

การรับส่งข้อมูลจะถูกกำหนดเป็นเฟรม โดยทุกๆ 1 มิลลิวินาที(ms) จะเกิดการรับส่งข้อมูลขึ้น 1 เฟรม ในแต่ละเฟรมจะแบ่งย่อยออกเป็นแพ็กเก็ต (packet) เริ่มต้นการทำงานของแต่ละเฟรมโดยโฮสต์จะต้องส่งสัญญาณเริ่มต้นเฟรมหรือ SOF (Start Of Frame) ออกไปเพื่อให้อุปกรณ์ทุกตัวรู้จังหวะการเริ่มเฟรม หลังจากนั้นโฮสต์ก็จะเริ่มส่งหรือรับข้อมูลต่างๆ ตามที่ได้จัดลำดับความสำเร็จไว้อุปกรณ์ต่างๆ ที่อยู่ภายในบัสจะต้องทำงานตามจังหวะที่โฮสต์กำหนดไว้เท่านั้น การส่งข้อมูลกลับไปยังโฮสต์จะสามารถทำได้ก็ต่อเมื่อได้รับการถามหรือร้องขอจากโฮสต์แต่เนื่องจากแต่ละเฟรมข้อมูลจะต้องรับส่งเสร็จภายใน 1 มิลลิวินาที นั่นหมายความว่าข้อมูลของอุปกรณ์ทุกๆ ตัวที่เชื่อมตัวต่อกับบัสจะต้องถูกกำหนดขนาดไม่ให้ใหญ่เกินกว่าที่จะสามารถรับส่งได้ภายใน 1 มิลลิวินาทีและเล็กพอที่จะทำให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ทุกๆ ตัวสามารถใช้งานบัสไปพร้อมๆ กันได้ดังนั้นในระบบบัส USB จึงจำเป็นต้องอาศัยซอฟต์แวร์ที่เข้ามาจัดการในด้านนี้และยังต้องอาศัยฮาร์ดแวร์ที่จะคอยกระจายการส่งและรวบรวมการรับข้อมูลจากอุปกรณ์ทุกๆ ตัวในระบบ โดยแบ่งเป็นองค์ประกอบทางด้านซอฟท์แวร์และฮาร์ดแวร์ดังต่อไปนี้

3.3 องค์ประกอบทางด้านซอฟท์แวร์

3.3.1 ไดรเวอร์อุปกรณ์ USB (USB Device Drivers)

ไดรเวอร์อุปกรณ์ USB คือโปรแกรมเก็บข้อมูลที่จำเป็นในการติดต่อไปยังอุปกรณ์แต่ละตัวเมื่อโปรแกรมใดมีความต้องการจะติดต่อกับอุปกรณ์ต่างๆ จะต้องแจ้งความต้องการนั้นๆ มายังไดรเวอร์อุปกรณ์USB เนื่องจากตัวไดรเวอร์นี้จะรู้ว่าถ้าต้องการติดต่อกับอุปกรณ์จะต้องติดต่อผ่านเอ็นด์พอยต์(Endpoint)ไหนด้วยรูปแบบใด ดังนั้นอุปกรณ์แต่ละตัวก็จะมีไดรเวอร์อุปกรณ์ USB เฉพาะตัว ซึ่งเมื่อถึงคราวต้องต้องนำอุปกรณ์นั้นมาต่อใช้งานกับเครื่องคอมพิวเตอร์จริงๆ ก็จะต้องนำไดรเวอร์ตัวเดียวกันมาติดตั้งเพิ่มเข้ากับระบบปฏิบัติการในคอมพิวเตอร์เพื่อให้ระบบรู้จักและติดต่อใช้งานอุปกรณ์ที่ติดตั้งเข้า มาใหม่นี้ได้ เช่น ถ้าต้องการติดต่อเพื่อรับข้อมูลจากคีย์บอร์ด ตัวไดรเวอร์อุปกรณ์ USB จะรู้ว่าต้องรับ ส่งข้อมูลด้วยอัตราเร็วต่ำ (slow speed) โดยใช้รูปแบบการถ่ายทอดข้อมูลแบบอินเตอร์รัปต์(interrupt transfer type) ผ่านเอ็นด์พอยต์ตัวหนึ่งของคีย์บอร์ด และตรวจสอบข้อมูลการกดเป็นช่วงระยะห่างค่า หนึ่งแต่ในบางอุปกรณ์ที่เป็นอุปกรณ์พื้นฐานของเครื่องคอมพิวเตอร์ เช่น เมาส์และคีย์บอร์ดจะมีการบรรจุไดรเวอร์อุปกรณ์USB ของอุปกรณ์เหล่านี้ไว้ภายในไบออสของเครื่องคอมพิวเตอร์เรียบร้อยแล้ว จึงไม่ต้องติดตั้งไดรเวอร์เพิ่มเติมสำหรับอุปกรณ์เหล่านี้ เพียงแต่เข้าไปเปิดการทำงาน ไบออสก็จะทำให้เครื่องคอมพิวเตอร์รู้จักอุปกรณ์เหล่านี้โดยอัตโนมัติ

3.3.2 ไดร์เวอร์ USB (USB Drivers)

ไดร์เวอร์ USB นั้นเป็นส่วนของซอพต์แวร์ที่ทำหน้าที่ในการจัดการแบ่งสันปันส่วนช่องสัญ ญาณในแต่ละเฟรมให้อุปกรณ์แต่ละตัวนั้นสามารถส่งข้อมูลร่วมกันไปในสายสัญญาณคู่เดียวได้ในเวลาเดียวกัน ไดรเวอร์อุปกรณ์ USB ของอุปกรณ์แต่ละตัวจะส่งการร้องรอเพื่อการติดต่อ(request) ลงมายังไดรเวอร์USB และเมื่อไดรเวอร์USB รับทราบความต้องการติดต่อของอุปกรณ์ครบทุกๆ ตัวที่เชื่อมต่ออยู่กับบัสแล้ว ก็จะพิจารณาว่า ในรอบการรับส่งข้อมูลหนึ่งๆ นั้นอุปกรณ์แต่ละตัวสามารถรับส่งข้อมูลได้มากเท่าใด หากปริมาณข้อมูลที่ต้องการรับ-ส่งมีขนาดมากก็จะตัดแบ่งออกเป็นส่วนๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

แล้วเก็บไว้เพื่อรอ ส่งในรอบถัดไป โดยปริมาณข้อมูลที่ส่งได้ของอุปกรณ์แต่ละตัวจะถูกพิจารณาจากชนิดของการถ่ายทอดข้อมูล (transfer type) ว่า อุปกรณ์ใดใช้การถ่ายทอดข้อมูลแบบใดและการรับส่งข้อมูลชนิดนั้นมีลำดับความสำคัญมากน้อยเพียงใด

3.3.3 ไดร์เวอร์โฮสต์คอนโทรลเลอร์ (USB Host Controller Driver)

หลังจากไดรเวอร์ USB พิจารณาแล้วว่าอุปกรณ์แต่ละตัวส่งข้อมูลได้เท่าใดบ้าง มันจะส่งข้อมูลของอุปกรณ์แต่ละตัวที่จะติดต่อในรอบการติดต่อนั้นๆ มายังไดรเวอร์โฮสต์คอนโทรลเลอร์จากนั้นไดรเวอร์โฮสต์คอนโทรลเลอร์จะจัดเรียงลำดับข้อมูลของอุปกรณ์แต่ละชนิดลงเป็นเฟรมข้อมูลเพิ่มเติมส่วนประกอบต่างๆ ของเฟรมข้อมูลให้ครบตามมาตรฐานการถ่ายทอดข้อมูลแบบUSBแล้วส่งข้อมูลทั้งหมดไปยังฮาร์ดแวร์USBโฮสต์คอนโทรลเลอร์เพื่อส่งข้อมูลทั้งหมดออกไปยังอุปกรณ์ต่างๆ ในรูปที่ 3.2 แสดงลำดับและขั้นตอนการทำงานของซอฟต์แวร์ควบคุมการทำงานของพอร์ตUSB

3.4 องค์ประกอบทางด้านฮาร์ดแวร์

3.4.1 USBโฮสต์คอนโทรลเลอร์ & USB รูตฮับ (USB Host Controller Driver & USB Root Hub)

ในส่วนของUSBโฮสต์คอนโทรลเลอร์นั้นมีหน้าที่ในการสร้างสัญญาณ ไฟฟ้าเพื่อใช้ในการติดต่อสื่อสาร โดยสัญญาณที่ได้จากตัวมันจะถูกส่งไปยัง USB รูตฮับ โดยมันจะทำหน้าที่ในการแปลข้อมูล รูปแบบขนานที่รับมาจาก USB โฮสต์คอนโทรลเลอร์ให้เป็นข้อมูลรูปแบบอนุกรมที่ใช้ในการส่ง

ต่อไป นอกเหนือจากนี้นั้น USB รูตฮับยังมีหน้าที่สำคัญอีก 4 ประการได้แก่

- ควบคุมการใช้พลังงานของอุปกรณ์ที่มาต่อร่วม
- ตรวจสอบการเชื่อมต่อของอุปกรณ์ว่ามีอุปกรณ์ต่อร่วมอยู่หรือไม่
- เปิดหรือเอ็นเอเบิลการใช้งานพอร์ตเมื่อมีอุปกรณ์ต่ออยู่ และปิดหรือดิสเอเบิลการใช้งานเมื่อปลดอุปกรณ์ออกไปแล้ว
- รายงานสถานะของแต่ละพอร์ตเมื่อไดรเวอร์โฮสต์คอนโทรลเลอร์ร้องขอมา

3.4.2 USB ฮับ (USB Hub)

หน้าที่หลักๆของ USBฮับคือ ขยายการเชื่อมต่อให้อุปกรณ์จำนวนมากๆ สามารถเชื่อมต่อเข้ากับระบบบัสได้ โดยการทำงานหลักของ USB ฮับนั้นมีอยู่ 2 ส่วนคือ ทำหน้าที่เป็นตัวทวนสัญญาณ (repeater) และตัวจัดการพลังงาน (power management) ในส่วนของการทวนสัญญาณUSB ฮับจะต้อง

61506
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รับสัญญาณจากโฮสต์มา แล้วส่งกระจายออกไปยังพอร์ตทุกๆ พอร์ต และรับสัญญาณจากแต่ละพอร์ต แล้วจับมารวมกันเพื่อส่งกลับไปที่โฮสต์สำหรับส่วนของการจัดการพลังงานนั้นมีหน้าที่เหมือนกับรูทฮับก็คือตรวจสอบว่ามีการต่ออยู่ของอุปกรณ์ที่พอร์ตใดบ้าง หากมีอุปกรณ์ต่ออยู่ก็เปิดการใช้งานพอร์ต นั้นๆ หากไม่มีอุปกรณ์ต่ออยู่ก็ปิดการใช้งาน ตรวจสอบการเชื่อมต่อหรือปลดออกของอุปกรณ์เพื่อราย งานผลเมื่อโฮสต์คอนโทรลเลอร์ร้องขอและป้องกันอุปกรณ์ที่ต่ออยู่ในแต่ละพอร์ตไม่ให้ดึงกระแส ไฟฟ้าเกินกว่าที่กำหนด

3.4.3 อุปกรณ์ USB (USB Device)

ส่วนประกอบนี้นั้นก็คือส่วนของอุปกรณ์ต่างๆที่นำมาต่อพ่วงกับพอร์ต USBนั่นเอง โดยการจัดประเภทของอุปกรณ์ USB นั้น สามารถจัดประเภทได้จากความเร็วในการส่งถ่ายข้อมูล และรูปแบบการใช้พลังงานของอุปกรณ์ โดยการจัดประเภทจากความเร็วในการส่งถ่ายข้อมูลนั้นแต่เดิมสามารถจัดได้เป็น 2 ประเภทได้แก่

- อุปกรณ์ความเร็วต่ำ (Low-Speed Devices) เช่น เมาส์, จอยสติ๊ก, คีย์บอร์ด เป็นต้น ซึ่งจะใช้ความเร็วในการส่งถ่ายเท่ากับ 1.5 Mbit/sec

- อุปกรณ์ความเร็วเต็มที่ (Full-Speed Devices) เช่น เครื่องพิมพ์, กล้องดิจิตอล, ซีดีรอมไดร์ฟ, เครื่องเล่น MP3 เป็นต้น ซึ่งจะใช้ความเร็วในการส่งถ่ายข้อมูลเท่ากับ 12 Mbit/sec

แต่ในปัจจุบันนั้น ได้มีการเพิ่มเติมส่วนของมาตรฐาน USB2.0 ซึ่งมีความเร็วการส่งถ่ายข้อมูลในระดับความเร็วสูง (High Speed) เท่ากับ 480 Mbit/sec เพิ่มเติมขึ้นมา ซึ่งจะกล่าวถึงต่อไปในภายหลัง สำหรับการจัดประเภทของอุปกรณ์จากการใช้พลังงานของตัวมันนั้น จะแบ่งออกได้เป็น 2 ประเภทได้แก่

- อุปกรณ์ที่ใช้พลังงานจากระบบบัส (Bus Powered Device) ได้แก่อุปกรณ์ที่ใช้พลังงานจากระบบบัสโดยตรง เช่น Flash Drive เป็นต้น

- อุปกรณ์ที่ใช้พลังงานจากตัวเอง (Self Powered Device) ได้แก่อุปกรณ์ที่ใช้แหล่งจ่ายพลังงานจากตัวเองโดยไม่พึ่งพลังงานจากระบบบัส

นอกจากนี้ยังมีอุปกรณ์ USB บางประเภทที่มีคุณสมบัติของ USB ฮับอยู่ด้วย นั่นคือสามารถนำอุปกรณ์อื่นๆ มาเชื่อมต่อเข้ากับตัวมันได้ เราเรียกอุปกรณ์ประเภทนี้ว่า Compound USB Device

3.5 รูปแบบการส่งถ่ายข้อมูลบนระบบบัส USB

การถ่ายทอดสัญญาณ(tranfer type) ของบัส USB นั้นแบ่งออกเป็น 4 ชนิดตามขนาดชนิดของข้อมูลและจังหวะการส่งข้อมูลดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. การถ่ายทอดสัญญาณแบบไอโซโครนัส(Isochronous tranfer)

2. การถ่ายทอดสัญญาณแบบบัลก์(Bulk transfer)

3. การถ่ายทอดสัญญาณแบบอินเตอร์รัปต์(Interrupt transfer)

4. การถ่ายทอดสัญญาณควบคุม(Control tranfer)

การถ่ายทอดสัญญาณ 3 ชนิดแรกใช้สำหรับข้อมูลทั่วไปที่ต้องการรับหรือส่งไปยังตัวอุปกรณ์ ส่วนการถ่ายทอดสัญญาณแบบที่ 4 การถ่ายทอดสัญญาณแบบไอโซโครนัสใช้ถ่ายทอดข้อมูลที่ต้องการความต่อเนื่องสูง เช่น ข้อมูลสเสียงเพลง ส่วนการถ่ายทอดสัญญาณแบบบัลก์ใช้สำหรับถ่ายทอดข้อมูลที่มีปริมาณมากๆ แต่ไม่ต้องการความต่อเนื่องของข้อมูล และในขณะที่การถ่ายทอดสัญญาณแบบอินเตอร์รัปต์ ใช้สำหรับถ่ายทอดข้อมูลที่มีจำนวนน้อยครั้งและมีปราณของข้อมูลไม่มากในการสั่งงานอุปกรณ์แต่ละครั้งนั้นโฮสต์จะต้องระบุเป้าหมายปลายทางของข้อมูลที่ต้องการจะรับหรือส่ง เป้าหมายปลายทางที่ว่านี้เป็นกลุ่มของรีจิสเตอร์ที่อยู่ในอุปกรณ์ชนิดต่างๆ 4 ชนิดข้างต้นแยกกันออกไป ดังนั้นอุปกรณ์แต่ละตัวจะมีจำนวนเอ็นด์พอยต์มากกว่า 1 เอ็นด์พอยต์เพื่อรองรับการทำงานรูปแบบต่างๆ ยกตัวอย่าง ซีดีรอม USB แบบอ่านเขียนได้จะต้องมีเอ็นด์พอยต์ที่รองรับการถ่ายทอดสัญญาณควบคุมเพื่อรับคำสั่งจากโฮสต์ 1 เอ็นด์พอยต์ พร้อมกันนั้นยังต้องมีเอ็นด์พอยต์ที่รองรับการถ่ายทอดสัญญาณแบบบัลก์เพื่อส่งข้อมูลที่อ่านได้หรือรับข้อมูลเพื่อเขียนแผ่นซีดี 1 เอ็นด์พอยต์ และต้องมีเอ็นด์พอยต์ที่รองรับการถ่าย ทอดสัญญาณไฮโซโครนัส เพื่อส่งข้อมูลเพลงซึ่งมีความต่อเนื่องในกรณ์ที่เล่นแผ่นซีดีเพลงอีก 1 เอ็นด์พอยต์ เป็นต้น

3.6 ดิสคริปเตอร์ : ความหมาย , ชนิดและรูปแบบการทำงาน

อุปกรณ์แต่ละตัวมีคุณสมบัติและการทำงานที่แตกต่างกัน โฮสต์จำเป็นต้องรู้คุณสมบัติทั้งหมดของอุปกรณ์แต่ละตัวเพื่อให้การสั่งงานเป็นไปอย่างถูกต้อง และเนื่องจากบัสข้อมูลทั้งหมดจะถูกใช้งานรับส่งข้อมูลร่วมกันระหว่างอุปกรณ์ทุกๆ ตัว สิ่งที่โฮสต์จำเป็นต้องรู้อีกอย่างคือ ปริมาณข้อมูลที่ต้องการส่ง(bandwidth) ของอุปกรณ์แต่ละตัวในบัส ดังนั้นเมื่อมีอุปกรณ์ตัวใหม่ต่อเข้ากับบัสโฮสต์ต้องอ่านข้อ มูลต่างๆ ที่จำเป็นเข้ามาเพื่อใช้อ้างอิงในการสั่งงานอุปกรณ์ และใช้พิจารณาว่าระบบบัสสามารถรองรับอุปกรณ์ที่มาเชื่อมต่อใหม่ได้หรือไม่ ข้อมูลเหล่านี้รวมเรียนว่า ดิสคริปเตอร์ของอุปกรณ์ (devicedes cripters) ซึ่งอุปกรณ์แต่ละตัวจะแจ้งรายละเอียดของตัวเองให้โฮสต์รู้ผ่านดิสคริปเตอร์ชนิดต่างๆ ซึ่งได้รับการแบ่งแยกเป็นชนิดตามข้อมูลที่จะแจ้งกลับไปยังโฮสต์ โดยาการแบ่งแยกชนิดของดิสคริปเตอร์นั้นจะจัดเป็นลำดับชั้น สาเหตุที่ต้องจัดเป็นระดับชั้นเป็นเพราะว่าอุปกรณ์แต่ละตัวนั้นอาจมีการทำงานที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลากหลายรูปแบบ เช่น แบ่งการทำงานออกเป็น 2 โหมด แต่ละโหมดมีหน้าที่การทำงานแตกต่างกัน และแต่ละหน้าที่ก็จะใช้กลุ่มของเอ็นด์พอยต์ที่แตกต่างกันซึ่งสามารถสรุปได้ดังนี้

3.6.1 ดีไวซ์ดิสคริปเตอร์ (Device descripter)

ทำหน้าที่หลักในการเก็บข้อมูลโดยทั่วไปของตัวอุปกรณ์ ซึ่งในอุปกรณ์แต่ละตัวจะมีดีไวซ์ดิสคริปเตอร์เพียง 1 ชุดเท่านั้น ภายในจะระบุข้อมูลที่ใช้ในการเชื่อมต่อขั้นแรก (default communications pipe) เพื่อใช้ในการกำหนดข้อมูลสถานะของอุปกรณ์เข้า กับโฮสต์ นอกจากนั้นยังเก็บข้อมูลของข้อกำหนดในโหมดการทำงานต่างๆ ของตัวอุปกรณ์รวมถึงจำนวนคอนฟิกิวเรชั่นด้วย เพราะในครั้งแรกที่อุปกรณ์เชื่อมต่อเข้ากับบัสนั้นโฮสต์ไม่มีทางรู้ได้เลยว่า ต้องติดต่อกับอุปกรณ์ที่เอ็นด์พอยต์ใด จึงจำเป็นต้องขอข้อมูลส่วนนี้ก่อนที่จะติดต่อกับส่วนอื่น

3.6.2 คอนฟิกิวเรชั่นดิสคริปเตอร์ (Configuration descripter)

ใช้เก็บข้อมูลที่จำเป็นของการทำงานในแต่ละโหมดการทำงานและเก็บจำนวนอินเตอร์เฟสที่ใช้งานในโหมดนั้นๆ เช่น อุปกรณ์บางตัวมีการทำงาน2โหมดคือ โหมดใช้พลังงานสูงและโหมดประหยัดพลังงาน ดิสคริปเตอร์ตัวนี้จะเก็บข้อมูลที่จำเป็นในการตั้งค่าต่างๆ ของโฮสต์เมื่อต้องการเลือกใช้โหมดการทำงานแต่ละโหมดของอุปกรณ์

3.6.3 อินเตอร์เฟสดิสคริปเตอร์ (Interface descripter)

ใช้เก็บข้อมูลหน้าที่การทำงานภายใน ซึ่งในแต่ละโหมดการทำงานหรือคอนฟิกิวเรชั่นอาจจะมีการเชื่อมต่อหรืออินเตอร์เฟสเพียง1แบบหรือมากกว่าเพื่อใช้งานในหน้าที่ต่างๆตัวอย่างที่เห็นได้ชัดที่สุดก็คือซีดีรอม โดยในตัวซีดีรอมจะมีการรับส่งข้อมูลปริมาณมากๆ(mass storage), การส่งข้อมูลเสียงออดิโอ และการส่งข้อมูลภาพจะเห็นได้ว่า มีอินเตอร์เฟสที่ใช้งานแยกกัน ในอินเตอร์เฟสดิสคริปเตอร์จะบรรจุข้อมูลการใช้งานอินเตอร์เฟสนั้นๆ โดยข้อมูลเหล่านี้จะระบุว่าอุปกรณ์ถูกจัดอยู่ในคลาส(class) หรือคลาสย่อย(subclass)ใด และมีเอ็นด์พอยต์จำนวนเท่าใดที่ใช้งานในอินเตอร์เฟสนี้บ้าง

3.6.4 เอ็นด์พอยต์ดิสคริปเตอร์ (Endpoint descripter)

ใช้เก็บข้อมูลคุณสมบัติแต่ละเอ็นด์พอยต์ เช่น ใช้การถ่ายทอดสัญญาณแบบใด (ไอโซโครนัส, บัลก์, อินเตอร์รัปต์ หรือสัญญาณควบคุม) และถ่ายทอดข้อมูลได้มากที่สุดครั้งละเท่าใด

3.6.5 สตริงค์ดิสคริปเตอร์ (String descripter)

เป็นดิสคริปเตอร์ที่ไม่ได้อยู่ภายในโครงสร้างเพราะดิสคริปเตอร์ชนิดนี้จะเก็บข้อมูลตัวอักษรที่สามารถอ่านเข้าใจได้ไว้อธิบายส่วนต่างๆ ของดิสคริปเตอร์ทั้งสี่ตัวข้างต้น เช่น เก็บชื่อบริษัทผู้ผลิต และ /หรือหมายเลขประจำตัวของอุปกรณ์ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.6 คลาสสเปกซิฟิกดิสคริปเตอร์(Class-specific descripter)

เป็นดิสคริปเตอร์พิเศษที่มีเฉพาะในอุปกรณ์บางตัวที่จัดอยู่ในบางคลาสเท่านั้น ภายในเก็บข้อมูล เฉพาะของคลาสนั้นๆ ที่อยู่นอกเหนือจากดิสคริปเตอร์พื้นฐาน 4 ชนิดแรก

3.7 การจัดการกับอุปกรณ์บนบัส USB ที่มีความเร็วต่างกัน

อุปกรณ์USB แบ่งตามความเร็วของการถ่ายทอดข้อมูลได้ 2 ชนิดคือ อุปกรณ์ความเร็วเต็มที่จะถ่ายทอดข้อมูลที่อัตรา 12Mb/sและอุปกรณ์ความเร็วต่ำถ่ายทอดข้อมูลที่อัตรา 1.5 Mb/s แต่เนื่องจากอุปกรณ์ทั้งสองประเภทนี้ต่ออยู่บนบัสเดียวกันทั้งหมด ดังนั้นพอร์ตของฮับที่ให้อุปกรณ์USB เข้ามาเชื่อมต่อจะต้องรองรับการทำงานได้ทั้ง 2 แบบและด้วยความที่เป็นระบบบัส อุปกรณ์ทุกๆ ตัวจะได้รับข้อมูลทุกๆแพ็กเก็ตที่ส่งเข้ามาไม่ว่าจะเป็นการส่งด้วยความเร็วสูงหรือต่ำดังนั้นจึงต้องมีการจัดการจราจรระหว่างข้อมูลที่ส่งด้วย โดยตั้งเป็นข้อกำหนดว่า พอร์ตของอุปกรณ์ความเร็วต่ำต้องไม่เปิดทำงานจนกว่าจะได้รับสัญญาณ “Preamble” จากโฮสต์ โดยหลังจากได้รับสัญญาณ Preamble แล้ว ฮับจะเปลี่ยนความเร็วในการถ่ายทอดข้อมูลไปสู่โหมดความเร็วต่ำ แล้วเปิดการทำงานของพอร์ตที่ความเร็วต่ำทำให้อุปกรณ์ที่เชื่อมต่อด้วยความเร็วต่ำได้รับข้อมูล ในทางกลับกัน หลังจากได้รับสัญญาณ Preamble แล้วอุปกรณ์ความเร็วเต็มที่ก็จะทราบทันทีว่าข้อมูลที่ตามหลังมาจะอยู่ในโหมดความเร็วต่ำ ไม่ต้องอ่านและตีความข้อมูลเหล่านั้น

3.8 การส่งสัญญาณในบัสUSB

สัญญาณที่ปรากฏบนสายสัญญาณระหว่างรูตฮับและอุปกรณ์จะส่งไปในแบบสัญญาณผลต่าง (differentialsignaling) เพื่อลดการแพร่กระจายสนามแม่เหล็กไฟฟ้า (EMI:ElectromagneticInterference) เนื่องจากตามธรรมชาติของสัญญาณไฟฟ้า เมื่อมีการเปลี่ยนแปลงระดับสัญญาณด้วยความเร็วมากๆ จะทำให้เกิดการแพร่กระจายของสนามแม่เหล็กและสนามไฟฟ้าออกมารอบๆ สายส่งสัญญาณ ซึ่งอาจรบกวนการทำงานของอุปกรณ์ต่างๆ รอบข้างได้ด้วยการส่งสัญญาณแบบนี้จะทำให้การเปลี่ยนแปลงของสัญญาณในสายนำสัญญาณเกิดขึ้นมาพร้อมกันและเกิดในลักษณะตรงข้าม ทำให้สนามแม่เหล็กไฟฟ้าที่เกิดขึ้นหักล้างกัน ไม่แพร่ออกมาภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9 กระบวนการกำหนดการทำงานของอุปกรณ์

ในส่วนนี้เป็นลำดับขั้นตอนในการเชื่อมต่ออุปกรณ์ USB ไปยังคอมพิวเตอร์ในเบื้องต้น โดยมีลำดับขั้นตอนดังต่อไปนี้

- ฮับตรวจสอบพบว่ามีการเชื่อมต่ออุปกรณ์ตัวใหม่เข้าสู่ระบบ แล้วแจ้งผลกลับไปยังโฮสต์คอนโทรลเลอร์
- โฮสต์คอนโทรลเลอร์สั่งให้ฮับเปิดการทำงานของแหล่งจ่ายไฟในโหมดประหยัด เพื่อให้อุปกรณ์ที่อาศัยพลังงานจากบัสสามารถทำงานได้
- โฮสต์คอนโทรลเลอร์สั่งให้ฮับรีเซตพอร์ตที่อุปกรณ์รีเซตค่าแอดเดรสและเอ็นด์พอยต์ของตัวเองให้เป็นค่าเริ่มต้น(default)
- โฮสต์อ่านดิสคริปเตอร์ต่างๆ จากตัวอุปกรณ์และพิจารณาว่าทรัพยากรของระบบพอเพียงต่อความต้องการของตัวอุปกรณ์หรือไม่ ทรัพยากรในที่นี้คือ พลังงานไฟฟ้าและปริมาณข้อมูลที่จะส่งของตัวอุปกรณ์ หากพิจารณาแล้วว่าไม่สามารถทำงานได้ก็จะสั่งให้ฮับปิดการทำงานของพอร์ตนั้น
- เมื่อโฮสต์พิจารณาแล้วว่าสามารถให้บริการแก่อุปกรณ์ตัวที่มาเชื่อมต่อได้จะควบคุมให้แหล่งจ่ายไฟ จ่ายพลังงานตามที่อุปกรณ์ต้องการ รวมถึงการตั้งค่าแอดเดรสและกำหนดค่าต่างๆ
- หลังจากตั้งค่าเรียบร้อย คอมพิวเตอร์ก็จะรู้จักกับอุปกรณ์ตัวใหม่และสามารถติดต่อกับอุปกรณ์ได้ทันทีโดยไม่จำเป็นต้องทำการ Restart เครื่องคอมพิวเตอร์

3.10 ระบบบัส USB 2.0

เป็นระบบบัสที่ได้รับการพัฒนาขึ้นอย่างมากจากระบบบัส USB 1.1 ซึ่งมีคุณสมบัติเด่นด้านความเร็วในการส่งถ่ายข้อมูลที่สูงถึง 480 Mbit/sec ทำให้ขอบเขตในการใช้งานระบบบัส USB กว้างขวางมากยิ่งขึ้น โดยอุปกรณ์ที่ใช้ระบบบัส USB 2.0 นั้นจำเป็นที่จะต้องต่อร่วมกับระบบ(โฮสต์และฮับ)ที่ใช้ระบบบัส USB2.0เช่นเดียวกันจึงจะสามารถส่งถ่ายข้อมูลที่ระดับความเร็วสูงนี้ได้แต่อย่างไรก็ตามระบบก็ยังสามารถใช้งานร่วมกับอุปกรณ์ที่เป็น USB 1.0/1.1 แต่เดิมได้ โดยส่วนของโฮสต์และฮับ จะทำหน้าที่ในการปรับระดับความเร็วในการส่งถ่ายข้อมูลให้ตรงกัน

ซึ่งคุณสมบัตินี้เป็นการเพิ่มความซับซ้อนให้กับวงจรและรูปแบบโปรโตคอลของโฮสต์และฮับแต่ก็เป็นการทำให้ไม่จำเป็นต้องใช้โฮสต์และฮับที่มีความเร็วแตกต่างกันหลายๆ ตัวในระบบ สำหรับในส่วนการอินเตอร์เฟส USB ของโครงงานนี้นั้นได้เลือกใช้ IC ที่ทำหน้าที่ในการแปลงรูปแบบการสื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารจาก USB ไปยังการสื่อสารในรูปแบบอนุกรม ซึ่งทำให้ไม่ต้องเข้าจัดการกับกระบวนการอินเตอร์เฟสกับUSBโดยตรงโดยจะกล่าวถึงรายละเอียดต่อ ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ระบบเครือข่าย (Network)

ในปัจจุบันการสื่อสารผ่านระบบเครือข่าย เป็นที่แพร่หลายมากในแง่การใช้งาน ,ใช้บริการ ,อำนวย ความสะดวก ในชีวิตประจำวันต่างๆเนื่องจากระบบสื่อสารที่รวดเร็วและสามารถเชื่อมโยง เข้ากับหลายๆระบบได้ ทำให้สามารถนำมาประยุกต์ใช้งานผ่านระบบเครือข่ายได้หลากหลายรูปแบบ

โดยระบบ TCP/IP นั้นเริ่มมาจาก องค์กรกลางที่เป็นผู้สร้างและกำหนดมาตรฐานคือ InterNIC แต่ InterNIC นั้นมิได้ทำงานเพียงลำพัง ยังมีองค์กรอื่นๆช่วยเหลือด้วย ที่สำคัญก็คือ IETF (Internet Engineering Task Force) ซึ่งเป็นผู้กำหนดมาตรฐานของโปรโตคอล TCP/IP ในแง่มุมต่างๆออกมาเป็น RFC(Request For Comment)

RFC เป็นข้อกำหนดมาตรฐานที่อธิบายถึงรปแบบของแพ็คเก็ตข้อมูลที่รับส่งกัน และอธิบายถึง State Machine ที่เกี่ยวข้องกับโปร โตคอล TCP/IP ว่าผู้รับและผู้ส่งจะพูดคุยกันอย่างไร RFC นั้นได้ถูกจำแนกออกเป็นหลายเบอร์ตามความหลากหลายของ TCP/IP

แต่ว่า RFC นั้นก็ยังเป็นเพียงมาตรฐานที่เขียนขึ้นมาเท่านั้น ต้องรอจนกว่าจะมีผู้พัฒนาซอฟต์แวร์หรือสร้างไดรเวอร์ขึ้นมาในลักษณะที่เป็นรูปธรรมโดยยึดอิงกับ RFC เสียก่อน ซึ่งในปัจจุบันได้กลายเป็นที่นิยมอย่างมากในการใช้งาน สามารถหาไดรเวอร์ และโปรแกรมที่ใช้งานผ่านมาตรฐาน TCP/IP ได้ทั่วไปใน Internet นั้นเอง

4.1 ระบบ TCP/IP

โปรโตคอล TCP/IP เป็นโปรโตคอลมาตรฐาน มาตรฐานหนึ่งที่มีการใช้งานกันอย่างแพร่หลายที่สุดในปัจจุบันที่ใช้ในระบบอินเตอร์เน็ต และ ระบบอีเทอร์เน็ต

TCP/IP เป็นการรูปแบบการสื่อสารข้อมูลในระบบอินเทอร์เน็ต ซึ่งแบ่งออกเป็น 4 ชั้น layer คล้าย OSI เพียงแต่ OSI มีทั้งหมด 7 ชั้น layer สำหรับ 4 layer ของระบบ IP มีดังนี้

1. **Application** ระดับนี้หมายถึงแอพพลิเคชั่นทั้งหมดของวินโดวส์ที่ทำงานบนเน็ตเวิร์ก ซึ่งครอบคลุมทั้งระดับ Application ,Presentation และ Session ในระดับ Application นั้น จะครอบคลุมถึง Application ทั้งหมดของวินโดวส์ ส่วนในระดับของ Presentation และ Session นั้นจะครอบคลุมเพียง Application Programing Interface (API) ที่รองรับการพัฒนาแอพพลิเคชั่นบนเน็ตเวิร์กเช่น WinSock ,NetBIOS เป็นต้น

2. **Transport**

3. **Netwok(Internet)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะขอกล่าวรวมทั้ง 2 เลเยอร์ไว้ด้วยกันเพราะมันคล้ายๆกัน สำหรับโปรโตคอลในระดับนี้ที่รู้จักกันทั่วๆไป ได้แก่ NetBEUI ,NWLink และ TCP/IP ในที่นี้เราสนใจเพียง TCP/IP เท่านั้น โดย TCP จะจัดอยู่ใน Transport Layer ส่วน IP จะจัดอยู่ใน Network Layer

TCP เป็นโปรโตคอลแบบที่มีการกำหนดช่วงการสื่อสารตลอดระยะเวลาที่มีการสื่อสาร (connection - oriented) ซึ่งจะยอมให้มีการส่งข้อมูลเป็นกระแสไบต์ (byte steam) โดยไม่มีข้อผิดพลาด ข้อมูลที่มีปริมาณมากจะถูกแบ่งเป็นส่วนเล็กๆที่เรียกว่า message ซึ่งจะถูกส่งไปยังผู้รับผ่านทางชั้นสื่อสารของอินเทอร์เน็ต ทางฝ่ายผู้รับจะนำ message มาเรียงต่อกันตามลำดับของข้อมูลตัวเดิม นอกจากนั้น TCP ยังมีความสามารถในการควบคุมการไหลของข้อมูลเพื่อป้องกันไม่ให้ผู้ส่ง ส่งข้อมูลเร็วเกินกว่าที่ผู้รับสามารถทำงานได้ทันอีกด้วย

ส่วน IP นั้นจะควบคุมส่วนสำคัญหลักๆ 3 ส่วนดังนี้

- IP Address : สำหรับการรับส่งข้อมูลในระบบอินเทอร์เน็ตและอีเทอร์เน็ตจะถูกกำหนดและอ้างอิงด้วยหมายเลขประจำเครื่องนั้นก็คือ IP Address
- Routing Configuration : ข้อดีของโปรโตคอล TCP/IP ก็คือมีความสามารถในการกำหนดเส้นทางสำหรับการรับส่ง ที่สามารถเลือกเส้นทางในการรับส่งข้อมูลได้อย่างอัตโนมัติถ้าหากเกิดเส้นทางบางเส้นทางเสียหาย ระบบกลไกในการกำหนดเส้นทางสำหรับการรับส่งข้อมูลของระบบ TCP/IP ก็จะเลือกเส้นทางให้เหมาะสมถูกต้องให้สามารถรับส่งข้อมูลได้
- Protocal ,Port ,Sockets : เป็นช่องทางสำหรับกำหนดทิศทางของการรับส่งข้อมูลนอกเหนือจากที่จะต้องกำหนดหลังจาก IP Address

4. Host - to – network มองง่ายๆว่า โปรโตคอลระดับนี้คือ เครือข่าย Ethernet ที่เสียบ เน็ตเวิร์กการ์ดแบบ Ethernet ที่ใช้หัวต่อแบบ RJ-45 เข้าที่เครื่องคอมพิวเตอร์แต่ละเครื่อง จากนั้นก็นำ สายอีกข้างของสาย UTP Cat 5 เข้ากับหัวต่อ RJ-45 และเสียบปลายสายอีกข้างหนึ่งเข้ากับอุปกรณ์ที่เรียกว่า Hub นั้นเอง

4.2 Server & Client

ในการเขียนโปรแกรมเพื่อใช้ในระบบเครือข่าย จุดหลักๆของระบบ จะแบ่งฝ่ายที่ต้องติดต่อรับส่งข้อมูลระหว่างกันออกเป็น 2 ส่วนได้แก่ Server (แม่ข่าย) และ Client (ลูกข่าย)

Server (แม่ข่าย) จะทำหน้าที่เสมือนกองอำนวยการ ,ประชาสัมพันธ์ รวมถึงผู้จัดการ ให้กับระบบ ,ลูกข่ายที่จะเป็นส่วนร้องขอข้อมูลจาก Server ในที่นี้จะเป็นส่วนที่เก็บข้อมูล ,จัดการ ,บริหารข้อมูลหรือทรัพยากรระบบเพื่อให้ฝ่ายลูกข่ายสามารถใช้บริการได้

Client (ลูกข่าย) จะทำหน้าที่ติดต่อกับ User เพื่อส่งข้อมูลหรือขอเรียกใช้บริการต่างๆจาก Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้ง Server และ Client ต่างก็ต้องมีตำแหน่งที่อยู่ (IP Address) ,ช่องทางการติดต่อ (Port) โดยทั้งสองฝ่ายจะต้องสามารถติดต่อกันได้ผ่านทางช่องทางการติดต่อเดียวกัน ซึ่งในการเขียนโปรแกรมนั้นเราสามารถกำหนดหมายเลขของ Port ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

คำสั่งในการควบคุมอุปกรณ์เครื่องมือวัด

ในการควบคุมอุปกรณ์เครื่องมือวัดต่างๆผ่านระบบบัส IEEE-488 นั้น จำเป็นจะต้องทราบถึงคำสั่งใช้งานของอุปกรณ์แต่ละตัว ซึ่งคำสั่งเหล่านี้ก็จะแตกต่างกันไปตามแต่ประเภทของอุปกรณ์ รวมถึงบริษัทผู้ผลิตโดยคำสั่งเหล่านี้จะถูกโปรแกรมไว้ในส่วนของโปรมแกรมควบคุมที่เขียนขึ้นทางฝั่งคอมพิวเตอร์ สำหรับอุปกรณ์เครื่องมือวัดที่ได้เลือกมาทดสอบในปริญญานิพนธ์นี้ประกอบด้วย

- HAMEG Programmable Power Supply รุ่น HM8142
- HAMEG Programmable Function Generator รุ่น HM8130
- HAMEG Storage Analog/Digital Oscilloscope รุ่น HM 1007

โดยมีรูปแบบคำสั่งต่างๆ ดังต่อไปนี้

5.1 HAMEG Programmable Power Supply รุ่น HM8142

เป็นแหล่งจ่ายพลังงานไฟฟ้ากระแสตรงขนาด 1A 2 Channel สามารถควบคุมผ่านระบบบัส IEEE-488 โดยมีคำสั่งดังสั่งการดังต่อไปนี้

คำสั่ง : RM1 / RM0

รูปแบบ : RM1

ฟังก์ชั่น : สั่งให้ Power Supply ทำงานในโหมดรีโมทโดยไม่สามารถควบคุมผ่านหน้าปัทม์ได้ แต่สามารถควบคุมการทำงานได้โดยผ่านระบบบัสแบบ IEEE-488 เท่านั้น และสามารถออกจากโหมดรีโมทได้โดยคำสั่ง RM0 หรือการกดปุ่ม LOCAL ที่หน้าปัทม์ของเครื่อง

รูปแบบ :RM0

ฟังก์ชั่น : ทำการยกเลิกโหมดรีโมท ให้เครื่องกลับสู่สถานะ LOCAL (สามารถควบคุมการทำงานของเครื่องได้โดยผ่านปุ่มหน้าปัทม์)

ข้อสังเกต : คำสั่ง RM 0 จะมีผลต่อคำสั่ง LK 1

คำสั่ง : MX1 / MX0

รูปแบบ : MX1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชั่น : เปลี่ยนโหมดรีโมทไปเป็นโหมดแบบผสมในโหมดนี้สามารถที่จะควบคุมการทำงานของเครื่องได้ทั้งโดยการส่งคำสั่งผ่านระบบแบบ IEEE-488 และการควบคุมผ่านทางหน้าปัทม์ ของเครื่อง

รูปแบบ : MX0

ฟังก์ชั่น : ยกเลิกโหมดแบบผสมและกลับเข้าสู่สถานะปกติที่ควบคุม โดยผ่านระบบบัสแบบ IEEE-488

คำสั่ง: LK1 / LK0

รูปแบบ : LK1

ฟังก์ชั่น : สั่งให้เครื่องเข้าสู่โหมดรีโมท โดยไม่สามารถกดปุ่ม LOCAL เพื่อที่จะให้เครื่องกลับสู่โหมด LOCAL ได้

รูปแบบ : LK0

ฟังก์ชั่น : สั่งให้เครื่องออกจากโหมดรีโมทที่ไม่สามารถกดปุ่ม LOCAL ทำให้สามารถกด ปุ่ม LOCAL เพื่อที่จะสามารถควบคุมการทำงานผ่านหน้าปัทม์ของเครื่องได้

ข้อสังเกต : โหมดรีโมทที่ไม่สามารถกดปุ่ม LOCAL นี้สามารถถูกยกเลิกได้โดยคำสั่ง RM0

คำสั่ง : SU1 และ SU2

รูปแบบ : SU1 VV.mV.mV หรือ SU2 01.34

ฟังก์ชัน : ตั้งค่าศักดาไฟฟ้าของแหล่งจ่ายที่ 1 หรือ 2

คำสั่ง : SI1 และ SI2

รูปแบบ : SI1 A.mAmAmA หรือ SI2 0.123

ฟังก์ชัน : ตั้งค่ากระแสไฟฟ้าของแหล่งจ่ายที่ 1 หรือ 2 เพื่อจำกัดค่ากระแสไฟฟ้า

คำสั่ง : RU1 และ RU2

รูปแบบ : RU1 หรือ RU2

ฟังก์ชัน : ค่าศักดาไฟฟ้าที่ส่งกลับโดยเครื่อง จะเป็นค่าศักดาไฟฟ้าที่ถูกกำหนดค่าโดยคำสั่งกำหนดค่า (SET)

ข้อสังเกต : ใช้คำสั่ง MUX เพื่อที่จะถามค่ากระแสไฟฟ้าจริงที่วัดได้จากเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง : RI1 และ RI2

รูปแบบ : RI1 หรือ RI2

ค่าที่ส่งกลับ : I1_1.000A หรือ I2_0.012A

ฟังก์ชัน : ค่ากระแสไฟฟ้าที่ส่งกลับโดยเครื่อง จะเป็นค่ากระแสไฟฟ้าที่ถูกกำหนดค่าโดยคำสั่งจำกัดค่ากระแสไฟฟ้า (SI1 หรือ SI2)

ข้อสังเกต : ใช้คำสั่ง MIX เพื่อถามค่ากระแสไฟฟ้าจริงที่วัดได้จากเอาท์พุทจริง

คำสั่ง : MU1 และ MU2

รูปแบบ : MU1 หรือ MU2

ค่าที่ส่งกลับ : U1:12.34V หรือ U2:12.24V

ฟังก์ชัน : ค่าศักดาไฟฟ้าที่ส่งกลับโดยเครื่อง ซึ่งเป็นค่าที่วัดได้จากเอาท์พุทจริง

ข้อสังเกต : คำสั่ง RUX ใช้เพื่อถามค่าศักดาไฟฟ้าที่ทำการกำหนด

คำสั่ง : MI1 และ MI2

รูปแบบ : MI1 หรือ MI2

ค่าที่ส่งกลับ : I1= +1.000A หรือ I2= -0.123A

ฟังก์ชัน : ค่ากระแสไฟฟ้าที่ส่งกลับโดยเครื่อง เป็นค่ากระแสไฟฟ้าที่วัดได้

ข้อสังเกต : คำสั่ง RIX ใช้เพื่อถามถึงค่ากระแสไฟฟ้าที่จำกัดเอาไว้ ถ้าเอาท์พุทถูกปิดไว้ค่าที่ส่งกลับจะเป็น I1=_1.000A

คำสั่ง : TRU

รูปแบบ : TRU

ฟังก์ชัน : กำหนดค่าศักดาไฟฟ้าของแหล่งจ่ายตัวที่ 1 และ 2 ให้มีค่าเท่ากันตามต้องการ

คำสั่ง : TRI

รูปแบบ : TRI

ฟังก์ชัน : กำหนดค่ากระแสไฟฟ้าของแหล่งจ่ายตัวที่ 1 และ 2 ให้มีค่าเท่ากันตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

รูปแบบ : STA

ค่าที่ส่งกลับ : OP1/0SR1/0ER0/1CV1/CC1/CV2/CC2/RM0/1

OP0 สวิทช์เอาท์พุทปิด

OP1 สวิทช์เอาท์พุทเปิด

SQ1 แสดงสถานะของอุปกรณ์ที่เปลี่ยนแปลง (CV ไปเป็น CC หรือ OP1 ไป เป็น OP0) (จะทำงานเฉพาะเมื่อ SRQ นั้น enable หรือมีค่าเป็น 1 , ดูคำสั่ง SR1)

SQ2 เมื่อสัญญาณ service request (SRQ) ถูกทำให้ enable แสดงว่าไม่มีการเปลี่ยนแปลงสถานะของเครื่องมือ

ER0 ไม่มีความผิดพลาด

ER1 ความร้อนสูงเกิน

CV1 แหล่งจ่ายตัวที่ 1 ทำการจ่ายศักดาไฟฟ้าคงที่

CC1 แหล่งจ่ายตัวที่ 1 ทำการจ่ายกระแสไฟฟ้าคงที่

CV2 แหล่งจ่ายตัวที่ 2 ทำการจ่ายศักดาไฟฟ้าคงที่

CC2 แหล่งจ่ายตัวที่ 2 ทำการจ่ายกระแสไฟฟ้าคงที่

ฟังก์ชัน : สั่งให้เครื่องส่งค่าสตริงที่เก็บสถานะปัจจุบันของเครื่องมาให้

คำสั่ง: OP1 และ OP0

รูปแบบ : OP1

ฟังก์ชัน : สวิทช์เอาท์พุทเปิด

รูปแบบ : OP0

ฟังก์ชัน : สวิทช์เอาท์พุทปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง : CLR (clear)

รูปแบบ : CLR

ฟังก์ชัน : ยกเลิกฟังก์ชั่นทั้งหมดของเครื่อง และทำการเริ่มต้นใหม่ที่สถานะศูนย์ (zero state) ในโหมดรีโมท คีย์บอร์ดไม่สามารถใช้งานได้และสวิทช์เอาท์พุทปิด รวมทั้งศักดาไฟฟ้า และกระแสไฟฟ้าจะถูกตั้งค่าให้เป็นศูนย์

คำสั่ง : VER

รูปแบบ : VER

ค่าที่ส่งกลับ : sw Vx.xhw Vx.xxxxxxx HAMEG/Paris KRP&VM

คำสั่ง : ID?

รูปแบบ : ID?

ค่าที่ส่งกลับ : HM8142-1

ฟังก์ชัน : แสดงถึงชื่อรุ่นของเครื่อง

5.2 HAMEG Programmable Function Generator รุ่น HM8130

ทำหน้าที่สร้างสัญญาณรูปแบบต่างๆ ป้อนให้แก่วงจรที่ทำการทดลอง และมีคำสั่งควบคุมฟังก์ชันในการทำงานผ่านระบบบัสแบบ IEEE-488 โดยสามารถแบ่งได้เป็นกลุ่มคำสั่งดังต่อไปนี้

หมวดคำสั่งที่ไม่ต้องมีข้อมูลประกอบ

คำสั่ง: SIN

รูปแบบ: SIN

ฟังก์ชัน : ฟังก์ชันการสร้างสัญญาณคลื่นรูปซายน์

คำสั่ง: TRI

รูปแบบ: TRI

ฟังก์ชัน: ฟังก์ชันการสร้างสัญญาณคลื่นรูปสามเหลี่ยม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง: SQR
รูปแบบ: SQR
ฟังก์ชัน : ฟังก์ชันการสร้างสัญญาณคลื่นรูปสี่เหลี่ยม

คำสั่ง: PLS
รูปแบบ: PLS
ฟังก์ชัน: ฟังก์ชันการสร้างสัญญาณคลื่นเป็นพัลซ์

คำสั่ง: RMS
รูปแบบ: RMS
ฟังก์ชัน : ฟังก์ชันการสร้างสัญญาณคลื่นรูปฟันเลื่อย โดยมีสัญญาณขาขึ้นเป็นบวก

คำสั่ง: RMN
รูปแบบ: RMN
ฟังก์ชัน : ฟังก์ชันการสร้างสัญญาณคลื่นรูปฟันเลื่อย โดยมีสัญญาณขาขึ้นเป็นลง

คำสั่ง: ARB
รูปแบบ: ARB
ฟังก์ชัน : ฟังก์ชันการสร้างสัญญาณคลื่นรูปแบบ arbitary

คำสั่ง: SW1/0
รูปแบบ: SW1/0
ฟังก์ชัน : การเลือกการกวาดสัญญาณเปิด/ปิด

คำสั่ง: CTM
รูปแบบ: CTM
ฟังก์ชัน : การเลือกโหมดการสร้างสัญญาณแบบต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง: GTM

รูปแบบ: GTM

ฟังก์ชัน : การเลือกโหมดการสร้างสัญญาณแบบGATED

คำสั่ง: TRM

รูปแบบ: TRM

ฟังก์ชัน : การเลือกโหมดการสร้างสัญญาณแบบTRIGGER

คำสั่ง: OT1/0

รูปแบบ: OT1/0

ฟังก์ชัน : การเลือกให้เอาท์พุทเปิด/ปิด

คำสั่ง: OF1/0

รูปแบบ: OF1/0

ฟังก์ชัน : การเลือกให้ออฟเซ็ทเปิด/ปิด

คำสั่ง: DFR

รูปแบบ: DFR

ฟังก์ชัน : ให้แสดงความถี่ของสัญญาณ

คำสั่ง: DST

รูปแบบ: DST

ฟังก์ชัน : ให้แสดงความถี่เริ่มต้นของสัญญาณ

คำสั่ง: DSP

รูปแบบ: DSP

ฟังก์ชัน : ให้แสดงความถี่สุดท้ายของสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง: DWT

รูปแบบ: DWT

ฟังก์ชัน : ให้แสดงความกว้างของสัญญาณพัลซ์

คำสั่ง: DSW

รูปแบบ: DSW

ฟังก์ชัน : ให้แสดงเวลาในการกวาดสัญญาณ

คำสั่ง: DAM

รูปแบบ: DAM

ฟังก์ชัน : ให้แสดงค่าแอมปลิจูดของเอาท์พุท

คำสั่ง: DOF

รูปแบบ: DOF

ฟังก์ชัน : ให้แสดงค่าออฟเซ็ท

คำสั่ง: RM0

รูปแบบ: RM0

ฟังก์ชัน : ยกเลิกโหมดรีโมท แล้วกลับเข้าสู่การควบคุมการทำงานผ่านหน้าปัทม์ของเครื่อง เงื่อนไขนี้สามารถทำได้โดยการกดปุ่ม LOCAL เช่นเดียวกัน

ข้อสังเกต: คำสั่งRM0จะยกเลิกคำสั่ง LK1

คำสั่ง: LK1

รูปแบบ: LK1

ฟังก์ชัน : ยกเลิก LOCAL ชั่วคราว โดยที่สถานะนี้การควบคุมการทำงานของเครื่องทำได้โดยผ่านทางระบบบัสเท่านั้น และการกลับสู่สถานะ LOCAL ไม่สามารถทำได้โดยการปุ่ม LOCAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง: LK0

รูปแบบ: LK0

ฟังก์ชัน : ยกเลิกสถานการณ์ยกเลิกปุ่ม LOCAL ชั่วคราว ทำให้สามารถกดปุ่ม LOCAL เพื่อที่จะกลับสู่สถานะ LOCAL ได้ ทำให้สามารถควบคุมการทำงานของเครื่องผ่านปุ่มหน้าปัทม์ได้อีกครั้ง

คำสั่ง: TRG

รูปแบบ: TRG

ฟังก์ชัน : TRIG คาบสัญญาณ (ปิดการกวาดสัญญาณ) หรือการกวาดสัญญาณที่สมบูรณ์ (การกวาดสัญญาณเปิด)

คำสั่ง: CLR

รูปแบบ: CLR

ฟังก์ชัน : รีเซ็ทเครื่องและกลับสู่สถานะเริ่มต้น คำสั่ง CLR เหมือนกับคำสั่ง SDC ตามมาตรฐานของ IEEE-488

คำสั่ง: ARC

รูปแบบ: ARC

ฟังก์ชัน : ลบข้อมูลที่เป็นการสร้างสัญญาณแบบใดๆ และรีเซ็ทตัวนับภายในให้ค่าเป็นศูนย์

คำสั่ง: ARE

รูปแบบ: ARE

ฟังก์ชัน : ยกเลิกการเขียนข้อมูลที่ใช้สร้างแบบ arbitary

หมวดคำสั่งที่ประกอบด้วยข้อมูลที่เป็นเลขทศนิยม

คำสั่งทั้งหมดที่ประกอบด้วยข้อมูลแบบทศนิยมจะประกอบด้วยตัวอักษร 3 ตัวแล้วตามด้วย ":" ความยาวของข้อมูลที่มากที่สุดจะเป็นเลขจำนวน 5 หลักรวมกับจุดทศนิยมด้วย โดยรูปแบบของคำสั่ง นั้นจะมีค่าเลขยกกำลังหรือไม่มีก็ได้ จะมีจุดทศนิยมหรือไม่มีก็ได้ แต่จะมีหน่วยเป็นโวลท์, เฮริทซ์และวินาที แต่จะไม่มีการส่งค่าหน่อยลงไปบนบัส ข้อมูลสามารถมีเครื่องหมายนำหน้าที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ถ้าจำเป็น แต่สัญญาณที่เป็นบวกไม่ต้องมีเครื่องหมายนำหน้าก็ได้ ระหว่างเครื่องหมายกับค่าที่กำหนดจะต้องติดกันห้ามเว้นวรรค

ตัวอย่าง : FRQ:1000.0 มีค่าเท่ากับ FRQ:1000.0

FRQ:1E3 มีค่าเท่ากับ FRQ:1E+3

FRQ:1.0000E+3 มีค่าเท่ากับ FRQ:10.000E+2

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

คำสั่ง : AMP

รูปแบบ:AMP:<ข้อมูล>

ฟังก์ชัน: กำหนดค่าแอมปลิจูดของสัญญาณ < > V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง : OFS

รูปแบบ:OFS:<ข้อมูล>

ฟังก์ชัน: กำหนดค่าออฟเซ็ทของสัญญาณ < > V

หมายเหตุ : ค่าความถี่และเวลากำหนดได้มากที่สุด 5 หลัก และค่าศักดาไฟฟ้ากำหนดได้มากที่สุด 3 หลัก ค่าแอมปลิจูดสามารถกำหนดได้ 2 วิธี ถ้าเป็นค่าที่ไม่มีเครื่องหมายจะถือว่าเป็นค่าศักดาไฟฟ้าแบบ peak to peak (ในกรณีของสัญญาณรูปพัลซ์นั้น ศักดาไฟฟ้าสูงสุดจะมีค่าเท่ากับครึ่งหนึ่งของค่านี้) ถ้าเป็นค่าที่มีเครื่องหมายนำหน้าจะถือว่าเป็นค่าศักดาไฟฟ้าสูงสุด

หมวดคำสั่งร้องขอ

คำสั่งเหล่านี้จะทำการเก็บค่าสตริงของข้อมูลที่สามารถอ่านออกมาได้ทันทีที่เครื่องถูกกำหนดให้เป็นตัวส่ง โดยมีคำสั่งต่างๆ และรูปแบบ คือ

คำสั่ง : FRQ?

รูปแบบ: FRQ?

ฟังก์ชัน: แสดงความถี่ของสัญญาณ

คำสั่ง : STT?

รูปแบบ:STT?

ฟังก์ชัน: แสดงความถี่เริ่มต้นของสัญญาณ

คำสั่ง : STP?

รูปแบบ:STP?

ฟังก์ชัน: แสดงความถี่สุดท้ายของสัญญาณ

คำสั่ง : SWT?

รูปแบบ:SWT?

ฟังก์ชัน: แสดงเวลาในการกวาดสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง : WDT?

รูปแบบ:WDT?

ฟังก์ชัน: แสดงความกว้างของสัญญาณพัลซ์

คำสั่ง : AMP?

รูปแบบ:AMP?

ฟังก์ชัน: แสดงค่าแอมปลิจูดของศักดาไฟฟ้าเอาท์พุท

คำสั่ง:OFS?

รูปแบบ:OFS?

ฟังก์ชัน: แสดงค่าออฟเซ็ท

คำสั่ง : ARD?

รูปแบบ:ARD?

ฟังก์ชัน: แสดงข้อมูลของสัญญาณแบบ arbitary

คำสั่ง : ID?

รูปแบบ:ID?

ฟังก์ชัน: แสดงหมายเลยรุ่นของเครื่อง

คำสั่ง : VER?

รูปแบบ:VER?

ฟังก์ชัน: แสดงเวอร์ชันของซอฟต์แวร์และฮาร์ดแวร์ของเครื่อง

คำสั่ง : STA?

รูปแบบ:STA?

ฟังก์ชัน: แสดงสถานะของเครื่อง โดยสตริงข้อมูลที่อ่านได้จะมีทั้งหมด 21 ตัวอักษร ซึ่งจะบอกถึงสถานะของเครื่อง โดยมีความหมาย คือ

OT0OF0SW0SINCTMDFRDAM

1. OT0 สวิทช์เอาท์พุทเปิด (ปิด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. OF0 การเซ็ทค่าออฟเซ็ทปิด (เปิด)

3. SW0 การกวาดสัญญาณปิด (เปิด)

4. SIN รูปแบบของสัญญาณ (ซายน์)

5. CTM โหมดของการทำงาน (โหมดการสร้างสัญญาณแบบต่อเนื่อง)

6. DFR แสดงรายละเอียด (ความถี่)

7. DAM แสดงรายละเอียด (แอมปลิจูด) ค่าเอาท์พุทจะอยู่ในรูปแบบเลขทศนิยมกับเลขยกกำลัง สตริงข้อมูลแต่ละค่าจะขึ้นต้นด้วยคำสั่งของตัวมันเอง เช่น

"FRQ:1.2345E+3

"OFS:-3.0E+0 "WDT:45.6E-6

สำหรับการส่งคำสั่งควบคุมเครื่องสามารถที่จะส่งคำสั่งไปควบคุมพร้อมกันหลายฟังก์ชัน โดยสามารถส่งคำสั่งเหล่านั้นไปพร้อมกันเป็นข้อมูลชุดเดียวกันได้ เช่น

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

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

5.3 HAMEG Storage Analog/Digital Oscilloscope รุ่น HM 1007

เป็น Osilloscope ที่สามารถวัดสัญญาณที่มีค่าความถี่สูงสุด 100 เมกกะเฮิร์ทซ์ โดย Oscilloscope นี้สามารถที่จะทำการบันทึกสัญญาณที่ได้จากการวัดมาเก็บไว้ในหน่วยความจำของเครื่อง และสามารถที่จะอ่านค่าของสัญญาณที่บันทึกเก็บไว้ผ่านระบบบัสแบบ IEEE 488 ได้ แต่ไม่สามารถที่จะควบคุมช่วงของเวลาต่อช่องและจำนวนช่องต่อโวลท์ได้ สำหรับคำสั่งใช้ควบคุมการทำงานผ่านระบบบัส IEEE 488 มีดังนี้ คือ

คำสั่ง :ID?

รูปแบบ:ID?

ฟังก์ชัน:ทำการส่งหมายเลขรุ่นของเครื่องลงบนหน่วยความจำของเครื่อง

คำสั่ง:DIG

รูปแบบ:DIG[channel code]

ฟังก์ชัน: 1.ส่งข้อมูลเก็บไว้ในหน่วยความจำของเครื่องเฉพาะช่องการวัดที่ 1

2.ส่งข้อมูลเก็บไว้ในหน่วยความจำของเครื่องเฉพาะช่องการวัดที่ 2

3.ส่งข้อมูลเก็บไว้ในหน่วยความจำของเครื่องเฉพาะช่องการวัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง:GET

รูปแบบ:GET [Channel code]

ฟังก์ชัน: 1.ส่งสัญญาณ TRIG โดยไม่กำหนดช่องการวัด

2.ส่งสัญญาณ TRIGโดยกำหนดช่องการวัด

คำสั่ง:STA

รูปแบบ:STA

ฟังก์ชัน: ตรวจสอบสถานะเซ็ทช่องการวัดของ Oscilloscope

คำสั่ง:TXT

รูปแบบ:TXT <สตริง@>

ฟังก์ชัน: ส่งสตริงไปยังการอินเตอร์เฟส

คำสั่ง:OFS

รูปแบบ:OFS

ฟังก์ชัน: ตรวจสอบการเลื่อนตำแหน่งของ Y จากหน่อยความจำอ้างอิง หลังจากสัญญาณอ้างอิงถูกเก็บไว้ในหน่อยความจำ

คำสั่ง:FRM

รูปแบบ:FRM [format]

ฟังก์ชัน: กำหนดรูปแบบข้อมูลที่ทำการส่ง (0 ข้อมูลเป็นแบบไบนารี)

คำสั่ง:V24

รูปแบบ:V24

ฟังก์ชัน: ส่งข้อมูลผ่านระบบบัสแบบ IEEE 488 ไปยังอุปกรณ์ภายนอกที่ต่อผ่านระบบบัสแบบ RS232C

คำสั่ง:PRN

รูปแบบ:PRN

ฟังก์ชัน: เริ่มส่งข้อมูลผ่านพอร์ตใดพอร์ตหนึ่งในจำนวน 3 พอร์ตของเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง:XYZ

รูปแบบ:XYZ

ฟังก์ชัน: เปลี่ยนโหมดการอ่านค่าสัญญาณเป็นการอ่านค่าสัญญาณในโหมด XY

สิ่งสำคัญสิ่งหนึ่งซึ่งขาดไม่ได้คือในการจบชุดคำสั่งของแต่ละข้อมูลจะต้องตามด้วยรหัส ASCII อักษรตัวที่ 13 (ในทางการเขียนโปรแกรม) หรือการกด Enter นั่นเอง ในส่วนของการส่งของชุดข้อมูลคำสั่งเหล่านี้นั้น จำเป็นที่จะต้องอาศัยสัญญาณควบคุมการรับส่งข้อมูล (handshake lines) และสัญญาณ EOI ในการจัดเรียงชุดข้อมูลด้วย เนื่องจากข้อมูลนี้นั้นถูกส่งออกไปครั้งละตัวอักษร ดังนั้นจึงต้องอาศัยสัญญาณ EOI เพื่อจัดเรียงข้อมูลแต่ละตัวอักษรนั้นเป็นชุดคำสั่งนั่นเอง นอกเหนือจากคำสั่งของอุปกรณ์เหล่านี้ หากเป็นการใช้งานร่วมกับอุปกรณ์อื่นๆ ก็สามารถทำได้โดยการเปลี่ยนแปลงคำสั่งของโปรแกรมที่เขียนขึ้นควบคุมให้เหมาะสมกับอุปกรณ์เครื่องมือวัดเหล่านั้นนั่นเอง

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การออกแบบส่วนการอินเตอร์เฟสของระบบ

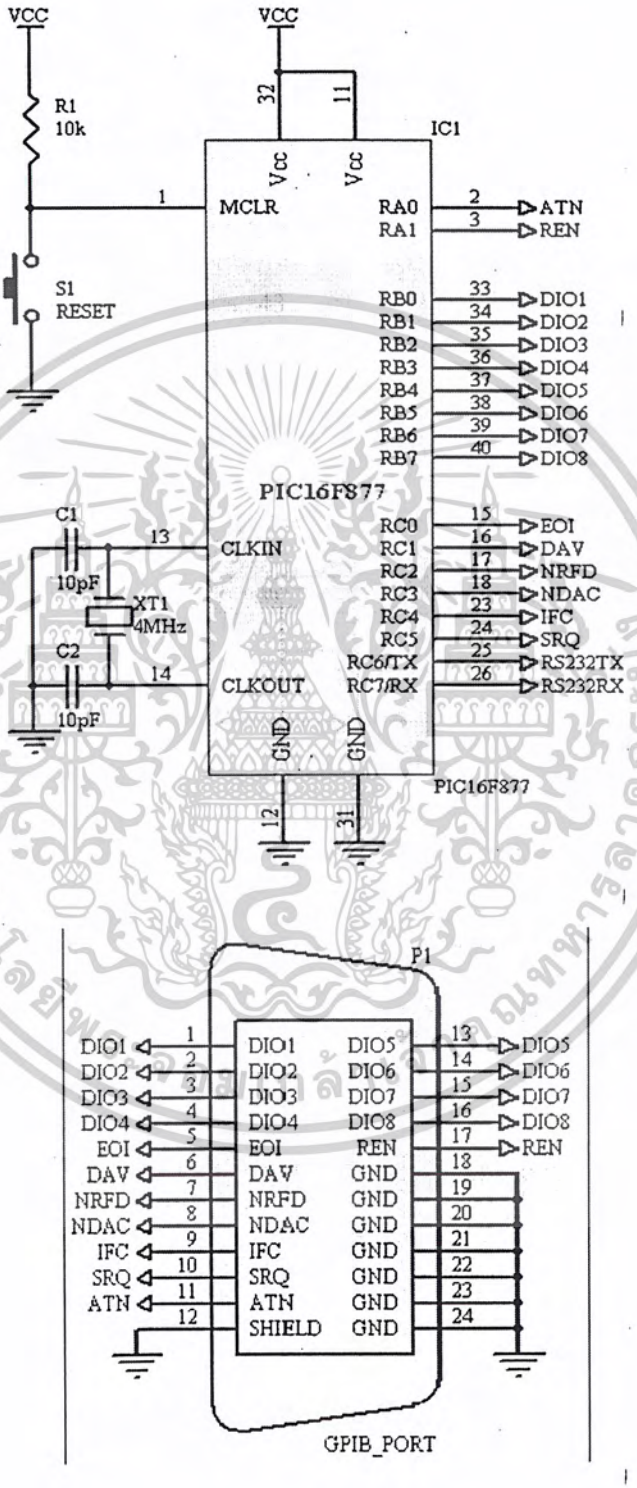
ในส่วนนี้นั้นจะกล่าวถึงวงจรในส่วนต่างๆ ของส่วนอินเตอร์เฟส ซึ่งแบ่งออกเป็นอินเตอร์เฟส และหน้าที่การทำงานของแต่ละส่วนประกอบ ส่วนประกอบหลักๆ ได้ดังนี้

6.1 Microcontroller

ส่วนนี้ถือเป็นหัวใจสำคัญของระบบ เนื่องจากทำหน้าที่ในการจัดเรียงข้อมูลที่รับเข้ามาในรูปแบบข้อมูลอนุกรมเพื่อส่งออกไปให้แก่เครื่องมือวัดในมาตรฐาน GPIB หรือในทางกลับกันคือรับข้อมูลจากอุปกรณ์เครื่องมือวัดทางพอร์ต GPIBเพื่อส่งให้กับคอมพิวเตอร์ในรูปแบบข้อมูลอนุกรม ซึ่งในที่นี้ได้เลือกใช้ Microcontroller เบอร์ PIC16F877 โดยมีคุณสมบัติในการจ่ายกระแสของแต่ละพอร์ตมากถึง 25mA (Sink/Source 25mA) ซึ่งเพียงพอที่จะจ่ายไปสั่งการอุปกรณ์เครื่องมือวัดที่ ต่อร่วมกันอยู่หลายๆ ตัวได้ อีกทั้งคุณสมบัติทางด้านความเร็วในการทำงาน โดยไมโคร คอนโทรลเลอร์ในตระกูลนี้จะใช้สัญญาณนาฬิกาเพียง 1 หรือ 2 สัญญาณในการทำงาน 1 คำสั่ง ทำให้สามารถทำงานได้เร็วกว่าไมโครคอนโทรลเลอร์บางตระกูลที่ความถี่เดียวกัน

จากรูปที่ 6.1จะเห็นได้ว่าพอร์ต B จะถูกใช้งานเป็นพอร์ต Data ในการส่งข้อมูล ซึ่งค่า Data ต่างๆ ที่จะส่งไปยัง GPIB นั้น ต้องรำลึกไว้เสมอว่าข้อมูลในมาตรฐาน GPIB นั้นจะเป็นในลักษณะตรงกันข้ามกับข้อมูลที่คอมพิวเตอร์รับรู้ดังนั้นค่าที่ส่งและรับเข้ามานั้นต้องมีการกลับค่าก่อนเสมอ สำหรับสัญญาณอื่นๆนั้น ต่อร่วมกับพอร์ตต่างๆ ตามรูปวงจร และสำหรับในส่วนของการสื่อสารไปยังคอมพิวเตอร์ จะเป็นการสื่อสารผ่านทาง USART (Universal Synchronous Asynchronous Receiver Transmitter) ซึ่งเป็นพอร์ตสื่อสารอนุกรมในตัวไมโครคอนโทรลเลอร์เอง โดยการติดต่อสื่อสารไปยังคอมพิวเตอร์โดย USB และ RS-232 นั้นจะกล่าวถึงต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



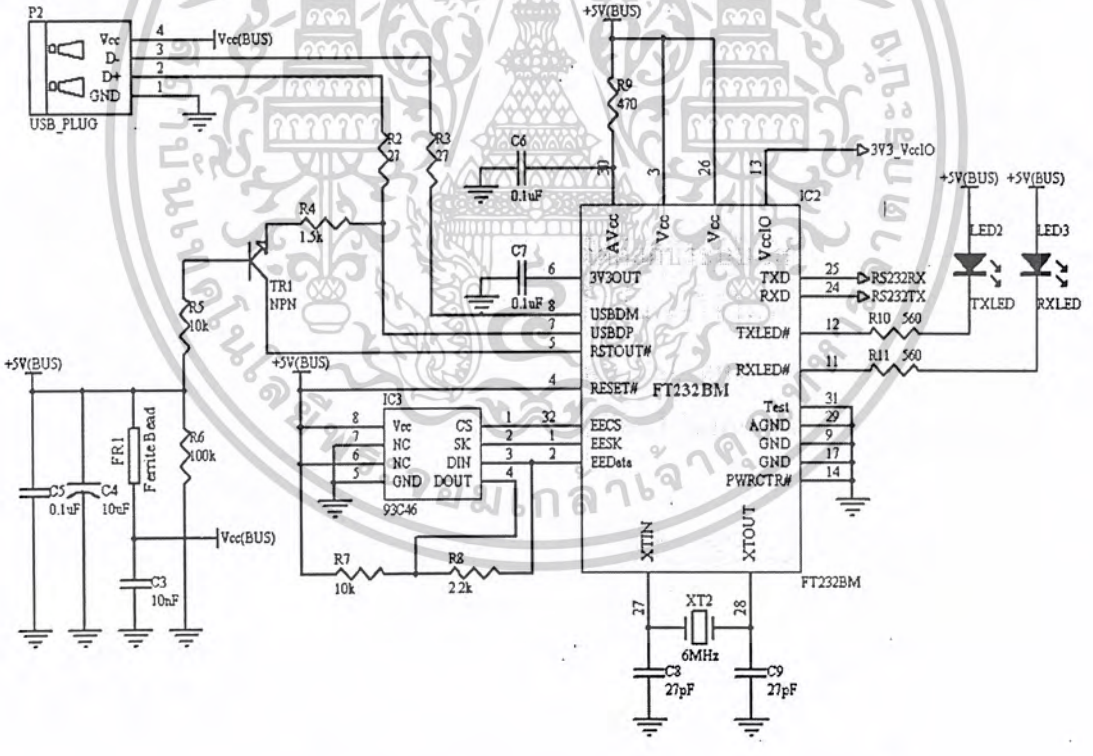
รูปที่ 6.1 แสดงการเชื่อมต่อระหว่างวงจรส่วนไมโครคอลโทรลเลอร์ และ พอร์ต GPIB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 USB Converter

สำหรับส่วนนี้นั้นเป็นส่วนที่ทำหน้าที่ในการเปลี่ยนแปลงการสื่อสารอนุกรมจากไมโครคอลโทรลเลอร์ไปสู่การทางสื่อสารผ่านระบบบัส USB ซึ่งในที่นี้นั้นได้เลือกใช้ IC เบอร์ FT232BM ของบริษัท Future Technology Devices International Limited :FTDI ซึ่งมีลักษณะการส่งถ่ายข้อมูลภายในแบบ FIFO โดย IC เบอร์นี้มีความสามารถในการจัดการกับโปรโตคอล USBได้อย่างสมบูรณ์ โดยไม่ต้องอาศัยเฟิร์มแวร์(Firmware)ในการติดต่อกับระบบบัส USB เพิ่มเติม

สำหรับในการติดต่อกับโปรแกรมที่เขียนขึ้นนั้น จะใช้ไดร์เวอร์พอร์ตอนุกรมเสมือน (Vitual COM Port : VCP) ซึ่งทำให้การเชื่อมต่อผ่านทาง USB ถูกมองเป็นการเชื่อมต่อในลักษณะของพอร์ตอนุกรม (COM) ทำให้มีความสะดวกในการเขียนโปรแกรมติดต่อไปยังพอร์ต USB เป็นอย่างมาก

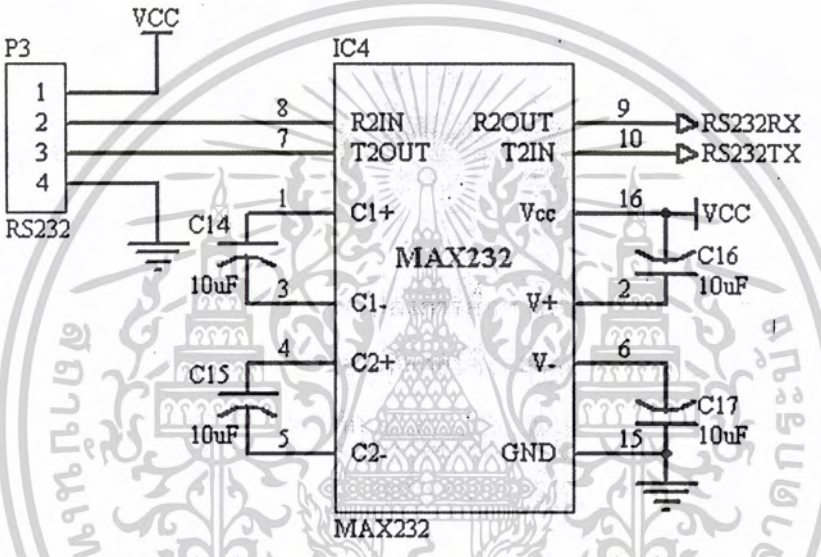


รูปที่ 6.2 แสดงวงจรในส่วนของ USB Convertor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 RS-232

วงจรในส่วนนี้จะทำหน้าที่เปลี่ยนแปลงการสื่อสารอนุกรมของไมโครคอนโทรลเลอร์ให้เป็นไปตามมาตรฐานของการสื่อสารทางคอมพิวเตอร์ซึ่งมีค่าลอจิกที่สูงกว่า โดยวงจรที่ใช้เป็นวงจรมาตรฐานซึ่งใช้ IC เบอร์ MAX232 เป็นหลัก

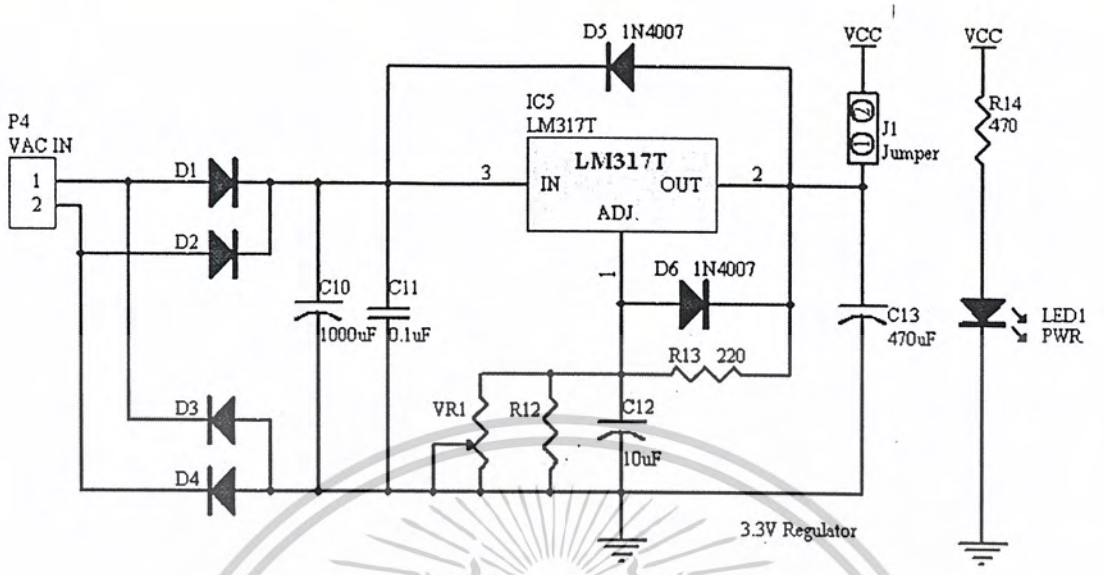


รูปที่ 6.3 แสดงวงจรในส่วนของ RS-232

6.4 Regulator

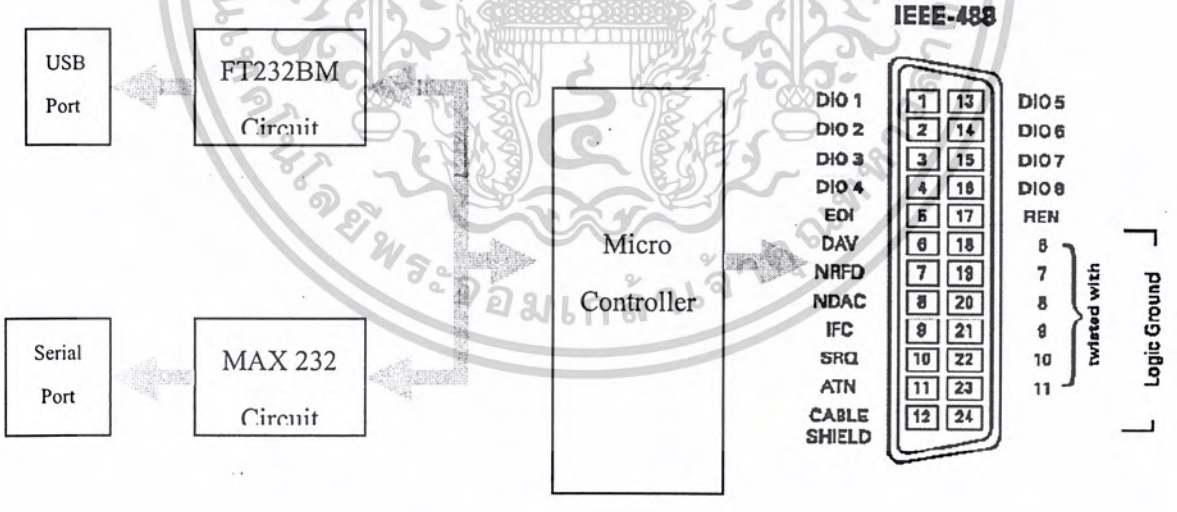
ในส่วนนี้ทำหน้าที่ในการสร้างแรงดัน 3.3 V ซึ่งในที่นี้เราใช้ IC เบอร์ LM317T เป็นตัวสร้างแรงดันเพื่อนำไปใช้ในส่วนของ USB Converter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



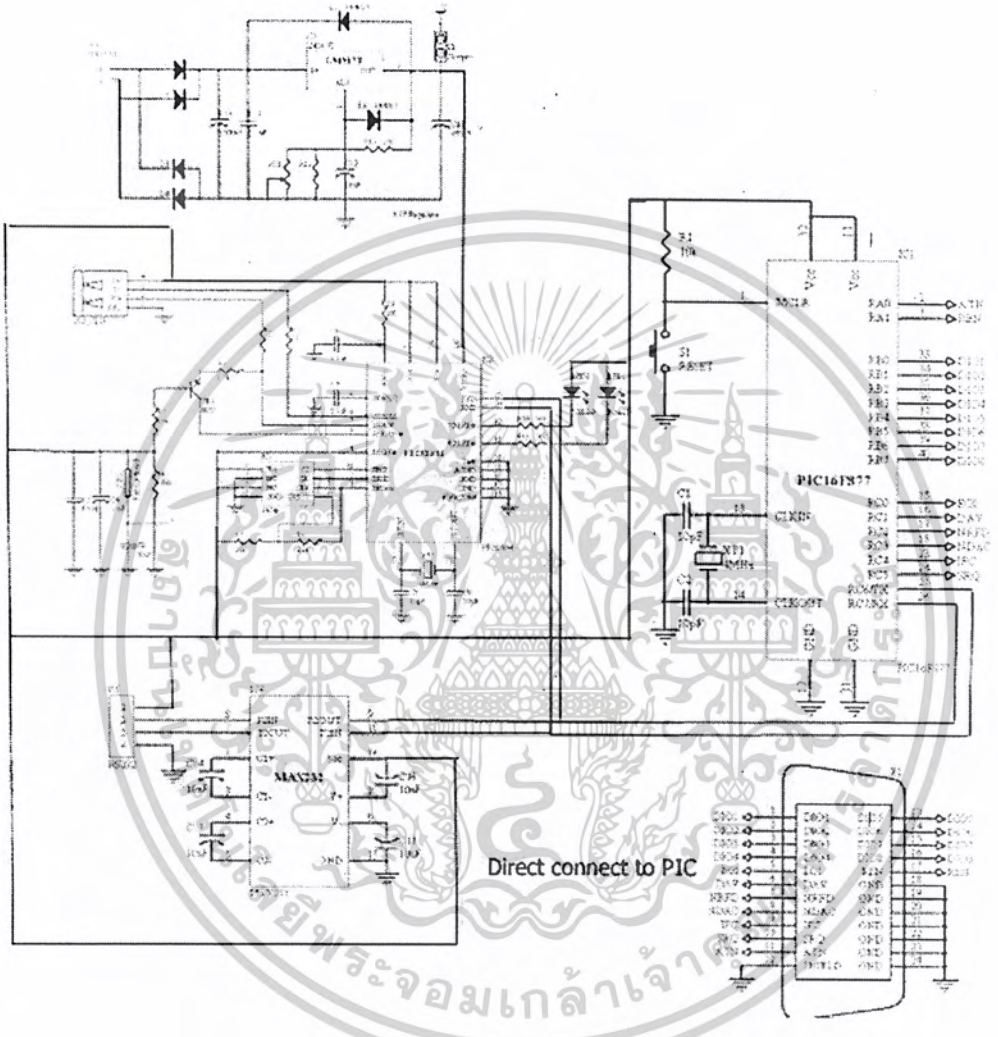
รูปที่ 6.4 วงจรในส่วนของ Regulator

จากส่วนประกอบทั้งหมดเมื่อนำมาเขียนเรียงกันในแผ่น PCB เดียวกันก็จะสามารถนำมาเป็นส่วนของฮาร์ดแวร์(Hardware)ที่ใช้ในโครงงานชิ้นนี้



รูปที่ 6.5 Block Diagrams ของวงจรรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.6 วงจรรวมของบอร์ดไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

การทำงานภายในไมโครคอลโทรลเลอร์

7.1 ขบวนการติดต่อไปยังอุปกรณ์เครื่องมือวัด

ในการเชื่อมต่อระบบไปยังอุปกรณ์เครื่องมือวัดต่างๆนั้น เนื่องจากอุปกรณ์เครื่องมือวัดที่ต่อร่วมอยู่ภายในระบบนั้นล้วนแต่ใช้สายส่งข้อมูล(Data)ชุดเดียวกัน ดังนั้นในช่วงเวลาหนึ่งจะมีอุปกรณ์ที่ได้รับหรือส่งข้อมูลเพียงอุปกรณ์เดียวเท่านั้น ดังนั้นส่วนที่จะทำหน้าที่ในการระบุอุปกรณ์ที่ได้รับอนุญาตก็คือ Address นั่นเอง ซึ่งในการระบุ Address นั้นจำเป็นจะต้องกระทำก่อนการรับหรือการส่งข้อมูลไปยังอุปกรณ์ใดๆ โดยลำดับขั้นตอนของการระบุAddress นั้นเป็นไปตามลำดับขั้นตอนดังนี้

ในขั้นตอนแรกนั้นสัญญาณที่มีความสำคัญในการกำหนดรูปแบบของข้อมูลที่สั่งการออกไปนั่นก็คือ สัญญาณ ATN ซึ่งมีความสำคัญต่อระบบดังต่อไปนี้

- ในขณะที่สัญญาณ ATN เป็น Low

ในขณะที่สัญญาณ ATN เป็น Low นั้นข้อมูลบนบัสข้อมูลของมันนั้นทางอุปกรณ์จะรับรู้ว่าข้อมูลเหล่านี้เป็นรหัสคำสั่งมาตรฐานของ GPIB ซึ่งในระหว่างที่ ATN เป็น Low นี้รหัสคำสั่งจะถูกส่งออกไปกี่คำสั่งก็ได้

- ในขณะที่สัญญาณ ATN เป็น High

เมื่อสัญญาณ ATN เป็น High นั้นข้อมูลบนบัสข้อมูลของมันจะหมายถึงชุดข้อมูลคำสั่ง ซึ่งชุดข้อมูลนี้นั้นจะแตกต่างกันไปสำหรับอุปกรณ์ที่แตกต่างกัน โดยการส่งข้อมูลเหล่านี้จะถูกส่งออกไปทีละตัวอักษร (ตามรหัส ASCII) และถูกควบคุมโดยสัญญาณ EOI เพื่อระบุชุดของนั่นเอง

ในการกำหนดการเริ่มติดต่อสื่อสารข้อมูลระหว่างกันนั้นจำเป็นจะต้องระบุ ADDRESS ของส่วนที่ทำการติดต่อ โดยในการกำหนดส่วนการเริ่มต้นนี้นั้นจะแบ่งออกได้เป็น 2 กรณี ได้แก่ ระหว่างตัวควบคุมกับตัวส่ง หรือตัวควบคุมกับตัวรับ ซึ่งในขณะที่ทางด้านหนึ่งทำหน้าที่เป็นผู้ส่งอีกด้านหนึ่งก็จะทำหน้าที่เป็นตัวรับนั่นเอง

-ขณะที่ตัวควบคุมติดต่อกับภาคส่ง

ในขณะนี้นั้นตัวควบคุมจะทำหน้าที่เสมือนเป็นด้านทางรับ (Listener) โดยในขณะนี้นั้น ATN จะถูกกำหนดให้เป็น Low เป็นอันดับแรกเพื่อกำหนดให้เข้าสู่โหมดของการสั่งงานคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

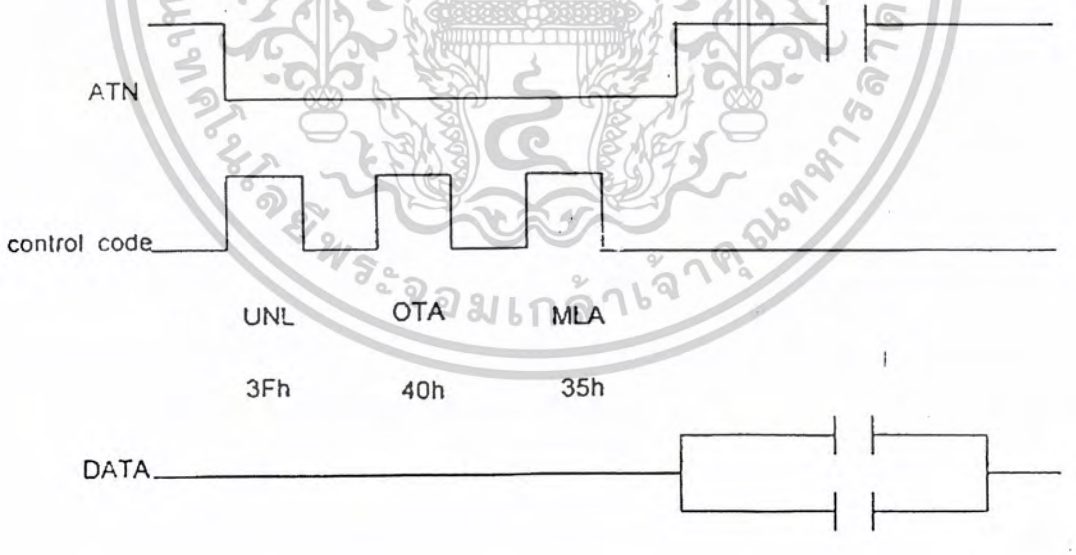
มาตรฐาน ดังแผนผังเวลาในรูปที่ 5.2 ซึ่งในส่วนของการกำหนดค่าคำสั่งต่างๆ นั้นมีรายละเอียดดังต่อไปนี้

UNL (3Fh) (Unlistener) คำสั่งนี้นั้นเป็นคำสั่งเคลียร์อุปกรณ์ที่ทำหน้าที่เป็นตัวรับที่ต่อพ่วงอยู่ภายในระบบทุกตัว ซึ่งจะทำให้ไม่มีอุปกรณ์ใดในระบบทำหน้าที่เป็นตัวรับอีก

OTA(40h+Address) (Our Talker Address) คำสั่งนี้นั้นเป็นการแจ้งค่า Address ของตัว ควบคุมให้อุปกรณ์ที่ต่อพ่วงอยู่บนบัสทุกตัวรับรู้ ซึ่ง Address ของตัวควบคุมนั้นจะเป็น 0 โดยในการส่งนั้นจะเป็นการส่งค่า 40h บวกค่า Address ของตัวควบคุมไป ได้แก่ 0จึงเป็นการส่งค่า 40h ออกไปนั่นเอง

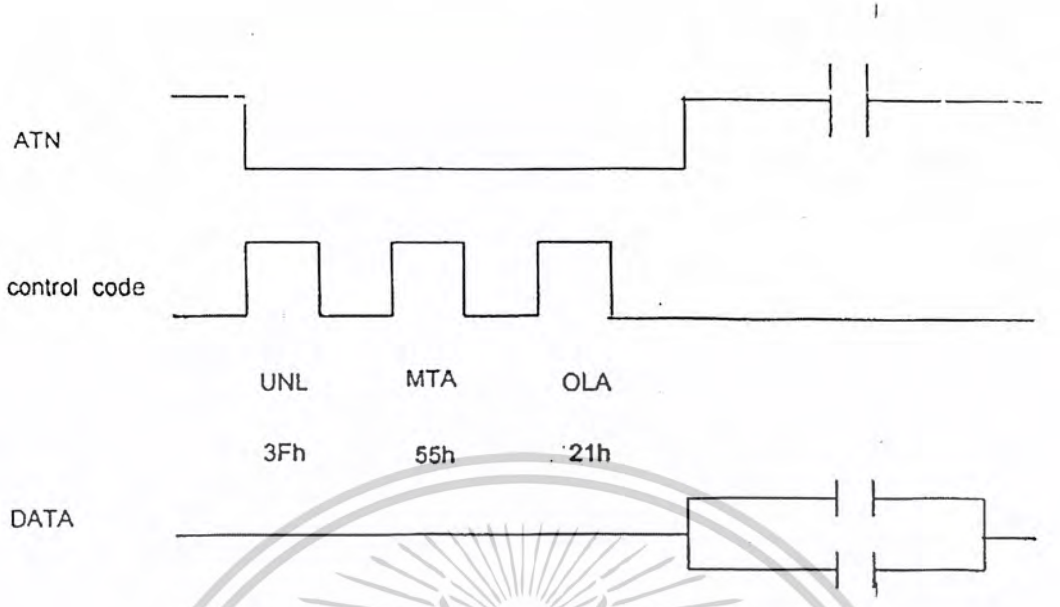
MLA (20h+Address) (My Listener Address) คำสั่งนี้นั้นเป็นการกำหนดค่าของอุปกรณ์ตัวส่งซึ่งเป็นไปในลักษณะเดียวกับกำหนด Address ของตัวควบคุม แต่จะเป็นการส่งค่า 20 บวกกับค่า Address อุปกรณ์ที่ต้องการให้ทำหน้าที่เป็นตัวส่ง ซึ่งในตัวอย่างรูปที่ 5.1 นี้นั้นได้แก่ 21 (15h) ดังนั้นค่าที่ส่งออกไปจึงเป็น 20h+15h = 35h ดังตัวอย่าง

หลังจากที่คำสั่งนี้ถูกสั่งงานแล้ว ตัวควบคุมจะทำการสั่งให้สัญญาณ ATN เป็น High เพื่อที่จะเริ่มทำการรับค่าที่ส่งมาจากตัวส่งต่อไป



รูปที่ 7.1 แสดงผังเวลาเมื่อตัวควบคุมติดต่อไปยังภาคส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7. 2 แสดงผังเวลาเมื่อตัวควบคุมติดต่อไปยังภาครับ

-ขณะที่ตัวควบคุมติดต่อกับภาครับ

ในขณะที่ตัวควบคุมทำหน้าที่เป็นภาคส่งเพื่อติดต่อกับภาครับนี้นั้นจะเป็นไปในลักษณะเดียวกับขณะที่ตัวควบคุมทำหน้าที่เป็นภาครับ โดยการสั่งการจะเริ่มต้นจากการที่กำหนดให้สัญญาณ ATN เป็น Low แล้วจึงเริ่มส่งชุดข้อมูลออกไป

UNL (3Fh) (Unlistener)เป็นคำสั่งยกเลิกอุปกรณ์ที่ทำหน้าที่เป็นตัวรับอยู่ทุกตัง เพื่อรอรับคำสั่งใหม่

MTA (40h+Address) (My Talker Address) เป็นคำสั่งที่จะแจ้ง Address ของตัวควบคุมที่จะให้ตัวรับรู้ว่าส่งมาจาก Address ใดโดยค่าในการส่งจะเป็น 40h+Address ซึ่งของตัวควบคุมซึ่งในตัวอย่างรูปที่ 7.2 นั้น Address ของตัวควบคุมจะเป็น 21 (15h) ดังนั้นค่าที่ส่งไปจึงเป็น 40h+15h = 55h ดังรูป

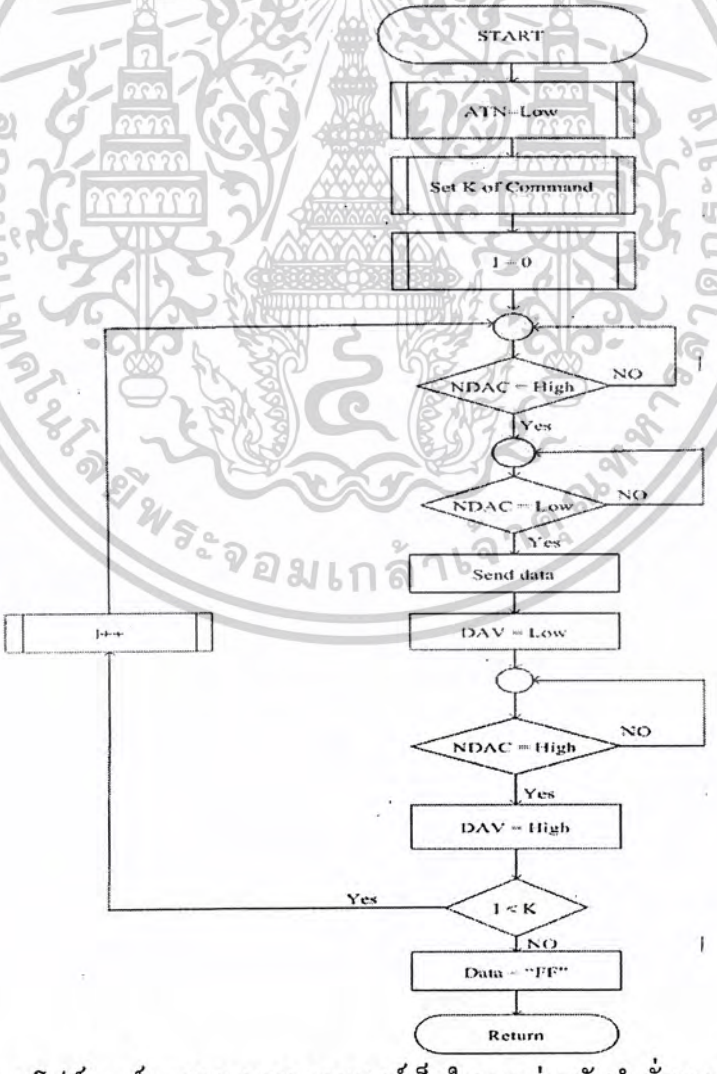
OLA (20h+Address) (Our Listener Address) เป็นคำสั่งที่ใช้กำหนด Address ของอุปกรณ์ที่ทำหน้าที่เป็นตัวรับ โดยค่าที่ส่งออกไปจะเป็น 20h+Address อุปกรณ์ตัวรับ ซึ่งดังรูปนั้น อุปกรณ์ตัวรับมี Address เป็น 1 ดังนั้นค่าที่ส่งออกไปจึงเป็น 21h นั่นเอง

- ขบวนการแฮนด์เช็คในการส่งรหัสคำสั่งมาตรฐาน

ในขั้นแรกนั้นเป็นการกำหนดให้สัญญาณ ATN เป็น Low เพื่อเป็นการบ่งบอกว่าเป็นการส่งรหัสคำสั่งมาตรฐาน (ในสภาวะปรกตินั้นสัญญาณ ATN ก็จะคงอยู่ในสภาวะ Output Low) ต่อมาก็จะเป็นการกำหนดจำนวดชุดของคำสั่งที่ต้องการจะทำการส่ง โดยเก็บค่าในตัวแปร K และ

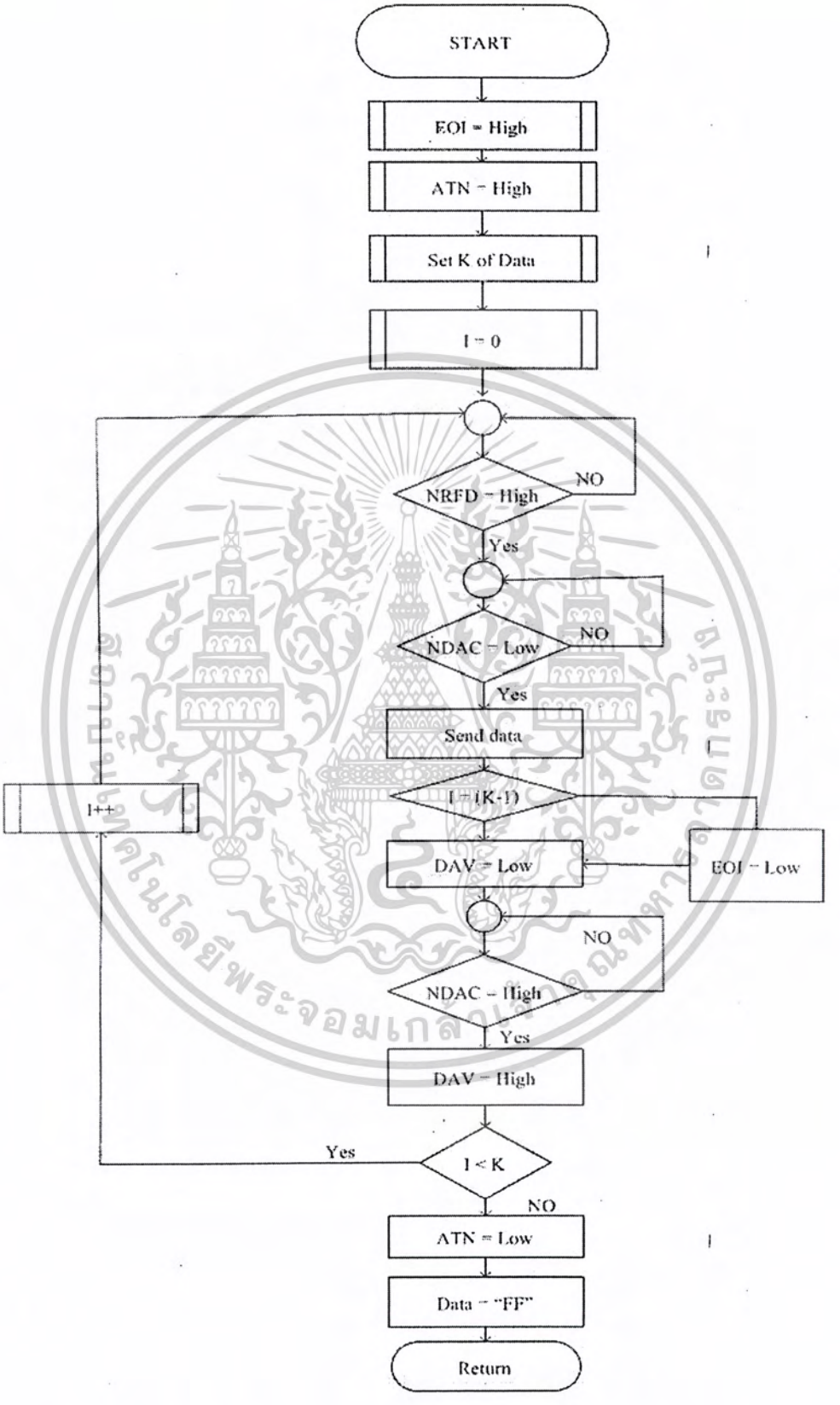
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดตัวแปรนับชุดข้อมูล (I) เป็น 0 หลังจากนั้นจะเริ่มเข้าสู่ขบวนการส่ง โดยตรวจสอบสัญญาณ NRFD ว่าเป็น High หรือไม่ และตรวจสอบสัญญาณ NDAC ว่าเป็น Low หรือไม่ เมื่อเป็นไปตามเงื่อนไขแล้วข้อมูลจะถูกส่งออกไปยังพอร์ต Data (ชุดข้อมูลจากไมโครคอนโทรลเลอร์ ต้องทำการ Invert ก่อนที่จะทำการส่งไปยังพอร์ต Data) หลังจากนั้นชั่วขณะหนึ่งสัญญาณ DAV จะถูกกำหนดให้เป็น Low เพื่อบ่งบอกไปยังทางภาคส่งว่าข้อมูลได้ถูกส่งออกไปแล้ว หลังจากนั้นทางภาคส่งจะตรวจสอบสัญญาณ NDAC อีกครั้งว่าเป็น High หรือไม่ ซึ่งเป็นการบ่งบอกว่าข้อมูลที่ส่งไปนั้นทางภาครับได้รับข้อมูลแล้ว ทางภาคส่งก็จะกำหนดให้สัญญาณ DAV เป็น High เพื่อเป็นการบ่งบอกการสิ้นสุดของการส่งข้อมูลใน 1 ชุดข้อมูล หลังจากนั้นจะเป็นการตรวจสอบว่าชุดรหัสคำสั่งนั้นสิ้นสุดแล้วหรือไม่ หากยังไม่จบชุดของรหัสก็จะทำการเข้าสู่กระบวนการส่งอีกครั้งเพื่อส่งรหัสคำสั่งที่เหลืออยู่ หลังจากขบวนการส่งจบสิ้นแล้ว ก็จะทำการเคลียร์พอร์ต Data โดยกำหนดข้อมูลให้เป็น FF เป็นอันเสร็จสิ้นขบวนการส่งรหัสคำสั่ง



รูปที่ 7.3 โฟล์วชาร์ตแสดงขบวนการแฮนด์เช็คในการส่งรหัสคำสั่งมาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.4 โฟล์วชาร์ตแสดงขบวนการแฮนด์เช็คในการส่งชุดข้อมูลคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 รูปแบบของชุดคำสั่งที่คอมพิวเตอร์ส่งมายังบอร์ดอินเตอร์เฟส GPIB

เนื่องจากชุดข้อมูลที่จะส่งไปยังอุปกรณ์ต่างๆนั้น จำเป็นต้องมีการระบุ Address ของอุปกรณ์นั้นๆ ดังนั้นเราจึงมีการออกแบบรูปแบบของข้อมูลที่ส่งดังนี้

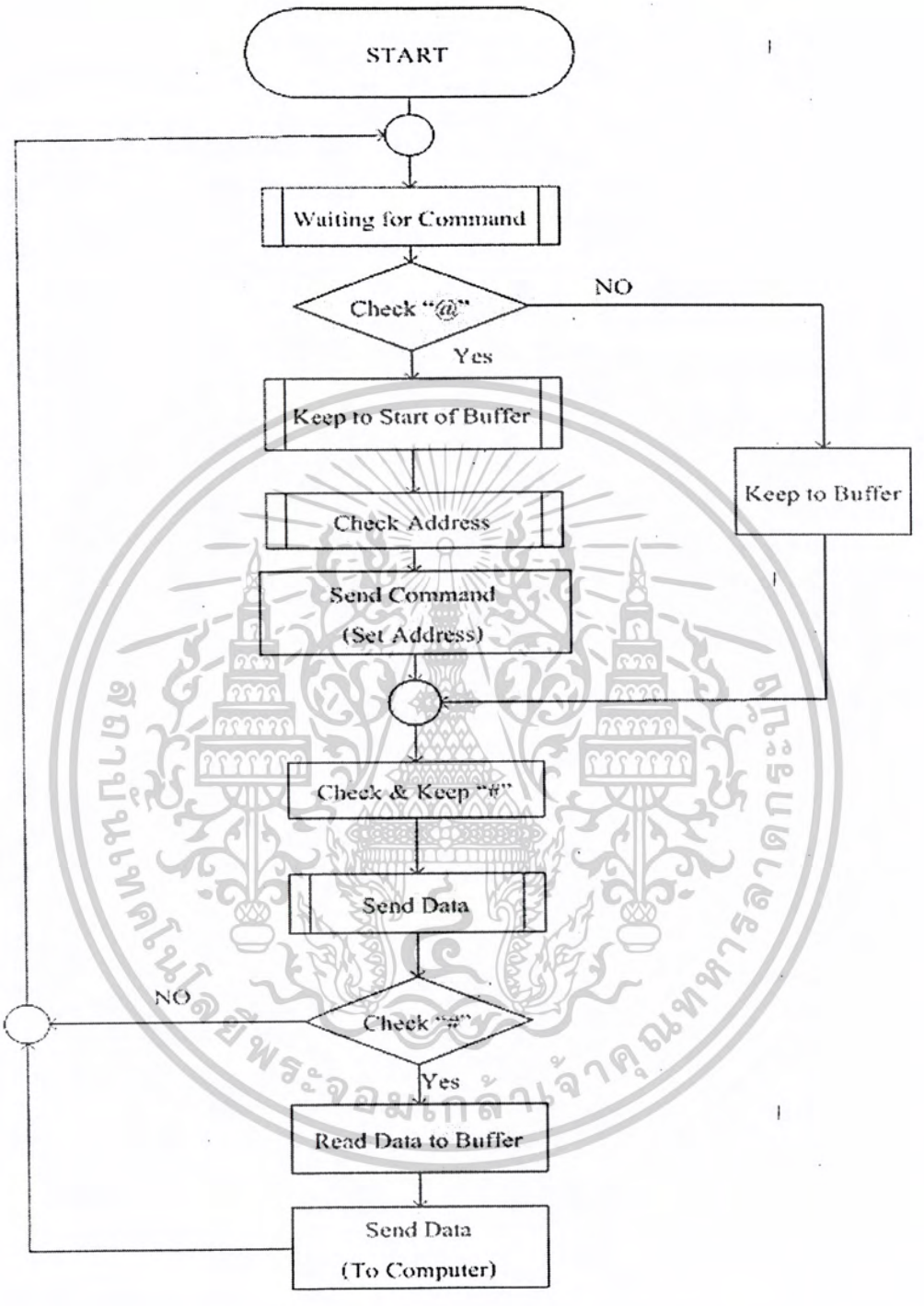
@[Address][Command][#]

@	เครื่องหมาย @ นั้นใช้บ่งบอกว่าหลังจากนี้จะเป็นชุดคำสั่ง
[Address]	ใช้ระบุ Address ของอุปกรณ์เครื่องมือวัดที่ต้องการติดต่อ
[Command]	เป็นส่วนของชุดคำสั่งที่ต้องการส่งไปยังไมโครคอลโทรลเลอร์
[#]	ถ้ามีเครื่องหมาย # แสดงว่าข้อมูลชุดที่สั่งไปนั้นๆ มีข้อมูลจะต้องรับกลับมา (ถ้าไม่มีข้อมูลจะต้องส่งกลับเวลาส่งชุดข้อมูลไปยังไมโครคอลโทรลเลอร์ไม่จำเป็นต้องมีเครื่องหมาย #)

7.3 โปรแกรมการทำงานของไมโครคอลโทรลเลอร์

ในส่วนของการทำงานนั้นไมโครคอลโทรลเลอร์จะคอยตรวจสอบว่าเจอเครื่องหมาย “@” ไหม ถ้าเจอจะเริ่มเขียนที่ตำแหน่งแรกของบัฟเฟอร์และทำการแยกระหว่าง Address และ ชุดคำสั่ง และตรวจสอบในตอนสุดท้ายด้วยว่าพบเครื่องหมาย “#” ไหม ซึ่งถ้าเจอเครื่องหมาย “#” แปลว่าจะต้องรอรับข้อมูลจากเครื่องมือวัดเพื่อส่งกลับไปยังคอมพิวเตอร์ต่อไปดังโฟร์ลชาร์ตของรูปที่ 7.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



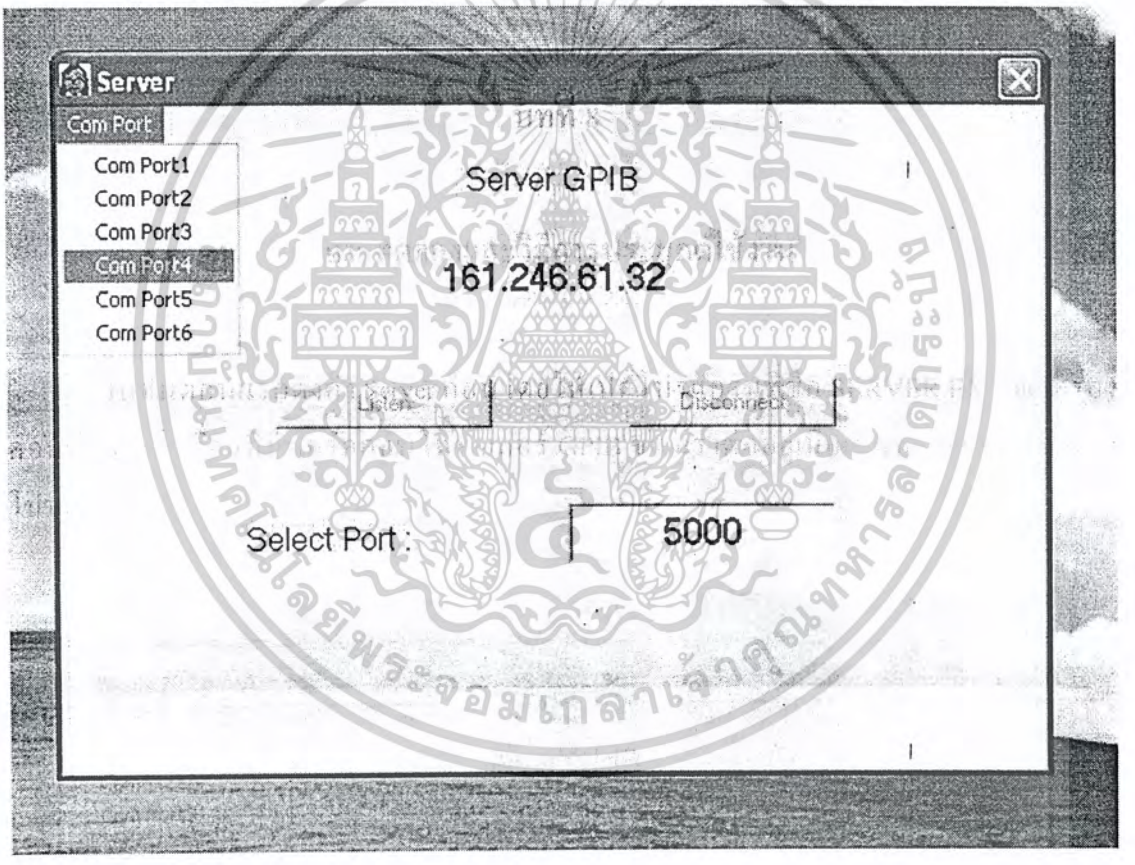
รูปที่ 7.5 โฟล์วชาร์ตแสดงการทำงานของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

การทดลองและวิธีการประยุกต์ใช้งาน

ในขั้นตอนแรกตั้งค่า Server ก่อนโดยให้เปิดโปรแกรมที่ชื่อ SERVER.EXE ณ เครื่องคอมพิวเตอร์เครื่องที่มีบอร์ดคอนโทรลเลอร์ GPIB ของเราต่ออยู่เมื่อเปิดมาจะเห็นหน้าต่างโปรแกรมดังนี้

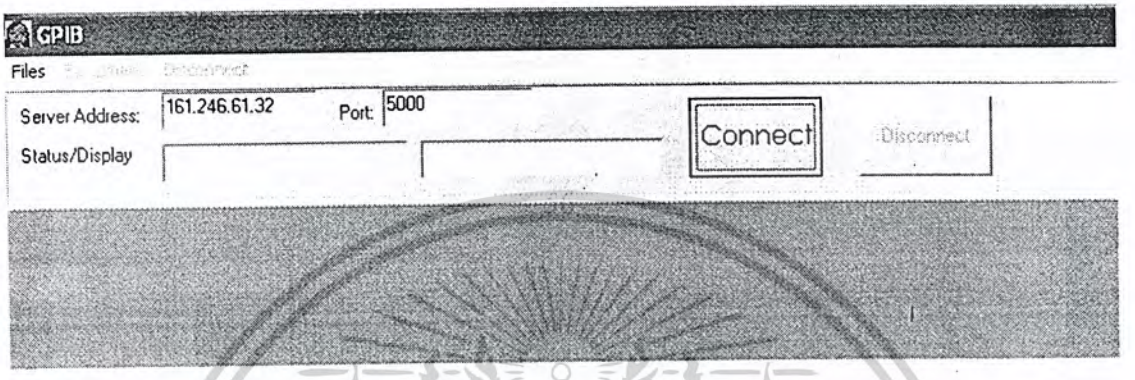


รูปที่ 8.1 รูปหน้าตา INTERFACE ของโปรแกรม SERVER.EXE

ให้ทำการเลือกที่ Com Port ให้ตรงกับการ Connect ของอุปกรณ์ (ถ้าเป็น USB จะเป็น Port 3,4,5 หรือ 6) หลังจากนั้นก็ตั้งค่าของ Socket Port โดย Socket Port นั้นก็คือช่องทางการติดต่อระหว่าง Server และ Client ซึ่งโปรแกรมทุกโปรแกรมที่ทำหน้าที่คนละอย่างกันจะมีค่า Socket Port ต่างกันด้วย หลังจากตั้งค่าของ Socket Port แล้วจะสามารถกดปุ่ม Listen ได้ ให้กดปุ่มนั้น หลังจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั้นก็จะถือว่าได้ตั้งค่า Server เรียบร้อยแล้ว หลังจากนั้นให้มาใช้งานจาก Client จากคอมพิวเตอร์อีกเครื่องซึ่งอยู่ในระบบเน็ตเวิร์กเดียวกัน



รูปที่ 8.2 รูปหน้าตา INTERFACE ของโปรแกรม CLIENT.EXE

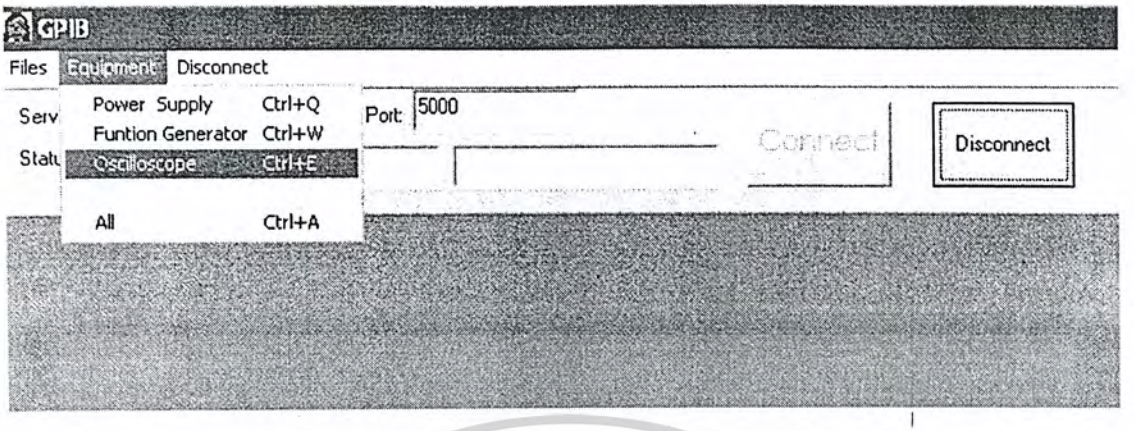
เมื่อโปรแกรม Client ถูกเปิดแล้วให้พิมพ์ค่าของ IP Address ของเครื่อง Server ที่ตำแหน่ง Server Address และทำการใส่ค่า Socket Port ให้ตรงกับ Socket Port ที่ได้ตั้งไว้ที่ Server และกด Connect เพื่อทำการติดต่อกับโปรแกรม Server ถ้าสามารถติดต่อได้ จะมี ไดอะล็อกบล็อก คำว่า "connect now" ขึ้นมาให้เห็น

หลังจากที่ Server และ Client สามารถติดต่อกันได้แล้วให้ทำการเลือกว่าเราจะสั่งงานอุปกรณ์ชนิดใด ซึ่งในที่นี้เราได้สั่งให้สามารถสั่งงานได้เพียง 3 ประเภท คือ

1. Power Supply รุ่น HM8142
2. Function Generator รุ่น HM8130
3. Analog/Digital Oscilloscope รุ่น HM1007

ซึ่งสามารถทำการเลือกได้ดังรูปที่ 8.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



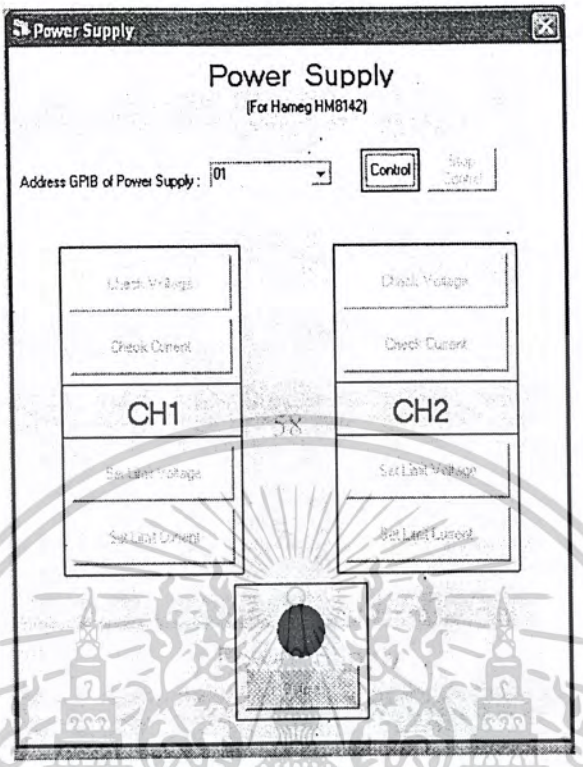
รูปที่ 8.3 รูปแสดงวิธีการเลือกอุปกรณ์เพื่อจะทำการควบคุม

8.1. การทดลองควบคุม Power Supply รุ่น HM8142

1. ทำการสั่งงานให้ Power Supply อยู่ในโหมด รีโมต ก่อน โดยการเลือก Address GPIB ให้ตรงกับอุปกรณ์หลังจากนั้นให้กดปุ่ม Control ที่โปรแกรมส่วนของ Power Supply

รูปที่ 8.4 สวิตช์ที่ใช้เซ็ท GPIB Address ของ Power Supply

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

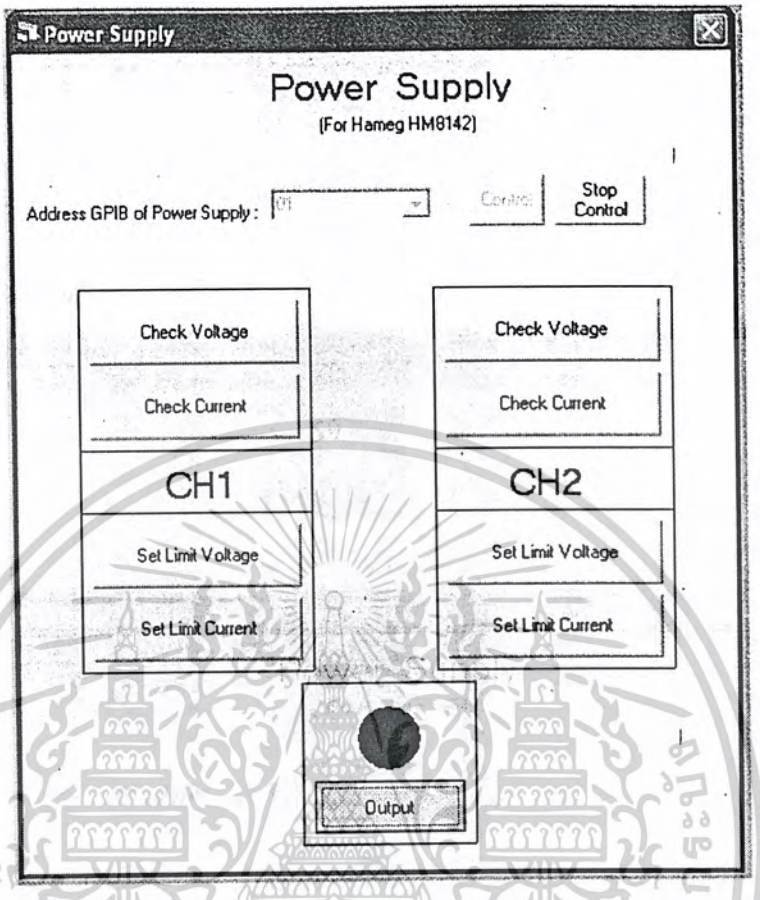


รูปที่ 8.5 หน้าตาโปรแกรมส่วนของ Power Supply ก่อนการกดปุ่ม Control

รูปที่ 8.6 Power Supply หลังจากที่อยู่ในโหมดรีโมต (สังเกตบริเวณที่วงไว้)

2. หลังจากที่ Power Supply ได้อยู่ในโหมดรีโมตแล้ว เราสามารถสั่งงานต่างๆ ได้ตามหน้า จอของโปรแกรมซึ่งมีดังนี้ Set limit Voltage ของ ch1, Set limit Current ของ ch1, Check limit Voltage ของ ch1, Check limit Current ของ ch1, Set limit Voltage ของ ch2, Set limit Current ของ ch2, Check limit Voltage ของ ch2, Check limit Current ของ ch2 และรวมถึง สั่งให้เปิด/ปิด Output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

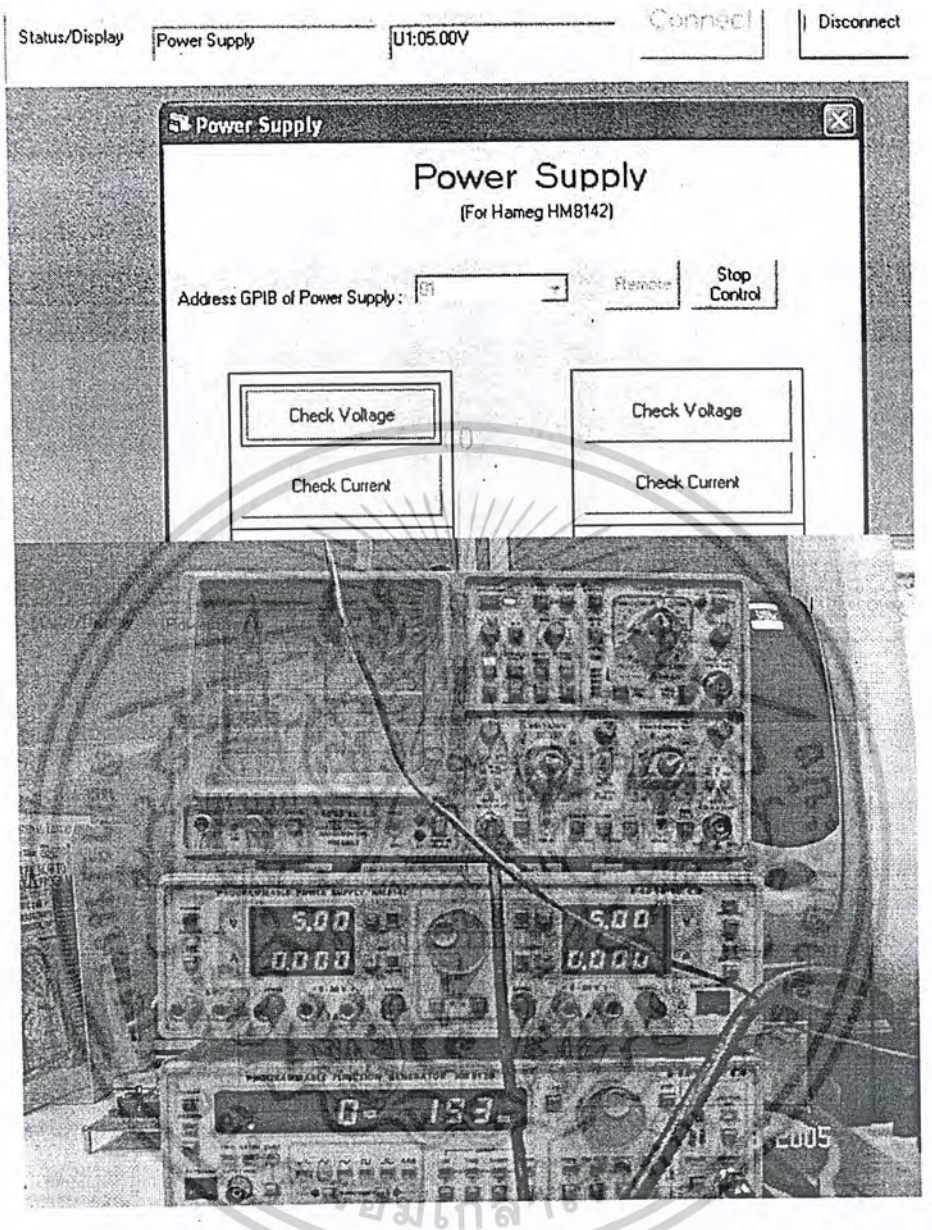


รูปที่ 8.7 หน้าตาของโปรแกรมส่วน Power Supply หลังจากกด Control แล้ว

3. จากนั้นเราลองใช้ทดสอบกดปุ่ม Set Limit Voltage ของ ch1 จะขึ้นหน้าต่างขึ้นมา 1 หน้าต่างให้เราใส่ค่า Voltage ที่ต้องการเซ็ทลงไป ผลที่ออกมาคือหน้าจอของอุปกรณ์ Power Supply ที่ส่วนของ ch1 จะเปลี่ยนเป็นค่าที่เราเซ็ททันที แต่ output จะยังไม่ออกจนกว่าเราจะกดปุ่ม Output และเมื่อเรากดปุ่ม Output วงกลมสีดำข้างบนปุ่มก็จะกลายเป็นสีแดงแสดงว่าอุปกรณ์ Power Supply มีการส่ง Output ออกไปยังขั้วบวกและขั้วลบของแต่ละ Channel แล้วให้เป็นไปตามค่าที่ได้ปรับไว้ ณ ก่อนที่วงกลมจะเป็นสีแดงและหลังจาก วงกลมเป็นสีแดงแล้ว (ขณะมี output ออกสู่ภายนอก) ก็สามารถเซ็ทเพื่อเปลี่ยนแปลงค่าต่างๆได้อีกด้วย

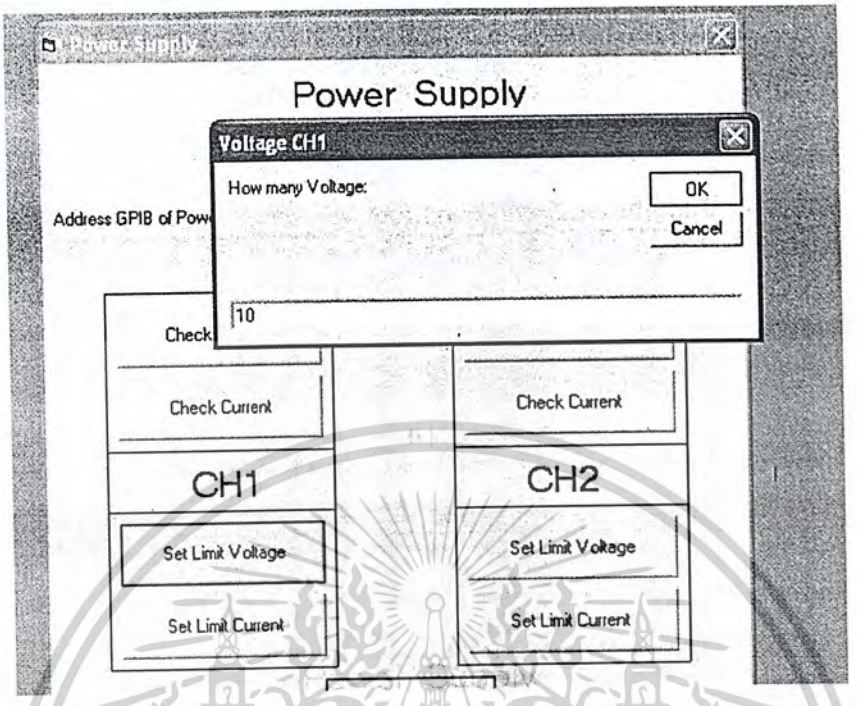
ส่วนการเซ็ทค่ากระแสนั้นจะเป็นการเซ็ทค่าลิมิตเท่านั้นและการCheck ค่ากระแสนั้นค่าที่อ่านกลับมาจะเป็นค่าที่ Power Supply จ่ายกระแส ณ ขณะนั้นเท่านั้น(ค่าที่อ่านกลับมาไม่ใช้ค่าที่เซ็ทลิมิตไว้) แสดงผลการทดลองได้ดังรูปที่ 8.8 และ 8.9และ 8.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

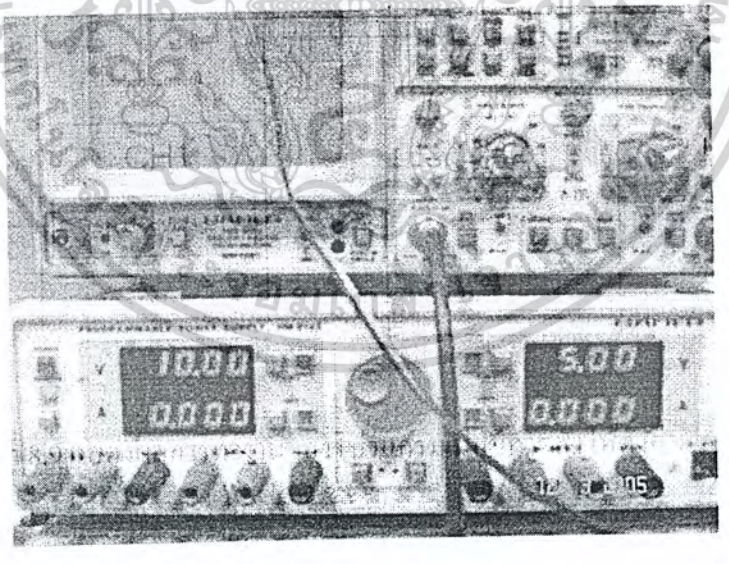


รูปที่ 8.8 แสดงผลการทำงานของโปรแกรมเมื่อมีการสั่งให้ตรวจสอบสถานะของแรงดันซึ่งจะแสดงผลที่ตำแหน่งของ Status/Display ด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.9 แสดงผลการทำงานของโปรแกรมเมื่อมีการสั่งให้ตั้งค่าแรงดัน

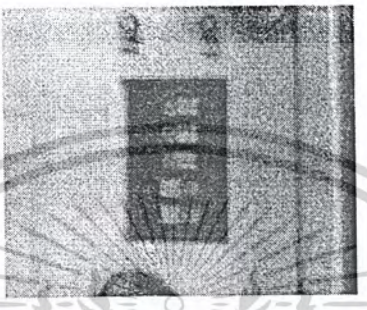


รูปที่ 8.10 ผลที่ได้หลังที่ได้สั่งงานจากโปรแกรมในรูปที่ 8.9

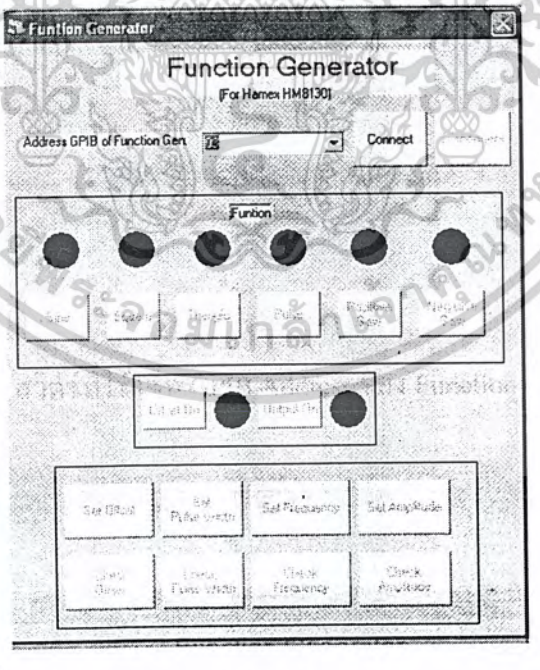
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.2. การทดลองควบคุม Function Generator รุ่น HM8130

1. สั่งงานให้Function Generatorอยู่ในโหมด รีโมตก่อนโดยการเลือก Address GPIB ให้ตรงกับอุปกรณ์หลังจากนั้นให้กดปุ่ม Connect ที่โปรแกรมส่วนของ Function Generator

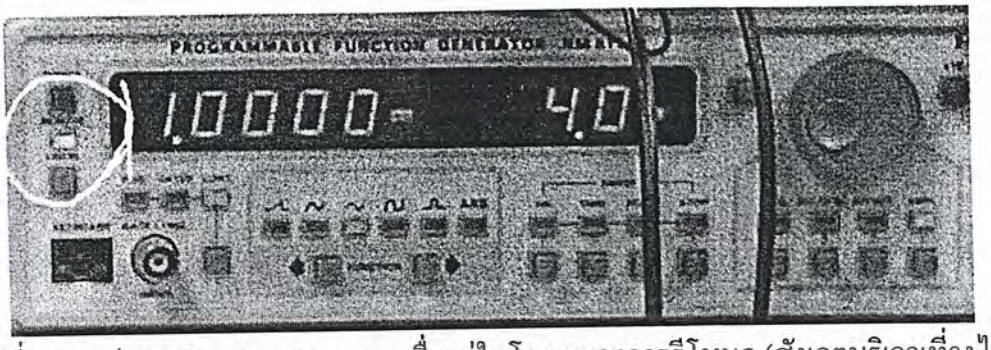


รูปที่ 8.11 สวิตช์ที่ใช้ เซ็ท GPIB Address ของ Function Generator



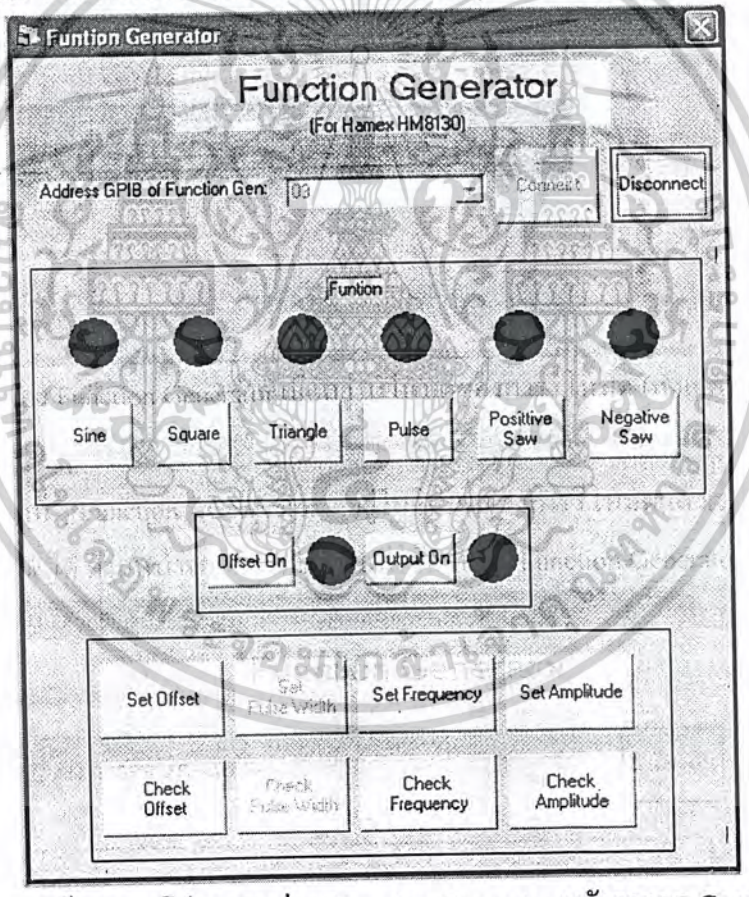
รูปที่ 8.12 หน้าตาโปรแกรมส่วนของ Function Generator ก่อนการกดปุ่ม Connect

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.13 รูปของ Function Generator เมื่ออยู่ในโหมดของการรีโหมด (สังเกตบริเวณที่วงไว้)

2.หลังจากที่ Function Generator อยู่ในโหมดของการรีโหมดแล้วเราก็สามารถสั่งงาน Function Generator ได้ ตามหน้าจอของโปรแกรมที่ใช้ควบคุม Function Generator

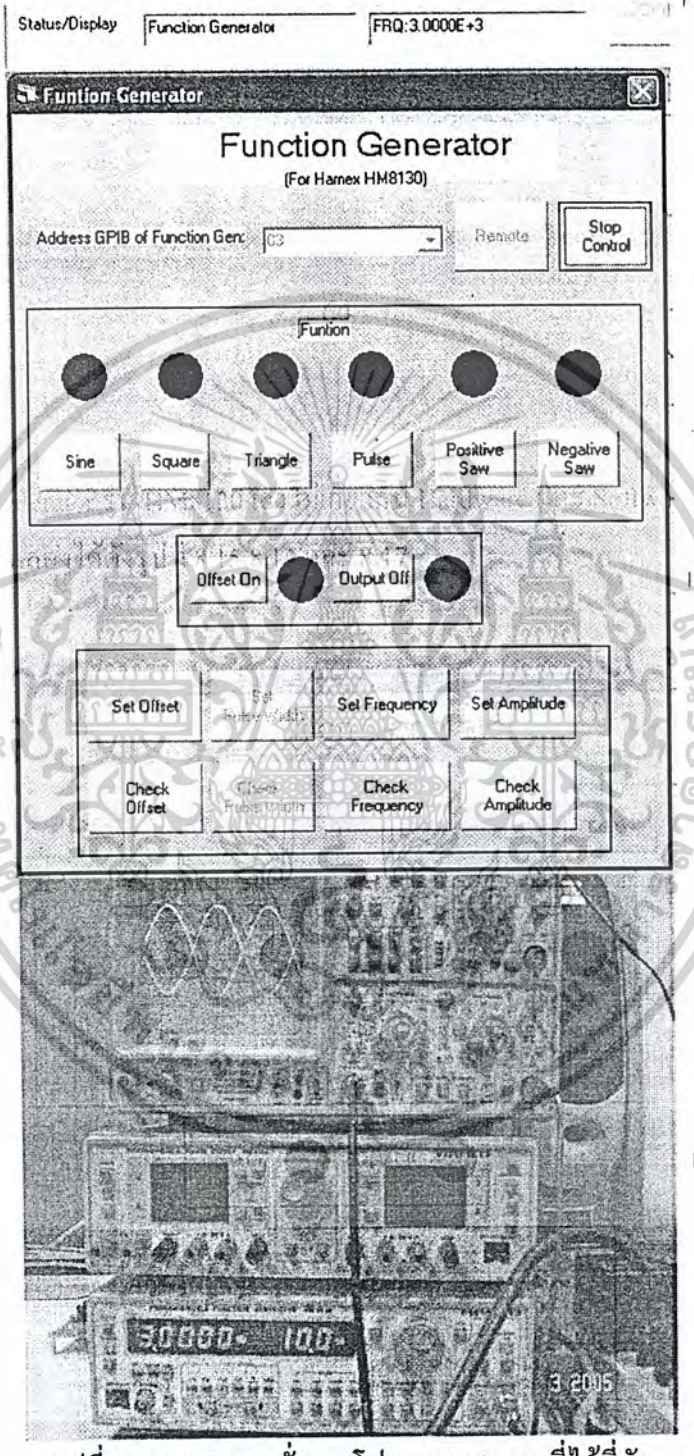


รูปที่ 8.14 หน้าตาของโปรแกรมส่วน Function Generator หลังจากกด Connect แล้ว

3. เมื่อต้องการเซ็ทอะไรก็เพียงกดปุ่มนั้นๆก็จะขึ้นหน้าต่างมาให้ใส่ค่าตัวเลขลงไปเหมือนกับเวลาเซ็ทค่าแรงดันและกระแสของ Power Supply ถ้าต้องการใช้รูปสัญญาณใดก็สามารถกดปุ่มนั้นๆได้เลยและถ้าต้องการให้ Function Generator ส่งสัญญาณออกสู่ภายนอกก็เพียงกดปุ่ม “Output On” หรือถ้าต้องการให้สัญญาณมี Offset ตามที่ตั้งไว้ก็เพียงกดปุ่ม “Offset On” เท่านั้น ส่วนคุณสม บัติต่างๆเช่นการรักษาระดับแรงดัน การรักษารูปร่างของสัญญาณ นั้นเป็นไปตามคุณสมบัติ

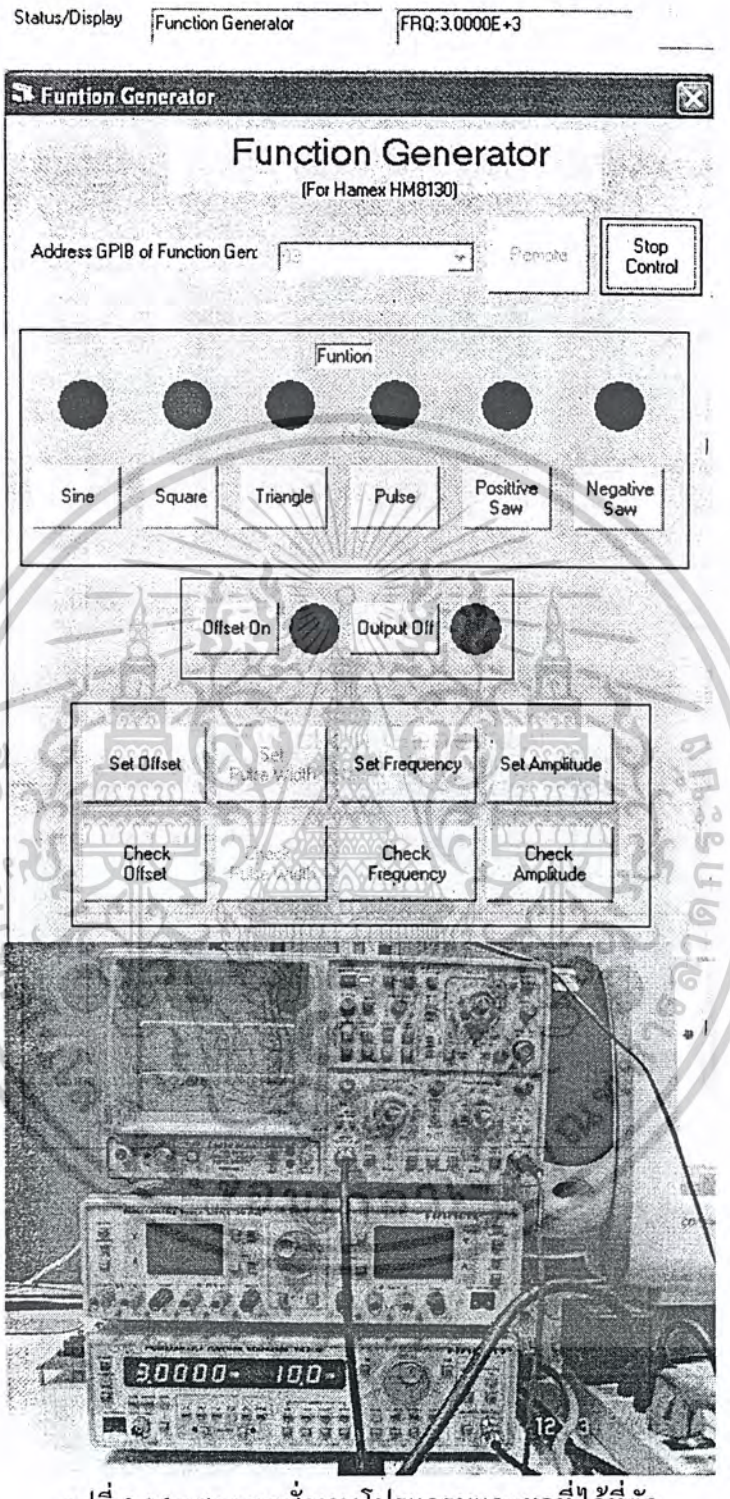
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของตัว Function Generator รุ่น HM8130 เองไม่เกี่ยวกับ Hardware และ Software ที่ใช้ในโครงงาน ชิ้นนี้แสดงผลการทดลองได้ดังรูปที่ 8.15,8.16 และ 8.17



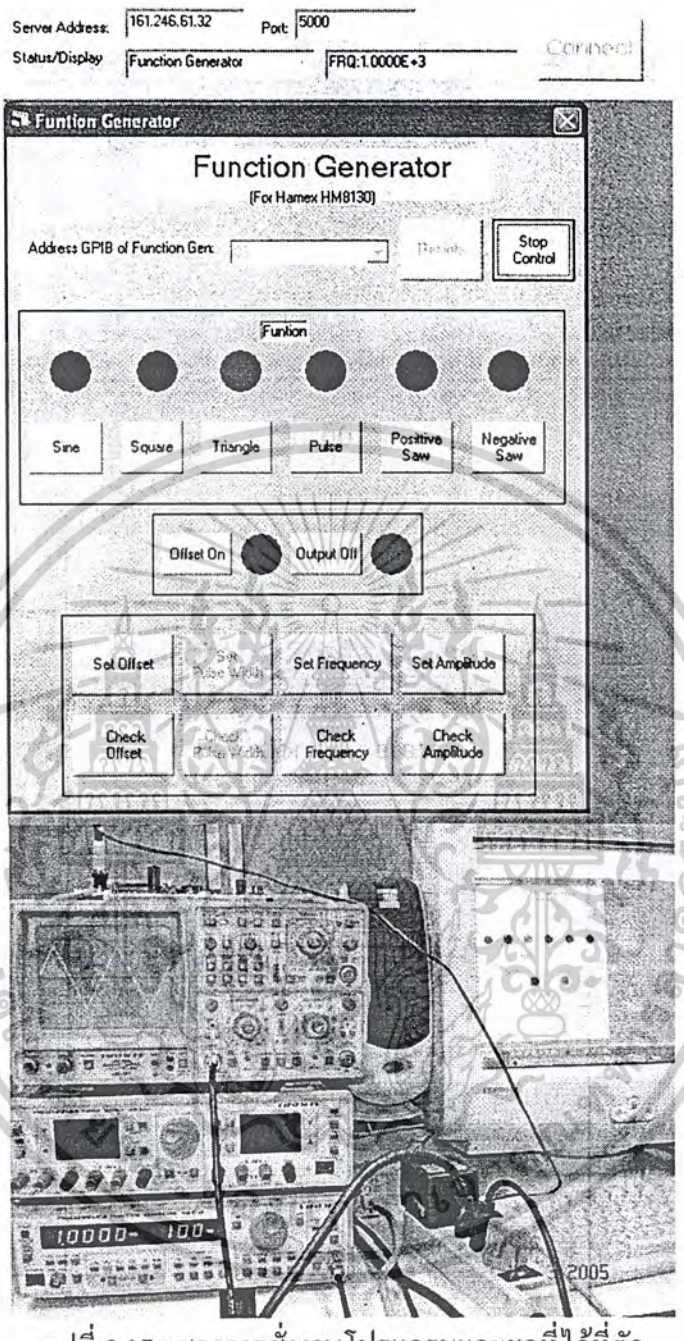
รูปที่ 8.15 แสดงการสั่งงานโปรแกรมและผลที่ได้ที่ตัว Function Gennerator หลังจากที่ทำการสั่งงานจากโปรแกรมแล้ว โดยสั่งงานให้เป็นคลื่น Sine ที่ความถี่ 3 kHz ค่า Amplitude 10V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.16 แสดงการสั่งงานโปรแกรมและผลที่ได้ที่ตัว Function Gennerator หลังจากที่ทำการสั่งงานจากโปรแกรมแล้ว โดยสั่งงานให้เป็นคลื่นสี่เหลี่ยม ที่ความถี่ 3 kHz ค่า Amplitude 10V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



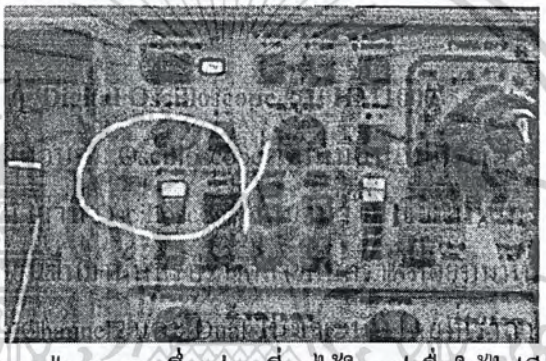
รูปที่ 8.17 แสดงการสั่งงานโปรแกรมและผลที่ได้ที่ตัว Function Gennerator หลังจากที่ทำการสั่งงานจากโปรแกรมแล้ว โดยสั่งงานให้เป็นคลื่นสามเหลี่ยม ที่ความถี่ 1 kHz ค่า Amplitude 10V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.3. การทดลองควบคุม Analog/Digital Oscilloscope รุ่น HM1007

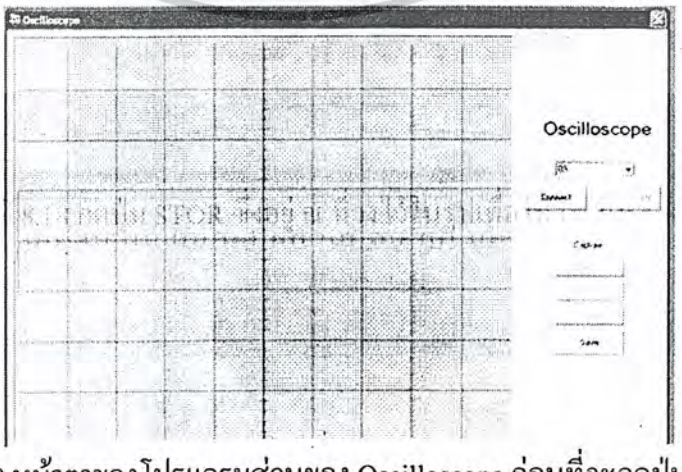
เราไม่สามารถสั่งงานควบคุม Oscilloscope ได้เหมือนอุปกรณ์ตัวอื่นที่นำมาทดลองแต่ Oscilloscope แต่เราสามารถนำค่าที่ Oscilloscope ส่งมานำไปเขียนเป็นกราฟเพื่อแสดงผลบนคอมพิวเตอร์ได้ และกราฟที่เขียนขึ้นมานั้นมีรูปร่างคล้ายกับกราฟที่อยู่บนหน้าจอ Oscilloscope ซึ่งสามารถอ่านได้ทั้ง Channel1 ,Channel2 และ Dual ในขณะนั้นโดยที่เราสามารถอ่านข้อมูลของ Channel2 ในขณะนั้นได้ทั้งๆที่หน้าจอ Oscilloscope ในขณะนั้นแสดงแค่ Channel1

1.สำหรับการรับค่าจาก Oscilloscope เข้ามาเพื่อประมวลผลเพื่อวาดกราฟในโปรแกรมของเรานั้นขั้นแรกเราต้องทำการกดปุ่ม Stor. บนหน้าปัทของ Oscilloscope ก่อน หลังจากนั้นก็เลือก GPIB Address บนโปรแกรมและกดปุ่ม Connect เพื่อติดต่อกับ Oscilloscope



รูปที่ 8.18 กดปุ่ม STOR ซึ่งอยู่ ณ ที่วงไว้ในรูปเพื่อให้ไฟสีเหลืองติด

รูปที่ 8.19 สวิตช์ใช้สำหรับ เซ็ทค่า GPIB Address ของ Oscilloscope

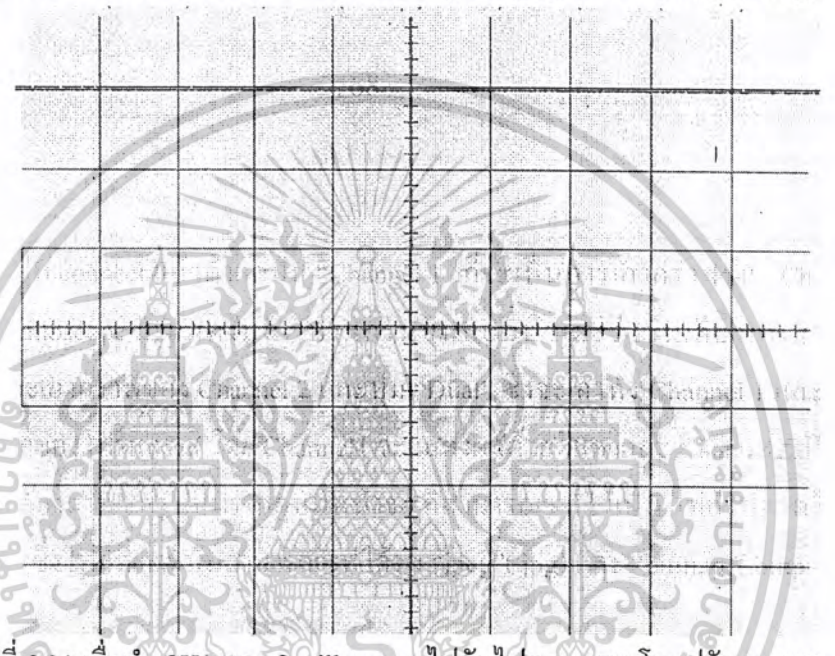


รูปที่ 8.20 หน้าตาของโปรแกรมส่วนของ Oscilloscope ก่อนที่จะกดปุ่ม connect

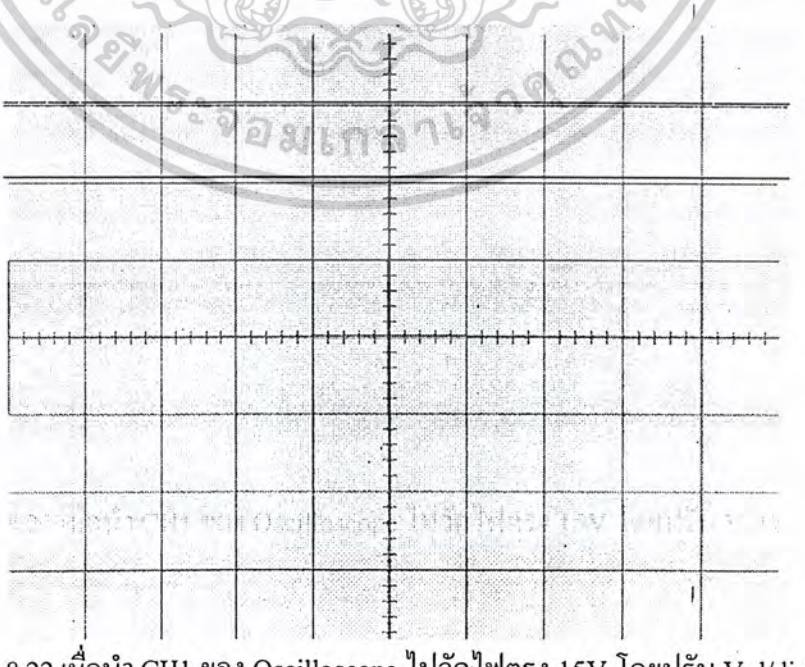
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.เมื่อกดปุ่ม connect แล้วถ้ากดปุ่ม “Channel 1 ก็จะเป็นการเอากราฟของ Channel 1 ณ ขณะนั้นของ Oscilloscope มาแสดงบน โปรแกรมบนคอมพิวเตอร์ เช่นเดียวกันสำหรับปุ่ม “Channeal 2 ซึ่งจะเอากราฟของ Channel 2 และปุ่ม “Dual” ซึ่งจะเอาทั้ง Channel 1 และ Channel 2 มาแสดงที่หน้าจอของโปรแกรม(โดย Channel1 จะแสดงผลด้วยสีแดงและ Channel2 เป็นสีดำ)

3. และถ้าต้องการเซฟรูปที่ปรากฏบนหน้าจอ ก็สามารถเซฟไว้เป็น ไฟล์รูปภาพได้โดยกดปุ่ม “Save” เท่านั้น ซึ่งไฟล์ที่เซฟ นั้นสามารถเซฟได้ทั้ง “jpeg”(.jpg) และ “bitmap”(.bmp)



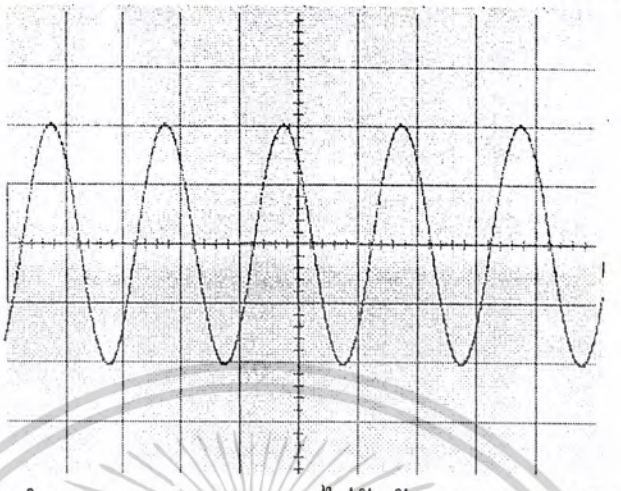
รูปที่ 8.21 เมื่อนำ CH1 ของ Oscilloscope ไปวัดไฟตรง 15V โดยปรับ Vol/div =5



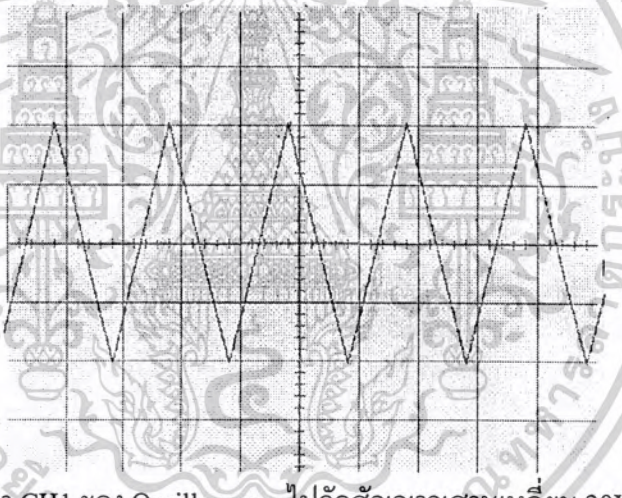
รูปที่ 8.22 เมื่อนำ CH1 ของ Oscilloscope ไปวัดไฟตรง 15V โดยปรับ Vol/div =5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ เมื่อนำ CH2 ของ Oscilloscope ไปวัดไฟตรง 10V โดยปรับ Vol/div =5

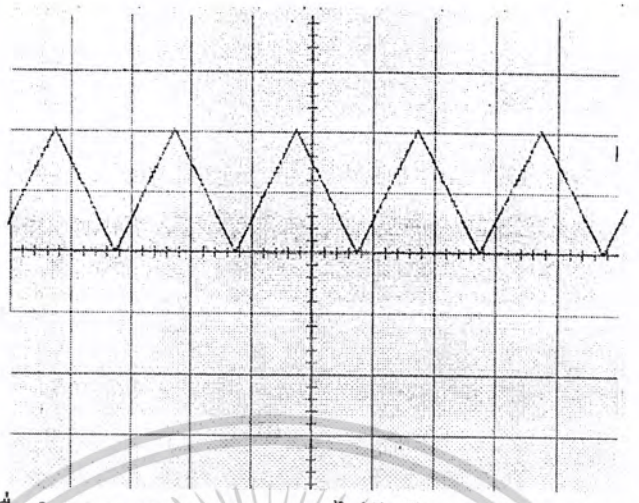


รูปที่ 8.23 เมื่อนำ CH1 ของ Oscilloscope ไปวัดสัญญาณ sine 20V Peak to Peak มีความถี่=1KHz โดยปรับ Vol/div = 5 และ Time/div = 0.5 ms

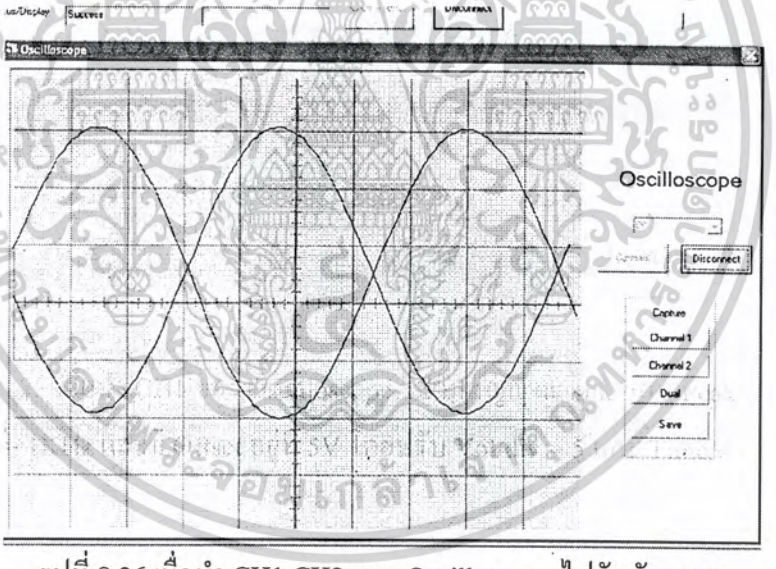


รูปที่ 8.24 เมื่อนำ CH1 ของ Oscilloscope ไปวัดสัญญาณสามเหลี่ยม 20V Peak to Peak มีความถี่=1KHz โดยปรับ Vol/div = 5 และ Time/div = 0.5 ms

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

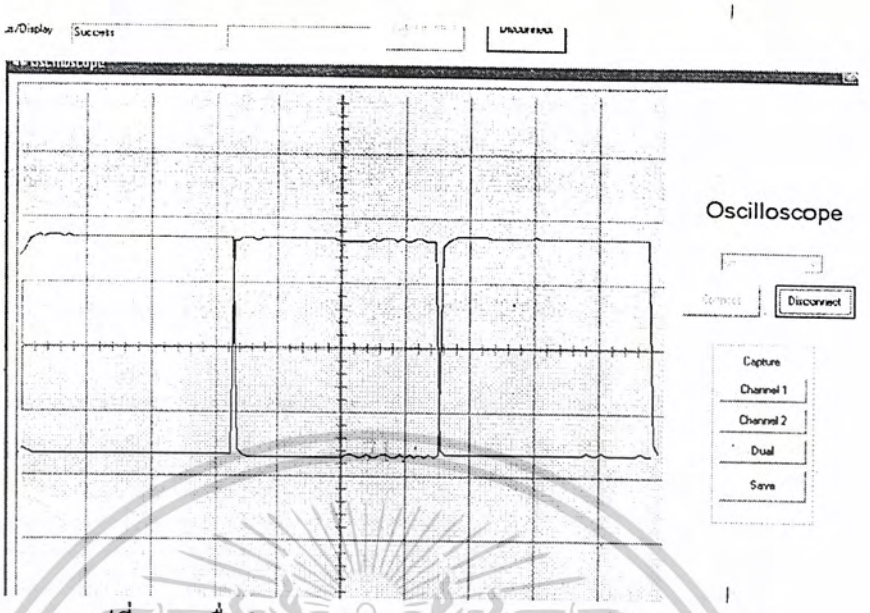


รูปที่ 8.25 เมื่อนำ CH1 ของ Oscilloscope ไปวัดสัญญาณ sine 10V Peak to Peak มีความถี่=1KHz และมี Offset อยู่ที่ 5V โดยปรับ Vol/div = 5 และ Time/div = 0.5 ms



รูปที่ 8.26 เมื่อนำ CH1,CH2 ของ Oscilloscope ไปวัดสัญญาณ sine 10V Peak to Peak มีความถี่=3KHz และมี Offset อยู่ที่ 2.5V โดยปรับ Vol/div = 2 และ Time/div = 50 us โดย ได้กดปุ่ม Inverse CH2 ไว้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.27 เมื่อนำ CH1,CH2 ของ Oscilloscope ไปวัดสัญญาณ
สี่เหลี่ยม 8V Peak to Peak มีความถี่=3KHz และมี Offset อยู่ที่ 0V
โดยปรับ Vol/div = 2 และ Time/div = 50 us โดย ได้กดปุ่ม Inverse CH2 ไว้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9

สรุปและข้อเสนอแนะในการนำไปพัฒนาต่อ

สรุป

-สำหรับ Power Supply และ Function Generator นั้นเราสามารถควบคุมและตรวจสอบผลที่เกิดขึ้นกับอุปกรณ์ได้ แต่สำหรับ Oscilloscope นั้นเราไม่สามารถควบคุมได้เนื่องจากไม่มีชุดคำสั่งสำหรับควบคุม มีเพียงแต่นำค่าต่างๆ ที่อยู่บนหน้าจอของ Oscilloscope มาแสดงเป็นเส้นกราฟบนหน้าจอคอมพิวเตอร์เท่านั้น

-สำหรับผลการทดลองนั้นเมื่อเราสามารถสั่งควบคุมอุปกรณ์ตัวนั้นๆได้แล้ว การกระทำต่างๆ เช่น การรักษาระดับแรงดัน การสร้างคลื่นสัญญาณ รูปต่างๆนั้นขึ้นอยู่กับความ สามารถของอุปกรณ์นั้นๆ ดังนั้นเราจึงไม่ได้เน้นมาก ผลการทดลองที่นำมาจึงเน้นไปทางการประยุกต์ ใช้งานมากกว่า

ปัญหาที่พบ

1. การส่งข้อมูลระหว่างเครือข่าย ยังมีความถูกต้องไม่สมบูรณ์ ยังมีเปอร์เซ็นต์ที่ข้อมูลอาจเกิดการผิดพลาดเนื่องจากระบบเน็ตเวิร์กได้ถึงแม้จะมีโอกาสน้อยก็ตาม

2. ในการส่งข้อมูลบางครั้งถ้าระบบเนตเวิร์กมีข้อมูลเต็ม(Busy) อาจจะเกิดการล้าช้าของการส่งอย่างมากบางครั้งอาจจะทำให้ข้อมูลในบางส่วนเกิดการสูญหาย

ข้อเสนอแนะสำหรับการนำไปพัฒนาต่อ

1. ทำให้มีเสถียรภาพมากขึ้น เนื่องจากบอร์ดที่ใช้อยู่นี้ยังไม่เสถียรมาก
2. พัฒนาให้ใช้งานกับอุปกรณ์ได้หลากหลายชนิดมากขึ้นและหลากหลายรุ่นมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1.กนก เจนจิระพงศ์เวช, "GPIB (IEEE-488)บัสอินเตอร์เฟส มาตรฐานและการใช้",วารสารเซมิคอนดัคเตอร์,ฉบับที่ 78(พฤษภาคม, สิงหาคม, กันยายน 2530)

2.นพดล มณีรัตน์, " การออกแบบและการส่งข้อมูลแบบขนาน สำหรับการควบคุมเครื่องมือวัดผ่านระบบมาตรฐาน"วิทยานิพนธ์วิศวกรรมศาตรมหาบัณฑิต คณะวิศวกรรมศาตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2540

3.สัลยุทธ์ สว่างวรรณ, "เครือข่ายคอมพิวเตอร์",เอช.เอ็น.กรุ๊ป จำกัด, 2547,หน้า40-44

ขอขอบคุณ

1. http://oldlook.hameg.de/en/index.htm (Website ของ Hameg)
2. http://thaiio.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้