GPIB(1EEE488) บัสอินเตอร์เฟสมาตรฐานและการใช้งาน

GPIB(1EEE488) STANDARD BUS INTERFACE AND APPLICATION

โดย

นายจิระศักดิ์ เอี๋ยวสกุล

นายธาราพงษ์ รุจิวงศ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมไฟฟ้า

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2534

008499

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2534

ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

เรื่อง GPIB(IEEE488) บัสอินเตอร์เฟสมาตรฐานและการใช้งาน

GPIB(IEEE488) Standard bus interface and application

จัดทำโดย

นาย จิระศักดิ์ เอี่ยวสกุล 32.1453 ; 3w

นาย ชาราพงษ์ รุจิวงศ์ 32.1460 ; 3w

.....อาจารย์ที่ปรึกษา

(อาจารย์ ศุภกิจ จูตะวิริยะ)

.....อาจารย์ที่ปรึกษา

(อาจารย์ วิจิตร กิณเรศ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GPIB[IEEE488]
บัสอินเตอร์เฟสมาตรฐานและการใช้งาน

นาย จิระศักดิ์ เอี๋ยวสกุล
นาย ธาราพงษ์ รุจิวงศ์

อาจารย์ที่ปรึกษา
อาจารย์ ศุภกิจ จูตะวิริยะ
อาจารย์ วิจิตร กิณเรศ

ปีการศึกษา 2534

บทคัดย่อ

ในการติดต่อส่งข้อมูลระหว่างอุปกรณ์นั้น เรามักจะคุ้นเคยกับระบบการเชื่อมโยงแบบอนุกรมหรือแบบขนาน แต่ปัจจุบันนี้เราพบว่าระบบบัสเชื่อมโยงเอนกประสงค์ (General Purpose - Interface Bus) เป็นระบบที่นิยมใช้กันมากในอุปกรณ์ทางไฟฟ้าหรือเครื่องมือวัดต่างๆทางอิเล็กทรอนิกส์ เช่น Power Supply , Pulse Generator and Digital Multimeter เป็นต้น แต่ส่วนใหญ่แล้วในบ้านเราจะไม่ค่อยคุ้นเคยกับระบบนี้ดี ประกอบกับราคาของการ์ด(Card) ที่จะต้องใช้ในการอินเทอร์เฟส(Interfaing) กับระบบนั้นมีราคาค่อนข้างสูง ฉะนั้นจึงไม่ค่อยมีใครนำ GPIB มาใช้ให้เกิดประโยชน์ ดังนั้นโครงงานนี้จึงถูกนำเสนอขึ้นมาเพื่อศึกษาถึงระบบการเชื่อมโยงต่อข้อมูลและการนำไปใช้งาน

เนื่องจากโครงงานนี้เราต้องการสร้างโดยให้ราคาต้นทุนต่ำ และพยายามออกแบบให้ใช้อุปกรณ์ที่มีขายในท้องตลาดบ้านเรา เช่น IC#8255 ที่ใช้เป็นพอร์ทควบคุมหรือ IC#74LS133 เป็นต้น เพื่อว่าผู้ที่สนใจโครงงานนี้สามารถนำไปสร้างเพื่อใช้งานได้เอง ดังนั้นราคาของการ์ดที่เราสร้างขึ้นมานี้จึงมีราคาถูกกว่าราคาที่มาจากบริษัทผู้ผลิตมากมายหลายเท่า ในขณะที่ประสิทธิภาพใกล้เคียงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GPIB[IEEE488]

Standard Bus Interface and Application

JIRASAK IEOSAKUN
THARAPONG RUJIWONG
ADVISOR
SUPAKIT JUTAVIRIYA
WIJIT KINNARET
ACADEMIC YEAR 2534

ABSTRACT

In communication System, almost we must to say to "Serial communication System and parallel communication System". For the time being, we found that GPIB(General Purpose Interface Bus) is favourally recieved for used in electrical equipment or Instrument such as, Power Supply, Pulse generator and Digital multimeter etc. Because it has high speed for access data and very convenience for used.

In Vice-Versa, The GPIB is not populerly used in Thailand. Because the GPIB card is very high price. Therefore, We just can't get used to it and isn't taken to utilized. Consequently, the project is presented for communication system study and application.

Due to, We want to build the GPIB card with low cost and We try to design the hardware with component that can be available in local.(For example, IC#8255 Port or Ic#741s133)

For the purpose that, Anyone who Interested in this project can be build it by himself for application. Therefore, the cost of GPIB card is very low than manufacturing Price(decrease by 500%) while the efficiency are the same.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้าที่

บทคัดย่อ	ก
ABSTRACT	ข
สารบัญ	ค
บทที่ 1 บทนำ	1
บทที่ 2 สัญญาณต่างๆในระบบบัสแบบขนาน	3
- โครงสร้างของ GPIB และ อุปกรณ์	5
- คุณลักษณะทางไฟฟ้า	6
- สัญญาณควบคุมที่สำคัญ	7
- ข้อควรระวังในการใช้ GPIB	11
- คำสั่งใช้งานของ GPIB	17
บทที่ 3 แผ่นวงจรพิมพ์ของโครงงาน	23
- ส่วนประกอบของแผ่นวงจรพิมพ์	23
- การออกแบบวงจรบัฟเฟอร์	24
- การออกแบบพอร์ทควบคุม	26
- การจัดกลุ่มสัญญาณ	27
- การออกแบบวงจรถอดรหัส	28
บทที่ 4 โปรแกรมสนับสนุนของแผ่นวงจรพิมพ์	33
- ส่วนประกอบของโปรแกรมจัดการระบบ	34
- ส่วนของโปรแกรมผู้ใช้งาน	37
- Flow Chart ของการทำงาน	38
บทที่ 5 วิธีการใช้โปรแกรม GPIB4.EXE	56
- GPIB MENU	57
- FILE MENU	67
- OPTION MENU	74
- QUIT MENU	76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ) หน้าที่

บทที่ 6 การทดลองและผลการทดลอง 77

บทที่ 7 บทวิจารณ์และสรุป 78

รายละเอียดโปรแกรม 79

- USER.PAS 79-1
- MENU.PAS 79-2
- GPIB4.PAS 79-25
- IEEE488.PAS 79-46

ภาคผนวก 80

กิตติกรรมประกาศ 81

บรรณานุกรม 82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ระบบบัสเชื่อมต่อข้อมูลเอนกประสงค์ เป็นระบบบัสมาตรฐานที่ใช้ในการติดต่อรับส่งข้อมูลระหว่างอุปกรณ์ มาตรฐานนี้ได้ผ่านการรับรองจากสถาบันวิศวกรไฟฟ้า และอิเล็กทรอนิกส์ (Institute of Electrical and Electronics Engineers : IEEE) แห่งสหรัฐอเมริกา และถูกจัดเข้าเป็นลำดับที่ IEEE-488 ในปี ค.ศ. 1975 และได้มีการปรับปรุงมาตรฐานให้ดียิ่งขึ้นในภายหลังเป็น IEEE-488 (1978) ซึ่งได้ใช้มาจนถึงปัจจุบันนี้ แต่เนื่องจากในคุณสมบัติของ IEEE-488 นี้มีความหมายถึงบัสเชื่อมต่อข้อมูลที่ใช้งานได้ทั่วไปจึงนิยมใช้คำว่า GPIB (General Purpose Interface Bus)

ประวัติความเป็นมา IEEE-488

บริษัทต่างๆ ที่เป็นผู้ผลิตเครื่องมือวัดในอเมริกา จึงได้ตกลงร่วมกันจัดหาระบบอินเทอร์เฟสมาตรฐานสำหรับที่จะใช้ร่วมกันขึ้นมา และในประเทศเยอรมันก็มีการพิจารณาระบบบัสมาตรฐานเช่นกัน โดยความร่วมมือช่วยเหลือของ IEC (International Electronic Commission) จนในปี 1972 อเมริกาโดย IEEE จึงได้มีการประชุมร่วมมือกันกับ IEC วางแผนพิจารณาระบบบัสมาตรฐานร่วมกัน

บริษัทฮิวเล็ทแพกการ์ด ผู้ผลิตเครื่องมือวัดรายใหญ่รายหนึ่งในอเมริกา ได้ทำการพัฒนาระบบบัสมาตรฐานของตัวเองอยู่ก่อนแล้วชื่อว่า HPIB (Hewlett Packard Interface Bus) จึงได้เสนอระบบบัสของตัวเองให้ IEEE พิจารณาและได้รับการยอมรับในที่สุดในปี 1975 เรียกว่ามาตรฐาน IEEE Std 488-1978 ระบบบัสนี้ก็คือ IEEE-488 ที่กล่าวถึงนั่นเอง สำหรับของ IEC ก็ได้มีการกำหนดมาตรฐานขึ้นมาอีกนัยหนึ่งเรียกว่า IEC 625-1 ตั้งใจที่จะให้เป็นมาตรฐานสากล โดยมีรายละเอียดทางเทคนิคทุกประการเหมือนกับ IEEE Std 488-1978 เพียงแต่ว่าการวางตำแหน่งของสัญญาณต่างๆ ในขั้วต่อต่างกันเล็กน้อยเท่านั้นเอง คำว่า GPIB จึงหมายรวมถึงทั้ง 2 มาตรฐานดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดประสงค์ของบัสก็คือ การส่งข่าวสารข้อมูลถ่ายทอดกันระหว่างเครื่องข่าวสารที่ว่านี้ก็มีทั้งข้อมูลที่เป็นผลลัพธ์จากการวัดค่าหรือจากกระบวนการต่างๆ และรวมถึงคำสั่งที่ใช้กำหนดสภาวะการทำงานของอุปกรณ์ต่างๆในระบบ

เมื่อมีการต่อเครื่องมือวัดหรืออุปกรณ์ที่ต้องการใช้ร่วมกันเข้ากับระบบบัสแล้ว จำเป็นต้องมีแบบแผนในการทำงานร่วมกันให้สอดคล้องกัน จึงต้องมีสัญญาณสำหรับควบคุมระบบ นอกเหนือจากข่าวสารข้อมูลที่ต้องมีการติดต่อกัน (สัญญาณควบคุมนี้ ใช้สำหรับควบคุมบัส และจัดระบบสำหรับการอินเทอร์เฟส อาจมองได้ว่าเป็นความเกี่ยวข้องทางฮาร์ดแวร์ไม่เกี่ยวข้องกับผู้ใช้งานภายนอก ส่วนคำสั่งสำหรับกำหนดการทำงานของอุปกรณ์ ในระบบเป็นงานทางด้านซอฟต์แวร์ที่เกี่ยวข้องกับการใช้งานโดยตรง)

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

สัญญาณต่างๆในระบบบัสแบบขนาน

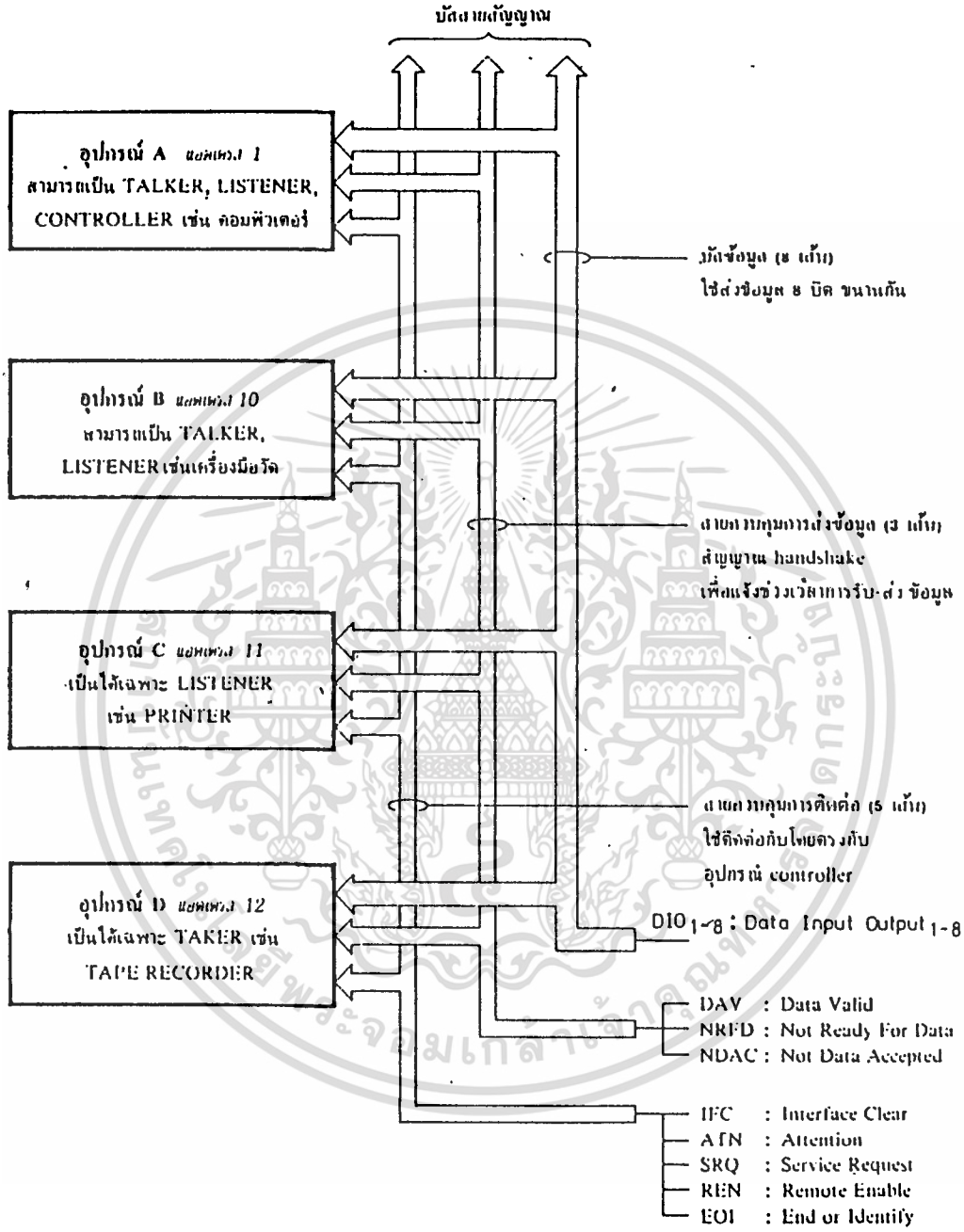
ที่มาของ GPIB นั้นจริงๆแล้วเกิดจากการรับรองของวิศวกรไฟฟ้าและอิเล็กทรอนิกส์ของอเมริกา (Institute of Electrical and Electronic Engineers : IEEE) ซึ่งยอมรับเข้าเป็นลำดับที่ IEEE-488 แต่เนื่องจากในคุณลักษณะสมบัติของ IEEE-488 นี้ มีความหมายถึงบัสอินเทอร์เฟสที่ใช้งานได้ทั่วไปจึงมีผู้นิยมใช้คำว่า GPIB

GPIB มีชื่อเต็มว่า General Purpose Interface Bus เป็นบัสมาตรฐานที่ใช้ในการติดต่อรับส่งข้อมูลระหว่างอุปกรณ์ได้หลายเครื่อง โดยมีข้อพิเศษสำหรับผู้ใช้ก็คือในกรณีที่ต้องการขยายอุปกรณ์เพิ่มเติมเข้ามาในระบบผู้ใช้ไม่จำเป็นต้องเพิ่มเติมส่วนวงจร หรืออุปกรณ์อื่นๆอีกเลย เพียงแต่ผู้ใช้เพิ่มสายเคเบิ้ลเชื่อมต่อเข้ามาขนานกับสายเคเบิ้ลหรือขั้วต่อเดิมเท่านั้น โดยใช้การแก้ไขเฉพาะส่วนของซอฟแวร์เท่านั้น

ในระบบที่ไม่ใหญ่โตนัก GPIB นั้นก็สามารถต่อเข้ากับอุปกรณ์หรือเครื่องมืออื่นๆได้สูงสุด 15 เครื่องโดยใช้สายสัญญาณเพียง 1 เส้นต่อขนานกันไปเรื่อยๆ ดังนั้นบัสสัญญาณของ GPIB จึงเป็นบัสแบบขนานโดยมีสัญญาณควบคุมร่วมด้วยเพื่อควบคุมทิศทางและเลือกตัวที่ต้องการติดต่อ

หากจะเปรียบเทียบกับมาตรฐานการรับ-ส่งสัญญาณของ RS-232C หรือ Centronics Interface. แล้ว GPIB มีข้อยุ่งยากและซับซ้อนกว่าในแง่การใช้งานเพราะต้องมีการใช้คำสั่งควบคุมอุปกรณ์แต่ละตัวก่อนที่จะรับ-ส่งข้อมูลกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 แผนผังแสดงอุปกรณ์ GPIB และการต่อสายสัญญาณต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของ GPIB

ส่วนประกอบพื้นฐานของ GPIB แสดงไว้ในรูปที่ 2.1 กล่าวคือ GPIB จะประกอบไปด้วย ผู้ส่ง(talker),ผู้รับ(listener) และผู้ควบคุม(controller)

Talker ทำหน้าที่ในการส่งข้อมูลโดยสามารถที่จะนำ talker จำนวนมากๆใส่ไว้ในระบบแต่จะมี talker เพียงตัวเดียวเท่านั้นที่กำลังทำงานอยู่

Listener ทำหน้าที่รับข้อมูล listener ก็เช่นเดียวกับ talker คือสามารถนำไปใส่ไว้ในระบบได้จำนวนมาก ดังนั้นจึงมีระบบที่ประกอบด้วย talker 1 ตัวและ listener หลายตัว

Controller เป็นตัวควบคุมสัญญาณต่างๆบนบัส โดยรับความต้องการของอุปกรณ์ที่จะส่งข้อมูล หรือกำหนด listener ทำการรับข้อมูล

อุปกรณ์ที่มี GPIB

อุปกรณ์หรือเครื่องมือที่มี GPIB นั้น แบ่งตามหน้าที่การทำงานได้ดังนี้

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 เช่น คอมพิวเตอร์ที่ทำหน้าที่ควบคุมระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

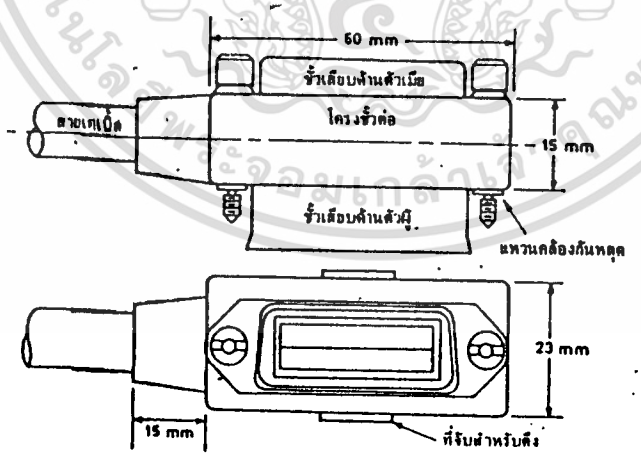
คุณลักษณะทางไฟฟ้าของ GPIB

คุณลักษณะทางไฟฟ้าซึ่งจะเป็นตัวกำหนดขีดจำกัดของ GPIB นั้นมีดังนี้

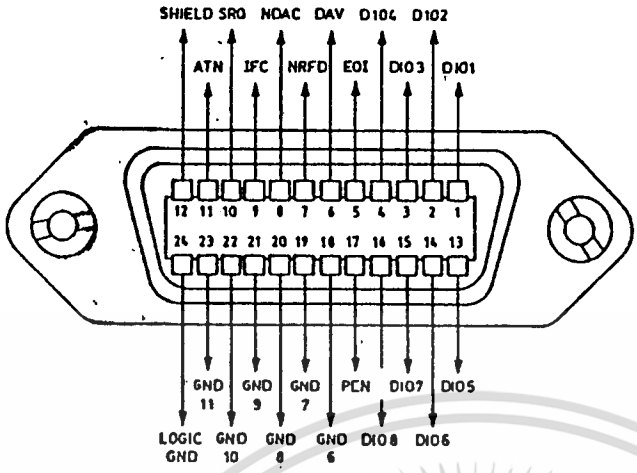
1. จำนวนอุปกรณ์ (talker,listener,controller) ที่ต่ออยู่กับสายสัญญาณ(bus) 1 เส้น จะต้องไม่เกิน 15 เครื่อง
2. สายเคเบิ้ลที่ใช้ต่อระหว่างอุปกรณ์แต่ละตัวจะต้องยาวไม่เกิน 4 เมตร และความยาวรวมของสายเคเบิ้ลทั้งหมดต้องไม่เกิน 20 เมตร
3. ความเร็วในการส่งข้อมูลต้องไม่เกิน 1 Mb/Sec(1 ล้านไบท์ต่อวินาที)
4. จำนวนของอุปกรณ์หรือเครื่องมือมากกว่าครึ่งหนึ่งต้องเปิดให้ทำงาน(จ่ายไฟ)

ขั้วต่อ GPIB

ขั้วต่อมาตรฐานของ GPIB เป็นแบบ Amphenal ขนาด 24 ขา และมีการจัดตำแหน่งสัญญาณที่ขาต่างๆดังรูปที่ 2.2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ขั้วต่อของ GPIB และการจัดขาของสัญญาณต่างๆ

สัญญาณควบคุมที่สำคัญ

สัญญาณ SQR และ ATN มีรายละเอียดดังนี้

SRQ (Serial Service Request)

เป็นสัญญาณอินพุตที่ป้อนให้กับคอมพิวเตอร์ ซึ่งอุปกรณ์ทุกตัวสามารถดึงให้สัญญาณเป็นลอจิก "ต่ำ" เพื่อแจ้งให้คอมพิวเตอร์ทราบว่า อุปกรณ์นั้นต้องการขอบริการ

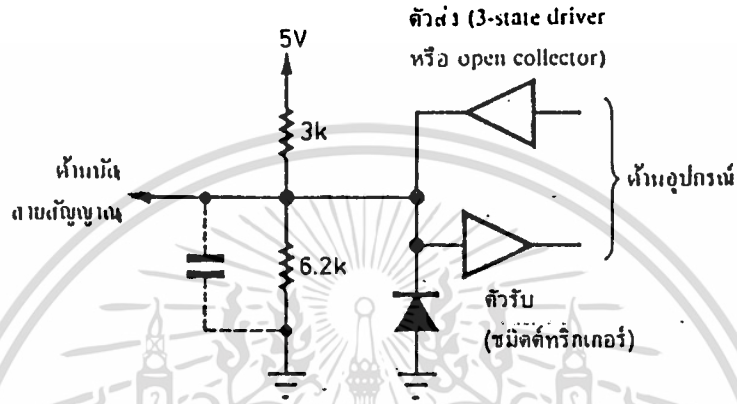
ATN (Attention)

ใช้สำหรับการเริ่มต้นการส่งคำสั่ง โดยคอมพิวเตอร์จะทำให้สัญญาณนี้เป็นลอจิก "ต่ำ" เพื่อกำหนดให้อุปกรณ์ทุกตัวที่ต่ออยู่กับสายสัญญาณทำตัวเป็น listener แล้วคอมพิวเตอร์จึงส่งสัญญาณแอดเดรสเพื่อกำหนดตำแหน่งของอุปกรณ์แต่ละตัว ถ้าหากอุปกรณ์ตัวที่ถูกเลือกไม่สามารถส่งสัญญาณกลับไปให้คอมพิวเตอร์ภายในช่วงเวลาที่กำหนด ตัวคอมพิวเตอร์จะสมมติว่าไม่มีอุปกรณ์ที่ตำแหน่งแอดเดรสนั้นอยู่ แล้วคอมพิวเตอร์จะแจ้งให้ผู้ใช้ทราบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

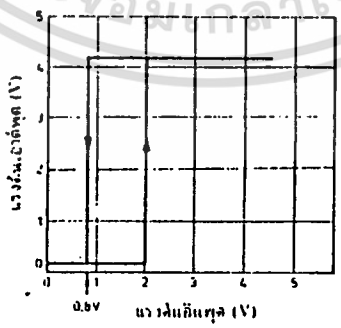
พิจารณาถึงวงจรไฟฟ้า

เมื่อพิจารณาถึงวงจรไฟฟ้าเมื่อมองย้อนเข้าไปยังจุดต่อสัญญาณบัสข้อมูล จะเห็นโหลดของ GPIB เป็นตัวต้านทานแบ่งแรงดัน 2 ตัวคือค่า 6.2k และ 3k ดังรูปที่ 2.3



รูปที่ 2.3 วงจรภายในจุดต่อบัสข้อมูลที่ใช้รับ-ส่งข้อมูล

ทางด้านเกตตัวส่ง(driver) เป็นแบบ open collector หรือแบบเกต 3 สถานะ (Tri-state driver) โดยการขับด้วยกระแสขนาด 48 mA ส่วนทางด้านเกตตัวรับ (receiver) ใช้แบบชมิตทริกเกอร์ (Schmitt trigger) เพื่อกำจัดสัญญาณรบกวน โดยมีคุณสมบัติการให้สัญญาณเอาท์พุต ดังรูปที่ 2.4



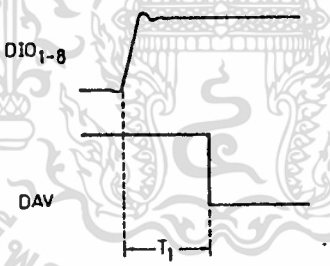
รูปที่ 2.4 คุณสมบัติฮิสเตอริซีสของตัว receiver แบบ Schmit trigger

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.4 ที่ระดับแรงดันอินพุทมีค่าน้อยกว่า 0.8v เอาท์พุทจะมีค่าต่ำด้วย แต่ถ้าเพิ่มแรงดันอินพุทขึ้นไปเรื่อยๆ จนมีค่ามากกว่า 2 v เอาท์พุทก็ยังคงมีค่าสูงอยู่แรงดันอินพุตต้องลดลงจนกระทั่งต่ำกว่า 0.8 v เอาท์พุทจึงกลับมามีค่าต่ำ ลักษณะเช่นนี้เป็นลักษณะของ " อิสเตอริซิส"

แต่อย่างไรก็ตามตัวเกต driver/receiver ที่ใช้กับ GPIB และมีจำหน่ายอยู่นั้น อาจมีคุณสมบัติแตกต่างไปจากที่กล่าวมาก็ได้ เช่นกรณีที่ใช้เป็นแบบอุปกรณ์แยกชิ้น(discrete เช่นทรานซิสเตอร์ทำหน้าที่เป็นสวิตช์) แล้วต่อปลาย (โหลด) ด้วยตัวต้านทานเมื่อปิดแหล่งจ่ายไฟจะทำให้แหล่งจ่ายไฟนั้นมีค่าเป็น 0 v.(ลัดวงจร) แต่ที่สัญญาณบัสข้อมูลยังมีตัวต้านทาน 6.2k และ 3k ต่อขนานกันอยู่ ดังนั้นจึงทำให้มีค่าโหลดประมาณ 2k ต่ออยู่สัญญาณบัสข้อมูลกับกราวด์ ซึ่งจะเป็นผลเสียทางช่วงของสัญญาณรบกวน(noise margin)

กรณีที่ใช้เกต driver/receiver เฉพาะของ GPIB เช่น ไอซีเบอร์ SN75160, MC6488A เป็นต้น ซึ่งเป็นโหลดประเภทอินดัคตีฟ เมื่อปิดแหล่งจ่ายไฟ จะทำให้ไม่มีโหลดเข้าไปเกี่ยวพันด้วยทำให้เกิดผลดีทางด้าน noise margin คือข้อมูลไม่ผิดพลาดได้ง่าย



รูปที่ 2.5 แสดงการเกิดสัญญาณ DIO_{1}-DIO_{8} และ DAV โดยมีค่าเวลาหน่วง T_{1}

008499
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนั้นความเชื่อถือได้ของการส่งข้อมูลยังขึ้นอยู่กับความยาวของสายเคเบิล ในกรณีที่อุปกรณ์มีจำนวนมากกว่า 11 เครื่องขึ้นไป ความยาวของสายเคเบิลจะต้องไม่เกิน 20 เมตร สำหรับกรณีที่อุปกรณ์มีจำนวนไม่เกิน 10 เครื่องนั้น ความยาวของสายเคเบิ้ล(L) จะมากกว่า 20 เมตร จึงควรจำกัดความยาวของสายเคเบิ้ลด้วยสูตร

$$L < 2N \text{ ... เมตร}$$

โดย N เป็นจำนวนอุปกรณ์

ดังนั้นความยาวของสายเคเบิ้ลระหว่างอุปกรณ์ก็เช่นกัน จะมีค่ามากที่สุด 4 เมตร ในทางปฏิบัติจริงๆแล้ว แม้ความยาวจะเกินกว่า 4 เมตรก็ตามยังสามารถทำงานได้โดยไม่มีความผิดพลาดเลย แต่ควรจะปฏิบัติตามมาตรฐานจะปลอดภัยกว่า

ทางด้านความเร็วในการส่งข้อมูล ซึ่งมีค่าสูงสุด 1Mb/Sec แต่ก็ยังมีขีดจำกัดอยู่อีกเช่นกัน กล่าวคือ หลังจากที่ป้อนข้อมูลออกทางสายสัญญาณ DIO_1-DIO_8 แล้วสายสัญญาณ DAV จะต้องแอคตีฟโดยมีเวลาหน่วง T_1 ตามรูปที่ 2.5 ซึ่งค่าเวลา T_1 จะเป็นตัวกำหนดความเร็วในการส่งข้อมูลดังนี้

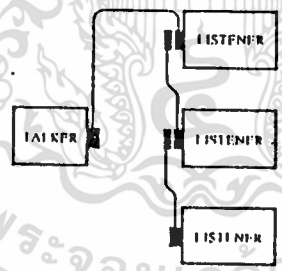
1. กรณีที่ใช้ driver เป็นแบบ open collector ขนาดกระแส 48 mA เวลาหน่วง T_1 จะมีค่ามากกว่า 2 µS และทุกๆระยะ 2 เมตร ติดตั้งโหลดมาตรฐานเข้าไปแล้วละก็ ความเร็วในการส่งระยะ 20 เมตร จะต้องมีค่าต่ำกว่า 250 kB/Sec
2. กรณีที่ใช้ driver เป็นแบบ Tri-state driver ขนาดกระแส 48 mA และติดตั้งโหลดมาตรฐานทุกๆระยะ 2 เมตร T_1 มีค่า 500 nS และความเร็วในการส่งในระยะ 20 m จะต้องมีค่าต่ำกว่า 500 kB/Sec
3. กรณีที่ใช้ Tri-state driver, T_1 = 350 nS และติดตั้งโหลดมาตรฐานทุกๆ 1 เมตร ความเร็วในการส่งในระยะ 15 เมตร จะต้องมีค่าต่ำกว่า 1 MB/Sec

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อควรระวังในการใช้ GPIB

กรณีที่ต้องการออกแบบหรือติดตั้งอุปกรณ์ GPIB จำเป็นต้องระมัดระวังสิ่งต่างๆดังต่อไปนี้

1. จะต้องไม่ยืดการเดินสายระหว่างขั้วคอนเน็คเตอร์ให้ยาวออกไป ซึ่งจะทำให้เกิด Coostalk และค่าตัวเก็บประจุในสายสัญญาณเพิ่มขึ้น
2. ปลายของสายชีลด์(ขาต่อเบอร์ 12) ต่อเข้ากับเฟรมตรงใกล้ๆกับขั้วต่อคอนเน็คเตอร์
3. เมื่อทำการต่อระหว่างอุปกรณ์ต่างๆ ด้วยสายเคเบิ้ลต้องระมัดระวังอย่าให้เกิดเป็นลูป(loop) ซึ่งในกรณีที่อุปกรณ์ใช้ภาคจ่ายไฟแบบ switchin อาจจะเกิดสัญญาณรบกวนเข้าไปในบัสสัญญาณได้ อาจทำให้เกิดการทำงานผิดพลาดได้
4. จะต้องไม่ให้มีการแพร่กระจายของสัญญาณรบกวนที่ไม่ต้องการออกไปจากสายเคเบิ้ล ซึ่งในกรณีที่อุปกรณ์เป็นไมโครคอมพิวเตอร์จะเกิดความถี่ฮาร์โมนิคประมาณ 100 mH จากความถี่นาฬิกาของระบบ ทำให้ไปรบกวนเครื่องรับวิทยุโทรทัศน์ได้ ซึ่งจะป้องกันได้โดยการใช้ขั้วต่อคอนเน็คเตอร์ที่มีการชีลด์ด้วยโครงโลหะอย่างดีตามมาตรฐานของ FCC



รูปที่ 2.6 ระบบการต่อ talk only และ listen only แบบง่ายๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบ TALK ONLY และ LISTEN ONLY

ในระบบ GPIB แบบง่ายๆที่ไม่ซับซ้อน จะประกอบไปด้วย talker และ listener ดังแสดงในรูปที่ 2.6 ซึ่งวิธีการใช้งานอย่างนี้เรียกว่า talk only,listen only.

talk only หมายความว่าอุปกรณ์ตัวหนึ่งถูกกำหนดให้เป็น talker(ส่งข้อมูล) ส่วน listen only หมายความว่าอุปกรณ์อีกตัวหนึ่งกำหนดให้เป็น listener(รับข้อมูล) ลักษณะเช่นนี้ใช้ต่อเมื่อต้องการส่งข้อมูลระหว่าง talker กับ listener โดยไม่มี controller

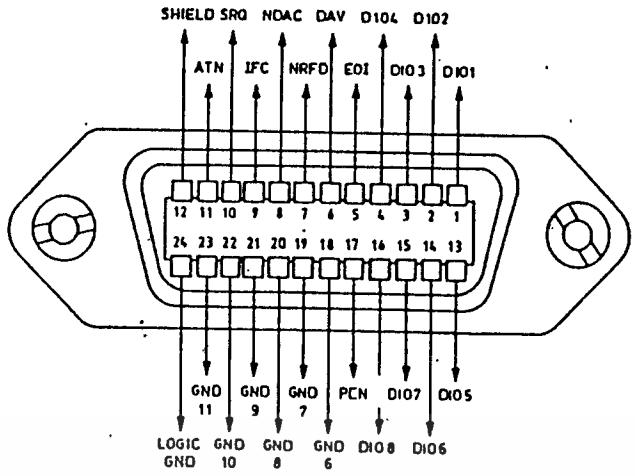
สำหรับระบบที่ทำงานโดยใช้ controller นั้น โดยทั่วไปแล้วไม่นิยมจะใช้เฉพาะกับกรณีที่ทำการตรวจสอบการส่งข้อมูลและเมื่อกำหนดให้เครื่องพิมพ์(printer) เป็น listener เท่านั้น ดังนั้นในระบบที่มี controller จะไม่มีอุปกรณ์ talk only ต่อเข้ากับบัสสัญญาณ ถ้าหากมีอุปกรณ์ talk only จะทำให้ไม่สามารถส่งข้อมูลจาก controller ได้ด้วยเหตุนี้ สำหรับอุปกรณ์ที่มีการทำงานเป็น talk only หรือ listen only จำเป็นต้องมีสวิทช์ on/off การทำงานติดมากับเครื่องด้วย

อุปกรณ์ที่มีระบบบัสเชื่อมต่อข้อมูลเอนกประสงค์

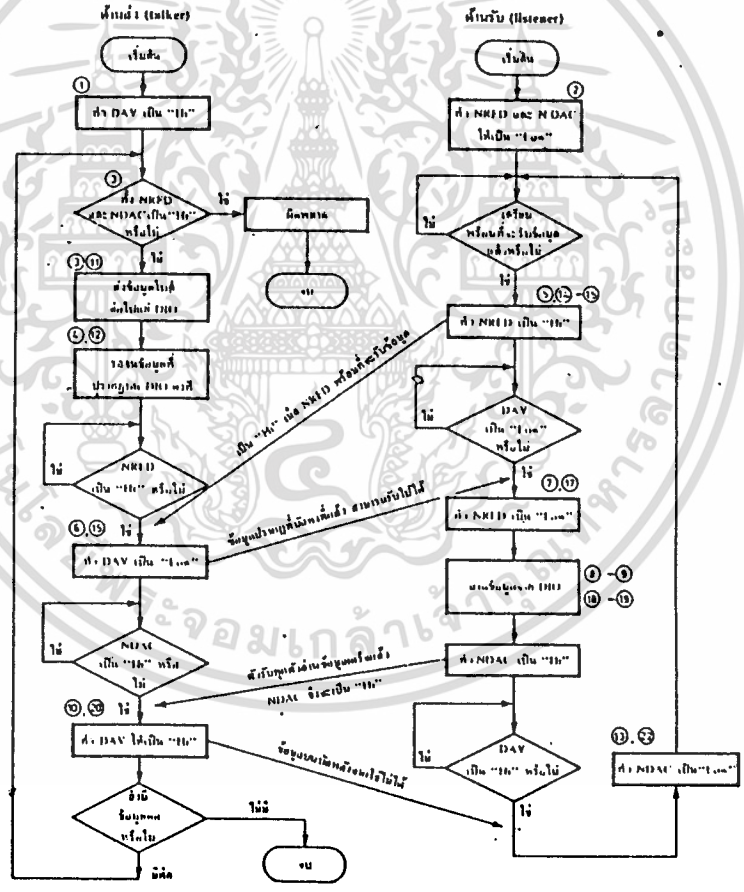
อุปกรณ์ที่มี GPIB แบ่งตามหน้าที่การทำงานได้ดังนี้

1. อุปกรณ์ที่ทำหน้าที่เป็นตัวส่งข้อมูล (talker) เท่านั้น เช่น เครื่องมือวัด เป็นต้น
2. อุปกรณ์ที่ทำหน้าที่เป็นตัวรับข้อมูล (Listener)เท่านั้น เช่น เครื่องพิมพ์ เครื่องบันทึก เป็นต้น
3. อุปกรณ์ที่ทำหน้าที่เป็นตัวรับและส่งข้อมูล (Listener and Talker) เช่น คอมพิวเตอร์ เครื่องมือวัดที่สามารถควบคุมได้จากภายนอก เป็นต้น
4. อุปกรณ์ที่ทำหน้าที่เป็นตัวรับข้อมูล ตัวส่งข้อมูล และตัวควบคุม (Listener,Talker and Controller) เช่น คอมพิวเตอร์ที่ทำหน้าที่ควบคุมระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

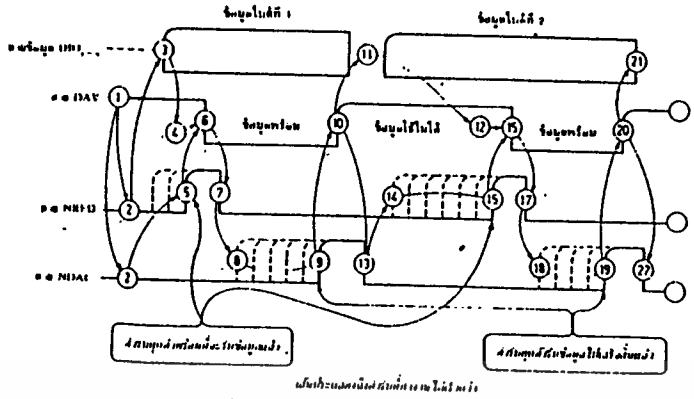


รูปที่ 2.7 ขั้วต่อของ GPIB และการจัดขาของสัญญาณต่างๆ



รูปที่ 2.8 ผังงานของการรับส่งข้อมูลแต่ละครั้งแสดงการโต้ตอบของสัญญาณ DAV, NRFD และ NDAC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 ผังเวลาแสดงความสัมพันธ์ของสัญญาณโต้ตอบในการรับส่งข้อมูล

<u>ความหมายของสัญญาณต่างๆในบัส</u>

GPIB เป็นระบบบัสแบบขนาด ดังรูป 2.7 ดังนั้นอุปกรณ์ทุกตัวที่ต่อร่วมบัสกันอยู่จึงขนานกันหมด สัญญาณต่างๆจากระบบบัสจึงปรากฏต่ออุปกรณ์ทุกตัวในจำนวนสายต่อทั้ง 24 เส้น ของบัสสามารถแบ่งออกได้เป็น 3 กลุ่มสัญญาณ คือ

กลุ่มสัญญาณควบคุมการรับ-ส่งข้อมูลประกอบด้วย

DAV (data valid) เมื่อถูกดึงเป็นลอจิก "Low" โดยอุปกรณ์ที่เป็นตัวส่ง(talker)เป็นการแจ้งแก่ระบบบัสว่า ตอนนี้ตัวส่งได้ทำการส่งข้อมูลลงไปที่สายสัญญาณข้อมูลเรียบร้อยแล้ว

NRFD (not ready for data) เมื่อมีลอจิก "Low" เป็นการแสดงว่าตอนนี้ระบบบัสยังไม่พร้อมที่จะรับข้อมูล เนื่องจากอุปกรณ์ในระบบบัสยังพร้อมไม่หมดทุกตัว ซึ่งสัญญาณเส้นนี้จะไม่เป็น "Hi"จนกว่าอุปกรณ์ในระบบบัสทุกตัวให้ลอจิก"Hi"ครบถ้วนแล้ว ใช้ประโยชน์สำหรับกรณีที่อุปกรณ์ที่ใช้ร่วมกันมีความเร็วในการทำงานแตกต่างกัน

NDAC (not data accept) สัญญาณเส้นนี้ ควบคุมโดยอุปกรณ์ตัวรับ(listener) จะมีลอจิก "Low"ในขณะที่ตัวรับกำลังเก็บข้อมูลจากสายข้อมูล และจะเป็น "Hi" เมื่ออ่านข้อมูลเสร็จเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มสัญญาณควบคุมการอินเตอร์เฟสประกอบด้วย

OUT
ATN (attention) เป็นสัญญาณจากอุปกรณ์ที่เป็นตัวควบคุม(controller) ใช้ในการสั่งให้อุปกรณ์ทุกตัวในระบบเตรียมพร้อมเพื่อรอรับคำสั่งต่อไป

OUT
IFC (interface clear) เป็นสัญญาณรีเซตหรือเคลียร์ระบบ กำเนิดได้โดยตัวควบคุมเท่านั้น เมื่ออุปกรณ์ในบัสได้รับสัญญาณเคลียร์นี้ จะกลับคืนสู่สภาวะเริ่มต้นใหม่ เป็นสภาวะแรกเริ่มก่อนการกำหนดฟังก์ชันเหมือนเมื่อแรกเปิดสวิทช์

OUT
REN (remote enable) สัญญาณเส้นนี้ควบคุมโดยอุปกรณ์ตัวควบคุมตัวเดียวเท่านั้นเช่นกัน ใช้สั่งให้อุปกรณ์เปลี่ยนจากโหมดที่ใช้งานปกติด้วยมือมาเป็นการควบคุมโดยตัวควบคุมแทน

IN
SRQ (service request) เป็นสัญญาณอินเทอร์รัพต์ เพื่อเป็นการบอกแก่ระบบว่า ขณะนี้อุปกรณ์ต้องการการติดต่อจากตัวควบคุมยกตัวอย่างเช่น โวลต์มิเตอร์ที่มีค่าที่อ่านได้พร้อมแล้วที่จะส่งให้แก่ระบบเพื่อไปยังอุปกรณ์ที่ต้องการข้อมูลนี้ หรือเมื่อเกิดมีการพบข้อผิดพลาดบางอย่าง

I/O
EOI (end of identify) สัญญาณที่ควบคุมได้ทั้งโดยอุปกรณ์ที่เป็นตัวควบคุมหรือตัวส่งก็ได้ ใช้แสดงว่าข่าวสารข้อมูลที่ส่งเป็นชุดนั้นสิ้นสุดลงแล้ว

I/O
กลุ่มสัญญาณข้อมูล (Data Line)

ประกอบด้วยสายสัญญาณจำนวน 8 เส้น สำหรับเป็นทางผ่านของข่าวสารข้อมูลของระบบ

สัญญาณลอจิกที่ใช้ใน GPIB บัสนี้มีลักษณะเป็นคอมพลีเมนต์ทั้งหมด คือ "1" เท่ากับ "Low" และ "0" เท่ากับ "Hi" ซึ่งตรงข้ามกับที่เราคุ้นเคยกัน

สัญญาณควบคุมที่มีอยู่อาจไม่ได้ใช้ทั้งหมดพร้อมกัน บางสายอาจไม่ได้ใช้งานในอุปกรณ์บางเครื่องก็ได้ แล้วแต่ความจำเป็นในการติดต่อสำหรับสายที่เหลืออีก 7 เส้นนั้นเป็นสายกราวนด์

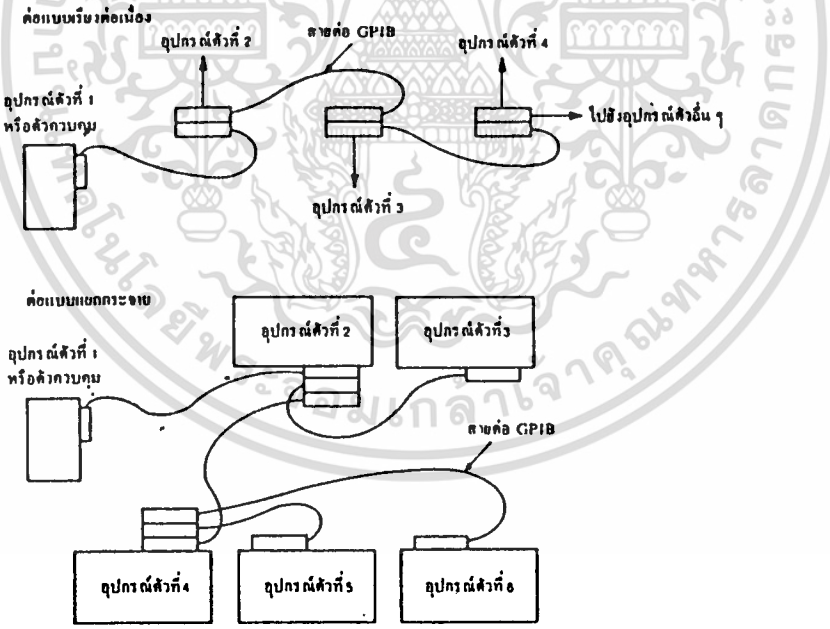
ในเรื่องของรายละเอียดทางด้านสัญญาณคงจะไม่กล่าวลึกไปกว่านี้ ซึ่งเป็นส่วนของผู้ออกแบบระบบที่จะใช้งานมากกว่าในส่วนผู้ใช้งานคงไม่จำเป็นนัก

สำหรับการต่ออุปกรณ์ต่างๆในระบบ GPIB อินเทอร์เฟสบัส มีอยู่ 2 วิธีคือแบบต่อเนื่องกันไปเรื่อยๆ จากเครื่องหนึ่งไปยังอีกเครื่องหนึ่ง และอีกวิธีคือ ต่อแบบกระจาย ดังแสดงในรูปที่ 2.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IEEE Standard		IEC Standard	
Pin	Designation	Pin	Designation
1	DIO1	1	DIO1
2	DIO2	2	DIO2
3	DIO3	3	DIO3
4	DIO4	4	DIO4
5	EOI	5	REN
6	DAV	6	EOI
7	NRFD	7	DAV
8	NDAC	8	NRFD
9	IFC	9	NDAC
10	SRQ	10	IFC
11	ATN	11	SRQ
12	SHIELD	12	ATN
13	DIO5	13	SHIELD
14	DIO6	14	DIO5
15	DIO7	15	DIO6
16	DIO8	16	DIO7
17	REN	17	DIO8
18	GND 6	18	GND 5
19	GND 7	19	GND 6
20	GND 8 (สายกราวด์ของ)	20	GND 7
21	GND 9 (สายที่ 6-11)	21	GND 8 (สายกราวด์ของ)
22	GND 10	22	GND 9 (สายที่ 6-12)
23	GND 11	23	GND 10
24	LOGIC GND	24	GND 11
		25	GND 12

ตารางที่ 1 เปรียบเทียบการจัดตำแหน่งสัญญาณระหว่าง IEEE Std488-1978 กับ IEC635-1



รูปที่ 2.10 ลักษณะการต่ออุปกรณ์เข้ากับระบบบัส GPIB ทั้งสองวิธี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งใช้งานของ GPIB

เมื่อมีการเชื่อมต่อหรืออินเทอร์เฟสอุปกรณ์เครื่องมือต่างๆ เข้าด้วยกันด้วย GPIB บัสแล้ว ตัวอย่างเช่น อาจจะเป็นการต่อคอมพิวเตอร์กับเครื่องวัดความถี่, ปริ๊นเตอร์ และดิสไดร์ฟ ตัวคอมพิวเตอร์ก็จะทำหน้าที่เป็นตัวควบคุมเพื่อบริหารอุปกรณ์อื่นๆ และเป็นตัวรับเพื่อรับข้อมูลจากเครื่องวัดความถี่ ขณะเดียวกันก็อาจจะเป็นตัวส่งค่าความถี่ที่วัดได้ให้แก่ตัวอื่นและปริ๊นท์เตอร์เป็นตัวรับข้อมูลมาพิมพ์ ส่วนดิสค์ไดร์ฟก็เป็นได้ทั้งตัวรับและตัวส่งข้อมูล

จะเห็นว่าอุปกรณ์แต่ละตัวสามารถเป็นได้หลายอย่าง ขึ้นอยู่กับว่าในการทำงานขณะหนึ่งๆ ต้องการให้เป็นอะไรซึ่งอุปกรณ์ทุกตัวที่ต่ออยู่ในบัสจะมีหมายเลขรหัสประจำตัวสำหรับการอ้างถึง (เหมือนกับบ้านที่ต้องมีเลขที่) เพื่อให้ตัวควบคุมสั่งการบังคับหน้าที่ต่างๆได้ และใช้ในการอ้างอิงจุดหมายปลายทางของข้อมูล

การสั่งการต่างๆเพื่อกำหนดหน้าที่การทำงาน และกำหนดฟังก์ชัน เช่น กำหนดช่วงการวัด โหมดการวัด หรืออื่นๆแก่เครื่องวัดที่ต่ออยู่ เหล่านี้นั้นเป็นตัวควบคุมจะเป็นตัวกำหนด โดยการส่งรหัสคำสั่งไปที่อุปกรณ์ โดยผ่านสายสัญญาณข้อมูล DI_1-DI_8 รหัสคำสั่งเหล่านี้จะถูกส่งไปในช่วงที่สายสัญญาณ ATN เป็น "Low" (ในระบบง่ายๆที่ไม่ซับซ้อน ซึ่งไม่จำเป็นต้องใช้ตัวควบคุม อุปกรณ์จะถูกกำหนดไว้ตายตัว เช่นให้เป็นตัวส่งอย่างเดียว(talk only) หรือรับอย่างเดียว (listen only)

คำสั่งสำหรับกำหนดหน้าที่การทำงานแบบต่างๆตามมาตรฐานของ GPIB มีอยู่ด้วยกัน 128 คำสั่ง ดังแสดงในตารางที่ 2 โดยแบ่งออกเป็น 5 กลุ่มคำสั่ง

รหัสที่ใช้ในระบบ GPIB บัสนั้น ใช้ร่วมกันทั้งรหัสข้อมูลและรหัสคำสั่ง นั่นคือรหัสเดียวกันมีความหมายได้ 2 อย่างคือเมื่อสายสัญญาณ ATN เป็น "Low"จะหมายถึงรหัสคำสั่ง แต่ถ้า ATN เป็น "Hi" รหัสนี้จะแทนข้อมูลที่เป็น ASCII แทน ซึ่งในตารางที่ 2 ก็ได้แบ่งความหมายออกเป็น 2 คอลัมน์ให้เห็น รหัสคำสั่งต่างๆมีดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.กลุ่มคำสั่งเจาะจงจุดหมาย(address command group) เป็นคำสั่งที่ส่งไปยังอุปกรณ์ที่เป็นตัวส่งหรือตัวรับที่กำหนดไว้ล่วงหน้าแล้ว ให้มีสภาพการทำงานตามที่ต้องการ คำสั่งในกลุ่มนี้ประกอบด้วย

GTL(go to local) สั่งให้อุปกรณ์กลับสู่สภาพการควบคุมปกติด้วยมือ

SDC(selected divice clear) สั่งให้อุปกรณ์เคลียร์ตัวเองสู่สภาพเริ่มต้นใหม่

PPC(parallel poll configure) เป็นคำสั่งสำหรับทำการจัดสรรสายสัญญาณของการทำกระบวนการตรวจสอบสภาพอุปกรณ์โดยวิธีขนานหรือ parallel poll(ซึ่งจะอธิบายภายหลัง) โดยใช้งานร่วมกับคำสั่งในกลุ่มรอง

GET(group excute trigger) ใช้สั่งเริ่มต้นการทำงานของอุปกรณ์ทีละหลายตัว

TCT(take control) เป็นการกำหนดให้อุปกรณ์ตัวส่งทำหน้าที่เป็นตัวควบคุม

ASCII — IEEE 488 BUS MESSAGES (COMMANDS AND ADDRESSES) HEX CODES

MSD / LSD	0		1		2		3		4		5		6		7	
	ASCII	MSG	ASCII	MSG	ASCII	MSG¹	ASCII	MSG¹	ASCII	MSG¹	ASCII	MSG¹	ASCII	MSG	ASCII	MSG
0	NUL		DLE		SP	00	0	16	@	00	P	16	`	MEANING DEFINED BY PCG CODE	p	MEANING DEFINED BY PCG CODE
1	SOH	GTL	DC1	LLO	!	01	1	17	A	01	Q	17	a		q	
2	STX		DC2		"	02	2	18	B	02	R	18	b		r	
3	ETX		DC3		#	03	3	19	C	03	S	19	c		s	
4	EOT	SDC	DC4	DCL	$	04	4	20	D	04	T	20	d		t	
5	ENQ	PPC	NAK	PPU	%	05	5	21	E	05	U	21	e		u	
6	ACK		SYN		&	06	6	22	F	06	V	22	f		v	
7	BEL		ETB		'	07	7	23	G	07	W	23	g		w	
8	BS	GET	CAN	SPE	(	08	8	24	H	08	X	24	h		x	
9	HT	TCT	EM	SPD	)	09	9	25	I	09	Y	25	i		y	
A	LF		SUB		*	10	:	26	J	10	Z	26	j		z	
B	VT		ESC		+	11	;	27	K	11	[	27	k		{	
C	FF		FS		,	12	<	28	L	12	\	28	l		|	
D	CR		GS		-	13	=	29	M	13	]	29	m		}	
E	SO		RS		.	14	>	30	N	14	^	30	n		~	
F	SI		US		/	15	?	UNL	O	15	_	UNT	o		DEL	

ADDRESSED COMMAND GROUP | UNIVERSAL COMMAND GROUP | LISTEN ADDRESS GROUP | TALK ADDRESS GROUP | SECONDARY COMMAND GROUP

PRIMARY COMMAND GROUP (PCG)

ตารางที่ 2 รหัสข้อมูลข่าวสารของระบบ GPIB บัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.กลุ่มคำสั่งควบคุม (universal command group) เป็นคำสั่งที่ส่งไปยังอุปกรณ์ทุกตัวที่ต่ออยู่ในบัส ประกอบด้วย

LLO (local lockout) เป็นการสั่งให้อุปกรณ์ล็อกอยู่ที่สภาวะควบคุมโดยปุ่มปรับตามปกติ

DCL (device clear) สั่งให้อุปกรณ์ทุกตัวกลับไปสู่สภาวะเริ่มต้น

PPL (parallel poll unconfigure) ใช้ยกเลิกกระบวนการตรวจสอบสภาพแบบขนานทั้งหมด

SPE (serial poll enable) เปลี่ยนโหมดของการตรวจสอบสภาพเป็นแบบอนุกรม ในโหมดนี้จะเป็นการส่งสถานะของเครื่องแทนการส่งข้อมูล

SPD (serial poll disable) ยกเลิกโหมดการตรวจสอบแบบอนุกรม

3. กลุ่มคำสั่งกำหนดอุปกรณ์ตัวรับ(listener address group) เป็นคำสั่งสำหรับกำหนดให้อุปกรณ์เป็นตัวรับตามรหัสหมายเลขจาก 0 ถึง 30 และมีคำสั่ง UNL (unlistener) สำหรับยกเลิก

4. กลุ่มคำสั่งกำหนดอุปกรณ์ตัวส่ง(talker address group) สำหรับกำหนดให้อุปกรณ์เป็นตัวส่ง ตามรหัสหมายเลขจาก 0 ถึง 30 และมีคำสั่ง UNL(unlistener) สำหรับยกเลิกเช่นกัน

คำสั่งในกลุ่มที่ 1 ถึง 5 นั้นจัดเป็น *กลุ่มคำสั่งหลัก(primary command group)* ที่มีความหมายตายตัว ยังมีคำสั่งอีกกลุ่มที่ขึ้นอยู่กับการกำหนดภายหลังนั่นคือ กลุ่มคำสั่งรอง

5. กลุ่มคำสั่งรอง (secondary command group) เป็นคำสั่งที่กำหนดรายละเอียดย่อยของอุปกรณ์แต่ละตัวที่ต่ออยู่ในระบบ ให้มีการทำงานอย่างไรตามจุดประสงค์ใช้งานของเครื่องมือนั้นเช่นเดียวกับการปรับปุ่มปรับต่างๆด้วยมือนั่นเอง คำสั่งรองนี้จะตามหลังคำสั่งหลัก คือจะให้หลังจากอุปกรณ์ต่างๆถูกกำหนดวางตัวในระบบเรียบร้อยแล้ว

คำสั่งต่างๆที่กล่าวไป ซึ่งใช้ในการกำหนดสภาวะการทำงานของอุปกรณ์ แต่ละสภาวะที่กำหนดไปนั้นเป็นอย่างไร และมีจุดประสงค์เพื่ออะไรเรามาดูกันต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Device Clear/Interface Clear

ดิไวซ์เคลียร์(device clear) ใช้ในการทำให้อุปกรณ์ที่ต่ออยู่ในบัสกลับไปอยู่ในสภาวะเหมือนเมื่อแรกเริ่มที่เปิดไฟเข้าเครื่อง อันเป็นสภาวะเริ่มต้น ยังไม่มีการกำหนดฟังก์ชันใดๆ สภาวะเริ่มต้นนี้จะแตกต่างกันไปแล้วแต่ว่าอุปกรณ์นั้นออกแบบมาไว้อย่างไร ดิไวซ์เคลียร์มีอยู่ 2 ลักษณะ คือ เคลียร์หมดทุกตัวที่ต่ออยู่ (DCL) กับเคลียร์เจาะจงอุปกรณ์ตัวใดตัวหนึ่ง (SDC)

แต่ว่าในการเคลียร์อุปกรณ์ให้อยู่ในสภาวะเริ่มต้นนั้น ไม่ได้หมายความว่าอินเทอร์เฟสฟังก์ชันของ GPIB จะถูกเคลียร์ให้ไปอยู่ในสภาวะเริ่มต้นด้วยแต่อย่างใด เป็นแต่การเคลียร์เฉพาะตัวอุปกรณ์เท่านั้น อินเทอร์เฟสฟังก์ชันก็คือ สภาพการอินเทอร์เฟสที่ได้กำหนดเอาไว้ในระบบประกอบด้วยฟังก์ชันต่างๆดังแสดงในตารางที่ 3

ฟังก์ชัน	สัญลักษณ์	การกลับสู่สภาพเริ่มต้นโดย IFC
source hand shake	SH	✓
acceptor hand shake	AH	✓
talker หรือ enlarge talker	T หรือ TE	✓
listener หรือ enlarge listener	L หรือ LT	✓
service request	SR	—
remote/local	RL	—
parallel poll	PP	—
device clear	DC	✓
device trigger	DT	✓
controller	C	✓

ตารางที่ 3 ฟังก์ชันการอินเทอร์เฟสในระบบ GPIB บัส

การเคลียร์สภาพการอินเทอร์เฟสไปอยู่ในสภาวะเริ่มต้นต้องใช้คำสั่ง *อินเทอร์เฟสเคลียร์ (Inierface Clear) แทน* ซึ่งจะทำให้ทุกฟังก์ชันถูกยกเลิกไป ยกเว้น SR (survice request),RL(remote/local) และ PP(parallel poll) นั่นก็คือถึงจะยกเลิกสภาพอินเทอร์เฟส แต่จะยังคงรักษาลักษณะการจัดของ SR,RL และ PP เอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Remote/Local

รีโมท(remote) เป็นการกำหนดให้อุปกรณ์ที่ต่ออยู่ในระบบ เช่น เครื่องมือวัดให้อยู่ในการควบคุมของอุปกรณ์ตัวอื่นแทน ซึ่งปุ่มปรับต่างๆบนหน้าปัดเครื่องจะไม่มีผลต่อการทำงาน ส่วนโลคัล (local) เป็นการควบคุมการทำงานของเครื่องมือวัดด้วยปุ่มปรับบนหน้าปัดตามปกติหรือควบคุมด้วยมือนั่นเอง

การใช้รีโมทมีประโยชน์ในแง่ที่ขณะที่ตัวควบคุม เช่น คอมพิวเตอร์กำลังติดต่อกับอุปกรณ์ตัวนั้นอยู่ หากไม่มีการตัดการควบคุมโดยปุ่มปรับบนหน้าปัดออก ถ้ามีใครมาปรับแต่งก็จะไม่ให้คอมพิวเตอร์มายุ่งเกี่ยวได้ การทำงานของ GPIB ในรีโมทและโลคัลมี 4 ลักษณะดังนี้

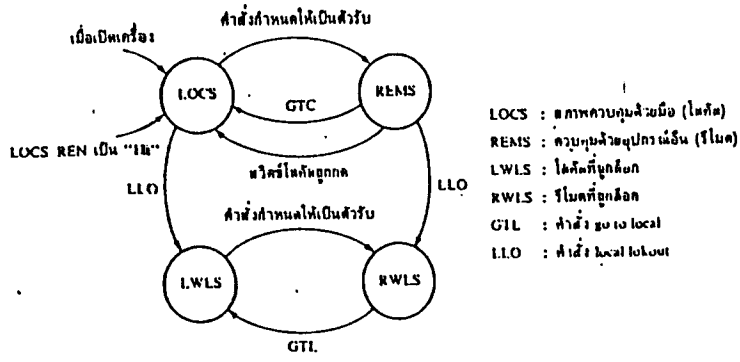
1.LOCS ก็คือโลคัลนั้นเอง อันเป็นสภาพการควบคุมที่ปุ่มปรับตามปกติ ซึ่งอุปกรณ์จะอยู่ในสภาพนี้เมื่อตอนเปิดไฟเข้าเครื่อง หรือ REN(remote enable) มีลอจิก "Hi" หรือเมื่อได้รับคำสั่ง GTL(go to local)

2.REMS คือรีโมท หมายถึงการตัดการควบคุมโดยปุ่มบนหน้าปัดออกกลายเป็นการควบคุมการทำงานโดยอุปกรณ์อื่นแทน สภาพรีโมทโดย REN มีลอจิก "Low" และจะถูกล็อกเอาไว้ เว้นแต่ว่าสวิทช์โลคัลที่ตัวอุปกรณ์จะถูกเปลี่ยนไปตำแหน่ง local

3.RWLS เป็นสภาพรีโมทที่ถูกล็อกเอาไว้เช่นกัน แต่ว่าจะตัดการควบคุมของสวิทช์โลคัลที่ตัวอุปกรณ์ออกไป ฉนั้น สภาพรีโมทโดย RWLS จึงมีนัยสำคัญสูงกว่า REMS อย่างไรก็ตามก็ยังถูกยกเลิกได้ด้วยคำสั่ง LLO(local lockout)

4.LWLS มีสภาพเช่นเดียวกับโลคัล แต่จะแตกต่างกันตรงที่สภาพโลคัลโดย LWLS นี้ เมื่อได้รับคำสั่งกำหนดอุปกรณ์ตัวรับ จะเปลี่ยนไปอยู่ในสภาพรีโมทแบบล็อคหรือ RWLS ทันที ในการที่จะมาที่สภาพ LWLS นี้ได้ก็มี 2 กรณีคือ เมื่ออยู่ในสภาพโลคัลธรรมดา(LOCS) แล้วได้รับคำสั่ง LLO(local lockout) หรือ เมื่ออยู่ใน RWLS แล้วได้รับคำสั่ง GTL(go to local)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 สภาพโลคัลและรีโมทลักษณะต่างๆและแผนภูมิแสดงการเปลี่ยนสภาวะ

ในรูปที่ 2.11 เป็นการแสดงการเปลี่ยนสภาวะต่างๆของโลคัลและรีโมททั้ง 4 ลักษณะตามที่กล่าวมา ซึ่งเมื่อ REN มีลอจิก "Hi" ก็จะกลายเป็นโลคัลทันทีไม่ว่าเดิมจะมีในสภาพใดและเมื่อ REN เป็น "Low" แล้วหากไม่มีคำสั่งกำหนดอุปกรณ์ตัวรับหรือคำสั่ง LLO เข้ามา ก็จะยังคงอยู่ในสภาพโลคัลเช่นนั้น

เมื่อเครื่องถูกกำหนดให้อยู่ในสภาพโลคัล หรือควบคุมด้วยมือแล้วหากคอมพิวเตอร์ส่งคำสั่งเข้ามาเพื่อกำหนดให้เป็นอุปกรณ์ตัวรับ เครื่องจะไม่สนใจคำสั่งดังกล่าวและหากถูกสั่งให้เป็นตัวส่งเครื่องก็จะส่งข้อมูลอย่างใดอย่างหนึ่งออกไปให้ นั่นคือ ระบบจะให้ความสำคัญต่อสภาพโลคัลมากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบฮาร์ดแวร์

แผ่นวงจรพิมพ์นี้ได้ออกแบบสำหรับต่อกับเครื่อง IBM PC โดยออกแบบการ์ดสำหรับเสียบลงในสล็อตของเครื่อง IBM PC ใช้ได้ทันทีโดยมีโปรแกรมสนับสนุนแผ่นวงจรพิมพ์ควบคุมการทำงานของแผ่นวงจรพิมพ์ รายละเอียดของแผนวงจรพิมพ์ในที่นี้จะอธิบายเพียงคร่าวๆซึ่งรายละเอียดทั้งหมดสามารถอ่านได้จาก หนังสือโครงงานแผ่นวงจรเชื่อมต่อข้อมูลเอนกประสงค์ตามมาตรฐาน IEEE-488

3.1 ส่วนประกอบของแผ่นวงจรพิมพ์

ประกอบด้วย วงจรรวม(IC) 8 ตัวดังนี้

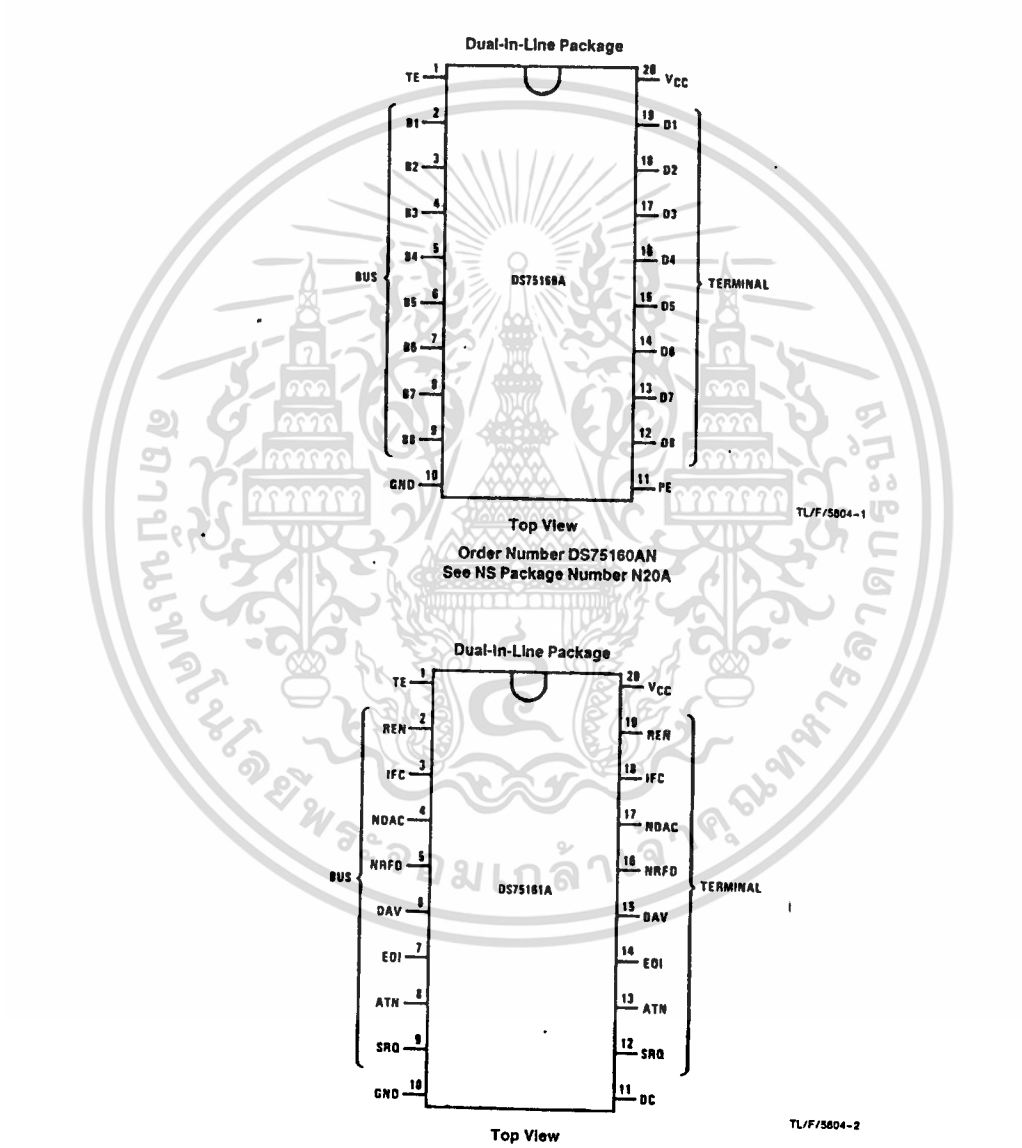
1.วงจรรวมเบอร์ 74ls04 (Hex Inverter)	2	ตัว
2.วงจรรวมเบอร์ 74ls133 (13 Input Nand)	2	ตัว
3.วงจรรวมเบอร์ 8255 (Programmable - Peripheral Interface)	2	ตัว
4.วงจรรวมเบอร์ DS75160A (Octal General-Purpose Interface bus Transceiver)	1	ตัว
5.วงจรรวมเบอร์ DS75161A (Octal General-Purpose Interface bus Transceiver)	1	ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบวงจรบัฟเฟอร์

ในส่วนของวงจรบัฟเฟอร์เราได้เลือกไอซีเบอร์ DA74160A และ DS74161A ซึ่งเป็นไอซีที่ออกแบบมาให้ใช้กับระบบบัสของ GPIB โดยเฉพาะ ซึ่งมีรายละเอียดดังนี้

Connection Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description

DS75160A

-เป็นอุปกรณ์ที่ 8 channel ที่มีการส่งผ่านสองทิศทาง(driver/receiver) ในตัวเดียวกัน โดยอาศัยการควบคุมทิศทางอินพุท คือ TE

-ต่ออยู่กับระบบบัสทั้ง 8 เส้นของ GPIB คือ DIO_1-DIO_8 ตามมาตรฐานของ IEEE-488

-เมื่อเกิด power down(vcc=0) จะไม่เกิดการ Load

-นอกจากนั้นพอร์ตเอาท์พุทยังสามารถกำหนดโหมดให้ Enable หรือ Disable การต่อใช้งานในแบบ Totempole ได้อีกด้วย

-เมื่อสัญญาณควบคุมอินพุท PE อยู่ในสภาวะ high บัสเอาท์พุทจะอยู่ในสภาวะ high-speed totempole mode และเมื่อ PE อยู่ในสภาวะ Low เอาท์พุทจะทำตัวเป็น open collector ซึ่งใช้ประโยชน์สำหรับการ polling แบบขนาน

DS75161A

-เป็นอุปกรณ์ที่มีการส่งผ่านได้สองทิศทาง และมี 8 channel เช่นเดียวกับ 75160A แต่มีโครงสร้างที่พิเศษเฉพาะคือใช้ทำหน้าที่จัดการเกี่ยวกับสัญญาณควบคุมในระบบบัสของ IEEE-488

-เป็นอุปกรณ์ที่ออกแบบมาให้ใช้งานร่วมกับ 75160A ประกอบเป็นบัส 16 เส้นต่ออินเทอร์เฟสระหว่าง IEEE-488 Bus กับ Single Controller Instrumentation system

-ในการใช้งานสัญญาณควบคุม SRQ,NDAC และ NRFD ในระบบบัสจะมีเอาท์พุทพอร์ทเป็นแบบ open collector

การควบคุมทิศทางของสัญญาณควบคุมถูกแบ่งออกเป็น 3 กลุ่มด้วยกันคือ

1.DAV,NDAC และ NRFD เป็นกลุ่มสัญญาณรับส่งข้อมูลซึ่งถูกควบคุมโดย TE อินพุท

2.ATN,SRQ และ IFC เป็นกลุ่มสัญญาณควบคุมการอินเทอร์เฟสถูกควบคุมโดย DC อินพุท

3.EOI ทิศทางการส่งผ่านข้อมูลจะถูกควบคุมทั้ง TE และ DC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบส่วนของพอร์ทควบคุม

เราได้ทราบแล้วว่าแบบมาตรฐานของ IEEE-488 นั้นประกอบด้วยบัสต่างๆจำนวน 18 เส้นซึ่งสามารถแบ่งได้เป็น 3 กลุ่มด้วยกัน คือ

1.กลุ่มสัญญาณควมคุมการรับส่งข้อมูล ซึ่งประกอบด้วย DAC,NRFD และ NDAC ซึ่งถูกควบคุมโดย TE

2.กลุ่มสัญญาณควบคุมการอินเทอร์เฟส ประกอบด้วย ATN,IFC,REN,SRQ ถูกควบคุมโดย DC และ EOI ควบคุมได้ทั้ง DC และ TE

3.กลุ่มสัญญาณข้อมูล เป็นเส้นทางผ่านข้อมูลของระบบจำนวน 8 เส้น

และนอกจากนั้นเรายังต้องทำการกำหนดสัญญาณควบคุมของ IC 75160A และ 75161A ด้วยนั้นคือสัญญาณ PE,TE,DC และ TE ตามลำดับ ดังนั้นเราจึงต้องกำหนดพอร์ททั้งสิ้นจำนวน 4 พอร์ทด้วยกัน และพอร์ทนี้จะต้องเป็นได้ทั้งอินพุทและเอาท์พุทพอร์ท ซึ่งตรงกับคุณสมบัติของ IC#8255 พอดี ซึ่งเราได้กล่าวในตอนแรกแล้วว่าภายใน 8255 1ตัวจะประกอบด้วยพอร์ทสามพอร์ทอิสระต่อกัน

ดังนั้นจึงขอกำหนดการทำงานของ 8255 ดังต่อไปนี้

1.ให้ 8255 ทำงานในโหมดศูนย์

2.กำหนดให้พอร์ท A เป็นพอร์ทของสัญญาณข้อมูล 8 บิท โดยกำหนดให้เป็นได้ทั้งอินพุทและเอาท์พุท

พอร์ท B เป็นพอร์ทที่ใช้สำหรับกำหนดทิศทางของสัญญาณข้อมูล และสัญญาณควบคุมโดยกำหนดให้เป็นเอาท์พุทพอร์ทเท่านั้น

พอร์ท C ใช้ทั้งพอร์ท c บนและพอร์ท c ล่าง เป็นได้ทั้งอินพุทพอร์ท และเอาท์พุทพอร์ทโดยจะใช้ 8255 2ตัว เพื่อความสะดวกในการควบคุมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดกลุ่มสัญญาณ

การจัดกลุ่มสัญญาณ จะพิจารณาจากทิศทางของสัญญาณที่กำหนดไว้ในวงจรรวมเบอร์ DS75160A และ DS75161A ซึ่งทำหน้าที่เป็นบัฟเฟอร์สำหรับการรับส่งข้อมูล และสัญญาณควบคุมสำหรับระบบ GPIB โดยเฉพาะ และพิจารณาถึงความสะดวกในการควบคุมการทำงานของ 8255

กลุ่มของสัญญาณต่างๆ

1.DIO_1-DIO_8 ต่อกับพอร์ต A ของ 8255(1)
2.NDAC,NRFD ต่อกับพอร์ต C บนของ 8255(1)
3.DAV,EOI ต่อกับพอร์ต C ล่างของ 8255(1)
4.SRQ ต่อกับพอร์ต C บนของ 8255(2)
5.REN,IFC,ANT ต่อกับพอร์ต C ล่างของ 8255(2)

และกลุ่มสัญญาณควบคุมต่างๆบน 8255 ซึ่งประกอบด้วย RD,WR และ RESET สามารถต่อเข้าโดยตรงกับ IBM SLOT ของคอมพิวเตอร์ได้เลย เพื่อใช้ในขบวนการอ่าน เขียน และรีเซตตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบวงจรถอดรหัส

เนื่องจากเราได้เลือกพอร์ตหมายเลข 300-31F ซึ่งได้กล่าวไว้แล้วในตอนต้นมาเป็นตัวกำหนดตำแหน่งของ 8255 ดังนั้นเราจะต้องใช้แอดเดรส A_0-A_9 มาทำการถอดรหัสดังรูป

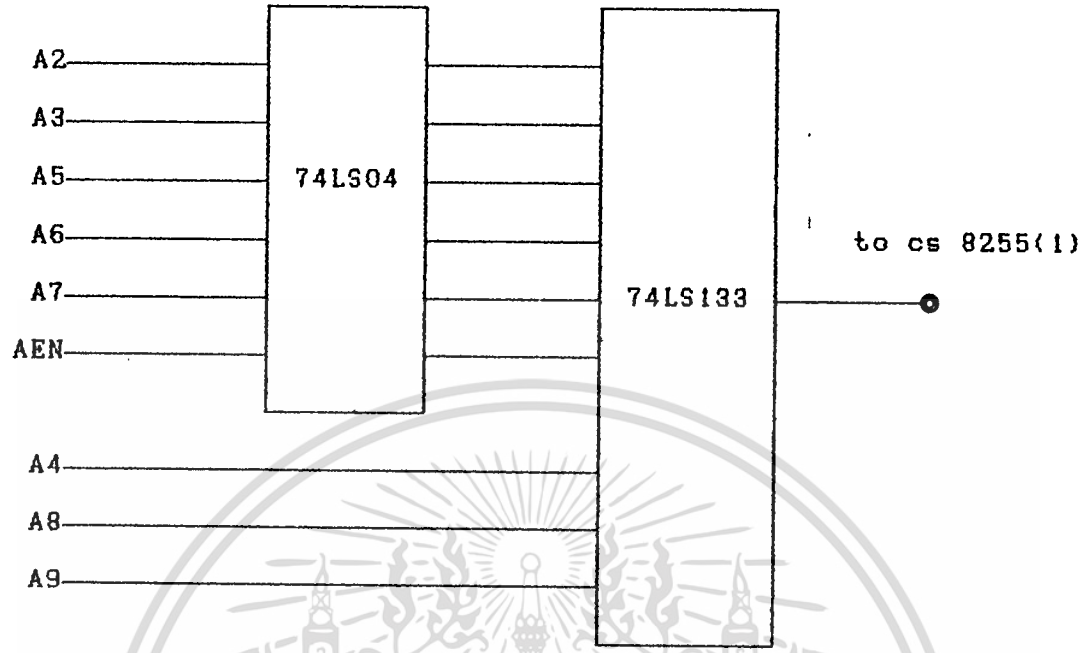
A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	PORT NO.	PORT NAME	SELECT
1	1	0	0	0	1	0	0	0	0	310	port A	
1	1	0	0	0	1	0	0	0	1	311	port B	8255
1	1	0	0	0	1	0	0	1	0	312	port C	(1)
1	1	0	0	0	1	0	0	1	1	313	control	
1	1	0	0	0	1	0	1	0	0	314	port A	
1	1	0	0	0	1	0	1	0	1	315	port B	8255
1	1	0	0	0	1	0	1	1	0	316	port C	(2)
1	1	0	0	0	1	0	1	1	1	317	control	

นั่นคือ พอร์ทแอดเดรส 0310h-0313h ถอดรหัสเพื่อเรียก 8255(1)

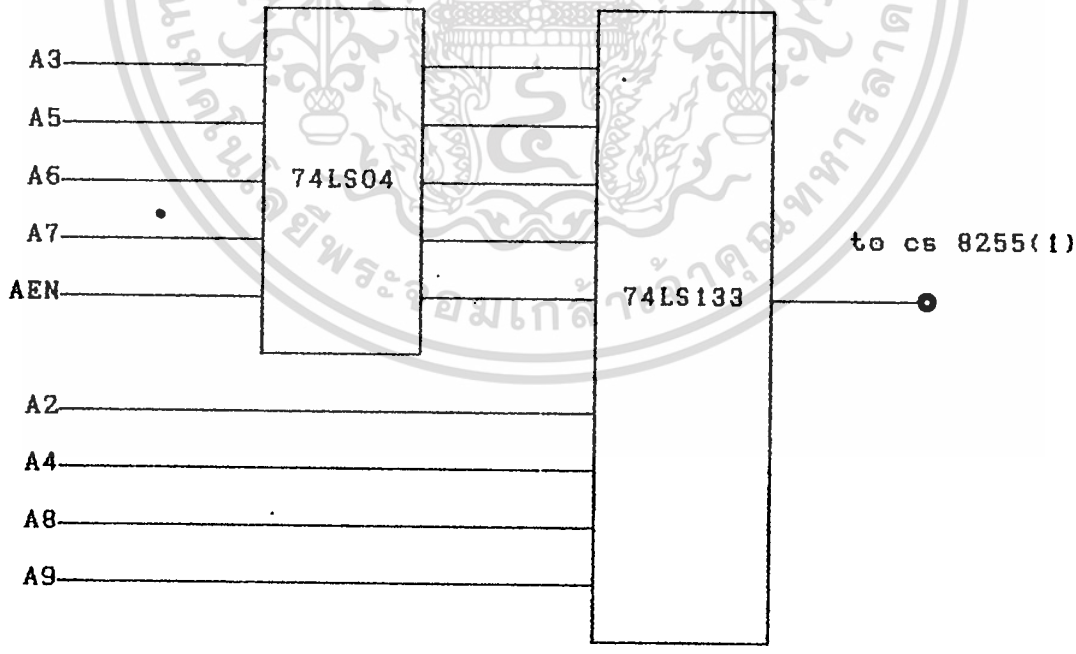
พอร์ทแอดเดรส 0314h-0317h ถอดรหัสเพื่อเรียก 8255(2)

เมื่อพิจารณาจากตารางจะเห็นว่าในส่วนของ 8255(1) ส่วนที่เปลี่ยนแปลงคือ A_0-A_1 ดังนั้นเราจึงสามารถออกแบบในส่วนของฮาร์ดแวร์ได้โดยใช้ INVERTER ร่วมกับ NAND GATE ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เช่นเดียวกันในส่วนของ 8255(2) เราก็สามารถออกแบบได้ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ A_0-A_1 นั้นต่อเข้าโดยตรงกับ 8255 ทั้ง 2 ตัวเลยเพื่อใช้ถอดรหัสพอร์ตภายใน ซึ่งหมายเลขพอร์ตแสดงไว้ในตารางด้านบนแล้ว นอกจากนั้นเราจะต้องนำสัญญาณ AEN (Address Enable :output) ซึ่งเป็นสัญญาณที่ใช้สำหรับแยกบัสแอดเดรสในการทำขบวนการ DMA เมื่อสัญญาณนี้แอคทีฟจะทำให้ DMA Controller สามารถควบคุมการทำงานแอดเดรสแทนการควบคุมของ cpu มาต่อร่วมอยู่ด้วยเพื่อไม่ให้เกิดการชนกันของข้อมูลเมื่อ CPU ทำขบวนการ DMA นั่นเอง ขบวนการ DMAเป็นขบวนการที่อุปกรณ์ต่างๆสามารถติดต่อกับหน่วยความจำได้โดยไม่ต้องผ่านการควบคุมของ CPU เป็นการเพิ่มความเร็วของการส่งผ่านข้อมูลระหว่างหน่วยความจำกับอุปกรณ์ไอโอนั้นๆได้ดียิ่งขึ้น

ในส่วนของ 8255 ยังไม่จบเพียงเท่านี้เรายังจะต้องส่งรหัสควบคุม(Control byte) เข้าไปยังพอร์ทควบคุมข้อมูล(port สุดท้ายใน 4 พอร์ท คือที่ A1กับ A0 เป็น 1 ซึ่งได้แก่พอร์ทหมายเลข 310 และ 317 ของ 8255 ทั้งสองตัว) ให้ทำงานในโหมดไหน และให้เป็นอินพุทหรือเอาท์พุทพอร์ท

ความหมายของบิทต่างๆของรหัสควบคุม(control byte) หรือรหัสสั่งงาน 8255 ในตอนเริ่มแรก

D7 - แสดงถึงรหัสควบคุมให้เริ่มทำงาน(1= ทำงาน)คือจะมีผลทำให้ 8255 รับรู้สิ่งต่อไปนี้ บิทต่างๆที่จะกำหนดให้ เพราะฉนั้นเวลาสั่งงานหรือสั่งหน้าที่ให้กับ 8255 บิทนี้จะเป็นลอจิก 1เสมอ

D6,D5 - เป็นการเลือกโหมดการทำงานของport A ซึ่งมีอยู่ 3 modeใน 8255จะได้กล่าวต่อไป

D4 - กำหนดให้ port A เป็นอินพุทหรือเอาท์พุทพอร์ท โดยที่

0 = output port

1 = input port

D3 - กำหนดให้ port C บนเป็นอินพุทหรือเอาท์พุทโดย

0 = output port

1 = input port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D2 - เป็นการเลือกโหมดให้กับพอร์ท B

0 = MODE 0

1 = MODE 1

D1 - กำหนดให้พอร์ท B เป็นอินพุทสหรือเอาท์พุทโดย

0 = output port

1 = input port

D0 - กำหนดให้พอร์ท C ล่างเป็นอินพุทพอร์ทหรือเอาท์พุทพอร์ทโดย

0 = output port

1 = input port

ซึ่งเราอาจจะแสดงได้ดังต่อไปนี้

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	*	*	0	0	*

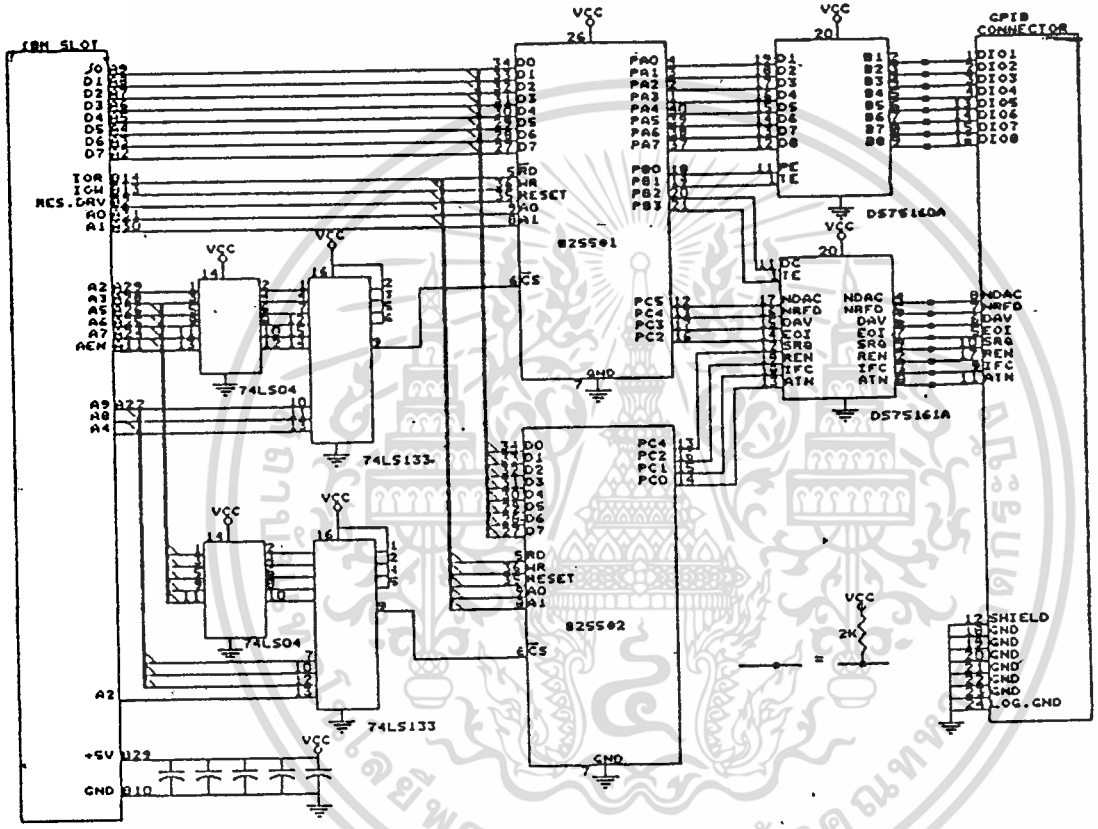
จะเห็นว่าเรายังไม่สามารถระบุได้แน่ชัดว่ารหัสคำสั่งเป็นอะไร ทั้งนี้เพราะพอร์ท A พอร์ท C บนและพอร์ท c ล่างสามารถเปลี่ยนแปลงได้ $2^3 = 8$ ค่าดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	*	*	0	0	*
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1
1	0	0	0	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0
1	0	0	1	0	0	0	1
1	0	0	1	1	0	0	0
1	0	0	1	1	0	0	1

โดยเราจะได้รหัสควบคุมคือ 80h,81h,88h,89h,90h,91h และ 99h ตามลำดับ จากที่ผ่านมาเราสามารถออกแบบ Hardware ของระบบบัสของ GPIB ได้ตามกระบวนกล่าวมาแล้วในตอนต้นทั้งหมด ในตอนต่อไปนี้ก็จะเป็นการรวมวงจรทั้งหมดที่กล่าวถึงมารวมกันวงจรใช้งานจริงในหน้าถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



IEEE-488 GPIB SCHEMETIC DIAGRAM

Size	Drawing By	REV
B	Mr.Jirasak I.,3M. Electrical Eng.	1.0
Date: January 1, 1980	Sheet of	

บทที่ 4

โปรแกรมสนับสนุนของแผ่นวงจรพิมพ์

ภาษาที่ใช้ในการเขียนโปรแกรมได้ใช้เทอร์โบปาสคาล เวอร์ชัน 5.5 โปรแกรมสนับสนุนที่เขียนไว้ทั้งหมดมี 4 โปรแกรม คือ

1. **โปรแกรม GPIB1.PAS** เป็นโปรแกรมจัดการระบบที่ต้องใช้ร่วมกับโปรแกรม USER.INC เวลาใช้งานต้องใช้โปรแกรม TURBO PASCAL VRTTION 5.5 ชื่อโปรแกรม TURBO.EXE มาเป็นตัวคอมไพล์โปรแกรม GPIB1.PAS ก่อนใช้งานทุกครั้ง
2. **โปรแกรม GPIB2.PAS** เป็นโปรแกรมที่ทำงานเหมือน GPIB.PAS แต่จะแสดงสภาวะการทำงานต่างๆทุกขั้นตอนของแต่ละคำสั่งที่เขียนใน USER.INC
3. **โปรแกรม GPIB3.EXE** เป็นโปรแกรมที่จะเขียนเป็นระบบเมนู เวลาใช้งานไม่ต้องใช้โปรแกรม TURBO.EXE มาคอมไพล์เหมือนกับสองโปรแกรมแรกที่กล่าวมาแล้ว ที่โปรแกรมนี้จะมีการแสดงสภาวะการทำงานต่างๆตามเมนูที่เราเลือก
4. **โปรแกรม GPIB4.EXE** เป็นโปรแกรมที่สร้างขึ้นเหมือนกับโปรแกรม GPIB3.EXE แต่จะไม่แสดงรายละเอียดของสภาวะการทำงานต่างๆทำให้การทำงานรวดเร็วขึ้นมาก

ในบทนี้จะอธิบายการทำงานและการใช้งานของโปรแกรม GPIB1.PAS(GPIB2.PAS มีการทำงานและการใช้งานเหมือน GPIB1.PAS จึงไม่อธิบายอีก) ส่วน GPIB4.EXE จะอธิบายการใช้งานในบทที่ 5 (GPIB3.EXE มีการทำงานและการใช้งานคล้าย GPIB4.EXE ดังนั้นจึงไม่อธิบาย)

โปรแกรมสนับสนุน GPIB.PAS นี้แบ่งออกเป็น 2 ส่วน คือ

4.1 ส่วนของโปรแกรมจัดการระบบ(System Program)

4.2 ส่วนของโปรแกรมของผู้ใช้งาน(Usre Program)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดมีดังต่อไปนี้

4.1 ส่วนของโปรแกรมจัดการระบบ (System Program)

ส่วนของโปรแกรมจัดการระบบเขียนเก็บไว้ในไฟล์ชื่อ GPIB1.PAS ซึ่งเป็นส่วนที่ใช้ในการควบคุมการทำงานของแผ่นวงจรให้เป็นไปตามมาตรฐาน GPIB โดยผู้ใช้ไม่จำเป็นต้องเข้ามาแก้ไขหรือเพิ่มเติมโปรแกรมในส่วนนี้ เพียงแต่ศึกษาการใช้แต่ละ Procedure ว่าใช้ทำอะไรและเรียกใช้ให้ถูกต้องก็พอ

รายละเอียดของโปรแกรมจัดการระบบ มีดังนี้คือ

ส่วนข้อกำหนด (Declaration Part) จะเป็นการกำหนดค่าต่างๆและตัวแปรสำหรับการนำไปใช้ในส่วนอื่นๆของโปรแกรม

ส่วนถ้อยแถลง (Statement Part) จะแบ่งออกเป็นหลายๆ Procedure ทำงานในหน้าที่ต่างๆกัน เพื่อความสะดวกในการใช้งาน ดังมีรายละเอียดการทำงานของแต่ละ Precedure มีดังต่อไปนี้

```
Procedure Address(Addr : integer; var MLA,MTA :byte);
```

ใน Procedure นี้จะเป็นการแปลงค่าแอดเดรสของอุปกรณ์ที่ตั้งค่าไว้ในระบบจากเลขฐานสิบ(เพื่อความสะดวกของผู้ใช้) ไปเป็นค่า MLA และ MTA(ซึ่งเป็นค่าที่กำหนดไว้ในมาตรฐาน) แล้วส่งค่าที่ได้คือ MLA(My Listen Address) และ MTA(My Talk Adress) กลับไปยัง Procedure ต่างๆที่เรียกใช้มันมี Flow Chart การทำงานดังรูปที่ 4-1

```
Procedure Source_Operation(Data : byte);
```

ใน Procedure นี้ เป็นการทำขบวนการ Source Operation สำหรับการส่งข้อความคือ Data (ข้อความ 1 ไบท์) มี Flow Chart การทำงานดังรูปที่ 4-2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Procedure Init_Source;

ใน Procedure นี้จะทำการกำหนดโหมดการทำงานของ 8255,75160A และ 75161A ในขณะเริ่มต้น ซึ่งเป็นส่วนที่ผู้ใช้จำเป็นจะต้องเรียกใช้เป็นอันดับแรกในโปรแกรมของผู้ใช้งาน มี Flow Chart การทำงานดังรูปที่ 4-3

Procedure Send_Command (Addr : Integer;Command : byte);

ใน Procedure นี้จะใช้สำหรับการส่งคำสั่งเพื่อควบคุมอุปกรณ์ให้เป็นไปตามต้องการ มี Flow Chart การทำงานดังรูปที่ 4-4

Procedure Send_Message (Addr : Integer;M : DataFormat);

ใน Procedure จะใช้สำหรับการส่งข้อความ(หลายๆไบท์) ที่ใช้ในการควบคุมอุปกรณ์ให้เป็นๆไปตามต้องการ มี Flow Chart การทำงานดังรูปที่ 4-5

Procedure Acceptor_Opertion(var DByte : byte);

ใน Procedure เป็นการทำขบวนการ Acceptor Operation สำหรับการรับข้อความทีละ 1 ไบท์ มี Flow Chart การทำงานดังรูปที่ 4-6

Procedure Get_Data (Addr : Inter;Var D:DataFormat;Num : integer);

ใน Procedure นี้ใช้ในการรับข้อมูลหลายๆไบท์กลับสู่คอมพิวเตอร์ (ทำหน้าที่เป็น Controller) และอ่านค่าเวลาขณะนั้นเข้ามาด้วย มี Flow Chart การทำงานดังรูปที่ 4-7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Procedure IFC;

ใน Procedure นี้ใช้สำหรับเคลียร์สัญญาณควบคุมทั้งหมด ให้กลับสู่สภาวะเริ่มต้นมี Flow Chart การทำงานดังรูปที่ 4-8

Procedure DCL;

ใน Procedure นี้ใช้สำหรับทำให้อุปกรณ์ทุกตัวที่ต่ออยู่ในระบบ GPIB กลับสู่สภาวะเริ่มต้น มี Flow Chart การทำงานดังรูปที่ 4-9

Procedure SPolling(Addr : integer ; var DByte : byte);

ใน Procedure นี้ใช้สำหรับการทำขบวนการตรวจสอบแบบอนุกรม (Serial Polling) มี Flow Chart การทำงานดังรูปที่ 4-10

Procedure Save-Data (D : DataFormat);

ใน Procedure นี้ใช้สำหรับการนำข้อมูลที่ได้จากอุปกรณ์และเวลาขณะที่รับข้อมูลเก็บเข้าไฟล์ที่ต้องการไว้ในแผ่นดิสเกตต์ โดยไฟล์ข้อมูลเหล่านี้จะมีชื่อของผู้ใช้มี Flow Chart การทำงานดังรูปที่ 4-11

{$I USER.inc}

ในส่วนนี้ จะเป็นส่วนที่ใช้ในการติดต่อกับโปรแกรมที่ผู้ใช้เขียนขึ้น โดยเก็บไว้ในไฟล์ชื่อ USER.INC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ส่วนของโปรแกรมของผู้ใช้งาน (User Program)

ส่วนของโปรแกรมของผู้ใช้งานนี้ ผู้ใช้งานจะต้องเขียนโปรแกรมขึ้นมาเองโดยเขียนอยู่ในไฟล์ชื่อ User.Inc เท่านั้น ในการเขียนโปรแกรมส่วนนี้ ผู้ใช้จะต้องศึกษาและจะต้องเข้าใจระบบบัสมาตรฐาน GPIB นี้พอสมควร

รูปแบบคำสั่งที่ได้กำหนดไว้ในส่วนของโปรแกรมจัดการระบบ ผู้ใช้สามารถเรียกใช้ได้โดยอาศัยรูปแบบต่อไปนี้

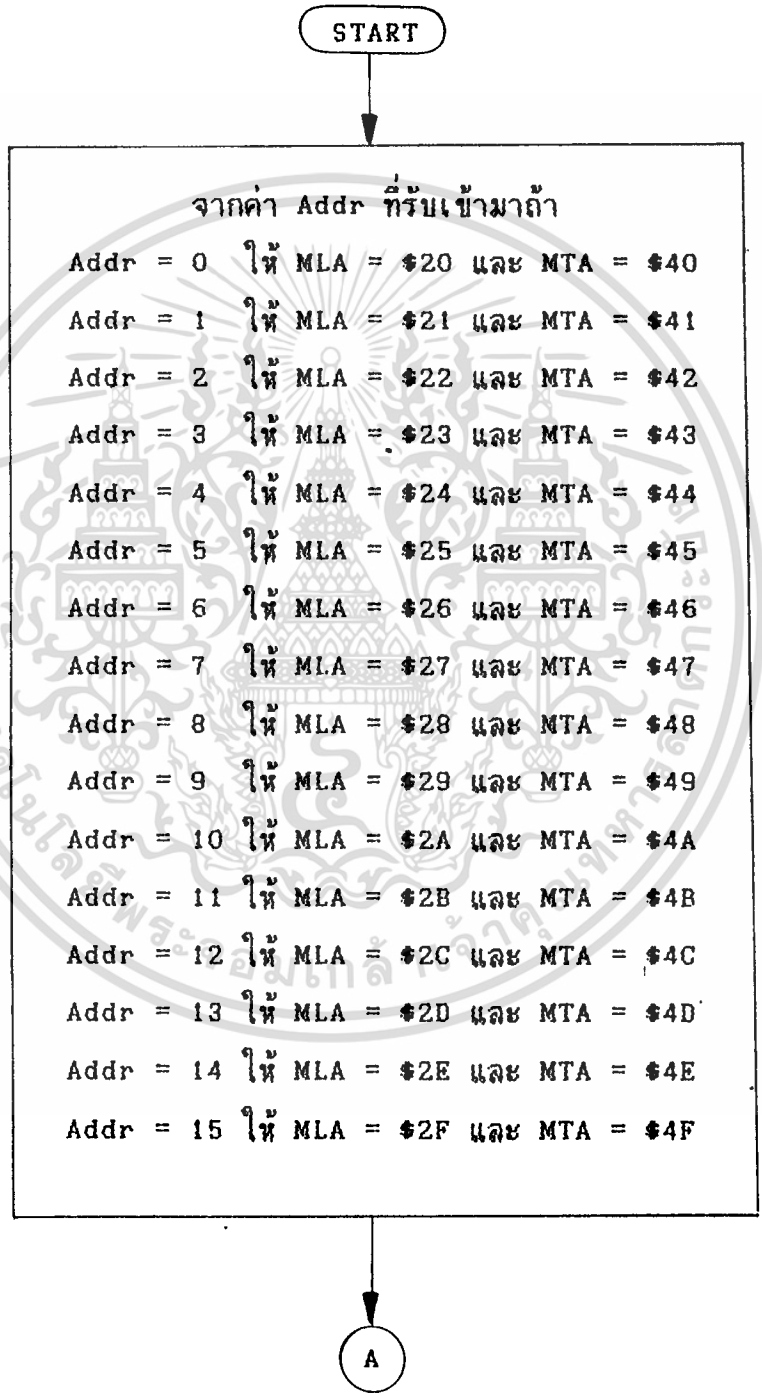
Device Clear

รูปแบบ DCL;
จุดประสงค์ คำสั่งนี้ใช้ในการสั่งให้อุปกรณ์ในระบบกลับสู่สภาวะเริ่มต้น
สัญญาณในบัส REN = ROW
IFC = HIGH

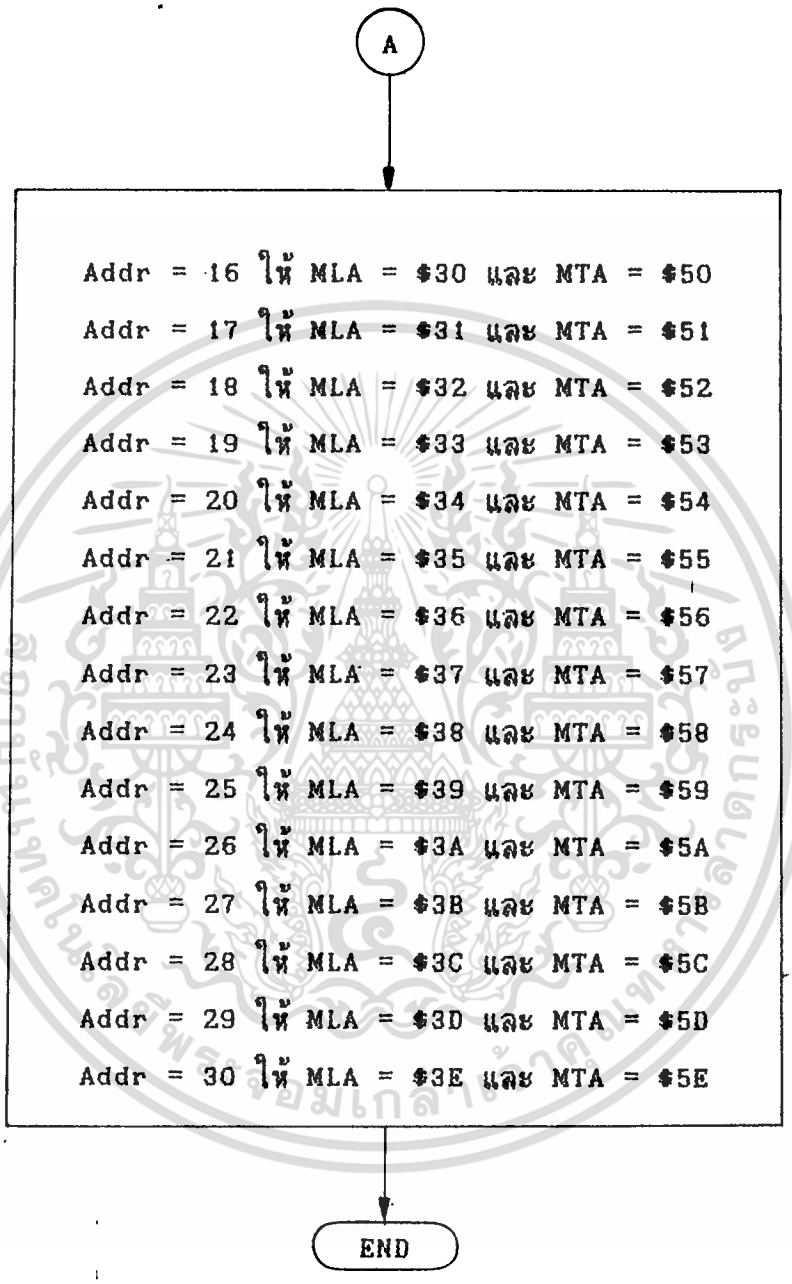
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-1 แสดง Flow Chart และ Procedure Address

```
Procedure Address (Addr : integer;var MLA,MLA : byte);
```



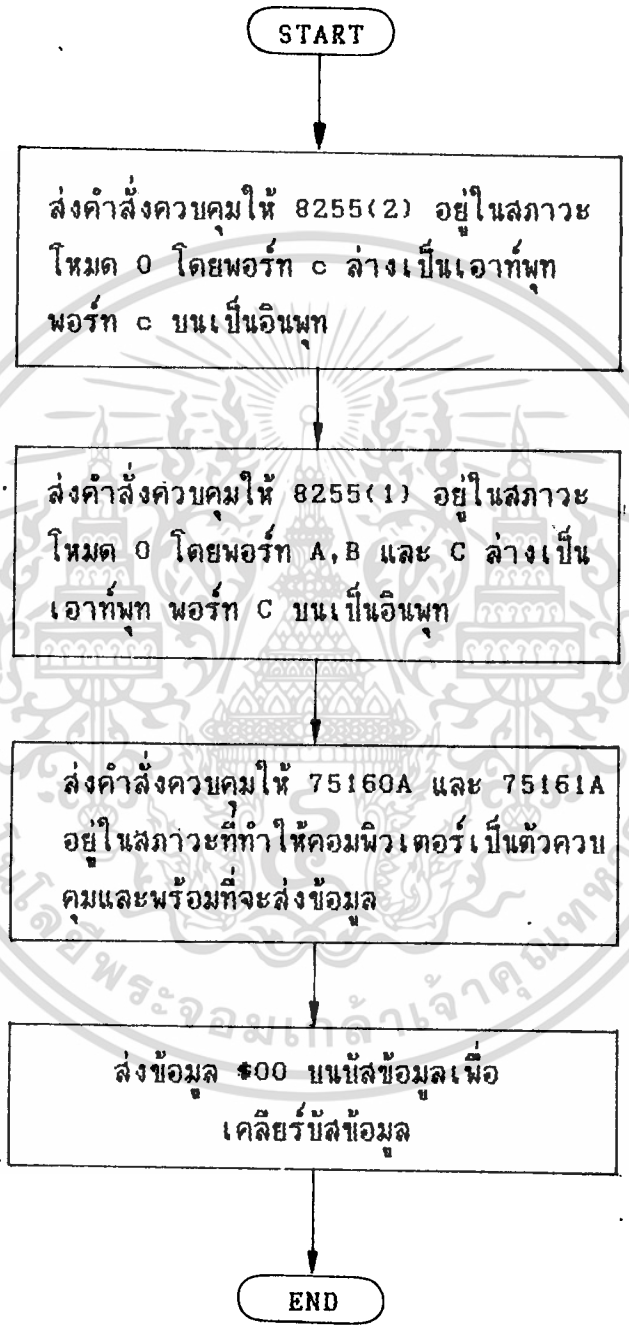
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-3 แสดง Flow Chart และ Procedure Init_Source

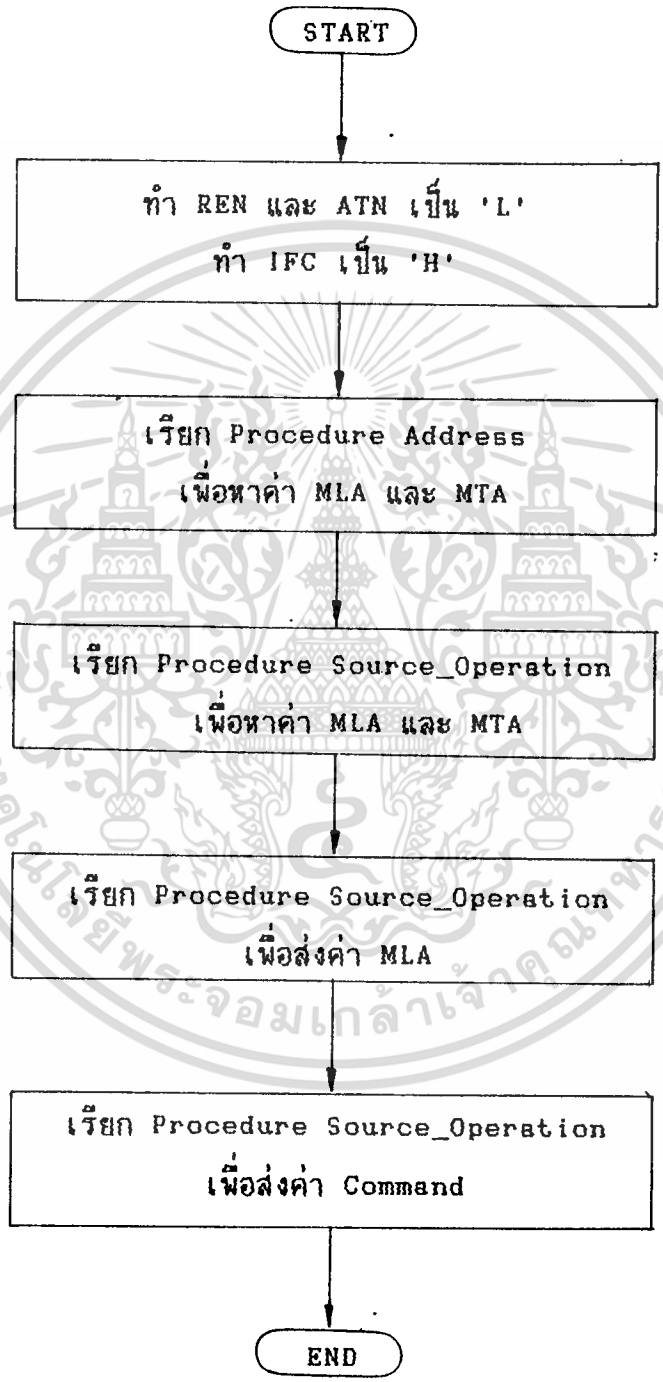
Procedure Init_Source;



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-4 แสดง Flow Chart ของ Procedure Send_Command

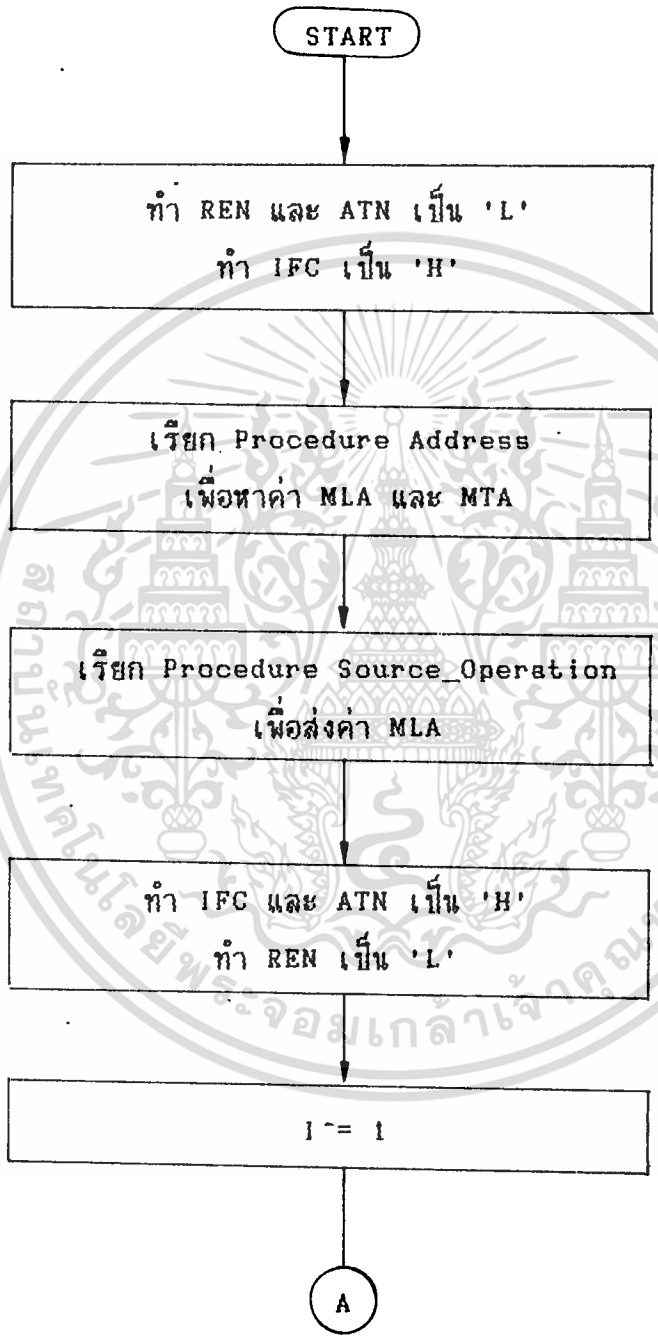
Send_Command (Addr : integer ; Command : byte);



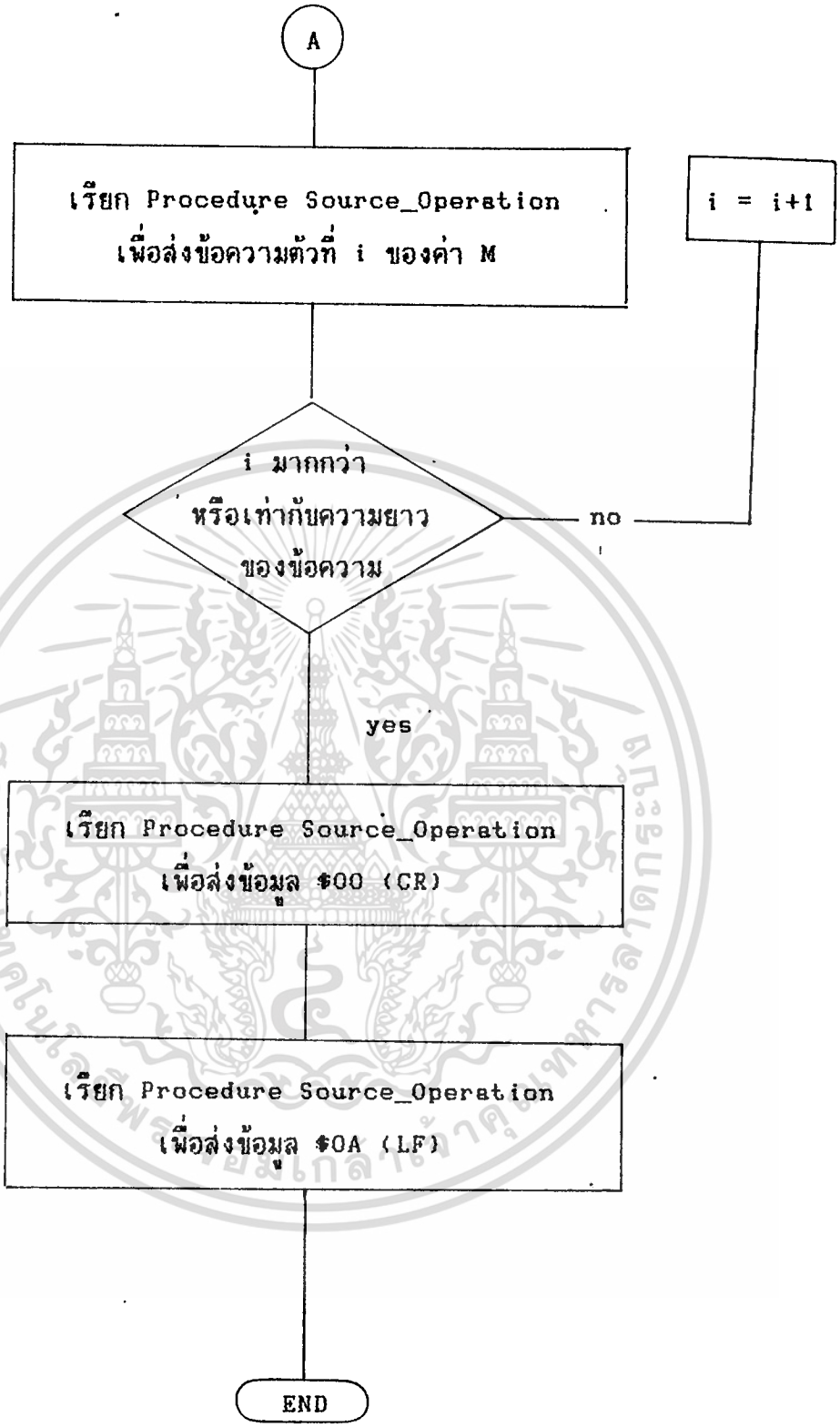
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4_5 แสดง Flow Chart และ Procedure Send_Message

```
Procedure Send_Message (Adde : integer;M : Data);
```



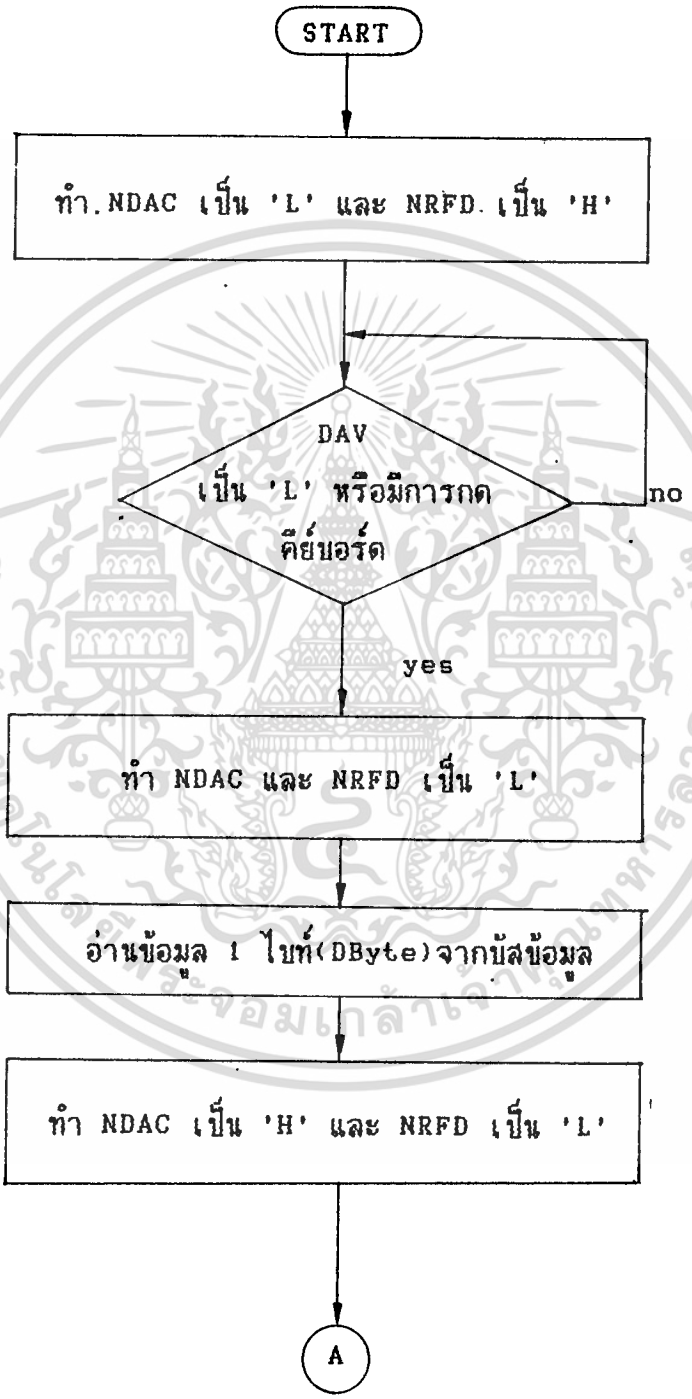
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



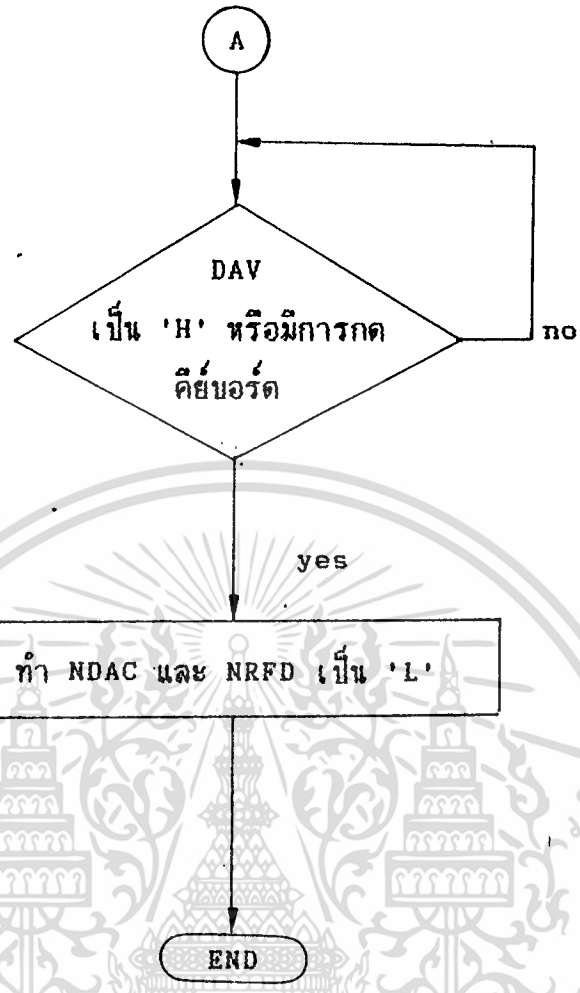
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-6 แสดง Flow Chart และ Procedure Acceptor_Operation

```
Procedure Acceptor_Operation (var DByte : byte);
```



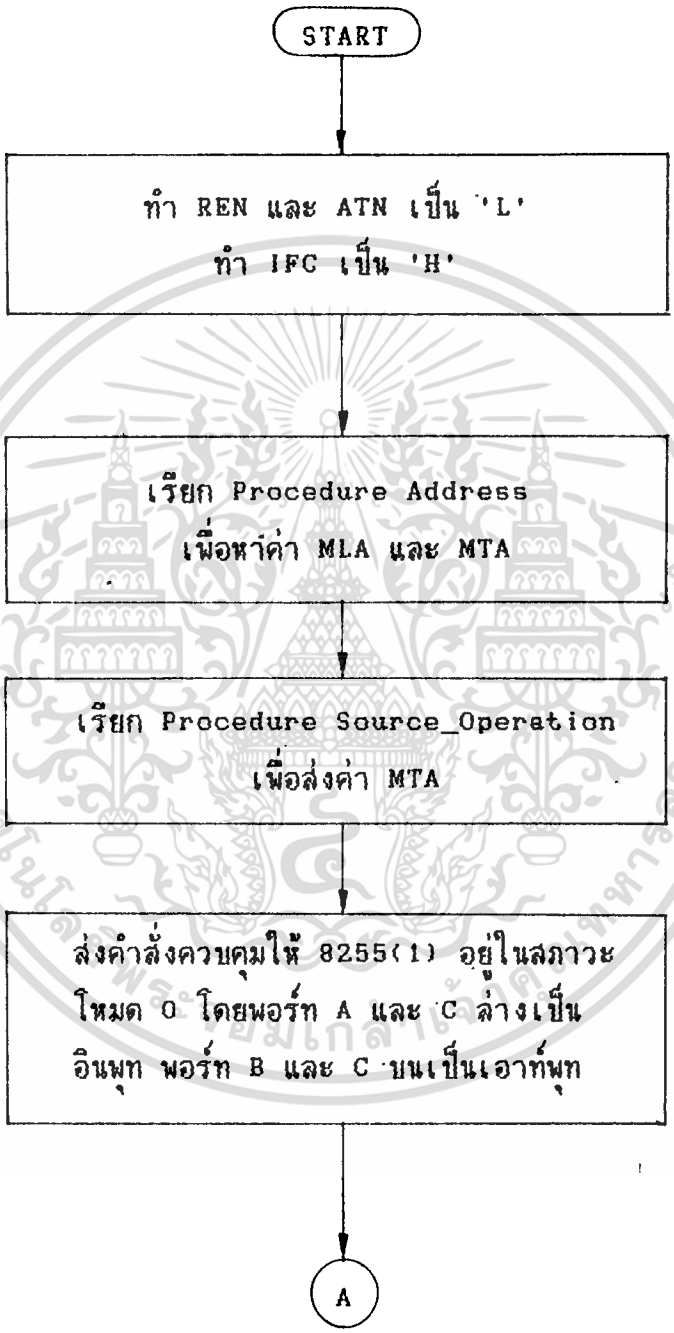
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



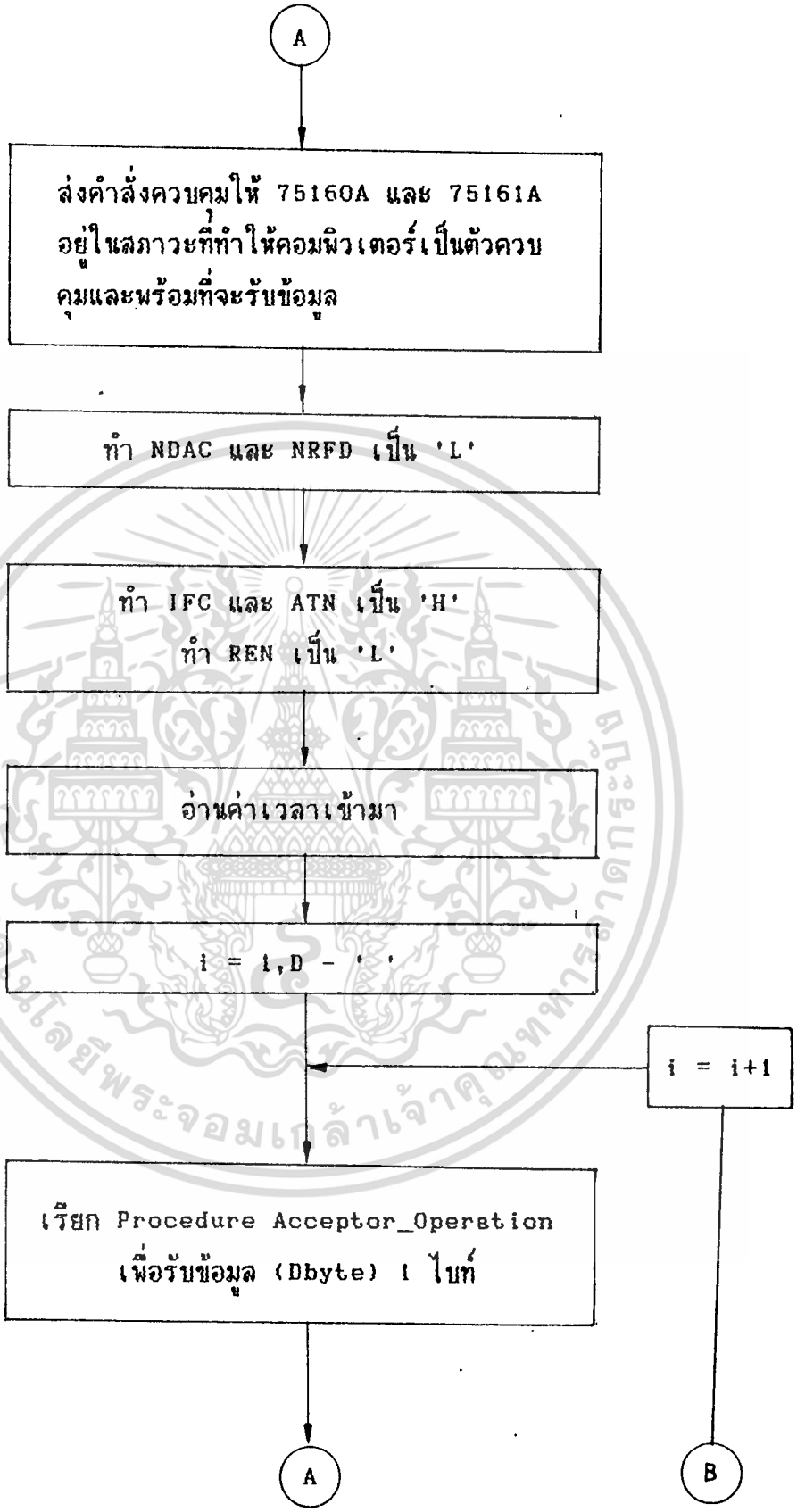
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-7 แสดง Flow Chart ของ Procedure Get_Data

```
Procedure Get_Data (Addr : integer;var D : DataFormat;Num : integer);
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

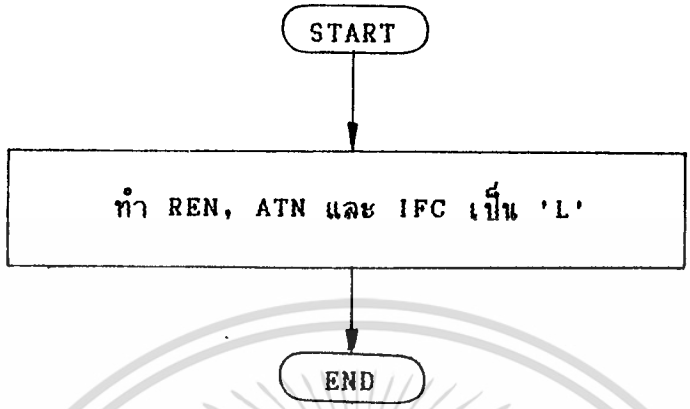


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

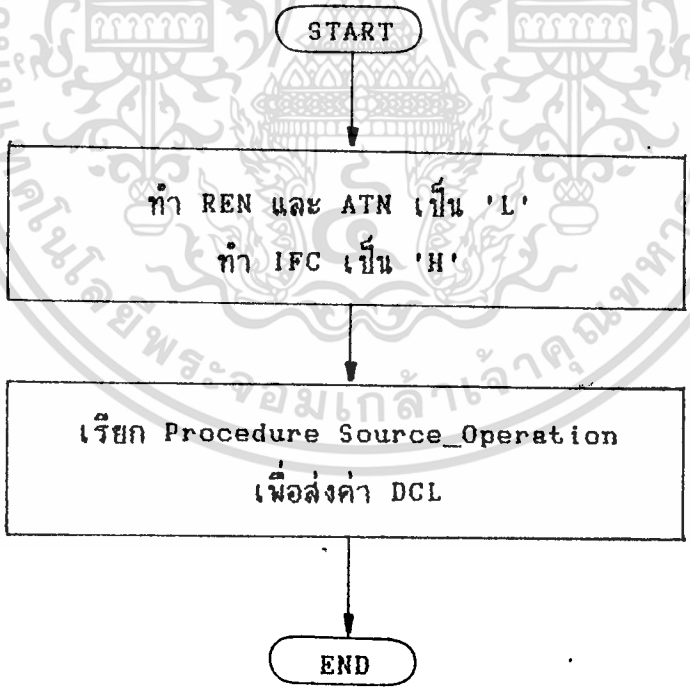
รูปที่ 4-8 แสดง Flow Chart ของ Procedure IFC

Procedure IFC;



รูปที่ 4-9 แสดง Flow Chart ของ Procedure DCL;

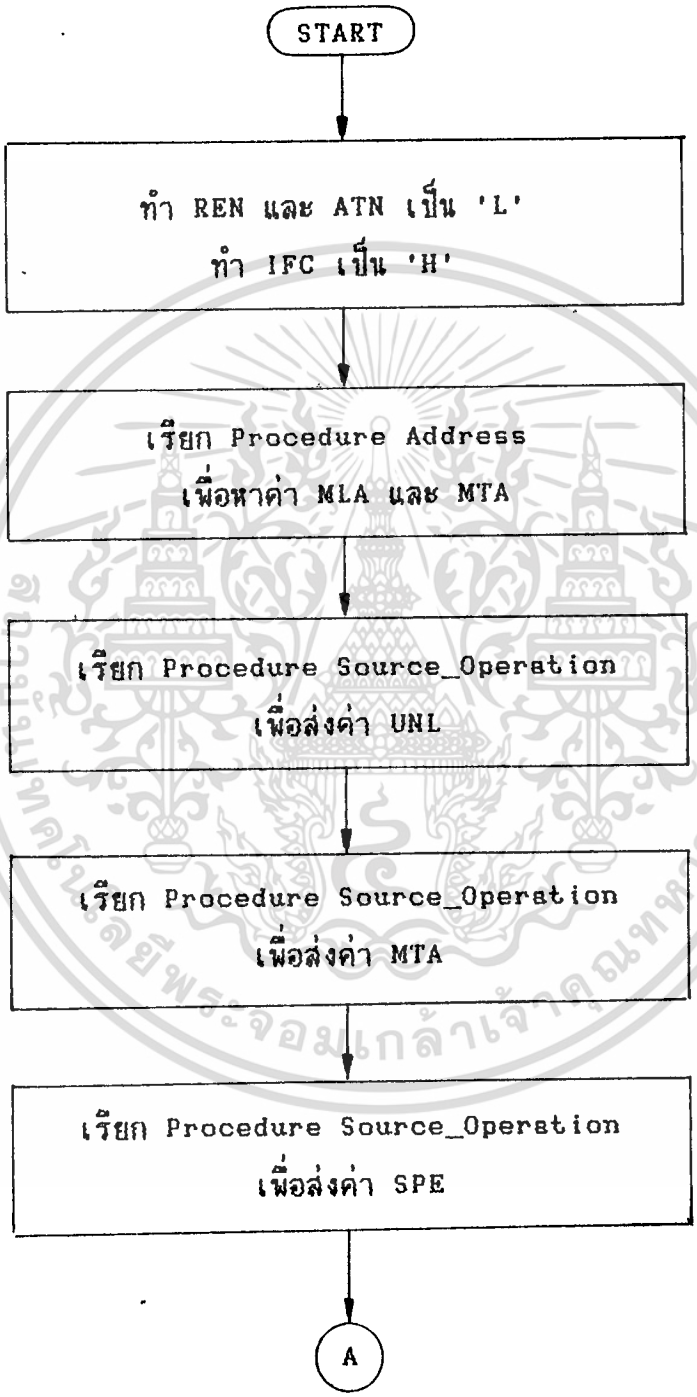
Procedure DCL;



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-10 แสดง Flow Chart ของ Procedure Spolling

Procedure Spokking (Addr : integer;var Dbyte:byte);



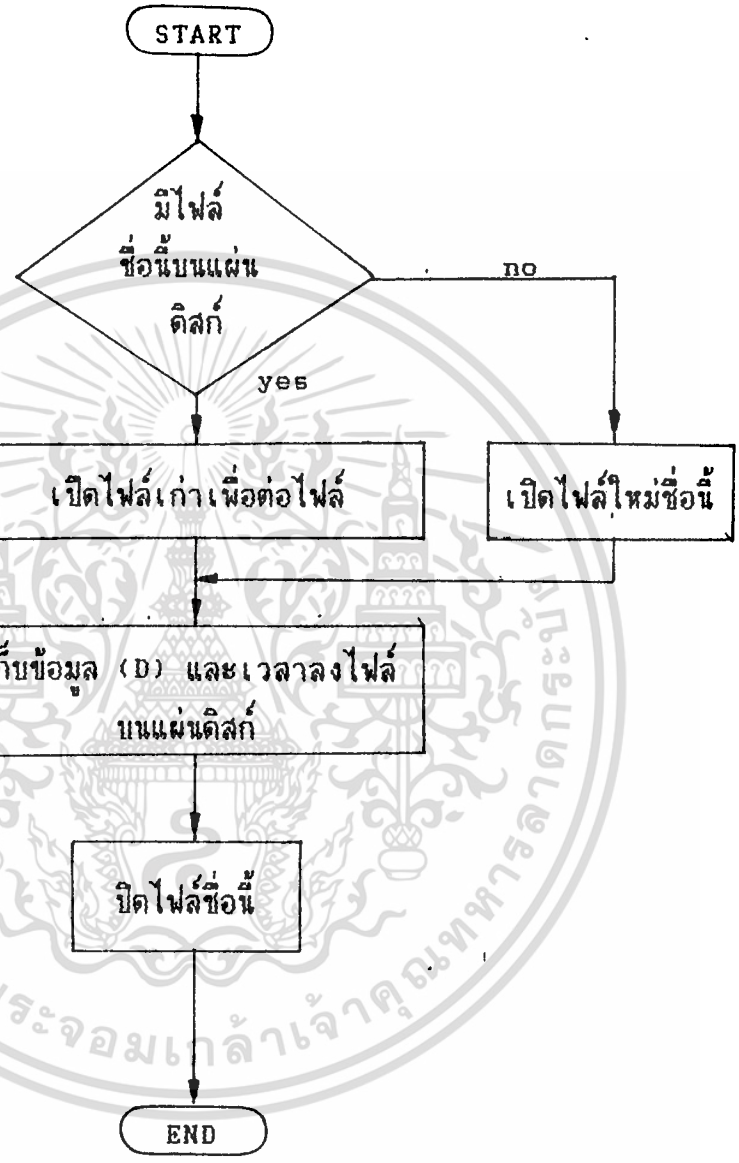
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-11 แสดง Flow Chart ของ Procedure Save_Data

```
Procedure Save_Data(D:DataFormat;Filename : str30);
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

วิธีการใช้โปรแกรม GPIB4.EXE

เนื่องด้วยทางผู้จัดทำเห็นว่าโปรแกรม GPIB1 มีความยุ่งยากแก่การใช้งานเพราะการใช้งานในแต่ละครั้งต้องให้ผู้เขียนโปรแกรม USER.INC พร้อมทั้งต้องคอมไพล์ใหม่ทั้งโปรแกรม ถ้าผู้ใช้ไม่มีความรู้ภาษาปาสคาลก็ไม่สามารถใช้งานได้ ดังนั้นเพื่อความสะดวกแก่ผู้ที่นำไปใช้งาน ทางผู้จัดทำจึงได้คิดทำเป็นโปรแกรมสำเร็จรูปขึ้นมาซึ่งได้ตั้งชื่อโปรแกรมนี้ว่า GPIB4.EXE

โปรแกรม GPIB4 ประกอบด้วย 4 เมนูหลักดังนี้คือ

1. GPIB MENU
2. FILE MENU
3. OPTION MENU
4. QUIT MENU

การเลือกใช้แต่ละเมนูหลัก ทำได้โดยกดคีย์ลูกศรซ้ายหรือขวาเพื่อเลื่อนแถบสีไปยังเมนูที่ต้องการ (ซึ่งในแต่ละเมนูหลักที่แถบสีปรากฏอยู่นั้น จะมีรายละเอียดแสดงอยู่ที่บรรทัดล่างสุดของจอ) แล้วกดคีย์ ENTER หรืออีกวิธีหนึ่งคือ กดตัวอักษรตัวแรกของแต่ละเมนูหลักที่ต้องการเลือกใช้ เมื่อขณะที่เลือกใช้เมนูหลักใดแล้ว ก็สามารถเลือกใช้เมนูหลักอื่นๆได้ โดยกดคีย์ลูกษรซ้ายหรือขวา และขณะที่กำลังใช้เมนูหลักใดแล้ว ก็สามารถเลือกใช้เมนูย่อยใดๆ ผู้ใช้สามารถกดคีย์ ESC เพื่อออกจากเมนูหลักหรือเมนูย่อยใดๆ ได้ทุกเวลาที่ต้องการ ในเมนูย่อยนั้นบางครั้งจะมีการให้ใส่ข้อมูลลงไป ผู้ใช้สามารถใส่ข้อมูลลงไปได้เลย และถ้าต้องการแก้ไขก็สามารถแก้ไขได้โดย กดคีย์ Ctrl และอักษร Y พร้อมกันเพื่อลบทั้งบรรทัดหรือกดคีย์ลูกศรขึ้นหรือลง คีย์ใดก็ได้ 1 คีย์ ตัวอักษรที่อยู่บรรทัดนั้นจะกระพริบ ผู้ใช้สามารถใส่ข้อมูลลงไปใหม่ได้เลยโดยมันจะลบตัวอักษรเก่าออกจากบรรทัดนั้นเอง ถ้าผู้ใช้ต้องการลบตัวอักษรตัวสุดท้ายสามารถทำได้โดยกดคีย์ Ctrl และอักษร H พร้อมกันหรือคีย์ Backspace และถ้าผู้ใช้ต้องการแทรกข้อความสามารถทำได้ โดยเลื่อนคีย์ลูกศรซ้ายหรือขวาไปยังตำแหน่งที่ต้องการแทรก แล้วกดตัวอักษรที่ต้องการแทรกลงไปได้เลย เมื่อผู้ใช้กรอกข้อมูลเสร็จก็ให้กดคีย์ ENTER ทุกครั้ง เพื่อให้เครื่องรับข้อมูลเข้าไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของแต่ละข้อมูลหลักมีดังต่อไปนี้

5.1 **GPIB MENU** เมื่อเลือก GPIB MENU จะมีหน้าต่าง(WINDOW) เกิดขึ้นใหม่ใต้คำว่า GPIB MENU จะประกอบด้วยคำสั่งต่างๆดังนี้คือ

5.1.1 Init.source สามารถเลือกใช้โดยการ เลื่อนลูกศรขึ้นหรือลงหรือกดตัวอักษร I ก็ได้ เพื่อให้แถบสีมาที่คำสั่งนี้ แล้วกดคีย์ ENTER ซึ่งที่คำสั่งนี้จะไปเรียกโปรแกรมย่อย Init_Source ให้ทำงาน

5.1.2 IFC สามารถเลือกใช้โดยการ เลื่อนลูกศรขึ้นหรือลง หรือกดอักษร I ก็ได้เพื่อให้แถบสีมาที่คำสั่งนี้แล้วกดคีย์ ENTER ซึ่งที่คำสั่งนี้จะไปเรียกโปรแกรมย่อย IFC ให้ทำงาน

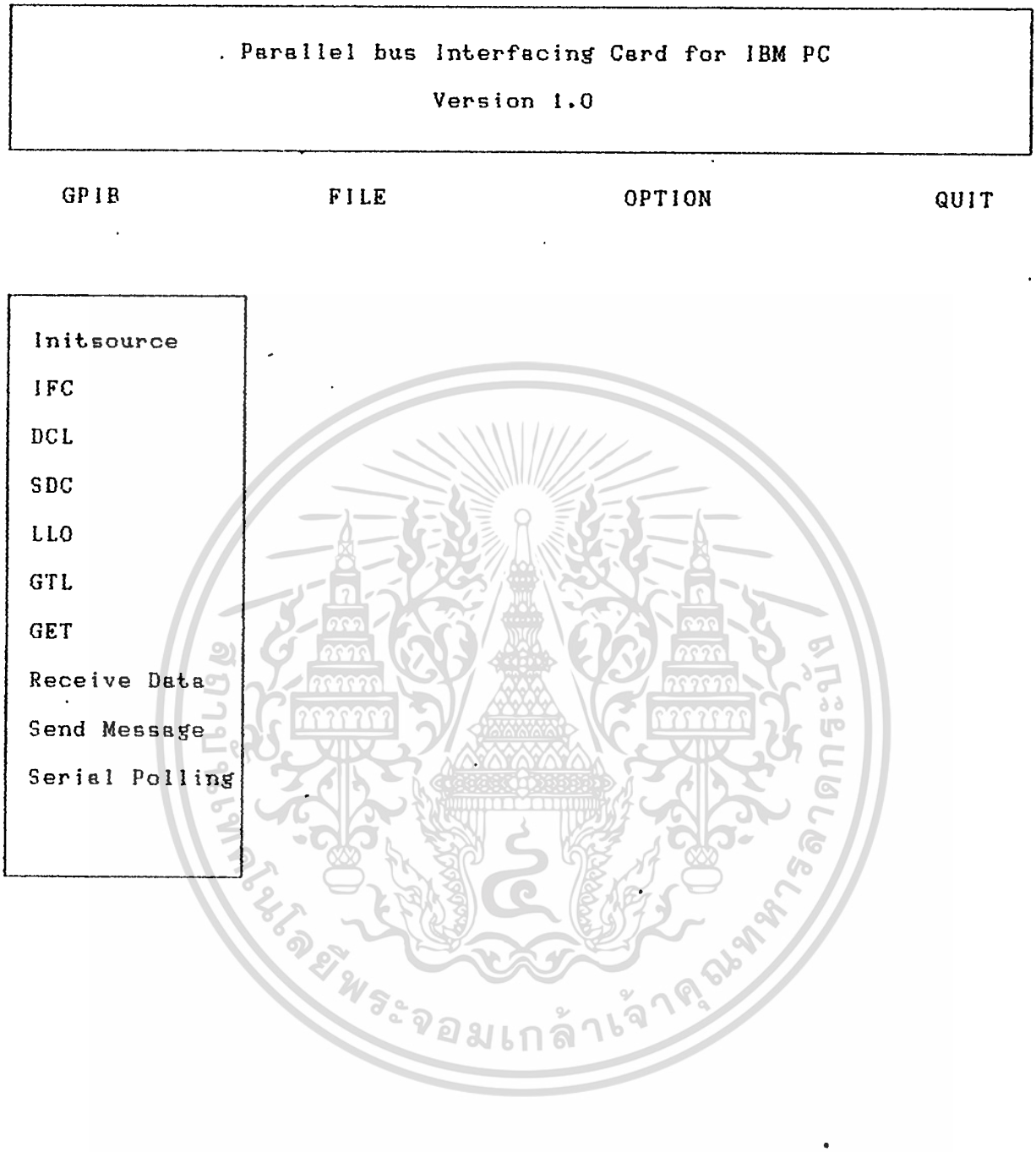
5.1.3 DCL สามารถเลือกใช้โดยการ เลื่อนลูกศรขึ้นหรือลง หรือกดอักษร D ก็ได้เพื่อให้แถบสีมาที่คำสั่งนี้ แล้วกดคีย์ ENTER ซึ่งที่คำสั่งนี้จะไปเรียกโปรแกรมย่อย DCL ให้ทำงาน ถ้ามีการเกิด BUS ERROR ขึ้น จะมีเสียงเตือนพร้อมทั้งมีตัวอักษร "BUS ERROR !" ปรากฏขึ้นที่มุมล่างทางขวาของจอ ดังรูปที่ 5-2 และจะยกเลิกคำสั่ง DCL

5.1.4 SCD สามารถเลือกใช้โดยการ เลื่อนลูกศรขึ้นหรือลง หรือกดอักษร S ก็ได้เพื่อให้แถบสีมาที่คำสั่งนี้ แล้วกดคีย์ ENTER ต่อมาจะกแสดงหน้าต่างขึ้นมาใหม่ให้ใส่ค่าแอดเดรส (Address) ดังรูปที่ 5-3 เมื่อใส่ค่าแอดเดรสแล้ว ก็จะไปเรียกโปรแกรมย่อย Send_Command เพื่อให้ส่งคำสั่ง SDC ออกไป

5.1.5 LLO สามารถเลือกใช้โดยการ เลื่อนลูกศรขึ้นหรือลง หรือกดอักษร L ก็ได้เพื่อให้แถบสีมาที่คำสั่งนี้ แล้วกดคีย์ ENTER ต่อมาจะแสดงหน้าต่างขึ้นมาใหม่ให้ใส่ค่าแอดเดรส (Address) เมื่อใส่ค่าแอดเดรสแล้ว ก็จะไปเรียกโปรแกรมย่อย Send_Command เพื่อให้ส่งคำสั่ง LLO ออกไป

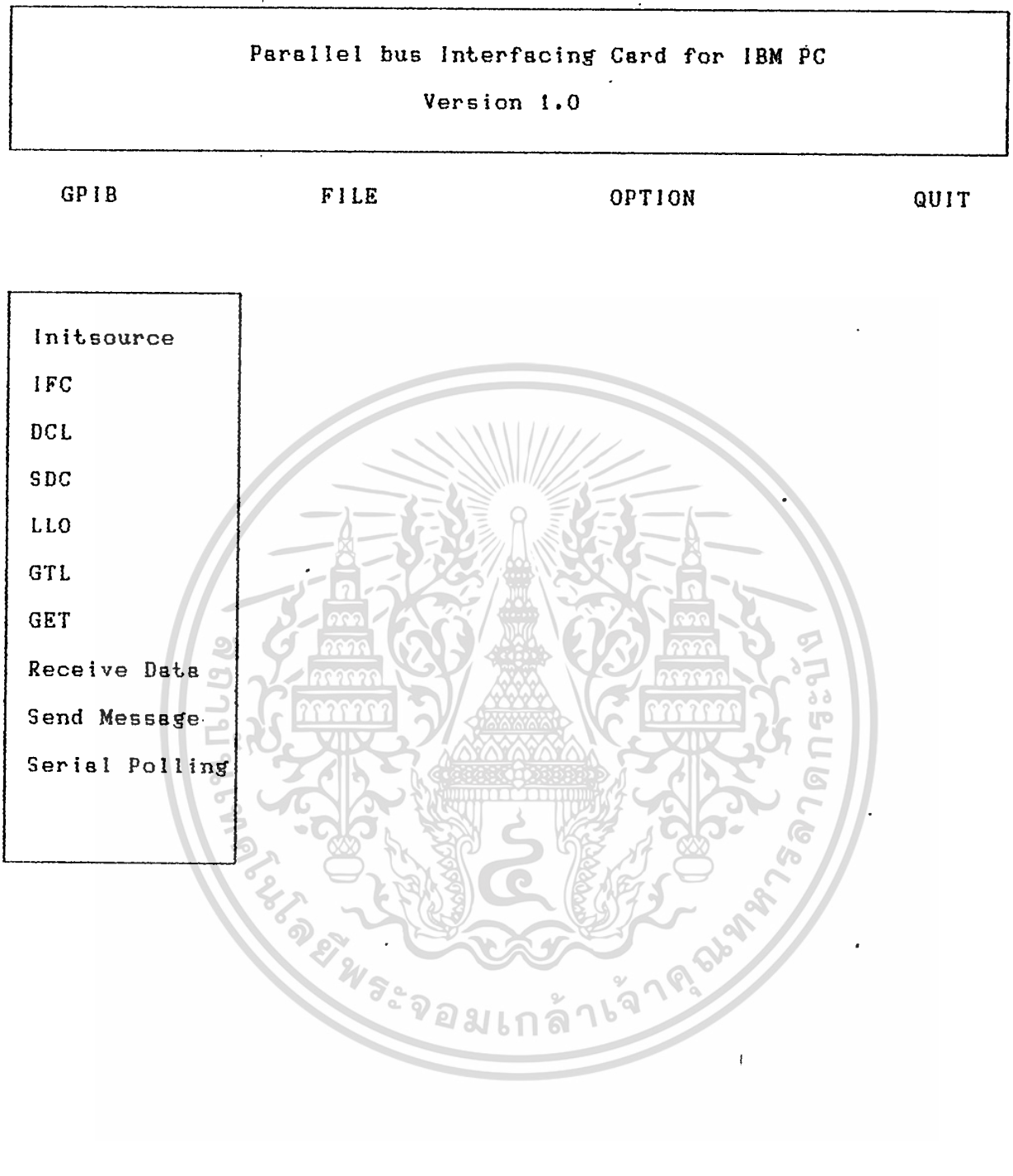
5.1.6 GTL สามารถเลือกใช้โดยการ เลื่อนลูกศรขึ้นหรือลง หรือกดอักษร G ก็ได้เพื่อให้แถบสีมาที่คำสั่งนี้ แล้วกดคีย์ ENTER ต่อมาจะแสดงหน้าต่างขึ้นมาใหม่ให้ใส่ค่าแอดเดรส (Address) เมื่อใส่ค่าแอดเดรสแล้ว ก็จะไปเรียกโปรแกรมย่อย Send_Command เพื่อให้ส่งคำสั่ง GTL ออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



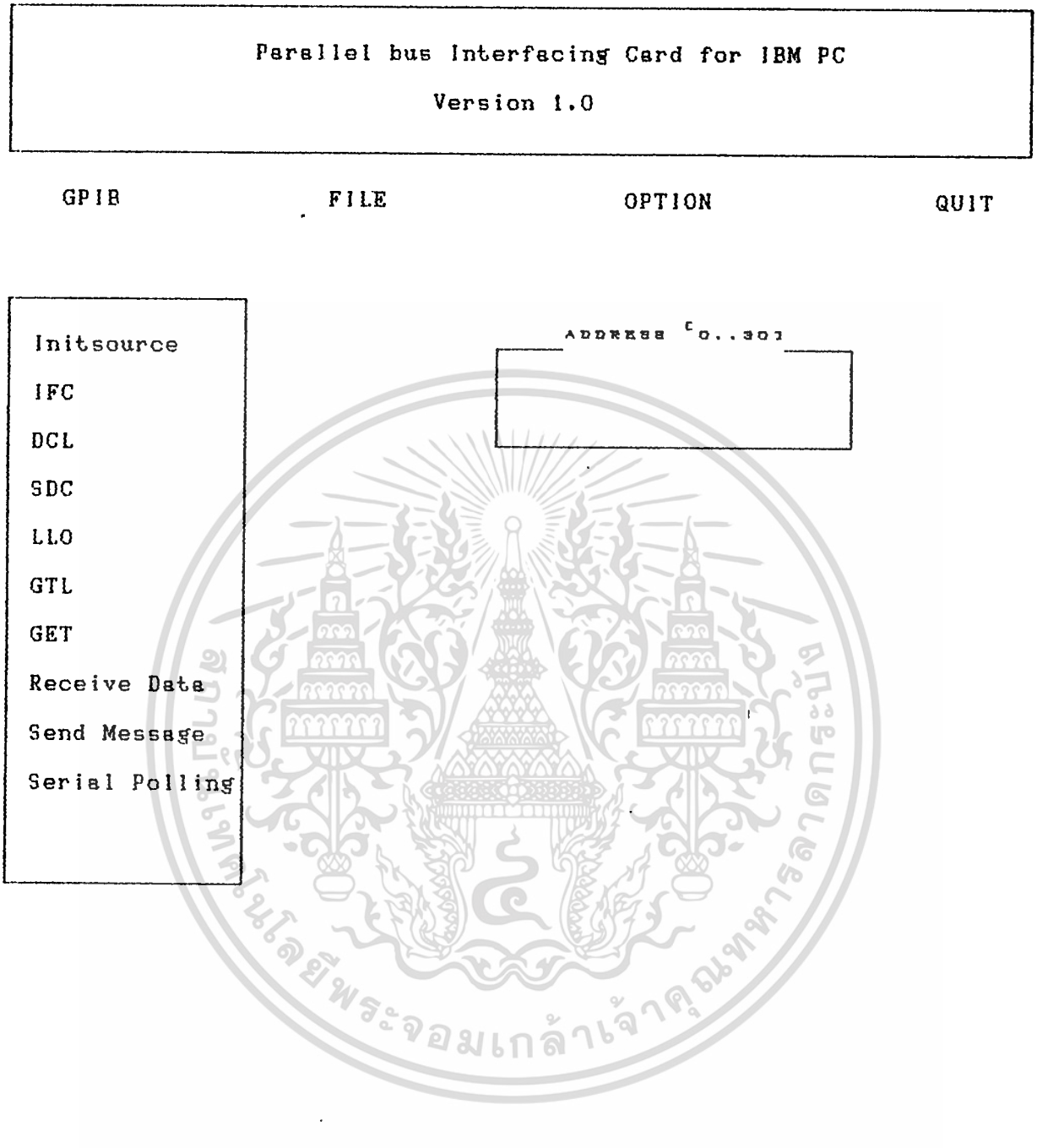
รูปที่ 5.1 แสดงภาพขณะจอเมื่อเลือก GPIB MENU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 แสดงภาพขณะจอภาพเกิด BUS ERROR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 แสดงหน้าต่างที่ให้ใส่ค่าแอดเดรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

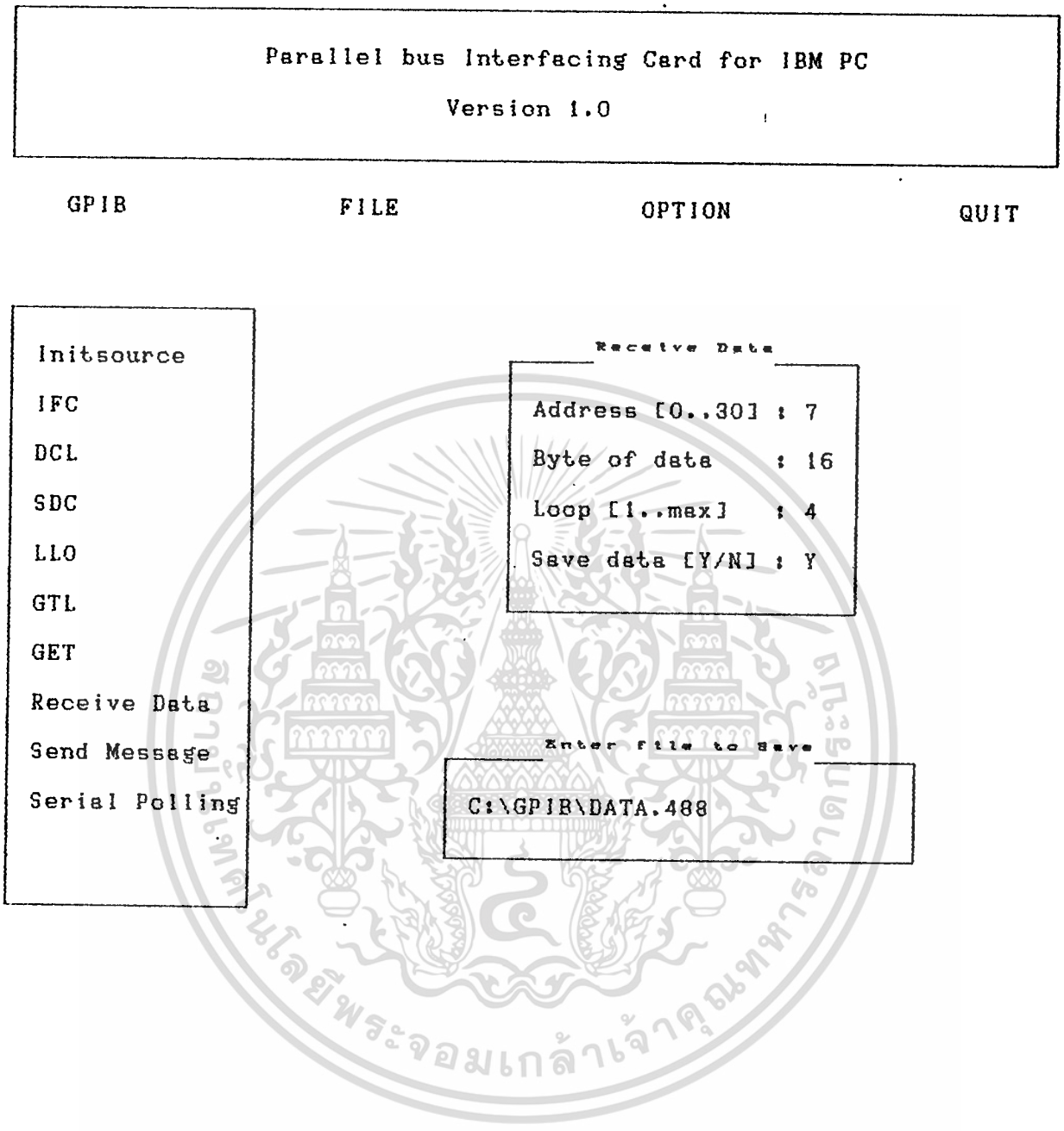
5.1.7 GET สามารถเลือกใช้โดยการ เลื่อนลูกศรขึ้นหรือลง หรือกดอักษร G ก็ได้เพื่อให้แถบสีมาที่คำสั่งนี้ แล้วกดคีย์ ENTER ต่อมาจะแสดงหน้าต่างขึ้นมาใหม่ให้ใส่ค่าแอดเดรส (Address) เมื่อใส่ค่าแอดเดรสแล้ว ก็จะไปเรียกโปรแกรมย่อย Send_Command เพื่อให้ส่งคำสั่ง GET ออกไป

5.1.8 RECEIVE DATA สามารถเลือกใช้โดยการ เลื่อนลูกศรขึ้นหรือลง หรือกดอักษร R ก็ได้เพื่อให้แถบสีมาที่คำสั่งนี้ แล้วกดคีย์ ENTER ต่อมาจะแสดงหน้าต่างขึ้นมาใหม่ให้ใส่ค่าแอดเดรส (Address),จำนวนข้อมูล(Bytes of data) จำนวนลูป(Loop) เมื่อใส่ค่าครบแล้ว ก็จะมีการถามว่าต้องการเก็บข้อมูลลงแผ่นดิสก์หรือไม่ ถ้าต้องการเก็บข้อมูลลงแผ่นดิสก์ ก็ให้กดตัวอักษร Y แล้วจะแสดงหน้าต่างขึ้นมาใหม่ให้ใส่ชื่อไฟล์ ดังรูปที่ 5-4 หรือถ้าไม่ต้องการเก็บข้อมูลลงแผ่นดิสก์ ก็ให้กดตัวอักษร N ต่อมาก็จะไปเรียกโปรแกรมย่อย Get_Data ให้ทำงานพร้อมทั้งอ่านเวลาขณะนั้นแล้ว จะแสดงหน่าต่างขึ้นมาใหม่เพื่อแสดงข้อมูลที่ได้พร้อมทั้งเวลา ดังรูปที่ 5-5

5.1.9 Send Message สามารถเลือกใช้โดยการ เลื่อนลูกศรขึ้นหรือลง หรือกดอักษร S ก็ได้ เพื่อให้แถบสีมาที่คำสั่งนี้ แล้วกดคีย์ ENTER ต่อมาจะแสดงหน้าต่างขึ้นมาใหม่ให้ใส่ค่าแอดเดรส (Address) เมื่อใส่ค่าแอดเดรสแล้วจะแสดงหน้าต่างขึ้นมาใหม่ให้ใส่ข้อความ (Message) ไม่เกิน 50 ตัวอักษรที่ต้องการส่งไปตั้ง(set) การทำงานของเครื่องที่อยู่ที่แอดเดรสนั้น ดังรูปที่ 5-6 ต่อมาจะไปเรียกโปรแกรมย่อย Send_Message ให้ทำงาน

5.1.10 Serial Polling สามารถเลือกใช้โดยการ เลื่อนลูกศรขึ้นหรือลง หรือกดอักษร S ก็ได้ เพื่อให้แถบสีมาที่คำสั่งนี้แล้วกดคีย์ ENTER ต่อมาจะแสดงหน้าต่างขึ้นมาใหม่ให้ใส่ค่าแอดเดรส (Address) เมื่อใส่ค่าแอดเดรสแล้ว จะเรียกโปรแกรมย่อย Spolling ให้ทำงาน หลังจากนั้นจะแสดงหน้าต่างขึ้นมาใหม่เพื่อแสดงค่าข้อมูล(data) ที่ได้รับจากเครื่องที่แอดเดรสนั้น ดังแสดงในรูปที่ 5-7

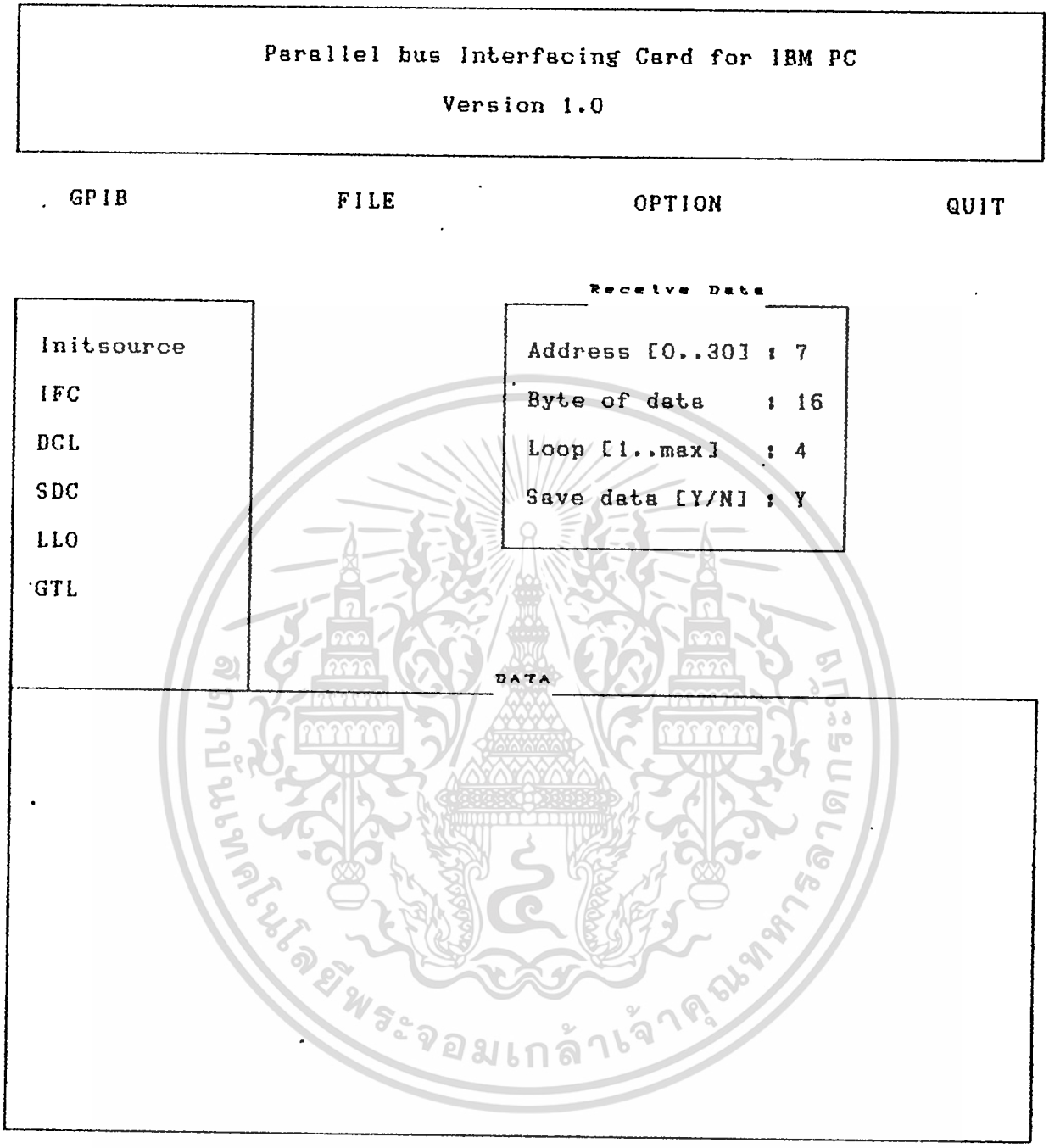
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Receive data from device.

รูปที่ 5.4 แสดงหน้าต่างที่เปิดขึ้นมาเมื่อเลือกคำสั่ง Receive Data

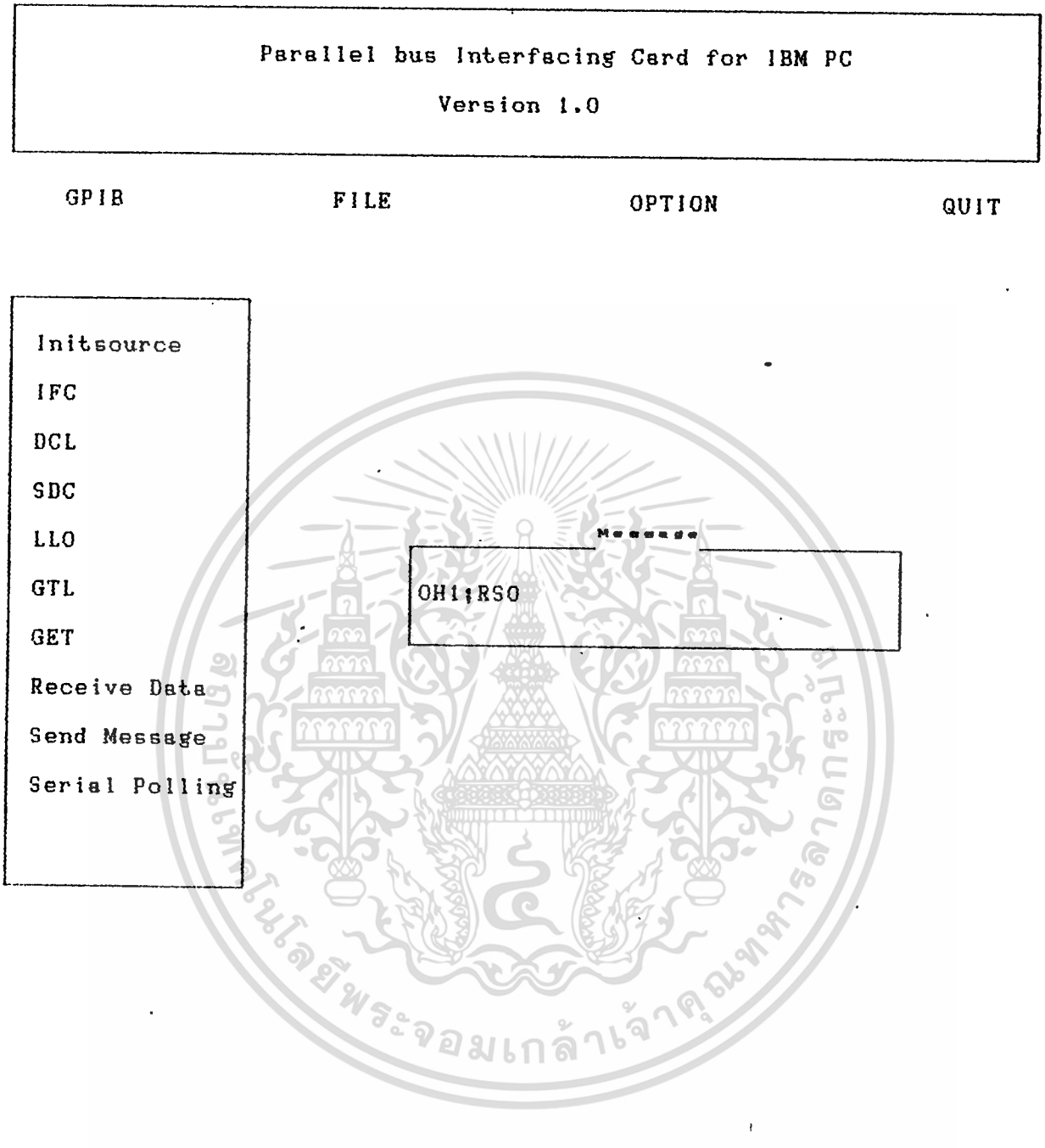
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Receive data from device.

รูปที่ 5.5 แสดงหน้าต่างที่เปิดขึ้นมาเพื่อแสดงข้อมูลที่ได้รับเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.6 แสดงหน้าต่างที่เปิดขึ้นมาเพื่อให้ใส่ข้อมูลที่ต้องการส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

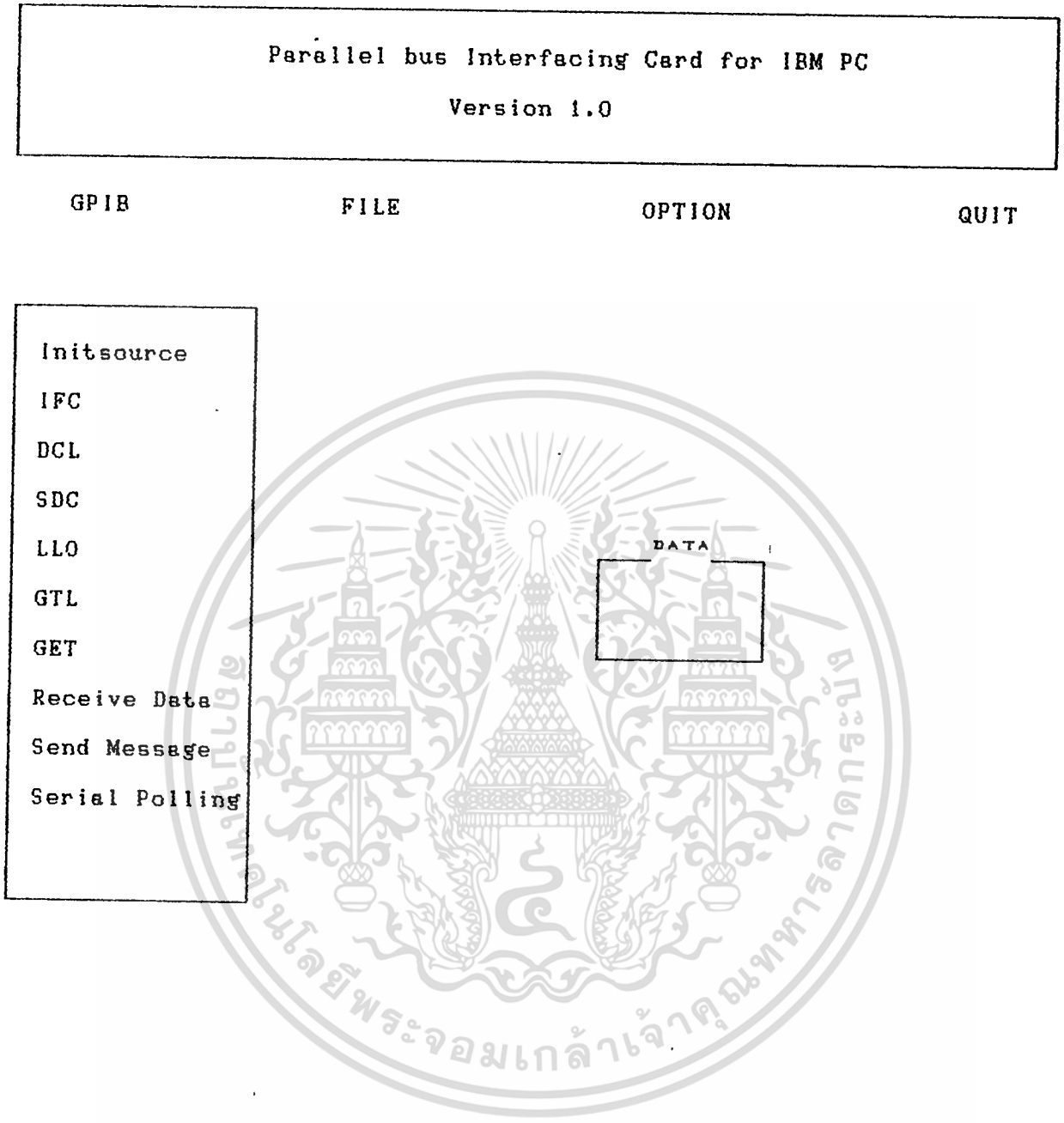
5.1.7 GET สามารถเลือกใช้โดยการ เลื่อนลูกศรขึ้นหรือลง หรือกดอักษร G ก็ได้เพื่อให้แถบสีมาที่คำสั่งนี้ แล้วกดคีย์ ENTER ต่อมาจะแสดงหน้าต่างขึ้นมาใหม่ให้ใส่ค่าแอดเดรส (Address) เมื่อใส่ค่าแอดเดรสแล้ว ก็จะไปเรียกโปรแกรมย่อย Send_Command เพื่อให้ส่งคำสั่ง GET ออกไป

5.1.8 RECEIVE DATA สามารถเลือกใช้โดยการ เลื่อนลูกศรขึ้นหรือลง หรือกดอักษร R ก็ได้เพื่อให้แถบสีมาที่คำสั่งนี้ แล้วกดคีย์ ENTER ต่อมาจะแสดงหน้าต่างขึ้นมาใหม่ให้ใส่ค่าแอดเดรส (Address),จำนวนข้อมูล(Bytes of data) จำนวนลูป(Loop) เมื่อใส่ค่าครบแล้ว ก็จะมีการถามว่าต้องการเก็บข้อมูลลงแผ่นดิสก์หรือไม่ ถ้าต้องการเก็บข้อมูลลงแผ่นดิสก์ ก็ให้กดตัวอักษร Y แล้วจะแสดงหน้าต่างขึ้นมาใหม่ให้ใส่ชื่อไฟล์ ดังรูปที่ 5-4 หรือถ้าไม่ต้องการเก็บข้อมูลลงแผ่นดิสก์ ก็ให้กดตัวอักษร N ต่อมาก็จะไปเรียกโปรแกรมย่อย Get_Data ให้ทำงานพร้อมทั้งอ่านเวลาขณะนั้นแล้ว จะแสดงหน่าต่างขึ้นมาใหม่เพื่อแสดงข้อมูลที่ได้พร้อมทั้งเวลา ดังรูปที่ 5-5

5.1.9 Send Message สามารถเลือกใช้โดยการ เลื่อนลูกศรขึ้นหรือลง หรือกดอักษร S ก็ได้ เพื่อให้แถบสีมาที่คำสั่งนี้ แล้วกดคีย์ ENTER ต่อมาจะแสดงหน้าต่างขึ้นมาใหม่ให้ใส่ค่าแอดเดรส (Address) เมื่อใส่ค่าแอดเดรสแล้วจะแสดงหน้าต่างขึ้นมาใหม่ให้ใส่ข้อความ (Message) ไม่เกิน 50 ตัวอักษรที่ต้องการส่งไปตั้ง(set) การทำงานของเครื่องที่อยู่ที่แอดเดรสนั้น ดังรูปที่ 5-6 ต่อมาจะไปเรียกโปรแกรมย่อย Send_Message ให้ทำงาน

5.1.10 Serial Polling สามารถเลือกใช้โดยการ เลื่อนลูกศรขึ้นหรือลง หรือกดอักษร S ก็ได้ เพื่อให้แถบสีมาที่คำสั่งนี้แล้วกดคีย์ ENTER ต่อมาจะแสดงหน้าต่างขึ้นมาใหม่ให้ใส่ค่าแอดเดรส (Address) เมื่อใส่ค่าแอดเดรสแล้ว จะเรียกโปรแกรมย่อย Spolling ให้ทำงาน หลังจากนั้นจะแสดงหน้าต่างขึ้นมาใหม่เพื่อแสดงค่าข้อมูล(data) ที่ได้รับจากเครื่องที่แอดเดรสนั้น ดังแสดงในรูปที่ 5-7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Serial polling.

รูปที่ 5.7 แสดงหน้าต่างที่เปิดขึ้นมาเพื่อแสดงข้อมูลที่ได้รับในกรณี Serial Polling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

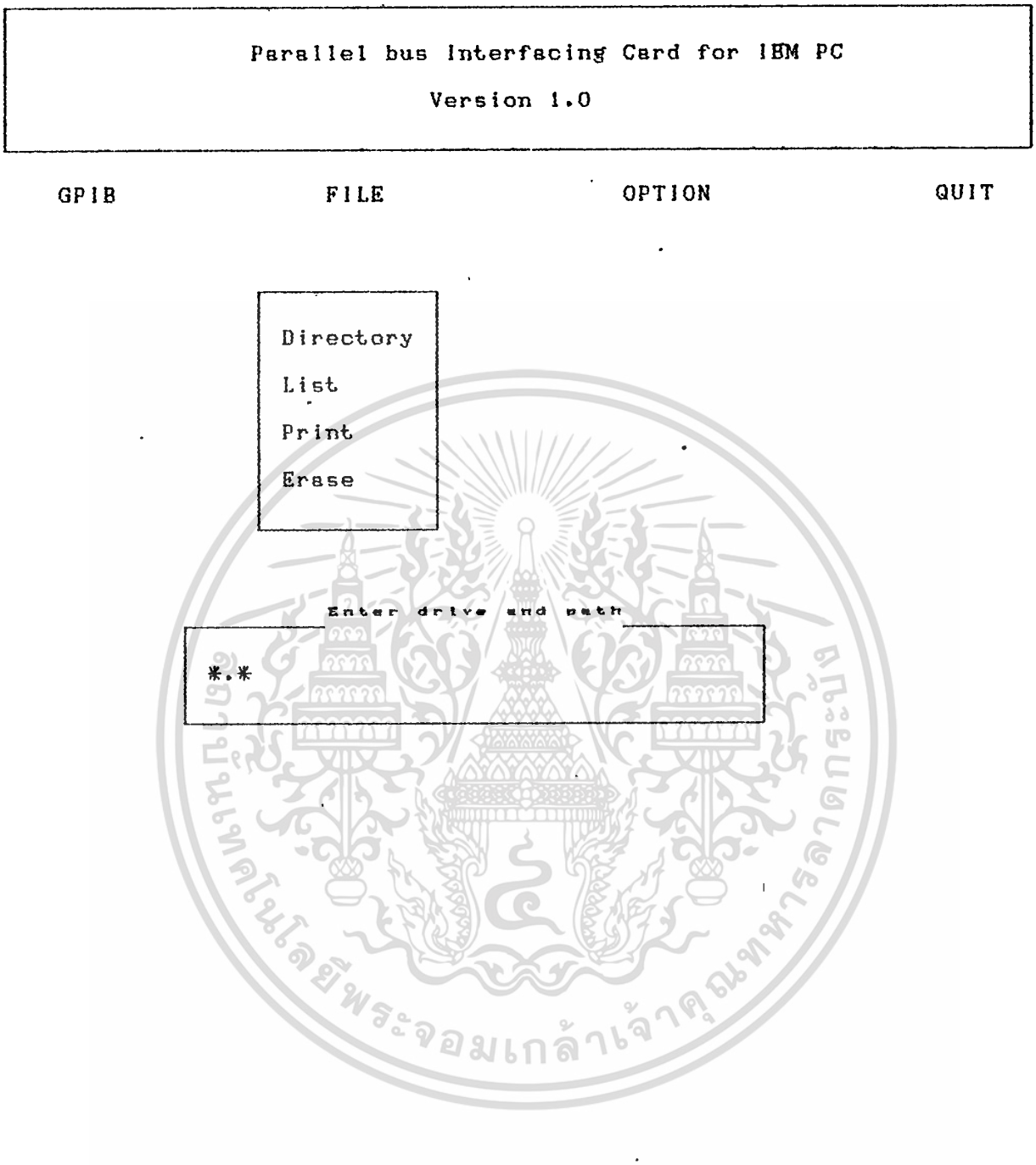
5.2 FILE MENU เมื่อเลือก FILE MENU จะมีหน้าต่าง(WINDOW) เกิดขึ้นมาใหม่ใต้คำว่า FILE ดังรูปที่ 5-8 ภายใน FILE MENU จะประกอบด้วยคำสั่งต่างๆดังนี้คือ

5.2.1 Directory สามารถเลือกใช้โดยการ เลื่อนลูกศรขึ้นหรือลงเพื่อให้แถบสีมาที่คำสั่งนี้ แล้วกดคีย์ ENTER หรือกดอักษร D ก็ได้ แล้วจะแสดงหน้าต่างขึ้นมาใหม่ให้ใส่ชื่อช่องเก็บข้อมูลพร้อมทั้งทางเดิน(Drive and Path) แล้วจะใส่หรือไม่ใส่ชื่อไฟล์ก็ได้ โดยถ้าไม่ใส่อะไรเลยจะถือว่าเลือกช่องเก็บข้อมูล และทางเดินขณะนั้น และจะแสดงทุกๆไฟล์(*.*) ดังรูปที่ 5-6 และรูปที่ 5-9 เมื่อขณะที่แสดงไฟล์ผู้ใช้สามารถเลื่อนลูกศรขึ้น-ลง-ซ้าย-ขวา เพื่อแถบสีไปยังไฟล์ที่ต้องการหรือกดตัวอักษรตัวแรกของไฟล์นั้น เพื่อดูขนาดของไฟล์ที่เลือกก็ได้ แล้วกดคีย์ ENTER หรือ ESC เพื่อออกจากคำสั่ง Directory

5.2.2 List สามารถเลือกใช้โดยการ เลื่อนลูกศรขึ้นหรือลงเพื่อให้แถบสีมาที่คำสั่งนี้ แล้วกดคีย์ ENTER หรือกดอักษร L ก็ได้คำสั่งนี้จะแสดงข้อมูลที่อยู่ในไฟล์ที่เลือก โดยจะมีการถามชื่อไฟล์ก่อน ถ้าต้องการเลือกไฟล์ก็ให้กดคีย์ ENTER ก็จะแสดงไฟล์ต่างๆเหมือนตอนเลือก Directory ซึ่งสามารถเลือกไฟล์ที่ต้องการแสดงโดยเลื่อนแถบสีไปยังไฟล์ที่ต้องการ หรือกดอักษรตัวแรกของไฟล์ที่ต้องการแสดงก็ได้แล้วกดคีย์ ENTER แต่ถ้าไม่ต้องการเลือกไฟล์ก็ให้ใส่ชื่อไฟล์ที่ต้องการแสดงแล้วกดคีย์ ENTER เมื่อข้อมูลที่อยู่ในไฟล์ได้แสดงที่จอ ผู้ใช้สามารถกดคีย์ต่างๆดังนี้

คีย์ <-- เพื่อเลื่อนบรรทัดขึ้น 1 บรรทัด
คีย์ Enter เพื่อเลื่อนบรรทัดลง 1 บรรทัด
คีย์ PgUp เพื่อเลื่อนหน้า(Page)ขึ้น 1 หน้า
คีย์ PgDn เพื่อเลื่อนหน้า(Page)ลง 1 หน้า
คีย์ Home เพื่อเลื่อนไปยังต้นไฟล์
คีย์ End เพื่อเลื่อนไปยังท้ายไฟล์
คีย์ Q เพื่อออกจาก List (หรือคีย์ ESC ก็ได้)

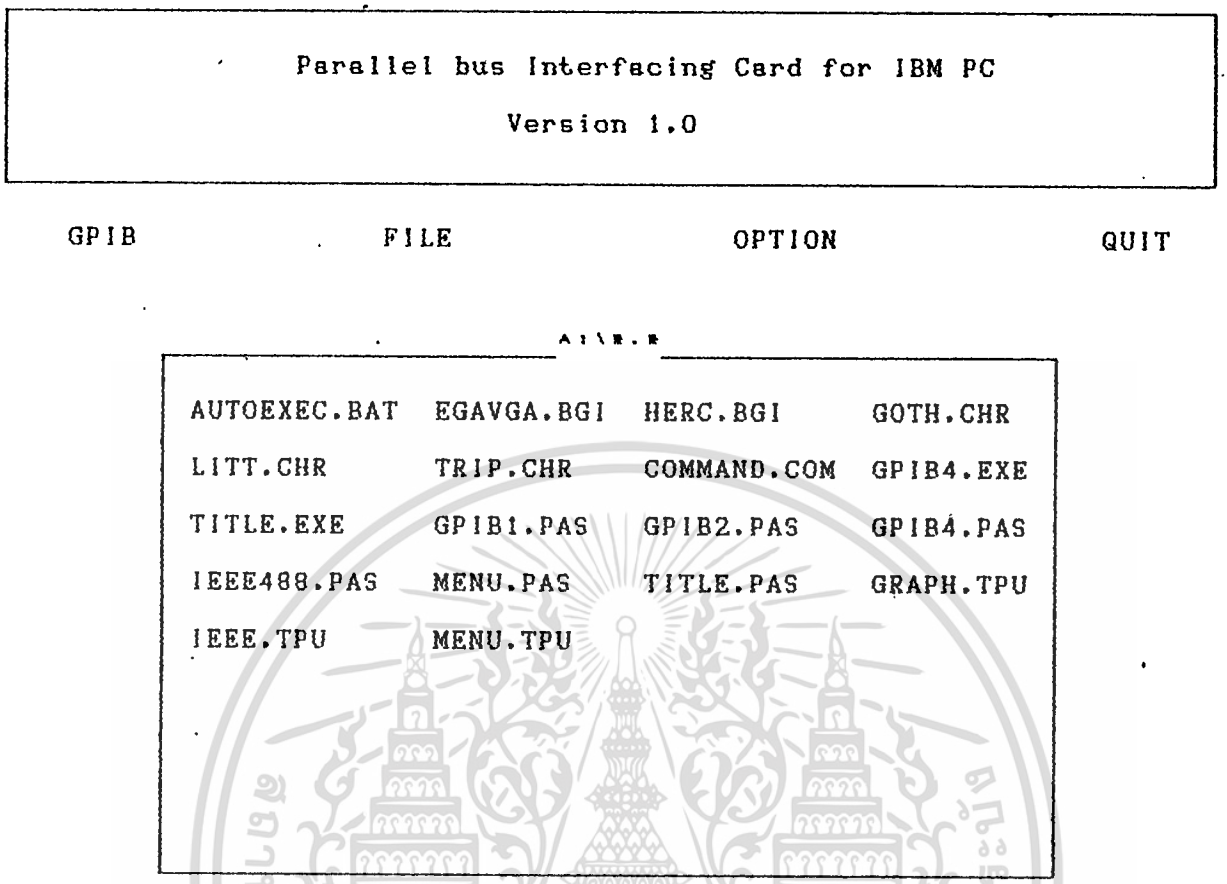
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Display files.

รูปที่ 5.8 แสดงภาพบนจอเมื่อเลือก FILE MENU

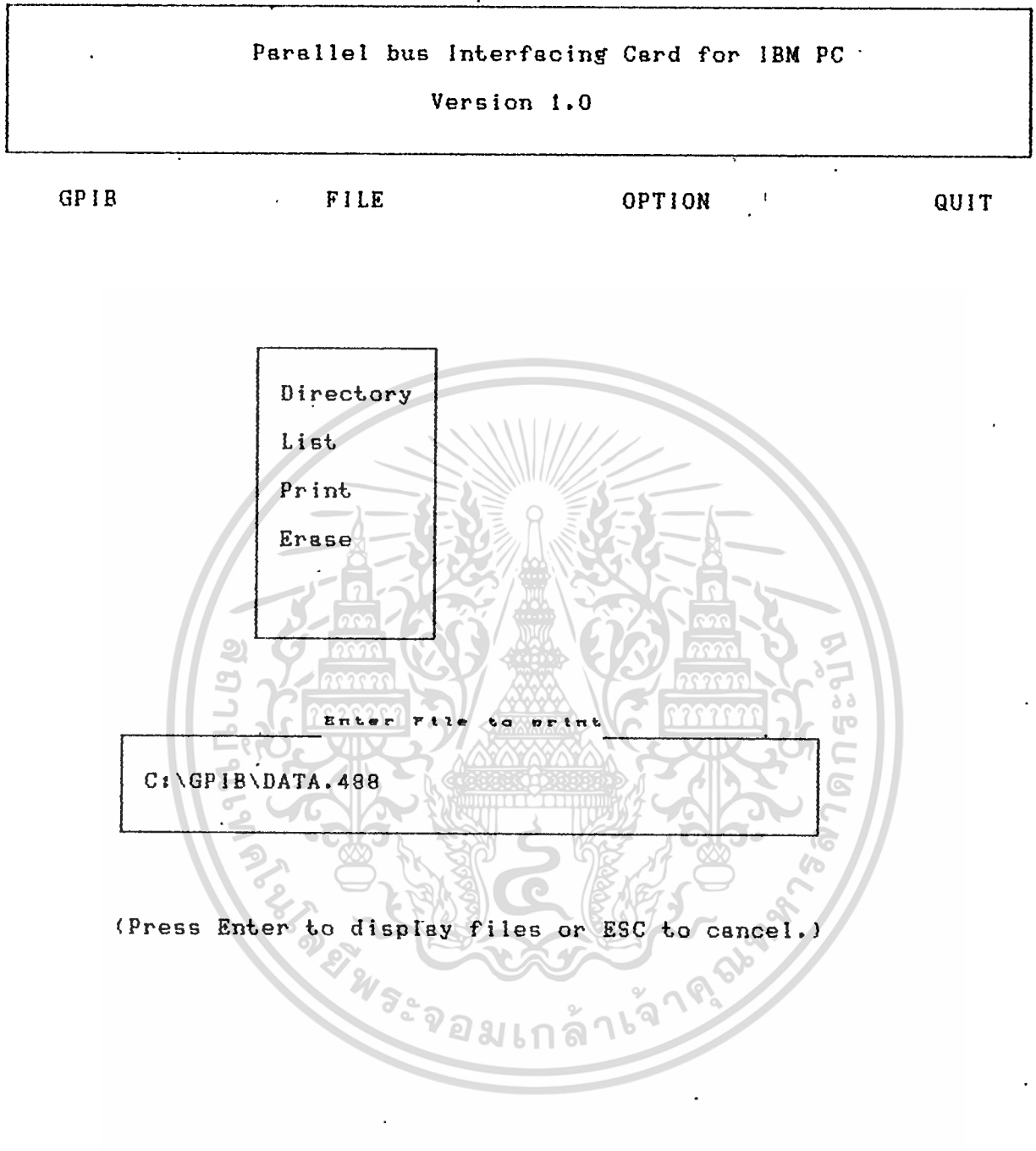
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.9 แสดงหน้าต่างเมื่อลือก Directory ให้แสดงทุกไฟล์

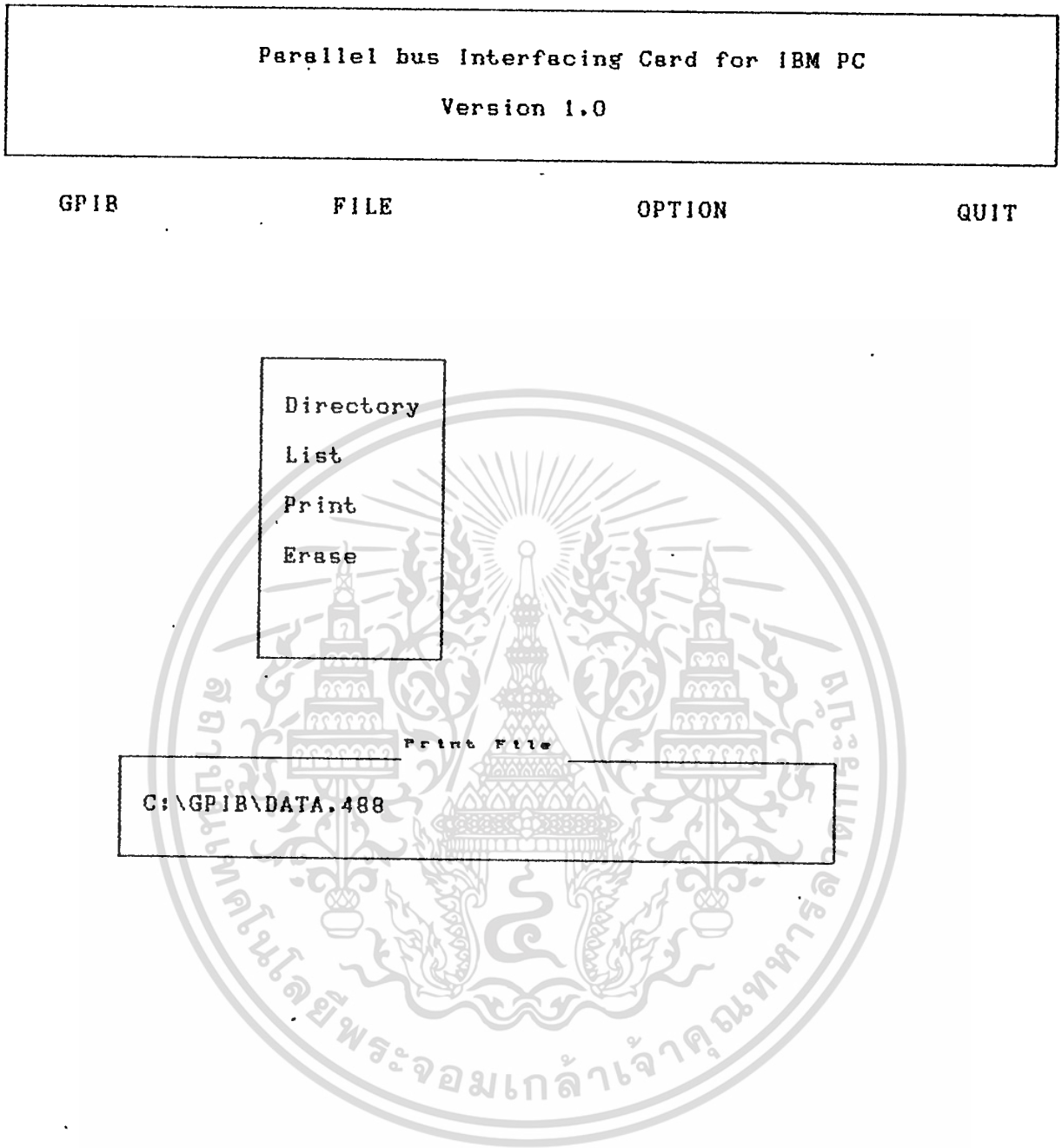
5.2.3 Print สามารถเลือกใช้โดยการ เลื่อนลูกศรขึ้นหรือลงเพื่อให้แถบสีมาที่คำสั่งนี้ แล้วกดคีย์ ENTER หรือกดอักษร P ก็ได้ คำสั่งนี้จะพิมพ์ข้อมูลที่อยู่ในไฟล์ที่เลือกออกทางเครื่องพิมพ์ (Printer) โดยจะแสดงหน้าต่างใหม่เพื่อถามชื่อไฟล์ก่อน ถ้าต้องการเลือกไฟล์ก็ให้กดคีย์ ENTER ก็แสดงไฟล์ต่างๆ เหมือนตอนเลือก Directory ซึ่งสามารถเลือกไฟล์ที่ต้องการพิมพ์โดยเลื่อนแถบสีไปยังไฟล์ที่ต้องการ หรือกดอักษรตัวแรกของไฟล์ที่ต้องการพิมพ์แล้วกดคีย์ ENTER ดังรูปที่ 5-10 ต่อมาก็จะแสดงหน้าต่างใหม่ เพื่อถามว่าเครื่องพิมพ์พร้อมหรือยัง ถ้าเครื่องพิมพ์พร้อมแล้วให้กดอักษร Y ก็จะแสดงหน้าต่างใหม่ เพื่อบอกชื่อไฟล์ที่กำลังพิมพ์พร้อมทั้งพิมพ์ไฟล์นั้น ดังรูปที่ 5-11 แต่ถ้าเครื่องพิมพ์ยังไม่พร้อมให้กดอักษร N ก็จะเป็นการยกเลิกคำสั่ง Print

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.10 แสดงหน้าต่างเมื่อเลือก Print กรณีให้ใส่ชื่อไฟล์

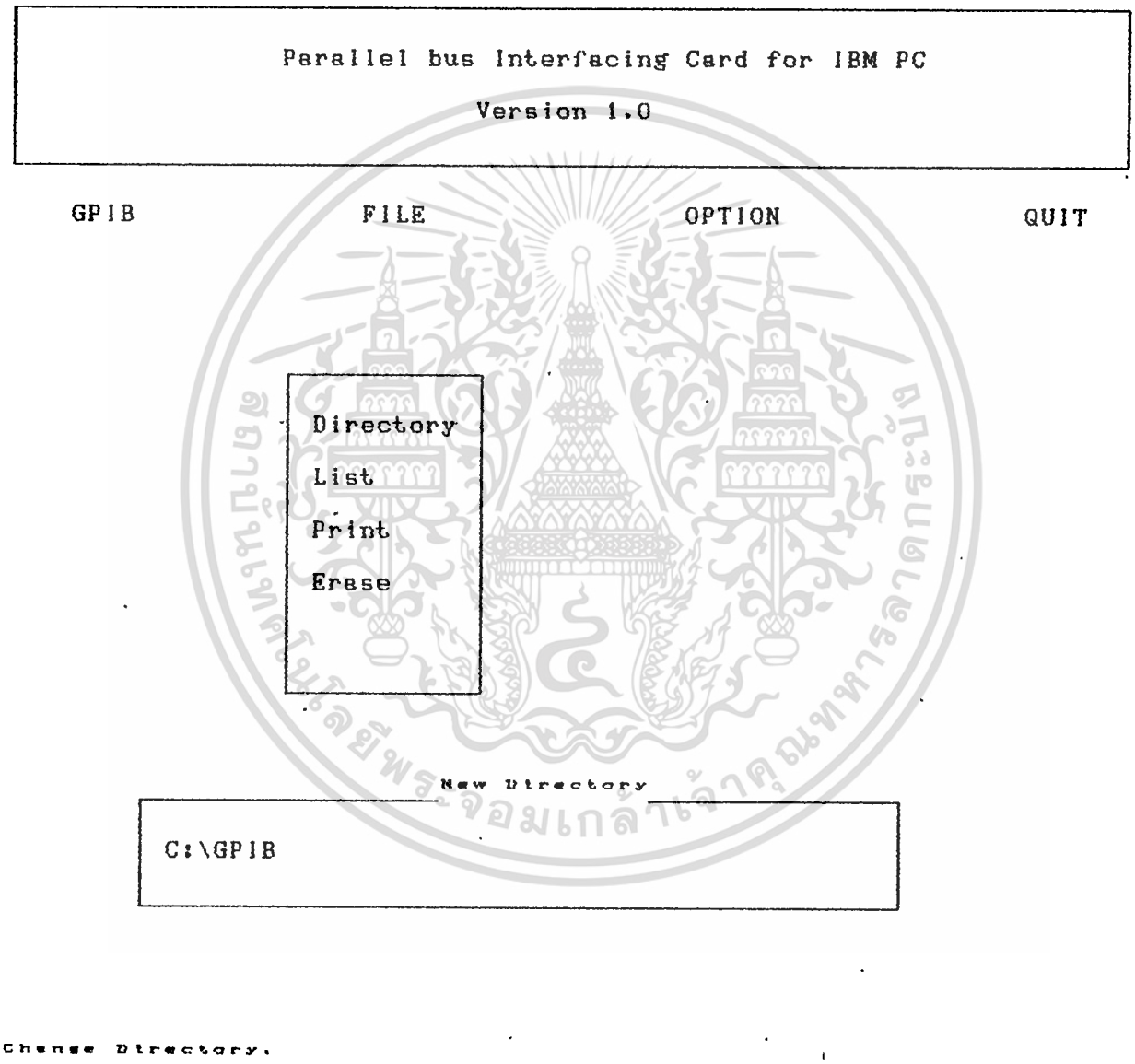
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.11 แสดงหน้าต่างเมื่อเลือก Print กรณีแสดงชื่อไฟล์ที่กำลังพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

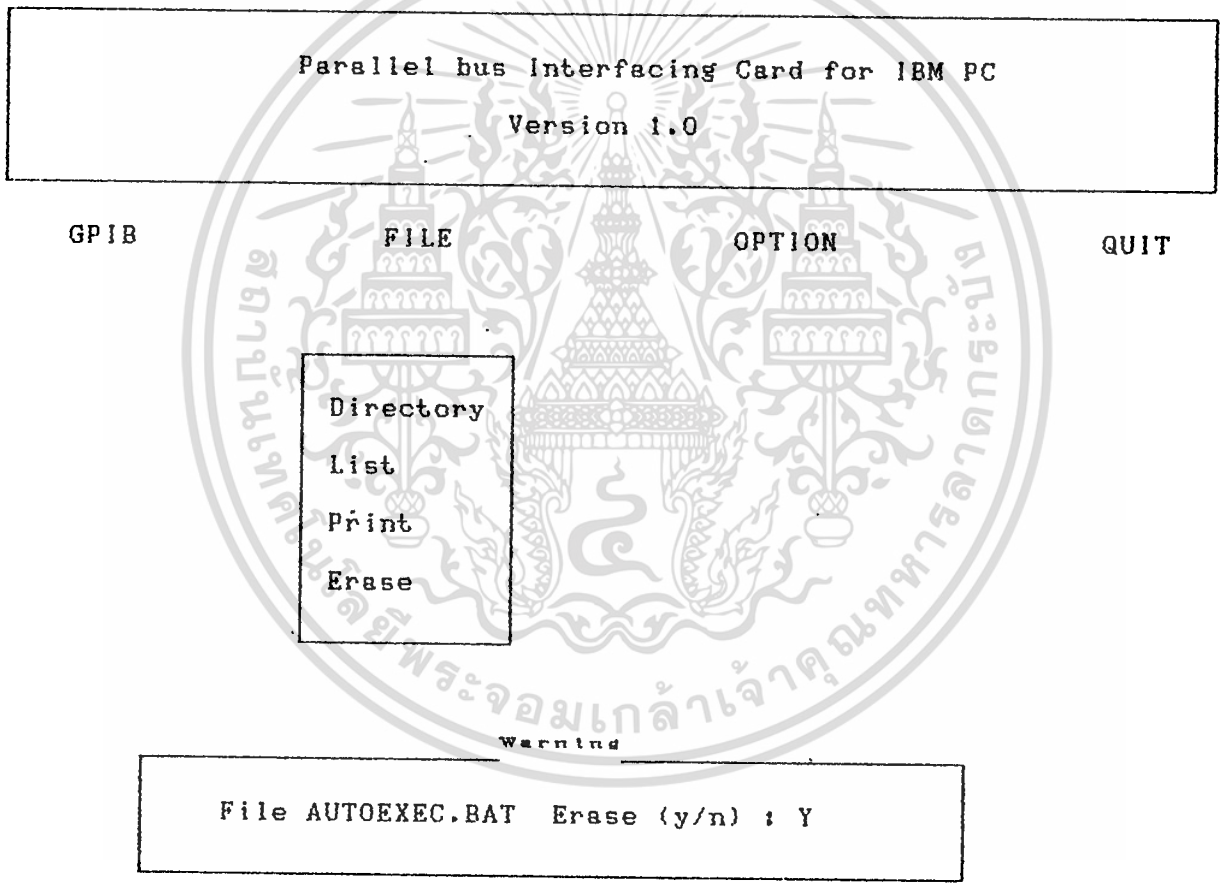
5.2.4 ChDir สามารถเลือกใช้โดยการ เลื่อนลูกศรขึ้นหรือลงเพื่อให้แถบสีมาที่คำสั่งนี้ แล้วกดคีย์ ENTER หรือกดอักษร C ก็ได้คำสั่งนี้ใช้สำหรับเปลี่ยนช่องเก็บข้อมูล และทางเดิน (Drive and Path) โดยจะแสดงหน้าต่างขึ้นมาเพื่อให้ใส่ช่องเก็บข้อมูล และทางเดินใหม่ (Drive and Path หรือ New Directory) ดังรูปที่ 5-12



รูปที่ 5.12 แสดงหน้าต่างเมื่อเลือก ChDir กรณีให้ใส่ New Directory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.5 Erase สามารถเลือกใช้โดยการ เลื่อนลูกศรขึ้นหรือลงเพื่อให้แถบสีมาที่คำสั่งนี้ แล้วกดคีย์ ENTER หรือกดอักษร E ก็ได้ คำสั่งนี้ใช้สำหรับลบไฟล์ที่ไม่ต้องการออกจากแผ่นดิสก์ โดยจะแสดงหน้าต่างเพื่อให้ใส่ชื่อไฟล์ที่ต้องการลบ ถ้าต้องการเลือกไฟล์ก็ให้กดคีย์ ENTER ก็จะแสดงไฟล์ต่างๆเหมือนตอนเลือก Directory ซึ่งสามารถเลือกไฟล์ที่ต้องการลบโดยเลื่อนแถบสีไปยังไฟล์ที่ต้องการ หรือกดอักษรตัวแรกของไฟล์ที่ต้องการลบก็ได้แล้วกดคีย์ ENTER แต่ถ้าไม่ต้องการเลือกไฟล์ก็ให้ใส่ชื่อไฟล์ที่ต้องการลบแล้วกดคีย์ ENTER ต่อมาก็จะแสดงหน้าต่างใหม่เพื่อถามว่าไฟล์ที่ได้เลือกไว้ต้องการลบใช่หรือไม่ ดังแสดงในรูปที่ 5-13 ถ้าต้องการลบให้กดอักษร Y แต่ถ้าไม่ต้องการลบให้กดอักษร N ก็จะยกเลิกการลบและออกจาก Erase



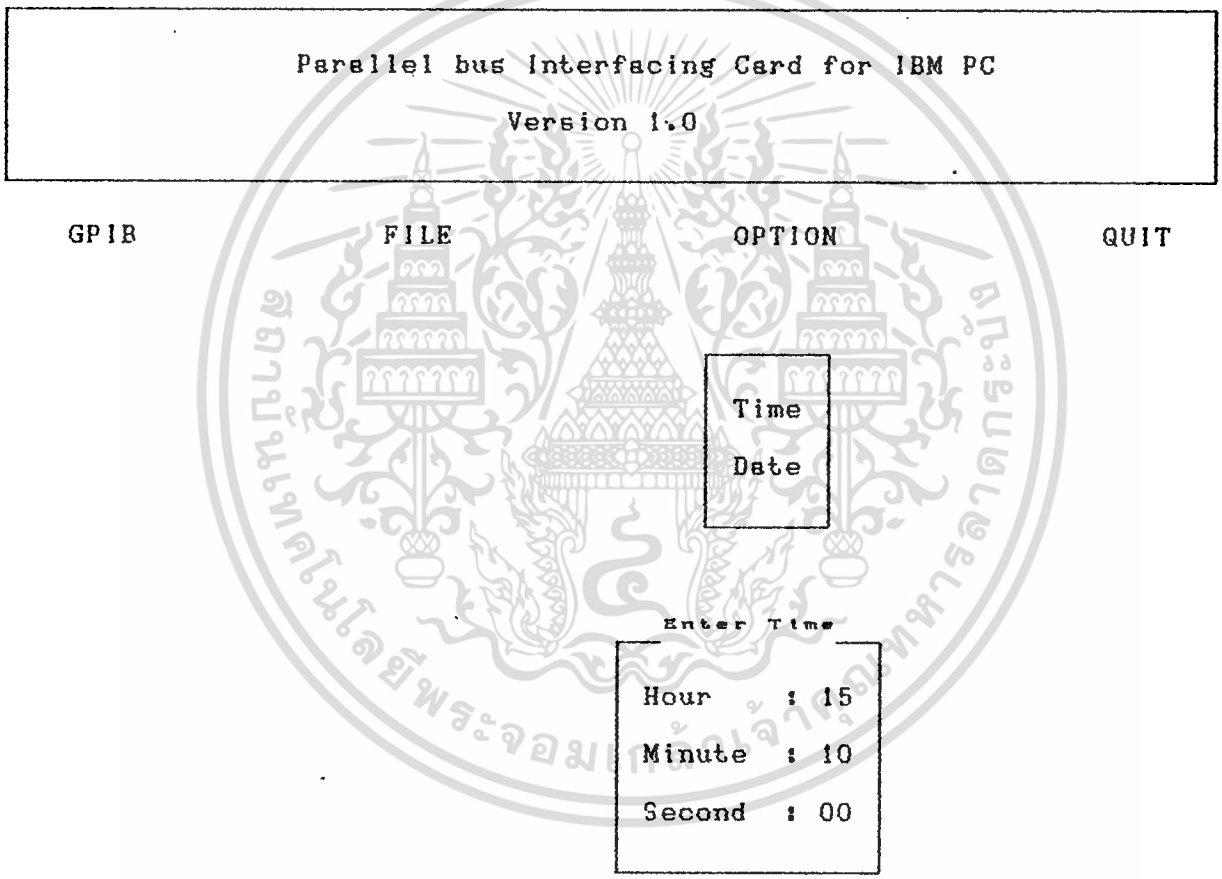
Erase file.

รูปที่ 5.13 แสดงหน้าต่างเมื่อเลือก Erase กรณีถามว่าต้องการลบไฟล์ใช่หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 **OPTION MENU** เมื่อเลือก OPTION MENU จะมีหน้าต่าง(window) เกิดขึ้นใหม่ใต้คำว่า OPTION ดังแสดงในรูปที่ 5-14 ภายใน OPTION MENU จะประกอบด้วยคำสั่งต่างๆดังนี้คือ

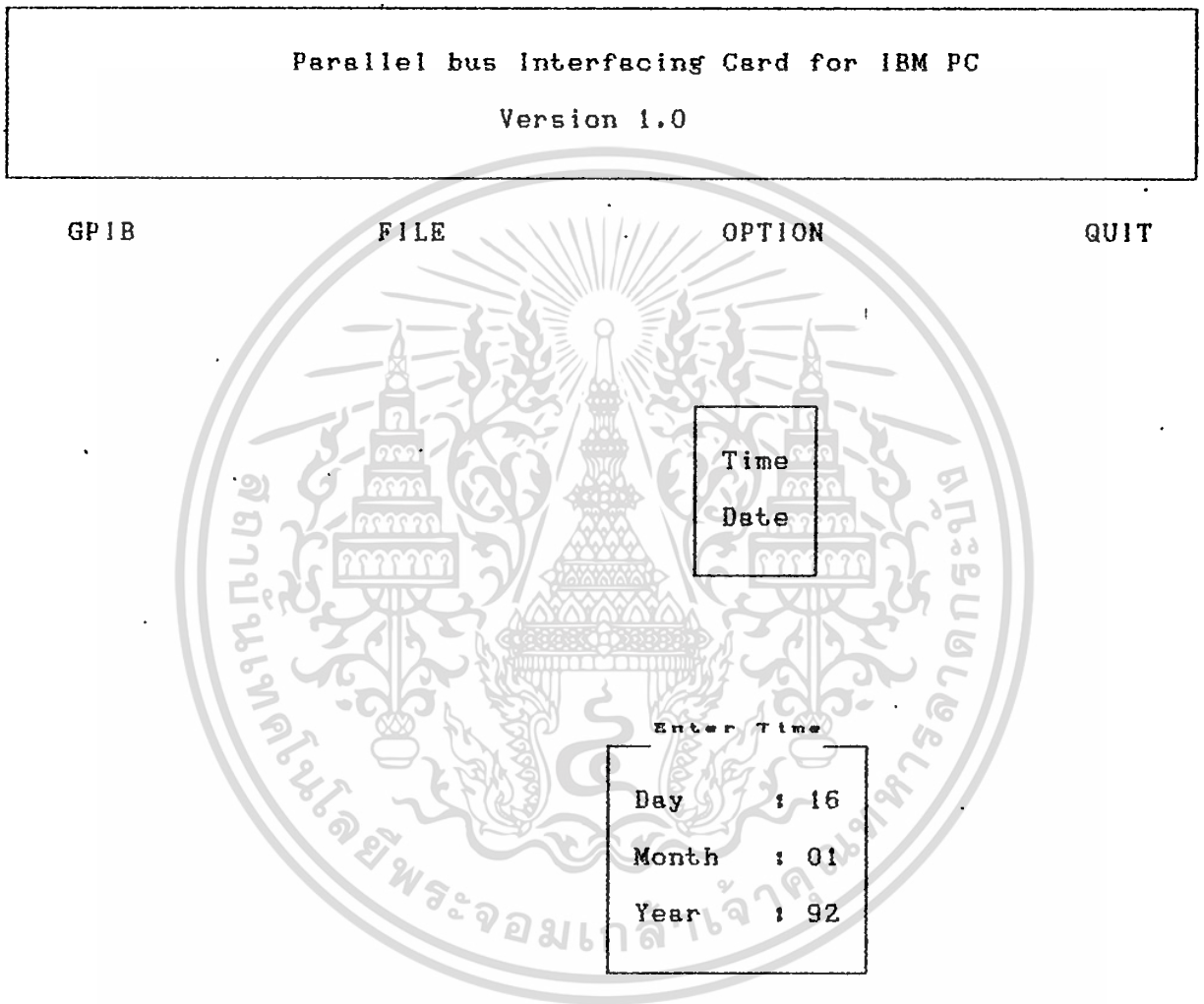
5.3.1 Time สามารถเลือกใช้ได้โดยการ เลื่อนลูกศรขึ้นหรือลงเพื่อให้แถบสว่างมาที่คำสั่งนี้ แล้วกดคีย์ ENTER หรือกดอักษร T ก็ได้ คำสั่งนี้ใช้สำหรับการตั้งเวลาให้แก่ระบบจัดการ (เหมือนป้อนเวลาตอนเริ่มเริ่มเดินเครื่อง) โดยจะแสดงหน้าต่างเพื่อให้ใส่เวลาเป็นชั่วโมง,นาที,วินาที ดังแสดงในรูปที่ 5-14



รูปที่ 5.14 แสดงหน้าต่างเมื่อเลือก Option กรณีให้ใส่เวลาที่ตั้งให้แก่ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

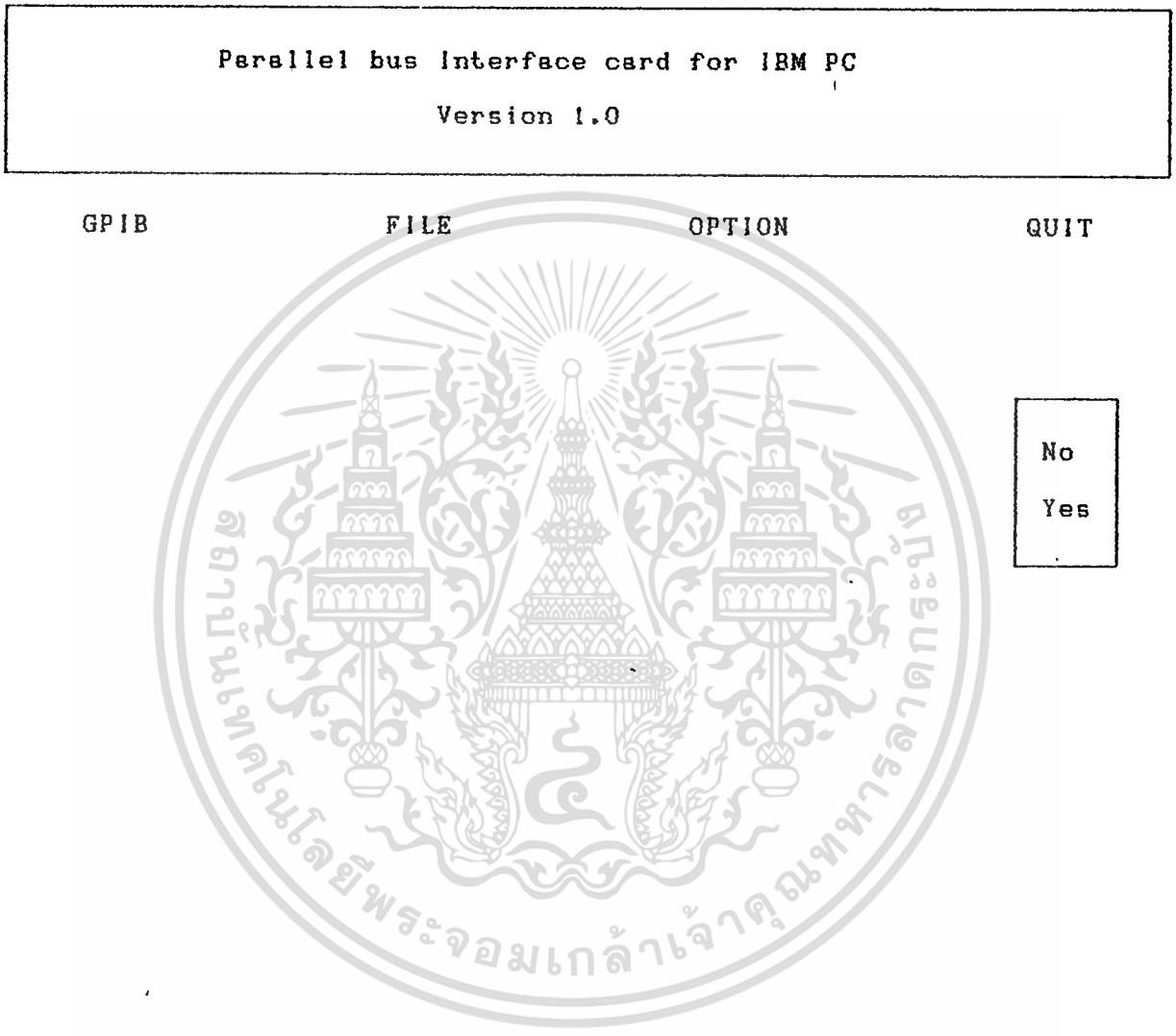
5.3.2 Date สามารถเลือกใช้ได้โดยการ เลื่อนลูกศรขึ้นหรือลงเพื่อให้แถบสว่างมาที่คำสั่งนี้ แล้วกดคีย์ ENTER หรือกดอักษร D ก็ได้ คำสั่งนี้ใช้สำหรับการตั้งวันให้แก่ระบบจัดการ (เหมือนป้อนเวลาตอนเริ่มเริ่มเดินเครื่อง) โดยจะแสดงหน้าต่างเพื่อให้ใส่วันเป็นวันที่, เดือน, ปี ดังแสดงในรูปที่ 5-15



รูปที่ 5.15 แสดงหน้าต่างเมื่อเลือก Option กรณีให้ใส่วันที่ตั้งให้แก่ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 **QUIT MENU** เมื่อเลือก QUIT MENU จะมีหน้าต่าง (window) เกิดขึ้นใหม่ใต้คำว่า QUIT ดังรูปที่ 5.16 ถ้าไม่ต้องการออกจากโปรแกรมก็ให้กด No และถ้าต้องการให้ออกจากโปรแกรมก็ให้กด Yes



รูปที่ 5.16 แสดงหน้าต่างเมื่อเลือก Quit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การทดลองและผลการทดลอง

ในการทดลองนี้ ได้ผลตามเป้าหมายที่ได้ตั้งไว้โดยสามารถใช้ Personal Computer ติดต่อกับอุปกรณ์และเครื่องมือวัดได้ เช่นการรับข้อมูลจากเครื่องมือวัดหรือการควบคุมการทำงานโดยการสั่งการทางคีย์บอร์ดของ PC เอง ซึ่งเป็นไปได้อย่างรวดเร็วและถูกต้อง

อนึ่งในครั้งแรกของการทำโครงงานนี้เราใช้การ wire warp ลง print เอนกประสงค์ ดังนั้นจึงพบปัญหาว่ามีสัญญาณรบกวนในระบบมาก และจากสัญญาณรบกวนนี้ทำให้ค่าของตัวเลขหรือตัวอักษรที่แสดงผลออกมาบนจอมอนิเตอร์ มีรูปร่างที่ผิดเพี้ยนไปจากเดิม และหลังจากที่คณะผู้จัดทำได้พยายามหาทางแก้ไขฮาร์ดแวร์ และออกแบบลายวงจรใหม่เพื่อหลีกเลี่ยงการซับซ้อนของลายวงจร โดยใช้ PC ช่วยในการออกแบบทำให้สามารถแก้ปัญหาการรบกวนของสัญญาณได้เป็นอย่างดี

ในการทดลองนี้ผู้จัดทำไม่สามารถใช้อุปกรณ์เครื่องมือวัดได้พร้อมๆกัน 14 เครื่อง(จากคุณสมบัติของ GPIB BUS นั้นสามารถนำอุปกรณ์มาต่อ PARALLEL กันในระบบได้ถึง 14 ตัว) แต่ในการทดลองนี้เราใช้เพียงแค่ 3 ตัวเท่านั้นคือ Digital Multimeter, Pulse Generator และ Temperature recorder โดยผลจากการทดลองกับเครื่องมือเหล่านี้อยู่ในเกณฑ์ที่น่าพอใจอย่างยิ่ง โดยค่าที่อ่านได้จาก Digital Multimeter หรือที่ Pulse Generator ซึ่งอุปกรณ์สองเครื่องนี้ถูก SET ให้เป็นตัว Talker ค่าที่อ่านได้บนมอนิเตอร์ตรงกับค่าที่แท้จริงกับค่าที่ปรากฏบนเครื่องมือวัดนั้นๆ ส่วน Temparature recorder นั้นถูก set ให้เป็นตัว Listener ก็สามารถเปลี่ยนแปลงฟังก์ชันตามที่ผู้ใช้ต้องการได้ถูกต้องและรวดเร็วมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

บทวิจารณ์และสรุป

จากผลการทดลองที่ได้นี้อยู่ในเกณฑ์ที่น่าพอใจ แต่สิ่งที่คณะผู้จัดทำหรือผู้ที่สนใจในโครงงานนี้ควรจะทำการแก้ไขหรือพัฒนาต่อไปก็คือ การปรับปรุงด้าน Hardware และ Software ในส่วนของ Hardware ถ้าเราเลือกใช้ไอซีเบอร์อื่นๆที่มีประสิทธิภาพและความเร็วในการทำงานสูงๆจะทำให้ประสิทธิภาพการทำงานดีขึ้น การรับส่งข้อมูลเป็นไปได้อย่างรวดเร็วและถูกต้อง ซึ่งประโยชน์ในส่วนนี้สามารถนำไปประยุกต์ใช้ในงานทางด้านไฟฟ้าและอิเล็กทรอนิกส์ได้เป็นอย่างดีเช่น การวัดสัญญาณทรานเซียนต์ การวัดความเร็วรอบมอเตอร์ เป็นต้น

ในส่วนของซอฟแวร์ ควรจะพัฒนาในลักษณะ Real Time Graphic ที่มีความละเอียดถูกต้องและข้อสำคัญคือพยายามพัฒนาให้มีความสะดวกและง่ายต่อการใช้งานมากที่สุด โดยอาจจะพัฒนาให้เป็นเมนูภาษาไทยเพื่อสื่อความหมายได้ง่ายและชัดเจนมากขึ้น โดยให้คำนึงถึงต้นทุนมากที่สุดขณะที่ประสิทธิภาพใกล้เคียงกับต่างประเทศ

สุดท้ายนี้คณะผู้จัดทำหวังเป็นอย่างยิ่งว่าโครงงานนี้คงจะเป็นประโยชน์ แก่ผู้ที่สนใจในด้านการติดต่อสื่อสารข้อมูลระหว่างเครื่องมือวัดและคอมพิวเตอร์ส่วนบุคคล ให้แพร่หลายมากขึ้น โดยข้อดีของโครงงานนี้ ทางคณะผู้จัดทำขอมอบให้แก่สถาบันพระจอมเกล้าลาดกระบัง และหากโครงงานนี้มีข้อบกพร่องประการใดทางคณะผู้จัดทำมีความยินดีรับข้อเสนอแนะทุกประการ

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MENU.PAS

```
Unit Menu;

Interface

Uses    Crt,Dos;

Const   single    ='        ';
        double    ='        ';

Type    str8      = string[8];
        str12     = string[12];
        str30     = string[30];
        str80     = string[80];
        switch    = (on,off);
        CharSet   = set of char;
        OneScreen = array[1..25,1..80,1..2] of byte;
        Screen    = record
                      ScreenData : OneScreen;
                      row,col    : byte;
                    end;

var     IsColorCard          : boolean;
        LastActive           : byte;
        VideoSegment         : word;
        Firstkey,Lastkey     : char;

Function  DirFiName(fname : str30) : str30;
Function  Upper(strg : string) : string;
Function  Space(num : byte ) : string;
Function  Ctoi(strg : string  ) : integer;
Function  Exist(FileName : string ) : boolean;
Procedure Tone(freq,time:word);
Procedure Color(T,B : byte );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Procedure Say(x,y : byte; message : str80);
Procedure Wdrt(x,y : byte; message : str80);
Procedure Shs(SW : switch);
Procedure Ssl(SW : switch);
Procedure Scur(SW : switch);
Procedure Suda(SW : switch);
Procedure Svcr;
Procedure Rscr;
Procedure SvcrT(var scr : Screen);
Procedure RscrF(scr : Screen);
Procedure Frame(border  : str8; tx,ty,bx,by : byte; title : str80);
Procedure Sftc(T,B : byte);
Procedure Prompt(col,row : byte; Menudata,Pmsg : str80);
Procedure Menuto(var option : byte);
Procedure Let(var strg; strglen : byte; Valid : CharSet);
Procedure ReadLet;

Implementation

Type    GetPtr    = ^GetRec;
        GetRec    = record
                      data              : string;
                      DataSeg,DataOfs   : word;
                      row,col,datalen   : byte;
                      ValidSet          : CharSet;
                      pre,next          : GetPtr;
        end;


        PromptPtr  =  ^PromptRec;
        PromptRec  =   record
                         x,y,Fntspc    : byte;
                         Hilgt         : str80;
                         msg           : string;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    PromptNo      : byte;
                    pre,next      : PromptPtr;
                end;
    ScreenPtr   =  ^ScreenRec;
    ScreenRec   =   record
                    ScrData : Screen;
                    next    : ScreenPtr;
                end;
Const   BiosSeg       = $40;
        Pcnt          : byte = 0;
        Udsl          : byte = 0;
        Hslt          : boolean = true;
        Uuda          : boolean = true;
        Slide         : boolean = false;
        FirstGet      : GetPtr  = nil;
        LastGet       : GetPtr  = nil;
        FirstPrompt   : PromptPtr  = nil;
        LastPrompt    : PromptPtr  = nil;
        LastScr       : ScreenPtr  = nil;

var     NowGet        : GetPtr;
        NowPrompt     : PromptPtr;
        NowScr        : ScreenPtr;
        ch            : char;
        Option,Hpc,Tpc,Bpc,Tmc,Bmc,Tlc,Blc,Tftc,Bftc : byte;
        bufhead       : word absolute BiosSeg:$1A;
        buftail       : word absolute BiosSeg:$1C;
        bufstart      : word absolute BiosSeg:$80;
        bufend        : word absolute BiosSeg:$82;

Function HeapAvail : boolean;
begin
     HeapAvail := (memavail > 4096);
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Procedure Tone(freq,time : word);
begin
     sound(freq);
     delay(time);
     nosound;
end;

Procedure Color(T,B : byte);
begin
     Textcolor(T);
     Textbackground(B);
end;

Procedure Say(x,y : byte; message : str80);
begin
     gotoXY(x,y);
     write(message);
end;

Procedure Wdrt( x,y : byte; message : str80);
var  i          : byte;
     location   : word;
begin
     dec(x); dec(y);
     location := (2*x) + (y*160);
     for i := 1 to length(message) do
        begin
          mem [VideoSegment:location] := ord(message[i]);
          inc(location,2);
        end;
end;

Procedure Wver(x,y : byte; message : str80);
var  i          : byte;
     location   : word;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
     dec(x); dec(y);
     location := (2*x) + (y*160);
     for i := 1 to length(message) do
        begin
          mem [VideoSegment:location] := ord(message[i]);
          inc(location,160);
        end;
end;

Procedure Wkbd(strg : string);
var       i,key : byte;
          tail  : word;
begin
     tail := buftail;
     if length(strg) > 0 then
     for i := 1 to length(strg) do
         begin
              inc(tail,2);
              if tail = 62 then tail := 30;
              if tail <> bufhead then
                 begin
                      key := ord(strg[i]);
                      mem[BiosSeg:buftail] := key;

                      buftail := tail;
                 end
                 else begin tone(500,100); exit; end;
              end;
     end;

Procedure Wfnk(strg : string);
var       i,key,tail:word;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
     tail := buftail;
     if length(strg) > 0 then
     for i := 1 to length(strg) do
         begin
              inc(tail,2);
              if tail = 62 then tail :=30;
              if tail <> bufhead then
                 begin
                      key := ord(strg[i]);
                      key := swap(key);
                      move(key,mem[BiosSeg:buftail],2);
                      buftail := tail;
                 end
                 else begin tone(500,100); exit; end;
                 end;
         end;

Function upper(strg : string) : string;
var       i : byte;
begin
     if length(strg) > 0 then
     for i := 1 to length(strg) do strg[i] := upcase(strg[i]);
     upper := strg;
end;

Function rtrim(strg : string) : string;
var       len : byte absolute strg;
begin
     while (len > 0) and (strg[length(strg)] = #32) do
           delete(strg,length(strg),1);
     rtrim := strg;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Function ltrim(strg : string) : string;
var     len : byte absolute strg;
begin
     while (len > 0) and (strg[1] = #32) do
          delete(strg,1,1);
     ltrim := strg;
end;

Function alltrim(strg : string) : string;
begin
     alltrim := rtrim(ltrim(strg)) ;
end;

Function Replicate(ch : char; num : byte) : string;
var strg : string;
    len  : byte absolute strg;
begin
     fillchar(strg[1],num,ch);
     len := num;
     Replicate := strg;
end;

Function Space(num : byte) : string;
begin
     Space := Replicate(#32,num);
end;

Function Ctoi(strg : string) : integer;
var      code,number : integer;
begin
     val(strg,number,code);
     if code = 0 then Ctoi := number
     else Ctoi := 0;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Function Exist(FileName : string) : boolean;
var      f : file;
begin
     assign(f,FileName);
     {$I-} reset(f); close(f); {$F+}
     Exist := (IOresult = 0);
end;

Procedure Shs(SW : switch);
begin
     if SW = on then Hslt := True
     else Hslt := False;
end;

Procedure Spc(H,T,B : byte);
begin
     Hpc := H;
     Tpc := T;
     Bpc := B;
end;

Procedure Ssl(SW : switch);
begin
     if SW = ON then Slide := True
     else Slide := False;
end;

Procedure Sct(Pch,Pcl : byte);
var      reg : registers;
begin
     with reg do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          begin
               ah := $01;
               ch := Pch;
               cl := Pcl;
          end;
          intr($10,reg);
end;

Procedure Scur(SW : switch);
begin
     if SW = off then Sct(13,0)
     else if IsColorCard then Sct(7,7)
     else Sct(13,13);
end;

Procedure Smc(T,B : byte);
begin
     Tmc := T; Bmc := B;
end;

Procedure Intp;
begin
     Slide := false;
     Uuda  := True;
end;

Procedure Suda(SW : switch);
begin
     if SW = ON then Uuda := True
     else Uuda := False;
end;

Procedure SvcrT(var scr : Screen);
begin
     with scr do begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        move(mem[VideoSegment:$0000],ScreenData,SizeOF(ScreenData));
        row := whereY; col := whereX;
        end;
end;

Procedure RscrF(scr : Screen);
begin
        with scr do begin
        move(ScreenData,mem[VideoSegment:$0000],SizeOf(ScreenData));
        GotoXY(col,row);
        end;
end;

Procedure Svcr;
begin
        if HeapAvail then
          begin
              new(NowScr);
              with NowScr^ do begin
              SvcrT(ScrData);
              NowScr^ .next := LastScr;
              LastScr := NowScr;
            end;
          end
        else write('Heap overflow');
end;

Procedure Rscr;
begin
        if LastScr <> nil then
        with LastScr^ do begin
        RscrF(ScrData);
        NowScr := LastScr;
        LastScr := LastScr^.next;
        dispose(NowScr);
    end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  else write('No screen saved');
end;

Procedure Sftc(T,B : byte);
begin
     Tftc := T;
     Bftc := B;
end;

Procedure Frame( border  : str8; tx,ty,bx,by : byte; title : str80);
var        i,OWMn,OWMx,OMN,OMX : Word;
           OldAttr,start,len : byte;
           strg : str80;
begin
     OWMn := WindMin;
     OWMx := WindMax;
     window(tx,ty,bx,by);
     clrscr;
     len := bx-tx-1;
     strg := border[1] + Replicate(border[5],len) + border[2];
     Wdrt(tx,ty,strg);
     if Title <> #00 then
        begin
             OldAttr := TextAttr;
             color(Tftc,Bftc);
             start := (length(strg) div 2) - (length(title) div 2);
             OMN := WindMin;
             OMX := WindMax;
             window(1,1,80,25);
             gotoXY(tx+start,ty);
             write(title);
             WindMin := OMN;
             WindMax := OMX;
             TextAttr := OldAttr;
        end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        strg := border[3] + replicate(border[6],len) + border[4];
        Wdrt(tx,by,strg);
        len := by-ty-1;
        strg := Replicate(border[7],len);
        Wver(tx,ty+1,strg);
        len := by-ty-1;
        strg := Replicate(border[8],len);
        Wver(bx,ty+1,strg);
        WindMin := OWMn;
        WindMax := OWMx;
end;

Procedure Cprompt;
begin
     while FirstPrompt <> nil do begin
          NowPrompt := FirstPrompt;
          FirstPrompt := FirstPrompt^.next;
          dispose(NowPrompt);
     end;
     FirstPrompt := nil;
     LastPrompt := nil;
     Pcnt := 0;
end;

Procedure WrtNPrompt(NowPrompt : PromptPtr);
var       OldX,OldY,OldAttr : byte;
begin
     OldAttr := TextAttr;
     with NowPrompt^ do
          begin
             say(x,y,Hilgt);
             OldX := whereX; OldY := whereY;
             TextColor(Hpc);
             say(x+Fntspc,y,Hilgt[Fntspc+1]);
             say(OldX,OldY,'');
          end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    TextAttr := OldAttr;
end;

Procedure Prompt(col,row : byte; Menudata, Pmsg : str80);
begin
     if HeapAvail then
        begin
           new(NowPrompt);
           inc(Pcnt);
           with NowPrompt^ do
                begin
                  msg := Pmsg;
                  X := col; Y := row;
                  PromptNo := Pcnt;
                  Hilgt := Menudata;
                  Fntspc := 0;
           while (Hilgt[Fntspc+1] = #32)
           Inc(Fntspc);
           WrtNPrompt(NowPrompt);
           pre := LastPrompt;
        end;
     if FirstPrompt = nil then FirstPrompt := NowPrompt;
     if LastPrompt <> nil then LastPrompt^ .next := NowPrompt;
     LastPrompt := NowPrompt;
     LastPrompt^.next := nil;
     end
     else write('Heap overflow');
end;

Procedure Lkhl;
begin
     Wkbd(#27);
     Wfnk(#75);
     Wkbd(#13);
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Procedure Rkhl;
begin
     Wkbd(#27);
     Wfnk(#77);
     Wkbd(#13);
end;

Procedure Ukhl;
begin
     Wfnk(replicate(#75,Udsl));
end;

Procedure Dkhl;
begin
     Wfnk(replicate (#77,Udsl));
end;

Procedure Suds(num : byte);
begin
     Udsl := num;
     Suds(off);
end;

Procedure Menuto(var option : byte);

Procedure WrtPrompt (NowPrompt : PromptPtr);
var       OldAttr,OldWindMin,OldWindMax : word;
begin
     OldAttr := TextAttr;
     with NowPrompt^ do
          begin
            Color(Tpc,Bpc);
            say(x,y,Hilgt);
            color(Tmc,Bmc);
            OldWindMin := WindMin;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            OldWindMax := WindMax;
            window(1,1,80,25);
            gotoXY(1,25); clreol;
            write(msg+space(79-length(msg)));
            WindMin := OldWindMin;
            WindMax := OldWindMax;
            gotoXY(Fntspc+x,y);
            end;
        TextAttr := OldAttr;
end;

Procedure UpMove;
begin
     with NowPrompt^ do
     if pre = nil then
        NowPrompt := LastPrompt
     else NowPrompt := pre;
end;

Procedure DownMove;
begin
     with NOwPrompt^ do
       if next = nil then
         NowPrompt := FirstPrompt
       else Nowprompt := next;
end;

Procedure HotKey(var FirstKey : char; var NowPrompt : PromptPtr);
var        FirstCh              : char;
           OldPrompt            : PromptPtr;
           EndSearch,FirstTime : boolean;
begin
     OldPrompt := NowPrompt;
     NowPrompt := NowPrompt^.next;
     if NowPrompt = nil then NowPrompt := FirstPrompt;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      EndSearch := false;
      FirstTime := true;
      while (NowPrompt <> nil) and not EndSearch do
         with NowPrompt^ do begin
         FirstCh := Hilgt[Fntspc+1];
         if UpCase(FirstKey) = UpCase(FirstCh) then
           begin
             WrtNPrompt(OldPrompt);
             WrtPrompt(NowPrompt);
             if Hslt then FirstKey := #13;
             EndSearch := true;
           end
         else NowPrompt := next;
         if FirstTime and (not EndSearch) and (NowPrompt = nil) then
           begin
            NowPrompt := FirstPrompt;
            FirstTime := false;
           end;
         end;
         if Not EndSearch then NowPrompt := OldPrompt;
end;

begin { MenuTo }
       if FirstPrompt = nil then begin option := 0; exit; end;
       NowPrompt := FirstPrompt;
       if (option > LastPrompt^.PromptNo)
       while (NowPrompt^.PromptNo <> option) and (NOwprompt <> nil) do
         NowPrompt := Nowprompt^.next;
       if NowPrompt^.PromptNo <> option then
         NowPrompt := FirstPrompt;
       WrtPrompt(NowPrompt);
       firstKey := #00;
       while not (firstkey in [#13,#27]) do
         begin
           firstkey := readkey;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            lastkey := firstkey;
            if (firstkey = #00) and keypressed then
              begin
               WrtNPrompt(NowPrompt);
               if keypressed then begin
               lastkey := readkey;
               case lastkey of
                 #75 : if Slide then Lkhl else UpMove;
                 #77 : if Slide then Rkhl else DownMove;
                 #72 : if Uuda then UpMove else Ukhl;
                 #80 : if Uuda then DownMove else Dkhl;
                 #71 : NowPrompt := FirstPrompt;
                 #79 : NowPrompt := LastPrompt;
               end;
             end;
             if not keypressed then WrtPrompt(NowPrompt);
            end
            else if not (FirstKey in [#13,#27]) then
              HotKey(FirstKey,NowPrompt);
      end;
      WrtPrompt(NowPrompt);
      LastActive := NowPrompt^.PromptNo;
      if firstkey = #27 then option := 0
      else option := NowPrompt^.PromptNo;
      Cprompt;
      Intp;
      Hslt := true;
      Udsl := 0;
end; { MenuTo }

Procedure Clet;
begin
      while Firstget <> nil do
      begin
        NowGet     := FirstGet;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        FirstGet  := FirstGet^.next;
        dispose(NowGet);
      end;
   FirstGet := nil;
   LastGet  := nil;
end;

Procedure Slc(tcolor,bcolor : byte);
begin
    Tlc := tcolor;
    Blc := bcolor;
end;

Procedure Let(var strg; strglen : byte; Valid : CharSet);
var OldAttr  : byte;
    len      : byte absolute strg;
begin
    OldAttr := TextAttr;
    if HeapAvail then begin
       new(NowGet);
       with NowGet^ do begin
         row      := whereY;     col      := whereX;
         DataSeg  := Seg(strg);  DataOfs  := ofs(strg);
         datalen  := strglen;    ValidSet := Valid;
         move(strg,data,len+1);
         if length(data) > strglen then data := copy
         pre := LastGet;
         color(Tlc,Blc);
         say(col,row,data+space(datalen-length(data)));
       end;
        if FirstGet = nil then FirstGet := NowGet;
        if LastGet <> nil then LastGet^.next := NowGet;
        LastGet := NowGet;
        LastGet^.next := nil;
       end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        TextAttr := OldAttr;
end;

Procedure ReadLet;

Procedure ReadOneGet(x,y : byte; var strg : string;len : byte);
var       FirstTime : boolean;
          i,OldAttr : byte;

Procedure FunctionKey(var i : byte);
begin
     if keypressed then
       begin
         lastkey := readkey;
         case lastkey of
         #83 : delete(strg,i,1);
         #75 : if i > 1 then dec(i);
         #77 : if i <= length(strg) then inc(i);
         #71 : i := 1;
         #79 : begin i := length(strg); if i < len then inc(i); end;
         #72,#80 : ch := #13;
       end;
     end;
end;

begin { ReadOneGet }
      OldAttr := TextAttr;
      FirstTime := True;
      color(Tlc,Blc);
      i := length(strg)+1;
      TextAttr := TextAttr + 128;
      repeat
        say(X,Y,strg+space(len-length(strg)));
        color(Tlc,Blc);
        gotoXY(x+i-1,y);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          ch := readkey;
          firstkey := ch;
          lastkey := firstkey;
          if FirstTime then
            begin
              if not (ch in [#0..#31]) then
                begin strg := '';i := 1; end;
              end;
            FirstTime := false;
            case ch of
              #00 : FunctionKey(i);
              ^H  : if i > 1 then begin dec(i); delete(strg,i,1); end;
              ^Y  : begin strg := '';i := 1; end;
              #1..#31:; { no action }
              else begin
                if ch in NowGet^.ValidSet then
                  begin insert(ch,strg,i); inc(i); end
                else tone(500,100); end;
            end; { case }
            if i > len then begin tone(500,100); ch := #13; end;
            if length(strg) > len then delete(strg,len+1,1);
          until (ch in [#13,#27]) or (i > len);
          strg := Alltrim(strg);
          with NowGet^ do move (strg,mem[DataSeg:DataOfs]
          say(x,y,strg+space(len-length(strg)));
          TextAttr := OldAttr;
end;

begin { ReadGet }
      lastkey := #00;
      NowGet := FirstGet;
      while not (lastkey = #27) and (NowGet <> nil) do
       with NowGet^ do
         begin
           ReadOneGet(Col,row,data,datalen);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            case lastkey of
            #72 : begin
                    NowGet := pre;
                    if NowGet = nil then
                    NowGet := LastGet;
                  end;
            #80 : begin
                    NowGet := next;
                    if NowGet = nil then NowGet := FirstGet;
                  end;
            else NowGet := next;
          end;
        end;
      Clet;
end;

Function DirFiName(fname : str30) : str30;
label    fin;
const    Lx = 9;
         Uy = 7;
         Rx = 71;
         Ly = 23;
         MaxFile = (Ly-Uy-1) * ((Rx-Lx) div 15);
var      Sr                : SearchRec;
         FiName            : str30;
         OLdWdMn,OldWdMx   : word;
         option,fcount     : byte;
         SrSize,dirname    : string[40];
         dirmap            : array[1..MaxFile] of str12;
begin
     OLdWdMn := WindMin;
     OldWdMx := WindMax;
     Svcr;
     fcount :=0;
     if (fname ='') or (fname[length(fname)] in [':','\','.'])
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  then fname := fname+'*.*';
dirname := upper(FExpand (fname));
frame(double,Lx,Uy,Rx,Ly,#32+dirname+#32);
window(Lx+1,Uy+1,Rx-1,Ly-1);
findfirst(fname,directory+archive,Sr);
if doserror <> 0 then
  begin
    Wdrt(32,20,'Can''t find file !');
    repeat until keypressed;
    goto fin;
  end;
while (doserror = 0) and (fcount < MaxFile) do
  begin
    inc(fcount);
    if whereX >= Rx-17 then gotoXY(1,whereY+1);
    with Sr do begin
      dirmap[fcount] := name;
      str(size:6,SrSize);
      prompt(whereX+1,whereY,#32+name+space(13-length(name)),
          +space(16)+'File .' + name + space(19-length(name))
          'Size .' + SrSize + ' bytes');
    end;

      findnext(Sr);
    end;
  option := 1;
  Scur(off);
  Shs(off);
  Suds((Rx-Lx) div 15);
  menuto(option);
  if option <> 0 then
    begin
      FiName := dirmap[option];
      DirFiName := FExpand(FiName);
    end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else DirFiName := '';
    fin: WindMin := OldWdMn;
         WindMax := OLdWdMx;
         Rscr;
end;

BEGIN { unit menu }
      IsColorCard := (mem[$40:$49] <> mono);
      if IsColorCard then
         begin
           VideoSegment := $B800;
           Color(15,0);
           Spc(15,15,1);
           Smc(15,1);
           Slc(15,0);
           Sftc(14,5);
         end
        else
         begin
          VideoSegment := $B000;
          Color(7,0);
          Spc(15,0,15);
          Smc(0,15);
          Slc(0,15);
          Sftc(15,0);
         end;
END. { unit menu }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GPIB4.PAS

```
Program Project_1992;

Uses Crt,Dos,Printer,Menu,IEEE488;

const   MainOption    : byte    = 1;
        Quit          :boolean = false;

var     OutPutScreen : Screen;
        heaptop      : ^integer;
        hr,mr,sr,fr  : word;
        FileSave     : Text;
        FileData     : str30;

Procedure ReadYN(var ans : char);

begin
     repeat
       ans := upcase(readkey);
       if ans = #27 then exit;
     until ans in ['Y','N'];
     write(ans);
end;

Procedure GPIBmenu;
const     col = 3;
          option : byte = 1;
var       GPIBmenuScr : Screen;
          a1,a2,a3 : string[3];

Procedure GetAddr(var addr : integer);
label roj;
begin
roj : Svcr;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Scur(on);
        frame(double,39,7,61,9,' ADDRESS [0..30] ');
        a1 := '';
        gotoxy(40,8);
        let(a1,21,[#48..#57]);
        readlet;
        Scur(off);
        Rscr;
        if Lastkey = #27 then exit;
        addr := Ctol(a1);
        if (addr > 30) or (a1 = '') then
          begin
            tone(500,250);
            goto roj;
          end;
end;

Procedure InitSource;
begin
     INIT_SOURCE;
end;

Procedure InterfaceClear;
begin
     IFC;
end;

Procedure DeviceClear;
begin
     DCL;
end;

Procedure SelectDeviceClear;
begin
     GetAddr(addr);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      if Lastkey = #27 then exit;
      SEND_COMMAND (ADDR,SDC);
end;

Procedure LocalLockOut;
begin
      GetAddr(addr);
      if Lastkey = #27 then exit;
      SEND_COMMAND (ADDR,LLO);
end;

Procedure GotoLocal;
begin
      GetAddr(addr);
      if Lastkey = #27 then exit;
      SEND_COMMAND (ADDR,GTL);
end;

Procedure GroupExecuteTrigger;
begin
      GetAddr(addr);
      if Lastkey = #27 then exit;
      SEND_COMMAND (ADDR,GET);
end;

Procedure ReceiveData;
label     wi1,wi2,wi3,wn;
var       ans : char;
          L,Loop : integer;

begin
      Svcr;
      Scur(on);
wi1:  a1 := ''; a2 := ''; a3 := '';
      frame(double,39,7,62,12,' Receive Data ');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    say(41,8,'Address [0..30] : '); let(a1,2,[#48..#57]);
    say(41,9,'Bytes of data   : '); let(a2,3,[#48..#57]);
    say(41,10,'Loop [1..max]  : '); let(a3,3,[#48..#57]);
    readlet;
    if Lastkey = #27 then goto wi3;
    addr := Ctoi(a1);
    if (addr > 30) or (a1 = '') or (a2 = '') or (a3 = '') then
       begin
         tone(500,250);
         goto wi1;
       end;
    num := Ctoi(a2);
    loop := Ctoi(a3);
    say(41,11,'Save data [Y/N] : '); ReadYN(ans);
    if ans = #27 then goto wi3;
    if ans = 'Y' then
       begin
       wn: FileData := '';
         Frame(double,27,14,75,16,' Enter File to Save ');
         gotoxy(28,15);
         let(FileData,47,[#0..#255]);
         readlet;
         if Lastkey = #27 then goto wi3;
         if FileData = '' then goto wn;
         FileData := upper(FileData);
         Assign(FileSave,FileData);
       end;
  frame(double,1,14,80,24,' DATA ');
  window(2,15,79,23);
  clrscr;
  For L := 1 to Loop do
      begin
         GET_DATA (ADDR,D,num);
         GetTime(hr,mr,sr,fr);
         writeln(D);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            writeln('Time=',hr,':',mr:2,':',sr:2,':',fr:2);
            writeln;
            if BusError then goto wi2;
            if ans = 'Y' then begin
              if Exist(FileData) then append(FileSave)
              else rewrite(FileSave);
    writeln(FileSave,D,'','Time=',hr:2,':',mr:2,':',sr:2,':',fr:2);
            close(FileSave);
            end;
          end;
      repeat until keypressed;
    wi2: window(1,1,80,25);
    wi3: Rscr;
         Scur(off);
    end;

Procedure SendMessage;
begin
     GetAddr(addr);
     if Lastkey = #27 then exit;
     M := '';
     Svcr;
     Scur(on);
     frame(double,24,11,75,13,' Message ');
     gotoxy(25,12);
     let(M,50,[#0..#255]);
     readlet;
     Rscr;
     Scur(off);
     if Lastkey = #27 then exit;
     SEND_MESSAGE (ADDR,M);
end;

Procedure SerialPolling;
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      GetAddr(addr);
      if Lastkey = #27 then exit;
      SPOLLING (ADDR,DByte);
      if BusError then exit;
      Svcr;
      frame(double,44,11,57,13,' DATA ');
      window(45,12,56,12);
      write(DByte);
      repeat until keypressed;
      window(1,1,80,25);
      Rscr;
end;
begin { GPIBmenu }
      SvcrT(GPIBmenuScr);
      repeat;
      Shs(off);
      frame(single,2,7,19,18,#00);
      prompt(col,      8,' Initsource     ','Initial the interface card.
      prompt(col,whereY+1,' IFC           ','Interface clear.');
      prompt(col,whereY+1,' DCL           ','Device clear.');
      prompt(col,whereY+1,' SDC           ','Select device clear');
      prompt(col,whereY+1,' LLO           ','Local look out.');
      prompt(col,whereY+1,' GTL           ','Go to local.');
      prompt(col,whereY+1,' GET           ','Group execute trigger.');
      prompt(col,whereY+1,' Receive Data  ','Receive data from device.')
      prompt(col,whereY+1,' Send Message  ','Send message to device.');
      prompt(col,whereY+1,' Serail Polling ','Serail Polling.');
      Ssl(on);
      menuto(option);
      case option of
        1 : InitSource;
        2 : InterfaceClear;
        3 : DeviceClear;
        4 : SelectDeviceClear;
        5 : LocalLockOut;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          6 : GotoLocal;
          7 : GroupExecuteTrigger;
          8 : ReceiveData;
          9 : SendMessage;
         10 : SerialPolling;
          0 : Option := LastActive;
      end; { case }
      until (LastKey = #27) or (LastKey = #75) or (LastKey = #77);
      RscrF(GPIBmenuScr);
      Shs(on);
end;

Procedure FileMenu;
const  col = 26;
       option : byte = 1;
var    fileMenuScr : Screen;

Procedure ReadDir;
var       fname : str30;
begin
      fname := '*.*';
      Scur(on);
      Frame(double,16,15,64,17,' Enter drive and path ');
      gotoxy(17,16);
      let(fname,47,[#0..#255]);
      readlet;
      Scur(off);
      if LastKey = #27 then exit;
      FileData := DirFiName(fname);
end;

Procedure List;
type      pter = ^dataP;
          dataP = record
                    LineNo    : word;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    data      : str80;
                    pre,next  : Pter;
                  end;
var   heaplist                  : ^integer;
      firstP,lastP,nowP,topP    : Pter;
      source                    : text;
      Sfile                     : SearchRec;
      buffer                    : str80;
      OldAttr                   : word;
Procedure ReadSource(var source : text);
var       CountLine : word;
begin
     firstP := nil;
     CountLine := 0;
     while not eof(source) do
     begin
       readln(source,buffer);
       inc(CountLine);
       new(nowP);
       with nowP^ do begin
         LineNo := CountLine;
         Data   := buffer;
       end;
   if firstP = nil then
   begin
   firstP := nowP;
   firstP^ .pre := nil;
   lastp := firstP;
   lastP^.next := nil;
   end
 else
   begin
      lastp^.next := nowP;
      nowP^.pre := lastP;
      lastP := nowP;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        lastP^.next := nil;
        end;
      end;
      close(source);
end;

Procedure Display(FileData : str30);
var       count     : integer;
          ch        : char;
          OldMin,OldMax : word;

Procedure Up_one_line;
begin
     if topP^.pre <> nil then
     begin
       topP := topP^.next;
       nowP := nowP^.next;
       gotoXY(1,1); insline;
       write(topP^.data);
     end;
end;

Procedure Down_one_line;
begin
     if nowP^.next <> nil then
     begin
       nowP := nowP^.next;
       topP := topP^.next;
       gotoXY(1,23);
       writeln;
       write(nowP^.data);
       end;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Procedure Down_one_page;
begin
      count := 0;
      if nowP^.next = nil then exit;
      topP := nowP;
      clrscr;
      write(nowP^.data);
      while  (count < 22) and (nowP^.next <> nil) do
      begin
        inc(count);
        nowP := nowP^.next;
        writeln;
        write(nowP^.data);
        end;
end;

Procedure Write_one_page;
begin
      count := 0;
      clrscr;
      write(nowP^.data);
      while  (count < 22) and (nowP^.next <> nil) do
      begin
      inc(count);
      nowP := nowP^.next;
      writeln;
      write(nowP^.data);
      end;
end;

Procedure Up_one_page;
begin
      count := 0;
      while  (count < 44) and (nowP^.pre <> nil) do
        begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          inc(count);
          nowP := nowP^.pre;
        end;
      topP := nowP;
      Write_one_page;
end;

Procedure First_line;
begin
      nowP := FirstP;
      topP := FirstP;
      Write_one_page;
end;

Procedure Last_line;
begin
      count := 0;
      nowP := lastP;
      while  (count < 22) and (nowP^.pre <> nil) do
       begin
        inc(count);
       nowP := nowP^.pre;
       end;
       topP := nowP;
       Write_one_page;
end;

Function FillSpc(no : word) : str80;
var       strg : str80;
begin
      str(no,strg);
      FillSpc := strg + space(4-length(strg));
end;
begin { display }
       nowP := firstP;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        topP := firstP;
        ch := #00;
        OldMin := WindMin;
        OldMax := WindMax;
        Window(1,2,80,24);
        Write_one_page;
        while not (ch in ['Q',#27]) do
        begin
          Wdrt(70,1, 'Line : '+ FillSpc(topP^.LineNo));
          Wdrt(70,25,'Line : '+ FillSpc(NowP^.LineNo));
          ch := upcase(readkey);
          if ch = ^C then Halt(1);
          if (ch = #00) and keypressed then
           begin
             lastkey := readkey;
             case lastkey of
               #72 : Up_one_line;
               #80 : Down_one_line;
               #73 : Up_one_page;
               #81 : Down_one_page;
               #71 : First_line;
               #79 : Last_line;
             end;
           end;
        end;
        WindMin := OldMin;
        WindMax := OldMax;
end;

Procedure AssignFile(FileData : str30);
var       OldAttr : word;
begin
     OldAttr := TextAttr;
     if IsColorCard then
     begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            textcolor(15);
            textbackground(1);
        end
    else
        begin
          textcolor(0);
          textbackground(7);
        end;
    gotoXY(1,25); clreol;
   write ('Command >> ⇑ Up ⇓ Down ⇟ PgDn ⇞ Home ⇟ End Q)uit ⇑');
    gotoXY(1,1); clreol;
    write('Reading File << ',FExpand(FileData));
    assign(source,FileData);
    reset(source);
    mark(heaplist);
    readsource(source);
    say(1,1,'File  >>  '+FileData); clreol;
    TextAttr := OldAttr;
end;
begin { List }
      clrscr;
      OldAttr := TextAttr;
      textcolor(14);
      textbackground(0);
      Scur(on);
      FileData := '';
      gotoXY(1,1);
      write('Enter file to display >'); clreol;
      Wdrt(16,3,'Press ENTER to dislay files ESC to cancel.)');
      let(FileData,56,[#0..#255]);
      readlet;
      Wdrt(16,3,'                                                    ');
      Scur(off);
      TextAttr := OldAttr;
      if Lastkey = #27 then exit;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if FileData = '' then FileData := DirFiName(FileData);
        if Lastkey = #27 then exit;
        FileData := upper(FileData);
        if Exist(FileData) then
          begin
            AssignFile(FileData);
            display(FileData);
            release(heaplist);
          end
          else
            begin
              tone(500,100);
              Wdrt(32,20,' File not found ! ');
              repeat until keypressed;
              end;
end;

Procedure Print;
var       PrintOut : Text;
          DataOut  : str80;
          OldAttr  : word;
          ans      : char;
begin
     Svcr;
     Scur(on);
     FileData := '';
     Frame(double,16,15,64,17,' Enter File to Print ');
     Wdrt(16,19,'(Press ENTER to display files or ESC to cancel.)');
     gotoxy(17,16);
     let(FileData,47,[#0..#255]);
     readlet;
     Scur(off);
     Rscr;
     if Lastkey = #27 then exit;
     if FileData = '' then FileData := DirFiName(FileData);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      if lastkey = #27 then exit;
      Frame(double,19,15,61,17,'');
      gotoxy(20,16);
      Scur(on);
      write(' Is your print ready? READY [Y/N] : ');
      ReadYN(ans);
      Scur(off);
      if ans = 'Y' then
         begin
          Frame(double,16,15,64,17,' Print File ');
          gotoxy(17,16);
          OldAttr := TextAttr;
          TextAttr := TextAttr+128;
          write(FExpand(FileData));
          TextAttr := OldAttr;
            if Exist(FileData) then
             begin
              assign(PrintOut,FileData);
              reset(PrintOut);
             while not EOF(PrintOut) do
               begin
                readln(PrintOut,DataOut);
               writeln(LST,DataOut);
              end;
          close(PrintOut);
         end
       else
         begin
           tone(500,100);
           Wdrt(32,20,'File not found !');
           repeat until keypressed;
         end;
      end;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Procedure ChangeDir;
var       DirName : DirStr;
begin
     GetDir(0,DirName);
     Scur(on);
     Frame(double,16,15,64,17,' NEW DIRECTORY ');
     GotoXY(17,16); let(DirName,47,[#0..#255]);
     readlet;
     Scur(off);
     if Lastkey <> #27 then
       begin
        {$I-} ChDir(DirName); {$I+}
        if IOresult <> 0 then Wdrt(32,20,'Invalid directory');
       end;
end;

Procedure EraseFile;
var       DelName   : str30;
          f         : file;
          ans       : char;
          dir       : DirStr;
          name      : NameStr;
          ext       : ExtStr;
          OldAttr   : word;
begin
     Svcr;
     Scur(on);
     DelName := '';
     Frame(double,16,15,64,17,'Enter File Name ');
     Wdrt(16,19,'(Press ENTER to display files or ESC to cancel.)');
     GotoXY(17,16); let(DelName,47,[#0..#255]);
     readlet;
     Scur(off);
     Rscr;
     if Lastkey <> #27 then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      begin
            if DelName = '' then DelName := DirFiName(DelName);
            if DelName <> '' then
                    begin
                      Sftc(142,4);
                      Frame(double,20,15,60,17,' Warning ');
                      FSplit(DelName,dir,name,ext);
                      tone(900,400);
                      Scur(on);
                      Say(22,16,'File >  ' + name + ext +' Erase (Y/N): ')
                      ReadYN(ans);
                      Scur(off);
                      if ans = 'Y' then
                      begin
                       assign(f,DelName);
                       {$I-} Erase(f); {$I+}
                       if IOresult <> 0 then
                       begin
                        tone(500,200);
                        Wdrt(30,23,'Can''t Erase This File');
                        repeat until keypressed;
                       end;
                    end;
               end;
          end;
end;

begin { FileMenu }
      SvcrT(FileMenuScr);
      Frame(single,col-1,7,col+11,13,#00);
      prompt(col,       8,' Directory ','Display files.');
      prompt(col,whereY+1,' List      ','Display data from file.');
      prompt(col,whereY+1,' Print     ','Print data from file.');
      prompt(col,whereY+1,' ChDir     ','Change directory.');
      prompt(col,whereY+1,' Erase     ','Erase file.');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Ssl(on);
        MenuTo(option);
        case option of
          1 : ReadDir;
          2 : List;
          3 : Print;
          4 : ChangeDir;
          5 : EraseFile;
          6 : option := LastActive;
        end;
        RscrF(FileMenuScr);
end;
Procedure OptionMenu;
const     col = 49;
          option : byte = 1;
var       OptionScr : Screen;

Procedure TimeSet;
label     tmn;
var       hh,mm,ss :string[2];
begin
tmn: Scur(on);
     hh := ''; mm := ''; ss := '';
     frame(double,45,13,60,17,' Enter Time ');
     say(47,14,'Hour   : '); let(hh,2,[#48..#57]);
     say(47,15,'Minute : '); let(mm,2,[#48..#57]);
     say(47,16,'Second : '); let(ss,2,[#48..#57]);
     readlet;
     Scur(off);
     if Lastkey = #27 then exit;
     hr := Ctol(hh);
     mr := Ctol(mm);
     sr := Ctol(ss);
     if (hr > 23) or (mr > 59) or (sr > 59) then goto tmn;
     SetTime(hr,mr,sr,0);
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Procedure DateSet;
var         yr,mn,dy : word;
            y1,m1,d1 : string[2];
begin
      Scur(on);
      d1 := '';m1 := '';y1 := '';
      frame(double,46,13,59,17, 'Enter Date ');
      say(48,14,'Day   : '); let(d1,2,[#48..#57]);
      say(48,15,'Month : '); let(m1,2,[#48..#57]);
      say(48,14,'Year  : '); let(y1,2,[#48..#57]);
      readlet;
      Scur(off);
      if Lastkey = #27 then exit;
      dy := Ctol(d1);
      mn := Ctol(m1);
      yr := Ctol(y1);
      SetDate(yr,mn,dy);
end;

begin { OptionMenu }
      SvcrT(OptionScr);
      frame(single,col-1,7,col+8,10,#00);
      prompt(col,       8,'  Time  ','Set real time to system.');
      prompt(col,whereY+1,'  Date  ','Set real date to system.');
      Ssl(on);
      menuto(option);
      case option of
       1 : TimeSet;
       2 : DateSet;
       0 : option := LastActive;
  end;
  RscrF(OptionScr);
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Procedure QuitMenu;
const       col = 73;
            option : byte = 1;
var         QuitScr : Screen;
begin { QuitMenu }
      SvcrT(QuitScr);
      frame(single,col-1,7,col+6,10,#00);
      prompt(col,       8,' No  ','Cancel.');
      prompt(col,WhereY+1,' Yes ','Exit program.');
      Ssl(on);
      menuto(option);
      case option of
        1 : Quit := false;
        2 : Quit := true;
        0 : option := LastActive;
   end;
   RscrF(QuitScr);
end;

Procedure MainMenu;
begin { MAinmenu }
      Suda(off);
      prompt( 3,5, ' GPIB ','> General Purpose Interface Bus command <')
 prompt(26,5,' File ','> Directory . List . Print . Chdir . Erase <');
 prompt(49,5,' Option ','> Time . Date <');
 prompt(73,5,' Quit ','> Exit Program <');
 MenuTo(MainOption);
 case MainOption of
   1 : GPIBmenu;
   2 : Filemenu;
   3 : OptionMenu;
   4 : QuitMenu;
   0 : MainOption := LastActive;
  end;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BEGIN { main }
      checksnow := true;
      directvideo := true;
      mark(heaptop);
      clrscr;
      Scur(off);
      frame(single,1,1,80,4,#00);
      textcolor(10);
      say(22,2,'PARALLEL Interface Card for IBM PC');
      say(35,3, 'Version 1.0');
      if IsColorCard then color(0,7);
      repeat MainMenu;
      until Quit;
      release(heaptop);
      color(15,0);
      clrscr;
      textmode (c40);
     { frame(single,1,1,40,4,#00);}
      textcolor(30);
      say(10,2,' ELECTRICAL ENGINEERING ');
      say(18,4,'KMITL');
      say(17,8,'THE END');
      gotoxy(1,6);
      Scur(on);
      readln;
      textmode(c80);
END.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IEEE488.pas

```
Unit IEEE488;

{ IEEE 488 Interfacing Card for IBM PC version 1.0  }
{          Adviser.....Supakit Jutaviriya           }
{          And.........Wijit Kinnaret               }
{          By..........Jirasak Ieosakun             }
{          And.........Tharapong Rujiwong           }

Interface

Uses       Crt,Menu;

const      GTL     = $01;      { Go to Local                }
           SDC     = $04;      { Select Device Clear        }
           PPC     = $05;      { Parallel Pull Configure    }
           GET     = $08;      { Group Exicute Trigger      }
           TCT     = $09;      { Take Control               }
           LLO     = $11;      { Local Lock Out             }
           PPU     = $15;      { Parallel Poll Enable       }
           SPE     = $18;      { Serial Poll Enable         }
           SPD     = $19;      { Serial Poll Disable        }
           UNL     = $3F;      { Unlisten                   }
           UNT     = $5F;      { Untalk                     }

type       DataFormat  =  string[50];

var        BusError       :  Boolean;
           i              :  integer;
           Addr           :  integer;

           MLA            :  byte;        { My Listen Address  }
           MTA            :  byte;        { My Talk Address    }
           M              :  DataFormat;  { Remote message     }
           D              :  DataFormat;  { Data               }
           num            :  integer;
           DByte          :  byte;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    option        :  byte;

  { portA      =  $310
    portB      =  $311
    portC1     =  $312
    control1   =  $313
    portC2     =  $316
    control2   =  $317 }

procedure Address (Addr :integer ; var MLA,MTA : byte);
procedure Source_Operation (Data : byte);
procedure Init_Source;
procedure Send_Command (Addr : integer ; Command : byte);
procedure Send_Message (Addr : integer ; M : DataFormat);
procedure Acceptor_Operation (var DByte : byte);
procedure Get_Data (Addr : integer ; var D : DataFormat ; Num : integer)
procedure IFC;
procedure DCL;
procedure SPolling (Addr : integer ; var Dbyte : byte);

Implementation

procedure Address (Addr :integer ; var MLA,MTA : byte);
  begin
     case Addr of
        0  :  begin  MLA  :=  $20  ;  MTA  :=  $40  ;  end;
        1  :  begin  MLA  :=  $21  ;  MTA  :=  $41  ;  end;
        2  :  begin  MLA  :=  $22  ;  MTA  :=  $42  ;  end;
        3  :  begin  MLA  :=  $23  ;  MTA  :=  $43  ;  end;
        4  :  begin  MLA  :=  $24  ;  MTA  :=  $44  ;  end;
        5  :  begin  MLA  :=  $25  ;  MTA  :=  $45  ;  end;
        6  :  begin  MLA  :=  $26  ;  MTA  :=  $46  ;  end;
        7  :  begin  MLA  :=  $27  ;  MTA  :=  $47  ;  end;
        8  :  begin  MLA  :=  $28  ;  MTA  :=  $48  ;  end;
        9  :  begin  MLA  :=  $29  ;  MTA  :=  $49  ;  end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        10  :  begin  MLA  :=  $2A  ;  MTA  :=  $4A  ;  end;
        11  :  begin  MLA  :=  $2B  ;  MTA  :=  $4B  ;  end;
        12  :  begin  MLA  :=  $2C  ;  MTA  :=  $4C  ;  end;
        13  :  begin  MLA  :=  $2D  ;  MTA  :=  $4D  ;  end;
        14  :  begin  MLA  :=  $2E  ;  MTA  :=  $4E  ;  end;
        15  :  begin  MLA  :=  $2F  ;  MTA  :=  $4F  ;  end;
        16  :  begin  MLA  :=  $30  ;  MTA  :=  $50  ;  end;
        17  :  begin  MLA  :=  $31  ;  MTA  :=  $51  ;  end;
        18  :  begin  MLA  :=  $32  ;  MTA  :=  $52  ;  end;
        19  :  begin  MLA  :=  $33  ;  MTA  :=  $53  ;  end;
        20  :  begin  MLA  :=  $34  ;  MTA  :=  $54  ;  end;
        21  :  begin  MLA  :=  $35  ;  MTA  :=  $55  ;  end;
        22  :  begin  MLA  :=  $36  ;  MTA  :=  $56  ;  end;
        23  :  begin  MLA  :=  $37  ;  MTA  :=  $57  ;  end;
        24  :  begin  MLA  :=  $38  ;  MTA  :=  $58  ;  end;
        25  :  begin  MLA  :=  $39  ;  MTA  :=  $59  ;  end;
        26  :  begin  MLA  :=  $3A  ;  MTA  :=  $5A  ;  end;
        27  :  begin  MLA  :=  $3B  ;  MTA  :=  $5B  ;  end;
        28  :  begin  MLA  :=  $3C  ;  MTA  :=  $5C  ;  end;
        29  :  begin  MLA  :=  $3D  ;  MTA  :=  $5D  ;  end;
        30  :  begin  MLA  :=  $3E  ;  MTA  :=  $5E  ;  end;
    end;
end;

procedure Source_Operation(Data : byte);
  begin
    Port[$312] := (Port[$312] OR $0C);        { make DAV=H   EOI=H }
    if (Port[$312] or $CF) = $FF then         { check  NRFD=NDAC=H }
      begin
        BusError := true;
        tone(300,300);
        wdrt (68,25,'BUS ERROR !');
        delay(1000);
        exit;
      end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    BusError := false;
    Port[$310] := $FF - Data;              { put Data on DIO lines          }
       repeat
       until ((Port[$312] or $EF) =$FF) or KeyPressed;       { check NRFD=H
    Port[$312] := (Port[$312] AND $F7); { make DAV=L                        }
       repeat
       until ((Port[$312] or $DF) = $FF) or KeyPressed; { check NDAC=H }
    Port[$312] := (Port[$312] or $0C);    { make DAV=H EOI=H                }
  end;

 procedure Init_Source;
 begin
      Port[$317] := $88;                   { CU2i CL20                      }
      Port[$313] := $88;                   { Alo  Blo  Culi  Cllo }
      Port[$311] := $0B;                   { 160=T;DAV=T,NRFD=NDAC=R }
      Port[$310] := $FF;                   { clear data }
 end;

 procedure Send_Command (Addr : integer ; Command : byte);
 begin
      Port[$316] := $02;                   { make REN=L IFC=H ATN=L }
      Address (Addr,MLA,MTA);
      Source_Operation (MLA);
      if BusError then exit;
      Source_Operation (Command);
 end;

 procedure Send_Message (Addr : integer ; M : DataFormat);
 var     message : char;
 begin
      Port[$316] := $02;                   { make PEN=L  IFC=H  ATN=L}
      Address (Addr,MLA,MTA);
      Source_Operation (MLA);
      if BusError then exit;
      Port[$316] := $03;                   { make PEN=L  IFC=H  ATN=H}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      for i :=1 to length(M) do          { send messages }
        begin
         message := M[i];
         Source_Operation(integer (message));
        end;
      Source_OPeration ($0D);             { send CR delimiter }
      Source_Operation ($0A);             { send LF delimiter }
end;

procedure Acceptor_Operation (var DByte : byte);
begin
      Port[$312] := $D0;                  { make NDAC=L NRFD=H }
        repeat
        until ((Port[$312] and $08) <> $08) or KeyPressed; {check DAV=L }
      Port[$312] := $C0;                  { make NDAC=L NRFD=L }
      DByte      := $FF - Port[$310];     { accept Data 1 byte }
      Port[$312] := $E0;                  { make NDAC=H NRFD=L }
        repeat
        until ((Port[$312] and $08) = $08) or KeyPressed;  { check DAV=H
      Port[$312] := $C0;
end;

procedure Get_Data (Addr : integer ; var D : DataFormat ; Num : integer)
begin
      Port[$316] := $02;                  { make REN=L IFC=H ATN=L }
      Address (Addr,MLA,MTA);
      Source_Operation (MTA);
      if BusError then exit;
      Port[$313] := $91;                  { ALi Blo CUo CLlo          }
      Port[$311] := $00;                  { 160=R;DAV=R,NRFD=NDAC=T   }
      Port[$312] := $C0;                  { make RDAC=L NRFD=L        }
      Port[$316] := $03;                  { make REN=L  IFC=H  ATN=H  }
      D := '';                            { clear Data  }
      for i := 1 to num do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          begin
               Acceptor_Operation (DByte);
               D := D+Char(DByte);
          end;
     Port[$313] := $88;                    { Alo  Blo  CUli  CLlo     }
     Port[$311] := $0B;                    { 160=T;DAV=T,NRFD=NDAC=R  }
     Port[$316] := $02;                    { make REN=L  IFC=H  ATN=L }
end;

procedure IFC;
begin
     Port[$316] := $00;                    { make REN=L  IFC=L  ATN=L }
end;

procedure DCL;
begin
     Port[$316] := $02;                    { make REN=L  IFC=H  ATN=L }
     Source_Operation ($14);
end;

procedure SPolling (Addr : integer ; var Dbyte : byte);
begin
     Port[$316] := $02;                    { make REN=L  IFC=H  ATN=L }
     Address (Addr,MLA,MTA);
     Source_Operation (UNL);
     if BusError then exit;
     Source_Operation (MTA);
     Source_Operation (SPE);
     Port[$313] := $91;                    { Ali  Blo  CUlo  CLli     }
     Port[$311] := $00;                    { 160=R;DAV=R,NRFD=NDAC=T  }
     Port[$312] := $C0;                    { make NDAC=L  NRFD=L      }
     Port[$316] := $03;                    { make REN=L  IFC=H  ATN=H }
     Acceptor_Operation (DByte);
     Port[$313] := $88;                    { Alo  Blo  CUli  CLlo     }
     Port[$311] := $0B;                    { 160=T;DAV=T,NRFD=NDAC=R  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Port[$316] := $02;                  { make REN=L  IFC=H  ATN=L }
        Port[$312] := $0C;                  { make DVA=H  EOI=H        }
        Source_Operation (SPD);
        source_Operation (UNT);
    end;

    begin
        BusError   := false;
        Port[$317] := $88;
        Init_source;
        IFC;
        DCL;
    end. { IEEE488 Unit }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

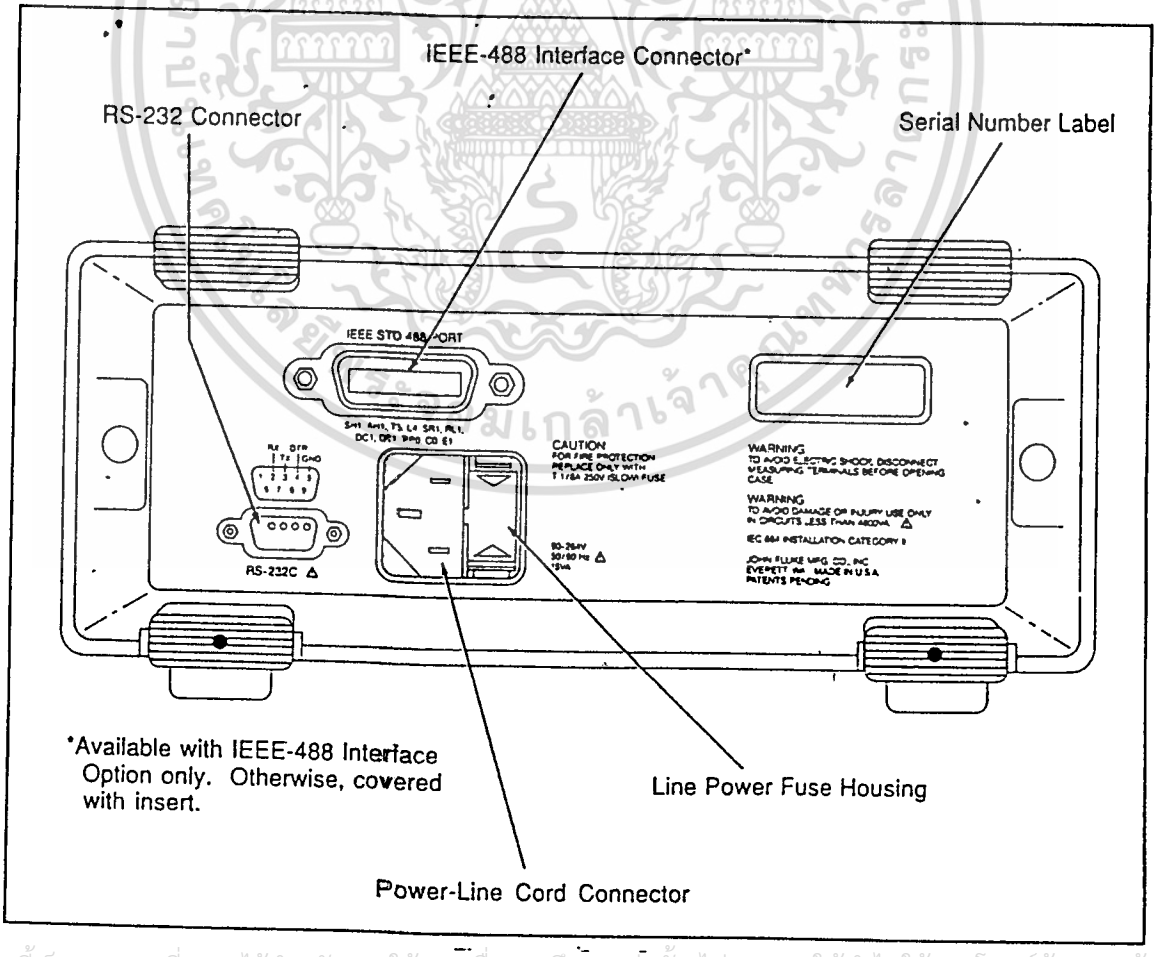
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GPIB (IEEE488)

General Purpose Interface Bus

บัสอินเตอร์เฟส มาตรฐานและการใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

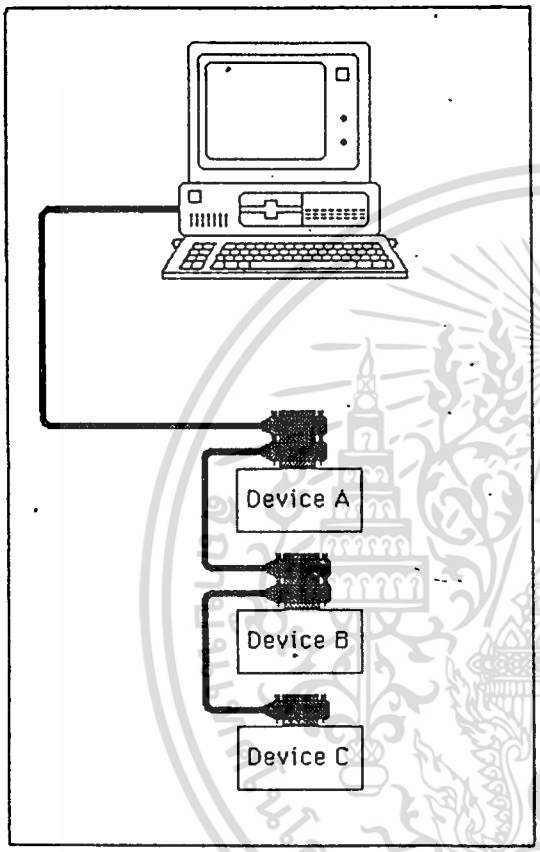


Figure 2.
Linear Configuration

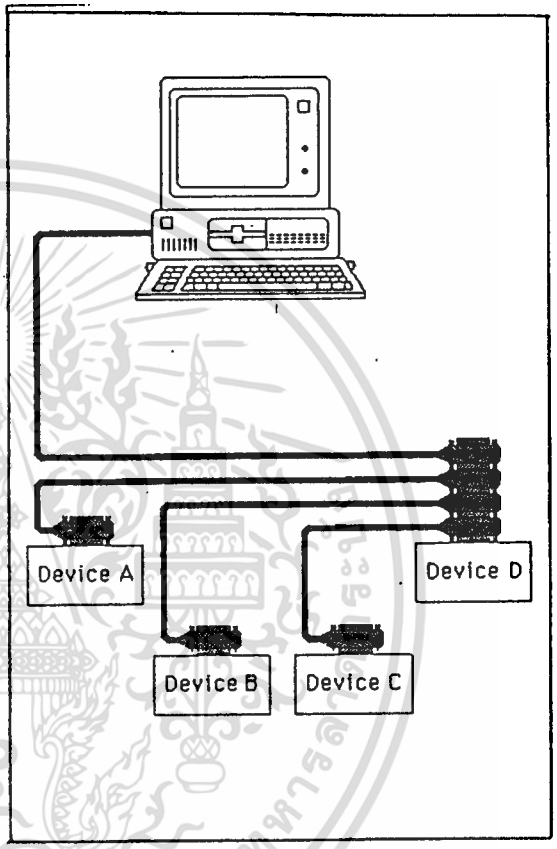
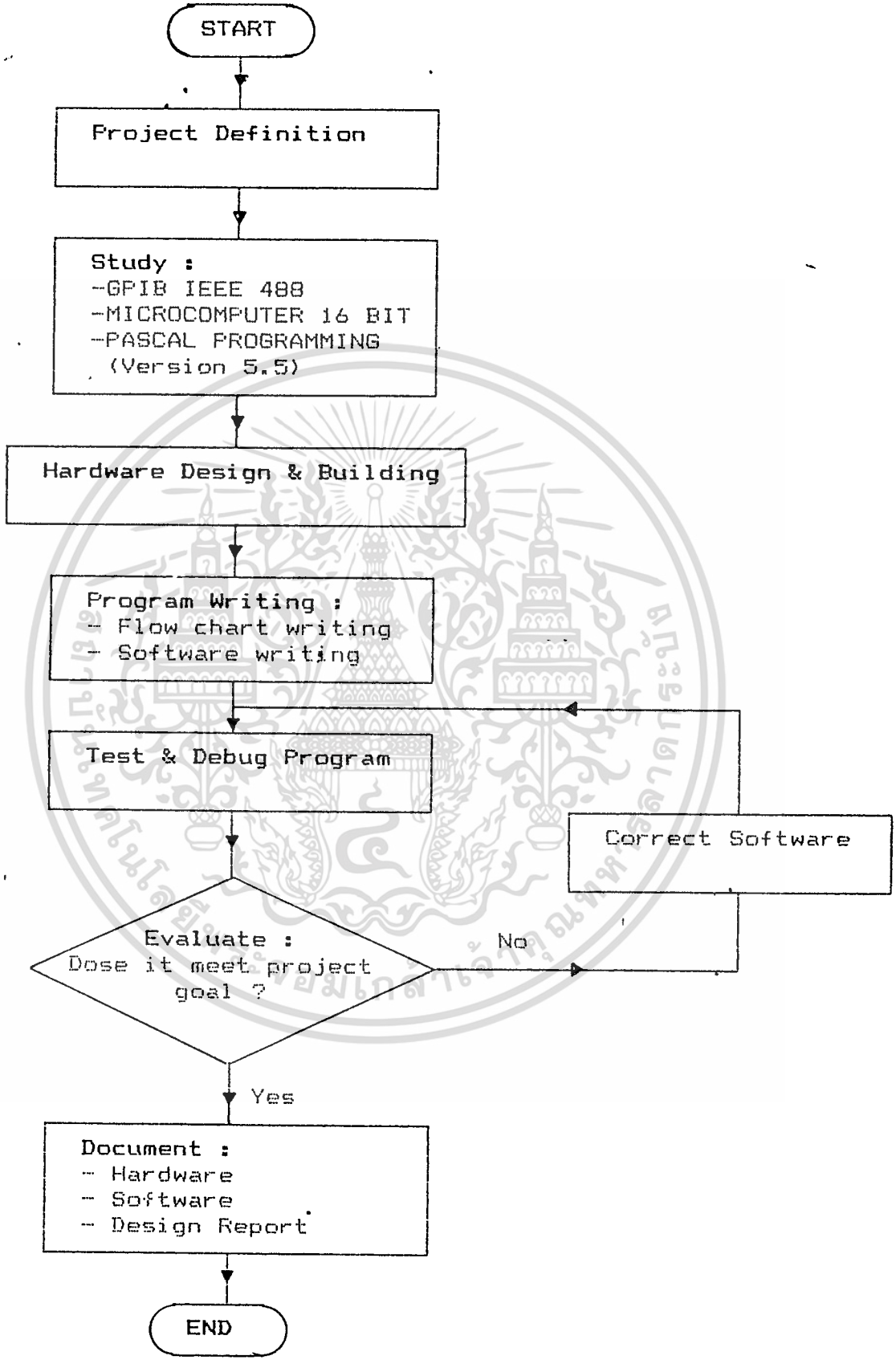


Figure 3.
Star Configuration

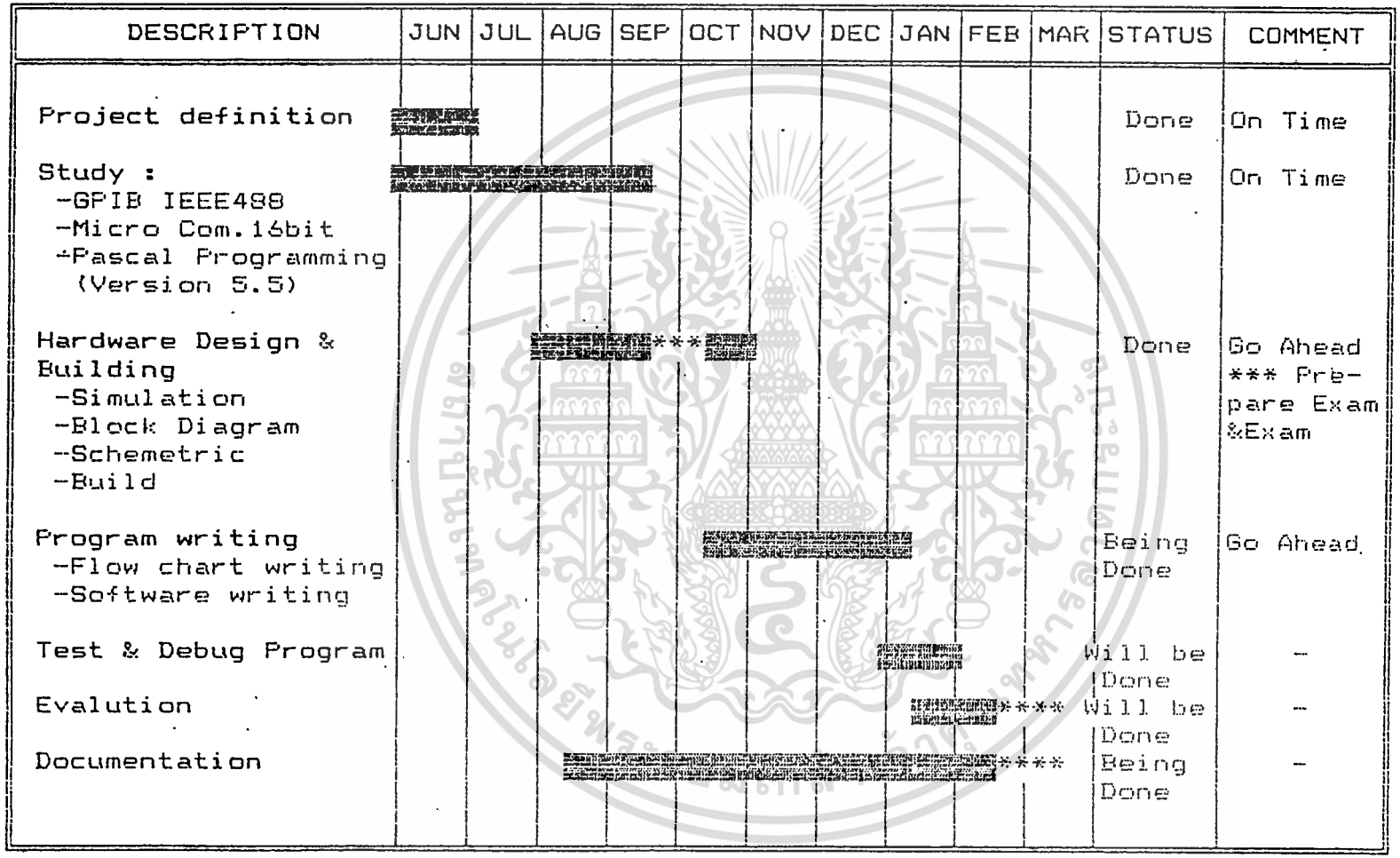
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROJECT FLOW CHART



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Schedual Project Planning



DESCRIPTION	JUN	JUL	AUG	SEP	OCT	NOV	DEC	JAN	FEB	MAR	STATUS	COMMENT
Project definition											Done	On Time
Study : -GPIB IEEE488 -Micro Com.16bit -Pascal Programming (Version 5.5)											Done	On Time
Hardware Design & Building -Simulation -Block Diagram -Schemetric -Build				***							Done	Go Ahead *** Pre-pare Exam &Exam
Program writing -Flow chart writing -Software writing											Being Done	Go Ahead.
Test & Debug Program											Will be Done	-
Evalution									****		Will be Done	-
Documentation										****	Being Done	-

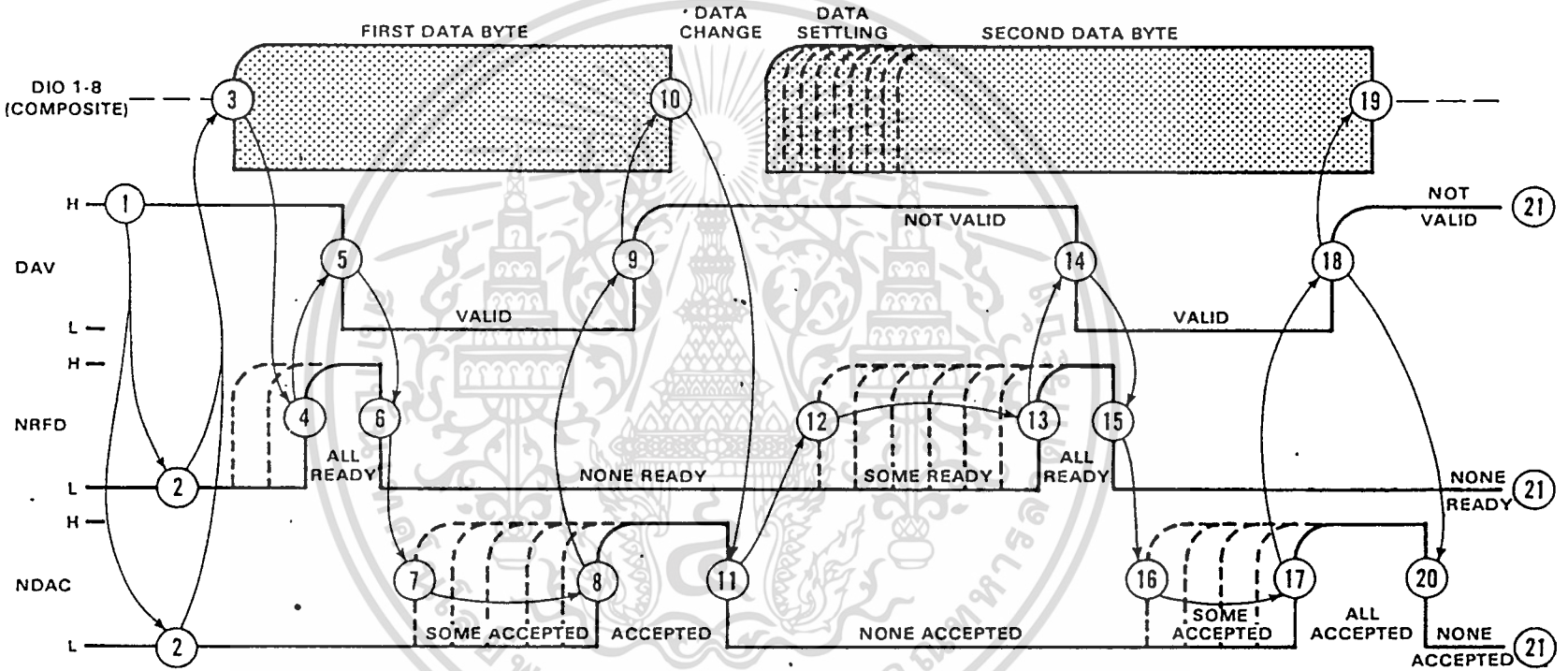


Figure 2-50. GPIB Three Wire Handshake state diagram.

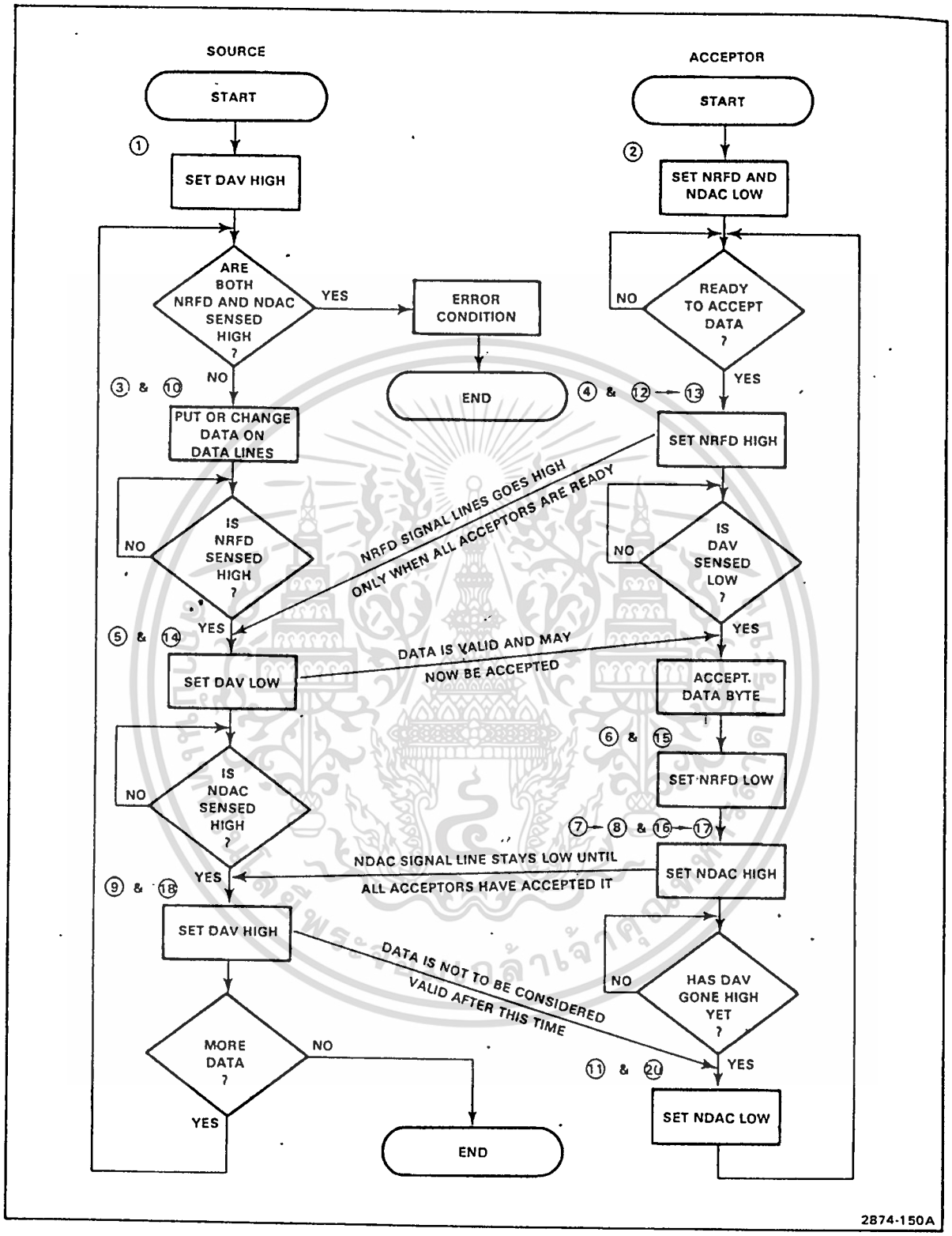
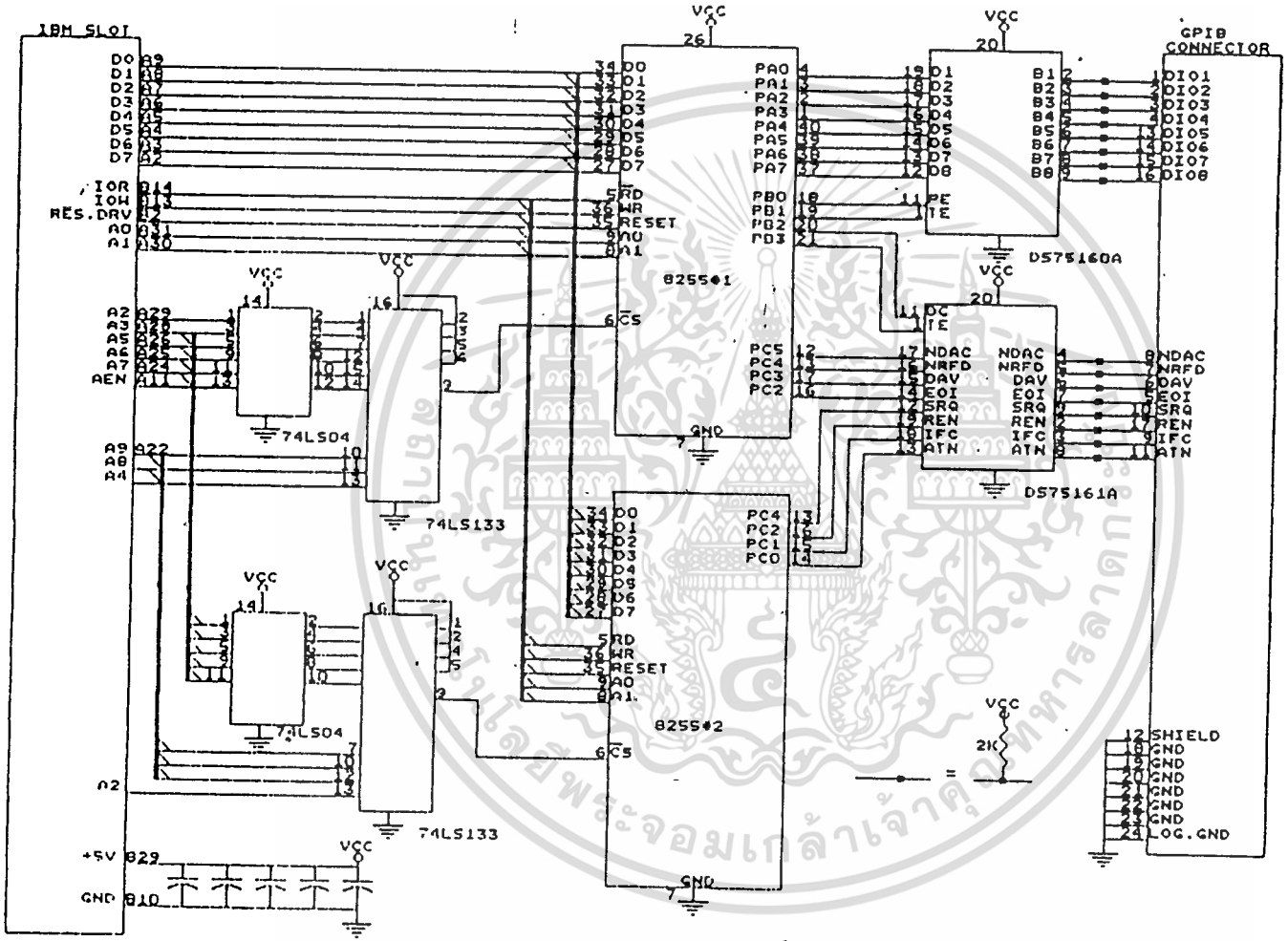
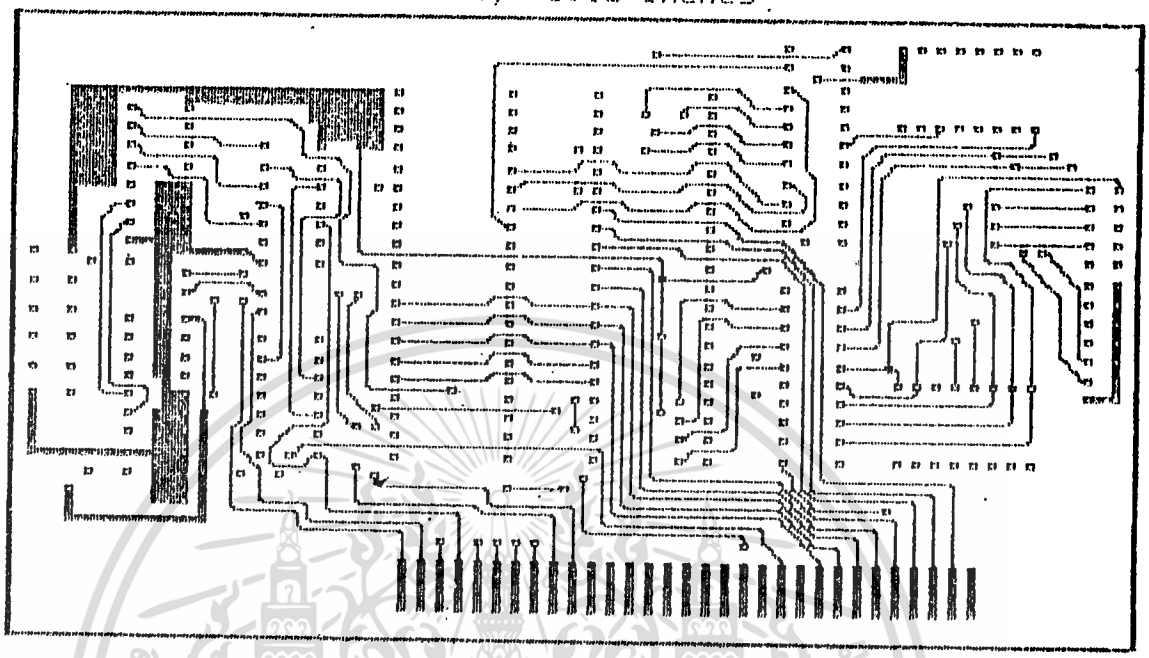


Figure 2-51. GPIB Handshake flowchart.

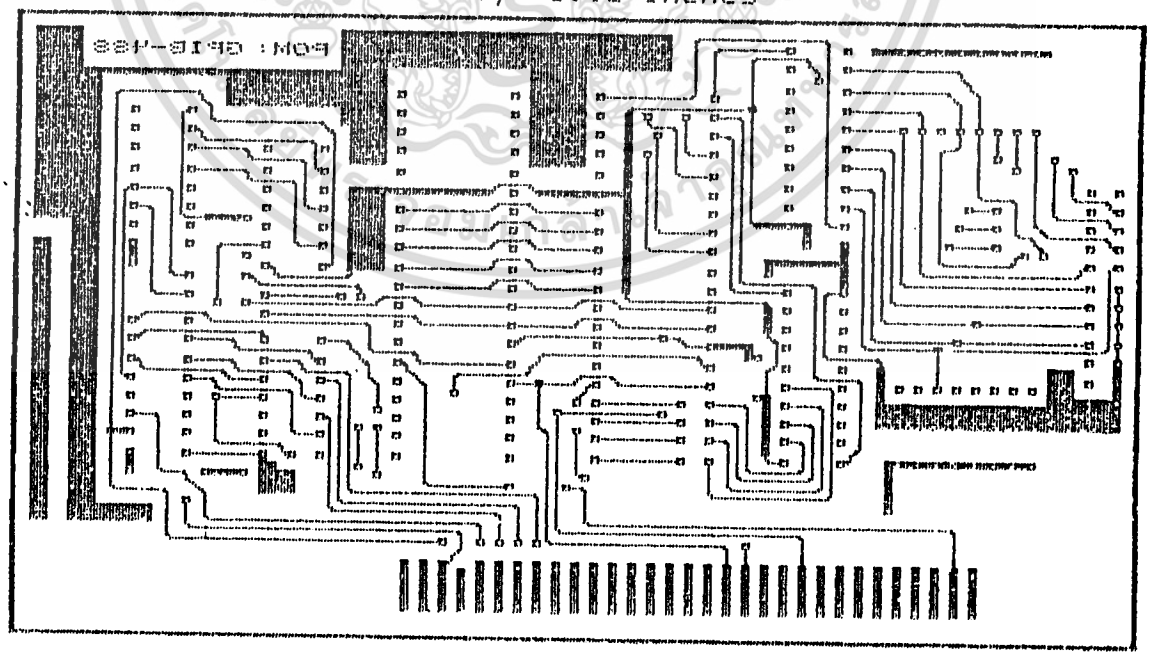
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


IBM SLOT
GPIB CONNECTOR
8255#1
8255#2
74LS04
74LS133
DS75160A
DS75161A
VCC
GND
+5V
RES.DRV
IOR
IOW
AEN
NDAC
NRFD
DAV
EOI
SRQ
REN
IFC
ATN
SHIELD
LOG.GND

IX checkplot 15 Mar 1992 17:27:43
pol
v1.2 r3 holes: 326 component side
approximate size: 5.75 by 3.05 inches



IX checkplot 15 Mar 1992 17:25:13
pol
v1.2 r3 holes: 326 solder side
approximate size: 5.75 by 3.05 inches



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5404/7404 Hex Inverter

	Schottky TTL					High-Speed TTL					Low-Power Schottky TL					Standard TTL					Low-Power TTL				
	Device Type	Package				Device Type	Package				Device Type	Package				Device Type	Package				Device Type	Package			
		C	P	M	CF		C	P	M	CF		C	P	M	CF		C	P	M	CF		C	P	M	CF
T. I.	SN54S04	J①			W①	SN54H04	J①			W①	SN54LS04	J①			W②	SN5404	J①			W②	SN54L04	J①	N①	T②	
	SN74S04	J①/N①				SN74H04	J①/N①				SN74LS04	J①/N①				SN7404	J①/N①				SN74L04	J①/N①	T②	
FAIRCHILD	FM54S04/FM9S04	D①				FM54H04/FM9H04	D①			F②	FM54LS04/FM9LS04	F①			F①	FM5404/FM9N04	D①			F②					
	FC74S04/FC9S04	D①/M①				FC74H04/FC9H04	D①/P①				FC74LS04/FC9LS04	D①/M①			F①	FM7404/FC9N04	D①/P①								
MOTOROLA						MC3108	L①			F①						MC5404	L①			F②					
						MC3008	L①/P①			F①						MC7404	L①/P①								
N. S. C.						DM54H04	J①/N①				SN74LS04	P①				DM54S04	J①/N①			W②	DM54L04	J①/N①			F②
	DM74S04		N①			DM74H04	J①/N①				DM54LS04					DM74S04	J①/N①				DM74L04	J①/N①			F②
											DM74LS04														
PHILIPS	N74S04		①			N74H04		①			N74LS04	①				FJH241/7404		①							
SIGNETICS	S54S04	F①	A①		W①	S54H04	F①	A①		W②						S5404	F①	A①		W②					
	N74S04	F①	A①		W①	N74H04	F①	A①			N74LS04	A①				N7404	F①	A①							
SIEMENS																FLH211		①							
FUJITSU											74LS04	M①				MB418	①	M①							
HITACHI	HD74S04	①	P①								HD74LS04	P①				HD7404/HD2522	①	P①							
MITSUBISHI	M5S004		P①								M74LS04	P①				M53204		P①							
NEC	74S04		C①								74LS04	C①				μPB235	D①								
TOSHIBA																TD3404A		P①							

Electrical Characteristics SN54LS04/SN74LS04

absolute maximum ratings over operating free-air temperature range

Supply voltage V_{CC}	7V	Operating free-air temperature range	SN54LS	−55°C to 125°C
Input voltage	7V		SN74LS	0°C to 70°C
		Storage temperature range		−65°C to 150°C

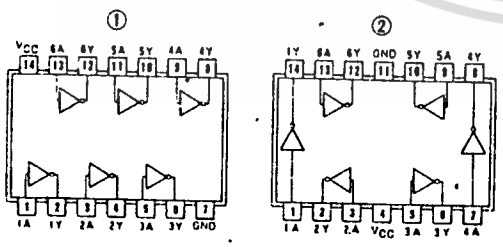
recommended operating conditions

	SN54LS04			SN74LS04			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			−400			−400	μA
Low-level output current, I_{OL}			4			8	mA
Operating free-air temperature, T_A	−55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range

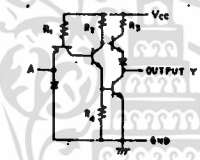
PARAMETER		TEST CONDITIONS †		MIN	TYP ‡	MAX	UNIT
V_{IH}	High-level input voltage			2			V
V_{IL}	Low-level input voltage					0.8	V
V_I	Input clamp voltage	V_{CC} = MIN, I_I = −18mA				−1.5	V
V_{OH}	High-level output voltage	V_{CC} = MIN, V_{IL} = V_{IL} max, I_{OH} = MAX		2.7	3.4		V
V_{OL}	Low-level output voltage	V_{CC} = MIN, V_{IH} = 2V, I_{OL} = 4mA				0.4	V
I_I	Input current at maximum input voltage	V_{CC} = MAX, V_I = 7V				0.1	mA
I_{IH}	High-level input current	V_{CC} = MAX, V_{IH} = 2.7V				20	μA
I_{IL}	Low-level input current	V_{CC} = MAX, V_{IL} = 0.4V				−0.4	mA
I_{OS}	Short-circuit output current *	V_{CC} = MAX	54LS Family	−20		−100	mA
			74LS Family	−20		−100	
I_{CCH}	Supply current	V_{CC} = MAX	Total, outputs high		1.2	2.4	mA
I_{CCL}	Supply current		Total, outputs low		3.6	6.6	mA
I_{CC}	Supply current	V_{CC} = 5V	Average per gate (50% duty cycle)		0.4		mA
t_{PLH}	Propagation delay time, low-to-high-level output	V_{CC} = 5V, C_L = 15PF,	T_A = 25°C, R_L = 2KΩ		9	15	ns
t_{PHL}	Propagation delay time, high-to-low-level output				10	15	ns

Pin Assignments (Top View)



positive logic :
$Y = \overline{A}$

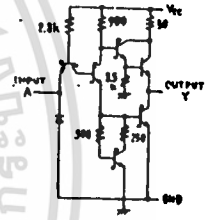
Schematics (each gate)



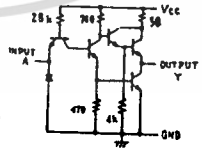
CIRCUIT	R1	R2	R3	R4
'04	4k	1.6k	130	1k
'L04	40k	20k	500	12k

Input clamp diodes not on SN54L/SN74L' circuits.

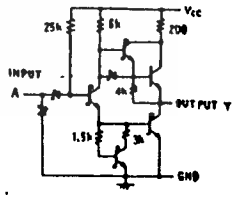
'04, 'L04 CIRCUITS



'S04 CIRCUIT



'H04 CIRCUIT



'L804 CIRCUIT

Resistor values shown are nominal and in ohms.

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡ All typical values are at V_{CC} = 5V, T_A = 25°C.
* Not more than one output should be shorted at a time, and for SN54H'/SN74H' and SN54S'/SN74S', duration of short-circuit should not exceed 1 second.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

54133/74133 13-Input Positive-NAND Gate

	Schottky TTL					High-Speed TTL					Low-Power Schottky TTL					Standard TTL					Low-Power TTL				
	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF
	SN54S133	J †			NJ																				
	SN74S133	J †	N †																						
FAIRCHILD	μA9S133 FW9S133	DJ			FJ						54LS133 FW9LS133	DJ			FJ										
	74S133 FC9S133	DJ	PJ								74LS133 FC9LS133	DJ	PJ		FJ										
MOTOROLA											SN74LS133		PJ												
NSC	DM74S133																								
SIGNETICS	S54S133																								
	N74S133																								
SIEMENS																									
FUJITSU																									
HITACHI	HD74S133	J	P †																						
MITSUBISHI	M54S133		MJ																						
NEC																									
TOSHIBA																									

Electrical Characteristics SN54S133 SN74S133

absolute maximum ratings over operating free-air temperature range

Supply voltage V_{CC}	7V	Operating free-air temperature range	SN54S	−55°C to 125°C
Input voltage	5.5V		SN74S	0°C to 70°C
Off-state output voltage	5.5V	Storage temperature range		−65°C to 150°C

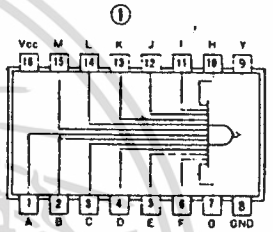
recommended operating conditions

	SN74S133 MIN	NOM	MAX	SN54S133 MIN	NOM	MAX	UNIT
Supply voltage V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			−1000			−1000	μA
Low-level output current, I_{OL}			20			20	mA
Operating free-air temperature T_A	−55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range

PARAMTERR		TEST CONDITIONS†		MIN	TYP‡	MAX	UNIT
V_{IH}	High-level input voltage			2			V
V_{IL}	Low-level input voltage					0.8	V
V_I	Input clamp voltage	V_{CC} = MIN, I_I = −18mA				−1.2	V
V_{OH}	High-level output voltage	V_{CC} = MIN, V_{IL} = V_{IL} max, I_{OH} = MAX	54 Family	2.5	3.4		V
			74 Family	2.7	3.4		
V_{OL}	Low-level output voltage	V_{CC} = MIN, V_{IH} = 2 V, I_{OL} = MAX				0.5	V
I_I	Input current at maximum input voltage	V_{CC} = MAX, V_I = 5.5V				1	mA
I_{IH}	High-level input current	V_{CC} = MAX, V_{IH} = 2.7V				50	μA
I_{IL}	Low-level input current	V_{CC} = MAX, V_{IL} = 0.5V				−2	mA
I_{OS}	Short-circuit output current	V_{CC} = MAX		−40		−100	mA
I_{CCH}	Supply current	V_{CC} = MAX	Total, outputs high		3	5	mA
I_{CCL}	Supply current		Total, outputs low		5.5	10	mA
I_{CC}	Supply current	V_{CC} = 5 V	Average per gate (50% duty cycle)		4.25		mA
t_{PLH}	Propagation delay time, low-to-high-level output	V_{CC} = 5 V, T_A = 25°C	C_L = 15pF, R_L = 280Ω	2	4	6	ns
			C_L = 50pF, R_L = 280Ω		5.5		
t_{PHL}	Propagation delay time, high-to-low-level output		C_L = 15pF, R_L = 280Ω	2	4.5	7	ns
			C_L = 50pF, R_L = 280Ω		6.5		

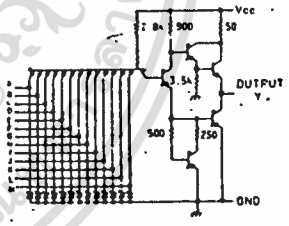
Pin Assignment (Top View)



positive logic:

$Y = \overline{ABCDEFGHIJKLM}$

Schematic(each gate)



'S133 CIRCUIT

Resistor values shown are nominal and in ohms.

†For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡All typical values are at V_{CC} = 5 V, T_A = 25°C.
§Not more than one output should be shorted at a time, and duration of short-circuit should not exceed 1 second.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

National Semiconductor

Bus Transceivers

DS75160A/DS75161A/DS75162A IEEE-488 GPIB Transceivers

General Description

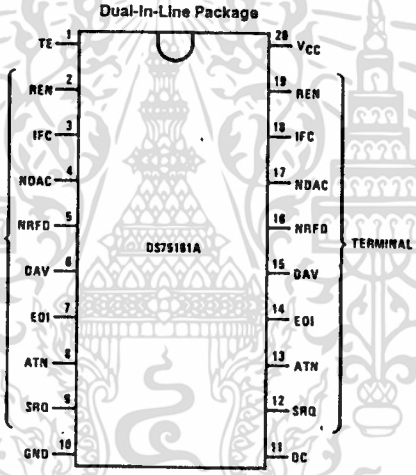
This family of high-speed-Schottky 8-channel bi-directional transceivers is designed to interface TTL/MOS logic to the IEEE Standard 488-1978 General Purpose Interface Bus (GPIB). PNP inputs are used at all driver inputs for minimum loading, and hysteresis is provided at all receiver inputs for added noise margin. The IEEE-488 required bus termination is provided internally with an active turn-off feature which disconnects the termination from the bus when V_{CC} is removed. A power up/down protection circuit is included at all bus outputs to provide glitch-free operation during V_{CC} power up or down.

The General Purpose Interface Bus is comprised of 16 signal lines—8 for data and 8 for interface management. The data lines are always implemented with DS75160A, and the management lines are either implemented with DS75161A in a single-controller system, or with DS75162A in a multi-controller system.

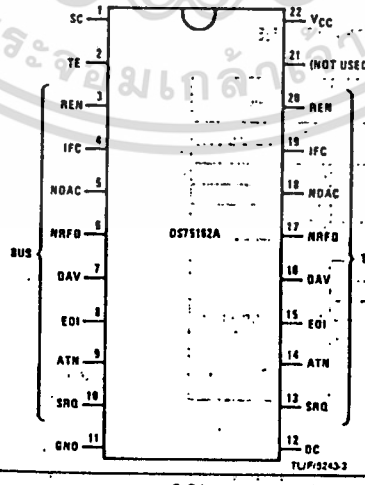
Features

- 8-channel bi-directional non-inverting transceivers
- Bi-directional control implemented with TRI-STATE® output design
- Meets IEEE Standard 488-1978
- High-speed Schottky design
- Low power consumption
- High impedance PNP inputs (drivers)
- 500 mV (typ) input hysteresis (receivers)
- On-chip bus terminators
- No bus loading when V_{CC} is removed
- Power up/down protection (glitch-free)
- Pin selectable open collector mode on DS75160A driver outputs
- Accommodates multi-controller systems

Dual-In-Line Package



Dual-In-Line Package



Order Number DS75160AN, DS75161AN or DS75162AN
See NS Package N20A or N22A

"TRI-STATE" is a registered trademark of National Semiconductor Corp.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description

DS75160A

This device is an 8-channel bi-directional transceiver with one common direction control input, denoted TE. When used to implement the IEEE-488 bus, this device is connected to the eight data bus lines, designated DIO_1–DIO_8. The port connections to the bus lines have internal terminators, in accordance with the IEEE-488 Standard, that are deactivated when the device is powered down. This feature guarantees no bus loading when $V_{CC} = 0V$. The bus port outputs also have a control mode that either enables or disables the active upper stage of the totem-pole configuration. When this control input, denoted PE, is in the high state, the bus outputs operate in the high-speed totem-pole mode. When PE is in the low state, the bus outputs operate as open collector outputs which are necessary for parallel polling.

DS75161A

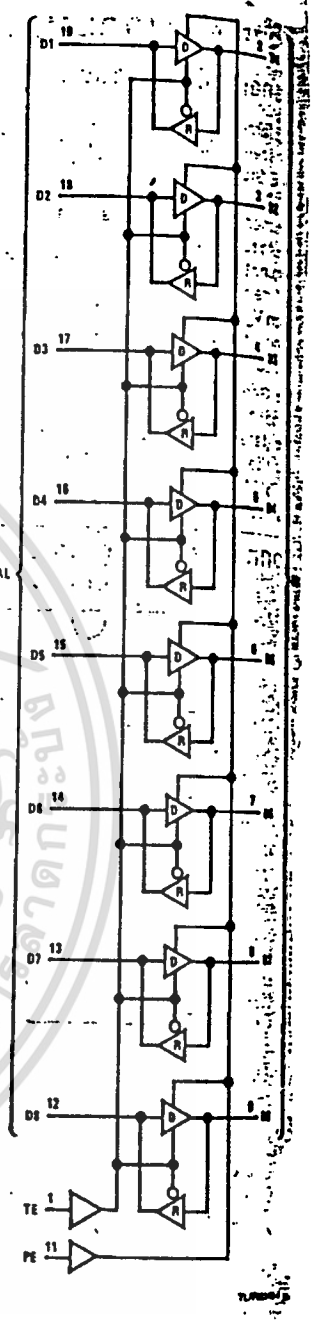
This device is also an 8-channel bi-directional transceiver which is specifically configured to implement the eight management signal lines of the IEEE-488 bus. This device, paired with the DS75160A, forms the complete 16-line interface between the IEEE-488 bus and a single controller instrumentation system. In compliance with the system organization of the management signal lines, the SRQ, NDAC, and NRFD bus port outputs are open collector. In contrast to the DS75160A, these open collector outputs are a fixed configuration. The direction control is divided into three groups. The DAV, NDAC, and NRFD transceiver directions are controlled by the TE input. The ATN, SRQ, REN, and IFC transceiver directions are controlled by the DC input. The EOI transceiver direction is a function of both the TE and DC inputs, as well as the logic level present on the ATN channel. The port connections to the bus lines have internal terminators identical to the DS75160A.

DS75162A

This device is identical to the DS75161A, except that an additional direction control input is provided, denoted SC. The SC input controls the direction of the REN and IFC transceivers that are normally controlled by the DC input on the DS75161A. This additional control function is instrumental in implementing multiple controller systems.

Logic Diagrams

DS75160A



Note 1: Denotes driver

Note 2: Denotes receiver

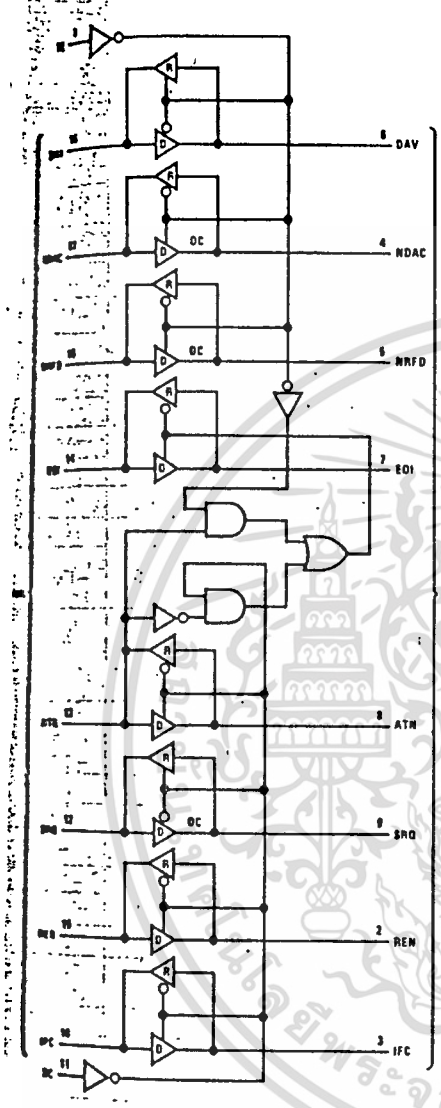
Note 3: Driver and receiver outputs are totem-pole configured

Note 4: The driver outputs of DS75160A can have their active disabled by switching the PE input (pin 11) to the logic low mode configures the outputs as open collector.

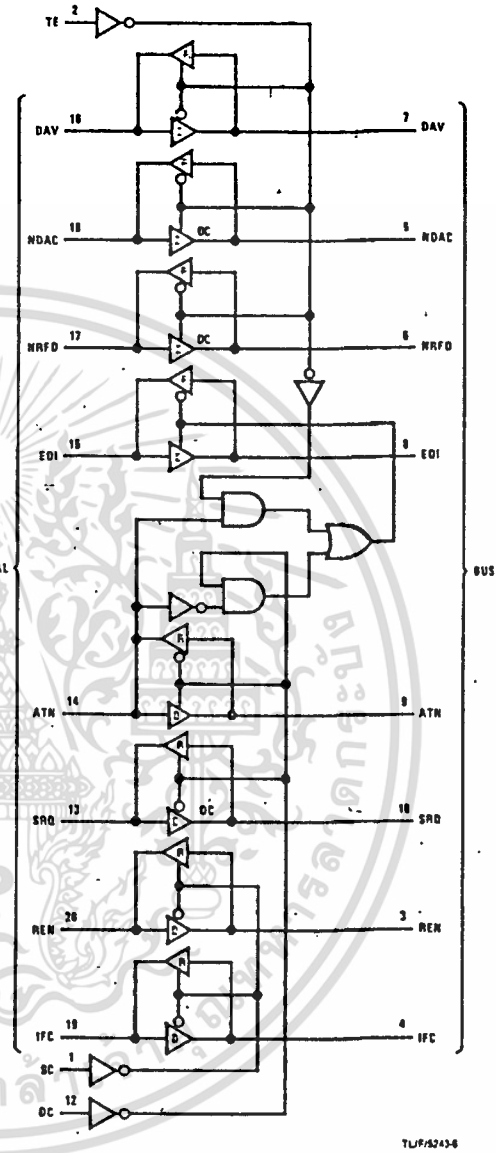
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Connection Diagrams (Continued)

DS75161A



DS75162A



Note 1: Denotes driver

Note 2: Denotes receiver

Note 3: Symbol "OC" specifies open collector output

Note 4: Driver and receiver outputs that are not specified "OC" are totem-pole configurations

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings (Note 1)

Supply Voltage, V_{CC}	7.0V
Input Voltage	5.5V
Storage Temperature Range	−65°C to 150°C
Maximum Power Dissipation* at 25°C Molded Package	1897 mW
Lead Temperature (Soldering, 10 seconds)	300°C

Operating Conditions

	Min
V_{CC}, Supply Voltage	4.75
T_A, Ambient Temperature	0
I_{OL}, Output Low Current Bus Terminal	

*Derate molded package 15.2 mW/°C above 25°C.

Electrical Characteristics (Notes 2 and 3)

Parameter			Conditions		Min	Typ	
V_{IH}	High-Level Input Voltage				2		
V_{IL}	Low-Level Input Voltage						
V_{IK}	Input Clamp Voltage		$I_I = -18$ mA			−0.8	−1.5
V_{HYS}	Input Hysteresis	Bus			400	500	
V_{OH}	High-Level Output Voltage	Terminal	$I_{OH} = -800\ \mu A$		2.7	3.5	
		Bus (Note 5)	$I_{OH} = -5.2$ mA		2.5	3.4	
V_{OL}	Low-Level Output Voltage	Terminal	$I_{OL} = 16$ mA			0.3	0.5
		Bus	$I_{OL} = 48$ mA			0.4	0.5
I_{IH}	High-Level Input Current	Terminal and TE, PE, DC, SC Inputs	$V_I = 5.5V$			0.2	100
			$V_I = 2.7V$			0.1	20
I_{IL}	Low-Level Input Current		$V_I = 0.5V$			−10	−100
V_{BIAS}	Terminator Bias Voltage at Bus Port	Bus	Driver Disabled	$I_{I(bus)} = 0$ (No Load)	2.5	3.0	3.7
I_{LOAD}	Terminator Bus Loading Current		Driver Disabled	$V_{I(bus)} = -1.5V$ to 0.4V	−1.3		
				$V_{I(bus)} = 0.4V$ to 2.5V	0		−3.2
				$V_{I(bus)} = 2.5V$ to 3.7V			2.5
				$V_{I(bus)} = 3.7V$ to 5V	0		−3.2
				$V_{I(bus)} = 5V$ to 5.5V	0.7		2.5
			$V_{CC} = 0$, $V_{I(bus)} = 0V$ to 2.5V				2.5
							40
I_{OS}	Short-Circuit Output Current	Terminal	$V_I = 2V$, $V_O = 0V$ (Note 4)		−15	−35	−75
		Bus (Note 5)			−35	−75	−150
I_{CC}	Supply Current	DS75160A	Transmit, TE = 2V, PE = 2V, $V_I = 0.8V$			85	125
			Receive, TE = 0.8V, PE = 2V, $V_I = 0.8V$			70	100
		DS75161A	TE = 0.8V, DC = 0.8V, $V_I = 0.8V$			84	125
		DS75162A	TE = 0.8V, DC = 0.8V, SC = 2V, $V_I = 0.8V$			85	125
C_{IN}	Bus-Port Capacitance	Bus	$V_{CC} = 5V$ or 0V, $V_I = 0V$ to 2V, f = 1 MHz			20	30

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: Unless otherwise specified, min/max limits apply across the 0°C to +70°C temperature range and the 4.75V to 5.25V power supply range. All typical values are for $T_A = 25°C$ and $V_{CC} = 5.0V$.

Note 3: All currents into device pins are shown as positive; all currents out of device pins are shown as negative; all voltages are referenced to ground unless otherwise specified. All values shown as max or min are so classified on absolute value basis.

Note 4: Only one output at a time should be shorted.

Note 5: This characteristic does not apply to outputs on DS75161A and DS75162A that are open collector.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Switching Characteristics $V_{CC} = 5.0V \pm 5\%$, $T_A = 0°C$ to $70°C$ (Note 1)

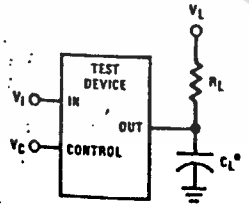
Parameter	From	To	Conditions	DS75160A			DS75161A			DS75162A			Units
				Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Propagation Delay Time, Low to High Level Output	Terminal	Bus	$V_L = 2.3V$, $R_L = 38.3\Omega$, $C_L = 30$ pF, *Figure 1*		10	20		10	20		10	20	ns
Propagation Delay Time, High to Low Level Output					14	20		14	20		14	20	ns
Propagation Delay Time, Low to High Level Output	Bus	Terminal	$V_L = 5.0V$, $R_L = 240\Omega$, $C_L = 30$ pF, *Figure 2*		14	20		14	20		14	20	ns
Propagation Delay Time, High to Low Level Output					10	20		10	20		10	20	ns
Output Enable Time to High Level	TE, DC, or SC (Note 2) (Note 3)	Bus	$V_I = 3.0V$, $V_L = 0V$, $R_L = 480\Omega$, $C_L = 15$ pF, *Figure 1*		19	32		23	40		23	40	ns
Output Disable Time from High Level					15	22		15	25		15	25	ns
Output Enable Time to Low Level			$V_I = 0V$, $V_L = 2.3V$, $R_L = 38.3\Omega$, $C_L = 15$ pF, *Figure 1*		24	35		28	48		28	48	ns
Output Disable Time from Low Level					17	25		17	27		17	27	ns
Output Enable Time to High Level	TE, DC, or SC (Note 2) (Note 3)	Terminal	$V_I = 3.0V$, $V_L = 0V$, $R_L = 3\ k\Omega$, $C_L = 15$ pF, *Figure 1*		17	33		18	40		18	40	ns
Output Disable Time from High Level					15	25		22	33		22	33	ns
Output Enable Time to Low Level			$V_I = 0V$, $V_L = 5V$, $R_L = 280\Omega$, $C_L = 15$ pF, *Figure 1*		25	39		28	52		28	52	ns
Output Disable Time from Low Level					15	27		20	35		20	35	ns
Output Pull-Up Enable Time (DS75160A Only)	PE (Note 2)	Bus	$V_I = 3V$, $V_L = 0V$, $R_L = 480\Omega$, $C_L = 15$ pF, *Figure 1*		10	17		NA			NA		ns
Output Pull-Up Disable Time (DS75160A Only)					10	15		NA			NA		ns

Note 1: Typical values are for $V_{CC} = 5.0V$ and $T_A = 25°C$ and are meant for reference only.

Note 2: Refer to Functional Truth Tables for control input definition.

Note 3: Test configuration should be connected to only one transceiver at a time due to the high current stress caused by the V_I voltage source when the output connected to that input becomes active.

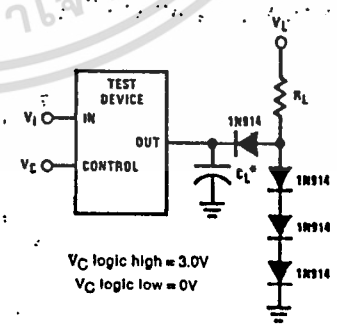
Switching Load Configurations



V_C logic high = 3.0V
V_C logic low = 0V

*C_L includes jig and probe capacitance

FIGURE 1 TL/F/5243-7



V_C logic high = 3.0V
V_C logic low = 0V

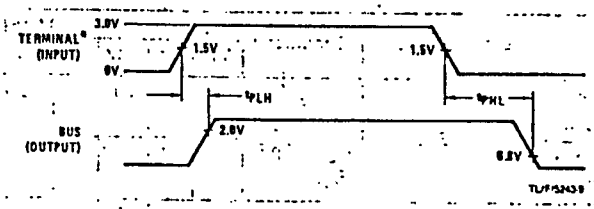
*C_L includes jig and probe capacitance

FIGURE 2 TL/F/5243-8

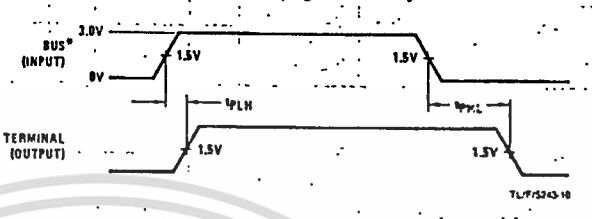
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Switching Waveforms

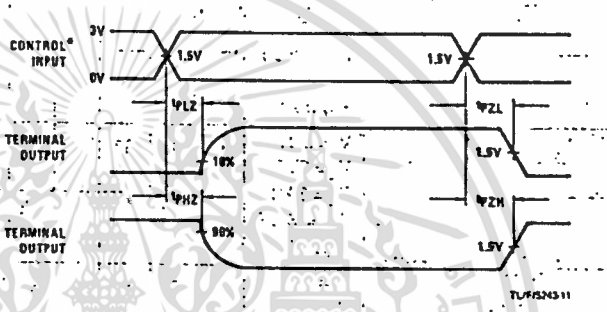
Transmit Propagation Delays



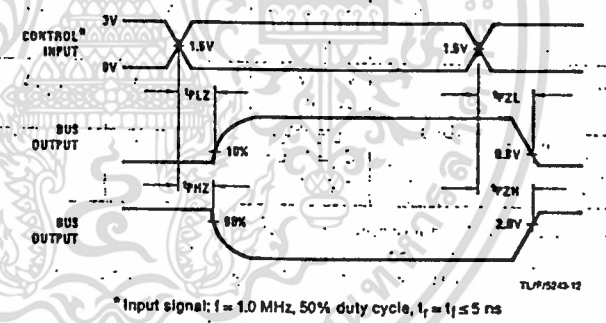
Receive Propagation Delays



Terminal Enable/Disable Times



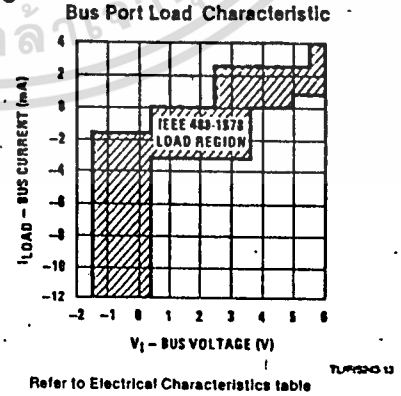
Bus Enable/Disable Times



*Input signal: f = 1.0 MHz, 50% duty cycle, $t_r = t_f \leq 5$ ns

Performance Characteristics

Bus Port Load Characteristic



Refer to Electrical Characteristics table

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ıal Truth Tables

DS75160A

Control Input Level		Data Transceivers	
TE	PE	Direction	Bus Port Configuration
H	H	T	Totem-Pole Output
H	L	T	Open Collector Output
L	X	R	Input

DS75161A

Control Input Level			Transceiver Signal Direction							
TE	DC	ATN*		EOI	REN	IFC	SRQ	NRFD	NDAC	DAV
H	H		R		R	R	T	R	R	T
H	L		T		T	T	R	R	R	T
L	H		R		R	R	T	T	T	R
L	L		T		T	T	R	T	T	R
H	X	H		T						
L	X	H		R						
X	H	L		R						
X	L	L		T						

DS75162A

Control Input Level				Transceiver Signal Direction							
SC	TE	DC	ATN*		EOI	REN	IFC	SRQ	NRFD	NDAC	DAV
H	H	H		R		T	T	T	R	R	T
H	H	L		T		T	T	R	R	R	T
H	L	H		R		T	T	T	T	T	R
H	L	L		T		T	T	R	T	T	R
L	H	H		R		R	R	T	R	R	T
L	H	L		T		R	R	R	R	R	T
L	L	H		R		R	R	T	T	T	R
L	L	L		T		R	R	R	T	T	R
X	H	X	H		T						
X	L	X	H		R						
X	X	H	L		R						
X	X	L	L		T						

H = High level input
L = Low level input
X = Don't care
T = Transmit, i.e., signal outputted to bus
R = Receive, i.e., signal outputted to terminal
* The ATN signal level is sensed for internal multiplex control of EOI transmission direction logic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการทำโปรเจคครั้งนี้ ซึ่งได้บรรลุผลสำเร็จเป็นอย่างดี ก็เพราะได้รับความช่วยเหลือและสนับสนุนเป็นอย่างดีจากผู้ที่มีรายชื่อดังต่อไปนี้

1. อาจารย์ ศุภกิจ จูตะวิริยะ และอาจารย์ วิจิตร กิณเรศ ผู้ซึ่งเป็นอาจารย์ที่ปรึกษาและสนับสนุนทั้งทางด้าน soft ware และ hard ware ที่ใช้ประกอบการทดลอง

2. รศ.ดร.โยธิน เปรมปราณีย์รัชต์ หัวหน้าภาควิศวกรรมระบบควบคุม และอาจารย์เกียรติวรรณ ทรงสัตย์ ที่ให้ความเอื้อเฟื้อในเรื่องเครื่องมือวัด(Function Generator)

3. คุณ วิชัย นอกเมือง ตำแหน่ง "Technical Support Manager" ของ National Semiconductor Bangkok Ltd. ซึ่งเป็นผู้อนุญาติให้ผู้จัดทำนำเครื่องมือ (Digital Multimeter) ใช้ประกอบการทดลองโปรเจค

4. คุณ อำนาจ วงษ์เนตร นักศึกษาภาคอิเล็กทรอนิกส์ ที่ให้ความช่วยเหลือในเรื่องเครื่องคอมพิวเตอร์ ในการพัฒนาซอฟแวร์และการจัดทำเอกสารตลอดการจัดทำโปรเจคในครั้งนี้

ขอขอบพระคุณเป็นอย่างยิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. Borland International, " Reference Guide and User's Guide ", Turbo Pascal,(1980).

2. Peter R. Rony, Microcomputer Interfacing with 8255 PPI chip, (1980).

3. Eugone Fisher and c.w. Jensen, PET/CMB and IEEE488 bus(GPIB) 2 nd Edition

4. Sevith, Waher J., Turbo pascal " An introduction to the art and science of programming"

5. Gondstrein, Larry Joel; Turbo pascal "An introduction to modern programming"

6. Paul A.Sand," Advanced Pascal programming Techniques ", Turbo Pascal,(1982).

7. บุญเลิศ เอี่ยมทัศนา," คู่มือเทอร์โบปาสคาล รุ่น 4.0-5.0 ",ISBN,974-509-009-3,2521

8. บุญเลิศ เอี่ยมทัศนา," เรียนรู้ภาษาปาสคาล ด้วยเทอร์โบปาสคาล 4.0-5.0 ", ISBN 974-509-020-4

9. กนก เจนจิรยงค์เวช," บัสอินเทอร์เฟส ",วารสารเซมิคอนดัคเตอร์อิเล็กทรอนิกส์, ฉบับที่ 78-79,หน้าที่ 214-218 และ 196-201,2530

10. พ.ต.ประพัฒน์ อุทโยภาศ," เรียนเทอร์โบปาสคาลด้วยตนเอง",ISBN 974-7967-52-9,2521

11. ยืน ภู่วรวรรณ," เทคโนโลยีฮาร์ดแวร์ IBM PC ",ISBN 974-509-167-7,2521

12. จิติ หนูแก้ว ," เทคนิคการเชื่อมต่อ IBM PC กับอุปกรณ์ภายนอกเพื่อประยุกต์ใช้ในงานต่างๆ ",ISBN 974-509-317-3,2521

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้