ปีการศึกษา 2532

การวิจัยและพัฒนาโครงข่ายคอมพิวเตอร์เนตเวอร์คเพื่อใช้ในการศึกษา

(M.C.S. NETWORKS)

โดย

นายนพร พรมพาทิชย์ เลขประจำตัว 291085

นายบัณฑิต เกียรติสิงห์นคร เลขประจำตัว 291096

อาจารย์ที่ปรึกษา

ดร. รัตติกร วราทุลศิริพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา ๒๕๓๓

ภาควิชา อิเลคทรอนิคส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

เรื่อง การวิจัยและพัฒนาโครงข่ายคอมพิวเตอร์เนตเวอร์คเพื่อใช้ในการศึกษา

ผู้จัดทำ

๑.นาย นพร พรมพานิชย์ เลขประจำตัว ๒๗.๑๐๘๕

๒.นาย บัณฑิต เกียรติสิงห์นคร เลขประจำตัว ๒๗.๑๐๙๖

.....อาจารย์ที่ปรึกษา

(อ.รัตติกร วรากูลศิริพันธ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

026946

การพัฒนาและออกแบบไมโครคอมพิวเตอร์เพื่อใช้ในการเรียนการสอน

นพร พรมพานิชย์

บัณฑิต เกียรติสิงห์นคร

ดร.รัตติกร วราภูลสิริพันธ์ อาจารย์ที่ปรึกษา

ปีการศึกษา ๒๕๓๒

บทคัดย่อ

โครงงานนี้ เป็นโครงงานที่เกี่ยวกับการเชื่อมต่อไมโครคอมพิวเตอร์ขึ้นเป็นระบบเพื่อนำไปใช้ในห้องเรียน โดยจัดให้มีการโต้ตอบกันได้ระหว่างผู้สอนและผู้เรียนโดยผ่านทางหน้าจอแสดงผล ช่วยให้ผู้เรียนทุกคนสามารถเห็นข้อความปรากฏขึ้นอย่างชัดเจน

หลักการทำงานจะอาศัยการส่งรับผ่านทางพอร์ท อาร์เอส-๒๓๒ (RS-232) ซึ่งเป็นพอร์ทส่งรับแบบอนุกรม (serial) ที่มีอยู่แล้วในคอมพิวเตอร์มาใช้งาน โดยต่อแบบไม่ใช้โมเด็ม (non-modem) ดังนั้นจึงไม่ต้องผ่านสายโทรศัพท์ และผู้จัดทำได้ทำการ์ด (card) ขึ้นมาเพิ่มเติมเพื่อใช้ในการควบคุม ภายในการ์ดมีส่วนโปรโตคอล (protocol),ส่วนควบคุมการเข้าขัดจังหวะ (interrupt), และส่วนควบคุมอื่นๆเพื่อให้เหมาะกับการใช้งาน โครงงานนี้เป็นโครงงานที่สามารถนำไปใช้ได้ในงานทั่วๆไป ไม่จำกัดว่าจะต้องใช้เฉพาะภายในห้องเรียนเท่านั้น เช่นสามารถนำไปติดภายในตัวอาคารหรือภายในร้านเพื่อใช้ติดต่อส่งข้อมูลภายใน ผู้ส่งข่าวสารไม่จำเป็นต้องนำสารไปส่งถึงตัวผู้รับด้วยตัวเองทำให้สามารถนำเวลาไปทำอย่างอื่นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MCS-NETWORK

NAPORN PROMPANIT

BUNDIT KIERDSINGNAKORN

ASSOCIATE PROFESSOR,

RUTTIKORN VARAKULSIRIPUN

1989

ABSTRACT

This is thesis is about connecting many computers to form a network which will be used in classes which cover large area. The intercommunication will be presented via display monitors so the users can see the messeges obviously.

The notion of this transmission is using the RS-232 port which had already attached in the most of the general computers. The performer had invented a card which includes some special functions such as protocols,interrupt controller by priority order and any others which suit the controllers.

The application of this project is not limited only in classes but specially also in other purposes such as for the broker and the stockholder, the manger and the secretaries or the warehouse authorities etc. For the sake of facilities of transmission data from one place to another the transmitter needn't to bring the messages to the receivers by him or herself.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
บทที่ 3 การคำนวณและการสร้าง	32
บทที่ 4 การทดลองและผล	38
บทที่ 5 บทวิจารณ์และสรุป	51
ภาคผนวก	52
กิตติกรรมประกาศ	75
หนังสืออ้างอิง	76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. บทนำ

ในปัจจุบันนับว่าคอมพิวเตอร์ได้เข้ามามีบทบาทมากในทุกสาขาวิชา เมื่อการใช้คอมพิวเตอร์มีมากขึ้น แน่นอนความต้องการในการสื่อสารข้อมูลด้วยคอมพิวเตอร์ย่อมมากขึ้นเป็นเงาตามตัว สำหรับในโครงงานนี้เป็นการทดลองและพัฒนาการนำเอาคอมพิวเตอร์มาต่อกันเป็นโครงข่ายเพื่อใช้ประโยชน์ในการเรียนการสอน โดยทั้งผู้เรียนและผู้สอนต่างก็สามารถติดต่อสื่อสารกันได้โดยผ่านคอมพิวเตอร์ ซึ่งถือเป็นการพัฒนาอีกรูปแบบหนึ่งที่นำมาใช้ได้ในปัจจุบันกับการเรียนการสอนในสมัยใหม่ ซึ่งจะสามารถอำนวยความสะดวกสบายให้ทั้งแก่ทั้งผู้เรียนและผู้ทำการสอน ทำให้เพิ่มความเพลิดเพลินและยังทำให้การเรียนเป็นไปอย่างสะดวกและรวดเร็วมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ทฤษฎีและหลักการ

2.1 การส่งผ่านข้อมูล

2.1.1 การอินเตอร์เฟสไมโครโปรเซสเซอร์

ดังที่ทราบมาแล้วว่าขบวนการทางคณิตศาสตร์ และลอจิกของระบบคอมพิวเตอร์เกิดขึ้นจากการทำงานของหน่วยประมวลผลกลาง (Central Processing Unit:CPU) นั่นเอง โดย CPU นี้จะติดตั้งอยู่บนแผ่นวงจรร่วมกับอุปกรณ์อื่นๆ เช่น หน่วยความจำ ROM (Read Only Memory) , RAM (Random Acess Memory) และอุปกรณ์อื่นๆ ดังนั้นการอินเตอร์เฟสไมโครโปรเซสเซอร์ก็คือ การทำงานร่วมกันระหว่าง CPU กับอุปกรณ์อื่นๆในการโอนย้ายข้อมูลระหว่างอุปกรณ์ต่างๆบนแผ่นวงจรนั่นเอง

นอกจาก CPU จะต้องทำงานสอดคล้องกับ ROM , RAM แล้ว ยังต้องอินเตอร์เฟสเข้ากับอุปกรณ์อินพุท , เอาท์พุทต่างๆอีกด้วย เพื่อเพิ่มประสิทธิภาพการทำงานให้สมบูรณ์ยิ่งขึ้น ในขบวนการต่างๆของการทำงานร่วมกันของอุปกรณ์อิเล็คทรอนิคส์ จะต่อเนื่องกันเป็นลูกโซ่ เช่นในการส่งข้อมูลจาก CPU ไปยังอุปกรณ์อื่นเป็นต้น

-ไมโครคอมพิวเตอร์เนตเวอร์ค

การติดต่อสื่อสารข้อมูลสมัยใหม่นี้ มีรากฐานมาจากความพยายามเชื่อมต่อระหว่างคอมพิวเตอร์กับคอมพิวเตอร์ เพื่อให้สามารถใช้คอมพิวเตอร์ได้อย่างมีประสิทธิภาพมากที่สุด จึงได้มีการเซตระบบเชื่อมต่อ เช่น กำหนดเป็นมาตรฐานพอร์ต มาตรฐาน RS232 , RS422 การเชื่อมต่อด้วยวิธีดังกล่าวนี้อาศัยระบบสื่อสารที่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อยู่แล้ว เช่น โทรศัพท์ ดังนั้นขอบเขตของการสื่อสารข้อมูลจึงอยู่ในรูปแบบที่จำกัด ต่อมาเมื่อมีการใช้คอมพิวเตอร์มากขึ้น ความต้องการติดต่อสื่อสาร จึงต้องการให้ติดต่อกันได้ในลักษณะเป็นการเชื่อมต่อหลายเครื่องพร้อมกันได้ในเวลาเดียวกัน ระบบเนตเวอร์คจึงได้รับการพัฒนาให้ดีขึ้นเป็นลำดับ

ลักษณะของเนตเวอร์คอาจจะเริ่มจากจุดเล็กๆ ซึ่งอยู่บนบอร์ดอิเล็คทรอนิคส์เดียวกัน ขยายตัวใหญ่ขึ้นเป็นระบบที่ทำงานร่วมกันในห้องทำงาน ในตึก ระหว่างตึก ระหว่างสถาบัน ระหว่างประเทศ การจัดแบ่งรูปแบบของเนตเวอร์คของคอมพิวเตอร์จึงแยกตามขนาดของเนตเวอร์ค ดังตารางที่ 1.1

ระยะทางระหว่างโปรเซสเซอร์		ลักษณะที่ตั้งของโปรเซสเซอร์	ชื่อเรียกเนตเวอร์ค
0.1	เมตร	บอร์ดอิเล็กทรอนิกส์	เครื่องจักรนิดดาต้าโฟลว์
1	เมตร	ระบบเดียวกัน	มัลติโปรเซสเซอร์
10	เมตร	ห้อง	โลคัลเนตเวอร์ค
100	เมตร	ตัวอาคาร	
1	กิโลเมตร	หน่วยงานเดียวกัน	
10	กิโลเมตร	เมือง	เนตเวอร์คระยะไกล
100	กิโลเมตร	ประเทศ	
1000	กิโลเมตร	ระหว่างประเทศ	เนตเวอร์คระยะไกลมาก
10000	กิโลเมตร	ระหว่างดวงดาว	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1.1 การแบ่งแยกลักษณะของคอมพิวเตอร์เนตเวอร์คตามระยะทางโปรเซสเซอร์ที่เชื่อมต่อกัน

2.1.2 วิธีของการถ่ายโอนข้อมูล

การถ่ายโอนข้อมูลสามารถแบ่งได้เป็น 2 แบบคือ

1. การถ่ายโอนข้อมูลแบบขนาน
2. การถ่ายโอนข้อมูลแบบอนุกรม

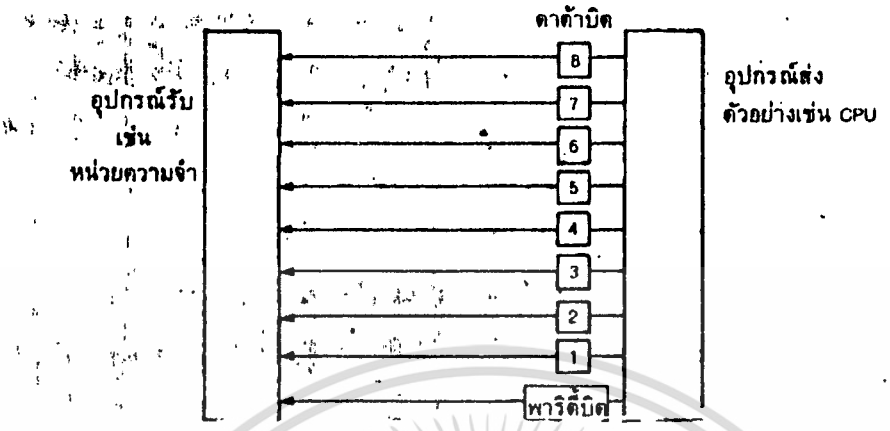
2.1.2.1. การถ่ายโอนข้อมูลแบบขนาน

ลักษณะของการส่งข้อมูลแบบขนาน ทำได้โดยการส่งข้อมูลออกมาทีละ 1 ไบต์ จากอุปกรณ์ส่งไปยังอุปกรณ์รับ ตัวกลางระหว่าง 2 เครื่องจะต้องมีช่องทางให้ข้อมูลเดินทางอย่างน้อย 8 ช่องทาง โดยมากจะเป็นสายขนานให้กระแสไฟฟ้าวิ่งมากกว่าจะเป็นตัวกลางชนิดอื่น เนื่องจากมีสัญญาณสูญหายไปกับความต้านทานของสาย ระยะทางระหว่าง 2 เครื่องจึงไม่ควรจะเกิน 100 ฟุต ปัญหาที่เกิดขึ้นหากระยะทางของสายมากกว่านี้ก็คือ ระดับของกราวนด์ในทางไฟฟ้าที่จุดรับผิดไปจากจุดส่ง ทำให้เกิดการผิดพลาดในการรับสัญญาณลอจิกทางฝ่ายรับ

นอกจากสายที่เป็นทางเดินของข้อมูลแล้วอาจจะมีทางเดินของสัญญาณควบคุมอื่นๆอีก เป็นต้นว่า บิตที่บอกพาริตี้ของสัญญาณ เพื่อเป็นการตรวจสอบความผิดพลาดของการรับสัญญาณที่ปลายทาง หรือสายที่ควบคุมการโต้ตอบ (Hand-shake)

จะเห็นว่าการส่งแบบขนานส่วนมากจะทำในระยะทางใกล้ๆ เนื่องจากจะต้องมีช่องทางเดินของสัญญาณมากกว่า 8 สาย และอุปกรณ์ที่ติดต่อแบบขนานกับคอมพิวเตอร์จะได้แก่ เครื่องพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

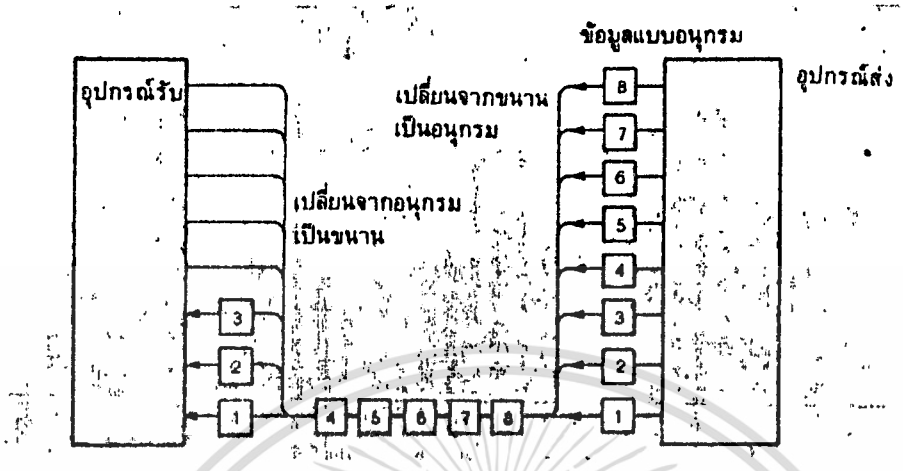


รูปที่ 1.1 การส่งข้อมูลแบบขนาน

2.1.2.2. <u>การถ่ายโอนข้อมูลแบบอนุกรม</u>

ในการถ่ายโอนข้อมูลแบบอนุกรม ข้อมูลจะถูกส่งออกมาทีละบิตระหว่างจุดส่งและจุดรับ จะเห็นว่าการส่งข้อมูลแบบนี้จะช้ากว่าแบบขนานที่กล่าวมาแล้ว ข้อดีของการถ่ายโอนข้อมูลแบบนี้คือ ตัวกลางการสื่อสารต้องการเพียงช่องเดียวหรือสายเพียงคู่เดียว ค่าใช้จ่ายในสื่อกลางจึงถูกกว่าแบบขนานสำหรับการส่งในระยะทางไกลๆ โดยเฉพาะเมื่อเรามีระบบการสื่อสารทางโทรศัพท์ไว้ใช้งานอยู่แล้ว ย่อมจะเป็นการประหยัดกว่าที่จะทำการติดต่อสื่อสารทีละ 8 ช่อง เพื่อการถ่ายโอนข้อมูลแบบขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.2 การส่งข้อมูลแบบอนุกรม

รูปที่ 1.2 แสดงให้เห็นการส่งข้อมูลแบบอนุกรม ข้อมูลจากจุดส่งจะถูกเปลี่ยนให้เป็นแบบอนุกรมเสียก่อนแล้วค่อยทยอยส่งออกทีละบิตไปยังจุดรับ ที่จุดรับจะต้องมีกลไกในการเปลี่ยนข้อมูลที่ส่งมาทีละบิต ให้เป็นสัญญาณแบบขนานซึ่งลงตัวพอดี นั่นคือ บิต 1 ลงที่บัสข้อมูลเส้นที่ 1 พอดี การที่จะทำให้การแปลงสัญญาณจากอนุกรมทีละบิตให้ลงพอดีนั้นจำเป็นต้องมีกลไกที่เหมาะสม เพื่อป้องกันการผิดพลาดในการรับ กลไกที่ว่านี้แบ่งเป็น 2 แบบ คือ

1) การสื่อสารแบบ Synchronous

2) การสื่อสารแบบ Asynchronous

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 **รูปแบบของการติดต่อสื่อสารแบบอนุกรม**

การติดต่อแบบอนุกรมอาจจะแบ่งตามรูปลักษณะได้ 3 แบบ ตามรูปที่ 1.3

1. แบบซิมเพล็กซ์ (Simplex) ข้อมูลส่งได้ในทางเดียวเท่านั้น บางครั้งที่เรียกว่าการส่งทิศทางเดียว (Unidirectional data bus)

2. แบบฮาล์ฟดูเพล็กซ์ (Half duplex) ข้อมูลสามารถส่งได้ทั้งสองสถานี แต่จะต้องผลัดกันส่งและผลัดกันรับ จะส่งและรับพร้อมกันไม่ได้

3. แบบฟูลดูเพล็กซ์ (Full duplex) ทั้งสองสถานีสามารถรับและส่งได้ในเวลาเดียวกัน

รูปที่ 1.3 รูปแบบของการติดต่อสื่อสารข้อมูลแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

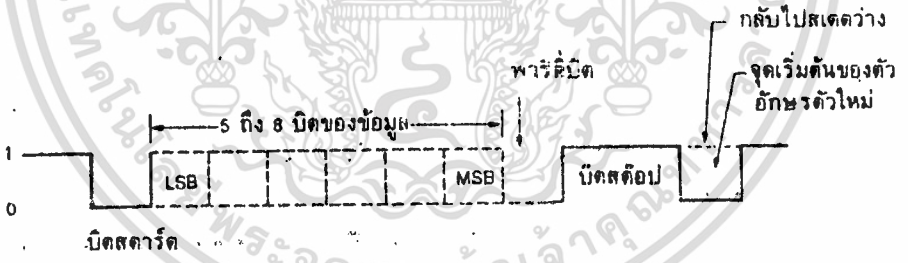
2.1.4 **ความเร็วในการถ่ายโอนข้อมูลแบบอนุกรม**

ความเร็วของการถ่ายโอนข้อมูลแบบอนุกรม หน่วยวัดเป็นบิตต่อวินาที (bps) หน่วยที่บรรยายถึงการเปลี่ยนแปลงของสัญญาณใน 1 วินาที เรียกว่าบอดเรต (baud rate) หรืออัตราบอด ซึ่งความสัมพันธ์ของอัตราบิตและอัตราบอดสามารถเขียนในรูปของสมการทางคณิตศาสตร์ได้ดังนี้

bit rate = baud rate * บิตใน 1 บอด

2.1.5 **การสื่อสารแบบ Asynchronous**

การส่งแบบอะซิงโครนัสนี้ จะมีลักษณะของสัญญาณแสดงได้ดังรูปที่ 1.4 เพื่อเพิ่มกลไกในการรับส่งอย่างถูกต้อง สัญญาณอะซิงโครนัสจะประกอบด้วยบิตเริ่มต้นหรือสตาร์ตบิต (start bit) และบิตสิ้นสุดหรือสต็อปบิต (stop bit)



รูปที่ 1.4 รูปแบบการสื่อสารแบบอะซิงโครนัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขณะที่สถานะของการส่งเป็นแบบว่าง (Idle) คือ ยังไม่มีสัญญาณส่งออกมา จะมีสัญญาณหรือมีแรงดัน (หรือกระแส) ตลอดเวลา เพื่อความแน่ใจว่าฝ่ายรับยังติดต่ออยู่กับฝ่ายส่ง เมื่อเริ่มจะส่งข้อมูล สัญญาณของอะซิงโครนัสจะเป็น 0 หนึ่งช่วงสัญญาณนาฬิกา บิตนี้เรียกว่า start bit ตามหลังของ start bit ก็จะเป็นข้อมูลสำหรับ 1 ตัวอักษร ซึ่งอาจจะมีขนาดตั้งแต่ 5 บิตจนถึง 8 บิต โดยบิตที่มีค่าน้อยที่สุด (LSB) จะถูกส่งออกมาก่อนไล่ไปจนถึงบิตที่มีค่ามากที่สุด (MSB) การเข้ารหัสอักขระนี้ส่วนมากจะนิยมใช้รหัส ASCII ตามหลังข้อมูลก็จะเป็นพาริตี้บิต ซึ่งอาจจะใช้หรือไม่ใช้ก็ได้ พาริตี้บิตทำหน้าที่เป็นตัวตรวจสอบความถูกต้องของสัญญาณที่ได้รับ พาริตี้บิตอาจจะเป็นแบบคู่ (Even) หรือแบบคี่ (Odd) หมายความว่าถ้าหากเป็นพาริตี้คู่ จำนวนบิตที่เป็น 1 ในช่วงบิตข้อมูลกับบิตพาริตี้รวมแล้วจะต้องเป็นจำนวนคู่ ผู้ส่งจะต้องทำการตรวจสอบข้อมูลแล้วใส่พาริตี้บิตเอง ฝ่ายรับเมื่อรับแล้วก็จะต้องตรวจสอบดูว่าเป็นจริงดังสถานการณ์ที่ตั้งเอาไว้หรือไม่ หากผิดพลาดก็หมายความว่าสัญญาณที่รับนั้นผิดไปจากสถานีส่งออกมา ทั้งนี้จะต้องผิดพลาดเป็นจำนวนคี่เท่านั้น คือผิดไป 1 บิต 3 บิต หรือ 5 บิต พร้อมกันจึงจะตรวจสอบได้ว่าผิด แต่ถ้าผิดเป็นจำนวนคู่ ผลรวมของจำนวนหนึ่งก็ยังเป็นคู่อยู่ดี สำหรับ Odd parity ก็จะมีลักษณะที่คล้ายกัน แต่จะตรวจสอบบิตที่เป็น 1 เป็นจำนวนคี่

หลังจากบิตพาริตี้แล้ว ก็จะเป็น stop bit ซึ่งมีสถานะเป็น 1 ความกว้างของ stop bit อาจจะเป็น 1 , 1.5 หรือ 2 พัลส์ของสัญญาณนาฬิกา แล้วแต่ผู้รับและผู้ส่งจะตกลงกันเอง

การเริ่มใช้พอร์ตอนุกรม จึงจำเป็นจะต้องตั้งค่าต่างๆสำหรับเป็นการส่งแบบอนุกรม ซึ่งได้แก่

1. ความเร็วในการส่ง
2. ความยาวรหัส 1 อักขระ
3. บิตตรวจสอบ
4. จำนวน stop bit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

026946

จะเห็นว่ากลไกในการซิงโครนัสของการสื่อสารอะซิงโครนัส มีลักษณะเป็นไปทีละตัวอักขระ จำนวนพัลส์ของสัญญาณที่ส่งออกยังมีบางส่วนที่ทำใช้ในการควบคุมการส่งอีก อันได้แก่ start bit, stop bit และ parity bit ทำให้ความเร็วการส่งอักขระต่อวินาทีน้อยลงไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ฮาร์ดแวร์ที่เกี่ยวข้องกับการสื่อสารข้อมูล

2.2.1 พอร์ต RS 232C

โดยปกติไมโครคอมพิวเตอร์จะมีพอร์ตที่เป็นแบบอนุกรม เรียกชื่อกันว่า RS 232C อยู่ในตัวเองอยู่แล้ว หลายเครื่องไม่มีมากับเครื่อง อย่างเช่น IBM PC จำเป็นจะต้องมีการ์ดที่เรียกว่าอะซิงโครนัสอะแดปเตอร์ (Asynchronous Communication Adapter) มาเสียบใส่

พอร์ต RS 232C นี้ทำหน้าที่รับและส่งข้อมูลในแบบอนุกรม เรียกว่า Universal Asynchronous Adapter เหตุที่มีชื่อเรียกว่า RS 232C ก็เนื่องจากสมาคมผู้ผลิตอุปกรณ์อิเล็คทรอนิคส์ของอเมริกา หรือ EIA ได้กำหนดมาตรฐานของอุปกรณ์การสื่อสารแบบอนุกรมเอาไว้ภายใต้ชื่อว่า RS 232C

หน้าที่สำคัญของการสื่อสารแบบอะซิงโครนัสก็คือ

-รับสัญญาณ

1. เปลี่ยนสัญญาณเข้ามาแบบอนุกรมให้เป็นแบบขนาน
2. ตรวจสอบความผิดพลาดของสัญญาณที่รับ
3. ตัด stop bit และ parity bit ออก
4. ส่งสัญญาณให้ CPU รู้ว่ารับสัญญาณไว้แล้ว

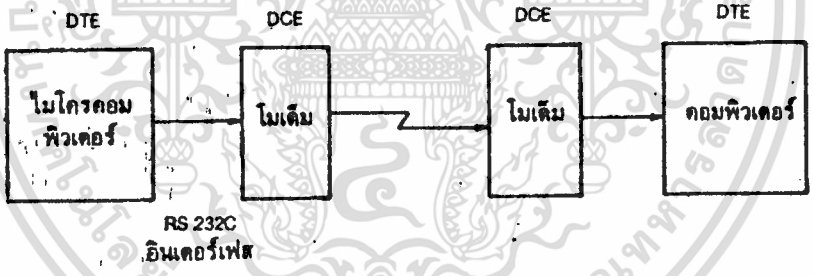
-ส่งสัญญาณ

1. เปลี่ยนสัญญาณแบบขนานจาก CPU แล้วส่งออกเป็นแบบอนุกรม
2. เพิ่ม stop bit และ parity bit
3. เพิ่มสัญญาณควบคุมโมเด็มที่เชื่อมต่อ (ถ้ามี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 มาตรฐาน RS 232C

มาตรฐาน RS 232C ซึ่ง RS ย่อมาจาก Recommend Standard ส่วน 232 เป็นหมายเลขบ่งบอกของมาตรฐานตัวนี้ C เป็นหมายเลขของฉบับท้ายสุดของมาตรฐานตัวนี้ จุดประสงค์ของมาตรฐานตัวนี้คือเพื่อบรรยายคุณลักษณะของการเชื่อมต่ออุปกรณ์รับส่งข้อมูลปลายทาง (Data Terminal Equipment : DTE) กับอุปกรณ์สื่อสารข้อมูล (Data Communication Equipment : DCE) สำหรับผู้ใช้ไมโครคอมพิวเตอร์ DTE จะหมายถึงตัวไมโครคอมพิวเตอร์ และ DCE หมายถึงโมเด็ม อุปกรณ์อื่นๆ เช่น เครื่องพิมพ์ที่รับสัญญาณแบบอนุกรมอาจจะเป็นได้ทั้ง DTE และ DCE ขึ้นอยู่กับผู้ผลิต ข้อแตกต่างของ DTE และ DCE จะเห็นได้จากรูปที่ 2.1 จากรูปนี้จะเห็นได้ว่า RS 232C มีส่วนสำคัญอย่างมากสำหรับการสื่อสารข้อมูลระหว่างไมโครคอมพิวเตอร์



รูปที่ 2.1 การใช้ RS 232C เชื่อมต่ออุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อสำคัญอีกประการหนึ่งของ RS 232C ก็คือ ความเร็วและระยะทางของการเชื่อมต่อ RS 232C สามารถเชื่อมต่อการถ่ายโอนข้อมูลได้จาก 0 - 20,000 บิตต่อวินาที ซึ่งเพียงพอสำหรับไมโครคอมพิวเตอร์ที่มีขนาดอัตราบอด 110 ถึง 9600 บอด ความยาวของสายเชื่อมต่อโดยสัญญาณตามมาตรฐานของ RS 232C จำกัดอยู่แค่ 50 ฟุต ซึ่งเพียงพอสำหรับการสื่อสารไมโครคอมพิวเตอร์กับอุปกรณ์รอบนอก

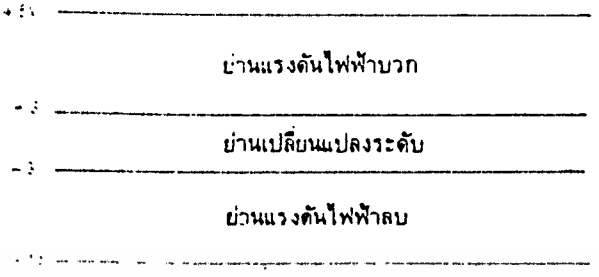
2.2.3 <u>ลักษณะของสัญญาณ RS 232C</u>

เพื่อเป็นหลักประกันว่าข้อมูลถูกส่งออกไปอย่างถูกต้อง และอุปกรณ์ถูกควบคุมอย่างถูกต้อง จำเป็นจะต้องมีข้อตกลงกันในเรื่องของสัญญาณที่ใช้ มาตรฐาน RS 232C กำหนดย่านของแรงดันไฟฟ้าในสัญญาณดังแสดงในตารางที่ 2.1 และรูปที่ 2.2

ตารางที่ 2.1

มาตรฐานของการใช้แรงดันไฟฟ้า			
แรงดันไฟฟ้า	สถานภาพลอจิก	สถานภาพของสัญญาณ	ฟังก์ชันในการควบคุม
บวก	0	สเปซ	ออน
ลบ	1	มาร์ค	ออฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



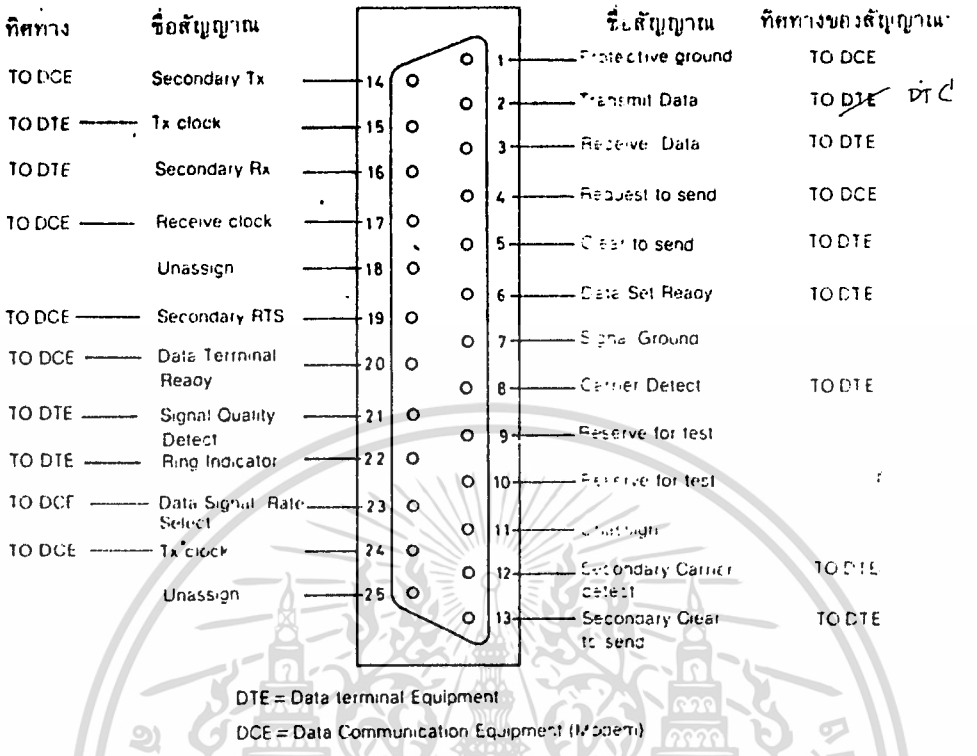
รูปที่ 2.2 ย่านของแรงดันไฟฟ้าที่ใช้ในสัญญาณ RS 232C

สำหรับไมโครคอมพิวเตอร์บางเครื่อง ให้แต่สัญญาณลอจิกออกมาเป็นสัญญาณของ RS 232C เลย เช่น อะซิงโครนัสอะแดปเตอร์ของ IBM PC ในกรณีเช่นนี้ ระยะทางของสายที่เชื่อมต่ออาจจะสั้นกว่า 50 ฟุต เนื่องจากระดับกำลังวนต์เปลี่ยนแปลงไป อันเนื่องมาจากการสูญเสียไปในความต้านทานของสาย

2.2.4 <u>การกำหนดจุดข้อต่อของ RS 232C</u>

มาตรฐานของ RS 232C กำหนดข้อต่อแบบ DB-25 แต่ละขาของข้อต่อกำหนดไว้ดังในรูปที่ 2.3 อย่างไรก็ตามผู้ผลิตไมโครคอมพิวเตอร์อาจจะใช้ข้อต่อชนิดอื่นที่นอกเหนือไปจาก DB-25 ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 การกำหนดของขั้วต่อ RS 232C

สัญญาณต่างๆมีหน้าที่ดังต่อไปนี้

-Transmit Data (TD ขาที่ 2)

เป็นสัญญาณที่ส่งออกจาก DTE (หรือตัวไมโครคอมพิวเตอร์) ไปยังโมเด็มหรือต่อเข้าโดยตรงกับไมโครคอมพิวเตอร์ตัวอื่น หรือเครื่องพิมพ์ เมื่อไม่มีสัญญาณส่งออก สถานภาพของลอจิกที่ขานี้จะมีค่าเท่ากับ "1" หรือเทียบเท่ากับ stop bit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-Recieve Data (RD ขาที่ 3)

เป็นทางของสัญญาณเข้าไปยัง DTE หรือไมโครคอมพิวเตอร์ เมื่อไม่มีสัญญาณรับเข้ามา ขานี้จะมีสถานภาพทางลอจิกเป็น "1"

-Request To Send (RTS ขาที่ 4)

ใช้สำหรับส่งสัญญาณไปยังโมเด็มหรือเครื่องพิมพ์ เป็นการเรียกร้องที่จะส่งสัญญาณมาทางขา 2 สัญญาณนี้ใช้คู่กับ CTS หรือ Clear to send อุปกรณ์รับหากได้รับสัญญาณ RTS จะตรวจสอบตัวเองว่าพร้อมจะรับสัญญาณได้หรือยัง หากพร้อมที่จะรับก็ส่งสัญญาณออกไปที่สาย CTS

-Clear To Send (CTS ขาที่ 5)

ดังอธิบายไว้ใน RTS เมื่อสัญญาณนี้อยู่ในสถานะ OFF (negative voltage หรือลอจิก "1") หมายความว่า อุปกรณ์รับกำลังบอกว่าพร้อมที่จะรับข้อมูลแล้ว

-Data Set Ready (DSR ขาที่ 6)

เมื่อสัญญาณสายนี้อยู่ในสถานะ ON (หรือลอจิก 0) เป็นการบอกไมโครคอมพิวเตอร์หรือฝ่ายส่งว่า โมเด็มต่อเข้ากับสายโทรศัพท์เรียบร้อยแล้วและพร้อมที่จะส่งได้แล้ว โมเด็มที่มีการหมุนหมายเลขอัตโนมัติจะส่งสัญญาณสายนี้ ไปบอกให้คอมพิวเตอร์รู้ว่าต่อโทรศัพท์ได้สำเร็จแล้ว

-Signal Ground (SG ขาที่ 7)

SG ทำหน้าที่เป็นระดับแรงดันอ้างอิงสำหรับทุกๆสายของสัญญาณ จะมีแรงดันเป็น "0" เมื่อเทียบกับสัญญาณตัวอื่น

-Carrierr Detect (CD ขาที่ 8)

โมเด็มจะส่งสัญญาณที่อยู่ในสถานะ ON (ลอจิก "0") ไปบอกไมโครคอมพิวเตอร์ เมื่อได้รับสัญญาณจากโมเด็มของอีกฝ่ายหนึ่ง สัญญาณนี้จะนำไปจุด LED บอกว่าได้รับสัญญาณจากโมเด็มอีกฝ่ายหนึ่งแล้ว ไฟ LED จะอยู่บนหน้าบิทม์ของโมเด็มเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-Data Terminal Ready (DTR ขาที่ 20)

คอมพิวเตอร์เปิดสัญญาณสายนี้ให้ ON (ลอจิก "0") เมื่อพร้อมที่จะติดต่อกับโมเด็ม โมเด็มส่วนมากจะไม่รายงานสถานภาพของตัวเอง (CD, DSR และ CTS) ให้คอมพิวเตอร์รู้ หากคอมพิวเตอร์ไม่เปิดสัญญาณ DTR

-Ring Indicator (RI ขาที่ 22)

สัญญาณนี้ใช้ในโมเด็มที่เป็นระบบตอบได้อัตโนมัติ (Auto-answer) สัญญาณนี้จะON เมื่อมีสัญญาณกระดิ่งมา และ OFF ระหว่างเสียงดังของกระดิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 แสดงคุณลักษณะโดยย่อของสัญญาณ RS 232C

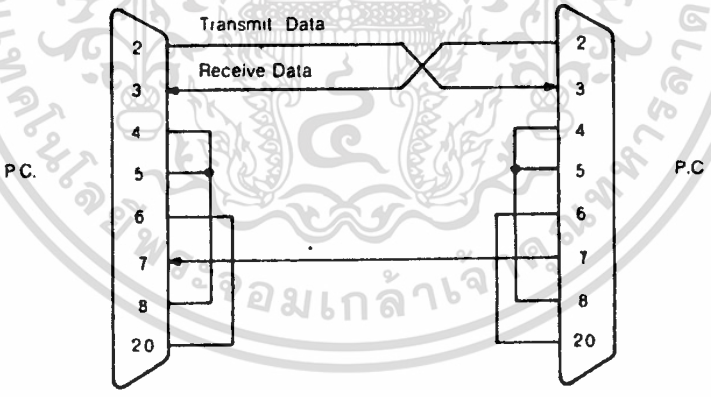
Driver output logic levels with 3k to 7k load	15V>O_h>5V -5>O_l>-15 โวลต์
Driver output voltage when open circuit	Vo <25 โวลต์
Driver output impedance with Power off	R_o >300 Ohms
Output short circuit current	I_o < 0.5 A
Driver slew rate	dv/dt < 30 V/s
Receiver input impedance	7k >R_{in}>3k
Receiver input voltage	+15 compatible with driver
Receiver output with open circuit input	MARK
Receiver output with +3V input	SPACE
Receiver output with -3V input	MARK
+15	LOGIC 0 = SPACE =
+5	CONTROL ON
+5	Noise Margin
+3	
+3	Transition Region
-3	
-3	Noise Margin
-5	
-5	LOGIC 1 = MARK =
-15	CONTROL OFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 <u>การเชื่อมต่อคอมพิวเตอร์กับคอมพิวเตอร์โดยตรง</u>

ในบางครั้งเราอาจจะต้องการย้ายข้อมูล จากคอมพิวเตอร์เครื่องหนึ่งไปยังอีกเครื่องหนึ่ง ถ้าหากว่าไมโครคอมพิวเตอร์ที่ใช้มีดิสเกตต์และเป็นชนิดเดียวกัน ก็เพียงแต่ก็อปปี้ลงบนดิสเกตต์แล้วก็นำไปใช้อีกเครื่องหนึ่ง แต่ถ้าเป็นคนละระบบหรือคนละ OS กัน วิธีดังกล่าวจะใช้ไม่ได้ผล เพราะการบันทึกลงบนแผ่นดิสก์ของคอมพิวเตอร์แต่ละระบบ หรือ OS แต่ละชนิดส่วนมากจะไปด้วยกันไม่ได้ วิธีการแก้ปัญหาก็คือ การต่อเครื่องคอมพิวเตอร์เข้าด้วยกันโดยใช้ RS 232C แล้วส่งถ่ายโอนข้อมูลจากเครื่องหนึ่งไปอีกเครื่องหนึ่ง วิธีการต่อแบบนี้เรียกว่า Null Modem คือไม่ใช้โมเด็มนั่นเอง (ในกรณีที่ระยะทางไม่เกิน 50 ฟุต)

วิธีการต่อ RS 232C เข้าระหว่างเครื่องคอมพิวเตอร์โดยตรงมีอยู่หลายวิธีตามแต่ขบวนการที่จะใช้ ถ้าไม่ต้องการมีการตรวจสอบสัญญาณกันก็ต่อ RD เข้า TD ของอีกเครื่องหนึ่งสายกราวนด์ต่อถึงกันดังรูปที่ 2.4 ก็สามารถใช้งานส่งถ่ายโอนข้อมูลได้แล้ว



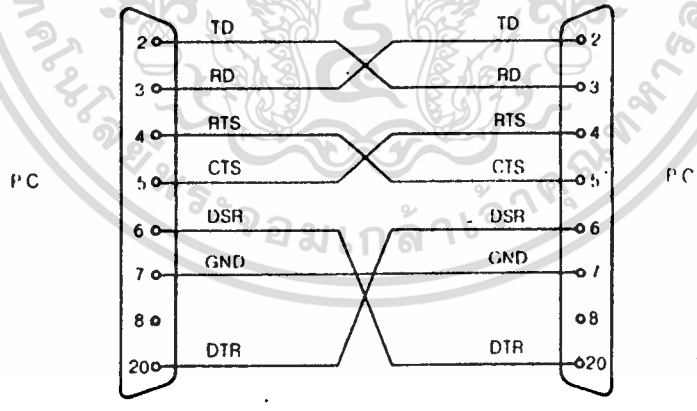
รูปที่ 2.4 การส่ง RS 232C ระหว่างไมโครคอมพิวเตอร์อย่างง่ายๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปกติ OS ที่ให้บริการเกี่ยวกับพอร์ต RS 232C จะส่งสัญญาณ RTS หรือ Request to sendออกมาที่ขา4ก่อน เมื่อ CS หรือ Clear to send ที่ขา 5 เป็นลอจิก"1" (หรือ ไฟลบ) จึงจะเริ่มทำการส่งข้อมูลที่โอเปอเรเตอร์บอกให้ส่ง ออกไปที่ขา 2 ในกรณีที่เป็นการต่อแบบง่ายๆ ดังแสดงในรูปที่ 2.4 จึงถือว่าเป็นการหลอกคอมพิวเตอร์ โดยเอาขา 4 RTS ต่อเข้ากับขา 5 หรือ CTS เพื่อให้คอมพิวเตอร์ส่งข้อมูลได้ทันที โดยไม่ต้องการความเรียบร้อยของฝ่ายรับ สำหรับขา 6 Data Set Ready ต่อเข้ากับขาที่ 20 Data Terminal Ready ก็ทำนองเดียวกัน โดยปกติคอมพิวเตอร์จะถามอุปกรณ์ที่มาต่อพ่วงกับ RS 232C ว่าพร้อมที่จะส่งแล้วหรือยัง โดยส่งสัญญาณถามที่ขา 20 และรอคำตอบที่ขา 6 ซึ่งก็มีลักษณะเหมือนกัน คือถามที่ขา 20 ก็ได้รับคำตอบกลับที่ขา 6 ทันที

ในการต่อแบบนี้ฝ่ายรับจะต้องรอรับอยู่ก่อนแล้ว ก่อนที่ฝ่ายส่งจะเป็นผู้ส่ง ไม่เช่นนั้นข้อมูลที่ส่งออกมาอาจหายได้ เพราะฝ่ายส่งไม่ได้ตรวจสอบความเรียบร้อยของฝ่ายรับก่อน

เราอาจจะต่อสายให้มีการตรวจสอบสัญญาณโต้ตอบ (Hand-shake) ที่ดีกว่านี้ได้ดังแสดงในรูปที่ 2.5



รูปที่ 2.5 การต่อไมโครคอมพิวเตอร์ผ่าน RS 232C แบบมี Hand shake

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีเช่นนี้จะมีการโต้ตอบที่ดีขึ้น เมื่อฝ่ายรับยังไม่พร้อมที่จะรับ ก็จะยังไม่มีสัญญาณ RTS ออกมา (พอร์ตอนุกรมยังไม่เปิด หรือ OPEN "COM1" ในภาษาเบสิคยังไม่ถูกเอ็กซีคิ้ว) ฝ่ายส่งซึ่งถือเอา RTS ของฝ่ายรับเป็น CTS ก็จะยังไม่ส่ง

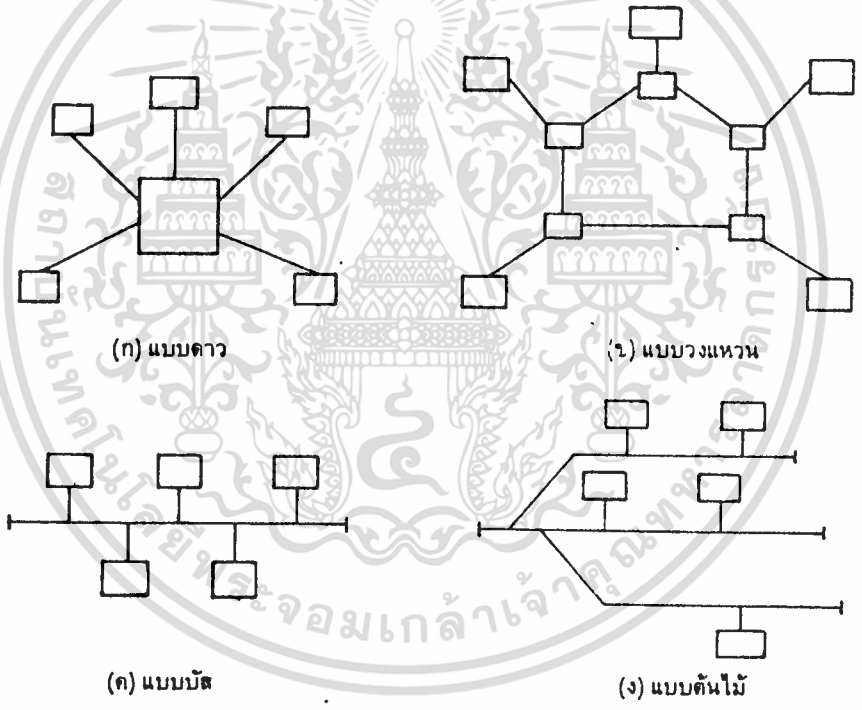
2.3 โครงข่ายของระบบเนตเวอร์ค

คำว่า โทโปโลยี (topology) หมายถึงรูปร่างของเนตเวอร์ค ที่พิจารณาจากการลากเส้นมาต่อรวมกันเป็นกิ่งก้านหรือรูปแบบของเนตเวอร์ค และหากเน้นด้วยคอมพิวเตอร์เนตเวอร์คก็จะหมายถึงรูปโครงเชื่อมต่อ โดยให้คอมพิวเตอร์เสมือนเป็นสถานีสายส่งสัญญาณ คือเส้นต่อของสถานี โดยในที่นี้จะกำหนดเป็นระบบข่ายแบบท้องถิ่นที่เรียกว่า Local Area Network หรือ LAN โครงข่ายของ LAN จึงเป็นรูปแบบที่นำเอาคอมพิวเตอร์หรืออุปกรณ์อื่นๆมาประกอบ โดยเชื่อมโยงกันและสายที่เชื่อมโยงนี้เราจะเรียกว่า ลิงค์ (link)

ปัญหาของการเชื่อมต่อระหว่างคอมพิวเตอร์หลายๆตัว จะได้แก่ ระบบเชื่อมโยงระหว่างจุดไปจุดอื่น ซึ่งถ้าหากใช้สถานีที่จะเชื่อมต่อกัน 2 สถานี ก็ต้องใช้ลิงค์ 1 ลิงค์ และถ้าเพิ่มเป็น 3 สถานีก็จะใช้ลิงค์ 3 ลิงค์ ถ้า 4 สถานี ก็จะต้องใช้จำนวน 6 ลิงค์ ซึ่งลิงค์ที่ใช้นี้ ถ้าหากเป็นลิงค์แบบเส้นทางเดียว คือรับเส้นหนึ่งและส่งอีกเส้นหนึ่ง ก็จะใช้สายเป็น 2 เท่า ในกรณี 2 สถานีจะต้องใช้สาย 2 เส้นนั่นเอง ดังนั้น ถ้าหากพิจารณาว่า มี N สถานี จะต้องใช้สายสัญญาณลิงค์ N(N-1) เส้น และหากลิงค์ที่พิจารณาเป็นแบบสองทิศทาง คือส่งไปและกลับได้แล้วละก็ จะทำให้เราใช้จำนวนลิงค์เหลือเพียง N(N-1)/2 และแต่ละสถานีจะมีจำนวน I-O port ถึง N-1 พอร์ต การต่อในลักษณะนี้เรียกว่าเป็น การต่อโดยตรง (direct connection) หรือ Mesh topology

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

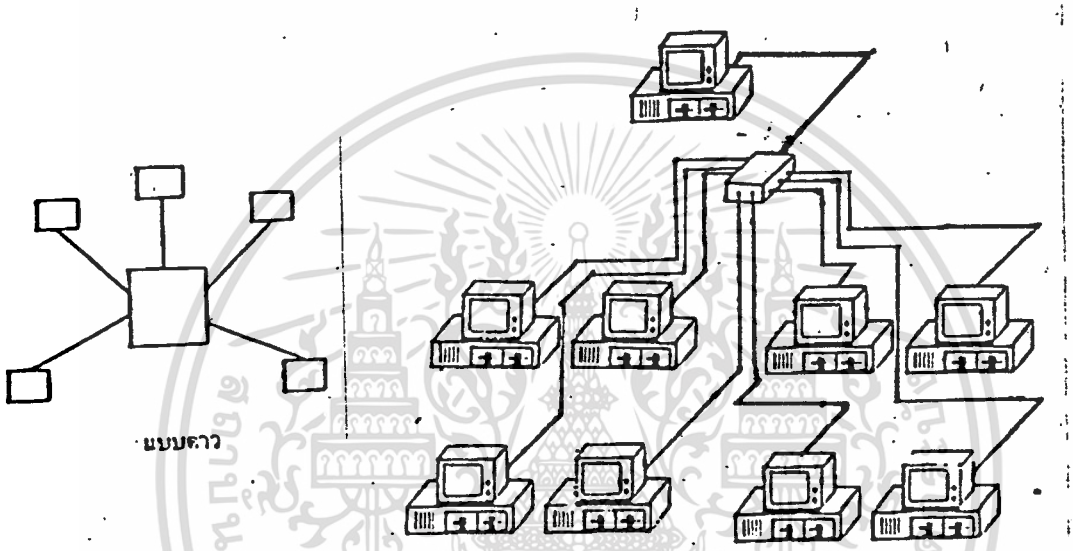
หากใช้จำนวนสถานีมากขึ้น การเชื่อมต่อจะต้องใช้ลิงค์มาก และจะเป็นการสิ้นเปลืองค่าใช้จ่ายมาก แต่ระบบการเชื่อมต่อถึงกันหมดนี้จะทำให้ระบบง่ายต่อการติดต่อสื่อสาร เพราะระบบการตรวจสอบทาง software จะง่ายขึ้น และจะไม่มีปัญหาเรื่องข้อมูลชนกันในสายตัวกลาง แต่อย่างไรก็ดี เราก็ถือว่าปัญหาทาง software เป็นปัญหาที่แก้ไขได้โดยง่าย จึงให้ความสำคัญในเรื่องของราคามากกว่า ดังนั้น topology ของเนตเวอร์คที่นิยมใช้กัน มีดังนี้ คือ โทโปโลยีในรูปแบบดาว (star) บัส (bus) ต้นไม้ (tree) และแบบริง (ring) ซึ่งลักษณะของโทโปโลยีต่างๆจะแสดงได้ดังรูปที่ 3.1



รูปที่ 3.1 โทโปโลยีของระบบเนตเวอร์ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งในโครงงานนี้จะใช้ โครงข่ายแบบดาว (Star Network) ซึ่งมีรูปแบบการต่อโดยนำสถานีต่างๆหลายสถานีมาต่อรวมกันกับคอมพิวเตอร์ศูนย์กลาง (Host Computer) หรือสวิทช์ชิ่งเซนเตอร์ (Switching Center) ดังแสดงในรูปที่ 3.2



รูปที่ 4.1 การต่อโครงข่ายแบบดาว

-ข้อดีของโครงข่ายรูปดาว

1) เหมาะสำหรับระบบที่มี การสื่อสารระหว่างหลายๆผู้ใช้เทอร์มินอลปลายทาง (End-user Terminal) กับคอมพิวเตอร์ศูนย์กลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) สามารถใช้ตัวกลางการสื่อสาร (Transmission Media) และความเร็วในการส่งข้อมูลที่แตกต่างกันได้

3) เหมาะสำหรับ terminal ที่มีความสามารถไม่มาก เช่น ใช้ในการเรียนการสอนระบบ Sound Lab

4) สามารถที่จะกำหนดการป้องกันการใช้โดยไม่ยินยอมได้ เช่น การกำหนดรหัสผ่าน (password) เป็นต้น

5) ง่ายต่อการควบคุมระบบการสื่อสารต่างๆ เช่น การตรวจสอบความผิดพลาดการส่งและรับข้อมูล

-ข้อเสียของโครงข่ายรูปดาว

1) ถ้าคอมพิวเตอร์ศูนย์กลางเสียไป ก็จะทำให้การติดต่อสื่อสารภายในระบบเสียไปด้วย

2) ต้องเพิ่มช่องต่ออุปกรณ์อินพุทและเอาท์พุท (I/O Port) ของคอมพิวเตอร์ศูนย์กลางให้เพียงพอกับจำนวนสายการสื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 สัญญาณต่างๆบนสล็อตของ IBM/PC

ภายใน IBM PC ได้มีการออกแบบให้สามารถเพิ่มเติมวงจรอินเตอร์เฟสเข้าไปในภายหลังได้ โดยผ่านทางสล็อตที่อยู่บนเมนบอร์ด (Main Board) สำหรับสล็อตบนเมนบอร์ดนี้จะมีจำนวน 5 สล็อต (สำหรับใน IBM PC/XT จะมี 8 สล็อต) ซึ่งแต่ละสล็อตจะมีจำนวนขาทั้งสิ้น 62 ขา แบ่งออกเป็น 2 ข้างๆละ 31 ขา ส่วนการเรียกตำแหน่งขาของสล็อตเหล่านี้จะขึ้นอยู่กับว่าขานั้นอยู่ข้างใด (ซ้ายหรือขวา) ของสล็อต โดยขาที่อยู่ทางด้านซ้ายของสล็อตจะเรียกโดยใช้อักษร "B" นำหน้าเลขตำแหน่งของขา ส่วนขาที่อยู่ทางด้านขวาของสล็อตจะเรียกโดยใช้อักษร "A" นำหน้าเลขตำแหน่งของขา

แต่ละขาของสล็อตเหล่านี้จะเชื่อมต่อกับเส้นสัญญาณต่างๆบนเมนบอร์ด ทำให้การสร้างวงจรอินเตอร์เฟสกับ IBM PC สามารถทำได้โดยสะดวก ซึ่งเส้นสัญญาณที่เชื่อมต่อกับขาของสล็อตเหล่านี้จะประกอบไปด้วย เส้นสัญญาณของบัสแอดเดรส (Address Bus) , บัสข้อมูล (Databus) , บัสควบคุมสำหรับการเขียน / อ่านข้อมูลจากหน่วยความจำหรืออุปกรณ์ I-O , เส้นสัญญาณสำหรับการของอินเตอร์รัพท์ของวงจรอินเตอร์เฟส , เส้นสัญญาณสำหรับการของ DMA , สัญญาณฐานเวลา (Timing Signal) ต่างๆที่ใช้ในระบบ , เส้นสัญญาณแสดงการรีเฟรชหน่วยความจำ และสัญญาณสำหรับการตรวจสอบความผิดพลาด (I-O Check)

นอกจากเส้นสัญญาณเหล่านี้แล้ว สล็อตบนเมนบอร์ดยังเชื่อมต่อกับแหล่งจ่ายไฟต่างๆที่ใช้ในระบบอีกด้วย คือ +5 Vdc, -5 Vdc, +12 Vdc และ -12 Vdc

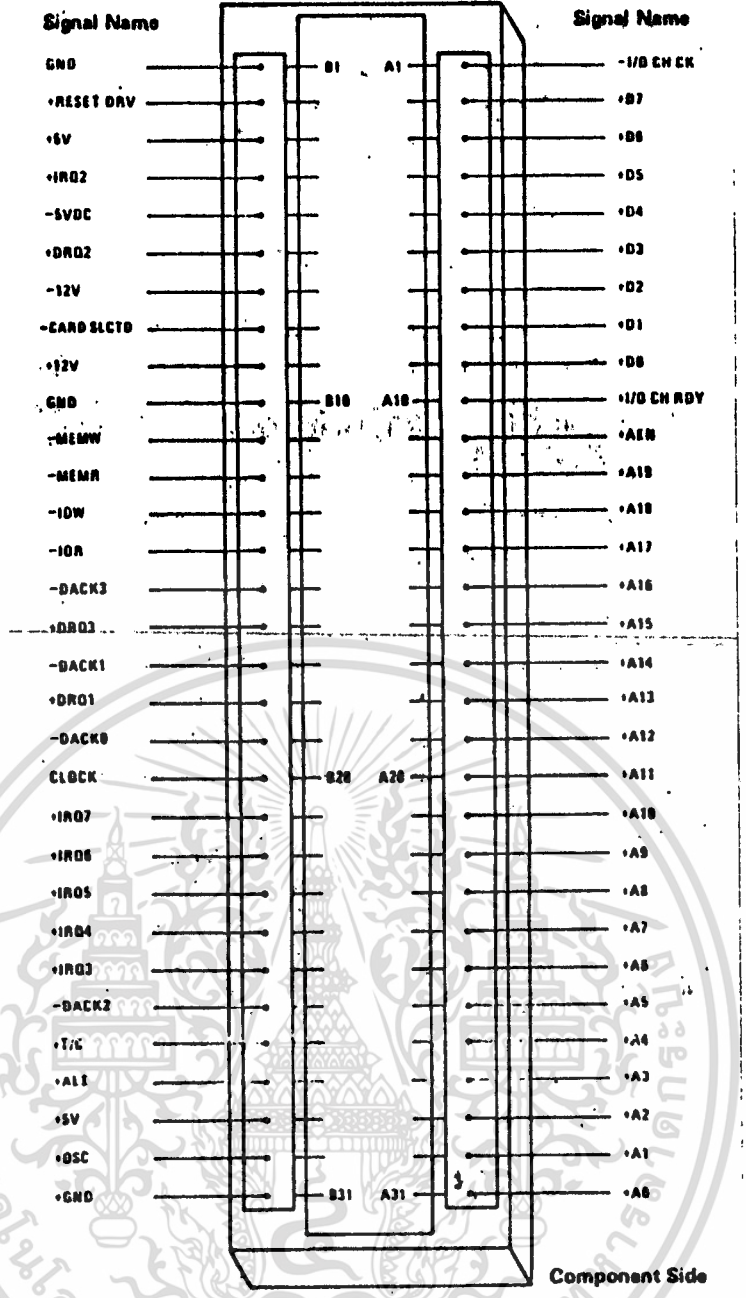
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.A.1 <u>รายละเอียดเกี่ยวกับสัญญาณต่างๆ</u>

-OSC (Oscillator : ขา B30)

ขานี้เป็นเอาท์พุทที่เชื่อมต่อกับสัญญาณ clock ที่มีความถี่สูงสุดบนเมนบอร์ด คือ 14.31818 MHz ซึ่งมีคาบเวลาประมาณ 70 nanosec. และมี duty cycle (ช่วงเวลาใน1 คาบที่สัญญาณ clock มีลอจิกเป็น "1" หารด้วยคาบเวลาทั้งหมด) ประมาณ 50% สัญญาณ clock อื่นๆของระบบ เช่น clock ที่ป้อนให้กับ 8088 หรือ ชิพซัพพอร์ทต่างๆนั้นจะถูกสร้างขึ้น โดยการหารสัญญาณ clock นี้ อย่างไรก็ตามสิ่งที่จะต้องคำนึงถึงในการใช้งานสัญญาณ OSC ก็คือ สัญญาณนี้จะไม่ synchronize กับสัญญาณอื่นๆบนบัสของระบบ ดังนั้นจึงไม่ควรที่จะนำสัญญาณจากขา OSC นี้ไปใช้เป็นสัญญาณ clock สำหรับวงจรภายนอกอื่นๆที่ทำงานร่วมกับระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 สัญญาณต่างๆบนสล็อตของ IBM PC

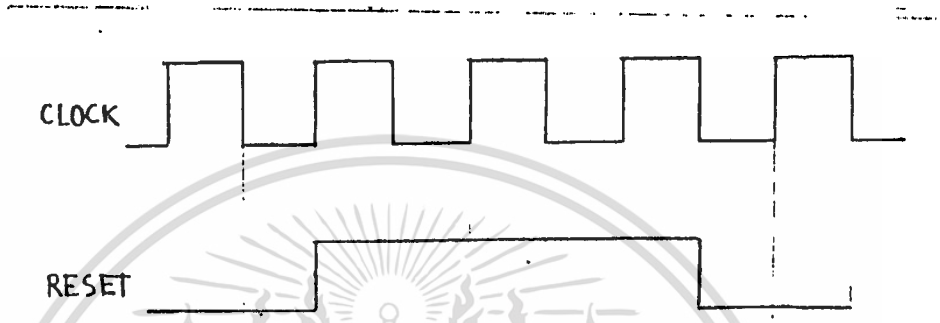
-CLK (Clock)

เป็นสัญญาณนาฬิกาที่อ้างอิงในระบบ โดยการนำสัญญาณ OSC มาหาร 3 จึงมีคาบเวลา 210 nanosec. หรือ 4.77 MHz และมี duty cycle 33% ทิศทางสัญญาณ output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-RESET DRV (Reset Drive)

เป็นสัญญาณที่ใช้ในการกำหนดค่าเริ่มต้นทางตรรกศาสตร์ของระบบ สัญญาณนี้จะ sync กับ CLK ช่วงเปลี่ยนสถานะ High to Low (falling edge of clock) สถานะการทำงาน Active High ดังแสดงในรูปที่ 4.2



รูปที่ 4.2 คาบเวลาของสัญญาณ RESET DRV

-A0 - A9 (Address bits 0 to 19)

สัญญาณนี้ใช้อ้างอิงตำแหน่งข้อมูลในหน่วยความจำ หรือ ตำแหน่งพอร์ตภายในระบบ ในจำนวน 20 บิตจะเข้าถึงข้อมูล 1 Megabyte โดยที่ A0 จะเป็นตำแหน่งต่ำสุด (LSB : Least Significant Bit) ในขณะที่ A19 จะเป็นตำแหน่งสูงสุด (MSB : Most Significant Bit) ซึ่งสัญญาณดังกล่าวได้มาจากไมโครโปรเซสเซอร์ หรือ DMA Controller สถานะการทำงาน Active High ทิศทางสัญญาณ output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-D0 - D7 (Data bit)

เป็นสายข้อมูลบิต 0 ถึง 7 สำหรับไมโครโปรเซสเซอร์ หน่วยความจำ และพอร์ตต่างๆ

โดย D0 เป็นตำแหน่งต่ำสุด (LSB) และ D7 เป็นตำแหน่งสูงสุด (MSB) ทิศทางการทำงานเป็น bidirectional data bus สถานะการทำงาน Active High

-ALE (Address Latch Enable)

เป็นสัญญาณที่ 8288 (Bus Controller) จับสัญญาณแอดเดรสที่ถูกต้องจากไมโครโปรเซสเซอร์ 8088 ทิศทางการทำงาน output สถานะการทำงาน Active High

-I/O Check (I/O Channel Check)

ใช้เป็นเครื่องมือในการตรวจจับความผิดพลาดพาริตี้ (parity error) ของหน่วยความจำหรือพอร์ตต่างๆ เมื่อสถานะเป็น LOW ไมโครโปรเซสเซอร์ก็จะรู้ว่าเกิดความผิดพลาดขึ้นในระบบ ทิศทางการทำงาน input สถานะการทำงาน Active Low

-I/O CHRDY (I/O Channel Ready)

โดยปกติสถานะการทำงานจะ High อยู่ แสดงว่าพร้อม หากหน่วยความจำหรือพอร์ทเกิดทำงานช้า อุปกรณ์เหล่านั้นก็จะทำให้สัญญาณนี้เป็น LOW เพื่อให้ไมโครโปรเซสเซอร์เพิ่มเวลาในส่วนการอ่านหรือเขียนข้อมูลกับอุปกรณ์เหล่านั้น ทิศทางการทำงาน input

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-IRQ2 - IRQ7 (Interrupt Request 2 to 7)

เป็นการจัดลำดับความสำคัญของสัญญาณอินเตอร์รัพท์ จากอุปกรณ์ต่างๆไปยังไมโครโปรเซสเซอร์ โดยเรียงลำดับความสำคัญต่ำสุดจาก IRQ7 ไปยังความสำคัญสูงสุด IRQ2 เมื่อเกิดการอินเตอร์รัพท์ขึ้น กล่าวคือสัญญาณเปลี่ยนสถานะ LOW เป็น HIGH ไมโครโปรเซสเซอร์จะตอบสนองโดยการกระโดดไปทำงานในส่วนที่เรียกว่า Interrupt Service Routine

ทิศทางการทำงาน input สถานะการทำงาน Active High

-IOR (I/O Read Command)

เป็นสัญญาณที่ใช้ในการติดต่อกับพอร์ตต่างๆ เพื่อให้ข้อมูลจากพอร์ตนั้นเข้าสู่ Data Bus สัญญาณนี้ถูกสร้างขึ้นจากไมโครโปรเซสเซอร์ หรือ DMA Controller ทิศทางการทำงาน output สถานะการทำงาน Active Low

-IOW (I/O Write Command)

เป็นสัญญาณที่ใช้ในการติดต่อกับพอร์ตต่างๆ เพื่อให้พอร์ทนั้นรับข้อมูลจาก Data Bus สัญญาณนี้ถูกสร้างขึ้นจากไมโครโปรเซสเซอร์ หรือ DMA Controller ทิศทางการทำงาน output สถานะการทำงาน Active Low

-MEMR (Memory Read Command)

เป็นสัญญาณที่ใช้ในการติดต่อกับหน่วยความจำ เพื่อให้ข้อมูลเข้าสู่ Data Bus สัญญาณนี้ถูกสร้างขึ้นจากไมโครโปรเซสเซอร์ หรือ DMA Controller ทิศทางการทำงาน output สถานะการทำงาน Active Low

-MEMW (Memory Write Command)

เป็นสัญญาณที่ใช้ในการติดต่อกับหน่วยความจำ เพื่อให้ข้อมูลจาก Data Bus เข้าสู่หน่วยความจำ สัญญาณนี้ถูกสร้างขึ้นจากไมโครโปรเซสเซอร์ หรือ DMA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Controller ทิศทางการทำงาน output สถานะการทำงาน Active Low

-DRQ1 - DRQ3 (DMA Request 1 to 3)

เป็นสัญญาณอินเตอร์รัพท์ที่เกิดจากอุปกรณ์ ซึ่งต้องการให้เกิดขบวนการ DMA โดยเรียงลำดับความสำคัญต่ำสุดจาก DRQ3 ไปยังความสำคัญสูงสุด DRQ1 ทิศทางการทำงาน input สถานะการทำงาน Active High

-DACK0 - DACK3 (DMA Acknowledge 0 to 3)

เป็นสัญญาณตอบรับการอินเตอร์รัพท์ DMA Controller ทิศทางการทำงาน output สถานะการทำงาน Active Low .

-AEN (Address Enable)

เมื่อสัญญาณนี้เป็น HIGH address จะถูกใช้ในขบวนการ DMA และใช้ในการติดต่อกับพอร์ตต่างๆ ทิศทางการทำงาน output

-T/C (Terminal Count)

เมื่อเกิดขบวนการ DMA สัญญาณนี้จะเป็น High ทิศทางการทำงาน output

-แรงดันต่างๆและกราวด์

ในระบบส่วนขยายพอร์ตจะมีแรงดัน +5 V อยู่ 2 ขา แรงดัน -5 V 1 ขา แรงดัน +12 V 1 ขา แรงดัน -12 V อยู่ 1 ขา และกราวด์ GND (Ground) อยู่ 3 ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

๓. การคำนวณและการสร้าง

การทำงานมีส่วนต่างๆที่เกี่ยวข้องกันอยู่ ๕ ส่วนคือ

๑. ส่วนที่ทำหน้าที่ถอดรหัสแอดเดรส
๒. ส่วนจดจำคำสั่งและกำหนดค่าเริ่มต้น
๓. ส่วนตัดต่อ (switching) เชื่อมโยงการส่งผ่านข้อมูล
๔. ส่วนควบคุมการขัดจังหวะ
๕. ส่วนโปรแกรม (program) ควบคุมการสั่งงานครุฟันธ์ (hardware)
๖. ส่วนโปรแกรมประยุกต์ (application program)

๓.๑. ส่วนที่ทำหน้าที่ถอดรหัสแอดเดรส

ส่วนนี้ใช้อุปกรณ์เบอร์ 74688 เป็นตัวถอดรหัสแอดเดรสจากบัสแอดเดรส (bus address) โดยเมื่อมีการอ้างถึงแอดเดรสที่ตรงกับแอดเดรสที่ตั้งไว้ระบบก็จะเริ่มสู่การทำงานโดยค่าแอดเดรสที่ตั้งไว้ในโครงงานนี้คือ 0380h - 0381h สำหรับอุปกรณ์เบอร์ 74244 และอุปกรณ์เบอร์ 74245 เป็นตัวขับกระแสให้กับบัสแอดเดรสและบัสดาต้า (data bus) เพราะในระบบมีการใช้บัสต่างๆเหล่านี้อยู่หลายส่วน โดยในการถอดรหัสนี้ เราจะกำหนดให้มีการถอดรหัสเฉพาะเมื่อมีการกล่าวอ้างถึงการติดต่อกับอุปกรณ์ภายนอกเท่านั้น

๓.๒. ส่วนจดจำคำสั่งและกำหนดค่าเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนนี้ใช้อุปกรณ์เบอร์ 8255 เป็นตัวเก็บข้อมูลโดยข้อมูลที่ถูกป้อนเข้าไปจะไปทำการควบคุมลักษณะการทำงานในรูปแบบต่างๆ รวมถึงการรีเซ็ทระบบในตอนเริ่มเปิดเครื่องใหม่ๆด้วย

๓.๓. ส่วนต่อเชื่อมโยงการส่งผ่านข้อมูล

ส่วนนี้จะทำหน้าที่ในการต่อเชื่อมโยงการส่งผ่านข้อมูลให้ถูกต้องกับจังหวะและสัมพันธ์กับผู้ใช้ โดยก่อนที่จะมีการส่งผ่านข้อมูลใดๆก็ยังมีส่วนที่ทำหน้าที่ตรวจสอบความพร้อมของเครื่องของผู้ส่งและเครื่องของผู้รับทุกตัวให้มีความพร้อมเพื่อป้องกันความผิดพลาดของการส่งและป้องกันการสูญหายของตัวข้อมูลเองอีกด้วย โดยส่วนที่ใช้สำหรับการตรวจสอบนี้เรียกว่าโปรโตคอล (Protocol)

โปรโตคอลที่ใช้ในโครงงานนี้จะอาศัยมาตรฐานที่เรียกว่า อาร์เอส-232 (RS-232) ซึ่งเป็นโปรโตคอลที่ใช้สำหรับการสื่อสารข้อมูลแบบอนุกรม โดยขาที่ใช้สำหรับโครงงานนี้แบ่งเป็น ๒ ส่วนคือ

ส่วนของเครื่องแม่ (Master)
สำหรับเครื่องแม่ขาที่ใช้มีดังนี้คือ

-ดีทีอาร์ (DTR) ขานี้มีหน้าที่ใช้สำหรับการอินเตอร์รัพท์เพื่อขอเป็นผู้ส่งข้อมูล เช่น ถ้าเป็นผู้สอนทำการอินเตอร์รัพท์ก็จะหมายถึงต้องการเปลี่ยนไปเป็นโหมดของการสอน แต่ถ้าเป็นผู้เรียนทำการอินเตอร์รัพท์ก็จะหมายถึงต้องการเปลี่ยนไปเป็นโหมดของการป้อนคำถาม

-ดีเอสอาร์ (DSR) ขานี้เป็นขาที่ใช้รับสัญญาณอินเตอร์รัพท์จากเครื่องอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-อาร์ทีเอส (RTS) ขานี้ใช้สำหรับการร้องขอเพื่อจะตรวจสอบความพร้อมของผู้ที่จะทำการรับข้อมูลว่ามีความพร้อมหรือยัง และใช้สำหรับการตอบรับการขอส่งข้อมูลในกรณีที่เป็นโหมดรับ

-ซีทีเอส (CTS) ขานี้ใช้สำหรับรับการร้องขอจากตัวส่งและรับสัญญาณการตอบรับจากตัวรับในกรณีที่เป็นโหมดส่ง

-อาร์ไอ (RI) ขานี้ใช้สำหรับผู้ทำการสอนทำการส่งสัญญาณล็อคไม่ให้ตัวลูกทำการอินเตอร์รัพท์ ซึ่งจะใช้ในกรณีที่ผู้สอนมีความต้องการที่จะทำการส่งข้อมูลแต่เพียงฝ่ายเดียว โดยไม่ต้องการให้ตัวลูกมาขัดจังหวะซึ่งจะทำให้การสอนเป็นไปอย่างราบรื่น

-ทีเอกซ์ (TX) ขานี้เป็นขาที่ใช้สำหรับการส่งผ่านข้อมูลจริงๆ โดยก่อนที่ขานี้จะทำงานได้ จะต้องผ่านขาที่เป็นโปรโตคอลมาก่อน

-อาร์เอกซ์ (RX) ขานี้ใช้สำหรับการรับข้อมูล โดยมีข้อกำหนดเหมือนกับขาอาร์เอกซ์

-สายดิน (GROUND) ขานี้ใช้เป็นขาที่เป็นขาสัญญาณอ้างอิงให้แก่ระบบ

ส่วนของเครื่องลูก (Slave)

สำหรับส่วนของเครื่องลูกขาที่ใช้มีดังนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-ดีทีอาร์ (DTR) (เหมือนของเครื่องแม่)

-ดีเอสอาร์ (DSR) (เหมือนของเครื่องแม่)

-อาร์ทีเอส (RTS) (เหมือนของเครื่องแม่)

-ซีทีเอส (CTS) (เหมือนของเครื่องแม่)

-ทีเอกซ์ (TX) (เหมือนของเครื่องแม่)

-อาร์เอส (RX) (เหมือนของเครื่องแม่)

-สายดิน (GROUND) (เหมือนของเครื่องแม่)

๓.๔. <u>ส่วนควบคุมการชัดจังหวะ</u>

ส่วนนี้ใช้เป็นตัวควบคุมการเข้าชัดจังหวะโดยตัวลูก ซึ่งเป็นไปได้ว่าเกิดการเข้าชัดจังหวะพร้อมกันยกตัวอย่างเช่นขณะอาจารย์สอนเสร็จและเปิดโอกาสให้นักศึกษาทำการถาม ซึ่งในเวลานี้ถ้าเกิดมีจำนวนเครื่องที่เป็นตัวลูกมากโอกาสที่จะเกิดการชนกันย่อมจะมีสูงมาก

แต่สำหรับกรณีทั่วๆไปโอกาสที่จะเข้าชัดจังหวะไม่พร้อมกันระบบก็สามารถที่จะทำงานได้ตามปกติ โดยไม่จำกัดรูปแบบของการเข้าชัดจังหวะว่าจะเป็นอย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของวงจรจะอาศัย ชิฟท์เรสจิสเตอร์ (Shift register) เป็นตัวช่วยในการตัดสินการณีที่มีการเข้าอินเตอร์รัพท์พร้อมกัน ยกตัวอย่างเช่น ในสถานะก่อนหน้าที่จะมีการเข้าอินเตอร์รัพท์พร้อมกันนั้น เครื่องลูกข่ายที่หนึ่งเป็นผู้ได้รับบริการจากเครื่องแม่ให้ถามก่อน เพราะฉะนั้นการอินเทอร์รัพท์พร้อมกันในครั้งนี้วงจรควบคุมการอินเตอร์รัพท์จะเปิดโอกาสให้กับเครื่องลูกข่ายที่สองในการรับบริการ อย่างไรก็ตามวงจรส่วนนี้จะยังไม่ค่อยมีประโยชน์เท่าไรนักในขั้นนี้ แต่ถ้ามีการขยายการบริการ โดยเพิ่มตัวลูกมากขึ้นแน่นอนโอกาสที่จะเกิดการชนกันย่อมจะมีสูงมาก และวงจรควบคุมการอินเตอร์รัพท์นี้จะได้ใช้ประโยชน์มากขึ้น

ในวงจรส่วนนี้ผู้จัดทำยังได้เพิ่มโหมดพิเศษที่ผู้ทำการสอนสามารถที่จะส่งจำเพาะเจาะจงลงไปได้อีกว่าต้องการจะติดต่อกับเครื่องไหนโดยเฉพาะ โดยข้อมูลที่ป้อนในส่วนนี้จะถูกป้อนโดยผ่านทางพอร์ท 8255

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

๓.๕. <u>ส่วนโปรแกรมควบคุมการส่งงานครุพันธ์</u>

ส่วนนี้จะมีส่วนของการควบคุมโดยผ่านโปรโตคอลเพื่อควบคุมการส่งผ่านข้อมูลให้มีความถูกต้องยิ่งขึ้น โดยการส่งผ่านของข้อมูลจะเกิดก็ต่อเมื่อฝ่ายรับและฝ่ายส่งมีความพร้อมทั้งคู่เท่านั้น และขณะทำการส่งผ่านของข้อมูลจากฝ่ายหนึ่งไปยังอีกฝ่ายหนึ่งผู้รับก็สามารถจะทำการขัดจังหวะเพื่อขอเป็นผู้ส่งข้อมูลเองได้

ในส่วนนี้ยังมีโปรแกรมที่ใช้สำหรับการแสดงที่หน้าจอทั้งของผู้ทำการสอนและผู้เรียนเพื่อให้ผู้ใช้ทั้งสองฝ่ายทราบว่าขณะนี้ตัวเองกำลังอยู่ในโหมดไหน โดยสำหรับโครงงานนี้ เนื่องจากได้ทำไว้สำหรับตัวลูกเพียงสองตัวเท่านั้น ผู้จัดทำจึงได้แบ่งส่วนของการเข้าขัดจังหวะของตัวลูกแสดงผลออกเป็นสองส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

๓.๖. **ส่วนโปรแกรมประยุกต์**

(ส่วนนี้อยู่ในโครงงานของอีกกลุ่มหนึ่งที่ประสานงานกัน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การทดลองและผล

จากหลักการทำงานที่ได้กล่าวไว้ข้างต้น เมื่อ load โปรแกรม "screen" และโปรแกรม "pulldown" เพื่อทดลองการส่งการรับผลที่ได้จะปรากฏทั้งที่ตัว Master และที่ตัว Slave ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

>

>screen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ogram manual :
is is the sample program that will simulate i/o interrupt
u required
age of this program is as followed :-
    F1  -- terminate this program
    F10 -- toggle between master & slave
 both windows (master & slave) you can type anything you wish
 you can do in any word processor program
u can use BACKSPACE to delete one character left
mark : each screen cannot scroll so that you
       must not type too long
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Master Job

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    Master Job

 am a teacher and I want to teach you ,
f you have any question please strike the 'F 10' key.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Master Job

m a teacher and I want to teach you ,
you have any question please strike the 'F 10' key.

' Let all study . This is the Physics class .
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Master Job

a teacher and I want to teach you .
ou have any question please strike the F10 key

Let all study . this is the Physics class .

Slave Job

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Master Job

a teacher and I want to teach you ,
ou have any question please strike the 'F 10 key.

Let all study . This is the Physics class .

Slave Job

on't understand what are you teaching.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Master Job

m a teacher and I want to teach you ,
you have any question please str.'e the 'F 10' key.

' Let all study . This is the Physics class .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Master Job

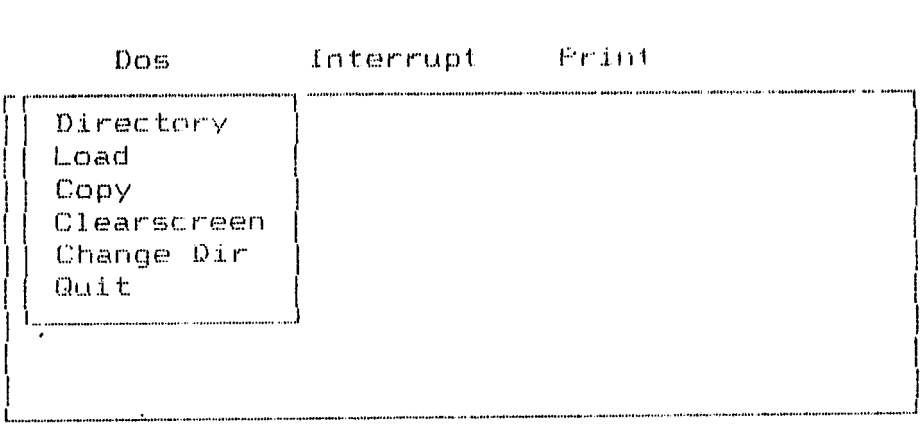
a teacher and I want to teach you ,
ou have any question pleAse strike the F 10' key.

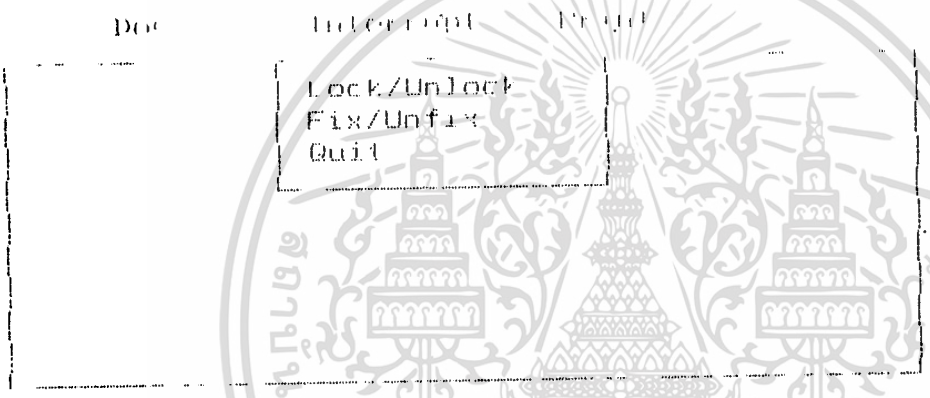
! Let all study . This is the Physics class .

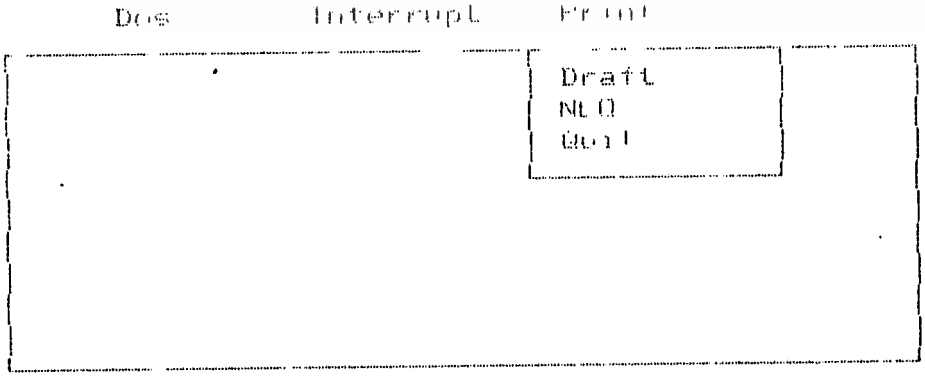
just testing if you can receive my message.
let's actually begin our class.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้







เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการทดลองที่ได้เป็นเพียงส่วนหนึ่งของลมุนพันธ์พื้นฐานเท่านั้นหากต้องการจะนำไปใช้งานจริง ควรเพิ่มให้มีฟังก์ชันพิเศษเฉพาะอย่างเพื่อให้เหมาะกับงานนั้นๆโดยเฉพาะ

หมายเหตุ

ส่วนของหน้าจอที่แสดงออกให้ดูนี้เป็นเพียงส่วนหนึ่งของลมุนพันธ์ทั้งหมดเท่านั้น เนื่องจากยังมีลมุนพันธ์บางส่วนที่ยังไม่ได้ปรับให้เข้ากับครุพันธ์บนการ์ด หากต้องการข้อมูลโดยละเอียดโปรดติดต่อกับผู้จัดทำโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

๕. บทวิจารณ์และสรุป

การสื่อสารในปัจจุบันเริ่มมีการตื่นตัวในการพัฒนาไปสู่การนำเอาคอมพิวเตอร์มาใช้มากขึ้นซึ่งจะทำให้การสื่อสารมีความรวดเร็วและมีประสิทธิภาพมากยิ่งขึ้นซึ่งในโครงงานนี้ก็เป็นส่วนหนึ่งซึ่งพยายามนำเอาคอมพิวเตอร์เข้ามาใช้ในการเรียนการสอนเพื่อเพิ่มประสิทธิภาพให้การเรียนเป็นไปได้อย่างสะดวกและรวดเร็ว

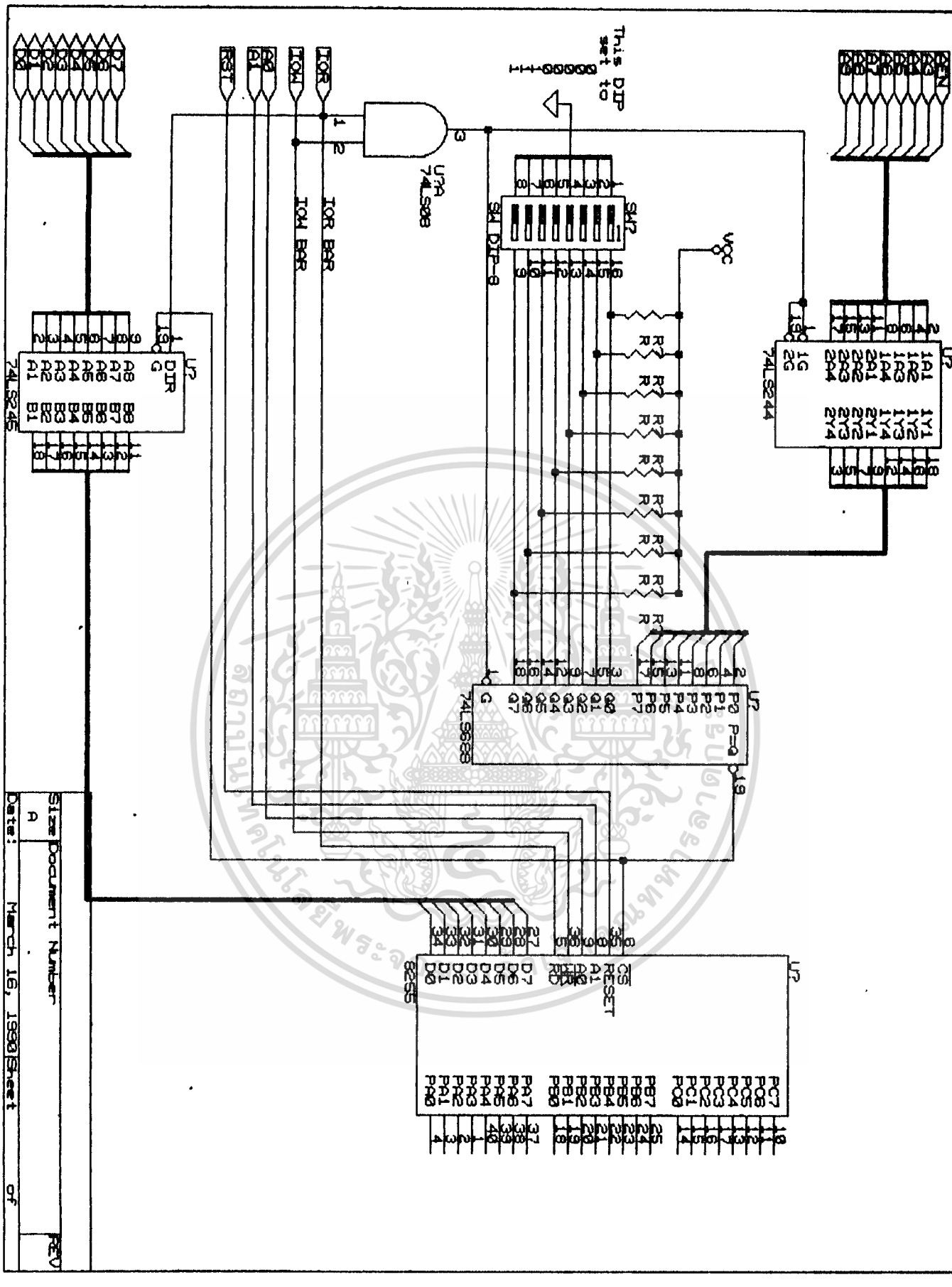
ในโครงงานนี้ระบบโครงข่ายมีเพียงตัวแม่ ๑ ตัวและตัวลูกอีก ๒ ตัวเท่านั้น แต่ถ้าหากต้องการนำไปใช้งานจริงๆก็สามารถทำได้ โดยการเปลี่ยนวงจรที่ทำหน้าที่ควบคุมการอินเตอร์รัพท์ให้สามารถทำการควบคุมการอินเตอร์รัพท์ได้มากขึ้น และเปลี่ยนส่วนที่ทำหน้าที่แสดงผลให้มีความซับซ้อนมากยิ่งขึ้นเพื่อให้สามารถควบคุมเครื่องที่เป็นตัวลูกได้อย่างทั่วถึง

โครงงานนี้นับว่าเป็นโครงงานที่มีความอ่อนตัวมากที่จะนำไปประยุกต์ใช้กับงานอื่นๆได้ไม่จำเป็นที่จะต้องเกี่ยวกับการเรียนการสอนเพียงอย่างเดียวขึ้นอยู่กับผู้ใช้ว่าต้องการประยุกต์ไปใช้งานในด้านใดโดยส่วนสำคัญที่จะช่วยเพิ่มให้การใช้งานเป็นไปอย่างสะดวกและทำให้น่าใช้มากยิ่งขึ้นคือซอฟแวร์ (Software) ซึ่งในส่วนนี้จะมีความแตกต่างไปขึ้นอยู่กับงานที่ใช้

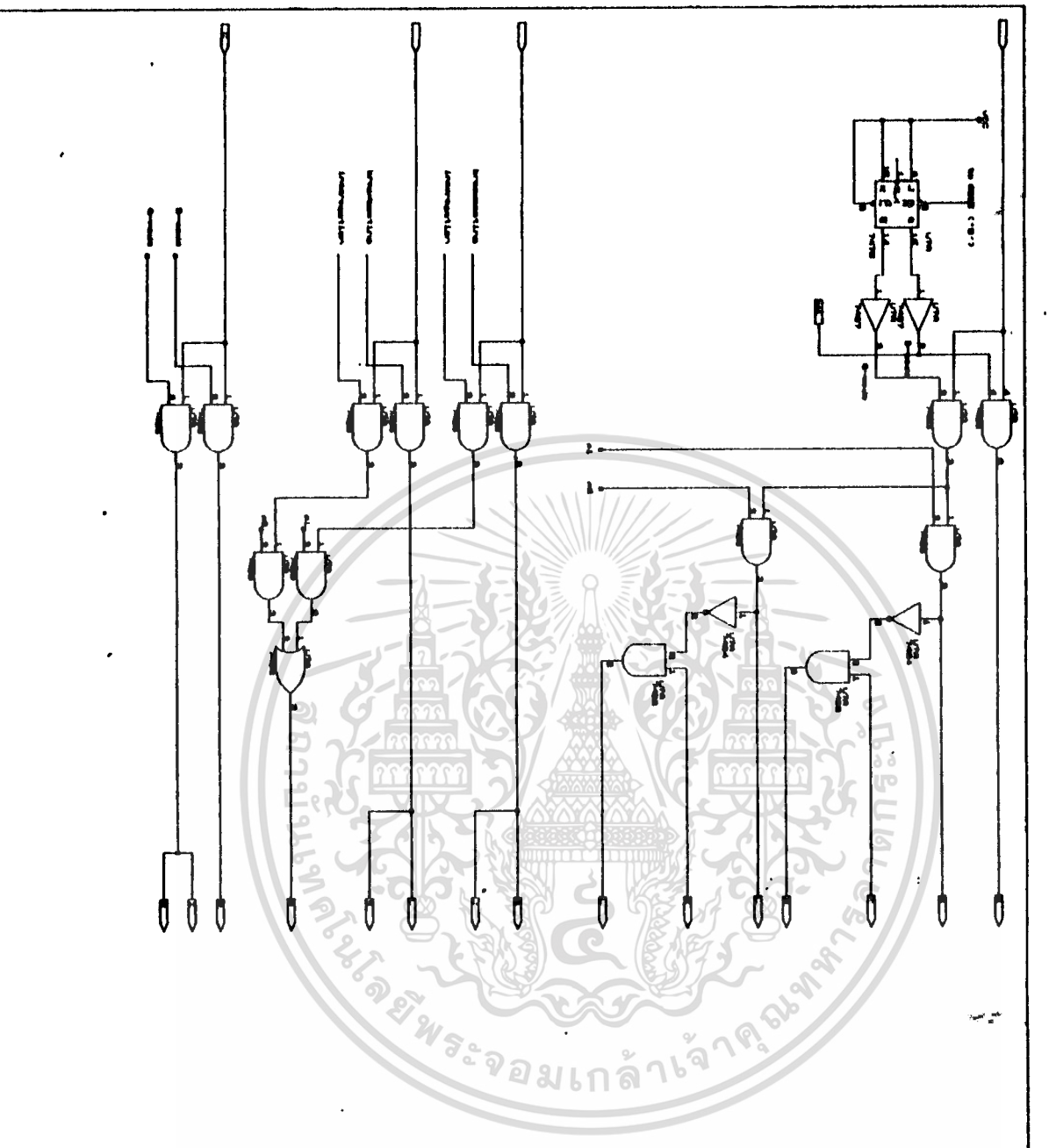
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

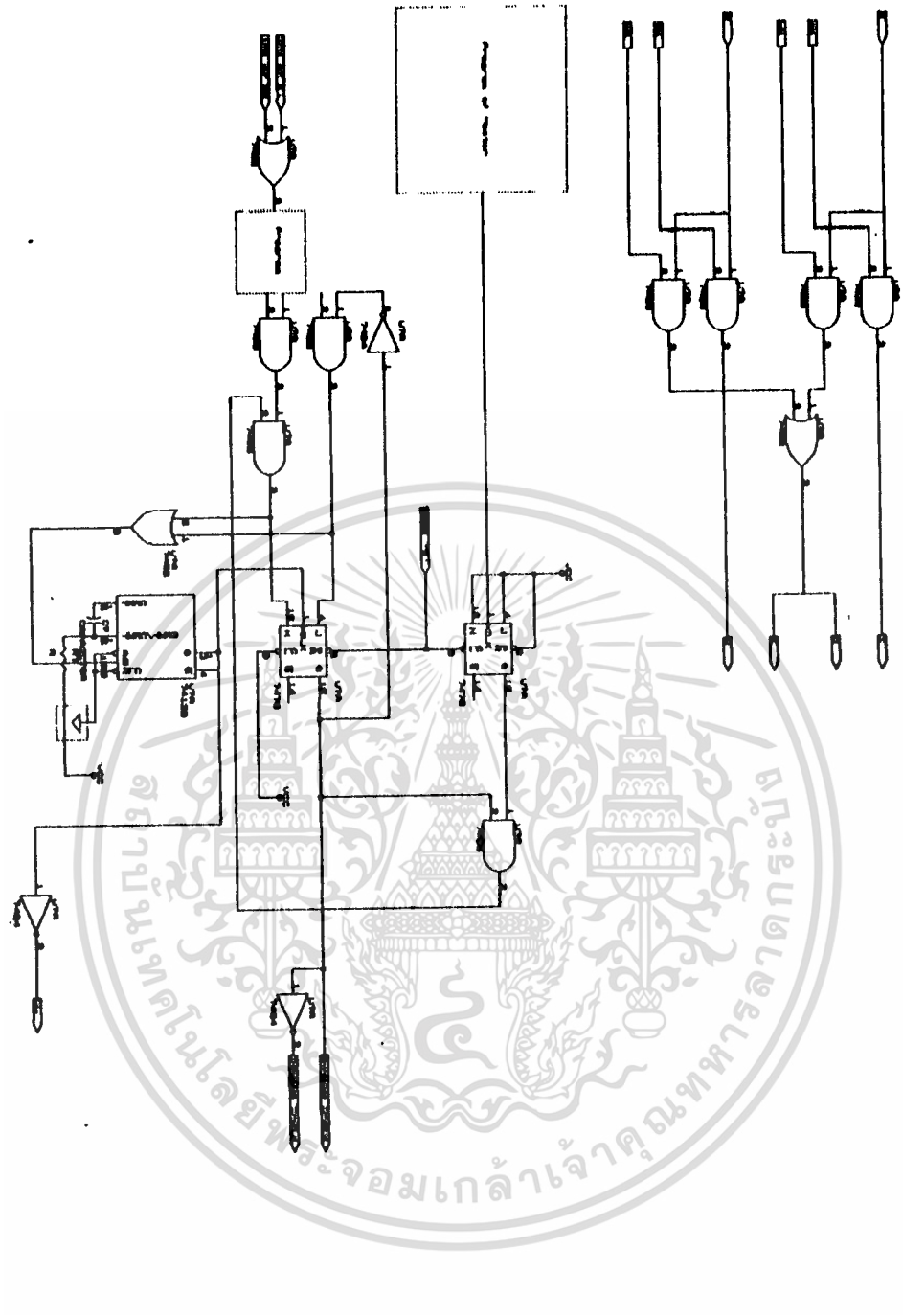
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



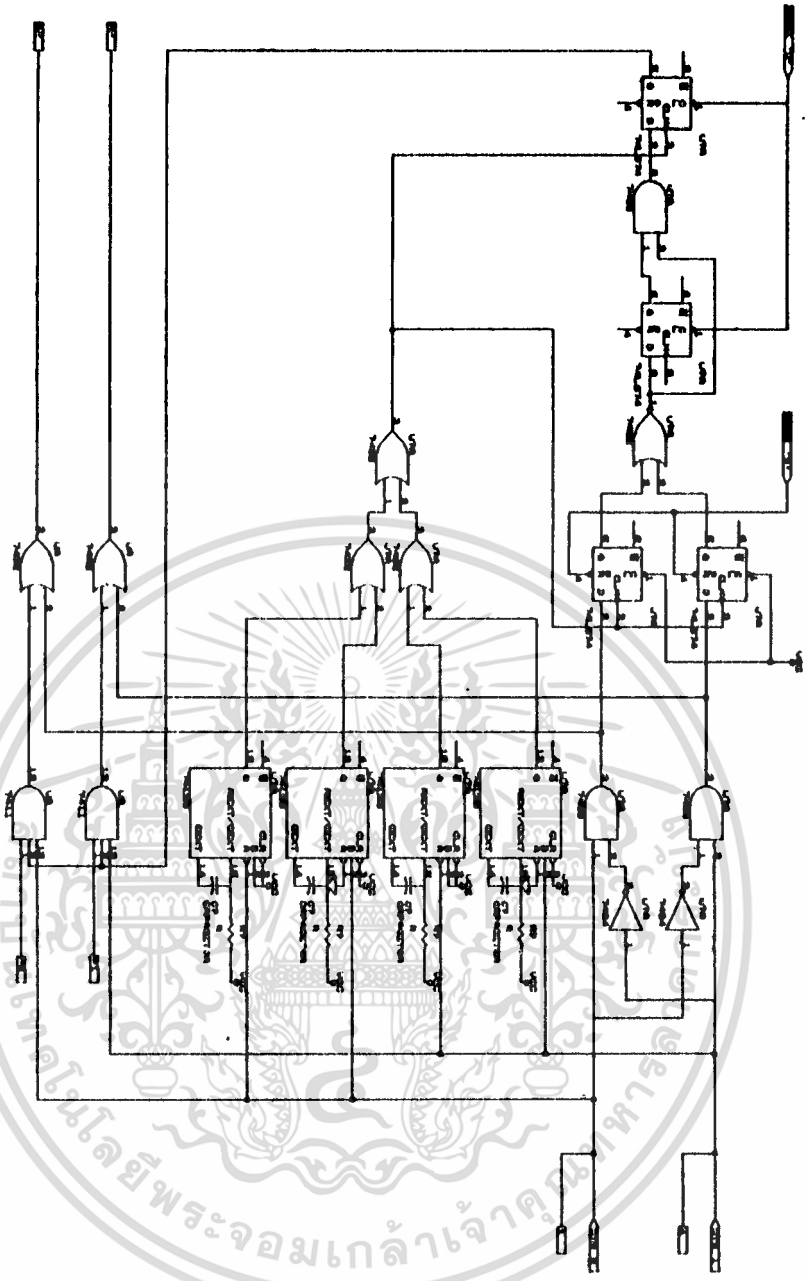
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



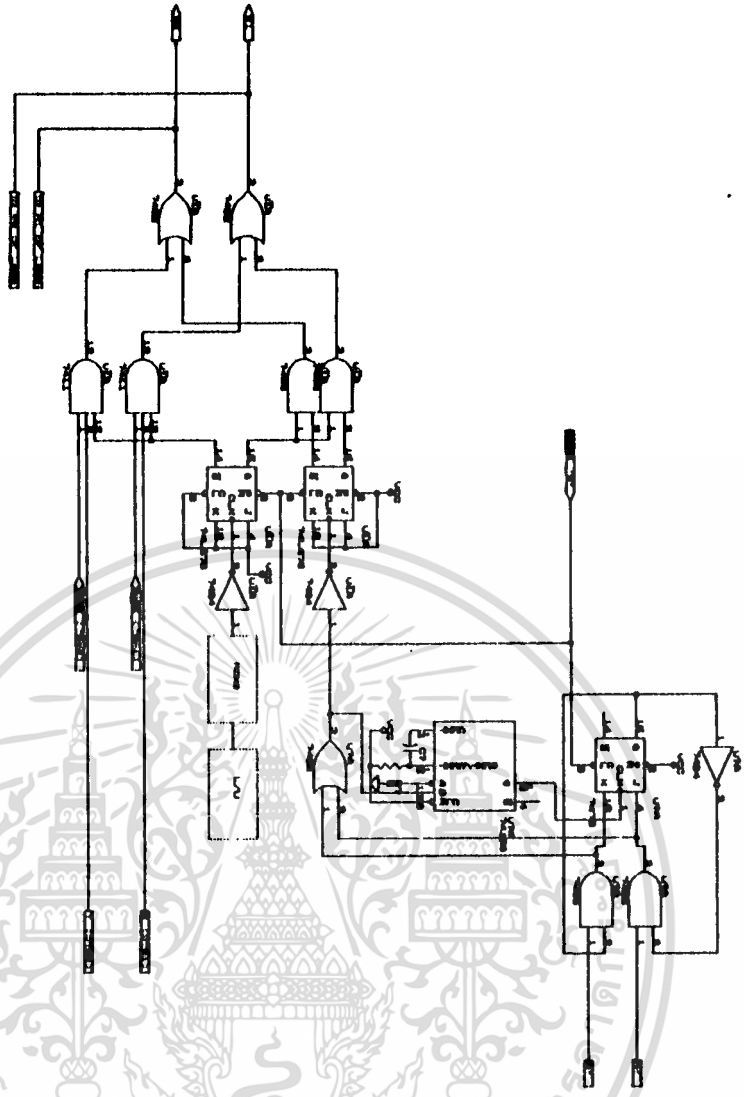
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



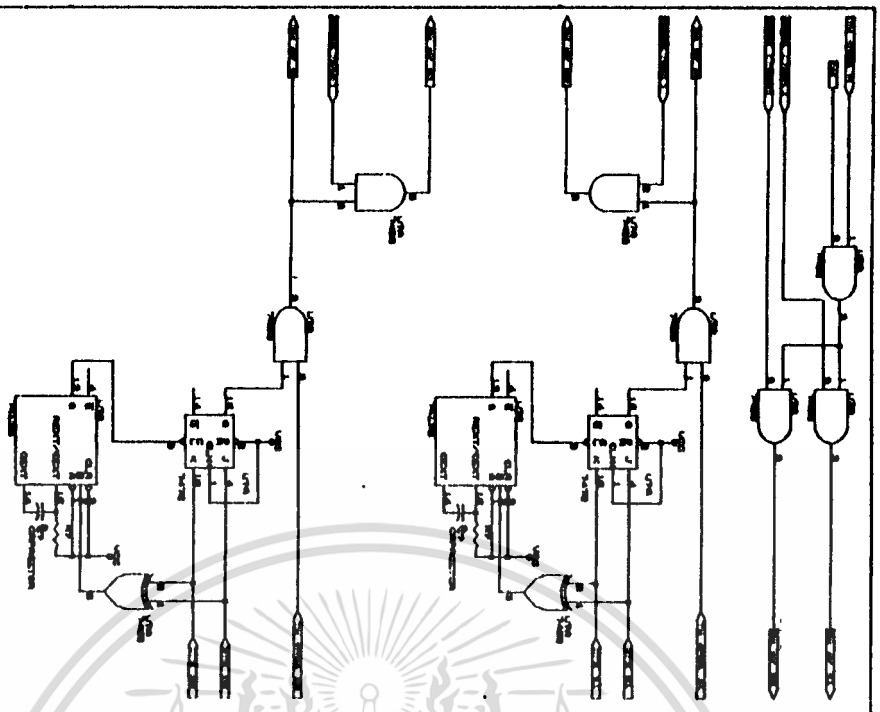
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include  <alloc.h>

#define   LOWDISPLAY      0x07
#define   NORMAL          0x0F
#define   REVLOW          0x70
#define   REVNORM         0x78
#define   BACKSPACE       0x08
#define   ENTER           0x0D
#define   SPACE           0x20
#define   F1              0x3B
#define   F10             0x44

void boarder (int left,int top,int right,int bottom)
{       int c;

        gotoxy (left-1,top-1); printf ("I");
        for (c=left;c<=right;c++)
        {       gotoxy (c,top-1); printf ("M"); }
        gotoxy (right+1,top-1); printf (";");
        gotoxy (right+1,top); printf (":");
        gotoxy (right+1,top+1); printf ("9");
        for (c=left;c<=right;c++)
        {       gotoxy (c,top+1); printf ("M"); }
        for (c=top+2;c<=bottom;c++)
        {       gotoxy (right+1,c); printf (":"); }
        gotoxy (right+1,bottom+1); printf ("<");
        for (c=left;c<=right;c++)
        {       gotoxy (c,bottom+1); printf ("M"); }
        gotoxy (left-1,bottom+1); printf ("H");
        for (c=top+2;c<=bottom;c++)
        {       gotoxy (left-1,c); printf (":"); }
        gotoxy (left-1,top+1); printf ("L");
        gotoxy (left-1,top); printf (":");
        window (left,top,right,bottom);
}

void slave_job ()
{   void *scr;
    int  size;
    char c;

    size = (76-2) * (22-14);
    scr = malloc (size);
    gettext (3,13,78,23,scr);
    window (1,1,80,25);
    boarder (4,14,77,22);
    textattr (REVLOW);
    gotoxy (1,1);
    cprintf ("                                        ");
    cprintf ("Slave Job                                ");
    textattr (NORMAL);
    window (1,1,80,25);
    window (4,16,77,22);
    clrscr ();
    while (1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{       c = getch ();
        if (c == 0)
        {  if ((c=getch()) == F10)
              break;
        }
        else
        {    if (c == ENTER)
             {  gotoxy (1,wherey()+1);
                if ((wherey()+1) > 7)
                   clrscr ();
             }
             else
                 if (c == BACKSPACE)
                 {  gotoxy (wherex()-1,wherey());
                    putch (SPACE);
                    gotoxy (wherex()-1,wherey());
                 }
                 else  putch (c);
        }
    }
    puttext (3,13,78,23,scr);
    free (scr);
}

void main_job ()
{   char  c;
    int   cur_x,cur_y;

    boarder (2,2,79,23);
    textattr (REVLOW);
    gotoxy (1,1);
    cprintf ("                                          ");
    cprintf ("Master Job                                ");
    textattr (NORMAL);
    window (2,4,79,23);
    while (1)
    {       c = getch ();
            if (c == 0)
            {  c = getch ();
               if (c == F1)
                  break;
               else
                   if (c == F10)
                   {  cur_x = wherex ();
                      cur_y = wherey ();
                      slave_job ();
                      window (2,4,79,23);
                      gotoxy (cur_x,cur_y);
                   }
            }
            else
            {    if (c == ENTER)
                 {  gotoxy (1,wherey()+1);
                    if ((wherey()+1) > 20)
                       clrscr ();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
            else
                if (c == BACKSPACE)
                {  gotoxy (wherex()-1,wherey());
                   putch (SPACE);
                   gotoxy (wherex()-1,wherey());
                }
                else   putch (c);
        }
    }
}

void manual ()
{   printf ("program manual : \n"
            "this is the sample program that will simulate "
            "i/o interrupt\nyou required\n"
            "usage of this program is as followed :- \n"
            "       F1  -- terminate this program\n"
            "       F10 -- toggle between master & slave\n"
            "in both windows (master & slave) you can type "
            "anything you wish\nas you can do in "
            "any word processor program\n"
            "you can use BACKSPACE to delete one character left\n"
            "remark : each screen cannot scroll so that you\n"
            "             must not type too long\n\n"
            "if any problems occur with you ..... \n"
            "please contact me\n"
            "akkamon srihiran\n"
            "  600/1092 sivalee pathumthani 12130 -- by mail\n"
            "  or tel. 5322599 <before 11:00 PM>\n");
    getch ();
}

void main ()
{   clrscr ();     manual ();
    clrscr ();     main_job ();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include  <dos.h>
#include  <alloc.h>
#include  <conio.h>

#define   LOWDISPLAY      0x07
#define   NORMAL          0x0f
#define   UNDLOW          0x01
#define   REVLOW          0x70
#define   REVNORM         0x78
#define   UP              0x48
#define   DOWN            0x50
#define   SP              0x20

typedef char charmenu[19];

cursorON ()
{
        union REGS r;
        r.h.ah = 1;
        r.h.ch = 12;
        r.h.cl = 13;
        int86 (0x10,&r,&r);
}

cursorOFF ()
{
        union REGS r;
        r.h.ah = 1;
        r.h.ch |= 20;
        int86 (0x10,&r,&r);
}


boarder (int left,int top,int right,int bottom)
{
        int c;
        gotoxy (left,top); printf ("I");
        for (c=left+1;c<=right-1;c++)
        {       gotoxy (c,top); printf ("M"); }
        gotoxy (right,top); printf (";");
        gotoxy (right,top+1); printf (":");
        gotoxy (right,top+2); printf ("9");
        for (c=left+1;c<=right-1;c++)
        {       gotoxy (c,top+2); printf ("M"); }
        for (c=top+3;c<=bottom-1;c++)
        {       gotoxy (right,c); printf (":"); }
        gotoxy (right,bottom); printf ("<");
        for (c=left+1;c<=right-1;c++)
        {       gotoxy (c,bottom); printf ("M"); }
        gotoxy (left,bottom); printf ("H");
        for (c=top+3;c<=bottom-1;c++)
        {       gotoxy (left,c); printf (":"); }
        gotoxy (left,top+2); printf ("L");
        gotoxy (left,top+1); printf (":");
        window (left,top,right,bottom);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
56  }
57
58  win (xlf,yup,xrt,ydn,text)
59  int     xlf,yup,xrt,ydn;
60  char    text[15];
61  {       void  *screen;
62          int   siz,i,j;
63
64          siz = (xrt - xlf) * (ydn - yup);
65          screen = malloc (siz);
66          window (1,1,80,25);
67          gettext (xlf,yup,xrt,ydn,screen);
68          window (xlf,yup,xrt,ydn);
69          clrscr ();
70          textattr (REVLOW);
71          gotoxy (1,2);  cprintf ("      %s      ",text);
72          gotoxy (1,3);  cprintf ("    press a key      ");
73          textattr (NORMAL);
74          getch ();
75          puttext (xlf,yup,xrt,ydn,screen);
76          free (screen);
77  }
78
79  light (int *cur)
80  {
81           int a,c,x,i;
82           charmenu menu1[7];
83           a = *cur;
84           strcpy (menu1[0]," Places input      ");
85           strcpy (menu1[1]," Transition input ");
86           strcpy (menu1[2]," Tokens input      ");
87           strcpy (menu1[3]," Input set         ");
88           strcpy (menu1[4]," Output set        ");
89           strcpy (menu1[5]," Fire              ");
90           strcpy (menu1[6]," Exit              ");
91           i = a; x = a+4;
92           gotoxy (2,x); textattr (REVNORM); cprintf ("%s",menu1[i]);
93           while ((c = getch ()) != 0x0d)
94           {
95                   if (c == UP)
96                   {
97                           gotoxy (2,x); textattr (LOWDISPLAY); cprin
     f ("%s",menu1[i]);
98                           if (x == 4)
99                           {       x = 11; i = 7; }
00                           gotoxy (2,--x); textattr (REVNORM); cprint
      ("%s",menu1[--i]);
01                   }
02                   if (c == DOWN)
03                   {
04                           gotoxy (2,x); textattr (LOWDISPLAY); cprin
     f ("%s",menu1[i]);
05                           if (x == 10)
06                           {       x = 3; i = -1; }
07                           gotoxy (2,++x);textattr (REVNORM); cprintf
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
("%s",menu1[++i]);
                }
                a = *cur = i;
        }
        textattr (NORMAL);
        switch (a)
        {
                case 0 : win (2,2,21,5,   "placeinput");
                         break;
                case 1 : win (2,10,21,13,  "transition");
                         break;
                case 2 : win (2,20,21,23, "  tokens  ");
                         break;
                case 3 : win (31,2,50,5,  "  input   ");
                         break;
                case 4 : win (60,2,79,5,  "  output  ");
                         break;
                case 5 : win (60,10,79,13,"   fire   ");
                         break;
                case 6 : win (60,20,79,23,"   exit   ");
                         break;
        }
}

menu (int *cursor)
{       window (1,1,80,25);
        boarder (31,7,50,17);
        textattr (REVLOW);
        gotoxy (2,2); cprintf ("    MAIN  MENU    ");
        gotoxy (2,4); textattr (LOWDISPLAY); cprintf (" Places inpu
t     ");
        gotoxy (2,5); textattr (LOWDISPLAY); cprintf (" Transition
input ");
        gotoxy (2,6); textattr (LOWDISPLAY); cprintf (" Tokens inpu
t     ");
        gotoxy (2,7); textattr (LOWDISPLAY); cprintf (" Input set
      ");
        gotoxy (2,8); textattr (LOWDISPLAY); cprintf (" Output set
      ");
        gotoxy (2,9); textattr (LOWDISPLAY); cprintf (" Fire
      ");
        gotoxy (2,10); textattr (LOWDISPLAY); cprintf (" Exit
      ");
        light (cursor);
}

main ()
{    int  i,j,size,cur;
     char  c;
     void  *scr;

     clrscr ();
     cursorOFF ();
     for (i=1;i<=24;i++)
         for (j=1;j<=80;j++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
5           {   gotoxy (j,i);
6               putch (176);
7           }
8       size = (51-30) * (18-6);
9       scr = malloc (size);
0       gettext (31,7,50,17,scr);
1       textattr (NORMAL);
2       cur = 0;
3       while (cur != 6)
4             menu (&cur);
5       puttext (31,7,50,17,scr);
6       free (scr);
7       getch ();
8       cursorON ();
9   }
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1     #include <conio.h>
2     #include <dos.h>
3     #include <stdio.h>
4     #define   Normal    0x07
5     #define   Reverse   0x70
6     #define   bmono     0xB000
7     #define  LINE_a   218    /* draw  Z */
8     #define  LINE_b   191    /* draw  ? */
9     #define  LINE_c   196    /* draw  D */
10    #define  LINE_d   179    /* draw 3  */
11    #define  LINE_e   192    /* draw  @ */
12    #define  LINE_f   217    /* draw  Y */
13    #define  UP       0x0048
14    #define  DOWN     0x0050
15    #define  LEFT     0x004B
16    #define  RIGHT    0x004D
17    #define  MAX_1    4
18    #define  MIN_1    9
19    #define  MIN_2    6
20    #define  MIN_3    6
21    #define  Col_1    10
22    #define  Col_2    23
23    #define  Col_3    36
24    #define  TRUE     1
25    #define  FALSE    0
26
27    int  x,y,X1,Y1,X2,Y2,Back_g,Row_win,Col_win,Min,Length,Select ;
28    int  Line_1,Line_2,Line_3,Ready,Ready_1,Ready_2,Ready_3;
29    unsigned int far *cursormode = (unsigned int far *)0x00400060L;
30    unsigned int far *videoport  = (unsigned int far *)0x00400063L;
31    char  buf_A[8*15*2],buf_B[5*18*2],buf_C[5*14*2],buf_1[8*15*2],
32          buf_2[5*18*2],buf_3[5*14*2];
33
34
35    char  *menu[3] = {" File          "," Display      "," Generate
      "};
36                     /* (L2,C10)        (L2,C23)        (L2,C36) *
37
38    char  *menu1[6] = {" Read Port   ",     /* L4,C10
      */
39                       " Load         ",     /* Col_L =  Col_1 - 1
      */
40                       " Write to     ",     /* Row_L =  MAX_1 - 1
      */
41                       " Directory    ",     /* Col_R =  23
      */
42                       " Change Dir   ",     /* Row_R =  MIN_1 + 1
      */
43                       " Quit         "};
44     char  *menu2[3] = {" Select Picture ",  /* L4,C23 */
45                       " Statics        ",  /* Col_L =  Col_2 - 1
      */
46                       " Continuous     "}; /* Row_L =  MAX_1 - 1
      */
47                                            /* Col_R =  39
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    */
48                                          /* Row_R =  MIN_2 + 1
    */
49
50      char  *menu3[3] = {" Sine Wave  ",      /* L4,C36 */
51                         " Triangle   ",      /* Col_L =  Col_3 - 1
    */
52                         " Square     "};     /* Row_L =  MAX_1 - 1
    */
53                                          /* Col_R =  48
    */
54                                          /* Row_R =  MIN_3 + 1
    */
55
56      void cursor_on()
57      {
58         outport(*videoport,10);
59         outport(*videoport + 1,((unsigned char)(*cursormode >> 8)) &
    0xDF);
60         outport(*videoport,11);
61         outport(*videoport+1,(unsigned char)*cursormode);
62      }
63
64      void cursor_off()
65      {
66         outport(*videoport,10);
67         outport(*videoport + 1,((unsigned char)(*cursormode >> 8))
    0x20);
68         outport(*videoport,11);
69         outport(*videoport+1,(unsigned char)*cursormode);
70      }
71
72      void set_text_attr(x,y,Back_g)
73      {
74         textattr(Back_g);
75         gotoxy(x,y);
76      }
77
78      void cls_window(X1,Y1,X2,Y2)
79      {
80        int  i,j,offset;
81        char   blank = ' ';
82        Back_g = Normal;
83
84        for(i=Y1 ; i<=Y2 ; i++)          /* Y1 and X1 Start from 1 */
85          {
86             for(j=X1 ; j<=X2 ; j++)
87              {
88                 offset  = ((i-1)*160)+2*(j-1);
89                 pokeb(bmono,offset ,blank);
90                 pokeb(bmono,offset +1,Back_g);
91              }
92          }
93      }
94
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
05      void change_attr(x,old_y,new_y,Length)
06      {
07        int offset,i,j;
08        if(old_y != new_y)
09         {
00            for(i=0 ; i<Length ; i++)
01            {
02               offset = ((old_y-1)*160) + 2*(x+i-1);
03               pokeb(bmono,offset + 1,Normal);
04            }
05         }
06            for(j=0 ; j<Length ; j++)
07            {
08               offset = ((new_y-1)*160) + 2*(x+j-1);
09               pokeb(bmono,offset + 1,Reverse);
10            }
 1      }
12
13      void draw_rectan(X1,Y1,X2,Y2)
 4      {
15         int j,dX,dY;
16         dX = X2 - X1;
17         dY = Y2 - Y1;
18         Back_g = Normal;
19         for(j=0 ; j<=(dX-2) ; j++)
20         {
21              set_text_attr(X1+1+j,Y1,Back_g);
22              cprintf("D");
23              set_text_attr(X1+1+j,Y2,Back_g);
24              cprintf("D");
25         }
26         for(j=0 ; j<=(dY-2) ; j++)
27         {
28              set_text_attr(X1,Y1+1+j,Back_g);
29              cprintf("3");
30              set_text_attr(X2,Y1+1+j,Back_g);
31              cprintf("3");
32         }
33         angle_t(X1,Y1,LINE_a);
34         angle_t(X2,Y1,LINE_b);
35         angle_t(X1,Y2,LINE_e);
36         angle_t(X2,Y2,LINE_f);
37
38       }
39
40      int  angle_t(X1,Y1,route)
41      {
42             int offset;
43             Back_g = Normal;
44             offset = ((Y1-1)*160) + 2*(X1-1);
45             pokeb(bmono,offset ,route);
46             pokeb(bmono,offset + 1,Back_g);
47      }
48
49      void start()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
50      {
51          int i;
52          clrscr();
53          cursor_off();
54          draw_rectan(8,3,55,13);
55          gettext(9,3,23,10,buf_1);
56          gettext(22,3,39,10,buf_2);
57          gettext(35,3,48,10,buf_3);
58          draw_rectan(9,3,23,10);
59          gotoxy(10,2);
60          printf("    File      Display      Generate ");
61          for(i=0 ; i<6 ; i++)
62            {
63               gotoxy(10,4+i);
64               printf("%s\n",menu1[i]);
65            }
66          change_attr(10,4,4,13);
67      }
68
69      int Left_Window_Key(Col_win,Line,Ready)
70      {
71          int i;
72          static int Ready_2,Ready_3;
73          switch(Col_win)
74      .   {
75            case Col_1:  Line_3 = Line;
76                         gettext(9,3,23,10,buf_A);
77                         cls_window(9,3,23,10);
78                         puttext(9,3,23,10,buf_1);
79                         Ready_3 = Ready;
80                         switch(Ready_3)
81                         {
82                           case TRUE:   puttext(35,3,48,7,buf_C);
83                                        change_attr(Col_3,Line_3,Line
    ,12);
84                                        break;
85                           case FALSE:  draw_rectan(35,3,48,7);
86                                        for(i=0 ; i<3 ; i++)
87                                        {   gotoxy(36,4+i);
88                                            printf("%s\n",menu3[i]);
89                                        }
90                                        change_attr(36,4,4,12);
91                                        break;
92                         }
93                         break;
94            case Col_2:  Line_1 = Line;
95                         gettext(22,3,39,7,buf_B);
96                         cls_window(22,3,39,7);
97                         puttext(22,3,39,7,buf_2);
98                         puttext(9,3,23,10,buf_A);
99                         change_attr(Col_1,Line_1,Line_1,13);
00                         break;
01            case Col_3:  Line_2 = Line;
02                         gettext(35,3,48,7,buf_C);
03                         cls_window(35,3,48,7);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        puttext(35,3,48,7,buf_3);
                        Ready_2 = Ready;
                        switch(Ready_2)
                        {
                         case TRUE:  puttext(22,3,39,7,buf_B);
                                     change_attr(Col_2,Line_2,Line_2,1
6);
                                     break;
                         case FALSE: draw_rectan(22,3,39,7);
                                     for(i=0 ; i<3 ; i++)
                                     {   gotoxy(23,4+i);
                                         printf("%s\n",menu2[i]);
                                     }
                                     change_attr(23,4,4,16);
                                     break;
                        }
                        break;
        }
    }

    int Right_Window_Key(Col_win,Line,Ready)
    {
       int i;
       static int Ready_2,Ready_3;
       switch(Col_win)
       {
         case Col_1:  Line_2 = Line;
                      gettext(9,3,23,10,buf_A);
                      cls_window(9,3,23,10);
                      puttext(9,3,23,10,buf_1);
                      Ready_2 = Ready;
                      switch(Ready_2)
                      {
                        case TRUE:   puttext(22,3,39,7,buf_B);
                                     change_attr(Col_2,Line_2,Line_1
,16);
                                     break;
                        case FALSE:  draw_rectan(22,3,39,7);
                                     for(i=0 ; i<3 ; i++)
                                     {   gotoxy(23,4+i);
                                         printf("%s\n",menu2[i]);
                                     }
                                     change_attr(23,4,4,16);
                                     break;
                      }
                      break;
         case Col_3:  Line_1 = Line;
                      gettext(35,3,48,7,buf_C);
                      cls_window(35,3,48,7);
                      puttext(35,3,48,7,buf_3);
                      puttext(9,3,23,10,buf_A);
                      change_attr(Col_1,Line_1,Line_1,13);
                      break;
         case Col_2:  Line_3 = Line;
                      gettext(22,3,39,7,buf_B);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
257                         cls_window(22,3,39,7);
258                         puttext(22,3,39,7,buf_2);
259                         Ready_3 = Ready;
260                         switch(Ready_3)
261                         {
262                          case TRUE:  puttext(35,3,48,7,buf_C);
263                                      change_attr(Col_3,Line_3,Line_3,.
    2);
264                                      break;
265                          case FALSE: draw_rectan(35,3,48,7);
266                                      for(i=0 ; i<3 ; i++)
267                                      {   gotoxy(36,4+i);
268                                          printf("%s\n",menu3[i]);
269                                      }
270                                      change_attr(36,4,4,12);
271                                      Ready_3   = TRUE;
272                                      break;
273                         }
274                         break;
275          }
276     }
277
278     main()
279     {
280        int quit = FALSE;
281        int old_y,new_y,flag_2 = 0,flag_3 = 0;
282        start();
283        Line_1  = MAX_1; Col_win = Col_1;
284        Line_2  = MAX_1; Line_3  = MAX_1;
285        Ready_2 = FALSE; Ready_3 = FALSE;
286        while(!quit)
287        {
288             Select = getch();
289             switch(Select)
290             {
291               case UP:      switch(Col_win)
292                             { case Col_1: new_y = Line_1; Min = MI
    _1;
293                                           Length  = 13;
294                                           break;
295                               case Col_2: new_y = Line_2; Min = MI
    _2;
296                                           Length  = 16;
297                                           break;
298                               case Col_3: new_y = Line_3; Min = MI
    _3;
299                                           Length  = 12;
300                                           break;
301                             }
302                             old_y = new_y;
303                             new_y = old_y -.1;
304                             if(new_y >= MAX_1)
305                                change_attr(Col_win,old_y,new_y,Len
    th);
306                             else{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
307                                     if(new_y < MAX_1)
308                                     {
309                                        old_y = MAX_1;
310                                        new_y = Min;
311                                        change_attr(Col_win,old_y,new_
    ,Length);
312                                     }
313                                   }
314                                switch(Col_win){
315                                case Col_1:   Line_1 = new_y;
316                                              break;
317                                case Col_2:   Line_2 = new_y ;
318                                              break;
319                                case Col_3:   Line_3 = new_y ;
320                                              break;
321                                }
322                                break;
323                  case DOWN:    switch(Col_win)
324                                { case Col_1: new_y = Line_1; Min = MI
    _1;
325                                              Length  = 13;
326                                              break;
327                                  case Col_2: new_y = Line_2; Min = MI
    _2;
328                                              Length  = 16;
329                                              break;
330                                  case Col_3: new_y = Line_3; Min = MI
    _3;
331                                              Length  = 12;
332                                              break;
333                                }
334                                old_y = new_y;
335                                new_y = old_y + 1;
336                                if(new_y <= Min)
337                                  change_attr(Col_win,old_y,new_y,Len
    th);
338                                else{
339                                     if(new_y > Min)
340                                       {
341                                        old_y = Min;
342                                        new_y = MAX_1;
343                                        change_attr(Col_win,old_y,new_
    ,Length);
344                                       }
345                                   }
346                                switch(Col_win){
347                                case Col_1:   Line_1 = new_y;
348                                              break;
349                                case Col_2:   Line_2 = new_y ;
350                                              break;
351                                case Col_3:   Line_3 = new_y ;
352                                              break;
353                                }
354                                break;
355                   case LEFT:   switch(Col_win)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
356                                 {  case Col_1:  if(flag_3==0)
357                                                 {  Ready_3 = FALSE;f
    _3++;}
358                                                 else
359                                                    Ready_3 = TRUE;
360                                                 Left_Window_Key(Col_
    ine_3,Ready_3);
361                                                 Col_win = Col_3;
362                                                 break;
363                                    case Col_2:  Left_Window_Key(Col_
    ine_1,0);
364                                                 Col_win = Col_1;
365                                                 break;
366                                    case Col_3:  if(flag_2==0)
367                                                 {  Ready_2 = FALSE;f
    _2++;}
368                                                 else
369                                                    Ready_2 = TRUE;
370                                                 Left_Window_Key(Col_
    ine_2,Ready_2);
371                                                 Col_win = Col_2;
372                                                 break;
373                                 }
374                                 break;
375                 case RIGHT:     switch(Col_win)
376                                 { case Col_1: if(flag_2==0)
377                                                 {  Ready_2 = FALSE;fla
    ++;}
378                                                 else
379                                                    Ready_2 = TRUE;
380                                                 Right_Window_Key(Col_1
    ne_2,Ready_2);
381                                                 Col_win = Col_2;
382                                                 break;
383                                    case Col_2: if(flag_3==0)
384                                                 {  Ready_3 = FALSE;fla
    ++;}
385                                                 else
386                                                    Ready_3 = TRUE;
387                                                 Right_Window_Key(Col_2
    ne_3,Ready_3);
388                                                 Col_win = Col_3;
389                                                 break;
390                                      case Col_3: Right_Window_Key(Col_3
    ne_1,0);.
391                                                 Col_win = Col_1;
392                                                 break;
393                                 }
394                                 break;
395                     case 'q' :  quit = TRUE;
396
397                                 default:;
398             }
399         }
400         cursor_on();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
401         clrscr();
402     }
403
404
405
406
407
408
409
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้จะไม่สามารถสำเร็จลงได้หากขาดคำแนะนำจากคณะอาจารย์และรุ่นพี่ปริญญาโทห้องอาจารย์รัตติกร ซึ่งช่วยอำนวยความสะดวกทำให้ผู้จัดทำสามารถทำโครงงานนี้ไปได้จนสำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. สุริยัน ศรัสวัสดิ์กุล, "ระบบสื่อสารข้อมูลคอมพิวเตอร์"

2. ทินกร คุ้ก,ชูชัย ตั้งธนสาร "การอินเตอร์เฟส IBM PC "

3. ยืน ภู่วรวรรณ,"ไมโครคอมพิวเตอร์เนตเวอร์ค"

4."โทรสารสาธารณะ" เอกสารเผยแพร่การให้บริการถ่ายสำเนาทางไกลผ่านดาวเทียม การสื่อสารแห่งประเทศไทย พ.ศ. 2528

5. "IDAR SERVICE" เอกสารเผยแพร่การบริการติดต่อศูนย์รวบรวมข้อมูล และศูนย์ประมวลผลระหว่างประเทศ การสื่อสารแห่งประเทศไทย พ.ศ. 2528

6. Larry E. Jordan and Bruce Churchill "Communications and Networking", Prentice-Hall,1983.

7. D.W. Davics, D.L.A. Barber W.L. Price & C.M. Solomonides, "Computer Networks and their Protocols", John Wiley & Son, 1983.

8. John E. McNamara, "Technical Aspect of Data Communication" Digital Equipment Corp., 1977.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. Ken Kreehmer, V.22bis : The Modem Specification that's got "trouble", Data Communications, McGraw Hill Magazine, April 1985.

10. Andrew S. Tanenbaum, "Computer Networks", Prentice_Hall International, Inc. Englewood, N.J. 07632, 1981.

11. Wushow Chou-Editor "Computer Communications" Prentice-Hall, Inc. Englewood, N.J. 07632, 1983.

12. Harvey A. Freeman and Kenneth J. Thurber "tutorial Microcomputer Networks", IEEE Computer society, IEEE Catalog Number EH0190-9, 1981.

13. George E. Davis " The Local Network Handbook ", Data Communications, McGraw-Hill Publications Company, Newyork, 1982.

14. William Stilings "Local Networks Technology", IEEE Computer Society Press, 1989.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้