โครงข่ายการสื่อสารระหว่างเครื่องคอมพิวเตอร์

(Master Control Slave Network)

โดย

นางสาว นพรัตน์ เจริญพร้อมสกุล

นางสาว ปริญญา ไทยแท้

นางสาว สุภาภรณ์ โอวาทสุวรรณ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตร์บัณฑิต

สาขาวิศวกรรมอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2536

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

033363

ปริญญานิพนธ์ปีการศึกษา 2536

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โครงข่ายการสื่อสารระหว่างเครื่องคอมพิวเตอร์ (MCS - NETWORK)

ผู้จัดทำ

1. นางสาว นพรัตน์ เจริญพร้อมสกุล	33100160
2. นางสาว ปริญญา ไทยแท้	33100214
3. นางสาว สุภาภรณ์ โอวาทสุวรรณ	33100445

.....

(ดร. รัตติกร วรากูลศิริพันธุ์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงข่ายการสื่อสารระหว่างเครื่องคอมพิวเตอร์
(Master Control Slave Network)

ผู้ร่วมงาน
นางสาว นพรัตน์ เจริญพร้อมสกุล 33100160
นางสาว ปริญญา ไทยแท้ 33100214
นางสาว สุภาภรณ์ โอวาทสุสรรณ 33100445
อาจารย์ที่ปรึกษา
ดร.รัตติกร วรากูลศิริพันธุ์

บทคัดย่อ

การสื่อสารข้อมูลระหว่างคอมพิวเตอร์ในโครงงานนี้ใช้มาตรฐาน RS-232 ซึ่งเป็นลักษณะการส่งข้อมูลแบบอนุกรม โครงข่ายนี้สามารถส่งถ่ายข้อมูลได้ทั้ง 2 ทิศทาง กล่าวคือที่เครื่องคอมพิวเตอร์ของผู้ใช้แต่ละเครื่อง สามารถกำหนดให้เป็นทั้งผู้รับหรือผู้ส่งซึ่งเป็นลักษณะการส่งแบบทิศทางหนึ่ง ในเวลาหนึ่ง โครงข่ายนี้ประกอบไปด้วยส่วนฮาร์ดแวร์และซอร์ฟแวร์ ซึ่งส่วนซอร์ฟแวร์นั้นเป็นส่วนที่สำคัญ ประกอบไปด้วย โปรแกรมที่ควบคุมพอร์ตสื่อสาร การโต้ตอบระหว่างผู้ใช้งาน ส่วนฮาร์ดแวร์นั้นช่วยให้ระบบสามารถขยายขอบเขตการสื่อสารออกไปได้ ลักษณะการใช้งานจำลองมาจากการสื่อสารในห้องเรียน เพื่อโต้ตอบกันระหว่างผู้สอนและผู้เรียน ซึ่งการสื่อสารถูกควบคุมโดยคอมพิวเตอร์ของผู้สอน

Data communication between the PC Computer in this project is based on RS-232 Standard ,which is the characteristic of serial communication. This network can transfer data in both direction unsimultaneously ,"half duplex" in technical word. The network includes hardware and software ,the significant part of the network is software program that consists of serial port controlling and network protocol, according to hardware that could be further expanded. The applications imitate from the basic classroom communication ,controled by center computer (Master Control Slave Network).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 : บทนำ	1
บทที่ 2 : ทฤษฎีและหลักการ	2
2.1 ส่วนประกอบเบื้องต้นในการสื่อสาร	
2.2 วิธีการโอนถ่ายข้อมูล	
2.3 ทิศทางการส่งข้อมูล	
2.4 ข้อกำหนด	
2.5 RS - 232	
2.6 การทำงานของ 8255	
บทที่ 3 : รายละเอียดการสร้างเน็ตเวิร์ค	13
3.1 หลักการทำงานของฮาร์ดแวร์	
3.2 เน็ตเวิร์คโปรโตคอล	
3.3 หลักการเขียนเมนู	
บทที่ 4 : ขั้นตอนการทำงานและผลการทดลอง	32
บทที่ 5 : บทสรุป	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในโลกแห่งการสื่อสารปัจจุบัน การสื่อสารกันระหว่างคอมพิวเตอร์กับคอมพิวเตอร์ดูจะเป็นสิ่งที่ขาดไม่ได้ ธุรกิจส่วนใหญ่มักจะใช้คอมพิวเตอร์กับโมเด็มในการรับส่งข้อมูลรวม ทั้งการใช้คอมพิวเตอร์ในเน็ตเวอร์คเพื่อทำงานอีกหลายหน้าที่ ในเรื่องที่เกี่ยวกับการโอน
ย้ายข้อมูล ไม่ว่าจะเป็นระบบบูลเลทินบอร์ด (Bulletin Board System) บริการออน ไลน์ คอมมิวนิเคชั่น (On Line Communication)ซึ่งเหล่านี้เป็นเพียงส่วนหนึ่งในหลายๆแอปปลิเคชั่น (Application) ของคอมพิวเตอร์เน็ตเวอร์ค

ดังนั้น กลุ่มผู้จัดทำจึ้งมีความคิดที่จะสร้างระบบคอมพิวเตอร์เน็ตเวอร์คขึ้นมา โดยจำลองให้เป็นระบบที่ใช้ในห้องเรียน เรียกว่า"MASTER CONTROL SLAVE NETWORK"ซึ่งโครงงานนี้ได้เริ่มขึ้นตั้งแต่ในภาคการศึกษาที่ 1 โดยได้เริ่มวางรูปแบบและนำเอาทฤษฎีและหลักการที่ได้ศึกษามาเริ่มทดลอง จนสามารถเชื่อมต่อและรับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ 2 เครื่องได้ และในภาคการศึกษาที่ 2 นี้ได้กำหนดลักษณะของเน็ตเวอร์ค และค้นคว้าเพิ่มเติม เพื่อสร้างวงจรในส่วนฮาร์ดแวร์ และในส่วนของซอร์ฟแวร์ก็ได้เพิ่มเติมในส่วนของเมนู และแอปปลิเคชั่น

ในส่วนรายละเอียดของหลักการและรูปแบบการทำงานของเน็ตเวอร์คนั้นจะอยู่ในบทต่างๆของวิทยานิพนธ์นี้ แต่ในบทนี้จะขอกล่าวโดยสังเขป คือระบบเน็ตเวอร์คสำหรับโครงงานนี้จะมีตัวควบคุมการทำงานของระบบอยู่ตรงกลางหนึ่งตัว โดยใช้เครื่องคอมพิวเตอร์ พีซี 1 เครื่อง เรียกว่า "MASTER" และมีเครื่องคอมพิวเตอร์ปลายทางหลายเครื่องเรียกว่า "SLAVE" เครื่อง MASTER จะควบคุมการใช้ช่องสัญญาณโดยผ่านทางคาร์ดอินเทอร์เฟสสำหรับรายละเอียดของฮาร์ดแวร์ ซอร์ฟแวร์ และขั้นตอนของโปรโตคอลที่ใช้ในเน็ตเวอร์คนี้ จะถูกอธิบายอย่างละเอียดในวิทยานิพนธ์ฉบับนี้ โดยเนื้อหาทั้งหมดจะถูกแบ่งเป็นบทๆ ดังนี้

บทที่ 1 บทนำ
บทที่ 2 ทฤษฎีและหลักการ
บทที่ 3 รายละเอียดกาสร้าง
บทที่ 4 ขั้นตอนการทำงานและผลการทดลอง
บทที่ 5 บทสรุป
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 ส่วนประกอบเบื้องต้นในการสื่อสาร

ส่วนประกอบหลักเบื้องต้นในการสื่อสารข้อมูลแบ่งได้เป็น 3 ส่วน คือ

1) แหล่งกำเนิด

2) ตัวกลาง

3) ตัวรับ

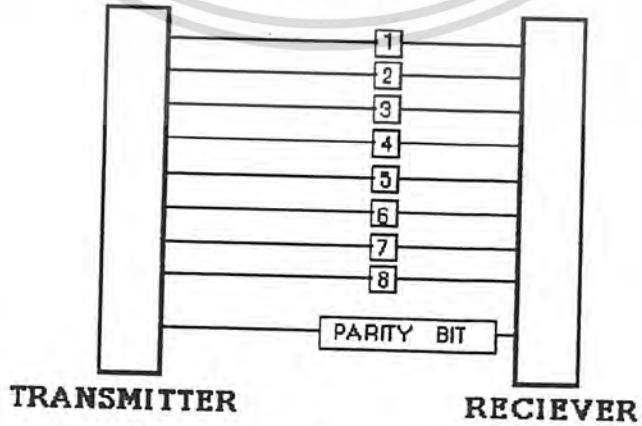
แสดงความสัมพันธ์ทั้ง 3 ส่วนดังรูป

รูปที่ 1 องค์ประกอบของการสื่อสาร

2.2 วิธีการโอนถ่ายข้อมูล

วิธีการรับและการส่งสัญญาณข้อมูลที่สื่อสารกันระหว่างเครื่อง แบ่งได้ดังนี้

1) การถ่ายโอนแบบขนาน ทำได้โดยการส่งข้อมูลออกมาพร้อมๆกันทุกบิทเพราะฉะนั้นช่องทางที่ข้อมูลใช้ในการเดินทาง จึงมีจำนวนเท่ากับจำนวนบิทที่รับส่ง การถ่ายโอนแบบนี้มักทำในระยะทางใกล้ๆ ทั้งนี้เพื่อป้องกันปัญหาเรื่องระดับทางกราวน์ไฟฟ้าระหว่างจุดส่งและจุดรับต่างกัน เนื่องจากความต้านทานสาย



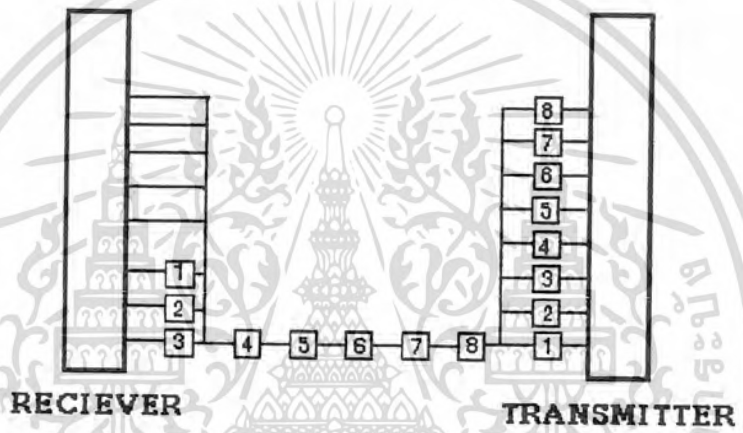
รูปที่ 2 ภาพแสดงการโอนถ่ายแบบขนาน

2) การถ่ายโอนแบบอนุกรม ทำได้โดยการส่งสัญญาณข้อมูลออกมาทีละบิทเรียงลำดับกันระหว่างจุดส่งและจุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รับ ถึงแม้จะทำได้ช้าแต่มีจุดเด่นคือประหยัดสายกว่า เพราะใช้สายเพียงคู่เดียวเท่านั้นการถ่ายโอนแบบนี้เหมาะสมกับการส่งในระยะไกลๆ เพราะเป็น วิธีประหยัดสายส่งได้เป็นจำนวนมาก การถ่ายโอนแบบนี้จำเป็นต้องได้รับการแปลงอย่างเหมาะสม เป็นสัญญาณแบบขนาน เมื่อมาถึงด้านรับ ซึ่งมีโปรโตคอล(PROTOCOL) หลักที่จัดการในเรื่องนี้ 2 ประเภทคือ

- ข้อกำหนดแบบซิงค์โครนัส(Synchronous protocol)
- ข้อกำหนดแบบอะซิงค์โครนัส(Asynchronous proyocol)



รูปที่3 ภาพแสดงการโอนถ่ายแบบอนุกรม

2.3 ทิศทางการส่งข้อมูล

ทิศทางการส่งข้อมูลจำแนกได้ 3 ลักษณะ คือ

1) การส่งแบบทิศทางเดียว (Simplex) การส่งแบบนี้ทิศทางการส่งจะคงที่ โดยกำหนดในเบื้องต้นว่า ฝ่ายใดเป็นฝ่ายส่งฝ่ายใดเป็นฝ่ายรับฝ่ายรับ

2) การส่งแบบฮาล์ฟดูเพล็กซ์ (Half duplex) เป็นการส่งในทิศทางใดทิศทางหนึ่งในเวลาหนึ่ง กล่าวคือทั้ง 2สถานีสามารถผลัดกันส่งได้ แต่จะส่งพร้อมกันไม่ได้ต้องผลัดกันรับผลัดกันส่ง

3) การส่งแบบฟลูดูเพล็กซ์ (Full duplex) เป็นการส่งในทิศทางใดก็ได้ในเวลาหนึ่งๆกล่าวคือทั้ง 2 สถานีสามารถรับส่งพร้อมกันได้ในเวลาเดียวกัน

2.4 ข้อกำหนด (PROTOCOL)

ข้อกำหนด(protocol) ในการสื่อสารข้อมูลนั้นมีความจำเป็นต้องกำหนดตกลงกันในเรื่องราวต่างๆสำหรับการควบคุมการไหลของข้อมูลในโครงข่ายเพื่อให้สามารถรับและส่งข้อมูลได้อย่างมีประสิทธิภาพและถูกต้อง

ISO (InternationalStandards Organization)ได้แบ่ง protocolออกเป็น 7 ระดับ ดังนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) Physical Layer หรือระดับ hardware เช่น RS-232C,RS-422,RS-485 เป็นต้น

2) Data Link Layer เป็นระดับที่ทำการเปลี่ยนแปลงการรับ-ส่ง ข้อมูลที่มีความไม่แน่นอนให้แน่นอนขึ้น โดยจัดรูปแบบเป็น block,frame พร้อมทั้งมีการตรวจสอบข้อผิดพลาด เช่น ตรวจสอบผลบวก(check_sum) CRC เพื่อให้รู้ว่าข้อมูลที่ได้รับมานั้นถูกต้องหรือไม่

3) Network Layer ทำการส่งข้อมูลเป็นแพกเกจ (package) เข้าไปในโครงข่าย packageอาจทำได้โดยอิสระโดยกำหนดที่อยู่ของผู้รับและผู้ส่ง วิธีนี้เรียกว่า data frame หรือ ส่งข้อมูลไปตามเส้นทางที่กำหนดเป็นลำดับทีละ package เรียกว่า virtual circuit

4) Transport Layer ทำหน้าที่รับผิดชอบในการประกันคุณภาพของการบริการภายในโครงข่าย หน้าที่สำคัญอีกอันหนึ่งคือควบคุมความมั่นคงในการรับ-ส่งข้อมูล ในระหว่างผู้รับและผู้ส่ง

ที่เหลืออีก 3 ชั้นทำหน้าที่ของโครงข่ายในระดับสูงได้แก่

5) Session Layer รับผิดชอบสำหรับปฏิกริยาที่มีความสำคัญเสมอกันระหว่างขบวนการสื่อสารที่ร่วมกัน 2 ขบวนการ

6) Presentation Layer ทำการแปลงรูปแบบ (format)หรือ code ที่จำเป็นสำหรับการส่งข่าสารไปยังปลายทาง

7) Application Layer ทำการสื่อสารโดยตรงกับโปรแกรมประยุกต์ ใช้งานของผู้ใช้ตามปกติแล้วนั้นงานที่เกี่ยวกับไมโครคอมพิวเตอร์มักเกี่ยวข้องกับ physical และ data link layer เท่านั้นในกำหนดprotocolจะต้องคำนึงถึงหน้าที่สำคัญ 2หน้าที่ดังนี้คือ

- การส่งสัญญาณตอบรับ (handshaking)เป็นการทำให้รู้ว่าวงจรการรับ-ส่งข้อมูลทำงานหรือไม่

-การกำหนดระเบียบการสื่อสาร (line discipline)เป็นการดำเนินการเกี่ยวกับการรับ-ส่งข้อมูลจริง,ขบวนการควบคุมข้อผิดพลาด (error control procedure),การส่งข้อมูลเป็น Block ,การจัดการข้อมูลอย่งถูกต้อง ซึ่งหน้าที่นี้ถูกเรียกกันว่า Protocol

ดังนั้นจึงขอกล่าวรายละเอียดเกี่ยวกับข้อกำหนดที่ใช้ในโครงงาน อันได้แก่ Asynchronous Communication Protocol หรือข้อกำหนดการสื่อสารแบบอะซิงโครนัส ในการสื่อสารแบบนี้ ข้อมูลที่จะส่งไปจะถูกรวมด้วยบิทเริ่มต้น (Start Bit) และบิทสุดท้าย (Stop Bit) ซึ่งใช้ทำให้ด้านรับสามารถซิงโครนัสกับด้านส่งได้ บิทข้อมูลต่างๆจะถูกส่งไปได้โดยการเปลี่ยนแปลงระดับแรงดัน (Logical State) ในสาย ความกว้างของแต่ละบิทจะเป็นตัวตัดสินความเร็วของการสื่อสาร ซึ่งวัดได้ 2 ลักษณะคือ

1) Data Rate (bit/sec) จำนวนที่สามารถส่งไปได้ใน 1 วินาที

2) Signal Rate (baud rate) จำนวนครั้งของการเปลี่ยนแปลงสัญญาณภายใน 1 วินาที

ทั้งนี้ทางด้านส่งและด้านรับจะต้องใช้อัตราเดียวกันในการรับส่งข้อมูล เมื่อไม่มีการสื่อสาร สายสื่อสารจะอยู่ในสภาวะ marking เมื่อเริ่มการสื่อสารจะส่งบิทเริ่มต้น ซึ่งมีสภาวะตรงข้ามคือ spacing ต่อจากนั้นจะทำการส่งข้อมูลบิทต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งทั้ง 2 ด้านจะต้องได้รับการกำหนดจำนวนบิทข้อมูลที่ใช้ในระบบการสื่อสาร ว่าเป็น 5,6,7 หรือ 8 บิทข้อมูล ในกรณีที่ให้มีการตรวจ parity bit ด้วย ทางด้านส่งจะส่งบิท parity ไปตามที่กำหนดโดยตามหลังบิทข้อมูลไป เพื่อเป็นการตรวจจับข้อผิดพลาดในขั้นหนึ่ง จากนั้นจะส่งบิทสุดท้ายออกไป ซึ่งอยู่ในสภาวะ marking การใช้บิทเริ่มต้นและบิทสุดท้าย เพื่อจะให้ทางด้านส่งและด้านรับสามารถส่งและรับได้ในเวลาเดียวกันตามลำดับในเวลาใดๆ ซึ่งก็คือ " การส่งแบบอะซิงโครนัส " นั่นเอง

รูปที่ 4 ภาพแสดงการส่งข้อมูลแบบอะซิงโครนัส

2.5 RS-232C

RS-232 เป็นโปรโตคอล(protocol) ในระดับฮาร์ดแวร์(hardware)ซึ่งมีลักษณะข้อกำหนดของสัญญาณตามมาตรฐานของ EIA มีชื่อเต็มดังนี้

RS-232C : interface between Data Terminal Equipment Data Communication Equipment Serial Binary Data Interchange

รหัส RS ย่อมาจากคำเต็มคือ Recommand Standard

232 เป็นหมายเลขของมาตรฐานนี้

C หมายถึงฉบับสุดท้ายของมาตรฐานนี้

ในตำแหน่งคาร์ดควบคุมการสื่อสารนั้นใช้หัวต่อชนิด DB-25P(male connector) ส่วนที่สายเคเบิ้ล(สายต่อระหว่างเครื่องPC)ใช้หัวต่อชนิดDB-25S(female connector)

มาตรฐาน RS - 232C นี้สามารถใช้ถ่ายโอนข้อมูลได้ด้วยอัตราเร็ว 0 ถึง 20,000 bps (110-9600 baud) โดยมีความยาวของสายสื่อสารไม่เกิน 50 ฟุต

สำหรับขาแต่ละขาและชื่อสัญญาณเพื่อใช้ในการเชื่อมต่อแสดงดังตาราง

Pin	Circuit	Description
1	AA	Protective Ground
2	BA	Transmit Data
3	BB	Received Data
4	CA	Request to Send
5	CB	Clear to Send
6	CC	Data Set Ready

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7	AB	Signal Ground
8	CF	Receive Line Signal Detector
9/10	-	(Reserved for Data Set Testing)
11	-	Unassigned
12	SCF	Secondary Received Line Signal Detector
13	SCB	Secondary Clear to Send
14	SBA	Secondary Transmitted Data
15	DB	Transmitted Signal Element Timing (DCE Source)
16	SBB	Secondary Received Data
17	DD	Received Signal Element Timing
18	-	Unassigned
19	SCA	Secondary Request to send
20	CD	Data Terminal Ready
21	CG	Signal Quality Dectector
22	CE	Ring Indicator
23	CH/CI	Data Signal Rate Select(DTE/DCE Source)
24	DA	Transmitted Signal Element Timing (DTE Source)
25	-	Unassigned

สำหรับ RS-232 นั้นจะไม่ใช้สัญญาณครบทุกๆขา โดยในคาร์ดควบคุมใช้การสื่อสารของ IBM PC กำหนดขาสัญญาณดังต่อไปนี้

Pin	Signal Name
2	Transmitted Data
3	Recieved Data
4	Request to Send
5	Clear to Send
6	Data Set Ready

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

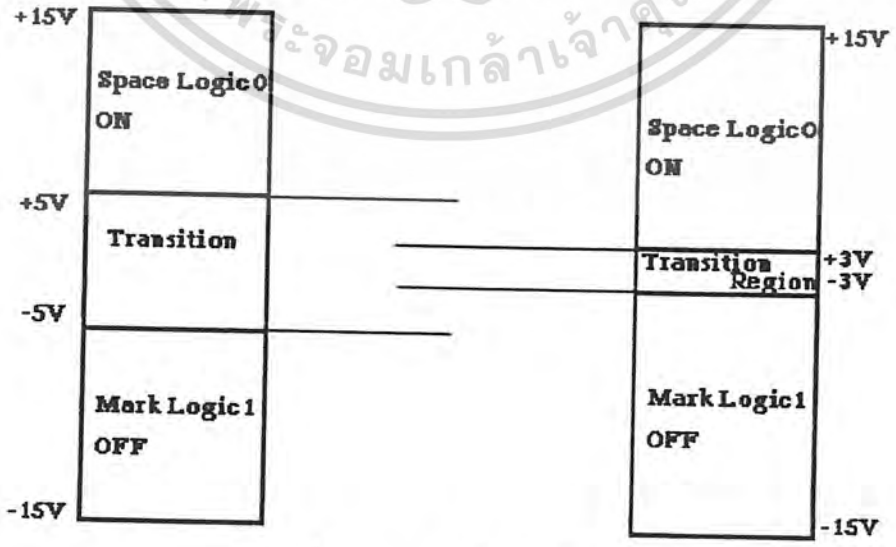
7	Signal Ground
8	Carrier Detect
20	Data Terminal Ready
22	Ring Indicator

ตารางที่ 2.1

RS-232 เป็นการเชื่อมต่อโดยแรงดัน ตามมาตรฐานของEIAใช้หัวแบบDB-25คาร์ดควบคุมจะทำการเปลี่ยนระดับสัญญาณจากระดับTTLไปเป็นระดับแรงดันตามข้อกำหนดของ EIAซึ่งsoftwareสามารถตรวจสอบเพื่อทราบสถานะของการเชื่อมต่อและอุปกรณ์รอบนอกได้

การเปลี่ยนแปลงระดับแรงดันสามารถอธิบายได้ตามรูปข้างล่าง จะถือสัญญาณเป็นmarking เมื่อแรงดันจากวงจรเปลี่ยนแปลงระดับแรงดัน(interchange circuit) วัดที่จุดต่อเป็นลบมากกว่า $-3V_{dc}$ เทียบกับระดับ ground และเป็น spacing เมื่อเป็นบวกมากกว่า $+3\ V_{dc}$ เทียบกับระดับ ground ขอบเขตระหว่าง $+3V_{dc}$ ถึง $-3V_{dc}$ จะถือว่าเป็น ทรานซิชั่นรีเจียน (transition region) และไม่มีความหมาย สำหรับแรงดันที่อยู่นอกเหนือค่าเหล่านี้ถือว่าไม่มีความหมายเช่นกัน

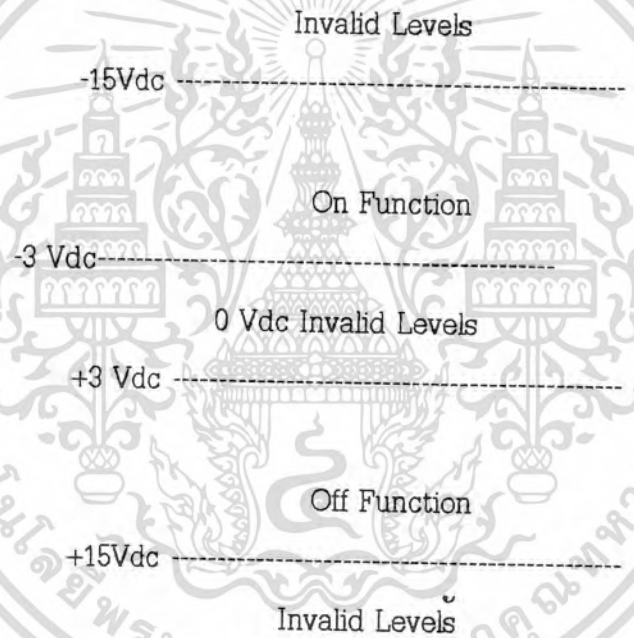
ในระหว่างการสื่อสารจะถือว่า marking คือ logic "1" และ spacingคือlogic "0" วงจรควบคุมการเชื่อมต่อจะ "on" เมื่อระดับแรงดันมากกว่า $+3V_{dc}$ และ "off" เมื่อระดับแรงดันเป็นลบมากกว่า $-3V_{dc}$ เทียบกับระดับกราวน์



รูปที่ 5 ภาพแสดงลักษณะสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดเกี่ยวกับระดับสัญญาณของคาร์ดควบคุมตามมาตรฐาน RS-232



Interface Voltage	Binary State	Signal Condition	Interface Contol Fn
Positive Voltage	Binary (0)	=Spacing	=ON
Negative Voltage	Binary (1)	=Marking	=OFF

ตารางที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 การทำงานของ 8255

เนื่องจากในโครงงานนี้มีการสร้างในส่วนของฮาร์ดแวร์ซึ่งเป็น คาร์ดอินเทอร์เฟส จึงจำเป็นต้องมีการใช้ไอซีเบอร์ 8255 ดังนั้นจะขอกล่าวรายละเอียดเกี่ยวกับการทำงานของไอซีเบอร์นี้

การทำงานของวงจร

Card Interface นี้ใช้ไอซี 8255A เป็นหลัก สายสัญญาณ Input และ Output ทั้ง 24 เส้น ของ 8255A มีระดับสัญญาณ TTL (Transistor Transistor Logic) จึงเป็นการง่ายในการใช้ 8255A Interface กับวงจร Logic อื่น ๆ รวมทั้ง CMOS ด้วย

ไอซี 74LS138 นำมาใช้ร่วมกับ 8255A เพื่อใช้ในการถอดรหัสตำแหน่งควบคุมให้การ์ด I/O ทำงาน การจัดขาของ 8255A และ 74LS138 แสดงดังรูปที่ 1

การทำงานหลักสามารถแบ่งออกได้เป็น 2 ส่วนใหญ่ ๆ คือ

1. ฮาร์ดแวร์ คือ ตัวการ์ด I/O ที่จะกล่าวถึง
2. ซอฟต์แวร์ คือ โปรแกรมที่จะมาควบคุมการทำงานของการ์ด

ส่วนประกอบของ Hardware นี้มีอุปกรณ์น้อย จึงง่ายต่อการตรวจซ่อม และการสร้าง วงจรนี้ใช้ไอซีเบอร์ 74LS138 ซึ่งเป็น Demultiplexer ใช้ในการถอดรหัส Address ของการ์ด I/O

Address ปกติของวงจรนี้จะถูกตั้งไว้ที่ 201H ซึ่งเป็น Address ของ Game Port ของ 8088, 80286, 80386 Address นี้เราสามารถตั้งใหม่ได้ โดยตำแหน่งที่สามารถตั้งได้จะมีค่าอยู่ในช่วง 201H ถึง 280H การตั้ง Address ใหม่นี้จะทำได้โดยการเลื่อนดิพสวิตช์ตามตำแหน่งต่าง ๆ ในตารางที่ 1

ตารางที่ 1 การตั้งตำแหน่ง Address ของ Card I/O

ตำแหน่ง SW on	ช่วง Address
S1	$XXD
S2	$XX3NC
S3	$XX9
S4	$XX7NC
S5	$XX5
S6	$XXBNC
S7	$XX1
S8	$XXFNC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

033362

ตำแหน่ง SW on	ช่วง Address
S9	$27X
S10	$26X
S11	$25X
S12	$24X
S13	$23X
S14	$22X
S15	$21X
S16	$20X

ตารางที่ 2.3

สมมติว่าเราต้องการตั้ง Address ใหม่ที่ตำแหน่ง 20DH ก่อนอื่นจะต้องเลื่อน Dip Switch ทุกตัวไปไว้ที่ตำแหน่ง "OFF" ก่อน จากนั้นจึงเลื่อน Dip Switch S1 และ S16 ไปอยู่ที่ตำปน่ง "ON" ก็จะได้ค่าของ Address ใหม่ตามที่ต้งการ สังเกต S2, S4, S6 และ S8 ค่า Address จะลงท้ายด้วย NC หมายถึงห้ามต่อกับวงจร

การควบคุมการ์ดนี้ใช้ IC1 และ IC2 เป็นตัวถอดรหัสการเลือกการ์ดจากเครื่อง PC เพื่อส่งสัญญาณ CS เลื่อชิพ 8255A ให้ทำงาน โดย CS จะ Active ที่ 0 ตามวงจรในรูปที่ 3 เครื่อง PC จะต้องส่ง Address มาที่ Address Bus A0 - AEN เป็นค่า 01000000XX IC1 และ IC2 ก็จะนำเอาค่านี้ไปถอดรหัสโดยค่า A0 และ A1 นี้ IC1 และ IC2 จะไม่นำไปใช้ แต่ 8255A จะนำไปใช้เอง เมื่อ CS แอกทีฟ "0" 8255A ก็จะทำงานโดยรับสัญญาณจาก A0 - A1 มาถอดรหัสเลื่อพิร์ตใช้งาน พร้อมกับการรับหรือส่งข้อมูลหรหว่างพอร์ดที่ถูกเลือกใช้กับบัสข้อมูลตามโหมดการทำงานที่ตั้งไว้ตอนเริ่มใช้งาน

ค่า Address ที่เครื่องคอมพิวเตอร์ส่งให้กับวงจรในรูปที่ 3 สามารถกำหนดได้ดังนี้

01000000000B หรือ 512 ในฐานสิบ เป็นการเลือกชิพ 8255A เพื่อให้ 8255A ควบคุมให้พอร์ต A ทำงาน

01000000001B หรือ 513 ในฐานสิบ เป็นการเลือกชิพ 8255A เพื่อให้ 8255A ควบคุมให้พอร์ต B ทำงาน

01000000010B หรือ 514 ในฐานสิบ เป็นการเลือกชิพ 8255A เพื่อให้ 8255A ควบคุมให้พอร์ต C ทำงาน

01000000011B หรือ 515 ในฐานสิบ เป็นการเลือกชิพ 8255A เพื่อให้ 8255A ควบคุม Register ควบคุม (Control Word Registers) รับข้อมูลการเซตโหมดการทำงานจากบัสข้อมูล (Data Bus) D0 - D7

Card I/O นี้จำเป็นต้องอาศัย Software ช่วยในการควบคุมการทำงานด้วย ซึ่งอาจเขียนได้จากหลาย ๆ ภาษาตามที่ถนัด สำหรับ Project นี้เราใช้ภาษา C ในการควบคุมการทำงานของ Card I/O นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของ 8255A

1. มี Input และ Output 24 ขาที่สามารถโปรแกรมได้
2. ระดับสัญญาณเดียวกันกับ TTL
3. คุณสมบัติเหมือนกับ Micro Processor ตระกูล Intel
4. สามารถเซตและรีเซตบิตได้โดยตรง

8255A สามารถโปรแกรมให้เป็น Input หรือ Output ก็ได้ สามารถกำหนดโหมดการทำงานได้ 3 โหมด คือ โหมด 0, โหมด 1 และ โหมด 2

โครงสร้างภายในของ 8255A

8255A มีโครงสร้างภายในดังแสดงในรูปที่ 11 การเรียกชื่อพอร์ตของ 8255A จะเรียกพอร์ตว่า พอร์ต A, B และ C โดยพอร์ต C แยกเป็น 2 ส่วนคือ PC0 - PC3 เรียกว่า พอร์ต C ล่าง และ PC4 - PC7 เรียกว่า พอร์ต C บน

จากรูปที่ 11 สามารถอธิบายหน้าที่การทำงานของแต่ละส่วนได้ดังต่อไปนี้

Data Bus Buffer

เป็น Buffer 8 บิต สองทิศทาง 3 สถานะ (3-State) ใช้ในการ Interface กับบัสข้อมูลของระบบ (System Data Bus) ข้อมูลจะถูกส่งหรือรับโดยผ่านทาง Buffer ขึ้นอยู่กับการประมวลผลของ CPU ว่าให้ทำการส่งหรือรับข้อมูล

Read/Write and Control Logic

ทำหน้าที่จัดการเกี่ยวกับการส่งผ่านข้อมูลและ Control หรือ Status ทั้งภายในและภายนอก ส่วนนี้จะรับอินพุท จาก Address Bus และ Control Bus ของ CPU แล้วนำการถอดรหัสในการควบคุมพอร์ต บลอ็กนี้มีสายสัญญาณควบคุมจากภายนอก ดังต่อไปนี้

Chip Select (CS) เมื่อขานี้ได้รับลอจิก "0" จะทำให้ 8255A ต่อเข้ากับระบบบัสของ CPU เพื่อให้ CPU อ่านหรือเขียนข้อมูลผ่านพอร์ตได้

Read (RD) เป็นขาสัญญาณอินพุตที่ต้องส่งเข้ามาจาก CPU เมื่อสัญญาณขานี้มีลอจิก "0" และ CS เป็น "0" ด้วย CPU จะทำการอ่านข้อมูลจากบัสข้อมูลของ 8255A

Write (WR) ขาสัญญาณการเขียนจะแอกทีฟเมื่อ CS เป็น "0" และ WR เป็น "0" โดยสัญญาณนี้จะถูกส่งมาจาก CPU ทำให้ CPU สามารถเขียนข้อมูลบนบัสข้อมูลของ 8255A ได้

Port Select 0 and Port Select 1 (A0 and A1) สัญญาณอินพุทที่สองขานี้จะต้องสัมพันธ์กับสัญญาณ RD, WR และ CS เพื่อใช้ในการเลือกพอร์ตใช้งานรายละเอียดดังตารางที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง รายละเอียดการเลือกใช้พอร์ต

A0	A1	RD	WR	CS	กระบวนการทางอินพุต (อ่าน)
0	0	0	1	0	PORT A - DATA BUS
0	1	0	1	0	PORT B - DATA BUS
1	0	0	1	0	PORT C - DATA BUS

กระบวนการทางเอาต์พุต (เขียน)

0	0	1	0	0	DATA BUS - PORT A
0	1	1	0	0	DATA BUS - PORT B
1	0	1	0	0	DATA BUS - PORT C
1	1	1	0	0	DATA BUS - CONTROL

ฟังก์ชันยกเลิกการทำงาน

X	X	X	X	1	DATA BUS - 3 STATE
1	1	0	1	0	ELEGA CONDITION
X	X	1	1	0	DATA BUS - 3 STATE

ตารางที่ 2.4

Reset (RESET) แอกทีฟ "1" สำหรับขานี้ เมื่อลอจิก "1" เข้ามาจะทำให้เกิดการเคลียร์รีจิสเตอร์ควบคุม (Control Register) และทุกพอร์ตจะถูกเซตสู่โหมดอินพุท

Group A and Group B Control แต่ละบล็อกจะรับคำสั่งจาก Read/Write Control Logic และรับ Control Word จากบัสข้อมูลภายใน เพื่อใช้ในการควบคุมพอร์ต

- Control Group A จะควบคุมพิร์ต A และ พอร์ต C บน (PC0 - PC3)
- Control Group B จะควบคุมพอร์ต B และพอร์ต C ล่าง (PC4 - PC7)

Ports A, B และ C 8255A ประกอบด้วยพอร์ต 8 ยิต 3 พอร์ต (พอร์ต A, B และ C) ซึ่งพอร์ตทั้งสามนี้สามารถกำหนดรูปแบบการใช้งานได้ โดยอาศัยซอฟต์แวร์ช่วยในการจัดการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

รายละเอียดการสร้างเน็ตเวอร์ค

ในบทนี้ จะอธิบายวิธีทดลองการทำงานของระบบเน็ตเวอร์ค ซึ่งถูกแบ่งออกเป็น 2 ส่วน คือส่วนที่เป็นวงจรฮาร์ดแวร์ และส่วนที่เป็นซอร์ฟแวร์หรือเน็ตเวอร์ค โปรโตคอล รวมทั้งส่วนของซอร์ฟแวร์ที่เขียนเป็นเมนู ซึ่งทั้ง 3 ส่วนจะต้องทำงานอย่างสอดคล้องกัน

3.1 หลักการทำงานของฮาร์ดแวร์

เนื่องจากการที่ใช้สายสัญญาณเพียง 3 เส้นคือ TX,RX,GNDดังนั้นจึงต้องมีการสวิทช์ช่องสัญญาณ โดยมีการทำ Hardware มาช่วยในการควบคุมและการสวิทช์ ซึ่ง Hardware นี้มีลักษณะเป็น Card Interface สามารถแบ่งการทำงานออกเป็น 2 ส่วน

-ส่วนควบคุมการใช้ช่องสัญญาณ

-ส่วนที่ทำหน้าที่สวิทช์

และในกรณีที่มีการ Polling จะต้องมีการใช้งานในส่วนเก็บค่าการขอใช้ช่องสัญญาณด้วย

ส่วนควบคุมการทำงาน

ส่วนนี้จะประกอบไปด้วย IC 8255 เป็นหลัก กำหนดแอดเดรสที่ 512 ถึง 515 โดยจะถูกตั้งให้ทำงานดังนี้คือ

PORT A :เป็น input port ทำหน้าที่รับข้อมูลจากส่วนเก็บค่าการขอใช้ช่องสัญญาณ เพื่อให้ Software ทางฝาย Master ทำการตรวจสอบว่า Slave ตัวใดขอใช้ช่องสัญญาณ
(ใช้ในกรณีที่ Master ทำการ polling)

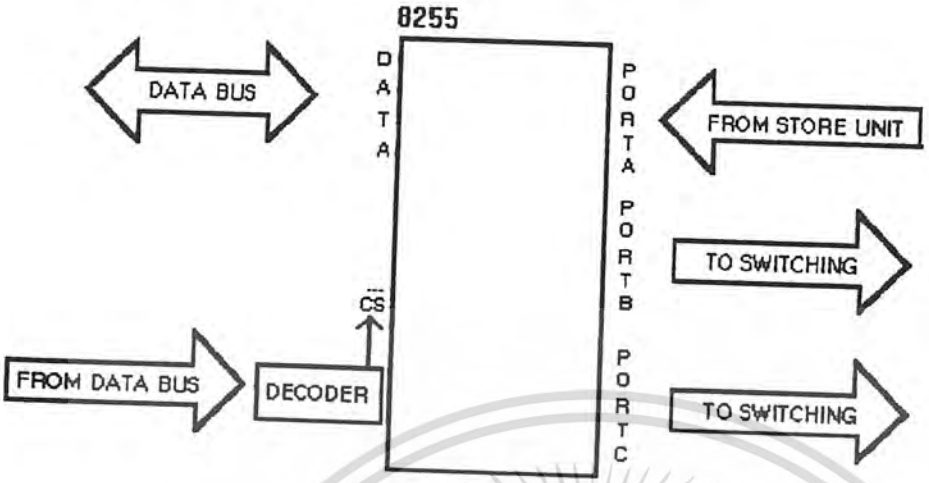
PORTB :เป็น output port ทำหน้าที่กำหนดการใช้ช่องสัญญาณที่ Master จะใช้ส่งข้อมูลไปยัง Slave

PORT C :เป็น output port ทำหน้าที่กำหนดการใช้ช่องสัญญาณที่ SLAVE จะส่งข้อมูลมายัง MASTER ซึ่ง port c นี้จะทำงานใน 2 กรณีคือ

-มีการ pollingโปรแกรมจะต้องทำการตรวจสอบว่าslave เครื่องใดต้องการใช้ช่องสัญญาณ แล้วจึงกำหนดค่าให้ที่ port c โดยทำงานร่วมกับ port a

-ไม่มีการ polling โปรแกรมจะทำการกำหนดที่ port c เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6 ภาพแสดงการทำงานของส่วนควบคุม

ส่วนสวิทช์

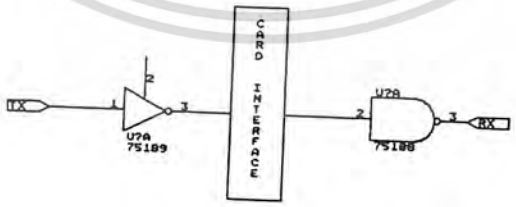
ในส่วนสวิทช์นี้มีการควบคุมการไหลของสัญญาณ 2 แบบคือ

-การไหลของข้อมูลจาก ขาTX ของฝาย MASTER ไปยัง ขาRX ของฝาย SLAVE โดยใช้ and gate ซึ่งจะทำการ and ระหว่างค่าที่ได้รับจาก port a กับขาTXของฝาย MASTER

-การไหลของข้อมูลจาก ขาTX ของฝายSLAVE ไปยังขาRX ของฝายMASTER โดยใช้ data selector (16 line to 1 line)

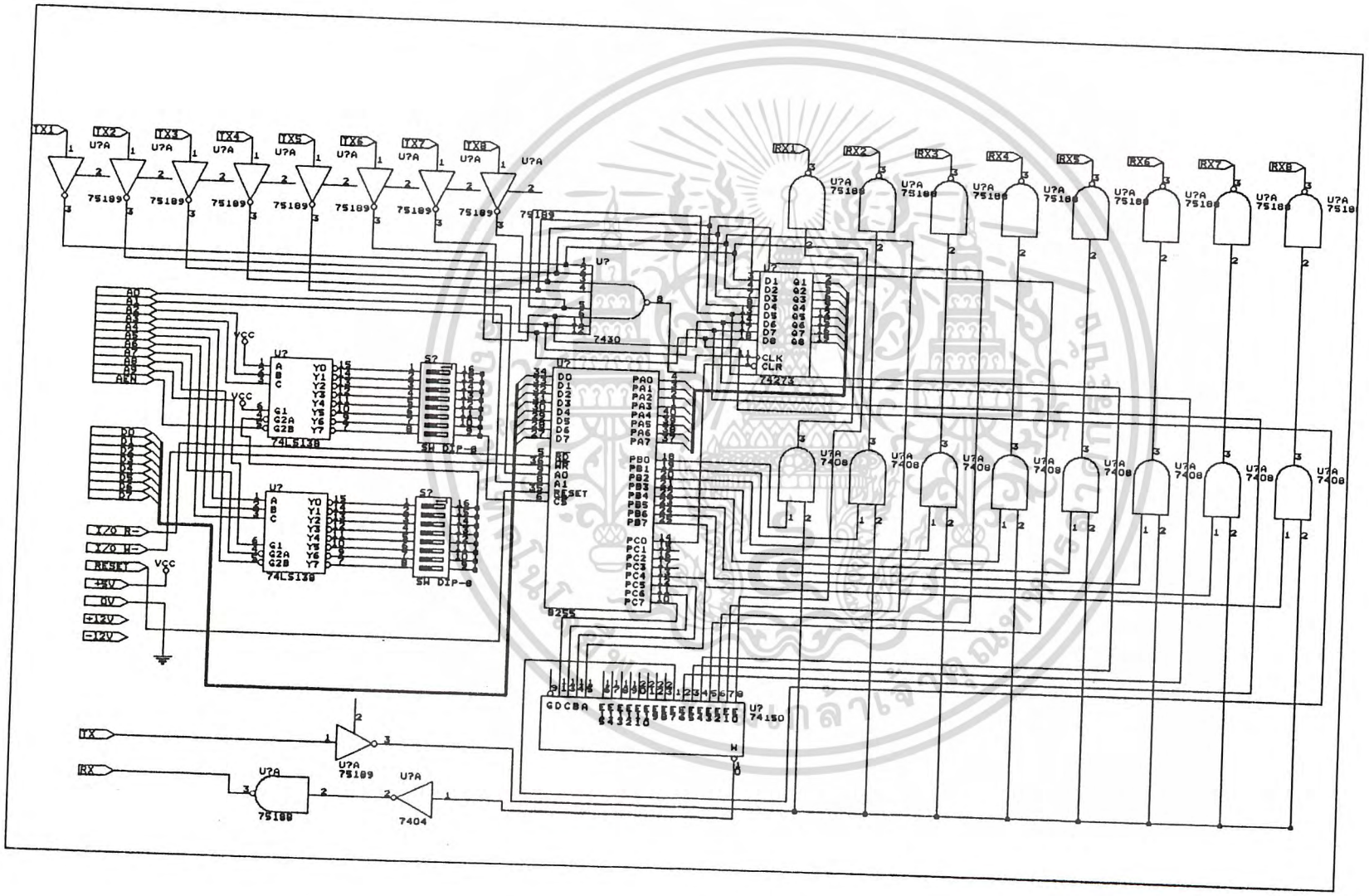
ส่วนรับส่งข้อมูล

สำหรับส่วนนี้เป็นส่วนที่จะต้องส่งข้อมูลจากขา TX ของคอมพิวเตอร์เครื่องหนึ่งไปยัง ขา RX ของเครื่องคอมพิวเตอร์อีกเครื่องหนึ่ง จึงต้องมีวงจร Driver ซึ่งใช้ ไอซีเบอร์ MC1488 และ MC1489



รูปที่7 ภาพแสดงส่วนรับส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8 ภาพแสดงวงจรส่วนฮาร์ดแวร์

3.2 เน็ตเวิร์คโปรโตคอล (Network Protocol)

เป็นส่วนของซอร์ฟแวร์ที่เขียนด้วยภาษาซี (แต่ก็มีส่วนเกี่ยวข้องกับฮาร์ดแวร์อยู่บ้าง) การทำงานอาจอธิบายเปรียบเทียบกับมาตรฐานการสื่อสารISOMODELได้ดังนี้คือ

1)Physical Layer

ในชั้นนี้เป็นชั้นเดียวที่เครื่องไอบีเอ็ม-พีซี มีอุปกรณ์ที่จะต้องใช้อยู่แล้ว คือ (UART) 8250 ซึ่งเป็นอุปกรณ์หลักของการทำงานในชั้นนี้ อนึ่งในโครงงานนี้ได้เพิ่มคาร์ดที่ทำหน้าที่สวิทช์การรับส่งข้อมูลระหว่าง MASTERและ SLAVE โดยเสียบคาร์ดที่ตัว MASTER และต่อสายรับ/ส่งข้อมูลจาก MASTER ไปยัง SLAVE โดยผ่านคาร์ดนี้

การสื่อสารในเน็ตเวิร์คนี้เป็นแบบ Asynchronous โดยใช้ฟังก์ชั่นของ Turbo C คือ Bioscom เป็นตัวกำหนด data bit,start bit,stop bit,parity bit และ baud rate

Outportb เป็นคำสั่งนำข้อมูลออกจากพอร์ตสื่อสาร

Inportb เป็นคำสั่งนำข้อมูลออกจากพอร์ตสื่อสาร

ทิศทางการสื่อสารเป็นแบบ Half Duplex นั่นคือในช่วงเวลาหนึ่งจะสื่อสารไปได้ในทิศทางเดียวเท่านั้น คือ จาก MASTER ไปยัง SLAVE หรือจาก SLAVE ไปยัง MASTER แต่เนื่องจากการสลับการทำงานจากฝ่ายรับไปเป็นฝ่ายส่ง ฝ่ายส่งไปเป็นฝ่ายรับ เกิดขึ้นอย่างรวดเร็ว จึงดูเหมือนกับว่าทั้ง 2 ฝ่ายสามารถรับส่งได้พร้อมๆกัน

2)Data Link และ Network Layer

ทำหน้าที่เชื่อมต่อสายระหว่าง MASTER กับ SLAVE ตามการทำงานในแต่ละแอปปลิเคชั่น และกำหนดเส้นทางการส่งข้อมูล (โดยการโปรแกรม 8255) มีขั้นตอนดังนี้คือ

-Detection Polling เฉพาะเมื่ออยู่ใน Broadcasting ทำได้โดยการเซ็ท 8255 ให้สามารถรับค่าสัญญาณซึ่งมาจาก Slave แต่ละตัว และตรวจสอบว่าเป็น SLAVE ตัวใดส่งมา

-ถ้าตกลงเครื่อง MASTER จะส่งสัญญาณตอบรับไปทันที และตัดการติดต่อจาก SLAVE เครื่องอื่น จนกว่าจะจบการติดต่อกับ SLAVE เครื่องดังกล่าว ถ้าไม่ตกลง MASTER ก็จะกลับสู่การทำงานปกติ คือรอรับข้อมูลจากคีย์บอร์ด

-ในกรณี Single Mode หรือการติดต่อเครื่องต่อเครื่อง MASTER จะเป็นผู้ตัดสินใจว่าจะติดต่อกับเครื่องใด และกำหนดเส้นทางการรับส่งข้อมูลเอง

3)Presentation Layer

คำสั่งการทำงานใน Layer เกี่ยวข้องกับการสร้างกรอบวินโดว์ และพูลดาวน์เมนู สำหรับแอปปลิเคชั่นต่างๆ ในโครงงาน เพื่อให้สะดวกต่อการใช้งาน ดังนั้นจึงต้องสร้างตำแหน่งในการจัดการวินโดว์ที่สร้างขึ้น การใช้คำสั่งอาจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบ่งเป็นกลุ่มได้ดังนี้

-คำสั่งสำหรับการสร้างกรอบวินโดว์ และพูลดาวน์เมนู โดยสามารถกำหนดขนาด,หมายเลขของกรอบ และฟังก์ชั่นคีย์ที่ใช้ได้กับพูลดาวน์เมนู

-คำสั่งที่ใช้เก็บหน้าจอปัจจุบันไปไว้ในหน่วยความจำ และคำสั่งเรียกกรอบหน้าจอกลับคืน

-คำสั่งสำหรับการปิดกรอบหน้าจอ และการ clear หน้าจอ

-คำสั่งเกี่ยวกับการรับ Standard Input Device เช่นจากแป้นพิมพ์ ,และคำสั่งสำหรับการจัดหน้าจอต่างๆ เช่นการเลื่อนตำแหน่ง Cursur,การเขียนข้อมูลที่ได้รับลงบนกรอบหน้าจอที่ต้องการ,การเลื่อนบรรทัดข้อมูล

คำสั่งต่างๆ เหล่านี้มีพื้นฐารการทำงานเดียวกับคำสั่งที่ใช้ใน Turbo C เพียงแต่ต้องอ้างอิงกับลำดับที่ของกรอบหน้าจอที่ต้องการใช้งานรวมทั้งขนาดของหน้าจอนั้นด้วย

4)Application Layer

ในโครงงานนี้ได้กำหนดรูปแบบการใช้งานสำหรับ User ไว้ 3 แบบได้แก่

1.Broadcasting Routine

2.Chat Mode Routine

3.File Transfer Routine

เมื่อเริ่มต้นเข้าสู่โปรแกรมหน้าจอของ MASTERจะปรากฏ Main Menu ให้เลือกว่าต้องการสื่อสารข้อมูลแบบใด ส่วนหน้าจอของ SLAVE จะเข้าสู่ Broadcasting ทันที

-Broadcasting Routine

หลักการคล้ายๆกับห้องโสตฯ กล่าวคือทางฝ่าย MASTER สามารถส่งข้อมูลที่ User กดบนแป้นพิมพ์และปรากฏบนหน้าจอของตนไปยังหน้าจอของ SLAVE ที่ต้องการได้หลายเครื่องในเวลาเดียวกัน การส่งข้อมูลจะเป็นแบบ Simplex คือส่งทิศทางเดียวจาก MASTERไปยังSLAVE เท่านั้น และจะส่งอย่างต่อเนื่องจนกว่าจะเกิอเหตุการณ์ดังต่อไปนี้เกิดขึ้น

:MASTER ต้องการออกจาก Broadcasting กลับสู่ Main Menu

:ระหว่างการส่งข้อมูลแบบ Broadcast ถ้า MASTER ทำการ polling slave request ถ้าพบว่ามีการติดต่อจาก SLAVE เครื่องใดเครื่องหนึ่ง และ MASTER อนุญาติให้ทำการติดต่อได้ ทั้งหน้าจอของฝ่าย SLAVE และ MASTER จะเข้าสู่การทำงานในรูปแบบที่ต้องการทันที

สำหรับ 2 รูปแบบที่เหลือ ถ้า MASTER เป็นผู้เรียกใช้งาน จะต้องออกจากหน้าจอ Broadcast เสียก่อน (ยกเลิกการทำงาน) แล้วไปเลือกจาก Main Menu แต่ถ้า SLAVE เป็นผู้เลือกใช้งาน ชั้น Data Link จะส่งสัญญาณ ENQไปยัง MASTER แล้วรอการตอบรับจาก MASTER โดยระหว่างนั้นการทำงานอยู่ใน Broadcast

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-Chat Mode Routine

เป็นการสื่อสารแบบ On Line คือทั้ง 2 ฝ่ายสามารถส่งรับข้อมูลโต้ตอบกันได้ทันที จนกว่าฝ่ายใดฝ่ายหนึ่งจะกด ESC ถือเป็นการยกเลิกการทำงาน

:ถ้า MASTER เป็นผู้เลือกใช้งาน ชั้น Data LInk จะทำการเชื่อมต่อสายส่งไปยัง SLAVE ที่ต้องการเพียงเครื่องเดียว เมื่อเชื่อมต่อได้แล้วก็จะเรียกหน้าจอสำหรับ On Line Communication ที่มี 2 หน้าต่างสำหรับข้อมูลที่ส่งออกไปและข้อมูลที่รับเข้ามาจากพอร์ต จากนั้นจึงส่งข้อมูลโต้ตอบกันโดยใช้แป้นพิมพ์

:ถ้า SLAVE เป็นผู้เรียกใช้งาน ชั้น Data Link จะส่งสัญญาณ ENQ ไปยัง MASTER แล้วตามด้วย CHATRQ จากนั้นก็รอรับสัญญาณตอบรับ CHATCONF จาก MASTER ระหว่างที่คอยนั้น SLAVE ก็ยังคงอยู่ใน Broadcast Routine คือรับข้อมูลจากพอร์ตอนุกรมมาแสดงบนหน้าจอไปเรื่อยๆ เมื่อได้รับสัญญาณตอบรับ CHATCONF ก็จะเข้าสู่หน้าจอ On line

เมื่อยกเลิกการทำงาน MASTER กลับสู่ Main Menu ส่วน SLAVE จะกลับสู่ Broadcast Window

-File Transfer Routine

ใช้สำหรับการส่งข้อมูลประเภท text file อีกฝ่ายหนึ่ง การเรียกใช้งานไม่ว่าฝ่ายใดเป็นผู้เรียกใช้จะต้องผ่านการใส่ชื่อ file ที่ต้องการส่งเสียก่อน จากนั้นเมื่อตรวจสอบแล้วว่าสามารถอ่านข้อมูลจาก file ได้ จึงจะมีการติดต่อกับพอร์ตข้อมูล เพื่อเชื่อมต่อพอร์ตข้อมูลระหว่างเครื่องต่อเครื่อง

:การส่ง file และรับ file เริ่มจากการส่งสัญญาณ SYN ไปยังฝ่ายรับ เมื่อได้รับสัญญาณ SYN ตอบกลับมาก็จะเริ่มส่งชื่อและขนาดของ file ไปยังฝ่ายรับ ฝ่ายรับจะเปิด file ตามขนาดและชื่อที่ได้รับ จากนั้นจึงทำการส่งและรับข้อมูลใน file จนเสร็จ

:สำหรับการเรียกใช้หรือการขอใช้ Routine นี้ ใช้สัญญาณ FILERQ และ FILECONF โดยใช้วิธีเดียวกับ Chat Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 หลักการเขียนเมนู

ในการติดต่อระหว่าง Master และ Slave นั้นมีโหมดการทำงานมากกว่า 1โหมดดังนั้นเพื่อความสะดวกในการใช้งาน จึงได้จัดทำเมนูเพื่อให้เข้าสู่การทำงานอื่นๆได้ง่ายขึ้น นอกจากนี้ยังจัดส่วนการทำงานให้อยู่ในแต่ละกรอบ WINDOW เพื่อสะดวกต่อการเก็บหน้าจอ คืนหน้าจอเดิม เป็นต้น เมนูในโครงงานนี้ทำงานใน TEXT MODE เราสามารถกำหนดสีตัวอักษร และสีหน้าจอ (Black ground) ตามค่าแอทตริบิวต์ (Attribute) ของตัวอักษร ซึ่งวิธีการกำหนดค่าแอทตริบิวต์ จะอยู่ในหัวข้อถัดไป

ข้อแตกต่างระหว่างเมนู POPUP PULLDOWN กับเมนูแบบธรรมดา คือ เมนูธรรมดาจะหยุดการทำงานของโปรแกรมขณะนั้น เมื่อมีการเรียกใช้เมนู แต่เมนูแบบ POPUP PULLDOWN จะหยุดการทำงานของโปรแกรมชั่วคราว ซึ่งทำให้ผู้ใช้เกิดความรู้สึกที่ดีกว่า ข้อแตกต่างระหว่าง POPUP และ PULLDOWN คือPOPUP ในแต่ละตัว จะไม่มีการเลือกเมนูย่อย (Sub menu) แต่แบบ PULLDOWN จะสามารถมีเมนูย่อยในตัวเลือกได้

<u>อะแดปเตอร์การแสดงผล</u>

การเขียนโปรแกรมเมนูใน TEXT MODE นี้ ใช้การเขียนควบคุมจอภาพโดยตรง ซึ่งอะแดปเตอร์การแสดงผลมีอยู่ 4 แบบ คือ โมโนโครม, CGE, EGA และ VGA ขนาดจอภาพที่ใช้อยู่โดยทั่วๆไปจะเป็นค่าเทียบกับตารางคือ โหมด 2,3,7 (มีขนาด 80x25) และไม่ว่าจะเป็นการแสดงผลแบบใดที่มุมซ้ายสุดจะเป็นตำแหน่ง(0,0)เสมอ

โหมด	แบบ	ขนาด	อะแดปเตอร์
0	text, b/w	40×25	CGA, EGA
1	text, 16 colors	40×25	CGA, EGA
2	text, b/w	80×25	CGA, EGA
3	text, 16 colors	80×25	CGA, EGA
4	graphics, 4 colors	320×200	CGA, EGA
5	graphics, 4 grey tones	320×200	CGA, EGA
6	graphics, b/w	640×200	CGA, EGA
7	text, b/w	80×25	monochrome
8	graphics, 16 colors	160×200	PCjr
9	graphics, 16 colors	320×200	PCjr
10	graphics, 4 colors PCjr, 16 colors EGA	640×200	PCjr, EGA
13	graphics, 16 colors	320×200	EGA
14	graphics, 16 colors	640×200	EGA
15	graphics, 4 colors	640×350	EGA

ตารางที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยปกติหน่วยความจำ (memory)แบบRAMของคอมพิวเตอร์จะถูกสงวนไว้สำหรับเก็บลักษณะตัวอักษรที่แสดงบนจอไว้ส่วนหนึ่ง หน่วยความจำนี้ในจอโมโนโครมจะเริ่มที่แอดเดรส B0000000H ส่วนจอแบบอื่นจะเริ่มที่แอดเดรส B8000000H

การแสดงผลของตัวอักษรในโหมด 80x25 นั้น อักษรแต่ละตัวจะต้องใช้หน่วยความจำจำนวน 2 ไบต์ โดยไบต์แรกเก็บรหัสแอสกี(ASCII)ของตัวอักษร ไบต์ที่ 2 เก็บแอตทริบิวต์
(Attribute)ของตัวอักษร สำหรับค่าแอทตริบิวต์ มีหลักการกำหนดดังนี้

เช่น 0xAB

เมื่อ A คือค่าสีของBlackground

B คือค่าสีของตัวอักษร

A และ B เป็นไปดังตารางที่ 3.2

หน่วยความจำที่ถูกจองไว้สำหรับการแสดงผล โดยปกติจะมากกว่าที่ใช้ในการแสดงผลแบบ 80x25 เพราะว่า เพื่อนำไปใช้ในการแสดงผลในโหมดกราฟฟิก และอีกเหตุผลหนึ่งคือ สามารถเก็บการแสดงผลในโหมดกราฟฟิกและอีกเหตุผลหนึ่งคือ สามารถเก็บการแสดงผลไว้ในหน่วยความจำ และสามารถเลือกหน้าจอไหนมาแสดงได้ เรียกหน่วยความจำของแต่ละหน้าจอว่า video page ซึ่งโดยทั่วไปดอสใช้ page 0 ในการแสดงผล

วิธีการติดต่อกับส่วนแสดงผลทำได้ 3 วิธี คือ

<u>วิธีแรก</u> ติดต่อกับ function call ของดอส วิธีนี้ถ้านำมาเขียน POPUP PULLDOWN โปรแกรมจะทำงานช้ามาก

<u>วิธีที่สอง</u> ทำงานโดยผ่าน function call ของ BIOS ซึ่งทำงานได้เร็วพอสมควร

<u>วิธีที่สาม</u> เขียนอ่านข้อมูลลงหน่วยความจำสำหรับการแสดงผลโดยตรง วิธีนี้การทำงานของโปรแกรมจะเร็วที่สุด

<u>การเก็บหน้าจอและการคืนหน้าจอ</u>

การเก็บตัวอักษรที่ปรากฏอยู่บนที่ตำแหน่งใดๆบนจอภาพ ทำได้โดยการอ่านตัวอักษรเข้ามาเก็บไว้โดยใช้อินเทอร์รัพ 16 ฟังก์ชันหมายเลข 8 ซึ่งเมื่อเรียกใช้จะส่งรหัสแอสกี และ แอทตริบิวต์ของตัวอักษร ณ ตำแหน่งที่เคอร์เซอร์ปรากฏอยู่ และเนื่องจากเมนูที่ใช้ในโครงงานเป็นลักษณะ PULLDOWN ดังนั้นเมื่อมีการเลือกเมนูแล้วอาจจะยังไม่คืนหน้าจอ(เพื่อว่าอาจมีการแสดงเมนูย่อยขึ้นมาอีก) การคืนหน้าจอนั้นอยู่ในส่วนอื่นของโปรแกรม(ไม่ได้อยู่ในฟังก์ชันเดียวกัน) ดังนั้นก่อนที่จะเขียนโปรแกรมควรคิดลำดับของเมนูก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Value	Text color	Blackground	SHOW
0	Blank	Blank	
1	Blue	Blue	
2	Green	Green	
3	Cyan	Cyan	
4	Red	Red	
5	Magenta	Magenta	
6	Brown	Brown	
7	Lightgray		
8	Drakgray		
9	Lightblue		
10	Lightgreen		
11	Lightcyan		
12	Lightred		
13	L.magen		
14	Yellow		
15	White		

ตารางที่ 3_2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Save a portion of the screen. */
void save_video(int num)
{
  register int i,j;
  char *buf_ptr;
  char far *v, far *t;

  buf_ptr = frame[num].p;
  v = vid_mem;
  for(i=frame[num].startx; i<frame[num].endx+1; i++)
    for(j=frame[num].starty; j<frame[num].endy+1; j++) {
      t = (v + (j*160) + i*2);
      *buf_ptr++ = *t++;
      *buf_ptr++ = *t;
      *(t-1) = ' ';  /* clear the window */
    }
}

/* Restore a portion of the screen. */
void restore_video(int num)
{
  register int i,j;
  char far *v, far *t;
  char *buf_ptr;

  buf_ptr = frame[num].p;
  v = vid_mem;
  t = v;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for(i=frame[num].startx; i<frame[num].endx+1; i++)
      for(j=frame[num].starty; j<frame[num].endy+1; j++) {
        v = t;
        v += (j*160) + i*2; /* compute the address */
        *v++ = *buf_ptr++;  /* write the character */
        *v = *buf_ptr++;    /* write the attribute */
    }
    frame[num].active = 0; /* deactivate */
    /*free(frame[num].p);*/
}
```

<u>วิธีการกำหนดกรอบเมนู</u> เราควรสร้างกรอบเมนูให้อยู่ในอาเรย์ของตัวแปรที่เก็บข้อมูลทั้งหมดที่มีความสัมพันธ์แต่ละเมนู โครงสร้างเมนูเป็นดังนี้

```
  struct menu_frame {
  int startx, endx, starty, endy;
  int curx,cury;
  unsigned char *p;
  char **menu;
  char *keys;
  char *header;
  int border, count;
  int active;
} frame[MAX_FRAME];
```

จากโครงสร้างนี้ MAX_FRAME เป็นค่าคงที่ที่เก็บจำนวนเมนูที่สามารถมีได้มากที่สุดในโปรแกรม ตัวแปร active ใช้ในการบอกสถานะว่าขณะนั้นเมนูปรากฏอยู่บนหน้าจอหรือไม่

ในโครงงานนี้นอกจากจะเป็น PULLDOWN เมนูแล้ว แต่ละ Application ยังอยู่ในแต่ละส่วนของวินโดว์ต่างๆ (POPUP) วินโดว์

<u>POPUP วินโดว์</u>

POPUP วินโดว์ คือส่วนของหน้าจอที่ใช้สำหรับทำงานอื่นที่ไม่ใช่งานบนหน้าจอปัจจุบัน POPUP วินโดว์จะปรากฏขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำ ไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาเสมือนกับทับสิ่งที่ปรากฏอยู่เดิม แต่หลักการจริงๆ คือจะมีการนำเอาข้อความเดิมที่ถูกทับไปเก็บไว้ในหน่วยความจำ แล้วแสดงวินโดว์นั้นออกมา เมื่อใช้งานวินโดว์นั้นเสร็จแล้ว ก็นำเอาข้อความเดิมกลับมาแสดงอีกครั้ง เพื่อทำงานต่อไป ทำให้สามารถใช้งานวินโดว์ได้หลายๆวินโดว์พร้อมๆกัน

POPUP วินโดว์ประกอบด้วยส่วนสำคัญ คือ

1) ส่วนของโปรแกรม Application

2) ส่วนจัดการเกี่ยวกับอินพุตและเอ้าท์พุตและกรอบของวินโดว์

จะเห็นว่าโปรแกรม Application ที่ใช้วินโดว์นั้น จะต้องไม่มีการเขียนตัวอักษรออกมานอกขอบเขตของวินโดว์ เนื่องจากขนาดของวินโดว์สามารถเปลี่ยนแปลงได้โดยไม่เกี่ยวข้องกับตัวโปรแกรม Application ที่ใช้งานบนวินโดว์ จึงเป็นหน้าที่ของส่วนที่จัดการวินโดว์เองที่จะป้องกันการเขียนเกินขอบเขตของวินโดว์ I/O และรูทีนเกี่ยวกับจอภาพของภาษาซี เช่น printf() หรือ getch() ไม่สามารถนำมาใช้ได้จำเป็นต้องเขียนฟังก์ชันที่จะใช้กับ I/O ของวินโดว์โดยเฉพาะขึ้นมา

ทฤษฎีการทำงานของวินโดว์คือ Application แต่ละอย่างจะมีวินโดว์ของตนเองแยกจากกัน เมื่อต้องการใช้งานโปรแกรมใด ก็ให้วินโดว์ของโปรแกรมนั้นเป็นวินโดว์ใช้งาน
คือจะทำงานได้กับวินโดว์นั้นเท่านั้น วินโดว์อื่นจะหยุดหมด เมื่อทำงานเสร็จแล้ววินโดว์นั้นจะหายไป ถ้ามีการเรียกวินโดว์ซ้อนกัน วินโดว์เดิมจะหยุดแล้วไปทำงานกับวินโดว์ใหม่ แต่วินโดว์เดิมจะไม่หายไป ซึ่งปกติแล้วถ้าไม่ใช้งานลักษณะของวินโดว์ จอภาพทั้งจอจะถูกลบไปหมดแต่ถ้าเป็นวินโดว์แล้ว งานเดิมก็เหมือนถูกหยุดไว้ชั่วคราวเท่านั้น

<u>ฟังก์ชันอินพุต/เอ้าท์พุตของวินโดว์</u>

ฟังก์ชัน **window_xy(num,x,y)**

```
int num,x,y;
{
 if(x<0|| x+frame[num].startx>=frame[num].endx-1)
 return 0;
 if(y<0 || y+frame[num].starty>=frame[num].endy-1)
 return 0;
 frame[num].curx = x;
 frame[num].cury = y;
 goto_xy(x,y);
 return 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
```

หมายเหตุ ตำแหน่งของเคอร์เซอร์ที่กำหนดในวินโดว์ใดวินโดว์หนึ่งจะอยู่ที่ตำแหน่งเดิมเสมอเมื่อเทียบกับวินโดว์ เช่น กำหนดตำแหน่ง 2,2 เคอร์เซอร์จะปรากฏที่ตำแหน่ง 2 แถวจากขอบบนของวินโดว์ และ 2 ตัวอักษรถัดจากขอบซ้ายของวินโดว์

ฟังก์ชัน **window_getche(num)**

```
int num;
{
  union inkey {
  char ch[2];
  int i;
}c;
  if (!frame[num].active) return 0;
  window_xy(num,frame[num].curx,frame[num].cury);
  c.i = bioskey(0);
  return c.i;
}
```

หมายเหตุ ฟังก์ชัน window_getche() ต่างจากฟังก์ชัน getch() คือฟังก์ชันนี้จะส่งค่ากลับเป็น scan code 16 บิต คือ รหัสตัวอักษรใน 8 บิตล่างและรหัสตำแหน่งใน 8 บิตบน

ฟังก์ชั่น **void window_gets(num,s)**

```
int num;
char *s;
{
  char ch,*temp;
  temp=s;
  for(;;)
{
  ch = window_getche(num);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        switch(ch)
      {
        case '\r':
          *s ='\0';
          return ;
        case BKSP :
          if(s>temp)
        {
            s--;
            frame[num].curx--;
            if(frame[num].curx<0) frame[num].curx = 0;
            window_xy(num,frame[num].curx,frame[num].cury);
            write_char(frame[num].startx+frame[num].curx+1,
            frame[num].starty+frame[num].cury+1,
            ' ',0x1B);
        }
        break;
        default :
            *s = ch;
            s++;
            break;
      }
    }
  }
```

ฟังก์ชัน **window_putchar(num,ch)**

```
int num;
char ch ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
   register int x,y;
   char far *v;
   if(!frame[num].active) return 0;
   x = frame[num].curx + frame[num].startx +1;
   y = frame[num].cury + frame[num].starty +1;
   v = vid_mem;
   v += (y*160) + (x*2);
   if((ch == '\r') || (x >= frame[num].endx))
   {
    if((y < frame[num].endy-1))
    {
     y++;
     x = frame[num].starty+1;
     v = vid_mem;
     v += (y*160)+ x*2;
     frame[num].cury++;
     frame[num].curx =0;
    }
    else
    {
    frame[num].curx = frame[num].startx;
    frame[num].cury = frame[num].starty;
    window_xy(frame[num].curx,frame[num].cury);
    }
}
 else
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        frame[num].curx++;
        *v++ = ch;
        *v++ = 0x1B;
    }
        window_xy(num,frame[num].curx,frame[num].cury);
        return 1;
}
```

ฟังก์ชัน **window_puts(num,str)**

```
    int num;
    char *str;
    {
     if(!frame[num].active) return 0;
     for(; *str;str++)
     window_putchar(num,*str);
     return 1;
    }
```

ฟังก์ชันแก้ยวกับจอภาพอื่นๆ

ฟังก์ชันเหล่านี้ได้แก่

ฟังก์ชัน	การทำงาน
window_cls()	ลบข้อความในหน้าต่าง
window_upline()	เลื่อนเคอร์เซอร์ขึ้น 1 บรรทัด
window_downline()	เลื่อนเคอร์เซอร์ลง 1 บรรทัด
window_bksp()	เลื่อนเคอร์เซอร์ย้อนหลัง 1 ตำแหน่ง
window_cleol()	ลบข้อความจนสิ้นสุดบรรทัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void window_cls(num)
int num;
{
   register int i,j;
   char far *v,far *t;

   v = vid_mem;
   t = v;
   for(i=frame[num].starty+1; i<frame[num].endy;i++)
     for(j=frame[num].startx+1;j<frame[num].endx;j++)
     {
       v = t;
       v +=(i*160) +j*2;
       *v++ = ' ';
       *v = 0x10;
       }
   frame[num].curx = 0;
   frame[num].cury = 0;
}
void window_cleol(num)
int num;
{
  register int i,x,y;
  x = frame[num].curx;
  y = frame[num].cury;
  window_xy(num,frame[num].curx,frame[num].cury);
    for (i = frame[num].curx; i<frame[num].endx-1; i++)
      window_putchar(num,' ');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    window_xy(num,x,y);
  }
  window_upline(num)
  int num;
  {
  if (frame[num].cury >0){
      frame[num].cury--;
      window_xy(num,frame[num].curx,frame[num].cury);
      return 1;
      }
      return 0;
  }
window_downline(num)
int num;
{
  if(frame[num].cury<frame[num].endy-frame[num].starty-1){
      frame[num].cury++;
      window_xy(num,frame[num].curx,frame[num].cury);
      return 1;
   }
   return 0;
}
window_shiftleft(num)
int num;
{
 if(frame[num].curx>0)
 {
   frame[num].curx--;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    window_xy(num,frame[num].curx,frame[num].cury);
    return 1;
   }
    return 0;
  }
  window_shiftright(num)
  int num;
  {
   if(frame[num].curx<frame[num].endx-frame[num].startx-1)
    {
    frame[num].curx++;
    window_xy(num,frame[num].curx,frame[num].cury);
    return 1;
    }
    return 0;
  }
  window_bksp(num)
  int num;
   {
   if(frame[num].curx != 0)
    {
    frame[num].curx--;
    window_xy(num,frame[num].curx,frame[num].cury);
    window_putchar(num, ' ');
    frame[num].curx--;
    window_xy(num,frame[num].curx,frame[num].cury);
}
  else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
    frame[num].cury--;
    frame[num].curx = frame[num].endx-4;
    window_xy(num,frame[num].curx,frame[num].cury);
    window_putchar(num,' ');
    frame[num].curx--;
    window_xy(num,frame[num].curx,frame[num].cury);
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ขั้นตอนการทำงานและผลการทดลอง

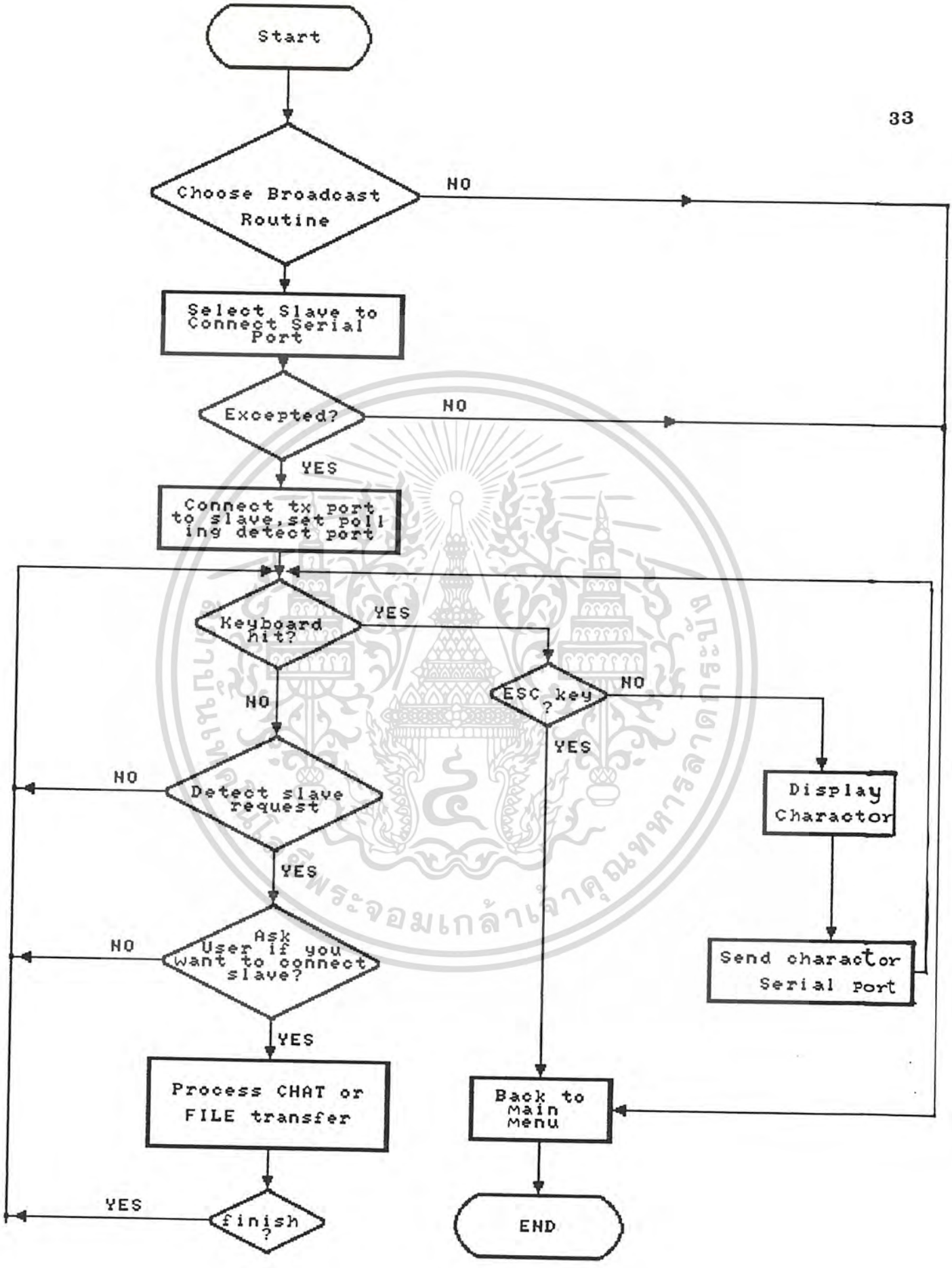
สำหรับขั้นตอนการทำงานนั้น เริ่มต้นทั้งทางฝ่าย SLAVE และ ฝ่าย MASTERจะต้องเรียกโปรแกรมสื่อสารขึ้นมาทางฝ่ายMASTERเมื่อเข้าสู่โปรแกรมจะต้องเลือกเข้า Mode คือ Broadcast Mode หรือ Single Mode จากนั้นจึงจะเลือก Application ว่าจะเป็น On line (CHAT) หรือ File transferส่วนทางฝ่าย SLAVE เมื่อเข้าสู่โปรแกรมจะอยู่ใน Broadcast อยู่แล้ว จากนั้นการทำงานจะขึ้นอยู่กับทางฝ่าย MASTER

เพื่อให้เห็นขั้นตอนการทำงานชัดเจนขึ้นจะขออธิบายเป็น Flow chart ต่างๆดังนี้คือ

- โฟล์วชาร์ทการทำงานใน Broadcast Mode ของฝ่าย MASTER
- โฟล์วชาร์ทการทำงานของฝ่าย SLAVE
- โฟล์วชาร์ทการเข้าสู่ CHAT Application ของฝ่าย MASTER
- โฟล์วชาร์ทการเข้าสู่ File transfer Application ของฝ่าย MASTER
- โฟล์วชาร์ทแสดงขั้นตอนการทำงานแบบ CHAT
- โฟล์วชาร์ทแสดงขั้นตอนการส่งไฟล์
- โฟล์วชาร์ทแสดงขั้นตอรการรับไฟล์

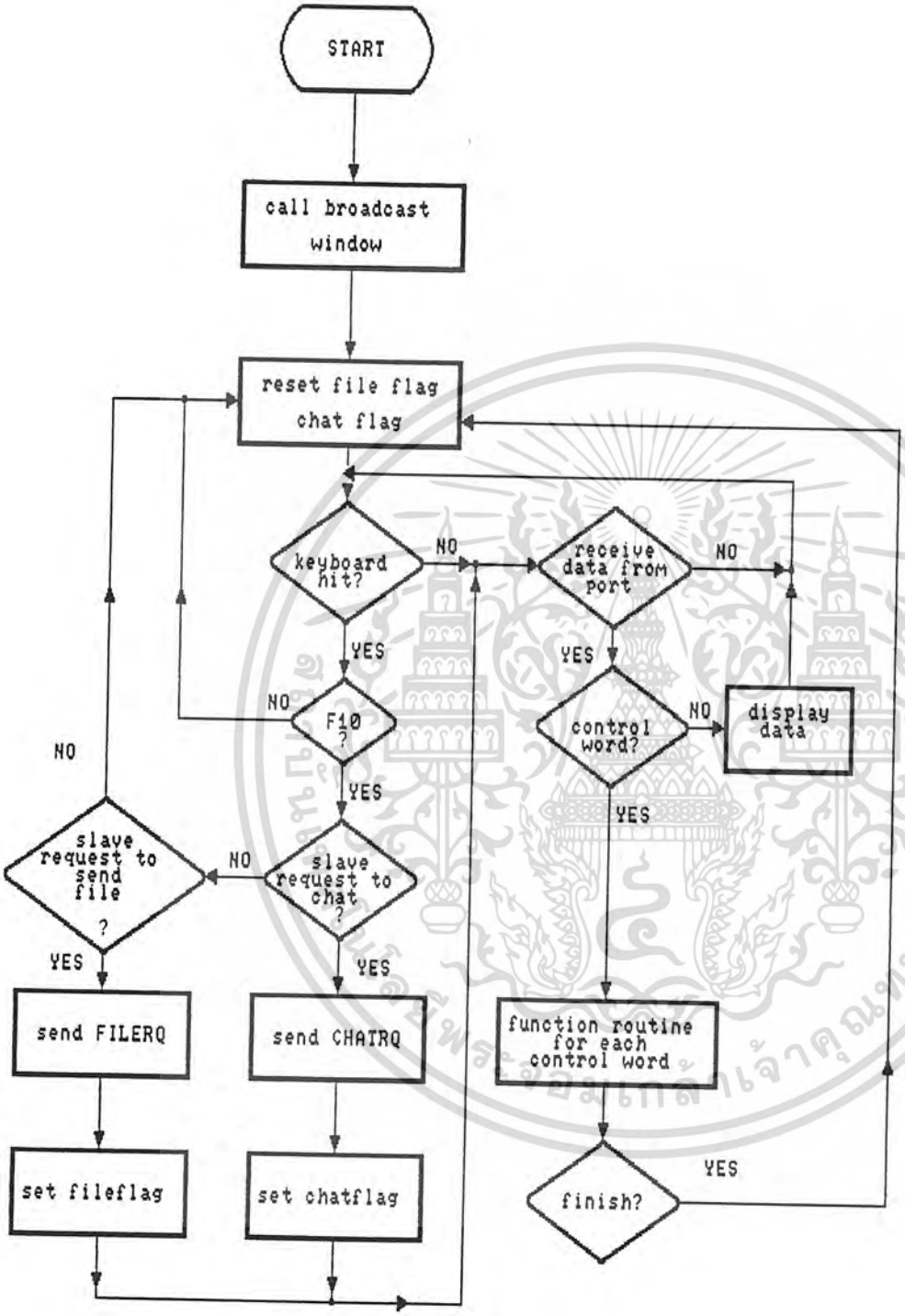
สำหรับผลการทดลองปรากฏว่าทั้งทางฝ่าย MASTER และ SLAVE สามารถสื่อสารกันได้ใน ทั้งแบบ On line Communication และ File Transfer นอกจากนี้ได้แสดงหน้าจอของการเข้าสู่การทำงานใน Mode ต่างๆไว้ด้วยในตอนท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



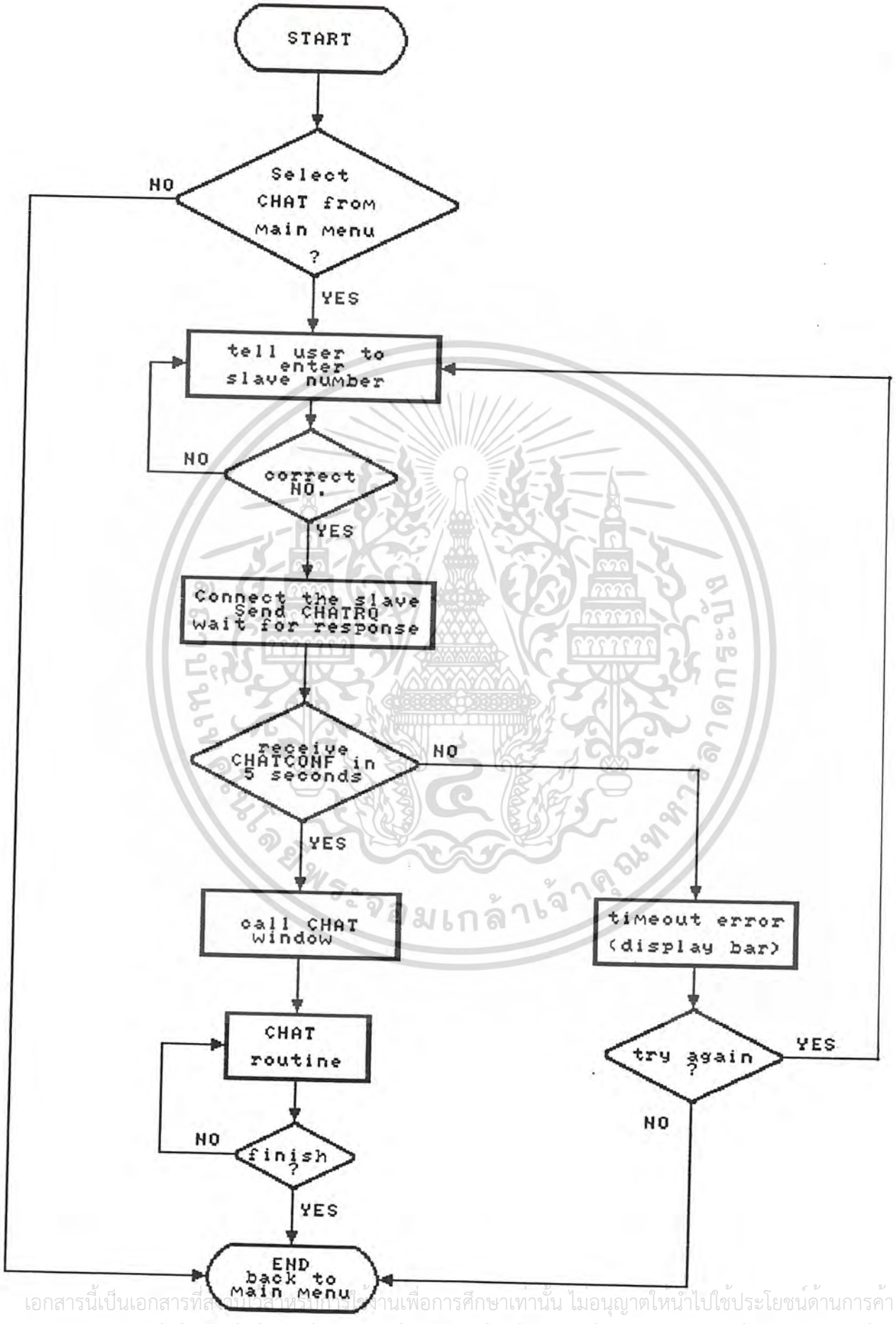
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 1 โฟลว์ชาร์ทแสดงการเข้าสู่การทำงาน
ในบรอดคาสท์โหมดของเครื่องมาสเตอร์



รูปที่ 2 โฟลว์ชาร์ทแสดงการเข้าสู่การทำงานของเครื่องสเลฟ

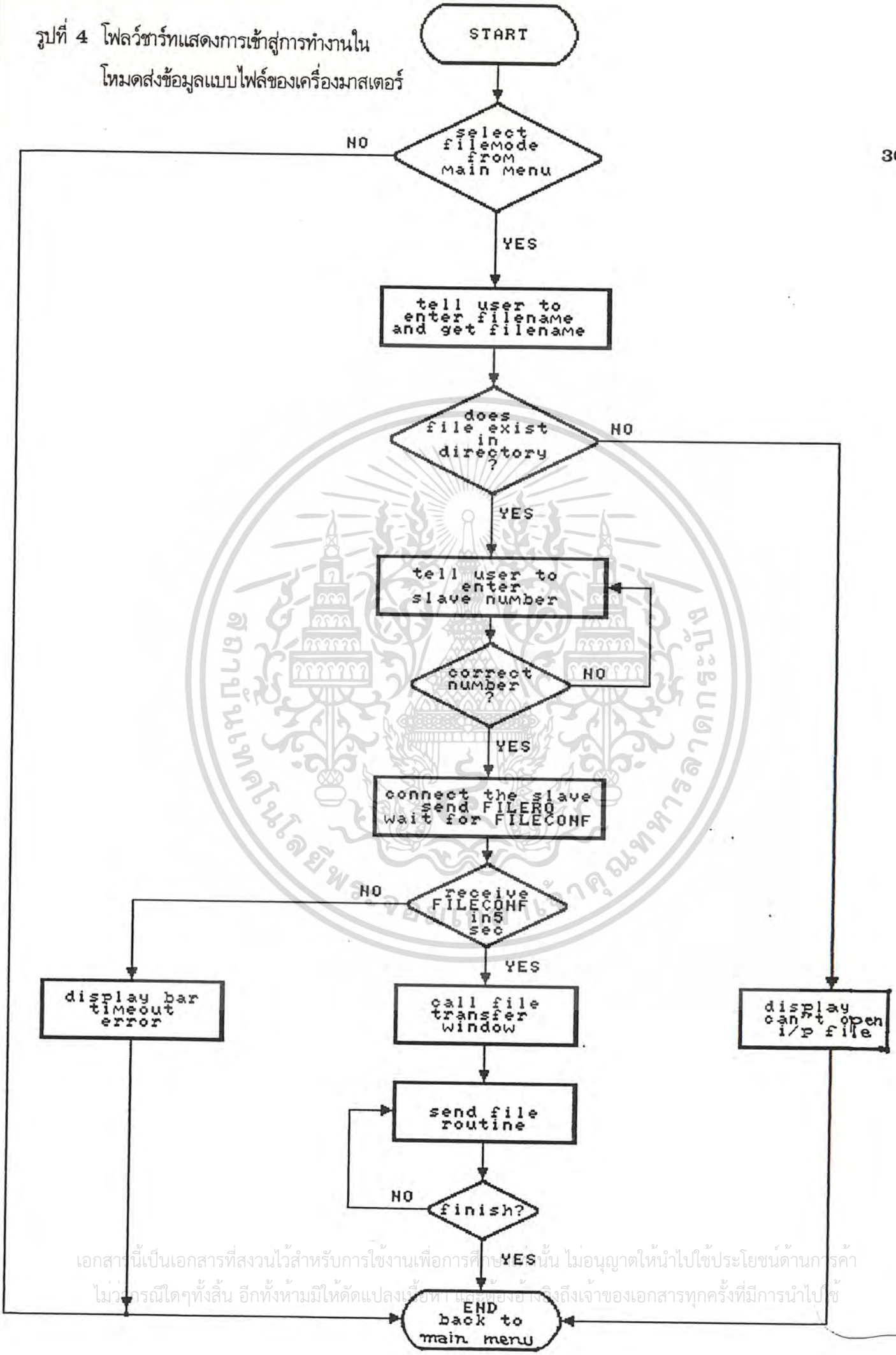
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



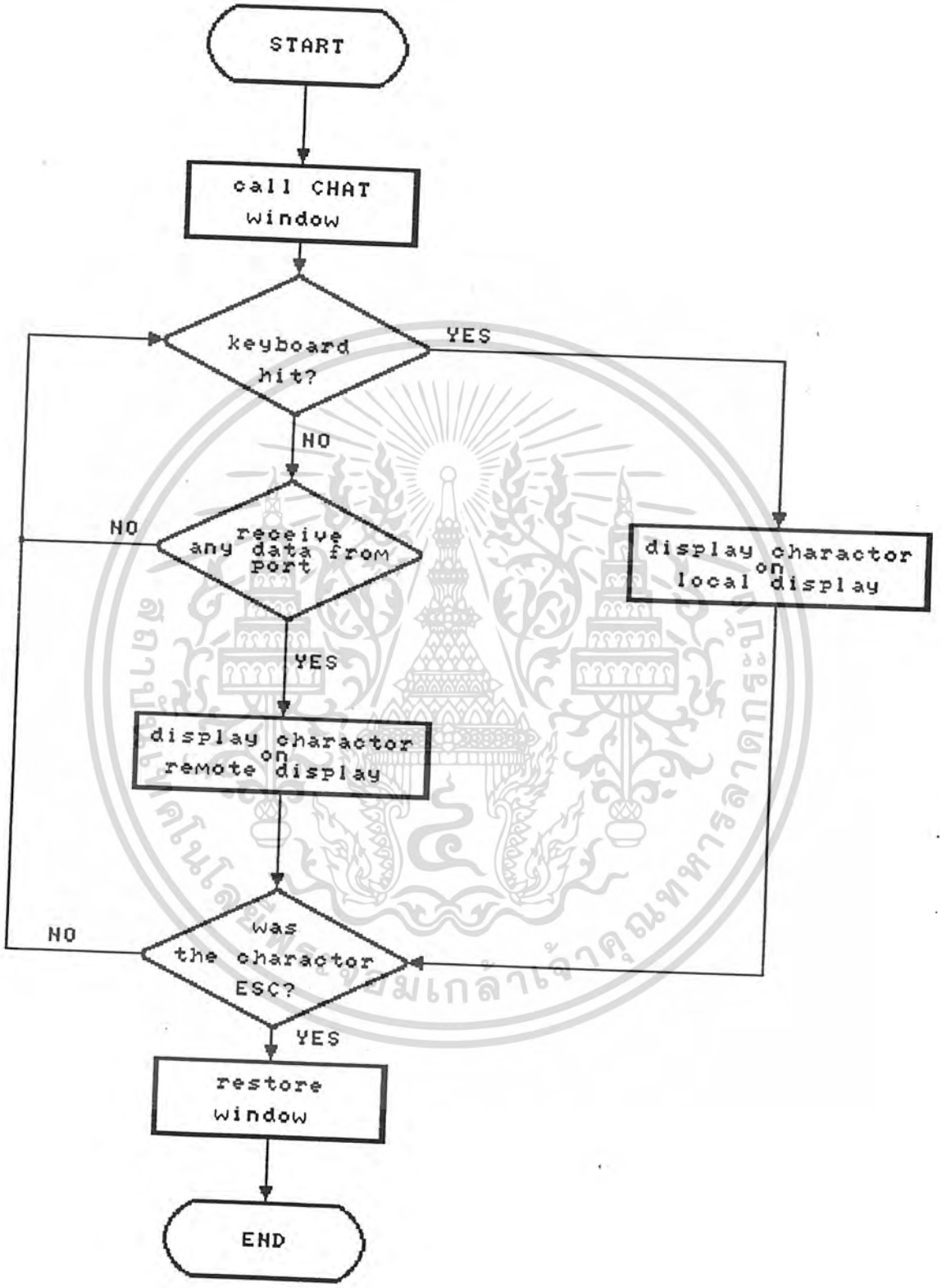
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3 โฟลว์ชาร์ทแสดงการเข้าสู่การทำงานในโหมดส่งข้อมูลแบบออนไลน์ของเครื่องมาสเตอร์

รูปที่ 4 โฟลว์ชาร์ทแสดงการเข้าสู่การทำงานใน
โหมดส่งข้อมูลแบบไฟล์ของเครื่องมาสเตอร์

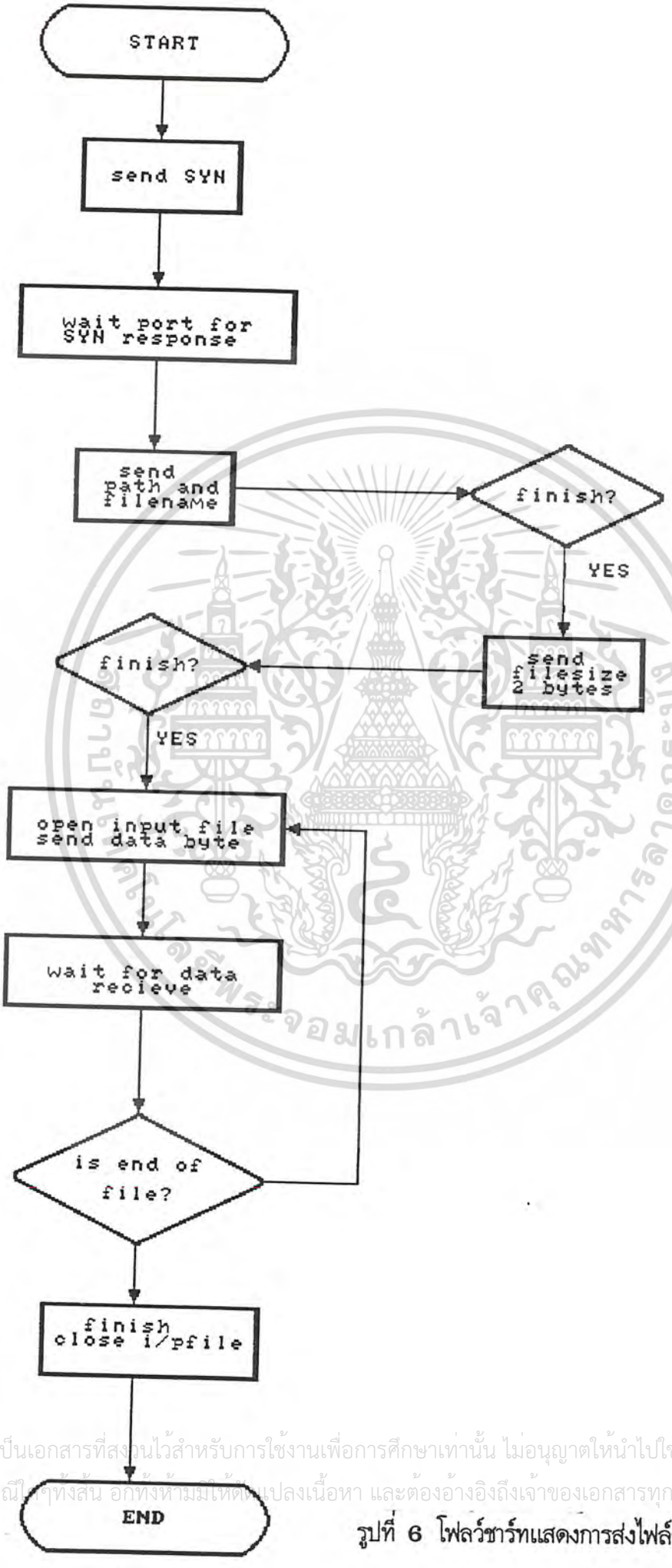


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



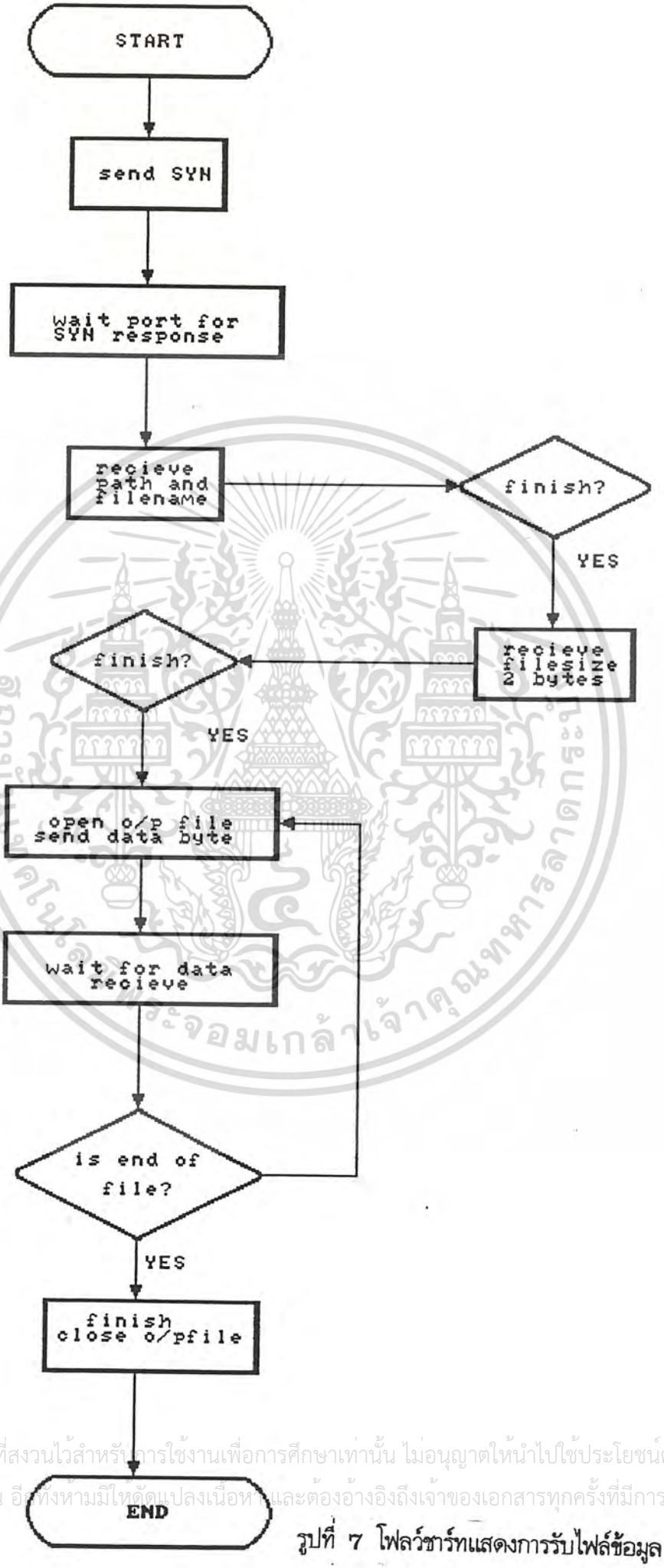
รูปที่ 5 โฟลว์ชาร์ทแสดงการรับ/ส่งข้อมูลแบบออนไลน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



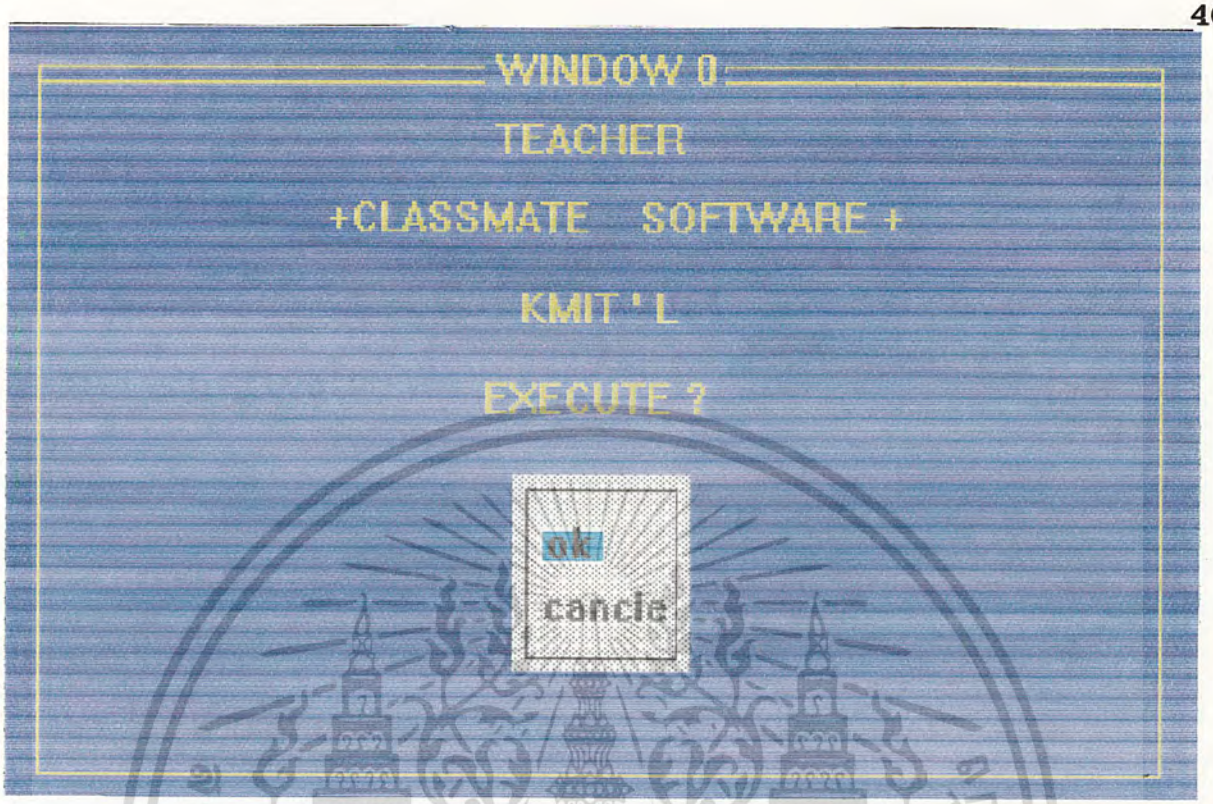
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6 โฟลว์ชาร์ทแสดงการส่งไฟล์ข้อมูล

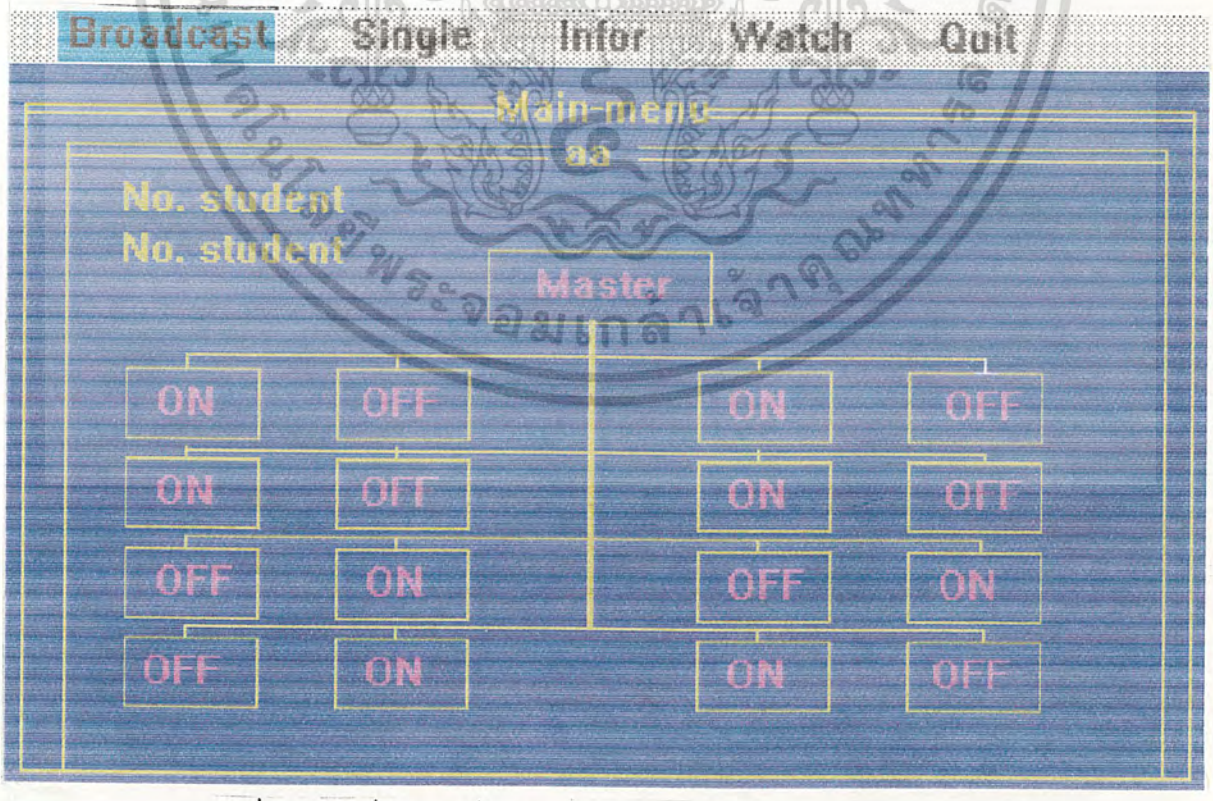


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 7 โฟลว์ชาร์ทแสดงการรับไฟล์ข้อมูล

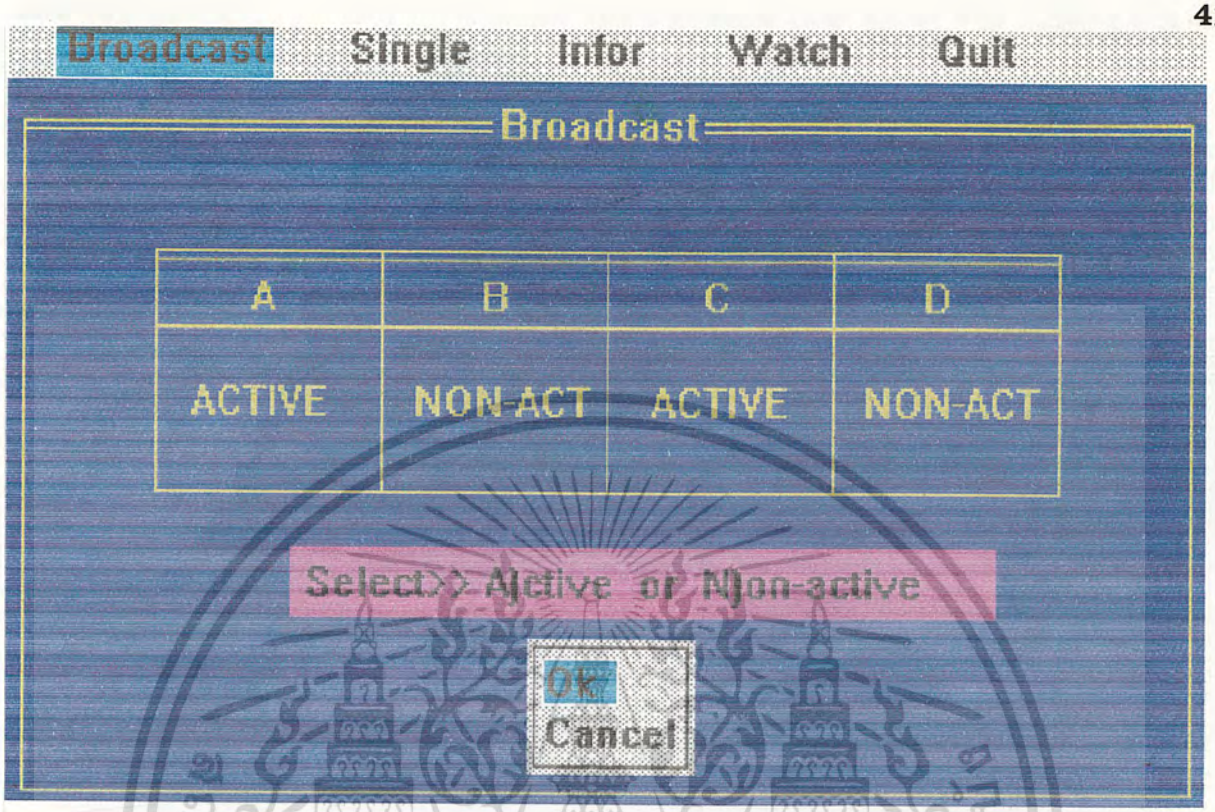


รูปที่ 8 หน้าจอที่ปรากฏเมื่อ MASTER เข้าสู่โปรแกรม

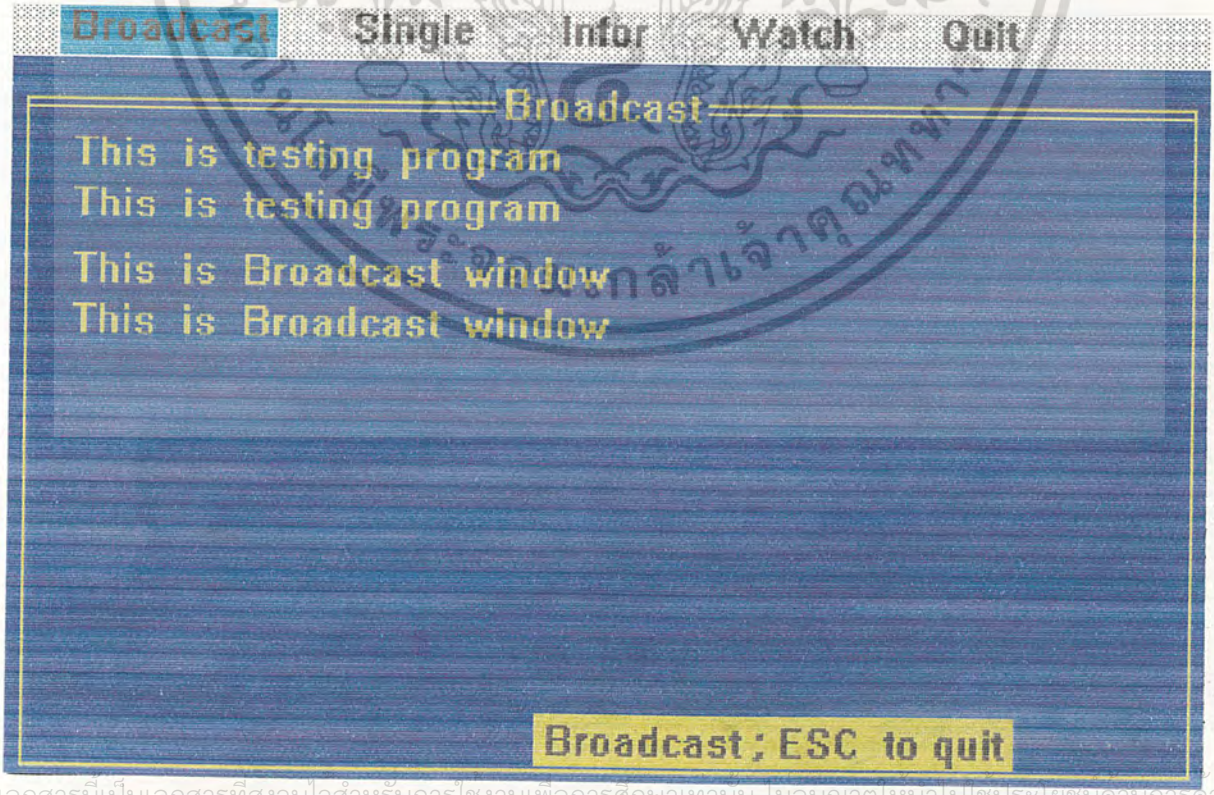


รูปที่ 9หน้าจอที่ปรากฏเมื่อMASTER ตกลงเข้าสู่ Main menu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

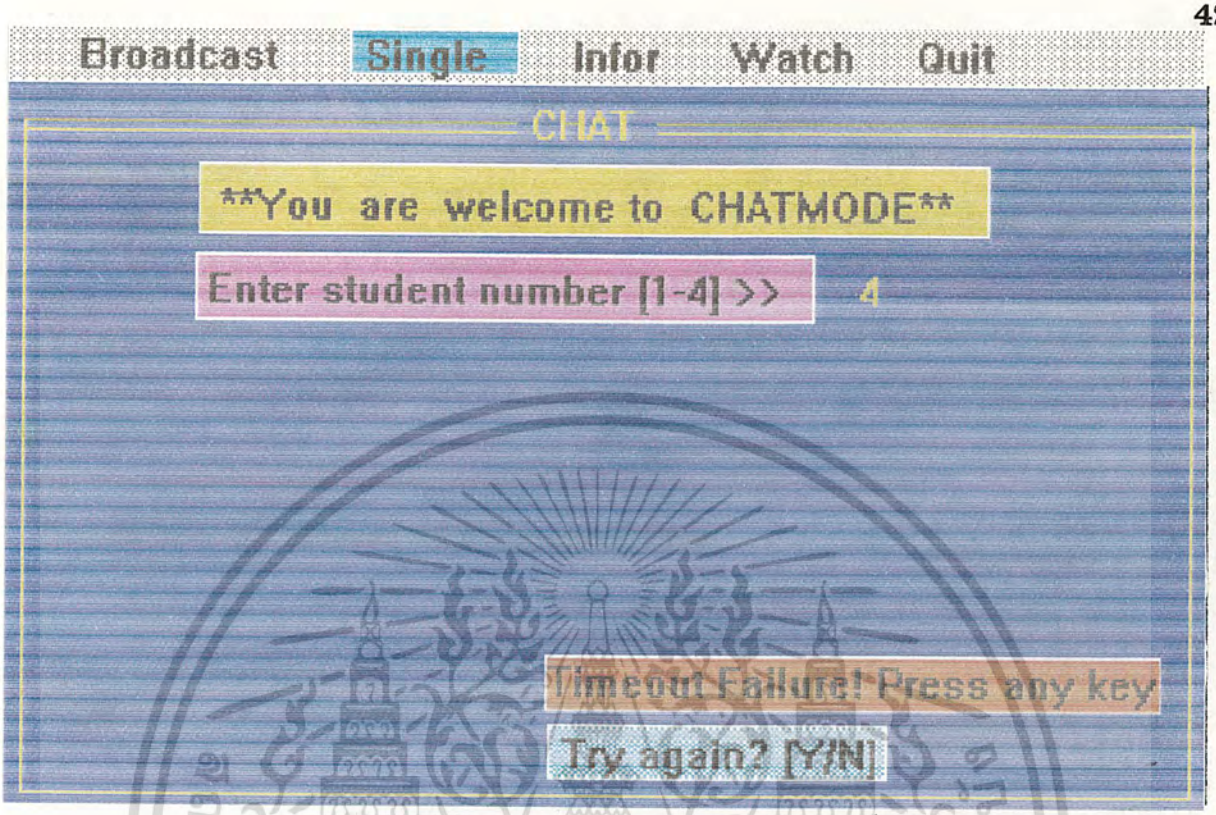


รูปที่ 10 หน้าจอที่ปรากฏเมื่อ MASTER เลือกโหมดการทำงานเป็นแบบ Broadcast และเลือกเครื่องคอมพิวเตอร์ที่ต้องการให้รับข้อมูล

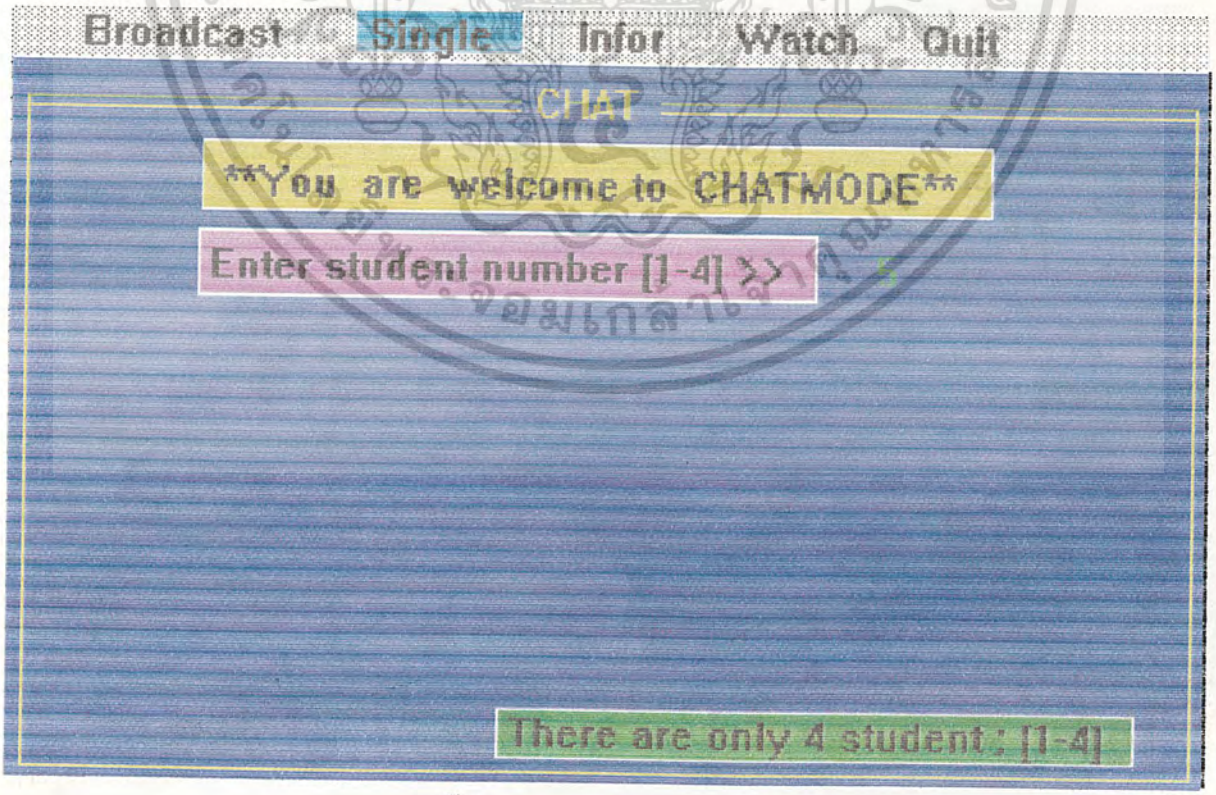


รูปที่ 11 หน้าจอที่ปรากฏเมื่อ MASTER เริ่มส่งข้อมูลไปยัง SLAVE ที่กำหนดไว้ในรูปที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

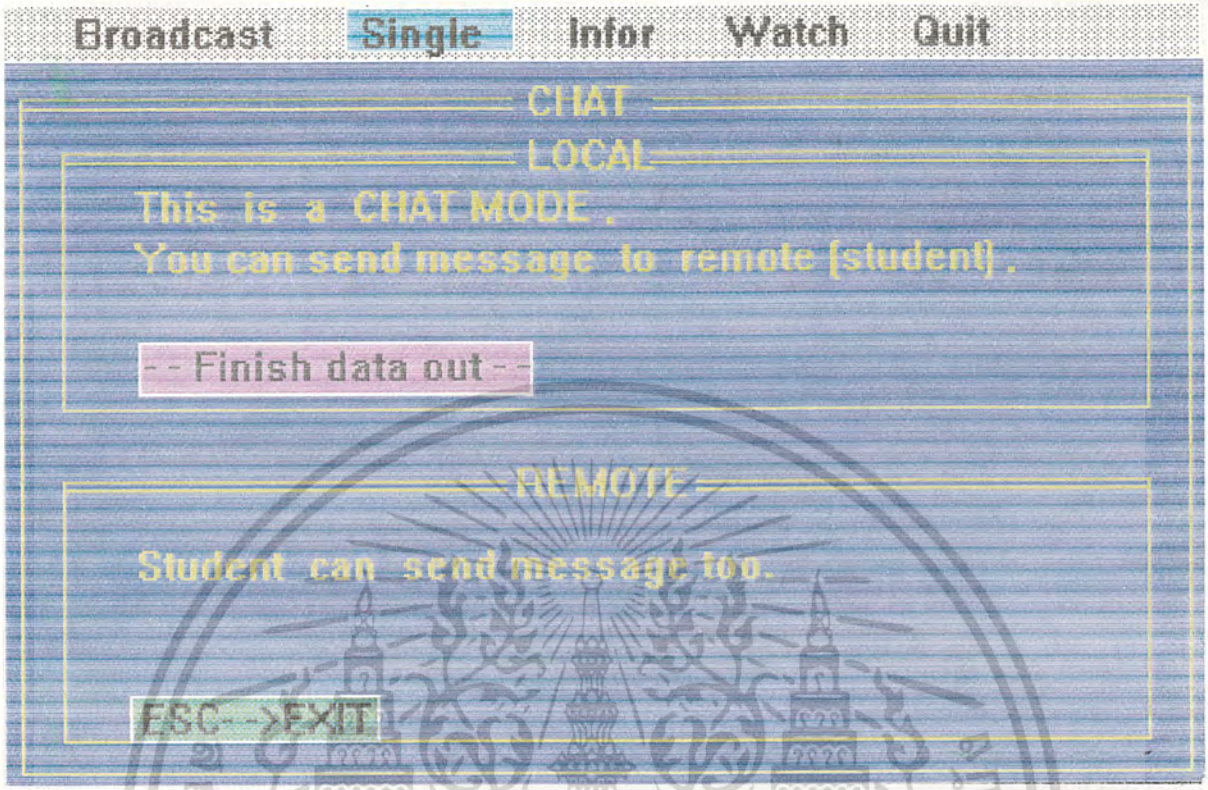


รูปที่ 12

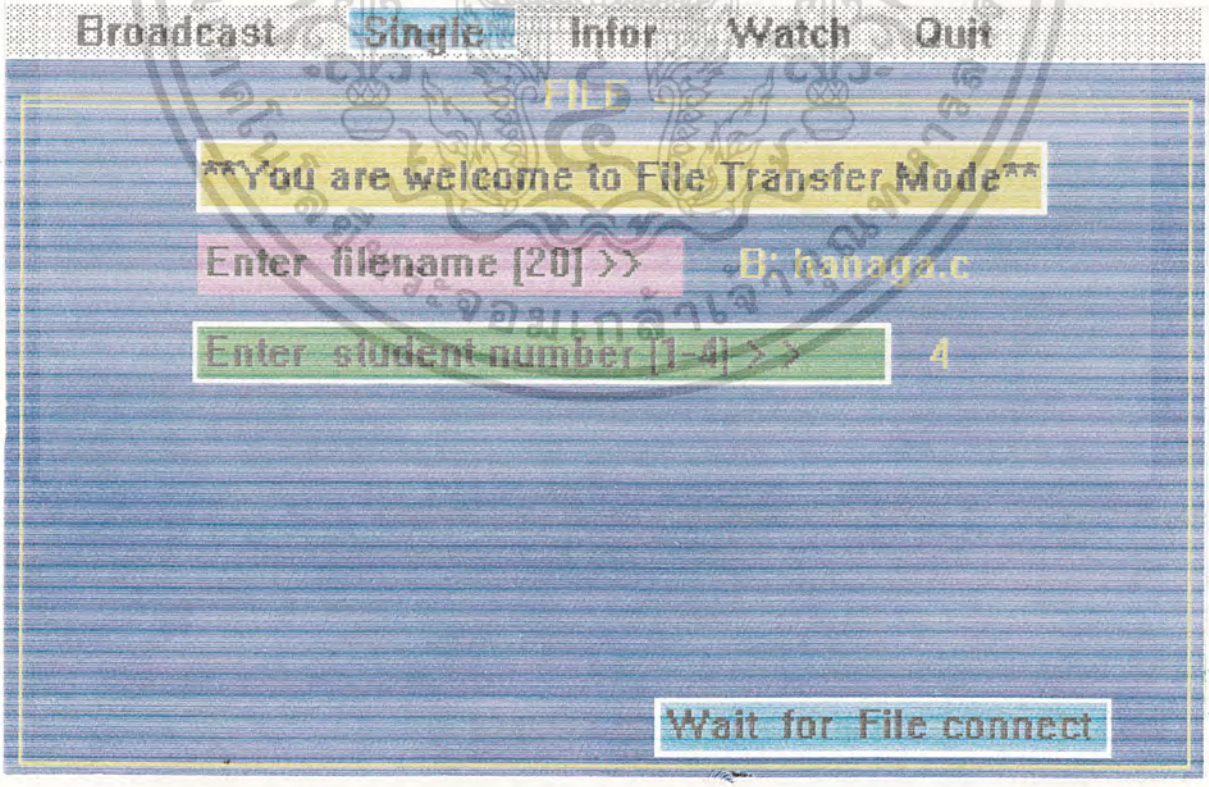


รูปที่ 13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

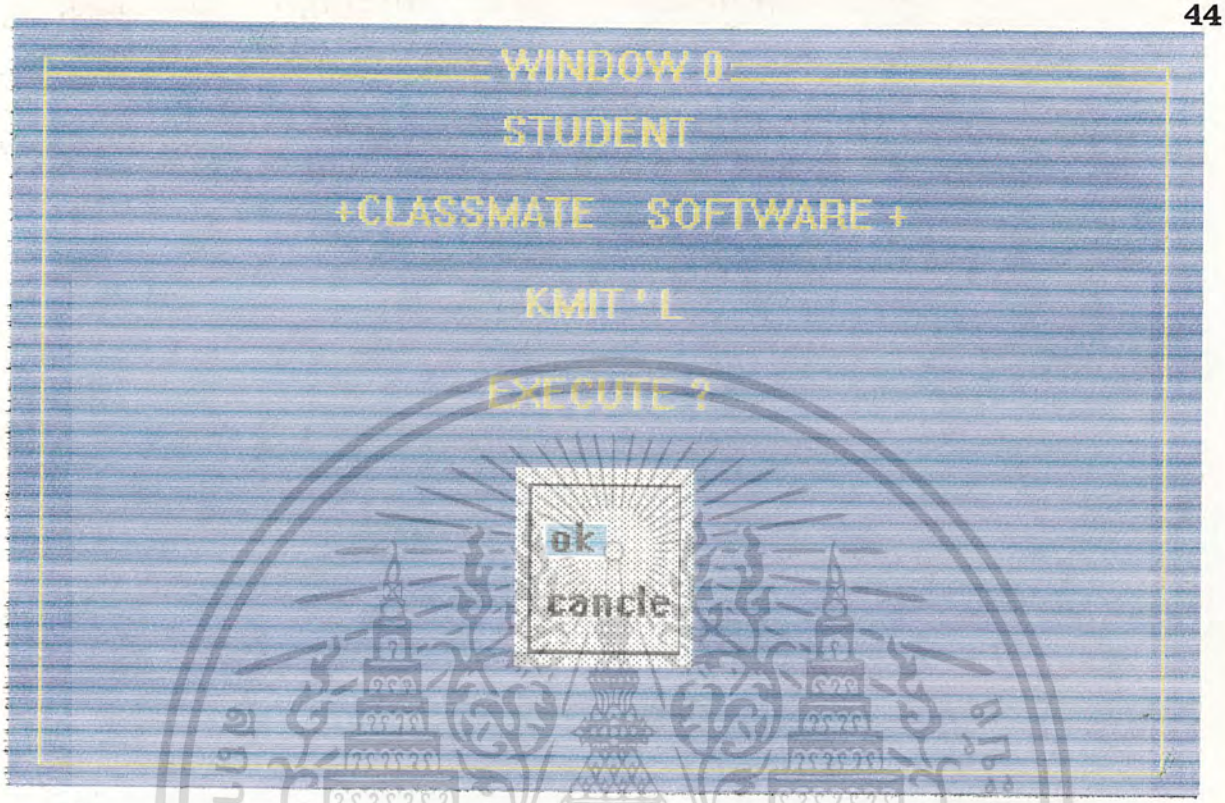


รูปที่ 14 หน้าจอที่ปรากฏเมื่อ MASTER เลือกโหมดการทำงานแบบ Single และเลือกรูปแบบการสื่อสารเป็นการส่งข้อความ

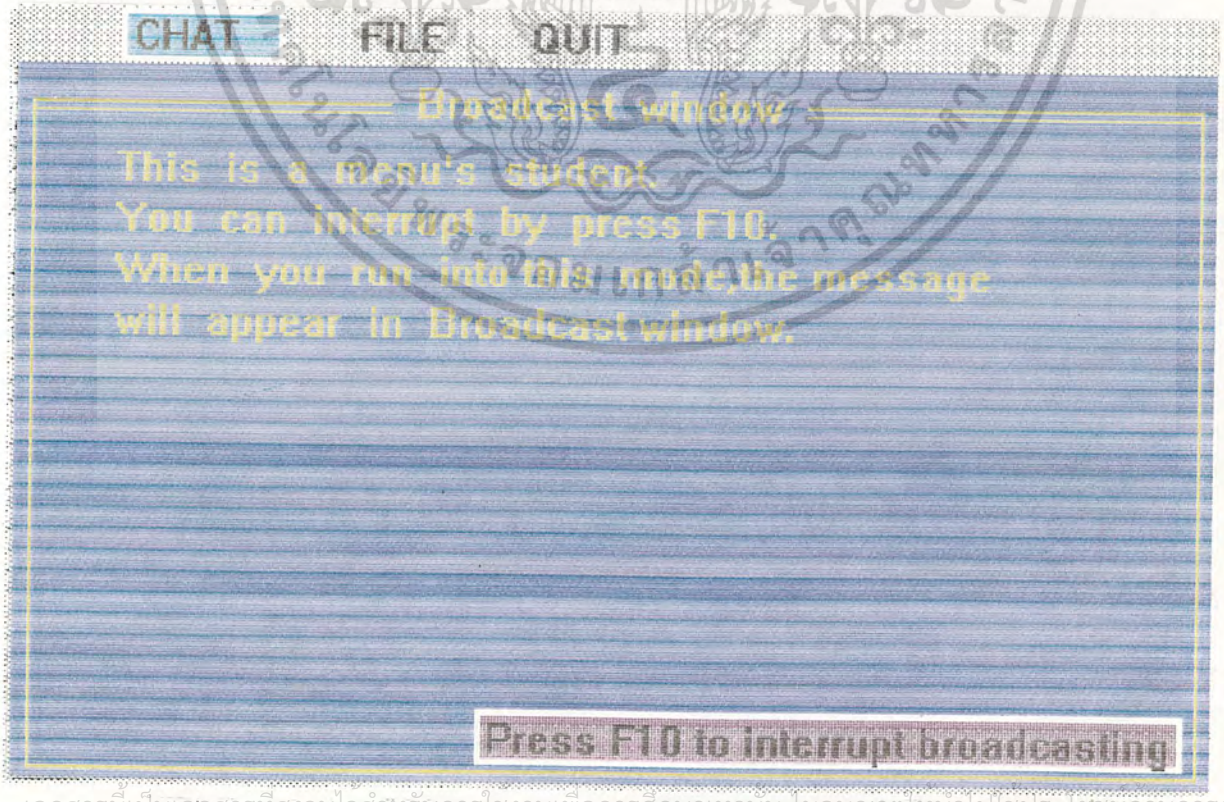


รูปที่ 15 หน้าจอที่ปรากฏเมื่อ MASTER เลือกโหมดการทำงานแบบ Single และเลือกรูปแบบการสื่อสารเป็นการส่งไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 16 หน้าจอที่ปรากฏเมื่อ SLAVE เข้าสู่โปรแกรมและจะรับข้อความที่ MASTER ส่งมาทันที

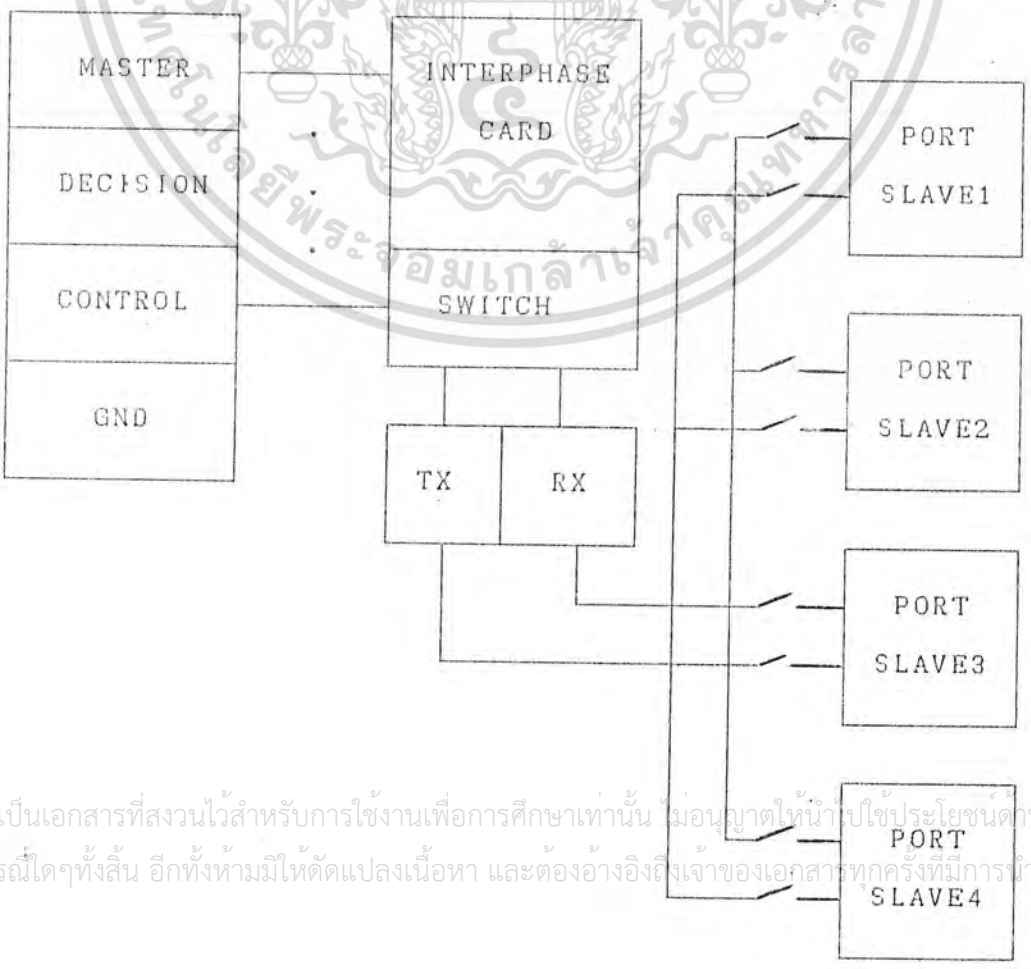


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่17หน้าจอที่ปรากฏเมื่อ SLAVE ต้องการขอติดต่อกับMASTER

บทที่ 5
บทสรุปโครงงาน

โครงงานนี้ออกแบบโดยมีวัตถุประสงค์หลักคือเพื่อให้การสื่อสารผ่านคอมพิวเตอร์ในห้องเรียนเป็นไปอย่างมีประสิทธิภาพมากขึ้น กล่าวคือผู้สอนสามารถติดต่อกับผู้เรียนผ่านทางคอมพิวเตอร์ได้โดยตรง และในทางกลับกันผู้เรียนก็ขอติดต่อไปยังผู้สอนได้เช่นกัน เมื่อสิ้นสุดการติดต่อแล้วจึงกลับสู่การใช้งานคอมพิวเตอร์ตามปรกติ ฮาร์ดแวร์ที่ใช้จึงเป็นแบบพื้นฐานไม่ยุ่งยาก สามารถใช้กับเครื่องพีซีใดๆ ที่มีการ์ดสื่อสารแบบอนุกรม ตามมาตรฐาน RS-232 โดยเพิ่มการ์ดควบคุมที่ออกแบบไว้เข้าที่เครื่อง MASTER เพียงเครื่องเดียว แล้วต่อเชื่อม twisted pair wire จากพอร์ตของการ์ดไปยังเทอร์มินัลแต่ละเครื่อง จากนั้นการควบคุมการสื่อสารใดๆภายในเครือข่ายจะมาจากเครื่องที่เป็นมาสเตอร์ ตามธรรมชาติของการสื่อสารในห้องเรียนที่ผู้สอนเป็นผู้ควบคุมการติดต่อ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Pull-down menu routines for text mode operation and
   short sample program.
*/
 #include <dos.h>
 #include <bios.h>
 #include <conio.h>
 #include <stdio.h>
 #include <stdlib.h>
 #include <ctype.h>
 #include <sys\timeb.h>
 #include <mem.h>
 #include <string.h>
 #include <process.h>

#define BORDER 1
#define ESC 0x1B
#define MAX_FRAME 30
#define REV_VID 0x70
#define NORM_VID 7          /**/
#define GREEN     0x20

 #define COM2       1
 #define STATUS     3
 #define SETS       0
 #define DATA       0x03
 #define STOP       0x04
 #define PARITY     0x08 /*0x08-odd;0x18-even*/
 #define BAUD       0x80 /*2400*/
 #define BYTE       DATA|STOP|PARITY|BAUD
 #define PORTBASE 0x2f8
 #define TX         PORTBASE
 #define RX         PORTBASE
 #define RCVRDY     0x100
 #define XMTRDY     0x2000
 #define ESC        0x1B
 #define BKSP       8
 #define ENQ        5
 #define ACK        6
 #define DISC       10
 #define CHATRQ       12
 #define CHATCONF     11
 #define FILERQ       14
 #define FILECONF     15
 #define SYN          22

  struct timeb t;
  long double time1,time2;
  float  timepast;

  struct node{
    int stnum;
    unsigned int txadd;
    unsigned int rxadd;
    int state;
    }term[4]={{1,0x01,0x00,1},{2,0x02,0x10,1},{3,0x04,0x20,0},{4,0x08,0x30,0}};
  void broad();
  void mchat();
  void msfile(),sendf();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    void recf();
    void initport();
    void settx();
    void setrx();
    unsigned int filesize() ;
 void save_video(int num);
 void restore_video(int num);
 void write_video(int x, int y, char *p, int attrib);
 void write_string(int x, int y, char *p, int attrib);
 void write_char(int x, int y, char ch, int attrib);
 void pd_driver(void),windows();
 void display_header(int num);
 void display_menu_x(int num),display_menu_y(int num);
 void draw_border(int num),draw_borderx(int num);
 void draw_bordery(int num),draw_tree(),draw_sub_tree();
 void draw(int startx,int starty,int endx,int endy);
 void draw_sub_string(),goto_xy(),window_gets();
 void window_cls(),window_cleol();
 void save(),restore();

 int make_menu(int num, char *menu[], char *keys,
               int count, int x, int y, int border);
 int get_resp(int num), pulldown(int num);
 int video_mode(void);
 char far *vid_mem;
 int readkey(void);

 struct menu_frame {
   int startx, endx, starty, endy;
   int curx,cury;
   unsigned char *p;
   char **menu;
   char *keys;
   char *header;
   int border, count;
   int active;
 } frame[MAX_FRAME],i;

 char *answer1[]= {
   "OK",
   "CANCEL",
 };
 char *answer2[]= {
   "BROADCAST",
   "SINGLE",
   "INFOR",
   "WATCH",
   "QUIT",
 };
 char *anssingle[]= {
    "CHAT",
    "FILE",
 };
 char LINK[16] ={ 0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1 };
 char VALUE[4][4]={
          {  0, 1, 2, 3 },
          {  4, 5, 6, 7 },
          {  8, 9,10,11 },
          { 12,13,14,15 },
    };
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char *ansinf[]={
    "baud 4800",
    "port COM2",
    "student 4",
    };

bkcolor(n)
 int n;
  {
     int i,j;
     clrscr();
     for(i=0;i<81;i++)
     {
       for(j=0;j<26;j++)
       {
           textattr(n);
           gotoxy(i,j);
           putch(0);
       }
     }
 }
 /*spacecolor(n,num,startx,starty,endx,endy)
 int n;
 {
   int num,startx,starty,endx,endy;
   frame[num].startx=startx;
   frame[num].starty=starty;
   frame[num].endx = endx;
   frame[num].endy = endy;

   for(startx=frame[num].startx;startx<frame[num].endx;startx++)
   {
     for(starty=frame[num].starty;starty<frame[num].endy;starty++)
     {
       textattr(n);
       goto_xy(frame[num].startx,frame[num].starty);
       putch(0);
     }
   }
 } */
 main(void)
{
  int p,o;
  gotoxy(0,0);
  bkcolor(0x11);

  /* first, create the menu frames */
  make_window(0, "WINDOW 0",1, 1,79, 23, BORDER);
  make_window(1, "MAIN_MENU ",1, 1, 79, 23, BORDER);
  make_window(2," Broadcast_window ",1,1,79,23,BORDER);
  make_window(3," single_window ",1,1,79,23,BORDER);
  make_window(4,"aa",5,2,75,23,BORDER);
  make_window(5,"FILE",1,1,79,23,BORDER);
  make_window(6,"CHAT",1,1,79,23,BORDER);
  make_window(7,"",1,1,79,3,BORDER);
  make_window(8,"LOCAL",2,4,78,13,BORDER);
  make_window(9,"REMOTE",2,14,78,23,BORDER);
  make_menu_y(10, answer1, "oc",2,35,17,BORDER);
  make_menu_x(11, answer2, "bsiwq",5,0,0,BORDER);
  make_menu_y(12,anssingle,"cf",2,13,1,BORDER);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   make_menu_y(13,answer1,"oc",2,35,15,BORDER);
   make_menu_y(14,ansinf,"bps",3,25,1,BORDER);
   bioscom(SETS,BYTE,COM2);
   initport();
   pd_driver(); /* activate the menu system */
   textattr(7);
   clrscr();
}

/* Demonstrate the pull-down menu functions. */
void pd_driver(void)
{ int l;
  int txcode;
  int q,p,r,s,set=1;
  int ans1,ans2,anszone,anssin,ansinf;
  int get[4]={0,0,0,0};
      clrscr();
      windows(0);
      q=0;
 while(q==0)
 {
   write_string(35,5,"TEACHER",0x1B);
   write_string(29,8,"+CLASSMATE SOFTWARE+",0x1B);
   write_string(36,11,"KMIT'L",0x1B);
   write_string(35,14,"EXECUTE?",0x1B);

   while((ans1=pulldown_y(10)) != -1)
   {
     if(ans1==0)
     {
          for(l=0;l<3;l++)
          {
          xmt_ch(ENQ);}
          write_string(frame[10].startx+2,frame[10].starty+ans1+1,frame[10].menu[ans1],0x30)
          restore_video(10);
          windows(1);
          windows(4);
          write_string(8,3,"NO. OF STUDENT",0x1B);
          write_string(8,4,"NO. OF STUDENT",0x1B);
          draw_tree(10,7);
          pulldown_x(11);
          restore_video(4);
          p=0;
          while(p==0)
          {
               set=1;
               while(set==1&&(ans2=pulldown_x(11))!= -1)
               {
                 write_string(frame[11].startx+4+(ans2*12),frame[11].starty,frame[11].menu[an
0x30);

                 switch(ans2)
                 {
                     case 0: /*Broadcast*/
                            windows(2);
                            s=0;
                            while(s==0)
                            {
                              /* function draw on/off */
                              drawzone();
                              write_string(25,17,"Select>> A)ctive  or  N)on-active",0x34);
                              window_xy(2,13,7);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        for(r=0;r<4;r++)
                        {
                        window_xy(2,13+r*15,7);
                        get[r]=get_res_zone(2);

                        }
                        while((anszone = pulldown_y(13)) != -1)
                        {
                         write_string(frame[13].startx+2,frame[13].starty+anszone+1,
                         frame[13].menu[anszone],0x30);
                         if(anszone==0)  /*zone*/
                           {
                            restore_video(13);
                            window_cls(2);
                            txcode=code(get);
                            broad(txcode);
                            restore_video(2);
                            break;
                           }
                         if(anszone==1)
                            break;
                        } /* end of loop anszone */
                        restore_video(13);
                        s = 1;
                      } /* end of loop s=0*/
                     restore_video(2);
                     window_cls(1);
                     break;
        case 1: /**Single**/
           while((anssin = pulldown_y(12)) != -1)
           {
               write_string(frame[12].startx+2,frame[12].starty+anssin+1,frame[12].menu[an
],0x30);
               switch(anssin)
               {

                   case 0: /*chat*/
                     restore_video(12);
                     windows(6);
                     mchat();
                     restore_video(6);
                     break; /*end chat*/
                  case 1:/*file*/
                     restore_video(12);
                     windows(5);
                      msfile();
                      restore_video(5);
                      break;
                      }
           }
           restore_video(12);
           break;
         /*end sing*/
               case 2:/****Infor**/
                     while((ansinf = pulldown_y(14)) != -1)
                     {
                         write_string(frame[14].startx+2,frame[14].starty+ansinf+1,frame
.menu[ansinf],0x30);
                     }
                     restore_video(14);
                     break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                case 3: /***Watch***/



                                        windows(4);
                                        write_string(8,3,"NO. OF STUDENT",0x1B);
                                        write_string(8,4,"NO. OF STUDENT",0x1B);
                                        draw_tree(10,7);
                                        getch();
                                        restore_video(4);
                                        break;

                        case 4: /*QUIT*/
                                set = 0;
                                xmt_ch(DISC);
                                break;
                    }  /*end of switch*/

                }   /*end of loop while ans2*/
                restore_video(11);
                p = 1;
            }  /*end of loop p=0*/
            restore_video(1);
        }   /*end if ans=0*/
        if(ans1==1)
        {
            xmt_ch(DISC);
            break;}
    }   /* end of loop while ans1*/
    restore_video(10);
    q = 1;
  }     /* end of loop q=0 */
  restore_video(0);
}



/*************************************************************/
/************************INCHAT*******************************/
window_inchat()
{
union inkey{
    char ch[2];
    int  in;
    }c;
char  r;
int end;
    windows(8);
    windows(9);
    write_string(3,23,"ESC-->EXIT",0x1B);
    window_xy(8,3,1);
    end=0;
    while(end==0)
    {
          if(kbhit())  {
                        c.in=0;
                        c.in=window_getche(8);
                        if(c.ch[0] != ESC)
                        {
                           xmt_ch(c.ch[0]);
                           switch(c.ch[0]) {
                                        case '\r' :
                                        window_putchar(8,'\r');
                                        break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

```
                                        case '\b' :
                                             window_bksp(8);
                                             break;
                                        default    :
                                            window_putchar(8,c.ch[0]);
                                            break;
                                        }    /**end switch***/
                                } /***end while ESC****/
                                else
                                {
                                xmt_ch(ESC);
                                write_string(5,12,"------Finish data out ----------",0x2A);
                                end=1;
                                }
                        }
        else {

                if(rcv_check() == RCVRDY)
                {
                     switch( r = rcv_ch())
                   {
                     case '\r' :
                               window_putchar(9,'\r');
                               break;
                     case '\b' :
                               window_bksp(9);
                               break;
                     case ESC :
                               write_string(5,22,"end data in",0x2A);
                               end=1;
                               break;
                     default :
                               window_putchar(9,r);
                               break;
                     }/***end switch ***/
                  }/*end if   */

              }/***end else****/
        } /*end while*/
        getch();
        window_cls(8);
        window_cls(9);
        restore_video(8);
        restore_video(9);

}

/*<><><><><><><><><><><><><><><><><BROAD&SING<><><><><><><><><><><><><><><><><><><><><><><><><><><><*/

                                                                        #include <dos
void broad(txcode)
{  int bro,j;
    int no,ans,fin;
    union inkey{
                char ch[2];
                int i;
                }c;
  /* broadwin();*/

  write_string(40,23,"Broadcasting;ESC to quit",0x34);
  window_xy(2,3,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
settx(txcode);
bro=1;
while(bro==1)
{   outportb(514,0xff);
    if(!kbhit())
    {
        for(j=0;j<5;j++)
        {
        no=detect();/*polling*/
        if(no>=0) break;
        }
        if(no>=0)/*get only a single slenq*/
        {

           settx(term[no].txadd);/*link only 1 slave*/
           setrx(term[no].rxadd);
           while(rcv_check()!=RCVRDY);
           switch(rcv_ch())
           {case CHATRQ:
               write_string(40,22,"Student CHAT Interrupt[Y/N]",0x34);/*pop-up*/
               for(;;)
               {
                    c.i=window_getche(2);
                    ans=toupper(c.ch[0]);
                    switch(ans)
                    {case 'Y':
                          xmt_ch(CHATCONF);
                          window_inchat();/*inchatwin */
                          settx(txcode);
                          write_string(40,22,"                                    ",0x11);/
-up*/
                           break;
                    case 'N':
                     write_string(40,22,"                                    ",0x11);/*pop-up
                       break;
                    default:
                       continue;
                       }
                    break;
                }
           break;
           case FILERQ:
              write_string(40,22,"Student FILETR Interrupt[Y/N]",0x34);/*pop-up*/
              for(;;)
              {
                    c.i=window_getche(2);
                    ans=toupper(c.ch[0]) ;
                    switch(ans)
                    {case 'Y':
                          xmt_ch(FILECONF);
                               windows(5);
                               recf();/*recfwin*/
                               settx(txcode);
                               window_cls(5);
                               restore_video(5);
                           write_string(40,22,"                                    ",0x11);/*p
p*/
                               break;
                          case 'N':
                           write_string(40,22,"                                    ",0x11);/*p
p*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    break;
                            default:



                                        continue;
                                    }
                                break;
                                }
                            break;
                        default :/*not chat or file rq*/
                            break;
                            }
                        }
                        /*more than 1 detector no detect*/
                }

                else /*if kbhit*/
                {       c.i=window_getche(2);
                        if(c.ch[0]!=ESC)
                          {
                            xmt_ch(c.ch[0]);
                            window_putchar(2,c.ch[0]);}
                         else
                          {
                            xmt_ch(c.ch[0]);
                            bro =0;}
                          }
                }
                write_string(40,23,"Leave Broadcasting;Back to main menu",0x34);
                getch();
            }

int code(int choice[4])
{ int x,i;
  x =0x00;
  for(i=0;i<4;++i)
   { if(choice[i]==1)
       x=(x|term[i].txadd);
       }
  return (x);
  }
void mchat()
 {
    union inkey{
              char ch[2];
              int i;
              }c;
    int chat;
    int n,no;
    char ans;
    int timeout;

     write_string(21,3,"** You are welcome to CHATMODE **",0x3);
     chat=1;
     while (chat==1)
     {
            write_string(5,8," Enter student Number [1-4]>>",0x30);/*chatrqwin*/
            window_xy(6,40,8);
            gotoxy(frame[6].startx+frame[6].curx,frame[6].starty+frame[6].cury);
            scanf("\n %d",&n);
            while((n<1)||(n>4))
            {
                write_string(40,21," There are only 4 students;[1-4] ",0x35);
                write_string(5,8," Enter student Number again[1-4]>>",0x30);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                window_xy(6,40,8);
                gotoxy(frame[6].startx+frame[6].curx,frame[6].starty+frame[6].cury);
                scanf("\n %d",&n);
                }
            no=n-1;
            write_string(40,21," Wait for Chatconnect..",0x34);
            write_string(65,21,"            ",0x11);
            settx(term[no].txadd);
            setrx(term[no].rxadd);
             timeout=0;
             ftime(&t);time1=t.time+t.millitm/1000.0;
             xmt_ch(CHATRQ);
            while(rcv_ch()!=CHATCONF)
            {
               ftime(&t);time2=t.time+t.millitm/1000.0;
               timepast=time2-time1;
               if(timepast>5)   {timeout=1;break;}
               xmt_ch(CHATRQ);
               }

             if(timeout==0)
            {     window_cls(6);
                  window_inchat();
                  chat=0;
                  }
            else
            {
                  write_string(40,21," Timeout Failure! Press any key ",0x35);
                  getch();
                  for(;;)
                  {
                        write_string(40,22," Try again?[Y/N]>>",0x35);
                        window_xy(6,22,19);
                        c.i=window_getche(6);
                        ans=toupper(c.ch[0]);
                        window_putchar(6,ans);
                        switch(ans)
                        { case 'Y':
                                    window_cls(6);
                                    break;/*backto chat*/
                          case 'N':
                                    chat=0;
                                    break;/*backto main*/
                          default:
                                    continue;
                                    }
                         break;
                     }/*out for ans*/
                }/*out chat*/
             }
      }

void msfile()
{
    FILE *fp;
    union inkey{
          char ch[2];
          int i;
          }c;
    int data,r;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int n,no;
char ans;
char fname[15];
int timeout;

 write_string(19,3,"** You are welcome to FILETRANSFER MODE **",0x34);
 write_string(5,6," Enter filename[20]>>",0x35);
 window_xy(5,28,4);
 gotoxy(frame[5].startx+frame[5].curx,frame[5].starty+frame[5].cury);
 window_gets(5,fname);
 if((fp=fopen(fname,"rb"))!=NULL)
 {
       data=1;
       while (data==1)
       {
        write_string(5,8," Enter student Number [1-4]>>",0x35);/*chatrqwin*/
        window_xy(5,40,8);
        gotoxy(frame[5].startx+frame[5].curx,frame[5].starty+frame[5].cury);
        scanf("\n %d",&n);
        while((n<1)||(n>4))
        {
             write_string(40,21," There are only 4 students;[1-4] ",0x35);
             write_string(5,8," Enter student Number again[1-4]>>",0x35);
             window_xy(5,40,8);
             scanf("\n %d",&n);
             }
       no=n-1;
       write_string(40,21," Wait for Fileconnect..",0x35);
       write_string(65,21,"          ",0x11);
       settx(term[no].txadd);
       setrx(term[no].rxadd);
        xmt_ch(FILERQ);
        timeout=0;
       ftime(&t);time1=t.time+t.millitm/1000.0;
        while(rcv_ch()!=FILECONF)
            {
                xmt_ch(FILERQ);
                ftime(&t);time2=t.time+t.millitm/1000.0;
                timepast=time2-time1;
                if(timepast>5) {timeout=1;break;}
                }
         if(timeout==0)
         {
             window_cls(5);
             sendf(fname,fp);
             window_cls(5);
             restore_video(5);
             data=0;
             }
         else {write_string(40,21," Timeout Failure! Press any key ",0x35);
                data=0;
               getch();
                 }

           }/*out data*/
      }
    else {write_string(40,21," Cannot open input file ",0x35);
          getch();
           }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }

void sendf(unsigned char name[20],FILE *fp)
{
   int p,j;
   char k;
   union {
           char x[2];
           unsigned int t;
           }ct;

    j=strlen(name);
    ct.t=filesize(fp);
    write_string(3,2,name,0x35);
    while(rcv_ch()!='.')
    {
      xmt_ch(SYN);
    }
    write_string(3,3," syn..",0x30);
    xmt_ch(j);/*******send size of file's name**********/
    wait();
    for(p=0;p<j;p++)     /*****send file's name******/
     {
       xmt_ch(name[p]);
       wait();
     }
   xmt_ch(ct.x[0]);
   wait();
   xmt_ch(ct.x[1]);
   wait();
   window_xy(10,3);
   for(p=1;p<=ct.t;p++)
   {
       k=getc(fp);
       xmt_ch(k);
       window_putchar(5,k);
       wait();
   }
   fclose(fp);
   write_string(40,21," Finish sending file ",0x30);
   getch();
}

void recf()
{
   FILE *fp;
   int p,j,n;
   char c;
   unsigned char name[20] = {"                    "};
   union {
          char x[2];
          unsigned int c;
           }ct;
   write_string(3,2," please wait..",0x30);
   xmt_ch(SYN);

   while(rcv_ch()!=SYN);    /*********syn***********/
         xmt_ch('.');
   write_string(15,2,"syn",0x30);
   while(rcv_ch()==SYN);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    j=rcv_ch();
    xmt_ch('.');
    for(p=0;p<j;p++)
     {   while(rcv_check()!=RCVRDY);
           name[p]=rcv_ch();
           xmt_ch('.');
     }
     write_string(3,3,name,0x30);
    name[p]=NULL;
    remove(name);
    fp=fopen(name,"wb");
    if(fp==NULL)
     {  write_string(40,21,"Cannot open file",0x30);
        return;
     }
   while(rcv_check()!=RCVRDY);
   ct.x[0]=rcv_ch();
   xmt_ch('.');
   while(rcv_check()!=RCVRDY);
   ct.x[1]=rcv_ch();
   xmt_ch('.');
   window_xy(3,4);
   for(n=1;n<=ct.c;n++)
   {
        while(rcv_check()!=RCVRDY);
        c=rcv_ch();
        window_putchar(5,c);
        putc(c,fp);
        xmt_ch('.');
   }
   fclose(fp);
    write_string(40,20," Finish receiving file ",0x30);
    write_string(40,21,name,0x30);
    gotoxy(60,22);
    printf("\nfilesize=%d\n",ct.c);
    getch();
}
 wait()
{
       while(rcv_check()!=RCVRDY)   ;
       while(rcv_ch()!='.');
}
unsigned int filesize(FILE  *fp)
{
  int i=0;
  do{
    getc(fp);
    i++;
  }while(!feof(fp));
  rewind(fp);
  return (i-1);
}

/*-----------------------------------------------------*/
 int rcv_check()  /**/
 {
   return(bioscom(STATUS,BYTE,COM2)&RCVRDY);
 }
 int rcv_ch()
 {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    return(inportb(RX));
  }
  int xmt_ch(ch)
    char  ch;
  {
    while((bioscom(STATUS,BYTE,COM2)&XMTRDY) == 0);
    outportb(TX,ch);
  }
  void initport()
  {
    outportb(515,0x90);/* A:in,B:out,C-upper:out ADD,C-lower:stop rq*/
    outportb(514,0xff);
  }
  void settx(port) /*set port to send data to a sl*/
     int port;
  {
    outportb(513,port);/*outprtB sl add*/
   }
  void setrx(port)/*set master rx to rec data from a selected sl*/
    int port;
   {
     outportb(514,port);
     }
  int detect(void) /*if >=0get a single rq ;-1 ifnot*/
  {  int dnode,z;
     dnode=-1;
     if((z=inportb(512))!=0x00)
        {
            switch(z)
            {
              case 0x0e :
               dnode=0;
               break;
              case 0x0d :
               dnode=1;
               break;
              case 0x0b :
               dnode=2;
               break;
              case 0x07 :
               dnode=3;
               break;
              default:
                break;
               }
            }
            return (dnode);
       }


/*-------------------------------------------------------*/

make_window(num,header,startx,starty,endx,endy,border)
 int num;
 char *header;
 int startx,starty;
 int endx,endy;
 int border;
 {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    unsigned  char *p;
    if(num>MAX_FRAME)
      {
          printf("Too many windows\n");
          return 0;
      }
     if((startx>79) || (startx<0) || (starty>24) || (starty<0))
      {
       printf("rang error");
       return 0;
      }
     if((endx>79) || (endy>24))
      {
          printf("window won't fit ");
          return 0;
      }
    p=(unsigned char*)malloc(2*(endx-startx+1)*(endy-starty+1));
    if(!p) exit(1);
    frame[num].startx = startx;  frame[num].endx = endx;
    frame[num].starty = starty;  frame[num].endy = endy;
    frame[num].p      = p;
    frame[num].header = header;
    frame[num].border = border;
    frame[num].active = 0;
    return 1;
}
make_menu_y(
  int num,       /* menu number */
  char *menu[], /* menu text */
  char *keys,    /* hot keys */
  int count,     /* number of menu items */

  int x, int y, /* X,Y coordinates of left hand corner */
  int border     /* no border if 0 */
)
{
  register int i, len;
  int endx, endy;
  unsigned char *p;

  if(num>MAX_FRAME) {
    printf("Too many menus\n");
    return 0;
  }
  if((y>24) || (y<0) || (x>79) || (x<0)) {
    printf("range error");
    return 0;
  }
  /* compute the size */
  len = 0;
  for(i=0; i<count; i++)
   if(len<strlen(menu[i])) len = strlen(menu[i]);
  endx =  x+len+3;
  endy =  count+1+y;
  if((endy+1>25) || (endx+1>80)) {
    printf("menu won't fit");
    return 0;
  }
  p = (unsigned char *) malloc(2 * (endx-x+1) * (endy-y+1));
  if(!p) exit(1); /* put your own error handler here */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    /* construct the frame */
    frame[num].startx = x; frame[num].endx = endx;
    frame[num].starty = y; frame[num].endy = endy;
    frame[num].p = p;
    frame[num].menu = (char **) menu;
    frame[num].border = border;
    frame[num].keys = keys;
    frame[num].count = count;
    frame[num].active = 0;
    return 1;
}

make_menu_x(num,menu,keys,count,x,y,border)
int num;
char *menu[];
char *keys;
int count;
int x,y;
int border;
{
     register int i,len;
     int endx,endy,choice,vmode;
     unsigned char *p;
     if(num >MAX_FRAME)
     {
        printf("Too many menus !\n");
        return 0;
     }
     if((y>24)||(y<0)||(x>79)||(x<0))
     {
       printf("Rang menu error !\n");
       return 0;
     }
     len = 0;
     for(i=0;i<count;i++)
     if(len<strlen(menu[i]))
       len=strlen(menu[i]);
     endy = y;
     endx = 80;
     if((endy>25)||(endx>80))
     {
        printf("Menu won't fit \n");
        return 0;
     }
     p = (unsigned char*)malloc(2*(endx-x+1)*(endy-y+1));
     if(!p)
       exit(1);
       frame[num].startx = x;
       frame[num].starty = y;
       frame[num].endx   = endx;
       frame[num].endy   = endy;
       frame[num].p      = p;
       frame[num].menu   = (char**) menu;
       frame[num].border = border;
       frame[num].keys   = keys;
       frame[num].count  = count;
       frame[num].active = 0;
     return 1;
}
/*-------------------------------------------------------------*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   void windows(num)
   int num;
   {
        int vmode,choice;
        int x,y;
        vmode = video_mode();
        if((vmode!=2)&&(vmode!=3)&&(vmode!=7))
        {
            printf("video must be in 80 column text mode ");
            exit(1);
        }
        if(vmode==7) vid_mem = (char far *) 0xB0000000;
        else vid_mem = (char far*) 0xB8000000;
        if(!frame[num].active)
        {
             save_video(num);
             frame[num].active = 1;
        }
        if(frame[num].border) draw_border(num);
        display_header(num);
        x = frame[num].startx + 1;
        y = frame[num].starty + 1;
        goto_xy(x,y);
   }
/* Display a pull-down menu and return selection. */
int pulldown_y(int num)
{
   int  vmode,choice;

   vmode = video_mode();
   if((vmode!=2) && (vmode!=3) && (vmode!=7)) {
     printf("video must be in 80 column text mode");
     exit(1);
   }
   /* set proper address of video RAM */
   if(vmode==7) vid_mem = (char far *) 0xB0000000;
   else vid_mem = (char far *) 0xB8000000;

   /* get active window */
   if(!frame[num].active) { /* not currently in use */
     save_video(num);       /* save the current screen */
     frame[num].active = 1; /* set active flag */
   }
   if(frame[num].border) draw_bordery(num);
   display_menu_y(num);     /* display the menu */
   return get_resp_y(num); /* return response */
}

 int pulldown_x(num)
 int num;
 {
    int vmode,choice;
    vmode = video_mode();
    if((vmode!=2)&&(vmode!=3)&&(vmode!=7))
    {
      printf("video must be in column 80");
      exit(1);
    }
    if(vmode==7) vid_mem = (char far *) 0xB0000000;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
   register int i;
   char far *v, far *t;

   v = vid_mem;
   t = v;
   /* draw vertical lines */
   for(i=frame[num].starty+1; i<frame[num].endy; i++) {
       v += (i*160) + frame[num].startx*2;
       *v++ = 179;
       *v = 0x1E;
       v = t;
       v += (i*160) + frame[num].endx*2;
       *v++ = 179;
       *v = 0x1E;
       v = t;
   }
   /* draw horizontal lines */
   for(i=frame[num].startx+1; i<frame[num].endx; i++) {
       v += (frame[num].starty*160) + i*2;
       *v++ = 205;
       *v = 0x1E;
       v = t;
       v += (frame[num].endy*160) + i*2;
       *v++ = 196;
       *v = 0x1E;
       v = t;
   }
   /* draw corners */
   write_char(frame[num].startx, frame[num].starty,(char) 213, 0x1E);
   write_char(frame[num].startx, frame[num].endy,(char) 192, 0x1E);
   write_char(frame[num].endx, frame[num].starty,(char) 184, 0x1E);
   write_char(frame[num].endx, frame[num].endy,(char) 217, 0x1E);
}
void draw_bordery(num)
int num;
{
    register int i,j,k,l;
    char far *v,far *t;
     v = vid_mem;
     t=v;
     for(k=frame[num].startx;k<frame[num].endx+1;k++)
     {
            for(l=frame[num].starty;l<frame[num].endy+1;l++)
            {
               v += l*160+(k*2);
               *v++ = 0;
               *v = 0x71;
               v = t;
            }
     }
     for(i=frame[num].startx+1;i<frame[num].endx+1;i++)
     {
            v += frame[num].starty*160+(i*2);
            *v++ = 196;
            *v    = 0x70;
             v    = t;
             v += frame[num].endy*160+(i*2);
             *v++ = 196;
             *v = 0x70;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                v = t;
        }
        for(i=frame[num].starty+1;i<frame[num].endy+1;i++)
        {
                v += i*160+(frame[num].startx*2);
                *v++ = 179;
                *v = 0x70;
                v = t;
                v +=(i*160)+frame[num].endx*2;
                *v++ = 179;
                *v = 0x70;
                v = t;
        }
       write_char(frame[num].startx,frame[num].starty,218,0x70);
       write_char(frame[num].startx,frame[num].endy,192,0x70);
       write_char(frame[num].endx,frame[num].starty,191,0x70);
       write_char(frame[num].endx,frame[num].endy,217,0x70);
   }
void draw_borderx(num)
int  num;
{
   register int i,j,k,l;
   char far *v,far *t;

   v = vid_mem;
   t = v;
   for(k=0;k<80;k++)
   {
      for(l=frame[num].starty;l<frame[num].endy+1;l++)
      {
        v += l*160+(k*2);
        *v++ = 0;
        *v = 0x71;
           v = t;
       }
     }
}
/*-------------------------------------------------------------*/
   /* Draw for zone window*/
   drawzone()
{

    draw(10,3,25,5);
    draw(25,3,40,5);
    draw(40,3,55,5);
    draw(55,3,70,5);
    draw(10,5,25,10);
    draw(25,5,40,10);
    draw(40,5,55,10);
    draw(55,5,70,10);
    write_char(10,3,213,0x1B);
    write_char(25,3,209,0x1B);
    write_char(40,3,209,0x1B);
    write_char(55,3,209,0x1B);
    write_char(70,3,184,0x1B);
    write_char(10,5,198,0x1B);
    write_char(25,5,216,0x1B);
    write_char(40,5,216,0x1B);
    write_char(55,5,216,0x1B);
    write_char(70,5,181,0x1B);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    write_char(10,10,192,0x1B);
    write_char(25,10,193,0x1B);
    write_char(40,10,193,0x1B);
    write_char(55,10,193,0x1B);
    write_char(70,10,217,0x1B);
    write_char(18,4,65,0x1C);
    write_char(33,4,66,0x1C);
    write_char(48,4,67,0x1C);
    write_char(63,4,68,0x1C);

}
void  draw(startx,starty,endx,endy)
{
  register int i;
  char far *v, far *t;

  v = vid_mem;
  t = v;

   for(i=starty+1;i<endy;i++){
      v += (i*160) + startx*2;
      *v++ = 179;
      *v = 0x1B;
      v = t;
      v += (i*160) + endx*2;
      *v++ =179;
      *v = 0x1B;
      v = t;
   }
   for(i=startx+1;i<endx;i++){
         v += (starty*160)+i*2;
         *v++ = 205;
         *v = 0x1B;
         v = t;
         v += (endy*160)+i*2;
         *v++ = 196;
         *v = 0x1B;
         v = t;
   }
}
/*-------------------------------------------------------*/
/* draw for menu watch*/
void draw_tree(startx,starty)
{
  register int k,i,j,x,y,row;
  char far *v,far *t;


  v = vid_mem;
  t = v;
  for(j=0;j<4;j++)
    {
    for(i=0; i<4;i++)
     {
      x = (startx-2)+i*20;
      y = (starty+2)+j*4;
      k = VALUE[j][i];
        window_xy(4,x,y);
        if(LINK[k]==1)
          write_string(x,y,"ON",0x1E);
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    draw(startx,starty,endx,endy);
    write_char(startx+7,endy,194,0x1B);
    write_char(startx,starty,213,0x1B);
    write_char(startx,endy,192,0x1B);
    write_char(endx,starty,184,0x1B);
    write_char(endx,endy,217,0x1B);
    write_string(startx+4,starty+1,"master",0x1C);
 }
/*-----------------------------------------------------------------*/
/* Input user's selection */
get_resp_y(int num)
{
  union inkey {
    char ch[2];
    int i;
  } c;
  int arrow_choice=0;
  int key_choice;
  int x, y;

  x = frame[num].startx+2;
  y = frame[num].starty+1;
  /* highlight the first selection */
  write_string(x, y+arrow_choice, frame[num].menu[0],0x30 );
  for(;;)
  {
    while(!bioskey(1));
    c.i = bioskey(0);     /* read the key */

    /* reset the selection to normal video */
    write_string(x,y+arrow_choice,frame[num].menu[arrow_choice],REV_VID); /* redisplay */

    if(c.ch[0]) { /* is normal key */
      /* see if it is a hot key */
      key_choice = is_in(frame[num].keys, tolower(c.ch[0]));
      if(key_choice) return key_choice -1 ;

      /* check for ENTER or space bar */
      switch(c.ch[0]) {
        case '\r': return arrow_choice;
        case ' ' : arrow_choice++;
          break;
        case ESC : return -1; /* cancel */
      }
    }
    else {  /* is special key */
      switch(c.ch[1]) {
        case 72: arrow_choice--; /* up arrow */
          break;
        case 80: arrow_choice++; /* down arrow */
          break;
      }
    }
    if(arrow_choice==frame[num].count) arrow_choice=0;
    if(arrow_choice<0) arrow_choice = frame[num].count-1;

    /* highlight the next selection */
    write_string(x,y+arrow_choice,
      frame[num].menu[arrow_choice],0x38);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
get_resp_x(num)
int num;
 {
      union inkey{
                 char ch[2];
                 int  i;
                 } c;
      int arrow_choice = 0, key_choice;
      int x,y;
      x = frame[num].startx+4;
      y = frame[num].starty;


      write_string(x,y,frame[num].menu[0],0x30);
      for(;;)
      {
             while(!bioskey(1));
             c.i = bioskey(0);
             write_string(x+(12*arrow_choice),y,frame[num].menu[arrow_choice],REV_VID);
             if(c.ch[0])
             {
                  key_choice = is_in(frame[num].keys,tolower(c.ch[0]));
                  if(key_choice) return key_choice -1;
                  switch(c.ch[0])
                  {
                       case '\r' : return arrow_choice;
                       case ' '  : arrow_choice++;
                                   break;
                       case ESC  : return -1;
                  }
             }
             else
             {
                       switch(c.ch[1])
                       {
                                case 75 : arrow_choice--;
                                          break;
                               case 77 : arrow_choice++;
                                          break;
                       }
               }
              if(arrow_choice==frame[num].count) arrow_choice = 0;
              if(arrow_choice<0)arrow_choice=frame[num].count-1;
              write_string(x+(12*arrow_choice),y,frame[num].menu[arrow_choice],0x38);
              }
     }
int get_res_zone(num)
int num;
{
  int active;
  char ans;
  union inkey{
       char ch[2];
       int i;
       } c;
  for(;;)
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

/* Restore a portion of the screen. */
void restore_video(int num)
{
   register int i,j;
   char far *v, far *t;
   char *buf_ptr;

   buf_ptr = frame[num].p;
   v = vid_mem;
   t = v;
   for(i=frame[num].startx; i<frame[num].endx+1; i++)
     for(j=frame[num].starty; j<frame[num].endy+1; j++) {
       v = t;
       v += (j*160) + i*2; /* compute the address */
       *v++ = *buf_ptr++;  /* write the character */
       *v = *buf_ptr++;    /* write the attribute */
   }
   frame[num].active = 0; /* deactivate */
   /*free(frame[num].p);*/
}
/* returns the current video mode */
video_mode(void)
{
  union REGS r;
  r.h.ah = 15;  /* get video mode */
  return int86(0x10, &r, &r) & 255;
}
is_in(s,c)
char *s,c;
{
    register int i;
    for(i=0;*s;i++)
       if(*s++==c) return i+1;
       return 0;
}

void goto_xy(x,y)
int x,y;
{
   union REGS r;
   r.h.ah = 2;
   r.h.dl = x;
   r.h.dh = y;
   r.h.bh = 0;
   int86(0x10,&r,&r);
}
window_xy(num,x,y)
int num,x,y;
{
  if(x<0|| x+frame[num].startx>=frame[num].endx-1)
   return 0;
  if(y<0 || y+frame[num].starty>=frame[num].endy-1)
   return 0;
   frame[num].curx = x;
   frame[num].cury = y;
   /*write_char(frame[num].curx,frame[num].cury,95,0x1B);*/
   return 1;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************************************************************************/
/************** function I/O for windows *************************************/
/*****************************************************************************/


window_puts(num,str)
int num;
char *str;
{
   if(!frame[num].active) return 0;
   for(; *str;str++)
    window_putchar(num,*str);
    return 1;
}


window_putchar(num,ch)
int num;
char ch ;
{
  register int x,y;
  char far *v;

  if(!frame[num].active) return 0;
   x = frame[num].curx + frame[num].startx +1;
   y = frame[num].cury + frame[num].starty +1;

   v = vid_mem;
   v += (y*160) + (x*2);
   if((ch == '\r') || (x >= frame[num].endx))
   {
      if((y < frame[num].endy-1))
      {
y++;
x = frame[num].startx+1;
v = vid_mem;
v += (y*160)+ x*2;
frame[num].cury++;
frame[num].curx =0;
}
      else
      {
  save(num);
  restore(num);
frame[num].curx = 0;
window_xy(num,frame[num].curx,frame[num].cury);
}
    }
  else
  {
    frame[num].curx++;
    *v++ = ch;
    *v++ = 0x1B;
    }
  window_xy(num,frame[num].curx,frame[num].cury);
  return 1;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    void save(int num)
    {   register int i,j;
        char *buf_ptr;
        char far *v,far *t;
        buf_ptr =frame[num].p;
        v=vid_mem;
        for(i=frame[num].starty+2;i<frame[num].endy;i++)
             for(j=frame[num].startx+1;j<frame[num].endx;j++)
             {
               t=(v+(i*160)+j*2);
               *buf_ptr++ = *t++;
               *buf_ptr++ = *t;
               }
       }
    void restore(int num)
    {
      register int i,j;
      char far *v,far *t;
      char *buf_ptr;
      buf_ptr=frame[num].p;
      v=vid_mem;
      t=v;
      for(i=frame[num].starty+1;i<frame[num].endy;i++)
         for(j=frame[num].startx+1;j<frame[num].endx;j++)
         {
             v=t;
             v +=(i*160)+j*2;
             *v++ =*buf_ptr++;
             *v   =*buf_ptr++;
             }
   }

  void window_gets(num,s)
  int num;
  char *s;
  {
    char ch,*temp;
    temp=s;
    for(;;)
    {
    ch = window_getche(num);
    switch(ch)
    {
case '\r':
     *s ='\0';
     return ;
case BKSP :
     if(s>temp)
     {
        s--;
        frame[num].curx--;
        if(frame[num].curx<0) frame[num].curx = 0;
 window_xy(num,frame[num].curx,frame[num].cury);
        write_char(frame[num].startx+frame[num].curx+1,
 frame[num].starty+frame[num].cury+1,
 ' ',0x1B);
        }
    break;
default :
     *s = ch;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        window_xy(num,frame[num].curx,frame[num].cury);
        window_putchar(num,ch);
        s++;
        break;
}
        }
   }

 window_getche(num)
 int num;
 {
      union inkey {
    char ch[2];
    int i;
    }c;
    if (!frame[num].active) return 0;
    window_xy(num,frame[num].curx,frame[num].cury);
    c.i = bioskey(0);
return c.i;
      }

void window_cls(num)
int num;
{
   register int i,j;
   char far *v,far *t;

   v = vid_mem;
   t = v;
   for(i=frame[num].starty+1; i<frame[num].endy;i++)
     for(j=frame[num].startx+1;j<frame[num].endx;j++)
     {
       v = t;
       v +=(i*160) +j*2;
       *v++ = ' ';
       *v = 0x10;
       }
    frame[num].curx = 0;
    frame[num].cury = 0;
 }

void window_cleol(num)
int num;
{
  register int i,x,y;
  x = frame[num].curx;
  y = frame[num].cury;
  window_xy(num,frame[num].curx,frame[num].cury);
    for (i = frame[num].curx; i<frame[num].endx-1; i++)
      window_putchar(num,' ');
    window_xy(num,x,y);
}


window_upline(num)
int num;
{
if (frame[num].cury >0){
    frame[num].cury--;
     window_xy(num,frame[num].curx,frame[num].cury);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        return 1;
        }
        return 0;
    }


window_downline(num)
int num;
{
    if(frame[num].cury<frame[num].endy-frame[num].starty-1){
        frame[num].cury++;
        window_xy(num,frame[num].curx,frame[num].cury);
        return 1;
     }
     return 0;
}
window_shiftleft(num)
int num;
{
 if(frame[num].curx>0)
 {
    frame[num].curx--;
    window_xy(num,frame[num].curx,frame[num].cury);
    return 1;
 }
    return 0;
}
window_shiftright(num)
int num;
{
    if(frame[num].curx<frame[num].endx-frame[num].startx-1)
     {
     frame[num].curx++;
     window_xy(num,frame[num].curx,frame[num].cury);
     return 1;
     }
     return 0;
}

window_bksp(num)
int num;
   {
   if(frame[num].curx != 0)
      {
      frame[num].curx--;
      window_xy(num,frame[num].curx,frame[num].cury);
      window_putchar(num, ' ');
      frame[num].curx--;
      window_xy(num,frame[num].curx,frame[num].cury);
        }
   else
      {
      frame[num].cury--;
      frame[num].curx = frame[num].endx-4;
      window_xy(num,frame[num].curx,frame[num].cury);
      window_putchar(num,' ');
      frame[num].curx--;
      window_xy(num,frame[num].curx,frame[num].cury);
   }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  main(void)
{
   int p,o;
   gotoxy(0,0);
   bkcolor(0x11);

   /* first, create the menu frames */
   make_window(0, "WINDOW 0",1, 1,79, 23, BORDER);
   make_window(1," Broadcast_window ",1,1,79,23,BORDER);
   make_window(5,"FILE",1,1,79,23,BORDER);
   make_window(6,"CHAT",1,1,79,23,BORDER);
   make_window(7,"",1,1,79,3,BORDER);
   make_window(8,"LOCAL",2,4,78,13,BORDER);
   make_window(9,"REMOTE",2,14,78,23,BORDER);
   make_menu_y(10, answer1, "oc",2,35,17,BORDER);
   make_menu_x(11, answer2, "cfq",3,0,0,BORDER);



   pd_driver(); /* activate the menu system */
   textattr(7);


   clrscr();

}

/* Demonstrate the pull-down menu functions. */
void pd_driver(void)
{
  FILE *fp;
  union inkey {
          char ch[2];
          int i;
          }x;
  int q,p,r,s,set=1;
  int ans1,ans2;
  int fileflag,chatflag;
  unsigned char name[20]={"                    "};
  char c;
     clrscr();
     bioscom(SETS,BYTE,COM2);
     windows(0);
     q=0;
while(q==0)
{
   write_string(35,5,"STUDENT",0x1B);
   write_string(28,8,"+CLASSMATE SOFTWARE+",0x1B);
   write_string(35,11,"KMIT'L",0x1B);
   write_string(34,14,"EXECUTE?",0x1B);

   while((ans1=pulldown_y(10)) != -1)
 {
    if(ans1==0)
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
write_string(frame[10].startx+2,frame[10].starty+ans1+1,frame[10].menu[ans1],0x30);
    restore_video(10);
    windows(1);
    chatflag=0;
    fileflag=0;
    p=1;
    while(p==1)
    {
    write_string(40,23," Press F10 to interrupt broadcasting ",0x30);
        if(!kbhit())
                {
                   if(rcv_check()==RCVRDY)
                   switch(c=rcv_ch())
                   {case CHATRQ:
                    while(rcv_ch()==CHATRQ) xmt_ch(CHATCONF);
                       window_inchat();/*inchatwin */
                       break;
                    case CHATCONF:
                       if(chatflag==1)
                             window_inchat();
                       chatflag=0;
                       break;
                    case FILERQ:
                    while(rcv_ch()==FILERQ) xmt_ch(FILECONF);
                       windows(5);
                       recf();/*recfwin*/
                       window_cls(5);
                       restore_video(5);
                       break;
                    case FILECONF:
                       if(fileflag==1)
                       {
                            windows(5);
                            sendf(name,fp);
                            window_cls(5);
                            restore_video(5);}
                       fileflag=0;
                       break;
               case '\b':
                       window_bksp(1);
                       break;
                   case DISC:
                       p=0;
                     break;
                   default:
                     window_putchar(1,c);
                     break;
                    }/*end sw*/
                }
  else/*kbhit*/
  {
     x.i=window_getche(1);

    if(x.ch[1]==68)
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                set=1;
                while(set==1&&(ans2=pulldown_x(11))!= -1)
                {
                write_string(frame[11].startx+4+(ans2*12),frame[11].starty,frame[11].menu[ans2
30);
                write_string(40,23," Press ESC to activate Broadcasting ",0x30);
                switch(ans2)
                {
                      case 0: /*chat*/
                           outportb(TX,ENQ);
                           xmt_ch(CHATRQ);
                           write_string(40,23," Send CHAT RQ ",0x30);
                           chatflag=1;
                           set=-1;
                           break;
                     case 1:/*file*/
                           windows(5);
                           write_string(19,3,"** You are welcome to FILETRANSFER MODE **",0x34)
                           write_string(5,6," Enter filename[20]>>",0x35);
                           window_xy(5,28,4);
                           window_gets(5,name);
                           if((fp=fopen(name,"rb"))!= NULL)
                             {
                                 outportb(TX,ENQ);
                                 xmt_ch(FILERQ);
                                 fileflag=1;
                                 }
                            else {write_string(40,21," Cannot open input file ",0x35);
                                       getch();
                                       fileflag=0;
                                         }
                           window_cls(5);
                           restore_video(5);
                           set=-1;
                           break;
                     case 2: /*QUIT*/
                            set=0;
                            p=0;
                            break;
                   }  /*end of switch*/
                 } /*end of loop while ans2*/
               restore_video(11);
             }/*else c.ch[1]!=60*/
           } /*end elsekbhit*/
         }  /*end of loop p=0*/
         restore_video(1);
     }   /*end if ans=0*/
     if(ans1==1)
          break;
   }   /* end of loop while ans1*/
   window_cls(10);
   restore_video(10);
   q = 1;
}    /* end of loop q=0 */
restore_video(0);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. PC/XT Personal computer XT System Technical Reference
2. ศิวัฒน์ ศิวะบวร, พรชัย จักร์ธำรงค์ และ จิระศักดิ์ ชัยวิริยะกุล, " การประยุกต์ใช้โปรแกรมภาษาซี " หน้า 1-80,ซีเอ็ดยูเคชั่น
3. JAMES W.Coffron, " Z80 Application ",SYBEX Inc
4. ไบรอัน ดับบลิว เคอร์นิกแฮน และ เดนนิส เอ็ม ริตชี เรียบเรียงโดย ดร. วิทยา วัชระวิทยากุล, " ภาษา และ การโปรแกรม C "

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้