โครงข่ายควบคุมจากศูนย์กลาง

CENTRAL CONTROLLED NETWORK

โดย
นายกิตติเดช นิ้มวงษ์เจริญสุข
นายชูศักดิ์ วิจิตสรัตน์
นายสราวุธ ชัยแป้น

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต
สาขาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา ๒๕๓๖

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

033265

ปริญญานิพนธ์
ประจำปีการศึกษา ๒๕๓๖
เรื่อง โครงข่ายควบคุมจากศูนย์กลาง
(Central Controlled Network)

ภาควิชา วิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า
เจ้าคุณทหาร ลาดกระบัง

ผู้จัดทำ

1. นายกิตติเดช นิ้มวงษ์เจริญสุข 33100025
2. นายชูศักดิ์ วิจิตสรัตน์ 33100095
3. นายสราวุธ ชัยแป้น 33100408

---------- อาจารย์ที่ปรึกษา
(ดร. สุวิพล สิทธิชีวภาค)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงข่ายควบคุมจากศูนย์กลาง

CENTRAL CONTRLLED NETWORK

ผู้จัดทำ นายกิตติเดช นิ้มวงษ์เจริญสุข
นายชูศักดิ์ วิจิตสรัตน์
นายสราวุธ ชัยแป้น
อาจารย์ที่ปรึกษา ดร.สุวิพล สิทธิชีวภาค
ปีการศึกษา 2536

บทคัดย่อ

งานวิจัยนี้เป็นการศึกษาวิจัยเกี่ยวกับ โครงข่ายควบคุมโดยศูนย์กลาง (CENTRAL CONTROLLED NETWORK) ซึ่งจะประกอบไปด้วยส่วนที่สำคัญ 2 ส่วน ซึ่งส่วนแรก ได้แก่ ส่วนที่เป็นฮาร์ดแวร์ (Hardware) และ ส่วนที่สองเป็นซอฟท์แวร์ (Software)ในส่วนของฮาร์ดแวร์นั้นได้แบ่งการทำงานออกเป็น3 ส่วนย่อยๆ คือ ส่วนตรวจจับสัญญาณการ ร้องขอ ซึ่งเป็นการตรวจจับรูปแบบของสัญญาณ ENQ ส่วนประมวลผล โดยใช้ไอซีเบอร์ 8255 ซึ่งจะรับข้อมูลจากส่วนตรวจจับเข้ามาทางพอร์ต A แล้วส่ง ผลการตรวจจับออกทางพอร์ตC ด้านล่าง เพื่อไปทำการควบคุมระบบการทำงานของส่วนที่ 3 ซึ่งคือ ส่วนการสวิตชิ่ง (Switching) ซึ่งก็จะทำการสวิตช์ เทอร์มินัลที่ได้รับการคัดเลือกเชื่อมต่อเข้ากับเซนเตอร์สำหรับส่วนของซอร์ฟแวร์นั้นแบ่งได้ 2 ส่วน คือ ส่วนที่ใช้สำหรับการควบคุมการทำงานของฮาร์ดแวร์ (hardware) และส่วนของโปรแกรมประยุกต์ (Application) โดยที่ส่วนของการควบคุมจะเป็นการปรับค่าของพารามิเตอร์ในการสื่อสารและการเชื่อมต่อ และส่วนของโปรแกรมประยุกต์นั้นเน้นการตอบสนองข้อมูลที่ เทอร์มินอล ต้องการ

ABSTRACT

This research concerns with CENTRAL CONTROLLED NETWORK .The system consists of 2 parts ;HARDWARE and SOFTWARE. Hardware is divided into 3 subparts. The first is data detecting part which uses to detect ENQ format. The second is a processing unit that utilized IC-8255. The last part is a switching unit which is controlled by second part. For software, which can seperated into 2 parts . One is used to control hardware and another is an application program. Controling hardware is setting communication parameter and swintching control. And application program is emphasized on Data enquirment.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เสารบัญ

หน้า

บทที่ 1 บทนำ....1
บทที่ 2 ทฤษฎีและหลักการ....3
2.1 รูปแบบการส่งข้อมูล....3
2.2 มาตราฐานการเชื่อมต่อ....15
2.3 การควบคุมการเชื่อมต่อข้อมูล....20
2.4 มีเดีย แอคเซส โปรโตคอล....22
2.5 โปรโตคอลสื่อสารอนุกรม....22
บทที่ 3 หลักการทำงานของฮาร์ดแวร์....25
บทที่ 4 วิธีการทำงานของ Network....37
4.1 เน็ตเวอร์ค โปรโตคอล....37
4.2 ขั้นตอนการสื่อสาร....40
บทที่ 5 หน้าที่และความสำคัญของโปรแกรม....42
5.1 หน้าที่ในการควบคุมชั้นรูทติ้งในตัวเซ็นเตอร์....42
5.2 หน้าที่ในการปรับค่าพารามิเตอร์ของการสื่อสารทั้งระบบให้ตรงกัน....43
5.3 หน้าที่ในการรับส่งข้อมูล....44
5.4 หน้าที่ในการบันทึกและการแสดงผล....45
บทที่ 6 ผลการทดลองและสรุป....58
แนวทางการพัฒนาต่อ....69
ภาคผนวก....70
หนังสืออ้างอิง....100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบัน เริ่มมีการนำเครื่องต่างชนิดกันมาต่อใช้งานร่วมกันในระบบเครือข่ายเดียวกันมากขึ้น ดังนั้นการจะนำเครื่องต่างๆเหล่านี้มาต่อใช้งานร่วมกันอย่างมีประสิทธิภาพ ผู้ออกแบบระบบควรมีความรู้ความเข้าใจเกี่ยวกับพื้นฐาน รวมทั้งข้อจำกัดต่างๆของระบบก่อน การนำเอาเครื่องคอมพิวเตอร์มาต่อกันเป็นระบบเครือข่ายจะช่วยการทำงานต่างๆในหน่วยงานได้ เช่น ในเรื่องของสำนักงานอัตโนมัติ (Office automation) ก็อาจจะนำระบบเครือข่ายคอมพิวเตอร์มาช่วยในการนัดหมายการประชุม การจัดตารางนัดหมายส่วนตัว และงานด้านการประมวลผลคำ (Word processing)ได้ โดยข้อมูลจะเก็บอยู่ในหน่วยเก็บข้อมูลที่เดียวกัน นอกจากนี้การพิมพ์ข้อมูลก็สามารถส่งงานพิมพ์ไปพิมพ์ที่เครื่องพิมพ์รวมของระบบได้ แทนที่จะต้องซื้อเครื่องพิมพ์ต่อกับเครื่องคอมพิวเตอร์ทุกๆเครื่อง

การต่อเครื่องคอมพิวเตอร์เป็นระบบเครือข่ายและเก็บข้อมูลไว้ที่ศูนย์กลาง นอกจากจะช่วยประหยัดค่าใช้จ่ายในการซื้ออุปกรณ์ราคาแพงๆ เช่น เครื่องพิมพ์แบบเลเซอร์ เพื่อต่อกับคอมพิวเตอร์ทุกเครื่องแล้ว ยังช่วยในการรักษาความปลอดภัยของข้อมูลอีกด้วย เช่น ข้อมูลระบบเงินเดือน อาจใช้เครื่องไมโครคอมพิวเตอร์ในการประมวลผล และใช้ระบบการรักษาความปลอดภัยของระบบเครือข่ายมาช่วยในการรักษาความปลอดภัยของข้อมูลได้

ซึ่งนอกเหนือจากด้านการรักษาความปลอดภัยของข้อมูลแล้วการนำระบบเครือข่ายคอมพิวเตอร์มาใช้ยังช่วยให้ได้ข้อมูลที่มีความถูกต้อง และทันต่อเหตุการณ์ด้วย เพราะข้อมูลถูกเก็บอยู่ที่ศูนย์กลาง ดังนั้นเมื่อข้อมูลถูกปรับปรุง ผู้อื่นในระบบเครือข่ายที่ได้รับอนุญาตให้ใช้งาน ก็สามารถเรียกข้อมูลส่วนนี้ได้ทันที

ในส่วนของโครงงานนี้ กลุ่มผู้จัดทำมีความคิดที่จะสร้างระบบโครงข่ายคอมพิวเตอร์ขึ้นโดยไม่จำเป็นต้องอาศัยระบบโครงข่ายที่มีขายอยู่ในท้องตลาดโดยได้ศึกษาทฤษฎีและรูปแบบต่างๆของระบบโครงข่ายที่มีใช้อยู่ทั่วไปเพื่อนำมาเป็นบรรทัดฐานในการสร้างโครงข่ายคอมพิวเตอร์ที่คิดขึ้น

ในส่วนรายละเอียดของหลักการและรูปแบบการทำงานของโครงข่ายนั้นจะอยู่ในบทต่างๆของวิทยานิพนธ์ฉบับนี้ สามารถกล่าวโดยสังเขปได้ดังนี้

ระบบโครงข่ายที่ใช้สำหรับโครงงานนี้ เป็นโครงข่ายที่เป็นแบบดาว (Star topology)กล่าวคือ โครงข่ายจะมีรูปร่างคล้ายดาวโดยจะมีตัวควบคุมการทำงานของระบบอยู่ตรงกลาง 1 ตัว และมีอุปกรณ์ปลายทางอยู่ทั้งหมด 8 ตัว เชื่อมต่ออยู่กับศูนย์กลางการควบคุมสำหรับโครงข่ายควบคุมจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(HUB) และมีอุปกรณ์ปลายทางซึ่งในที่นี้จะเรียกว่าเทอร์มินัล (Terminal) ซึ่งมีอยู่ 8 ตัว เชื่อมต่ออยู่โดยในโครงข่ายนี้ได้ใช้พีซีเป็น เทอร์มินัลด้วยเช่นกัน

การติดต่อสื่อสารของโครงข่ายนั้นสามารถแบ่งอย่างกว้าง ๆ เป็น 2 แบบ คือ

1. การติอต่อระหว่างเทอร์มินัล (Terminal) โดยผ่านฮับ (HUB)
2. การติดต่อระหว่างเทอร์มินัล (Terminal) กับฮับ (HUB)

ถ้าเรามองให้ลึกลงไปแล้วการติดต่อทั้งสองแบบ จะมีพื้นฐานเหมือนกันคือการสื่อสารกันระหว่างเทอร์มินัลกับฮับ เพราะฉะนั้นการอธิบายรูปแบบการทำงานของโครงข่ายทั้งหมด จึงขอกล่าวเพียงลักษณะการสื่อสารของเทอร์มินัลกับฮับเท่านั้น โดยใช้โปรโตคอล (Protocol) ในแบบที่คิดขึ้นเพื่อให้ทำงานร่วมกับฮาร์ดแวร์ (Hardware) ที่ทำขึ้น เพราะหากว่าเทอร์มินัลกับอับคู่หนึ่งสามารถติดต่อสื่อสารกันได้ ย่อมหมายความว่ามีความเป็นไปได้ที่โครงข่ายจะทำงานได้ตามต้องการ และส่วนที่เหลือก็จะทำหน้าที่คัดเลือกเทอร์มินัลที่จะติดต่อกับฮับ โดยการทำงานเหล่านี้จะใช้ฮาร์แวร์ที่เพิ่มเติมขึ้นมาร่วมกับอะแดปเตอร์สื่อสารแบบอนุกรม (Serial card) ของคอมพิวเตอร์ ซึ่งอุปกรณ์เหล่านี้จะถูกควบคุมโดยซอฟแวร์ (Software) อีกทีหนึ่งโดยส่วนฮาร์แวร์ที่เพิ่มเติมเข้ามา จะทำหน้าที่เป็นสวิทชิ่ง (Switching) โดยจะคัดเลือกเทอร์มินัลที่ต้องการติดต่อกับฮับ ซึ่งทุกเทอร์มินัลจะมีลักษณะการร้องขอการเชื่อมต่อในทฤษฎีของมีเดีย แอคเซส คอนโทรล (Mediam Access Control)แบบเซ็นทรัลลี่ คอนโทรล ดีมานด์ แอสไซน์เมนท์ เทคนิค (Centrally Control Demand Assignment Technique) และในส่วนของโปรโตคอล (Protocol) ที่ใช้ในโครงข่ายนี้ได้นำทฤษฎีแบบ ไบท์-โอเรียนโปรโตคอล (Byte-Oriented Protocol) มาประยุกต์ และนำมาใช้ใน โฟลคอนโทรล (FLOW CONTROL) แต่จะใช้เพียง 4 รหัสของแอสกี (ASCII) เท่านั้น และในส่วนของรายละเอียดของเนื้อหาทั้งหมดจะกล่าวถึงในบทต่อ ๆ ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎี และหลักการ

2.1 รูปแบบการส่งข้อมูล

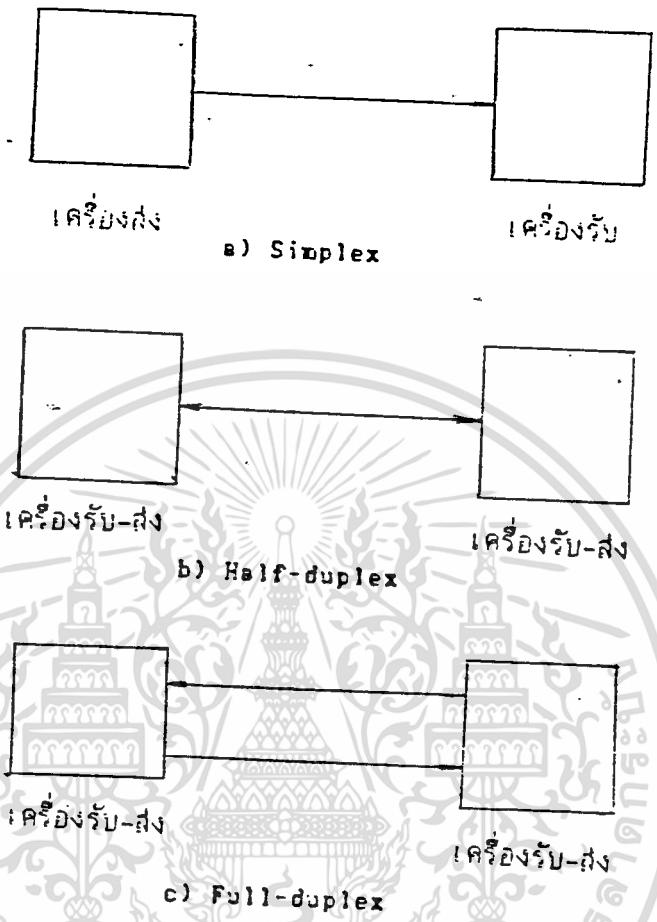
ในการส่งข้อมูลไปบนตัวกลางการสื่อสารนั้น ข้อมูลที่อยู่ในรูปของเลขฐาน_สองจะถูกเปลี่ยนให้เป็นสัญญาณทางไฟฟ้าก่อนแล้วจึงทำการส่งไปยังจุดรับ สำหรับวิธีการในการส่งข้อมูลนั้นสามารถจำแนกออกตามคุณสมบัติต่าง ๆ ได้หลายวิธีด้วยกันในที่นี้จะกล่าวถึงลักษณะทางไฟฟ้าที่ใช้แทนข้อมูลสำหรับการส่งวิธีการส่งข้อมูลจำแนกตามลักษณะทิศทางการส่ง จำแนกตามลักษณะการจัดข้อมูลในการส่ง จำแนกตามความสัมพันธ์ของข้อมูล เป็นต้น

2.1.1 การจำแนกวิธีการส่งข่าวสารตามทิศทางการส่งภายในสาย

ตัวกลางที่ใช้ในการสื่อสารข้อมูลอาจมีได้หลายชนิดแต่ตัวกลางที่นิยมใช้กันมากก็คือ สายสื่อสาร (Communication Line) ซึ่งตามปกติแล้วก็คือสายโทรศัพท์ (Telephone Line) โดยการส่งข้อมูลออกไปตามสาย เราสามารถพิจารณาตามทิศทางการส่งข้อมูลภายในสาย แล้วแบ่งการส่งข้อมูล 3 ชนิด คือ

1. การส่งแบบทิศทางเดียว (Simplex)
2. การส่งแบบทิศทางใดทิศทางหนึ่ง (half duplex)
3. การส่งแบบสองทิศทาง (Full-duplex)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 การส่งข้อมูลแบบจำแนกตามทิศทางการส่งภายในสาย

2.1.2 การจำแนกวิธีการส่งตามลักษณะการจัดข้อมูล

นอกจากการจำแนกวิธีการส่งตามลักษณะทิศทางการส่งแล้วยังสามารถจำแนกวิธีการส่งออกตามลักษณะการจัดข้อมูลที่ใช้ส่งได้อีกเป็น 2 วิธี คือ

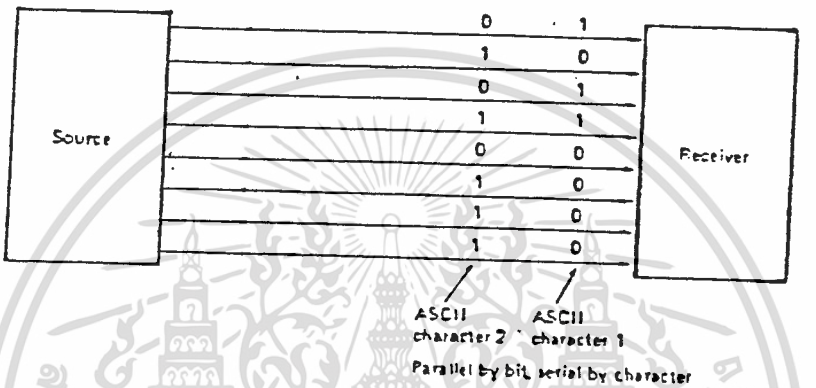
1) การส่งแบบขนาน (Parallel Transmission)

2) การส่งแบบอนุกรม (Serial Transmission)

การส่งข้อมูลแบบขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการส่งแบบขนานนั้น เราจะนำเอาทุก ๆ บิทของรหัสที่ประกอบเป็นอักษรหนึ่งตัวส่งออกไปพร้อม ๆ กันในเวลาเดียวกัน ซึ่งจะต้องใช้แซ็นแน็ล หรือ สายส่งเท่ากับจำนวนของบิทที่ประกอบเป็นอักษรหนึ่งตัว นั่นหมายความว่าหากเรามีรหัสขนาด 8 บิทก็ต้องมีแซ็นแน็ลสำหรับการส่งจำนวนเท่ากับ 8 แซ็นแน็ล ตัวอย่างเช่นในรหัส ASCII มีจำนวนบิทแทนตัวอักษรแต่ละตัวจำนวน 8 บิท เมื่อเราใช้วิธีการส่งแบบขนานต้องมีสายส่งทั้งหมด 8 เส้นทั้งทางด้านส่งและทางด้านรับ ดังรูปที่ 2.2



รูปที่ 2.2 การส่งข้อมูลแบบขนาน

การส่งข้อมูลแบบอนุกรม

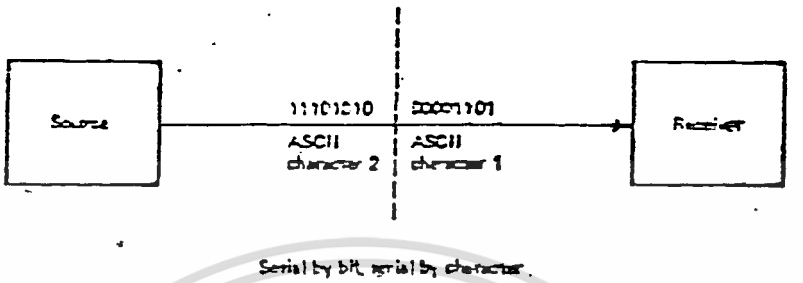
วิธีการส่งข้อมูลที่นิยมแพร่หลายมากที่สุดคือ การส่งข้อมูลแบบอนุกรมในการส่งแบบอนุกรมนั้นบิททั้งหมดของตัวอักษรหนึ่งตัวจะถูกนำมาส่งออกไปทีละบิทติดต่อกันไปตามแซ็นแน็ลซึ่งมีอยู่เพียงแซ็นแน็ลเดียว ดังรูปที่ 5.12 ทางด้านเครื่องรับเมื่อรับข้อมูลมาแล้วก็จะนำมาจัดเป็นตัอักษรขึ้นใหม่ ให้ตรงกับชุดของตัวอักษรที่ทางด้านส่งมา ซึ่งวิธีการดังกล่าวนี้จะต้องประกอบด้วยความสัมพันธ์ ในการทำงานระหว่างด้านรับ และด้านส่ง เพื่อให้สามารถได้ตัวอักษรที่ถูกต้องกลับมาใช้งาน

สำหรับการส่งข้อมูลในระบบการสื่อสารนั้น จะใช้วิธีการส่งแบบอนุกรมเพราะการใช้งานหลักของการสื่อสารคือการส่งข้อมูลในระยะทางไกล ๆ สำหรับการส่งข้อมูลแบบอนุกรมนั้น สิ่งหนึ่งที่จะต้องระมัดระวังคือ ความผิดพลาดเกี่ยวกับความสัมพันธ์ของบิทและของตัวอักขระที่ส่งเพราะเรานำเอาบิทของอักขระหลาย ๆ ตัวมาส่งเรียงกันมาออกทางด้านรับจะต้องสามารถรับบิทได้อย่างถูกต้อง และจะต้องแยกอักขระที่รับมาเป็นอักขระเดิมที่ทางด้านส่งออกมาได้การส่งข้อมูลแบบอนุกรมนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะใช้ฟอร์ทอนุกรม COM 1 และ COM 2 ในการติดต่อซึ่งมีมาตรฐานการติดต่อหลายรูปแบบเช่น

CCITT v.28, CCITT v.24 หรือ RS-232 แต่ที่นิยมแพร่หลายคือ RS-232



รูป 2.3 การส่งข้อมูลแบบอนุกรม

2.1.3 การจำแนกวิธีการส่งตามความสัมพันธ์ของข้อมูล

สำหรับวิธีการส่งข้อมูลที่จำแนกตามความสัมพันธ์ของข้อมูลที่ส่งนั้น เราอาจแบ่งออกได้เป็น 2 วิธี ดังรายละเอียดต่อไปนี้

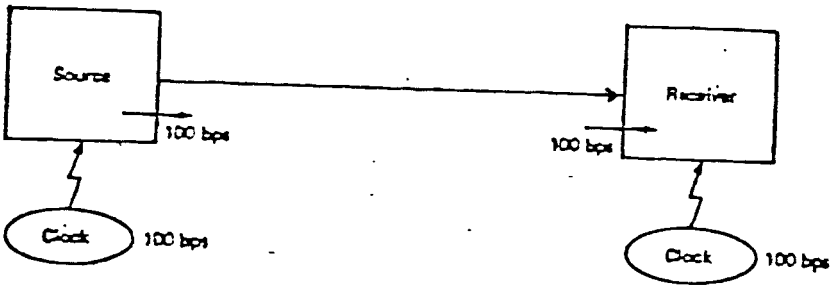
1) การส่งข้อมูลแบบสัมพันธ์ (Synchronous transmission)

สำหรับความสัมพันธ์ระหว่างข้อมูลที่เราส่งในระบบการสื่อสารนั้น ราจะต้องิจารณาถึงความสัมพันธ์สองชนิดคือ ความสัมพันธ์ของบิท (Bit Synchronization) ที่ประกอบกันเป็นอักขรหนึ่งตัว และความสัมพันธ์ของอักขระ (Character Synchronization) ที่ประกอบกันเป็นบล็อคในการส่งที่เราจะพิจารณาในตอนนี้

ความสัมพันธ์ของบิท

สำหรับความสัมพันธ์ของบิท หมายถึงว่าทางด้านรับจะต้องรับบิทต่าง ๆ ที่ทางด้านส่งทำการส่งมาได้อย่าถูกต้อง ซึ่งทางด้านรับจะต้องรับบิทจากสายส่งเมื่อใดหลังจากรับมาแล้วที่ 2, 3 และตัวต่อ ๆ ไปเมื่อไร ซึ่งสามารถทำได้โดยการเพิ่มสัญญาณนาฬิกาเข้าไปที่จุดปลายของระระบบทั้งสองด้าน ดังรูป 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

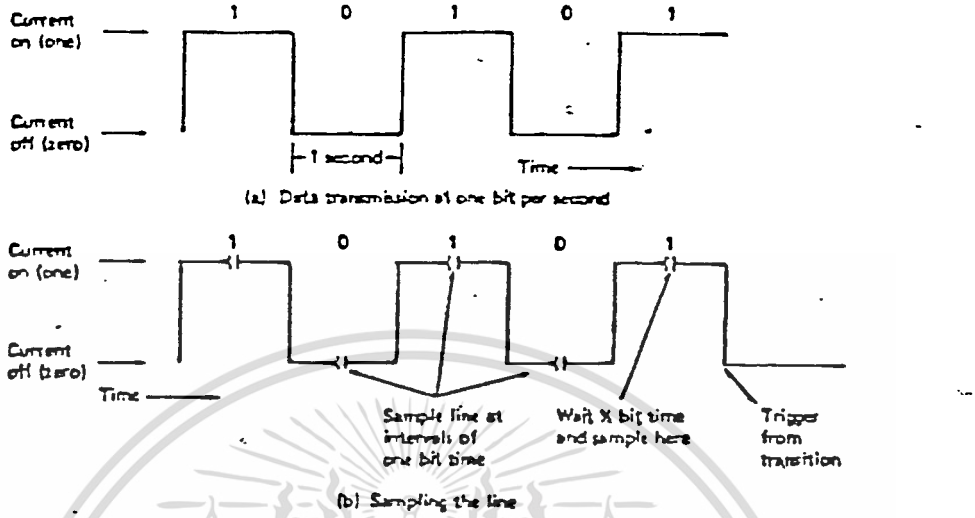


รูป 2.4 การเพิ่มคล๊อคเพื่อความสัมพันธ์ของบิท

สัญญาณคล๊อคทางด้านส่งจะเป็นตัวกำหนดเวลาว่า เมื่อใดจะส่งบิทข้อมูลออกไปตามแซ็นแนล และสัญญาณคล๊อคทางด้านรับจะเป็นตัวกำหนดว่า จะรับข้อมูลจากแซ็นแนลด้วยอัตราความถี่เท่าใด เช่น ถ้าเราต้องการส่งข้อมูลด้วยความเร็ว100 ต่อวินาที เราก็ใช้สัญญาณคล๊อคที่มีความถี่ 100 บิทต่อวินาทีสำหรับด้านลง ทางด้านลงก็จะลงบิทออกไปตามสายลง 100 บิท ใน 1 วินาที ทางด้านรับที่รับสัญญาณที่ส่งมาจะต้องรับบิททุก ๆ 1/100 วินาที ดังนั้นเราต้องสร้างสัญญาณคล๊อคทางด้านรับให้มีค่า 100 บิทต่อวินาที เพื่อที่ทางด้านรับจะได้ข้อมูลจากสายจำนวน 100 ครั้งใน 1 วินาที

สำหรับวิธีการในการ "ส่งบิทออกมาตามสาย" และ "การสุ่มตัวอย่างบิทในทางด้านรับ" นั้นจะแตกต่างกันไปสำหรับระบบแต่ละระบบ เช่นในวงจรไฟฟ้าเราอาจแทนค่า 1 ด้วยการทำให้กระแสไหลในวงจร (ดวงไฟ "ติด") และแทนค่า 0ด้วยการตัดกระแสออกจากวงจร (ดวงไฟ "ดับ") ถ้าหากเราต้องการส่งแถวของบิทที่ประกอบด้วย 0 และ 1 บิทต่อวินาที เรากระทำดังแสดงในรูป 2.5 ซึ่งเป็นตัวอย่างของค่าข้อมูลในช่วงเวลาใด ๆ ช่วงหนึ่ง ขนาดความกว้างของแต่ละบิทเท่ากันเพราะสัญญาณคล๊อคเป็นตัวควบคุมความกว้างของแต่ละบิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.5 ตัวอย่างเกี่ยวกับการรับส่งแบบง่าย ๆ

ทางด้านรับนั้นหากเป็นมนุษย์ก็สามารถมองเห็นการติดดับของดวงไฟ จึงสามารถแปลชุดของบิทที่เข้ามาได้ แต่หากเราใช้อุปกรณ์ปลายทางอิเลคทรอนิคส์แทนดวงไฟและโอเปอเรเตอร์ เราจะต้องแน่ใจว่าอุปกรณ์อิเลคทรอนิคส์นั้นสามารถรับข่าวสารมาได้อย่างถูกต้องตามปกติแล้วอุปกรณ์ปลายทางอิเลคทรอนิคส์จะสุ่มตัวอย่างโดยการตรวจดูสถานะของสาย (0 หรือ 1) ณ เวลาใด ๆ ในช่วงสั้น ๆ ว่าตาของสายเป็น 0 หรือ 1 ซึ่งเวลาที่สุ่มตัวอย่างนี้ควรจะเป็นช่วงกลางของบิทแต่ละบิท หากสุ่มตัวอย่างในช่วงเวลาที่กำหนดไว้สำหรับเป็นช่วงการเปลี่ยนแปลงระหว่าง 0 และ 1 นั้นอาจจะทำให้ได้ค่าที่ไม่ถูกต้อง ปกติแล้วการสุ่มตัวอย่างจะอยู่ที่จุดกลางของความกว้างของบิทพอดี ซึ่งสามารถกระทำได้ โดยการใช้จุดของการเปลี่ยนแปลงจาก 1 ไป 0 หรือ 0 ไป 1 เป็นจุดอ้างอิงในตอนเริ่มแรก โดยการรอเวลาผ่านไปเท่ากับ 1/2 ของเวลาของความกว้างหนึ่งบิท แล้วจึงทำการสุ่มตัวอย่าง ณ จุดนั้น ๆ หลังจากการสุ่มค่าของบิทแรกแล้วก็รอเวลาผ่านไปเท่ากับความกว้างของบิทหนึ่งบิท จึงทำการสุ่มค่าของบิทที่สอง สำหรับบิทที่สามและบิทต่อ ๆ ไปก็ทำในทำนองเดียวกัน ดังรูป 2.56 ซึ่งการสุ่มตัวอย่างค่าบิทแต่ละบิทเกิดที่กึ่งกลางของความกว้างของบิท ถ้าหากความเร็วของ

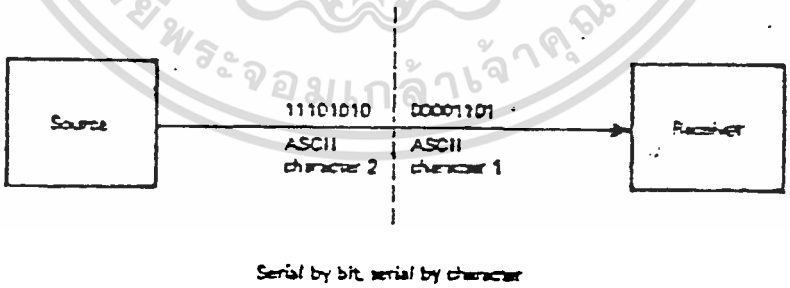
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักวิทยบริการฯ จุฬาลงกรณ์มหาวิทยาลัย

ทางปฏิบัติแล้วคล็อคของทางด้านรับและส่งจะเป็นอิสระจากกัน ค่าความเร็วจึงอาจแตกต่างกันบ้างเล็กน้อยแต่ความเร็วที่แตกต่างกันแม้เพียงเล็กน้อย จะเพิ่มขึ้นเรื่อย ๆ เพราะสัญญาณนี้เกิดติดต่อกันเป็นวัฏจักร จึงทำให้เกิดความผิดพลาดในการรับสัญญาณของทางด้านรับได้จำเป็นต้องมีการปรับความเร็วของคลีอค ทั้งทางด้านรับ และส่งให้สัมพันธ์กันที่เรียกว่า ริชิงโครไนเซ่น (resynchronization)

ความสัมพันธ์ของตัวอักขระ

ในการรับข่าวสารตามสายนั้น แม้ว่าเราจะมีการจัดเกี่ยวกับความสัมพันธ์ของบิทแล้วก็ตามยังมีปัญหาที่ตามมาอีกก็คือ - ในการส่งนั้นเรานำเอาบิทของอักขะหลาย ๆ ตัวมารวม ๆ กันเป็นบล๊อค ฉะนั้นแม้ว่าบิทต่าง ๆ ได้รับมาถูกต้องก็ตามเรายังต้องทราบว่ากลุ่มของบิทที่แสดงถึงตัวอักขระต่าง ๆ นั้นเริ่มต้นที่บิทใด รูป2.6 แสดงถึงการส่งชุดของบิทของอักษรในรหัสแบบอัสกีย์ 2 ตัวที่ส่งไปอย่างอนุกรมตามสายสื่อสาร โดยมีข้อมูลขนาด 8 บิท จำนวน 2 ชุด สำหรับตัวอักษรตัวที่ 1 และตัวที่ 2 ตามลำดับเมื่อบิทของข้อมูลทั้งสองชุดนี้ถูกส่งไปติดต่อกันทางด้านรับเมื่อรับบิทดังกล่าวมาแล้ว จะทราบได้อย่างไรว่าชุดของบิทจำนวน 8 บิทของตัวอักษรแต่ละตัวเริ่มที่บิทใด วิธีแก้ปัญหานี้กระทำได้ถ้าหากว่าเราทราบว่าบิทใดเป็นบิทเริ่มต้นของตัวอักษร และถ้าหากทราบว่า 1) ในตัวอักษรหนึ่งตัวนั้นมีกี่บิท และทราบว่า 2)ความเร็วของการส่งบิทต่าง ๆ มาตามสาย โดยการนับจำนวนบิทที่ได้รับมาตามสายหลังจากทราบบิทแรกก็จะสามารถแยกตัวอักษรออกจากกันได้



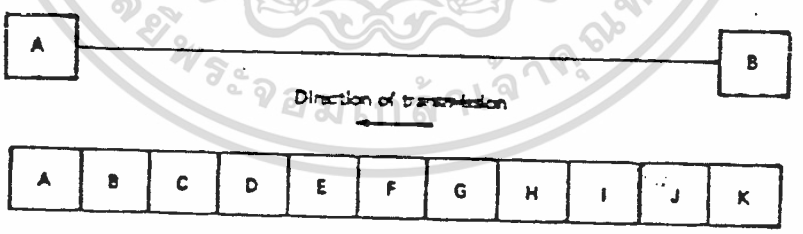
รูป 2.6 การส่งข้อมูลเป็นตัวอักษร 2 ตัวแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

033265

เทคนิคการส่งแบบสัมพันธ์นั้นใช้สำหรับส่งข้อมูลทั้งชุดไปครั้งเดียว ในการส่งแบบนี้นั้นช่วงความกว้าง (เวลา) ระหว่างบิทแต่ละบิทจะมีค่าเท่ากัน และสำหรับระบบที่มีการส่งครั้งละ 1 ตัวอักษรนั้น ช่วงเวลาระหว่างการสิ้นสุดของบิทสุดท้ายของตัวอักษรตัวหนึ่งกับการเริ่มต้นของบิทแรกของตัวอักษรตัวถัดไปจะมีค่า 0 หรือมีเวลาเท่ากับเวลาทั้งหมด สำหรับการส่งหนึ่งตัวอักษรทำให้การส่งมีลักษณะคล้ายกับการส่งข่าวสารในรูปของเลขฐานสองที่มีจำนวนบิทติดต่อกันไป โดยไม่ได้แยกว่าความยาวใดเป็นของช่วงตัวอักษรใด ในระบบเช่นนี้บิทแต่ละบิทจะมีความยาวเท่ากัน

ในรูป 2.7 แสดงให้เห็นถึงการส่งอักษรต่าง ๆ ไปแบบการส่งสัมพันธ์โดยตัวอักษรแต่ละตัวมีช่วงเวลาห่างกันเท่ากับศูนย์ ทางด้านรับนั้นเพียงหาว่าบิทแรกของตัวอักษรตัวแรกคือบิทใด และทราบขนาด หรือจำนวนบิทในหนึ่งตัวอักษร พร้อมทั้งความเร็วในการส่งก็จะสามารถแยกข่าวสารของแต่ละอักษรออกมาได้ ในกรณีที่ข้อมูลเป็นรหัสแบบอัสกีย์ ตัวอักษรแต่ละตัวจะประกอบด้วยบิท B บิท หากทราบว่าบิทใดคือบิทเริ่มต้นของตัวอักษร โดยวิธีการนับมากลุ่มละ B บิท เราก็สามารถแยกตัวอักษรต่าง ๆ ได้เพื่อให้การหาบิทแรกของตัวอักษรตัวแรกเป็นไปอย่างถูกต้องจึงมักส่งชุดของข้อมูลชุดหนึ่งก่อนหน้าการส่งข้อมูลตัวอักษร โดยการส่งตัวอักษรควบคุมความสัมพันธ์ (SYN Transmission Control Chracter : TC) ชุดของข่าวสารดังกล่าวนี้ประกอบด้วยบิทขนาด B บิท คือ 00010110 (มิพาริติคี่) สำหรับทางด้านรับนั้นจะถูกออกแบบมาให้นำบิทที่รับมาเปรียบเทียบกับชุดของบิทของตัวอักษรควบคุมความสัมพันธ์ (Looking for SYNC) โดยกระทำทุกครั้งที่รับบิทใหม่เข้ามาจนกว่าจะได้ชุดของบิทที่ต้องการดังกล่าว

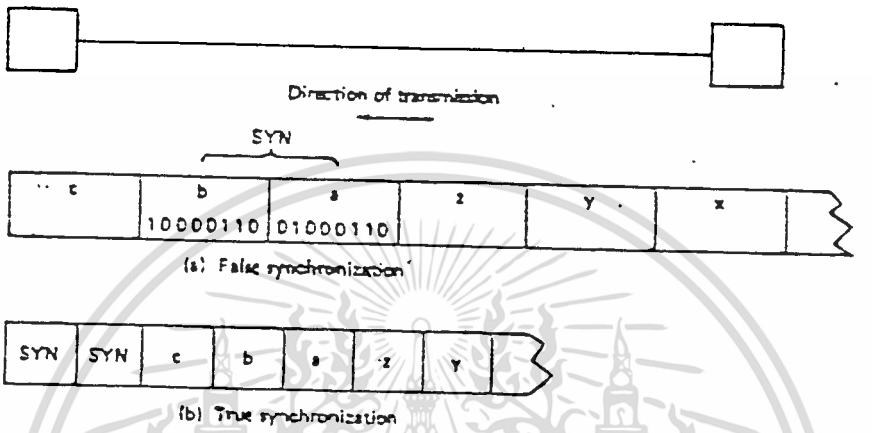


รูป 2.7 การส่งแบบสัมพันธ์

แต่วิธีการดังกล่าวนี้อาจจะเกิดผิดพลาดได้ ถ้าหากชุดของบิทของตัวอักษรที่ตามกันมามี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

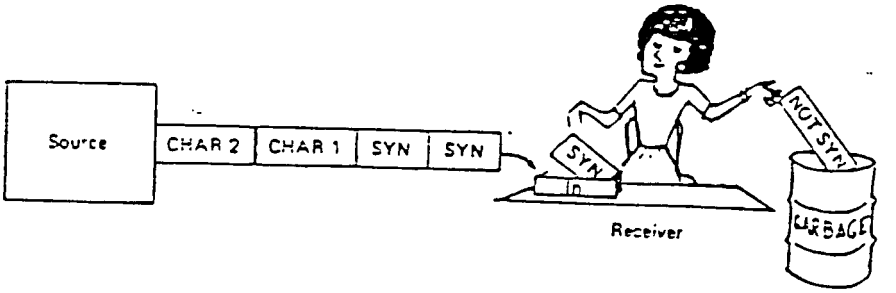
ลักษณะของบิทเหมือนกับชุดของบิทของตัวอักษรควบคุมความสัมพันธ์ ทำให้ความสัมพันธ์ที่ได้เกิดผิดพลาดไป กรณีดังกล่าวแสดงในรูป 2.8a เพื่อแก้ไขความผิดพลาดดังกล่าวนี้เราจึงส่งตัวอักษร SYN จำนวน 2 ตัวไปก่อนข้อมูลดังรูป 2.8b



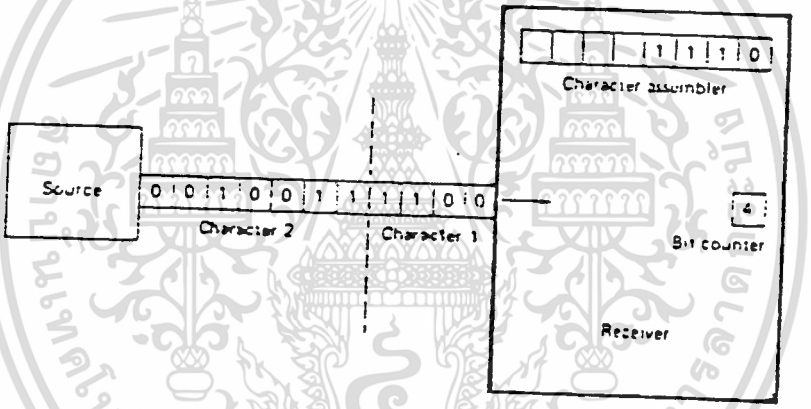
รูป 2.8 การส่งตัวอักษร SYN ไป 2 ตัวเพื่อป้องกันความผิดพลาด

สำหรับทางด้านรับเมื่อรับข่าวสารและตรวจสอบได้ว่าเป็นตัวอักษร SYNแล้ว จะต้องนำชุดของบิทจำนวน 8 บิทต่อไปมาตรวจสอบอีกว่าเป็นอักษร SYN หรือไม่ หากชุดที่ 2 เป็นอักษร SYN ก็แสดงว่าชุด 8 บิทที่นำมาติดนั้นถูกต้อง ถ้าหากว่าB บิทชุดแรกเป็นอักษร SYN แต่ชุดที่ 2 ไม่ใช่ก็จะต้องตรวจสอบต่อไป รูป 2.9 และ รูป 2.10 แสดงถึงวิธีการดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.9 การส่งแบบสัมพันธ์ที่มี SYN หลายตัว

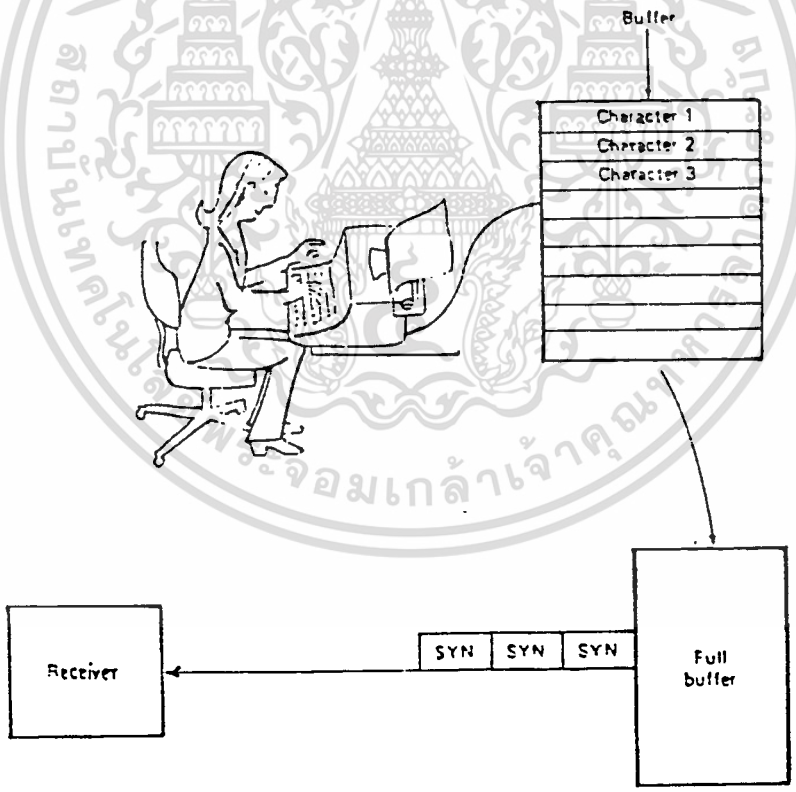


รูป 2.10 การส่งแบบสัมพันธ์

ในระบบส่วนมากมักจะมีตัวอักษร SYN นำหน้าข้อมูล 3 - 4 ตัว เพื่อให้แน่ใจได้ว่าชุดของบิทสำหรับอักขระที่รับต่อมาคือข้อมูลที่ถูกต้อง นอกจากวิธีการส่งตัวอักษร SYN หลายตัวติดต่อกันสำหรับการสร้างความสัมพันธ์ของตัวอักษรดังได้กล่าวมาแล้วนั้น ยังมีวิธีการบางชนิดที่ส่งตัวอักษรสองชนิดที่แตกต่างกันไปเพื่อสร้างความสัมพันธ์ โดยทางด้านรับ เมื่อตรวจพบตัวชุดของบิทที่ตรงกับอักษรตัวแรกที่กำหนดแล้วจะต้องตรวจสอบต่อไปว่าอักษรตัวที่สองนั้น ตรงกับที่กำหนดไว้หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการส่งแบบสัมพันธ์นี้ ส่วนมากแล้วข่าวสารที่จะส่งนั้นมักจะถูกรวบรวมหรือเก็บรักษาไว้ในอุปกรณ์ปลายก่อนที่จะส่งออกไป ทั้งนี้เพราะในการส่งแบบสัมพันธ์นั้น อักษรทุกตัวจะต้องถูกส่งออกไปด้วยเวลาที่คงที่ ซึ่งเป็นการยากที่โอเปอเรเตอร์ที่ป้อนข่าวสารจะกระทำได้ และนอกจากนี้ความเร็วในการส่งของสายนั้นมีความเร็วสูงกว่าการทำงานของโอเปอเรเตอร์ ฉะนั้นอุปกรณ์ปลายโดยทั่วไปจึงมักจะมีส่วนสำหรับเก็บข่าวสารที่เรียกว่า บัฟเฟอร์ (Buffer) สำหรับเก็บรวบรวมข่าวสารที่โอเปอเรเตอร์ป้อนเข้าไป แล้วส่งไปด้วยความเร็วการทำงานของอุปกรณ์ปลายเอง อุปกรณ์เก็บรวบรวมข่าวสารนี้ก็คืออุปกรณ์ทางดิติตอลที่เรียกว่า เมมโมรี่(Memory) นั่นเอง สำหรับการใส่ตัวอักษร SYN โอเปอเรเตอร์ไม่จำเป็นต้องใส่เข้าไป แต่อุปกรณ์ปลายจะใส่เข้าไปให้เองก่อนการส่งข้อมูลไปตามสาย สำหรับปลายอีกด้านหนึ่งนั้นก็ต้องมีอุปกรณ์ที่เรียกว่าบัฟเฟอร์ เพื่อรับข้อมูลที่ส่งมาจากทางด้านส่ง วิธีการส่งแบบสัมพันธ์นี้ทำให้เราสามารถใช้สายสื่อสารได้อย่างเต็มที่ รูป 2.11แสดงถึงการส่งแบบสัมพันธ์ที่มีบัฟเฟอร์

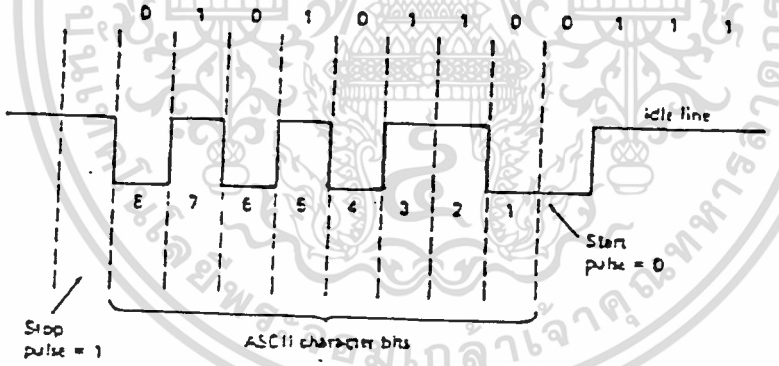


รูป 2.11 การใช้บัฟเฟอร์ในการส่งแบบสัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งข้อมูลแบบไม่สัมพันธ์

การส่งข้อมูลแบบนี้ เราสามารถส่งข้อมูลออกไป โดยไม่จำเป็นต้องมีความสัมพันธ์ระหว่างตัวอักษรว่าจะต้องมีเวลาที่แน่นอนอย่างไร โดยเราอาจจะส่งตัวอักษรติดต่อกันไป หรืออาจจะเว้นช่วงเวลาสำหรับการส่งตัวอักษรเป็นเวลาเท่าใดก็ได้ ในกรณีเช่นนี้ ทางด้านรับจะต้องสร้างความสัมพันธ์ขึ้นใหม่สำหรับอักษรแต่ละตัวที่รับมา ฉะนั้นทางด้านรับจะต้องทราบถึงบิทเริ่มต้นของอักขระแต่ละตัวว่าเริ่มต้นที่ใดวิธีการดังกล่าวสามารถกระทำได้โดยการเพิ่มบิทที่เรียกว่า Start Pulse โดยเติมเข้าไปข้างหน้าชุดของบิททุก ๆ ตัวอักขระ สำหรับบอกให้ด้านรับทราบถึงบิทเริ่มต้นของอักขระใด ๆ ในทางปฏิบัติด้านรับสามารถทราบว่ามี Start Pulse หรือยังโดยการตรวจสอบสถานะของสายซึ่งจะมีสถานะเป็น 0 หรือ 1 ก็ได้ โดยทั่วไปเมื่อไม่มีการส่งข่าวสารมักกำหนดให้สายมีสถานะเป็น 1 เรียกว่า NAK สถานะตรงข้ามคือ 0 เรียกว่า SPACE เมื่อต้องการส่งตัวอักษร 1 ตัว เครื่องส่งจะต้องส่งบิทที่บอกถึงการเริ่มต้นเรียกว่า บิทเริ่มต้น (Start Bit) หรือ Start Pulse โดยการส่ง 0 ออกไปตามสาย ด้านรับจะทราบว่าบิทที่ตามมาเป็นข่าวสาร แสดงดังรูปที่5.10



รูปที่ 2.12 การส่งข้อมูลแบบไม่สัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

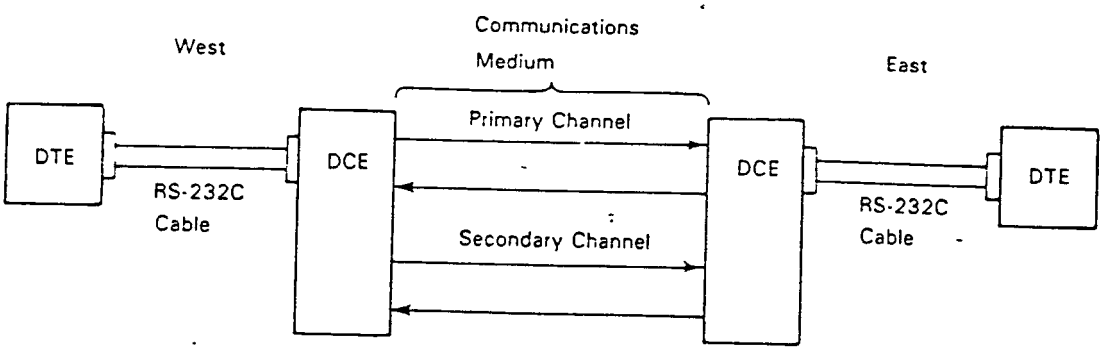
ทางด้านรับเมื่อรับทราบถึงการเปลี่ยนสถานะของสายการเริ่มต้นการทำงานจะเริ่มภายหลังเวลาผ่านไปเท่ากับครึ่งเวลาของความกว้างขนาด 1 บิท โดยจะสุ่มตัวอย่าง (ตรวจสอบ) สายว่าขณะนั้นมีสถานะ 0 หรือ 1 ถ้าเป็น 0 แสดงว่ามีการส่งบิทเริ่มต้น หลังจากนั้นจะสุ่มตัวอย่างสถานะของสายทุก ๆ ช่วงเวลา 1 บิทเพื่อหาค่าเวลารหัสของตัวอักขระที่ส่งมา ในการสุ่มตัวอย่างครั้งแรกนั้นหลังจากได้รับการเปลี่ยนแปลงจาก 1 ไป 0 เป็นเวลา 1 บิท หากค่าสถานะของสายที่รับมามีค่า 1 ก็แสดงว่าบิทที่รับมาไม่ใช่บิทเริ่มต้น อาจเป็นเพียงสัญญาณรบกวนที่เรียกว่า สัญญาณรบกวนอิมพัลซ์ (Noiseimpylse) ในกรณีนี้ทางด้านรับจะทิ้งสัญญาณนี้ไป การตรวจสอบบิทเริ่มต้นนี้กระทำทุกครั้งที่มีอักขระตัวใหม่เข้ามาเราจึงสามารถเว้นระยะห่างระหว่างตัวอักขระให้มีเวลาเท่าใดก็ได้ และเมื่อส่งครบบิทก็จะมีบิทสำหรับบอกการสิ้นสุดเรียกว่า พัลซ์การสิ้นสุด (Stop Pulse) เพื่อให้ด้านรับมีเวลาสำหรับการเตรียมรับข้อมูลตัวต่อไป ในรูปที่ 2.12 จะพบว่าบิทของข้อมูลนั้นมีบิทเริ่มต้นนำหน้าและปิดท้ายด้วยบิทหยุดทุก ๆ ครั้งตามขนาดของบิทที่แทนตัวอักษรหนึ่งตัว พัลซ์การหยุดมีค่าเป็น 1 ส่วนระยะเวลาของพัลซ์ดังกล่าวนี้จะแตกต่างออกไปสำหรับแต่ละระบบสำหรับรหัสอัสกีย์พัลซ์การหยุดมีขนาดความกว้างเท่ากับขนาดความกว้างของบิทข้อมูลหนึ่งหรือสองบิท ความจริงแล้วช่วงเวลาของบิทการหยุดนั้นเป็นช่วงเวลาสำหรับทางด้านรับจะทำงานหลังจากรับข้อมูลมาครบหนึ่งตัวอักขระ เช่นการพิมพ์ในเครื่องพิมพ์การเจาะรูปบนเทปของเครื่องเจาะเทป เป็นต้น ในการทำงานของอุปกรณ์แบบจักรกลนี้มีการทำงานที่ช้ามาก จึงอาจต้องมีเวลาของพัลซ์การหยุดเท่ากับเวลาของ 2 บิท แต่อุปกรณ์ประเภทเครื่องจักรไฟฟ้ามีความเร็วในการทำงานมากกว่า จึงมีพัลซ์การหยุดเพียง 1 บิท

เนื่องจากอักขระทุกตัวที่ส่งออกมามีความเป็นอิสระต่อกันและทุก ๆ อักขระจะมีบทเริ่มต้นและบิทสิ้นสุดอย่างน้อยอักขระละ 1 บิท หากเกิดสัญญาณรบกวน อาจทำให้เกิดความผิดพลาดกับข้อมูลได้ ความผิดพลาดนี้จะเกิดขึ้นกับข้อมูลเพียงอักขระเดียว เพราะอักขระแต่ละตัวมีความสัมพันธ์กับบิทเริ่มต้นกับบิทสิ้นสุดของตัวมันเองเท่านั้น

2.2 มาตรฐานการเชื่อต่อ RS-232C

ข้อกำหนดตาม RS-232 C นั้น บอกถึงการเดินสายในเคเบิลที่ต่อระหว่างDTE (DATA TERMINAL EQUIPMENT) กับ DCE (data communication Equipment) ตามรูปที่ 2.13 เป็นเคเบิลที่ต่อกับปลั๊ก 25 ขาที่เสียบเข้ากับคอนเน็คเตอร์ ของพอร์ทอนุกรมที่หลัง PC หรืออุปกรณ์ปลายทางต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.13 แบบจำลองของวงจรการสื่อสารที่ใช้ RS-232 C

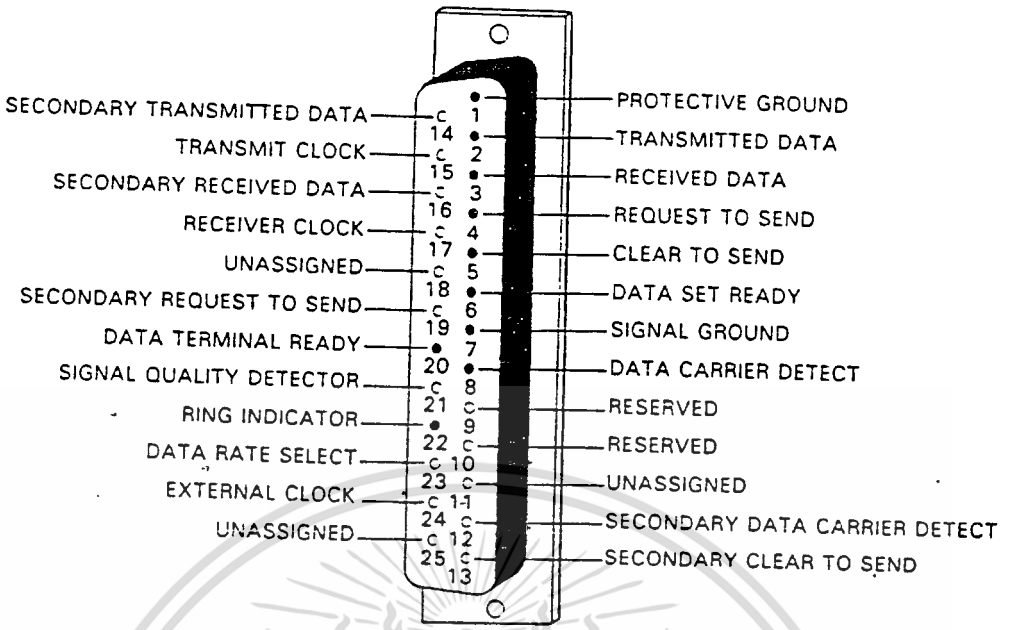
ปลั๊กนี้มีลักษณะดังแสดงในรูป 2.14 ประการแรกจะกำหนดระดับโวลเตจ และคุณสมบัติทางไฟฟ้าอย่างอื่นของสายในเคเบิล พร้อมทั้งอธิบายหน้าที่ของมันซึ่งจะได้กล่าวถึงในหัวข้อต่อไป และเพื่อที่จะให้เข้าใจว่า DTE ควบคุมโมเด็มได้อย่างไร เราจำเป็นต้องเข้าใจแบบจำลอง หรือรูปแสดงแนวความคิดของตัวกลางการสื่อสารที่ใช้ในข้อกำหนด RS-232-c ด้วยรูปแบบจำลองนี้มีดังแสดงในรูป 2.13 ต่อไปในหนังสือเล่มนี้เมื่อกล่าวถึงอุปกรณ์ที่อยู่ทางด้านซ้ายมือ และด้านขวามือของตัวกลางสื่อสาร เราจะอ้างว่าเป็นอุปกรณ์ด้านตะวันตก และด้านตะวันออกตามลำดับ

ตัวกลางของการสื่อสาร ประกอบด้วย สัญญาณหลัก (primary channel) และ **ช่องสัญญาณรอง** (secondary channel) ขอให้ผู้อ่านจงเข้าใจว่าช่องสัญญาณนั้นเป็นช่องทาง (pipeline) ที่สัญญาณไหลผ่านเท่านั้น ดังนั้นช่องสัญยาณหลัก และช่องสัญญาณรองนั้นอาจจะมีอยู่ในสายคู่เดียวกัน หรือคนละคู่ก็ได้

เกี่ยวเนื่องกับช่องสัญญาณ ก็คือแนวความคิดเรื่อง คลื่นพาห์ (carrier)คลื่นพาห์คือสัญญาณอนาลอก (ปกติเป็นคลื่นรูปไซน์) ที่ความถี่ที่เหมาะสมที่จะเคลื่อนผ่านตัวกลางการสื่อสารได้ ตัวพาห์เป็นตัวบรรทุกข้อมูล ที่เรียกว่า มอดดูเลทข้อมูลเพื่อนำข้อมูลผ่านตัวกลางการสื่อสารไปสู่จุดหมายมีวิธีการมอดดูเลทหลายวิธี ซึ่งเราจะกล่าวถึงต่อไปในบทข้างหน้า

ในแต่ละช่องสัญญาณ มีสัญญาณคลื่นพาห์ 2 สัญญาณ แต่ละสัญญาณใช้สำหรับการส่งแต่ละทิศทาง ถ้าคลื่นพาห์ทั้งสองสามารถใช้ได้พรอมกัน ช่องสัญญาณก็จะเป็นชนิดฟูลดูเพล็กซ์ แต่ถ้าต้องใช้สลับเวลากัน ช่องสัญญาณนั้นก็เป็นชนิด ฮาล์ฟดูเพล็กซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 คอนเน็คเตอร์ RS 232-C

ช่องสัญญาณมีค่า ความจุ ซึ่งมีค่าเท่ากับ จำนวนบิทต่อวิทนาทีที่มากที่สุดที่ช่องสัญญาณยอมให้ผ่านตัวมันไปได้ ช่องสัญญาณรองไม่จำเป็นที่จะต้องมีใช้อยู่ในทุกโมเด็ม แต่ถ้ามีอยู่แล้วค่าความจุ (ความเร็ว) ของมันจะน้อยกว่าค่าความจุของช่องสัญญาณหลัก เพราะถ้าไม่เป็นเช่นนั้นมันก็จะไม่มีแนวความแตกต่างระหว่างช่องสัญญาณหลักกับช่องสัญญาณรอง ข้อกำหนด RS-232-C กำหนดว่ามันไม่สามารถจะประยุกต์ใช้ได้กับตัวกลางการสื่อสารที่มีค่าสูงกว่า 20,000 บิทต่อวิทนาทีRS-232 c ปล่อยการเลือกหลายอย่างไว้ให้ขึ้นกับผู้ออกแบบโมเด็มเองเป็นต้นว่า

- ช่องสัญญาณจะเป็น ฮาล์ฟ หรือ ฟูลดูเพล็กซ์
- ตัวกลางการสื่อสารจะเป็นแบบสวิตซ์ หรือแบบเช่าเฉพาะ
- ช่องสัญญาณจะเป็นแบบซิงโครนัสหรืออะซิงโครนัส
- ช่องสัญญาณรองจะมีหรือไม่

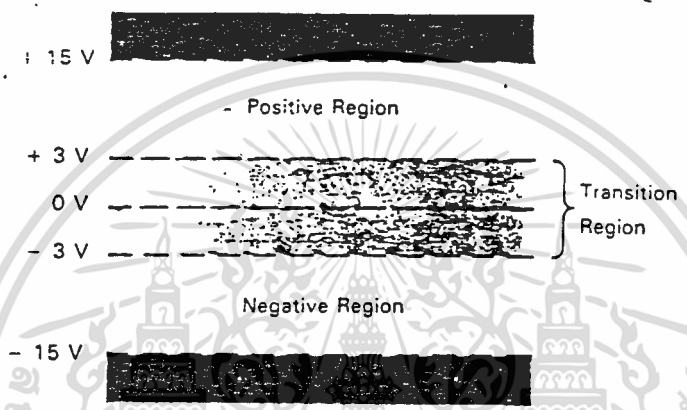
เหล่านี้ล้วนแล้วแต่ผู้ออกแบบจะต้องการอย่างไรก็ได้ ผลก็คือ จำนวนของสายในเคเบิล RS-232C ที่ใช้ควบคุม DCE จะถูกใช้ไม่เท่ากัน ขึ้นอยู่กับการออกแบบ บางทีอาจจะไม่มีโมเด็มแบบหนึ่งแบบใดเลยก็ได้ที่ใช้สายครบหมดทุกสายตาม RS-232 C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อกำหนดทางไฟฟ้า

มีรายละเอียดเกี่ยวกับคุณสมบัติของสัญญาณไฟฟ้าบนสายแต่ละสายในเคเบิลRS 232 C มากมายแต่เพราะเราเกี่ยวข้องอยู่กับสัญญาณไบนารี

ข้อจำกัดเกี่ยวกับโวลเตจจึงถูกกำหนดในสองบริเวณดังแสดงในรูป 6.3 บริเวณบวก (postive region) อยู่ระหว่าง +3 โวลท์ DC ถึง 5 โวลท์ DC



รูป 2.15 ระดับโวลเตจของสัญญาณที่ RS-232 C กำหนดให้

สัญญาณจะมีสถานะอยู่ในบริเวณนี้ได้ไม่เกิน 1 มิลลิวินาที ในบริเวณเปลี่ยนถ่ายสถานะนี้จะไม่มีการกำหนดสถานภาพให้กับสัญญาณแต่อย่างใด

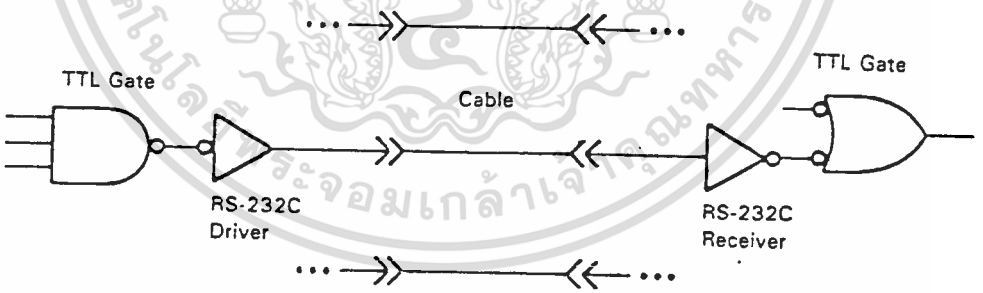
บริเวณเหล่านี้ใช้เกี่ยวพันกับสถานะไบนารีของสายสัญญาณในลักษณะที่เป็นเอกเทศ การแปลความหมายของระดับโวลเตจขึ้นอยู่กับหน้าที่ของสาย ซึ่งกำหนดแบ่งเป็นไปตามตาราง 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

_ตาราง 2.1 หน้าที่ของสายและความหมายของโวลเตจที่กำหนดให้

WIRE FUNCTION	VOLTAGE LEVEL	
	Positive	Negative
Data	SPACE (0)	MARK (1)
Modem Control & Timing	On (asserted)	Off (negated)

เนื่องจากระดับไซลเตจดังกล่าวไม่พอเหมาะที่จะใช้ได้กับ TTL ดังนั้นอินเตอร์เฟสสู่เคเบิล RS-232 C จึงต้องการวงจรภาคขับ และภาครับเป็นพิเศษ ดังแสดงในรูป 2.16 ภาคขับ และภาครับนี้สามารถหาได้ในรูป IC ตามท้องตลาดทั่วไป



รูป 2.16 วงจรอินเตอร์เฟส RS-232 C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RS-232 C นั้นจำกัดค่าความจุไฟฟ้าของสายสัญญาณ (วัดเทียบกับกราวด์)อย่างมากสุดคือ 2500 PF สำหรับสายเคเบิลที่มีฉนวนและช่วงว่างระหว่างสายอย่างสามัญทั่วไป สายยาวประมาณ 50 ฟุต จะมีค่าความจุไฟฟ้าประมาณนั้น ดังนั้นถ้าไม่ใช้สายชนิดพิเศษแล้ว ระยะห่างมากสุดระหว่าง DTE และ DCE คือ 50 ฟุต หรือประมาณ 15 เมตร

2.3 การควบคุมการเชื่อมต่อข้อมูล

2.3.1 คาร์แรกเตอร์ - โอเรียน (Character - oriented) โปรโตตอล สำหรับการสื่อสารทั่วไป สามารถแบ่งได้เป็น 2 ประเภท คือ บิท-โอเรียนโปรโตคอล (Bit-Oriented Protocol)ไบท์โอเรียน โปรโตคอล (Byte-Oriented Protocol) หรือคาร์แรกเตอร์ - โอเรียน โปรโตคอล (Character-Oriented Protocol)

สำหรับเนิดเวอร์คที่ทำขึ้นได้ใช้ประเภท Byte-Oriented Protocolตัวอย่างหนึ่งของ Byte-Oriented Protocol ที่ใช้กันทั่วไปบนพีซี ก็คือ ASCII (Ametrican Standard Code for Information Interchange)ดังนั้นการสื่อสารของเนิดเวอร์คนี้จึงนำรหัสแอสวีมาใช้ (รูปที่ 1) โดยรหัสแอสวีที่สำคัญที่ถูกนำมาใช้ในโฟล คอนโทรล ของเน็ตเวอร์คนี้จะใช้ 4 ห้อง เท่านั้น คือACK, ENQ, ETX, SYM ซึ่งมีคำอธิบายดังนี้

ACK (Acknowledge) : คาแรกเตอร์นี้จะถูกใช้โดยทางด้านรับเพื่อบอกให้ทราบว่า ได้รับข้อมูลที่ไม่ผิดพลาดแล้ว

ENQ (Enquity) : มื่อคอมพิวเตอร์ 2 เครื่อง เริ่มที่จะติดต่อสื่อสารกัน เครื่องทางด้านส่งจะส่ง NEQ เพื่อให้ทราบว่า เครื่องปลายทางอีกด้านยังเชื่อต่ออยู่หรือไม่

ENQ (End of Text) : คาแรกเตอร์นี้ถูกใช้ในระบบซิงโครนัสเพื่อที่จะเริ่มหรือรักษาการซิงโครไนซ์ ในเส้นทางการสื่อสารในขณะที่ไม่มีข้อมูลไหลอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

High-order bits

Bits 4 3 2 1	7 6 5 = 0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0 0 0 0	NUL	DLE	SP	0	@	P	\	p
0 0 0 1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	STX	DC2	"	2	B	R	b	r
0 0 1 1	ETX	DC3	#	3	C	S	c	s
0 1 0 0	EOT	DC4	$	4	D	T	d	t
0 1 0 1	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	ACK	SYN	&	6	F	V	f	v
0 1 1 1	BEL	ETB	'	7	G	W	g	w
1 0 0 0	BS	CAN	(	8	H	X	h	x
1 0 0 1	HT	EM	)	9	I	Y	i	y
1 0 1 0	LF	SUB	*	:	J	Z	j	z
1 0 1 1	VT	ESC	+	;	K	[	k	{
1 1 0 0	FF	FS	,	<	L	\	l	:
1 1 0 1	CR	GS	—	=	M	]	m	}
1 1 1 0	SO	RS	.	>	N	^	n	~
1 1 1 1	SI	US	/	?	O	—	o	DEL

Low-order bits

*Adapted from *Data Communications: A User's Guide* by Kenneth Sherman. Reston Publishing Company, Inc.: 1981, p. 95. Reprinted with permission.

รูป 2.17 รหัส ASC II

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 มีเดีย แอคเซส โปรโคคอล (MEDIA ACCESS PROTOCOL)

ในระบบ (LAN) ที่มีอยู่ในปัจจุบัน จะมีส่วนของมีเดีย แอคเซส คอนโทรล โปรโตคอล (Media Access Control Protocols) ซึ่งสามารถแบ่งได้เป็น 5 กลุ่ม ดังนี้

1. ฟิกซ์ แอสไซน์เมนท์ เทคนิค (Fixed Assignment Technigue)
2. แรนดอม แอคเซส เทคนิก (Random Access Technigue)
3. เซ็นทรัลลี่ คอนโทรล ดีมานด์ แอสไซน์เมนท์ เทคนิก (Centrally Controlled Demand Assignment Technigue)
4. ดีมานด์ แอสไซน์เมนท์วิดธ์ ดีสทรีบิวท์ คอนโทรล (Demand Assignment With Distributed control)
5. เทคนิคอื่น ๆ (Other Technigue)

อนึ่ง สำหรับโครงข่ายการควบคุมจากศูนย์กลางที่ทำขึ้นได้ใช้เซ็นทรัลสี่คอนโทรล ดีมานด์ แอสไซน์เมนท์ เทคนิก ซึ่งเทคนิคนี้มีความยืดหยุ่นมากกว่า 2 เทคนิคแรก โดยจะมีส่วนควบคุมศูนย์กลางเป็นตัวกำหนดความต้องการ ซึ่งจะแตกต่างกันที่เทคนิคแรกจะไม่สามารถปรับเปลี่ยนตามความต้องการของผู้ใช้ และถ้าระบบต้องการให้มีดีเลย์สั้น ระบบอาจจะเสียหายได้ ส่วนเทคนิคที่ 2 จะมีปัญหาการชนกันของข้อมูลดังนั้นเทคนิกที่ 3 ผู้ใช้จะสามารถแสดงความต้องการการส่งข้อมูล ก่อนที่ช่องสัญญาณจะถูกแบ่งให้สำหรับข้อมูลนั้น ซึ่งกรรมวิธีแบ่งของสัญญาจะทำให้ 2 วิธี คือ วิธีพอลลิ่ง (polling) ซึ่งส่วนควบคุมศูนย์กลางจะสอบถามไปยังเครื่องปลายทางว่ามีข้อมูลจะส่งหรือไม่ ส่วนอีกวิธีซึ่งเป็นวิธีที่ใช้ในเน็คเวอร์คนี้ คือ วิธีรีเซิร์ฟ (Reserve)ซึ่ง Terminal จะเป็นผู้ร้องขอการใช้ช่องสัญญาณ เมื่อต้องการส่งข้อมูล วิธีนี้จะมีความเหมาะสมที่สุดในการใช้พีซีเป็นมีเดีย และยังมีทรูพุท (Throughput) ที่สูงอีกด้วย แต่ถ้าเป็น ทีดีเอ็มเอ (TDMA) หากมีการปิด หรือไม่ใช้ในบาง Terminal ก็จะเป็นการสูญเสียเวลาที่จัดไว้ให้ของแต่ละ Terminal ไป

2.5 โปรโตคอลของการสื่อสารแบบอนุกรม

เมื่อพิจารณาการส่งข้อมูลในแบบอนุกรมจะพบว่า ปัญหาหนึ่งที่เกิดขึ้นอยู่เสมอก็คือ การตัดสินว่าข้อมูลที่ได้รับนั้นมีจุดเริ่มต้นที่ใด ดังนั้นจึงมีการกำหนดข้อตกลงในการสื่อสารขึ้นเพื่อแก้ปัญหานี้ ข้อตกลงดังกล่าวเราเรียกว่า โปรโตคอล(Protocol) ของการสื่อสารข้อมูลแบบอนุกรม สามารถแบ่งออกได้เป็น 2 ประเภทใหญ่ คือ โปรโตคอลสำหรับการสื่อสารข้อมูลอนุกรมแบบซิงโครนัส (Synchronous) และโปรโตคอลสำหรับการสื่อสารข้อมูลอนุกรมแบบอซิงโครนัส (Asynchronous) การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สื่อสารแบบซิงโครนัสนั้น ข้อมูงจะถูกส่งออกไปอย่างสม่ำเสมอ ช่วงเวลาระหว่างบิท และระหว่างเติมเพื่อกำกับการส่งว่าควรจะส่งเมื่อใด และควรจะหยุดเมื่อใดระบบที่เป็นซิงโครนัสจะเป็นระบบที่มีความเร็วสูง แต่ก็ยังต่ำกว่าการสื่อสารแบบขนาน

การสื่อสารแบบอซิงโครนัส เป็นหัวใจของการสื่อสารข้อมูลผ่านทางสายโทรศัพท์ในปัจจุบัน การสื่อสารแบบนี้ช่วงเวลาระหว่างบิทจะมีค่าเช่นเดียวกับซิงโครนัส แต่จะแตกต่างกันที่ระยะห่างระหว่างเวิร์ค ซึ่งไม่มีข้อกำหนดว่าจะห่างกันเป็นระยะเวลานานเท่าไร กล่าวคือ ระหว่างเวิร์คนั้นจะห่างกันกี่วินาที กี่นาที กี่ชั่วโมง กี่วันกี่เดือน หรือกี่ปีก็ได้ทั้งสิ้น ขึ้นอยู่กับว่าฝ่ายรับจะรอบรับได้หรือไม่

เมื่อไม่มีข้อกำหนดทางด้านระยะเวลาระหว่างเวิร์ดแล้ว ทางผู้ส่งและผู้รับจะเข้าใจตรงกันได้อย่างไรว่าที่ใดค่อจุดเริ่มต้น และจุดสิ้นสุดของแต่ละเวิร์ด เพื่อแก้ปัญหานี้จึงมีการกำหนดข้อตกลงเกี่ยวกับฟอร์แมตของข้อมูลที่จะส่งให้ทางผู้รับสามารถเข้าใจว่า จุดใดเป็นจุดเริ่มต้นของเวิร์ค ข้อกำหนดดังกล่าวจะกำหนดให้แต่ละเวิร์คต้องขึ้นต้นด้วยบิทที่เรียกว่า start bit หรือบิทเริ่มต้น ซึ่งจะต้องเป็นลอจิกศูนย์เสมอ จากนั้นตามด้วยบิทข้อมูลที่ต้องการส่ง ซึ่งมีความยาว 5 ถึง B บิท ถัดจากบิทข้อมูลก็จะเป็นพาริติบิทซึ่งจะทำหน้าที่เป็บิทสำหรับตรวจสอบความถูกต้องหรือไม่ บิทพาร์ทตี้นี้มี 2 ประเภทคือ even parity (มีค่าเป็น 1) เมื่อจำนวนบิทที่เป็นลอจิก

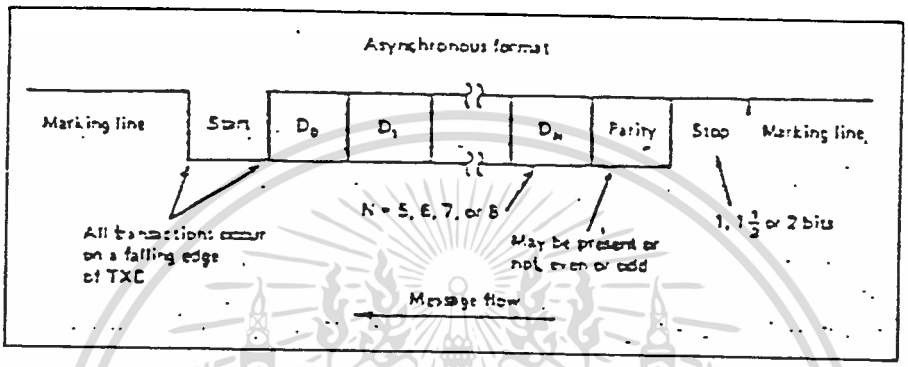
1 ในบิทที่เป็นข้อมูลมีจำนวนเป็นคี่ และ odd parity ซึ่งจะเซ็คเมื่อจำนวนบิทที่เป็นลอจิก 1 ในบิทที่เป็นข้อมูลมีจำนวนเป็นคู่ ในการส่งข้อมูลบ่งครั้งอาจจะไม่มีการใช้บิทพาริตี้ก็ได้ ถ้าหากการสื่อสารในครั้งนั้นมีความน่าเชื่อถือสูง คือ มีสัญญาณรบกวนต่ำเป็นการเพิ่มความเร็วในการส่งข้อมูลด้วย บิทสุดท้ายในฟอร์แมตก็คือ stop bit ทำหน้าที่บอกทางผู้รับว่าขณะนี้มีข้อมูลที่ได้รับนั้นครบเวิร์ดแล้วขอให้เตรียมตัวรับเวิร์ดต่อไป stop bit นี้ถูกกำหนดให้เป็นลอจิก 1 เสมอ ทั้งนี้เพื่อให้ระบบสามารถ

ตรวจสอบบิทเริ่มต้นได้ (บิทเริ่มต้นมีลอจิกเป็นศูนย์) stop bit อาจมีความยาวเป็น บิท 1.5 บิท หรือ 2 บิท ก็ได้

จากฟอร์แมตดังกล่าว จะเห็นว่าเรามีฟอร์แมตสำหรับการสื่อสารมากมายไปหมด เช่น 5E1 (5Data bit, Even paity, 1 Stop bit), 7E1 (7Data bit,Even parity, 1 Stop bit) เป็นต้น ในการใช้งานทั่วไปเรานิยมใช้กันอยู่เพียง2 ฟอร์แมตคือ 7E1 และ EN1 จะเลือกใช้ฟอร์แมตใดขึ้นอยู่กับสภาพของสายส่งสัญญาณว่ามีสัญญาณรบกวนมากเพียงใด ถ้าหากสายส่งสัญยาณรบกวนมากควรจะใช้ 7E1_ __แต่ถ้าสายส่งมีสภาพติดสัญญาณรบกวนต่ำควรใช้ BN1 เพราะความเร็วมากกว่า ทั้งนี้ต้องมีการตกลงกันล่วงหน้าระหว่างผู้ส่งกับผู้รับว่าจะใช้ฟอร์แมตใดในการสื่อสารลักษณะของข้อมูลที่ถูกส่งออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปจะมีลักษณะดังรูป 2.18



รูป 2.18 ตัวอย่างของฟอร์แมตข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่3
หลักการทำงานของฮาร์ดแวร์

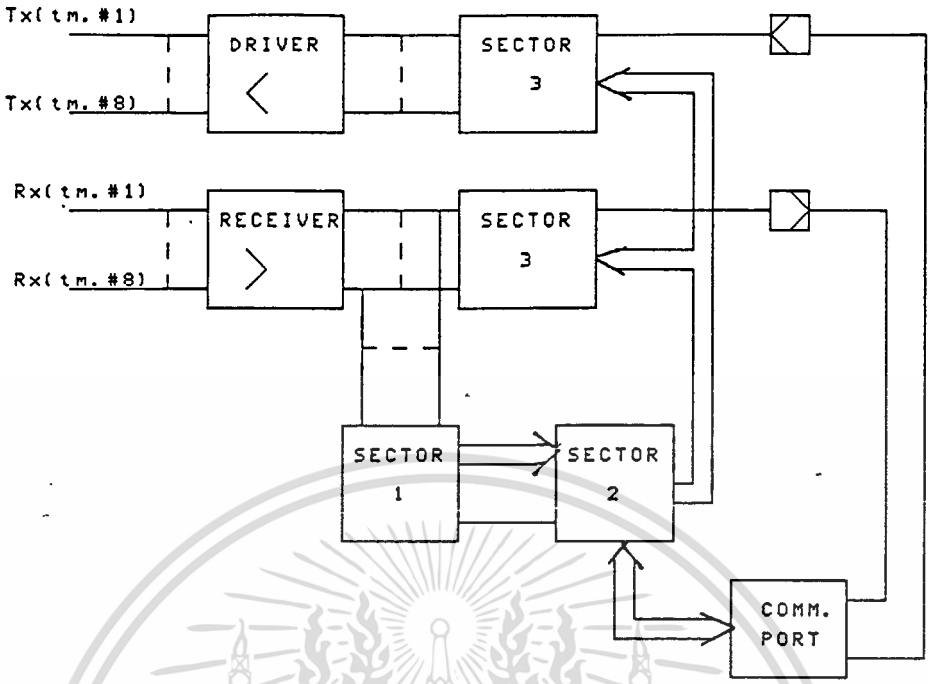
หลักการทำงานของฮาร์ดแวร์ (HARDWARE) ได้เน้นเฉพาะการเชื่อมต่อระหว่างตัวเทอร์มินัลกับฮับ โดยที่ฮับเป็นศูนย์กลางการทำงานและการเชื่อมต่อทั้งหมด ซึ่งมีหลักการทำงานดังนี้เนื่องจากการที่เลือกใช้สัญญาเพียง 3 เส้น ก็คือTx, Rx และ Gnd. โดยขา Rx เป็นขาสัญญาณที่เทอร์มินัลส่งข้อมูลให้กับนั้นส่วนขา Tx เป็นขา สัญญาณที่นั้นส่งข้อมูลไปยังเทอร์มินัลส่งผลให้ การรับรู้ถึงการขอใช้ช่องสัญญาณ(Channel) ที่ตัวควบคุม (Controllor) มีได้ทางเดียว คือ ต้องส่งสัญญาณRx เท่านั้น และในสภาวะปกติสัญญาณ Rx ที่ผ่านท่างตัวรับ (Receiver) มาแล้วจะมีสถานะเป็นไฮ (High) ดังนั้น การที่เทอร์มินัลใด ๆ จะขอใช้ช่องสัญญาณก็ควรจะต้องส่งสัญญาณมาบอก โดยการทำให้ Rx ที่สภาพเปลี่ยนจากไฮไปเป็นโล (Low) ซึ่งจะทำให้ตัวควบคุมสามารถตรวจจับได้ แล้วจึงทำการสวิตช์ช่องสัญญาณถูกต้อง ดังนั้นฮาร์ดแวร์ ของส่วนที่จึงถูกแบ่งออกเป็น 3 ส่วนด้วยกันคือ

ส่วนที่ 1ก็คือ ส่วนตรวจจับการขอใช้ช่องสัญญญา ซึ่งจะเก็บผลรอให้ส่วนควบคุม (ส่วนที่ 2) มาอ่านแล้วเคลียร์ (Clear) จากนั้นจะสามารถตรวจจับได้อีกต่อไป

ส่วนที่ 2 ก็คือ ส่วนควบคุมการใช้ช่องสัญญาณ จะนำผลจากส่วนแรกไปทำการประมวลผลว่าจะตัดสินให้เทอร์มินัลใดได้ใช้ช่องสัญญาณ แล้วจะส่งผลไปให้ส่วนที่ 3

ส่วนที่ 3 ก็คือ ส่วนที่ทำหน้าที่สวิตช์เทอร์มินัลเข้ากับตัวควบคุม ดูตัวอย่างภาพประกอบดังรูปที่3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



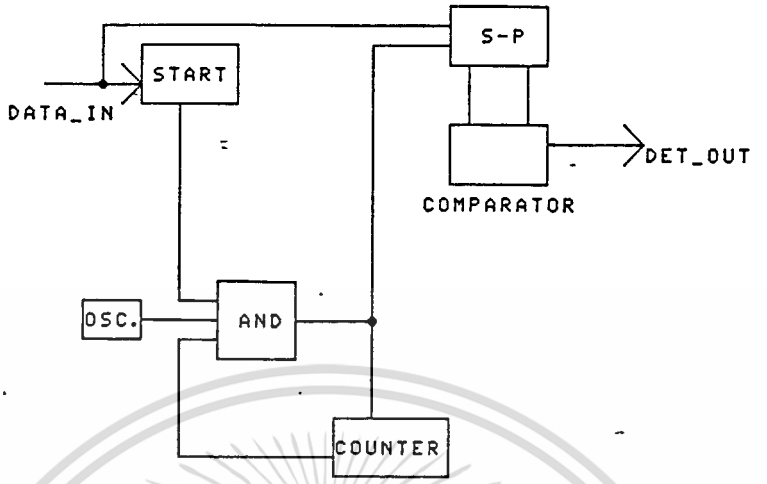
รูปที่3.1แสดงลักษณะความสัมพันธ์ของแต่ละส่วน

ลักษณะการทำงานของวงจรแต่ละส่วนเป็นดังนี้

1. ส่วนตรวจจับการขอใช้ช่องสัญญาณ

ส่วนตรวจจับ (Detection) จะเริ่มต้นทำงาน ด้วยการเคลียร์การทำงานของทุก ๆ ยูนิต (Unit) ก่อน ซึ่งผลจากการเคลียร์จะทำให้ผลของการตรวจจับ (Detect Out) ของแต่ละยูนิตจะมีค่าเป็นไฮ เป็นผลให้สถานะของสัญญาณเอนาเบิล ซึ่งเกิดจากตรรกของการต่อเกท (Gate) แบบโอเพ็นคอลเลกเตอร์ (Open Collector) (74LS09) มีค่าเป็นไฮซึ่งหมายถึงว่า เมื่อทำการเคลียร์ส่วนตรวจจับจะมีผลให้ส่วนตรวจจับพร้อมที่จะทำงานได้ทันทีและการทำงานของส่วนตรวจจับก็คือ เมื่อมีข้อมูลเข้ามานั้น เป็นไปตามที่กำหนด (ในกรณีนี้จะกำหนดให้เป็นเลข5 ฐาน 16 <5_{16}> ซึ่ง จะเป็นรหัสเอ็นไควร์รี่ (ENQ)) ก็จะทำให้ผลการตรวจจับของหน่วยตรวจจับนั้น ๆ มีค่าเป็น โล ซึ่งก็จะทำให้สัญญาณเอนาเบิลเป็นโล ทันที ซึ่งในลักษณะนี้จะทำให้หน่วยตรวจจับทุกหน่วยดิสเอเบิล (Disable) ซึ่งจะหมายความว่า ที่ขณะเวลานั้น หากมีการส่ง ENQ ออกมาจากเทอร์มินัล ใด ๆ ก็ตามเป็น เทอร์มินัล แรกและหน่วยตรวจจับได้แล้วเทอร์มินัล อื่น ๆ จะถูกไม่สนใจอีกซึ่งลักษณะวงจรจะมีดังในรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.2 แสดงลักษณะของวงจรตรวจจับการขอใช้ช่องสัญญาณ

ลักษณะของวงจรสามารถแบ่งเป็นส่วนต่างๆได้ดังนี้

1. ส่วนกำเนิดความถี่ที่มีความสัมพันธ์กับความเร็วของการส่งข้อมูล
2. ส่วนตรวจจับ สตาร์ท บิท (Start Bit)
3. ส่วนแปลงข้อมูลจากแบบอนุกรม (Serial) เป็นแบบขนาน (Parallel) (S-P)
4. ส่วนวงจรนับ
5. ส่วนเปรียบเทียบข้อมูล

1. ส่วนกำเนิดความถี่

ส่วนนี้จะทำการผลิตความถี่ที่ 7200 เฮิร์ท ซึ่งจะถูกนำไปผ่านวงจรหาร 3 อีกทีหนึ่ง และได้ความถี่ที่ 2400 เฮิร์ท ซึ่งเป็นความถี่ที่เราใช้ในการแปลงข้อมูลจากแบบอนุกรมไปเป็นแบบขนาน และจะต้องเท่ากับอัตราการส่งข้อมูลที่เราเซ็ทเอาไว้ การทำงานของวงจรจะใช้ตัวออสซิลเลเตอร์ ซึ่งทำการผลิตความถี่ที่ 1.8432 เมกะเฮิร์ท แล้วนำไปผ่านวงจรหารความถี่ ซึ่งจะหารความถี่ลง 16 เท่า จำนวน 2 ชุด ในโครงงานนี้ได้ใช้ไอซีเบอร์ 74LS93 ซึ่งมีคุณสมบัติของวงจรหาร 16 จำนวน 2 ชุด

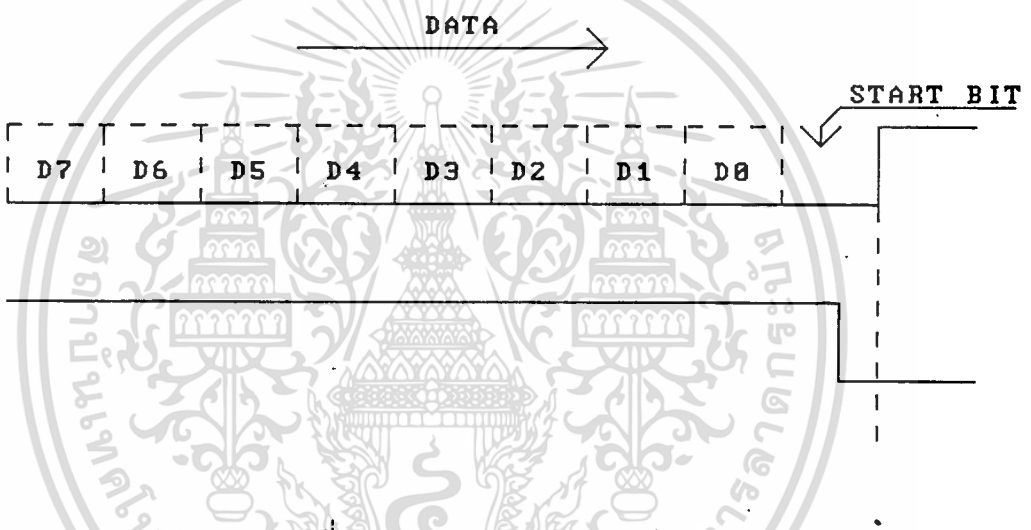
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยเอาท์พุทที่ออกจากวงจรหารความถี่ชุดแรกจะมีค่าเท่ากับ 115200 เฮิร์ท (1.8432MHz/16) ซึ่งเป็นอินพุทของวงจรหารความถี่ชุดที่สอง โดยความถี่ที่ออกมามีค่าเท่ากับ 7200 เฮิร์ท (115200Hz./16) ซึ่งเป็นความถี่ที่เราต้องการ

2. ส่วนตรวจจับสตาร์ทบิท

ในส่วนนี้จะใช้ 74LS73 (U1:A) เป็นตัวตรวจจับโดยกำหนดให้ขา เค (K) เป็นโล ส่วนขา เจ (J) นั้นจะถูกต่อกับเอนาเบิล ซึ่งเมื่อเริ่มทำงาน ครั้งแรกสุดจะมีค่าเป็น ไฮ ดังนั้น เมื่อมีสตาร์ทบิทเข้ามาที่ขาดล็อค จะทำให้ คิว (Q) มีสถานะเป็น ไฮ ดังรูปที่ 3.3

การเปลี่ยนแปลงสถานะของ Q ใน U1:A นี้จะมีผลให้ออสซิเลทเต็ทซิกแนลอินพุท (Oscillator Signal Input) ถูกหารโดย74HCT393และได้สัญญาณ คล็อค (Clock) ออกมาทางขา Q1



รูปที่3.3 แสดงการตรวจจับสตาร์ทบิท

3. ส่วนแปลงข้อมูลจากข้อมูลแบบขนานมาเป็นแบบอนุกรม

สัญญาณคล็อคจาก 74HCT393 (U4:A) จะผ่าน 74LS08 (U5:D) ไปยัง 74HCT164 (U2) ซึ่งจะทำหน้าที่แปลง ข้อมูลจากแบบอนุกรมเป็น แบบขนาน และในขณะเดียวกันสัญญาณคล็อคที่เข้ามาก็ จะถูกควบคุมจำนวนลูก คลื่นด้วยส่วนวงจรนับ

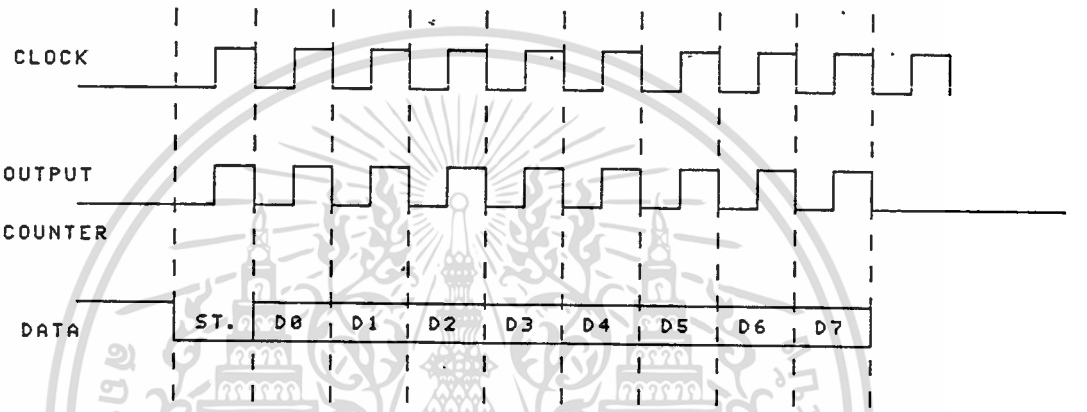
4. ส่วนวงจรนับ

สัญญาณคล็อคที่ออกจาก 74LS08 (U5:D) ส่วนหนึ่งจะถูกป้อนกลับนำ ไปเข้า 74HCT393 (U4:B) ซึ่งจะทำหน้าที่นับจำนวนลูกคลื่นให้ได้ 9 ลูกคลื่น (เนื่องจากเริ่มทำการนับ ตั้งแต่บิทเริ่มต้น 1 บิทและบิทข้อมูลอีก 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิท) โดย U4:B จะทำงานร่วมกันกับ U5:C และ U6:A ซึ่งเมื่อนับ ได้ครบ 9 แล้วจะทำให้ U5:D เป็น โล ทำให้ไม่มีสัญญาณคล็อคผ่านเข้าไปที่ U2 อีก ดังรูปที่ 3.4

เมื่อถึงขั้นนี้แล้ว จะเป็นการสิ้นสุดของการแปลงข้อมูล จากแบบอนุกรมไปเป็นแบบขนาน ซึ่งการทำงานหลังจากนี้ก็จะเป็นหน้าที่ของส่วนเปรียบเทียบข้อมูล



รูปที่3.4 แสดงลักษณะของสัญญาณคล็อค

5. ส่วนเปรียบเทียบข้อมูล

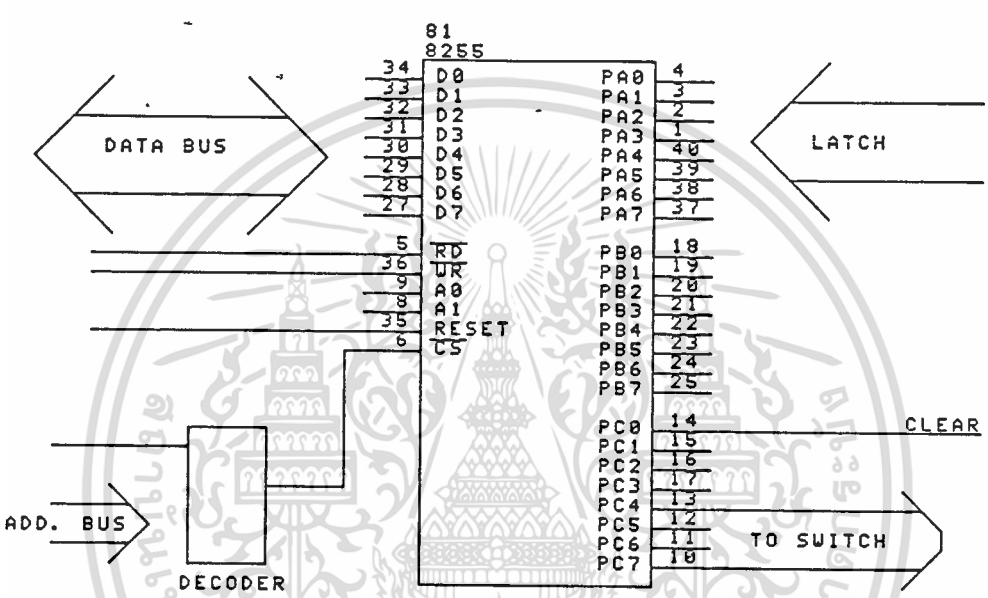
โดยมี74F521(U3)ซึ่งทำหน้าที่เป็นคอมพาราเตอร์ (Comparater)

ซึ่งจะทำการเปรียบเทียบข้อมูลที่ได้จากส่วนที่ 3 กับข้อมูลที่ถูกกำหนดไว้โดยดิพสวิตซ์ (DIP_SWITCH;U7) และจะให้ผลการเปรียบเทียบออกมาเพื่อให้ ส่วนควบคุมอ่านเข้าไปวิเคราะห์ต่อไป

2. ส่วนควบคุมการใช้ช่องสัญญาณ (Channel)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนนี้ประกอบไปด้วยไอซีเบอร์ 8255 เป็นหลัก โดยไอซีจะถูกตั้งให้ทำงานโดยกำหนดให้พอร์ต A เป็นอินพุทพอร์ต (Input port) ซึ่งส่วนพอร์ต B นั้น ในวงจรส่วนแรกจะไม่ใช้ และพอร์ต C นั้นเป็นเอาท์พุทพอร์ทโดยพอร์ตAเป็นพอร์ตที่ทำการรับข้อมูลจากส่วนตรวจจับการขอใช้ช่องสัญญาณเข้ามา เพื่อทำการคัดเลือกเทอร์มินัลที่มีสิทธิใช้ช่องสัญญาณตามวิธีแบบสุ่ม จากนั้นก็จะส่งผล ซึ่งเป็นค่าของเทอร์มินัลที่ได้รับเลือก ให้ใช้ช่องสัญญาณก่อนออกมาทางพอร์ต C ด้านล่าง (Lower) และข้อมูลทางพอร์ตนี้ ก็จะถูกส่งไปให้ส่วนการสวิตช์ต่อไป นอกจากนี้บิทที่ 0 (PC0)ของพอร์ต C ยังทำหน้าที่ส่งสัญญาณเคลียร์ (Clear) ไปให้ส่วนตรวจจับช่องสัญญาณด้วย ลักษณะของวงจรแสดงตามรูปที่3.5



รูปที่3.5 แสดงส่วนควบคุมการใช้ช่องสัญญาณ

3. ส่วนการสวิตช์เทอร์มินัลเข้ากับตัวควบคุมหรือคอมพอร์ตของฮับ(HUB)

ในส่วนนี้จะมีสวิตช์ที่ควบคุมการไหลของข้อมูลอยู่ 2 แบบ คือ

-การไหลของข้อมูลจากเทอร์มินัลเข้าสู่ฮับ (Terminal to HUB)

-การไหลของข้อมูลจากฮับเข้าสู่เทอร์มินัล (HUB to Tetminal)

ในแบบแรกนั้นไม่มีปัญหา เนื่องจากได้เลือกใช้ไอซีเบอร์ 74LS150 ซึ่งเป็น 16 line to 1 line Data Selector แต่สำหรับแบบที่ 2 นั้นก็ไม่สามารถหาไอซีที่ทำหน้าที่เป็น 1 line to 16 line Data Selector ได้ ดังนั้นจึงได้ดัดแปลงโดยนำไอซีเบอร์ 74LS154 ซึ่งทำหน้าที่เป็น 4 line to 16 line Decoder มาใช้แทน โดยพิจารณาจาก Function Table ดังมีภาพประกอบในรูปที่ 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

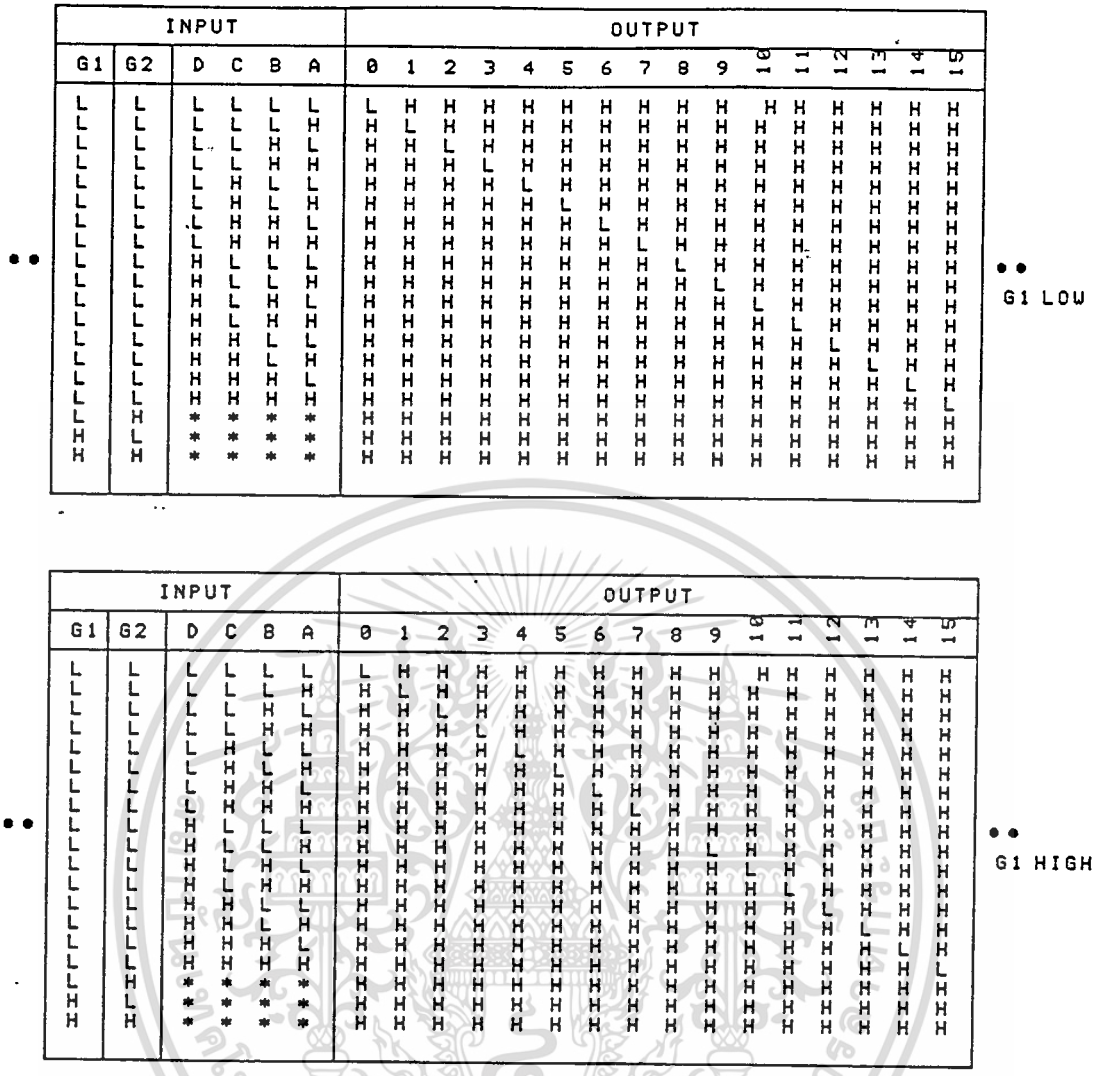
INPUT						OUTPUT													
G1	G2	D	C	B	A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	H	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	H	H	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	H	*	*	*	*	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H

รูปที่3.6 แสดงฟังก์ชั่นเทเบิล

หากสภาวะของ G1 และ G2 เป็นโล ทั้งคู่จะเห็นได้ชัดว่าเอาท์พุทที่ถูกชี้โดยอินพุทจะมีสภาพเป็น โล และหาก G1 หรือ G2 จะเป็น ไฮ ก็จะเห็นได้ว่าได้ว่าเอาท์พุททุกตัวจะกลายเป็น ไฮ ทันที

จากที่เคยกล่าวไว้แล้วว่าในสภาพปกติของสัญญาณ Rx จะมีสภาพเป็นไฮตลอด ดังนั้น หากให้อินพุทชี้ไปที่ค่าใดค่าหนึ่ง สมมุติว่าเป็น 8 จะได้ผลดังรูปที่ 3.7

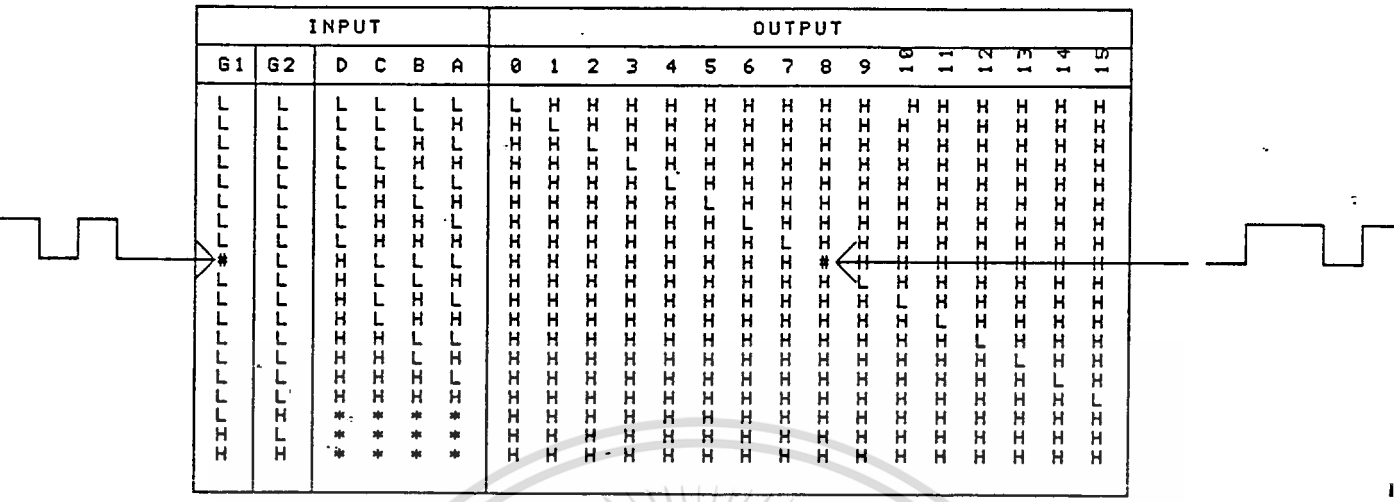
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.7 แสดงฟังก์ชั่นเทเบิลของเอาท์พุทที่ 8

กำหนดให้ G2 เป็นโล และ G1 เป็น low ด้วย จะเห็นว่าเอาท์พุท 8 เป็นโลด้วย และในทางกลับกัน ให้ G1 เป็น ไฮ ด้วย ซึ่งทั้ง ๆ ที่ตามปกตินั้น แล้วกรณีที่เกิดการดีสเอเบิ้ล (Disable) ของไอซีนี้ตามหน้าที่ปกติของมันแต่ เนื่องจากสภาวะนี้กลายเป็นสภาวะปกติของสัญญาณ Rx ด้วยคือ ไม่มีผลกระทบต่อสัญญาณ Rx ของเทอร์มินัลเครื่องอื่น ๆ ดังนั้นหากทำการเปลี่ยนสถานะของ G1 เป็นอะไรก็ตามเอาท์พุทที่ถูกชี้โดยอินพุทก้จะถูกเปลี่ยนแปลงตามไปด้วย ดังเหตุผลดังกล่าวข้างต้น ดังรูปที่3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

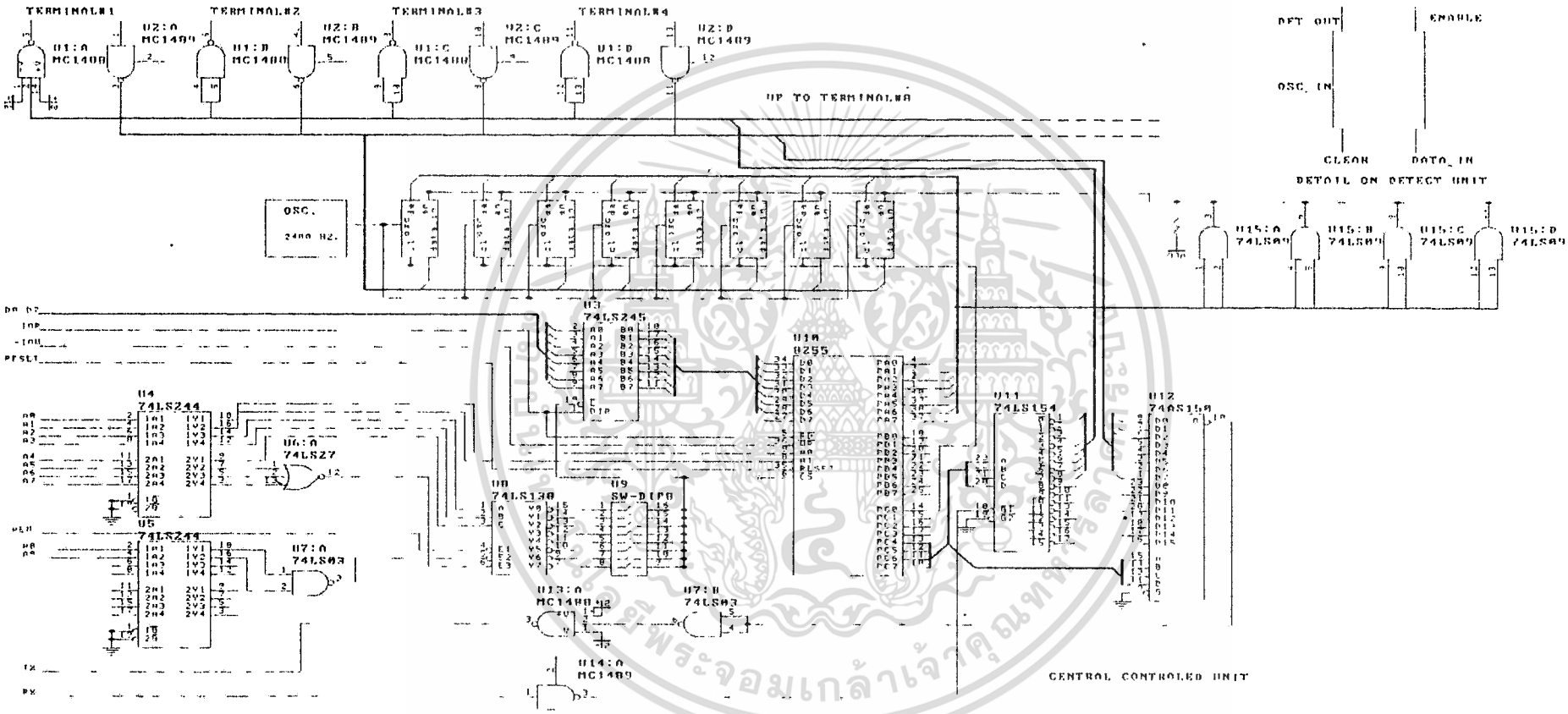


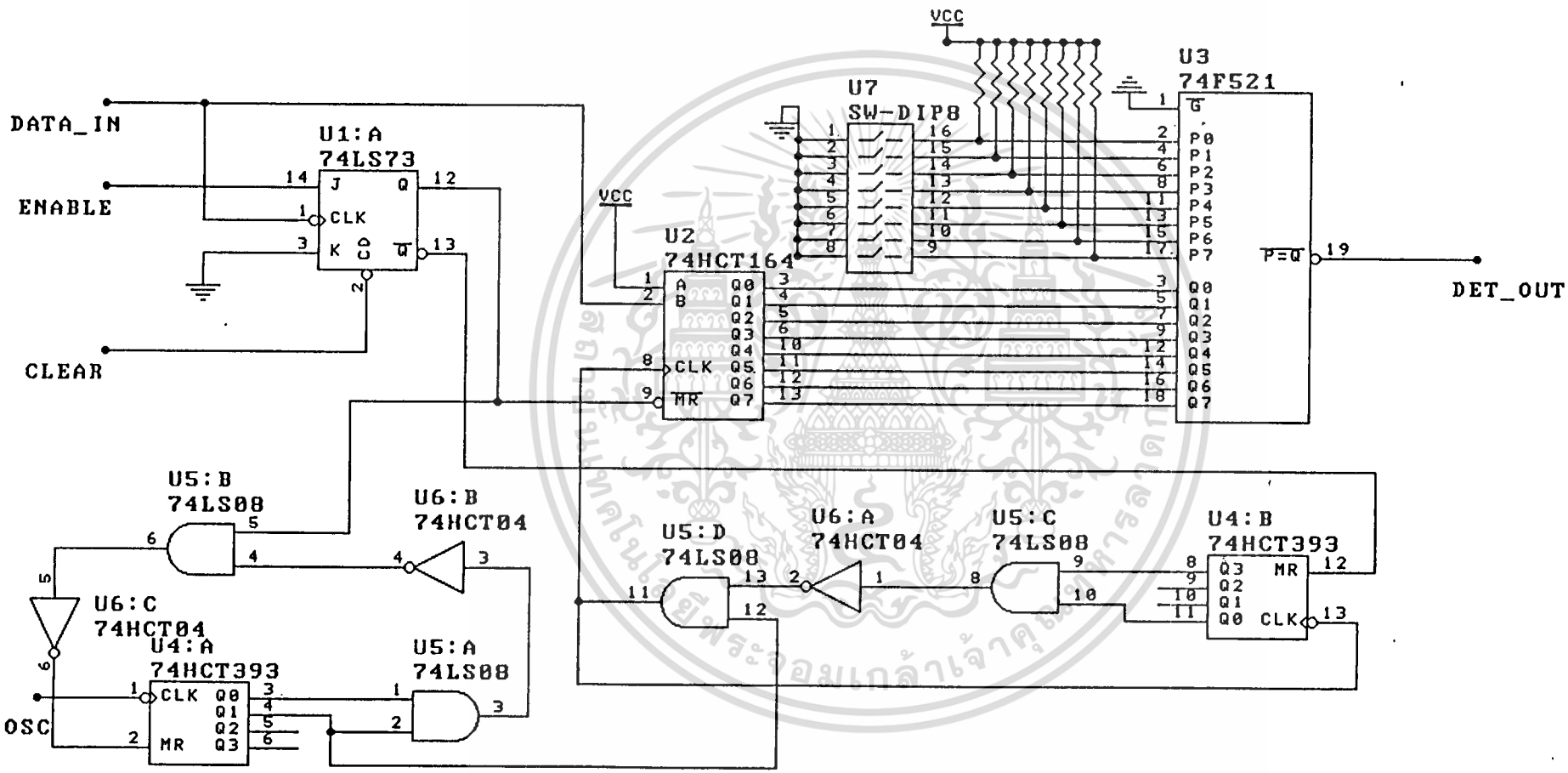
INPUT						OUTPUT													
G1	G2	D	C	B	A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	L	L	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
L	L	H	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H
L	L	H	H	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H
L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L
L	H	*	*	*	*	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H

รูปที่3.8 แสดงฟังก์ชั่นเทเบิลที่เปลี่ยนแปลง

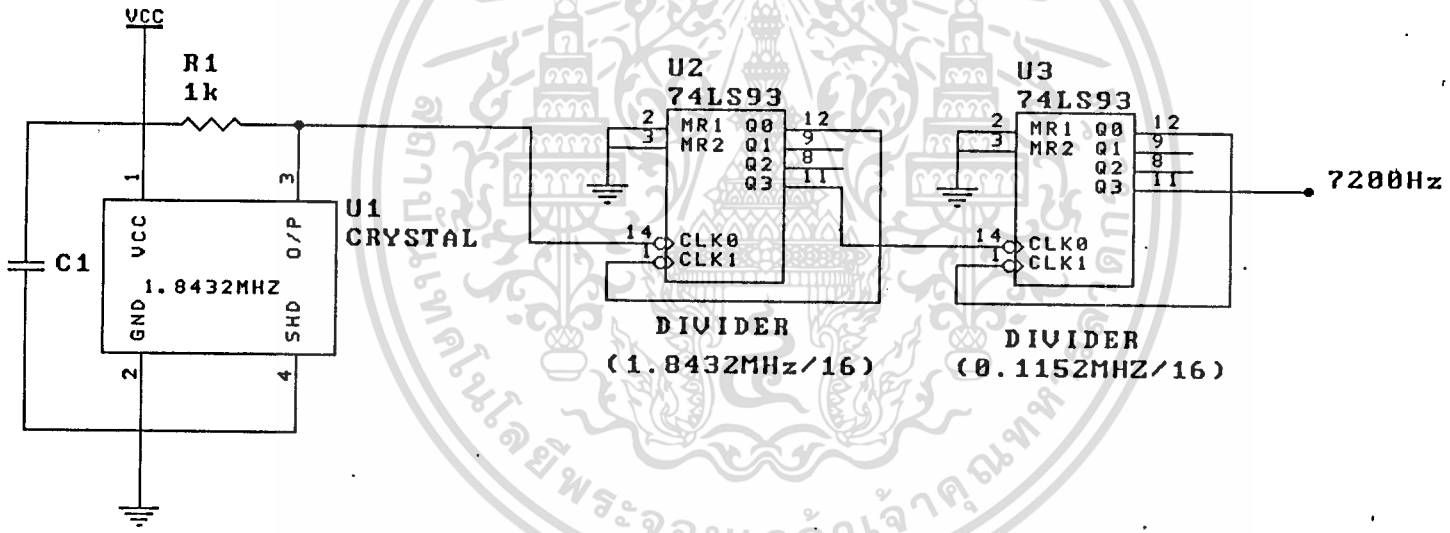
ดังนั้นไอซีตัวนี้จะถูกนำมาทำหน้าที่เป็น1 line to 16 line Data selector โดยจะรับหมายเลขเทอร์มินัลที่จะถูกสวิตช์มาจากส่วนควบคุมอีกที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


TERMINAL#1
TERMINAL#2
TERMINAL#3
TERMINAL#4
U1:A MC1488
U2:A MC1489
U1:B MC1488
U2:B MC1489
U1:C MC1488
U2:C MC1489
U1:D MC1488
U2:D MC1489
UP TO TERMINAL#8
OSC.
2400 HZ.
DET OUT
ENABLE
OSC. IN
CLEAR
DATA IN
DETAIL ON DETECT UNIT
U15:A 74LS00
U15:B 74LS00
U15:C 74LS00
U15:D 74LS00
U3 74LS245
U10 8255
U4 74LS244
U6:A 74LS27
U5 74LS244
U7:A 74LS03
U8 74LS138
U9 SW-DIP8
U11 74LS154
U12 74LS154
U13:A MC1488
U7:B 74LS03
U14:A MC1489
CENTRAL CONTROLED UNIT


DATA_IN
ENABLE
CLEAR
OSC
U1:A
74LS73
U2
74HCT164
U3
74F521
U7
SW-DIP8
VCC
DET_OUT
U5:B
74LS08
U6:B
74HCT04
U6:C
74HCT04
U4:A
74HCT393
U5:A
74LS08
U5:D
74LS08
U6:A
74HCT04
U5:C
74LS08
U4:B
74HCT393

DETECT UNIT


VCC
R1
1k
C1
U1
CRYSTAL
1.8432MHZ
VCC
O/P
GND
SHD
U2
74LS93
MR1
MR2
Q0
Q1
Q2
Q3
CLK0
CLK1
DIVIDER
(1.8432MHz/16)
U3
74LS93
DIVIDER
(0.1152MHZ/16)
7200Hz

OSCILATOR

บทที่ 4
วิธีการทำงานของ Network

ในบทนี้ จะอธิบายวิธีทดลองการทำงานของระบบเน็ตเวอร์ค ซึ่งถูกแบ่งออกเป็น 2 ส่วน คือ ส่วนที่เป็นซอร์ฟแวร์หรือเน็ตเวอร์ค โปรโตคอล และส่วนที่เป็นวงจรฮาร์ดแวร์ ซึ่งทั้ง 2 ส่วนจะต้องทำงานอย่างสอดคล้องกัน

4.1 เน็ตเวอร์ค โปรโตคอล (Network Protocol)

ส่วนต่อไปนี้จะกล่าวถึงการทำงานโดยรวมของเน็ตเวอร์ค ซึ่งจะอธิบายให้เข้าใจในรูปกฎการสื่อสารโปรโตคอล (Protocol) ที่ติดขึ้นสำหรับใช้ในเน็ตเวอร์ค นี้ โดยใช้หลักการของเลเยอร์ โปรโตคอล (Layer Protocol)

การวิจัยนี้ สามารถแบ่งการทำงานของ เทอร์มินอลออกได้เป็น 3 ชั้น คือ

1. ฟิสิคัล เลเยอร์ (Physical Layer)
2. ดาต้าลิงค์ เลเยอร์ (Datalink Layer)
3. แอปปลิเคชั่น เลเยอร์ (Application Layer)

ส่วนการทำงานของเซ็นเตอร์นั้น สามารถแบ่งได้เป็น 4 ชั้น โดยมีการเพิ่มรูติ้ง เลเยอร์ (Routing Layer) เข้าไปในชั้นล่างของ ฟิสิคัล เลเยอร์ อีกชั้นหนึ่ง

การทำงานของฟิสิคัลเลเยอร์ และ ดาต้าลิงค์เลเยอร์ นั้น จะเหมือนกันทั้งในเทอร์มินอลและเซ็นเตอร์ แต่แอปปลิเคชั่นในเซ็นเตอร์และเทอร์มินอลนั้นจะแตกต่างกัน ดังนั้นจึงขออธิบายการทำงานของชั้นฟิสิคัลและดาต้าลิงค์ก่อน

ฟิสิคัล เลเยอร์ (Physical Layer)

ชั้นนี้จะเป็นชั้นเดียวใน 4 ชั้นที่เครื่อง ไอบีเอ็ม-พีซี มีอุปกรณ์ที่จะต้องใช้อยู่แล้ว คือ ยูอาร์ต (UART) 8250 ซึ่งเป็นอุปกรณ์หลักในการทำงานของชั้นนี้ โดยนำฟังก์ชั่นเหล่านี้มาใช้คือการเปลี่ยนข้อมูลไปมาระหว่าง ซีรียล อะซิงโครนัส กับพาราเลล (Parallel) เปลี่ยน บอดเรต (Baud Rate) เลือกรูปแบบของข้อมูลอะซิงโครนัส ส่งสัญญาณ ฮาร์ดแวร์ อินเทอร์รัพท์ (Hardware Interrupt) เมื่อข้อมูลพร้อมตรวจสอบพาริตี้ เออเรอร์ (Rarity Error) วิธีการควบคุมการทำงานฟังก์ชั่นเหล่านี้ในเน็ตเวอร์คจะอยู่ในส่วนของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่ของชั้นฟิสิคัลในเน็ตเวอร์คนี้ คือ

-เปลี่ยนข้อมูลระหว่าง ซีเรียล ในทวิสท์-แพร์ วาร์ย กับข้อมูล พราเลล และใช้ความเร็วข้อมูลใน ทวิส-แพร์ วาร์ย ที่ 2,400 บอด

- ทำการสื่อสารแบบ ฮาร์ฟ ดูเพล็ก (Half Duplex)

- ตรวจสอบพาริตี้ (Parity)

- เซทแฟลก (Set Flag) เพื่อบอกว่ามีข้อมูลเข้ามา

ดาต้าลิงค์ เลเยอร์ (Datalink Layer)

ใช้ในฐานะที่เป็นมีเดีย แอคเซส โปรโตคอล จึงมีหน้าที่ดังนี้

- คอยรับไบท์ข้อมูลจากชั้นฟิสิคัล มาประกอบเป็นแพคเก็ท (Packet) เก็บไว้ในบัฟเฟอร์ (Buffer) แล้วรอให้แอปปลิเคชั่นมาอ่าน

- รับแพคเก็ทจากแอปปลิเคชั่น และส่งต่อไปให้ฟิสิคัลทีละไบท์

- ควบคุมไบท์ ซิงโครนัส(Byte Sychronous) โดยใช้หลักการสตอบ แอนด์ เวท(Stop and Wait) เพื่อป้องกันการโอเวอร์รัน เออเรอร์ (Overrun Error)

หลักในการรับส่งข้อมูลระหว่าง ดาต้าลิงค์ เลเยอร์ กับ ฟิสิคัล เลเยอร์

ดาต้าลิงค์ เลเยอร์ จะเป็นตัวทำ มีเดีย แอคเซส ซึ่งในเน็ตเวอร์คนี้จะเป็นแบบ เซ็นทรัลลี่ คอนโทรล ดีมานด์ แอสไซน์เมนท์ เทคนิก โดย เทอร์มินอล ที่มีแพคเก็ทจะส่งจะต้องแย่งชันแนลจาก เซ็นเตอร์ โดยฮาร์ดแวร์ในชั้นรูทติ้งจะเป็นตัวบังคับให้ต้องใช้เทคนิกนี้ ซึ่งมีลักษณะดังนี้

การรับแพคเก็ท ก่อนจะรับแพคเก็ทข้อมูล ชั้นฟิสิคัลจะทำการขอการเชื่อมติอมายังชั้นดาต้าลิงค์ ด้วยการส่งเอ็นไควร์ลี่(ENQ)มาอย่างต่อเนื่อง ชั้นดาต้าลิงค์จะต้องตอบรับโดยการส่งแอคโนเลดจ์(ACK)ไปให้ชั้นฟิสิคคัล หลังจากนั้น รหัสตัวต่อไปจากชั้นฟิสิคัลจึงถือเป็นข้อมูล และเมื่อชั้นฟิสิคัลเซ็ทแฟลกบอกว่าได้รับข้อมูลแล้ว ชั้นดาต้าลิงค์จึงอ่านข้อมูล 1 ไบต์นั้น พร้อมทั้งส่งรหัสแอคโนเลดจ์ไปให้ฟิสิคัล 1 ไบต์ ชั้นดาต้าลิงค์จึงจะได้รับตัวต่อไป ขบวนการจะเป็นเช่นนี้ไปจนกว่าจะรับหมดทั้งแพคเก็ท

การส่งแพคเก็ท ก่อนจะส่งแพคเก็ทข้อมูลจะต้องทำการแย่งชันแนลของฮับกับเทอร์มินัลอื่นๆ โดยการส่งเอ็นไควร์ลี่ไปให้ชั้นฟิสิคัลอย่างต่อเนื่องจนกว่าจะได้รับแอคโนวเลดจ์จากชั้นฟิสิคัล จึงสามารถส่งข้อมูลไบต์แรกไปให้ชั้นฟิสิคัลได้ ต่อจากนั้นจะต้องรอรับแอคโนวเลดจ์อีก จึงจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถส่งข้อมูลไบต์ต่อไปให้ชั้นฟิสิคัลได้ ขบวนการจะเป็นเช่นนี้จนกว่าจะส่งหมดทั้งแพคเก็ท

แอปปลิเคชั่น เลเยอร์ (Application Layer)

หน้าที่ของแอปปลิเคชั่นเลเยอร์ มี 3 หน้าที่ คือ

1. รอรับแพคเก็ทจากชั้นดาต้าลิงค์ แล้วนำข้อมูลนั้นมาแยกจัดตามความเหมาะสม เช่น เก็บข้อมูลลงแผ่น แสดงผลทางจอ หรือส่งแพคเก็ทชนิดอื่นๆตอบต้อกลับไป

2. นำข้อมูลที่จะส่งมาแบ่งเป็นแพคเก็ท แล้วส่งไปให้ชั้นดาต้าลิงค์

3. หน้าที่ที่ 3 นี้ จะเป็นแอปปลิเคชั่นของฮับเท่านั้น ซึ่งมีหน้าที่ คือ ต้องพิจารณาว่าแพคเก็ทที่รับมานั้นเป็นของตัวเอง หรือเป็นแพคเก็ทที่จะส่งไปให้เทอร์มินัลอื่น เพื่อที่จะจัดการเก็บข้อมูลนั้นได้ถูกต้อง หรือจะส่งแพคเก็ทกลับไปให้ชั้นดาต้าลิงค์ส่งไปให้เทอร์มินัลต่อไป

รูทติ้ง เลเยอร์ (Routing Layer)

ชั้นนี้เป็นเอกลักษณ์พิเศษของฮับ ฮับจะเปรียบเสมือนกับมีเดียของเน็ทเวอร์ค และเทอร์มินัลคู่ใดต้องการสื่อสารกันจะต้องส่งข้อมูลผ่านฮับ ในช่วงขณะหนึ่งๆจะมีเพียง 1 เทอร์มินัลเท่านั้นที่สามารถเชื่อมต่อ เพื่อส่งหรือรับข้อมูลกับฮับ หน้าที่ของชั้นรูทติ้งก็คือ เลือกการเชื่อมต่อเทอร์มินัลเข้ากับฮับ เพราะฉะนั้นชั้นอื่นๆของฮับก็จะไม่ต้องสนใจกับการสวิชชิ่ง เนื่องจากว่า เมื่อแพคเก็ทผ่านชั้นแอปปลิเคชั่นมาจนถึงชั้นฟิสิคัลแล้ว ชั้นรูทติ้งจะเป็นชั้นที่ดูแลให้ข้อมูลไปถึงเทอร์มินัลที่ถูกต้อง และชั้นรูทติ้งยังจะคอยดูแลเลือกรับข้อมูลจากเทอร์มินัลที่สามารถส่งข้อมูลตามกฎของมีเดีย แอคเซส โปรโตคอล ที่กำหนดขึ้น ซึ่งได้กล่าวไว้แล้วในชั้นดาต้าลิงค์

ชั้นรูทติ้งนี้ เป็นส่วนของอาร์ดแวร์ที่ทำขึ้น ซึ่งจะทำงานร่วมกับซอร์ฟแวร์ ในฮาร์ดแวร์จะมีส่วนหนึ่งที่เรียกว่าแลตว์ยูนิท (Latch Unit) ซึ่งทำหน้าที่เป็นตัวบอกซอร์ฟแวร์ว่าเทอร์มินัลใดสามารถได้ชันแนลสื่อสาร จากนั้นซอร์ฟแวร์จึงจะสั่งให้ฮาร์ดแวร์อีกส่วนหนึ่งที่เรียกว่าสวิชชิ่งทำการเชื่อมต่อและรับข้อมูลกับเทอร์มินัลนั้นๆ สำหรับการอินเตอร์เฟส (Interface) ระหว่างชั้นรูทติ้งกับชั้นฟิสิคัลนั้นจะใช้ซีเรียลพอร์ต (Serial Port) เพราะฉะนั้นข้อมูลชั้นรูทติ้งที่จัดให้ชั้นฟิสิคัลของฮับ ก็จะเป็นแบบอะซิงโครนัสเช่นเดียวกับชั้นฟิสิคัลในเทอร์มินัล ส่วนวิธีการโปรแกรมควบคุมชั้นรูทติ้งจะอยู่ในบทที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ขั้นตอนการสื่อสาร

1. เครื่อง คอมพิวเตอร์ ที่เป็นฮับจะทำการตั้งค่าพารามิเตอร์สำหรับการสื่อสาร ซึ่งก็มี ค่าความเร็ว ให้เป็น 2400 bit/sec ความยาวข้อมูล 8 bit 1 stop bit และทำการควบคุมการ์ด Network ให้ IC8255 มี Port A & Port B เป็น Input Part Part C เป็น output Part จากนั้น Center ก็จะรอการร้องขอจาก Terminal ต่างๆ

2. ทางด้าน Terminal จะเริ่มต้นด้วยการ Run Programe ของตนเองเพื่อเลือก Application ที่ตนสนใจ ซึ่งจะแยกการอธิบายออกเป็นหัวข้อย่อยๆ ดังต่อไปนี้

2.1 เลือก Video text ในการติดต่อ Program จะทำการเปิด ไฟล์ค้นหา เพื่อให้ Terminal นั้นเลือกบุคคลที่ตนสนใจ จากนั้นจะ set ค่า command = 1 และนำชื่อไฟล์ ไปเก็บไว้ใน buffer

2.2 เลือก Talk to Center ผู้ใช้ Terminal นั้นจะเขียนข้อความที่ตนเองต้องการจะสื่อสารกับ center จากนั้นจะ set ค่า command = 2 และนำข้อมูลเก็บไว้ใน buffer

2.3 เลือก File Transfer ผู้ใช้จะเลือกชื่อ ไฟล์ที่ต้องการได้จาก center เก็บไว้ใน buffer และ set ให้ค่า command = 3

2.4 เลือก Email ผู้ใช้ทำการพิมพ์จดหมายลงใน block ข้อความเพื่อนำไปส่งยัง terminal ที่ต้องการ จากนั้น จะเก็บเนื้อความจดหมายไว้ใน buffer แล้วทำการ set ค่า command ให้เท่ากับ 4

2.5 เลือก Others เข้าสู่โปรแกรมหมอดู ผู้ใช้เขียนวัน เดือน ปีเกิดของตนเอง และโปรแกรมจะทำหน้าที่เลือกไฟล์ให้ เก็บชื่อไฟล์นี้ไว้ใน buffer และให้ command = 5

3. byte ต่างๆ จะถูกรวมเข้าเป็น message จากนั้น terminal จะทำการ initial ค่าต่างๆ ของการสื่อสาร ให้ตรงกับ center

4. Terminal จะส่งการร้องขอไปยัง center หากได้รับการตอบรับแล้ว ก็จะส่ง message ออกไปให้ center ทันที โดยใช้วิธี stop and wait

5. Center รับ message ได้แล้วทำการถอด message นั้นออกแล้วเก็บ data ไว้ใน buffer จากนั้นตรวจสอบ command เพื่อให้ทราบว่าขณะนั้น terminal ที่ส่ง message มากำลัง run application ใดอยู่

6. Center จะทำการอ่าน "dest" ว่าต้องการส่งต่อไปยัง terminal อื่น หรือไม่ หรือต้องการส่งให้กับ center เอง และทำการ access data ที่ได้มา เช่น นำไปเปิดไฟล์ หรือ นำ data ที่ได้มาออกที่ monitor เพื่อให้ center พิมพ์ตอบสนองไป หรือนำ data นั้นไปส่งต่อให้กับ terminal อื่นๆ

7. Message ที่ center เตรียมไว้นั้นจะถูกส่งไปยัง terminal ก็ต่อเมื่อได้รับสัญญาณตอบรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก terminal นั้นแล้ว

8. หลังจากจบการสื่อสารกับ terminal นั้นแล้ว center จะกลับไป clear ตัวเอง เพื่อให้พร้อมที่จะรับข้อมูลจาก terminal อีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
หน้าที่และความสำคัญของโปรแกรม

โปรแกรมที่เขียนขึ้นนั้น ถูกเขียนโดยภาษาซี (สาเหตุที่เขียนโปรแกรมโดยใช้ภาษาซี เพราะว่า ภาษาซีเป็นภาษาระดับกลางที่สามารถแอคเซสกับฮาร์ดแวร์ของพีซีได้ดีพอ ๆ กับแอสเซมบลี้ นอกจากนั้นภาษาซียังเป็นภาษาที่เขียนและสามารถพัฒนาโปรแกรมได้ง่ายกว่าแอสเซมบลี้)

สำหรับหน้าที่ของโปรแกรมนั้น จะแบ่งออกเป็นหัวข้อย่อย ๆ ดังนี้

5.1 หน้าที่ในการควบคุมชั้น รูทติ้ง ในตัวเซ็นเตอร์

เนื่องจากการสร้างฮาร์ดแวร์ขึ้นใช้งานนั้น จำเป็นที่จะต้องมีซอฟแวร์เข้ามาควบคุมอีกชั้นหนึ่งในส่วนของรูทติ้ง ดังนั้นซอฟแวร์ในส่วนแรกของเซ็นเตอร์จึงถูกเขียนขึ้น ซึ่งมีรายละเอียดการทำงานดังต่อไปนี้

1) เนื่องจากฮาร์ดแวร์นั้นใช้ไอซี-8255 เราจึงจำเป็นที่จะต้องควบคุมในส่วนนี้ด้วย โดยใช้คำสั่ง outportb(0x303,0x92);

'0x303' คือ คอนโทรลพอร์ตของไอซี 8255 ถูกบังคับจากการออกแบบทางฮาร์ดแวร์

'0x92' คือ จะเป็นการกำหนดหน้าที่ให้พอร์ตเอ และพอร์ตบีเป็นพอร์ตอินพุต และพอร์ตซีเป็นพอร์ตเอาท์พุต

2) outportb(0x302,0xF0); เป็นการส่งคำสั่งไปยังพอร์ตซี เพื่อทำการเคลียร์ส่วนของการรับการร้องขอ โดย '0x302' นั้น หมายถึงพอร์ตซีของไอซี 8255 ด้วยเหตุผลเดียวกันกับที่ได้อธิบายไปแล้วในข้อ 1 นั้นเอง

3) outportb(0x302,0xF1); เป็นการส่งคำสั่งไปยังพอร์ตซี เพื่อทำการ เตรียมพร้อมที่จะรับการร้องขอจากเทอร์มินอลต่าง ๆ

4) do {

```
switch(inportb(PORT_A))
 {
   case 0x7F: outportb(PORT_C,0x01);
        outportb(com,ACK); break;
   case 0xBF: outportb(PORT_C,0x11);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                outportb(com,ACK); break;
            case 0xDF: outportb(PORT_C,0x21);
                outportb(com,ACK); break;
            case 0xEF: outportb(PORT_C,0x31);
                outportb(com,ACK); break;
        }
    }while(inportb(PORT_A)==0xFF);
```

จากโปรแกรมข้างต้นเป็นส่วนสำคัญในการประมวลผล โดยตรวจเช็คจากพอร์ตเอ ของ 8255 ซึ่งสถานะปกติจะมีค่าเป็น 0xFF แต่หากเทอร์มินอลที่ 1 ร้องขอมา ค่าในพอร์ตเอจะเป็น 0x7F หรือถ้าเทอร์มินอลที่ 2 ร้องขอมาค่าในพอร์ตเอ จะเป็น 0xBF แทนเป็นต้น หลังจากที่สามารถตรวจจับการร้องขอจากเทอร์มินอลได้แล้ว ซอฟแวร์จะสั่งให้เชื่อมเส้นทางเข้ากับเทอร์มินอลนั้น แล้วส่ง 'ACK' กลับไปให้เทอร์มินอลด้วย เพื่อให้เทอร์มินอลรับรู้ว่าสามารถเชื่อมต่อระหว่างเซ็นเตอร์กับเทอร์มินอลนั้นได้แล้ว

5.2 หน้าที่ในการปรับค่าพารามิเตอร์ของการสื่อสารทั้งระบบให้ตรงกัน

นั่นคือ ทำการตั้งค่าบอดเรท ความยาวของข้อมูล, ความยาวสต๊อปบิทและพาริตี้บิท ดังต่อไปนี้

```
Set_Comm() {
    outportb(com+3,80); /*ปรับค่า DLAB เป็น 1*/
    outportb(com,30); /*ปรับบอดเรทให้เท่ากับค่า 2,400 บอด*/
    outportb(com+1,00);
    outportb(com+3,03); /*เป็นการกำหนดความยาวข้อมูลเท่ากับ 8 สต๊อปบิทเท่ากับ 1 และไม่
                          มีพาริตี้*/
}
```

โดยให้ com เท่ากับค่า 0x3F8 ในกรณีที่ใช้คอมมิวนิเคชั่นพอร์ตที่ 1 และเท่ากับค่า 0x2F8 ในกรณีของพอร์ตที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 หน้าที่ในการรับและส่งข้อมูล

เนื่องจากเป็นการรับและการส่งข้อมูล โดยการใช้หลักการสต๊อปแอนด์เวท เพื่อประโยชน์ในการป้องกันการเกิดโอเวอร์รันเออเรอร์

1) การส่งข้อมูล จากข้อมูลที่อยู่ในลักษณะเป็นไบท์ เราจะนำมาต่อกันเป็นสตรีม ซึ่งเป็นรูปแบบที่ถูกกำหนดไว้แล้ว ดังนี้

ไบท์ที่ 0	=	SYN
ไบท์ที่ 1	=	ต้นทาง
ไบท์ที่ 2	=	ปลายทาง
ไบท์ที่ 3	=	คำสั่ง
ไบท์ที่ 4 ขึ้นไป	=	ข้อมูล
3 ไบท์สุดท้าย	=	ETX

จากนั้นจะทำการสั่ง โดยการส่งข้อมูลไป 1 ไบท์ แล้วรอตรวจเช็คที่ไบท์ สถานะของ 8250 ในบิทที่ 0 หากเป็น '1' แสดงว่ามีข้อมูลเข้ามา หากเป็น '0' แสดงว่าข้อมูลนั้นถูกอ่านไปแล้ว เมื่อทราบว่ามีข้อมูลเข้ามา จะตรวจว่าเป็น 'ACK' หรือไม่ ถ้าใช่จะทำการส่งข้อมูลในไบท์ต่อไป จนกระทั่งข้อมูลถูกส่งออกไปจนหมดแสดงได้ดังนี้

```
Send_Command_Packet(command,data)
    nt command;
    char *data;
{
     buff_send[0] = SYN;
     buff_send[1] = scr;
     buff_send[2] = dest;
     buff_send[3] = command;
     strcpy(&buff_send[4],data);
     buff_send[end+4+1] = ETX;
     buff_send[end+4+2] = ETX;
     buff_send[end+4+3] = ETX;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for(count=0;count<end+7;count++)
        {
            outportb(com,buff_send[count]);
            while(!(inportb(com+5)&0*01));
            while(inportb(com)!=ACK);
        }
}
```

2) การรับข้อมูล จะทำการรับข้อมูลทีละ 1 ไบท์ สลับกับการส่ง 'ACK' จนกระทั่งได้รับ ETX ครบ 3 ไบท์ ดังนี้

```
Receive_Command_Packet(command,data)
int command;
char *data;
{
        int count,stop=0;
        buff_rec[0] = NULL;
        for(count=0;stop!=3;count++)
            {
                if((inportb(com) = buff_send[count])==ETX) stop++;
                do {
                        outportb(com,ACK);
                } while(inportb(com)!=ENQ);
            }
}
```

5.4 หน้าที่ในการบันทึกและการแสดงผล ในรูปแบบต่าง ๆ ได้แก่

1) วีดีโอเท็กซ์ เป็นโปรแกรมที่สามารถแสดงได้ทั้งตัวอักษร และภาพโดยโปรแกรมที่เขียนขึ้นนั้น เป็นการจำลองข้อมูลของนักศึกษาภาคโทรคมนาคม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยแบ่งโปรแกรมนี้ออกเป็น 2 ส่วน

- ส่วนแรกจะอยู่ที่เทอร์มินอล คือ จะมีชื่อและเลขประจำตัวให้เลือก หลังจากเลือกแล้ว จะถูกส่งเป็นชื่อไฟล์ที่ร้องขอจากเซ็นเตอร์ เมื่อเซ็นเตอร์ส่งไฟล์นั้นมาให้เป็นที่เรียบร้อย จะถูกนำออกมาแสดงผล

- ส่วนที่สอง จะอยู่ที่เซ็นเตอร์ คือ เป็นไฟล์ภาพ และประวัติของนักศึกษาแต่ละคน

จะแสดงโปรแกรมส่วนแรก ดังนี้

ส่วนของการแสดงชื่อและเลขประจำตัว

```
search_data(int gnd,int foregnd)  {
FILE *fptr;
register int x1=15,y1=6,x2=65,i;
register char countrecord,count;

        if((fptr = fopen("person.dat","r+b")) == NULL)       /*เปิดไฟล์เพื่ออ่าน*/
        {
                warnning(30,"ไม่สามารถเปิดไฟล์ PERSON.DAT ได้");
                return FALSE;
        }
    /* เป็นการอ่านชื่อนักศึกษาทั้งหมดที่มีโดยมีจำนวนนักศึกษา = numrecord */
        for(countrecord = 0;countrecord < numrecord;countrecord++)
        {
                /*อ่านไฟล์ไปเก็บไว้ใน structure Datarec*/
                fread(&Datarec,sizeof(struct datatype),1,fptr);
                textbackground(gnd);textcolor(foregnd);
                /*for loop นี้ วน 2 รอบเพื่ออ่านชื่อ 1 ครั้งและID. 1 ครั้ง */
                for(count = 0;count < 2;count++)
                        {
                        /*copy ข้อมูลจาก Datarec.Info ไปไว้ใน Search_Dat */
                                strcpy(Search_Dat[countrecord].Info[count],Datarec.Info[count]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    if(countrecord<14)
                    {
                        gotoxy(x1+5+(30*count),y1+3+countrecord);
                        cprintf("%s",Datarec.Info[count]);
                    }
                }
                /* Seek position of fptr from start of file */
                fseek(fptr,countrecord*sizeof(Datarec),SEEK_SET);
                fwrite(&Datarec,sizeof(struct datatype),1,fptr);
            }
            fclose(fptr);
            textbackground(BLACK); textcolor(YELLOW);
            for(i=x1+1;i!=x2-1;i++) {  gotoxy(i,y1+3); cprintf(" "); }
            for(count=0;count<2;count++)
            {
                gotoxy(x1+5+(30*count),y1+3);
                cprintf("%s",Search_Dat[0].Info[count]);
            }
            return TRUE;
        }
```

/* เป็นโปรแกรมที่ใช้ในการเลื่อนแถบเพื่อเลือกชื่อและID.ที่เราสนใจ */

```
run_search()
{
register int x1=15,x2=65,y1=6,y2=23,i,key,y;
register char count = 0,countrecord = 0;
y = y1+3;
do {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
key = bioskey(0);
if(key==UP) {
/*if(y==23) y=22;*/
if(countrecord==0) putch(0x07);
else {
        textbackground(RED); textcolor(WHITE);
        for(i=x1+1;i!=x2-1;i++) {  gotoxy(i,y); cprintf(" "); }
        for(count=0;count<2;count++)
        {
                gotoxy(x1+5+(30*count),y);
                cprintf("%s",Search_Dat[countrecord].Info[count]);
        }
        if(y==y1+3)
        { movetext(x1+1,y1+3,x2-1,y2-2,x1+1,y1+4); y++;
}
y--;
countrecord--;
textbackground(BLACK); textcolor(YELLOW);
for(i=x1+1;i!=x2-1;i++) {  gotoxy(i,y); cprintf(" "); }
for(count=0;count<2;count++)
{
        gotoxy(x1+5+(30*count),y);
        cprintf("%s",Search_Dat[countrecord].Info[count]);
}
}
}
if(key==DOWN) {
if(countrecord==numrecord-1) putch(0x07);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else {
                textbackground(RED); textcolor(WHITE);
                for(i=x1+1;i!=x2-1;i++) {  gotoxy(i,y); cprintf(" "); }
        for(count=0;count<2;count++)
        {
                gotoxy(x1+5+(30*count),y);
                cprintf("%s",Search_Dat[countrecord].Info[count]);
        }
        if(y==y2-1)
        { movetext(x1+1,y1+4,x2-1,y2-1,x1+1,y1+3); y--;
        }
        y++;
        countrecord++;
        textbackground(BLACK); textcolor(YELLOW);
        for(i=x1+1;i!=x2-1;i++) {  gotoxy(i,y); cprintf(" "); }
        for(count=0;count<2;count++)
        {
                gotoxy(x1+5+(30*count),y);
                cprintf("%s",Search_Dat[countrecord].Info[count]);
        }
        }
        }
        if(key==ESC) return FALSE;
        }while((key!=ENTER)&&(key!=ESC));
        display(countrecord);
}
```

ส่วนของการแสดงผลออกทางจอภาพ (ทั้งตัวอักษร และภาพ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
display_image(text)
char *text;
{
FILE *fp;
static char results[8][16] = {"Ok",
"Bad file",
"Bad read",
"Memory error",
};
char path[81];
int r;

strmfe(path,text,"BMP");   /* เป็นส่วนที่ใช้ในการเพิ่ม ".bmp" */
strupr(path);              /*เพิ่ม path */
if((fp = fopen(path,"rb")) != NULL) {
fi.setup=dosetup;
fi.closedown=doclosedown;   /* เป็นส่วนที่แสดงภาพและข้อมูล */
r=unpackbmp(fp,&fi);
printf("\%s",results[r]);
fclose(fp);
} else printf("Error opening %s",path);
}
/* this function is called before an image is unpacked */
dosetup(fi)
_FILEINFO *fi;
{
_union REGS r;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
_if(!getbuffer((long)fi->width*(long)fi->depth,fi->width,fi->depth))
_   return(MEMORY_ERROR);

r.x.ax=0x0013;
_int86(0x10,&r,&r);

setvgapalette(fi->palette,256,fi->background);

_return(GOOD_READ);
}

/* This function a called after an image has been unpacked. It must
   display the image and deallocate memory. */
doclosedown(fi)
_FILEINFO *fi;
{
union REGS r;
int c,i,n,x=0,y=0;

if(fi->width > SCREENWIDE) n=SCREENWIDE;
else n=fi->width;

do {
for(i=0;i<SCREENDEEP;++i) {
c=y+i;
if(c>=fi->depth) break;
memcpy(MK_FP(0xa000,SCREENWIDE*i),getline(c)+x,n);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Display_Text();
c=GetKey();
switch(c) {
case CURSOR_LEFT:
if((x-STEP) > 0) x-=STEP;
____else x=0;
____break;
___case CURSOR_RIGHT:
____if((x+STEP+SCREENWIDE) < fi->width) x+=STEP;
____else if(fi->width > SCREENWIDE)
____    x=fi->width-SCREENWIDE;
____else x=0;
break;
___case CURSOR_UP:
____if((y-STEP) > 0) y-=STEP;
else y=0;
____break;
___case CURSOR_DOWN:
if((y+STEP+SCREENDEEP) < fi->depth) y+=STEP;
____else if(fi->depth > SCREENDEEP)
____    y=fi->depth-SCREENDEEP;
____else y=0;
____break;
___case HOME:
____x=y=0;
____break;
___case END:
____if(fi->width > SCREENWIDE)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
____    x=fi->width-SCREENWIDE;
____else x=0;
if(fi->depth > SCREENDEEP)
____    y=fi->depth-SCREENDEEP;
else y=0;
break;
__}
_} while(c != 27);

_freebuffer();

_r.x.ax=0x0003;
_int86(0x10,&r,&r);
_return(GOOD_READ);
}

Display_Text(int countrecord)
{
FILE*fptr;
register int x1,y1,x2,y2,i;
register char count;

if((fptr = fopen("person.dat","r")) == NULL)
{
warnning(25,"ไม่สามารถเปิดไฟล์ PERSON.DAT ได้");
return FALSE;
}
/* จัดขอบของเมนู */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(Num_of_Fields%2 == 0)y1 = 15-(Num_of_Fields+6)/2;
else y1 = (15-(Num_of_Fields+6)/2)-1;
y2 = 15+(Num_of_Fields+6)/2;
x1 = 15; x2 = 65;
draw_border(x1,y1,x2,y2,WHITE,RED);
gotoxy((x1+x2)/2-6,y1+1); textcolor(BLUE);
cprintf("ทะเบียนนักศึกษา");
fseek(fptr,countrecord*sizeof(Datarec),SEEK_SET);
fread(&Datarec,sizeof(struct datatype),1,fptr);
textbackground(WHITE);textcolor(RED);
for(count = 0;count < Num_of_Fields;count++)
{
gotoxy(x1+2,y1+3+count);cprintf("%s",field[count]);
gotoxy(x1+16,y1+3+count);cprintf(": %s",Datarec.Info[count]);
}
getch();
returned_screen(x1,y1,x2,y2,BLUE);
fclose(fptr);
return TRUE;
}
```

2) การรับและการส่งไฟล์ ซึ่งทางเทอร์มินอลจะทำการส่งชื่อไฟล์ที่ต้องการจะได้รับนั้น ไปให้กับเซ็นเตอร์ และเมื่อเซ็นเตอร์สามารถรับชื่อไฟล์ได้แล้ว จะเปิดไฟล์นั้น แล้วส่งเนื้อความของไฟล์นั้นให้กับเทอร์มินอลที่ร้องขอมา จะสามารถแสดงได้ดังต่อไปนี้

<u>แสดงส่วนของการรับไฟล์</u>

```
recieve_file()
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int fhand,byte_rx;
unsigned char fname;
        union {
unsigned int word;
unsigned char byte[2];
}size;
char ch;

    fname = getch();

for(byte_rx=0;byte_rx<2;byte_rx++) {
while(!(inportb(COMM2+5)&0x01));
size.byte[byte_rx] = inportb(COMM2);
outportb(COMM2,ACK);
}
for(byte_rx=0;byte_rx<size.word;byte_rx++) {
while(!(inportb(COMM2+5)&0x01));
ch = inportb(COMM2);
putch(ch); /* แสดงออกทางหน้าจอด้วย */
outportb(COMM2,ACK);
}
}
```

<u>แสดงส่วนของการส่งไฟล์</u>

```
send_file() {
union {
unsigned int word;
unsigned char byte[2];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}size;
FILE *fp;
int fhand,byte_s,byte_rx;
unsigned char fname;
char ch;
clrscr();
printf("\n            WAIT...\n");

      while(!(inportb(COMM2+5)&0x01)); /*ตรวจข้อมูลในบัฟเฟอร์*/
      fname=inportb(COMM2);            /*รับชื่อไฟล์*/
      outportb(COMM2,ACK);
      if(fname==ESC) exit(0);

 if((fhand = open("main.dat",O_RDONLYIO_BINARY))==-1) {
perror("\nSORRY!");
exit(EXIT_FAILURE);
 }
 fp = fdopen(fhand,"rb");     /*เปิดไฟล์*/
 size.word = filelength(fhand);
 for(byte_s=0;byte_s<2;byte_s++) {
outportb(COMM2,size.byte[byte_s]); /*ส่งขนาดไฟล์*/
while(!(inportb(COMM2+5)&0x01));   /*เช็คบัฟเฟอร์*/
while(inportb(COMM2)!=ACK);        /*ตรวจว่าเป็น ACK หรือไม่*/
 }
/* loop นี้เป็นการส่ง ไฟล์ */
for(byte_s=0;byte_s<size.word;byte_s++) {
ch = getc(fp);
outportb(COMM2,ch);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*putch(ch);  */                    /*ใช้แสดงผลออกทางหน้าจอ*/
while(!(inportb(COMM2+5)&0x01));
while(inportb(COMM2)!=ACK);
}
fclose(fp);
}
```

ส่วนของโปรแกรมอื่น ๆ จะเป็นการรับส่งข้อมูลในลักษณะเดียวกัน

คือ เป็นลักษณะของการส่งข้อมูลเป็นสตรีม ซึ่งโปรแกรมที่เพิ่มเติมขึ้นมาจากที่ได้กล่าวไปแล้วนั้น มีทอล์คทูเซ็นเตอร์ (TALK TO CENTER) โปรแกรมพรีดิคชั่น (PREDICTION) เป็นต้น

<u>หมายเหตุ</u> โปรแกรมทั้งหมดจะถูกแสดงไว้ที่ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

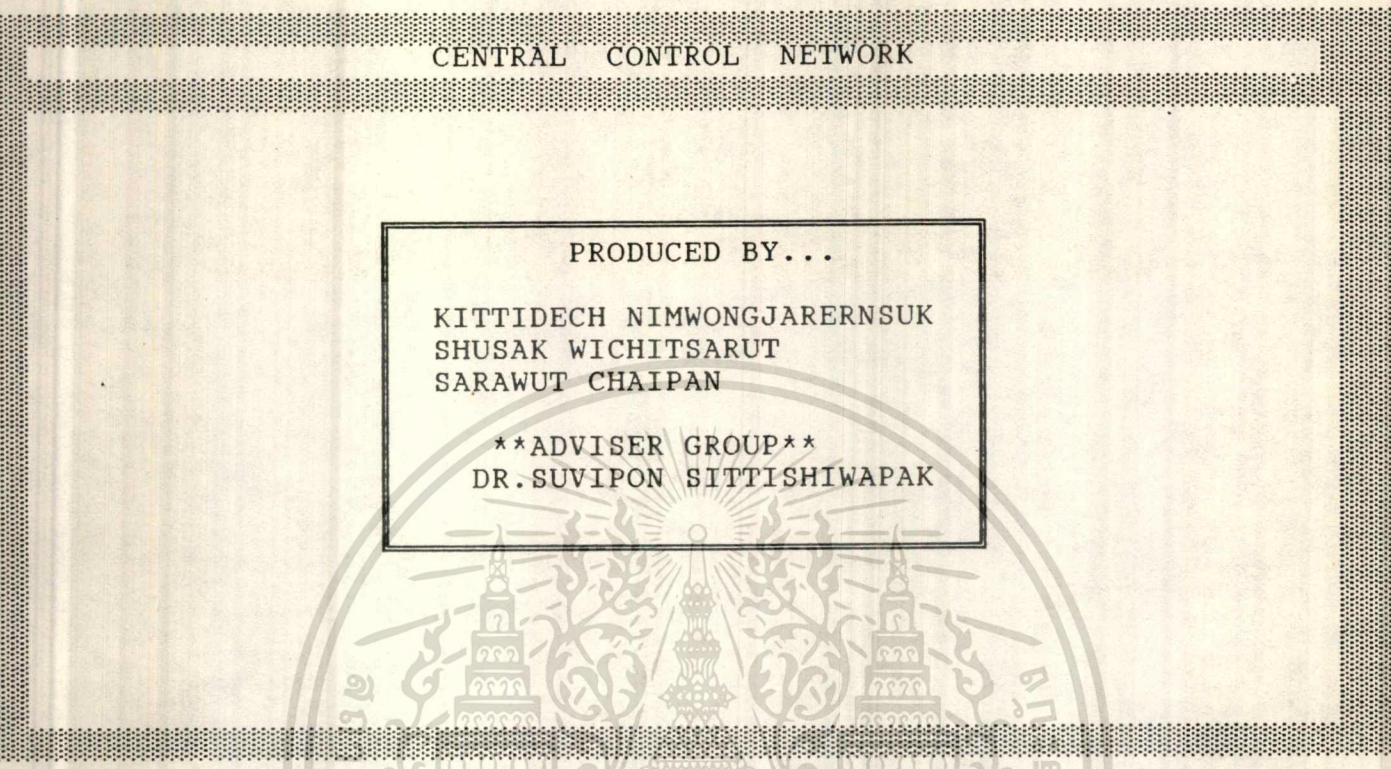
บทที่6
ผลการทดลองและสรุป

สำหรับโครงงานนี้ ได้ทดลองโดยใช้เครื่องคอมพิวเตอร์ 3 เครื่อง โดยใช้เป็นฮับ 1 เครื่องและเทอร์มินัลอีก 2 เครื่อง การเชื่อมต่อระหว่างฮับกับเทอร์มินัลผ่านทางสาย RS-232C ผลการทดลองมีดังนี้

ในส่วนของฮาร์ดแวร์ ได้ทำการตรวจสอบฮาร์ดแวร์โดยใช้ดีบัก ซึ่งประกอบไปด้วยการเซทค่าพารามิเตอร์ในการสื่อสาร และการควบคุมการเชื่อมต่อ จากการทดลองได้ผลว่า สามารถตรวจจับสัญญาณเอ็นไควรี่ได้และทำการส่งข้อมูลติดต่อระหว่างเซนเตอร์กับเทอร์มินัลได้

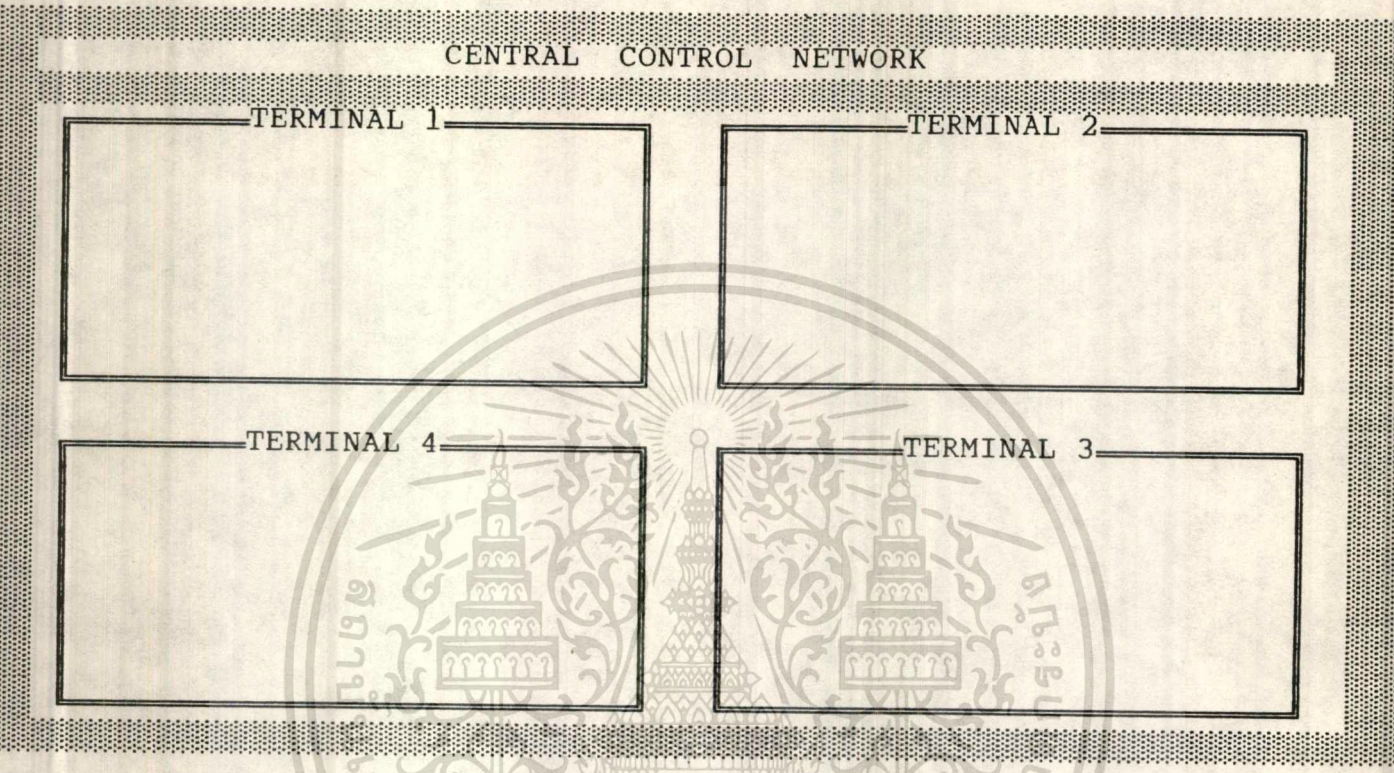
ในส่วนของซอฟแวร์ สามารถที่จะใช้คำสั่งควบคุมการทำงานของฮาร์ดแวร์และสามารถที่จะรับส่งข้อมูลได้ สำหรับแอปปลิเคชั่นนั้นมีวีดีโอเท็กซ์ ทอร์คทูเซนเตอร์ อิเลคทรอนิคส์เมล์ และไฟล์ทรานเฟอร์และในการแสดงผลของการทดลองนั้นสามารถดูได้จากรูป ซึ่งได้แสดงไว้แล้ว ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



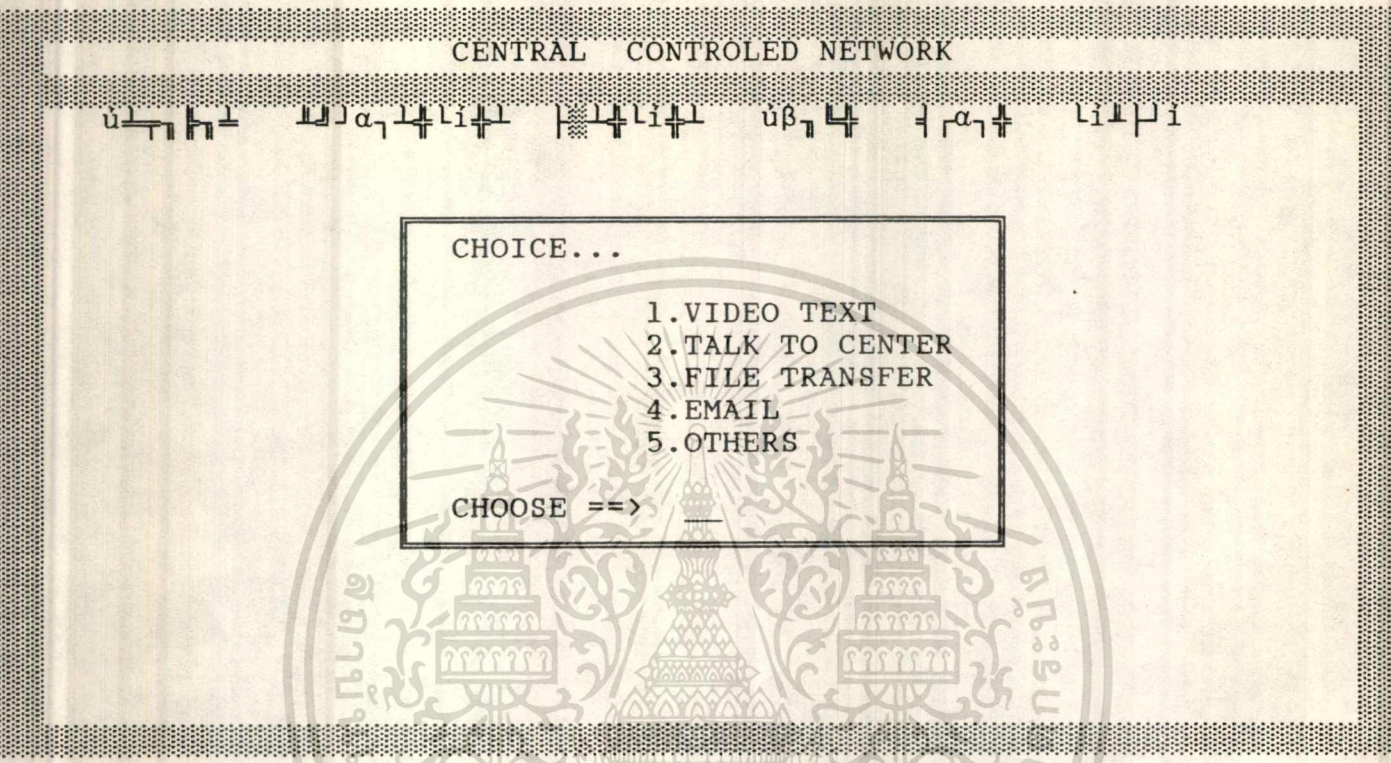
รูปที่ 6.1 รูปแสดงหน้าจอเริ่มต้นการทำงานที่เซนเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



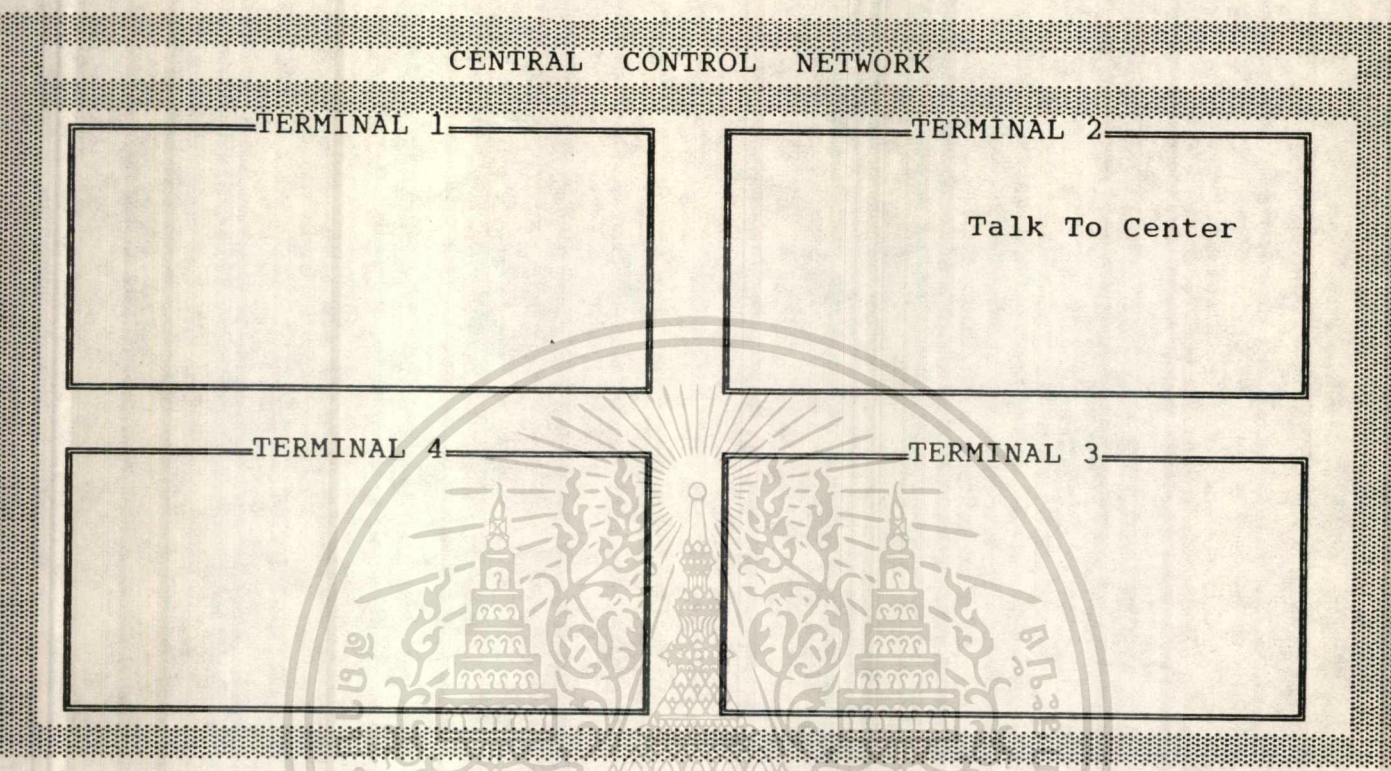
รูปที่ 6.2 รูปแสดงความพร้อมของเซนเตอร์ที่จะรับการร้องขอจากเทอร์มินอลต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.3 รูปแสดงหน้าจอเริ่มต้นการทำงานที่เทอร์มินอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

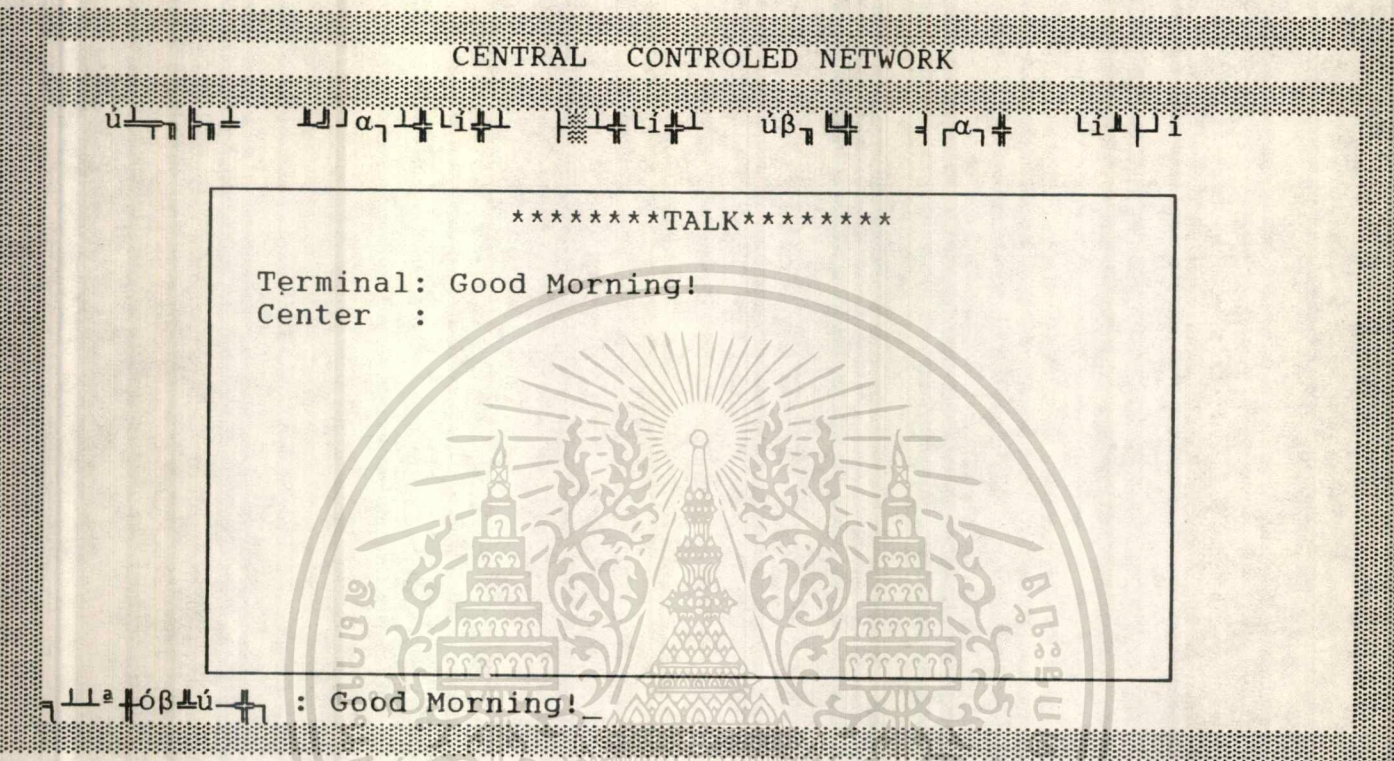


รูปที่ 6.4 แสดงการทำงานที่เซนเตอร์ เมื่อเทอร์มินอลที่ 2 ได้รับการตอบรับแล้ว
โดยทางเทอร์มินอลที่ 2 นั้นต้องการติดต่อกับเซนเตอร์แบบทอร์คทูเซนเตอร์ (Talk To Center)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

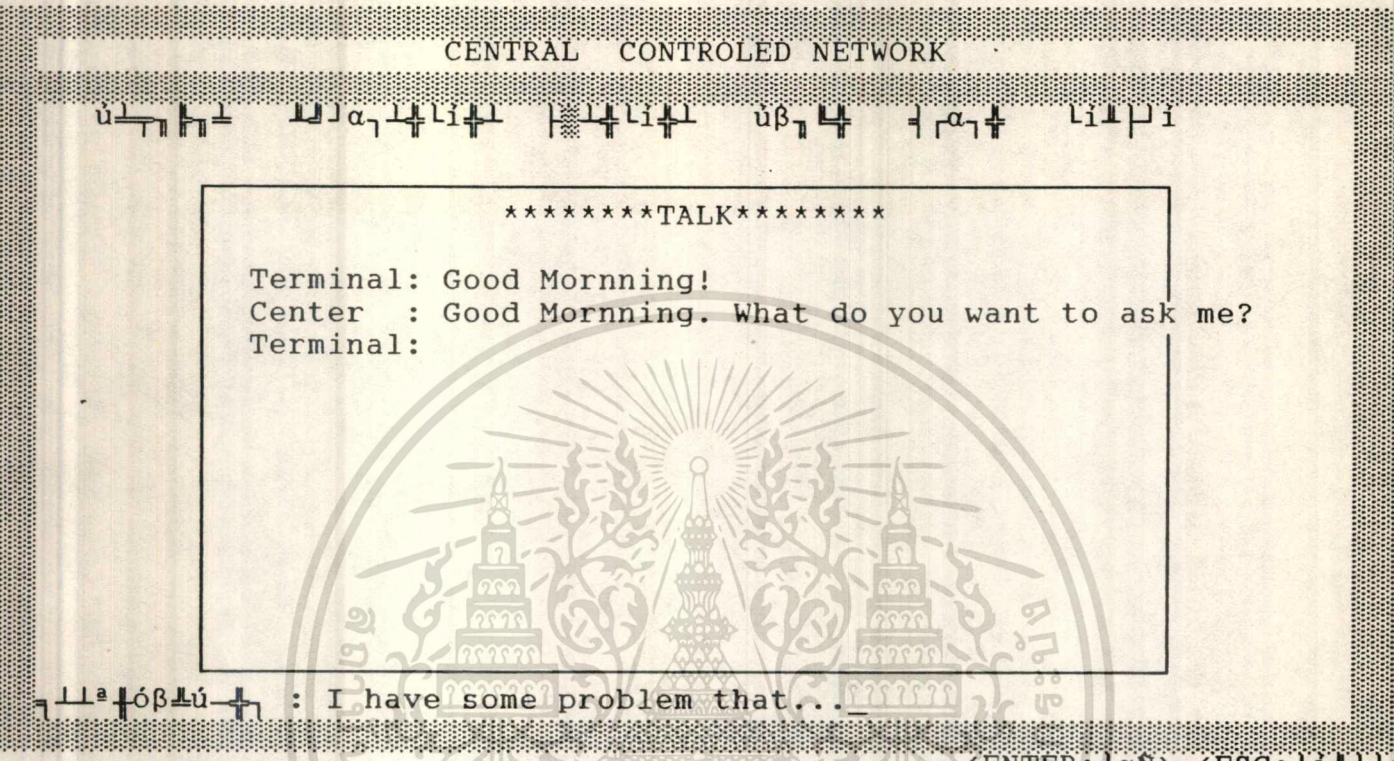
รูปที่ 6.5 แสดงเมนูทอร์คทูเซนเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



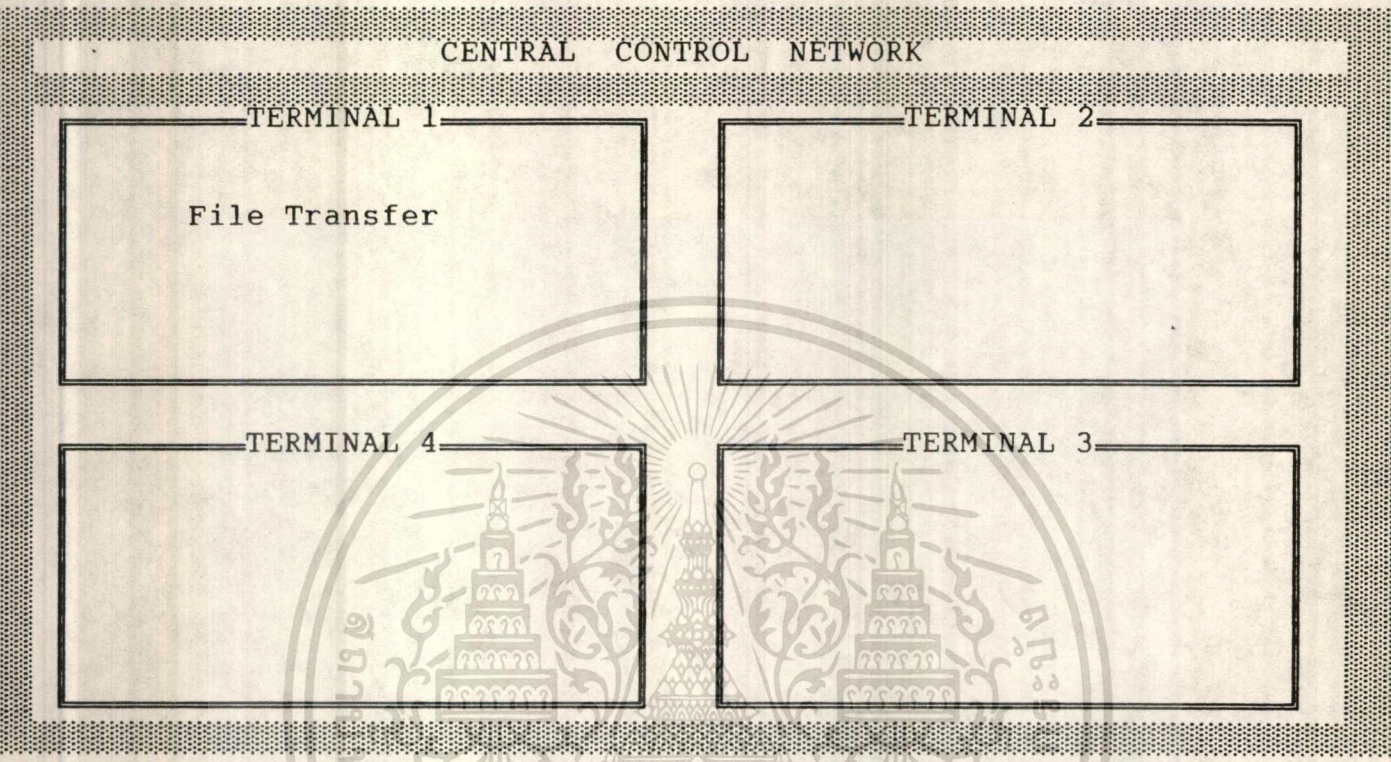
รูปที่ 6.6 แสดงการติดต่อและรอรับข้อมูลจากเซนเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



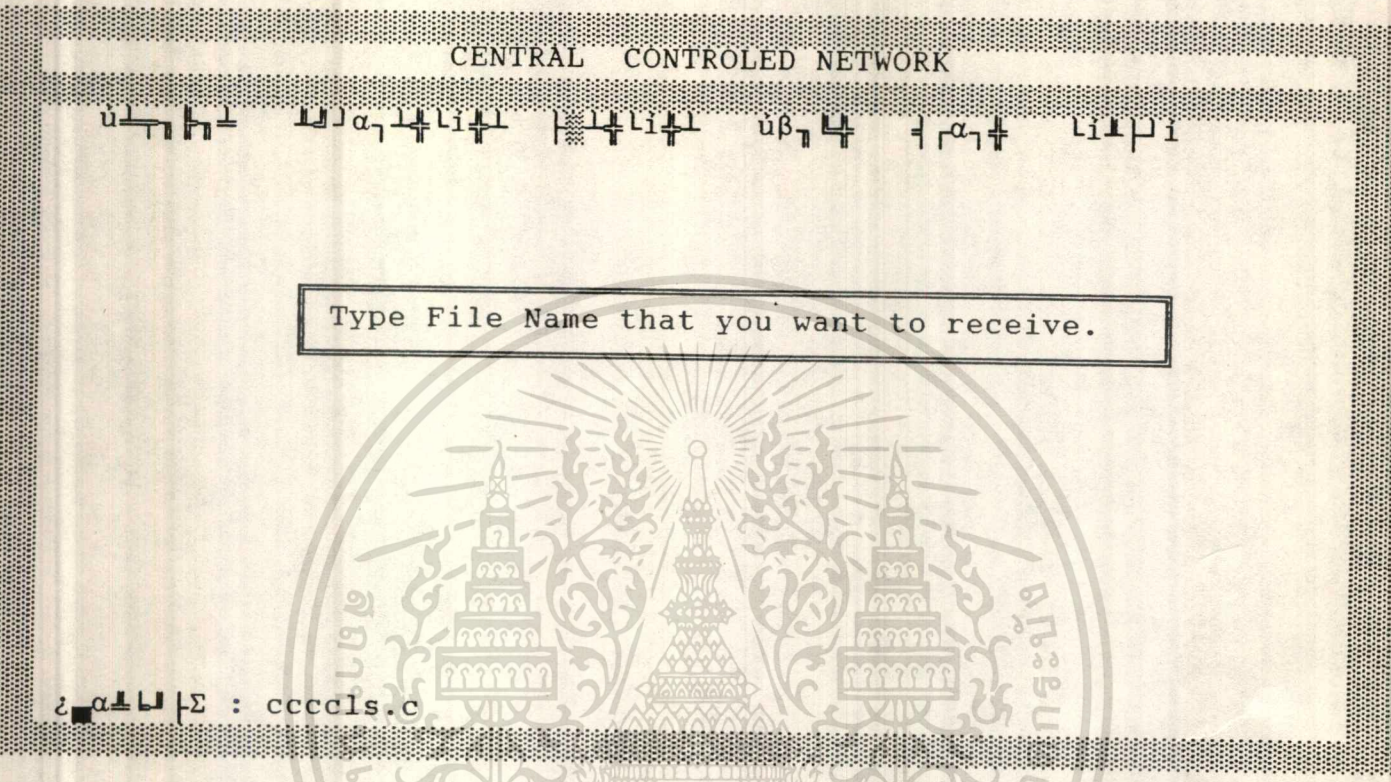
รูปที่ 6.8 แสดงการบริการของเซนเตอร์ที่ให้กับเทอร์มินอล 1 ในการรับส่งไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.7 แสดงข้อมูลที่เซนเตอร์ส่งมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.9 แสดงเมนูที่เทอร์มินอลทำการขอไฟล์จากเซนเตอร์และการพิมพ์ชื่อไฟล์"cccls.c"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

การทำงานของโครงงานนี้สามารถแบ่งได้เป็น 2 ส่วน คือ ส่วนของฮาร์ดแวร์และส่วนของแอปปลิเคชั่นซอร์ฟแวร์ ซึ่งสามารถสรุปได้ดังนี้

-ส่วนของฮาร์ดแวร์

การทำงานของฮาร์ดแวร์ในโครงงานนี้ ได้เปลี่ยนแปลงวงจรบางส่วนในส่วนตรวจจับสัญญาณ ENQ จากเดิมนั้นการตรวจจับจะตรวจจับจากช่องสัญญาณเพียงบิทเดียว ซึ่งถ้าหากมีนอยซ์(Noise) หรือสัญญาณที่ไม่ต้องการเข้ามาในบิทใดๆ จะทำให้ส่วนตรวจจับสามารถตรวจจับได้ทันที ซึ่งทำให้เกิดการผิดผลาดขึ้นได้ ดังนั้นจึงได้ทำการเปลี่ยนแปลงวงจรส่วนนี้ โดยให้มีการตรวจจับสัญญาณทั้ง 8 บิท ซึ่งจะทำให้การตรวจจับมีความแน่นอนกว่า ทำให้การทำงานเป็นไปตามที่วางไว้ทุกประการ

-ส่วนของแอปปลิเคชั่น ซอร์ฟแวร์

แอปปลิเคชั่นซอร์ฟแวร์ที่ใช้ในโครงงานนี้ก็มี วีดีโอเท็กซ์ ทอร์คทูเซนเดอร์ อิเลคทรอนิคส์เมล์ และไฟล์ทรานเฟอร์ โดยลักษณะการทำงานจะเป็นดังนี้ คือ ตัวศูนย์กลางจะทำงานให้กับเทอร์มินอลหนึ่ง ๆ จนเสร็จงานนั้นแล้วจึงรอรับการร้องขอจากเทอร์มินอลต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวทางการพัฒนาต่อ

ในส่วนของการพัฒนาต่อนั้น สามารถแยกพิจารณาได้เป็น 2 ส่วนคือ ส่วนของฮาร์ดแวร์ และส่วนของซอร์ฟแวร์ ในส่วนของฮาร์ดแวร์นั้น การติดต่อระหว่างเทอร์มินัลกับเทอร์มินัลนั้น สามารถทำได้โดยผ่านทางฮับ ซึ่งเราสามารถปรับปรุงส่วนนี้ได้โดยให้มีการเชื่อมต่อระหว่างตัวเทอร์มินัลโดยตรงไม่ต้องผ่านฮับ ทำให้การติดต่อสะดวกและรวดเร็วยิ่งขึ้น ในส่วนของแอปปลิเคชั่นซอร์ฟแวร์ ควรจะเพิ่มส่วนของโปรแกรมฝังตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*Ò¹ÁÑ¡Á¿ Ð¿·Ö µÙà ÐµÊÁä¿Ù·ÊÃ */
#include <dos.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>
#include <io.h>
#include <alloc.h>
#include <fcntl.h>
#include <conio.h>
#include <stdio.h>

#define UP    0x4800
#define DOWN  0x5000
#define LEFT  0x4B00
#define RIGHT 0x4D00
#define ESC   0x011B
#define ENTER 0x1C0D
#define TRUE              1
#define FALSE             0
#define MAXFIELD          20 /*¦Ï·Ä·ÈÝÄ¢áÊ¹ÁÌÄÝ³ÙÇØ¥Ç×²*/
#define FIELDLENGTH       25 /*¦Ï·Ä·³ÝÄÇØ¥Ç×²¢Ê¥Ñ³àÃÌÈÝÄ¢áÊ¹ÁÌÄÝ³Ù*/
#define DATALENGTH        35 /*¦Ï·Ä·³ÝÄÇØ¥Ç×²¢Ê¥¨ÜàÊ-Ç¡×Ã*/
#define MAXDATA          30 /*¦Ï·Ä··Ý¡ÀÛ¡ÆÌÇØ¥Ç×²µÙà¸Ý·µÛ¡Ô²á*/

        unsigned        char Num_of_Fields = 0;
        char    Search_Options[MAXFIELD+1];
        long    numrecord = 0;
        char    field[MAXFIELD][FIELDLENGTH + 1];
        struct  datatype            {
                                            char    Info[MAXFIELD][DATALENGTH+1];
                                            char    Image[13];
                                            char    Chronology[400];
                                        } Datarec,Search_Dat[MAXDATA];
        char main_menu[3][15] = {" £ÏÑ·Ì·Ì "," µÙà¿Ì "," À¡ÐÃÙ¡ "};
        struct Menu {
                int x;
                int y;
                char *name;
        }Main_Menu[6];

#include "c:\language\pred\database\vdo_text.c"
#include "c:\language\pred\database\tran.c"
#include "c:\language\pred\database\email.c"
#include "c:\language\pred\database\talk.c"
/*#include "c:\language\pred\predict\predict.c"*/

menu()
{
   int i;
   textbackground(BLUE);
   textcolor(WHITE);
   gotoxy(1,2);
   cprintf ("±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±");
   gotoxy(1,4);
   cprintf ("±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±");
   gotoxy (1,3);
   cprintf ("±±±±±±                    CENTRAL  CONTROLED NETWORK                     ±±±±±±");
   for(i=5;i<24;i++) {
       gotoxy (1,i);
       cprintf ("±±±±±±                                                                        ±±±±±±");
   }
   gotoxy (1,24);
   cprintf ("±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±±");
   textbackground(WHITE);
```

เอกสารนี้เป็น...ใช้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ...ห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   textcolor(BLACK);
   gotoxy(7,5);
   cprintf ("   £ÏÑ·Ì·Ï   Ð¼Ùà¿ÁÍÀ¡ÍÁ  Ã²ÁÍÀ¡ÍÁ   £á·ÈÍ   µÚà¿Í   À¡ÐÃÙ¡          ")·
   intro();
   mainmenu();
}

intro()
{
   returned_screen(25,8,56,18,BLUE);
   draw_double_border(25,8,56,18,BLUE,LIGHTCYAN);
   textbackground(BLUE);
   textcolor(YELLOW); gotoxy(35,9); cprintf("PRODUCED BY...");
   textcolor(WHITE); gotoxy(28,11); cprintf("·ÍÀ¡Ù³³ÙÐ²¯ ·Ùá¿Ä¥£äÐ¦ÁÙ«Ç×¢");
   textcolor(YELLOW); gotoxy(28,12); cprintf("·ÍÀ¯ØÁÝ¡²Ùä ·Í¿Ç¡×ÃÊÌÔÁ");
   textcolor(CYAN); gotoxy(28,13); cprintf("·ÍÀÁÝ¥ÇÁÁ£ä ·Í¿Ç¡×ÃÊÌÔÁ");
   textcolor(YELLOW); gotoxy(31,15); cprintf("**ADVISER GROUP**");
   textcolor(WHITE); gotoxy(30,16); cprintf("²Á.Ç×ÄÙ¼Ã  ÇÙµ¶Ù¯ÚÄ¾Í£");
   getch();
   returned_screen(26,9,55,16,BLUE);
}

mainmenu()
{
   draw_double_border(25,8,56,18,BLUE,LIGHTCYAN);
   textcolor(YELLOW); gotoxy(28,9); cprintf("CHOICE...");
   textcolor(WHITE); gotoxy(28,11); cprintf("1.VIDEO TEXT");
   textcolor(YELLOW); gotoxy(38,12); cprintf("2.TALK TO CENTER");
   textcolor(CYAN); gotoxy(38,13); cprintf("3.FILE TRANSFER");
   textcolor(YELLOW); gotoxy(38,14); cprintf("4.EMAIL");
   textcolor(WHITE); gotoxy(38,15); cprintf("5.OTHERS");
   textcolor(YELLOW); gotoxy(28,17); cprintf("CHOOSE ==> ");
   textcolor(YELLOW+BLINK); gotoxy(40,17); cprintf("_");
}

shift()
{
        int key,i;
        int x[6] = {9,51,59},y=5;
        textbackground(LIGHTCYAN); textcolor(BLACK);
     gotoxy(9,5);       cprintf(" £ÏÑ·Ì·Ï ");
        i=0;
        do {
                key = bioskey(0);
                if((key==0x231)||(key==0x2C3)||(key==0x4F31))
                {
                        search();
                        mainmenu();
                }
                if((key==0x332)||(key==0x32F)||(key==0x5032))
                {
                        talk_to_center();
                        mainmenu();
                }
                if((key==0x433)||(key==0x42D)||(key==0x5133))
                {
                        transfer_file();
                        mainmenu();
                }
                if((key==0x534)||(key==0x5BE)||(key==0x4B34))
                {
                        email();
                        mainmenu();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
                if((key==0x635)||(key==0x6B4)||(key==0x4C35))
                {
                        /*prediction();*/
                        mainmenu();
                }
                if(key==RIGHT) {
                        textbackground(WHITE); textcolor(BLACK);
                        gotoxy(x[i],y); cprintf("%s",main_menu[i]);
                        if(i==2) i=0;
                        else     i++;
                        textbackground(LIGHTCYAN);      textcolor(BLACK);
                        gotoxy(x[i],y); cprintf("%s",main_menu[i]);
                } if(key==LEFT) {
                        textbackground(WHITE);  textcolor(BLACK);
                        gotoxy(x[i],y); cprintf("%s",main_menu[i]);
                        if(i==0) i=2;
                        else     i--;
                        textbackground(LIGHTCYAN);      textcolor(BLACK);
                        gotoxy(x[i],y); cprintf("%s",main_menu[i]);
                } if(key==ESC)  i = ESC;
        } while((key!=ESC)&&(key!=ENTER));
     textbackground(WHITE);     textcolor(BLACK);
        gotoxy(x[i],y); cprintf("%s",main_menu[i]);
        return i;
}

draw_border(x1,y1,x2,y2,gnd,foregnd)
        int x1,y1,x2,y2,gnd,foregnd;
{
        register int i,j;
        textbackground(gnd);    textcolor(foregnd);
        for(j=y1;j<y2;j++) {
                gotoxy(x1,j);
                for(i=x1;i<x2;i++)      cprintf(" ");
        }
        for(i=x1+1;i<x2;i++) {
                gotoxy(i,y1);   putch(196);
                gotoxy(i,y2);   putch(196);
        }
        for(i=y1+1;i<y2;i++) {
                gotoxy(x1,i);   putch(179);
                gotoxy(x2,i);   putch(179);
        }
        gotoxy(x1,y1); putch(218);
        gotoxy(x1,y2); putch(192);
        gotoxy(x2,y1); putch(191);
        gotoxy(x2,y2); putch(217);
}

draw_double_border(int xul,int yul,int xlr,int ylr,int gnd,int foregnd) {
        register int i=0;
     textbackground(gnd);       textcolor(foregnd);
        gotoxy(xul,yul);
        for(i=xul;i<=xlr;i++)   putch(205);
        gotoxy(xul,ylr);
        for(i=xul;i<=xlr;i++)   putch(205);
        for(i=yul;i<=ylr;i++)
        {
                gotoxy(xul,i);  putch(186);
                gotoxy(xlr,i);  putch(186);
        }
        gotoxy(xul,yul);putch(201);
        gotoxy(xlr,yul);putch(187);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        gotoxy(xlr,ylr);putch(188);
        gotoxy(xul,ylr);putch(200);
        return(0);
}

returned_screen(x1,y1,x2,y2,gnd)
        int x1,y1,x2,y2,gnd;
{
        register int i,j;
        textbackground(gnd);    textcolor(gnd);
        for(j=y1;j<=y2;j++) {
                gotoxy(x1,j);
                for(i=x1;i<=x2;i++)     cprintf(" ");
        }
}
warnning(x,text)
        int x;
        char *text;
{
        draw_border(20,10,60,13,WHITE,RED+BLINK);
        gotoxy(x,11);
        textcolor(RED);
        cprintf("%s",text);
        getch();
        returned_screen(20,10,60,13,BLUE);
}

help() {
        warnning(35,"£ÏÑ·Ì·Ï");
}

add() {
        int i;
        draw_border(15,10,65,14,WHITE,RED);
        textcolor(BLACK);
        gotoxy(24,11);
        cprintf("¡ÍÁÐ¼Ùà¿ÁÍÀ¡ÍÁ·Ýá·ÐÃÜÊ¡¡ÁÌµÏÔ²á 2 ÃÝ¡Æ±Ì");
        gotoxy(40,13);
        cprintf("ÈÁÚÊ");
        i = decide(20,13,"Ð¼Ùà¿¦Ï·Ã··Ý¡ÁÛ¡ÆÍ",47,13,"Ð¼Ùà¿ÈÝÄ¢áÊ¹ÁÍÄÝ³Ù",WHITE,BLACK,RED+BLINK);
        returned_screen(15,10,65,14,BLUE);
        if(i==1)  createfield();
        if(i==0)  createdata();
}

remove_() {
        int i;
        draw_border(15,10,65,14,WHITE,RED);
        textcolor(BLACK);
        gotoxy(24,11);
        cprintf("¡ÍÃÃ²ÁÍÀ¡ÍÁ·Ýá·ÐÃÜÊ¡¡ÁÌµÏÔ²á 2 ÃÝ¡Æ±Ì");
        gotoxy(40,13);
        cprintf("ÈÁÚÊ");
        i = decide(20,13,"Ã²¦Ï·Ã··Ý¡ÁÛ¡ÆÍ",47,13,"Ã²ÈÝÄ¢áÊ¹ÁÍÄÝ³Ù",WHITE,BLACK,RED+BLINK);
        returned_screen(15,10,65,14,BLUE);
        if(i==0)  removedata();
        if(i==1)  removefield();
}

search() {
        draw_border(15,6,65,23,LIGHTRED,WHITE);
        textcolor(YELLOW);
        gotoxy(24,7);
        cprintf("¨ÙàÊ··Í¿Ç¡×Ã          ÐÃ¢¹ÁÌ¦Ï³ÝÄ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(search_data(LIGHTRED,WHITE)==TRUE)   run_search();
        returned_screen(15,6,65,23,BLUE);
}

from()
{
        int i;
        returned_screen(25,8,56,18,BLUE);
        draw_border(15,20,65,23,WHITE,BLUE);
        i = decide(20,22,"ÑÈÃà¥¢áÊ¿ØÃ",48,22,"Produced By",WHITE,BLUE,BLUE+BLINK);
        if(i==2) return FALSE;
        draw_double_border(25,8,56,17,BLUE,LIGHTCYAN);
     textbackground(BLUE);
        if(i==0)
        {
                textcolor(YELLOW); gotoxy(35,9); cprintf("ÑÈÃà¥¢áÊ¿ØÃ...");
                gotoxy(28,11); cprintf("1. Ñ²·¡µÌÐ¸ÚÀ·£±ÌÄÙÀÄ¡ÁÁ¿");
                gotoxy(28,12); cprintf("2. »àÎÀ¡Ù¦¡ÁÁ¿·Ý¡ÁÛ¡ÆÎ");
                gotoxy(28,13); cprintf("3. ÇÏ·Ý¡¥Î·..");
                textcolor(WHITE+BLINK); gotoxy(28,16); cprintf("¡²£ÚÀäÓ² Õ Ô¹ÀÝ¥Ð¿·Ø");
                getch();
                returned_screen(25,8,56,18,BLUE);
        }
        if(i==1) intro();
        returned_screen(15,20,65,23,BLUE);
}


turn_off() {
        register int i;
        draw_border(25,10,55,14,WHITE,RED);
        textcolor(BLACK);
        gotoxy(35,11);
        cprintf("DÃÙ¡¦ÎÁµÏ¥Î·");
        i = decide(35,13,"Ó¨à",42,13,"Ô¿àÓ¨à",WHITE,BLACK,BLACK+BLINK);
        returned_screen(25,10,55,14,BLUE);
        return i;
}


decide(x_yes,y_yes,text1,x_no,y_no,text2,gnd,foregnd,blink)
        int x_yes,y_yes,x_no,y_no,gnd,foregnd,blink;
        char *text1,*text2;
{
        register int key,i=0;
        textbackground(gnd);
        textcolor(blink);
        gotoxy(x_yes,y_yes);
        cprintf("%s",text1);
        textcolor(foregnd);
        gotoxy(x_no,y_no);
        cprintf("%s",text2);
        do {
                key = bioskey(0);
                if((key==RIGHT)||(key==LEFT)||(key==UP)||(key==DOWN)) {
                        if(i==0) { textcolor(foregnd); gotoxy(x_yes,y_yes);
                                        cprintf("%s",text1);
                                        textcolor(blink); gotoxy(x_no,y_no);
                                        cprintf("%s",text2);  i=1;
                        } else {
                        if(i==1) { textcolor(blink); gotoxy(x_yes,y_yes);
                                        cprintf("%s",text1);
                                        textcolor(foregnd); gotoxy(x_no,y_no);
                                        cprintf("%s",text2);  i=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        }
                    }
                }
                if(key==ESC) i=2;
        } while((key!=ESC)&&(key!=ENTER));
        return i;
}

main()
{
        int choice;
        clrscr();
        menu();
        if(readfield()) checkrecord();
        for(;;) {
                choice = shift();
                if(choice==ESC)  { clrscr(); exit(0); }
                switch(choice) {
                        case 0: help(); mainmenu(); break;
                        case 1: from(); mainmenu(); break;
                        case 2: choice = turn_off();
                                if(choice==0)   { clrscr();     exit(0);        }
                                else            { mainmenu();   break;  }
                        default: break;
                }
        }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*== programme send 5 byte by stop and wait method.==*/
/*== if it work,use send2.c and rec2.c to improve programme ==*/


#include <dos.h>
#include <stat.h>
#include <fcntl.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <io.h>
#include "chat.c"


#define COMM2  0x2F8
#define ACK      6
#define ENQ      5
#define ESC     27
#define SYN     22


set_com_var()
{
        outportb(COMM2+3,80);
        outportb(COMM2,60);
        outportb(COMM2+1,00);
        outportb(COMM2+3,03);
}

test_s_pack()
{
    char buffer[5]={SYN,'A','B','C','D'};
    int byte_rx;
      outportb(COMM2,SYN);
      while(inportb(COMM2)!=ACK);
    printf("test send.\n");
    for(byte_rx=1;byte_rx<5;byte_rx++) {
        outportb(COMM2,buffer[byte_rx]);
        putch(buffer[byte_rx]);     /*no need*/
        while(!(inportb(COMM2+5)&0x01));
        while(inportb(COMM2)!=ACK);
    }
    printf("\nSEND PACKET ALREADY.");
}

rec_menu()
{
        int fhand,byte_rx;
        unsigned char fname;
        union {
        unsigned int word;
        unsigned char byte[2];
        }size;
        char ch;

   fname = getch();

        outportb(COMM2,fname);
        while(!(inportb(COMM2+5)&0x01));
        while(inportb(COMM2)!=ACK);
        if(fname == ESC) exit(0);

        for(byte_rx=0;byte_rx<2;byte_rx++) {
                while(!(inportb(COMM2+5)&0x01));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                size.byte[byte_rx] = inportb(COMM2);
                outportb(COMM2,ACK);
        }
        clrscr();
        for(byte_rx=0;byte_rx<size.word;byte_rx++) {
                while(!(inportb(COMM2+5)&0x01));
                ch = inportb(COMM2);
                putch(ch);
                outportb(COMM2,ACK);
        }

}
acting()
{
        unsigned char fname;

        fname = getch();
        outportb(COMM2,fname);
        while(!(inportb(COMM2+5)&0x01));
        while(inportb(COMM2)!=ACK);
        if(fname == ESC) exit(0);
        switch(fname)
        {
            case '1': chat(); break;
            case '2':  break;
        }
}
main()
{
        clrscr();
        set_com_var();
        test_s_pack();
        clrscr();
        printf("\n\n=====Press any key to connect or ESC key to halt.=====\n");
        rec_menu();
        printf("\n\n\n\n\nChoose choice number.---> ");
        rec_menu();
        printf("\n\n\n\n\nChoose choice number.---> ");
        acting();
        getch();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*== this programme made for receive 5 bytes from programme send.c==*/

#include <dos.h>
#include <stat.h>
#include <fcntl.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <io.h>
#include "chat.c"

#define COMM2  0x2F8
#define ACK      6
#define ENQ      5
#define ESC     27
#define SYN     22

set_com_var()
{
        outportb(COMM2+3,0x80);
        outportb(COMM2,0x60);
        outportb(COMM2+1,0x00);
        outportb(COMM2+3,0x03);
}

test_r_pack()
{
    int byte_rx;
    char buffer[5];

        outportb(COMM2,SYN);
        while(inportb(COMM2)!=SYN);
        outportb(COMM2,ACK);
        printf("test receive.\n");

        for(byte_rx=1;byte_rx<5;byte_rx++) {
            while(!(inportb(COMM2+5)&0x01));
            buffer[byte_rx]=inportb(COMM2);
            putch(buffer[byte_rx]);     /*no need*/
            outportb(COMM2,ACK);
        }
        printf("\nRECEIVED PACKET ALREADY.");
}

send_menu() {
       union {
              unsigned int word;
              unsigned char byte[2];
       }size;
       FILE *fp;
       int fhand,byte_s,byte_rx;
       unsigned char fname;
       char ch;
       clrscr();
       printf("\n          WAIT...\n");

            while(!(inportb(COMM2+5)&0x01));
            fname=inportb(COMM2);
            outportb(COMM2,ACK);
            if(fname==ESC) exit(0);

        if((fhand = open("main.dat",O_RDONLY|O_BINARY))==-1) {
             perror("\nSORRY!");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                exit(EXIT_FAILURE);
        }
        fp = fdopen(fhand,"rb");
        size.word = filelength(fhand);
        for(byte_s=0;byte_s<2;byte_s++) {
                outportb(COMM2,size.byte[byte_s]);
                while(!(inportb(COMM2+5)&0x01));
                while(inportb(COMM2)!=ACK);
        }
        clrscr();
        printf("SEND MENU");
        for(byte_s=0;byte_s<size.word;byte_s++) {
                ch = getc(fp);
                outportb(COMM2,ch);
        /*      putch(ch);  */
                while(!(inportb(COMM2+5)&0x01));
                while(inportb(COMM2)!=ACK);
        }
        fclose(fp);
}

send_sub() {
        union {
                unsigned int word;
                unsigned char byte[2];
        }size;
        FILE *fp;
        int fhand,byte_s,byte_rx;
        unsigned char fname,choice;
        unsigned char name[5][14] = {"submain1.dat","submain2.dat",
                        "submain3.dat","submain4.dat","submain5.dat"};
        char ch;
        clrscr();
        printf("\n          WAIT...\n");

              while(!(inportb(COMM2+5)&0x01));
              fname=inportb(COMM2);
              outportb(COMM2,ACK);
              if(fname==ESC) exit(0);
        switch(fname) {
                case '1': choice = 0; break;
                case '2': choice = 1; break;
        }
        if((fhand = open(name[choice],O_RDONLY|O_BINARY))==-1) {
                perror("\nSORRY!");
                exit(EXIT_FAILURE);
        }
        fp = fdopen(fhand,"rb");
        size.word = filelength(fhand);
        for(byte_s=0;byte_s<2;byte_s++) {
                outportb(COMM2,size.byte[byte_s]);
                while(!(inportb(COMM2+5)&0x01));
                while(inportb(COMM2)!=ACK);
        }
        clrscr();
        printf("SEND SUB MENU");
        for(byte_s=0;byte_s<size.word;byte_s++) {
                ch = getc(fp);
                outportb(COMM2,ch);
        /*      putch(ch);*/
                while(!(inportb(COMM2+5)&0x01));
                while(inportb(COMM2)!=ACK);
        }
        fclose(fp);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

acting() {
              unsigned char fname;
              while(!(inportb(COMM2+5)&0x01));
              fname=inportb(COMM2);
              outportb(COMM2,ACK);
              if(fname==ESC) exit(0);
         switch(fname) {
                case '1': chat(); break;
                case '2': break;
        }
}

main()
{
        clrscr();
        set_com_var();
        test_r_pack();
        send_menu();
        send_sub();
        acting();
        getch();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
readfield()
{
        FILE *fp;

        if((fp = fopen("field.nam","r")) == NULL)
        {
                warnning(25,"Ó¿àÇÍ¿ÎÁ´Ð'Ù²Ó½ÃäÑÇ²¥ÁÎÀÃÌÐÉÚÀ²Ó²á");
                return FALSE;
        }
        while(!feof(fp))
                fscanf(fp,"%s\n",field[Num_of_Fields++]);
        fclose(fp);
        return TRUE;
}

void checkrecord()
{
  int handle;

  if((handle = _open("person.dat",O_BINARY | O_RDONLY)) != -1)
  {
        numrecord = filelength(handle)/sizeof(Datarec);
    _close(handle);
  }
}

createfile()
{
        FILE    *fp;
      if ((fp = fopen("person.dat","a+b")) ==  NULL)
                if ((fp = fopen("person.dat","wb")) ==  NULL)
                {
                        warnning(30,"Ó¿àÇÍ¿ÎÁ´Ð'Ù²Ó½ÃäÁÎÀ¨ÙàÊ·Ý¡ÁÛ¡ÆÎÓ²á");
                        return FALSE;
                }
        fwrite(&Datarec,sizeof(struct datatype),1,fp);
        numrecord++;
        fclose(fp);
}

createdata()  /*Ð¼ÙÀà¿ÁÎÀ¨ÙàÊ·Ý¡ÁÛ¡ÆÎ */
{
        register int x1,y1,x2,y2,i,row,key;
        if(Num_of_Fields != 0) {
                if(Num_of_Fields%2 == 0)        y1 = 15-(Num_of_Fields+6)/2;
                else y1 = (15-(Num_of_Fields+6)/2)-1;
                y2 = 15+(Num_of_Fields+6)/2;
                x1 = 15; x2 = 65;
                draw_border(x1,y1,x2,y2,WHITE,RED);
                for(i=0;i<Num_of_Fields;i++)
                {
                        gotoxy(x1+2,y1+3+i);
                        cprintf("%s : ",field[i]);
                }
                gotoxy(x1+2,y2-3);      cprintf("Ô¼Ãä¾Î¼ :");
                gotoxy((x1+x2)/2-3,y1+1);
                textcolor(BLUE);
                cprintf("µÎÐ.ÚÀ·¹ÁÎÄá³Ù");
                textbackground(BLACK);
                textcolor(YELLOW);
                for(row=Num_of_Fields;row>=0;row--)
                {
                        textbackground(BLACK);
                        textcolor(YELLOW);
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        gotoxy(x1+20,y2-3-row);
                        for(i=0;x1+20+i!=x2-3;i++)
                                cprintf(" ");
                        gotoxy(x1+20,y2-3-row);
                        if(row != 0)    gets(Datarec.Info[Num_of_Fields-row]);
                        else            gets(Datarec.Image);
                        textcolor(RED);
                        textbackground(WHITE);
                        gotoxy(x1+20,y2-3-row);
                        for(i=0;x1+20+i!=x2-3;i++)
                                cprintf(" ");
                        gotoxy(x1+20,y2-3-row);
                        if(row != 0)    cprintf("%s",Datarec.Info[Num_of_Fields-row]);
                        else            cprintf("%s",Datarec.Image);
                }
                gotoxy(x1+4,y2-2);      textcolor(BLACK);
                cprintf("¢áÊ¿ØÃ¢áÌ¥³á··Ýá·´Ø¡³áÊ¥µÝá¥È¿³Ó¨àÈÁÜÊÔ¿à ?");
                gotoxy(x2-2,y2-2);      textcolor(BLACK+BLINK);
                cprintf("Y");
                for(;;)
                {
                        key = bioskey(0);
                        if(key==ESC)
                        {       returned_screen(x1,y1,x2,y2,BLUE); return FALSE;
                        }
                        if((key==ENTER)||(key==0x15DD)||(key==0x1559)) break;
                        else if(key==0x316E||key==0x314E) { createdata(); break; }
                        else putch(0x07);
                }
                returned_screen(x1,y1,x2,y2,BLUE);
                createfile();
        }
        else
        {
                warnning(30,"Ó¿ã¿ÚÈÝÄ¢áÊ¹ÁÌÃÝ³ÙÓÊá¸Ý·µÛ¡");
                return FALSE;
        }
}

add_data(char oldfield)
{
        FILE    *fptr;
        register int x1,y1,x2,y2,i;
        register char countrecord,count,temp;

        if((fptr = fopen("person.dat","r+b")) == NULL)
        {
                warnning(25,"Ó¿àÇ¦¿ÌÁ´Ð¹Ù²Ô½Ãä PERSON.DAT Ô²á");
                return FALSE;
        }
        if(Num_of_Fields%2 == 0)        y1 = 15-(Num_of_Fields+6)/2;
        else y1 = (15-(Num_of_Fields+6)/2)-1;
        y2 = 15+(Num_of_Fields+6)/2;
        x1 = 15; x2 = 65;
        draw_border(x1,y1,x2,y2,WHITE,RED);
        gotoxy((x1+x2)/2-6,y1+1); textcolor(BLUE);
        cprintf("µÌÐ¸ÚÀ·¹ÁÌÃá³ÙÐ¼Ùà¿Ð³Ù¿");
        for(countrecord = 0;countrecord < numrecord;countrecord++)
        {
                fread(&Datarec,sizeof(struct datatype),1,fptr);
                temp = oldfield;
                textbackground(WHITE);  textcolor(RED);
                for(count = 0;count < Num_of_Fields-1;count++)
                {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                gotoxy(x1+2,y1+3+count);        cprintf("%s",field[count]);
                gotoxy(x1+16,y1+3+count);cprintf(": %s",Datarec.Info[count]);
            }
            gotoxy(x1+2,y1+3+count+1); cprintf("%s",field[temp]);
            textbackground(BLACK); textcolor(YELLOW);
            gotoxy(x1+16,y1+3+count+1);
            for(i=0;x1+16+i!=x2-3;i++)      cprintf(" ");
            gotoxy(x1+16,y1+3+count+1); cprintf(":");
            gotoxy(x1+17,y1+3+count+1);     gets(Datarec.Info[temp++]);
            /* Seek position of fptr from start of file */
            fseek(fptr,countrecord*sizeof(Datarec),SEEK_SET);
            fwrite(&Datarec,sizeof(struct datatype),1,fptr);
            returned_screen(x1+1,y1+2,x2-1,y2-1,WHITE);
        }
        returned_screen(x1,y1,x2,y2,BLUE);
        fclose(fptr);
        return TRUE;
}

createfield()
{
        FILE *fp;
        register int x1,y1,x2,y2,i;
        register char old_field;
        if((fp = fopen("field.nam","a")) == NULL)
        {
            warnning(25,"Ó¿àÇ¦¿¦Á´Ð¹Ù²Ó½ÃäÑÇ²¥Á¦ÀÃ¦ÐÉÚÀ²Ó²á");
            return FALSE;
        }
        old_field = Num_of_Fields; /* Ó¨áó·Ó½Ãä add_data */
        if(Num_of_Fields <= MAXFIELD) {
            if(Num_of_Fields%2 == 0)        y1 = 15-(Num_of_Fields+6)/2;
            else y1 = (15-(Num_of_Fields+6)/2)-1;
            y2 = 15+(Num_of_Fields+6)/2;
            x1 = 15; x2 = 65;
            draw_border(x1,y1,x2,y2,WHITE,RED);
            draw_border(x1+1,y2-3,x2-1,y2-1,WHITE,RED);
            for(i=0;i<Num_of_Fields;i++)
            {
                gotoxy(x1+2,y1+3+i);
                cprintf("%s :",field[i]);
            }
            gotoxy(x1+3,y2-2);
            cprintf("¨ÙàÊÈÝÄ¢áÊ :");
            gotoxy((x1+x2)/2-3,y1+1);
            textcolor(BLUE);
            cprintf("Ð¼Ùà¿ÈÝÄ¢áÊ");
            textbackground(BLACK);
            textcolor(YELLOW);
        gotoxy(x1+16,y2-2);
            for(i=0;x1+16+i!=x2-3;i++)
                cprintf(" ");
        gotoxy(x1+16,y2-2);
            gets(field[Num_of_Fields]);
            returned_screen(x1,y1,x2,y2,BLUE);
            if(strcmp(field[Num_of_Fields],"") == 0) return FALSE;
            else
            {
                y2 = y2-2;
                draw_border(x1,y1,x2,y2,WHITE,RED);
                for(i=0;i<=Num_of_Fields;i++)
                {
                    gotoxy(x1+2,y1+3+i);
                    cprintf("%s :",field[i]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        }
                        gotoxy((x1+x2)/2-7,y1+1);
                        textcolor(BLUE);
                        cprintf("<< ÑÇ²¥ÈÝÄ¢áÊµÝá¥È¿² >>");
                        i = decide(55,13,"Dịß.¢áÊ¿ØÃ",56,15,"Àị ĐÃÙị",WHITE,BLACK,RED+BLINK);
                        if(i==0)
                        {
                                fprintf(fp,"%s\n",field[Num_of_Fields++]);
                                returned_screen(x1,y1,x2,y2,BLUE);
                                fclose(fp);
                                add_data(old_field);
                                return TRUE;
                        }
                        else { returned_screen(x1,y1,x2,y2,BLUE); return FALSE; }
                }
        } else {
                draw_border(15,11,65,14,YELLOW,RED);
                gotoxy(16,12);
                cprintf("ÈÝÄ¢áÊµÚà¿ÚÔ²á³áÊ¥Ô¿èDịÙ· 10 ÈÝÄ¢áÊ");
                getch();
                returned_screen(15,11,65,14,BLUE);
        }
}

run_remove()
{
        register int x1=15,x2=65,y1=6,y2=23,i,key,y;
        register char count = 0,countrecord = 0,delrecord;
        FILE *fptr,*fp;
        y = y1+3;
        do {
                key = bioskey(0);
                if(key==UP) {
                        if(countrecord==0) putch(0x07);
                        else {
                                textbackground(CYAN); textcolor(BLUE);
                                for(i=x1+1;i!=x2-1;i++) {  gotoxy(i,y); cprintf(" "); }
                                for(count=0;count<2;count++)
                                {
                                        gotoxy(x1+5+(30*count),y);
                                        cprintf("%s",Search_Dat[countrecord].Info[count]);
                                }
                                if(y==y1+3)
                                { movetext(x1+1,y1+3,x2-1,y2-2,x1+1,y1+4); y++;
                                }
                                y--;
                                countrecord--;
                                textbackground(BLACK); textcolor(YELLOW);
                                for(i=x1+1;i!=x2-1;i++) {  gotoxy(i,y); cprintf(" "); }
                                for(count=0;count<2;count++)
                                {
                                        gotoxy(x1+5+(30*count),y);
                                        cprintf("%s",Search_Dat[countrecord].Info[count]);
                                }
                        }
                }
                if(key==DOWN) {
                        if(countrecord==numrecord-1) putch(0x07);
                        else {
                                textbackground(CYAN); textcolor(BLUE);
                                for(i=x1+1;i!=x2-1;i++) {  gotoxy(i,y); cprintf(" "); }
                                for(count=0;count<2;count++)
                                {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        cprintf("%s",Search_Dat[countrecord].Info[count]);
                                }
                                if(y==y2-1)
                                { movetext(x1+1,y1+4,x2-1,y2-1,x1+1,y1+3); y--;
                                }
                                y++;
                                countrecord++;
                                textbackground(BLACK); textcolor(YELLOW);
                                for(i=x1+1;i!=x2-1;i++) {  gotoxy(i,y); cprintf(" "); }
                                for(count=0;count<2;count++)
                                {
                                        gotoxy(x1+5+(30*count),y);
                                        cprintf("%s",Search_Dat[countrecord].Info[count]);
                                }
                        }
                }
                if(key==ESC) return FALSE;
        }while((key!=ENTER)&&(key!=ESC));
        delrecord = countrecord;
      if((fp = fopen("temp.dat","w")) != NULL)
        {
                if((fptr = fopen("person.dat","r")) != NULL)
                {
                        for(countrecord=0;countrecord<numrecord;countrecord++)
                        {
                                fseek(fptr,countrecord*sizeof(Datarec),SEEK_SET);
                                fread(&Datarec,sizeof(struct datatype),1,fptr);
                                if(countrecord!=delrecord)
                                        fwrite(&Datarec,sizeof(struct datatype),1,fp);
                        }
                fclose(fptr);
                }
        fclose(fp);
        }
        if(remove("person.dat") != 0)
        { warnning(30,"Ã¸Ô½Ãä person.dat Ô¿àÔ²á"); return FALSE;
        }
        if(rename("temp.dat","person.dat") != 0)
        {  warnning(30,"Ã¸Ô½Ãä temp.dat Ô¿àÔ²á"); return FALSE;
        }
        numrecord--;
        return TRUE;
}

removedata()
{
        int x1=15,y1=6,x2=65,y2=23;
        draw_border(x1,y1,x2,y2,CYAN,BLUE);
        textcolor(BLUE);
        gotoxy(35,7);
        cprintf("ÑÇ²¥ÁÌÀ¯ÙàÊµýá¥È¿²");
        if(search_data(CYAN,BLUE)==TRUE) run_remove();
        returned_screen(x1,y1,x2,y2,BLUE);
}

removefield()
{
        FILE *fp,*fptr;
        char data[FIELDLENGTH+1];
        register int x1,y1,x2,y2,i,deldat,countrecord,count;

        if(Num_of_Fields == 0)
        {
                warnning(30,"Ô¿àÇ¦¿ÌÁ´Ã¸ÁÌÀ¡ÌÁÔ²á");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        return FALSE;
    } else {
        if(Num_of_Fields%2 == 0)        y1 = 15-(Num_of_Fields+6)/2;
        else y1 = (15-(Num_of_Fields+6)/2)-1;
        y2 = 15+(Num_of_Fields+6)/2;
        x1 = 15; x2 = 65;
        draw_border(x1,y1,x2,y2,WHITE,RED);
        draw_border(x1+1,y2-3,x2-1,y2-1,WHITE,RED);
        for(i=0;i<Num_of_Fields;i++)
        {
                gotoxy(x1+2,y1+3+i);
                cprintf("%s :",field[i]);
        }
        gotoxy(x1+3,y2-2);
        cprintf("ÈÝÄ¢áÊµÚà¹áÊ¥¡ÍÁÃ. :");
        gotoxy((x1+x2)/2-3,y1+1);
        textcolor(BLUE);
        cprintf("Ã.ÈÝÄ¢áÊ");
        textbackground(BLACK);
        textcolor(YELLOW);
        gotoxy(x1+22,y2-2);
        for(i=0;x1+18+i!=x2-7;i++)
                cprintf(" ");
        gotoxy(x1+22,y2-2);
        gets(data);
        if(strcmp(data,"")==0)
        {
                returned_screen(x1,y1,x2,y2,BLUE);
                return FALSE;
        }
        else
        {
                for(i=0;i<Num_of_Fields;i++)
                {
                        if(strcmp(data,field[i])==0) { deldat=i; break; }
                }
        }
        if(i==Num_of_Fields)
        {
                returned_screen(x1,y1,x2,y2,BLUE);
                warnning(35,"Ô¿à¿ÚÈÝÄ¢áÊ·Úá!");
                return FALSE;
        }
        returned_screen(x1,y1,x2,y2,BLUE);
        y2 = y2-3;
        draw_border(x1,y1,x2,y2,WHITE,RED);
        for(i=0;i<Num_of_Fields;i++)
        {
                gotoxy(x1+2,y1+3+i);
                if(i != deldat)         cprintf("%s :",field[i]);
        }
        gotoxy((x1+x2)/2-7,y1+1);
        textcolor(BLUE);
        cprintf("<< ÑÇ²¥ÈÝÄ¢áÊµÝá¥È¿² >>");
        i = decide(55,13,"Ð¡B.¢áÊ¿ØÃ",56,15,"À¡ÐÃÙ¡",WHITE,BLACK,RED+BLINK);
        if(i==0)
        {
                for(i=deldat;i<Num_of_Fields;i++)
                        strcpy(&field[i],&field[i+1]);
                if((fp = fopen("field.nam","w")) == NULL)
                {
                        warnning(25,"Ô¿àÇÍ¿ÍÁ´Ð¹Ù²Ô½ÃäÑÇ²¥ÁÍÀÃÌÐÊÚÀ²Ô²á");
                        return FALSE;
                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        for(i=0;i<Num_of_Fields-1;i++)
                                fprintf(fp,"%s\n",field[i]);
                        fclose(fp);
                        returned_screen(x1,y1,x2,y2,BLUE);
                        /*Ã.¢áÊ¿ØÃ¢É¥Ñ³àÃÌ.×ÉÃ*/
                        if((fp = fopen("temp.dat","w")) != NULL)
                        {
                                if((fptr = fopen("person.dat","r")) != NULL)
                                {
                                        for(countrecord=0;countrecord<numrecord;countrecord++)
                                        {
                                                fseek(fptr,countrecord*sizeof(Datarec),SEEK_SET);
                                                fread(&Datarec,sizeof(struct datatype),1,fptr);
                                                for(count = deldat;count < Num_of_Fields;count++)
                                                        strcpy(Datarec.Info[count],Datarec.Info[count+1]);
                                                fwrite(&Datarec,sizeof(struct datatype),1,fp);
                                        }
                                fclose(fptr);
                                }
                        fclose(fp);
                        }
                        if(remove("person.dat") != 0)
                        { warnning(30,"Ã.Ó½Ãä person.dat Ó¿àÔ²á"); return FALSE;
                        }
                        if(rename("temp.dat","person.dat") != 0)
                        {  warnning(30,"Ã.Ó½Ãä temp.dat Ó¿àÔ²á"); return FALSE;
                        }
                        Num_of_Fields--;
                        return TRUE;
                }
                else { returned_screen(x1,y1,x2,y2,BLUE); return FALSE; }
        }
}

search_data(int gnd,int foregnd)
{
        FILE    *fptr;
        register int x1=15,y1=6,x2=65,i;
        register char countrecord,count;

        if((fptr = fopen("person.dat","r+b")) == NULL)
        {
                warnning(30,"Ó¿àÇÌ¿ÌÁ´Ð¹Ù²Ó½Ãä PERSON.DAT Ô²á");
                return FALSE;
        }
        for(countrecord = 0;countrecord < numrecord;countrecord++)
        {
                fread(&Datarec,sizeof(struct datatype),1,fptr);
                textbackground(gnd);    textcolor(foregnd);
                for(count = 0;count < 2;count++)
                {
                        strcpy(Search_Dat[countrecord].Info[count],Datarec.Info[count]);
                        if(countrecord<14)
                        {
                                gotoxy(x1+5+(30*count),y1+3+countrecord);
                                cprintf("%s",Datarec.Info[count]);
                        }
                }
                /* Seek position of fptr from start of file */
                fseek(fptr,countrecord*sizeof(Datarec),SEEK_SET);
                fwrite(&Datarec,sizeof(struct datatype),1,fptr);
        }
        fclose(fptr);
        textbackground(BLACK); textcolor(YELLOW);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        for(i=x1+1;i!=x2-1;i++) {  gotoxy(i,y1+3); cprintf(" "); }
        for(count=0;count<2;count++)
        {
                gotoxy(x1+5+(30*count),y1+3);
                cprintf("%s",Search_Dat[0].Info[count]);
        }
        return TRUE;
}

run_search()
{
        register int x1=15,x2=65,y1=6,y2=23,i,key,y;
        register char count = 0,countrecord = 0;
        y = y1+3;
        do {
                key = bioskey(0);
                if(key==UP) {
                        /*if(y==23) y=22;*/
                        if(countrecord==0) putch(0x07);
                        else {
                                textbackground(RED); textcolor(WHITE);
                                for(i=x1+1;i!=x2-1;i++) {  gotoxy(i,y); cprintf(" "); }
                                for(count=0;count<2;count++)
                                {
                                        gotoxy(x1+5+(30*count),y);
                                        cprintf("%s",Search_Dat[countrecord].Info[count]);
                                }
                                if(y==y1+3)
                                { movetext(x1+1,y1+3,x2-1,y2-2,x1+1,y1+4); y++;
                                }
                                y--;
                                countrecord--;
                                textbackground(BLACK); textcolor(YELLOW);
                                for(i=x1+1;i!=x2-1;i++) {  gotoxy(i,y); cprintf(" "); }
                                for(count=0;count<2;count++)
                                {
                                        gotoxy(x1+5+(30*count),y);
                                        cprintf("%s",Search_Dat[countrecord].Info[count]);
                                }
                        }
                }
                if(key==DOWN) {
                        if(countrecord==numrecord-1) putch(0x07);
                        else {
                        textbackground(RED); textcolor(WHITE);
                                for(i=x1+1;i!=x2-1;i++) {  gotoxy(i,y); cprintf(" "); }
                                for(count=0;count<2;count++)
                                {
                                        gotoxy(x1+5+(30*count),y);
                                        cprintf("%s",Search_Dat[countrecord].Info[count]);
                                }
                                if(y==y2-1)
                                { movetext(x1+1,y1+4,x2-1,y2-1,x1+1,y1+3); y--;
                                }
                                y++;
                                countrecord++;
                                textbackground(BLACK); textcolor(YELLOW);
                                for(i=x1+1;i!=x2-1;i++) {  gotoxy(i,y); cprintf(" "); }
                                for(count=0;count<2;count++)
                                {
                                        gotoxy(x1+5+(30*count),y);
                                        cprintf("%s",Search_Dat[countrecord].Info[count]);
                                }
                        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
                if(key==ESC) return FALSE;
        }while((key!=ENTER)&&(key!=ESC));
        display(countrecord);
}

display(int countrecord)
{
        FILE    *fptr;
        register int x1,y1,x2,y2,i;
        register char count;

        if((fptr = fopen("person.dat","r")) == NULL)
        {
                warnning(25,"Ô¿àÇÌ¿ÍÁ´Ð¹Ù³Ô½Ãä PERSON.DAT Ô²á");
                return FALSE;
        }
        if(Num_of_Fields%2 == 0)        y1 = 15-(Num_of_Fields+6)/2;
        else y1 = (15-(Num_of_Fields+6)/2)-1;
        y2 = 15+(Num_of_Fields+6)/2;
        x1 = 15; x2 = 65;
        draw_border(x1,y1,x2,y2,WHITE,RED);
        gotoxy((x1+x2)/2-6,y1+1); textcolor(BLUE);
        cprintf("µÌÐ¸ÚÀ·'ÁÌÄá³Ù¸×ÊÊÃ");
        fseek(fptr,countrecord*sizeof(Datarec),SEEK_SET);
        fread(&Datarec,sizeof(struct datatype),1,fptr);
        textbackground(WHITE);  textcolor(RED);
        for(count = 0;count < Num_of_Fields;count++)
        {
                gotoxy(x1+2,y1+3+count);        cprintf("%s",field[count]);
                gotoxy(x1+16,y1+3+count);cprintf(": %s",Datarec.Info[count]);
        }
        getch();
/*      display_image("c:\\windows\\leaves");*/
        returned_screen(x1,y1,x2,y2,BLUE);
        fclose(fptr);
        return TRUE;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Ò¹ÁÑ¡Á¿ Talk to center Ð'B· ÑÊ¼¼ÃÙÐ£¨Ýà· µÚà 2 .· ÐµÊÁä¿Ù·ÊÃ */
email()
{
        char mail[100][39];
        register int x,y,count,old_position,row;

        draw_border(15,7,65,23,WHITE,BLUE);
        textcolor(RED); gotoxy(30,8);
        cprintf("***ELECTRONIC MAIL***");
        textcolor(BLUE);        gotoxy(18,10);
        cprintf(": ¦Ì¡ Terminal µÚà 1");     /*¼ÊÁÄ¿¡Ý. node.c ÑÃáÄ 1 ÓÈáÓ¨á %d,scr);*/
        textcolor(BLUE);        gotoxy(18,11);
        cprintf(": ³áÊ¥¡ÌÁÇà¥¦³È¿ÎÀ·ÚáÔ¹ÀÝ¥ Terminal µÚà ");
        putch(getch());  /*³áÊ¥ÓÇà£à¦·ÚáÓÈá¡Ý. dest*/
        returned_screen(19,15,61,22,BLACK);
        draw_double_border(20,15,60,22,BLACK,LIGHTCYAN);
        textbackground(WHITE); textcolor(RED);  gotoxy(33,14);
        cprintf("Ð·ÜáÊµÚàÐ¢ÚÀ·¦³È¿ÎÀ");
        textbackground(BLACK);
        x=21;y=16;count=0;row=0;
        gotoxy(x,y);
        do {
                old_position = count;
                if(x==59)
                {
                        x=21; count=0; row++;
                        if(y==21)
                        {
                                movetext(21,17,59,21,21,16);
                                textcolor(BLACK); gotoxy(21,21);
                                cprintf("                                       ");
                        }
                        else    y++;
                }
                do {
                        textcolor(YELLOW+BLINK); gotoxy(x,y); cprintf("_");
                }while(!kbhit());
                textcolor(YELLOW); gotoxy(x,y);
                mail[row][count] = getche();
                if(mail[row][count]==13)
                {
                        textcolor(BLACK); gotoxy(x,y); cprintf(" ");
                        old_position = count;
                        x=58;
                }
                if(mail[row][count]==8)
                {
                        if((y==16)&&(x==21)) { putch(0x07); x--; count--; }
                        else if(x==21)
                        {
                                textcolor(BLACK); gotoxy(x,y); cprintf(" ");
                                gotoxy(58,y); printf(" ");
                                x=57; count=old_position; y--; row--;
                        }
                        else
                {
                                textcolor(BLACK); gotoxy(x,y); cprintf(" ");
                                gotoxy(x-1,y); printf(" ");
                                x-=2; count-=2;
                        }
                }
                x++; count++;
        } while(mail[row][count-1]!=27);
        returned_screen(15,6,65,23,BLUE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Ò'ÁÑ¡Á¿ Talk to center Ð'B· ÑÊ¥¼ÃÙÐ£¯Ýà· µÙà 2 .· ÐµÉÁä¿Ù·ÊÃ */
talk_to_center()
{
        char talk[100];
        register int x,y,x1,y1,x2,count;

        draw_border(15,7,65,22,WHITE,RED);
        textcolor(RED); gotoxy(31,8);
        cprintf("********TALK********");
        textcolor(RED); gotoxy(18,10);
        cprintf("Terminal: ");
        returned_screen(7,23,74,23,BLACK);
        textbackground(WHITE); textcolor(RED+BLINK);    gotoxy(55,25);
        cprintf("<ENTER:Áà¥> <ESC:À¡ÐÃÙ¡>");
        textbackground(BLACK); textcolor(YELLOW);
        gotoxy(7,23); cprintf("¸ÁÁ¦×¢éÊÉÃÎ¿ :");
        x=22;y=23;count=0;x1=28;y1=10;x2=62;
        do {
                talk[count]=NULL;
          do {
                        textcolor(YELLOW+BLINK); gotoxy(x,y); cprintf("_");
                }while(!kbhit());
                textcolor(YELLOW); gotoxy(x,y);
                talk[count] = getche();
                if(talk[count]==13)
                {
                        talk[count]=NULL;
                        textbackground(WHITE); textcolor(RED);  gotoxy(x1,y1);
                        if(count<x2-x1) cprintf("%s",talk);
                        else break; /*µÍ'àÉÓ'áÉÙ¡*/
                        y1++;
                        textbackground(WHITE); textcolor(BLACK);
                        gotoxy(18,y1);  cprintf("Center  : ");
                        y1++;
                        textbackground(WHITE); textcolor(RED);
                        gotoxy(18,y1);  cprintf("Terminal: ");
                        returned_screen(7,23,74,23,BLACK);
                        textbackground(BLACK); textcolor(YELLOW);
                        gotoxy(7,23); cprintf("¸ÁÁ¦×¢éÊÉÃÎ¿ :");
                        x=21;/*curser position*/
                        talk[count]=13;
                        count=0;
                }

                if(talk[count]==8)
                {
                        if(x==22) { putch(0x07); x--; count--; }
                        else
                {
                                textcolor(BLACK); gotoxy(x,y); cprintf(" ");
                                gotoxy(x-1,y); printf(" ");
                                x-=2; count-=2;
                        }
                }

                x++; count++;
        } while(talk[count-1]!=27);
        returned_screen(15,6,65,22,BLUE);
        returned_screen(7,23,74,23,BLUE);
        textbackground(WHITE); textcolor(WHITE);
        gotoxy(55,25);  cprintf("<ENTER:Áà¥> <ESC:À¡ÐÃÙ¡>");
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
        BMP reader - copyright (c) 1991
        Alchemy Mindworks Inc.
*/

#include "stdio.h"
#include "dos.h"
#include "alloc.h"

/*                              uncomment this line if you
#include "memmangr.h"           will be linking in the memory manager
*/

#define GOOD_READ       0       /* return codes */
#define BAD_FILE        1
#define BAD_READ        2
#define MEMORY_ERROR    3

#define SCREENWIDE      320     /* mode 13 screen dimensions */
#define SCREENDEEP      200
#define STEP            32      /* size of a step when panning */

#define HOME            0x4700  /* cursor control codes */
#define CURSOR_UP       0x4800
#define CURSOR_LEFT     0x4b00
#define CURSOR_RIGHT    0x4d00
#define END             0x4f00
#define CURSOR_DOWN     0x5000

#define RGB_RED         0
#define RGB_GREEN       1
#define RGB_BLUE        2
#define RGB_SIZE        3

#define pixels2bytes(n) ((n+7)/8)

#define greyvalue(r,g,b)        (((r*30)/100) + ((g*59)/100) + ((b*11)/100))

typedef struct {
        int width,depth,bytes,bits;
        int background;
        char palette[768];
        int (*setup)();
        int (*closedown)();
        } FILEINFO;

typedef struct {
        char id[2];
        long filesize;
        int reserved[2];
        long headersize;
        long infoSize;
        long width;
        long depth;
        int biPlanes;
        int bits;
        long biCompression;
        long biSizeImage;
        long biXPelsPerMeter;
        long biYPelsPerMeter;
        long biClrUsed;
        long biClrImportant;
        } BMPHEAD;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น รวมทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char *farPtr(char *p,long l);
char *getline(unsigned int n);
char *mono2vga(char *p,int width);
char *ega2vga(char *p,int width);
char *rgb2vga(char *p,int width);
int dosetup(FILEINFO *fi);
int doclosedown(FILEINFO *fi);

int putline(char *p,unsigned int n);

char masktable[8]={0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01};

FILEINFO fi;
char *buffer=NULL;

display_image(text)
        char *text;
{
        FILE *fp;
        static char results[8][16] = {  "Ok",
                                        "Bad file",
                                        "Bad read",
                                        "Memory error",
                                        };
        char path[81];
        int r;

                strmfe(path,text,"BMP");
                strupr(path);
                if((fp = fopen(path,"rb")) != NULL) {
                        fi.setup=dosetup;
                        fi.closedown=doclosedown;
                        r=unpackbmp(fp,&fi);
                        printf("\%s",results[r]);
                        fclose(fp);
                } else printf("Error opening %s",path);
}

/* unpack a BMP file */
unpackbmp(fp,fi)
        FILE *fp;
        FILEINFO *fi;
{
        BMPHEAD bmp;
        char *p,*pr;
        int i,n;

        /* set a default monochrome palette */
        memcpy(fi->palette,"\000\000\000\377\377\377",6);

        /* get the header */
        if(fread((char *)&bmp,1,sizeof(BMPHEAD),fp)==sizeof(BMPHEAD)) {

                /* check the header */
                if(!memcmp(bmp.id,"BM",2)) {

                        /* set the details */
                        fi->width=(int)bmp.width;
                        fi->depth=(int)bmp.depth;
                        fi->bits=bmp.bits;

                        /* work out the line width */
                        if(fi->bits==1) fi->bytes=pixels2bytes(fi->width);
                        else if(fi->bits==4) fi->bytes=pixels2bytes(fi->width)<<2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else if(fi->bits==8) fi->bytes=fi->width;
else if(fi->bits==24) fi->bytes=fi->width*3;

/* round up to an even dword boundary */
if(fi->bytes & 0x0003) {
        fi->bytes |= 0x0003;
        ++fi->bytes;
}

/* get the palette */
if(fi->bits > 1 && fi->bits <=8) {
        n=1<<fi->bits;
        for(i=0;i<n;++i) {
                fi->palette[(i*RGB_SIZE)+RGB_BLUE]=fgetc(fp);
                fi->palette[(i*RGB_SIZE)+RGB_GREEN]=fgetc(fp);
                fi->palette[(i*RGB_SIZE)+RGB_RED]=fgetc(fp);
                fgetc(fp);
        }
}
else if(fi->bits==24) {
        for(i=0;i<256;++i)
            memset(fi->palette+(i*RGB_SIZE),i,RGB_SIZE);
}

/* allocate a line buffer */
if((p=malloc(fi->bytes)) != NULL) {

        if((fi->setup)(fi) != GOOD_READ) {
                free(p);
                return(MEMORY_ERROR);
        }

        /* find the start of the image data */
        fseek(fp,bmp.headersize,SEEK_SET);

        /* read all the lines */
        for(i=0;i<fi->depth;++i) {
                if(fread(p,1,fi->bytes,fp) != fi->bytes) {
                        freebuffer();
                        free(p);
                        return(BAD_READ);
                }

                /* translate the line types into VGA */
                switch(fi->bits) {
                        case 1:
                                pr=mono2vga(p,fi->width);
                                if(pr != NULL) {
                                        putline(pr,fi->depth-1-i);
                                        free(pr);
                                }
                                else {
                                        freebuffer();
                                        free(p);
                                        return(MEMORY_ERROR);
                                }
                                break;
                        case 4:
                                pr=ega2vga(p,fi->width);
                                if(pr != NULL) {
                                        putline(pr,fi->depth-1-i);
                                        free(pr);
                                }
                                else {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        freebuffer();
                        free(p);
                        return(MEMORY_ERROR);
                }
                break;
        case 8:
                putline(p,fi->depth-1-i);
                break;
        case 24:
                pr=rgb2vga(p,fi->width);
                if(pr != NULL) {
                        putline(pr,fi->depth-1-i);
                        free(pr);
                }
                else {
                        freebuffer();
                        free(p);
                        return(MEMORY_ERROR);
                }
                break;
                        }
                }
                (fi->closedown)(fi);
                free(p);
                return(GOOD_READ);
            } else return(MEMORY_ERROR);
        } else return(BAD_FILE);
    } else return(BAD_READ);
}

/* convert a monochrome line into an eight bit line */
char *mono2vga(p,width)
    char *p;
    int width;
{
    char *pr;
    int i;

    if((pr=malloc(width)) != NULL) {
        for(i=0;i<width;++i) {
            if(p[i >> 3] & masktable[i & 0x0007])
                pr[i]=1;
            else
                pr[i]=0;
        }
        return(pr);
    } else return(NULL);
}

/* convert a four bit line into an eight bit line */
char *ega2vga(p,width)
    char *p;
    int width;
{
    char *pr;
    int i,j=0;

    if((pr=malloc(width)) != NULL) {
        for(i=0;i<width;) {
            pr[i++]=(p[j] >> 4) & 0x0f;
            pr[i++]=p[j] & 0x0f;
            ++j;
        }
        return(pr);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        } else return(NULL);
}

/* convert an RGB line into an eight bit line */
char *rgb2vga(p,width)
        char *p;
        int width;
{
        char *pr;
        int i;

        if((pr=malloc(width)) != NULL) {
                for(i=0;i<width;++i) {
                        pr[i]=greyvalue(p[RGB_RED],p[RGB_GREEN],p[RGB_BLUE]);
                        p+=RGB_SIZE;
                }
                return(pr);
        } else return(NULL);
}

/* this function is called before an image is unpacked */
dosetup(fi)
        FILEINFO *fi;
{
        union REGS r;

        if(!getbuffer((long)fi->width*(long)fi->depth,fi->width,fi->depth))
           return(MEMORY_ERROR);

        r.x.ax=0x0013;
        int86(0x10,&r,&r);

        setvgapalette(fi->palette,256,fi->background);

        return(GOOD_READ);
}

/* This function a called after an image has been unpacked. It must
   display the image and deallocate memory. */
doclosedown(fi)
        FILEINFO *fi;
{
        union REGS r;
        int c,i,n,x=0,y=0;

        if(fi->width > SCREENWIDE) n=SCREENWIDE;
        else n=fi->width;

        do {
                for(i=0;i<SCREENDEEP;++i) {
                        c=y+i;
                        if(c>=fi->depth) break;
                        memcpy(MK_FP(0xa000,SCREENWIDE*i),getline(c)+x,n);
                }
                c=GetKey();
                switch(c) {
                        case CURSOR_LEFT:
                                if((x-STEP) > 0) x-=STEP;
                                else x=0;
                                break;
                        case CURSOR_RIGHT:
                                if((x+STEP+SCREENWIDE) < fi->width) x+=STEP;
                                else if(fi->width > SCREENWIDE)
                                    x=fi->width-SCREENWIDE;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        else x=0;
                        break;
                case CURSOR_UP:
                        if((y-STEP) > 0) y-=STEP;
                        else y=0;
                        break;
                case CURSOR_DOWN:
                        if((y+STEP+SCREENDEEP) < fi->depth) y+=STEP;
                        else if(fi->depth > SCREENDEEP)
                            y=fi->depth-SCREENDEEP;
                        else y=0;
                        break;
                case HOME:
                        x=y=0;
                        break;
                case END:
                        if(fi->width > SCREENWIDE)
                            x=fi->width-SCREENWIDE;
                        else x=0;
                        if(fi->depth > SCREENDEEP)
                            y=fi->depth-SCREENDEEP;
                        else y=0;
                        break;
                }
        } while(c != 27);

        freebuffer();

        r.x.ax=0x0003;
        int86(0x10,&r,&r);
        return(GOOD_READ);
}

/* get one extended key code */
GetKey()
{
        int c;

        c = getch();
        if(!(c & 0x00ff)) c = getch() << 8;
        return(c);
}

/* set the VGA palette and background */
setvgapalette(p,n,b)
        char *p;
        int n,b;
{
        union REGS r;
        int i;

        outp(0x3c6,0xff);
        for(i=0;i<n;++i) {
                outp(0x3c8,i);
                outp(0x3c9,(*p++) >> 2);
                outp(0x3c9,(*p++) >> 2);
                outp(0x3c9,(*p++) >> 2);
        }
        r.x.ax=0x1001;
        r.h.bh=b;
        int86(0x10,&r,&r);
}

/* make file name with specific extension */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
strmfe(new,old,ext)
        char *new,*old,*ext;
{
        while(*old != 0 && *old != '.') *new++=*old++;
        *new++='.';
        while(*ext) *new++=*ext++;
        *new=0;
}

/* if you don't use in the memory manager, these functions
   will stand in for it */

#if !MEMMANGR

/* return a far pointer plus a long integer */
char *farPtr(p,l)
        char *p;
        long l;
{
        unsigned int seg,off;

        seg = FP_SEG(p);
        off = FP_OFF(p);
        seg += (off / 16);
        off &= 0x000f;
        off += (unsigned int)(l & 0x000fL);
        seg += (l / 16L);
        p = MK_FP(seg,off);
        return(p);
}

/* save one line to memory */
putline(p,n)
        char *p;
        unsigned int n;
{
        if(n >= 0 && n < fi.depth)
            memcpy(farPtr(buffer,(long)n*(long)fi.width),p,fi.width);
}

/* get one line from memory */
char *getline(n)
        unsigned int n;
{
        return(farPtr(buffer,(long)n*(long)fi.width));
}

#pragma warn -par
getbuffer(n,bytes,lines)
        unsigned long n;
        int bytes,lines;
{
        if((buffer=farmalloc(n)) == NULL) return(0);
        else return(1);
}

freebuffer()
{
        if(buffer != NULL) farfree(buffer);
        buffer=NULL;
}
#endif /* !MEMMANGR */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. Gred E. Keiper "LOCAL AREA NETWORKS " McGraw-Hill Book company

2. น.ต.ดร.ไพศาล สงวนหมู่ ,ร.ศ.ยืน ภู่วรรณ "การสื่อสารข้อมูลและไมโครคอมพิวเตอร์เนตเวอร์ค" บริษัท ซีเอ็ดยูเคชั่น จำกัด

3. รศ.ดร. วิวัฒน์ กิรานนท์ "การสื่อสารข้อมูลดิจิตอล" สถาบันเทคโนโลยี พระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

4. William L.Schweber "DATA COMMUNICATION" McGraw-Hill international Edition

5. The Peter Norton Programming Library "ADVANCE C PROGRAMMING"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้