



การสื่อสารระยะไกล ด้วย RS-485

TELECOMMUNICATION BY RS-485

โดย

นายเกษิษฐ์ อ่อนชุม

นายชัยยุทธ หงษ์โสกา

นายประพนธ์ วงษ์ธรรม

วัน เดือน ปี..... 24.ค.ค. 2541
เลขทะเบียน..... 039154
เลขเรียกหนังสือ..... 100393 ก 111 ก..

ปฏิญานีพจน์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่น ๆ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

039154

การสื่อสารระยะไกล ด้วย RS-485
TELECOMMUNICATION BY RS-485

โดย

นายเกษิษฐ์

อ่อนนุช 38013177

นายรัชบุท

หงษ์โสภา 38013180

นายประพนธ์

วงษ์ธรรม 38013193

อาจารย์ที่ปรึกษา

อ.ชินภัทร

นันทจิวารัชย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาบัตรเรื่อง

การสื่อสารระยะไกลด้วย RS-485

TELECOMMUNICATION BY RS-485

จัดทำโดย

นายเกษิษฐ์ อ่อนชุม

นายชัยยุทธ หงษ์โสภา

นายประพจน์ วงษ์ธรรม

อาจารย์ที่ปรึกษา

อ.ชินภัทร นันทจิวากรชัย



ปริญญาบัตรฉบับนี้ได้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ลงชื่อ

อาจารย์ที่ปรึกษา

(อ.ชินภัทร นันทจิวากรชัย)

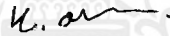
วันที่ ๒๐ / ๒๖๖ / ๒๕๖๑

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ ประสบความสำเร็จได้ ก็เพราะได้รับความร่วมมือจากหลายฝ่าย รวมทั้งอาจารย์ที่คอยให้คำแนะนำ ทั้งในด้านทฤษฎีและปฏิบัติ คณะผู้จัดทำขอขอบพระคุณผู้ที่มีส่วนสำคัญที่สุดของปริญญานิพนธ์นี้ **อ.ชินภัทร นันทจิวงกรชัย** ที่ได้ช่วยเป็นอาจารย์ที่ปรึกษา ในการจัดทำปริญญานิพนธ์ฉบับนี้ ให้สำเร็จลุล่วงไปด้วยดี

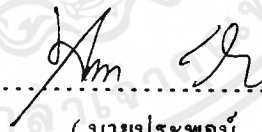
และขอขอบคุณ คณะอาจารย์และเจ้าหน้าที่ภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ รวมทั้งเพื่อนๆ นักศึกษาทุกคน และผู้มีส่วนร่วมในการจัดทำปริญญานิพนธ์ฉบับนี้ ให้สำเร็จตาม วัตถุประสงค์ทุกท่าน



(นายเกษิษฐ์ อ่อนนวม)



(นายชัยยุทธ หงษ์โสภ)



(นายประพจน์ วงษ์ธรรม)

คณะผู้จัดทำ

การสื่อสารระยะไกล ด้วย RS-485

นายเกษิษฐ์	อ่อนชุม
นายชัยยุทธ	หงษ์โสภ
นายประพจน์	วงษ์ธรรม
อ.ชินภัทร	นันทจิวารัชย์ (อาจารย์ที่ปรึกษา)

บทคัดย่อ

ในการสื่อสารข้อมูลจะมีทั้งการสื่อสารแบบอนุกรมและแบบขนาน โดยทั่วไปจะใช้การสื่อสารแบบอนุกรมตามมาตรฐาน RS-232C แต่จะทำการส่งข้อมูลได้ไม่ไกลนัก ในโครงการนี้จะใช้การสื่อสารตามมาตรฐาน RS-485 ซึ่งเป็นแบบอนุกรมและแบบสมดุล ข้อดีของระบบนี้คือสามารถรับ-ส่งข้อมูลได้ในระยะไกล ประมาณ 4000 ฟุต, ทนต่อสัญญาณรบกวนได้ดี และสามารถใช้งานในช่วงแรงดันที่กว้างขึ้น จุดเด่นของระบบ RS-485 คือ ใช้งานในระบบหลายจุด คือ สามารถติดต่อกันระหว่างไมโครคอมพิวเตอร์กับไมโครคอมพิวเตอร์ หรือไมโครคอนโทรลเลอร์ ได้ถึง 32 ตัว และสามารถใช้ร่วมกับโปรโตคอลที่กำหนดมาใช้งานในระบบได้ สำหรับในโครงการนี้ เราจะเสนอการใช้เมโมรี่แบบสัมผัส (Touch Memory) ร่วมกับระบบ RS-485

TELECOMMUNICATION BY RS-485

Mr. Kadit	Ohnchum
Mr. Chaiyoot	Hongsopha
Mr. Prapot	Wongtham
Mr. Chinapat	Nuntajiwakornchai (Advisor)

Abstract

In Data Communication there are Serial Communication and Parallel Communication. The most popular used is RS-232C but its disadvantages in transmit and receive data in short distances. In this thesis we used RS-485. It is a serial data communication and balance voltage. The advantages of this method are it able to transmit and receive data about 4000 feet, good protection of noise, transmit and receive data clearly and it available for use in a wide range. The most important advantage of this system that it can be use in a multiprocessor or multidrop with microprocessor units. In this thesis we will communicate between microcomputer and microcontroller. And this communication can be use with our buliding protocol " RS-485 PROTOCOL " Furthermore in this we will show the application of RS-485 communication system, Touch memory in access control system.

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
Abstract	III
สารบัญ	IV
บทที่ 1 บทนำ	1
บทที่ 2 รูปแบบของการสื่อสารข้อมูล	3
2.1 การสื่อสารข้อมูลแบบขนาน	3
2.2 การสื่อสารข้อมูลแบบอนุกรม	4
2.2.1 การสื่อสารข้อมูลแบบอนุกรมที่แบ่งตามทิศทางของข้อมูล	5
2.2.2 การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232C	6
2.2.3 การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-422A	7
2.2.4 การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-485	8
2.3 รูปแบบของการสื่อสารข้อมูลแบบ RS-485	11
2.3.1 เครือข่ายที่ใช้สายนำสัญญาณสองเส้น (Two Wire)	11
2.3.2 เครือข่ายที่ใช้สายนำสัญญาณสี่เส้น (Four Wire)	12
2.4 รายละเอียดของการสื่อสารข้อมูลแบบ RS-485	13
2.4.1 คุณสมบัติเฉพาะของตัวส่ง RS-485	13
2.4.2 คุณสมบัติเฉพาะของตัวรับ RS-485	13
2.4.3 คุณสมบัติเฉพาะของคู่สายสัญญาณ RS-485	13
2.4.4 ความหมายของยูนิค โหลด	13
2.4.5 คุณสมบัติเฉพาะของคู่ตัวรับ-ส่ง	14
2.5 การคำนวณหาจำนวนคู่ตัวรับ-ส่ง ของ RS-485	14
2.6 RS-485 โพรโตคอล (RS-485 Protocol)	14
2.7 ลักษณะของเครือข่ายแบบบัส	19
2.8 วิธีการเข้าสู่ช่องสัญญาณ	20
บทที่ 3 ทักษะโมรี (Touch Memory)	23
3.1 คุณสมบัติของทักษะโมรี	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

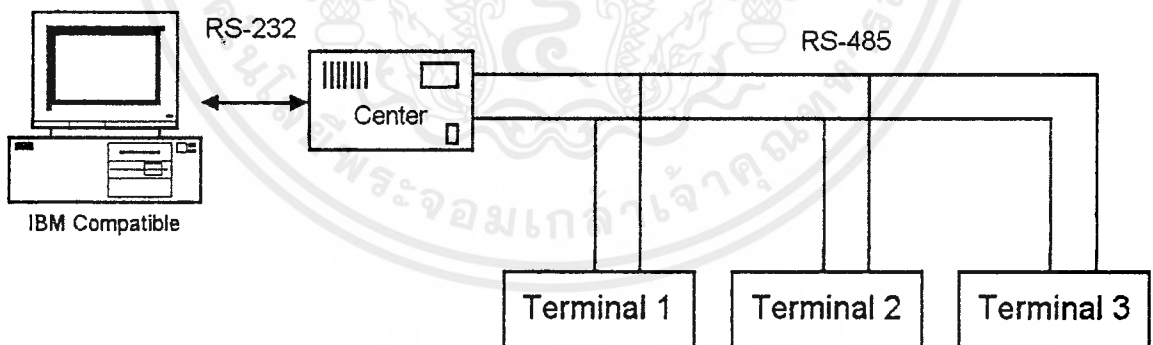
3.2 การทำงานของทัชเม โมรี	25
3.2.1 การเขียนข้อมูลลงบนทัชเม โมรี	25
3.2.2 การอ่านข้อมูลจากทัชเม โมรี	26
บทที่ 4 โครงสร้างของระบบ	27
4.1 โครงสร้างการทำงานของชุดควบคุมหลัก	28
4.1.1 หลักการทำงานของชุดควบคุมหลัก (Master Control)	28
4.2 โครงสร้างการทำงานของชุดลูกข่าย	30
4.2.1 หลักการทำงานของชุดลูกข่าย (Slave)	30
บทที่ 5 การทดลอง	32
บทที่ 6 สรุปและวิจารณ์	33
ภาคผนวก	
ก. วงจร	
ข. ลายวงจรและแผ่นพิมพ์	
ค. โฟล์ดชาร์ตการทำงานของระบบ	

บทที่ 1

บทนำ

ในการสื่อสารข้อมูลอนุกรมระหว่างคอมพิวเตอร์ด้วยกัน และระหว่างคอมพิวเตอร์กับอุปกรณ์รอบข้าง ด้วยตัวรับและตัวส่งแบบไม่สมดุลย์ เช่น RS-232C และ RS-423A นั้น จะพบว่า มีข้อบกพร่องต่างๆ เกิดขึ้นหลายประการ เช่น ข้อมูลเกิดการผิดพลาดได้ง่าย เมื่อมีการรับส่งมูลสูง และการใช้คู่สายสัญญาณรับ-ส่งข้อมูลที่มีความยาวมากๆ ในการรับส่งข้อมูลที่มีระยะทางไกลๆ ตลอดถึงการวางคู่สายสัญญาณส่งผ่านบริเวณที่มีสัญญาณรบกวนสูง ด้วยสาเหตุดังกล่าวจึงได้มีการพัฒนามาตรฐานการสื่อสารข้อมูลอนุกรมแบบสมดุลย์ขึ้นมา(ตัวรับ-ตัวส่งไม่ใช่สายสัญญาณร่วมกัน) เพื่อแก้ปัญหาเหล่านั้น ซึ่งการสื่อสารอนุกรมแบบสมดุลย์นี้จะมีการประยุกต์ใช้งานที่แตกต่างกัน ขึ้นอยู่กับความเหมาะสมและประเภทของงานแต่ละประเภท

ในปริณิญาณิพนธ์รายงานนี้จะเป็นการเสนอรูปแบบของการสื่อสารแบบอนุกรมแบบ RS-485 ซึ่งทำงานร่วมกับโปรโตคอลสำหรับการสื่อสารข้อมูลในระบบ และนอกจากนี้เรายังสามารถพัฒนาระบบการสื่อสารข้อมูลแบบอนุกรมแบบ RS-485 โดยจะแสดงให้เห็นการประยุกต์ใช้งานหน่วยความจำแบบสัมผัส (Touch Memory) เป็นการสนับสนุนระบบการสื่อสารข้อมูล โดยมีรูปแบบของระบบดังนี้ คือ



รูปที่ 1.1 แสดงโครงสร้างของการสื่อสารข้อมูล RS-485

สำหรับรายละเอียดของปริญญาานิพนธ์มีดังนี้

- บทที่ 1 เป็นบทนำ กล่าวถึงการสื่อสารข้อมูลของ ไมโครคอมพิวเตอร์และสาเหตุของการพัฒนาระบบ RS-485
- บทที่ 2 เป็นการกล่าวถึงรูปแบบของการสื่อสารข้อมูลแบบอนุกรม, แบบขนาน
- บทที่ 3 เป็นการกล่าวถึงรายละเอียดของทัชเม โมรี (Touch Memory) ที่ได้นำมาประยุกต์ใช้งาน
- บทที่ 4 กล่าวถึงโครงสร้างของระบบ
- บทที่ 5 กล่าวถึงผลการทดลอง ในส่วนต่างๆ ของระบบที่ได้ทำการออกแบบ
- บทที่ 6 เป็นบทสรุปผลการทดลอง และบทวิจารณ์ผลการทดลอง

ในส่วนท้ายของปริญญาานิพนธ์นี้เป็นภาคผนวก มีรายละเอียดดังนี้

- ภาคผนวก ก แสดงรูปวงจรรวมทั้งหมดของส่วนต่างๆ แต่ละส่วน
- ภาคผนวก ข แสดงรูปวงจรพิมพ์ลายวงจรทั้งหมดของแต่ละส่วน
- ภาคผนวก ค แสดงไฟล์ชาร์ตการทำงานของระบบ

บทที่ 2

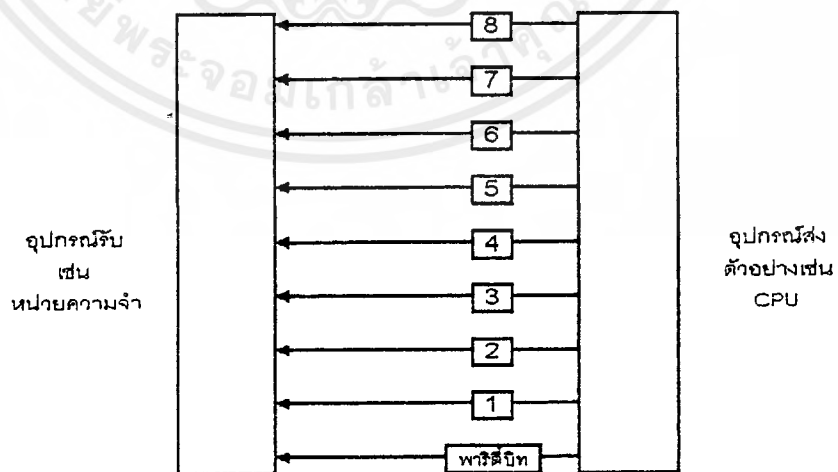
รูปแบบของการสื่อสารข้อมูล

โดยทั่วไป หลักใหญ่ของการส่งข้อมูลในคอมพิวเตอร์ หรือระหว่างคอมพิวเตอร์ด้วยกัน จะมีลักษณะของการส่งหรือการสื่อสารข้อมูลอยู่ 2 ลักษณะ คือ การสื่อสารข้อมูลแบบขนาน และการสื่อสารข้อมูลแบบอนุกรม ซึ่งแต่ละลักษณะจะมีรายละเอียดดังนี้ คือ

2.1 การสื่อสารข้อมูลแบบขนาน

ลักษณะของการสื่อสารข้อมูลแบบขนานนั้น จะเป็นการสื่อสารข้อมูลที่ข้อมูลจะรับ-ส่ง โดยผ่านสายหรือช่องทางสัญญาณพร้อมกันหลายๆ เส้น ดังแสดงในรูปที่ 2.1 โดยที่จำนวนของสัญญาณจะมีจำนวนไม่แน่นอนขึ้นอยู่กับโครงสร้างการประมวลผลข้อมูลของระบบนั้นๆ ข้อดีของการสื่อสารข้อมูลแบบนี้ คือ สามารถสื่อสารข้อมูลกันได้ในระยะเวลาดังนั้นๆ แต่ก็มีข้อเสีย คือ จะสิ้นเปลืองสายสัญญาณเป็นจำนวนมาก และถ้ายังใช้ในการสื่อสารข้อมูลในระยะทางไกลๆ นอกจากจะสิ้นเปลืองค่าใช้จ่ายจำนวนมากแล้ว ยังทำให้สัญญาณถูกลดทอนลงไปด้วย

ดังนั้นโดยทั่วไปแล้วการสื่อสารข้อมูลแบบขนานนิยมนำไปใช้กับการสื่อสารข้อมูลในระยะทางสั้นๆ ที่ต้องการสื่อสารข้อมูลด้วยอัตราเร็ว เช่น การเชื่อมต่อของสัญญาณระหว่างหน่วยประมวลผลกลางกับอุปกรณ์รอบข้าง หรือ การสื่อสารข้อมูลของเครื่องมือวัดและอุปกรณ์ต่างๆ กับเครื่องไมโครคอมพิวเตอร์



รูปที่ 2.1 แสดงโครงสร้างของการสื่อสารข้อมูลแบบขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การสื่อสารข้อมูลแบบอนุกรม

ลักษณะของการสื่อสารข้อมูลแบบอนุกรมนั้นจะเป็นการสื่อสารที่จะทำการรับ-ส่งข้อมูล โดยใช้สายจำนวนน้อย ซึ่งปกติจะใช้เพียง 1 คู่ เท่านั้น คือ สายสัญญาณที่จะใช้เป็นสายข้อมูล และสายกราวนด์

ลักษณะของการรับ-ส่งข้อมูลนั้น ข้อมูลจะถูกส่งออกไปหรือรับเข้ามาในลักษณะที่เป็นบิตต่อบิต ซึ่งถ้าหากเปรียบเทียบกันกับการสื่อสารแบบขนานที่จำนวนข้อมูลและอัตราเร็วในการสื่อสารข้อมูลเท่ากันแล้วจะพบว่า การสื่อสารข้อมูลแบบอนุกรมจะต้องใช้เวลาในการรับ-ส่งข้อมูลมากกว่าอย่างแน่นอน แต่เมื่อพิจารณาถึงข้อดีของการสื่อสารข้อมูลแบบอนุกรมแล้วจะพบว่า ข้อดีของการสื่อสารข้อมูลแบบอนุกรมคือ การใช้สายสัญญาณน้อยกว่า และสามารถส่งสัญญาณได้ในระยะทางที่ไกลกว่า แม้ว่าอัตราการลดทอนหรือผิดเพี้ยนของสัญญาณที่มีผลจากความยาวของสายสัญญาณจะมีค่าเท่ากับการสื่อสารข้อมูลแบบขนาน แต่การสื่อสารข้อมูลแบบอนุกรมจะมีวิธีการที่จะลดผลจากการลดทอนของสัญญาณนี้ โดยอาศัยหลักการรับ-ส่งสัญญาณแบบดิฟเฟอเรนเชียล

ดังนั้นการสื่อสารข้อมูลแบบอนุกรมจึงเหมาะสำหรับใช้กับการสื่อสารข้อมูลในระยะไกล หรือการสื่อสารที่ต้องการใช้สายหรือช่องสัญญาณในการรับ-ส่งข้อมูลจำนวนน้อย เช่นการสื่อสารข้อมูลโครงข่ายแบบท้องถิ่น (Local Area Network:LAN)



รูปที่ 2.2 แสดงโครงสร้างของการสื่อสารข้อมูลแบบอนุกรม

2.2.1 การสื่อสารข้อมูลแบบอนุกรมที่แบ่งตามทิศทางของข้อมูล

นอกจากที่ได้ไปแล้วนั้นจะพบว่า การสื่อสารข้อมูลแบบอนุกรมยังสามารถที่จะแบ่งตามลักษณะของทิศทางในการสื่อสารข้อมูล ตามโครงสร้างและความต้องการของระบบ ได้ดังต่อไปนี้ คือ

ก.) การสื่อสารข้อมูลในทิศทางเดียวตลอดเวลาหรือแบบซิมเพล็กซ์(Simplex)

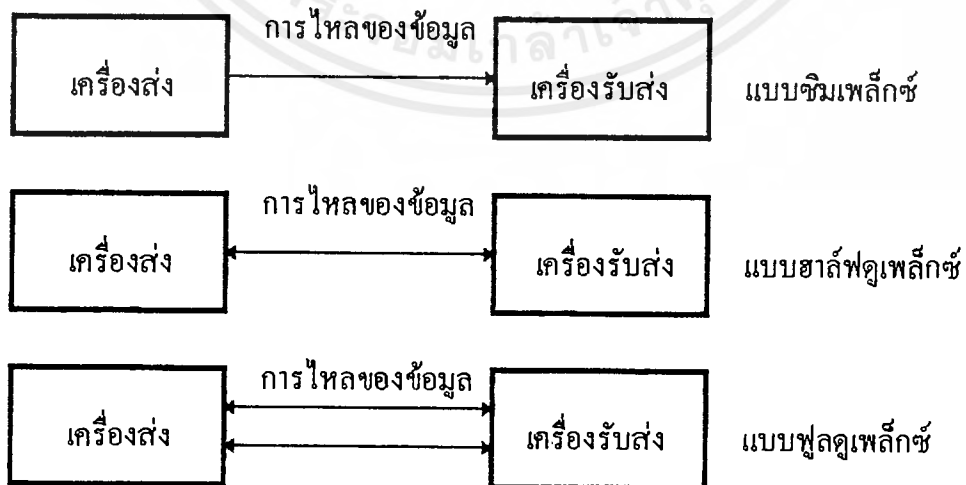
เป็นการสื่อสารข้อมูลที่ข้อมูลสามารถส่งได้ในทางเดียวเท่านั้น เมื่อทำการสื่อสารในทิศทางใดก็จะใช้ทิศทางนั้นตลอดเวลาไม่มีการเปลี่ยนแปลงทิศทาง เช่น การส่งสัญญาณภาพจากสถานีโทรทัศน์ไปยังเครื่องรับโทรทัศน์ หรือการส่งข้อมูลจากศูนย์บริการไปยังวิทยุติดตามตัว

ข.) การสื่อสารข้อมูลแบบสองทิศทางแต่คนละเวลาหรือแบบฮาล์ฟดูเพล็กซ์(Half duplex)

เป็นการสื่อสารข้อมูล ที่สามารถส่งข้อมูลได้ทั้ง 2 ทิศทาง โดยจะทำการส่งในลักษณะของการผลัดกันรับและส่ง โดยในขณะเวลาหนึ่งนั้นสัญญาณจะไปได้ในทิศทางเดียวเท่านั้น ดังนั้นอุปกรณ์แต่ละตัวที่จะเชื่อมต่อหรือสื่อสารข้อมูลในลักษณะนี้ จะต้องเป็นได้ทั้งตัวรับและตัวส่ง ซึ่งมีชื่อเรียกว่า *ทรานซีฟเวอร์* (Tranceiver) และจะต้องมีวงจรที่เล็กกว่า ณ เวลานั้นจะทำงานเป็นตัวรับหรือตัวส่ง

ค.) การสื่อสารข้อมูลแบบสองทิศทางตลอดเวลาหรือแบบฟูลดูเพล็กซ์(Full duplex)

เป็นการสื่อสารข้อมูลที่คล้ายกับแบบฮาล์ฟดูเพล็กซ์ แต่เป็นการสื่อสารข้อมูลใน 2 ทิศทางแบบตลอดเวลา



รูปที่ 2.3 รูปแบบของการติดต่อสื่อสารข้อมูลแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232C

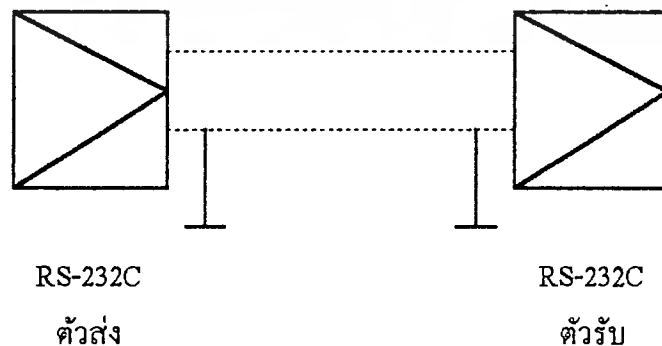
การสื่อสารข้อมูลแบบอนุกรมที่มีการใช้งานกันอยู่ในปัจจุบันนี้นั้น ได้มีการกำหนดมาตรฐานในการรับ-ส่งข้อมูลไว้หลายแบบด้วยกัน แต่ที่ได้รับความนิยมนำมาใช้งานอย่างกว้างขวาง ก็คือ การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232C

ทั้งนี้เนื่องจากความต้องการในการสื่อสารข้อมูลผ่านทางเครือข่ายโทรศัพท์ที่มีมากขึ้นเรื่อยๆ ดังนั้นจึงได้มีการกำหนดมาตรฐานที่เรียกว่า RS-232C ขึ้น เพื่อใช้เป็นมาตรฐานแก่อุปกรณ์ที่ถูกผลิตขึ้นจากบริษัทต่างๆ ในสหรัฐอเมริกา Bell System operating telephone companies เป็นบริษัทหลักบริษัทแรกที่เป็นผู้ผลิตและติดตั้งระบบสื่อสารข้อมูล และเป็นผู้ผลิตอุปกรณ์ต่างๆ ที่ใช้ในการอินเทอร์เฟซอุปกรณ์ดิจิทัลกับเครือข่ายโทรศัพท์รายใหญ่ อุปกรณ์นี้ก็คือ Bell modem ซึ่งถูกพัฒนาโดย Bell Laboratories และถูกใช้เป็นมาตรฐานในงานอุตสาหกรรมจนถึงปัจจุบันนี้ (บริษัทต่างๆ มักจะลอกข้อกำหนดต่างๆ ของ Bell ไปใช้งาน เพื่อให้สินค้าของบริษัทนั้นๆ สามารถใช้กับอุปกรณ์ของ Bell ได้) ดังนั้นความต้องการข้อมูลเกี่ยวกับข้อกำหนดในการอินเทอร์เฟซกับโมเด็มจึงมีเพิ่มขึ้นเรื่อยๆ เพื่อตอบสนองต่อความต้องการนี้ EIA, Bell System และผู้ผลิตโมเด็มรายอื่นๆ จึงได้ร่วมมือกันตั้งมาตรฐาน RS-232C ขึ้น

มาตรฐาน RS-232C ได้ถูกตีพิมพ์โดย EIA ในปี ค.ศ. 1969 โดยตัวอักษร RS แทน "Recommended Standard" 232 แทนหมายเลขของมาตรฐาน ส่วนอักษร C แสดงให้เห็นว่ามาตรฐานนั้นได้รับการแก้ไขกี่ครั้ง

การที่มาตรฐานนี้เป็นที่นิยมใช้ ก็เนื่องจากเป็นระบบการสื่อสารข้อมูลที่ใช้ในเครื่องไมโครคอมพิวเตอร์ IBM PC ซึ่งเป็นคอมพิวเตอร์ที่มีใช้อย่างแพร่หลายตั้งแต่อดีตมาจนถึงปัจจุบัน

มาตรฐาน RS-232C จะมีโครงสร้างการสื่อสารเป็นแบบจุดต่อจุดเท่านั้น โดยมีลักษณะสมบัติทางไฟฟ้าและทางกายภาพ ดังแสดงในตารางที่ 2.1 และรูปที่ 2.4



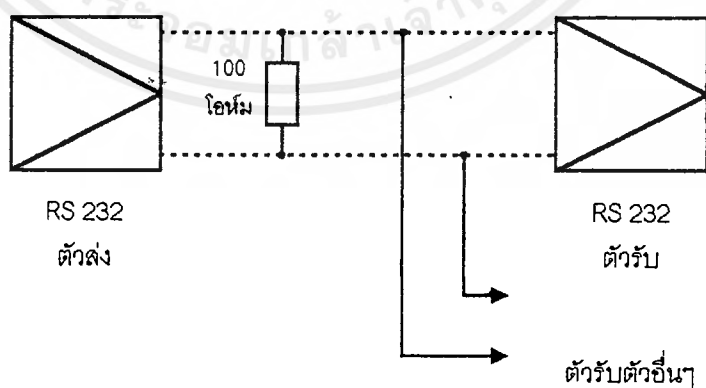
รูปที่ 2.4 แสดงโครงสร้างของการสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-422A

ในการออกแบบระบบการสื่อสารข้อมูลที่ได้กล่าวมาแล้วนั้น ได้มีการพยายามที่จะออกแบบให้มีการสื่อสารข้อมูลได้รวดเร็วยิ่งขึ้น และมีระยะทางในการสื่อสารข้อมูลที่ยาวขึ้นด้วย ซึ่งที่ผ่านมาการสื่อสารข้อมูลตามมาตรฐาน RS-232C ได้ออกแบบมาเพื่อใช้เชื่อมโยงกับโมเด็มเท่านั้น จึงไม่ได้คำนึงถึงความเร็วและระยะทางในการสื่อสาร แต่ปัจจุบันได้มีการออกแบบระบบมาเพื่อรองรับความต้องการของการใช้งานที่ต้องการให้รับและส่งข้อมูลได้ไกลขึ้น คือ มาตรฐาน RS-422A ซึ่งจะใช้สัญญาณแบบดิฟเฟอเรนเชียล ดังรูปที่ 2.5

หลักการของการสื่อสารข้อมูลตามมาตรฐาน RS-422A คือ สัญญาณที่จะรับ-ส่ง จะเป็นการเปรียบเทียบระหว่างสัญญาณ 2 เส้น เทียบกับมาตรฐาน RS-232C ที่สัญญาณทุกสัญญาณจะเทียบกับกราวด์ ซึ่งในการสื่อสารในระยะทางไกลๆ แล้วนั้น สัญญาณจะถูกลดทอนไปและเมื่อสัญญาณถูกลดทอนถึงจุดๆ หนึ่ง สัญญาณก็จะผิดพลาดไปจากความเป็นจริง ซึ่งก็จะมีผลทำให้การรับ-ส่งข้อมูลเกิดความผิดพลาดขึ้น แต่สำหรับ RS-422A ซึ่งใช้สัญญาณดิฟเฟอเรนเชียลแล้ว การลดทอนของสัญญาณจะ ไปลดทอนสัญญาณทั้งสองสายด้วยค่าที่เท่ากันหรือใกล้เคียงกัน และความแตกต่างของสัญญาณ หรือระยะสัญญาณทั้งสองเส้น จากตัวส่งไปยังตัวรับก็ยังคงมีค่าเท่าเดิมหรือมีการเปลี่ยนแปลงน้อย จึงทำให้ผลของการลดทอนต่อสัญญาณที่การสื่อสารข้อมูลในระยะทางไกลๆ มีผลต่อสัญญาณดิฟเฟอเรนเชียลน้อยกว่า การสื่อสารข้อมูลแบบนี้จึงสามารถส่งข้อมูลได้ไกลกว่า และอัตราการสื่อสารข้อมูลสูงกว่า ดังแสดงการเปรียบเทียบการสื่อสารข้อมูลดังตารางที่ 2.2



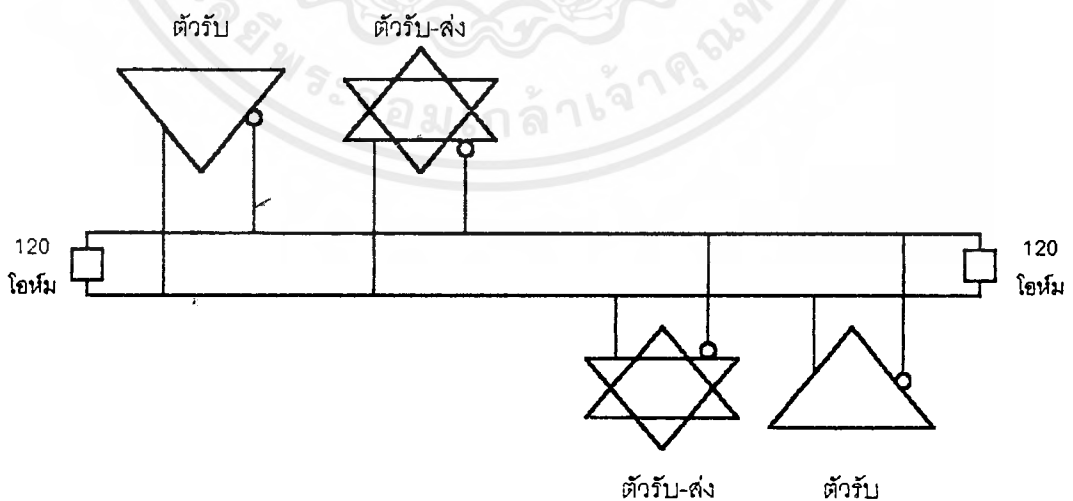
รูปที่ 2.5 แสดงโครงสร้างของการสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-422A

2.3.4 การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-485

การสื่อสารข้อมูลตามมาตรฐานที่ได้กล่าวมาข้างต้นคือ RS-232C นั้น เป็นมาตรฐานการสื่อสารข้อมูลในแบบที่ใช้การสื่อสารข้อมูลระหว่างอุปกรณ์ หรือแบบจุดต่อจุด (Point to Point) ส่วน RS-422A นั้นเป็นมาตรฐานที่พัฒนามาจาก RS-232C เพื่อให้สามารถทำการสื่อสารข้อมูลได้ในระยะทางที่ไกลขึ้น และอัตราการสื่อสารข้อมูลเพิ่มขึ้น แต่ก็ยังเป็นการสื่อสารข้อมูลจากอุปกรณ์หนึ่งไปยังอุปกรณ์อื่นๆ ได้สูงสุดเพียงไม่เกิน 10 คู่เท่านั้น และไม่สามารถส่งข้อมูลย้อนกลับจากอุปกรณ์ 10 คู่ นั้นกลับมายังอุปกรณ์หลักได้ หรืออาจกล่าวได้ว่าการสื่อสารข้อมูลตามมาตรฐาน RS-422A นั้นเป็นการสื่อสารข้อมูลแบบซิมเพล็กซ์ คือทิศทางข้อมูลเป็นแบบทางเดียวตลอดเวลา ดังนั้นจึงไม่สามารถทำการออกแบบระบบให้เป็นโครงข่ายข้อมูลได้

ด้วยเหตุนี้จึงได้มีการพัฒนาการสื่อสารแบบใหม่ขึ้น เพื่อรองรับความต้องการดังกล่าวนี้ ซึ่งก็คือ มาตรฐาน RS-485

สำหรับมาตรฐาน RS-485 นั้น จะเป็นมาตรฐานที่อาศัยหลักการทำงานของ สัญญาณแบบดิฟเฟอเรนเชียลเช่นเดียวกับมาตรฐาน RS-422A แต่จะสามารถสื่อสารได้ทั้ง 2 ทิศทางในสายสัญญาณเพียงคู่เดียว ซึ่งเป็นการสื่อสารข้อมูลแบบฮาล์ฟดูเพล็กซ์ จากผลของการใช้สัญญาณในลักษณะดิฟเฟอเรนเชียลนี้ จะทำให้ได้ระยะทางและความเร็วในการสื่อสารข้อมูลสูง เช่นเดียวกับมาตรฐานการสื่อสารข้อมูล RS-422A แต่มาตรฐาน RS-485 นั้นสามารถที่จะสื่อสารระหว่างอุปกรณ์ทั้งการรับและการส่งได้สูงสุดถึง 32 ตัว หรืออาจกล่าวได้ว่า การสื่อสารข้อมูลตามมาตรฐาน RS-485 นั้นเป็นการสื่อสารแบบหลายจุด(Multi-Point Communication) โดยโครงสร้างในการสื่อสารข้อมูลแบบ RS-485 แสดงดังรูปที่ 2.6



รูปที่ 2.6 แสดง โครงสร้างของการสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พารามิเตอร์	<u>RS-232C</u>	<u>RS-423A</u>	<u>RS-422A</u>	<u>RS-485</u>
โหมดการทำงาน	Single-ended	Single-ended	Differential	Differential
จำนวนของตัวรับ และตัวส่งที่ข้อม รับได้	1 ตัวส่ง 1 ตัวรับ	1 ตัวส่ง 10 ตัวรับ	1 ตัวส่ง 10 ตัวรับ	32 ตัวส่ง 32 ตัวรับ
ความยาวของคู่ สายสัญญาณรับ ส่งข้อมูล	50 ฟุต	4000 ฟุต	4000 ฟุต	4000 ฟุต
อัตราการส่งข้อมูล สูงสุด(บิตต่อ วินาที)	20 k	100 k	10 M	10 M
แรงดันไฟฟ้า โหมคร่วมสูงสุด	± 2.5 V	± 6 V	+6 V -2.5 V	12 V -7 V
Driver output	± 5 V ต่ำสุด ± 15 V สูงสุด	± 3.6 V ต่ำสุด ± 6.0 V สูงสุด	± 2 V ต่ำสุด	± 1.5 V ต่ำสุด
Driver load (W)	3 k ถึง 7 k	450 ต่ำสุด	100 ต่ำสุด	60 ต่ำสุด
Driver slew rate	30 V/uS สูงสุด	-	NA	NA
กระแสลิมิต เมื่อเอาต์พุตลัด วงจร	500 mA ลัดวงจรกับ Vcc หรือ Gnd	150 mA ลัดวงจรกับ Gnd	150 mA ลัดวงจรกับ Gnd	150 mA ลัดวงจรกับ Gnd 250 mA ลัดวงจรกับ 8 V หรือ 12 V
ค่าความต้านทาน เอาต์พุตของตัว ส่ง(W)	NA - Power On 300 - Power Off	NA - Power On 60 k - Power Off	NA - Power On 60 k - Power Off	120 k Power On,Off
ค่าความต้านทาน อินพุตของตัวรับ (W)	3 k ถึง 7 k	4 k	4 k	12 k
ความไวตัวรับ	± 3 V	± 200 mV	± 200 mV	± 200 mV

ตารางที่ 2.1 การเปรียบเทียบมาตรฐานการสื่อสารข้อมูลของ EIA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบการสื่อสารข้อมูล	แบบขนาน	แบบอนุกรม
1. ระยะทาง	ปกติจะน้อยกว่า 100 ฟุต	ส่งได้ตั้งแต่ระยะทางสั้นๆ ไปจนถึงระยะทางเป็นไมล์
2. ความเร็ว	อัตราความเร็วสูงมากในระยะที่ไม่ไกลมากนัก กำหนดได้เป็นจำนวนบิตต่อวินาที	อัตราความเร็วของข้อมูลที่ใช้กันอยู่ทั่วไปจะอยู่ในช่วง 0 ถึง 2 ล้านบิตต่อวินาที
3. ระดับของสัญญาณ	ในการอินเตอร์เฟสจะใช้ระดับสัญญาณที่ใช้กับอุปกรณ์ที่ทีแอล(TTL) คือสัญญาณลอจิก 1 และ 0 จะแทนด้วยระดับแรงดัน +5 V และ 0 V ตามลำดับ	ใช้มาตรฐานของ EIA RS-232C คือมีระดับสัญญาณไฟฟ้าขนาด 12 V หรืออาจจะใช้มาตรฐาน 20 mA current loop หรืออาจจะใช้ระดับสัญญาณของ TTL ก็ได้(ใช้กันน้อยมาก)
4. ความผิดพลาดของสัญญาณ	ถ้าส่งในระยะทางไกลๆ ความผิดพลาดของข้อมูลจะเกิดขึ้นง่าย	การผิดพลาดของสัญญาณจะมีน้อยลง
5. ค่าใช้จ่าย	ถ้าส่งในระยะทางไกลๆ จะสิ้นเปลืองค่าใช้จ่ายมาก เพราะต้องใช้สายส่งสัญญาณหลายเส้น	สิ้นเปลืองน้อยกว่าหลายเท่า ถึงแม้ว่าจะใช้อุปกรณ์เปลี่ยนสัญญาณของข้อมูลจากแบบขนานไปเป็นแบบอนุกรมแล้วส่งผ่านสายส่งใช้อุปกรณ์ในการแปลงสัญญาณกลับมาเป็นแบบขนานอีก ก็ยังลงทุนน้อยกว่า

ตารางที่ 2.2 แสดงข้อเปรียบเทียบระหว่างการส่งข้อมูลแบบขนานและแบบอนุกรม

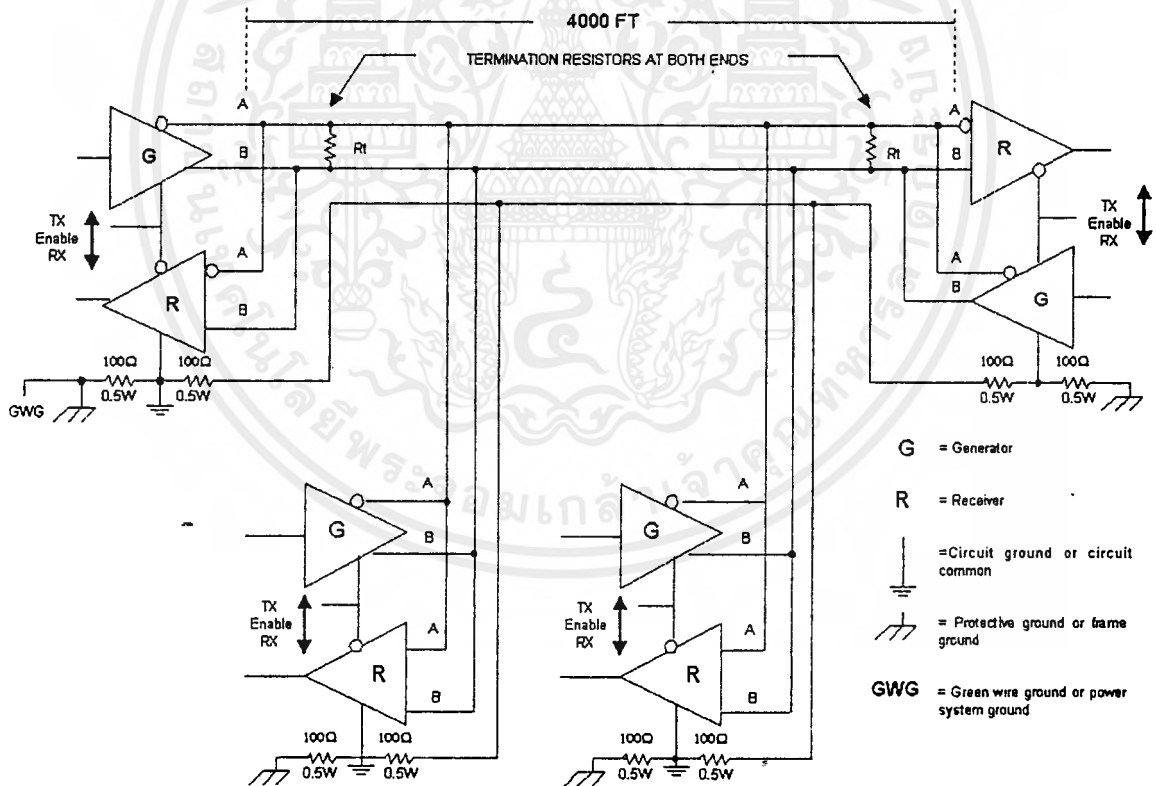
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 รูปแบบของการสื่อสารข้อมูลแบบ RS-485

รูปแบบหรือลักษณะการต่อใช้งานของ RS-485 ในลักษณะของเครือข่าย สามารถแบ่งออกได้เป็น 2 รูปแบบ คือ

2.3.1 เครือข่ายแบบที่ใช้สายนำสัญญาณสองเส้น (Two Wire)

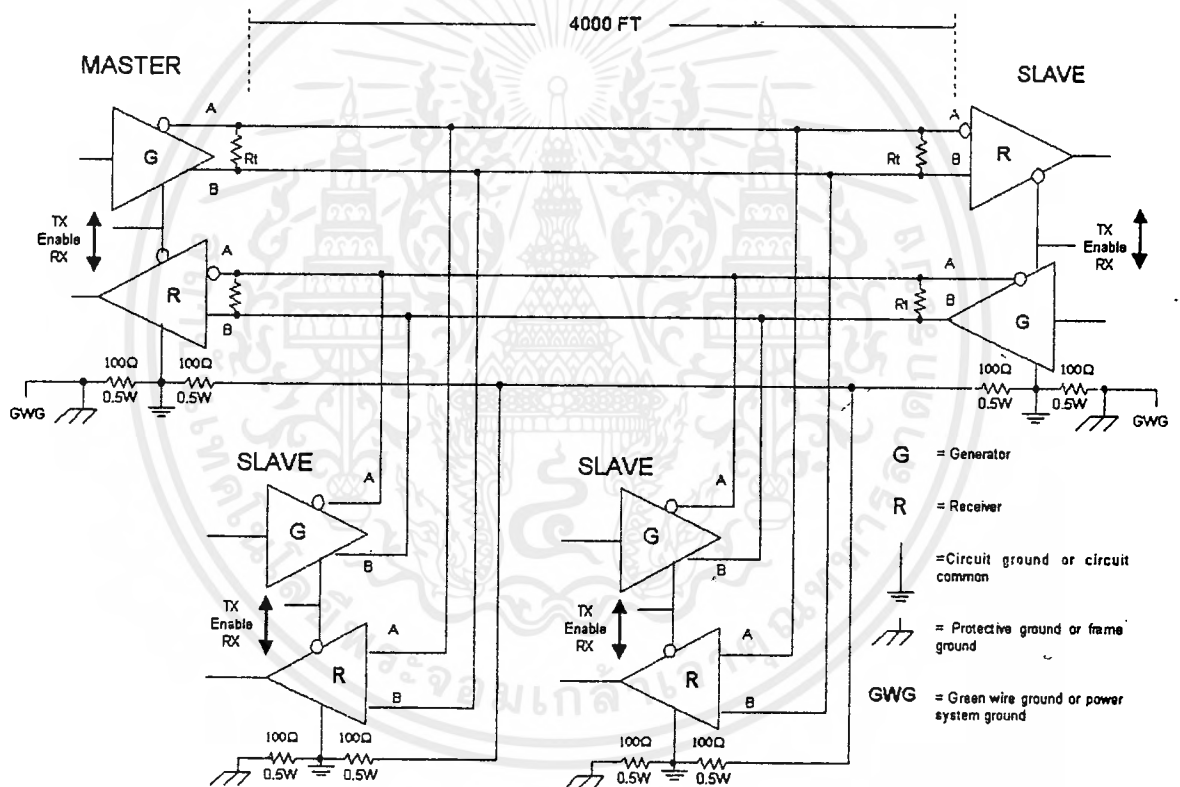
การสื่อสารข้อมูล RS-485 แบบสองสาย นี้ ถือเป็นการสื่อสารแบบฮาล์ฟดูเพล็กซ์ คือ ลักษณะของการรับ-ส่ง ข้อมูล จะเป็นไปในลักษณะผลัดกันรับ-ส่ง โดยการกำหนดว่าจะเป็นการให้เป็นการรับหรือส่ง จะถูกกำหนดด้วยตัวแม่ข่าย(MASTER) ข้อดีของเครือข่ายแบบนี้ คือ จะช่วยประหยัดสายสัญญาณที่ใช้ในการวางเครือข่าย ส่วนข้อเสีย คือ ความเร็วในการรับ-ส่ง ข้อมูลจะช้าลง



รูปที่ 2.7 แสดงเครือข่ายของ RS-485 แบบที่ใช้สายนำสัญญาณสองเส้น(Two Wire)

2.3.2 เครื่องข่ายแบบที่ใช้สายนำสัญญาณสี่เส้น (Four Wire)

การสื่อสารข้อมูล RS-485 แบบที่ใช้สายนำสัญญาณสี่เส้นนี้ ถือเป็นการสื่อสารแบบ ฟูลดูเพล็กซ์ คือ ลักษณะของการรับ-ส่งข้อมูลสามารถทำการรับและส่งข้อมูลได้ในเวลาเดียวกัน ทั้งนี้ เพราะว่ามี บัสข้อมูล(DATA BUS) อยู่ จำนวน 4 เส้น ข้อดีของเครือข่ายแบบนี้ คือ ความเร็วของการรับ-ส่งข้อมูลจะเร็วกว่า แบบที่ใช้สายนำสัญญาณสองเส้น แต่ก็มีข้อเสีย คือ จะสิ้นเปลืองค่าใช้จ่ายในส่วนของการสายนำสัญญาณมากกว่าแบบที่ใช้สายนำสัญญาณสองเส้น



รูปที่ 2.8 แสดงเครือข่ายของ RS-485 แบบที่ใช้สายนำสัญญาณสี่เส้น

2.4 รายละเอียดของการสื่อสารข้อมูลแบบ RS-485

รายละเอียดต่างๆ ของการสื่อสารข้อมูลแบบ RS-485 สามารถกล่าวได้ดังนี้ คือ

2.4.1 คุณสมบัติเฉพาะของตัวส่ง RS-485

สามารถกล่าวได้ดังนี้ คือ

- ตัวส่ง 1 ตัว สามารถขับโหลดได้สูงสุด 32 ชุด โดยที่โหลด 1 ชุด จะประกอบด้วยตัวส่ง 1 ตัว และตัวรับ 1 ตัว และค่าของความต้านทานที่ต่อคร่อมระหว่างสายสัญญาณมีค่า 60 Ω
- เอาต์พุตของตัวส่งในสถานะออฟ มีกระแสรั่วไหลไม่เกิน 100 μA ในช่วงแรงดันไฟฟ้าระหว่าง -7 ถึง 12 V
- เอาต์พุตของตัวส่งให้แรงดันไฟฟ้าเอาต์พุต 1.5 ถึง 5 V ในช่วงแรงดันไฟฟ้าระหว่าง -7 ถึง 12 V
- ตัวส่งมีวงจรป้องกันตัวเองที่ส่วนเอาต์พุต ในกรณีที่ตัวส่งหลายๆ ตัว ส่งข้อมูลออกมาพร้อมๆ กัน

2.4.2 คุณสมบัติเฉพาะของตัวรับ RS-485

- ค่าความต้านทานอินพุตมีค่าสูง โดยมีค่าไม่น้อยกว่า 12 k Ω
- ตัวรับ มีค่าแรงดันไฟฟ้าอินพุต ระหว่าง -7 ถึง 12 V
- ตัวรับ สามารถตอบสนองต่อสัญญาณที่แตกต่างจากสัญญาณโหมคร่วมได้ ± 200 mV

2.4.3 คุณสมบัติเฉพาะของคู่สายสัญญาณ RS-485

- คู่สายสัญญาณที่ใช้สำหรับการรับ-ส่งสัญญาณ ในระบบ RS-485 นั้น ควรพันสลับเป็นเกลียวเพื่อการลดทอนสัญญาณรบกวน

2.4.4 ความหมายของยูนิตโหลด(Unit Load)

ยูนิตโหลด เป็นจำนวนที่มากที่สุดของตัวรับและตัวส่ง ที่สามารถใช้งานบนคู่สายสัญญาณรับ-ส่ง คู่หนึ่ง โดยจะขึ้นกับค่า Unit Load (U.L.) ซึ่ง RS-485 ขอมรับได้ที่ 32 Unit Load ต่อคู่สายสัญญาณ 1 คู่

นิยามของยูนิตโหลด เป็นโหลดที่ใช้กระแส 1 mA ที่แรงดันไฟฟ้าโหมคร่วม 12 V ซึ่ง โหลดนี้ประกอบด้วย ตัวส่ง และ/หรือ ตัวรับ แต่ไม่รวมค่าความต้านทานที่เกิดจากความต้านทานที่ต่อคร่อมคู่สายสัญญาณรับ-ส่ง

2.4.5 คุณสมบัติเฉพาะของคู่ตัวรับ-ส่ง

- เป็นไปตามมาตรฐาน RS-485, RS-422A, CCITT V.11 และ X.27
- เอาท์พุทของตัวส่งเป็นแบบ 3 สเตท (State) ยกเว้น SN 75179B
- เอาท์พุทตัวส่งสามารถขับกระแสได้สูงสุด 60 mA
- ค่าความต้านทานอินพุทของตัวรับมีค่าประมาณ 20 k(น้อยที่สุด)
- ตัวรับมีค่าความไวอินพุท (input sensitivity) ประมาณ ± 200 mV
- ใช้ไฟเลี้ยง 5 V

2.5 การคำนวณหาจำนวนคู่ตัวรับ-ส่ง RS-485

ตัวส่ง

ค่าของกระแสรั่ว เมื่ออยู่ในสภาวะออฟที่ 12 V มีค่าไม่เกิน 0.1 mA
 ตัวส่งมีค่า (U.L.) = $0.1 \text{ mA} / 1.0 \text{ mA} = \text{U.L.}$

ตัวรับ

ค่าของกระแสอินพุทที่แรงดันไฟฟ้าอินพุท 12 V มีค่าไม่เกิน 0.6 mA
 ตัวรับมีค่า (U.L.) = $0.6 \text{ mA} / 1.0 \text{ mA} = 0.6 \text{ mA}$

เมื่อพิจารณาโหลด 1 (ตัวรับและตัวส่ง) มีค่า = $(0.1+0.6)\text{mA} / 1.0 \text{ mA}$
 $= 0.7 \text{ U.L.}$

สายสัญญาณรับ-ส่ง ข้อมูลคู่หนึ่งสามารถรองรับตัวรับ-ส่งได้ = $32 / 0.7 = 45$ คู่

2.6 RS-485 Protocol

RS-485 Protocol เป็นเงื่อนไขของการรับและส่งข้อมูลที่ได้พัฒนาขึ้นมาเพื่อใช้กับระบบการสื่อสารข้อมูลแบบ RS-485 โดยได้ประยุกต์มาจาก DDCMP Protocol, BSC Protocol, HDLC Protocol และ Xebec Protocol เพื่อสร้างชุดของข้อความในการรับ-ส่งข้อมูล และการประยุกต์ โดยมีรายละเอียดดังต่อไปนี้

1 เฟรมของข้อมูลประกอบด้วย HEADER PORTION และ DATA PORTION

SYN	SYN	SYN	HEADER	DATA PORTION	SYN	SYN	SYN
-----	-----	-----	--------	--------------	-----	-----	-----

HEADER PORTION

ในส่วนของ HEADER PORTION ของ RS-485 Protocol - ได้ประยุกต์มาจากบางส่วนของ

DDCMP (Digital Data Communication Message Protocol)

HDLC (High Level Data Link Control)

Xebec Protocol

โดยมีรายละเอียดดังต่อไปนี้คือ

SOH	TYD	EXC	CDAT	FSEQTF	Fserr	ADDRESS	LRC #1
8 bit	2 bit	4 bit	10 bit	8 bit	8 bit	24 bit	8 bit

SOH : Start of Header (8 bit)

เป็นส่วนแสดงการเริ่มต้นของข้อความของ HEADER PORTION มีขนาด 8 บิต

TYD : Type of Data (2 bit)

เป็นส่วนที่ใช้แสดงถึงชนิดของข้อมูลที่บรรจุอยู่ในส่วนของ DATA PORTION

TPD	ลักษณะของข้อความ
00	ข้อความเป็นข้อมูล
01	ข้อความเป็นคำสั่ง
10	ข้อความเป็นคำสั่งและพารามิเตอร์
11	ข้อความเป็นข้อความตอบรับจากตัวตอบรับ

CDAT : Count of Data (10 bit)

เป็นส่วนแสดงจำนวนของข้อมูลที่บรรจุอยู่ใน DATA FIELD ของ DATA PORTION โดยนับข้อมูลแบบไบนารีได้สูงสุด 1 Kbyte/1 เฟรม (สามารถขยายได้ 8 Kbyte/1 เฟรม)

EXC : Expansion of Count (4 bit)

เป็นส่วนแสดงบิตขยายของ CDAT มีขนาด 4 บิต

บิตแรก = 0 แสดงว่า CDAT มีขนาด 10 บิต

= 1 แสดงว่า CDAT มีขนาด 13 บิต (เพิ่มจาก 3 บิต ของ EXC)

FSEQTF : File Sequence Transfer (8 bit)

เป็นส่วนแสดงการรับ-ส่งข้อมูลอย่างต่อเนื่องที่หลายๆ เฟรม มีขนาด 8 บิต ประกอบด้วย

FSQ	NFs	P/F	Nfr
1 bit	3 bit	1 bit	3 bit

FSQ : Frame Sequence bit (1 bit)

0 : ส่งข้อความทีละเฟรม ไม่ต้องพิจารณาอีก บิต ของ FSEQTF และ Fserr

1 : ส่งข้อความแบบทีละหลายๆ เฟรม พิจารณาอีก 7 บิตของ FSEQTF และ Fserr (ส่งต่อเนื่องไม่เกิน 8 เฟรมต่อครั้ง)

NFs : Number of Frame Sequence Begin to Send (3 bit)

แสดงลำดับเฟรมข้อความที่จะส่ง (0-7) มีขนาด 3 บิต

NFr : Number of The Next Frame be Receive (3 bit)

แสดงลำดับเฟรมข้อความต่อไปที่จะรับ (0-7) มีขนาด 3 บิต

P/F : Poll/Final bit (1 bit)

กรณีเซนต์อร์ส่งข้อความให้เทอร์มินอล บิต P/F เป็น P : Poll

P = 1 ยินยอมให้เทอร์มินอลสามารถตอบรับได้

P = 0 ยังไม่ยินยอมให้เทอร์มินอลตอบรับได้

กรณีเซนต์อร์ส่งข้อความให้เทอร์มินอล บิต P/F เป็น F : Final

F = 1 ยินยอมให้เซนต์อร์สามารถตอบรับได้

F = 0 ยังไม่ยินยอมให้เซนต์อร์ตอบรับได้

Fserr : Frame Sequence error

แสดงข้อความที่ผิดพลาด มีขนาด 8 บิต Fserr ประกอบด้วย

F7	F6	F5	F4	F3	F2	F1	F0
1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit

ตัวอย่าง เช่น ถ้าเราส่งข้อมูล 5 เฟรม (0-4) ผล เฟรม 1 และ เฟรม 3 เกิดผิดพลาด

Fserr = 00001010b

ADDRESS : Address of bode (24 bit)

เป็นส่วนแสดงค่าตำแหน่งของตัวรับที่ตัวส่งต้องการติดต่อด้วย มีขนาด 24 บิต

เมื่อตัวรับเป็นเซนเตอร์

ADDRESS = CXX : C = 43 h (ASCII)
: XX = 01-32 (ASCII 2 BYTE)

เมื่อตัวรับเป็นเทอร์มินอล

TERMINAL = TXX : T = 54h (ASCII)

LRC # 1 : Longitudinal Redundancy Check

เป็นส่วนแสดงความผิดพลาดและเป็นรหัสเช็คความผิดพลาดของ HEADER มีขนาด 8 บิต

DATA PORTION

ในส่วนของ DATA PORTION ของ RS-485 Protocol ได้ประยุกต์มาจาก บางส่วนของ

HDLC Protocol : High-Level Data Link Control Protocol

BSC Protocol : Binary Synchronous Communication Protocol

Xebec Protocol.

โดยมีรายละเอียดของ DATA PORTION ดังนี้

STX	DATA FIELD	ETX	LRC # 2
8 bit	1-8 Kbyte	8 bit	8 bit

STX : Start of Text (8 bit)

เป็นแสดงการเริ่มต้นของข้อความส่วน DATA PORTION มีขนาด 8 บิต

ETX : End of Text (8 bit)

เป็นส่วนแสดงการสิ้นสุดของข้อความ ส่วน DATA PORTION มีขนาด 8 บิต

LRC # 2 : Longitudinal Redundacy Check (8 bit)

เป็นส่วนแสดงการตรวจสอบของ DATA PORTION มีขนาด 8 บิต

DATA FIELD (1-8 Kbyte)

เป็นส่วนแสดงข้อมูลที่ใช้ในการติดต่อ มีขนาดไม่เกิน 1024 ตัวอักษรต่อเฟรมข้อความ (ขยายได้ 8192 ตัวอักษร) โดยมีรูปแบบดังนี้

COMMAND	SEP	SDB	DATA	EDB
8 bit	8 bit	8 bit	ASCII CODE	8 bit

COMMAND (8 bit)

คือ ชุดคำสั่งของ RS-485 Protocol

SEP : Seperator (8 bit)

เป็นตัวกั้นระหว่างคำสั่งกับข้อมูลหรือพารามิเตอร์

SDB : Start of Data Block (8 bit)

เป็นส่วนแสดงการเริ่มต้นของข้อมูล หรือพารามิเตอร์

EDB : End of Data Block (8 bit)

เป็นส่วนแสดงการสิ้นสุดของข้อมูลหรือ พารามิเตอร์

๐๗๕๖๖๖

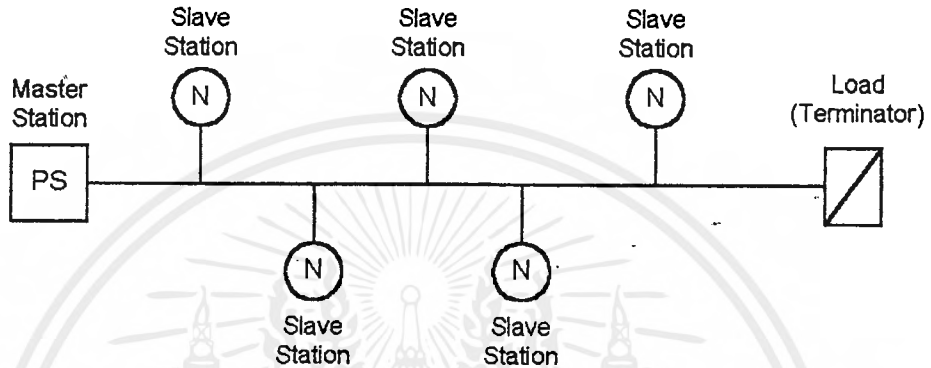
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



DATA : Data or Parameter (ASCII)

เป็นข้อมูล หรือ พารามิเตอร์ รูปแบบรหัส ASCII

2.7 ลักษณะของเครือข่ายแบบบัส(Bus Network)



รูปที่ 2.9 แสดงเครือข่ายแบบบัส

-เทอร์มินเนเตอร์(Terminator) จะทำหน้าที่ดูดกลืนสัญญาณ ไม่ให้เกิดการสะท้อนกลับ ซึ่งจะทำให้เกิดการแทรกสอด (Interference)

ในการขยายวงจรวงจรของเครือข่ายแบบบัสนั้น ทำได้โดยใช้รีพีทเตอร์ (Repeater) ทุก 1 เซกเมนต์(Segment)

ข้อดีของเครือข่ายแบบบัส

1. ง่ายในการเพิ่มโหนด(Node) ใหม่
2. ไม่มีปัญหาการเลือกเส้นทางที่ซับซ้อน
3. เครือข่ายแบบบัสนี้เหมาะกับสถานะที่ทำการรับ-ส่งข้อมูลอย่างหนาแน่น
4. ในการส่งระยะทางไกลๆ สามารถทำได้โดยการเพิ่มบัสรีพีทเตอร์(Bus Repeation)
5. เหมาะกับเทคนิคการเข้าถึงช่องสัญญาณแบบแข่งขัน

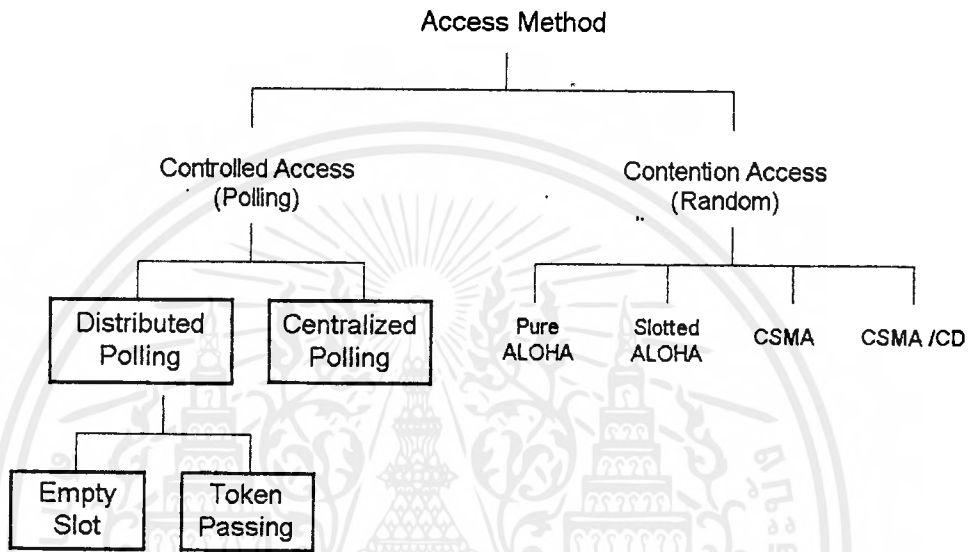
ข้อเสียของเครือข่ายแบบบัส

1. ไม่มีความปลอดภัยในการรับ-ส่งข้อมูล
2. ถ้าไม่มีการควบคุม จะเกิดการชนกันของข้อมูลขึ้น
3. จำเป็นจะต้องมีวงจรมีขยาย(Amplifier) หรือเครื่องทวนสัญญาณRepeater ที่ดี ซึ่งอุปกรณ์เหล่านี้มีราคาแพง

2.8 วิธีการเข้าสู่ช่องนำสัญญาณ

การเข้าสู่ช่องนำสัญญาณ จะมีอยู่ 2 วิธี คือ

1. การเข้าถึงแบบควบคุม Controlled Access หรือ Polling Technic
2. การเข้าถึงแบบแข่งขัน Contention Access หรือ Random Access



รูปที่ 2.10 แสดงรูปแบบของการเข้าถึงข้อมูล

2.8.1 การเข้าถึงข้อมูลแบบควบคุม (Controlled Access)

แบ่งออกได้เป็น 2 วิธี คือ

1. Centralized Polling

- เป็นการควบคุมการเข้าถึงช่องนำสัญญาณ โดยโหนด(Node) ควบคุมศูนย์กลาง หรือ Central Hub
- โหนดแต่ละโหนดในเครือข่าย ต้องคอยรับสัญญาณอนุญาต จากโหนดศูนย์กลาง แล้วส่งข่าวสารภายในเวลาที่อนุญาต แล้วคอยสัญญาณอนุญาตครั้งต่อไป

2. Distributed Polling

วิธีการเข้าถึงแบบนี้ แต่ละโหนดผู้ใช้ (User Node) สามารถบรรจุกลไกภายใน (An internal mechanism) ให้สอดคล้องกับ user node อื่นๆ

- เฟรมควบคุมพิเศษนี้ จะถูกกระจายไปยังแต่ละ user node
- User node ซึ่งรับเฟรมอนุญาตนี้ สามารถส่งข้อมูลของตนเองได้
- โดยวิธีการนี้พูดได้ว่า แต่ละ user node มีการควบคุมท้องถิ่น (local control) และมีความสำคัญเท่ากัน

- มีเพียง user เดียวที่อนุญาตให้ส่งข้อมูลเป็นเวลาหนึ่งเท่านั้น

วิธี Distributing Polling นี้แบ่งออกได้เป็น 2 วิธี คือ

ก. เทคนิค Empty Slot หรือ Slot Ring

ข. เทคนิค Token Passing

2.8.2 การเข้าถึงแบบแข่งขัน (Contention Access)

แบ่งได้ดังต่อไปนี้คือ

1. Pure ALOHA

ก. User node ที่มีข้อมูลจะส่ง จะทำการส่งข้อมูลออกไปด้วย up link frequency

ข. เมื่อ user node ใด ส่งข้อมูลออกไปแล้ว จะเก็บสำเนา (copy) ของข้อมูลนั้น รอจนกว่าจะได้รับสัญญาณที่สะท้อนกลับจากดาวเทียม แล้วตรวจสอบ ถ้าพบ collision จะรอ 1 back off time แล้วส่งข้อมูลเดิมซ้ำอีกครั้ง ถ้าไม่พบ collision จะลบสำเนาของข้อมูลที่ส่งเพื่อการส่งข้อมูลครั้งต่อไป

2. Slot ALOHA

User node ที่ต้องการส่งข้อมูล จะต้องรอจุดเริ่มต้น ของ slot time จึงจะส่งข้อมูลออกไปได้ภายใน 1 Slot time

3. CSMA : Carrier Sense Multiple Access

ก. User node ที่ต้องการส่งข้อมูล จะต้องตรวจสอบสถานะภาพของช่องสัญญาณ และจะส่งได้ต่อเมื่อพบช่องสัญญาณอยู่ในสภาพไอดีล (Idle) หรือไม่พบสัญญาณแคะเรียร์ (Carrier) ของ user อื่น

ข. ถ้าพบช่องสัญญาณอยู่ในสภาพไม่ว่าง (Busy) หรือพบสัญญาณ Carrier ก็จะทำให้การหน่วง(Delay) ตัวเองออกไป 1 back off time แล้วย้อนกลับไปทำตามข้อ ก. และถ้าหากทำการส่งข้อมูลเสร็จแล้ว จะรอรับการตอบสนอง(Response) จากผู้รับ ถ้าเป็นลบ(Negative) จะทำการส่งซ้ำอีกครั้ง



บทที่ 3

หน่วยความจำแบบสัมผัส

(Touch Memory)

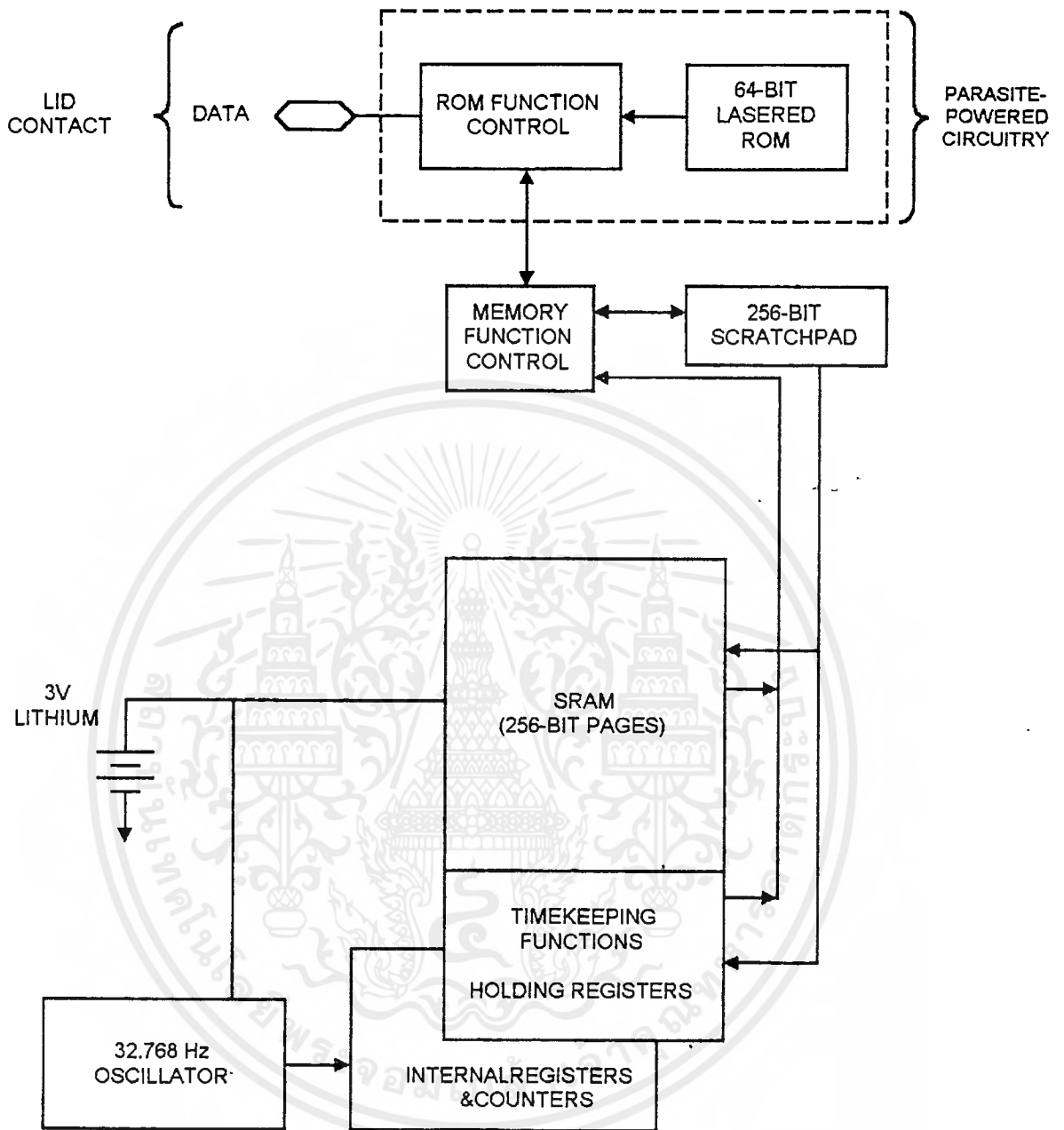
ในโครงงานนี้ได้มีการนำเอา ทักษะโมรี มาทำการประยุกต์ใช้งานร่วมกับการสื่อสารข้อมูล ในระบบ RS-485

3.1 คุณสมบัติของ ทักษะโมรี

ทักษะโมรี ที่ได้นำมาประยุกต์ใช้งานร่วมกับการสื่อสาร RS-485 ในโครงงานนี้ เป็น ทักษะโมรี ของบริษัท DALLAS Semiconductor เบอร์ DS 1993 ซึ่งมีคุณสมบัติต่างๆ ดังนี้ คือ

- เป็นหน่วยความจำที่มีขนาด 4096 บิต หรือ 4 กิโลบิต ในการเขียน/อ่าน ข้อมูล
- สามารถทำงานได้ระหว่างอุณหภูมิ -40 ถึง +70 องศา
- สามารถเก็บบันทึกข้อมูลไว้ได้ นานกว่า 10 ปี
- สามารถตรวจสอบรหัสและข้อมูล ได้โดยการสัมผัสเพียงชั่วขณะ
- จะทำการอ่านข้อมูลเมื่อทำการประทับหรือสัมผัสลงบนตัวอ่านข้อมูล
- เป็นการติดต่อสื่อสารไปยังสายข้อมูลหลัก ที่ประหยัด โดยการใช้สัญญาณดิจิทัลเพียงสัญญาณเดียว ที่ระดับ 16.3 กิโลบิตต่อวินาที ในการติดต่อ
- มี real time/calander อยู่ในรูปแบบของไบนารี

และบล็อกไดอะแกรมแสดงการทำงานของ DS 1993 นี้ สามารถแสดงได้ดังรูปที่ 3.1



รูปที่ 3.1 แสดงบล็อก ไดอะแกรมของทัชเม โมรี DS1993

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การทำงานของทัชเมโมรี่ DS 1993

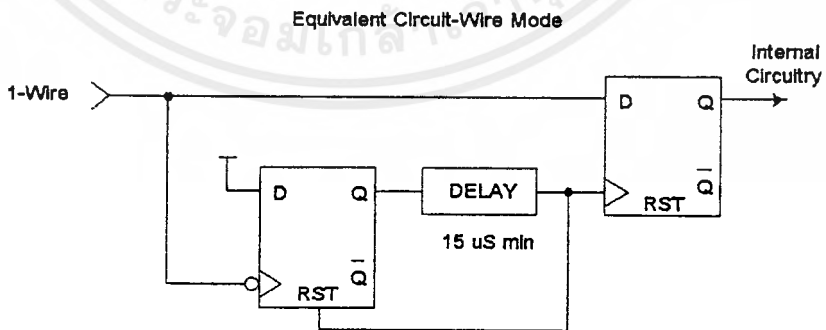
การทำงานของทัชเมโมรี่ DS 1993 นี้ ประกอบด้วยการทำงานในหลายๆ ส่วน ซึ่งสามารถแบ่งออกได้ดังนี้ คือ

3.2.1 การเขียนข้อมูลลงบนทัชเมโมรี่

หลักการการทำงานของการเขียนข้อมูลนั้น การเขียนข้อมูลจะมีข้อแตกต่างกันระหว่างข้อมูล 0 และ 1 จากไทมิ่งในการเขียนข้อมูล(Write Timing) การเขียนข้อมูล 0 ความกว้างของ t_{LOW0} จะต้องทำให้ไทมิ่งของสัญญาณที่เป็นระดับ Low มีค่าประมาณ 60-120 μ S หรือมีค่ามากกว่า t_{LOW1} แต่ไม่เกิน t_{SLOT} ส่วนการเขียนข้อมูล 1 ค่าของ T_{low1} จะมีค่าประมาณตั้งแต่ 1-15 μ S ถ้าไทมิ่งของการเขียนข้อมูล 1 มีค่ามากกว่านี้ จะกลายเป็นการเขียนข้อมูล 0 ไป



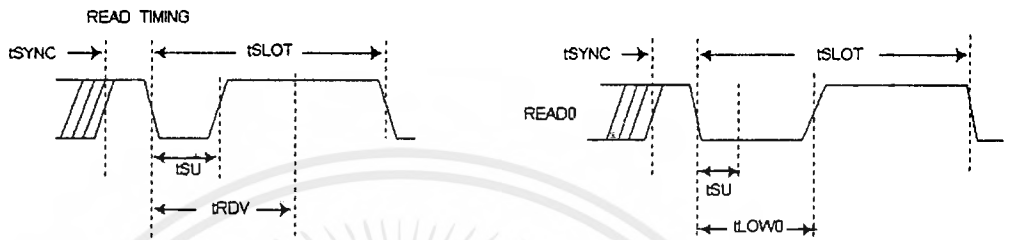
รูปที่ 3.2 แสดงไทมิ่งไดอะแกรม(Timing Diagram) ของการเขียนข้อมูล DS1993



รูปที่ 3.3 แสดงวงจรสมมูลย์ของโหมดการเขียนข้อมูล(Equivalent Circuit-Write Mode)

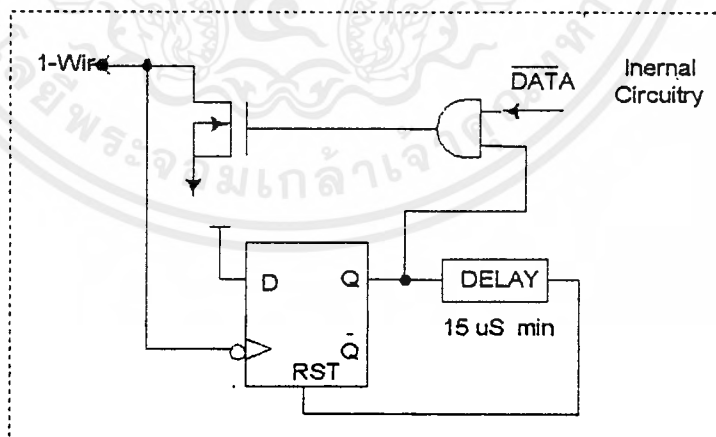
3.2.2 การอ่านข้อมูลจากทัชเมโมรี

การอ่านข้อมูลที่เป็น 0 หรือ 1 วัดได้จากไทม์ความกว้างของ t_{RDV} แล้วนำไทม์มาเปรียบเทียบกับเป็น 0 หรือ 1



รูปที่ 3.4 แสดงไทม์ไดอะแกรมของการอ่านข้อมูลจาก DS1993

Equivalent Circuit-Read Mode

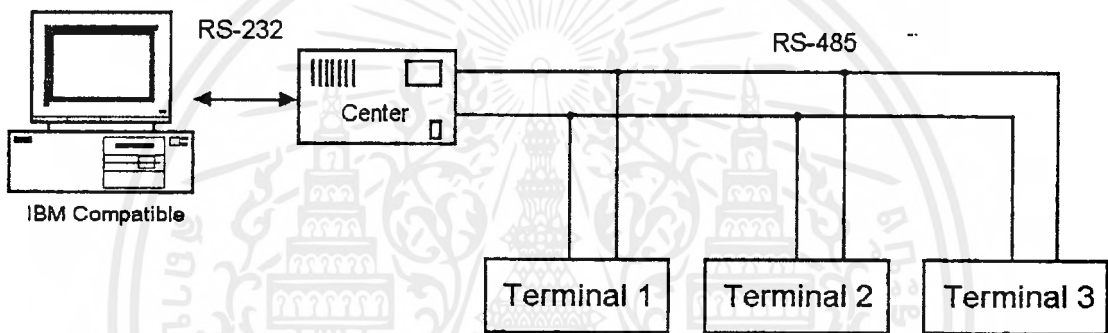


รูปที่ 3.5 แสดงวงจรสมมูลของโหมดการอ่านข้อมูล (Equivalent Circuit Read Mode)

บทที่ 4

โครงสร้างของระบบ

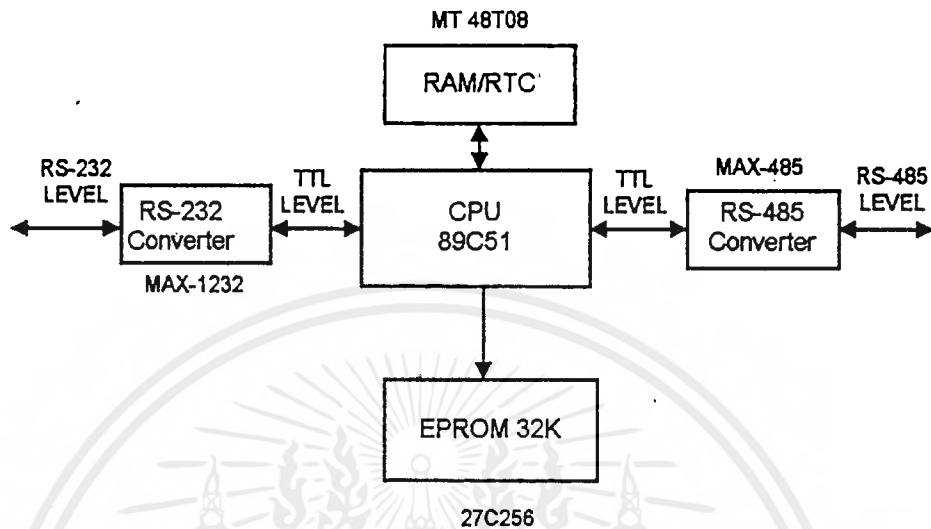
โครงสร้างของระบบที่นำมาประยุกต์ใช้งานกับการสื่อสารด้วยระบบ RS-485 นั้น เป็นการนำไปประยุกต์ใช้งานในการควบคุมการเข้า-ออก ซึ่งในโครงสร้างของระบบจะประกอบด้วย ส่วนควบคุมแม่ข่าย(Master Control) และหน่วยควบคุมลูกข่าย(Slave Control) ต่อเชื่อมกันด้วยระบบ RS-485 โดยใช้ สายข้อมูล(DATA BUS) เพียง 2 เส้น



รูปที่ 4.1 แสดงโครงสร้างของระบบการสื่อสาร RS-485

ในการสื่อสารข้อมูลด้วยระบบ RS-485 จะใช้สายข้อมูล (DATA BUS) เพียง 2 เส้น โดยที่ชุดแม่ข่าย 1 ชุด สามารถใช้เชื่อมต่อกับชุดลูกข่าย ได้ถึง 32 ชุด โดยที่ชุดลูกข่าย แต่ละชุดจะมีตำแหน่งที่อยู่(Address) ของตัวเอง เพื่อเป็นตำแหน่งอ้างอิงในการติดต่อกับชุดแม่ข่าย ได้อย่างถูกต้อง และเนื่องจากในระบบจะมีการใช้สายข้อมูล เพียง 2 เส้น ดังนั้นการติดต่อกับชุดลูกข่าย แต่ละตัวจะต้องทำคนละเวลากัน แต่เนื่องจากการรับ-ส่งข้อมูลของระบบกระทำด้วยความเร็วสูง จึงดูเหมือนว่า ชุดลูกข่าย แต่ละชุดสามารถติดต่อกับชุดแม่ข่าย ได้ในขณะเดียวกัน

4.1 โครงสร้างการทำงานของชุดควบคุมแม่ข่าย (Master Control)



รูปที่ 4.2 แสดงบล็อกไดอะแกรมของชุดควบคุมหลัก

จากรูปจะพบว่า สัญญาณควบคุมที่มาจากคอมพิวเตอร์จะเข้ามาในรูปแบบของสัญญาณ RS-232 ซึ่งระดับแรงดันของสัญญาณจะอยู่ที่ $\pm 12V$ ดังนั้นเพื่อให้สามารถติดต่อกับซีพียู ซึ่งมีระดับสัญญาณเป็นที่ทีแอล (TTL) ได้ จำเป็นจะต้องแปลงสัญญาณก่อน โดยใช้ IC MAX-232 ซึ่งไอซีนี้จะทำหน้าที่ทั้งการแปลงจากระดับแรงดัน $\pm 12V$ เป็นระดับแรงดัน 5 โวลต์ กับแรงดัน ศูนย์ โวลต์ แล้ว ยังแปลงระดับแรงดัน 5 โวลต์ และ 0 โวลต์ เป็นระดับแรงดัน $\pm 12V$ ด้วย เพื่อให้ใช้ในขณะส่งสัญญาณกลับให้กับคอมพิวเตอร์ด้วย

ในส่วนของการส่งข้อมูลด้าน RS-485 จะใช้ ไอซีเบอร์ MAX-485 โดยจะทำหน้าที่เปลี่ยนระหว่างระดับสัญญาณที่ทีแอล เป็นระดับสัญญาณตามมาตรฐาน RS-485

4.1.1 หลักการการทำงานของชุดควบคุมแม่ข่าย (Master Control)

การทำงานของชุดควบคุมแม่ข่ายนั้น สามารถอธิบายได้ดังนี้ คือ

ก.) ซีพียู (CPU)

ซีพียู ที่ใช้จะเป็นเบอร์ 89C51 ซึ่งจะมีหน่วยความจำสำหรับเก็บโปรแกรม (ROM) รวมอยู่ภายในซีพียู เลข ทำให้ไม่ต้องใช้หน่วยความจำในการเก็บโปรแกรมภายนอกซีพียู อีก โดยการทำงานนั้น ซีพียูจะคอยรับคำสั่งที่ส่งมาจากคอมพิวเตอร์ โดยผ่านทาง RS-232 และทำการประมวลคำสั่ง และส่งคำสั่งให้กับชุดลูกข่าย

ข.) RS-232 คอนเวอร์เตอร์ (RS-232 Converter)

จะใช้ไอซีเบอร์ MAX-232 เพื่อทำหน้าที่ในการเปลี่ยนระดับแรงดันจาก $\pm 12V$ เป็นระดับ ทีทีแอล 5 V เพื่อให้เหมาะสมกับสัญญาณที่ซีพียูต้องการ และเมื่อซีพียูต้องการส่งสัญญาณให้กับ คอมพิวเตอร์ MAX-232 ก็จะเปลี่ยนระดับแรงดัน ทีทีแอล 5.V เป็น $\pm 12V$ เพื่อให้เหมาะสมสำหรับการเชื่อมต่อ กับระบบที่ใช้สัญญาณ RS-232

ค.) RS-485 คอนเวอร์เตอร์ (RS-485 Converter)

จะใช้ไอซีเบอร์ MAX-485 ทำหน้าที่ในการเปลี่ยนระดับแรงดันขนาด $\pm 5V$ เป็นระดับสัญญาณทีทีแอล ตามรูปแบบที่ระบบ RS-485 ต้องการ

ง.) หน่วยความจำ(RAM)

หน่วยความจำที่ใช้เก็บข้อมูลในโครงการนี้ เป็นไอซีเบอร์ MT48T08 โดย ไอซีตัวนี้ จะมีหน่วยความจำสำหรับเก็บข้อมูลอยู่ 8 กิโลไบต์ และมีแบตเตอรี่อยู่ในตัว เพื่อเก็บรักษาข้อมูลมิให้สูญหายขณะที่ไม่มีแหล่งจ่ายไฟ

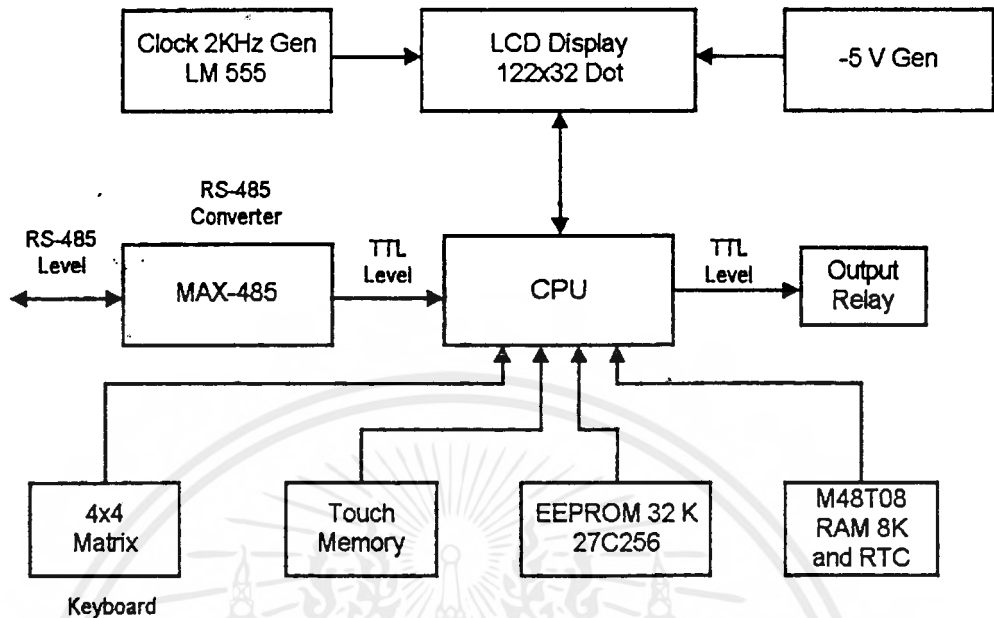
จ.) ระบบฐานเวลาจริง(Real Time Clock)

ระบบฐานเวลาจริง จะอยู่ภายในตัวของ ไอซี MT48T08 โดยการอ่านเขียนข้อมูล จะทำเหมือนกับการอ่านเขียนข้อมูลกับหน่วยความจำทุกประการ

ฉ.) ระบบตรวจสอบการทำงานของ CPU

ระบบตรวจสอบการทำงานของ CPU จะใช้ไอซีเบอร์ MAX-232 ซึ่งจะคอยตรวจสอบการทำงานของซีพียู ว่ามีการทำงานปกติหรือไม่ เมื่อซีพียู เกิดมีปัญหาหรือหยุดการทำงานขึ้นมา ไอซี MAX-232 ก็จะทำหน้าที่รีเซ็ตซีพียู เพื่อทำงานใหม่ รวมทั้งดูแลระบบไฟเลี้ยงของระบบด้วยถ้าแรงดันไฟเลี้ยงต่ำกว่า 4.8 V ก็จะสั่งให้ซีพียู ทำการรีเซ็ตเช่นกัน

4.2 โครงสร้างการทำงานของชุดลูกข่าย



รูปที่ 4.4 แสดงบล็อกไดอะแกรมของชุดลูกข่าย

ในส่วนของชุดลูกข่ายนั้น จะมีลักษณะการทำงานในส่วนของการรับ-ส่งสัญญาณ RS-485 ดังนี้คือ เมื่อมีการส่ง ข้อมูลมาจากชุดแม่ข่าย ในลักษณะของข้อมูลเป็นบล็อกของคำสั่ง ก็จะนำคำสั่งนั้นมาตีความหมายโดยจะเช็คว่าเป็นแอดเดรส(Address) ของตัวเองหรือไม่ ถ้าไม่ใช่แอดเดรสของตัวเองก็จะไม่สนใจคำสั่งบล็อกนั้น แต่ถ้าเป็นแอดเดรส ของตัวเองก็จะทำงานตามคำสั่งนั้น

4.2.1 หลักการทำงานของชุดลูกข่าย

การทำงานของชุดลูกข่ายนั้น สามารถอธิบายได้ดังนี้คือ

ก.) CPU

จะใช้ชิพพียู เบอร์ 89C51 ซึ่งเป็นชิพพียูที่มีหน่วยความจำสำหรับเก็บโปรแกรมอยู่ภายใน โดยสามารถเก็บโปรแกรมได้ขนาด 4 กิโลไบต์

ข.) หน่วยความจำ (Memory : ROM)

หน่วยความจำภายนอกจะใช้ไอซีเบอร์ 93C46 โดยเป็น EEPROM ที่มีหน่วยความจำขนาด 128 ไบต์ โดย EEPROM จะยังสามารถเก็บข้อมูลเอาไว้ได้ ถึงแม้ว่าจะไม่มีไฟ

เสียง และสามารถอ่าน-เขียน ได้ด้วยไฟฟ้า ทำให้ง่ายต่อการใช้งาน

ค.) RS-485 คอนเวอร์เตอร์ (RS-485 Converter)

จะใช้ไอซีเบอร์ MAX-485 ทำหน้าที่ระดับสัญญาณระหว่างทีทีแอล กับ RS-485 เพื่อให้ระบบสามารถติดต่อกันได้

ง.) หน่วยแสดงผล (Display)

ชุดแสดงผลจะใช้ LCD Display Module DMC 162 เป็นตัวแสดงผลบอกสถานะการทำงาน โดยจะแสดงผลได้ในขนาด 2 บรรทัด 16 ตัวอักษร

จ.) หน่วยรับข้อมูลอินพุต (Keyboard Input)

การป้อนคำสั่ง จะป้อนผ่านคีย์บอร์ด เมตริกซ์ (Keyboard Matrix) ขนาด 4x4 โดยจะเป็นการป้อนรหัสผ่านและรหัสควบคุมต่างๆ

ฉ.) หน่วยรับข้อมูลควบคุมอินพุตการเข้า (Input Access Control)

จะใช้ชิพหน่วยความจำแบบสัมผัส (Touch Memory) ซึ่งหน่วยความจำแบบนี้จะเก็บรหัสควบคุมการเข้า-ออกเอาไว้ และสามารถแก้ไขรหัสได้โดยป้อนผ่าน Keyboard และคอมพิวเตอร์

ช.) หน่วยควบคุมดูแลการทำงานของ CPU

จะใช้ไอซีเบอร์ MAX-232 ทำหน้าที่คอยตรวจเช็คการทำงานของชิพยู ถ้าชิพยูหยุดการทำงาน ไอซี MAX 1232 ก็จะทำหน้าที่รีเซ็ตชิพยู เพื่อให้เริ่มต้นการทำงานใหม่

ซ.) หน่วยเอาต์พุต (Output)

หน่วยเอาต์พุตจะเป็นรีเลย์ (Relay) เพื่อใช้ในการควบคุม การเปิด/ปิด อุปกรณ์ไฟฟ้าต่างๆ ตามที่ต้องการ

บทที่ 5

การทดลองและผลการทดลอง

ในกรทำการทดลองโครงการนี้นั้น ได้ทำการทดลองโดยใช้สายนำสัญญาณในการส่งข้อมูล เป็นสายโทรศัพท์(ซึ่งถือว่าเป็นสายนำสัญญาณที่มีคุณภาพค่อนข้างต่ำ) ความยาวประมาณ 20 เมตร

การทดลองจะแบ่งออกเป็น 2 โหมด คือ

1. โหมดการทำงานแบบเครื่องเดียว (Stand Alone Mode)
2. โหมดการทำงานแบบเป็นเครือข่าย (Network Mode)

5.1 การทดลองในโหมดการทำงานแบบเครื่องเดียว (Stand Alone Mode)

เครื่องจะเข้าสู่การทำงานในโหมดนี้เมื่อเปิดเครื่องครั้งแรกแล้ว เครื่องไม่สามารถตีเทค (Detect) เจอสัญญาณที่สื่อสารกันอยู่ในบัส(Bus) ภายในเวลาที่กำหนด เครื่องก็จะปรับตัวเองเข้าสู่การทำงานในโหมดการทำงานแบบเครื่องเดียว (Stand Alone) โดยอัตโนมัติ ซึ่งการทำงานในโหมดนี้ มีขั้นตอนในการทำงานดังต่อไปนี้ คือ

ในสภาวะปกติ เครื่องจะแสดงข้อความภาษาไทยว่า “ ยินดีต้อนรับ ” และจะคอยรับข้อมูลจากอินพุตที่เป็นทัชเมโมรี่ เมื่อไม่มีข้อมูลเข้ามาทางอินพุต ก็จะกลับไปแสดงค่าเวลาต่อ และจะวนลูปอยู่อย่างนี้ตลอดไป แต่หากว่ามีอินพุตจากทัชเมโมรี่ เข้ามา เครื่องก็จะอ่านรหัสจากทัชเมโมรี่ เข้ามาเปรียบเทียบกับข้อมูลในฐานข้อมูล (Data Base) ว่ามีข้อมูลไหนตรงกันบ้าง ซึ่งในฐานข้อมูลจะเก็บรหัสและรายชื่อของนักศึกษาไว้ ถ้าข้อมูลในฐานข้อมูลตรงกันกับข้อมูลที่ผ่านจากทัชเมโมรี่ เครื่องก็จะแสดงรายชื่อของนักศึกษาออกมาที่จอ LCD Graphic และรีเลย์ก็จะเปิด แต่ถ้าวัดไม่ตรงกันเครื่องก็จะส่งสัญญาณแสดงความผิดพลาด (Error) ออกมาที่ LED แล้วกลับมาทำงานในลูปปกติต่อไป

สำหรับข้อมูลที่จะบันทึกลงในฐานข้อมูลนั้น สามารถบันทึกเข้าทางคีย์บอร์ด โดยผ่านคีย์บอร์ด แบบเมตริกซ์ 16 คีย์ (4x4 Matrix Keyboard)

การทดลอง

ขั้นตอนที่ 1

สำหรับการทดลองการทำงานในโหมด Stand Alone นั้น ในขั้นแรกจะต้องบันทึกรหัสของทัชเมโมรี่ ที่จะใช้ลงในหน่วยความจำของเครื่องก่อน โดยจะเข้าสู่โหมดการเพิ่มรหัสนักศึกษาใหม่ และนำทัชเมโมรี่ ไปประกบที่อินพุต ซอกเก็ต (Input Socket) เครื่องก็จะอ่านรหัสจากทัชเม

รี เข้าไปเก็บในหน่วยความจำ และให้เราป้อนรหัสนักศึกษาตามลงไปเป็นขั้นตอนสุดท้าย ก็เป็นอันเสร็จขั้นตอนการเพิ่มรหัสนักศึกษาใหม่



รูปที่ 5.1 แสดงเครื่องเครื่องลูกข่ายในโหมด Stand Alone

ขั้นตอนที่ 2

ขั้นตอนต่อไปคือการทดลองนำทัชเมโมรี ที่ได้ทำการเพิ่มรหัสเข้าไปแล้วนี้ นำไปประกบที่อินพุท ซอกเก็ต เครื่องก็จะอ่านข้อมูลเปรียบเทียบกับข้อมูลในฐานข้อมูล ถ้าข้อมูลตรงกันรีเลย์ก็จะเปิด

ผลการทดลอง

ผลการทดลองที่ได้นั้นจะพบว่าเครื่องสามารถทำงานในโหมด Stand Alone ได้อย่างถูกต้อง โดยเมื่อนำทัชเมโมรีที่ยังไม่ได้มีการบันทึกข้อมูลลงในฐานข้อมูลมาประกบที่อินพุท ซอกเก็ตแล้ว เครื่องก็จะไม่ทำการอ่านข้อมูลจากทัชเมโมรี ทำให้รีเลย์ยังคงอยู่ในสถานะปิดหรือออฟ แต่เมื่อนำทัชเมโมรีที่ได้มีการบันทึกข้อมูลลงในฐานข้อมูลแล้วมาประกบที่อินพุท ซอกเก็ต เครื่องก็จะทำการอ่านข้อมูลจากทัชเมโมรี และแสดงรหัสนักศึกษาที่จอ LCD รีเลย์ก็จะทำงานโดยจะเปลี่ยน

สถานะเป็นเปิดหรือออน

เอกสารนี้เป็นเอกสารที่เผยแพร่เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การทดลองในโหมดการทำงานแบบเป็นเครือข่าย (Network Mode)

เครื่องจะเข้าสู่โหมดการทำงานในแบบเครือข่าย เมื่อเปิดเครื่องแล้วเจอสัญญาณปรากฏในบัส การทำงานในโหมดนี้มีการสื่อสารอยู่ 2 รูปแบบ คือ การสื่อสารข้อมูลระหว่างตัวแม่ข่ายและตัวลูกข่าย และอีกรูปแบบหนึ่งคือ การสื่อสารข้อมูลระหว่างคอมพิวเตอร์กับตัวแม่ข่าย

ลักษณะการทำงานโดยรวมๆ ของการทำงานในแบบเครือข่าย คือ ชุดลูกข่ายจะคอยรับข้อมูลจากอินเทอร์เน็ตเข้ามา ซึ่งเป็นข้อมูลจากทัชเมโมรี่ และจะส่งข้อมูลนี้ไปยังชุดแม่ข่าย และชุดแม่ข่ายก็จะส่งข้อมูลต่อไปยังเครื่องคอมพิวเตอร์ และคอมพิวเตอร์ก็จะมีกระบวนการเปิดฐานข้อมูลค้นหาและเปรียบเทียบข้อมูลที่รับเข้ามากับข้อมูลในฐานข้อมูล และส่งผลของการเปรียบเทียบกลับไปชุดแม่ข่าย และชุดแม่ข่ายก็จะส่งข้อมูลกลับไปชุดลูกข่าย เมื่อชุดลูกข่ายได้รับข้อมูลการเปรียบเทียบก็จะประมวลผลข้อมูลนี้ ถ้าข้อมูลตรงกันก็จะทำให้รีเลย์ทำงานโดยอยู่ในสถานะเปิด และบันทึกสถานะของรีเลย์เอาไว้ สำหรับโปรแกรมหรือซอฟต์แวร์บนคอมพิวเตอร์ก็จะบันทึกเวลาเอาไว้ด้วย เพื่อการประมวลผลในภายหลัง ส่วนการสื่อสารข้อมูลระหว่างชุดแม่ข่าย กับชุดลูกข่ายนั้น จะมีการสื่อสารข้อมูลกันเป็นบัสๆ โดยมีโปรโตคอล คอยควบคุมการรับ-ส่ง

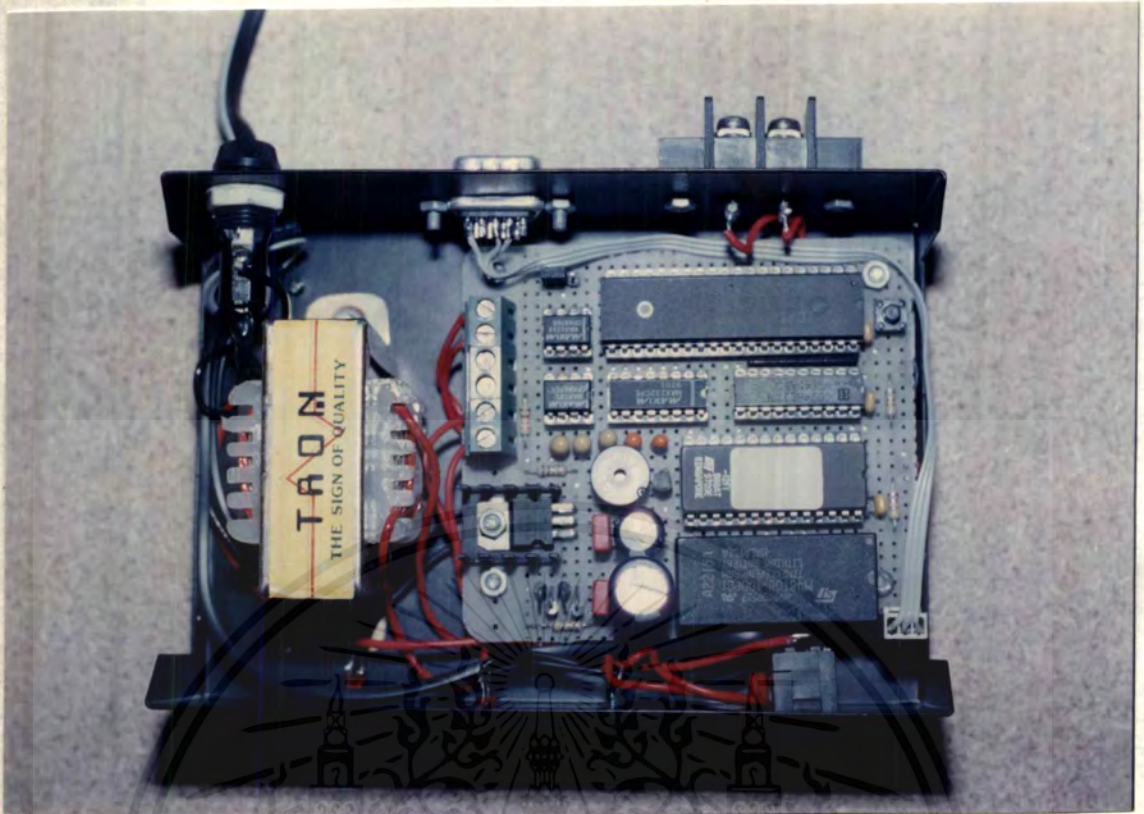
การทดลอง

การทดลองการทำงาน คือนำทัชเมโมรี่ที่ได้มีการโปรแกรมเอาไว้ล่วงหน้าแล้ว มาประกบที่อินเทอร์เน็ต ซอกเก็ต และคอยสังเกตการทำงานของรีเลย์ ถ้ารีเลย์ทำงานหรือออน-ก็แสดงว่า ระบบเครือข่ายทำงานถูกต้องสมบูรณ์

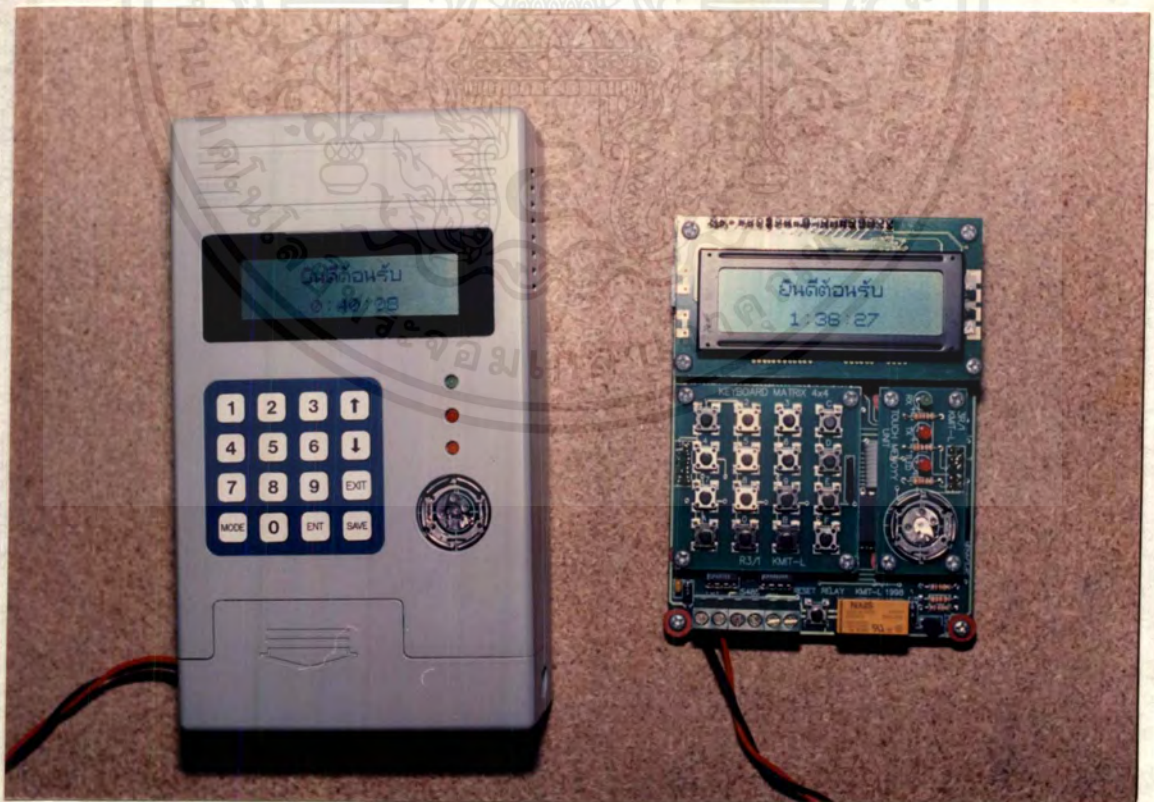
ผลการทดลอง

เมื่อทำการทดลองโดยนำเอาทัชเมโมรี่ที่ได้มีการบันทึกข้อมูลลงในฐานข้อมูลแล้ว มาประกบที่อินเทอร์เน็ต ซอกเก็ต จะพบว่าเครื่องสามารถทำการอ่านข้อมูลจากทัชเมโมรี่ได้ รีเลย์ก็จะทำงานโดยอยู่ในสถานะออน และเมื่อนำเอาทัชเมโมรี่ที่ยังไม่ได้ทำการบันทึกข้อมูลไว้มาประกบที่อินเทอร์เน็ต ซอกเก็ต ปรากฏว่าเครื่องไม่ทำการสั่งให้รีเลย์ออน ทั้งนี้เพราะข้อมูลจากทัชเมโมรี่ไม่ตรงกับข้อมูลในฐานข้อมูล

จากผลการทดลองแสดงว่าเครื่องสามารถทำงานในโหมดเครือข่ายและโหมด Stand Alone ได้อย่างถูกต้องสมบูรณ์ตามที่ได้ทำการออกแบบไว้



รูปที่ 5.2 แสดงเครื่องแม่ข่ายในโหมดเครือข่าย



รูปที่ 5.3 แสดงเครื่องลูกข่ายในโหมดเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปและวิจารณ์

จากการทดลอง จะพบว่าโครงการสามารถทำงานได้ตามที่ได้ออกแบบไว้ คือ สามารถทำการรับ-ส่งข้อมูลได้อย่างถูกต้อง ซึ่งอ้างอิงจากการทดลองโดยใช้สายโทรศัพท์ ความยาวประมาณ 20 เมตร เป็นสายนำสัญญาณ สำหรับระยะทางสูงสุดที่ระบบหรือโครงการที่จะสามารถทำการรับ-ส่งข้อมูลได้นั้นยังไม่ได้มีการทำการทดลองในโครงการนี้ ทั้งนี้เนื่องจากข้อจำกัดในด้านเวลาในการจัดทำโครงการนี้ที่มีจำกัด แต่จากคุณลักษณะของการสื่อสารแบบ RS-485 นั้น สามารถที่จะทำสื่อสารข้อมูลได้ในระยะทางสูงสุดถึง 4000 ฟุต

จากผลการทดลองที่ได้จะพบว่า สามารถที่จะควบคุมอุปกรณ์แอท์ซูทต่างๆ ที่ซูดลูกข่ายได้ โดยการควบคุมผ่านคอมพิวเตอร์ โดยที่การควบคุมนั้นจะไม่มีารผิดเพี้ยน ซึ่งเป็นผลมาจากการออกแบบระบบที่มีความถูกต้องและเชื่อถือได้ในการทำงาน และใช้อุปกรณ์ที่มีคุณภาพ

6.1 ปัญหาที่พบ

ปัญหาที่พบจากการทำโครงการนี้ เกิดขึ้นเนื่องจากชิพยิว ที่ใช้เป็นเบอร์ 89C51 ของซูดแม่ข่าย มีซูดพอร์ทอนุกรม (Serial Port) ให้เพียง 1 ซูด แต่โครงสร้างของระบบนั้น ต้องการซูดพอร์ทอนุกรม ถึง 2 ซูด คือ ซูดที่ใช้ติดต่อระหว่างคอมพิวเตอร์กับซูดแม่ข่าย ที่เป็น RS-232 และซูดที่ติดต่อระหว่างซูดแม่ข่าย กับซูดลูกข่าย ซึ่งเป็น RS-485

และปัญหาอีกประการที่พบในการทดลองก็คือ ปัญหาในการอ่านข้อมูลจากอุปกรณ์ทัชเมโมรี เนื่องจากอุปกรณ์ทัชเมโมรี มีขา Data In, Data Out เป็นขาเดียวกัน ทำให้การเขียนโปรแกรมเพื่ออ่านและเขียนข้อมูลกระทำได้ยาก

6.2 แนวทางการแก้ไขปัญหาและการพัฒนา

การแก้ปัญหาในเรื่องของจำนวนพอร์ทอนุกรม ไม่พอของชิพยิว 89C51 กระทำได้โดยใช้โปรแกรมในการจำลองพอร์ทอนุกรม ขึ้นมา 1 พอร์ท โดยใช้พอร์ทของชิพยิวเป็นตัวจำลองขา Tx และ Rx ของพอร์ทอนุกรม และปัญหาของการอ่าน-เขียนข้อมูลของอุปกรณ์ทัชเมโมรี สามารถทำได้โดยการพยายามปรับเปลี่ยนซอร์ฟแวร์ โดยเปลี่ยน ไทมมิง (Timing) ในการอ่าน-เขียนข้อมูลไปเรื่อยๆ จนโปรแกรมมีความแน่นอนในการอ่านเขียนข้อมูล

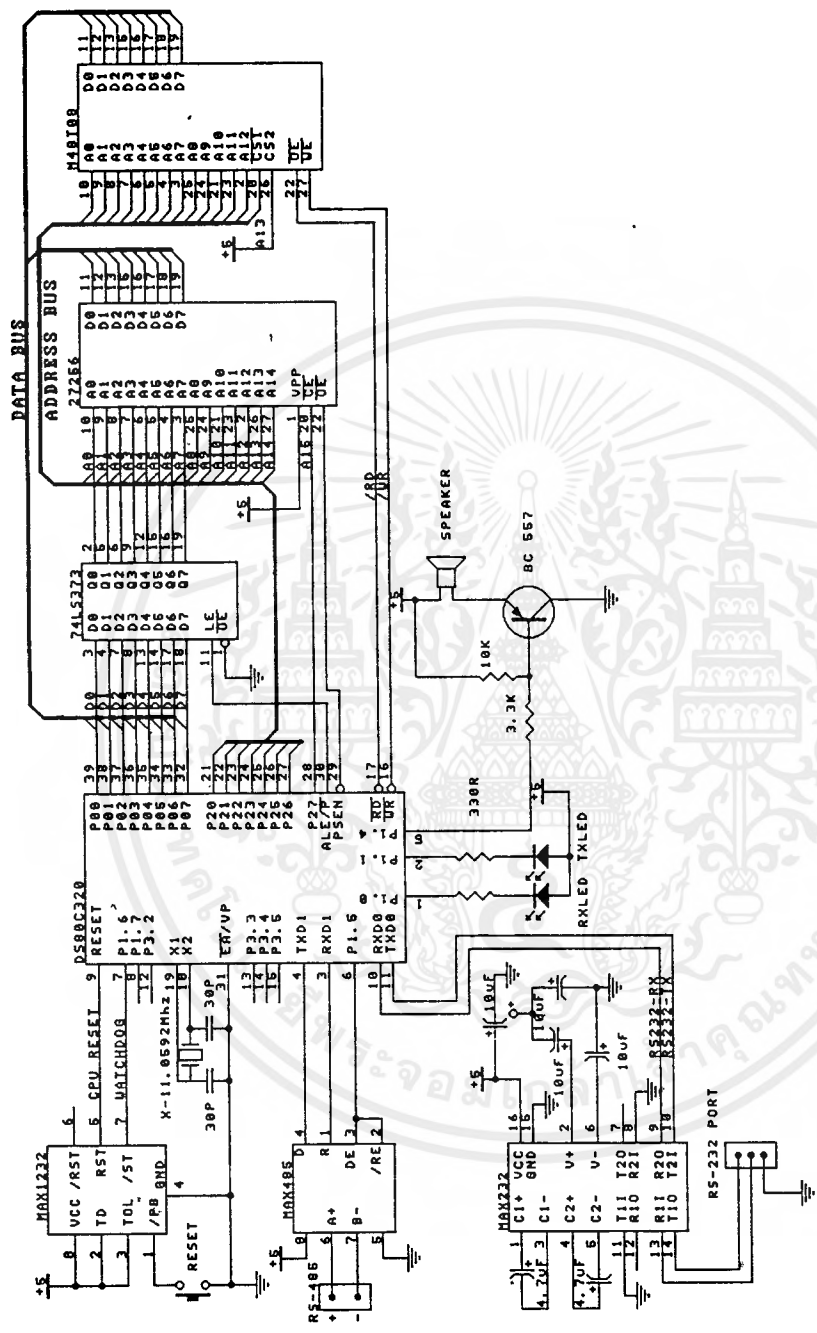
ส่วนแนวทางในการพัฒนานั้น จะเป็นการพัฒนาโปรแกรมให้มีความสามารถเพิ่มขึ้น และ ซอร์ฟแวร์ สำหรับการสั่งงานควบคุมผ่านทางคอมพิวเตอร์ นั้นสามารถที่จะพัฒนาให้เป็น ซอร์ฟแวร์ ที่ทำงานภายใต้ระบบปฏิบัติการวินโดวส์ (Windows) โดยที่ซอร์ฟแวร์ ที่ใช้ในโครงการ ขณะนี้เป็น ซอร์ฟแวร์ ที่ทำงานภายใต้ระบบปฏิบัติการดอส (DOS)



ภาคผนวก

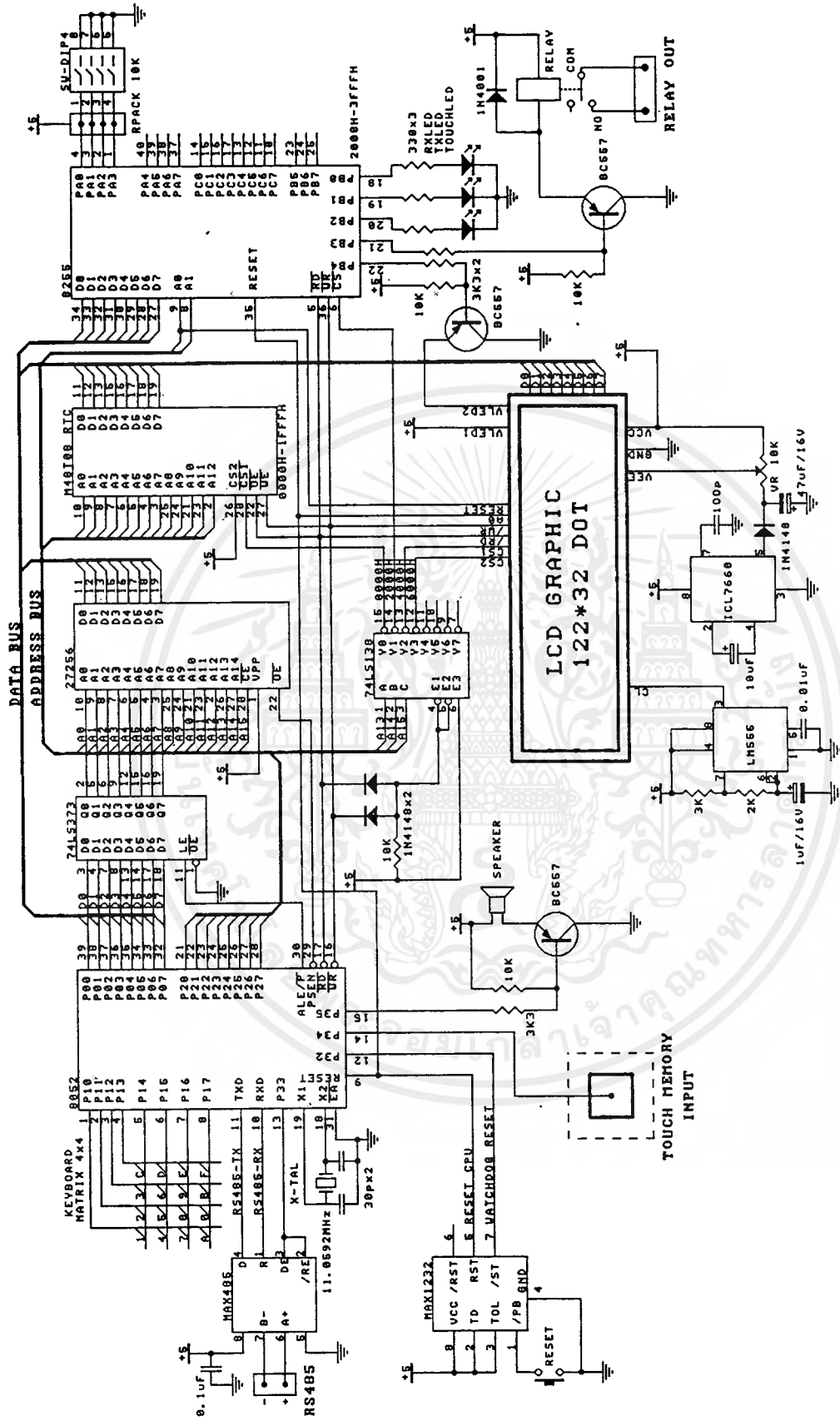


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



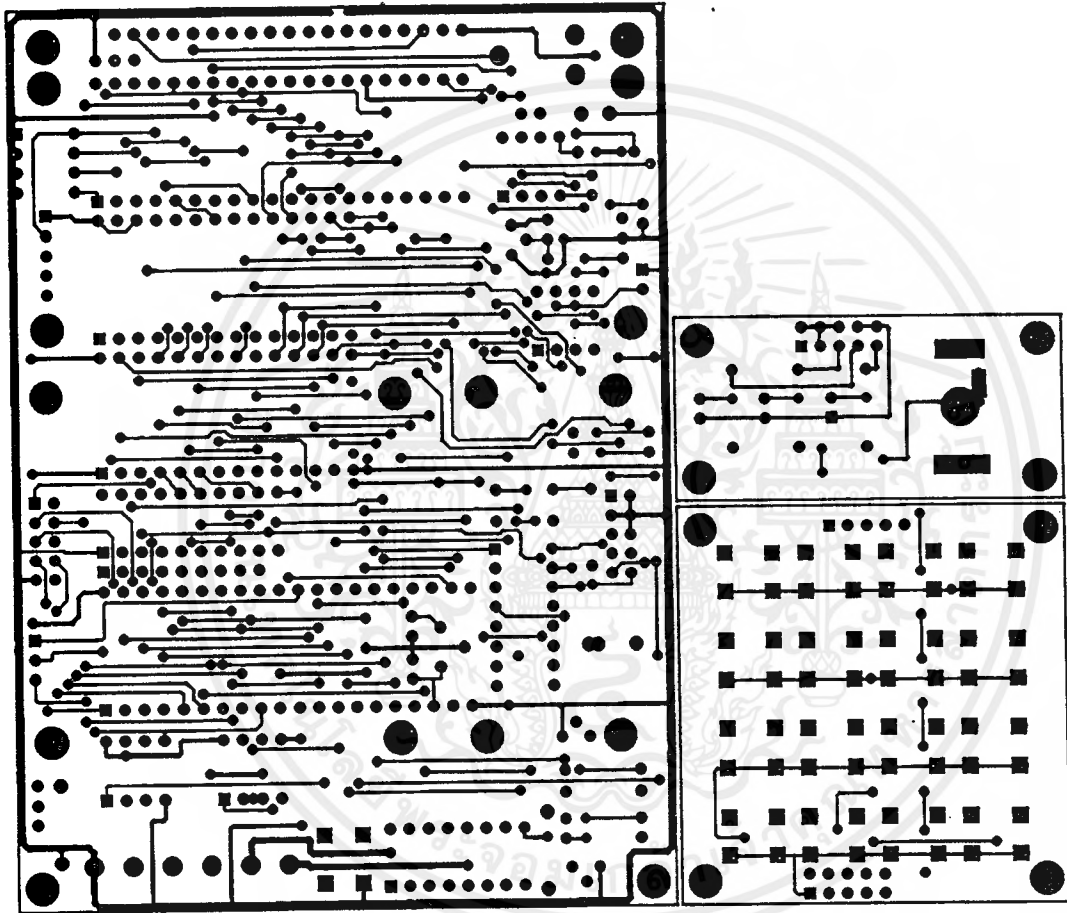
รูปแสดงวงจรของชุดแม่ข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีคุณนำไปใช้



รูปแสดงวงจรของชุดฝึกข่าบ

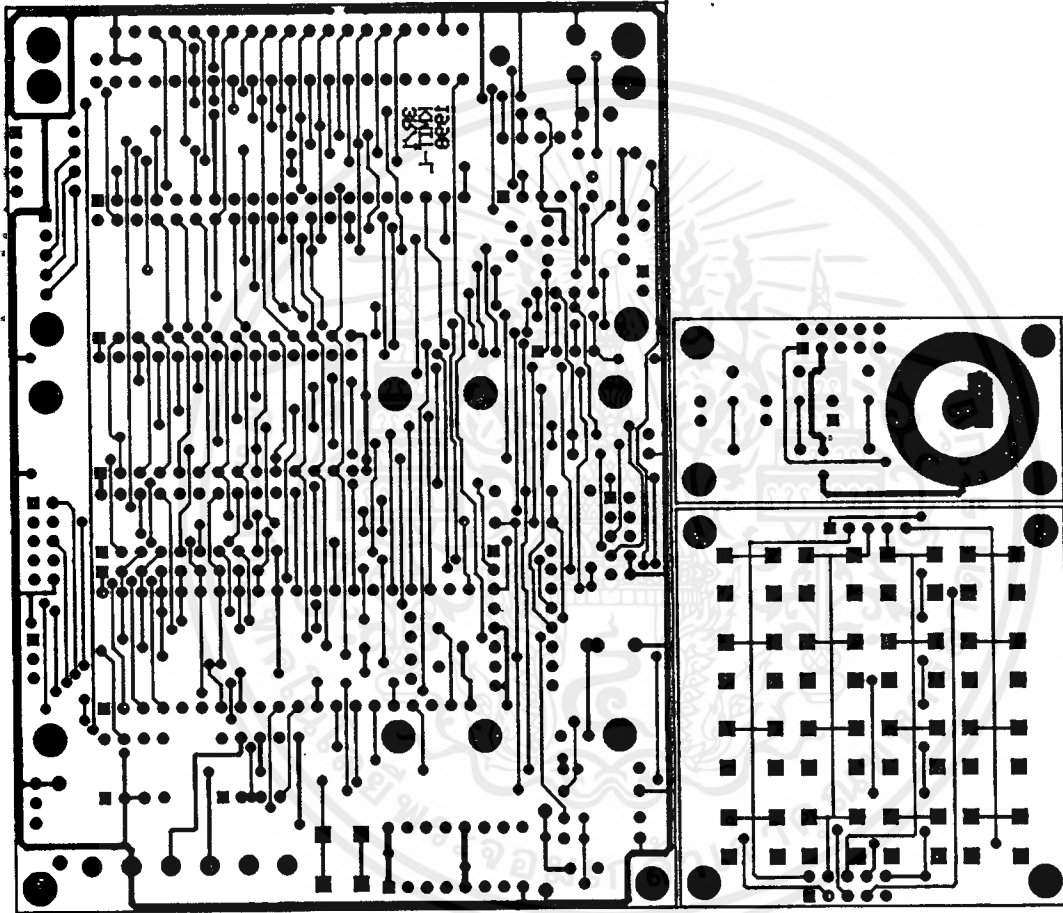
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่สามารถใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



PRJ3R_2S Top Layer

รูปแสดงแผ่นวงจรพิมพ์ของชุดลูกข่าย(ด้านบน)

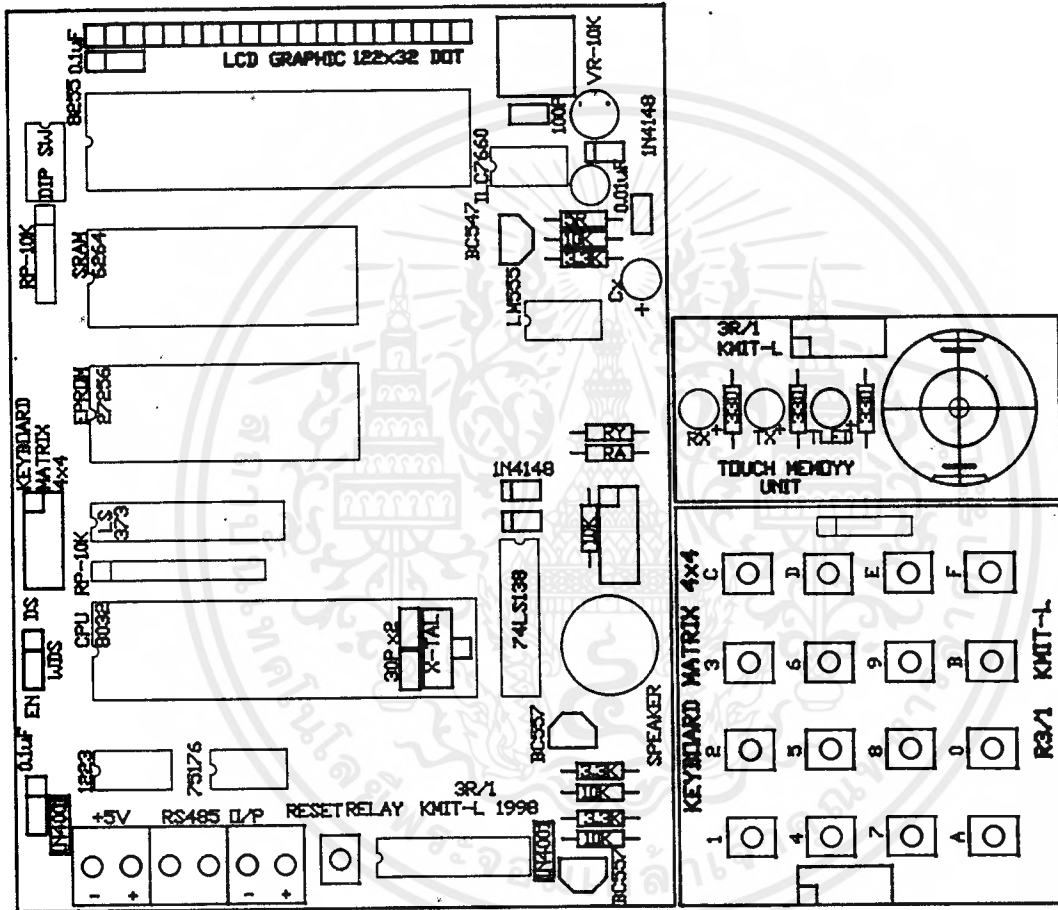
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



PRJ3R_2S Botton Layer

รูปแสดงแผ่นวงจรพิมพ์ของชุดลูกข่าย(ด้านล่าง)

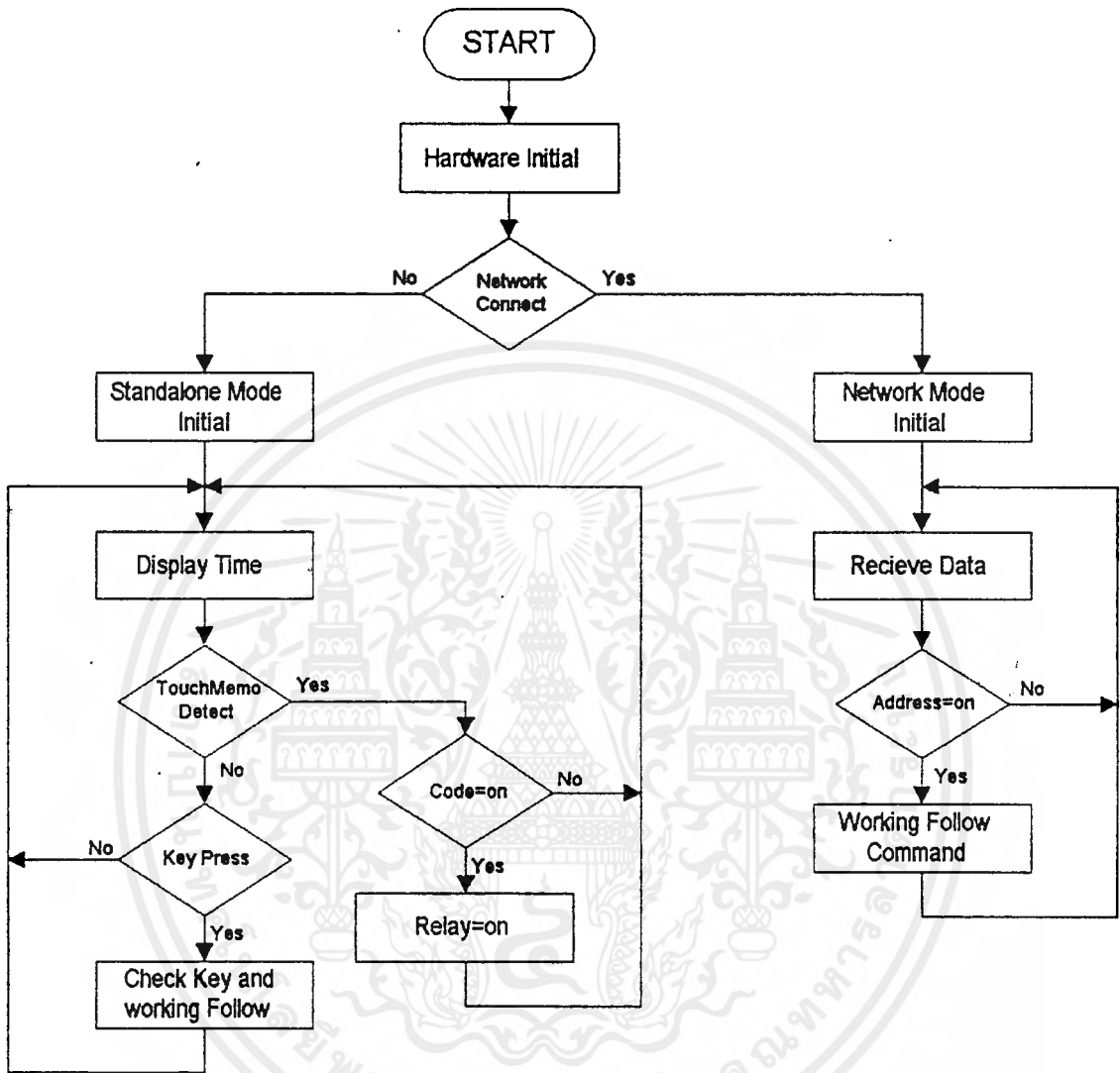
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



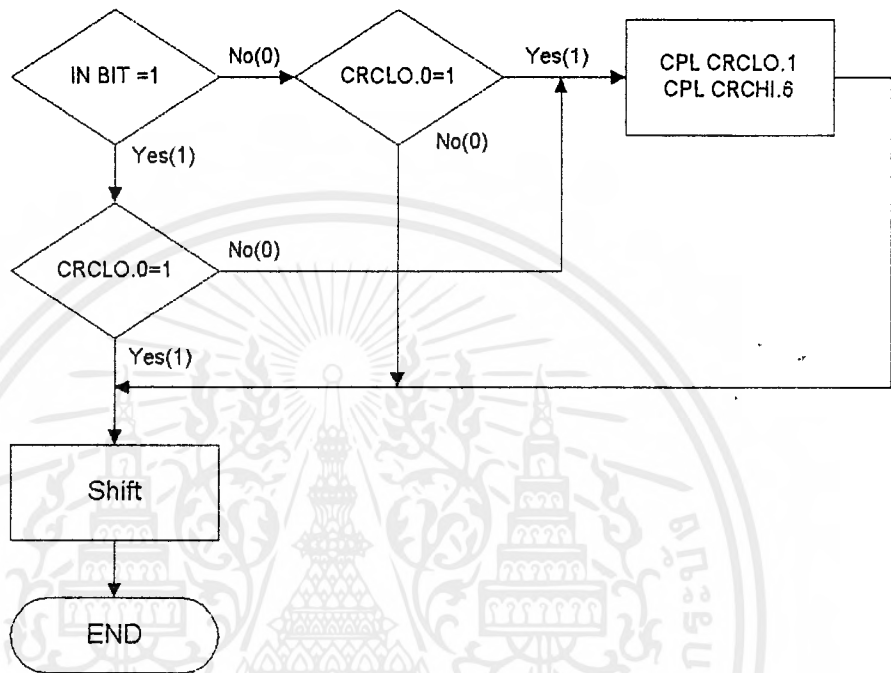
PRJ3R_2S Check Plot

รูปแสดงการวางอุปกรณ์บนแผ่นวงจรพิมพ์ของชุดลูกข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงโฟลว์ชาร์ตของโปรแกรมหลัก



รูปแสดงโฟลว์ชาร์ตโปรแกรมเช็ค CRC 16 บิต

```

+***** REMOTE MONITOR FOR NETWORK SYSTEM *****
" | KMITL PROJECT 3R/1
" | 38013177 KADIT ONCHUM
" | 38013180 CHAIYOOT HONGSOPHA
" | 38013193 PRAPOT WONGTAM
" | -----

```

```

bit MSECINT;
UCHAR CRCLO;
UCHAR CRCHI;

```

```

void main(void)

```

```

{
    RX = &Receive;
    TX = &Transmit;
    SCNO = 0x52;
    SCNO1 = 0x52;
    TMOD = 0x20;
    TH1 = 0xe8;
    TR1 = 1;
    TMOD = TMOD | 0x01;
    TL0 = TIMER0_LOW;
    TH0 = TIMER0_HIGH;
    EA = 1;
    Beep(26,400);
    while(1)
    {
        SendClock(1,1);
        Delay(200);
        SendClock(2,1);
        Delay(600);
    }
}

```

```

void SendClock(char addr,char dcom)

```

```

{
    TX->DAaddress = addr;
    TX->Command = dcom;
    TX->Datalength = 8;
    ReadTime();
    Protocaldata[0] = 0x00;
    Protocaldata[1] = Clock.hour;
    Protocaldata[2] = Clock.min;
    Protocaldata[3] = Clock.sec;
    Protocaldata[4] = Clock.date;
    Protocaldata[5] = Clock.month;
    Protocaldata[6] = Clock.year;
    Protocaldata[7] = Clock.day;
    Txprotocal();
}

```

```

void Delay(unsigned int delaytime)

```

```

{
    bit msecintf;
    TL0 = TIMER0_LOW;
    TH0 = TIMER0_HIGH;
    ET0 = 1;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TRO = 1;
msecintf = MSECINT;
while(delaytime)
{
    while(msecintf == MSECINT);
    delaytime--;
    msecintf = MSECINT;
}
ETO = 0;
TRO = 0;
}

timer0() interrupt 1 using 0
{
    EA = 0;
    MSECINT = ~MSECINT;
    TLO = TIMERO_LOW;
    TH0 = TIMERO_HIGH;
    EA = 1;
}

void Beep(char freq,int time)
{
    int i;
    for(i=0;i<time;i++)
    {
        SOUNDBIT = 0;
        BeepDelay(freq);
        SOUNDBIT = 1;
        BeepDelay(freq);
    }
}

void BeepDelay(int del)
{
    while(del)
        del--;
}

void ReadTime(void)
{
    unsigned char xdata *rtcptr;
    rtcptr = 0x1ff8; *rtcptr = 0x40; rtcptr++;
    Clock.sec = *rtcptr; rtcptr++;
    Clock.min = *rtcptr; rtcptr++;
    Clock.hour = *rtcptr; rtcptr++;
    Clock.day = *rtcptr; rtcptr++;
    Clock.date = *rtcptr; rtcptr++;
    Clock.month = *rtcptr; rtcptr++;
    Clock.year = *rtcptr;
    rtcptr = 0x1ff8; *rtcptr = 0x00;
}

void WriteTime(void)
{
    unsigned char xdata *rtcptr;
    rtcptr = 0x1ff8; *rtcptr = 0x80; rtcptr++;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <reg52.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <intrins.h>
#include "thaifont.h"
#include "engfont.h"
#include "student.h"

#define XBYTE ((unsigned char volatile xdata *)0)

#define PORTA XBYTE [0X2000]
#define PORTB XBYTE [0X2001]
#define PORTC XBYTE [0X2002]
#define CPORT XBYTE [0X2003]
#define CS1_I XBYTE [0X4000]
#define CS1_D XBYTE [0X4001]
#define CS2_I XBYTE [0X6000]
#define CS2_D XBYTE [0X6001]
#define IDINDEX XBYTE [0x1998]
#define TXLED_ON (PORTB &= 0XFE)
#define TXLED_OFF (PORTB |= 0X01)
#define RXLED_ON (PORTB &= 0XFD)
#define RXLED_OFF (PORTB |= 0X02)
#define TMRLED_ON (PORTB &= 0XFB)
#define TMRLED_OFF (PORTB |= 0X04)
#define RELAY_ON (PORTB &= 0XF7)
#define RELAY_OFF (PORTB |= 0X08)
#define BLK_OFF (PORTB &= 0XEF)
#define BLK_ON (PORTB |= 0X1F)

#define TIMER0_HIGH 0xFC
#define TIMER0_LOW 0x66
#define UCHAR unsigned char
#define UINT unsigned int
#define ULONG unsigned long
#define RESET 0XE2
#define DISP_ON 0XAf
#define PAGE_0 0XB8
#define COL_0 0X00
#define ON 0X00
#define OFF 0X01
#define SET 0X01
#define CLR 0X00

```

```

sbit MAX485 = P3^3; sbit WDS = P3^2;
sbit TOUCH = P3^4; sbit SOUNDBIT = P3^5;
sbit COL0 = P1^0; sbit ROW0 = P1^4;
sbit COL1 = P1^1; sbit ROW1 = P1^5;
sbit COL2 = P1^2; sbit ROW2 = P1^6;
sbit COL3 = P1^3; sbit ROW3 = P1^7;

```

```

typedef struct QUEUE
{
    UCHAR SAaddress;
    UCHAR DAaddress;
    UCHAR Command;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    UCHAR Datalength;
    UCHAR Dataptr;
    UCHAR CRC16LO;
    UCHAR CRC16HI;
    UCHAR head;
    UCHAR tail;
    UCHAR Queue[0xff];
    }QUEUE;

xdata    QUEUE Transmit,Receive;
data     QUEUE *RX,*TX;

typedef struct RTC
{
    UCHAR sec;
    UCHAR min;
    UCHAR hour;
    UCHAR day;
    UCHAR date;
    UCHAR month;
    UCHAR year;
}RTC;
idata    RTC Clock;

typedef struct StdBuffer
{
    char StdID[9];
    char StdKeyCode[17];
} StdBuffer;
StdBuffer xdata StdBuff[50];

/*-----*/
bit    RXprotocal(void);
bit    receive_c(UCHAR *c);
void    TXprotocal(void);
void    Beep(char freq,int time);
void    DelBeep(int del);
void    Delay(unsigned int delaytime);
void    ReadTime(void);
void    WriteTime(void);
void    Lcd1wd(unsigned char dat);
void    Lcd1wi(unsigned char com);
void    Lcd1Busy(void);
void    Lcd2wd(unsigned char dat);
void    Lcd2wi(unsigned char com);
void    Lcd2Busy(void);
void    LcdBit(char x,char y,bit b);
void    LcdByte(char x,char y,unsigned char c);
void    LcdChar(char x,char y,unsigned char c);
void    LcdThaiChar(char x,char y,unsigned char c);
void    LcdThaiString(char x,char y,char *cp);
void    LcdString(char x,char y,char *cp);
void    LcdUpdate(void);
void    LcdSet(void);
void    LcdClr(void);
void    ClrThaiBuf(void);
void    Network(void);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void Standalone(void);
void TimeDisp(void);
void ClearLcdbuf(void);
void TimeEdit(void);
void SetID(void);
void EditID(void);
char GetKeyCode(void);
void TouchCnv(void);
void Tx485Char(UCHAR tx485);
char StandaloneMenu(char menu, char maxmenu);

```

```
char Getkey(void);
```

```
extern void CalcCRC16(UCHAR crcdata);
extern void Tmrrom(void);
```

```
/*-----*/
```

```

bit Success;
bit MSECINT;
UCHAR Rcvchar;
UCHAR Myaddress;
UCHAR CRC8;
UCHAR CRCLO;
UCHAR CRCHI;
UCHAR TEMPX;
UCHAR OLDCHAR;
UCHAR xdata LcdBuf[488];
UCHAR xdata Protocaldata[48];
UCHAR xdata ThaiBuf[16];
UCHAR xdata PrintBuf[15];
UCHAR xdata Codedata[9];
UCHAR xdata TouchBuf[8];
char xdata TmrStrbuf[17];
char xdata TmrStrbuf1[9];
char xdata TmrStrbuf2[9];

code UCHAR *S_Mode[] = {
    {"çïŒïËŒŒ ũ/ç/£"},
    {"ÀŒçïËŒŒ ÁúéŒ"},
    {"çïŒïúúíŒŒ;íÀàŒÀèŒŒúéŒŒ"},
    {"éŒúúíŒŒ;íÀàŒÀèŒŒ"},
    ""
};

```

```

void main (void)
{
    long int i;
    char mode;
    RX = &Receive;
    TX = &Transmit;
    RX->head = RX->tail = 0;
    TX->head = TX->tail = 0;
    SCON = 0x52;
    TMOD = 0x20;
    TH1 = 0xe8;
    TR1 = 1;
    TMOD |= 0x01;
    TL0 = TIMER0_LOW;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TH0    = TIMER0_HIGH;
PS     = 1;
EA     = 1;
LcdSet();LcdClr();
CPORT = 0x90;PORTB = 0xff;
Myaddress = PORTA;Myaddress &= 0x0f;
Beep(21,500);
MAX485 = 0;
while(1)
{
    mode = 0;
    for(i=0;i<20000;i++)
    {
        if(RI)
        {
            mode = 1;
            break;
        }
    }
    switch(mode)
    {
        case 0:Standalone();break;
        case 1:Network();break;
    }
}

void Network(void)
{
    LcdClr();
    LcdString(20,10,"NETWORKING");
    LcdUpdate();
    Beep(21,1000);
    ES = 1;
    MAX485 = 0;
    ClearLcdbuf();
    while(1)
    {
        while(!RXprotocal());
        {
            if ((CRCLO == RX->CRC16LO) && (CRCHI == RX->CRC16HI))
            {
                sprintf(PrintBuf, "%2bx:%02bx:%02bx",Protocaldata[0],
                Protocaldata[2],Protocaldata[3]);
                LcdThaiString(20,1,PrintBuf);
                sprintf(PrintBuf, "%02bx/%02bx/%02bx",Protocaldata[4],
                Protocaldata[5],Protocaldata[6]);
                LcdThaiString(20,15,PrintBuf);
                LcdUpdate();
            }
        }
    }
}

void Standalone(void)
{
    char getmenu,key,i;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    UCHAR oldsec;
    ClearLcdbuf();
    LcdString(20,10,"STAND ALONE");
    LcdUpdate();
    Beep(21,1000);
    Delay(1500);
    ClearLcdbuf();
    while(1)
    {
        ReadTime();
        if(oldsec != Clock.sec)
        {
            sprintf(PrintBuf, "%2bx:%02bx:%02bx", Clock.hour,
                Clock.min,Clock.sec);
            LcdThaiString(30,0,"๐๑;ÇíçÕě;ûîπ");
            LcdThaiString(20,16,PrintBuf);
            LcdUpdate();
            oldsec = Clock.sec;
        }
        Tmrrom();
        TouchCnv();
        if(CRC8 == TouchBuf[7])
        {
            for(i=0;i<39;i++)
            {
                key = strcmp(StdBuff[i].StdKeyCode,TmrStrbuf);
                if(key == 0)
                {
                    TMRLED_ON;
                    RELAY_ON;
                    Beep(21,2500);
                    RELAY_OFF;
                    TMRLED_OFF;
                    break;
                }
            }
        }
        key = Getkey();
        if(key == 'A')
        {
            getmenu = StandAloneMenu(0,3);
            switch(getmenu)
            {
                case 0:
                {
                    TimeEdit();
                    ClearLcdbuf();
                    break;
                }
                case 1:
                {
                    TimeDisp();
                    ClearLcdbuf();
                    break;
                }
                case 2:
                {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        SetID();
        ClearLcdbuf();
        break;
    }
    case 3:
    {
        EditID();
        ClearLcdbuf();
        break;
    }
}
}
}

void TimeEdit(void)
{
    ClearLcdbuf();
    LcdString(0,3," This Function ");
    LcdString(0,16," not Avilable");
    LcdUpdate();
    while(Getkey() == ' ');
}

void SetID(void)
{
    int CalcID;
    char keycodeok;
    UCHAR ForCalID[5];
    keycodeok = GetKeyCode();
    if(!keycodeok) return;
    ForCalID[0] = Codedata[4];
    ForCalID[1] = Codedata[5];
    ForCalID[2] = Codedata[6];
    ForCalID[3] = Codedata[7];
    ForCalID[4] = '\0';
    CalcID = atoi(ForCalID);
    CalcID -= 3176;
    ClearLcdbuf();
    LcdThaiString(0,0,StudentName[CalcID*2]);
    LcdThaiString(0,16,StudentName[(CalcID*2)+1]);
    LcdUpdate();
    strcpy(StdBuff[CalcID].StdID,Codedata);
    while(Getkey() == ' ');
    ClearLcdbuf();
    LcdThaiString(0,0," ÀÙÀ°Ø£Õë ");
    LcdThaiString(0,16," SECURITY CODE");
    LcdUpdate();
    do
    {
        TMRLED_ON;
        Tmrrom();
        TMRLED_OFF;
    }while(CRC8 != TouchBuf[7]);
    TMRLED_ON;
    Beep(21,400);
    ClearLcdbuf();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

    }
}

j = TouchBuf[i];
j &= 0x0f;
if((0 <= j) && (j <= 9))
{
    j |= 0x30;
    TmrStrbuf1[(i*2)+1] = j;
}
else
{
    if((0x0a <= j) && (j <= 0x0f))
    {
        j -= 9;
        j |= 0x60;
        TmrStrbuf1[(i*2)+1] = j;
    }
}

}

TmrStrbuf1[8] = '\0';
for(i=0;i<4;i++)
{
    j = TouchBuf[i+4];
    j &= 0xf0;
    j = j >> 4;
    if((0x00 <= j) && (j <= 0x09))
    {
        j |= 0x30;
        TmrStrbuf2[i*2] = j;
    }
    else
    {
        if((0x0a <= j) && (j <= 0x0f))
        {
            j -= 0x09;
            j |= 0x60;
            TmrStrbuf2[i*2] = j;
        }
    }

    j = TouchBuf[i+4];
    j &= 0x0f;
    if((0x00 <= j) && (j <= 0x09))
    {
        j |= 0x30;
        TmrStrbuf2[(i*2)+1] = j;
    }
    else
    {
        if((0x0a <= j) && (j <= 0x0f))
        {
            j -= 0x09;
            j |= 0x60;
            TmrStrbuf2[(i*2)+1] = j;
        }
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

if(('0' <= key) && (key <= '9'))
{
    LcdThaiChar((i*8)+31,16,key);
    LcdUpdate();
    Codedata[i] = key;
    if(i!=7){i++;ready = 0;}
    else{i=7;ready = 1;}
}
else
if(key == 'A') /* delete */
{
    if((1<=i)&&(i<=7))
    {
        if(i==7)
        {
            LcdThaiChar((i*8)+31,16,'-');
            LcdUpdate();
            i--;ready=0;
            j = 1;
        }
        else
        {
            if(j)
            {
                LcdThaiChar((i*8)+31,16,'-');
                LcdUpdate();
                j = 0;ready=0;
            }
            else
            {
                i--;
                LcdThaiChar((i*8)+31,16,'-');
                LcdUpdate();ready=0;
            }
        }
    }
    else
    {Beep(20,50);Beep(23,150);}
}
else
if(key == 'E') /* exit */
{
    LcdClr();
    return(0);
}
else
if(key == 'B') /* Enter */
{
    if(ready)
    {
        Codedata[8] = '\0';
        Beep(21,300);
        LcdClr();
        return(1);
    }
    else Beep(25,400);
    LcdClr();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        return(0);
    }
}

void TimeDisp(void)
{
    UCHAR oldsec;
    char key;
    LcdClr();
    while(1)
    {
        ReadTime();
        if(oldsec != Clock.sec)
        {
            sprintf(PrintBuf, " %2bx:%02bx:%02bx", Clock.hour, Clock.min, Clock.sec);
            LcdThaiString(20,1,PrintBuf);
            sprintf(PrintBuf, " %02bx/%02bx/%02bx", Clock.date, Clock.month, Clock.year);
            LcdThaiString(20,15,PrintBuf);
            LcdUpdate();
            oldsec = Clock.sec;
        }
        key = Getkey();
        if(key == 'E')
        {
            Beep(21,100);
            return;
        }
    }
}

char StandaloneMenu(char menu, char maxmenu)
{
    char key;
    ClearLcdbuf();
    LcdThaiChar(0,1,'>');
    LcdThaiString(9,1, S_Mode[menu]);
    LcdThaiString(9,16,S_Mode[menu+1]);
    LcdUpdate();
    while(1)
    {
        while((key = Getkey()) == ' ');
        if((key == 'D') && (menu != maxmenu)) menu++;
        else
        if((key == 'C') && (menu != 0)) menu--;
        else
        if(key == 'B')
        {
            Beep(21,200);
            return(menu);
        }
        ClearLcdbuf();
        LcdThaiString(9,1, S_Mode[menu]);
        LcdThaiString(9,16,S_Mode[menu+1]);
        LcdThaiChar(0,1,'>');
        LcdUpdate();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void LcdThaiString(char x,char y,char *cp)
```

```
{  
    TEMPX = x;  
    while(*cp)  
        LcdThaiChar(TEMPX,y,*cp++);  
}
```

```
void LcdThaiChar(char x,char y,unsigned char c)
```

```
{  
    char i;  
    int index;  
    index = (c * ThaiCharHeight);  
    if(((0xa0 <= c) && (c <= 0xad)) || ((0xb1 <= c) && (c <= 0xd0))  
        || ((0x20 <= c) && (c <= 0x7f)) || (c == 0xd2) || (c == 0xe0) || (c  
    {  
        ClrThaiBuf();  
        for(i=0;i<8;i++)  
            ThaiBuf[i+6] = ThaiFontTable[index+i];  
        for(i=0;i<16;i++,y++)  
            LcdByte(x,y,ThaiBuf[i]);  
  
        TEMPX += 8;  
        OLDCHAR = c;  
    }  
    else  
        if(c == 0xe2)  
        {  
            ClrThaiBuf();  
            for(i=0;i<8;i++)ThaiBuf[i+6] = ThaiFontTable[index+i];  
            ThaiBuf[3] = 0x98;ThaiBuf[4] = 0x64;ThaiBuf[5] = 0x02;  
            for(i=0;i<16;i++,y++)LcdByte(x,y,ThaiBuf[i]);  
            TEMPX += 8;OLDCHAR = c;  
        }  
    else  
        if(c == 0xe3)  
        {  
            ClrThaiBuf();  
            for(i=0;i<8;i++)ThaiBuf[i+6] = ThaiFontTable[index+i];  
            ThaiBuf[3] = 0x1c;ThaiBuf[4] = 0x22;ThaiBuf[5] = 0x2a;  
            for(i=0;i<16;i++,y++)LcdByte(x,y,ThaiBuf[i]);  
            TEMPX += 8;OLDCHAR = c;  
        }  
    else  
        if(c == 0xe4)  
        {  
            ClrThaiBuf();  
            for(i=0;i<8;i++)ThaiBuf[i+6] = ThaiFontTable[index+i];  
            ThaiBuf[3] = 0x00;ThaiBuf[4] = 0x36;ThaiBuf[5] = 0x28;  
            for(i=0;i<16;i++,y++)LcdByte(x,y,ThaiBuf[i]);  
            TEMPX += 8;OLDCHAR = c;  
        }  
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
if(c == 0xd3)
{
ClrThaiBuf();
for(i=0;i<8;i++)ThaiBuf[i+6] = ThaiFontTable[index+i];
ThaiBuf[4] = 0x03;ThaiBuf[5] = 0x03;
for(i=0;i<16;i++,y++)LcdByte(x,y,ThaiBuf[i]);
TEMPX += 8;OLDCHAR = c;
}

else
if(c == 0xb0)
{
ClrThaiBuf();
for(i=0;i<8;i++)ThaiBuf[i+6] = ThaiFontTable[index+i];
ThaiBuf[13] = 0x54;ThaiBuf[14] = 0x6c;
for(i=0;i<16;i++,y++)LcdByte(x,y,ThaiBuf[i]);
TEMPX += 8;OLDCHAR = c;
}

else
if(c == 0xae)
{
ClrThaiBuf();
for(i=0;i<8;i++)ThaiBuf[i+6] = ThaiFontTable[index+i];
ThaiBuf[13] = 0x50;ThaiBuf[14] = 0x5c;ThaiBuf[15] = 0x6c;
for(i=0;i<16;i++,y++)LcdByte(x,y,ThaiBuf[i]);
TEMPX += 8;OLDCHAR = c;
}

else
if(c == 0xaf)
{
ClrThaiBuf();
for(i=0;i<8;i++)ThaiBuf[i+6] = ThaiFontTable[index+i];
ThaiBuf[13] = 0x40;ThaiBuf[14] = 0x54;ThaiBuf[15] = 0x6a;
for(i=0;i<16;i++,y++)LcdByte(x,y,ThaiBuf[i]);
TEMPX += 8;OLDCHAR = c;
}

else
if( ((0xd4 <= c) && (c <= 0xd7)) || (c ==0xd1 ) )
{
index += 5;
for(i=0;i<3;i++)
ThaiBuf[i+3] = ThaiFontTable[index+i];
for(i=0;i<16;i++,y++)
LcdByte(TEMPX-8,y,ThaiBuf[i]);
OLDCHAR = c;
}
else
if((0xe7 <= c) && (c <= 0xed))
{
if( ((0xd4 <= OLDCHAR) && (OLDCHAR <= 0xd7)) || (OLDCHAR == 0
{
index += 5;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        for(i=0;i<3;i++)
            ThaiBuf[i] = ThaiFontTable[index+i];
        for(i=0;i<16;i++,y++)
            LcdByte(TEMPX-8,y,ThaiBuf[i]);
        OLDCHAR = c;
    }
    else
    {
        index += 5;
        for(i=0;i<3;i++)
            ThaiBuf[i+3] = ThaiFontTable[index+i];
        for(i=0;i<16;i++,y++)
            LcdByte(TEMPX-8,y,ThaiBuf[i]);
        OLDCHAR = c;
    }
}
else
if((c == 0xd8) || (c == 0xd9))
{
    for(i=0;i<2;i++)
        ThaiBuf[i+14] = ThaiFontTable[index+i];
    for(i=0;i<16;i++,y++)
        LcdByte(TEMPX-8,y,ThaiBuf[i]);
}
}

void LcdString(char x,char y,char *cp)
{
    while(*cp)
    {
        LcdChar(x,y,*cp++);
        x += 8;
    }
}

void LcdChar(char x,char y,unsigned char c)
{
    int index;
    char i;
    index = (int)((c & 0x7f) * CharHeight);
    for(i=0;i<CharHeight;i++)
    {
        LcdByte(x,y,(FontTable[index]));
        index++;
        y++;
    }
}

void LcdByte(char x,char y,unsigned char c)
{
    char i,j,k;
    bit pix;
    for(i=x,j=0;j<8;i++,j++)
    {
        k = _crol_(0x01,j);
        if(c & k) pix = 1;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        else pix = 0;
        LcdBit(i,y,pix);
    }
}

void LcdBit(char x,char y,bit b)
{
    int index;
    unsigned char dat;
    unsigned char i = 0x01;
    i = (i << (y*8));
    index = (122*(y/8))+x;
    dat = LcdBuf[index];
    if(b)
        {dat |= i;LcdBuf[index] = dat;}
    else
        {i = ~i;dat &= i;LcdBuf[index] = dat;}
}

void LcdUpdate(void)
{
    char x,y;int z;
    for(y=0;y<4;y++)
        {
            Lcd1wi(0xb8+y);Lcd2wi(0xb8+y);
            for(x=0;x<122;x++)
                {
                    if(x < 61)
                        {z = (122*y)+x;Lcd1wi(x);Lcd1wd(LcdBuf[z]);}
                    else
                        {z = (122*y)+x;Lcd2wi(x-61);Lcd2wd(LcdBuf[z]);}
                }
        }
}

void LcdSet(void)
{
    Lcd1wi(RESET);Lcd1wi(DISP_ON);Lcd1wi(PAGE_0);Lcd1wi(COL_0);
    Lcd2wi(RESET);Lcd2wi(DISP_ON);Lcd2wi(PAGE_0);Lcd2wi(COL_0);
}

void LcdClr(void)
{
    int i;for(i=0;i<488;i++)LcdBuf[i] = 0x00;
    LcdUpdate();
}

void ClearLcdbuf(void)
{
    int i;
    for(i=0;i<488;i++)LcdBuf[i] = 0x00;
}

void Lcd1wd(unsigned char dat)
{Lcd1Busy();CS1_D = dat;}

void Lcd1wi(unsigned char com)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

} {Lcd1Busy();CS1_I = com;}

void Lcd1Busy(void)
{ unsigned char i;do{i = CS1_I;}while(i & 0x80);}

void Lcd2wd(unsigned char dat)
{Lcd2Busy();CS2_D = dat;}

void Lcd2wi(unsigned char com)
{Lcd2Busy();CS2_I = com;}

void Lcd2Busy(void)
{ unsigned char i;do{i = CS2_I;}while(i & 0x80);}

void ClrThaiBuf(void)
{
char i;for(i=0;i<16;i++)ThaiBuf[i] = 0x00;
}

void ReadTime(void)
{
unsigned char xdata *rtcptr;
rtcptr = 0x1ff8;*rtcptr = 0x40;rtcptr++;
Clock.sec = *rtcptr;rtcptr++;
Clock.min = *rtcptr;rtcptr++;
Clock.hour = *rtcptr;rtcptr++;
Clock.day = *rtcptr;rtcptr++;
Clock.date = *rtcptr;rtcptr++;
Clock.month = *rtcptr;rtcptr++;
Clock.year = *rtcptr;
rtcptr = 0x1ff8;*rtcptr = 0x00;
}

void WriteTime(void)
{
unsigned char xdata *rtcptr;
rtcptr = 0x1ff8;*rtcptr = 0x80;rtcptr++;
*rtcptr = Clock.sec;rtcptr++;
*rtcptr = Clock.min;rtcptr++;
*rtcptr = Clock.hour;rtcptr++;
*rtcptr = Clock.day;rtcptr++;
*rtcptr = Clock.date;rtcptr++;
*rtcptr = Clock.month;rtcptr++;
*rtcptr = Clock.year;
rtcptr = 0x1ff8;*rtcptr = 0x00;
}

char Getkey(void)
{
UCHAR col,row;
char keyout;
code char ScanData[4][4] = { {'1','2','3','C'},
{'4','5','6','D'},
{'7','8','9','E'},
{'A','0','B','F'} };

for(col=0;col<4;col++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    P1 = 0xff;
    switch(col)
    {
        case 0: COL3 = 0; break;
        case 1: COL2 = 0; break;
        case 2: COL1 = 0; break;
        case 3: COL0 = 0; break;
    }
    if(!ROW3) row = 0;
    else if(!ROW2) row = 1;
    else if(!ROW1) row = 2;
    else if(!ROW0) row = 3;
    else row = 0xff;
    if(row != 0xff)
    {
        Delay(10);
        if(!ROW3) row = 0;
        else if(!ROW2) row = 1;
        else if(!ROW1) row = 2;
        else if(!ROW0) row = 3;
        else row = 0xff;
        if(row != 0xff)
        {
            keyout = ScanData[row][col];
            if(('0' <= keyout) && (keyout <= '9'))
                Beep(21,60);
            else Beep(23,50);
            Delay(100); return keyout;
        }
    }
}
keyout = ' '; return keyout;
}

void Beep(char freq,int time)
{
    int i;
    for(i=0;i<time;i++)
    {
        SOUNDBIT = 0; DelBeep(freq);
        SOUNDBIT = 1; DelBeep(freq);
    }
}

void DelBeep(int del)
{
    while(del) del--;
}

void Delay(unsigned int delaytime)
{
    bit msecintf;
    TLO = TIMERO_LOW; THO = TIMERO_HIGH;
    ES = 0; ETO = 1; TR0 = 1;
    msecintf = MSECINT;
    while(delaytime)
    {
        while(msecintf == MSECINT); delaytime--;
        msecintf = MSECINT;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    ETO = 0;TRO = 0;ES = 1;
}

void Tx485Char(UCHAR tx485)
{
    while(!TI);
    TI = 0;
    SBUF = tx485;
}

void TXprotocal(void)
{
    char i;
    MAX485 = 1;
    TXLED_ON;
    for(i=0;i<100;i++);
    Tx485Char('{');
    Tx485Char(TX->DAaddress);
    Tx485Char(Myaddress);
    Tx485Char(TX->Command);
    Tx485Char(RX->Datalength);
    CRCLO = 0;
    CRCHI = 0;
    for(i=0;i<RX->Datalength;i++)
    {
        CalcCRC16(Protocaldata[i]);
        Tx485Char(Protocaldata[i]);
    }
    Tx485Char(TX->CRC16LO);
    Tx485Char(TX->CRC16HI);
    Tx485Char('}');
    Tx485Char(' ');
    for(i=0;i<100;i++);
    MAX485 = 0;
    TXLED_OFF;
}

bit RXprotocal(void)
{
    while (!receive_c(&Rcvchar));
    if(Rcvchar != '{')return 0;
    while (!receive_c(&Rcvchar));
    if(Rcvchar != Myaddress).return 0;
    RXLED_ON;
    while (!receive_c(&Rcvchar));
    RX->SAaddress = Rcvchar;
    while (!receive_c(&Rcvchar));
    RX->Command = Rcvchar;
    while (!receive_c(&Rcvchar));
    RX->Datalength = Rcvchar;
    RX->Dataptr = 0;CRCLO = CRCHI = 0;
    while(RX->Dataptr != RX->Datalength)
    {
        while (!receive_c(&Rcvchar));
        Protocaldata[RX->Dataptr] = Rcvchar;
        CalcCRC16(Rcvchar);RX->Dataptr++;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    while (!receive_c(&Rcvchar));
    RX->CRC16LO = Rcvchar;
    while (!receive_c(&Rcvchar));
    RX->CRC16HI = Rcvchar;
    while (!receive_c(&Rcvchar));
    if(Rcvchar != '}')return 0;
    RXLED_OFF;return 1;
}

bit receive_c(UCHAR *c)
{
    Success = 0;
    if(RX->head != RX->tail)
    {
        *c = RX->Queue[RX->head];
        RX->head++;
        Success = 1;
    }
    return(Success);
}

timer0() interrupt 1 using 0
{
    EA = 0;MSECINT = ~MSECINT;
    TLO = TIMERO_LOW;TH0 = TIMERO_HIGH;EA = 1;
}

serial() interrupt 4 using 1
{
    ES = 0;
    if(RI)
    {
        RI = 0;
        if((RX->tail+1) != RX->head)
        {
            RX->Queue[RX->tail] = SBUF;
            RX->tail++;
        }
    }
    else TI = 0;ES = 1;
}

```

```

/***** ThaiFont.h *****/
#define ThaiCharWidth 7
#define ThaiCharHeight 8
unsigned char code ThaiFontTable[] = {
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 00, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 01, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 02, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 03, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 04, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 05, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 06, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 07, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 08, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 09, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 0A, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 0B, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 0C, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 0D, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 0E, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 0F, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 10, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 11, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 12, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 13, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 14, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 15, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 16, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 17, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 18, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 19, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 1A, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 1B, */
0x08,0x00,0x00,0x00,0x08,0x00,0x00,0x00, /* 1C, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 1D, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 1E, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 1F, */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 20, */
0x08,0x08,0x08,0x08,0x08,0x08,0x00,0x00, /* 21, ! */
0x24,0x24,0x24,0x00,0x00,0x00,0x00,0x00, /* 22, " */
0x14,0x14,0x14,0x7F,0x14,0x7F,0x14,0x14, /* 23, # */
0x08,0x7E,0x09,0x09,0x3E,0x48,0x48,0x3F, /* 24, $ */
0x02,0x45,0x22,0x10,0x08,0x04,0x22,0x51, /* 25, % */
0x0E,0x11,0x11,0x0A,0x04,0x4A,0x31,0x31, /* 26, & */
0x02,0x02,0x01,0x00,0x00,0x00,0x00,0x00, /* 27, ' */
0x08,0x04,0x02,0x02,0x02,0x02,0x02,0x04, /* 28, ( */
0x08,0x10,0x20,0x20,0x20,0x20,0x20,0x10, /* 29, ) */
0x08,0x49,0x2A,0x1C,0x7F,0x1C,0x2A,0x49, /* 2A, * */
0x00,0x08,0x08,0x08,0x7F,0x08,0x08,0x08, /* 2B, + */
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 2C, , */
0x00,0x00,0x00,0x7E,0x00,0x00,0x00,0x00, /* 2D, - */
0x00,0x00,0x00,0x00,0x00,0x60,0x60,0x00, /* 2E, . */
0x00,0x40,0x20,0x10,0x08,0x04,0x02,0x00, /* 2F, / */
0x1C,0x22,0x51,0x49,0x45,0x22,0x1C,0x00, /* 30, 0 */
0x08,0x0E,0x08,0x08,0x08,0x08,0x3E,0x00, /* 31, 1 */
0x3E,0x41,0x40,0x3E,0x01,0x01,0x7E,0x00, /* 32, 2 */
0x3E,0x41,0x40,0x38,0x40,0x41,0x3E,0x00, /* 33, 3 */
0x30,0x28,0x24,0x22,0x7F,0x20,0x20,0x00, /* 34, 4 */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0x7F, 0x01, 0x3F, 0x40, 0x41, 0x41, 0x3E, 0x00,
0x3E, 0x41, 0x01, 0x3F, 0x41, 0x41, 0x3E, 0x00,
0x7F, 0x20, 0x10, 0x08, 0x04, 0x02, 0x01, 0x00,
0x3E, 0x41, 0x41, 0x3E, 0x41, 0x41, 0x3E, 0x00,
0x3E, 0x41, 0x41, 0x7E, 0x40, 0x41, 0x3E, 0x00,
0x00, 0x08, 0x00, 0x00, 0x00, 0x08, 0x00, 0x00,
0x00, 0x00, 0x00, 0x02, 0x00, 0x00, 0x00, 0x00,
0x10, 0x08, 0x04, 0x02, 0x01, 0x02, 0x04, 0x08,
0x00, 0x00, 0x00, 0x7F, 0x00, 0x7F, 0x00, 0x00,
0x08, 0x10, 0x20, 0x40, 0x20, 0x10, 0x08, 0x00,
0x3E, 0x41, 0x41, 0x20, 0x10, 0x08, 0x08, 0x00,
0x3E, 0x41, 0x41, 0x59, 0x55, 0x3D, 0x01, 0x41,
0x1C, 0x22, 0x41, 0x41, 0x7F, 0x41, 0x41, 0x00,
0x3F, 0x41, 0x41, 0x3F, 0x41, 0x41, 0x3F, 0x00,
0x7C, 0x02, 0x01, 0x01, 0x01, 0x02, 0x7C, 0x00,
0x1F, 0x21, 0x41, 0x41, 0x41, 0x21, 0x1F, 0x00,
0x7F, 0x01, 0x01, 0x1F, 0x01, 0x01, 0x7F, 0x00,
0x7F, 0x01, 0x01, 0x1F, 0x01, 0x01, 0x01, 0x00,
0x3E, 0x41, 0x01, 0x79, 0x41, 0x41, 0x3E, 0x00,
0x41, 0x41, 0x41, 0x7F, 0x41, 0x41, 0x41, 0x00,
0x3E, 0x08, 0x08, 0x08, 0x08, 0x08, 0x3E, 0x00,
0x7C, 0x10, 0x10, 0x10, 0x10, 0x11, 0x0E, 0x00,
0x41, 0x21, 0x11, 0x0F, 0x11, 0x21, 0x41, 0x00,
0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x7F, 0x00,
0x41, 0x63, 0x55, 0x49, 0x41, 0x41, 0x41, 0x00,
0x41, 0x43, 0x45, 0x49, 0x51, 0x61, 0x41, 0x00,
0x1C, 0x22, 0x41, 0x41, 0x41, 0x22, 0x1C, 0x00,
0x3F, 0x41, 0x41, 0x3F, 0x01, 0x01, 0x01, 0x00,
0x1C, 0x22, 0x41, 0x41, 0x51, 0x22, 0x5C, 0x00,
0x3F, 0x41, 0x41, 0x3F, 0x11, 0x21, 0x41, 0x00,
0x3E, 0x41, 0x01, 0x3E, 0x40, 0x41, 0x3E, 0x00,
0x7F, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x00,
0x41, 0x41, 0x41, 0x41, 0x41, 0x22, 0x1C, 0x00,
0x41, 0x41, 0x41, 0x41, 0x22, 0x14, 0x08, 0x00,
0x41, 0x41, 0x41, 0x49, 0x49, 0x2A, 0x14, 0x00,
0x41, 0x22, 0x14, 0x08, 0x14, 0x22, 0x41, 0x00,
0x41, 0x22, 0x14, 0x08, 0x08, 0x08, 0x08, 0x00,
0x7F, 0x20, 0x10, 0x08, 0x04, 0x02, 0x7F, 0x00,
0x3E, 0x02, 0x02, 0x02, 0x02, 0x02, 0x3E, 0x00,
0x00, 0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x00,
0x3E, 0x20, 0x20, 0x20, 0x20, 0x20, 0x3E, 0x00,
0x3E, 0x41, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x20, 0x20, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x3E, 0x40, 0x7E, 0x41, 0x7E, 0x00,
0x01, 0x01, 0x3D, 0x43, 0x41, 0x41, 0x3F, 0x00,
0x00, 0x00, 0x7E, 0x01, 0x01, 0x01, 0x7E, 0x00,
0x40, 0x40, 0x5E, 0x61, 0x41, 0x41, 0x7E, 0x00,
0x00, 0x00, 0x3E, 0x41, 0x7F, 0x01, 0x3E, 0x00,
0x38, 0x44, 0x04, 0x1F, 0x04, 0x04, 0x04, 0x00,
0x00, 0x00, 0x7E, 0x41, 0x7E, 0x40, 0x3E, 0x00,
0x01, 0x01, 0x3D, 0x43, 0x41, 0x41, 0x41, 0x00,
0x08, 0x00, 0x0E, 0x08, 0x08, 0x08, 0x3E, 0x00,
0x40, 0x00, 0x70, 0x40, 0x40, 0x41, 0x3E, 0x00,
0x01, 0x01, 0x41, 0x21, 0x1F, 0x21, 0x41, 0x00,
0x0E, 0x08, 0x08, 0x08, 0x08, 0x08, 0x3E, 0x00,
0x00, 0x00, 0x37, 0x49, 0x49, 0x49, 0x49, 0x00,

/* 35, 5 */
/* 36, 6 */
/* 37, 7 */
/* 38, 8 */
/* 39, 9 */
/* 3A, : */
/* 3B, ; */
/* 3C, < */
/* 3D, = */
/* 3E, > */
/* 3F, ? */
/* 40, @ */
/* 41, A */
/* 42, B */
/* 43, C */
/* 44, D */
/* 45, E */
/* 46, F */
/* 47, G */
/* 48, H */
/* 49, I */
/* 4A, J */
/* 4B, K */
/* 4C, L */
/* 4D, M */
/* 4E, N */
/* 4F, O */
/* 50, P */
/* 51, Q */
/* 52, R */
/* 53, S */
/* 54, T */
/* 55, U */
/* 56, V */
/* 57, W */
/* 58, X */
/* 59, Y */
/* 5A, Z */
/* 5B, [*/
/* 5C, \ */
/* 5D,] */
/* 5E, ^ */
/* 5F, _ */
/* 60, ` */
/* 61, a */
/* 62, b */
/* 63, c */
/* 64, d */
/* 65, e */
/* 66, f */
/* 67, g */
/* 68, h */
/* 69, i */
/* 6A, j */
/* 6B, k */
/* 6C, l */
/* 6D, m */

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```
/****** EngFont.h *****/
```

```
#define CharWidth 7
```

```
#define CharHeight 12
```

```
unsigned char code FontTable[] = {  
0x00,0x00,0x00,0x00,0x46,0x29,0x11,0x29,0x46,0x00,0x00,0x00, /* 00,  
0x00,0x00,0x00,0x1E,0x22,0x1E,0x22,0x22,0x1E,0x02,0x02,0x01, /* 01,  
0x00,0x00,0x00,0x00,0x43,0x24,0x14,0x0C,0x04,0x06,0x06,0x06, /* 02,  
0x06,0x09,0x01,0x01,0x02,0x06,0x09,0x09,0x06,0x00,0x00,0x00, /* 03,  
0x00,0x00,0x0C,0x02,0x01,0x0F,0x01,0x02,0x0C,0x00,0x00,0x00, /* 04,  
0x34,0x38,0x04,0x02,0x01,0x01,0x0E,0x10,0x0C,0x00,0x00,0x00, /* 05,  
0x00,0x00,0x00,0x00,0x1A,0x25,0x24,0x24,0x24,0x20,0x20,0x20, /* 06,  
0x0C,0x12,0x21,0x21,0x3F,0x21,0x21,0x12,0x0C,0x00,0x00,0x00, /* 07,  
0x00,0x00,0x00,0x01,0x01,0x01,0x01,0x09,0x06,0x00,0x00,0x00, /* 08,  
0x00,0x00,0x01,0x09,0x05,0x03,0x05,0x29,0x11,0x00,0x00,0x00, /* 09,  
0x01,0x02,0x04,0x04,0x04,0x04,0x0C,0x12,0x21,0x00,0x00,0x00, /* 0A,  
0x00,0x00,0x00,0x00,0x09,0x09,0x09,0x09,0x17,0x01,0x01,0x01, /* 0B,  
0x00,0x00,0x00,0x23,0x22,0x12,0x0A,0x06,0x02,0x00,0x00,0x00, /* 0C,  
0x08,0x1C,0x02,0x0C,0x02,0x01,0x1E,0x20,0x18,0x00,0x00,0x00, /* 0D,  
0x00,0x00,0x00,0x0C,0x12,0x21,0x21,0x12,0x0C,0x00,0x00,0x00, /* 0E,  
0x00,0x00,0x00,0x7E,0x15,0x14,0x14,0x14,0x14,0x00,0x00,0x00, /* 0F,  
0x00,0x00,0x00,0x0C,0x12,0x21,0x21,0x13,0x0D,0x01,0x01,0x01, /* 10,  
0x00,0x00,0x00,0x7C,0x12,0x21,0x21,0x12,0x0C,0x00,0x00,0x00, /* 11,  
0x00,0x00,0x00,0x7E,0x09,0x08,0x08,0x08,0x08,0x00,0x00,0x00, /* 12,  
0x00,0x00,0x00,0x23,0x12,0x12,0x12,0x12,0x0C,0x00,0x00,0x00, /* 13,  
0x04,0x04,0x0E,0x15,0x15,0x15,0x0E,0x04,0x04,0x00,0x00,0x00, /* 14,  
0x00,0x00,0x00,0x00,0x23,0x14,0x08,0x14,0x62,0x00,0x00,0x00, /* 15,  
0x00,0x00,0x00,0x08,0x49,0x2A,0x2A,0x2A,0x1C,0x08,0x08,0x08, /* 16,  
0x00,0x00,0x00,0x22,0x41,0x49,0x49,0x49,0x36,0x00,0x00,0x00, /* 17,  
0x00,0x1C,0x22,0x41,0x41,0x41,0x22,0x22,0x63,0x00,0x00,0x00, /* 18,  
0x7C,0x04,0x04,0x04,0x04,0x04,0x05,0x06,0x04,0x00,0x00,0x00, /* 19,  
0x00,0x00,0x10,0x20,0x7F,0x20,0x10,0x00,0x00,0x00,0x00,0x00, /* 1A,  
0x00,0x00,0x04,0x02,0x7F,0x02,0x04,0x00,0x00,0x00,0x00,0x00, /* 1B,  
0x08,0x1C,0x2A,0x08,0x08,0x08,0x08,0x08,0x08,0x00,0x00,0x00, /* 1C,  
0x00,0x00,0x08,0x00,0x7F,0x00,0x08,0x00,0x00,0x00,0x00,0x00, /* 1D,  
0x7F,0x02,0x04,0x08,0x30,0x08,0x04,0x02,0x7F,0x00,0x00,0x00, /* 1E,  
0x00,0x06,0x49,0x30,0x06,0x49,0x30,0x00,0x00,0x00,0x00,0x00, /* 1F,  
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 20,  
0x08,0x08,0x08,0x08,0x08,0x08,0x00,0x00,0x08,0x00,0x00,0x00, /* 21,  
0x24,0x24,0x24,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 22,  
0x14,0x14,0x14,0x7F,0x14,0x7F,0x14,0x14,0x14,0x00,0x00,0x00, /* 23,  
0x08,0x7E,0x09,0x09,0x3E,0x48,0x48,0x3F,0x08,0x00,0x00,0x00, /* 24,  
0x02,0x45,0x22,0x10,0x08,0x04,0x22,0x51,0x20,0x00,0x00,0x00, /* 25,  
0x0E,0x11,0x11,0x0A,0x04,0x4A,0x31,0x31,0x4E,0x00,0x00,0x00, /* 26,  
0x02,0x02,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 27,  
0x08,0x04,0x02,0x02,0x02,0x02,0x02,0x04,0x08,0x00,0x00,0x00, /* 28,  
0x08,0x10,0x20,0x20,0x20,0x20,0x20,0x10,0x08,0x00,0x00,0x00, /* 29,  
0x08,0x49,0x2A,0x1C,0x7F,0x1C,0x2A,0x49,0x08,0x00,0x00,0x00, /* 2A,  
0x00,0x08,0x08,0x08,0x7F,0x08,0x08,0x08,0x00,0x00,0x00,0x00, /* 2B,  
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x06,0x04,0x04,0x02, /* 2C,  
0x00,0x00,0x00,0x00,0x7F,0x00,0x00,0x00,0x00,0x00,0x00,0x00, /* 2D,  
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x04,0x00,0x00,0x00, /* 2E,  
0x00,0x40,0x20,0x10,0x08,0x04,0x02,0x01,0x00,0x00,0x00,0x00, /* 2F,  
0x3E,0x41,0x61,0x51,0x49,0x45,0x43,0x41,0x3E,0x00,0x00,0x00, /* 30,  
0x08,0x0C,0x08,0x08,0x08,0x08,0x08,0x08,0x3E,0x00,0x00,0x00, /* 31,  
0x3E,0x41,0x40,0x20,0x18,0x04,0x02,0x01,0x7F,0x00,0x00,0x00, /* 32,  
0x3E,0x41,0x40,0x40,0x3C,0x40,0x40,0x41,0x3E,0x00,0x00,0x00, /* 33,  
0x20,0x30,0x28,0x24,0x22,0x7F,0x20,0x20,0x20,0x00,0x00,0x00, /* 34
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0x7F, 0x01, 0x01, 0x01, 0x3F, 0x40, 0x40, 0x41, 0x3E, 0x00, 0x00, 0x00, /* 35,
0x3E, 0x41, 0x01, 0x01, 0x3F, 0x41, 0x41, 0x41, 0x3E, 0x00, 0x00, 0x00, /* 36,
0x7F, 0x41, 0x20, 0x10, 0x08, 0x04, 0x04, 0x04, 0x04, 0x00, 0x00, 0x00, /* 37,
0x3E, 0x41, 0x41, 0x41, 0x3E, 0x41, 0x41, 0x41, 0x3E, 0x00, 0x00, 0x00, /* 38,
0x3E, 0x41, 0x41, 0x41, 0x7E, 0x40, 0x40, 0x41, 0x3E, 0x00, 0x00, 0x00, /* 39,
0x00, 0x00, 0x08, 0x00, 0x00, 0x00, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, /* 3A,
0x00, 0x00, 0x00, 0x02, 0x00, 0x00, 0x00, 0x00, 0x06, 0x04, 0x04, 0x02, /* 3B,
0x10, 0x08, 0x04, 0x02, 0x01, 0x02, 0x04, 0x08, 0x10, 0x00, 0x00, 0x00, /* 3C,
0x00, 0x00, 0x00, 0x7F, 0x00, 0x7F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 3D,
0x04, 0x08, 0x10, 0x20, 0x40, 0x20, 0x10, 0x08, 0x04, 0x00, 0x00, 0x00, /* 3E,
0x3E, 0x41, 0x41, 0x20, 0x10, 0x08, 0x08, 0x00, 0x08, 0x00, 0x00, 0x00, /* 3F,
0x3E, 0x41, 0x41, 0x59, 0x55, 0x3D, 0x01, 0x41, 0x3E, 0x00, 0x00, 0x00, /* 40,
0x1C, 0x22, 0x41, 0x41, 0x7F, 0x41, 0x41, 0x41, 0x41, 0x00, 0x00, 0x00, /* 41,
0x3F, 0x42, 0x42, 0x42, 0x3E, 0x42, 0x42, 0x42, 0x3F, 0x00, 0x00, 0x00, /* 42,
0x3C, 0x42, 0x01, 0x01, 0x01, 0x01, 0x01, 0x42, 0x3C, 0x00, 0x00, 0x00, /* 43,
0x3F, 0x42, 0x42, 0x42, 0x42, 0x42, 0x42, 0x42, 0x3F, 0x00, 0x00, 0x00, /* 44,
0x7F, 0x01, 0x01, 0x01, 0x0F, 0x01, 0x01, 0x01, 0x7F, 0x00, 0x00, 0x00, /* 45,
0x7F, 0x01, 0x01, 0x01, 0x0F, 0x01, 0x01, 0x01, 0x01, 0x00, 0x00, 0x00, /* 46,
0x3C, 0x42, 0x01, 0x01, 0x01, 0x71, 0x41, 0x42, 0x3C, 0x00, 0x00, 0x00, /* 47,
0x41, 0x41, 0x41, 0x41, 0x7F, 0x41, 0x41, 0x41, 0x41, 0x00, 0x00, 0x00, /* 48,
0x3E, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x3E, 0x00, 0x00, 0x00, /* 49,
0x7C, 0x10, 0x10, 0x10, 0x10, 0x10, 0x10, 0x11, 0x0E, 0x00, 0x00, 0x00, /* 4A,
0x41, 0x21, 0x11, 0x09, 0x07, 0x09, 0x11, 0x21, 0x41, 0x00, 0x00, 0x00, /* 4B,
0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x7F, 0x00, 0x00, 0x00, /* 4C,
0x41, 0x63, 0x55, 0x49, 0x49, 0x41, 0x41, 0x41, 0x41, 0x00, 0x00, 0x00, /* 4D,
0x41, 0x43, 0x45, 0x49, 0x51, 0x61, 0x41, 0x41, 0x41, 0x00, 0x00, 0x00, /* 4E,
0x1C, 0x22, 0x41, 0x41, 0x41, 0x41, 0x41, 0x22, 0x1C, 0x00, 0x00, 0x00, /* 4F,
0x3F, 0x41, 0x41, 0x41, 0x3F, 0x01, 0x01, 0x01, 0x01, 0x00, 0x00, 0x00, /* 50,
0x1C, 0x22, 0x41, 0x41, 0x41, 0x41, 0x51, 0x22, 0x5C, 0x00, 0x00, 0x00, /* 51,
0x3F, 0x41, 0x41, 0x41, 0x3F, 0x09, 0x11, 0x21, 0x41, 0x00, 0x00, 0x00, /* 52,
0x3E, 0x41, 0x01, 0x01, 0x3E, 0x40, 0x40, 0x41, 0x3E, 0x00, 0x00, 0x00, /* 53,
0x7F, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x00, 0x00, 0x00, /* 54,
0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x3E, 0x00, 0x00, 0x00, /* 55,
0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x22, 0x14, 0x08, 0x00, 0x00, 0x00, /* 56,
0x41, 0x41, 0x41, 0x49, 0x49, 0x49, 0x55, 0x63, 0x41, 0x00, 0x00, 0x00, /* 57,
0x41, 0x41, 0x22, 0x14, 0x08, 0x14, 0x22, 0x41, 0x41, 0x00, 0x00, 0x00, /* 58,
0x41, 0x41, 0x22, 0x14, 0x08, 0x08, 0x08, 0x08, 0x08, 0x00, 0x00, 0x00, /* 59,
0x7F, 0x40, 0x20, 0x10, 0x08, 0x04, 0x02, 0x01, 0x7F, 0x00, 0x00, 0x00, /* 5A,
0x3C, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x3C, 0x00, 0x00, 0x00, /* 5B,
0x00, 0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x00, 0x00, 0x00, 0x00, /* 5C,
0x1E, 0x10, 0x10, 0x10, 0x10, 0x10, 0x10, 0x10, 0x1E, 0x00, 0x00, 0x00, /* 5D,
0x3E, 0x41, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 5E,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7F, 0x00, 0x00, 0x00, /* 5F,
0x20, 0x20, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 60,
0x00, 0x00, 0x00, 0x3C, 0x22, 0x21, 0x21, 0x31, 0x2E, 0x00, 0x00, 0x00, /* 61,
0x01, 0x01, 0x01, 0x1D, 0x23, 0x21, 0x21, 0x23, 0x1D, 0x00, 0x00, 0x00, /* 62,
0x00, 0x00, 0x00, 0x1E, 0x21, 0x01, 0x01, 0x21, 0x1E, 0x00, 0x00, 0x00, /* 63,
0x20, 0x20, 0x20, 0x2E, 0x31, 0x21, 0x21, 0x31, 0x2E, 0x00, 0x00, 0x00, /* 64,
0x00, 0x00, 0x00, 0x1E, 0x21, 0x21, 0x3F, 0x01, 0x3E, 0x00, 0x00, 0x00, /* 65,
0x18, 0x24, 0x04, 0x04, 0x1F, 0x04, 0x04, 0x04, 0x04, 0x00, 0x00, 0x00, /* 66,
0x00, 0x00, 0x00, 0x2E, 0x31, 0x21, 0x21, 0x31, 0x2E, 0x20, 0x21, 0x1E, /* 67,
0x01, 0x01, 0x01, 0x0D, 0x13, 0x21, 0x21, 0x21, 0x21, 0x00, 0x00, 0x00, /* 68,
0x00, 0x08, 0x00, 0x0C, 0x08, 0x08, 0x08, 0x08, 0x08, 0x00, 0x00, 0x00, /* 69,
0x00, 0x00, 0x00, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x21, 0x1E, /* 6A,
0x01, 0x01, 0x01, 0x11, 0x09, 0x07, 0x09, 0x11, 0x21, 0x00, 0x00, 0x00, /* 6B,
0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x00, 0x00, 0x00, /* 6C,
0x00, 0x00, 0x00, 0x37, 0x49, 0x49, 0x49, 0x49, 0x49, 0x00, 0x00, 0x00, /* 6D,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0x00, 0x00, 0x00, 0x1D, 0x23, 0x21, 0x21, 0x21, 0x21, 0x00, 0x00, 0x00, /* 6E,
0x00, 0x00, 0x00, 0x1E, 0x21, 0x21, 0x21, 0x21, 0x1E, 0x00, 0x00, 0x00, /* 6F,
0x00, 0x00, 0x00, 0x1D, 0x23, 0x21, 0x21, 0x23, 0x1D, 0x01, 0x01, 0x01, /* 70,
0x00, 0x00, 0x00, 0x2E, 0x31, 0x21, 0x21, 0x31, 0x2E, 0x20, 0x20, 0x20, /* 71,
0x00, 0x00, 0x00, 0x1D, 0x23, 0x01, 0x01, 0x01, 0x01, 0x00, 0x00, 0x00, /* 72,
0x00, 0x00, 0x00, 0x1E, 0x21, 0x06, 0x18, 0x21, 0x1E, 0x00, 0x00, 0x00, /* 73,
0x00, 0x04, 0x04, 0x1F, 0x04, 0x04, 0x04, 0x24, 0x18, 0x00, 0x00, 0x00, /* 74,
0x00, 0x00, 0x00, 0x21, 0x21, 0x21, 0x21, 0x21, 0x1E, 0x00, 0x00, 0x00, /* 75,
0x00, 0x00, 0x00, 0x11, 0x11, 0x11, 0x11, 0x0A, 0x04, 0x00, 0x00, 0x00, /* 76,
0x00, 0x00, 0x00, 0x41, 0x49, 0x49, 0x49, 0x49, 0x36, 0x00, 0x00, 0x00, /* 77,
0x00, 0x00, 0x00, 0x21, 0x12, 0x0C, 0x0C, 0x12, 0x21, 0x00, 0x00, 0x00, /* 78,
0x00, 0x00, 0x00, 0x21, 0x21, 0x21, 0x21, 0x31, 0x2E, 0x20, 0x21, 0x1E, /* 79,
0x00, 0x00, 0x00, 0x3F, 0x10, 0x08, 0x04, 0x02, 0x3F, 0x00, 0x00, 0x00, /* 7A,
0x38, 0x04, 0x04, 0x04, 0x02, 0x04, 0x04, 0x04, 0x38, 0x00, 0x00, 0x00, /* 7B,
0x08, 0x08, 0x08, 0x00, 0x00, 0x08, 0x08, 0x08, 0x00, 0x00, 0x00, 0x00, /* 7C,
0x0E, 0x10, 0x10, 0x10, 0x20, 0x10, 0x10, 0x10, 0x0E, 0x00, 0x00, 0x00, /* 7D,
0x06, 0x49, 0x30, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 7E,
0x7F, 0x7F, 0x7F, 0x7F, 0x7F, 0x7F, 0x7F, 0x7F, 0x7F, 0x00, 0x00, 0x00 /* 7F,
};



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



1. ยืน ภู่วรรณ , “ การสื่อสารข้อมูลและไมโครคอมพิวเตอร์ ”, บริษัทซีเอ็ดยูเคชั่น จำกัด,
240 หน้า, 2532
2. ชันวา ศรีประโมง , “ การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม ”, โครงการตำราวิชาการ
มหาวิทยาลัยเทคโนโลยีมหานคร, 450 หน้า, 2537
3. ชูชัย ชนสารตั้งเจริญ , “ การสื่อสารข้อมูล ” โครงการตำราเรียน ฟิสิกส์เซนเตอร์, 200 หน้า,
2531
4. Dallas Semiconductor , “ DS 1993 Touch Memory Chip ”, Dallas Semiconductor Data Book,
20 P, 1994
5. Maxim Inc , “ MAX 232 & MAC 691 Maxim New Releases Data Book Guide Series Book1 ”
, Maxim Integrated Product, 1992

