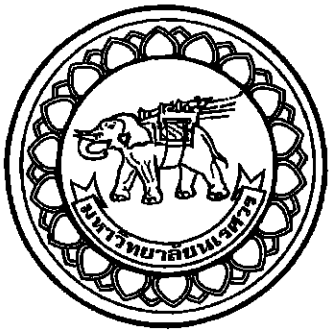
อภินันทนาการ



ระบบควบคุมการเข้าออกอาคารและห้องทำงาน

ROOM AND BUILDING ACCESS SYSTEM

นางสาวกรรณิกา รุจิรวณิชเทพ รหัส 43362375

นายกำพล จันต๊ะคาด รหัส 43362391

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2546

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	ระบบควบคุมการเข้าออกอาคารและห้องทำงาน
ผู้ดำเนินโครงงาน	นางสาวกรรณิกา รุจิรวณิชเทพ รหัส 43362375
	นายกำพล จันต๊ะคาด รหัส 43362391
อาจารย์ที่ปรึกษา	ดร.ยงยุทธ ชนบดีเฉลิมรุ่ง
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2546

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ
(ดร.ยงยุทธ ชนบดีเฉลิมรุ่ง)

.....กรรมการ
(อาจารย์พนัส นัถฤทธิ์)

.....กรรมการ
(ดร.สุรเชษฐ์ กานต์ประชา)

หัวข้อโครงงาน	ระบบควบคุมการเข้าออกอาคารและห้องทำงาน
ผู้ดำเนินโครงงาน	นางสาวกรรณิกา รุจิรวณิชเทพ รหัส 43362375
	นายกำพล จันต๊ะคาด รหัส 43362391
อาจารย์ที่ปรึกษา	ดร.ยงยุทธ ชนบดีเฉลิมรุ่ง
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2546

.....

บทคัดย่อ

โครงงานนี้จัดทำขึ้นเพื่อพัฒนาระบบตรวจสอบการเข้าใช้ห้อง ให้สามารถบันทึกเก็บข้อมูลบุคคลที่เข้าใช้ห้องนั้น ๆ ได้ในระยะทางไกล เพื่อความสะดวกของผู้ใช้ห้อง สำหรับการแลกเปลี่ยนข้อมูลระหว่าง คอมพิวเตอร์ และ บอร์ดควบคุม จะส่งผ่านทางตัวแปลงสัญญาณ ซึ่งตัวแปลงสัญญาณนี้จะทำการแปลงมาตรฐาน RS232 ไปเป็น RS422 ในช่วงเวลาที่บอร์ดได้รับคำสั่งจากคอมพิวเตอร์ จะทำการสั่ง โซลินอยด์ทำการปิดเปิดประตู ซึ่งบอร์ดควบคุมนี้อยู่ห่างจากคอมพิวเตอร์ประมาณ 100 เมตร และมีการควบคุม 2 บอร์ดควบคุม 2 ประตู

ผลที่ได้รับจากโครงงานนี้ คือ อุปกรณ์ทางด้านฮาร์ดแวร์และซอฟแวร์สำหรับใช้รับค่าและตรวจบันทึกข้อมูล เพื่อใช้ในการเปิด-ปิดประตูที่อยู่ในระยะไกล และคอมพิวเตอร์สามารถใช้ควบคุมเป็นระบบเครือข่ายที่มีมากกว่า 1 เอาท์พุต

Project Title	Room and Building Accesses System		
Name	Miss kannikar	Ruchirawanithep	ID 43362375
	Mr.Kampol	Jantakard	ID 43362391
Project Advisor	Dr. Yongyoot	Chonbodychaloemrung	
Major	Electrical Engineering		
Department	Electrical and Computer Engineering		
Academic Year	2003		

.....

ABSTRACT

This project is created to develop the room access system for storing database of users who can be convenient access any permissible rooms far from the control room. Signal and data between the computer and control boards can be exchanged by passing through the transforming box. The transforming box will transform RS232 signal into RS422 signal .As soon as the control board receives the command, it will command the solenoid to open or close the door. The control board is about 100 meters away from the computer. There are two-control board and two door in this project.

The result of this project is equipment that consists of hardware for opening or closing the long distance door and software for storing information .The computer can be used as the network control system with more than one output.

กิตติกรรมประกาศ

โครงงานวิศวกรรมเรื่องระบบควบคุมการเข้าออกอาคารและห้องทำงาน ที่ทำสำเร็จเป็นรูปเล่มได้เนื่องจากได้รับความกรุณาของอาจารย์ยงยุทธ ชนบดีเฉลิมรุ่ง ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ มหาวิทยาลัยนเรศวร ผู้เป็นอาจารย์ที่ปรึกษาได้วางเมื่อมีการสับเปลี่ยนการติดต่อบอร์ดทดลองทั้งสองจะต้องทำการรูดบัตรซ้ำอีกรากฐาน และประสิทธิวิชาความรู้ทางด้านนี้ในการดำเนินงาน ทั้งยังได้เอื้อเฟื้อ ให้ความช่วยเหลือเป็นอย่างดีให้แก่ผู้ร่วมงาน จึงขอแสดงความขอบคุณเป็นอย่างสูง ณ ที่นี้ด้วย

ขอแสดงความขอบคุณผู้ที่ช่วยให้โครงงานนี้สำเร็จได้ด้วยดี

อาจารย์ยงยุทธ ชนบดีเฉลิมรุ่ง ที่ปรึกษาโครงงาน

นางสาวกรรณิกา รุจิรวณิชเทพ

นายกำพล จันต๊ะคาด

สารบัญ

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

สารบัญรูป (ต่อ)

บทที่ 1
บทนำ

1.1 หลักการและเหตุผล

จากอดีตที่ผ่านมา โลกของเรามีการพัฒนาด้านเทคโนโลยีไม่มากนัก ประกอบกับการดำรงชีวิตในแต่ละวันของมนุษย์ก็ยังไม่มีความรีบเร่ง แข่งขัน และแสวงหาความสะดวกสบายเหมือนเช่นปัจจุบัน

ปัจจุบันโลกก้าวล้ำด้วยผลงานทางด้านเทคโนโลยีที่อำนวยความสุขสบายให้แก่มนุษย์ในทุก ๆ ด้าน นับตั้งแต่คิดด้วยสมองกล สื่อสารด้วยใยแก้วนำแสง ฟังเพลงด้วยเลเซอร์ ไปจนถึงบอกเวลาด้วยนาฬิกาอะตอม และคงไม่มีใครปฏิเสธได้ว่าไม่เคยสัมผัสกับคอมพิวเตอร์ไม่ว่าโดยทางตรงและทางอ้อม สำหรับคอมพิวเตอร์ที่ใช้งานโดยทั่วไปไม่ได้จำกัดให้ทำงานในเฉพาะด้าน และมีจำนวนมากที่สุดในปัจจุบันก็คือ เครื่องที่เรียกกันว่า PC ซึ่งย่อมาจาก Personal Computer ถ้าแปลกันตรงตัว ก็คือ เครื่องคอมพิวเตอร์สำหรับใช้งานส่วนบุคคล ที่เรียกกันอย่างแพร่หลายกันอย่างนี้ก็เพราะเป็นการเปรียบเทียบคอมพิวเตอร์ขนาดใหญ่หรือเมนเฟรมที่มีไว้สำหรับใช้งานร่วมกันหลาย ๆ คนในสมัยก่อน แต่มาถึงปัจจุบันเครื่องพีซีถูกพัฒนาจนมีความสามารถทำงานได้มากมายมหาศาล หรือแม้แต่จะเป็นเซิร์ฟเวอร์ที่ใช้พร้อมกับหลาย ๆ คนหรือเชื่อมต่อกับเครื่องอื่น ๆ ได้อีกมากมายเกินกว่าที่เคยเป็นในยุคแรกมาก แต่ก็ยังคงถูกเรียกกว่าพีซีอย่างเช่นเดิม เครื่องเหล่านี้เองทำให้เกิดความก้าวหน้าไปไกลมากของเทคโนโลยี แต่ในทางกลับกัน ก็ก่อให้เกิดผลเสียในด้านความปลอดภัย เพราะเทคโนโลยีและคอมพิวเตอร์ถูกนำมาใช้ในทางที่มิชอบ ในเรื่องของการโจรกรรมทั้งทางข้อมูลข่าวสารและทรัพย์สินทำให้ความปลอดภัยของมนุษย์ลดน้อยลง จึงทำให้เกิดแนวความคิดที่จะนำระบบควบคุมการเข้า-ออกอาคารและห้องทำงานมาป้องกันข้อมูลและทรัพย์สินเพื่อให้เกิดความปลอดภัยและอำนวยให้เกิดความสะดวกของผู้ใช้งานมากยิ่งขึ้น

1.2 วัตถุประสงค์ของการศึกษาโครงงาน

1. เพื่อพัฒนาระบบการควบคุมการเข้า-ออกอาคารและห้องทำงานให้มีประสิทธิภาพในการรักษาความปลอดภัยของทรัพย์สินในแต่ละอาคารและแต่ละห้องทำงานภายนอกเวลาราชการ
2. เพื่อพัฒนาระบบควบคุมการเข้า-ออกอาคารและห้องทำงานให้มีความสะดวกรวดเร็วในการเข้า-ออก อาคารและห้องทำงานมากยิ่งขึ้น
3. เพื่อศึกษาการรับ-ส่งข้อมูลในระยะไกล

1.3 ขอบข่ายของงาน

มีเป้าหมายควบคุมการเข้า-ออกอาคารและห้องทำงานนอกเวลาราชการ โดยใช้คอมพิวเตอร์ส่วนกลางมาควบคุมประตูที่ติดตั้งบาร์โค้ด 3 แห่ง และประตูมีระยะที่ห่างจากส่วนควบคุมออกไปไม่น้อยกว่า 500 เมตร ผู้ที่จะสามารถผ่านเข้า–ออก แต่ละประตูได้จะต้องได้รับอนุญาตจากส่วนควบคุมก่อน โดยต้องมีบัตรนิสิตและรหัสที่ยืนยันความถูกต้องถึงจะผ่านเข้า-ออกได้โดยมี

- ไมโครคอนโทรลเลอร์ ควบคุมแต่ละประตู
- มีวงจรรับ – ส่ง สัญญาณ RS 485 – 422 แปลงเป็น RS 232
- มีฐานข้อมูลใน Computer

1.4 กิจกรรมการดำเนินงาน

กิจกรรม	2545		2546									
	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.
ศึกษาหลักการทำงานของระบบที่จะนำมาพัฒนาโครงงาน												
ศึกษามาตรฐานการรับส่งข้อมูลแบบ RS 232,RS 422,RS485												
เรียนรู้โปรแกรมควบคุมการทำงาน												
พัฒนาระบบฐานข้อมูล												
ออกแบบวงจร												
ประกอบวงจร												
ทดสอบการทำงานและแก้ไขข้อผิดพลาด												
ทำวิทยานิพนธ์												

1.5 ผลที่คาดว่าจะได้รับ

1. สามารถนำเอาความรู้ในเรื่องมาตรฐานสายส่ง RS – 422 และ RS – 485 มาใช้แทนระบบRS – 232 เพื่อพัฒนาให้การรับส่งได้ไกลยิ่งขึ้น
2. รู้จักการใช้โปรแกรมเพื่อควบคุมการทำงานของคอมพิวเตอร์และ ไมโครคอนโทลเลอร์
3. รู้จักการแก้ไขปัญหาเฉพาะหน้า
4. สามารถนำระบบที่พัฒนาจากระบบเดิมมาประยุกต์ใช้ในการอำนวยความสะดวกและระบบการรักษาความปลอดภัยในงานลักษณะต่าง ๆ ได้

1.6 งบประมาณ

1. ค่าเอกสาร	150 บาท
2. ค่าอุปกรณ์ และ เครื่องมือ	1,050 บาท
3. ค่าใช้จ่ายอื่น ๆ เช่น ค่าจัดทำรูปเล่มโครงงาน	800 บาท
รวมเป็นเงินทั้งสิ้น	2,000 บาท

บทที่ 2
ทฤษฎีพื้นฐานของระบบควบคุมการเข้า-ออก อาคารและห้องทำงาน

2.1. สถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล 51

Single Chip Microcontroller System 51 Family Architectural

2.1.1 บทนำ

ไมโครคอนโทรเลอร์แบบชิพเดี่ยว คือ ไมโครคอมพิวเตอร์แบบที่มีขนาดเล็กโดยบรรจุไว้ในแผงวงจรรวมเพียงชิพเดี่ยวเหมาะสำหรับงานควบคุมอุปกรณ์อื่น ๆ แบบอัตโนมัติเพราะผู้ใช้สามารถเขียนโปรแกรมควบคุมการทำงานได้ตามต้องการ ไมโครคอนโทรเลอร์แบบชิพเดี่ยวตระกูล 51 หรือ MCS51 อันได้แก่ เบอร์ 8051, 8052 และอื่น ๆ

2.1.2 โครงสร้างของ AT 89C2051

ในโปรเจคนี้เราได้เลือก micro controller เบอร์ AT89C2051 มาซึ่งตัวนี้เป็น microcomputer แบบ low voltage, high-performance 8 bit กับความจำอย่างเดียวแบบ Flash Programmable และลบได้ขนาด 22 KByte มันเข้ากันได้กับ MSC 51 และสามารถ program โดยใช้ชุดคำสั่ง MCS – 51 มีลักษณะดังต่อไปนี้

- Compatible with Mcs-51 Product
- 2k Bytes of Reprogrammable Flash Memory
  - Endurance: 1000 Write/Erase Cycles
- 2.7 Volt to 6 Volt Operating range
- Fully Static Operation : 0Hz to 24 MHz
- Two level Program Memory Lock
- 128×8-bit Internal Ram
- 15 Programmable I/O Lines
- Two 16 –bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial UART Channel
- Direct LED Drive Outputs
- On-chip Analog Comparator

- Low power Idle And power down Mode

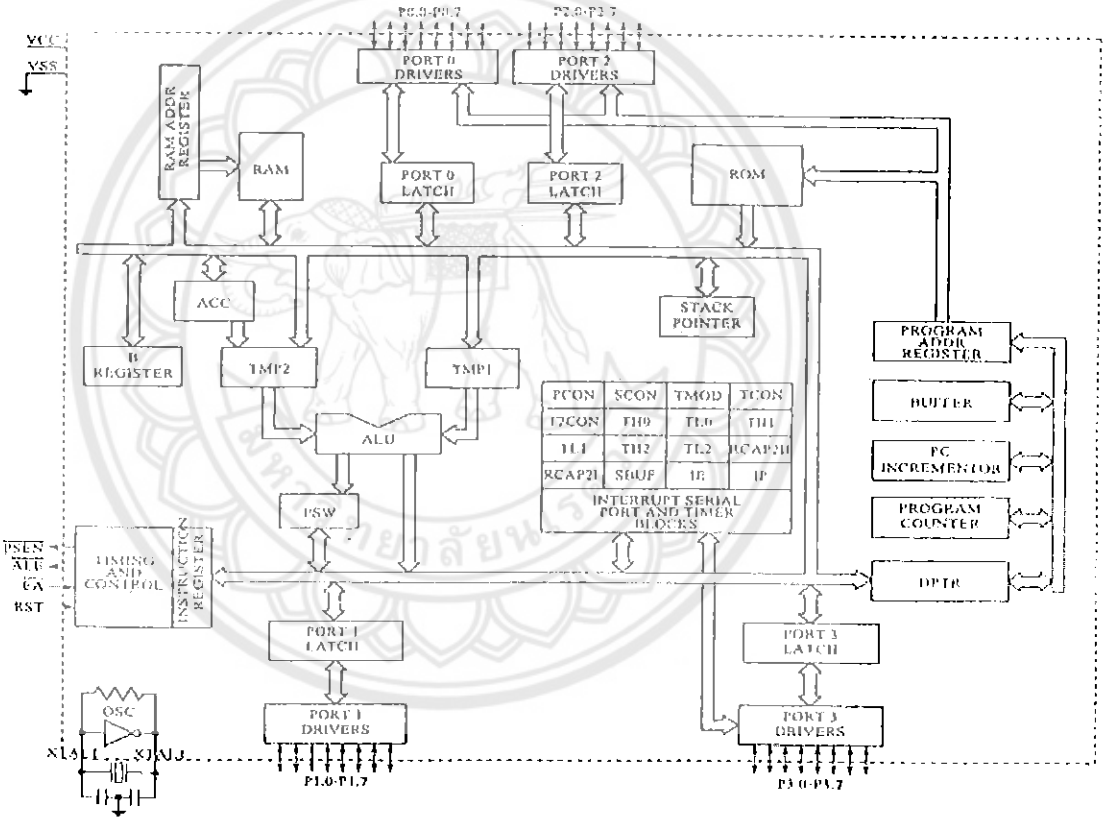
ซึ่งหลักการทำงานของMicro ตัวนี้ก็จะทำงานคล้ายกับการทำงานของ Micro ตัวอื่น ๆ ในที่นี้จะยกการทำงานของ 8051 มาเพื่อใช้ในการอ้างอิง

2.1.3 สถาปัตยกรรมของ 8051

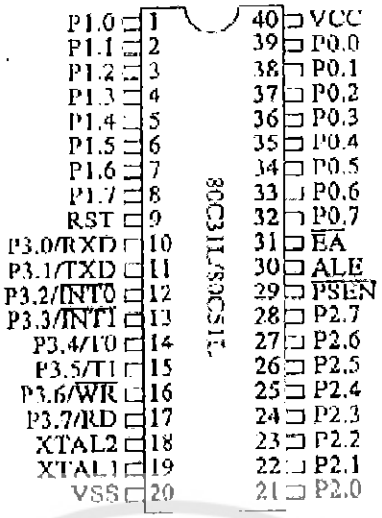
8051 ไมโครคอนโทรลเลอร์ที่บรรจุอยู่ในวงจรรวมแบบ Dual Inline Package (DPI) ซึ่งแต่ละข้างของ 8051 มีขาอย่าข้างละ 20 ขารวมทั้งหมด 40 ขานั้นจะใช้งานต่าง ๆ กันดังนี้คือ

VCC

ขา 40 เป็น ขาที่ต้องป้อนไฟเลี้ยง +5 V เข้าไปเพื่อให้วงจรรวมทำงานได้ระดับโวลเตจของลอจิก 0 และ 1 ของ 8051 จึงต่อเข้าอุปกรณ์ลอจิกแบบ TTL ได้โดยตรง



รูปที่ 2.1 สถาปัตยกรรมภายใน 8051



รูปที่ 2.2 ไดอะแกรมขาของ 8051 แบบ DIP

VSS

ขา 20 เป็นขาที่ต่อกับกราวด์ (Ground) ของแหล่งจ่ายไฟ การต่ออุปกรณ์ทั้งหมดจะต้องมีกราวด์ของอุปกรณ์ต่อเข้าด้วยกัน

Port

เป็นพอร์ทขนานขนาด 8 บิท อยู่ที่ขา 32 ถึง 39 เริ่มจากบิท 0 ถึง บิท 7 ตามลำดับ ดังในรูปที่ 2.4 แต่ละขาจะเขียนว่า PO.O, PO.1,.....PO.7 นั้น PO.7 หมายถึงบิท 7 ของพอร์ท 0 ซึ่งเป็นบิทที่มีนัยสำคัญสูงสุด (Most Significant) และ PO.O คือบิท 0 ของพอร์ท 0 เป็นบิทนัยสำคัญต่ำสุด (Least Significant) พอร์ท 0 นี้ใช้ได้ทั้งการรับ – ส่งตำแหน่งและข้อมูลกับหน่วยความจำหรือใช้เป็นพอร์ท รับ – ส่ง ข้อมูลก็ได้ ข้อมูลที่ส่งออกทางพอร์ท 0 จะถูก Latch ไว้ที่ขาของพอร์ท โครงสร้าง แต่ละบิทของพอร์ท 0 เป็นแบบ Open Drain Bidirectional

Port 1

เป็นพอร์ทขนานขนาด 8 บิท ในรูปที่ 2.2 คือขา P1.0 ถึง P1.7 (ขา 1-8) P1.0 หมายถึงบิท 0 ของพอร์ท 1 ซึ่งเป็นบิท Least Significant Bit และ บิท P1.7 หมายถึงบิทที่ 7 ของพอร์ท 1 ซึ่งเป็นบิท Most Significant Bit

Port 2

พอร์ทขนานขนาด 8 บิท คือ ขา P2.0 ถึง P2.7 (บิท 0 ถึงบิท 7 ของพอร์ท 2) ในรูปที่ 2.2 ลักษณะโครงสร้างจะเหมือนกับ Port 0 แตกต่างกันที่ใน Port 2 นั้นภาค Driver จะใช้งานเพียง 2 ลักษณะคือ

1. ใช้ส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อ ค่าตำแหน่งนี้เป็น 8 บิท บนของค่าตำแหน่ง
2. ใช้เป็นพอร์ทรับและส่งข้อมูลกับภายนอก

Port 3

คือขา P3.0 ถึง P3.7 หรือขา 10.17 ตามลำดับในรูปที่ 2.2 แต่ละบิทของพอร์ท 3 จะมีฟังก์ชั่นอื่น ดังนี้

P3.0/RXD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม

P3.1/TXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม

P3.2/INT0 (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

P3.4/INT1 (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

P3.4/T0 (Timer/Counter 0 External Input) ขารับสัญญาณเข้าไปยังวงจร Timer/Counter 0 ที่ทำหน้าที่นับจำนวน ไซเคิลของสัญญาณ T0 นี้หรือสัญญาณนาฬิกาก็ได้

P3.5/T1 (Timer/Counter1 External Input) ขารับสัญญาณเข้าไปยังวงจร Timer/Counter 1 ซึ่งมีทำงานเหมือนกับ T0

P3.6WR (External Data Memory Write Strobe) ขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 8051

P3.7/RD (External Data Memory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก

รายละเอียดการใช้งานแต่ละขาจะกล่าวต่อไปในหัวข้อ 2.2

RST

ขารีเซ็ทขานี้จะใช้ทำการรีเซ็ทการทำงานของ 8051 ที่ขา RST ภายใน 8051 จะมีตัวต้านทานต่อระหว่างขานี้กับกราวด์ (Ground) ถ้าป้อนสัญญาณที่มีสภาวะลอจิก 1 เข้าไปที่ขานี้จะเป็นการรีเซ็ทการทำงานของ 8051 ดังนั้นจึงสามารถต่อตัวเก็บประจุ (Capacitor) ภายนอกระหว่างขา RST กับไฟเลี้ยง +5 โวลท์ เพื่อให้เกิดการรีเซ็ทเมื่อเริ่มป้อนไฟเลี้ยงให้กับ 8051 ซึ่งเรียกว่า Power on reset การรีเซ็ทจะทำให้ค่าในรีจีสเตอร์ต่างๆ เปลี่ยนไปเป็นค่าหนึ่งดังในตารางที่ 2.2

ในตารางที่ 2.1 ช่องทางขวาเป็นค่ารีจีสเตอร์ที่อยู่ทางซ้ายเมื่อสิ้นสุดการรีเซ็ทในรีจีสเตอร์ SBUF เมื่อสิ้นสุดการรีเซ็ทจะมีค่าที่ไม่แน่นอน และพอร์ทจะอยู่ในสภาวะลอจิก 1 ทุกบิท ตลอดเวลาที่สัญญาณของขา RST เป็น HIGHT อยู่

ตารางที่ 2.1 ค่าของรีจิสเตอร์เมื่อเกิดการรีเซ็ท 8051

REGISTER	CONTENT
PC	0000H
ACC	00H
B	00H
PSW	00H
SP	00H
DPTR	00H
P0-P3	0X000000B
IP	00H
IE	00H
TMOD	00H
TCON	00H
T2CON	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
TH2	00H
TL2	00H
RCAP2H	00H
RCAP2L	00H
SCON	00H
SBUF	Indeterminate
IOCON	00H

เมื่อสัญญาณที่ขา RST กลับเป็น 0 ก็จะออกจากการรีเซ็ท 8051 จะเริ่มการทำงานจากคำสั่งที่อยู่ใน Program Memory ตำแหน่ง 0000H เพราะค่าของรีจีสเตอร์ (PC, Program Counter) ซึ่งใช้ชี้ตำแหน่งโปรแกรมที่จะทำงานถูกเปลี่ยนให้เป็น 0000H ดังนั้นผู้ใช้จะต้องเขียนโปรแกรมมาเก็บไว้ที่ตำแหน่ง 0000H ในเครื่องไมโครคอมพิวเตอร์แบบบอร์ดเดี่ยว(Single Board Microcomputer) จะมีโปรแกรมที่เขียนเก็บไว้เริ่มจากตำแหน่ง 0000H นี้เรียกว่า มอนิเตอร์โปรแกรม (Monitor Program) ที่จะคอยรับการกดแป้นพิมพ์ (Keyboard) และแสดงผลทางตัวแสดงผล (Display) แบบ 7 Segment

ALE

Address Latch Enable ขานี้จะส่งสัญญาณที่มีความถี่ 1.6 เท่าของสัญญาณนาฬิกาจากออสซิลเลเตอร์สัญญาณนี้จะส่งออกมาตลอดเวลายกเว้นบางครั้งของการติดต่อกับหน่วยความจำสำหรับข้อมูลภายนอก 8051 สัญญาณนี้จะใช้บอกกับอุปกรณ์ภายนอก 8051 ว่าขณะนี้สัญญาณนี้

Active (เป็นลอจิก 1) จะมีการส่งข้อมูลที่เป็น 8 บิทล่าง ของตำแหน่งหน่วยความจำภายนอก 8051 ที่ต้องการติดต่อออกไปทางพอร์ท 0 อุปกรณ์ภายนอกจะใช้สัญญาณนี้เป็นการ Latch ข้อมูลไว้เพราะพอร์ท 0 จะส่งค่าตำแหน่งหน่วยความจำออกมาเพียงชั่วขณะเท่านั้น ซึ่งในเวลาต่อมาพอร์ท 0 จะใช้ รับ-ส่งข้อมูลกับหน่วยความจำภายนอก สัญญาณ ALE จะสามารถต่อเข้ากับอุปกรณ์ TTL ชนิด LS ได้ถึง 8 อินพุท

PSEN

Program Store Enable เป็นขาที่ 29 ในรูปที่ 2.2 ขานี้ปกติจะให้ลอจิก 1 แต่จะส่งลอจิก0 เมื่อต้องการอ่านคำสั่ง (Fetch Instruction) ที่จะนำไปทำงานมาจากหน่วยความจำสำหรับโปรแกรมภายนอก 8051 ในกรณีที่อ่านคำสั่งซึ่งเก็บอยู่ในหน่วยความจำสำหรับโปรแกรมภายใน 8051 แล้วสัญญาณนี้จะไม่เปลี่ยนลอจิกเป็น 0 ขา PSEN นี้สามารถต่อไปยังขาอินพุทของ TTL ชนิด LS ได้ถึง 8 อินพุท

EA

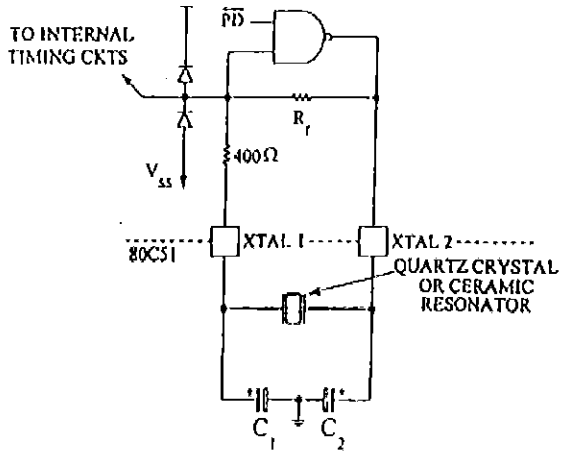
External Access ขา 31 ในรูปที่ 2.2 ขานี้เป็นขาอินพุทที่ต่อเข้าไปยังวงจร Timing and Control ในรูปที่ 2.2 เพื่อควบคุมการสร้างสัญญาณ PSEN ถ้าป้อนสัญญาณลอจิก 0 เข้าไปที่ขา EA นี้แสดงว่าโปรแกรมในตำแหน่ง 0000H ถึง 0FFFH ที่ต้องการให้ทำงานถูกเก็บไว้ภายนอก 8051 จะต้องสร้างสัญญาณ PSEN ออกไปยังภายนอก เพื่อทำการ FETCH คำสั่งเข้ามาทำงาน แต่ถ้าสัญญาณที่ป้อนให้ขา EA เป็น 1 หมายความว่าโปรแกรมในตำแหน่ง 0000H ถึง 0FFFH ถูกเก็บไว้ภายใน 8051 การทำงานในตำแหน่งหน่วยความจำช่วงนี้จะอ่านคำสั่งต่างๆ จาก ROM ภายใน 8051

XTAL 1

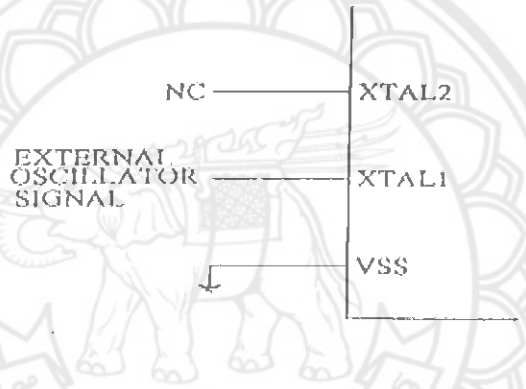
ขาที่ 19 ของรูปที่ 2.2 ขานี้จะต่อเข้ากับขาของ Inverting Amplifier (วงจรขยายแบบป้อนกลับเฟสสัญญาณ) ที่ประกอบเป็นวงจรออสซิลเลเตอร์ ในรูปที่ 2.3 จะเห็นวงจรภายในของออสซิลเลเตอร์ NAND Gate จะทำหน้าที่เป็นวงจรขยายแบบกลับเฟสของสัญญาณที่จะควบคุมให้มีการออกซิลเลตหรือไม่ก็ขึ้นอยู่กับสัญญาณ PD ซึ่งต่อมาจากบิท PD ของรีจีสเตอร์ PCON ถ้าต้องการใช้สัญญาณนาฬิกา (Clock Signal) จากภายนอกมาเป็นสัญญาณนาฬิกา การควบคุมการทำงานของ 8051 ก็ให้ป้อนสัญญาณเข้ามาที่จุดนี้ แต่ถ้าต้องการใช้วงจรออสซิลเลเตอร์ภายในก็ให้ต่อ Crystal หรือเซรามิคเรโซเนเตอร์ดังรูปที่ 2.3 คาปาซิเตอร์ในวงจรควรมีค่าประมาณ 20 pF

XTAL 2

ขาที่ 18 ของรูปที่ 2.2 ขานี้เป็นจุดเอาท์พุทของวงจรขยายแบบกลับเฟสสัญญาณที่ประกอบเป็นวงจรออสซิลเลเตอร์ (อินพุทคือขา XTAL 1) ถ้าจะใช้สัญญาณนาฬิกาที่สร้างมาจากภายนอกมาเป็นสัญญาณนาฬิกาของ 8051 แล้ว ให้ปล่อยขานี้ลอยไว้แล้วป้อนสัญญาณนาฬิกาจากภายนอกเข้ามาที่ขา XTAL 1 ดังรูปที่ 2.4



รูปที่ 2.3 วงจรออสซิลเลเตอร์ภายใน 8051



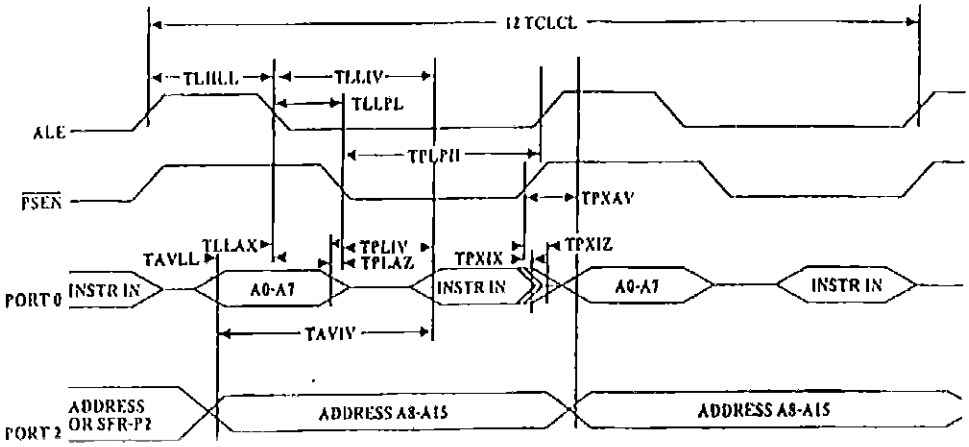
รูปที่ 2.4 8051 ที่ทำงานโดยสัญญาณที่มาจากภายนอก

1.1.4 ไดอะแกรมเวลาของการติดต่อกับหน่วยความจำ

การอ่านข้อมูลจากหน่วยความจำสำหรับโปรแกรมภายนอก 8051 นั้นลำดับสัญญาณตามเวลา (Timing Diagram) ของสัญญาณที่ทำการอ่านคำสั่งมีดังรูปที่ 2.5

การอ่านคำสั่ง (Fetch) จาก Program Area ภายนอกจะเริ่มจาก 8051 ส่งสัญญาณลอจิก 1 ออกมาทางขา ALE ขณะนี้สัญญาณ PSEN จะเป็น 1 จากนั้น Port 0 จะส่งค่าตำแหน่งหน่วยความจำ 8 บิทล่างและพอร์ท 2 จะส่งค่าตำแหน่งหน่วยความจำ 8 บิทบนออกมาแล้วสัญญาณ ALE จะกลับเป็น 0 อุปกรณ์ภายนอกจะสามารถใช้ขอบขาลงของสัญญาณ ALE เพื่อ Latch ตำแหน่งหน่วยความจำที่พอร์ท 0 ไว้ จากนั้นพอร์ท 0 ก็จะยกเลิกส่งค่าตำแหน่งหน่วยความจำเข้าสู่สภาวะ High Impedance และสัญญาณ PSEN จะเป็น 0 เพื่อเตรียมรับคำสั่งที่ส่งออกจากหน่วยความจำภายนอกเข้าไปยัง 8051 เพื่อทำงานต่อไป เมื่อคำสั่งถูกอ่านเข้าไปเก็บใน Instruction Register

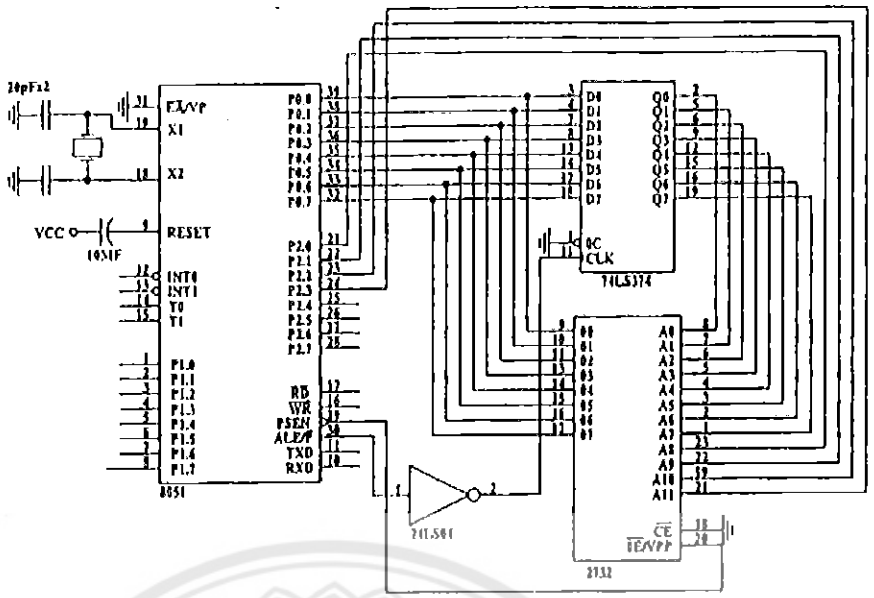
EXTERNAL PROGRAM MEMORY READ CYCLE



รูปที่ 2.5 Timing Diagram ของการอ่านโปรแกรมจากหน่วยความจำภายนอก

จากรูป 2.5 สัญญาณ PSEN จะกลับเป็น 1 พร้อมกับสัญญาณ ALE ก็จะกลับเป็น High เพื่อการอ่านคำสั่ง ต่อไปทำงาน ข้อมูลในพอร์ท 2 จะคงที่ตลอดเวลา ตั้งแต่สัญญาณ ALE เป็น 1 จนกระทั่งสัญญาณ ALE เปลี่ยนเป็น 0 และกลับเป็น 1 อีกครั้งหนึ่งจากนั้นจะเริ่มลำดับการ Fetch ข้อมูลไบท์ที่ 2 จากหน่วยความจำสำหรับโปรแกรม ซึ่งจะมีการเปลี่ยนแปลงของสัญญาณตามเวลาเหมือนกับการ Fetch ไบท์แรกนั่นเอง จาก Timing Diagram ดังกล่าวจะออกแบบวงจรที่มี Program Memory อยู่ภายนอก 8051 ได้ดังตัวอย่างในรูปที่ 2.6

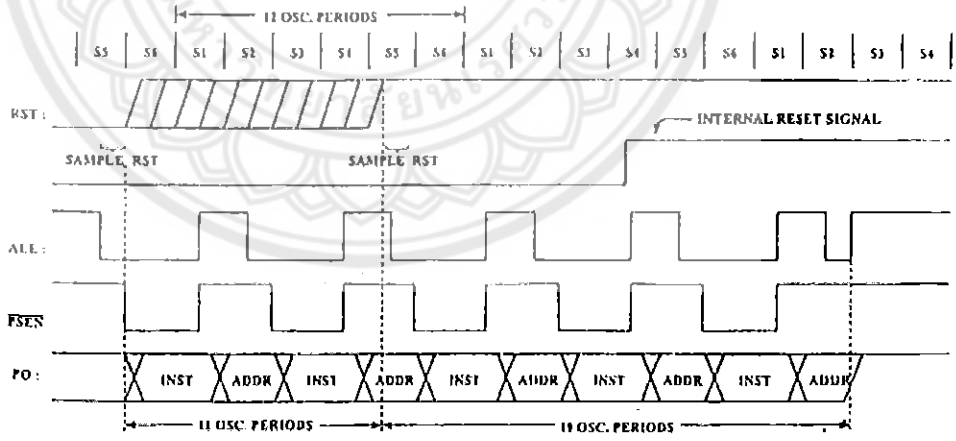
74LS374 ในรูปที่ 2.6 จะทำหน้าที่ Latch ตำแหน่งในหน่วยความจำ 8 บิทล่างที่เวลาขอบขาลงของสัญญาณ ALE ซึ่งสัญญาณ ALE จะถูกกลับให้เป็นตรงข้ามโดย Inverter 74LS04 ก่อนที่จะป้อนให้กับขา CK ของ 74LS374 และที่ขอบขาขึ้นของสัญญาณที่ออกจาก 74LS04 จะ Latch ตำแหน่งหน่วยความจำ ข้อมูลที่ออกจาก 74LS374 จะเป็นค่า 8 บิทล่างของตำแหน่ง หน่วยความจำที่ต้องการติดต่อ ในวงจรได้ต่อค่าตำแหน่งหน่วยความจำ 8 บิทเข้ากับ A0 ถึง A7 ของ EPROM และข้อมูลจากพอร์ท 2 บิท P2.0 ถึง P2.3 จะต่อเข้ากับ A8-A11 ของ EPROM โดยตรง เพราะค่าตำแหน่งหน่วยความจำ 8 บิทบนที่ออกมาจากพอร์ท 2 จะคงที่ตลอดเวลา ขา PSEN ของ 8051 จะถูกต่อเข้ากับขา OE ของ EPROM 2716 ดังนั้นเมื่อสัญญาณ PSEN มีสภาวะลอจิกเป็น 0 ก็จะส่งคำสั่งที่เก็บใน EPROM ณ ตำแหน่งที่ชี้โดยข้อมูลที่ขา A0 ถึง A11 ออกมายังพอร์ท 0 และถูก 8051 เก็บไปทำงานต่อไป



รูปที่ 2.6 วงจรที่มี Program memory อยู่ภายนอก 8051

2.1.5 การรีเซ็ท

เมื่อสัญญาณที่มีสภาวะลอจิก 1 เข้าไปทาง RST จะไม่ได้เกิดการรีเซ็ทขึ้นทันทีทันใด แต่ลำดับการรีเซ็ทจะแสดงได้ดังไดอะแกรมตามเวลาในรูปที่ 2.7



รูปที่ 2.7 ไดอะแกรมเวลาของการรีเซ็ท

ในรูปที่ 2.7 เป็น Timing Diagram ของการรีเซ็ทสภาวะลอจิกของสัญญาณที่ขา RST จะถูกอ่านเข้ามาที่เวลา S5P2 (เฟส 2 State 5) ของทุกๆ ไซเคิลของเครื่อง ในกรณีที่เป็นคำสั่งซึ่งมีการทำงานเสร็จสิ้นใน 2 ไซเคิลของเครื่อง ก็จะตรวจสอบเฉพาะสัญญาณที่อ่านเข้ามาในไซเคิลที่

2 ของการทำงาน ดังนั้นในการรีเซ็ทจะต้องป้อนสัญญาณที่มีสภาวะลอจิก 1 เข้าไปที่ ขานี้เป็นเวลาอย่างน้อย 2 ไซเคิลของเครื่องหรือ 24 ไซเคิลของสัญญาณนาฬิกาที่สร้างจากวงจร ออสซิลเลเตอร์ภายใน 8051 เพื่อให้แน่ใจว่าสัญญาณรีเซ็ทจะถูกอ่านเข้าไปตรวจสอบและทำงาน ขณะที่ทำการรีเซ็ท 8051 ออสซิลเลเตอร์จึงจะต้องทำงานอยู่ด้วย เมื่อ 8051 สุ่มข้อมูลที่ขา RST แล้วตรวจพบว่าเป็นสภาวะลอจิก 1 ก็จะสร้างสัญญาณรีเซ็ทขึ้นภายในที่เวลา S2P4 ของไซเคิลของเครื่องถัดไป ข้อมูลที่แต่ละพอร์ทส่งออกมาจะยังคงปรากฏที่พอร์ทจนกว่าจะเกิดการรีเซ็ทขึ้นซึ่งใช้เวลา 19 ไซเคิลของสัญญาณจากออสซิลเลเตอร์นับตั้งแต่เวลา S5P2 ในไซเคิลของเครื่องที่พบสัญญาณรีเซ็ท ในระหว่างเวลา 19 ไซเคิลนี้จะยังคงมีการ Fetch คำสั่งเข้าไปทำงานได้อยู่

สภาวะของสัญญาณลอจิกที่ขา RST จะถูกอ่านเข้าไปตรวจสอบที่เวลา S5P2 ของทุกๆ ไซเคิลของเครื่อง ดังนั้นถึงแม้ว่าสัญญาณที่ขา RST จะมีลอจิกเป็น 1 มาก่อนก็จะยังไม่เกิดการตรวจสอบสัญญาณรีเซ็ท ดังในรูปที่ 2.7 สัญญาณที่ขา RST จะมีลอจิกเป็น 1 มาตั้งแต่ State ที่ 6 ก็จะไม่เกิดอะไรขึ้นจนกระทั่ง 1 ไซเคิลของออสซิลเลเตอร์ซึ่งเป็น S5P2 จึงจะเกิดการตรวจสอบสัญญาณที่ขา RST ถ้าคำสั่งนั้นมีการทำงานมากกว่า 1 ไซเคิลของเครื่อง 8051 ก็จะต้องทำงาน ในคำสั่งนั้นให้เสร็จสิ้นเสียก่อนจึงจะเริ่มการรีเซ็ทได้โดย 8051 จะดูสภาวะของสัญญาณที่ขา RST ของ S5P2 ในไซเคิลของเครื่องสุดท้ายเท่านั้น ดังนั้นใน S5P2 ของไซเคิลเครื่องแรกๆ ในคำสั่ง อาจมีสภาวะลอจิกที่ขา RST เป็น 1 แต่ที่ S5P2 ของไซเคิลของเครื่องสุดท้ายมีสภาวะลอจิกที่ ขา RST เป็น 0 ก็จะไม่เกิดการรีเซ็ทขึ้นที่เวลา S5P2 เมื่อตรวจสอบสภาวะสัญญาณที่ขา RST แล้วพบว่าเป็น 1 จะต้องรอไปจนถึงเวลา S4P2 ของไซเคิลถัดไปจึงจะทำให้สัญญาณรีเซ็ทภายในเปลี่ยนสภาวะลอจิกจาก 0 เป็น 1 ในระหว่างเวลา S5P2 ที่ตรวจพบสัญญาณ RST มีลอจิกเป็น 1 จนกระทั่ง S4P2 ของไซเคิลของเครื่องถัดไปจะยังคงมีการ Fetch คำสั่งเข้าไปทำงานอีก 2 คำสั่ง เมื่อสัญญาณรีเซ็ทภายในเปลี่ยนเป็น 1 ก็จะเริ่มการรีเซ็ท โดยการเขียนข้อมูล 0 ไปยัง Special Function Register ทุกตัวยกเว้นพอร์ท 0 ถึงพอร์ท 3 Stack Pointer และรีจิสเตอร์ SBUF ดังตารางที่ 2.2 ระหว่างนี้ข้อมูลใน RAM ภายใน 8051 จะไม่เปลี่ยนแปลงข้อมูลในระหว่างการเขียนข้อมูลลงไปยัง SFR จะยังมีการ Fetch คำสั่งเข้ามาทำงานอีก 1 คำสั่งจนกว่าจะถึง S3P1 ของไซเคิลที่ 2 (นับแต่ไซเคิลของเครื่องที่ตรวจพบลอจิก 1 ที่ขา RST) ก็จะทำให้สภาวะลอจิกที่ขา ALE และ PSEN ค้างอยู่ที่สภาวะลอจิก 1 และจะเป็นอย่างนี้ไปจนกว่าสภาวะลอจิกที่ขา RST เป็น 0 เวลานับตั้งแต่พบสัญญาณลอจิก 1 ที่ขา RST ที่เวลา S5P2 จนถึงเวลา ALE และ PSEN ค้างอยู่ที่ 1 จะเท่ากับ 19 ไซเคิลของออสซิลเลเตอร์เมื่อสัญญาณขา RST ถูกเปลี่ยนกลับเป็นลอจิก 0 8051 จะรออีก 1 ถึง 2 ไซเคิลของเครื่อง สัญญาณ ALE และ PSEN จะเริ่มเปลี่ยนแปลงเพื่อเริ่มกระบวนการ Fetch คำสั่งเข้าไปทำงานเริ่มจากคำสั่งในหน่วยความจำสำหรับโปรแกรมตำแหน่ง 0000H

2.2 รีจีสเตอร์ฟังก์ชั่นพิเศษ (Special Function Register, SFR)

2.2.1 บทนำ

ใน 8051 จะใช้วิธีการกำหนดชื่อให้กับตำแหน่งของหน่วยความจำสำหรับข้อมูลภายใน (Internal Data Memory) ที่เรียกว่าการ Symbolize เช่น การให้ชื่อหน่วยความจำแต่ละตำแหน่งในแต่ละ Bank ซึ่งอยู่ช่วงหน่วยความจำตำแหน่ง 00H ถึง 1FH แล้วในคำสั่งจะอ้างอิงหน่วยความจำแต่ละตำแหน่งโดยการใช้ชื่อ R0, R1, R2, R3, R4, R5, R6 และ R7 หน่วยความจำตำแหน่งเหล่านี้จะเรียกอีกอย่างหนึ่งว่าเป็นรีจีสเตอร์ซึ่งมีหน้าที่ในการเก็บหรือพักข้อมูล หรือใช้สำหรับการกระทำบางอย่าง รีจีสเตอร์กลุ่มหนึ่งใน 8051 ที่เรียกว่า Special Function Register (SFR) เป็นรีจีสเตอร์ที่ใช้สำหรับงานเฉพาะ คือ ข้อมูลที่ถูกนำมาไปเก็บไว้ในรีจีสเตอร์เหล่านี้จะมีความหมายเฉพาะตัวของรีจีสเตอร์ ที่แต่ละตำแหน่งของ SFR อาจจะไม่ใช่เป็นหน่วยความจำ (RAM) แต่อาจเป็นตัวนับ (Counter Register), Shift Register หรือ Latch ซึ่งการอ้างอิงข้อมูลในแต่ละตำแหน่งนั้น 8051 จะถือเสมือนว่าเป็นหน่วยความจำตำแหน่งหนึ่ง จึงเรียกการมองข้อมูลแต่ละตำแหน่งนี้ว่า Memory Map I/O รีจีสเตอร์กลุ่มนี้มีดังในตารางที่ 2.2

ในตารางที่ 2.2 ช่อง Symbol ทางซ้ายจะเป็นสัญลักษณ์ของรีจีสเตอร์ในช่องถัดมา คือชื่อของรีจีสเตอร์ตามสัญลักษณ์ที่อยู่ทางซ้าย ในช่องขวาสุดจะเป็นตำแหน่งของหน่วยความจำสำหรับข้อมูลภายใน 8051 ที่แทนด้วยชื่อหรือสัญลักษณ์ทางซ้ายนั่นเอง เช่น ในบรรทัดแรก คือรีจีสเตอร์ชื่อ Accumulator ที่มีสัญลักษณ์ ACC รีจีสเตอร์นี้คือหน่วยความจำสำหรับข้อมูลภายใน 8051 ที่ตำแหน่ง 0E0H การอ่านหรือเขียนข้อมูลกับรีจีสเตอร์เหล่านี้สามารถทำได้โดยการใช้คำสั่งในกลุ่มการเคลื่อนย้ายข้อมูล (MOV A,#25H หรือ MOV 0E0H, #25H) และรีจีสเตอร์บางตัวในกลุ่มนี้ยังสามารถใช้คำสั่งกลุ่ม Boolean Instruction เพื่อการทำงานกับแต่ละบิทในรีจีสเตอร์เหล่านี้ได้ จากตารางที่ 2.2 รีจีสเตอร์ที่มีเครื่องหมาย * อยู่ข้างหน้าจะสามารถใช้คำสั่งในกลุ่ม Boolean Instruction จัดการกับแต่ละบิทได้ รีจีสเตอร์ที่มีเครื่องหมาย + นำหน้าหมายความว่ารีจีสเตอร์นั้นมีเฉพาะใน 80C52 และ 83C154 เท่านั้นไม่มีใน 8051

รูปที่ 2.8 ในช่องสี่เหลี่ยมเล็กๆ จะเป็นตำแหน่งของบิทนั้นในแต่ละรีจีสเตอร์ เช่น ในช่องซ้ายสุดของรีจีสเตอร์ TCON มีค่า 8FH ซึ่งเป็นค่าตำแหน่งบิท 7 ของหน่วยความจำตำแหน่ง 88H ถ้าต้องการให้บิทนี้มีค่าเป็น 0 ก็สามารถทำได้โดยใช้คำสั่ง

```
CLR 8FH
```

หรือจะทำให้บิทนี้เป็น 1 ก็ทำได้โดยใช้คำสั่ง

```
SETB 8FH
```

ตารางที่ 2.2 Special Function Register (SFR)

Symbol	Name	Address
*ACC	Accumulator	0E0H
*B	B Register	0FOH
*PSW	Program Status Word	0D0H
SP	Stack Pointer	81H
DPTR	Data Pointer 2 Bytes	
DPL	Low Byte	82H
DPH	High Byte	83H
*P0	Port 0	80H
*P1	Port 1	90H
*P2	Port 2	0A0H
*P3	Port 3	0B0H
*IP	Interrupt Priority Control	0B8H
*IE	Interrupt Enable Control	0A8H
TMOD	Timer/Counter Mode Control	89H
*TCON	Timer/ Counter Control	88H
*+T2CON	Timer/ Counter2 Control	0C8H
TH0	Timer/ Counter 0 High Byte	8CH
TL0	Timer/ Counter 0 Low Byte	8AH
TH1	Timer/ Counter 1 High Byte	8DH
TL1	Timer/ Counter 1 Low Byte	8BH
+TH2	Timer/ Counter 2 High Byte	0CDH
+TL2	Timer/ Counter 2 Low Byte	0CCH
+RCAP2H	T/C 2 Capture Reg. High Byte	0CBH
Symbol	Name	Address
+RCAP2L	T/C 2 Capture Reg. Low Byte	0CAH
*SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H
*IOCON (1)	IO Control	F8H

+80C52 and 83C154 only

(1)83C154 only

2.2.2 Special Function Register

รีจีสเตอร์ในกลุ่ม Special Function Register มีดังนี้

Accumulator ตำแหน่งหน่วยความจำภายในเท่ากับ 0EOH

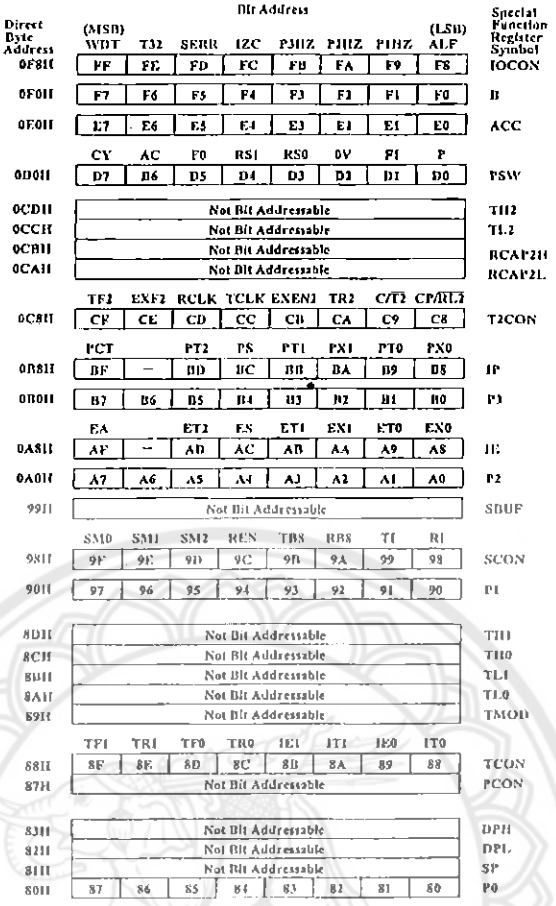
รีจีสเตอร์นี้มีขนาด 8 บิท เป็นรีจีสเตอร์ที่ใช้กันมาก ซึ่งในรหัสคำสั่งช่วยจำจะอ้างอิงถึงรีจีสเตอร์นี้โดยใช้สัญลักษณ์ A เช่น MOV A,#15H คำสั่งที่จะอ่านหรือเก็บข้อมูลกับหน่วยความจำภายนอกจะต้องกระทำผ่านรีจีสเตอร์นี้เท่านั้น เช่น MOVX @R0,A หรือ MOVX A,@R0 เป็นต้น และข้อมูลที่อยู่ภายในรีจีสเตอร์นี้ก็สามารถที่จะให้โปรแกรมตรวจสอบเพื่อกระโดดการทำงานไปยังตำแหน่งอื่นได้เช่น JZ rel

B Register ตำแหน่งหน่วยความจำภายในเท่ากับ 0F0H

เป็นรีจีสเตอร์ขนาด 8 บิท ที่ใช้ในคำสั่งการคูณ (MUL AB) และคำสั่งการหาร (DIV AB) เท่านั้นโดยรีจีสเตอร์ B นี้จะเก็บตัวคูณและผลลัพธ์บิท 8 ถึง 15 ในคำสั่งการคูณ ส่วนในคำสั่งหารนั้น โดยรีจีสเตอร์ B จะเก็บตัวหารและผลการหาร การเขียนข้อมูลไปยังรีจีสเตอร์นี้จะต้องใช้คำสั่งเคลื่อนย้ายข้อมูลไปยังตำแหน่ง 0F0H เช่น MOV 0F0H,25H จะเป็นการกำหนดค่า 25H ให้กับรีจีสเตอร์ B

Program Status Word ตำแหน่งหน่วยความจำภายในเท่ากับ 0D0H

เป็นรีจีสเตอร์ขนาด 8 บิท ที่แต่ละบิทจะบอกสภาวะต่างๆ แต่ละบิทของ PSW จะสามารถกำหนดให้เป็น 1 หรือ 0 ได้ด้วยคำสั่ง SETB หรือ CLR ตามลำดับ ค่าตำแหน่งบิท 0 ถึงบิท 7 ของรีจีสเตอร์ PSW เท่ากับ 0DH ถึง D7H



รูปที่ 2.8 แผนภาพค่าตำแหน่งหน่วยความจำแต่ละบิท

Stack Pointer ตำแหน่งหน่วยความจำภายในเท่ากับ 081H

เป็นรีจีสเตอร์ขนาด 8 บิท รีจีสเตอร์นี้จะใช้ชี้ตำแหน่งหน่วยความจำภายใน 8051 ที่ใช้เก็บตำแหน่ง (Address) เดิมของโปรแกรมทำงานคำสั่ง CALL หรือตำแหน่งที่จะใช้เก็บข้อมูลด้วยคำสั่ง PUSH และตำแหน่งที่จะอ่านข้อมูลออกมาในคำสั่ง POP เมื่อทำการรีเซ็ท 8051 โดยการป้อนสัญญาณสภาวะลอจิก 1 เข้าไปที่ขา RST ของ 8051 จะทำให้ข้อมูลในรีจีสเตอร์นี้มีค่าเป็น 07H หมายความว่ารีจีสเตอร์ SP ชี้หน่วยความจำภายใน 8051 ที่ตำแหน่ง 07H ค่าของ SP จะเปลี่ยนแปลงไปโดยการใช้คำสั่งเคลื่อนย้ายข้อมูลหรือการทำงานของคำสั่ง PUSH, POP และ CALL

Data Pointer Register ตำแหน่งหน่วยความจำภายในเท่ากับ 82H และ 83H

รีจีสเตอร์ DPTR มีขนาด 16 บิท หน้าที่ของรีจีสเตอร์นี้ใช้สำหรับชี้ตำแหน่งในหน่วยความจำรีจีสเตอร์ DPTR นี้สามารถใช้อ้างอิงตำแหน่งหน่วยความจำได้สูงสุด 60 x 1024 ตำแหน่ง เช่นคำสั่ง MOVX A,@DPTR หรือใช้ชี้ตำแหน่งโปรแกรมที่ต้องการกระโดดข้ามไปทำงาน เช่นคำสั่ง JMP @A+DPTR รีจีสเตอร์ DPTR นี้ประกอบด้วยรีจีสเตอร์ขนาด 8 บิท 2 ตัวคือ DPH

ซึ่งอยู่ที่ตำแหน่ง 83H และ DPL ซึ่งที่ตำแหน่ง 82H ในหน่วยความจำสำหรับข้อมูลภายใน 8051 ดังนั้นการแก้ไขข้อมูลในรีจีสเตอร์ DPTR จึงทำได้ทั้งทีละ 16 บิท เช่น คำสั่ง MOV DPTR,#data 16 หรือจัดการทีละ 8 บิท โดยการแก้ไขข้อมูลใน DPH หรือ DPL ด้วยคำสั่ง MOV 83H,#data 8 หรือ MOV 82H,#data 8

PORT 0 ถึง 3 ตำแหน่งหน่วยความจำภายในเท่ากับ 80H, 90H, 0A0H, 0B0H

Special Function Register ชื่อ P0, P1, P2 และ P3 เป็นรีจีสเตอร์ขนาด 8 บิทของหน่วยความจำสำหรับข้อมูลภายใน 8051 ที่ตำแหน่ง 80H, 90H, 0A0H และ 0B0H ตามลำดับ การเขียนข้อมูลลงไปยังหน่วยความจำแต่ละตำแหน่งเป็นการส่งข้อมูลไปยังพอร์ทนั้นๆ ของ 8051 ข้อมูลที่เขียนออกไปจะถูก LATCH ค้างไว้และปรากฏที่บิทของพอร์ท เช่น MOV 80H, #18H จะ ปรากฏสภาวะลอจิก LLHHLLL ที่ขาบิท 7 ถึง 0 ของพอร์ท 0 ตามลำดับในการอ่านข้อมูลจากรีจีสเตอร์แต่ละตัวก็จะเป็นการอ่านสภาวะลอจิกของสัญญาณที่ปรากฏอยู่ที่แต่ละขาของพอร์ทนั้นๆ เช่น MOV A,80H เป็นการอ่านสภาวะลอจิกจากพอร์ท 0 เข้ามายัง Accumulator การอ่านข้อมูลจากพอร์ทจะต้องเขียนข้อมูล 11111111B ไปไว้ที่พอร์ทนั้นๆ เสียก่อน ทุกบิทของพอร์ท 0 ถึง 3 จะสามารถแก้ไขเปลี่ยนแปลงได้โดยใช้คำสั่ง SETB bit และ CLR bit

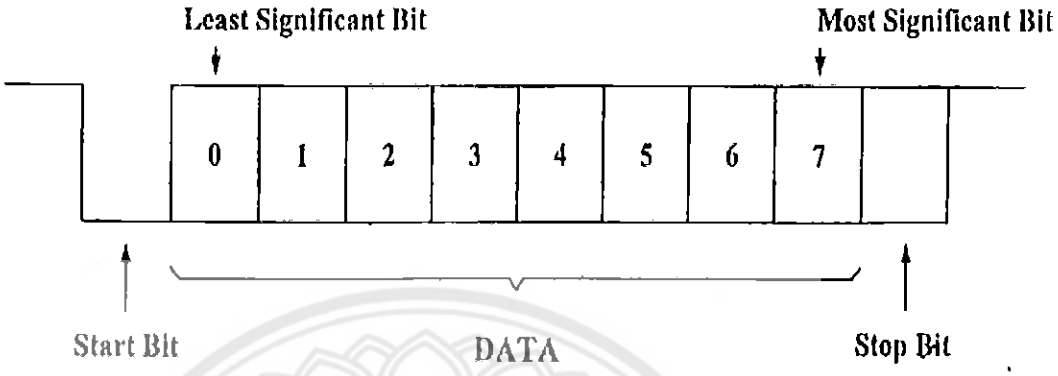
Serial Data Buffer ตำแหน่งหน่วยความจำภายในเท่ากับ 99H

รีจีสเตอร์นี้มีขนาด 8 บิทและมีตำแหน่งของหน่วยความจำสำหรับข้อมูลภายใน 8051 เท่ากับ 99H โครงสร้างภายในแล้วรีจีสเตอร์นี้มี 2 ตัวที่มีชื่อเดียวกัน ตัวหนึ่งสำหรับเก็บข้อมูลที่จะส่งแบบอนุกรมออกจาก 8051 และอีกตัวหนึ่งสำหรับรับข้อมูลแบบอนุกรมที่เข้ามา ดังนั้น Serial Port ของ 8051 จึงเรียกว่ามีการทำงานแบบ Full Duplex เพราะสามารถส่งและรับข้อมูลได้ในเวลาเดียวกันเนื่องจากรีจีสเตอร์สำหรับส่งและรับแยกออกจากกัน ข้อมูลที่ต้องการจะส่งออกก็ให้เขียนไปยังรีจีสเตอร์ SBUF แล้วสั่งงานให้ส่งข้อมูลออกมา ข้อมูลในรีจีสเตอร์จะเริ่มส่งออกโดยเริ่มจากบิท 0 ถึง 7 ตามลำดับ ถ้าข้อมูลมีข้อมูลเข้ามาทางขา RXD ถูกเก็บไว้ในรีจีสเตอร์นี้ โดยถือว่าข้อมูลบิทแรกที่เข้ามาคือ บิท 0

Serial Port จะสามารถกำหนดการรับ-ส่งข้อมูลแบบอนุกรมได้ 4 โหมด (MODE) โดยการกำหนดในรีจีสเตอร์ SCON (Serial Port Control Register) แต่ละโหมดการทำงานของ Serial Port มีดังนี้

MODE 0 : ในโหมดนี้จะมีการรับหรือส่งข้อมูลแบบอนุกรมทางขา RXD และ TXD จะส่งสัญญาณ Clock ที่ใช้สำหรับเลื่อน (Shift) ข้อมูล 1 ชุดของข้อมูลจะประกอบด้วยข้อมูล 8 บิทเท่านั้น และจะเริ่มการรับ – ส่งข้อมูลจากบิท 0 จนถึงบิท 7 ตามลำดับ อัตราการส่งข้อมูลแบบอนุกรมจะเท่ากับ 1/12 เท่าของความถี่สัญญาณนาฬิกาที่ใช้กับ 8051

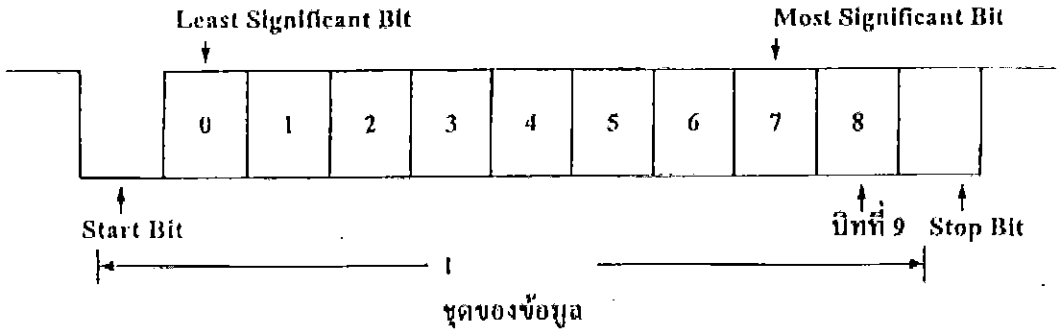
MODE 1 : ข้อมูลที่รับ-ส่ง 1 ชุดในโหมดนี้จะมี 10 บิท ผ่านทางขา RXD และ TXD ตามลำดับ เริ่มต้นการรับส่งข้อมูลด้วย Start bit 1 บิท (ลอจิกเป็น 0), ข้อมูล 8 บิท (เริ่มจากบิท 0), Stop bit 1 บิท (ลอจิกเป็น 1) การส่งข้อมูลโหมดนี้มีดังรูปที่ 2.9



รูปที่ 2.9 ชุดข้อมูลอนุกรมในโหมด 1

เมื่อรับข้อมูลอนุกรมเข้ามาข้อมูล 8 บิท จะถูกเก็บไว้ในรีจีสเตอร์ SBUF และ Stop Bit จะถูกเก็บไปที่บิท RB8 ในรีจีสเตอร์ SCON ในการส่งข้อมูลออกก็จะเขียนข้อมูลที่ต้องการส่งไปยังรีจีสเตอร์ SBUF อัตราการส่งข้อมูลในโหมดนี้สามารถกำหนดได้ตามต้องการโดยจะขึ้นกับการเกิด Overflow ใน Timer 1

MODE 2 : การรับ-ส่งข้อมูลของโหมด 2 1 ชุดจะมี 11 บิท ข้อมูลจะส่งออกผ่านทางขา TXD และรับเข้ามาทางขา RXD ข้อมูลแต่ละชุดจะเริ่มต้นด้วย Start bit 1 บิท, ข้อมูล 8 บิท (เริ่มจากบิท 0), ข้อมูลบิทที่ 9 จำนวน 1 บิท และ Stop bit อีก 1 บิท ข้อมูลบิทที่ 9 ที่จะส่งออกนี้สามารถกำหนดได้ว่าจะให้เป็น 1 หรือ 0 โดยการกำหนดในบิท RB8 ของรีจีสเตอร์ SCON บิทนี้มีประโยชน์มากในการส่งข้อมูลแบบอนุกรมเช่นอาจส่งค่าพาริตี้ของข้อมูลไปเป็นบิทที่ 9 เพื่อว่าเมื่อปลายทางรับข้อมูลแล้วจะได้ใช้ตรวจสอบว่าข้อมูลที่รับเข้ามา 8 บิทมีพาริตี้บิทตรงกับบิทที่ 9 หรือไม่ ถ้าไม่ตรงก็แสดงว่ามีข้อผิดพลาดเกิดขึ้นระหว่างการส่งข้อมูล เมื่อรับข้อมูลเข้ามานั้นข้อมูลบิทที่ 9 ก็จะถูกนำไปเก็บในบิท RB8 ของรีจีสเตอร์ SCON ชุดข้อมูลที่รับ-ส่งจะมีดังรูปที่ 2.10



รูปที่ 2.10 ชุดข้อมูลอนุกรมในโหมด 2

อัตราการส่งข้อมูลจะกำหนดให้เป็น 1/32 หรือ 1/64 เท่าของความถี่สัญญาณนาฬิกาที่ใช้กับ 8051 โดยการกำหนดบิท SMOD ในรีจีสเตอร์ PCON

MODE 3 : การส่งข้อมูลในโหมดนี้ 1 ชุดมี 11 บิท เหมือนกับโหมด 2 ทุกประการ แตกต่างกันตรงอัตราการส่งข้อมูลเท่านั้น คือ อัตราการส่งข้อมูลในโหมด 3 นี้สามารถกำหนดได้ตามต้องการ โดยจะขึ้นอยู่กับการเกิด Overflow ใน Timer 1 เหมือนกับโหมด 1

SCON (Serial Port Control Register) ตำแหน่งหน่วยความจำภายในเท่ากับ 98H

รีจีสเตอร์ SCON มีขนาด 8 บิท ใช้สำหรับควบคุมการส่งและรับข้อมูลผ่านทาง Serial Port แต่ละบิทของข้อมูลในรีจีสเตอร์นี้มีความหมายเฉพาะดังตารางที่ 2.3

ตารางที่ 2.3 Serial Port Control Register (SCON)

SCON : SERIAL PORT CONTROL REGISTER. BIT ADDRESSABLE.

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

RM0	SCON.7	Serial Port mode specifer. (NOTE1).
SM1	SCON.6	Serial Port mode specifer. (NOTE1).
SM2	SCON.5	Enable the multiprocessor communication feature in mode 2&3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2=1 then RI will not be activated if a valid stop was not received. In mode 0, SM2 should be 0. (See Table 9).

REN	SCON.4	Set/Cleared by software to Enable/Disable reception.
TB8	SCON.3	The 9th that will be transmitted in mode 2&3.Set/Cleared by software.
RB8	SCON.2	In mode 2&3, is the 9th data bit that was received. In mode 1, if SM2=0, RB8 is the stop that was received. In mode 0, RB8 is not used.
TI	SCON.1	Transmit interrupt flag. Sea by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other mode. Must be cleared by software.
RI	SCON.0	Receive interrupt flag. Set by hardware at the end the 8th bit time in mode 0, or halfwaythrough the stop bit time in the order modes (except see SM2). Must be cleared by software.

SM0	SM1	Mode	Description	Baud Rate
0	0	0	SHIFT REGISTER	Fosc./12
0	1	1	8-Bit UART	Variable
1	0	2	9-Bit UART	Fosc./64 OR Fosc./32
1	1	3	9-Bit UART	Variable

MODE	SCON	SM2 VARIATION
0	10H	Single Processor
1	50H	Environment
2	90H	(SM2 = 0)
3	D0H	
0	NA	Multiprocessor
1	70H	Environment
2	B0H	(SM2 = 0)
3	F0H	

TIMER Resister TH0, TL0, THI, TLI

ตำแหน่งหน่วยความจำภายในเท่ากับ 8CH, 8AH, 8DH, 8BH

ใน 8051 จะมีวงจร Timer อยู่ 2 ชุด คือ Timer 0 และ Timer 1 (8052 จะมี Timer อีก 1 ชุด) ใน Timer แต่ละชุดจะมีรีจีสเตอร์ขนาด 8 บิท อยู่ 2 ตัว เพื่อเก็บค่าการนับของ Timer ได้สูงสุดถึง 16 บิทใน Timer0 รีจีสเตอร์นี้คือ TH0, TL0 และใน Timer 1 คือ รีจีสเตอร์ TH1, TL1 TLx (x หมายถึง 0 หรือ 1) จะเก็บค่าของการนับ 8 บิทล่างและ THx จะเก็บค่าของการนับ 8 บิท บน ผู้ใช้จะสามารถกำหนดการทำงานของวงจร Timer ในโหมด Timer หรือโหมด Counter ได้โดยการกำหนดในรีจีสเตอร์ชื่อ TMOD (Time/Counter Mode Control Rehister) การทำงานเป็น Timer นั้นจะให้รีจิสเตอร์ใน Timer 0 หรือ 1 ทำการนับจำนวนไซเคิล (Cycle) ของสัญญาณนาฬิกาบางบิทในรีจีสเตอร์ TCON เพื่อบอกสภาวะ Timer Overflow นี้ ในการให้วงจร Timer ทำงานเป็น Counter ก็คือการใช้รีจีสเตอร์ TLx และ THx ทำการนับจำนวนไซเคิลของสัญญาณที่เข้ามาทางขา T0 หรือ T1 ของ 8051 สัญญาณที่เข้ามาทางขา T0 หรือ T1 อาจจะมาจากอุปกรณ์ตรวจจับ (Sensor) ก็ได้ แต่สภาวะของสัญญาณนี้จะต้องมีระดับโวลเตจของสภาวะลอจิก 0 หรือ 1 เป็นแบบ TTL คือลอจิก 0 จะต้องมีโวลเตจไม่เกิน 0.6 โวลท์ และลอจิก 1 จะต้องมีโวลเตจมากกว่า 2.4 โวลท์

TMOD Timer/Counter mode register

ตำแหน่งหน่วยความจำภายในเท่ากับ 89H

TMOD เป็นรีจีสเตอร์ขนาด 8 บิท มีหน้าที่ควบคุมการทำงานของ Timer 0 และ Timer 1 แต่ละบิทในรีจีสเตอร์นี้มีความหมายเฉพาะดังตารางที่ 2.4

ตารางที่ 2.4 TMOD Time/Counter Mode Register

NOTE 1 :

M0	M1	Operating Mode	
0	0	0	13-bit Timer
0	1	1	16-bit Timer/Counter
1	0	2	8-bit Auto-Reload Timer/Counter
1	1	3	(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits, TH0 is an 8-bit Timer and is controlled by Timer 1 control bits.
1	1	3	(Timer1) Timer/Counter 1 stopped.

ในรูปที่ 2.1 M0 เป็นชื่อของบิท 0 และ GATE ทางซ้ายสุดเป็นชื่อของบิท 7 รีจีสเตอร์นี้แบ่งข้อมูลออกเป็น 2 ชุด ชุดละ 4 บิท คือ บิท 0-3 ใช้สำหรับควบคุมการทำงานของ Timer 0 และบิท 4-7 ใช้ควบคุมการทำงานของ Timer 1 หน้าที่ในการควบคุม Timer ของแต่ละบิทที่มีชื่อเดียวกันจะเหมือนกัน

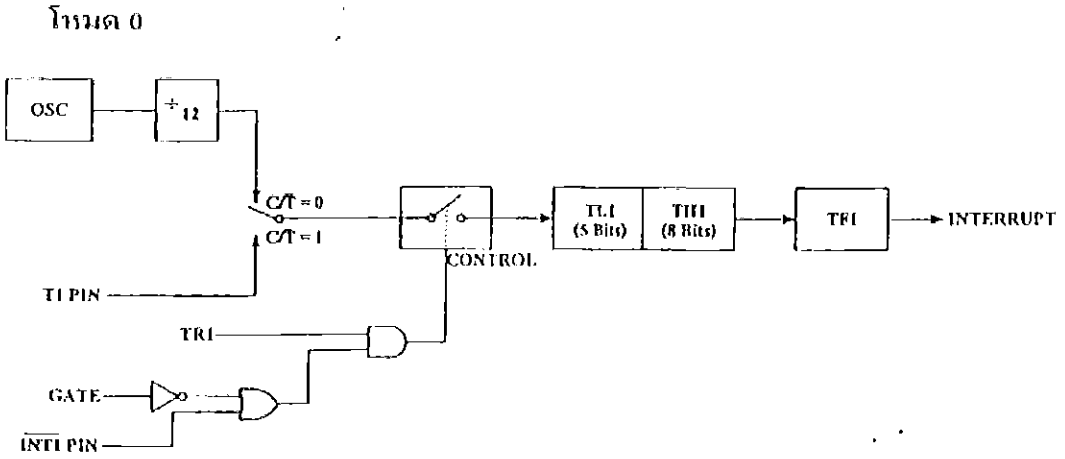
GATE เป็นบิทที่ใช้ควบคุมให้ Timer ทำงานหรือไม่ ถ้าบิทนี้ของ Timer x ถูกตั้งเป็น 1 จะทำให้ Timer ทำงานก็ต่อเมื่อที่ขา INTx มีสภาวะลอจิกเป็น 1 และบิท TRx ในรีจีสเตอร์ TCON เป็น 1 ด้วย

C/T บิทนี้ใช้สำหรับเลือกการทำงานของ Timer ว่าจะใช้เป็น Timer หรือ Counter ถ้าบิทนี้เป็น 1 ก็หมายความว่าเลือกการทำงานเป็น Counter ซึ่งจะนับจำนวนไซเคิลของสัญญาณที่เข้ามาทางขา Tx

M1, M0 เป็น 2 บิท ที่ใช้ร่วมกันเพื่อเลือกโหมดการทำงานของ Timer การทำงานโหมด 0, 1 และ 2 ของ Timer 0 จะเหมือนกับ Timer 1 แต่ในโหมด 3 การทำงานของทั้งสองจะต่างกัน ค่าใน M1 และ M0 จะเลือกโหมดการทำงานดังนี้

M1	M0	การทำงาน
0	0	โหมด 0 รีจีสเตอร์ THx และ TLx ทำตัวเป็นตัวนับ 13 บิท ค่าการนับ 8 บิทบนมาจาก 8 บิทของ THx และอีก 5 บิทล่างมาจากค่า 5 บิทล่างของรีจีสเตอร์ TLx โดยที่ 3 บิทบนของ TLx จะไม่ต้องสนใจเลย
0	1	โหมด 1 รีจีสเตอร์ THx และ TLx ทำตัวเป็นตัวนับ 16 บิทค่าจากการนับ 8 บิทบนอยู่ในรีจีสเตอร์ THx และค่าจากการนับ 8 บิทล่างอยู่ในรีจีสเตอร์ TLx
1	0	โหมด 2 ในการนับของรีจีสเตอร์ TLx ขนาด 8 บิทเมื่อนับถึงค่าสูงสุดคือ FFH เมื่อทำการนับต่อไปจะเกิดการ Overflow แล้วก็จะ "Reload" เอาข้อมูลจาก THx เข้าไปยัง TLx เพื่อเป็นค่าเริ่มต้นในการนับครั้งใหม่
1	1	โหมด 3 การทำงานของ Timer 0 และ Timer 1 จะต่างกันดังที่จะกล่าวต่อไป

การทำงานของแต่ละโหมดจะมีรายละเอียดดังนี้



รูปที่ 2.11 Timer mode 0 : 13 bit count

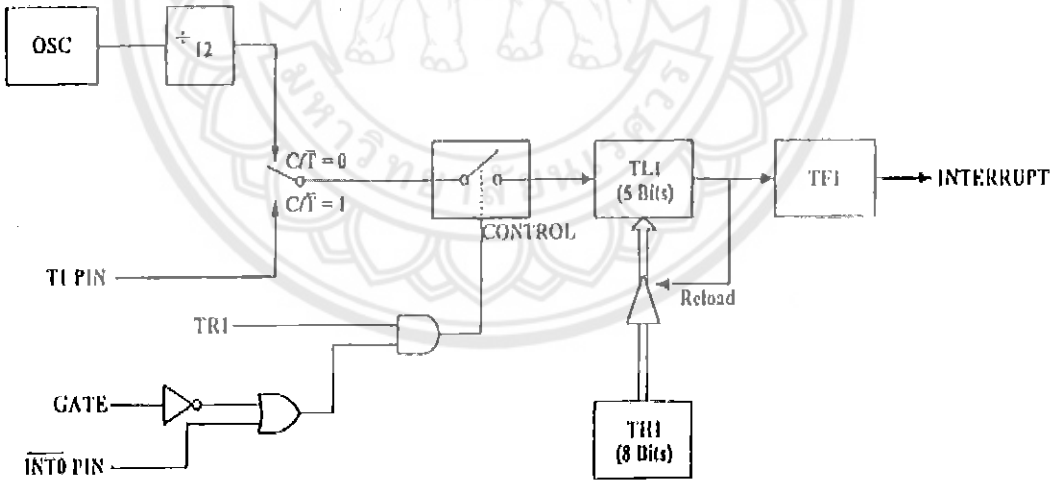
รูปที่ 2.13 เป็นไดอะแกรมของวงจร Timer ภายใน 8051 ที่ทำงานในโหมด 0 ซึ่ง Timer 0 และ Time 1 ก็จะมีการทำงานเหมือนกันทุกประการ ในการอธิบายนี้จะใช้วงจร Timer 1 จากรูปจะเห็นสวิทช์ C/T ซึ่งถ้ากำหนดค่าในบิท C/T ของ TMOD เป็น 0 จะทำให้สวิทช์อยู่ในตำแหน่งบน เพื่อให้สัญญาณนาฬิกาที่ออกจากวงจรออสซิลเลเตอร์ผ่านวงจรหาร 12 ไปยังสวิทช์ Control ถ้าออสซิลเลเตอร์ผลิตสัญญาณนาฬิกาความถี่ 12 MHz ก็จะมีสัญญาณความถี่ 1 MHz ออกจากวงจรหาร 12 ถ้าบิท C/T เป็น 1 จะทำให้สวิทช์ C/T อยู่ในตำแหน่งข้างล่าง เพื่อให้สัญญาณที่เข้ามาทาง T1 (หรือ T0 ถ้าเป็น Timer0) ผ่านไปยังสวิทช์ Control สัญญาณที่เข้ามายังสวิทช์ Control จะส่งผ่านไปยังวงจรนับหรือไม่ก็ขึ้นอยู่กับสัญญาณควบคุมที่ออกมาจาก AND GATE ถ้าบิท TR1 (หรือ TR0 ถ้าเป็น Timer 0) ในรีจีสเตอร์ TCON เป็น 0 จะทำให้สภาวะของสัญญาณที่ออกจาก AND GATE เป็น 0 เสมอ และจะไม่มีสัญญาณใดออกจากสวิทช์ Control ไปยังวงจรนับเลย รีจีสเตอร์ TL1 และ TH1 จะไม่ทำงาน แต่ถ้าบิท TR1 เป็น 1 จะทำให้สภาวะของสัญญาณออกจาก AND GATE ไปควบคุมสวิทช์ Control ขึ้นกับสภาวะ INT1 (หรือ INT0 ถ้าเป็น Timer 0) และข้อมูลที่บิท GATE ของรีจีสเตอร์ TMOD ถ้าบิท GATE เป็น 0 หรือสัญญาณที่ขา INT1 มีสภาวะลอจิกเป็น 1 จะทำให้สัญญาณควบคุมสวิทช์ Control เป็น 1 ทำให้มีสัญญาณออกไปยังตัวนับรีจีสเตอร์ TL1 และ TH1 (หรือ TL0 และ TH0 ถ้าเป็น Timer 0) รีจีสเตอร์ TL1 จะทำการนับโดยมีการนับเพียง 5 บิทเท่านั้น (ทำหน้าที่เป็นวงจร Prescaler ขนาด 5 บิท) ซึ่งนับได้ตั้งแต่ 0-31 เมื่อ TL1 นับสัญญาณที่ออกจากสวิทช์ Control ครบ 32 ไซเคิล จะมีสัญญาณส่งไปยัง TH1 หนึ่งไซเคิลบิท 5-7 ของ TLx ที่ไม่ได้ใช้งาน ก็จะไม่ต้องสนใจการทำงานของ Timer 0 และ 1 ในโหมดนี้เหมือนกับการทำงานของ Timer ในไมโครคอนโทรลเลอร์เบอร์ 8048 ทุกประการ

โหมด 1

ในโหมดนี้จะมีการทำงานของสงจรภายในของ Timer 0 หรือ 1 เหมือนกับโหมด 0 ทุกประการ แตกต่างกันที่ TLx จะถูกใช้งานทั้ง 8 บิท ทำให้ผลการนับใน TLx และ THx จะมีถึง 16 บิท

โหมด 2

ในรูปที่ 2.12 เป็นไดอะแกรมของวงจร Timer 1 ใน 8051 ที่ทำงานโหมด 2 Timer 0 และ Timer 1 มีการทำงานในโหมด 2 เหมือนกันโดยจะสามารถกำหนดให้ทำหน้าที่เป็น Timer หรือ Control ได้โดยบิท C/T และควบคุมการนับได้โดยข้อมูลในบิท TR1 และ GATE ในรีจีสเตอร์ TMOD กับสัญญาณที่เข้า INTx เมื่อเริ่มการทำงาน ข้อมูลในรีจีสเตอร์ TH1 จะถูกโหลด (Load) ไปยังรีจีสเตอร์ TL1 ทำให้รีจีสเตอร์ TH1 และ TL1 มีค่าเหมือนกันเมื่อเกิดการนับจำนวนไซเคิลของสัญญาณที่ออกจากสวิทช์ Control จะทำให้ค่าจากการนับในรีจีสเตอร์ TL1 เพิ่มขึ้นไปเรื่อยๆ ทีละ 1 จนถึง 0FFH ในการนับครั้งต่อไป จะทำให้บิท TF1 ในรีจีสเตอร์ TCON ไม่เป็น 1 และข้อมูลในรีจีสเตอร์ TH1 จะถูกโหลดไปยังรีจีสเตอร์ TL1 เพื่อเป็นค่าเริ่มต้นการนับต่อไป



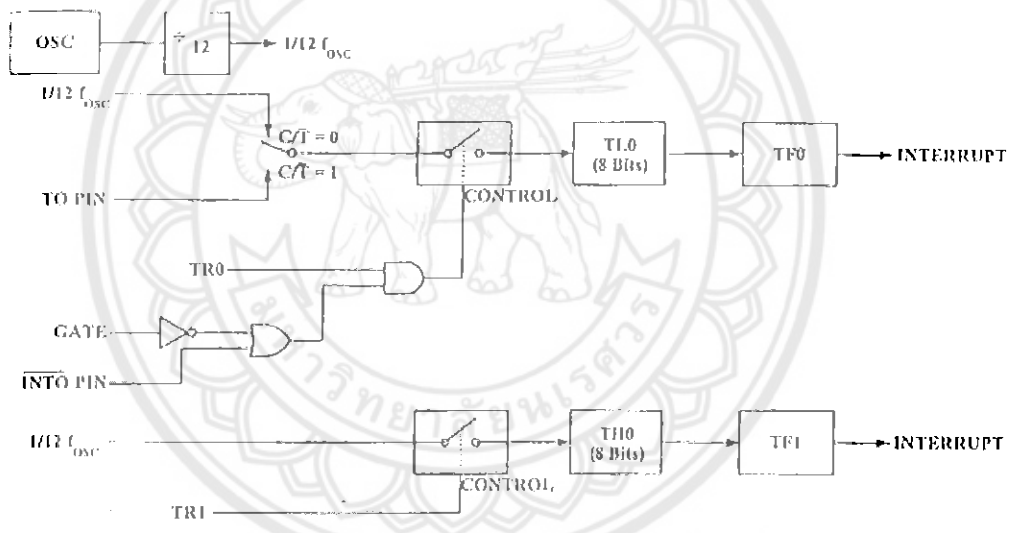
รูปที่ 2.12 Timer Mode 2

โหมด 3

การทำงานโหมด 3 ของ Timer 0 และ 1 จะต่างกันที่ Timer 1 ในโหมด 3 จะไม่ทำงาน Timer 0 ในโหมด 3 จะทำงานเป็นตัวนับที่เสมือนเป็นตัวนับ 8 บิทอยู่ 2 ตัวคือ TL0 และ TH0 ทำงานแยกกันดังรูปที่ 2.13

รีจีสเตอร์ TL0 จะเป็นตัวนับ 8 บิท ที่มีการนับสัญญาณจากออสซิลเลเตอร์หารด้วย 12 หรือนับสัญญาณที่เข้ามาทางขา T0 ขึ้นกับบิท C/T ในรีจีสเตอร์ TMOD และการนับจะควบคุมโดยบิท TR0 และ GATE ในรีจีสเตอร์ TMOD กับสภาวะลอจิกของสัญญาณที่ขา TNT0 เหมือนกับในการทำงานของโหมด 0, 1 และ 2 แต่ค่าจากการนับนี้สูงสุดจะมีเพียง 255 เท่านั้น เมื่อค่าการนับเปลี่ยนจาก 0FFH เป็น 00H คือเกิดการ Overflow จะทำให้บิท TF0 ถูก Set เป็น 1 และอาจเกิดการขัดจังหวะ (Interrupt) การทำงานของโปรแกรมได้ ถ้ามีการกำหนดค่าในรีจีสเตอร์ IE และ IP

ตัวนับอีกตัวคือรีจีสเตอร์ TH0 จะทำงานในโหมดของ Timer เท่านั้น คือจะนับจำนวนไวเคิลของสัญญาณที่ออกจากออสซิลเลเตอร์แล้วหารด้วย 12 การนับจะควบคุมได้ด้วยบิท TR1 ในรีจิสเตอร์ TMOD ถ้าบิทนี้เป็น 1 ก็จะมีสัญญาณเข้าไปยัง TH0 แต่ถ้าบิทนี้เป็น 0 ก็จะไม่มีสัญญาณเข้าไปยัง TH0



รูปที่ 2.13 Timer 0 mode 3

TCON Timer Control Register ตำแหน่งหน่วยความจำภายในเท่ากับ 088H

รีจีสเตอร์ขนาด 8 บิทนี้ใช้ควบคุมการทำงานและบอกสภาวะของ Timer 0 และ Timer 1 แต่ละบิทของรีจีสเตอร์นี้จะทำงานต่างกันดังรูปที่ 2.16

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1	TCON.7	Timer1 overflowflag.Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.
TR1	TCON.6	Timer1 run control bit.Set/cleared by software to turn Timer/Counter1 ON/OFF.
TF0	TCON.5	Timer0 overflowflag. Set by hardware when the Timer/Counter0 overflows. Cleared by hardware as processor vectors to the service routine.
IE1	TCON.3	External Interrupt 1 edge flag. Set by hardware when External interrupt edge is detected. Cleared by hardware when interrupt is processed.
IT1	TCON.2	Interrupt 1 type control bit.Set/cleared by software to specify falling edge/flow level triggered External Interrupt.
IE0	TCON.1	External Interrupt 0 edge flag. Set by hardware when External Interrupt edge detected. Cleared by hardware when interrupt is processed.
IT0	TCON.0	Interrupt 0 type control bit.Set/cleared by software to specify falling edge/low level triggerd External Interrupt.

รูปที่ 2.14 TCON Timer Control register

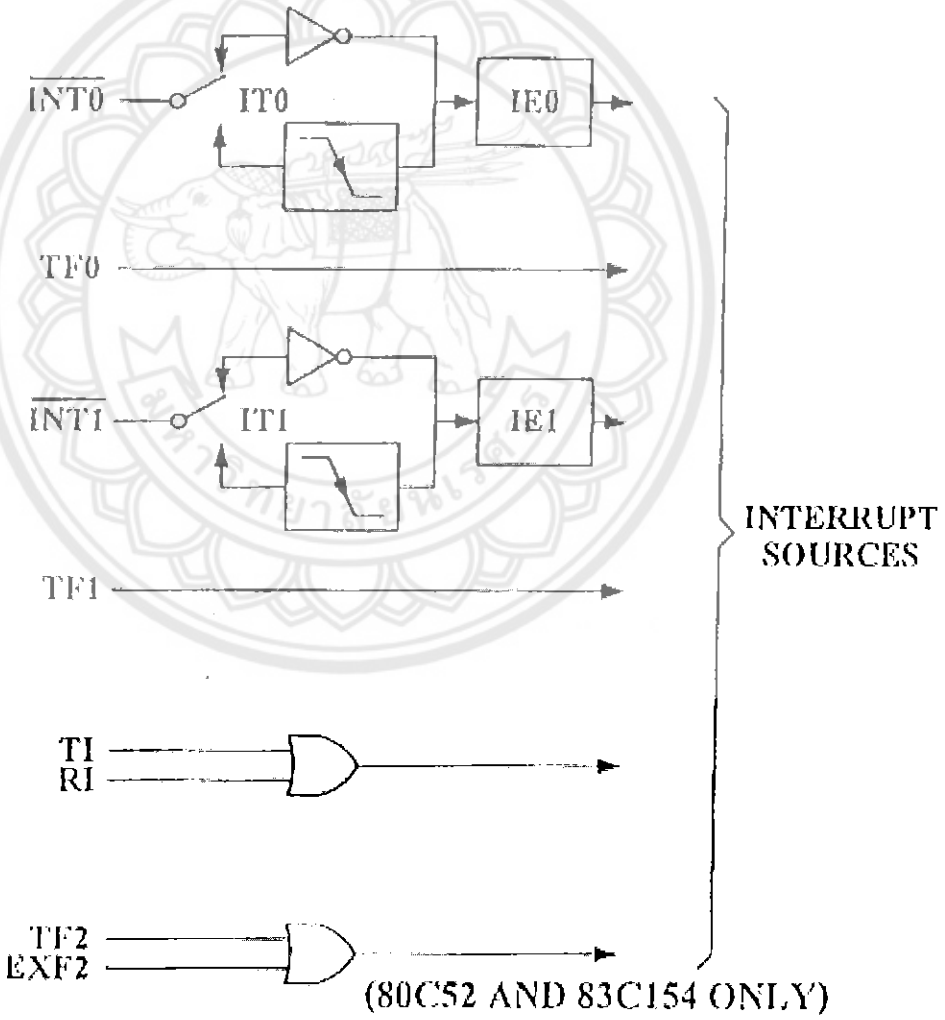
IE Interrupt Enable Register ตำแหน่งหน่วยความจำภายในเท่ากับ 0A8H

การขัดจังหวะการทำงาน (Interrupt) เป็นการที่มีสัญญาณหนึ่งหรือคำสั่งหนึ่งที่ (ไม่ใช่คำสั่ง CALL หรือ JMP) ที่จะทำให้การทำงานตามปกติของโปรแกรมถูกขัดจังหวะ แล้วข้ามไปทำงานยังตำแหน่งหนึ่งตำแหน่งใดที่กำหนดไว้ เมื่อทำงานในโปรแกรมขัดจังหวะเสร็จสิ้นก็จะกลับมาทำงานในโปรแกรมที่ตำแหน่งก่อนจะไปทำงานยังโปรแกรมขัดจังหวะ โปรแกรมที่ถูกกระโดดไปทำงานเรียกว่า โปรแกรมตอบสนองการขัดจังหวะ (Interrupt Service Routine) ใน 8051 จะสามารถขัดจังหวะด้วยสัญญาณจาก 6 แหล่งดังรูปที่ 2.15 ถ้าเป็น 8052 หรือ 83154 จะสามารถขัดจังหวะได้ด้วยสัญญาณจาก 8 แหล่ง คือ สัญญาณในชุดล่างสุดของรูปที่ 2.15 สัญญาณขัดจังหวะที่ 5 ในรูปที่ 2.15 จะสามารถทำให้เกิดการขัดจังหวะได้ 2 วิธี คือ มีข้อมูลเข้ามาทาง

พอร์ทอนุกรมเก็บอยู่ที่รีจีสเตอร์ SBUF และในกรณีที่ข้อมูลใน SBUF ส่งออกไปทางพอร์ทอนุกรมหมดแล้วไม่ว่าเกิดกรณีใดๆ ก็ทำให้เกิดการขัดจังหวะขึ้น

สัญญาณภายนอกที่เข้ามายัง 8051 ทางขา INT0 และ INT1 จะสามารถทำให้เกิดการขัดจังหวะการทำงาน 8051 ได้ (สัญญาณที่ 1 และ 3 ในรูปที่ 2.15) โดยสภาวะของสัญญาณนั้นเปลี่ยนจาก 1 เป็น 0 หรือ เมื่อสัญญาณนั้นเป็น 0 แล้วแต่การกำหนดในบิท IT0 และ IT1 ของรีจีสเตอร์ TCON จะทำให้บิท IE0 กับ IE1 เป็นตัวสร้างสัญญาณขัดจังหวะต่อไป

จาก Timer0 และ Timer 1 เมื่อค่าการนับในแต่ละโหมดถึงค่าสูงสุดในโหมดนั้นแล้ว เมื่อทำการนับต่อไป ค่าการนับต่อไปจะเป็น 0 (หรืออาจเป็นค่าที่ Reload จาก THx ในโหมด 2) และทำให้บิท TF0, TF1 เป็น 1 ซึ่งสัญญาณจาก 2 บิทนี้ จะสามารถทำให้เกิดการขัดจังหวะได้เช่นกัน ดังเช่นสัญญาณขัดจังหวะที่ 2 และ 4 ในรูปที่ 2.15



รูปที่ 2.15 แหล่งกำเนิดสัญญาณขัดจังหวะ

แหล่งกำเนิดสัญญาณทั้ง 6 ที่สามารถทำให้เกิดการขัดจังหวะได้ 5 แบบนี้ ผู้ใช้สามารถกำหนดให้สัญญาณใดบ้างเกิดการขัดจังหวะเรียกว่า Enable หรือไม่ให้เกิดการขัดจังหวะเรียกว่า Disable โดยการกำหนดในรีจีสเตอร์ IE (Interrupt Enable Register) ซึ่งมี 8 บิท แต่ละบิทสามารถ Enable ให้ขัดจังหวะได้จากแต่ละสัญญาณ ดังตารางที่ 2.5

ถ้าต้องการ Enable บิทใด ก็ให้โปรแกรมกำหนดค่าในบิทนั้นเป็น 1 ถ้าค่าในบิทนั้นเป็น 0 หมายถึง Disable การ Disable จะทำให้ไม่มีการขัดจังหวะการทำงานของโปรแกรมเนื่องจากสัญญาณขอขัดจังหวะนั้น ๆ EX0 เป็นชื่อบิท 0 และ EA เป็นชื่อของบิท 7

การกำหนดให้บิทใด Enable หรือ Disable นั้นจะเป็นไปโดยอิสระ ไม่ขึ้นแก่กัน จึงสามารถกำหนดให้บิทใดหรือมากกว่า 1 บิท Enable ก็ได้ ดังนั้น 8051 จึงมีรีจีสเตอร์อีกตัวที่ใช้เลือกว่า ถ้ามีสัญญาณขอการขัดจังหวะโปรแกรมเข้ามาพร้อมกันมากกว่า 1 แล้ว จะทำโปรแกรมตอบสนองการขัดจังหวะอันใดก่อน รีจีสเตอร์นั้นคือ IP Interrupt Priority Register

ตารางที่ 2.5 Interrupt Enable Register

(MSB) (LSB)

EA	X	ET2	ES	ET1	EX1	ET0	EX0

(MSB) (LSB)

$\overline{EA}$	X	ET2	ES	ET1	EX1	ET0	EX0

Symbol	Position	Function
$\overline{EA}$	IE.7	disabled all interrupts. If $\overline{EA}$=0, no interrupt will be acknowledged. If $\overline{EA}$=1 each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
-	IE.6	reserved
ET2	IE.5	enables or disables the Timer2 Overflow or capture interrupt. If ET2=0,the Timer 2 interrupt is disabled.
ES	IE.4	enables or disables the Serial Port interrupt.If ES=0, the Serial Port interrupt is disabled.
ET1	IE.3	enables or disables the Timer1 Overflow interrupt. If ET1=0, the Timer1 interrupt is disabled.
EX1	IE.2	enbles or disables External Interrupt 1. If EX1=0, External Interrupt is disabled.
ET0	IE.1	enables or disables the Timer 0 Overflow interrupt. If ET0=0,the Timer 0 interrupt is disabled.
EX0	IE.0	enables or disables External Interrupt. If EX0=0, External Interrupt 0 is disabled.

IP Interrupt Priority register ตำแหน่งหน่วยความจำภายในเท่ากับ 0B8H

ในการตอบสนองต่อสัญญาณขัดจังหวะของ 8051 นั้น ถ้าสัญญาณขัดจังหวะทั้งหมดเข้ามาพร้อมกัน8051จะต้องเลือกทำงานโปรแกรมตอบสนองการขัดจังหวะโดยการตรวจสอบสัญญาณเรียงตามลำดับ ซึ่งเรียกว่าวิธีการ Polling สัญญาณขัดจังหวะหนึ่งจะถูกตรวจสอบก่อน แล้วสัญญาณอื่นๆ จะถูกตรวจสอบต่อมา ถ้าสัญญาณนั้นขอขัดจังหวะ 8051 จะสร้างคำสั่ง CALL เป็นพิเศษขึ้นมาเพื่อไปทำงานโปรแกรมตอบสนองการขัดจังหวะของสัญญาณนั้น เมื่อเสร็จสิ้นแล้วก็จะกลับมาทำงานในโปรแกรมเดิมก่อนการขัดจังหวะ ทำให้เสมือนว่าสัญญาณแต่ละสัญญาณมี

ลำดับความสำคัญไม่เท่ากัน สัญญาณขัดจังหวะจะมีลำดับความสำคัญดังนี้ โดยเรียงจากลำดับความสำคัญสูงสุดถึงต่ำสุด

1. IE0
2. TF0
3. IE1
4. TF1
5. RI + TI

แต่ในการใช้งานบางครั้ง จำเป็นที่จะต้องให้สัญญาณใดสัญญาณหนึ่งมีลำดับความสำคัญสูงสุด (Highest Priority) เพื่อจะทำงานโปรแกรมตอบสนองการขัดจังหวะได้ก่อนการขัดจังหวะของสัญญาณอื่น จะสามารถกำหนดความสำคัญของการขัดจังหวะได้ใหม่โดยการกำหนดข้อมูลในบิทของรีจิสเตอร์ IP (Interrupt Priority Register) ตามตำแหน่งของแต่ละบิทในตารางที่ 2.7

ตารางที่ 2.6 IP : Interrupt Priority Register

(MSB) (LSB)

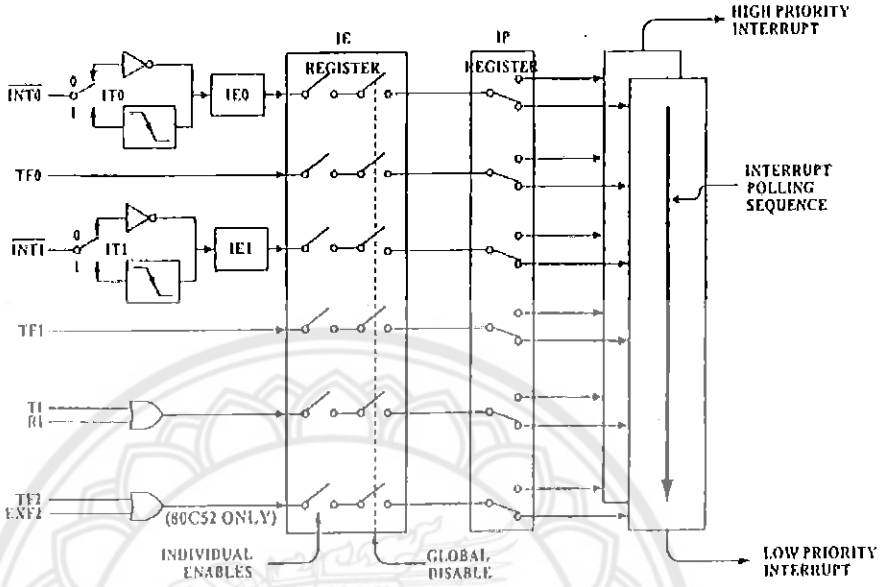
X	X	PT2	PS	PT1	PX1	PT0	PX0

Symbol	Position	Function
PCT	IP.7	PCT = 1,only one level
—	IP.6	reserved
PE2	IP.5	defines the Timer 2 interrupt priority level. PT2 = 1 programs it to the higher priority level.
PS	IP.4	defines the Serial Port interrupt priority level. PS = 1 programs it to the higher priority level.
PT1	IP.3	defines the Timer 1 interrupt priority level. PT1 = 1 programs it to the higher priority level.
PT0	IP.1	defines the Timer 0 interrupt priority level. PT0 = 1 programs it to the higher priority level.
PX0	IP.0	defines the External0 interrupt priority level. PX0 = 1 programs it to the higher priority level.

จากรูปที่ 2.16 เป็นภาพแสดงระบบขัดจังหวะของ 8052 ซึ่งแตกต่างจากของ 8051 ตรงที่ 8052 จะมีสัญญาณขัดจังหวะมาจาก TF2, EFX2 คือชุดล่างในภาพ

ในรูปจะเห็นว่าแต่ละสัญญาณจะมีสวิทช์ควบคุมอยู่ 3 ตัว 2 ตัวแรกอยู่ในกรอบสี่เหลี่ยม IE Register และอีก 1 สวิทช์อยู่ในกรอบ IP Register สวิทช์ตัวแรกทางซ้ายสุดจะควบคุมด้วยข้อมูลแต่ละบิท บิท 0 ถึงบิท 5 ของรีจิสเตอร์ IE ถ้าข้อมูลเป็น 1 จะทำให้สวิทช์นั้นปิดวงจร (Closed circuit) การควบคุมสวิทช์ทางซ้ายสุดของแต่ละสัญญาณจะไม่ขึ้นแก่กัน (Individual) สวิทช์ที่ 2 ถัดมาของทุกสัญญาณจะควบคุมร่วมกันด้วยบิท EA ในรีจิสเตอร์ IE ถ้าบิทนี้เป็น 0

สวิทช์ที่ 2 ของทุกสัญญาณจะเปิดวงจร (Opened circuit) ทำให้ไม่มีสัญญาณขอขัดจังหวะผ่านไปได้ สวิทช์ที่ 3 ทางขวาสุดจะใช้สำหรับเลือกว่าสัญญาณนั้นจะอยู่ในกลุ่มลำดับความสำคัญสูงสุด (High Priority Interrupt) หรือลำดับความสำคัญต่ำ (Low Priority Interrupt)



รูปที่ 2.16 ระบบการขัดจังหวะของ 8052 และ 83154

ถ้าต้องการให้สัญญาณใดมีลำดับความสำคัญสูงก็ให้กำหนดบิทนั้นในรีจีสเตอร์ IP เป็น 1 สวิทช์ที่ 3 จะเลื่อนไปอยู่ในตำแหน่งบน ถ้าไม่ต้องการก็กำหนดให้บิทนั้นเป็น 0 บิทใดเป็น 1 เรียกว่า สัญญาณนั้นอยู่ในกลุ่มลำดับความสำคัญสูง และบิทใดเป็น 0 เรียกว่าสัญญาณนั้นอยู่ในกลุ่มลำดับความสำคัญต่ำ ถ้ากลุ่มลำดับความสำคัญสูงมีเพียง 1 สัญญาณก็จะเรียกว่า สัญญาณนั้นมีลำดับความสำคัญสูงสุด ในกลุ่มลำดับความสำคัญเดียวกันก็จะมีการจัดลำดับความสำคัญเฉพาะกลุ่มโดยวิธี Polling เหมือนเดิม เช่น กรณีที่มีการกำหนดในบิทของรีจีสเตอร์ IP ให้มีลำดับความสำคัญสูงหรือต่ำเหมือนกันแล้วเกิดมีความต้องการขอขัดจังหวะเรียงตามลำดับความสำคัญ 5 ลำดับ ที่กล่าวมาแล้วข้างต้น เช่นให้ PT1, PX1 และ PT0 เป็น 1 เมื่อมีสัญญาณจะตอบสนองการทำงานขัดจังหวะของ Timer 0, External Interrupt 1 และ Timer 1 ตามลำดับ ในขณะที่ 8051 กำลังทำงาน โปรแกรมตอบสนองการขัดจังหวะของสัญญาณเขัดจังหวะที่มีลำดับความสำคัญต่ำอยู่ ถ้ามีสัญญาณขัดจังหวะมีลำดับความสำคัญสูงกว่าเกิดขึ้น การทำงานของโปรแกรมก็จะกระโดดไปทำงานในตำแหน่งโปรแกรมตอบสนองการขัดจังหวะของสัญญาณที่มีลำดับความสำคัญสูง เสร็จแล้วจึงกลับมาทำงานที่โปรแกรมตอบสนองการขัดจังหวะลำดับความสำคัญต่ำต่อไป แต่ละบิทของรีจิสเตอร์ IP นั้นจะบอกลำดับความสำคัญของแหล่งกำเนิดสัญญาณขัดจังหวะดังนี้

PX0 บิท 0 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจากภายนอก 8051 คือ INT0

PT0 บิท 1 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจาก Timer 0

PX1 บิท 2 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจากภายนอก 8051 คือ INT1

PT1 บิท 3 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจาก Timer 1

PT2 บิท 5 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจาก Timer 2 บิทนี้ใช้เฉพาะใน 8052 ที่มี Timer 2

PS บิท 3 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจาก Serial Port ในกรณีที่มีข้อมูลเข้ามาหรือส่งข้อมูลออกสิ้นสุดแล้ว

บิทที่เหลือจะไม่มีการใช้งาน

PCON (Power Control Register) ตำแหน่งหน่วยความจำภายในเท่ากับ 87

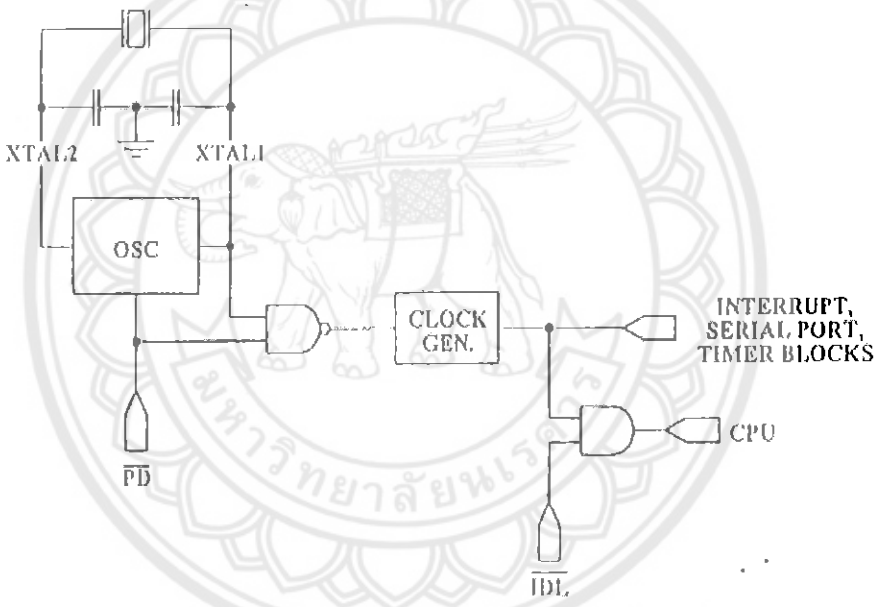
8051 เป็นไมโครคอนโทรลเลอร์ที่สร้างขึ้นด้วยเทคโนโลยีทั้งแบบ CHMOS และ HMOS ซึ่งแบบ CHMOS มีข้อดีตรงที่ใช้กำลังไฟต่ำกว่าแบบ HMOS ดังนั้นต่อไปในอนาคตจะมีแต่เฉพาะรุ่น CHMOS เท่านั้น นอกจากนี้แล้ว 8051 ยังมีข้อเสียตรงที่สามารถลดการใช้กำลังไฟลงได้โดยการทำงานใน Idel Mode และ Power Down Mode ใน Idle Mode นั้นสัญญาณนาฬิกาจากออสซิลเลเตอร์จะป้อนให้เฉพาะส่วน Interrupt, Serial Port และ Timer ในส่วนอื่นจะไม่มีสัญญาณนาฬิกาไปเลี้ยง แต่มีไฟเลี้ยงให้กับทุกส่วนในวงจร การใช้กำลังไฟจึงลดลงมาก ส่วนใน Power Down Mode นั้น ออสซิลเลเตอร์จะหยุดการทำงาน ทำให้ไม่มีสัญญาณนาฬิกาไปเลี้ยงส่วนใดๆ ในวงจรเลย แต่ข้อมูลภายในรีจิสเตอร์จะยังคงอยู่ไม่สูญหายไป

Idle Mode

ในรูปที่ 2.17 ขณะที่ 8051 ทำงานตามปกติไปจนถึงคำสั่งที่ทำให้บิท 0 ของรีจิสเตอร์ PMOD มีค่าเป็น 1 ก็จะเข้าสู่การทำงานใน Idle Mode โดยสัญญาณ IDL จะเป็น LOW (สัญญาณจะตรงข้ามกับข้อมูลในบิท 0) ขณะนี้สัญญาณนาฬิกาจากออสซิลเลเตอร์จะไม่ออกจาก AND GATE ไปยังส่วน CPU โดยจะจ่ายเฉพาะส่วน Interrupt, Timer และ Serial Port ในขณะนี้ 8051 จะเสมือนหยุดการทำงาน โดยข้อมูลใน Stack Pointer, Program Counter, Program Status Word, Accumulator และรีจิสเตอร์อื่นๆ จะไม่เปลี่ยนแปลง ข้อมูลที่พอร์ทต่างๆ จะยังคงค่าเดิมไว้เหมือนกับก่อนเข้าสู่ Idle Mode และสัญญาณ ALE กับ PSEN จะเป็นลอจิก High ขณะที่การใช้กระแสไฟของ 8051 จะลดต่ำลงมาก เนื่องจากภายใน 8051 จะไม่เกิดการเปลี่ยนสถานะลอจิก การที่จะออกจาก Idle Mode ทำได้ 2 วิธี

ตารางที่ 2.7 PCON : Power Control Register

Symbol	Position	Name and Function
SMOD	PCON.7	Double Baud rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either modes 1,2 or 3.
HPD	PCON.6 83C154 only)	Hard Power Down bit. Setting this bit allow CPU to enter in Power Down state on an external event (1 to 0 transition) on bit T1 (p.3-5) the CPU quit the Hard Power Down mode when bit T1 (p.3-5) go high or when reset is activated.
RPD	PCON.5 83C154 only)	Recover from Idle or Power Down bit. When 0 RPD has no effect. When 1, RPD permits to exit from Idle or Power Down with any non enabled interrup source (except timex 2). In this case the program s tart at the next address. When interrupt is enabled the appropriate interrupt routine is serviced.
—	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit.
GF0	PCON.2	General-purpose flag bit.
PD	PCON.1	Power Down bit. Setting this bit activated power down operation.
IDL	PCON.0	Idle mode bit. Setting this bit activated idle mode operation.



รูปที่ 2.17 Power down และ Idle mode

วิธีที่ 1 โดยการขัดจังหวะจากสัญญาณขัดจังหวะทั้ง 6 ที่กล่าวมาแล้ว เมื่อมีสัญญาณขอขัดจังหวะจากแหล่งใดก็ตาม จะทำให้บิท 0 ของรีจิสเตอร์ PCON มีค่าเป็น 0 และการทำงานของ 8051 จะออกจาก Idle Mode โดยกระโดดไปทำงานยังตำแหน่งของโปรแกรมตอบสนองการขัดจังหวะนั้นๆ เมื่อเสร็จสิ้นการทำงานในโปรแกรมตอบสนองการขัดจังหวะโดยการทำงานคำสั่ง RETI ก็จะกลับมาทำงานยังคำสั่งที่อยู่ต่อจากคำสั่งที่ทำให้บิท 0 ของรีจิสเตอร์ PCON เป็น 1 ซึ่งทำให้การทำงานเข้าสู่ Idle Mode เช่นคำสั่งที่ตำแหน่ง 2000H คือ MOV PCON,#1H ที่เป็นคำสั่งที่ทำให้บิท IDC มีค่า 1 ดังนั้นเมื่อทำงานที่คำสั่งนี้เสร็จสิ้นก็จะหยุดการทำงาน และเมื่อ

เกิดการขัดจังหวะเนื่องจากสัญญาณขัดจังหวะใดๆ ก็ตาม 8051 จะออกจาก Idle Mode ไปทำงานที่โปรแกรมตอบสนองการขัดจังหวะ เมื่อเสร็จสิ้นการทำงานโปรแกรมตอบสนองการขัดจังหวะแล้วจะกระโดดมาทำงานที่ตำแหน่งของคำสั่งต่อจากคำสั่ง MOV PCON,#1H

วิธีที่ 2 ก็คือการป้อนสัญญาณที่มีสภาวะลอจิก 1 เข้าไปยังขา RST เพื่อทำการรีเซ็ท 8051 สัญญาณรีเซ็ทนี้จะต้องมีลอจิกเป็น 1 ในระหว่างนี้ 8051 จะทำงานในคำสั่งต่อจากคำสั่งที่ทำให้บิท 0 ของ PCON เป็น 1 เข้าสู่ Idle Mode ต่อไปอีก 2 – 3 คำสั่ง ก่อนที่ทุกอย่างจะเข้าสู่การรีเซ็ท ดังนั้น จะต้องระวังคำสั่งที่อยู่ต่อจากคำสั่งที่ทำให้เข้าสู่ Idle Mode อาจทำให้ข้อมูลบนพอร์ทเปลี่ยนแปลงจนทำให้อุปกรณ์ที่มาต่อเสียหายเมื่อกลับออกจาก Idle Mode

ในวิธีที่ 1 นั้นแสดงว่าการเข้าสู่โปรแกรมตอบสนองการขัดจังหวะเป็นได้ 2 กรณี คือ ขณะที่ทำงานตามปกติแล้วมีสัญญาณขัดจังหวะ ก็จะกระโดดไปทำงานในโปรแกรมตอบสนองการขัดจังหวะ หรือในกรณีที่อยู่ใน Idle Mode แล้วมีสัญญาณขัดจังหวะก็อาจจะกระโดดไปทำงานในโปรแกรมตอบสนองการขัดจังหวะ จึงอาจให้โปรแกรมกำหนดข้อมูลในบิท GF0 หรือ GF1 หรือทั้งสอง เพื่อให้โปรแกรมตอบสนองการขัดจังหวะรู้ได้ว่า การเข้าสู่โปรแกรมตอบสนองการขัดจังหวะนั้นมาจากกรณีใด

Power Down Mode

ในการเข้าสู่ Power Down Mode นั้นจะทำได้โดยการใช้โปรแกรมกำหนดให้บิท PD หรือบิท 1 ของรีจิสเตอร์ PCON มีค่าเป็น 1 เช่น MOV PCON,#2 เมื่อ 8051 ทำงานที่คำสั่งนี้เสร็จสิ้นสัญญาณ PD ในรูปที่ 2.17 จะเป็น 0 เพราะจะตรงข้ามกับข้อมูลในบิท PD ทำให้การทำงานเข้าสู่ Power Down Mode ทันที ในโหมดนี้เองออสซิลเลเตอร์จะหยุดการทำงาน ทำให้ไม่มีสัญญาณนาฬิกาไปยังส่วนต่าง ๆ ภายใน 8051 ดังนั้นจะไม่มีการทำงานใด ๆ รวมทั้งข้อมูลในรีจิสเตอร์ทุกตัวจะไม่เปลี่ยนแปลง และข้อมูลใน RAM ภายในก็จะไม่เปลี่ยนแปลง ขณะนี้สัญญาณออกจากขา ALE และ PSEN จะเป็น 0 การใช้กำลังไฟของ 8051 จะต่ำมาก อีกทั้งสามารถลดไฟเลี้ยงวงจรที่ขา Vcc ลงได้จนถึง 2 โวลท์ โดยไม่ทำให้ข้อมูลใด ๆ ใน 8051 สูญหายไป การออกจาก Power Down Mode ทำได้วิธีเดียว คือ การป้อนสัญญาณลอจิก 1 เข้าไปที่ขา RST ของ 8051 ซึ่งทำให้เข้าสู่การรีเซ็ท 8051 แต่จะทำให้ข้อมูลใน SFR เปลี่ยนแปลงไป ถ้าในขณะที่อยู่ใน Power Down Mode มีการลดไฟเลี้ยงวงจรจะต้องให้ไฟเลี้ยงวงจรกลับมาอยู่ที่ 5 โวลท์ก่อนที่จะเข้าสู่การรีเซ็ท

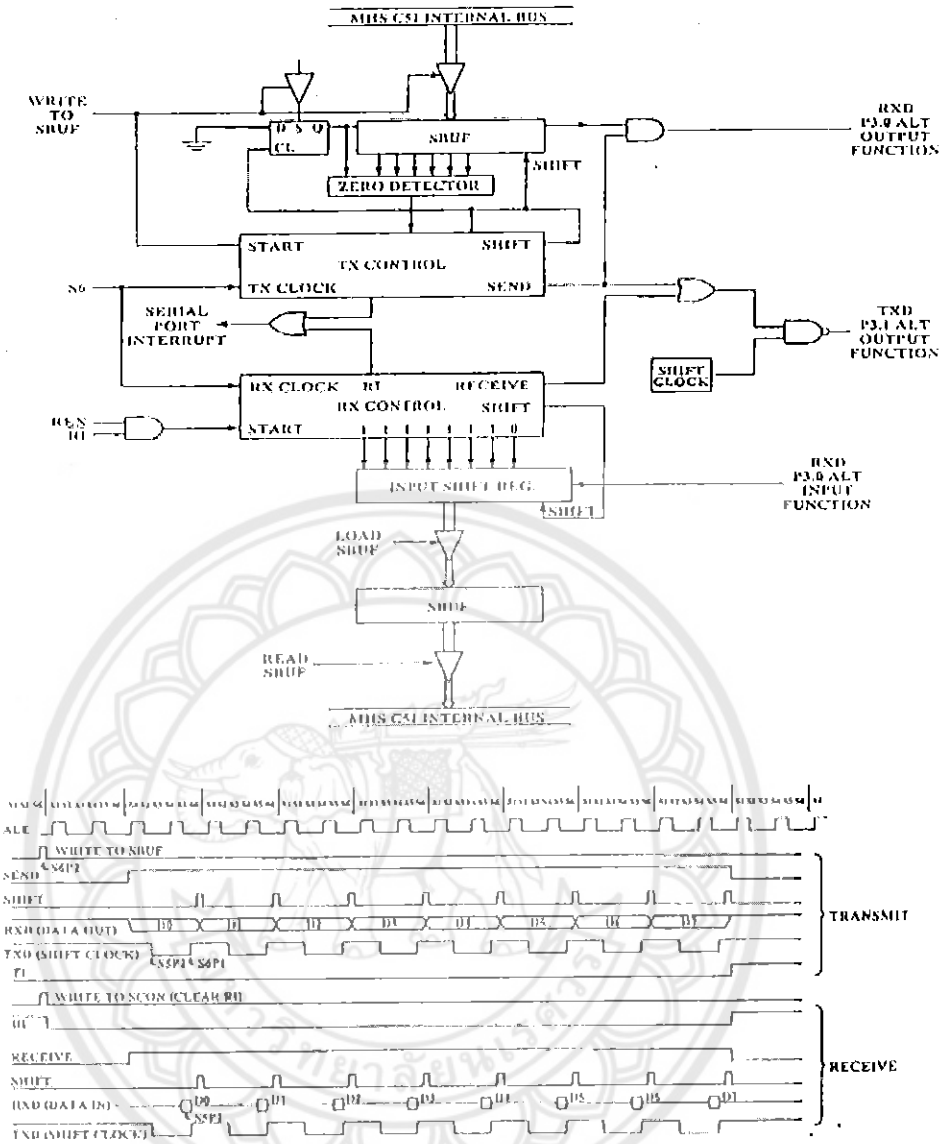
2.2.3 การรับ-ส่งข้อมูลทางพอร์ทอนุกรม

ในการรับ – ส่งข้อมูล แบบอนุกรมผ่านทางพอร์ทอนุกรมนั้น จะต้องมีการกำหนดโหมดการทำงานในรีจิสเตอร์ SCON และในบางโหมดของการทำงานจะสามารถกำหนดอัตราการส่งข้อมูลได้โดยการโปรแกรมใน Timer ข้อมูลที่จะส่งออก หรือรับเข้าทางพอร์ทอนุกรมจะอยู่ที่รีจิสเตอร์ SBUF การทำงานของภายในแต่ละโหมดมีดังนี้

การทำงานโหมด 0

การส่งข้อมูล

การส่งข้อมูลจะเริ่มจากการทำงานของคำสั่งให้เคลื่อนย้ายข้อมูลเข้ามายังรีจิสเตอร์ SBUF โดยจะมีสัญญาณ Write to SBUF ซึ่งเกิดขึ้นในช่วงเวลา S6P2 สัญญาณนี้จะทำให้ข้อมูลจากบัสภายใน 8051 (8051 Internal Bus) ทางด้านบนถูกนำไปเก็บที่รีจิสเตอร์ SBUF อีกทั้งยังทำการเซ็ทให้ D FLIP- FLOP ที่อยู่ทางซ้ายของ SBUF มีสภาวะลอจิกที่ขา Q เป็น 1 สัญญาณ Write to SBUF ซึ่งต่อเข้าไปยังขา Start ของ TX Control จะบอกให้ TX Control เริ่มส่งข้อมูล โดยจะหน่วงเวลาไว้ 1 ไซเคิลของเครื่อง เมื่ออ่านเวลา 1 ไซเคิลของเครื่องไปแล้ว ขึ้นไซเคิลใหม่ สัญญาณ SEND จะเปลี่ยนสภาวะลอลจิกจาก 0 เป็น 1 และเริ่มส่งข้อมูลบิท 0 จากนี้ในทุกเวลา S6P2 สัญญาณ Shift จะเปลี่ยนเป็น 1 ออกจากวงจร TX Control (เวลาอื่นสัญญาณนี้จะเป็น 0) ทำให้ข้อมูลใน SBUF ถูกเลื่อนออกไปทีละบิตตรงขอบขาลงของสัญญาณ Shift ในทุกๆ ไซเคิลของเครื่อง ขณะที่ข้อมูลถูกเลื่อนออกไปทางขวานี้ ข้อมูลจาก D FLIP – FLOP จะเอนมาทางซ้าย ข้อมูลจาก D FLIP – FLOP บิทแรกที่เลื่อนเข้ามาจะเป็น 1 เพราะถูกรีเซ็ทไว้ตอนแรก แต่บิทต่อมาจะเป็น 0 เพราะขา D จะถูกต่อไว้ที่กราวน์ ขณะที่สัญญาณ SEND มีลอจิกเป็น 1 จะทำให้เอาท์พุทของ OR GATE มีลอจิกเป็น 1 ดังนั้นสัญญาณจากวงจร Shift Clock จะถูกส่งออกไปทางขา P3.1 (TXD) โดยสัญญาณนี้จะเป็น 1 ในช่วงเวลา S6P1 ถึง S6P2 ของไซเคิลเครื่องถัดไป และจะเป็น 0 ในช่วงเวลา S3P1 ถึง S5P2 ในไซเคิลของเครื่องเดียวกัน สัญญาณจากขา TXDที่ส่งออกไปนี้ก็เพื่อให้อุปกรณ์ปลายทางสามารถรับข้อมูลได้ถูกต้องเพราะถูกส่งออกไปพร้อมกับข้อมูลใน SBUF ข้อมูลจำนวน 8 บิทจะถูกเลื่อนออกไปทางขา RXD จนครบทั้ง 8 บิท เมื่อข้อมูลบิทสุดท้ายถูกส่งออกไปจะทำให้ 1 ที่เกิดจาก D FLIP – FLOP ในตอนเริ่มต้นการส่งข้อมูลถูกเลื่อนมาอยู่างขวาสุดของรีจิสเตอร์ SBUF และทางซ้ายทั้งหมดจะเป็น 0 ทำให้วงจร Zero Detector ซึ่งตรวจสอบค่า 0 ส่วนสัญญาณไปบอกวงจร TX Control ว่าสิ้นสุดการส่งข้อมูลออก เมื่อสิ้นสุดการส่งข้อมูล สัญญาณ SEND จะเปลี่ยนสภาวะลอจิก 1 เป็น 0 ที่ขอบขาลงของสัญญาณ Shift และการส่งข้อมูลบิทสุดท้ายออกไปจะทำให้บิท TI ในรีจิสเตอร์ SCON เปลี่ยนจาก 0 เป็น 1 บอกการสิ้นสุดของการส่งข้อมูล



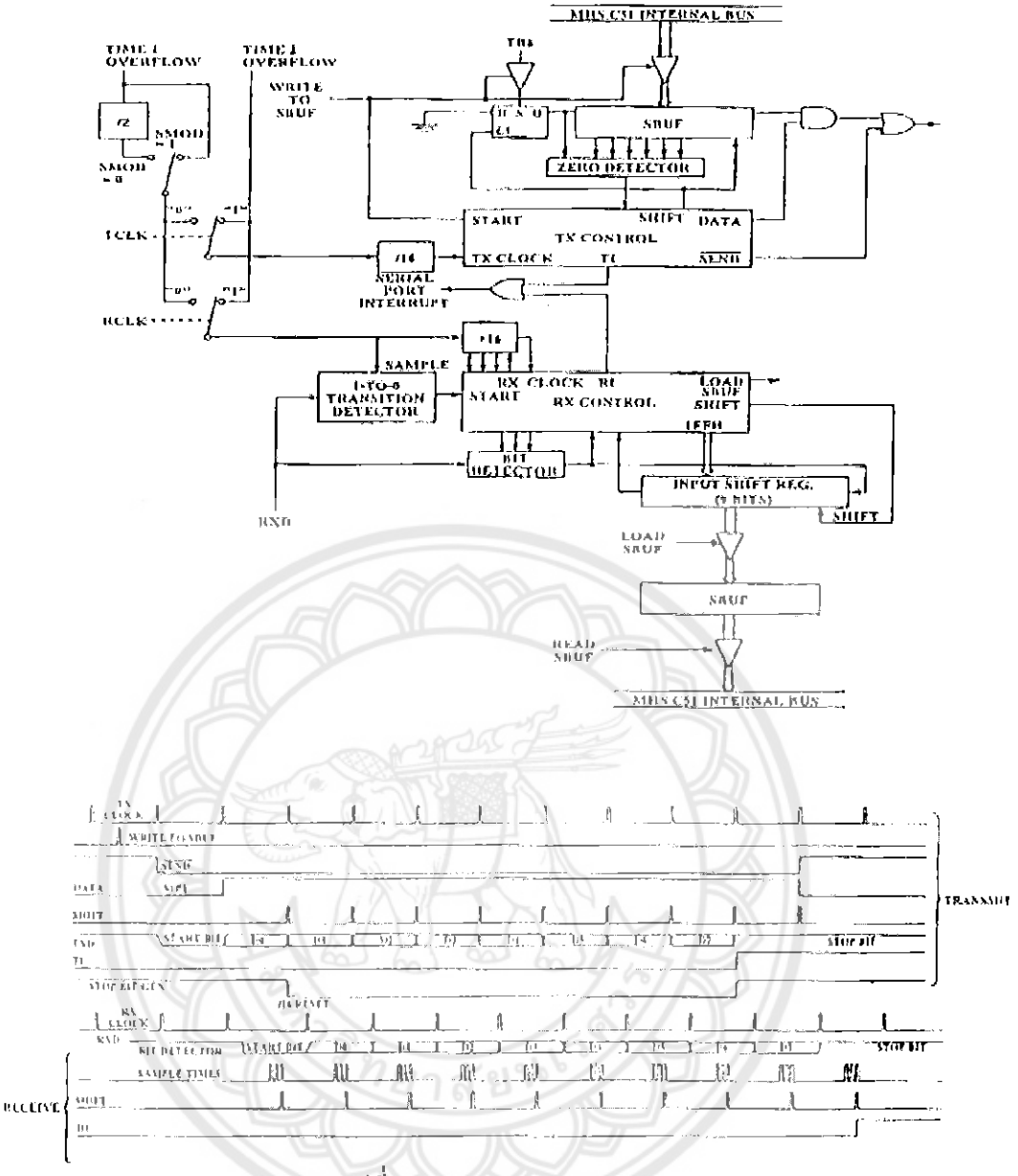
รูปที่ 2.18 Serial Port Mode 0

การรับข้อมูล

การรับข้อมูลทางพอร์กอนุกรมในโหมด 0 จะเริ่มต้นรับข้อมูลเข้ามาทางขา RXD ก็ต่อเมื่อมีการทำงานของคำสั่ง Set ค่าบิท REN เป็น 1 และ Clear บิท R1 เป็น 0 ในรีจิสเตอร์ SCON ซึ่งการทำงานของคำสั่งนี้จะทำให้เริ่มรับข้อมูลที่เวลา S6P2 เสร็จแล้วจะหน่วงเวลาไปจนถึง S6P2 ในไซเคิลของเครื่องถัดไปก็จะทำให้สภาวะลอจิกของสัญญาณ Receive เปลี่ยนจาก 1 เป็น 0 เพื่อเริ่มส่งสัญญาณ Clock จากวงจร Shift Clock ถูกส่งออกไปทางขา TXD โดยสัญญาณ Clock ที่ขานี้จะมีสภาวะเป็น 1 ตั้งแต่ S6P1 จนถึง S6P2 ของไซเคิลเครื่องถัดไป และมีสภาวะลอจิกเป็น 0 ในช่วงเวลา S3P1 ถึง S5P2 แต่ก่อนที่สัญญาณ Receive จะเปลี่ยนเป็น 1 นั้นวงจร RX Control จะเขียนข้อมูล 111111110H เข้าไปยัง Input Shift Register จากนั้นใน

ทุก ๆ เวลา S6P2 ซึ่งสัญญาณ Shift มีสภาวะลอจิกเป็น 1 (นอกนั้นที่เวลาอื่นจะมีลอจิกเป็น 0) จะเลื่อนข้อมูลใน Input Shift Register ไปทางซ้ายทีละ 1 บิท เมื่อครบ 7 ครั้ง ข้อมูล 0 ซึ่งอยู่ทางซ้ายสุดของรีจิสเตอร์ในตอนเริ่มต้นจะเลื่อนมาอยู่ทีละ 1 บิท เมื่อครบ 7 ครั้ง ข้อมูล 0 และบอกให้กับ RX Control ว่ามีข้อมูลบิทสุดท้ายอีก 1 บิทที่จะเข้ามา เมื่อข้อมูลสุดท้ายเข้ามาที่เวลา S5P2 ของไซเคิลเครื่องที่ 9 (ข้อมูลที่เข้ามายังพอร์ทอนุกรมจะถูกอ่านเข้าไปที่เวลา S5P2 แต่จะถูกเก็บเข้าไปที่ Input Shift Register ที่เวลา S6) นับตั้งแต่เริ่มรับข้อมูลจะทำให้ที่เวลา S1P1 ของไซเคิลเครื่องที่ 10 นับตั้งแต่เริ่มมีสัญญาณ Write to SCON (Clear RI) จะเกิดการส่งข้อมูล Input Shift Register ไปเก็บยังรีจิสเตอร์ SBUF ต่อไป และในขณะเดียวกันก็จะทำให้บิท RI ซึ่งถูก Clear ตั้งแต่เริ่มต้นรับข้อมูลถูก Set ให้เป็น 1 และจะทำให้สัญญาณ Receive เป็น 0 ทำให้ไม่มีสัญญาณ Clock ออกไป

ในโหมดนี้จะมีการส่งข้อมูลชุดละ 10 บิท คือ Start bit 1 บิท ข้อมูล 8 บิท และ Stop 1 บิท อัตราการส่งข้อมูลในโหมดนี้จะขึ้นกับอัตราการเกิด Overflow ใน Timer 1 ข้อมูลนี้จะถูกส่งออก 1 บิททุกๆ 16 หรือ 32 ครั้งของการเกิด Overflow ขึ้นใน Timer 1 (การ Overflow คือการที่ข้อมูลใน Timer Register มีค่าเพิ่มขึ้นจนถึงค่าสูงสุด เมื่อเพิ่มค่าไปอีกก็จะกลับเป็น 0) ใน 8052 จะสามารถกำหนดอัตราการส่งข้อมูลขึ้นกับอัตราการเกิด Overflow ของ Timer 2 ได้ดังรูปที่ 2.18



รูปที่ 2.19 Serial Port Mode 1

การทำงานโหมด 1

- การส่งข้อมูล

จากรูปที่ 2.21 SMOD จะเป็นตัวเลือกว่า สัญญาณ Timer 1 Overflow ที่ส่งไปยังวงจรหาร 16 จะถูกหาร 2 ก่อนหรือไม่ ถ้า SMOD เป็น 1 สัญญาณ Timer 1 Overflow จะไม่ถูกหาร แต่ถ้า SMOD เป็น 0 สัญญาณ Timer 1 Overflow จะถูกกหาร 2 ก่อนที่จะเข้าวงจรหาร 16 การส่งข้อมูลจะเริ่มจากการที่มีคำสั่งเขียนข้อมูลไปยังรีจิสเตอร์ SBUF จะมีสัญญาณ Write to SBUF เกิดขึ้นเพิ่มรับข้อมูลจาก Internal Bus ด้านบนไปเก็บยังรีจิสเตอร์ SBUF และทำให้เอาท์พุทของ D FLOP – FLOP ทางซ้ายของ SBUF มีค่าเป็น 1 และเป็นบิทที่ 9 ของการส่งสัญญาณ Write to SBUF ยังส่งไปยัง TX Control ด้วย ขณะนี้ข้อมูลในวงจรหาร 16 มีค่า

เป็นอะไรไม่ทราบ จึงจะรอจนกว่าข้อมูลในวงจรหาร 16 นับเพิ่มขึ้นจนถึงค่าสูงสุดแล้ววนกลับเป็น 0 คือเกิดการวนกลับทำให้เกิดการส่งข้อมูลที่เวลา S1P1 ของไซเคิลเครื่องถัดไป (การส่งข้อมูลจะสัมพันธ์กับการเกิด Overflow ในวงจรหาร 16) สัญญาณ SEND จาก TX Control เปลี่ยนสภาวะลอจิกเป็น 0 แล้วเริ่มส่งข้อมูลที่เป็น Start bit (0) ออกไป เมื่อส่ง Start bit ออกไปแล้ววงจร TX Control ก็จะทำให้สัญญาณ DATA เป็น 1 เพื่อเลื่อนข้อมูลใน SBUF ออกไปเริ่มจากบิท 0 จนถึงบิท 7 การส่งข้อมูลนี้จะเกิดขึ้นเมื่อสัญญาณ TX Clock เปลี่ยนสถานะจาก 0 เป็น 1 ดังในรูปที่ 2.19 ข้อมูลถูกเลื่อนออกไปนั้นจะมี 0 ถูกเลื่อนเข้ามาทางซ้ายของรีจิสเตอร์ SBUF เมื่อข้อมูลเลื่อนออกไปทั้ง 8 บิทแล้ว บิทที่ 9 ซึ่งเป็น 1 และตอนต้นอยู่ทางซ้ายสุดจะถูกเลื่อนมาอยู่ในตำแหน่งสุดท้ายทางขวาของรีจิสเตอร์ SBUF และทางซ้ายของหลักนี้จะมี 0 อยู่ทั้ง 8 บิท ใน SBUF ทำให้ Zero Detector รู้ว่าเป็นข้อมูลบิทสุดท้ายแล้วที่ส่งออกโดยจะมีสัญญาณมาบอกกับวงจร TX Control ด้วย เมื่อ TX Control ส่งสัญญาณ Shift ออกไปเป็นการส่งข้อมูลบิทสุดท้าย (บิท 7) ออกไปก็จะรออีก 1 TX Control ด้วย เมื่อ TX Control ส่งสัญญาณ Shift ออกไปเป็นเป็นการส่งข้อมูลบิทสุดท้าย (บิท 7) ออกไป ก็จะรออีก 1 TX Clock (Bit Clock) ก็จะทำให้ขา TXD ส่งข้อมูล Stop Bit (1) ออกมา สัญญาณ DATA ซึ่งมีสภาวะเป็นลอจิก 1 มาตั้งแต่เริ่มส่งข้อมูลบิท 0 ก็จะกลับเป็น 0 และบิท TI จะเป็น 1 เพื่อบอกการสิ้นสุดการส่งข้อมูลทั้งหมดจะสิ้นสุดลงเมื่อสัญญาณ TX Clock ไซเคิลที่ 10 นับตั้งแต่สัญญาณ SEND เปลี่ยนสภาวะเป็น 0

- การรับข้อมูล

การรับข้อมูลจะเกิดขึ้น กับอัตราการเกิด Overflow ใน Timer 1 แล้วหาร 2 หรือไม่ขึ้นกับค่าของบิท SMOD สัญญาณนี้จะไปเข้าวงจรหาร 16 และเป็นตัวกำหนดอัตราการรับข้อมูล การรับข้อมูลจะเริ่มจากวงจร 1 – TO – 0 Transition Detector พบว่าสัญญาณที่ขา RXD เปลี่ยนจาก 1 เป็น 0 ซึ่งหมายถึงมีข้อมูล Start bit เข้ามา การตรวจสอบนี้กระทำที่อัตราเดียวกับสัญญาณที่เข้าวงจรหาร 16 เมื่อพบการเปลี่ยนสถานะลอจิกที่ขา RXD ก็จะเริ่มการรับข้อมูล ขณะนี้จะรีเซ็ทวงจรหาร 16 ให้มีค่าเป็น 0 เพื่อสร้างสัญญาณ RX Clock ให้เข้าจังหวะ (Synchronous) กับข้อมูลที่เข้ามา โดยสัญญาณ RX Clock จะเป็น 1 เมื่อการนับของวงจรหาร 16 มีค่าเป็น 15 ขณะที่วงจรหาร 16 นับถึง 7, 8 และ 9 จะมีการตรวจสอบข้อมูลที่เข้ามาทางขา RXD เพื่อเป็นการตรวจว่าข้อมูลนั้นเป็นอะไร ถ้าอย่างน้อยข้อมูล 2 ใน 3 เป็นค่าใดก็จะถือว่าข้อมูลที่เข้ามาเป็นค่านั้น ถ้าในการตรวจสอบ Start Bit แล้วพบว่าผิดพลาด คือไม่เป็น 0 ก็จะรีเซ็ทการทำงานเพื่อไปตรวจสอบการเปลี่ยนสถานะจาก 1 เป็น 0 ของข้อมูลที่ขา RXD ใหม่ แต่ถ้าพบ Start Bit ก็จะเก็บข้อมูลทั้งหมดที่เข้ามา โดยเลื่อนข้อมูลเข้าไปยัง Input Shift Register ที่มีสัญญาณควบคุมการเลื่อนข้อมูล (Shift) ส่งมาจาก RX Control ในตอนเริ่มต้นการรับข้อมูลจะมีการเขียนข้อมูล 1FFFH ไปเก็บใน Input Shift Register ขณะที่ข้อมูลถูกเลื่อนเข้าไปทางขวา

ของ Input Shift Register ก็จะมี 1 ถูกเลื่อนออกไปทางซ้ายทุกครั้งที่มีข้อมูลเข้ามา เมื่อ Start Bit ที่รับเข้ามาถูกเลื่อนไปถึงซ้ายสุดของ Input Shift Register ก็จะมีสัญญาณไปบอก RX Control Block หลังจากข้อมูลบิทสุดท้ายเข้ามาแล้วก็จะโหลด (Load) เอาข้อมูล 8 บิทไปเก็บในรีจิสเตอร์ SBUF พร้อมทั้ง Set ค่าในบิท R1 และ RB8 ของรีจิสเตอร์ SCON แต่การโหลดข้อมูลไปเก็บนี้จะเกิดขึ้นได้ก็ต่อเมื่อ

1. RI = 0 และ

2. SM = 0 หรือถ้า SM2 = 1 จะต้องได้รับ Stop Bit เป็น 1

ถ้าไม่มีสภาวะใดสภาวะหนึ่งดังกล่าว ข้อมูลที่รับเข้ามาก็จะถูกทิ้งไปคือไม่โหลดไปเก็บในรีจิสเตอร์ SBSF ถ้ามีสภาวะดังกล่าวถูกต้อง Stop Bit จะถูกนำไปเก็บในรีจิสเตอร์ SBUF และ บิท RI จะเป็น 1

แต่ไม่ว่าทั้ง 2 กรณีจะเกิดขึ้นหรือไม่ ก็จะกลับไปสู่การตรวจสภาวะเปลี่ยนจาก 1 เป็น 0 ที่ขา RXD เพื่อรับข้อมูลต่อไป

ในการรับข้อมูลอนุกรมโหมด 1 นี้ อัตราการส่งข้อมูลแต่ละบิท (Buad Rate) จะขึ้นกับอัตราการเกิด Overflow ใน Timer1 ดังสมการ

Buad Rate =$(2^{SMOD}/32)$ X (Timer 1 Overflow Rate)

ในขณะที่ใช้ Timer1 เป็นตัวกำหนด Buad Rate นี้จะต้อง Disable ไม่ให้เกิดการขัดจังหวะเนื่องมาจากการ Overflow Timer 1 อาจใช้ในโหมดของ Timer หรือ Counter ก็ได้ ซึ่งเมื่อการนับในรีจิสเตอร์ตัวนับมีค่าสูงสุดแล้วกลับมาเป็น 0 ก็จะเกิด Overflow เช่นเดียวกัน แต่โดยปกติแล้วจะใช้ Timer1 นี้ในโหมดของ Timer ที่มีการทำงานแบบ Auto Reload โหมด 2 ตัว เพื่อว่าเมื่อค่าในการนับโดยรีจิสเตอร์ TL1 ถึงค่าสูงสุดก็จะโหลดค่าในรีจิสเตอร์ TH1 มาไว้ใน TL1 สำหรับเป็นค่าเริ่มต้นการนับต่อไป ซึ่ง Buad Rate จะมีค่าดังสมการข้างล่าง คือ

Buad Rate =$(2^{SMOD}/32)$ X (Oscillator frequency)/(12x[256-(TH)])

โดยที่ SMOD เป็นบิทหนึ่ง ในรีจิสเตอร์ PCON

เช่นความถี่ของออสซิลเลเตอร์เท่ากับ 11.059 MHz บิท SMOD = 0 รีจิสเตอร์ TH1 มีค่า E8H, Timer 1 ทำงานในโหมด 2 จะได้ว่าอัตราการส่ง-รับข้อมูลแบบอนุกรมจะมีค่าเท่ากับ

= $(2^0/32) \times (11.059 \times 10^6)/(12 \times [256-232])$

= $(1/32) \times (11.059 \times 10^6)/(12 \times 24)$

= 1200 บิท/วินาที

การทำงานโหมด 2 และโหมด 3

โหมด 2 และ โหมด 3 ของการรับ – ส่งข้อมูลทางพอร์ทอนุกรมจะเหมือนกันแตกต่างเฉพาะตรงที่อัตราการส่งข้อมูลเท่านั้น ในโหมด 2 จะเลือกอัตราการรับ-ส่ง ข้อมูลได้เป็น 1/32

หรือ 1/64 เท่าของความถี่สัญญาณออสซิลเลเตอร์ ส่วนในโหมด 3 จะสามารถกำหนดอัตราการส่งข้อมูลจะขึ้นกับอัตราการเกิด Overflow ของ Time เช่นเดียวกับการรับ-ส่งข้อมูลโหมด 1

ในรูปที่ 2.20 จะเห็นว่าสัญญาณจากเฟส 2 ของสัญญาณนาฬิกาซึ่งมีความถี่เท่ากับ 1/2 ของสัญญาณจากออสซิลเลเตอร์จะถูกเลือกโดยสวิทช์ SMOD ว่าจะให้หาร 2 หรือไม่ สวิทช์ SMOD นี้จะสามารถกำหนดตำแหน่งให้อยู่ข้างบนหรือข้างล่างโดยการกำหนดค่าบิท SMOD ในรีจิสเตอร์ PCON ให้เป็น 1 หรือ 0 ตามลำดับ สัญญาณที่ออกจากวงจรหาร 16 จะมีความถี่เป็น 1/32 หรือ 1/64 เท่าของสัญญาณจากออสซิลเลเตอร์

ในรูป 2.21 สัญญาณ Timer 1 Overflow อันเกิดจากการนับถึงค่าสูงสุดใน Timer 1 แล้วค่าการนับจะกลับเป็น 0 ใหม่ สัญญาณนี้จะถูกเลือกโดยสวิทช์ SMOD ให้หาร 2 หรือไม่ก่อนที่จะส่งไปยังวงจรหาร 16 เพื่อสร้างเป็นสัญญาณควบคุมการเลื่อนข้อมูลหรือออกทางพอร์ทอนุกรมต่อไป การเลือกตำแหน่งของสวิทช์ SMOD ก็โดยการกำหนดค่าในบิท SMOD ของ รีจิสเตอร์ PCON ถ้าบิทนี้เป็น 1 สัญญาณ Timer 1 Overflow จะถูกส่งไปยังวงจรหาร 16 โดยตรง แต่ถ้าบิทนี้เป็น 0 สัญญาณ Timer 1 Overflow จะถูกหาร 2 ก่อนแล้วจึงส่งเข้าไปยังวงจรหาร 16 การกำหนดอัตราการส่งข้อมูลจะทำให้ได้เหมือนกับในการรับ-ส่งข้อมูลพอร์ทอนุกรมโหมด 1 ที่กล่าวมาแล้วทุกประการ

การรับ-ส่งข้อมูลในโหมด 2 และ 3 นั้น 1 ชุดของข้อมูลจะประกอบด้วย 1 Start Bit, 9 Data Bit 1 Stop Bit ทั้งหมดนี้จะส่งออกหรือรับเข้าทางพอร์ทอนุกรมทีละบิทเรียงตามลำดับ จากรูปที่ 2.20 และ 2.21 จะสามารถอธิบายการทำงานภายใน 8051 ในการส่งหรือรับข้อมูลได้ดังนี้

- การส่งข้อมูล

การส่งข้อมูลจะเริ่มจากมีการเขียน (Write) ข้อมูลลงไปยังรีจิสเตอร์ SBUF ทำให้มีสัญญาณ Write to SBUF เกิดขึ้นแล้วข้อมูล 8 บิท จาก Internal Data Bus ของ 8051 ถูกเขียนลงไปในรีจิสเตอร์ SBUF และข้อมูลจากบิท TB8 ของรีจิสเตอร์ SCON จะเขียนลงไปยัง D FLIP FLOP เช่นกันรวมเป็น 9 บิท ดังรูป ที่เวลา S1P1 ของไซเคิลเครื่องแรก หลังจากสัญญาณ TX Clock นับครบ 1 รอบ จะทำให้สัญญาณ SEND ที่ออกจากวงจร TX Control เปลี่ยนสภาวะลอจิกเป็น 0 เพื่อเริ่มการส่งข้อมูลโดยจะเริ่มส่ง Start Bit (0) ออกไปทางขา TXD เมื่อสัญญาณ TX Clock เปลี่ยนเป็น 1 ครั้งต่อไปก็จะเริ่มส่งข้อมูลบิท 0 ออกไป และสัญญาณ DATA จะเป็น 1 ที่เวลานี้ด้วย TX Control จะส่งสัญญาณ Shift ไปยังรีจิสเตอร์ SBUF ทุกครั้งที่สัญญาณ TX Clock เป็น 1 เพื่อเลื่อนข้อมูลออกไปทางขวา และจะเลื่อน 0 เข้ามาทางซ้ายขณะที่บิท TB8 อยู่ทางขวาสุด เตรียมส่งออกนั้น ข้อมูลทางซ้ายทุกบิทจะเป็น 0 วงจร Zero Detector จะมีสัญญาณไปบอก TX Control ในการส่งข้อมูลบิทสุดท้าย คือ TB8 ออกไปตามเวลาของสัญญาณ TX Clock (Bit Time) เมื่อส่งข้อมูลบิท TB8 ออกไปเสร็จสิ้นแล้วก็จะรอเวลา 1 Bit

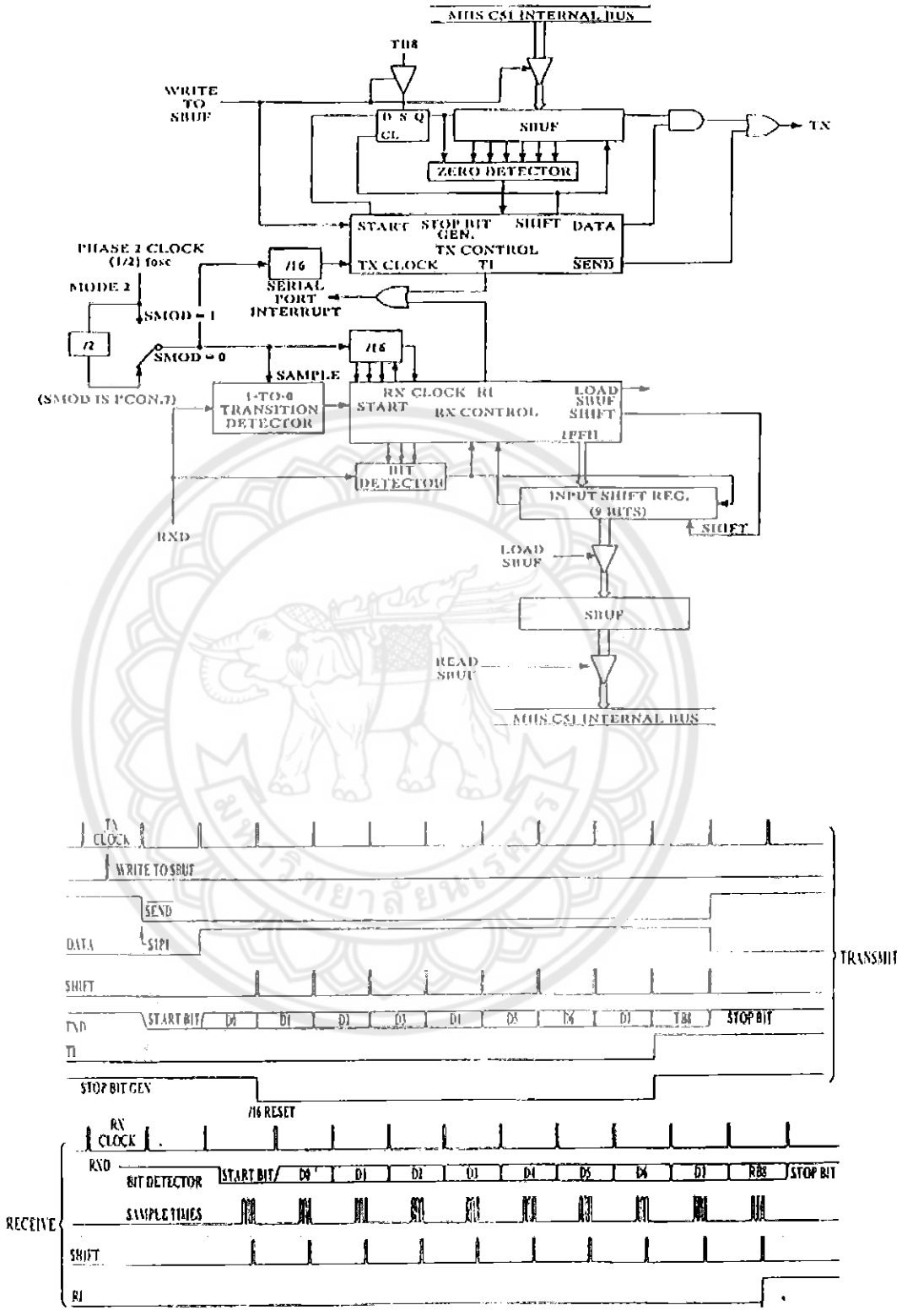
Time สัญญาณ SEND ก็จะกลับเป็น 1 และบิท TI ในรีจิสเตอร์ SCON จะเป็น 1 เพื่อบอกสิ้นสุดการส่งข้อมูลและสัญญาณ DATA ก็จะกลับเป็น 0 จากนั้นสัญญาณที่ออกจาก TXB ก็คือ Stop Bit นั่นเอง

- การรับข้อมูล

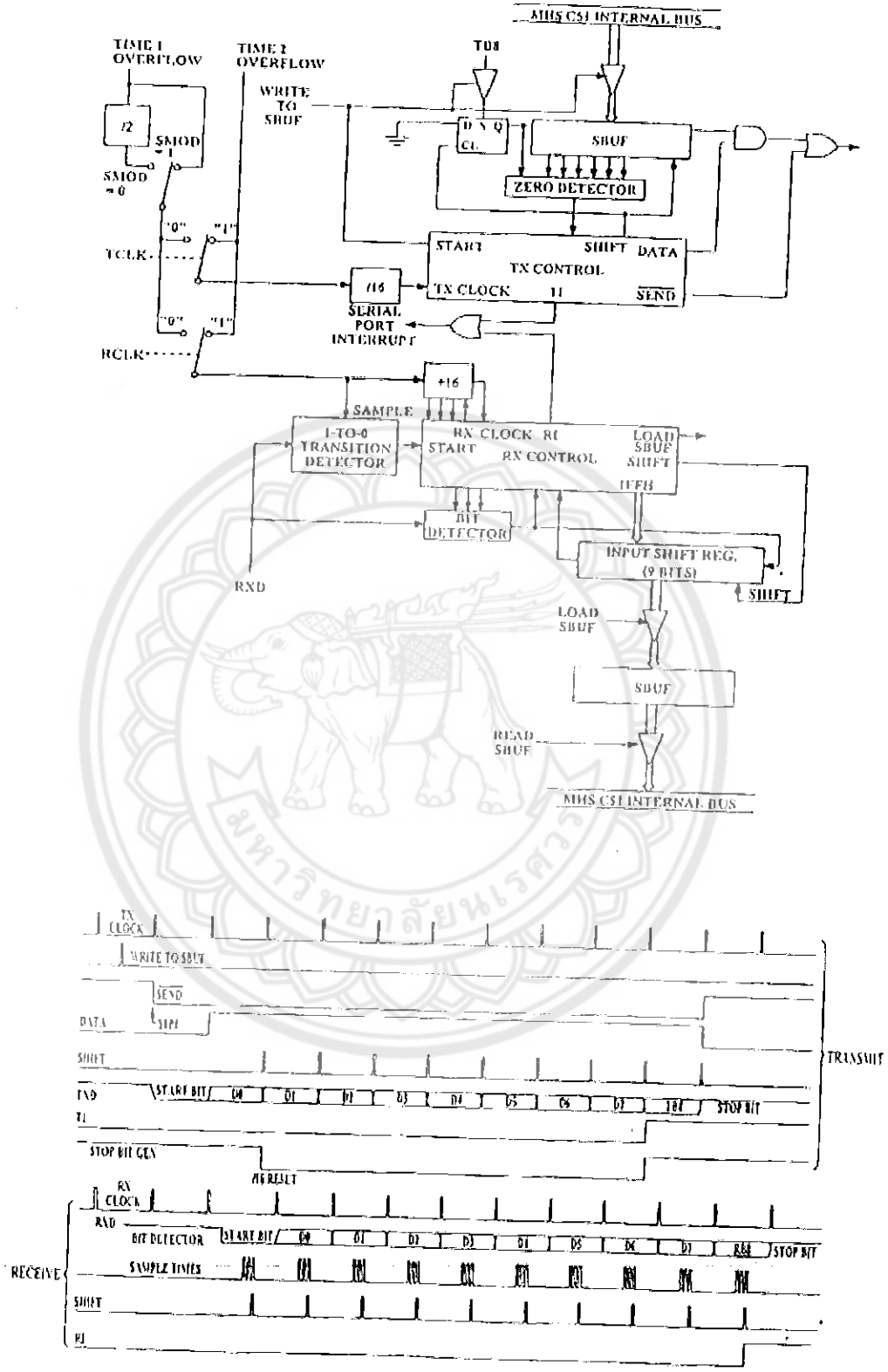
การรับข้อมูลที่เข้ามาทางขา RXD ของ 8051 จะเริ่มต้นเมื่อวงจร 1 – to – 0 Transition Detector ทำการตรวจสอบข้อมูลที่ขา RXD 16 เท่า ของอัตราการส่งข้อมูลแล้วพบว่าสัญญาณที่ขา RXD เปลี่ยนจาก 1 เป็น 0 ทำให้วงจรหาร 16 ถูกรีเซ็ทไปด้วย และจะเขียนข้อมูลในรีจีสเตอร์ SBUF เป็น 1 FFH (มี 9 บิท) เมื่อ Counter ในวงจรหาร 16 นับถึง 7, 8 และ 9 จะมีการตรวจสอบข้อมูลที่เข้ามาทางขา RXD ถ้า 2 ใน 3 เหมือนกัน ก็ถือว่าข้อมูลที่เข้ามาคือค่านั้น ถ้าข้อมูลที่รับเข้ามาบิทแรกไม่เป็น 0 คือไม่ใช่ Start Bit (เพราะอาจตรวจพบการเปลี่ยนจากสัญญาณขา RXD แต่การตรวจสอบการเปลี่ยนสถานะจาก 1 เป็น 0 ใหม่ แต่ถ้าถูกต้องก็จะรับข้อมูลเข้ามาทีละ 1 บิท ข้อมูลนี้จะถูกเลื่อนเข้าไปเก็บทางขวาของ Input Shift Register และ 1 จะถูกเลื่อนออกไปทางซ้ายโดยสัญญาณ Shift จนกระทั่งสัญญาณ Start Bit ซึ่งเป็น 0 ถูกเลื่อนเข้ามาถึงซ้ายสุดของ Input Shift Register จะบอกให้ RX Control รู้ว่าจะต้องมีข้อมูลบิทสุดท้ายเข้ามาอีก 1 บิท เมื่อข้อมูลบิทสุดท้ายเข้ามาจะเกิดการโหลดข้อมูลจาก Input Shift Rigister ไปยัง SBUF, RB8 และเซ็ท RI ให้เป็น 1 แต่สิ่งนี้จะเกิดขึ้นได้ต้องมีสภาวะดังนี้ด้วย

1. บิท R1 เป็น 0 และ
2. SM2 = 0 หรือถ้า SM2 = 1 ข้อมูลบิทที่ 9 ต้องเป็น 1

ถ้าไม่มีสภาวะดังกล่าวทั้ง 2 ข้อมูลที่รับมาจะถูกทิ้งไปไม่ถูกนำไปเก็บที่ SBUF และ RI ก็จะไม่ถูกเซ็ท แต่ถ้าเกิดขึ้นทั้ง 2 กรณี ข้อมูลจะถูกนำไปเก็บที่รีจิสเตอร์ SBUF 8 บิท และบิทสุดท้ายจะนำเก็บที่บิท RB8 ของรีจิสเตอร์ SCON หลังจากนั้น 1 Bit Time ไม่ว่าจะมีการเก็บข้อมูลหรือไม่ ก็จะเข้าสู่การตรวจสอบสถานะการเปลี่ยนสัญญาณจาก 1 เป็น 0 เพื่อเตรียมรับข้อมูลต่อไป



รูปที่ 2.20 Serial Port Mode 2

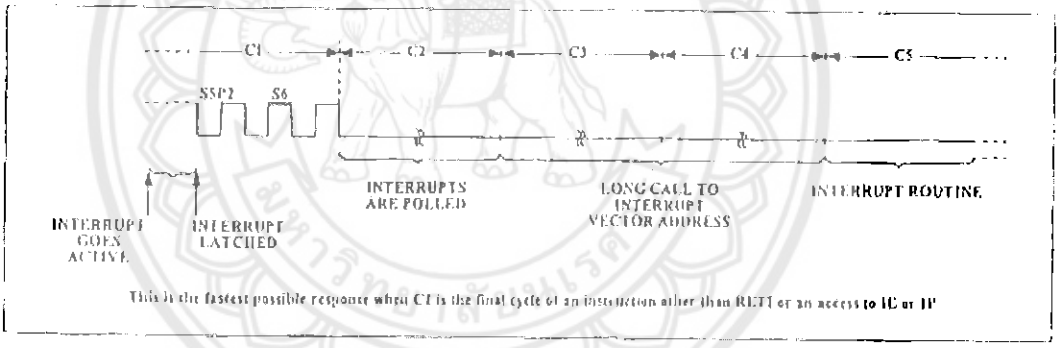


รูปที่ 2.21 Serial Port Mode 3

2.2.4 การขัดจังหวะ (Interrupt)

การขัดจังหวะ คือ สภาวะหนึ่งที่คอมพิวเตอร์กำลังทำงานอยู่แล้วถูกขัดจังหวะด้วยสัญญาณหรือคำสั่งพิเศษที่ทำให้คอมพิวเตอร์ต้องละจากงานที่กำลังทำอยู่ ไปทำงานในโปรแกรมตอบสนองการขัดจังหวะนั้น เมื่อเสร็จแล้วก็จะกลับมาทำงานเดิมต่อไปได้ ใน 8051 จะสามารถขัดจังหวะการทำงานได้ 6 แหล่ง คือ

1. INT0, TNT1 เป็น 2 ขาของ 8051 ที่จะรับสัญญาณจากภายนอก การขัดจังหวะจะเกิดขึ้นถ้าสัญญาณที่ขาดังกล่าวมีสภาวะลอจิกเป็น 0 หรือเปลี่ยนจาก 1 เป็น 0 โดยเลือกด้วยการกำหนดในบิท IT0 หรือ IT1 ในรีจิสเตอร์ TCON
2. TR0, TF1 เป็นบิทหนึ่งที่จะบอกการทำงานของ Timer 0, Timer 1 เมื่อเกิด Overflow ขึ้นใน Timer จะทำให้บิทนี้เป็น 1 และเกิดการขัดจังหวะการทำงานของ 8051
3. TI, RI เป็น 2 บิทในรีจิสเตอร์ SCON ถ้าบิทนี้ถูกเซ็ทให้เป็น 1 โดยฮาร์ดแวร์อันเนื่องจากเสร็จสิ้นการส่งหรือรับข้อมูลจะสามารถทำให้เกิดการขัดจังหวะได้



รูปที่ 2.22 ไดอะแกรมเวลาของการตอบสนองการขัดจังหวะ

8051 จะทำการอ่านสัญญาณจากทั้ง 6 แหล่งที่เวลา S5P2 ของทุก ๆ ไซเคิลของเครื่อง (Machine Cycle) เข้ามาเก็บ และในช่วงของไซเคิลของเครื่องถัดไปก็จะตรวจสอบสถานะของสัญญาณทั้ง 6 ที่เก็บเข้ามาถ้าสัญญาณนั้นมีการขัดจังหวะที่ถูกต้อง 8051 ก็จะละทิ้งการทำงานเดิมไว้ชั่วคราวแล้วสร้างคำสั่ง LCALL ขึ้นมาภายใน 8051 เพื่อไปทำงานในโปรแกรมตอบสนองการขัดจังหวะแต่ถ้าสัญญาณนั้นเมื่อทำงานในโปรแกรมตอบสนองการขัดจังหวะเสร็จสิ้น ก็จะสามารถกลับมาทำงานเดิมได้โดยใช้คำสั่ง RETI เป็นคำสั่งสุดท้ายในโปรแกรมตอบสนองการขัดจังหวะ สัญญาณขัดจังหวะจากแต่ละแหล่งจะมีตำแหน่งหน่วยความจำที่จะเก็บโปรแกรม ตอบสนองการขัดจังหวะไว้ต่างกันดังนี้

สัญญาณที่ขอขัดจังหวะ	ตำแหน่งเริ่มต้นโปรแกรมตอบสนองการขัดจังหวะ	
1	INT0	0003H
2	TF0	000BH
3	INT1	0013H
4	TF1	001BH
5	TI, RI	0023H

2.3 การสื่อสารอนุกรม

2.3.1 รูปแบบข้อมูลในคอมพิวเตอร์

การที่จะทำความเข้าใจการส่งผ่านข้อมูลสิ่งแรกคือต้องทำความเข้าใจกับวิธีที่ข้อมูลถูกเก็บไว้ภายในคอมพิวเตอร์ก่อน

2.3.2 บิทและไบท์

ในเลขฐานสิบ มีตัวเลขอยู่สิบตัวคือ 0 ถึง 9 การเพิ่มศูนย์หนึ่งตัวเข้าทางท้ายเป็นการคูณจำนวนด้วยสิบในเลขฐานสองมีเพียงตัวเลขสองตัวคือ 0 กับ 1 การเพิ่มศูนย์เข้าทางท้ายจำนวนเป็นการคูณจำนวนด้วยสอง

ตัวเลขศูนย์หรือหนึ่งแต่ละตัวในเลขฐานสองเรียกว่า บิท (bit) 8 บิทจะเป็น 1 ไบท์ (byte) ผลที่ตามมาคือ ค่าของหนึ่งไบท์จึงเป็นได้ตั้งแต่ 00000000 ถึง 11111111 หรือ 0 ถึง 255 ในฐานสิบบิทที่อยู่ทางขวาสุดของไบท์เรียกว่า บิทศูนย์ บิทที่อยู่ซ้ายสุดเรียกว่า บิทเจ็ด บิทศูนย์เรียกว่าบิทที่มีนัยสำคัญต่ำสุด (least significant bit) และ บิทเจ็ด เรียกว่า บิทที่มีนัยสำคัญสูงสุด (most significant bit)

คอมพิวเตอร์เกือบทั้งหมดทำงานในระบบเลขฐานสอง เพราะว่ามันเป็นการง่ายที่จะแปลงรหัส 0 และ 1 เป็นแรงดันไฟฟ้าบวกและลบ ในคอมพิวเตอร์ส่วนใหญ่หน่วยที่เล็กที่สุดของหน่วยความจำที่อ้างอิงได้โดยการอ้างแอดเดรส คือ ไบท์ ดังนั้น เมื่อข้อมูลถูกเก็บและจัดการในคอมพิวเตอร์ ตามปกติจึงมันถูกแปลงให้เป็นไบท์ที่เรียงลำดับกัน

2.3.3 การเข้ารหัสข้อความ

เมื่อข้อความ (อักขระเครื่องหมายวรรคตอนและอื่น ๆ) ถูกเก็บในคอมพิวเตอร์แต่ละตัวอักษรที่แตกต่างกันจะถูกแทนด้วยจำนวนที่ต่างกันจำนวนเหล่านี้โดยปกติมีค่าจาก 0 ถึง 127 หรือจาก 0 ถึง 255 เนื่องจากไบท์หนึ่งสามารถมีค่าจาก 0 ถึง 225 มันจึงเป็นธรรมชาติที่จะให้หนึ่งไบท์แทนตัวอักษรหรือเครื่องหมายวรรคตอนแต่ละตัวในข้อมูลที่เป็นข้อความ มีสองวิธีที่ต่างกันสำหรับการ จับคู่ตัวอักษรกับจำนวน คือ EBCDIC (Extended Binary Coded Decimal Interchange Code) ซึ่งถูกใช้ในคอมพิวเตอร์ ชนิดอื่นของไอบีเอ็มยกเว้นไอบีเอ็มพีซี ASCII (American Standard Code For Information Interchange) ซึ่งถูกใช้ในคอมพิวเตอร์ส่วนใหญ่ ตาราง ASCII อย่างเป็นทาง

การให้จำนวนระหว่าง 32 ถึง 126 แทนตัวเลข อักษร เครื่องหมายวรรคตอนและสัญลักษณ์ที่ใช้กันทั่วไปอื่น ๆ จำนวนจาก 0 ถึง 31 และ 127 มีความหมายพิเศษ เช่น Carriage return, Line feed และตัวอักษรที่ไม่สามารถแสดงผลได้อื่น ๆ ตัวอย่างเช่น ตัว A ถูกเก็บเป็นเลขฐานสิบ 65 ในฐานสองคือ 01000001 คอมมาถูกเก็บในเลขฐานสิบ 44 ซึ่งเป็น 00101100 ในฐานสองเนื่องจากจำนวน 127 สามารถถูกเก็บในหนึ่งไบท์ โดยจะเหลืออีกหนึ่งบิท เนื่องจากเราให้ชื่อบิทในไบท์หนึ่งตั้งแต่ศูนย์ถึงเจ็ด จะเป็นได้ว่ารหัส ASCII ใช้ได้เพียงบิทศูนย์ถึงหก บิทเจ็ดถูกสำรองไว้

ตารางที่ 2.8 แสดงจำนวน 35 ในเลขฐานสอง

หมายเลขบิท	7	6	5	4	3	2	1	0
ค่าถ้าถูกเซ็ท	128	64	32	16	8	4	2	1
การเซ็ท	0	0	1	0	0	0	1	1
ค่าตามที่เซ็ท	0	0	32	0	0	0	2	1

คอมพิวเตอร์หลายชนิดใช้เต็มที่ทั้งแปดบิทสำหรับการเข้ารหัส ทำให้มีรหัสที่แตกต่างกัน 256 ตัว 128 ตัวแรกเป็นไปตาม ASCII และส่วนที่เหลือถูกใช้สำหรับอักขระต่างชาติ สัญลักษณ์ทางคณิตศาสตร์ อักษรกราฟฟิก และอื่น ๆ ตามแต่การออกแบบ โชคไม่ดีที่ไม่มีมาตรฐานสำหรับอักขระเพิ่มเติม (Extended Character) ซึ่งมักจะมีความหมายแตกต่างกันบนคอมพิวเตอร์คนละชนิด

2.3.4 รหัส ASCII ชนิดพิเศษ

รหัส 32 ตัวแรกในตาราง ASCII มีความหมายพิเศษ ดังในตารางที่จะนำเสนอต่อไป มีหลายตัวได้รับการออกแบบเพื่อวัตถุประสงค์ทางการสื่อสารโดยเฉพาะ

รหัส 1 ถึง 26 ถูกอ้างเป็น Ctrl-A ถึงCtrl-Z ด้วยเช่นกัน และพวกมันสามารถถูกสร้างด้วยแป้นพิมพ์ของคอมพิวเตอร์ โดยการกดปุ่ม Ctrl ค้างไว้ และกดปุ่มตัวอักษรที่เหมาะสมพร้อมกัน

ตาราง ที่ 2.9 รหัส ASCII ชนิดพิเศษ

รหัส	อักขระ	ความหมาย
0	NULL	วิธีหนึ่งที่จะทำให้เกิดการหน่างเวลาอย่างจงใจ ในอดีตมันมีความจำเป็นที่จะส่ง null หลังจาก carriage return เพื่อให้เครื่องพิมพ์ปัดแคร่ไปทางซ้ายสุดของหน้ากระดาษ ปัจจุบันเครื่องพิมพ์ทำงานได้เร็วขึ้น null จึงถูกใช้เพื่อจุดประสงค์อื่นหลายอย่าง
1	SQH	Start of heading แสดงว่าข้อความที่ตามมาเป็นส่วนหนึ่งของหัวข้อ
2	STX	Start of text แสดงจุดเริ่มต้นของข้อความจริงของข่าวสาร
3	ETX	End of text แสดงจุดสิ้นสุดของข้อความ
4	EOT	End of transmission แสดงการสิ้นสุดการส่ง
5	ENQ	Enquiry โดยปกติถูกใช้เป็นส่วนหนึ่งของซอฟแวร์ แฮนด์เช็คกิ้ง ในการขอให้คอมพิวเตอร์ฝ่ายรับ ตอบรับการได้รับข่าวสาร
6	ACK	Acknowledge การตอบรับการได้รับข่าวสาร
7	BEL	ส่งเสียงออกทางเทอร์มินัล
8	BS	Backspace
9	HT	Horizontal tab
10	LF	Line feed ทำให้ขึ้นบรรทัดใหม่ในตำแหน่งเดิม
11	VT	Vertical Tab
12	FF	From Feed เลื่อนหน้ากระดาษไปหนึ่งหน้า
13	CR	Carriage return เลื่อนไปที่ต้นบรรทัด บางครั้งทำให้เกิด Line Feed
14	SO	Shif out กำหนดจุดเริ่มต้นของรหัสควบคุมพิเศษ บ่อยครั้งที่ใช้ Esc แทน
15	SI	Switch in กำหนดจุดสิ้นสุดของรหัสควบคุมที่เริ่มต้นโดย SO
16	DLE	Data Link Escape เหมือน Esc
17	DC1	
18	DC2	Device Control 1 ถึง 4 รหัสสำรองไว้ให้ใช้ตามความต้องการ บางครั้งใช้ในซอฟต์แวร์แฮนด์เช็คกิ้ง
19	DC3	
20	DC4	
21	NAK	Negative acknowledgement บ่งชี้ว่าข้อมูลที่ส่งไปนั้นไม่ได้ถูกรับอย่างถูกต้อง ตัวอย่างเช่น พบความผิดพลาดทางพาริตี้

ตาราง ที่ 2.9 รหัส ASCII ชนิดพิเศษ (ต่อ)

รหัส	อักขระ	ความหมาย
22	SYN	Synchonous idele เหมือน NULL แต่ถูกใช้ในการสื่อสารแบบซิงโครนัส เพื่อดูแล ให้อุปกรณ์สองตัวซิงโครไนต์กัน ระหว่างการส่ง
23	ETB	End of transmission block ถูกใช้ในที่ซึ่งการส่งข้อมูลถูกแบ่งออกเป็นบล็อก เพื่อวัตถุประสงค์ในการตรวจสอบข้อผิดพลาด
24	CAN	Cancel บ่งชี้ว่า ข้อมูลที่ถูกส่งไปควรถูกทิ้งไป
25	EM	End of medium บ่งชี้ว่ามาถึงปลายทางของเทปกระดาษ
26	SUB	Substitute แก้ไขตัวอักษรที่ถูกส่งการผิดพลาด ถูกใช้เพื่อบ่งชี้จุดสิ้นสุดของการส่งด้วยเช่นการ
27	ESC	Escape บ่งชี้จุดเริ่มของตัวอักษรที่ติดตามมาว่ามีความหมายพิเศษ
28	FS	
29	GS	File group, Record และ Unit separator ตามลำดับใช้ เพื่อกำหนดขอบเขตระหว่างส่วนของข้อความ
30	RS	
31	US	
32	DEL	บ่งชี้ว่า ตัวอักษรที่มาก่อนมันควรถูกลบ

(ดังนั้น 1 = Ctrl-A,2=Ctrl-B เป็นต้น) บางรหัสสามารถถูกป้อนเข้าโดยการกดปุ่มเฉพาะ เช่น Tab สำหรับ รหัส 9 หรือ Return สำหรับรหัส 13

2.3.5 การเข้ารหัสข้อมูลที่ไม่ใช่ข้อความ

แน่นอนว่าทุกอย่างที่ถูกเก็บในคอมพิวเตอร์ไม่ได้อยู่ในรูปของข้อความเสมอไป คำสั่งของโปรแกรม ข้อมูลตัวเลข และกราฟิกอิมเมจ เป็นตัวอย่างข้อมูลที่ไม่ได้ถูกเก็บในรูปแบบ ASCII ข้อมูลประเภทนี้โดยปกติถูกเข้ารหัสให้ใช้ทุกค่าที่เป็นไปได้ของหนึ่งไบท์ จำนวนถูกเก็บในรูปแบบไบนารี และสามารถขยายไปเป็นหลายไบท์ คำสั่งของโปรแกรมมักจะประกอบด้วยหนึ่งหรือสองไบท์ เราเรียกข้อมูลประเภทนี้ว่า ข้อมูลไบนารี (Binary Data) (แม้ว่าข้อความจะถูกเก็บในรูปแบบไบนารีเช่นกัน)

เนื่องจากไบท์ที่เก็บข้อมูลซึ่งไม่ใช่ข้อความสามารถเป็นค่าใด ๆ ก็ได้ ในเวลาที่มันตรงกับค่าที่มีความหมายพิเศษในตาราง ASCII ทำให้เกิดความยุ่งยากในการส่งข้อมูล ถ้าอุปกรณ์ฝ่ายรับเกิดแปลไบท์ที่ไม่ใช่ข้อความว่าหมายถึงสิ้นสุดข่าวสาร ในกรณีที่ข้อมูลไม่สามารถถูกส่งในรูปแบบ

ข้อมูลดิบ เพราะว่าไบท์ที่อยู่กลางข่าวสารอาจตรงกับสัญลักษณ์สิ้นสุดข่าวสารโดยบังเอิญ และทำให้อุปกรณ์ฝ่ายรับหยุดรับข้อมูล การแก้ไขปัญหานี้ต้องมีโปรโตคอลที่แน่นอน

2.3.6 การแปลงเป็นรูปแบบอนุกรม

คอมพิวเตอร์เกือบทั้งหมดเก็บและจัดการข้อมูลในแบบขนาน หมายความว่าเมื่อไบท์หนึ่งถูกส่งจากส่วนหนึ่งของคอมพิวเตอร์ไปยังส่วนอื่น มันไม่ได้ถูกส่งไปครั้งละหนึ่งบิท แต่จะถูกส่งไปหลายบิทพร้อมกันผ่านตัวนำในแบบขนาน จำนวนบิทที่ถูกส่งในครั้งหนึ่งแปรผันไปตามเครื่อง แต่โดยปกติจะเป็นแปดหรือทวีคูณของแปด เพราะฉะนั้น คอมพิวเตอร์สามารถทำงานกับหนึ่งไบท์เป็นอย่างน้อยในครั้งหนึ่ง ๆ

เนื่องจากการสื่อสารจากคอมพิวเตอร์ไปยังอุปกรณ์อื่นหลายชนิด เป็นแบบอนุกรมหมายความว่าข้อมูลถูกส่งไปทีละหนึ่งบิท ตัวเชื่อมต่อการสื่อสารต้องสามารถนำไบท์ที่รับมาแบบขนานส่งออกไปทีละบิทได้

จากที่กล่าวมาแล้วว่าสายข้อมูลในการสื่อสารแบบอนุกรม มีเพียงสภาวะ MARK และ SPACE ซึ่งในกรณีของการเชื่อมต่อโดยตรงเท่ากับแรงดันไฟฟ้าลบหรือบวกตามลำดับ ข้อมูลใด ๆ ที่ถูกส่งต้องถูกแปลงให้เป็นลำดับของ MARK และ SPACE ก่อน สำหรับการส่งข้อมูล MARK แทนค่าหนึ่ง และ SPACE แทนค่าศูนย์

2.3.7 การสื่อสารแบบซิงโครนัสและอะซิงโครนัส

เมื่อข้อมูลถูกแปลงให้เป็นรูปแบบอนุกรมแล้วมีวิธีการส่งข้อมูลอยู่สองแบบคือ ซิงโครนัส (Synchronous) และอะซิงโครนัส (Asynchronous) เมื่อข้อมูลถูกส่งมาจากการพิมพ์ที่แป้นพิมพ์การส่งและรับจะเป็นแบบอะซิงโครนัส คือคนที่พิมพ์ไม่สามารถที่จะพิมพ์ได้อย่างต่อเนื่อง ดังนั้นเมื่อคอมพิวเตอร์รับตัวอักษรแต่ละตัวจะมีช่องว่างระหว่างตัวอักษรที่ไม่สม่ำเสมอ ทำให้อุปกรณ์ฝ่ายรับไม่อาจคาดหมายได้ว่า ตัวอักษรต่อไปจะมาถึงเมื่อใด จากการขาดความต่อเนื่องนี้ จึงมีความจำเป็นต้องใส่บิทพิเศษก่อนและหลังตัวอักษรแต่ละตัว เพื่อบ่งบอกจุดเริ่มต้นและสิ้นสุดของตัวอักษร

บิทพิเศษนี้เรียกว่า บิทเริ่มต้น (Start Bit) นอกจากนี้ ยังมีอีกบิทหนึ่งคือ บิทพาริตี้ (Parity Bit)ที่มักจะถูกใส่เพิ่มเข้าไปเพื่อใช้ตรวจสอบความผิดพลาด วิธีนี้เรียกว่า การสื่อสารแบบอะซิงโครนัส (Asynchronous Communication)

เมื่อตัวอักษรถูกส่งไปเป็นกลุ่มตามความเร็วของเครื่อง ช่วงห่างระหว่างกันก็จะสม่ำเสมอ จึงไม่มีความจำเป็นต้องมีบิทเริ่มต้นและบิทจบสำหรับตัวอักษรแต่ละตัว เพราะว่าเมื่อตัวอักษรแรกถูกรับไป อุปกรณ์ฝ่ายรับสามารถคาดหมายการมาถึงของตัวอักษรถัดไปได้ กล่าวอีนัยหนึ่ง คือ มันสามารถเข้าจังหวะตัวมันเองกับคอมพิวเตอร์ฝ่ายส่งได้ วิธีแบบนี้เรียกว่า การสื่อสารแบบซิงโครนัส (Synchronous Communication)

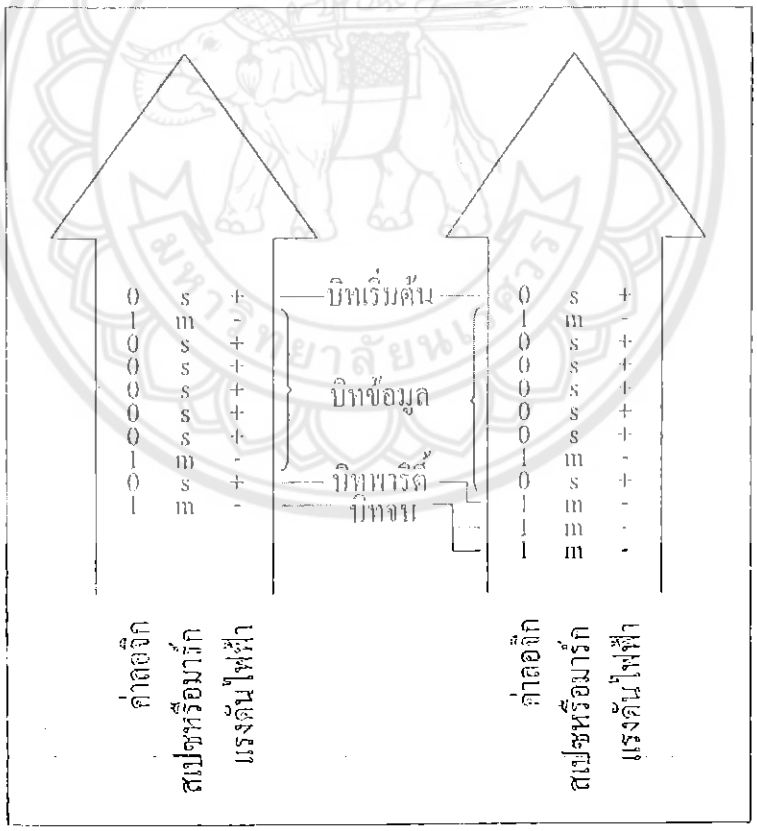
เนื่องจากการสื่อสารแบบอะซิงโครนัสต้องการบิทเริ่มต้นและบิทจบเพิ่มเข้าไปในแต่ละตัวอักษร จึงมีความยาวในการส่งไฟล์มากว่าการสื่อสารแบบซิงโครนัส ประมาณ 20 เปอร์เซ็นต์ ความแตกต่างนี้อาจสังเกตไม่เห็นเมื่อแหล่งข้อมูลที่ส่งมาจากการพิมพ์ที่เทอร์มินัล

นอกจากนี้ในโลกของไอบีเอ็มเมนเฟรมซึ่งเทอร์มินัลแบบซิงโครนัสเป็นอุปกรณ์สามัญ การสื่อสารแบบอนุกรมส่วนใหญ่ เกิดขึ้นในแบบอะซิงโครนัสซึ่งประยุกต์เข้ากับการสื่อสารเกือบทั้งหมดระหว่างไมโครคอมพิวเตอร์เทอร์มินัล และระบบยูนิกซ์ ด้วยเหตุนี้ส่วนที่เหลือของหัวข้อนี้จะให้ความสนใจที่การสื่อสารแบบอะซิงโครนัสเป็นหลัก

2.3.8 การจัดเฟรม

ในกรณีการสื่อสารแบบอะซิงโครนัส บิทที่เป็นตัวแทนของหนึ่งไบท์ ซึ่งเรียกว่าบิทข้อมูล (Data Bit) จะถูกนำและตามด้วยบิทเริ่มต้น บิทจบและบิทพาริตี้ กระบวนการนี้เรียกว่า การจัดเฟรม (Framing)

จำนวนของบิทที่แทนหนึ่งตัวอักษรแปรผันไปตามโปรโตคอลสื่อสารที่ใช้ จำสวนที่ว่าหมายถึงจำนวนของบิทข้อมล หรือความยาวเวิร์ด



รูปที่ 2.23 ตัวอย่างการส่งตัวอักษร A สองแบบ

2.3.9 บิทเริ่มต้น

บิทเริ่มต้นถูกใส่เพิ่มที่จุดเริ่มต้นของเฟรมเสมอ เพื่อเตือนอุปกรณ์ฝ่ายรับว่าข้อมูลกลับมาถึงและเพื่อเข้าจังหวะกลไกที่แยกแต่ละบิท บิทเริ่มต้นคือ SPACE หรือไบนารี 0

ในการเชื่อมต่อโดยตรง SPACE หรือ 0 ถูกส่งเป็นแรงดันไฟฟ้าบวกแรงดันไฟฟ้าระหว่างเฟรมจะเป็นลบ ดังนั้นที่จุดเริ่มต้นของแต่ละเฟรมแรงดันไฟฟ้าจะเปลี่ยนจากลบเป็นบวก

2.3.10 บิทข้อมูล

มาตรฐานหรือโปรโตคอลการสื่อสารแบบอนุกรม ทำให้เกิดการส่งตัวอกษรที่ยาวต่างกันเมื่อซอฟแวร์สื่อสารให้คุณเลือกความยาวเวิร์ด มันกำลังถามว่าคุณต้องการส่งตัวอักษรเจ็ดบิทหรือแปดบิท(บางครั้งความยาวอื่นถูกใช้แต่แทบไม่ค่อยมี) ถ้าข้อมูลทั้งหมดถูกส่งไปแบบ ASCII เวิร์ดขนาดเจ็ดบิทก็เพียงพอ จำไว้ว่าตาราง ASCII กำหนดจำนวนจาก 0 ถึง 127 ซึ่งทั้งหมดสามารถแทนได้ด้วยเจ็ดบิท

ถ้าข้อมูลที่ส่งไม่ใช่ ASCII (เช่นข้อความที่ใช้ชุดอักขระเพิ่มเติมหรือข้อมูลไบนารี) ทั้งแปดบิทของแต่ละไบท์จึงมีความจำเป็น คุณไม่สามารถใช้โปรโตคอลเจ็ดบิทได้ ถ้าข้อมูลไม่ถูกแปลงเป็นรูปแบบเจ็ดบิทเสียก่อน

2.3.11 บิทพาริตี้

การตรวจสอบพาริตี้เป็นวิธีหนึ่งในการทดสอบว่า ข้อมูลที่ส่งได้ถูกรับไปอย่างถูกต้องหรือไม่ อุปกรณ์ฝ่ายส่งจะเพิ่มบิทพาริตี้อีกหนึ่งบิท เป็นค่า 0 หรือ 1 ขึ้นอยู่กับบิทข้อมูล อุปกรณ์ฝ่ายรับจะตรวจสอบว่าบิทพาริตี้มีความสัมพันธ์ที่ถูกต้องกับบิทอื่นหรือไม่ ถ้าไม่ แสดงว่าบางสิ่งต้องผิดพลาดในระหว่างการส่ง พาริตี้สามารถคำนวณได้จากวิธีต่อไปนี้

- พาริตี้คู่ (Even Parity) หมายความว่า จำนวนของบิทข้อมูลที่เป็น 1 และค่าของบิทพาริตี้รวมกัน เป็นคู่ เช่น ตัว A ในฐานสองคือ 01000001 เมื่อนับจำนวนของบิทที่เป็น 1 จะได้ 2 ซึ่งเป็นเลขคู่ ดังนั้นบิทพาริตี้ต้องเป็น 0 ถ้าตัวอักษร A ที่รับได้มีพาริตี้เป็น 1 แสดงว่าเกิดความผิดพลาดในระหว่างการส่ง
- พาริตี้คี่ (Odd Parity) หมายความว่า จำนวนทั้งหมดของบิทข้อมูลที่เป็น 1 บวกกับค่าของบิทพาริตี้ เป็นจำนวนคี่ ดังนั้นสำหรับตัวอักษร A บิทพาริตี้ควรถูกเซ็ทเป็น 1 เพื่อให้จำนวนของบิทที่เป็น 1 ทั้งหมดเป็น 3 ซึ่งเป็นจำนวนคี่
- ไม่มีพาริตี้ (Null Parity) หมายถึงไม่มีพาริตี้
- SPACE (บางครั้งเรียกว่า Bit Trimmimg) คือ บิทพาริตี้ที่เป็น 0 เสมอมีประโยชน์ในการตรวจสอบข้อผิดพลาดบางอย่าง เมื่อการส่งข้อมูลเป็นขยะมาก บางครั้งบิทพาริตี้อาจกลายเป็น 1 แสดงว่า เกิดข้อผิดพลาดพาริตี้แบบนี้สามารถใช้เพื่อส่งอักษรเจ็ดบิท

ให้กับอุปกรณ์ที่ต้องการตัวอักษรแปดบิทได้เช่นกัน อุปกรณ์ฝ่ายรับจะถือว่า บิทพาริตี้เป็นบิทสุดท้ายของข้อมูล

- MARK (บางครั้งเรียกว่า Bit Forcing) ทำงานเหมือนกับพาริตี้แบบ SPACE ยกเว้นแต่บิทพาริตี้จะเป็น 1 เสมอ เนื่องจาก 1 ในตำแหน่งนั้นสามารถที่จะถูกตีความรวมเข้ากับค่าของจำนวนได้ อุปกรณ์ หรือคอมพิวเตอร์ฝ่ายรับต้องถูกโปรแกรมไม่ให้สนใจมัน

2.3.12 บิทจบ

สองบิทอย่างน้อยต้องมีหนึ่งบิทเสมอ เพื่อประกันว่ามีแรงดันไฟฟ้าลบอย่างน้อยเป็นช่วงเวลาหนึ่งก่อนที่เฟรมถัดไปจะมาถึงเพื่อที่จะสามารถแยกแยะเฟรมถัดไปได้ จากบิทเริ่มต้นที่เป็นบวกของมันบิทจบมากกว่าหนึ่งบิทโดยทั่วไปจะใช้เมื่ออุปกรณ์ฝ่ายรับต้างการเวลาเพิ่มขึ้นก่อนที่มันจะสามารถจัดการกับตัวอักษรที่เข้ามาตัวถัดไปได้

หนึ่งบิทครึ่ง หมายความว่า ความยาวของบิทนั้นมากกว่าบิทปกติ บิทจบบังคับให้มีช่องว่างอย่างน้อยระหว่างเฟรม พวกมันถูกส่งเป็นไบนารีหนึ่งซึ่งในการเชื่อมต่อโดยตรงจะเป็นแรงดันไฟฟ้าลบ

บิทจบสองบิทมักจะถูกใช้ที่อัตราบอด 110 ซึ่งเป็นอัตราการส่งข้อมูลต่ำสุด ที่ใช้กันทั่วไป เพื่อให้สอดคล้องกับความต้องการของเครื่องโทรพิมพ์รุ่นเก่าซึ่งใช้อัตราบอดต่ำและต้องการเวลาพิเศษเพื่อประมวลตัวอักษร

2.3.13 เบรก

ดังที่ได้อธิบายมาก่อนเมื่อกล่าวถึงบิทเริ่มต้นว่า ระหว่างตัวอักษร สายข้อมูลโดยปกติอยู่สภาวะ MARK (แรงดันไฟฟ้าลบ,ไบนารีหนึ่ง) ถ้าตัวอักษรประกอบด้วยศูนย์ทั้งหมด พร้อมด้วยแปดบิทข้อมูลและพาริตี้คู่ สภาวะ SPACE จะปรากฏอยู่สิบบิท คือ บิทเริ่มต้น บิทข้อมูลทั้งแปดและบิทพาริตี้ ซึ่งเป็นสภาวะ SPACE ที่ยาวที่สุด ก่อนจะสิ้นสุดเมื่อถึงบิทจบ ดังนั้น ที่อัตรา 150 บิทต่อวินาที สภาวะ SPACE ตามปกติจะไม่มากไปกว่า 1/15 วินาที หรือ 66067 มิลลิวินาที

สภาวะ SPACE ที่นานกว่านี้ โดยปกติเป็น 100 ถึง 600 มิลลิวินาทีถูกใช้เป็นสัญญาณพิเศษเรียกว่า เบรก (Break) เบรกบางครั้งถูกใช้เสมือนกับ Ctrl-C ของเมนเฟรมบนพีซี มันจะขัดจังหวะไม่ว่าโปรแกรมอะไรกำลังทำงานอยู่ และกลับคืนสู่ระบบปฏิบัติการ หรือ เมนูที่อยู่ในระดับบนภายในโปรแกรม เช่นเดียวกับ Ctrl-C หรือ Break มันมีประโยชน์สำหรับการหนีออกจากโปรแกรมที่เข้ามาลูปไม่รู้จบ

2.3.14 อัตราบอด

อัตราบอด (Baud Rate) แสดงจำนวนของสัญญาณแต่ละหน่วยในหนึ่งหน่วยวินาที มันถูกตั้งชื่อตาม Baudot ซึ่งเป็นผู้บุกเบิกการสื่อสารชาวฝรั่งเศส ในการออกแบบไบนารีมันเป็นสิ่งเดียวกับบิทต่อวินาที (bps) หรือจำนวนของเลขฐานสองที่ถูกส่งในหนึ่งวินาที ทั้งสองคำนี้มีความแตก

ต่างกัน แต่มันมักจะทำให้สับสน 200,000 คน อาจบอกว่าพวกเขามีโมเด็ม 1200 บอด และไม่มีสักคนที่มีจริง ๆ ที่จริงแล้วพวกเขามีโมเด็ม 1200 bps

ในการเชื่อมต่อ RS 232 โดยตรงสัญญาณจะเป็นหนึ่งในสองสถาน ในเวลาขณะใดขณะหนึ่งอัตราบอดและ bps จึงเท่ากัน อย่างไรก็ตามจะเห็นได้ว่าเมื่อสัญญาณหนึ่งถูกส่งผ่านระหว่างโมเด็ม มันสามารถเป็นหนึ่งในหลายสถานะ ความยาวของสัญญาณอาจเป็น 1/600 วินาที (600 บอด) แต่เนื่องจากมากกว่าสองบิทของข้อมูลสามารถถูกส่งไป พร้อมกับการเปลี่ยนแปลงแต่ละสภาวะอัตราบิทต่อวินาทีจะสูงกว่าอัตราบอด

มีจุดน่าสังเกตคือทั้งอัตราบอดและ bps อ้างถึงอัตราที่บิทภายในหนึ่งเฟรมถูกส่ง ช่องว่างระหว่างเฟรมอาจมีความยาวแปรเปลี่ยนได้ เช่นจากการพิมพ์ตัวอักษรด้วยอัตราแตกต่างกัน ดังนั้นทั้งอัตราและ bps จึงไม่ได้หมายถึงอัตราที่ข้อมูลถูกส่งไปจริง ๆ

อัตราบิทต่อวินาทีโดยทั่วไปอยู่ในอนุกรม 100,150,300,600,1200,2400,4800,9600 และ 19200 อัตราที่ใช้กันมากที่สุดสำหรับการสื่อสารทางโมเด็มคือ 1200 และ 2400 อัตรา 1200 ใช้กันมากสำหรับการสื่อสารระหว่างคอมพิวเตอร์กับเครื่องพิมพ์และ 9600 นั้นจะใช้กันมากสำหรับการเชื่อมต่อเทอร์มินัลกับคอมพิวเตอร์

2.3.15 การแก้ปัญหา

เมื่ออุปกรณ์สองตัวสื่อสารซึ่งกันและกัน พวกมันต้องตกลงกันในเรื่อง อัตราบอด ความยาวเวิร์ด จำนวนบิทจบ และพาริตี้ ถ้าพบว่าไม่ได้รับอะไรเลย ความผิดพลาดอาจอยู่ที่การเชื่อมต่อทางกายภาพ เช่น ข้อมูลกำลังถูกส่งบนสายผิดเส้น สายขาด หรือไม่ได้รับสัญญาณแฮนด์เช็คกิ้งที่ถูกต้อง ถ้าได้รับขยะ ความผิดพลาดอาจอยู่ในหัวข้อที่กล่าวต่อไปนี้

2.3.16 อัตราบอดไม่ตรงกัน

ถ้าอุปกรณ์สองตัวถูกตั้งอัตราบอดต่างกัน อุปกรณ์ฝ่ายรับอาจพยายามที่จะแปลข้อมูล (ถ้ามันไม่ได้ถูกโปรแกรมให้รายงานข้อผิดพลาดทางพาริตี้และทางเฟรม) โดยปกติคุณจะเห็นว่าจำนวนข้อมูลที่ได้รับแตกต่างจากที่ถูกส่งมา

2.3.17 ความผิดพลาดทางพาริตี้

ความผิดพลาดทางพาริตี้ (Parity Error) บ่งบอกว่าข้อมูลถูกทำลายในระหว่างการส่งอย่างไรก็ตาม มันอาจหมายความว่าอุปกรณ์ทั้งสองไม่ได้ถูกตั้งให้มีพาริตี้ (คู่,คี่,หรือไม่มี) หรือความยาวเวิร์ดตรงกัน

2.3.18 ความยายเวิร์ดไม่ตรงกัน

ถ้าเวิร์ดขนาดแปดบิทกำลังถูกส่งและอุปกรณ์ฝ่ายรับคาดหวังที่จะรับเวิร์ดขนาดเจ็ดบิท คุณอาจไม่พลความแตกต่างในการส่งข้อความ เพราะว่าเพียงแค่เจ็ดบิทแรกที่มีนัยสำคัญ เนื่องจากบิทศูนย์ถูกส่งก่อน และบิทเจ็ดไม่ถูกใช้ในการส่ง ASCII ปกติแล้วการขาดหายไปของมันจึงไม่มีความสำคัญอย่างไรก็ตาม อุปกรณ์ฝ่ายรับอาจพยายามแปลความหมายบิทที่เกินมาเป็นบิทพาริตี้ และ

รายงานข้อผิดพลาด ดังนั้นข้อผิดพลาดทางพาริตี้จึงไม่จำเป็นที่จะต้องหมายถึงข้อมูลถูกทำลายในการส่ง มันอาจบอกถึงความยาวเวิร์ดไม่ตรงกันก็ได้

ถ้าส่งเวิร์ดขนาดเจ็ดบิทโดยที่อุปกรณ์ฝ่ายรับต้องการเวิร์ดขนาดแปดบิทบิทพาริตี้อาจถูกนำไปรวมเป็นบิทที่แปด เนื่องจากบิทพาริตี้อาจเป็น 1 สำหรับตัวอักษรครึ่งหนึ่ง และเป็น 0 สำหรับอีกครึ่งหนึ่ง จึงพบไม่บ่อยครั้งว่าอุปกรณ์ฝ่ายรับจะแทรกอักขระเพิ่มเติม เช่น อักขระกราฟิกในจำนวนครึ่งหนึ่งของอักษรที่รับได้

2.3.19 บิทจบ

ไม่ควรจะมีปัญหาถ้าบิทจบสองบิทถูกส่งมา แม้มีเพียงบิทเดียวที่ต้องการบิทจบที่เกินมาเพียงแต่รวมเข้าในช่องว่างระหว่างตัวอักษร อย่างไรก็ตามการส่งหนึ่งบิทจบ เมื่อต้องการสองบิทอาจทำให้เกิดปัญหาขึ้นอยู่กับคุณลักษณะของอุปกรณ์ฝ่ายรับ เรื่องนี้ไม่เป็นปัญหากับอุปกรณ์โมเด็ม

2.3.20 ความผิดพลาดทางเฟรม

ความผิดพลาดทางเฟรมบ่งบอกความไม่ตรงกันของจำนวนบิท ซึ่งมักจะถูกรายงานเมื่อไม่ได้รับบิทจบตามที่คาดหวัง

2.4 การรับ - ส่งข้อมูลในระยะไกล

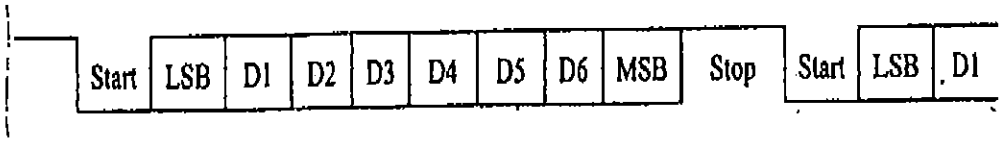
2.4.1 บทนำ

ปัจจุบันคอมพิวเตอร์กำลังได้รับความนิยมในการนำมาใช้งานด้านต่าง ๆ มากมายตามสำนักงานสถาบันการศึกษา หรือแม้แต่บ้านพักอาศัยก็ได้มีการนำคอมพิวเตอร์มาใช้ในงานที่เกี่ยวข้องกับการจัดการข้อมูล และงานเอกสารกันอย่างแพร่หลาย และมีแนวโน้มที่จะนั่งทำงานอยู่หน้าจอคอมพิวเตอร์นานขึ้น จนดูเหมือนว่าจะใช้คอมพิวเตอร์ในการทำงานทุกอย่างและด้วยเทคโนโลยีทางด้านฮาร์ดแวร์และโปรแกรมคอมพิวเตอร์ใหม่ ๆ ในปัจจุบัน ทำให้คอมพิวเตอร์มีความสามารถมากขึ้น ดังนั้นเราจึงได้นำคอมพิวเตอร์มาประยุกต์ใช้ร่วมกับงานในหลาย ๆ ด้าน รวมถึงด้านการสื่อสารข้อมูลระหว่างคอมพิวเตอร์กับคอมพิวเตอร์หรือ ระหว่างคอมพิวเตอร์กับอุปกรณ์ต่อพ่วงภายนอก แต่สิ่งหนึ่งที่เป็นส่วนประกอบสำคัญที่ทำให้การสื่อสารข้อมูลเป็นไปได้อย่างมีประสิทธิภาพนั้นก็คือ ระบบของการสื่อสารที่ทันสมัย ซึ่งนิยมใช้การสื่อสารแบบอนุกรม เพราะเสียค่าใช้จ่ายน้อยกว่าวิธีอื่น ๆ มาก

2.4.2 ความรู้ทั่วไปเกี่ยวกับการสื่อสารอนุกรม Asynchronous

การสื่อสารอนุกรมแบบ Asynchronous นั้นจัดเป็นการสื่อสารอนุกรมแบบใช้สายสัญญาณเส้นเดียว สามารถสื่อสารกันได้โดยการใช้ความเร็วของการรับส่งเป็นจุดอ้างอิงว่าจะรับส่งกันด้วยความเร็วกี่บิตใน 1 วินาที ส่งกันครั้งละกี่บิต โดยเริ่มนับจาก Bit Start เป็นจุดเริ่มต้นของการรับส่ง ซึ่งส่วนมากแล้วในการสื่อสารข้อมูลส่วนมากนั้นจะเลือกใช้วิธีแบบนี้ในการรับ-ส่ง โดยแบ่งสัญญาณออกเป็น 2 เส้น คือ รับและส่งอย่างละ 1 เส้น โดยในการส่งสัญญาณออกไปในสายส่งแต่

ละครั้งนั้นจะเริ่มต้นด้วยบิตข้อมูลเริ่มต้น ส่วนมากมีค่าเป็น 0 (start Bit) เพื่อใช้บ่งบอกให้ฝ่ายรับทราบว่ามีการเริ่มส่งข้อมูลแล้ว จากนั้นจึงตามด้วยข้อมูลอื่นบิต ๆ ต่อเนื่องกันไปจนครบซึ่งบิตสุดท้ายจะต้องเป็นบิตหยุด ซึ่งมีค่าเป็น 1 (stop Bit) เมื่อส่งข้อมูลครบ 1 ชุด แล้วถ้ามีข้อมูลชุดใหม่ที่ต้องการส่งข้อมูลใหม่ที่ต้องการส่งอีกก็จะเริ่มส่ง Start Bit ของข้อมูลชุดถัดไปอีก อย่างนี้เรื่อย ๆ .



รูปที่ 2.24 สัญญาณของการสื่อสารอนุกรมแบบ Asynchronous

ระบบของการสื่อสารแบบอนุกรม Asynchronous จัดเป็นระบบสื่อสารที่มีประสิทธิภาพดีอีกแบบหนึ่งที่มีการพัฒนาขีดความสามารถในการสื่อสารกันเรื่อยมาเป็นลำดับ แม้แต่ในปัจจุบันนี้ก็ยังเป็นที่นิยมใช้งานกันอยู่อย่างแพร่หลาย โดยเฉพาะอย่างยิ่งการสื่อสารกับเครื่องคอมพิวเตอร์ PC ที่ใช้งานกันอยู่ในปัจจุบันนี้เองก็ยังมีการบรรจุเอาวงจรการสื่อสารอนุกรม Asynchronous รวมไว้ในระบบด้วยทุกเครื่องเสมอ หรืออาจเรียกได้ว่า มันเป็นอุปกรณ์มาตรฐานของเครื่องคอมพิวเตอร์ที่ขาดไม่ได้เลยก็ว่าได้ ทั้งนี้แล้วสาเหตุคงเนื่องมาจากการสื่อสารแบบอนุกรม Asynchronous นี้มีขีดความสามารถสำหรับทำการรับ-ส่งข้อมูล กันได้ผลดีและเสียค่าใช้จ่ายน้อย จึงส่งผลให้การสื่อสารแบบนี้ได้รับการพัฒนาและปรับปรุงคุณภาพเรื่อยมาเป็นลำดับจนกลายเป็นมาตรฐานไปในที่สุด โดยระบบของการสื่อสารแบบนี้เราสามารถพบเห็นกันได้ทั่วไปแต่อาจมีชื่อเรียกที่แตกต่างกันไปบ้าง ทั้งนี้ก็ขึ้นอยู่กับวงจรภาคที่ใช้เปลี่ยนแปลงระดับสัญญาณ LOGIC TTL จากภาคส่งก่อนที่จะส่งสัญญาณนั้นเข้าไปในสายส่งสัญญาณ และ วงจรที่ใช้แปลงระดับของสัญญาณที่รับมาได้จากสายส่ง ก่อนจะส่งให้กับวงจรของภาครับอีกครั้งหนึ่ง ซึ่งนิยมเรียกกันว่าวงจร Line Driver นั่นเอง

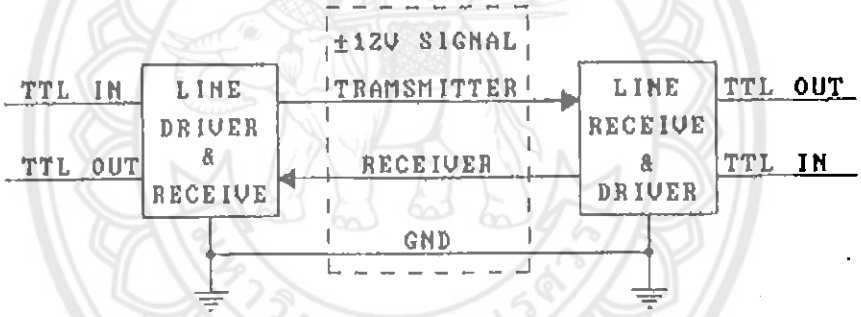
2.4.3 การรับ – ส่ง ข้อมูลด้วย RS 232

ระบบการรับ – ส่ง ข้อมูลแบบ RS 232 นี้ ถือกำเนิดขึ้นครั้งแรกในปี คศ. 1969 โดยสมาคมผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์แห่งสหรัฐอเมริกา ซึ่งข้อกำหนดของมาตรฐานระบบนี้จะครอบคลุมทั้งข้อกำหนดที่เป็นลักษณะทางกล เช่น ลักษณะของขั้วต่อสัญญาณที่ใช้การจัดเรียงขาสัญญาณต่าง ๆ รวมไปถึงข้อกำหนดที่เป็นคุณสมบัติทางไฟฟ้าของสัญญาณที่ใช้ในการรับ – ส่ง ข้อมูลระหว่างอุปกรณ์ต้นทางกับปลายทางเพื่อเป็นจุดอ้างอิงให้กับบริษัทผู้ผลิตต่าง ๆ ที่จะสร้างอุปกรณ์ที่ใช้ในการสื่อสารอนุกรม Asynchronous ได้ใช้เป็นมาตรฐานเดียวกันในการผลิตออกจำหน่าย

เมื่อนำสัญญาณในการรับ – ส่ง ของวงจรการสื่อสารอนุกรมแบบ Asynchronous นี้ ไปผ่านวงจร Line Driver เพื่อเปลี่ยนระดับของสัญญาณ จากระดับโลจิก TTL ของภาคส่งให้มีขนาด

สูงขึ้นเป็น $\pm$ 12 v เพื่อให้สามารถส่งสัญญาณไปในสายส่งสัญญาณให้ได้ระยะทางไกลขึ้น และในส่วนของภาครับเอง ก็ต้องทำการเปลี่ยนระดับของสัญญาณที่รับได้จากสายส่งสัญญาณที่เป็น $\pm$ 12 v ให้กลับมาเป็นระดับโลจิก TTL มาตรฐานเพื่อส่งสัญญาณให้กับวงจรภาครับอีกครั้งหนึ่ง โดยวงจร Line Driver แบบนี้จะเรียกกันโดยทั่วไปว่า RS 232 โดยคุณสมบัติของวงจร Line Driver แบบนี้จะสามารถ รับ – ส่ง ข้อมูลกันได้ผลดีในระยะทางประมาณ 50 ฟุต (15 เมตร) ทั้งนี้ก็เนื่องมาจากเมื่อสายส่งมีความยาวมาก ๆ แล้ว จะทำให้เกิดการสูญเสียของระดับแรงดันในสายส่งจนวงจรภาครับไม่สามารถตรวจสอบระดับของสัญญาณที่ต่ำเกินไปได้ จึงทำให้การรับส่งข้อมูลในระยะทางไกล ๆ ไม่ได้ผลเท่าที่ควรและเกิดความผิดพลาดมากขึ้น โดยการรับ – ส่ง ที่ใช้มาตรฐานสัญญาณแบบ RS 232 ที่พบเห็นกันได้ทั่วไปได้แก่ Serial Port ของเครื่องคอมพิวเตอร์ส่วนบุคคลที่นิยมเรียกกันว่า Com Port หรือ Port Mouse บางคนจึงนิยมเรียกกันว่า Com 1 หรือ Com 2 นั่นเอง

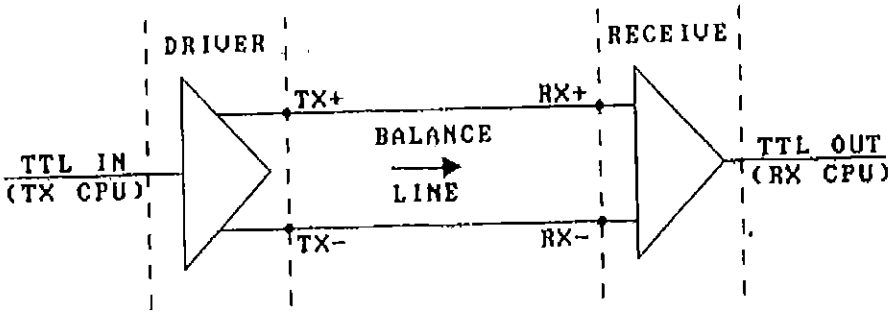
เนื่องจากการ รับ – ส่ง ข้อมูลแบบอนุกรมโดยใช้วงจร Line Driver แบบ RS 232 นั้นมักมีข้อจำกัดในเรื่องของระยะทางซึ่งไม่สามารถปรับปรุงให้ส่งได้ไกลขึ้นกว่าที่เป็นอยู่ได้และยังไม่สามารถเชื่อมต่อกันครั้งละหลาย ๆ อุปกรณ์ในเวลาเดียวกันได้



รูปที่ 2.25 แสดงวิธีการรับ – ส่ง ข้อมูลแบบ RS 232 ในอุดมคติ

2.4.4 มาตรฐาน RS – 422

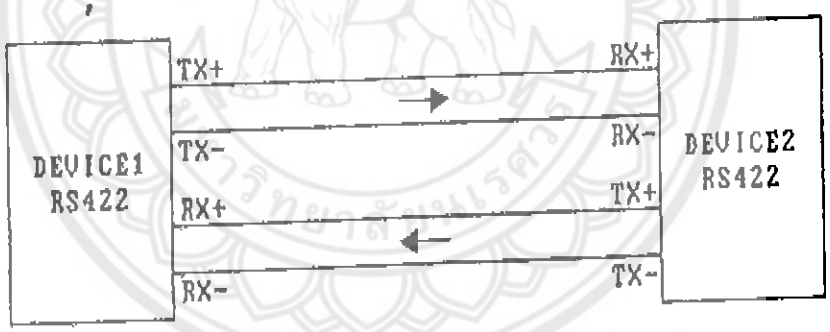
การสื่อสารแบบอนุกรม ด้วย RS – 422 จะใช้เทคนิคการส่งแบบ Balance Line ซึ่งวงจร Driver แบบนี้จะสามารถส่งสัญญาณ ที่มีค่าระหว่าง $\pm$ 2 volt ถึง $\pm$ 6 volt ได้ดี และในส่วนของ วงจรภาครับเองก็ยังสามารถตรวจจับสัญญาณที่มีขนาดต่ำถึง 200 mv. ได้ ถ้าใช้อุปกรณ์ได้ตรงตามมาตรฐานที่กำหนดไว้ สามารถส่งได้ไกลถึง 1000 ฟุต แต่ถ้าความเร็วที่ใช้ในการรับส่งมีค่า ต่ำกว่า 10 Mbps. จะสามารถส่งได้ไกลถึง 4000 ฟุต (1,200 เมตร) แต่อย่างไรก็ตามต้องพิจารณากับองค์ประกอบอื่น ๆ ด้วย เช่น คุณภาพสายสัญญาณที่ใช้ในการรับส่ง, คุณภาพของ ขั้วต่อสัญญาณ, และระดับของสัญญาณรบกวนที่สายสัญญาณเดินผ่าน



รูปที่ 2.26 แสดงวิธีการรับ – ส่ง ข้อมูลแบบ RS 422 ในอุดมคติ

- *การเชื่อมต่อ RS-422 แบบ Full duplex*

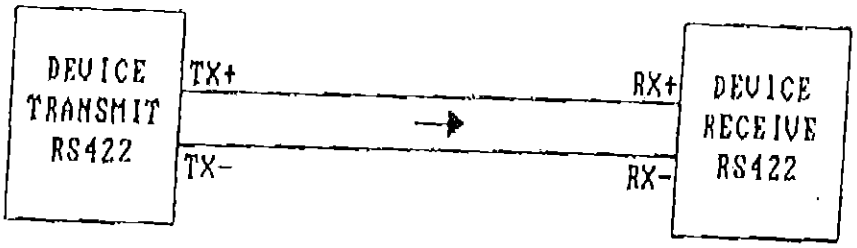
เป็นการรับส่งข้อมูลแบบ 2 ทิศทาง ได้พร้อมกันซึ่งสามารถรับส่งได้พร้อมกันตลอดเวลาตามต้องการ ในการเชื่อมต่อแบบนี้ จะต้องใช้วงจร Line Diver ถึง 2 ชุด คือรับ 1 ชุด และส่ง อีก 1 ชุด โดยจะมีสายสัญญาณชุดละ 1 คู่ (2 เส้น) การต่อแบบ Full Duplex นี้ มีลักษณะคล้ายกับการพูดโทรศัพท์ โดยไม่จำเป็นให้อีกฝ่ายหนึ่งพูดจบก็ได้ แต่วิธีการแบบนี้จะใช้กับอุปกรณ์แบบ Point to Point



รูปที่ 2.27 แสดง ลักษณะการต่อสาย RS 422 แบบ Full Duplex

- *การเชื่อมต่อ RS422แบบ simplex*

เป็นการรับข้อมูลแบบทิศทางเดียว โดยกำหนดทิศทางไว้คงที่ ไม่สามารถเปลี่ยนได้ด้วยโปรแกรม โดยทิศทางสามารถกำหนดเป็นแบบรับเข้า หรือ ส่งออก อย่างเดียว



รูปที่ 2.28 แสดงลักษณะการต่อสาย RS 422 แบบ Simplex

2.4.5 การรับส่ง ข้อมูลด้วย RS – 485

การรับส่งข้อมูลแบบ RS – 485 จะมีลักษณะคล้ายกับ RS – 422 ซึ่ง ใช้การส่งแบบสมดุล (Balanced Transmission) คือ เป็นการส่งข้อมูลที่ใช้ผลต่างของระดับแรงดันของสัญญาณ 2 เส้น สามารถส่งข้อมูลด้านความเร็วสูง (สูงสุดได้ถึง 2.5 เมกะบิตต่อวินาที และระยะทางได้ไกลมากขึ้น จาก RS – 422) คือ 1000 ม.

การส่งข้อมูลด้วย RS – 485 มีอยู่ 2 แบบ คือ ใช้สายสัญญาณ 4 เส้น และใช้สายสัญญาณ 2 เส้น

1. *การใช้สัญญาณแบบ 4 เส้น*

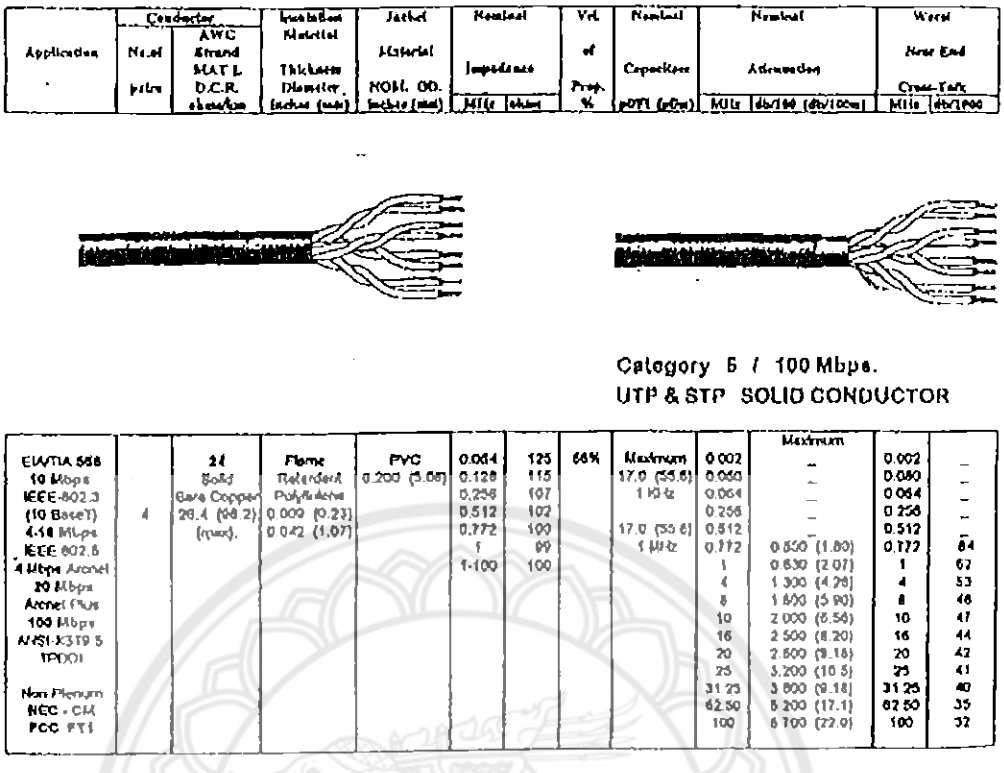
สัญญาณข้อมูลที่รับและส่งจะใช้สัญญาณคนละคู่กัน คือ ส่ง 2 เส้น และรับ 2 เส้น ส่งสัญญาณสื่อสารแบบ Full Duplex ได้ คือสามารถส่งและรับข้อมูลได้ในเวลาเดียวกัน

2. *แบบใช้สายสัญญาณ 2 เส้น*

สัญญาณที่รับและส่งจะใช้สายสัญญาณร่วมกัน โดยจะใช้การสื่อสารแบบ Half Duplex คือ ไม่สามารถรับและส่งข้อมูลจะได้ในเวลาเดียวกัน และในเครือข่ายหนึ่งจะมีผู้ส่ง ได้เพียงผู้เดียวเท่านั้น จะส่งพร้อมกันไม่ได้

2.4.6 สายสัญญาณที่ใช้กับ RS – 422 / RS – 485

สำหรับสายที่จะใช้ในระบบ Line Driver แบบ RS – 422 / RS – 485 นั้นจะต้องเป็นสายที่ออกแบบมาเพื่อใช้กับงานด้านการสื่อสารโดยเฉพาะ เช่น สายสัญญาณแบบ Unshiled Twis Pair (UTP) ซึ่งเป็นสายคู่ตีเกลียวแบบไม่มี Shiled ซึ่งเหมาะสำหรับใช้งานภายในอาคาร ซึ่งไม่มีแหล่งกำเนิดสัญญาณรบกวนแผ่กระจายออกมาในรัศมีที่พาดผ่านไปมากนัก แต่ถ้าเส้นทางที่สายสัญญาณพาดผ่านไปนั้น มีระดับของสัญญาณรบกวนมากก็ต้องใช้สายแบบ Shiled Twis Pair (STP) ซึ่งเป็นสายคู่ตีเกลียวแบบมี Shiled นั่นเอง โดยในการนำไปต่อใช้งานนั้นต้องถูกคู่ด้วย



Category 5 / 100 Mbps.
UTP & STP SOLID CONDUCTOR

Application	Conductor: No. of pairs	Conductor: AWG Strand MAT'L D.C.R. ohms/km	Insulation Material Thickness Diameter inches (mm)	Jacket Material Nom. OD inches (mm)	Nominal Impedance MHz	Nominal Impedance ohms	Vel. of Prop. %	Nominal Capacitance pF/ft (pF/m)	Nominal Attenuation MHz	Nominal Attenuation dB/100 ft (dB/100m)	Worst Near End Cross-Talk MHz	Worst Near End Cross-Talk dB/1000
EIA/TIA 568		24	Flame	PVC	0.064	125	66%	Maximum	0.002	–	0.002	–
10 Mbps		Solid	Retardant	0.200 (5.08)	0.128	115		17.0 (55.8)	0.050	–	0.050	–
IEEE-802.3		Bare Copper	Polyolefine		0.256	107		1 kHz	0.064	–	0.064	–
(10 BaseT)	4	28.4 (93.2)	0.020 (0.23)		0.512	102			0.256	–	0.256	–
4-16 Mbps		(nom).	0.042 (1.07)		0.772	100		17.0 (55.8)	0.512	–	0.512	–
IEEE 802.5					1	99		1 kHz	0.772	0.650 (1.80)	0.772	64
4 Mbps Arcnet					1-100	100			1	0.650 (2.07)	1	62
20 Mbps									4	1.300 (4.26)	4	53
Arcnet Plus									8	1.800 (5.90)	8	46
100 Mbps									10	2.000 (6.56)	10	47
ANSI-X3T9.5									16	2.500 (8.20)	16	44
TP(DDI									20	2.800 (9.18)	20	42
									25	3.200 (10.5)	25	41
Non-Plenum									31.25	3.800 (9.18)	31.25	40
NEC - CM									62.50	5.200 (17.1)	62.50	35
FCC PT1									100	6.700 (22.0)	100	32

รูปที่ 2.29 แสดงตัวอย่าง Specification ของสายสัญญาณที่ใช้กับระบบ RS 422/485

2.5 มาตรฐานการรับส่งข้อมูลที่จะนำมาใช้

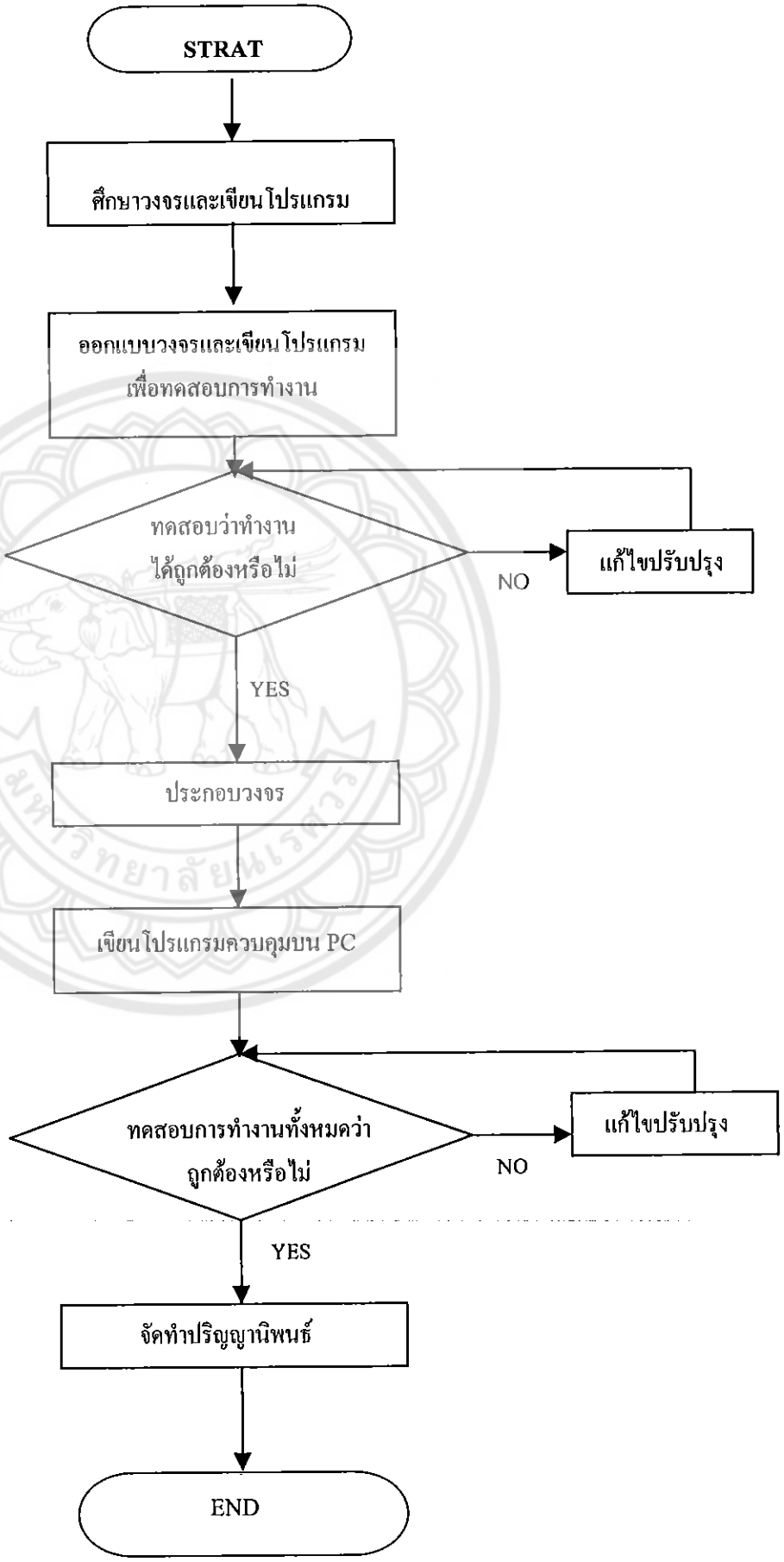
จากการศึกษาคุณสมบัติและรายละเอียดของแต่ละมาตรฐานการรับส่งข้อมูล แล้วแต่ละระบบก็จะมีข้อดี และข้อเสียแตกต่างกันไป ซึ่งก็จะขึ้นอยู่กับจุดประสงค์ของงานที่จะนำไปใช้ ในส่วนนี้มีจุดประสงค์ที่จะสื่อสารข้อมูล ในระยะทางที่ไกลออกไปและสามารถที่จะทำการเชื่อมต่ออุปกรณ์ในการทำงานหลายจุด เป็นแบบ Network เมื่อพิจารณามาตรฐาน RS–232 แล้ว ไม่สามารถจะนำมาประยุกต์ใช้ในส่วนนี้ได้เนื่องจากความจำกัดในระยะทาง ซึ่งเมื่อไกลออกไปสัญญาณจะถูกรบกวนและอ่อนกำลังลง มาตรฐาน RS–422 สามารถนำมาใช้ได้ เนื่องจากสามารถรับส่งข้อมูลออกไปในระยะไกลขึ้นมากกว่า RS–232 แต่ถ้าหากต้องการสื่อสารแบบ Full Duplex แล้ว RS–422 จะสามารถทำได้เพียงแค่ 2 จุด (แบบ Point to point) ไม่สามารถทำได้หลายจุด ยกเว้นจะสื่อสารแบบ Simplex RS–422 จะสามารถต่อออกไปได้ถึง 32 จุด แต่มีข้อจำกัดคือ สามารถส่งได้อย่างเดียว ฉะนั้นจึงมาพิจารณาที่มาตรฐาน RS–485 ซึ่งสามารถรับส่งสัญญาณได้ไกลกว่า RS–422 และสามารถสื่อสารได้ทั้งแบบ Half Duplex และแบบ Full Duplex

บทที่ 3
ขั้นตอนและวิธีดำเนินงาน

3.1 ขั้นตอนการดำเนินงาน

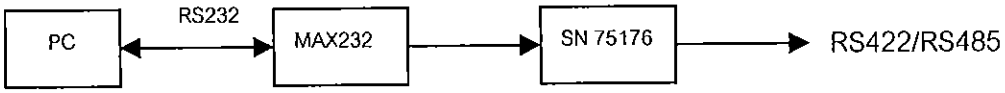
1. ศึกษาข้อมูลเกี่ยวกับโครงสร้างและการทำงานของไมโครคอนโทรลเลอร์ตระกูล MCS 51 การอินเตอร์เฟสแบบ RS232, RS422 การส่งข้อมูลแบบอนุกรมของเครื่องไมโครคอมพิวเตอร์
2. ออกแบบและประกอบวงจรบนบอร์ดควบคุม และวงจรแปลงการรับ–ส่ง ข้อมูลจาก RS232 ไปเป็น RS422
3. เขียนโปรแกรมทดสอบการทำงานของบอร์ดควบคุม และทดลองวงจรแปลงการรับ–ส่งข้อมูลจาก RS232 ไปเป็น RS422
4. เขียนโปรแกรมบนเครื่องไมโครคอมพิวเตอร์เพื่อติดต่อสั่งงานกับบอร์ดควบคุม
5. ออกแบบและสร้างชุดจำลองการทำงานทั้งหมด
6. ต่อวงจรควบคุมเข้ากับชุดจำลองการทำงานและทำการทำงานของโครงงานข้อมูลเกี่ยวกับโครงสร้างและการทำงานของไมโครคอนโทรลเลอร์ตระกูล MCS51 การอินเตอร์เฟสแบบ RS232, RS422 การส่งข้อมูลแบบอนุกรมของเครื่องไมโครคอมพิวเตอร์ ได้นำเสนอไว้ในบทที่ 2 แล้ว

แผนภูมิขั้นตอนการทำโครงงานปริญญานิพนธ์



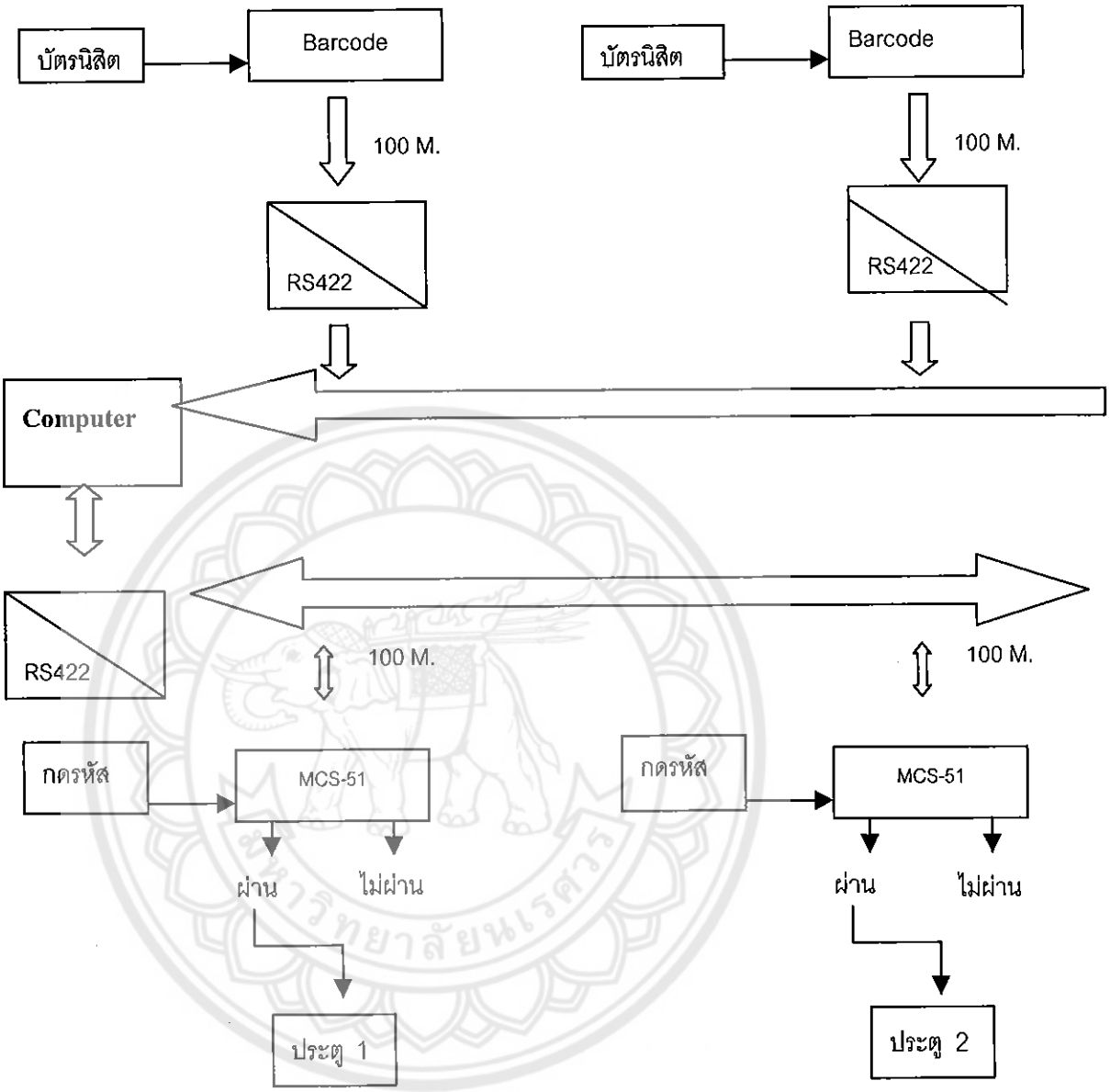
3.2 วงจรแปลงการรับ – ส่ง ข้อมูลจากมาตรฐาน RS 232 ไปเป็น RS 422/485

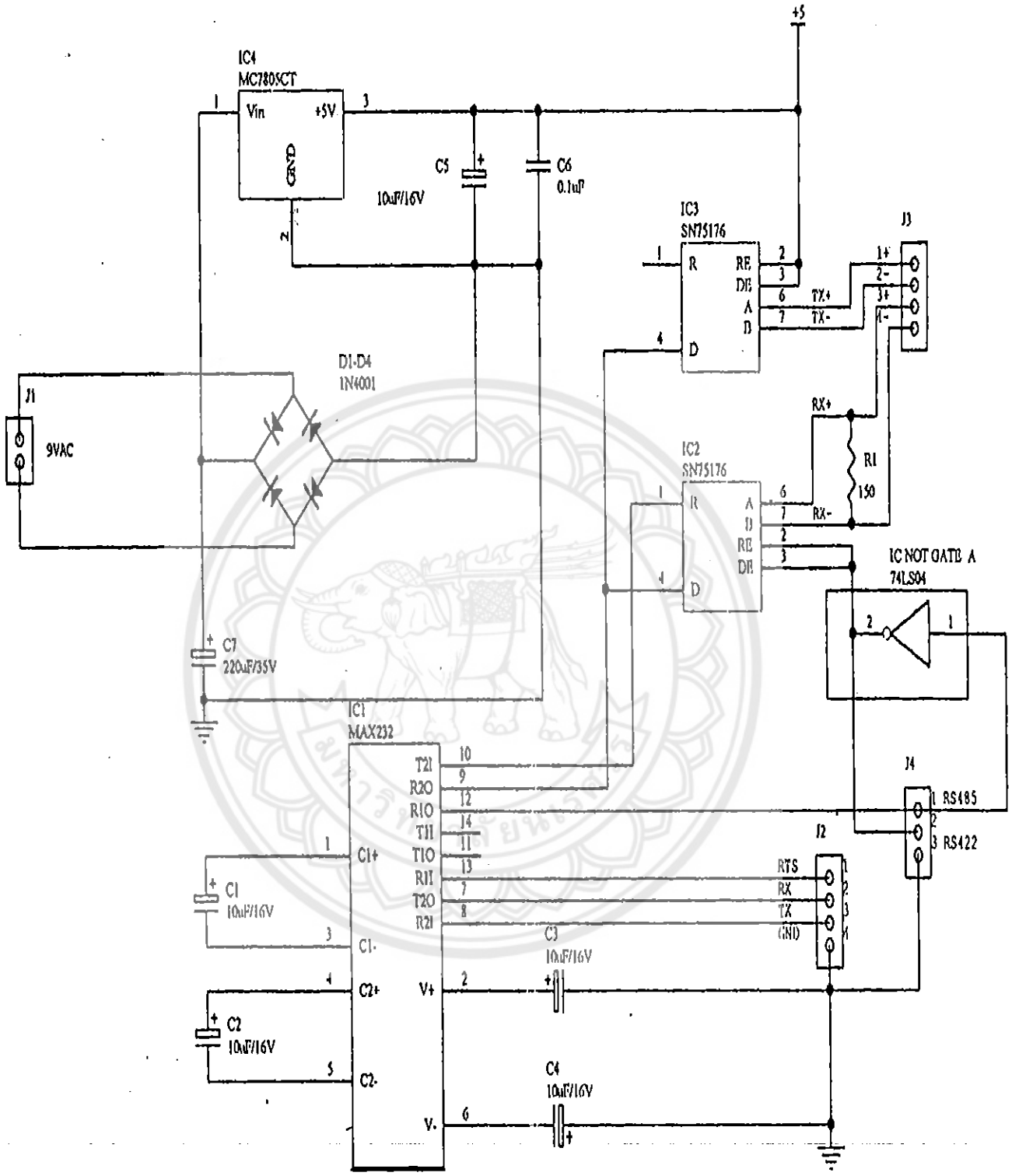
จากวงจรสามารถเขียนบล็อกไดอะแกรมแสดงลำดับของการแปลงสัญญาณในมาตรฐาน RS 232 เป็น RS 422/485



ไอซี MAX232 จะทำหน้าที่แปลงแรงดันในมาตรฐาน RS232 จากคอมพิวเตอร์ให้อยู่ในรูปแบบสัญญาณทีทีแอล (TTL) และยังทำการแปลงสัญญาณแบบ ทีทีแอล กลับไปเป็นสัญญาณในมาตรฐาน 232 เพื่อส่งกลับไปยังคอมพิวเตอร์ ส่วนไอซี 75176 อีก 2 ตัวทำหน้าที่แปลงสัญญาณแบบ ทีทีแอลที่ออกจาก MAX232 ให้เป็นสัญญาณในมาตรฐาน RS422 ไอซี 75176 นี้ จะถูกกำหนดให้ทำหน้าที่เป็นทั้งภาครับและภาคส่งด้วยสัญญาณควบคุมที่ขา 2 และ ขา 3 ของไอซี โดยถ้าสัญญาณควบคุมเป็นลอจิก "1" ไอซี 75176 นี้จะทำหน้าที่เป็นภาคส่ง ถ้าสัญญาณควบคุมเป็นลอจิก 0 ไอซี 75176 จะทำหน้าที่เป็นภาครับ การใช้งานแบบ RS422 จะต้องใช้ ไอซี 75176 ทั้ง 2 ตัว ทำหน้าที่เป็นภาครับและภาคส่งนอกจากการแปลงสัญญาณรับส่งข้อมูลจาก RS232 ไปเป็น RS422 แล้ว วงจรนี้ยังสามารถแปลงการรับส่งข้อมูลจาก RS232 ไปเป็น RS485 ได้ด้วย โดยใช้ ไอซี 75176 เพียงตัวเดียว ซึ่งจากวงจรใช้ IC 2 โดยสัญญาณควบคุมว่าจะให้ IC2 ทำหน้าที่เป็นภาครับหรือภาคส่งนั้น ใช้สัญญาณ RTS จากพอร์ทอนุกรม RS232 ของคอมพิวเตอร์ โดยปรกติสัญญาณ RTS ของคอมพิวเตอร์ จะมีลอจิก "1" และจะเปลี่ยนเป็นลอจิก 0 เมื่อคอมพิวเตอร์ต้องการที่จะส่งข้อมูลออกทางพอร์ทอนุกรม RS232 ซึ่งจะเห็นได้ว่า สัญญาณ RTS มีระดับลอจิกตรงกันข้ามกับการใช้งานควบคุม IC75176 อย่างที่ควรจะเป็น กล่าวคือ คอมพิวเตอร์ต้องการจะส่งข้อมูล ตัว IC75176 ของชุดแปลงการรับส่งข้อมูล จะต้องทำหน้าที่เป็นภาคส่งด้วยเพื่อให้สอดคล้องกัน และสัญญาณ RTS ควรจะเป็นลอจิก "1" เพื่อควบคุมให้ IC75176 ทำหน้าที่เป็นภาคส่ง วิธีที่จะทำให้สัญญาณ RTS ถูกต้องตามเงื่อนไขการควบคุม ทำได้โดยการนำสัญญาณ RTS มาผ่านอินเวอร์เตอร์หรือน็อทเกท แล้วนำเอาท์พุตของน็อทเกทไปเข้าที่ขา 2 และ ขา 3 ของไอซี 75176 ก็จะได้สภาวะการทำงานที่ถูกต้องภาคจ่ายไฟของวงจรได้จากการนำกระแสไฟสลับ 9 โวลท์ มาผ่านทางวงจรบริดจ์เพื่อแปลงให้เป็นแรงดันไฟฟ้ากระแสตรงประมาณ 12 โวลท์ แล้วกรองแรงดันให้เรียบด้วย C7 แล้วจึงรักษาระดับแรงดันให้คงที่ ที่ 5 โวลท์ ด้วย IC7805 จ่ายให้แก่วงจร

3.3 Block Diagram แสดงระบบควบคุมการเข้า – ออกอาคารและห้องทำงาน





รูปที่ 3.1 วงจรของชุดรับ – ส่งข้อมูล RS232 เป็น RS422/485

3.4 Visual Basic Programming Interface Hardware

การใช้ VB เขียนโปรแกรมติดต่อ I/O ผ่านทาง Port ของเครื่องคอมพิวเตอร์ ไม่ว่าจะเป็นทาง Serial Port(RS-232) หรือที่รู้จักในชื่อ Com1,Com2 และ Parallel Port หรือ Printer Port นั้นเอง หรืออาจใช้ Card I/O 8255 ซึ่งเป็นการขยายPort I/O ของ Parallel ก็สามารถทำการติดต่อกับ Hardware ภายนอกผ่านPort ได้ อีกทั้งสามารถติดต่อผ่านระบบ Network โดยผ่านช่องทางการติดต่ออย่าง TCP/IP

3.4.1 Serial Port(RS-232)

สามารถติดต่อกับอุปกรณ์ต่างๆที่มีการติดต่อกับอุปกรณ์ภายนอกผ่านทาง RS-232 เช่น เครื่องชั่งน้ำรวมถึงโหลดเซล(เป็นเซ็นเซอร์ชนิดหนึ่งใช้สำหรับวัดน้ำหนักซึ่งที่ชุดแสดงผลภายในเป็นชุดไมโครคอนโทรลเลอร์ จะมีสัญญาณรับส่งทาง RS-232),เครื่องวัดงานทางด้านไฟฟ้า,ไมโครคอนโทรลเลอร์,ควบคุมอุปกรณ์ไฟฟ้า,โอนถ่ายข้อมูลในฮาร์ดดิสระหว่างเครื่องคอมพิวเตอร์ด้วยกัน,ควบคุมแสตปปิ้งมอเตอร์ เป็นต้น ข้อดีของการติดต่อข้อมูลกันผ่านทาง RS-232 ก็คือสามารถใช้ได้ในระยะทางไกลๆระหว่างอุปกรณ์ ที่ติดต่อกัน

เนื่องจากที่ Microsoft Visaul Basic 5,6 จะมีตัวคอนโทรลชื่อ MS Comm ที่ใช้ติดต่อกับ Serial Port(RS-232) ให้ไว้อยู่แล้วไม่จำเป็นต้องเขียนโค๊ดให้ยุ่งยากทำให้การพัฒนาโปรแกรมในด้านนี้ได้เร็วและเป็นมารตฐานในทิศฐานการเขียนโปรแกรมเดียวกันของทุกโปรแกรม

3.4.2 Parallel Port

สามารถทำการประยุกต์ใช้งานได้ดีเพราะสามารถทำงานได้ทีละ 8 บิตในการติดต่อข้อมูลกับอุปกรณ์ภายนอกให้ระดับแรงดันที่ใช้กับอุปกรณ์ TTL ที่สัญญาณลอจิกเป็น"1"เท่ากับ +5 โวลท์ และลอจิกเป็น"0"เท่ากับ 0 โวลท์ในเมื่อมีข้อดีก็ต้องมีข้อเสีย ก็คือไม่สามารถทำงานในระยะทางที่ไกลๆระหว่างอุปกรณ์ที่ติดต่อกันจะเกิดความผิดผลาดของข้อมูลขึ้นง่ายเนื่องจากระดับแรงดันไม่สม่ำเสมออีกทั้งสิ้นเปลืองค่าใช้จ่ายในการต้องใช้สายสัญญาณหลายเส้น โครงงานที่นำมาใช้เช่น ควบคุมแสตปปิ้งมอเตอร์ 3 แกน XYZ,อิเล็กโตลนิวเมติกควบคุมโซลินอยวาล์ว,ควบคุมอุปกรณ์ภายในบ้านหรือสถานที่พิเศษ เป็นต้น

Microsoft Visual Basic 4,5,6 จะไม่มีฟังก์ชั่นสำหรับติดต่อพอร์ตโดยตรง เมื่อกับตอนยังอยู่บน DOS ดังเช่น BASIC,QBASIC ที่ใช้ฟังก์เช่น OUT เป็นต้น จากใน MSDN ของ Visual Studio สามารถติดต่อทางพอร์ตได้โดยใช้ API(Application Programming Interface) โดยจะต้องมีไฟล์ DLL ไว้สำหรับเรียกฟังก์ชั่นเพื่อติดต่อพอร์ตโดยจะเขียนด้วยภาษา VC++,C++,Pascal เป็นต้น ซึ่งก็จะมีวิธีสร้างไฟล์ DLL อยู่ในนี้แล้ว เขียนด้วย C++

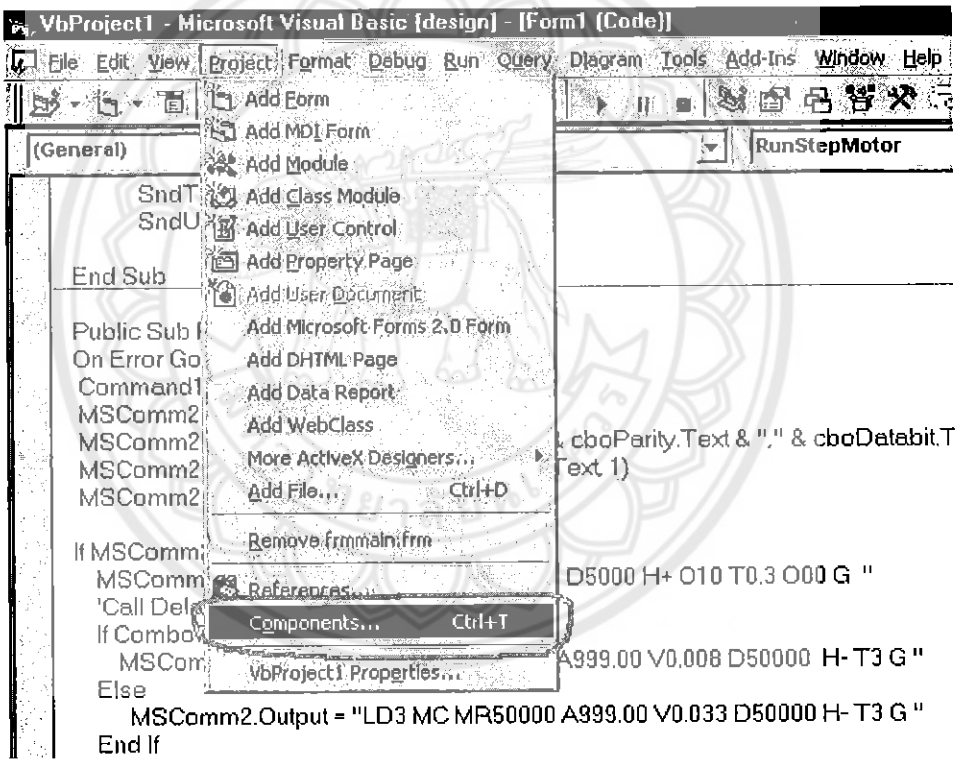
3.4.3 สรุป

การใช้ Visaul Basic บนวินโดวส์ไม่จำเป็นต้องเขียนโค๊ดที่คอนข้างยาว สามารถเขียนโปรแกรมได้ง่ายขึ้นซึ่งจะมีตัวคอนโทรลชื่อ MSComm สามารถติดต่อผ่าน RS-232 ได้ และ Winsock ที่สามารถติดต่อผ่าน TCP/IP

3.4.4 Visual Basic เขียนโปรแกรมติดต่อ I/O Serial Port

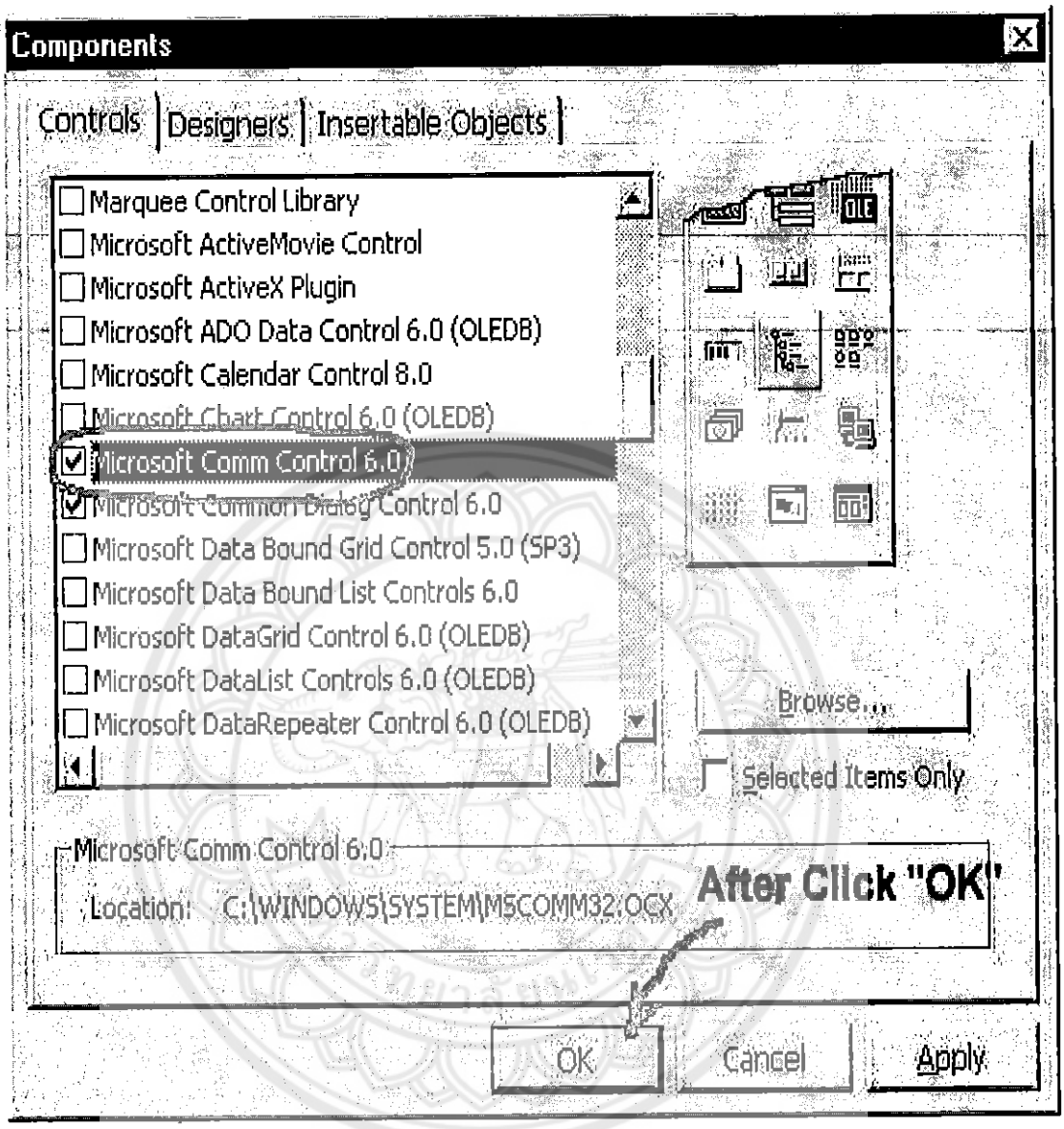
สามารถทำได้โดยใช้ VB Control ที่ชื่อว่า MSComm โดยที่ต้อง กำหนด Custom Control เข้าไปที่ เมนู Project--->Companents แล้วเลือกที่ช่อง MSComm ก็จะปรากฏ เป็นรูปไอคอนโทรศัพท์สีเหลือง ให้คลิกที่ไอคอนลากนำมาไว้บน Form ใน Project ของโปรแกรมเรา โดยสามารถทำตามวิธีที่กล่าวมา ได้ดังรูปต่อไปนี้

1. ขั้นที่ตอนแรก เลือกที่เมนูบาร์ด้านบนของโปรแกรม Visual Basic ดังรูปที่ 3.2



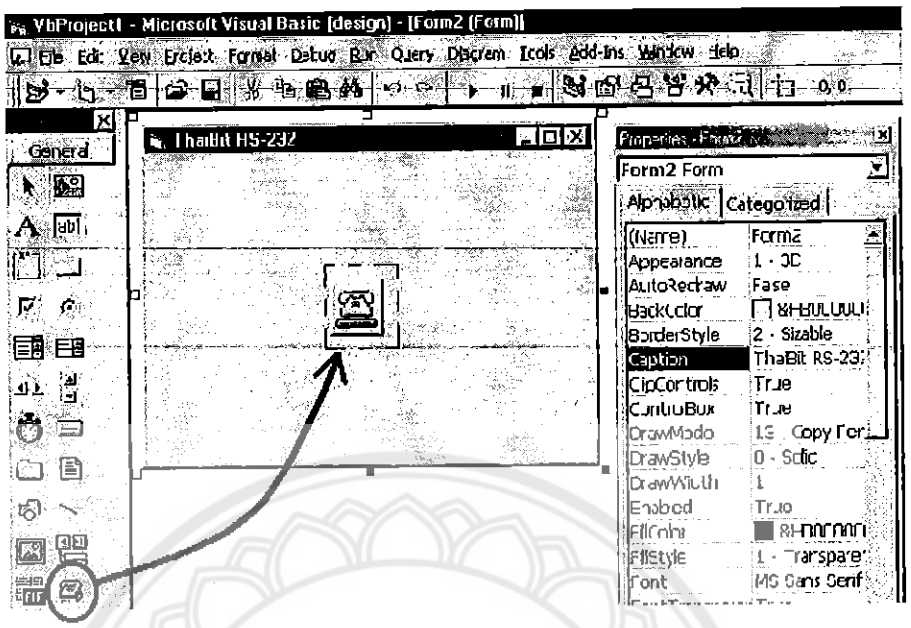
รูปที่ 3.2 การเลือกเมนูบาร์ของโปรแกรม visual basic

2. ขั้นที่สอง เลือกชื่อ Control ชื่อ Microsoft Comm Control 6 ดังรูป



รูปที่ 3.3 การเลือกชื่อจาก control

3. ขั้นที่สาม ลากControlชื่อMicrosoft Comm จากToolBox มาไว้บนFormดังรูปที่ 3.4



รูปที่ 3.4 ขั้นตอนการเลือก Microsoft comm จาก toolbox มาไว้บนForm

3.4.5 การเขียนโปรแกรมติดต่อกับ Serial Port สามารถทำได้ 2 วิธี คือ

1. การติดต่อแบบอินเตอร์รัพต์

ขบวนการอินเตอร์รัพท์ อุปกรณรอบข้างเกือบทุกชิ้นจะต้องปฏิบัติงานอยู่เพื่อส่งสัญญาณไปให้แก่ซีพียูเสมอ ถ้าอุปกรณ์นั้นพร้อมที่จะรับส่ง ที่เคยเจอจากการทำโครงงานอุปกรณ์ จะส่งเป็นรหัสแอสกี้ เราจะเขียนโปรแกรมอินเตอร์รัพต์ โดยเมื่อที่ข้อมูลเข้ามาก็จะทำให้มี CommEvent กับ OnComm Event

2.การติดต่อแบบโพลลิ่ง

ในระบบพีซี การโพลมีบ้างที่ใช้การส่งผ่านข้อมูลระหว่าง Terminal กับ CPU กรณีข้อมูลเป็นประเภทไบท์ที่ส่งจากคีย์บอร์ด โดยวิธีการนี้จะตรวจสอบ คีย์บอร์ดว่ามีข้อมูลส่งมาหรือเปล่า โดยจะตรวจสอบตลอดเวลา การทำงานกับข้อมูลที่รับเข้ามาจะตรวจสอบด้วยความเร็วที่สูงกว่าอัตราความเร็วข้อมูลที่ส่งเข้ามาทาง คีย์บอร์ด การที่ CPU ส่งสัญญาณออกไปตรวจสอบพบว่ามีข้อมูลที่ต้องส่งเข้ามา เรียกว่า"Wet Poll" ซึ่งจะเสียช่วงเวลา 90 เปอร์เซ็นต์ คาบเวลาที่เสียไปนั้นเราเลี่ยงไปใช้เทคนิค การโพลแบบ "Round Robin" แทน แต่ในVBจะใช้การตรวจสอบข้อมูลที่มาจาก Serial Port ตลอด โดยจะใช้ Control Timer เข้ามาช่วยในการเขียนโปรแกรมซึ่งสามารถตรวจสอบได้ถึงระดับ 1 มิลลิวินาที หรือจะใช้ Do....Loop ก็ได้

ในตัวคอนโทรล MSComm มี Event ที่ใช้เพียง Event เดียวเท่านั้นเอง ก็คือ OnComm Event ซึ่งจะใช้ในการติดต่อแบบอินเตอร์รัพต์ การเขียนโปรแกรมติดต่อ Serial Port แบบธรรมดาจะใช้ comEvent เพียง comEvReceive,comEvSend ถ้าเป็นการติดต่อสื่อสารแบบ โมเด็มจะใช้หลายตัวในการตรวจสอบสัญญาณ ผมไม่อยากแจ้งรายละเอียดเยอะเพราะมีใน Help Visual Basic อยู่แล้ว

3.5 องค์ประกอบในการใช้ MSComm

3.5.1 การตั้งค่าติดต่อกับพอร์ต

- **ComPort** คือ เราต้องกำหนดหมายเลข Port ที่ใช้ต่อRS-232 (Com1,Com2)รายละเอียดดูในเมนูด้านซ้าย Serial Port Detail
- **Setting** คือ เราต้องกำหนดอัตรา Baud,Parity,Data(จำนวนบิต),Stop ตัวอย่าง 9600,n,8,1 เป็นต้น
- **HandShaking** คือ เราจะกำหนดได้ 4 แบบ 1.comNone 2.comXonXoff 3. comRTS 4.comTRSXonXoff

3.5.2 การกำหนด Buffer ในการรับข้อมูลเข้ามา

- **OutBuffersize** คือ การกำหนด Buffer ในการส่งข้อมูลออกไป
- **Rthreshold** คือ การที่เรากำหนดการเกิด Event-driven ในการรับข้อมูลเข้ามา
- **Sthreshold** คือ การที่เรากำหนดการเกิด Event-driven ในการรับข้อมูลออกไป
- **Inputlen** คือ จำนวนของข้อมูลที่ไปอ่านใน Buffer รับข้อมูล
- **EOFEnable** คือ การที่บอกว่าสิ้นสุดของไฟล์(EOF) End of File

3.5.3 ด้านฮาร์ดแวร์

- **ParityReplace** คือ ค่าของคาแลกเตอร์ที่จะแทนในเมื่อเกิด Parity Error
- **NullDiscard** คือ การกำหนดให้รับหรือไม่รับ NULL CHARACTER
- **RTSEnable**คือ ทำให้มีสัญญาณ RTS (Request To Send)
- **DTSEnable**คือ ทำให้มีสัญญาณ DTR(Data Terminal Ready)

3.6 การกำหนดคุณสมบัติของ MSComm Control ให้สามารถติดต่อกับพอร์ตได้

3.6.1 Property ชื่อ CommPort คือ เลือกคอมพอร์ตที่เราจะต่อใช้งาน การเขียนโค๊ด

ตัวอย่าง MSComm1.CommPort=1

ในที่นี้เลือกจะใช้ Com1อยู่ที่ด้านหลังเครื่องคอมฯ

3.6.2 Property ชื่อ Settingsคือ การตั้งค่าของการรับส่งข้อมูล ซึ่งจะต้องรู้ด้วยว่าอัตราบอด ของอุปกรณ์ที่จะติดต่อด้วยเป็นเท่าไร โดยมีรายละเอียดการใส่ต่างๆค่าดังนี้

MSComm1.Settings="Baud(อัตราการรับส่งข้อมูล),Parity(ถ้าไม่ใช้ใส่ N,จำนวนบิตข้อมูล,บิตสต๊อบ"

การเขียนโค๊ด

ตัวอย่าง MSComm1.Settings="1200,N,8,1"

3.6.3Property ชื่อ InputLenคือ กำหนดขนาดขณะที่มีข้อมูลเข้ามาให้ไปอ่านข้อมูลทั้งหมดที่อยู่ในบัฟเฟอร์ มาดูการเขียนโค๊ดกัน

ตัวอย่าง MSComm1.InputLen=1

3.6.4. Property ชื่อ PortOpen คือ จะเปิดให้พอร์ตใช้งานหรือไม่ ถ้าเปิด =True ถ้าปิด =False มาดูการเขียนโค๊ดกัน

ตัวอย่าง MSComm1.PortOpen=True

3.6.5. Property ชื่อ Rthresholdคือ ทำให้เกิดการกระตุ้นด้วย Event-driven เมื่อมีข้อมูลในบัฟเฟอร์รับข้อมูล(Comport)มันทำให้เกิดCommEvent ใน OnComm Event มาดูการเขียนโค๊ดกัน

ตัวอย่าง MSComm1.Rthreshold =1

จากรายละเอียดที่กล่าวมา เราจะมาเขียนใน โพรซีเดอร์ VB ซึ่งจะไว้ที่ Sub Form_Load() หรือจะสร้าง Sub ขึ้นใหม่ในกรณีที่จะเรียกใช้ภายหลัง

```
Private Sub Form_Load()
MSComm1.Settings="1200,N,8,1"
MSComm1.CommPort=1
MSComm1.InputLen=1
MSComm1.PortOpen=True
MSComm1.Rthreshold =1
End Sub
```

3.7 วิธีของการรับส่งข้อมูลจาก Serial Port

จากวิธีเขียนโค๊ดด้านบนเป็นการกำหนดค่าเริ่มต้นให้กับคอมพอร์ตและเปิดใช้การรับและส่งของพอร์ต RS-232 ดังนั้นก็สามารถจะรับและส่งข้อมูลทางพอร์ตได้ โดยใช้ Property ดังนี้

Output =ซึ่งจะเป็นการส่งข้อมูลไปที่พอร์ต

Input =เป็นส่วนของการรับข้อมูลจากพอร์ต แต่ในส่วนนี้จะต้องนำคำสั่งไปเขียนที่ Event Property OnComm จะอยู่ใน Sub MSComm_OnComm ซึ่ง จะอ่านข้อมูลเข้ามาจากทางพอร์ต RS232 นั้นเอง

ตัวอย่างเช่น ถ้าต้องการที่จะพิมพ์ข้อมูลส่งออกพร้อมๆกับขณะที่เราพิมพ์ไปด้วยคุณก็เพียงไปเขียนโค๊ดไว้ที่ Event KeyPress ของ Control TextBox ที่เราจะให้เป็นตัวส่งข้อมูลโดยเขียนดังนี้

```
Sub txtRXTX_KeyPress(KeyAscii As Integer)
        MSComm1.Output=Chr$(KeyAscii)
End Sub
```

ส่วนการใช้ Prooerty Input ต้องนำมาไว้ที่ Event OnComm ดังนี้

```
Private Sub MSComm1_OnComm()
Dim StrData As Variant 'กำหนดชนิดตัวแปรเพราะต้องการให้เป็นอะไรก็ได้
    Str=MSComm1.Input
    Text1.Text=StrData
End Sub
```

จากผลการทดลองครั้งแรกที่ทำในส่วนของการรับข้อมูล Property Input ซึ่งถ้าเขียนโค๊ดดังข้างต้นนั้นจะยังใช้ไม่ในการที่จะอ่านข้อมูลและแสดงออกมาซึ่งจะต้องเขียน ฟังก์ชั่นขึ้นชื่อ DataShow มาตรวจสอบซึ่งโค๊ดดังนี้

ฟังก์ชั่น DataShow

```
Public Static Sub DataShow(TextShow As Control, Data As String)
Const SpeedBaud = 16000
Dim LngSize As Long, X
LngSize = Len(TextShow.Text)
If TermSize > SpeedBaud Then
    TextShow.Text = Mid$(TextShow.Text, 4097)
    LngSize = Len(TextShow.Text)
End If
TextShow.SelStart = SpeedBaud
Do
    X = InStr(Data, Chr$(8))
    If X Then
        If X = 1 Then
        TextShow.SelStart = SpeedBaud - 1
        TextShow.SelLength = 1
        Data = Mid$(Data, X + 1)
        Else
```

```
    Data = Left$(Data, X - 2) & Mid$(Data, X + 1)
    End If
    End If
Loop While X
Do
    X = InStr(Data, Chr$(10))
    If X Then
    Data = Left$(Data, X - 1) & Mid$(Data, X + 1)
    End If
    Loop While X
    X = 1
    Do
    X = InStr(X, Data, Chr$(13))
    If X Then
    Data = Left$(Data, X) & Chr$(10) & Mid$(Data, X + 1)
    X = X + 1
    End If
Loop While X
TextShow.SelText = Data
TextShow.SelStart = Len(TextShow.Text)
End Sub
```

การเรียกใช้ฟังก์ชั่น DataShow :

```
Call DataShow Text1,(StrConv((StrData),vbUnicode))
```

ซึ่งจะทำให้มีตัวอักษรแสดงออกที่ TextBox หลักการเขียนโปรแกรมนี้ใช้ติดต่อกับอุปกรณ์ที่สามารถส่งค่า Acsii ทางSerial RS-232 ได้ อย่างเช่น Card แสดงผลของเครื่อง ชั่งน้ำหนัก,Card วัดอุณหภูมิ เพียงเราตั้งค่าของ Com Port ให้ตรงกับที่อุปกรณ์กำหนดมาก็ใช้

การใช้ Even ในโพรซีเดอร์ OnComm()

ตารางแสดงค่าต่างๆที่ใช้ใน MS Comm เราสามารถนำ Event ในตารางดังกล่าวมาใช้เขียนใน โพซีเดอร์ OnComm() โดยการใช้เงื่อนไขแบบ Select case ซึ่งสามารถเขียนได้ดังนี้

```
Private Sub MSComm1_OnComm()
Select Case MSComm1.CommEvent
Case comEvReceive
```

```
Dim Buffer As Variant
    Buffer = MSComm1.Input
    ShowData txtRXTX, (StrConv((Buffer), vbUnicode))
  Case comEvSend : 'ส่วนนี้จะใส่เงื่อนไขให้ทำอะไรก็ได้
Case comEvCTS
  Case comEvDSR
  Case comEvCD
  Case comEvRing
  Case comEvEOF
  Case comBreak
  Case comCDTO
  Case comCTSTO
  Case comDCB
  Case comDSRTO
  Case comFrame
  Case comOverrun
  Case comRxOver
  Case comRxParity
  Case comTxFull
End Select
End Sub
```

ถ้าใช้ไมโครคอนโทรลเลอร์ เช่น 8051,PIC เป็นต้น เราก็ต้องเขียนโปรแกรมคำสั่งโปรโตคอลกำหนดในไมโครคอนโทรลเลอร์ เสียก่อนว่าถ้ารับค่าจาก Serial Port มาแล้วให้ทำอะไร เพื่อที่เวลาจะส่งค่าไปที่ไมโครคอนโทรลเลอร์ได้รู้จักคำสั่งว่าจะให้ตัวไมโครคอนโทรลเลอร์ทำอะไรหรือส่งค่าอะไรกลับไปให้ทั้งนี้ต้องมีพื้นฐานความรู้ด้านภาษาแอสแซมบี้ด้วย

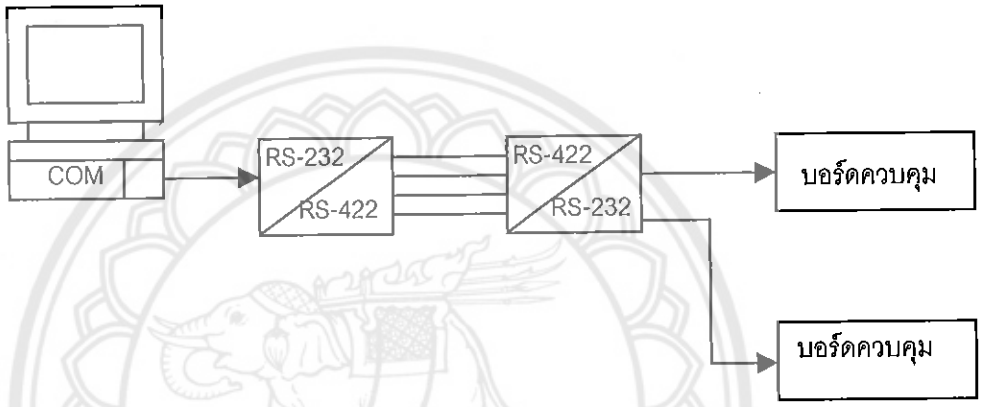
บทที่ 4

วิธีการดำเนินการทดลองและผลการทดลองโครงงาน

4.1 การทดลองการทำงานของบอร์ดควบคุมโดยใช้การรับส่งข้อมูลแบบ RS-422

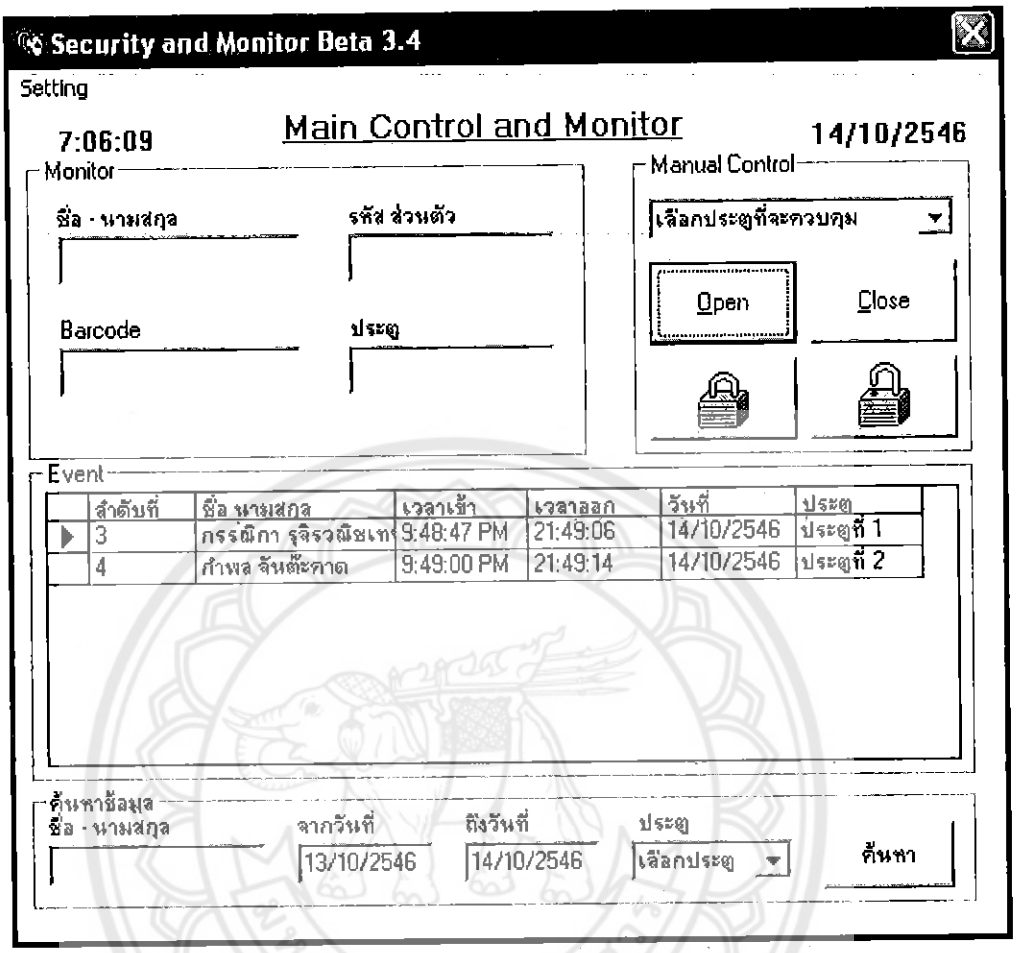
- **การเตรียมการก่อนการทดลอง**

1. ต่ออุปกรณ์ดังรูป



2. โปรแกรมทดลองบนเครื่องคอมพิวเตอร์เป็นโปรแกรมที่เขียนขึ้นด้วยโปรแกรม Visual basics 6
3. ทำการ Reset ตัวอ่านบาร์โค้ดด้วย Card reset
4. เข้าไปเพิ่มข้อมูลของผู้ทำการทดลองในระบบฐานข้อมูล
5. เปิดหน้าต่างโปรแกรมหลักแล้วทดสอบสั่งการปิดเปิดทางคอมพิวเตอร์เพื่อทดสอบว่าสามารถสื่อสารกันได้หรือไม่

4.2 การใช้งานของโปรแกรม

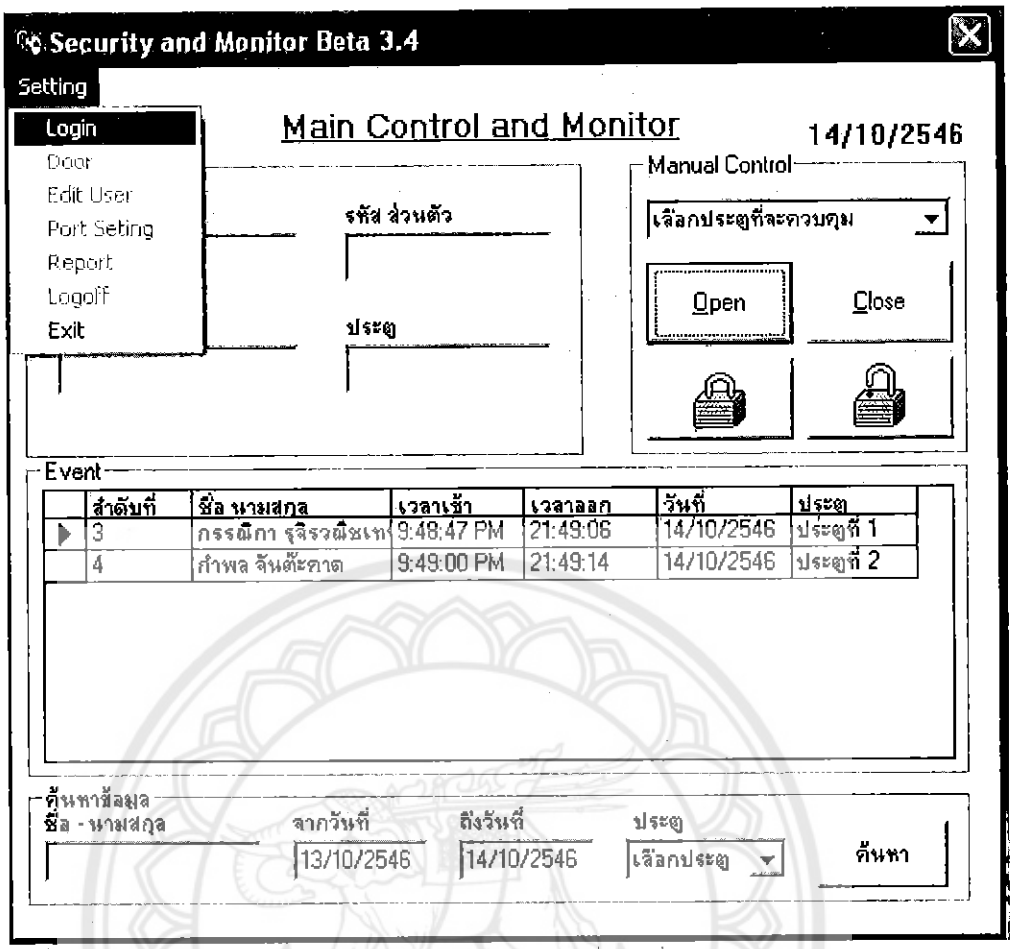


รูปที่ 4.1 หน้าต่างหลัก Main Control And Monitor

เป็นส่วนของ Monitor ซึ่งจะเป็นส่วนที่แสดงข้อมูลที่ส่งมาจากบอร์ดควบคุม คือ ชื่อ, รหัสส่วนตัว, บาร์โค้ด , ประตู

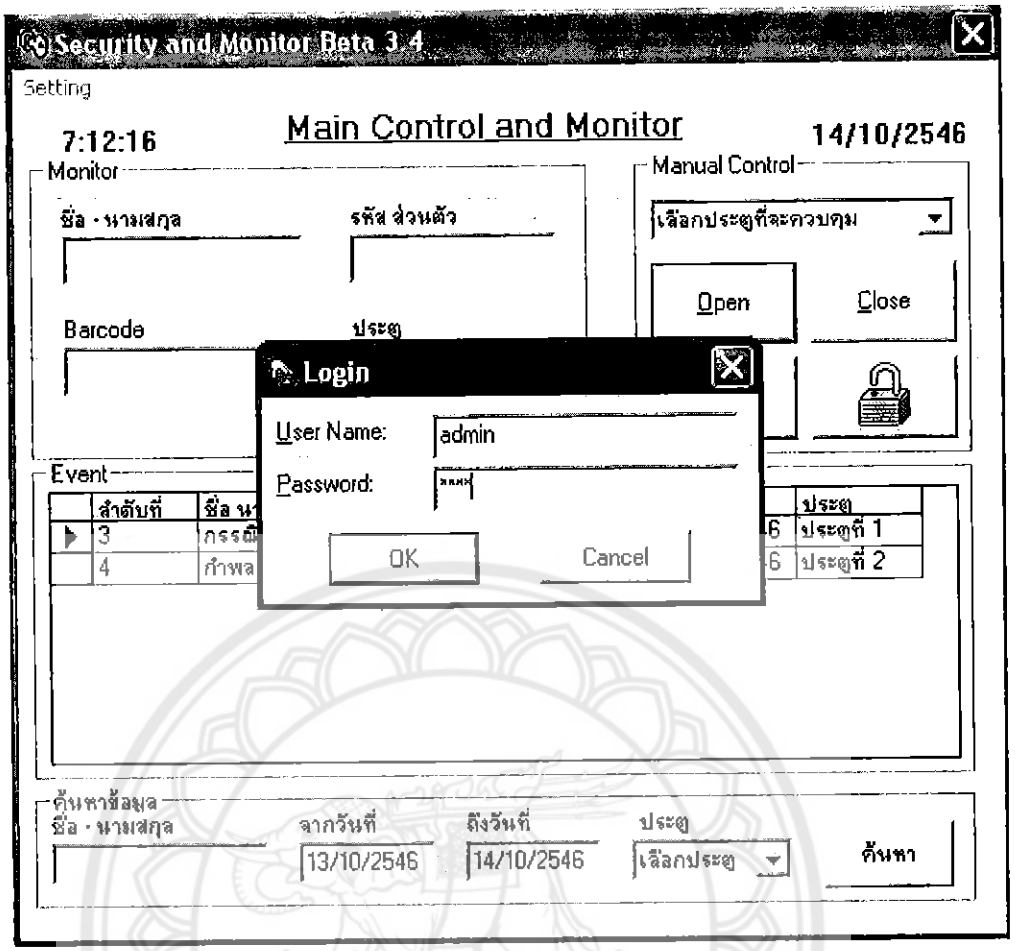
Manual Control คือ ส่วนที่สามารถควบคุมประตูให้ ปิด, เปิด, ล็อค จากหน้าจอ Monitor ได้

Event คือ ส่วนที่บันทึกเหตุการณ์ของข้อมูลการใช้ประตู



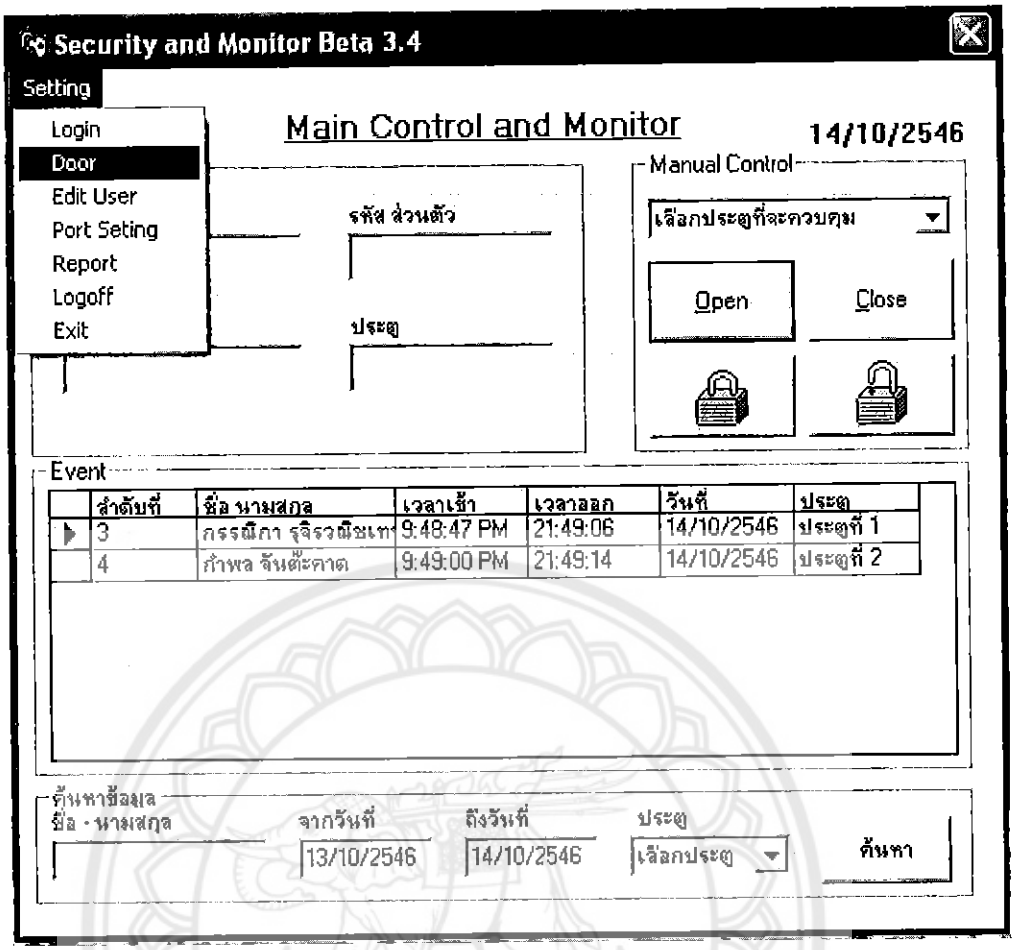
รูปที่ 4.2 การตั้งค่าต่าง ๆ ของโปรแกรม

ในกรณีที่จะตั้งค่าต่าง ๆ ต้องเป็นบุคคลที่มีหน้าที่ควบคุมการทำงานของระบบได้ โดยอันดับแรกต้องเข้าไปที่ Login ก่อน

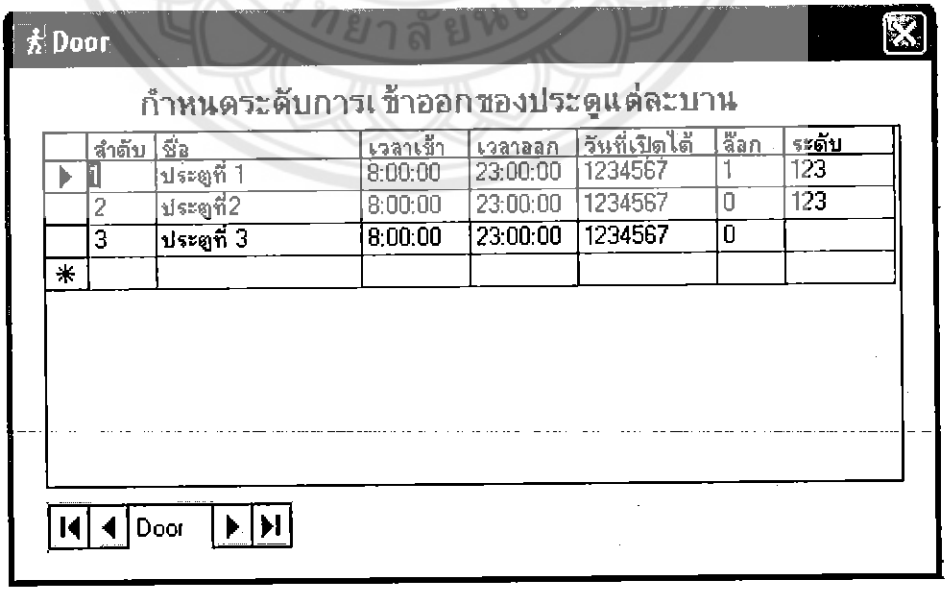


รูปที่ 4.3 Login

ก่อนจะทำการ Login ต้องใส่ชื่อและ Password ที่ถูกต้องในโปรแกรมนี้ ตั้ง User ไว้ชื่อ admin และรหัสผ่านคือ 2375 ถ้าหากจะแก้ไขทั้ง User และ Password ก็สามารถทำได้ในเมนูการตั้งค่าต่าง ๆ ที่จะกล่าวต่อไป

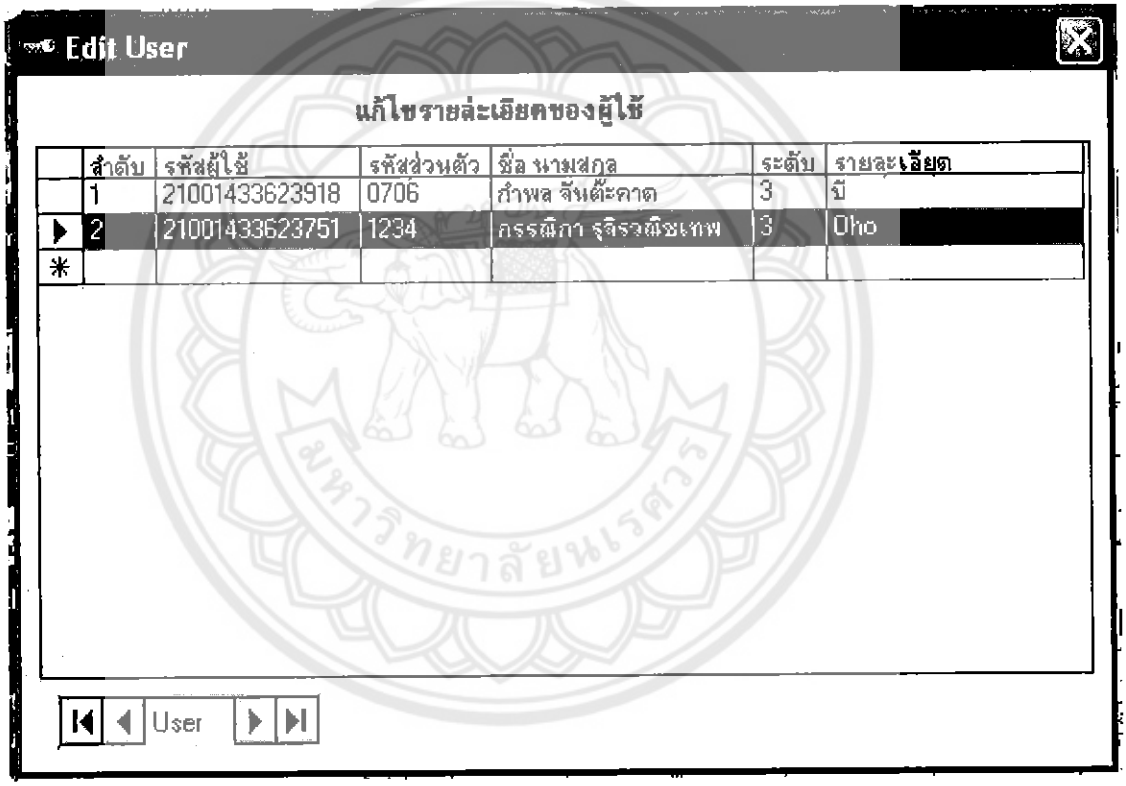


รูปที่ 4.4 เมนูตั้งค่า



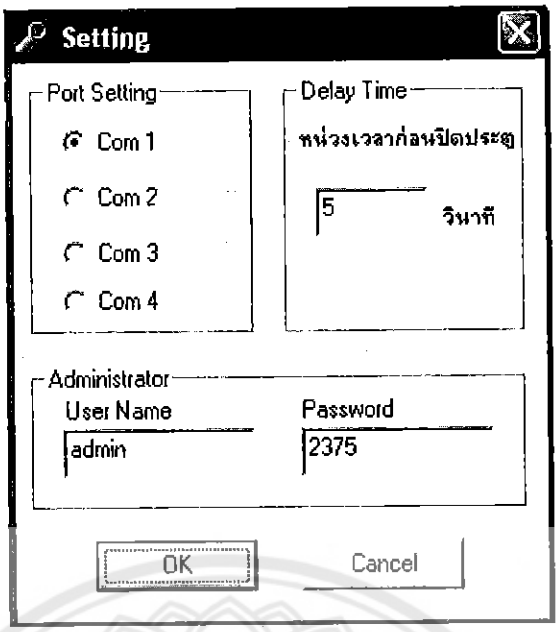
รูปที่ 4.5 (Door) การเพิ่มฐานข้อมูล และ จำนวนประตู

เมื่อเข้าไปเมนูแรก คือการเข้าไปเพิ่มเติมแก้ไขฐานข้อมูลของประตู วิธีเพิ่มประตูก็คือเข้าไปพิมพ์ข้อมูลของประตูใหม่ที่เพิ่มเข้ามา สำหรับลำดับนั้นไม่ต้องพิมพ์เพราะคอมพิวเตอร์จะรันให้เอง ใส่ชื่อประตู แล้วไปช่องกำหนดเวลาเข้า และเวลาออกที่จะอนุญาตให้สามารถเข้าออกได้ในช่วงเวลาที่กำหนด สำหรับช่องวันที่เปิดได้นั้นสามารถเลือกได้ว่าจะให้ใช้งานได้ในวันไหนบ้างโดยการเลือกตัวเลข 1-7 (1 คือ วันอาทิตย์, 2 คือ วันจันทร์... , 7 คือ วันเสาร์) ช่องต่อไปก็คือช่องบ่งบอกสถานะ "1" คือ ล็อคไว้ไม่มีใครสามารถเข้าใช้ได้ "0" คือไม่ได้ล็อค และช่องสุดท้ายคือการกำหนดระดับของผู้ใช้ "1" คือ นิสิตทั่วไป "2" คือ ระดับอาจารย์ "3" คือระดับ admin (ผู้ควบคุมระบบ) เมื่อทำการเพิ่มฐานข้อมูลเสร็จในแต่ละบรรทัดก็ให้กด ลูกศรให้เลื่อนไปอีกบรรทัดหนึ่งเพื่อให้คอมพิวเตอร์บันทึกข้อมูล



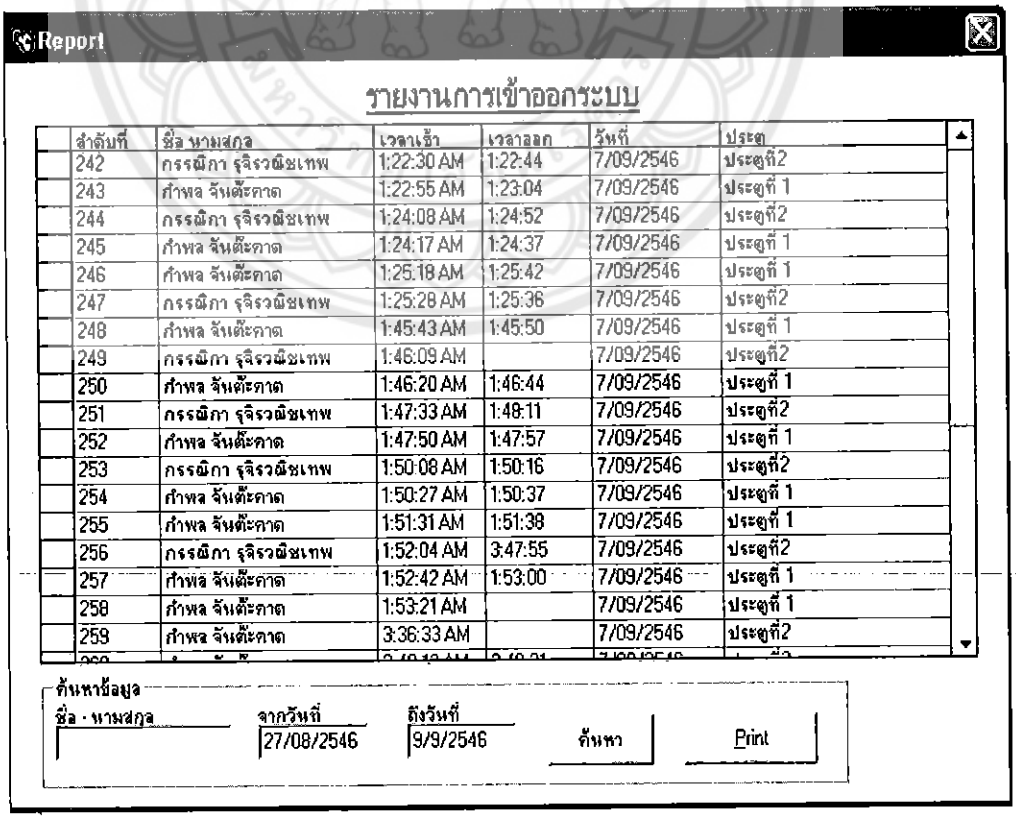
รูปที่ 4.6 Edit User เมนูเพิ่มฐานข้อมูลข้อผู้ใช้

เมื่อเข้ามาที่เมนู Edit user ก็จะปรากฏหน้าต่างนี้ หลังจากนั้นก็คือกรอกข้อมูลส่วนบุคคลเข้าไปให้ครบทุกช่องก็สามารถเพิ่มฐานข้อมูลได้



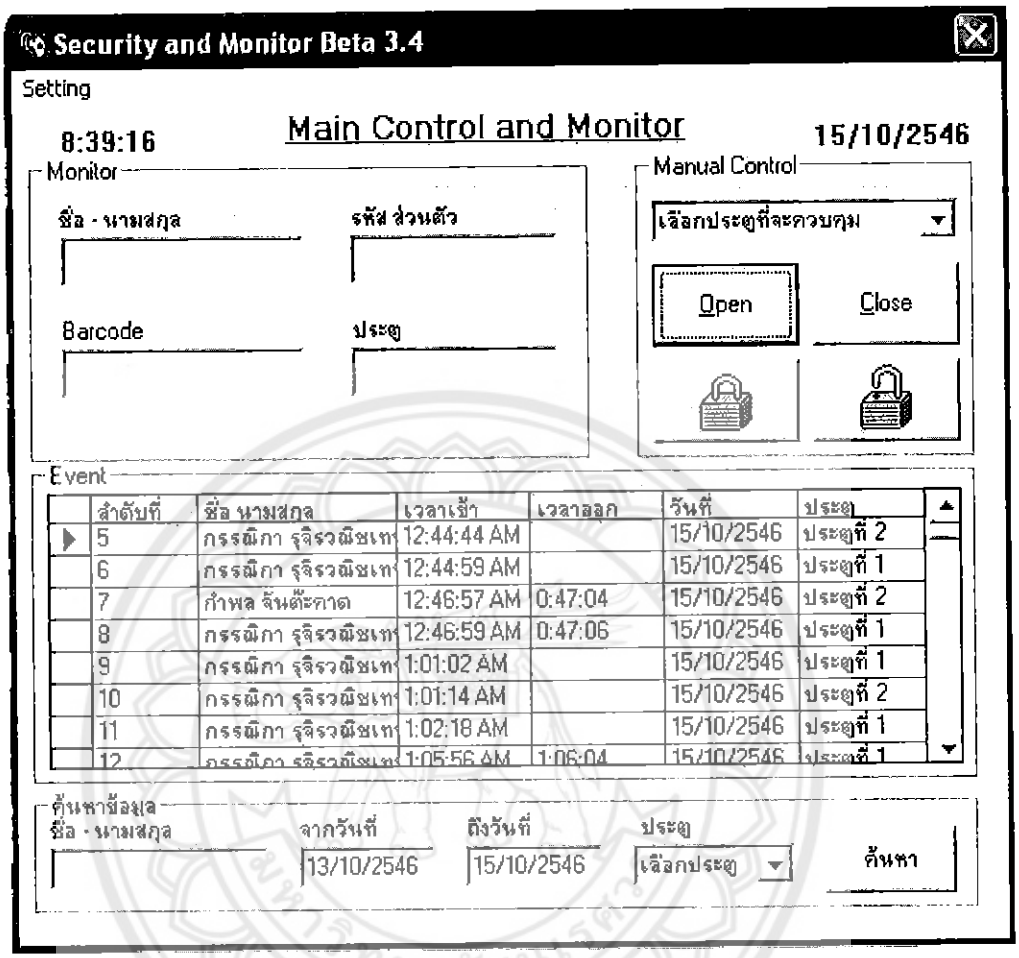
รูปที่ 4.7 การกำหนด Port Com, เวลาก่อนปิดประตู และ ข้อมูลของผู้เข้าไป

ในเมนูนี้สามารถเลือก Com port, ระยะเวลาที่ประตูเปิดและสามารถแก้ไขข้อมูลของผู้ควบคุมได้



รูปที่ 4.8 ตารางรายงานการเข้าและออกของประตู

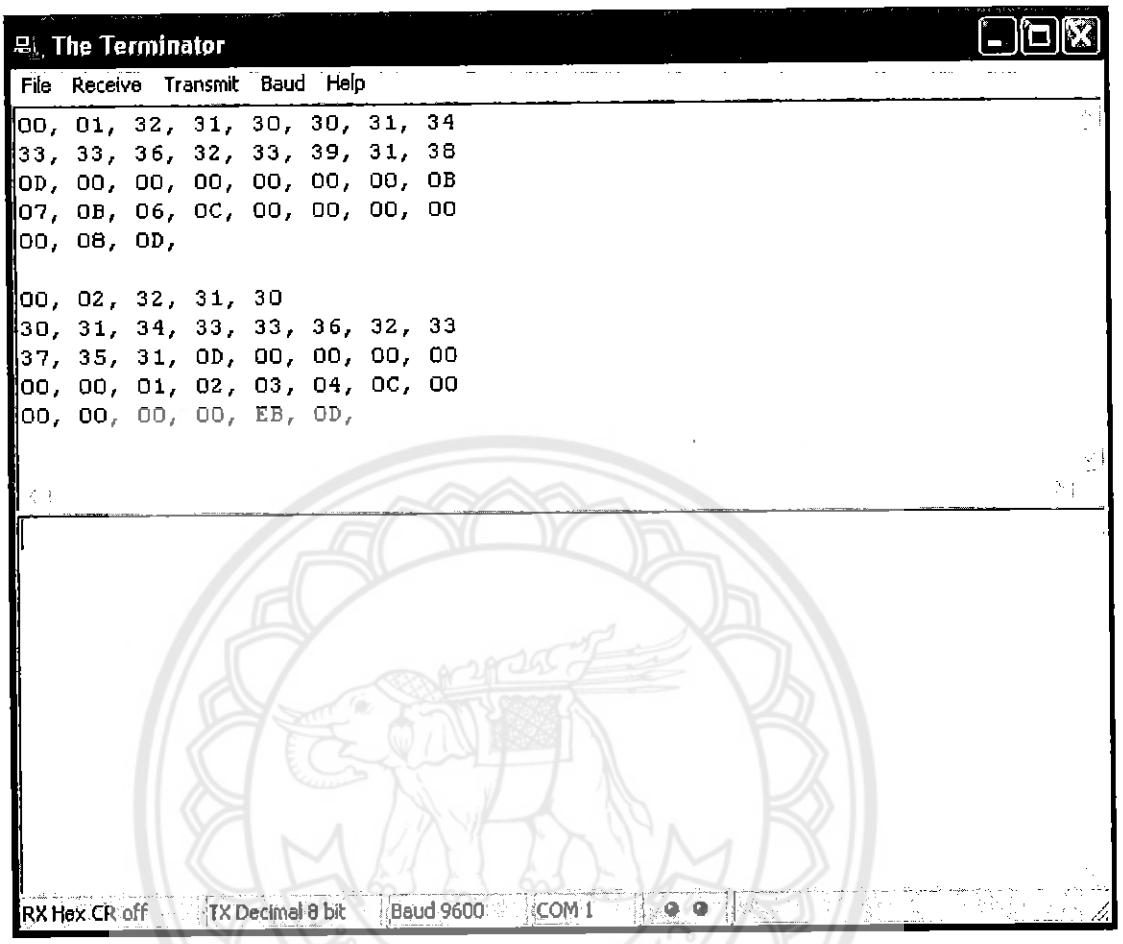
จากรูปที่ 4.8 เป็นส่วนรายงานการเข้าออกทั้งหมด สามารถค้นหาข้อมูลได้จากหน้าต่างนี้



รูปที่ 4.9 เมนูการค้นหาข้อมูลของการใช้ในช่วงเวลาที่ต้องการ

ในเมนูการค้นหานี้เราสามารถ Key ข้อมูลลงไปในช่องใดช่องหนึ่งก็ได้ไม่จำเป็นต้องใส่ข้อมูลครบทุกช่องก็สามารถค้นหาข้อมูลการใช้ในช่วงที่ต้องการทราบข้อมูลได้ แต่ถ้าใส่ข้อมูลให้ครบทุกช่องก็มีผลให้สามารถค้นหาข้อมูลได้ระเอียดมากขึ้น

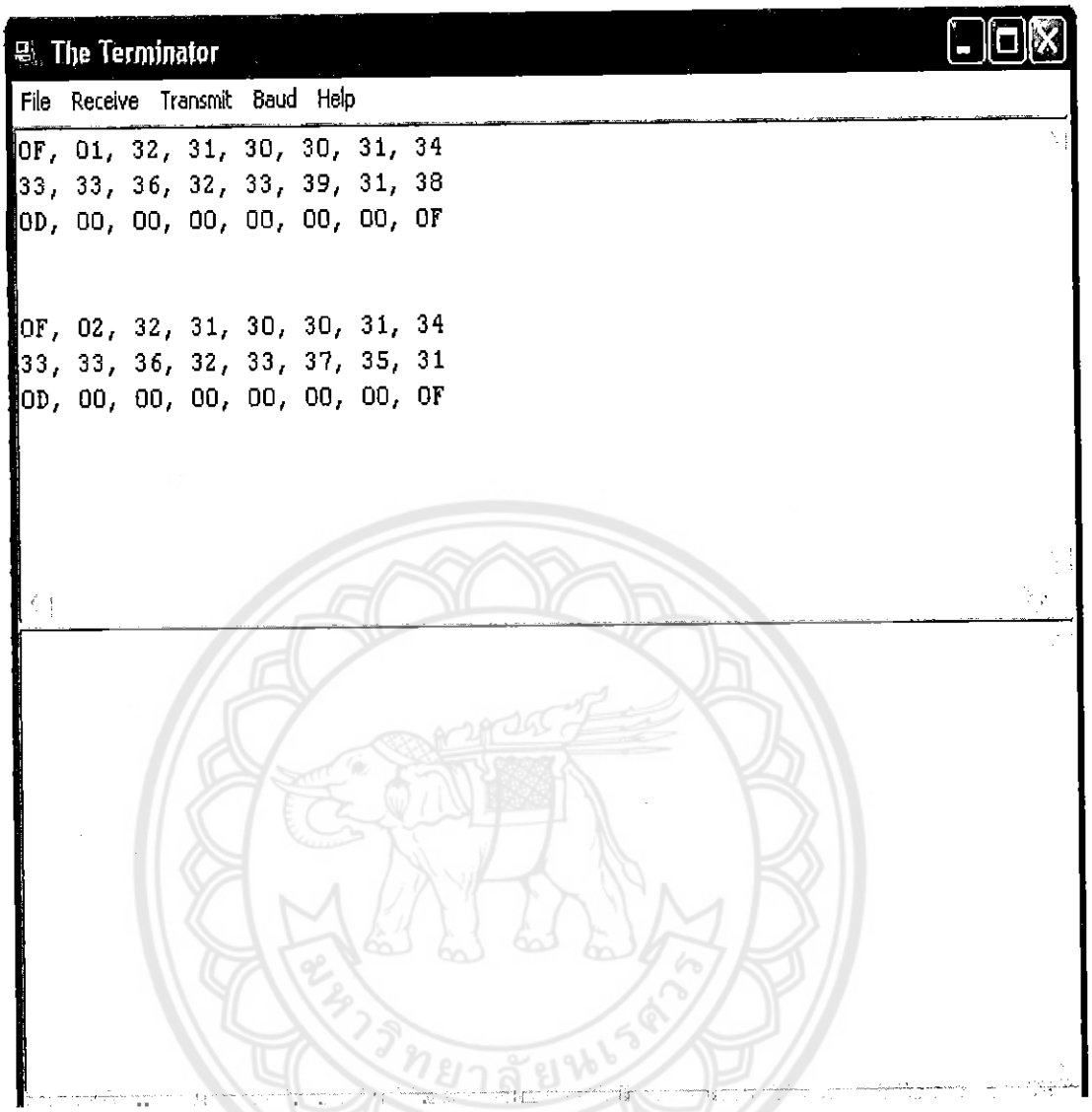
4.3 การใช้งานของ TERMINAL ในการตรวจสอบค่าต่าง ๆ ดังจะอธิบายต่อไปนี้



รูปที่ 4.10 แสดงการรับข้อมูลการเช็คเข้าขึ้นมาจากบอร์ดควบคุม 1 และ 2 ตามลำดับ

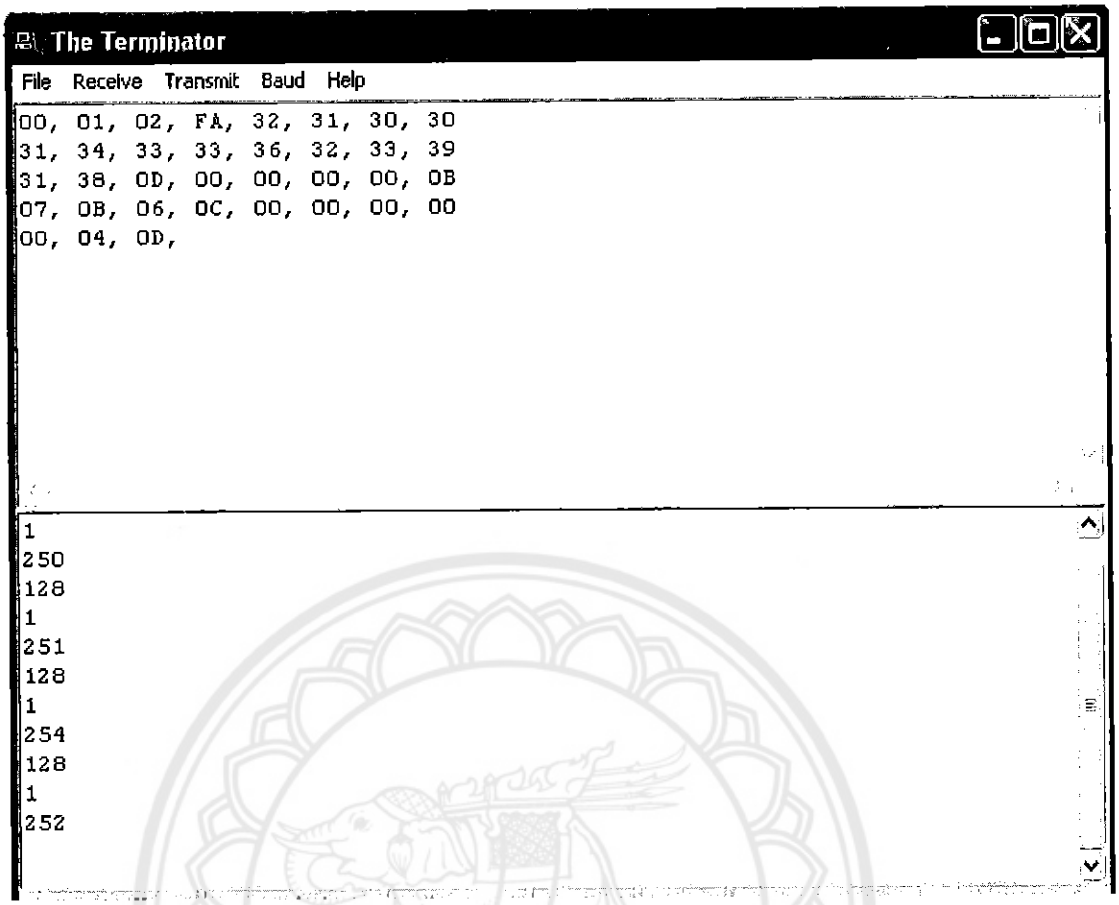
โปรแกรม Terminator

การทดสอบข้อมูลที่ส่งมาทั้งสองบอร์ด เริ่ม Start ที่ 00 แล้วตามด้วย 01 หมายถึง หมายเลขเครื่อง ตามด้วยรหัส ASCII เช่น32 = 2 , 31= 1 จะเป็นข้อมูลส่วนของบาร์โค้ดที่จะต้องนำด้วย 3 ข้อมูลทั้งหมดของตัวที่1 คือ 2100143362391 แล้วจะตามด้วย บิต Stop คือ 0D หลังจากนั้นก็จะเป็นส่วนของ Pass Wordที่สามารถรับได้ 10 ตัว(กำหนดไว้) หลังจาก Pass word เลข 0C ที่ปรากฏคือ การกด Enter ก่อนที่การส่งข้อมูลจะสิ้นสุดตัวเลขก่อนตัวสุดท้ายคือผลรวมของข้อมูลทั้งหมดเริ่มต้นที่ ID จนถึงตำแหน่งที่ 33 เป็นตัวเลขที่ใช้เช็คความถูกต้องของข้อมูลที่ส่งระหว่าง คอมพิวเตอร์กับ ไมโครคอนโทรลเลอร์ ส่วนเทคนิคการบวกคือ บวกกันแบบฐาน 16 ถ้าเกิน FF ก็ต้องลบด้วย FF แล้วเก็บเศษที่เหลือ เพื่อป้องกันไม่ให้เกิน 8 บิตแล้วก็จะตามบิดจบคือ 0D



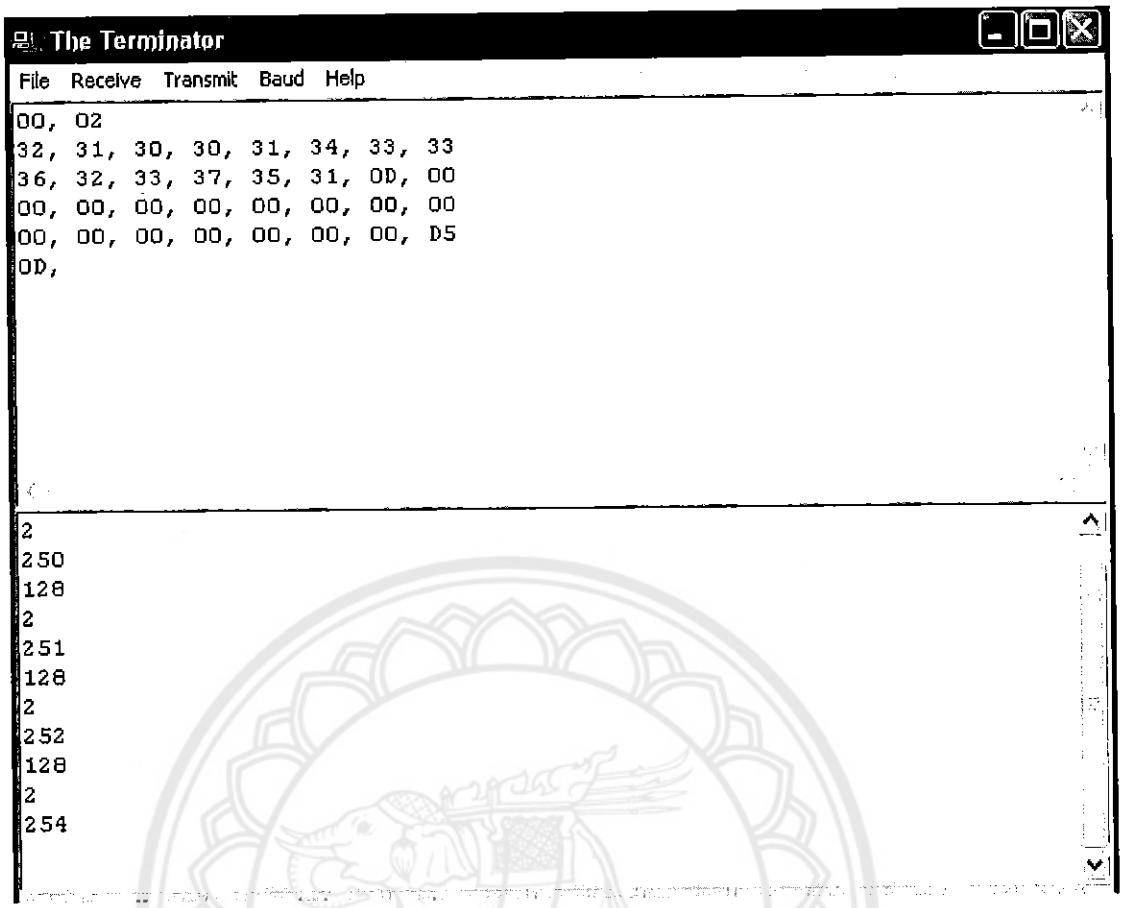
รูปที่ 4.11 แสดงการรับข้อมูลการเช็คออกขึ้นมาจากบอร์ดควบคุม 1 และ 2 ตามลำดับ

หน้าต่างนี้ ปีนการแสดงข้อมูลของการเช็คออกจากประตูโดยที่จะรับมาแค่บาร์โค้ดเพราะไม่มีการกด Pass word แล้วจึงนำมาเปรียบเทียบกับฐานข้อมูลว่าผู้ใช้มีสถานะอยู่ในห้องหรือไม่ถ้าใช่ก็จะยอมรับให้ผ่าน ซึ่งข้อมูลชุดนี้จะเริ่มต้นที่ 0F และปิดท้ายที่ 0F



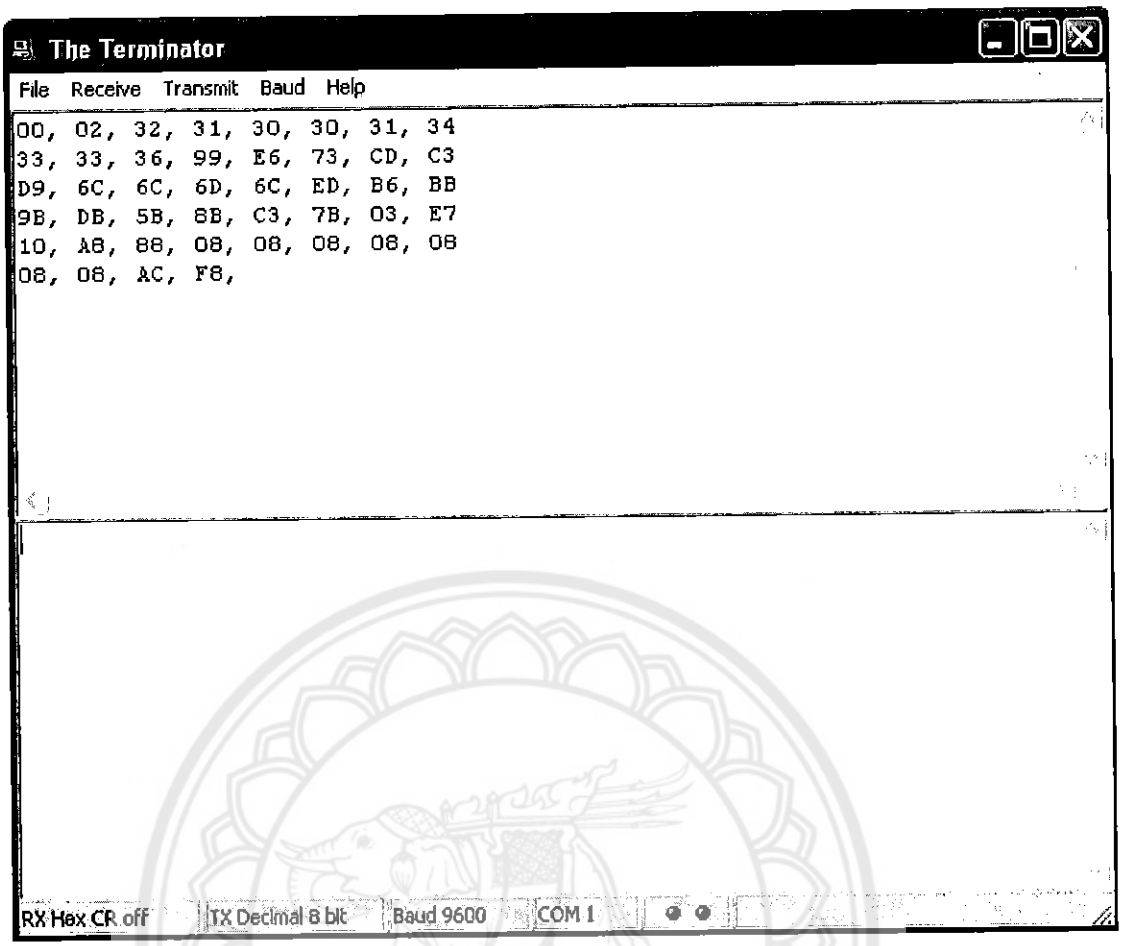
รูปที่ 4.12 แสดงการส่งคำสั่งจากคอมพิวเตอร์ไปยังบนบอร์ดควบคุมตัวที่ 1 เพื่อทดสอบการสื่อสาร

หน้าต่างนี้แสดงการส่งข้อมูลคำสั่งจากคอมพิวเตอร์ไปให้กับบอร์ดควบคุมบอร์ดที่ 1 โดยส่งไป 3 ชุด คือชุดแรก เป็นการส่งข้อมูลไป (128=แล้วแต่การกำหนด)ให้บอร์ดรับรู้ว่าคอมจะส่งคำสั่งมาให้ ส่วนข้อมูลชุดที่ 2 คือเป็น ID ของบอร์ด (1,2,...) และชุดที่ 3 คือ คำสั่งต่าง ๆ ที่เรากำหนดขึ้นมา 250 = เปิดประตู, 251= ปิดประตู, 252 = แสดงไฟเขียวกระพริบแสดงว่าเข้าไม่ได้, 254 = สั่งให้บอร์ดส่งข้อมูลขึ้นมา



รูปที่ 4.13 แสดงการส่งคำสั่งจากคอมพิวเตอร์ไปยังบอร์ดควบคุมตัวที่ 2 เพื่อทดสอบการสื่อสาร

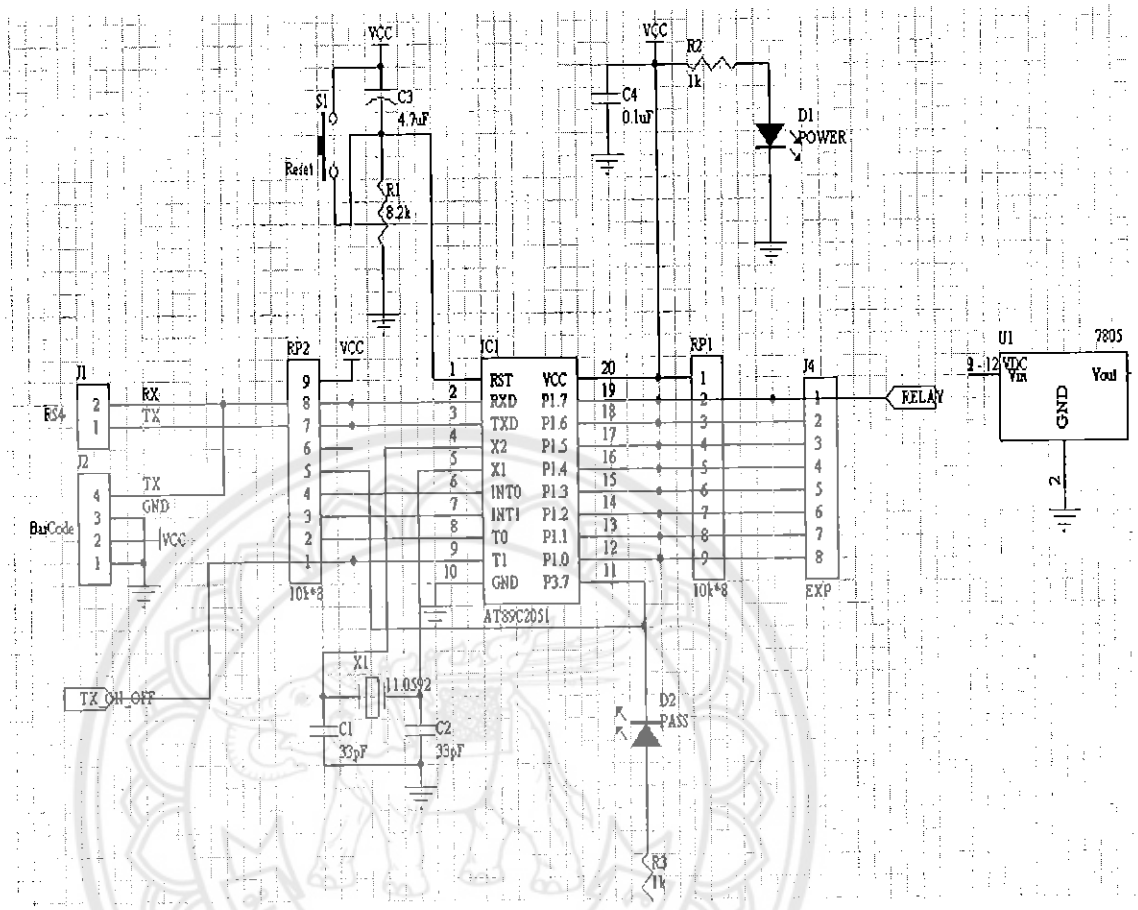
หน้าต่างนี้แสดงการส่งข้อมูลคำสั่งจากคอมพิวเตอร์ไปให้กับบอร์ดควบคุมบอร์ดที่ 2 โดยส่งไป 3ชุด คือชุดแรก เป็นการส่งข้อมูลไป (128=แล้วแต่การกำหนด) ให้บอร์ดรับรู้ว่าคอมจะส่งคำสั่งมาให้ ส่วนข้อมูลชุดที่ 2 คือเป็น ID ของบอร์ด (1,2,...) และชุดที่ 3 คือ คำสั่งต่าง ๆที่เรากำหนดขึ้นมา 250 = เปิดประตู , 251= ปิดประตู , 252 = แสดงไฟเขียวกระพริบแสดงว่าเข้าไม่ได้ , 254 = สั่งให้บอร์ดส่งข้อมูลขึ้นมา



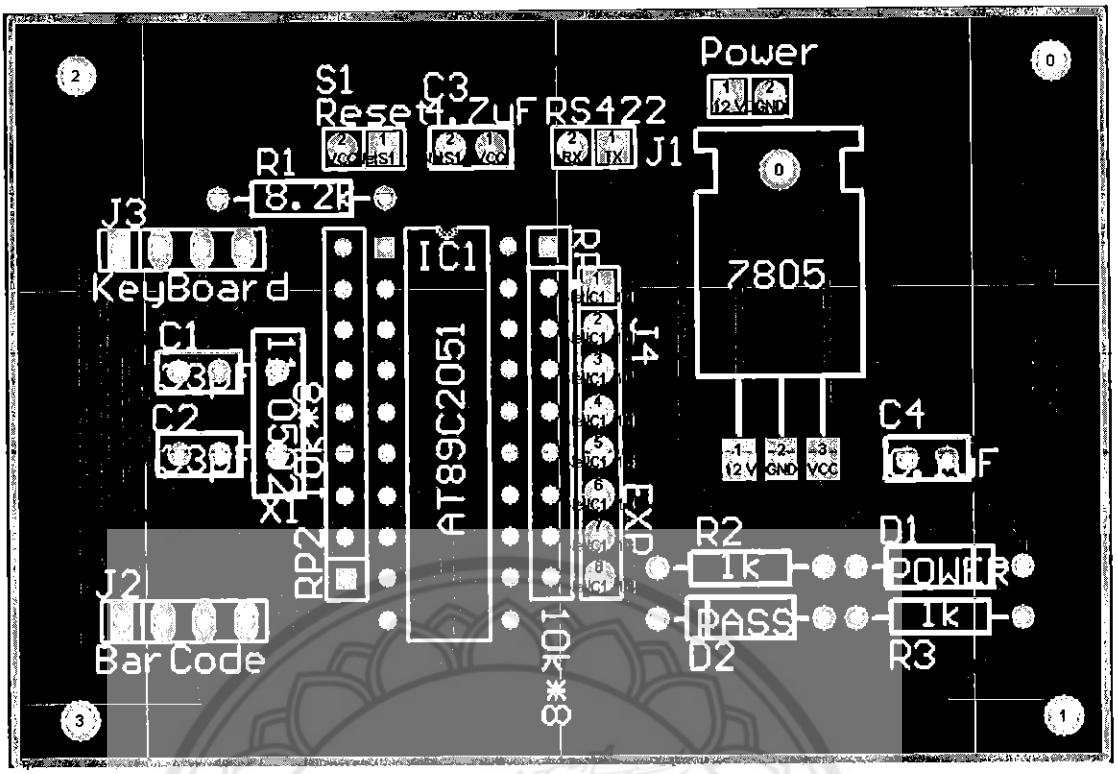
รูปที่ 4.14 แสดงการส่งข้อมูลมาซ้อนทับกัน

หน้าต่างนี้แสดงการส่งข้อมูลมาพร้อมกันจะทำให้ข้อมูล 2 ชุด ของบอร์ดทดลองทั้งสองมาซ้อนทับกันเป็นสาเหตุที่ทำให้เกิด Error สังเกตดูจากข้อมูลจะเห็นว่าบอร์ดที่ 2 สามารถส่งมาได้ก่อนแล้วจะมีการซ้อนกันของสัญญาณแล้วจะเป็นตัวเลขที่ไม่ได้อยู่ในตาราง ASCII ทำให้คอมพิวเตอร์ไม่สามารถรับสัญญาณได้

4.4 การติดต่อกับพอร์ทต่าง ๆ ของ MCS-51 AT89C2051



รูปที่ 4.15 วงจรของบอร์ดควบคุม

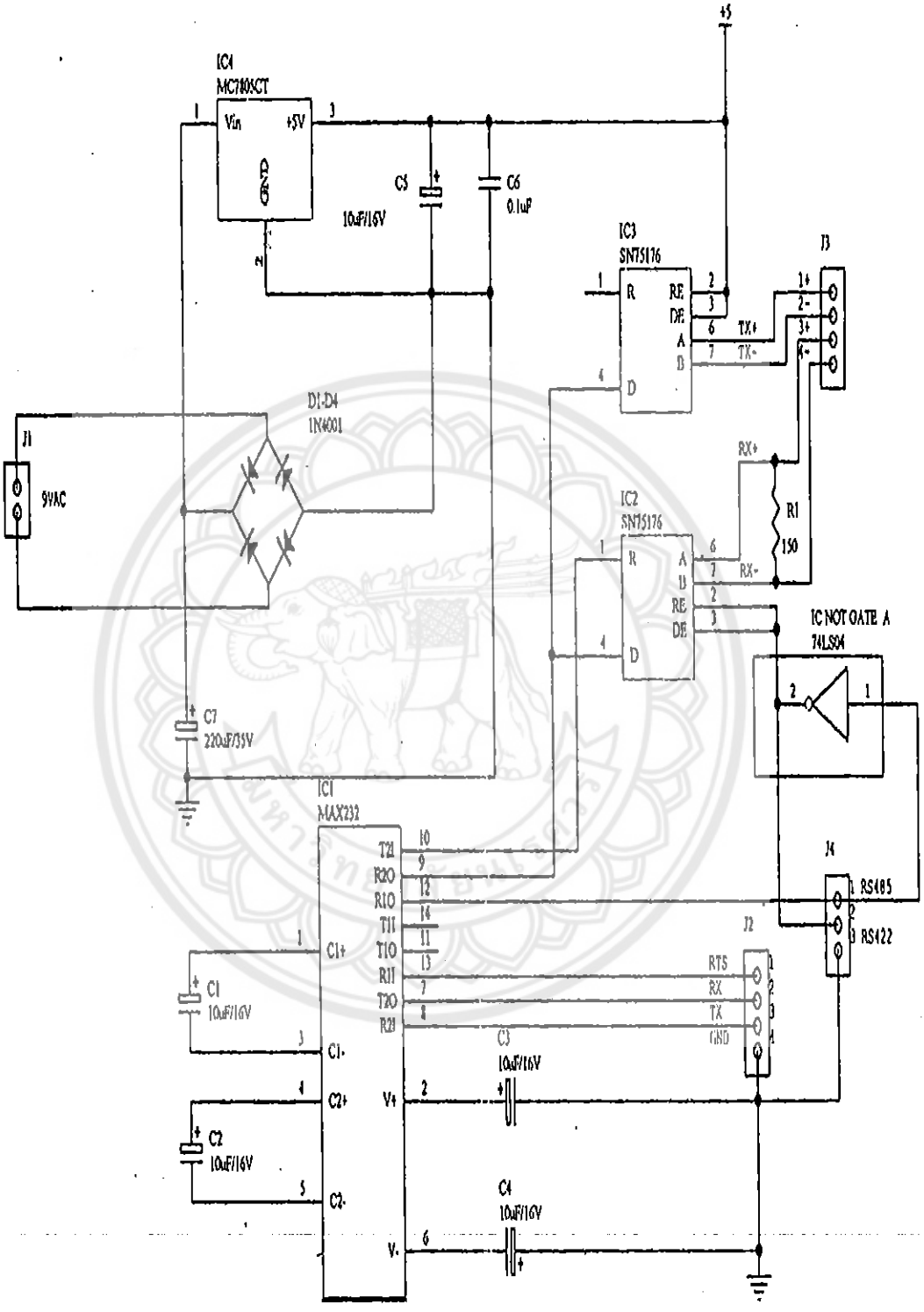


รูปที่ 4.16 วงจรพิมพ์ของบอร์ดควบคุม

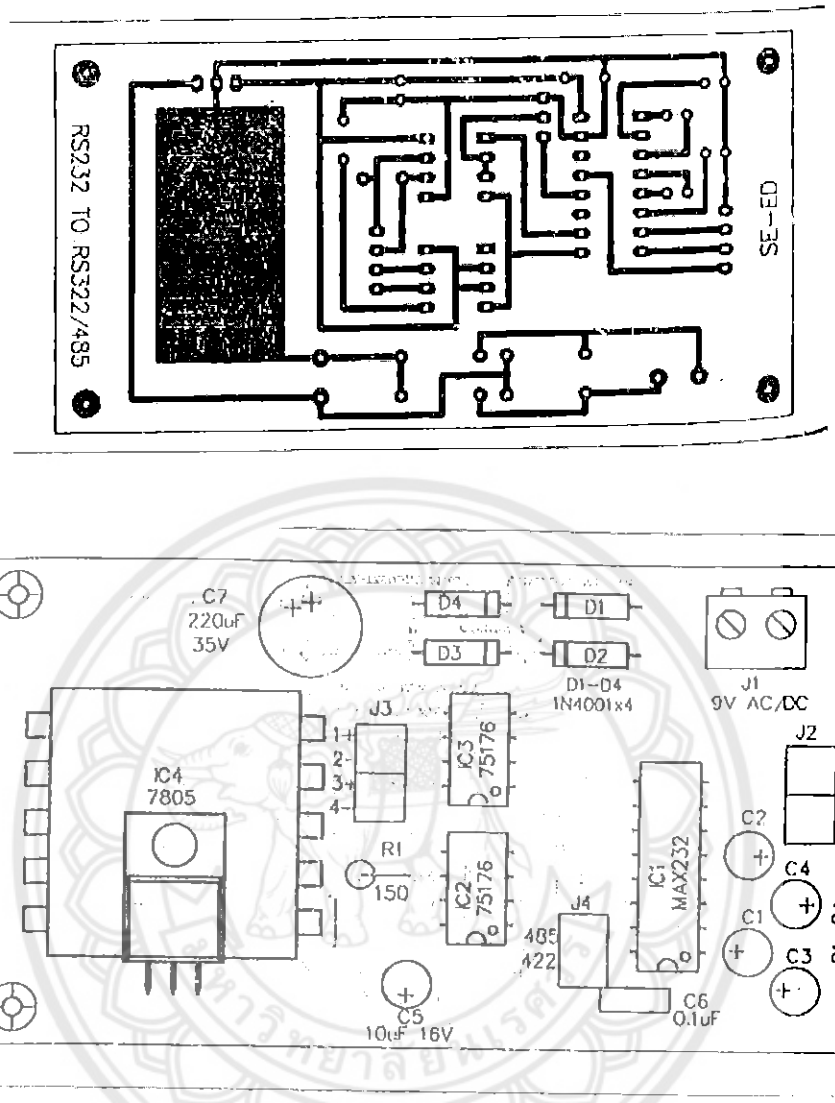
4.4.1 วงจรบอร์ดควบคุม

ในวงจรของบอร์ดควบคุมประกอบด้วย IC MCS 51 ของ ATMEL AT89C89C2051 ซึ่งเป็น CPU ที่มี 20 ขา ที่ติดต่อกับอุปกรณ์ต่อพ่วงดังนี้ ชุดแปลงสัญญาณ RS 422/485 โดยขา tx จะต่อกับ R-Pack ที่ขา 8 แล้วเข้าสู่ MCS-51 ที่ขา 3 (TXD) และ สัญญาณ (RXD), และสัญญาณ Rx ต่อผ่าน R-Pack ขาที่ 7 เข้าสู่ AT89C2051 ขา 2 (RXD), อุปกรณ์ barcode reader จะติดต่อกับ MCS-51 โดยขา tx ของ barcode reader จะต่อเข้ากับขา RXD ของ MCS-51 โดยขา RXD ของ MCS-51 คือขาที่ 2 ทั้งสัญญาณ RS 422 จาก computer และ barcode จะเป็นส่วนของการรับ-ส่งข้อมูล ส่วนขาที่ใช้ควบคุม mosfet IRFZ 44n เพื่อ trick on-off relay

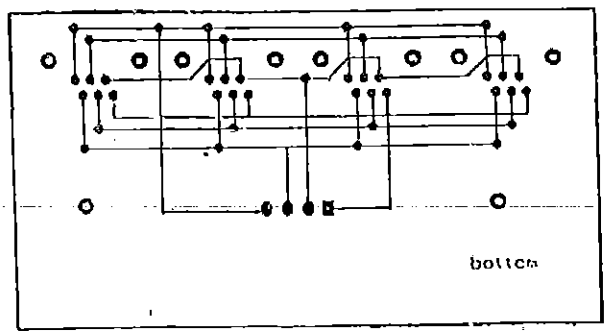
สำหรับแหล่งจ่ายไฟของวงจรนี้จะรับภาคจ่ายไฟมาจาก IC 7805 ซึ่งจะรักษาระดับแรงดันที่ +5 v โดยจ่ายเข้าที่ขา 20 และขา 10 ซึ่งเป็นขากราวด์



รูปที่ 4.17 วงจรของชุดแปลงสัญญาณ RS 232-422/485



รูปที่ 4.18 วงจรพิมพ์ชุดแปลงสัญญาณ RS232 เป็น RS422/485



รูปที่ 4.19 วงจรพิมพ์ Connector RJ11

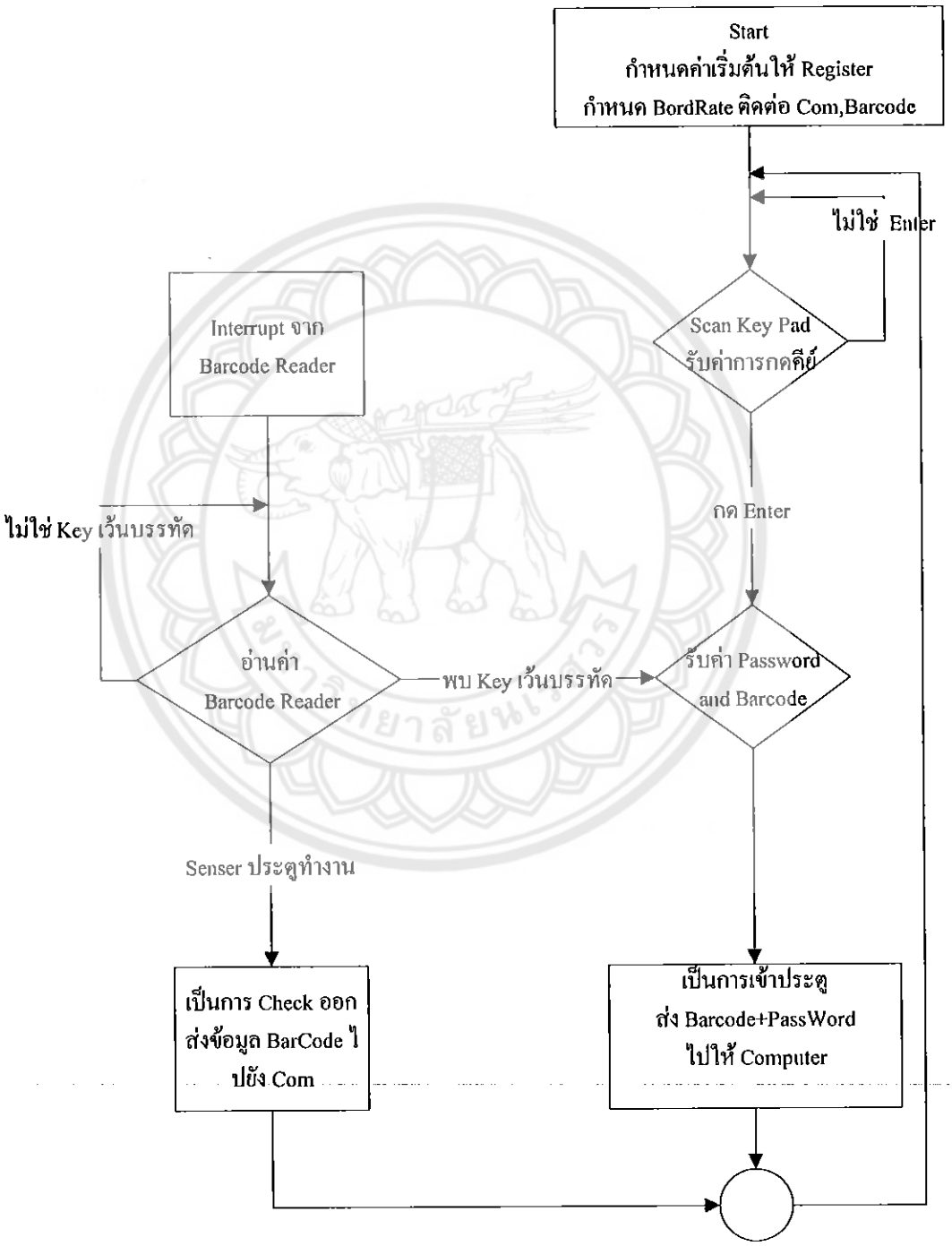
4.4.2 บอร์ดชุดแปลงสัญญาณ

ในชุดแปลงสัญญาณจาก RS 232 ไปเป็น RS 422/485 โดยปกตินั้นมาตรฐานของ RS 232 จะเป็นสัญญาณแบบ TTL เมื่อผ่าน line drive ก็จะเปลี่ยนจากสัญญาณ TTL มาเป็นระดับแรงดันที่ต่างกัน ± 12 v เทียบกับกราวด์ มาตรฐาน RS232 นี้จะมีข้อจำกัดทางด้านสัญญาณรบกวน ซึ่งส่งได้จำกัดในระยะ 50 ฟุต (15 เมตร)

ในโครงงานนี้ต้องการพัฒนาให้สามารถควบคุมได้ระยะไกล จึงต้องหามาตรฐานการส่งใหม่โดยใช้มาตรฐาน RS 422/485 ซึ่งทั้งคู่นี้จะมีการสื่อสารในลักษณะเดียวกัน คือใช้แบบ balance line หมายถึง ไม่ได้เทียบกราวด์ แต่เป็นการใช้ระดับสัญญาณที่ต่างกันคือ ±2 v ถึง ±6 v ฉะนั้นจึงทำการแปลงสัญญาณ RS 232 มาเป็น RS 422/485 เริ่มต้นจากที่รับสัญญาณ TTL มาจากคอมพิวเตอร์แล้วมาเข้า line driver MAX 232 ซึ่งเปลี่ยนสัญญาณ TTL มาเป็นลอจิกที่มีความต่างศักย์คือ ± 12 v แล้วส่งไปให้ IC 78176 ซึ่งมีหน้าที่ทำให้สัญญาณเปลี่ยนมาเป็นแบบ balance line คือมี Tx+, Rx+, Tx-, Rx- สามารถส่งให้ไกลขึ้นได้ 4,000 ฟุต

สำหรับภาคจ่ายไฟนั้นจะรับ AC 9 v มาเข้าที่ rectifier ผ่าน C 7 เพื่อกรองกระแสไฟให้เรียบแล้วส่งไปที่ IC 7805 เพื่อรักษาระดับแรงดัน 5 v เพื่อจ่ายไฟให้กับวงจร

การทำงานของชุดควบคุม



รูปที่ 4.20 การทำงานของชุดควบคุม

4.4.3 การทำงานของชุดควบคุม (MICRO)

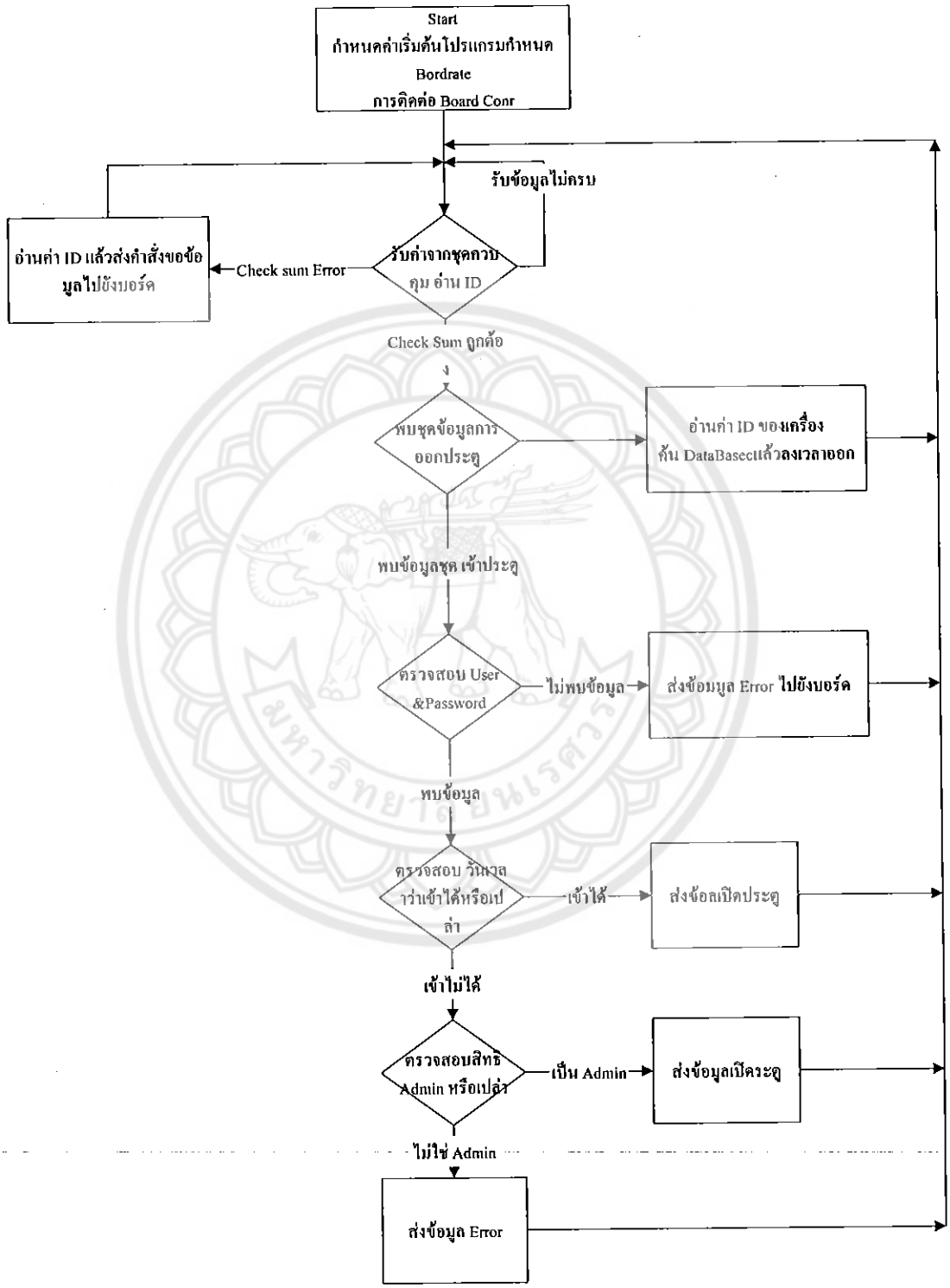
เริ่มต้นที่การกำหนดค่าเริ่มต้น Register ต่าง ๆ เช่น การกำหนด Baud Rate, Port Com, ID, Key Pad และที่เกี่ยวข้อง

ในสถานการณ์ปกติ Board จะทำการ scan Key Pad อยู่ตลอดเพื่อรอการอินเตอร์รัพท์ จาก barcode reader เมื่อมีการอ่านค่าจาก barcode reader board ก็จะรอรับข้อมูลจาก Key Pad หรือ sensor ถ้าหากมีการอ่านค่าบาร์โค้ดเข้ามาแล้วมีการกด Key Pad บอร์ดก็จะรับรู้ว่าเป็นการเช็คเข้า และอีกกรณีที่มีการอ่านบาร์โค้ดแล้ว sensor ทำงานบอร์ดก็จะรับรู้ว่าเป็นการออก

กรณีการเช็คเข้า คือ มีการกด Key Pad หลังจากมีการอ่านข้อมูลจากบาร์โค้ด ขั้นตอนต่อไปคือ บอร์ดควบคุมจะทำการส่งข้อมูลไปให้คอมพิวเตอร์ ข้อมูลนี้ก็คือ start bit + ID + Barcode + Password + stop bit

กรณีการเช็คออก คือ เมื่อมีการอ่านบาร์โค้ดเข้ามาแล้ว sensor ทำงาน บอร์ดจะรู้ว่าเป็นการเช็คออก และในระหว่างนั้นบอร์ดจะส่งข้อมูลไปให้คอมพิวเตอร์ ข้อมูลก็คือ start bit + ID + Barcode ++ stop bit

การทำงานของโปรแกรม



รูปที่ 4.21 การทำงานของโปรแกรมบน WINDOW

4.4.4 ขั้นตอนการทำงานของโปรแกรมบน WINDOW

เริ่มต้นที่การกำหนดค่าต่าง ๆ เช่น MSComm เพื่อติดต่อกับบาร์โค้ดและค่าต่าง ๆ ที่เกี่ยวข้อง ขั้นตอนต่อไปคือ การรอรับค่าจากบอร์ดควบคุม เมื่อรับค่ามาแล้วจะมีขบวนการเช็ค sum ถ้าการเช็ค sum error ก็จะส่งคำสั่งไปขอข้อมูลใหม่ (ไฟเขียวกระพริบ) ถ้าหากว่าข้อมูลที่รับมาและ password ถูกต้องแล้วก็จะตรวจสอบว่าเป็นการเช็คเข้าหรือเช็คออก ถ้าหากเป็นการเช็คออกก็จะส่งคำสั่งไปเพื่อเปิดประตูและลงเวลาออก

ถ้าเป็นการเช็คเข้าจะมีการตรวจสอบ Password ถ้าหากบาร์โค้ดและ password ไม่ถูกต้องก็จะส่งคำสั่ง error ไปแต่ถ้าข้อมูลถูกต้องก็ไปขั้นตอนการตรวจสอบเวลาที่กำหนดการเข้าออกจาก data base ว่าเป็นช่วงเวลาที่เข้าได้หรือไม่ถ้าอยู่ในช่วงเวลาที่เข้าได้ประตูก็จะเปิด ถ้าเข้าไม่ได้ก็ไปขั้นตอนการตรวจสอบว่าอยู่ในระดับ admin หรือไม่ ถ้าเป็น admin ก็จะสั่งเปิดประตูได้ แต่ถ้าไม่ใช่ admin ก็จะเข้าไม่ได้และส่งคำสั่ง error ไปยังบอร์ด

4.5 การทดลอง

1. ทำการติดต่อกับบอร์ดควบคุม (Log on) โดยการใช้บัตรประจำตัวรูดที่ตัวอ่านบาร์โค้ด
2. กดปุ่ม * แล้วตามด้วยรหัสผ่าน แล้วกด #เป็นการEnter
3. สังเกตดูไฟ LED ว่าไฟเขียวติดหรือไม่
4. เมื่อไฟเขียวติดก็สังเกต ที่โซลินอยด์
5. ขั้นตอนการออก ทำการรูดบัตรกับตัวอ่านบาร์โค้ดที่อยู่ข้างในแล้วเดินผ่าน Sensor แล้วประตูจะเปิด
6. ทดลองการใช้งานร่วมกันทั้งสองบอร์ด

บทที่ 5
สรุปผลการทดลองโครงงาน ปัญหา และข้อเสนอแนะ

5.1 สรุปผลการทดลองโครงงาน

จากการศึกษา ออกแบบ ทดลอง รวมทั้งการแก้ไขปัญหาต่าง ๆ โครงงานนี้สามารถติดต่อควบคุม ส่งได้ครั้งละหลายบอร์ดแต่จะสามารถรับข้อมูลที่ส่งมาจากบอร์ดควบคุมครั้งละ 1 บอร์ด เมื่อมีการส่งข้อมูลมาพร้อมกันก็จะมีปัญหาในด้านสัญญาณ ชนกัน จะเกิดการซ้อนทับของสัญญาณแล้วทำให้ตัวเลขเปลี่ยนไปทั้งชุดมีผลทำให้คอมพิวเตอร์ไม่สามารถรับข้อมูลได้ทั้งหมด เนื่องจากตัวเลขที่เกิดจากการซ้อนทับกันอยู่นอกตาราง ASCII

การทำงานจะเริ่มต้นที่ มีการติดต่อที่บอร์ดควบคุม โดยการใช้บัตรประจำตัวของผู้ใช้รูดกับบาร์โค้ดรีดเดอร์แล้วกด Password บอร์ดควบคุมจะส่งข้อมูลไปเปรียบเทียบกับฐานข้อมูลของผู้ใช้ที่คอมพิวเตอร์ตรวจสอบความถูกต้อง ถ้า Password ถูกต้อง คอมพิวเตอร์จะสั่งให้บอร์ดควบคุมเปิดประตูและ ไฟเขียวจะติดค้างแต่ถ้า ข้อมูลไม่ถูกต้องไฟเขียวจะกระพริบประตูไม่เปิด สถานะการเข้าออกประตูจะแสดงที่มอนิเตอร์ และจะมีการบันทึกข้อมูลของผู้ใช้ไว้ เช่น ชื่อ, รหัส, และเวลาเข้า-ออก ในขั้นตอนการออกผู้ใช้จะใช้แค่บัตรรูดแล้วประตูจะเปิดออก แล้วผู้ใช้ต้องเดินผ่าน Sensor ระบบถึงจะบันทึกเวลาออก โดยการทำงานนั้น สามารถควบคุมการเข้า-ออก ห้องที่ติดอุปกรณ์ไว้ที่ประตู และเป็นการควบคุมระยะไกล สายสัญญาณที่ใช้สามารถใช้ความยาวได้เกินพิสัย ของ RS-232 (15-30 เมตร) เนื่องจากว่าการใช้การสื่อสารในมาตรฐาน RS-422 ทำให้สามารถส่งข้อมูลได้สูงสุดถึง 1000 เมตร แต่ในการทดลองนี้ได้ใช้ความยาวสายในช่วงความยาว ประมาณ 100 เมตร

ในการสั่งงานให้จ่ายกระแสไปที่ โซลินอยด์บางครั้งบอร์ดควบคุมไม่ทำตามที่สั่ง (ที่บอร์ดมีไฟเขียวติดแล้วแต่ โซลินอยด์ไม่เปิด) ต้องทำการรูดบัตรและทำซ้ำขั้นตอนเดิมอีกครั้ง จึงจะทำงาน จึงเป็นข้อผิดพลาดที่พบในการทดลองและจะเกิดขึ้นในการนำไปใช้งานจริง

5.2 ปัญหาและการแก้ไข

โครงงานนี้มีข้อบกพร่องหลายประการ ผู้จัดทำมีความประสงค์จะเสนอแนะแนวทางในการแก้ไขและได้พบความบกพร่องบางประการหากได้รับการปรับปรุงให้ดีขึ้นในขั้นต่อไป

1. แผ่นวงจรพิมพ์ในโครงงานนี้มีการออกแบบแก้ไขเปลี่ยนแปลง เพิ่มเติมส่วนที่มีการเดินสายและมีการตัดบางเส้น ยังไม่ได้ปรับปรุงแก้ไข

2. สายสัญญาณที่ใช้ยังเป็นสายที่ยังไม่เหมาะสม ทำให้เกิดสัญญาณรบกวน หรือมีปัญหาเมื่อใช้สายความยาวที่ต่างกันจะทำให้เกิดความไม่สมดุลของความต้านทาน ทำให้ข้อมูลที่ส่งไปเป็นขยะได้ แนวทางในการแก้ไขคือ เลือกใช้สายประเภท STP คือเป็นสายคู่ตีเกรียว ที่มีการชิลด์เพื่อลดสัญญาณรบกวน หรือเพิ่มเติมวงจรประเภทลดสัญญาณรบกวน เช่น ดิฟ-แอมป์

3. เมื่อส่งสัญญาณพร้อมกันจะมีปัญหาในด้านสัญญาณ ชนกัน จะเกิดการซ้อนทับของสัญญาณแล้วทำให้ตัวเลขเปลี่ยนไปทั้งชุดมีผลทำให้คอมพิวเตอร์ไม่สามารถรับข้อมูลได้ทั้งหมด เนื่องจากตัวเลขที่เกิดจากการซ้อนทับกันอยู่นอกตาราง ASCII

4. ระบบควรจะมีการต่อกราวด์

เอกสารอ้างอิง

[1] ฉันทวุฒิ พืชผล, พิชิต สันติกุลานนท์. **คู่มือเรียน Visual Basic 6 .** พิมพ์ครั้งที่ 3 . กรุงเทพฯ : บริษัทโปรวิชัน

[2] ศุภชัย สมพานิช. **Database Programming ด้วย Visual Basics ฉบับมืออาชีพ.** พิมพ์ครั้งที่ 2 กรุงเทพฯ: สำนักพิมพ์ อินโฟเพรส. 2543

[3] วรพจน์ กรแก้ววัฒนกุล , ชัยวัฒน์ ลิ้มวิไล. **MCS 51 Flash Microcontroller ฉบับ AT89C5x ของ Atmel.** พิมพ์ครั้งที่ 1. กรุงเทพฯ : บริษัท อินโนเวทีพเอ็กเพอริเมนท์ จำกัด.

[4] กฤษฎา ใจเย็น. "เครื่องแปลงพอร์ทอนุกรม RS232 ให้เป็น RS422/485" วารสารเซมิคอนดัคเตอร์อิเล็คทรอนิคส์ 168 (กุมภาพันธ์ 2540) 40-41

[5] สุนทร วิฑูรสุรพจน์ "การใช้งานไมโครคอนโทรลเลอร์ตระกูล 8051" กรุงเทพฯ: บริษัท เอช.เอ็น.กรุ๊ป จำกัด 2537

ประวัติผู้เขียนโครงงาน

ชื่อ นางสาวกรรณิกา รุจิรวณิชเทพ

ภูมิลำเนา 2/4 หมู่ 4 ต. วัดจันทร์ อ. เมือง จ. พิษณุโลก

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนสุโขทัยวิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

e-mail : eve_aegreat@yahoo.com

ชื่อ นายกำพล จันต๊ะคาด

ภูมิลำเนา 98 หมู่ 5 ต.สบบง กิ่งอำเภอ ภูซาง จ.พะเยา

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนภูซางวิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

e-mail : b69@catcha.com