

ระบบควบคุมการเข้าออกด้วยลายนิ้วมือ
Access Control System Using Fingerprint



เลขหมู่.....
เลขทะเบียน...**61429**
วัน,เดือน,ปี...**17 ก.ค. 2549**

b. **1155๑1๒๑**
i.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมระบบควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2547

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมการเข้าออกด้วยลายนิ้วมือ

Access Control System Using Fingerprint

ผู้จัดทำ

1. นายชัชชัย วณิชกุลพิทักษ์
2. นายณัฐพล สิงห์ทิพย์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมการเข้าออกด้วยลายนิ้วมือ

Access Control System Using Fingerprint

นายชัชชัย วนิชกุลพิทักษ์ 45015315

นายณัฐพล สิงห์ทิพย์ 45015320

อาจารย์ที่ปรึกษา

อาจารย์ เกียรติวรรณ ทรงสตัย

ปีการศึกษา 2547

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการพัฒนาระบบควบคุมการเข้าออกด้วยลายนิ้วมือ โดยการประยุกต์ไมโครคอนโทรลเลอร์ตระกูล เอ็มซีเอส 51 เป็นหน่วยประมวลผลกลางระบบที่ออกแบบขึ้นมาจะประกอบด้วยหน่วยการทำงานย่อยๆ 3 หน่วย ได้แก่ เซิร์ฟเวอร์โมดูล, เทอร์มินอลโมดูล และ หน่วยจ่ายตัดกำลังไฟฟ้า เซิร์ฟเวอร์โมดูลกับเทอร์มินอลโมดูลจะเชื่อมต่อระหว่างกันด้วยโครงข่ายสื่อสารมาตรฐานอาร์เอส 485 ซึ่งมีสามารถขยายขีดความสามารถในการติดต่อ เซิร์ฟเวอร์ 1 ตัวต่อเทอร์มินอลได้สูงสุด 32 ตัว การติดต่อสื่อสารเป็นแบบอนุกรมเลือกใช้โปรโตคอลรูปแบบ เอชดีแอลซีเป็นแนวทางในการออกแบบเทอร์มินอลโมดูลแต่ละหน่วยจะทำหน้าที่อ่านค่าจากลายนิ้วมือเพื่อนำค่าไปประมวลและจัดการได้โดยเซิร์ฟเวอร์ ส่วนหน่วยจ่ายตัดกำลังไฟฟ้าจะเป็นตัวกลไกที่ทำหน้าที่เปิดปิดการใช้งานและเหมาะสมในการติดตั้ง

ABSTRACT

This thesis presents the development of the Access Control system using fingerprint. The application of microcontroller MCS-51 as the central processor each components is constructed. The designed system composes of three units, namely, server module, terminal mode, and power supply unit. Server module and terminal mode are connected together by RS-485 network that can extend the ability to maximum reach up to 32 terminals. The serial communication in specific system is based on HDLC protocol design. The terminal module has to read from a fingerprint in order to be managed by the server module. For the power supply unit has the functions that like ON/OFF the controlled objects. It is designed to suitable to install.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้าที่
บทคัดย่อ	I
สารบัญ	II
สารบัญภาพ	IV
สารบัญตาราง	VI
บทที่ 1 บทนำ	1
1.1 กล่าวนำ	1
1.2 วัตถุประสงค์ในการออกแบบสร้างและพัฒนาระบบ	3
1.3 หลักการที่ใช้ในการออกแบบ	3
1.4 รายละเอียดของรายงานนี้จะแบ่งส่วนที่ดำเนินการ	3
บทที่ 2 หลักการและทฤษฎี	5
2.1 การอ่านลายนิ้วมือ	5
2.2 พื้นฐานการรับส่งข้อมูล	7
2.3 รูปแบบของการรับส่งข้อมูลแบบอนุกรม	8
2.4 MCS-51 กับ การรับส่งข้อมูลแบบอนุกรม	11
2.5 หน่วยความจำข้อมูล	22
2.6 การอ่านและเขียนข้อมูลกับหน่วยความจำ	23
2.7 การขยายพอร์ตอินพุตเอาต์ด้วยอุปกรณ์ระบบบัส I^2C	26
2.8 การใช้งาน ไอซีสร้างฐานเวลาจริงหรือรีล ไทม์คล็อก(RTC)	33
2.9 มาตรฐานการสื่อสารข้อมูลอนุกรมแบบอาร์เอส 485	40
บทที่ 3 โครงสร้าง และการออกแบบระบบ	42
3.1 ส่วนประมวลผลกลางและจัดเก็บข้อมูล	42
3.2 ส่วนควบคุมการปิด-เปิด ประตูและเครื่องคอมพิวเตอร์หรือหน่วยตัด-จ่ายกำลังไฟฟ้า	43
บทที่ 4 การทดลองและผลการทดลอง	54
4.1 การทดลองส่วนของภาคจ่ายไฟ	54
4.2 การทดลองส่วนของวงจรบอร์ดหลัก	54
4.3 การทดลองส่วนของวงจรส่วนตัดจ่ายกำลังงานไฟฟ้า	55
4.4 การทดลองในส่วนของภาคแสดงผล	56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง	หน้าที่
บทที่ 5 บทสรุป วิจัยรณั ปัญหาที่พบ และการพัฒนา	58
ภาคผนวก ก. การใช้งานเครื่องสแกนลายนิ้วมือ	
ภาคผนวก ข. ชุดคำสั่งสำหรับส่วนสแกนลายนิ้วมือ	
เอกสารอ้างอิง	
กิตติกรรมประกาศ	



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่	หน้าที่
1.1 แผนผังระบบควบคุมการเข้าออกและปิด-เปิดเครื่องคอมพิวเตอร์	2
2.1 สายนิ้วมือ	6
2.2 การรับส่งข้อมูลแบบขนาน	7
2.3 การส่งข้อมูลแบบซิงโครนัส	8
2.4 การส่งข้อมูลแบบอนุกรม	9
2.5 บิตต่างๆของข้อมูลที่ส่งแบบอนุกรม	9
2.6 การรับส่งข้อมูลระหว่างรีจิสเตอร์กับบัสภายใน	11
2.7 ไคอะแกรมเวลาการส่งข้อมูล	14
2.8 ไคอะแกรมเวลาการรับข้อมูล	15
2.9 การส่งข้อมูลออกโดยใช้รีพรีจิสเตอร์ช่วย	15
2.10 การรับส่งข้อมูลในโหมด 1	16
2.11 แสดงการกำหนด Baud Rate ในโหมดต่างๆ	20
2.12 แสดงการเก็บข้อมูลแต่ละบิตของสเตติกแรม	22
2.13 แสดง โครงสร้างภายในของสเตติกแรม	23
2.14 แผนภาพของหน่วยความจำต่างๆ	23
2.15 แสดงกลุ่มสัญญาณที่ใช้ในการติดต่อกับหน่วยความจำ	24
2.16 แสดงสัญญาณในการอ่านข้อมูล	25
2.17 แสดงสัญญาณในการเขียนข้อมูล	25
2.18 ฟังแสดงการเชื่อมต่อของอุปกรณ์ต่างๆ บนระบบบัส IC	26
2.19 แสดงวงจรเอาต์พุตของอุปกรณ์ในระบบ I^2C	27
2.20 การต่อตัวต้านทานพูลอัปบนสายสัญญาณในระบบบัส I^2C	28
2.21 การต่อตัวต้านทาน R_s เพื่อลดสัญญาณรบกวนขนาดใหญ่ที่อาจเข้ามาในบัส	28
2.22 ไคอะแกรมเวลาแสดงสถานะต่างๆในบัส I^2C	30
2.23 รูปแบบของข้อมูลกำหนดแอดเดรสที่ใช้ในการอ้างถึงแบบ 7 บิต	31
2.24 การจัดขาของไอซี	33
2.25 โครงสร้างภายในของไอซีรีลไทม์คล็อกเบอร์ DS1307	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ(ต่อ)

รูปที่	หน้าที่
2.26 จัดสรรหน่วยความจำแรมภายใน DS1307	36
2.27 รูปแบบของข้อมูลสำหรับติดต่อ DS1307 ในโหมดการเขียนและการอ่านข้อมูล	37
2.28 แสดงการเชื่อมต่อ MCS-51 กับไอซีรีลไทม์คัลเลอร์ DS1307	39
3.1 MAIN โปรแกรม	44
3.2 โปรแกรมการลงทะเบียน	45
3.3 โปรแกรมการเข้าใช้งาน	46
3.4 โปรแกรมการเลิกใช้งาน	47
3.5 โปรแกรมการ Up-load ข้อมูลไปคอมพิวเตอร์	48
3.6 โปรแกรมการทำงานส่วนตัดจ่ายกำลังไฟฟ้า	49
3.7 วงจรส่วนบอร์ดหลัก	50
3.8 วงจรส่วนตัดจ่ายกำลังไฟฟ้า	51
3.9 ลายวงจรบอร์ดหลัก	52
3.10 ลายวงจรของส่วนตัดจ่ายกำลังไฟฟ้า	53
4.1 วงจรส่วนของบอร์ดหลัก	54
4.2 การแสดงผลทางจอLCD	55
4.3 วงจรส่วนตัด-จ่ายกำลังงาน	55
4.4 การตัด-จ่ายกำลังงาน	56
4.5 การแสดงผลทางจอคอมพิวเตอร์	57
4.6 การแสดงผลทางจอคอมพิวเตอร์	57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่		หน้าที่
2.1	แสดงโหมดต่างๆ ของการรับส่งแบบอนุกรม	12
2.2	บิตต่างๆ ของรีจิสเตอร์ SCON	13
2.3	แสดงความถี่สัญญาณพิกาที่ใช้กำหนด Baud Rate ค่าต่างๆ	21



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 กล่าวนำ

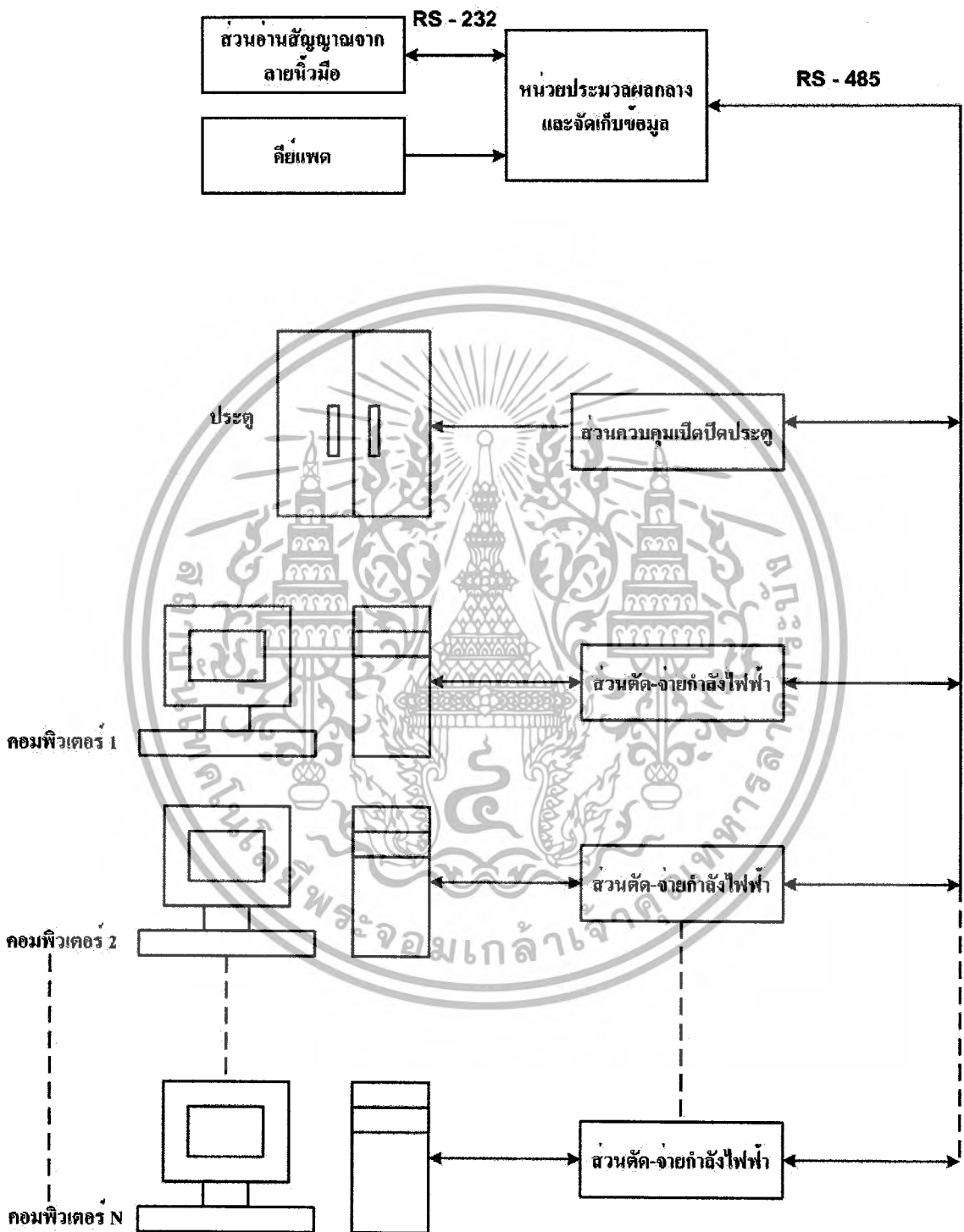
ปัจจุบันคอมพิวเตอร์มีบทบาทอย่างมากในชีวิตประจำวันไม่เพียงแต่การใช้งานเพียงเครื่องเดียวตามลำพังแต่ยังได้มีการเชื่อมต่อโครงข่ายของคอมพิวเตอร์ขึ้นเพื่อเพิ่มประสิทธิภาพในการใช้งาน ดังนั้นจึงมีความจำเป็นอย่างยิ่งที่จะต้องมียุคที่ระบบที่สามารถควบคุมตรวจสอบประมวลผลและรักษาความปลอดภัยของข้อมูลบนคอมพิวเตอร์หรืออนุญาตเฉพาะผู้ที่มีสิทธิ์ในการใช้คอมพิวเตอร์เครื่องนั้นๆ เท่านั้นซึ่งจะทำให้เกิดความสะดวกและปลอดภัยมากยิ่งขึ้นในที่นี้ระบบควบคุมการเข้าออกด้วยลายนิ้วมือจึงได้ถูกนำมาใช้และพัฒนาให้สามารถควบคุมการใช้งานห้องคอมพิวเตอร์และเครื่องคอมพิวเตอร์ได้

โดยในโครงการนี้ได้เลือกนำเสนอระบบควบคุมการเข้าออกด้วยลายนิ้วมือมาประยุกต์ใช้งานร่วมกับการกรงครหัสส่วนบุคคล ซึ่งเป็นระบบที่ออกแบบไว้โดยใช้ไมโครคอนโทรลเลอร์ในการควบคุมผ่านมาตรฐานการติดต่อแบบอนุกรม RS-232 และ RS-485 โดยได้นำเอาข้อมูลจากการกรงครหัสมาประยุกต์ใช้ร่วมกับข้อมูลที่ได้จากตัวบุคคล (Human DATA) ซึ่งได้จากการสแกนลายนิ้วมือนั่นเอง

โดยจะพิจารณา จากกรงครหัสสร่วมกับอุปกรณ์สแกนลายนิ้วมือมาควบคุมการใช้งานเปิดปิดเครื่องคอมพิวเตอร์ที่อยู่ในอยู่ในโครงข่ายของเราได้โดยในการออกแบบระบบนั้นจะประกอบไปด้วย ส่วนใหญ่ๆ 4 ส่วนด้วยกันคือ

1. ส่วนตรวจสอบและแปลงสัญญาณจากการกรงครหัส
2. ส่วนตรวจสอบและแปลงสัญญาณจากลายนิ้วมือ
3. ส่วนควบคุมการเปิดปิดประตูและเครื่องคอมพิวเตอร์หรือหน่วยจ่ายต่อกำลังไฟฟ้า
4. ส่วนประมวลผลและจัดเก็บข้อมูล

โดยในการทำงานของระบบนั้นทั้ง 4 ส่วนจะถูกเชื่อมต่อถึงกันโดยใช้มาตรฐานการติดต่อแบบอนุกรม RS-232 และ RS-485



รูปที่ 1.1 แผนผังระบบควบคุมการเข้าออกและปิด-เปิดเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ในการออกแบบสร้างและพัฒนาระบบ

- 1.สามารถออกแบบระบบควบคุมการเข้าออกโดยใช้ไมโครคอนโทรลเลอร์ ตระกูล MCS-51 เป็นหน่วยประมวลผล
- 2.สามารถใช้ติดต่อสื่อสารแบบอนุกรมผ่านโครงข่าย RS-485 แบบฮาร์ฟดิพล็กซ์ (Half – Duplex) ระหว่างเทอร์มินอล โมดูล กับ ส่วนเซิร์ฟเวอร์ โมดูล ซึ่งใช้คู่สายเพียง 2 เส้นในการรับส่งสัญญาณ
- 3.สามารถนำมามาตรฐานสื่อสาร เอชดีแอลซี (HDLC) มาประยุกต์ออกแบบเพื่อสร้างระบบควบคุมการเข้าออกนี้
- 4.สามารถนำเครื่องสแกนลายนิ้วมือประยุกต์ใช้การเก็บข้อมูลและอ่านข้อมูลได้
- 5.สามารถพัฒนาหรือแก้ไขโปรแกรมการติดต่อสื่อสารข้อมูล การขอใช้บริการ และการควบคุมเข้าออก ระหว่างเซิร์ฟเวอร์ (Server) กับ เทอร์มินอล (Terminal) หรือเซิร์ฟเวอร์ กับ คอมพิวเตอร์ให้มีประสิทธิภาพ
- 6.สามารถออกแบบตัวเซิร์ฟเวอร์, เทอร์มินอล และ หน่วยจ่ายกำลังไฟฟ้าให้มีประสิทธิภาพ ใช้งานได้ตามจุดประสงค์อย่างเหมาะสม

1.3 หลักการที่ใช้ในการออกแบบ

ระบบควบคุมการเข้าออกโดยใช้เครื่องสแกนลายนิ้วมือจะใช้หลักการของ

1. การอ่านข้อมูลจากเครื่องสแกนลายนิ้วมือ
2. การติดต่อสื่อสารอนุกรมของ ไมโครคอนโทรลเลอร์ เอ็มซีเอส51 แบบ มัลติไมโคร โปรเซสเซอร์ (Multi-microprocessor communication)
3. การติดต่อสื่อสารแบบอนุกรมตามมาตรฐาน เอชดีแอลซี โปรโตคอล
4. การใช้โครงข่าย อาร์เอส 485 (RS-485) และ อาร์เอส 232 (RS-232)
5. การใช้ภาษาแอสเซมบลีในการ โปรแกรมการทำงาน

1.4 รายละเอียดของรายงานนี้จะแบ่งส่วนที่ดำเนินการ

- บทที่1 เป็นบทกล่าวนำ บอกวัตถุประสงค์และหลักการออกแบบ
- บทที่2 กล่าวถึงทฤษฎีและหลักการต่างๆ ได้แก่ รูปแบบรวมทั้งหลักการพื้นฐานติดต่อสื่อสาร รวมทั้งการควบคุมส่วนอ่านลายนิ้วมือ การติดต่อสื่อสารเบื้องต้นและโครงสร้างการติดต่อสื่อสารแบบอนุกรม

- บทที่3 กล่าวถึงวิธีการสร้างและออกแบบ รวมทั้งโครงสร้างของเซิร์ฟเวอร์ และเทอร์มินอล
- บทที่4 เป็นการทดลองการทำงานของระบบในส่วนต่างๆ และผลการทดลอง
- บทที่5 บทสรุป เป็นการสรุปผลและการวิจารณ์ ระบบที่ออกแบบและสร้างขึ้นว่ามีข้อดีหรือข้อเสียอย่างไร รวมทั้งแนวทางในการแก้ไขและพัฒนาต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

หลักการและทฤษฎี

2.1 การอ่านลายนิ้วมือ

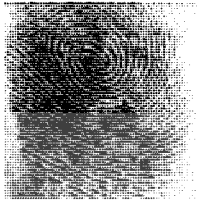
ลายนิ้วมือของแต่ละคน เริ่มปรากฏขึ้นตั้งแต่เป็นตัวอ่อนอายุ 3 ถึง 4 เดือนในครรภ์มารดา ซึ่งเป็นผิวหนังส่วนที่มีร่อง (Furrow) และมีสัน (Ridge) เอาไว้ใช้สำหรับอำนวยความสะดวกในการหยิบจับสิ่งของ สันและร่องที่ปรากฏนี้มีคุณลักษณะที่สำคัญสองประการ คือ ไม่มีการเปลี่ยนแปลงรูปแบบตามกาลเวลา (แต่อาจเปลี่ยนขนาดได้) และ การมีรูปแบบเฉพาะในแต่ละคน

ลายนิ้วมือไม่เปลี่ยนแปลงรูปแบบ (Permanence) ตั้งแต่แรกเกิด จนกระทั่งวันที่เราตาย แต่อาจเปลี่ยนแปลงขนาดได้ตามขนาดร่างกาย เหมือนกับการที่เราวาดรูปไว้บนลูกโป่ง ซึ่งไม่ว่าลูกโป่งจะเล็กหรือถูกเป่าให้พองใหญ่อย่างไร ก็ยังเป็นรูปเดิมเพียงแต่มีขนาดใหญ่ขึ้นเท่านั้น

การที่ลายนิ้วมือมีรูปแบบเฉพาะในแต่ละคน (Individuality) เป็นคุณสมบัติที่สำคัญอีกอย่างหนึ่งของลายนิ้วมือนั้น ตั้งแต่เริ่มมีการใช้เก็บและเปรียบเทียบลายนิ้วมือโดยใช้วิธีสมัยใหม่ ซึ่งมีมาร้อยกว่าปีแล้ว ยังไม่มีการตรวจพบว่ามีกรณีเหมือนกันของลายนิ้วมือ อีกทั้งถ้าจะอธิบายด้วยหลักการทางคณิตศาสตร์ และวิทยาศาสตร์ ก็มีการศึกษาของ Sir Francis Galton (1892) ซึ่งได้ประมาณไว้ว่า โอกาสที่คนสองคนจะมีลายนิ้วมือเหมือนกันนั้นมีความน่าจะเป็นอยู่ที่ $1/64,000,000,000$ ซึ่งเป็นการประเมินค่าโดยใช้การแบ่งรายละเอียดรูปแบบของลายนิ้วมือออกเป็นส่วนๆ และหาความน่าจะเป็นของการซ้ำกันของแต่ละส่วนนั้น แล้วนำความน่าจะเป็นของแต่ละส่วนมาคูณกันเพื่อหาความน่าจะเป็นทั้งหมด ท่าน Sir Francis Galton นี้เป็นผู้ที่เริ่มทำการวิจัยอย่างจริงจังกับลายนิ้วมือ และถือว่าเป็นบุคคลแรกที่ศึกษาถึงการใช้นิ้วมือในการระบุตัวบุคคล เป็นบุคคลแรกที่ทำการพิสูจน์ว่าลายนิ้วมือของแต่ละคนมีลักษณะเฉพาะ (Individuality) และ ไม่มีการเปลี่ยนแปลงรูปแบบ (Permanence) อีกทั้งยังเป็นผู้ที่กำหนดและแบ่งแยกประเภทของรูปแบบลายนิ้วมือที่ใช้กันอยู่จนถึงปัจจุบันนี้

ลายนิ้วมือของแต่ละคนนั้นมีลักษณะเฉพาะมากจนกระทั่งแม้แต่ คู่แฝดแท้ (Identical Twin) ก็ยังมีลายนิ้วมือที่แตกต่างกัน (แต่มีรูปแบบ DNA เหมือนกัน) อย่างไรก็ตามรูปแบบของลายนิ้วมือนั้นมีลักษณะความคล้ายกันของคนภายในครอบครัว หรือพูดได้อีกอย่างหนึ่งว่า รูปแบบของลายนิ้วมือมีการถ่ายทอดกันทางพันธุกรรม ซึ่งรูปแบบของลายนิ้วมือ สามารถแบ่งออกได้เป็นประเภทใหญ่ๆ ได้ 3 ประเภท คือแบบลายก้นหอย (Whorl) ลายมัดหวาย (Loop) และ ลายโค้ง (Arch)

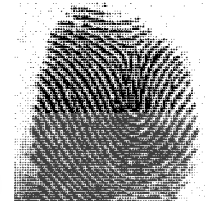
ลายก้นหอย (Whorl)



ลายมัดหวาย (Loop)



ลายโค้ง (Arch)



รูปที่ 2.1 แสดงลายนิ้วมือต่าง ๆ

รูปแบบลายนิ้วมือสามารถแบ่งย่อยให้ละเอียดขึ้นไปได้เป็น ลายมัดหวายเอียงขวา (Right Loop) ลายมัดหวายเอียงซ้าย (Left Loop) , ลายโค้งสูงแบบกระโจม (Tented Arch) เป็นต้น ลายนิ้วมือแบบลายก้นหอย (Whorl) มีประมาณ 30% ลายนิ้วมือแบบลายมัดหวาย (Loop) มีประมาณ 65% และลายนิ้วมือแบบลายโค้ง (Arch) มีประมาณ 5% การแบ่งลายนิ้วมือออกเป็นหลายประเภทนี้เพื่อวัตถุประสงค์ในการเพิ่มความรวดเร็วในการตรวจสอบลายนิ้วมือ แต่ไม่ได้เป็นสิ่งที่ใช้ในการบอกความเหมือน หรือความแตกต่างระหว่างลายนิ้วมือ แต่เป็นการใช้ลักษณะของสัน (Ridge) ของลายนิ้วมือเช่น การสิ้นสุดของสัน (Ridge Ending), สันแบบลายจุด (Dots) , สันที่แตกแขนง (Bifurcations) หรือรูปแบบต่างๆของสันที่เกิดขึ้น เป็นสิ่งที่ใช้ในการเปรียบเทียบลายนิ้วมือ

ในการทำงานของระบบของเรานั้นส่วนตรวจสอบและแปลงสัญญาณจากลายนิ้วมือมีความสำคัญมาก เพราะเป็นส่วนที่ใช้ในด้านของการรักษาความปลอดภัย เนื่องจากสัญญาณที่ได้จากส่วนนี้ จะเป็นสัญญาณเฉพาะของแต่ละบุคคล ไม่สามารถลอกเลียนแบบกันได้ โดยในส่วนนี้เราได้ใช้อุปกรณ์คือ โมดูลสแกนลายนิ้วมือรุ่น TS-FID 335416 EB ซึ่งต้องใช้งานร่วมกับรหัส ซึ่งในที่นี้ได้มาจากการกรอรหัส และในการเชื่อมต่อได้ทำการเชื่อมต่อกับหน่วยประมวลผลกลาง โดยมาตรฐาน RS-232 หรือ RS-485

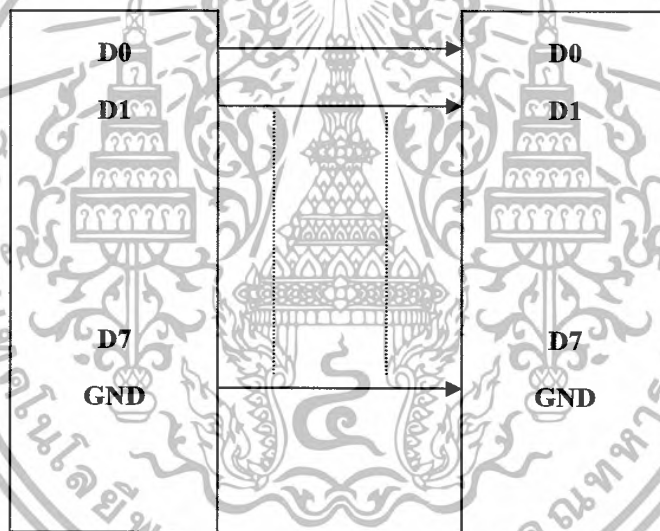
โดยในการตรวจสอบลายนิ้วมือนั้น จะต้องรอรับรหัสจากบัตรแม่เหล็กเมื่อหน่วยประมวลผลกลางได้รับและแปลงสัญญาณจากบัตรแม่เหล็กก็จะส่งรหัสมาให้กับส่วนตรวจสอบและแปลงสัญญาณจากลายนิ้วมือ ซึ่งจะตรวจสอบว่าลายนิ้วมือของผู้ใช้ตรงกับลายนิ้วมือที่มีการลงทะเบียนไว้หรือไม่ จากนั้นก็จะส่งสัญญาณกลับไปให้หน่วยประมวลผลกลางเพื่อทำการวิเคราะห์ต่อไป

2.2 พื้นฐานการรับส่งข้อมูล

การรับส่งข้อมูลในระบบคอมพิวเตอร์โดยทั่วไปจะหมายถึง การรับส่งข้อมูลเป็นจำนวนไบนารี ให้กับอุปกรณ์ที่เกี่ยวข้องกับคอมพิวเตอร์ซึ่งอาจแบ่งประเภทของการรับส่งข้อมูลได้ 2 แบบ

- 1) การรับส่งข้อมูลแบบขนาน (Parallel)
- 2) การรับส่งข้อมูลแบบอนุกรม (Serial)

การรับส่งข้อมูลแบบขนานจะเป็นการรับส่งข้อมูล จำนวน 1 ไบนารี ออกไปทางพอร์ท ในเวลาเดียวกันในระบบคอมพิวเตอร์ 1 ไบนารีจะมีจำนวน 8 บิต คือ D0 – D7 ถ้ามีการส่งข้อมูลแบบขนานจะใช้สายสัญญาณอย่างน้อย 9 เส้น คือ สาย Data 8 เส้น และสายกราวด์ 1 เส้นดังรูป



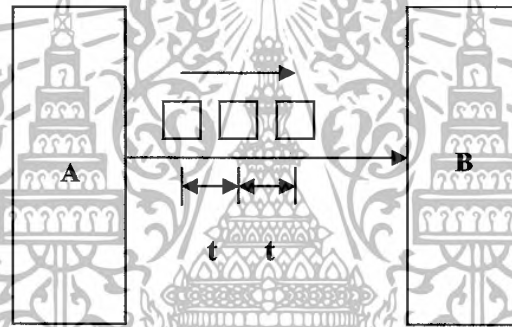
รูปที่ 2.2 การรับส่งข้อมูลแบบขนาน

การรับส่งข้อมูลแบบอนุกรม คือ การรับส่งข้อมูลที่ละบิต จนครบ 1 ไบนารี ถ้าต้องการส่งข้อมูล 1 ไบนารี คือ D0-D7 อาจจะส่งบิต D0 ออกไปก่อนแล้วตามด้วย D1 ไปเรื่อยๆจนถึง D7 การส่งข้อมูลทั้งสองแบบมีข้อดีข้อเสียแตกต่างกันคือ การส่งข้อมูลแบบขนานสามารถส่งข้อมูลได้เร็ว คือส่งทีละบิตจะได้ข้อมูลครบ 1 ไบนารี แต่ถ้าต้องส่งเป็นระยะไกลๆ จะสิ้นเปลืองสายสัญญาณมาก ถ้าเป็นการส่งแบบอนุกรม เมื่อต้องการส่งข้อมูลเป็นระยะไกลๆ จะช่วยประหยัดสายเนื่องจากจะใช้สายอย่างน้อยเพียง 2

เส้น คือ สายสัญญาณกับสายกราวด์ แต่การรับส่งข้อมูลจะใช้เวลานานเนื่องจากการส่งทีละบิต ในส่วนนี้จะกล่าวถึงพื้นฐานการรับส่งข้อมูลแบบอนุกรม โดยจะเน้นที่ตัว MCS-51 เป็นสำคัญ

การรับส่งข้อมูลแบบซิงโครนัส (Synchronous Input / Output)

การรับส่งข้อมูลแบบนี้ไม่ว่าจะเป็นการส่งแบบอนุกรมหรือแบบขนาน ข้อมูลแต่ละไบต์ที่ถูกส่งออกไปจะมีช่วงเวลาห่างกันแน่นอน เช่น การส่งข้อมูลจาก A ไป B ดังรูปที่ 2.3 Data 1 จะห่างจาก Data2 เป็นเวลา t และ Data 3 จะห่าง Data 2 เป็นเวลา t เช่นกัน ระบบนี้เหมาะกับงานที่ไม่มีความยุ่งยากมาก



รูปที่ 2.3 การส่งข้อมูลแบบซิงโครนัส

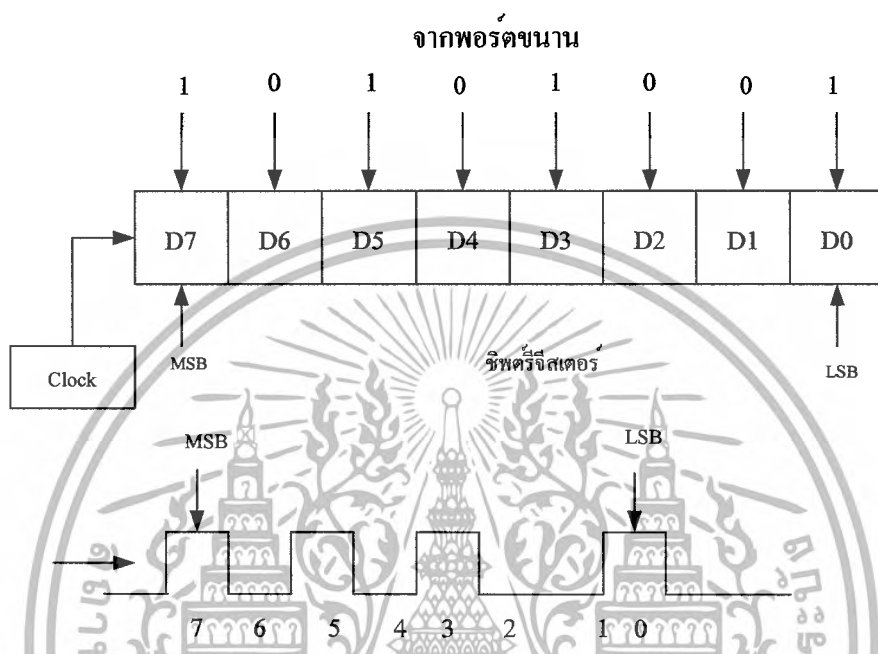
การรับส่งข้อมูลแบบอะซิงโครนัส (Asynchronous Input / Output)

การรับส่งข้อมูลแบบนี้ ข้อมูลที่ส่งออกไปจะไม่มีเวลาที่แน่นอน ซึ่งจะขึ้นอยู่กับความพร้อมของผู้ส่งและผู้รับ โดยจะมีสายสัญญาณตรวจสอบความพร้อมของระบบทั้งสองว่าพร้อมที่จะติดต่อกันหรือไม่ โดยสัญญาณที่เพิ่มขึ้นมาจากระบบแบบซิงโครนัส เรียกว่า สายสเตตัส (Status Line)

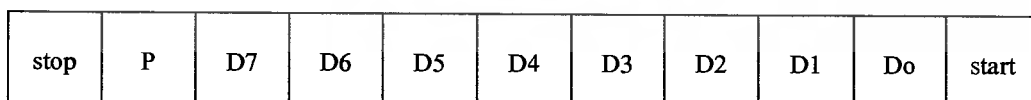
2.3 รูปแบบของการรับส่งข้อมูลแบบอนุกรม

เมื่อไมโครคอมพิวเตอร์ต้องการจะรับส่งข้อมูลแบบอนุกรม ตัวไมโครคอมพิวเตอร์จะส่งข้อมูลออกไปทางพอร์ทซึ่งเป็นพอร์ทแบบขนานก่อน จากนั้นจะมีอุปกรณ์มาต่อที่พอร์ทเพื่อแปลงข้อมูลแบบขนานให้เป็นแบบอนุกรมอีกทีหนึ่ง (Parallel-to-Serial Conversion) ตัวแปลงข้อมูลนี้อาจจะพิจารณาได้ง่ายๆ ว่าเป็น Shift Register ดังรูปที่ 2.4 เมื่อข้อมูลที่จะส่งอยู่ใน Shift Register แล้วตัวสัญญาณนาฬิกา

จะเป็นตัวกระตุ้นให้ข้อมูลบิตต่ำออกไปในเวลาแรก จากนั้นจะส่งบิตต่อไปตามออกมาจากรูปที่ 2.4 จะเป็นการส่งข้อมูล A9H ออกไป



สำหรับตัวรับข้อมูลแบบอนุกรมนั้นเมื่อตัวรับข้อมูลทำงานจะเป็นการรับเข้ามาใน Shift Register แล้วส่งข้อมูลให้ไมโครคอมพิวเตอร์แบบขนานอีกทีหนึ่ง (Serial-to-Parallel) ระบบคอมพิวเตอร์ในปัจจุบันจะมีตัวแปลง Parallel-to-Serial และ Serial-to-Parallel อยู่ในชิพไอซี เรียกว่า Universal Asynchronous Receiver Transmitter (UART) การส่งข้อมูลแบบอนุกรมนั้นจะต้องมีการเพิ่มเติมข้อมูลบางอย่างเข้าไปเพื่อให้การรับส่งข้อมูลสามารถทำงานได้ถูกต้องมากขึ้นโดยมีการเติมค่าบิตต่างๆลงไปตามรูปที่ 2.5



รูปที่ 2.5 บิตต่างๆของข้อมูลที่ส่งแบบอนุกรม

ถ้ามีการส่งข้อมูลแบบ 8 บิตจะต้องส่งบิตแรกออกไปก่อน เรียกว่า บิตเริ่มต้น(Start Bit)ถ้ามีการส่งข้อมูลหลายๆไบต์ออกมา บิตนี้จะเป็นตัวบอกว่ามีข้อมูลใหม่มาแล้ว โดยทั่วไปบิตเริ่มต้นมักมีระดับ

ลอจิกเป็น"0" ต่อจากบิตเริ่มต้นจะเป็นข้อมูลบิต D0 ถึง D7 จากนั้นจะตามด้วยบิตตรวจสอบความถูกต้อง (Parity Bit) ถ้าข้อมูล 8 บิตที่ส่งออกมา จำนวนของบิตที่มีค่าเป็น "1" เป็นจำนวนคู่ บิตนี้จะมีค่าเป็น 0 แต่ถ้าจำนวนของบิตที่มีค่าเป็น "1" เป็นจำนวนคี่ บิตนี้จะมีค่าเป็น 1 จากนั้นข้อมูลที่จะส่งออกไปจะตามด้วยบิตสิ้นสุดข้อมูล (Stop Bit) เพื่อเป็นการบอกว่าข้อมูลที่ส่งมา 8 บิตนั้นหมดแล้ว ตัวบิต Stop อาจมีจำนวนมากกว่า 1 บิตก็ได้ เช่น $1\frac{1}{2}$ บิต, 2 บิต

บิตตรวจสอบความถูกต้องหรือบิตพริตี้จะมีสองลักษณะคือพริตี้คู่ (Even Parity) และพริตี้คี่ (Odd Parity) ซึ่งเราสามารถเลือกได้ ถ้าหากระบุเป็นพริตี้คู่หมายความว่าข้อมูลที่ส่งไปหรือไบต์นั้นมีจำนวนลอจิก"1" รวมกับบิตพริตี้เป็นจำนวนคู่บิตส่วนถ้าระบุเป็นพริตี้คี่ก็หมายความว่า จะมีจำนวนลอจิก "1" ของ ไบต์ข้อมูลที่ส่งไปพร้อมกับค่าบิตพริตี้เป็นจำนวนคี่บิต ตัวอย่างเช่นถ้าเราส่งไบต์ข้อมูลที่มีค่าเป็น B2H หรือ 10110010B ออกไป ถ้าระบบระบุว่าเป็นพริตี้คี่แล้วค่าของบิตพริตี้จะต้องเป็นลอจิก "1" ซึ่งในไบต์ข้อมูลจะมีลอจิก"1" จำนวน 4 บิต และบิตพริตี้เป็น"1" ดังนั้นมีลอจิก "1" ทั้งหมด 5 บิต ซึ่งเป็นคี่บิตแต่ถ้าระบบระบุว่าเป็นระบบพริตี้คู่แล้วค่าบิตพริตี้จะต้องเป็นลอจิก"0" เมื่อทางภาครับได้รับข้อมูลเข้ามาก็ตรวจสอบว่าข้อมูลทั้งหมดมีลอจิก "1" เป็นคู่หรือคี่บิต ตรงกับการทำงานของระบบซึ่งออกแบบไว้หรือไม่ ซึ่งจะเป็นการตรวจสอบความถูกต้องได้ระดับหนึ่ง

การสื่อสารข้อมูลแบบอนุกรมนี้ การกำหนดอัตราเร็วในการรับส่งข้อมูลจะบอกเป็นบิตต่อวินาที(Bit Per Second:BPS) ที่เรียกว่าอัตราบอดหรือบอดเรต(Baud Rate) โดยค่ามาตรฐานที่ใช้กันมีหลายค่าได้แก่ 100, 150, 300, 600, 1200, 2400, 4800, 9600 และ 19200 บิตต่อวินาที

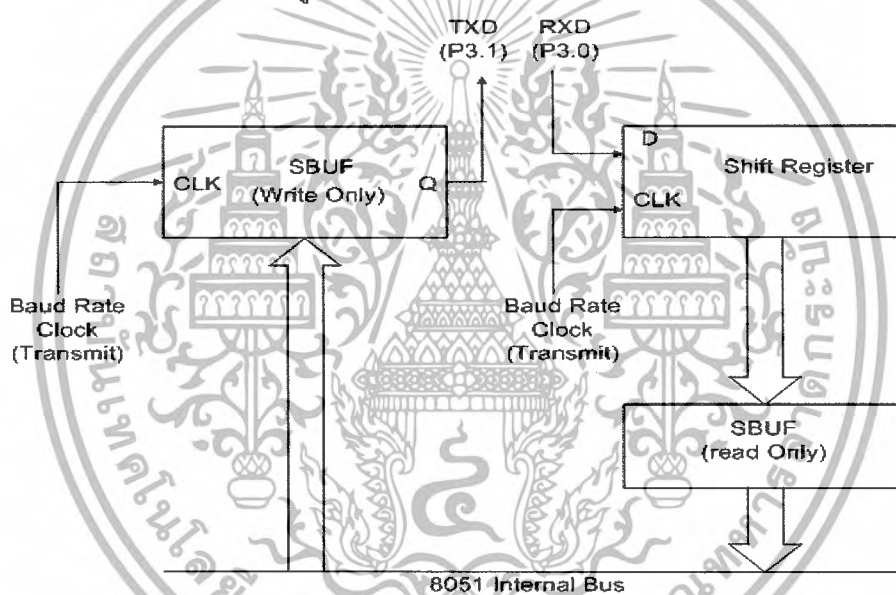
ใน MCS-51 นั้นจะมีพอร์ตสำหรับรับส่งข้อมูลแบบอนุกรมอยู่ในชิพ โดยเราไม่ต้องต่ออุปกรณ์ภายนอกเพิ่ม และเป็นพอร์ตสื่อสารอนุกรมแบบฟูลดูเพล็กซ์ (Full-duplex) คือสามารถรับส่งข้อมูลได้สองทิศทางในเวลาเดียวกัน การควบคุมการใช้งานของพอร์ตอนุกรมของ MCS-51 นี้สามารถโปรแกรมให้ทำงานรูปแบบต่างๆ ได้ 4 ประเภทหรือ 4 โหมดการทำงาน โดยโปรแกรมผ่านทางรีจิสเตอร์ SCON โดยโหมด 0 จะเป็นการสื่อสารแบบซิงโครนัสหรือแบบเข้าจังหวะเวลา ส่วนโหมด 1, 2, และ 3 เป็นการสื่อสารแบบอะซิงโครนัส

การส่งข้อมูลแบบอนุกรมนั้น ความเร็วของการส่งจะมีค่าเป็น บิตต่อวินาทีเรียกว่า "Baud Rate"

2.4 MCS-51 กับการรับส่งข้อมูลแบบอนุกรม

การรับส่งข้อมูลแบบอนุกรมกับไมโครคอนโทรลเลอร์ MCS-51 นั้น ภายในชิพ MCS-51 จะมี UART อยู่ในตัว ซึ่งเป็นข้อดีของไมโครคอนโทรลเลอร์ ถ้าเป็นไมโครโปรเซสเซอร์ เช่น เบอร์ Z-80 ถ้าต้องการรับส่งข้อมูลแบบอนุกรมจะต้องนำชิพ UART มาประกอบด้วย

พอร์ทอนุกรมของ MCS-51 จะใช้ขา TXD และ RXD ในการรับส่งข้อมูล โดยขาทั้งสองจะอยู่ในพอร์ท 3 คือ P3.1 หรือขา 11 เป็น TXD และ P3.0 หรือขา 10 เป็น RXD พอร์ทอนุกรมของ MCS-51 สามารถทำงานแบบ Full Duplex ได้ คือ สามารถส่งและรับข้อมูลในเวลาเดียวกันได้ โดยในการรับและส่งข้อมูลจะมีบัฟเฟอร์สำหรับเก็บข้อมูลให้ใช้



รูปที่ 2.6 การรับส่งข้อมูลระหว่างรีจิสเตอร์กับบัฟเฟอร์ภายใน

รีจิสเตอร์ที่สำคัญในการรับส่งข้อมูล คือ SBUF และ SCON ซึ่งเป็นรีจิสเตอร์ที่อยู่ใน Special Function Registers โดยรีจิสเตอร์ Serial Post Buffer (SBUF) จะอยู่ในตำแหน่ง 99H ถ้าเขียนข้อมูลไปที่ตำแหน่งนี้จะเป็นการส่งข้อมูลออกทางพอร์ทอนุกรม และถ้าอ่านข้อมูลจากตำแหน่งนี้จะเป็นการรับข้อมูลจากพอร์ทอนุกรม โดยใน SBUF จะประกอบด้วยบัฟเฟอร์ 2 ตัว สำหรับส่งและรับข้อมูล ดังรูปที่

2.6

สำหรับ Serial Port Control Register (SCON) ซึ่งอยู่ที่ตำแหน่ง 98 H จะเป็นรีจิสเตอร์ที่สามารถที่สามารถเข้าถึงข้อมูลระดับบิตได้ รีจิสเตอร์นี้จะทำหน้าที่ควบคุมและบอกสถานะต่างๆ ของการรับส่งข้อมูลแบบอนุกรม

สำหรับความเร็วของการส่งข้อมูล (Baud Rate) สามารถหาได้จากตารางสัญญาณนาฬิกาที่ใช้กับ MCS-51

Serial Port Control Register

MCS-51 มีโหมดการทำงานของพอร์ทอนุกรมหลายโหมด ซึ่งสามารถโปรแกรมโหมดการทำงานได้ โดยเขียนข้อมูลไปยังรีจิสเตอร์ SCON ความหมายของแต่ละบิตแสดงดังตารางที่ 1 และ 2

ตารางที่ 2.1 แสดงโหมดต่างๆ ของการรับส่งแบบอนุกรม

SM0	SM1	MODE	ความหมาย	Baud Rate
0	0	0	Shift Register	เปลี่ยนแปลงไม่ได้(Oscillator Frequency/12)
0	1	1	8-bit UART	สามารถเปลี่ยนแปลงได้โดยการกำหนดจาก Time
1	0	2	9-bit UART	เปลี่ยนแปลงไม่ได้(Oscillator Frequency /32 หรือ /64
1	1	3	9-Bit UART	สามารถเปลี่ยนแปลงได้โดยการกำหนดจากTime

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 บิตต่างๆ ของรีจิสเตอร์ SCON

บิต	ชื่อ	ตำแหน่ง	ความหมาย
SCON.7	SM0	9FH	บิตเลือกโหมดการทำงานบิต 0
SCON.6	SM1	9EH	บิตเลือกโหมดการทำงานบิต 1
SCON.5	SM2	9DH	บิตเลือกโหมดการทำงานบิต 2
SCON.4	REN	9CH	บิตแฟลคกำหนดยอมให้มีการรับข้อมูล
SCON.3	TB8	9BH	ค่าของบิต 9 สำหรับการส่งข้อมูลในโหมด 2 และ 3 สามารถ Set และ Clear ได้โดย Software
SCON.2	RB8	9AH	ค่าของบิต 9 เมื่อรับข้อมูลเข้ามา
SCON.1	TI	99H	บิตแฟลคแสดงการอินเทอร์รัพท์ภายหลังการส่งข้อมูลออกไปโดยจะ Set เมื่อส่งข้อมูลออกไปหมดแล้วและ
SCON.0	RI	98H	สามารถ Clear ได้ด้วย Software แฟลคแสดงการอินเทอร์รัพท์ภายหลังรับข้อมูลเข้ามา สามารถ clear ได้ด้วย Software

ก่อนที่จะใช้พอร์ทอนุกรมจะต้องโปรแกรมให้กับ SCON เสียก่อนเพื่อกำหนดโหมดการทำงานและลักษณะต่างๆ เช่น

MOV SCON, #01010010 B

เป็นการกำหนดให้พอร์ทอนุกรมทำงานในโหมด 1 และอินาเบิลให้มีการรับข้อมูลพร้อมกับการกำหนดให้ TI เป็น 1

ในการส่งข้อมูลทุกโหมดสามารถทำได้โดยเขียนข้อมูลไปยัง SBUF เมื่อข้อมูลถูกส่งไปแล้วบิต TI จะถูกเซตเป็น "1" ในการส่งข้อมูล จะต้องคอยตรวจบิต TI เพราะถ้า TI ยังไม่เป็น "1" แสดงว่าข้อมูลยังส่งไปไม่หมด ถ้าหากมีการเขียนข้อมูลไปต่อกันไปไปยัง SBUF จะทำให้เกิดข้อผิดพลาดขึ้น สำหรับในการรับข้อมูลบิต REN จะต้องเซตให้เป็น "1" ยกเว้นโหมด 0 เพื่ออนุญาตให้รับข้อมูลได้ เมื่อข้อมูลรับเข้ามาเรียบร้อยแล้ว บิต RI จะถูกเซตเป็น "1"

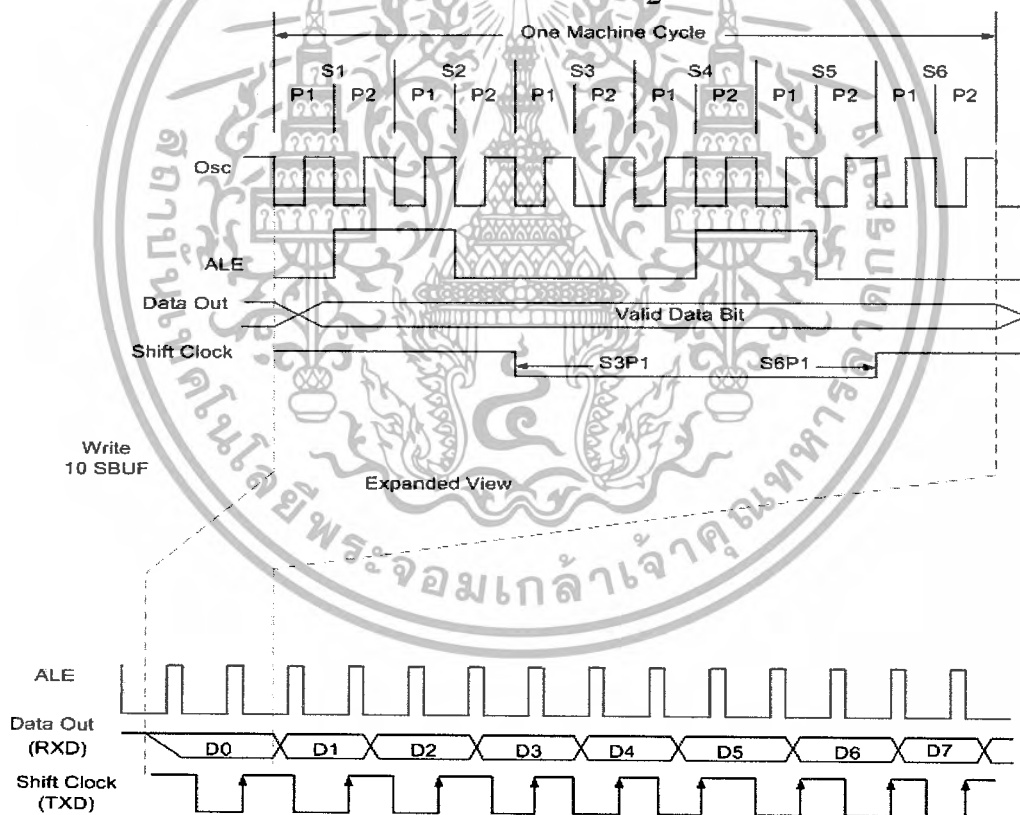
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mode of Operation

ใน MCS-51 การสื่อสารทางพอร์ตอนุกรมจะมีอยู่ 4 ประเภทหรือ 4 โหมด ซึ่งจะกำหนดได้ที่บิต SM0 และ SM1 ใน SCON โดยจะมี 3 โหมด เป็นการสื่อสารแบบ Asynchronous ในลักษณะของข้อมูลที่ส่ง จะมีบิตเริ่มต้น (Start Bit) และบิตจบ (Stop Bit) คล้ายกับการสื่อสารแบบ RS-232 ในระบบคอมพิวเตอร์ อีกโหมดหนึ่งจะเป็นการใช้พอร์ตอนุกรมในลักษณะชิพตริจิสเตอร์

8-Bit Shift Register (Mode 0)

การทำงานในโหมดนี้จะใช้ขา RXD ในการรับส่งข้อมูลโดยต่อกับ Shift Register ภายนอก ส่วนขา TXD จะเป็น Output Shift Clock เพื่อกระตุ้นรีจิสเตอร์ภายนอกให้เลื่อนบิตถ้ามีการส่งข้อมูลหรือรับข้อมูล 8 บิต จะเริ่มที่บิตต่ำสุดก่อน โดยมีค่า Baud Rate เท่ากับ $\frac{1}{2}$ ของความถี่ที่ใช้บนชิพ

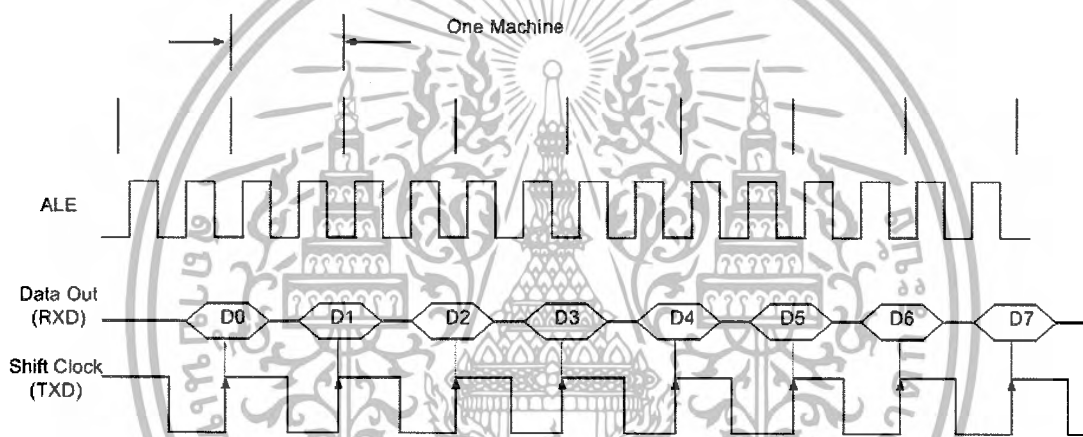


รูปที่ 2.7 ไตแกรมเวลาการส่งข้อมูล

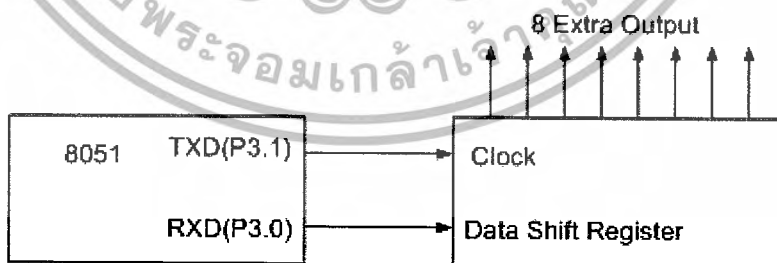
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการส่งข้อมูลจะทำโดยเขียนข้อมูลไปที่รีจิสเตอร์ SBUF ข้อมูลจะถูกส่งออกมาทางขา RXD (P3.0) โดยจะสอดคล้องกับสัญญาณที่ออกมาทางขา TXD ซึ่งสัญญาณทางขา TXD จะถูกส่งออกมาทุกๆ Machine Cycle โดยจะเป็นลอจิก"0" ใน S3P1 และจะกลับเป็นลอจิก "1" ใน S6P1 ซึ่งแสดงได้ดังรูปที่ 2.7

สำหรับการรับข้อมูลจะรับได้เมื่อเซตขา Receiver Enable Bit (REN)เป็น"1"และเคลียร์ขา Receiver Interrupt Bit(RI)เป็น"0" ข้อมูลจะเข้าสู่ MCS-51 เมื่อ Clock Shift ถูกส่งออกไปทาง TXD ที่ขอบขาขึ้นของ Clock Shift บิตต่ำจะถูกส่งเข้ามาก่อนดังรูปที่ 2.8



รูปที่ 2.8 ไตอะแกรมเวลาการรับข้อมูล



รูป 2.9 การส่งข้อมูลออกโดยใช้พอร์จิสเตอร์ช่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการประยุกต์ใช้งานโหมคนี้จะต้องมีไอซีชิพตรีจิสเตอร์มาต่อภายนอก เช่น ถ้าหากต้องการส่งข้อมูลออกมาพอร์ทอนุกรม อาจต่อวงจรได้ดังรูปที่ 2.9 โดยใช้ไอซี Serial – to –Parallel Shift Register โดยข้อมูลส่งออกมาทาง RXD และใช้ TXD เป็น Clock

8-Bit UART with Variable Baud Rate (Mode 1)

ในโหมคนี้จะเป็นการรับส่งข้อมูลแบบ 10 บิต ซึ่งประกอบด้วยบิตเริ่มต้น (เป็น “0”) ข้อมูล 8 บิตและบิตจบ (เป็น “1”) นอกจากนี้ยังสามารถกำหนดค่า Baud Rate ได้โดยค่า Baud Rate นี้จะแปรตามตัวจับเวลาตัวที่ 1 ในโหมคนี้ จะส่งข้อมูลออกจาก TXD และรับข้อมูลเข้าทาง RXD ถ้าเป็นการรับข้อมูลเข้าตัว Stop Bit จะเข้ามายังบิต RB8 ใน SCON



รูปที่ 2.10 การรับส่งข้อมูลในโหมค 1

ค่า Baud Rate ที่ใช้ในการรับส่งข้อมูลจะกำหนดโดย Timer 1 หลังจากโปรแกรมไปใน Timer 1 แล้วสามารถเลือกค่า Baud Rate ได้อีกสองค่าคือ ค่าจาก Timer 1 Overflowหาร 32 กับค่าจาก Timer 1 Overflowหาร 16

การรับส่งข้อมูลทำได้โดยการเขียนข้อมูล 8 บิตที่ SBUF โดยบิตที่ 9 (Stop Bit) ให้เขียนลงใน TB8 ใน SCON จากนั้นข้อมูลจะถูกส่งออกมาทาง TXD โดยส่ง Start Bit ออกมาก่อน ตามด้วยข้อมูล 8 บิตและจบด้วย Stop Bit เมื่อข้อมูลถูกส่งออกไปหมดแล้วบิต Interrupt Flag (TI) จะเป็น “1” ดังนั้นในการเขียนข้อมูลใหม่ลงไปจะต้องตรวจสอบบิตนี้

ในการรับข้อมูล จะเริ่มจากเมื่อมีการเปลี่ยนแปลงลอจิกจาก 1 เป็น 0 ทางขา RXD หมายความว่าเริ่มรับบิตต้น จากนั้นข้อมูลอีก 8 บิตจะถูกเก็บลงใน SBUF และ Stop Bit จะถูกเก็บในบิต RB8 ของรี

จีเตอร์ SCON เมื่อข้อมูลเข้ามาครบแล้ว บิต Interrupt Flag (RI) จะถูกเซต ดังนั้นในการอ่านข้อมูลจะอ่านได้เมื่อบิต RI ถูกเซตแล้ว เมื่ออ่านข้อมูลไปแล้วจะต้องเคลียร์บิตนี้

9 – Bit UART with Fixed Baud Rate (Mode 2)

การทำงานในโหมดนี้ไม่สามารถกำหนดค่า Baud Rate ได้ ซึ่งค่า Baud Rate จะมีสองค่าคือ 1/64 และ 1/32 ของสัญญาณนาฬิกาบนชิพ การรับส่งข้อมูลจะเป็นชุดข้อมูล 9 บิต บิตเริ่มต้น บิตหยุดรวมเป็น 11 บิต โดยข้อมูล 9บิตจะเป็นจำนวนข้อมูล 8 บิตและบิตที่โปรแกรมอีก 1 บิตโดยบิตนี้จะเป็นบิตที่ 9 ซึ่งจะเป็น Parity บิต ในการส่งข้อมูลจะต้องเขียนไปที่ บิต TB8 ในรีจิสเตอร์ SCON สำหรับการรับข้อมูลบิตที่ 9 จะถูกเก็บในบิต RB8

9 – Bit UART with Variable Baud Rate (Mode 3)

การทำงานในโหมดนี้จะคล้ายกับโหมด 2 แต่สามารถกำหนดค่า Baud Rate ได้ โดยการโปรแกรมไปที่ Timer 1 หลังจากโปรแกรมแล้วยังสามารถเลือกได้อีก 2 ค่าคือ ความถี่การ Overflow ของ Timer 1 ทหารด้วย 16 และหารด้วย 32

การกำหนดค่าเริ่มต้นให้รีจิสเตอร์ในการรับส่งข้อมูล

การรับข้อมูล ถ้าจะให้ MCS-51 รับข้อมูลทางพอร์ทอนุกรมจะต้องโปรแกรมไปที่บิต Receiver Enable (REN) ในรีจิสเตอร์ SCON ให้เป็นลอจิก “1” ซึ่งอาจทำได้สองวิธีดังนี้

SETB REN

เป็นการเซตบิต REN ให้เป็น “1” หรืออาจทำโดยใช้คำสั่ง

MOV SCON, #xxx1xxxxB

ซึ่งเป็นการย้ายข้อมูลที่ทำให้บิต REN เป็น 1 สำหรับค่า X หมายความว่า เป็นอะไรก็ได้ขึ้นกับการใช้งานในโหมดต่างๆ

ข้อมูลชุดแบบ 9 บิต ในการรับส่งข้อมูลที่มีข้อมูลแบบ 9 บิต ได้แก่ การใช้งานในโหมด 2 และโหมด 3 การส่งข้อมูลบิตที่ 9 จะถูกเขียนในบิต TB8 โดยการเขียนโปรแกรม สำหรับการรับข้อมูลเมื่อข้อมูลเข้ามาถึงบิตที่ 9 จะถูกเขียนลงในบิต RB8

การเพิ่มบิต Parity การส่งข้อมูลแบบ 9 บิต สามารถใช้บิตที่ 9 เป็นบิต Parity ได้ ซึ่งบิต Parity จะอยู่ใน Program Status Word (PSW) โดยจะถูกเซตหรือเคลียร์ทุกๆ เมฆขึ้นไซเคิลที่เกี่ยวข้องกับ Accumulator เช่น ถ้าจะส่งข้อมูล 8 บิต ตามด้วยบิต Even Parity เป็นบิตที่ 9 สามารถเขียนโปรแกรมได้ดังนี้

MOV C, P ; อ่านค่าบิต P มาเก็บใน C
 MOV TB8, C ; นำค่าบิต Parity เขียนลงใน TB8
 MOV SBUF, A; ส่งข้อมูลไปทางพอร์ทอนุกรม

ถ้าเป็นแบบ Odd Parity ให้แก้ไขข้อมูลที่อ่านได้จากบิต Parity เสียก่อนที่จะส่งออกไป ซึ่งเขียนโปรแกรมดังนี้

MOV C, P ; อ่านค่าบิต Parity มาเก็บใน C
 CPL C ; กลับค่าให้เป็น Odd Parity
 MOV TB8, C ; เขียนค่าลงใน TB8
 MOV SBUF, A ; ส่งข้อมูลออกทางพอร์ทอนุกรม

การส่งข้อมูลแบบมี Parity บิตด้วยไม่ใช่จะส่งได้แบบ 9 บิต หรือ โหมด 2 และ 3 เท่านั้น ในโหมด 1 ซึ่งส่งข้อมูลแบบ 8 บิตก็สามารถทำได้ อย่างเช่นการส่งรหัส ASCII จะใช้บิตข้อมูล 7 บิต สำหรับบิตที่เหลืออีกหนึ่งบิตจะเป็นบิต Parity รวมเป็น 8 บิต ซึ่งสามารถเขียนโปรแกรมได้ด้วยดังนี้

CLR ACC.7 ; เคลียร์ค่าบิต 7 เพื่อใช้เป็น Parity
 MOV C, P ; นำบิต Parity มาเก็บใน C
 MOV ACC.7, C ; เขียนค่าบิต Parity ลงในรีจิสเตอร์ A
 MOV SBUF, A ; ข้อมูลออกทางพอร์ทอนุกรม

แฟลคอินเทอร์รัพท์ เมื่อมีการรับส่งข้อมูลเสร็จสิ้นจะมีผลต่อแฟลคอินเทอร์รัพท์ (RI และ TI) ในรีจิสเตอร์ SCON ซึ่งบิตเหล่านี้จะถูกเซตโดย Hardware แต่ต้องเคลียร์ด้วย Software

บิต RI ถ้าถูกเซตหมายความว่าพเพอร์ที่ไว้รับข้อมูลเต็มให้อ่านข้อมูลไปแล้ว และบิตนี้สามารถใช้อินเทอร์รัพท์ MCS-51 ได้แต่ถ้าเขียนโปรแกรมจะใช้วิธีตรวจเช็คบิตนี้ ถ้าเป็น "1" หมายความว่าให้อ่านข้อมูลมาเก็บในรีจิสเตอร์ A ได้ แต่ก่อนอ่านจะต้องเคลียร์ RI เสียก่อนเพื่อจะได้รับข้อมูลถัดไปได้ ซึ่งเขียนโปรแกรมได้ดังนี้

WAIT: JNB RI, WAIT ; ถ้าบิตนี้ไม่เป็น "1" จะทำงานอยู่ที่เดิม
 CLR RI ; เคลียร์ RI
 MOV A, SBUF ; อ่านค่ามาเก็บใน A

บิต TI เมื่อส่งข้อมูลออกไปแล้วบิตนี้จะถูกเซต เป็นการบอกว่าบัพเฟอร์ส่งข้อมูลว่างแล้วให้ส่งข้อมูลใหม่เข้าไปได้ ซึ่งสามารถใช้บิตนี้อินเทอร์รัพท์ MCS-51 ได้เช่นกัน แต่ถ้าเขียนโปรแกรมคอยตรวจเช็คอาจเขียนได้ดังนี้

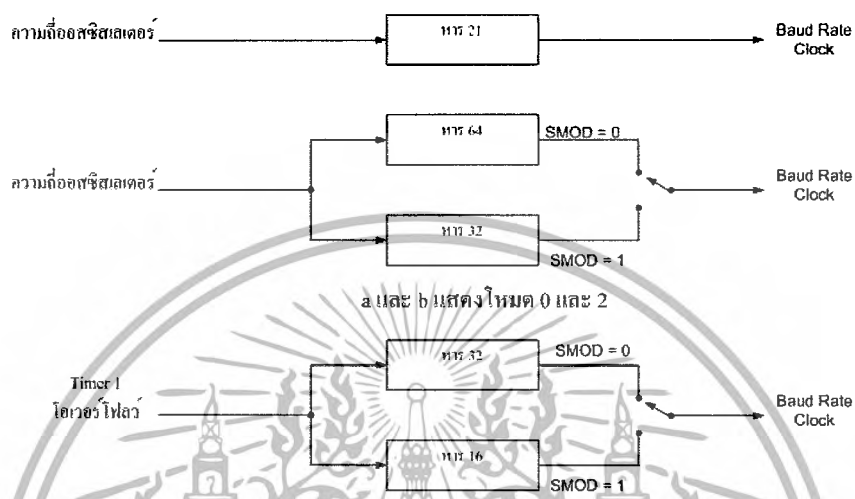
```
WAIT: JNB RI, WAIT ; ตรวจ TI ว่าเป็น "1" หรือยัง
CLR RI ; เคลียร์ TI
MOV SBUF, A ; เขียนข้อมูลลงไป
```

อัตราการส่งข้อมูลของพอร์ทอนุกรม

จากการศึกษาการรับส่งข้อมูลในโหมดต่างๆ พบว่าในโหมด 0 และโหมด 2 ไม่สามารถกำหนด Baud Rate เองได้ โดยในโหมด 0 ค่า Baud Rate จะมีค่าเท่ากับความเร็วของ Oscillator หากด้วย 12 ในโหมด 1 จะมีสองค่าคือ ความถี่ Oscillator หากด้วย 32 และหากด้วย 64 สองค่านี้เรียกว่า SMOD 0 และ SMOD 1 ซึ่งสามารถกำหนดได้ในรีจิสเตอร์ PCON บิตที่ 7 ในรีจิสเตอร์ PCON นี้ไม่สามารถเข้าถึงข้อมูลระดับบิตได้ การเขียนข้อมูลลงไปทีละบิตจะต้องใช้วิธีที่เรียกว่า "Read-Modify-Write" คืออ่านค่าขึ้นมาแก้ไขแล้วเขียนลงไปใหม่ ตัวอย่างเช่น

```
MOV A, PCON ; อ่านค่าจาก PCON มาเก็บในรีจิสเตอร์ A
SETB ACC.7 ; เซตบิต 7 (SMOD)
MOV PCON, A ; เขียนค่าลงไปใหม่ใน PCON
```

สำหรับโหมด 1 และโหมด 3 สามารถกำหนดค่า Baud Rate ได้โดยการโปรแกรมใน Timer 1 ในการโปรแกรมแต่ละครั้งจะมี SMOD สองค่าเช่นกัน ค่า Baud Rates ของโหมดต่างๆ แสดงได้ดังรูปที่ 2.11



รูป 2.11 แสดงการกำหนด Baud Rate ในโหมดต่างๆ

การใช้ Timer 1 กำหนด Baud Rate Clock

การกำหนดค่าลงใน Timer 1 ทำได้โดยการโปรแกรมไปที่ TMOD ให้ทำงานแบบ 8-bit Auto Reload Mode (โหมด 2) โดยเขียนค่าไปที่โปรแกรมไปริจิสเตอร์ TMOD ได้ดังนี้

MOV TMOD, #0010xxxxB

ค่า x หมายความว่า เป็นอะไรก็ได้ เพราะบิตเหล่านี้ใช้ใน Timer 0

ถ้าต้องการ Baud Rate ต่ำๆ สามารถใช้ 16-bit Mode ได้ โดยโปรแกรมเป็น TMOD = 0001xxxxB ค่า Baud Rate ที่ส่งออกมาจะมีค่าเท่ากับ ความถี่ของ Timer 1 เกิด Overflow หารด้วย 32 (หรือหารด้วย 16 ถ้าเป็น SMOD = 1)

การคำนวณหาอัตราบอดที่กำหนดด้วย Timer 1 สามารถทำได้โดยใช้สมการต่อไปนี้

$$\text{Baud Rate} = \frac{2^{SMOD}}{32} * \frac{\text{ความถี่ออสซิลเลเตอร์}}{12 * [256 - TH1]}$$

โดยที่ SMOD เป็นค่าของบิตภายในรีจิสเตอร์ PCON ซึ่งอาจมีค่าเป็น 0 หรือ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TH1 เป็นค่าภายในรีจิสเตอร์ TH1 ซึ่งใช้สำหรับ Reload ค่าของการนับเวลารูปแบบทั่วไปของการหาค่า Baud Rate ในโหมด 1 และ 3 สามารถหาได้ดังนี้

$$\text{Baud Rate} = \text{Timer 1 Overflow Rate} / 32$$

ถ้าต้องการค่า Baud Rate เท่ากับ 1,200 สามารถคำนวณค่าความถี่ Overflow ของ Timer 1 ได้ดังนี้

$$1,200 = \text{Timer 1 Overflow Rate} / 32$$

จะได้ Timer 1 Overflow Rate เท่ากับ 38.4 kHz

ถ้าระบบ MCS-51 ใช้ความถี่สัญญาณนาฬิกาจาก Crystal เท่ากับ 12 MHz ตัว Timer 1 จะได้รับ Clock เท่ากับ 1 MHz หรือ 1,000 kHz ถ้าเราต้องการ Timer 1 Overflow เท่ากับ 38.4 kHz ดังนั้นค่าอัตรา Overflow มีค่าเท่ากับ $1,000/38.4 = 26.04$ Clock โดยค่า Overflow จะเกิดเมื่อเกิดการเปลี่ยนจาก FFH เป็น 00H ดังนั้นจะต้องให้ Timer 1 นับไป 26 Count ดังนั้นค่าที่จะให้รีจิสเตอร์ TH1 มีค่าเท่ากับ -26 ซึ่งใช้เป็นค่า Reload ดังนั้นเขียนคำสั่งได้ดังนี้

```
MOV TH1, #-26
```

ตัวโปรแกรมแอสเซมบลีทั่วไปจะแปลงค่า -26 เป็น 0E6H เนื่องจากที่ผ่านมาจะเห็นว่าความถี่ Baud Rate จะมีความสัมพันธ์กับค่าสัญญาณนาฬิกาที่ใช้ จาก Crystal ในตารางที่ 2.3 จะเป็นค่าที่ต้องกำหนดใน Timer1 เมื่อต้องการค่า Baud Rate ต่างๆจะเห็นว่าถ้าใช้ความถี่ 12,000 MHz ค่า Baud Rate จะมีข้อผิดพลาด

ตารางที่ 2.3 แสดงความถี่สัญญาณนาฬิกาที่ใช้กำหนด Baud Rate ค่าต่างๆ

ค่า Baud Rate	Cryst	SMOD โหมด	ค่าใน TH1	ค่า Baud Rate ที่ได้	ผิดพลาด
9600	12.000	1	-7(F9H)	8923	7%
2400	12.000	0	-13(F3H)	2040	0.16%
1200	12.000	0	-16(E6H)	1020	0.16%
19200	11.059	1	-3(FDH)	19200	0
9600	11.059	0	-3(FDH)	9600	0
2400	11.059	0	-12(F4H)	2400	0
1200	11.059	0	-24(E8H)	1200	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

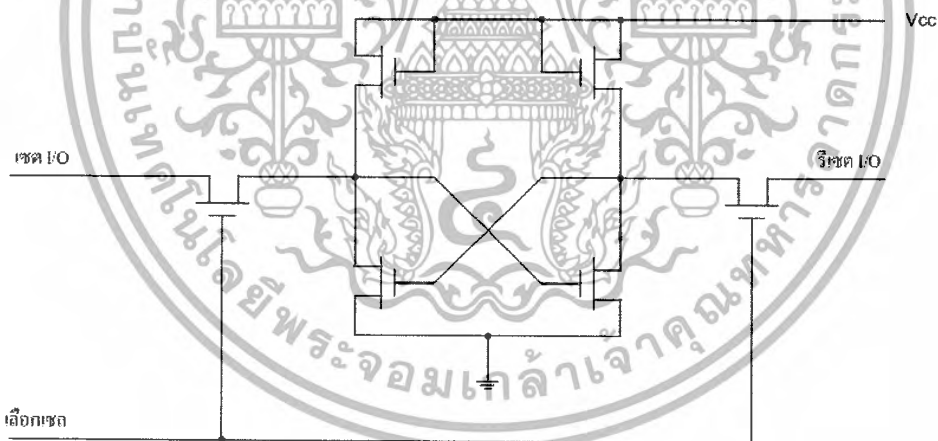
2.5 หน่วยความจำข้อมูล

หน่วยความจำชนิดนี้ CPU สามารถจะอ่านและเขียนข้อมูลได้ เรียกว่า RAM(Random Access Memory) เป็นหน่วยความจำที่ใช้เก็บข้อมูลจากการประมวลผลของ CPU ข้อมูลภายใน RAM จะคงอยู่ตลอดเวลาที่มีแหล่งจ่ายไฟต่ออยู่กับหน่วยความจำ ตัวหน่วยความจำ RAM นี้แบ่งออกได้เป็นสองชนิดใหญ่ๆ คือ

- สแตติกแรม(Static RAM)
- ไดนามิกแรม(Dynamic RAM)

สแตติกแรม (Static RAM)

เป็นหน่วยความจำ RAM ที่สามารถอ่านและเขียนข้อมูลได้ และข้อมูลจะยังคงอยู่ตลอดถ้ามีไฟเลี้ยง โครงสร้างภายในในการเก็บข้อมูลแต่ละบิตจะสร้างเป็นฟลิปฟล็อป ทำให้การเก็บข้อมูลแต่ละบิตจะต้องสร้างจากทรานซิสเตอร์ออกมาหลายตัวดังรูปที่ 2.12 แสดงโครงสร้างของสแตติกแรมในการเก็บข้อมูล 1 บิต

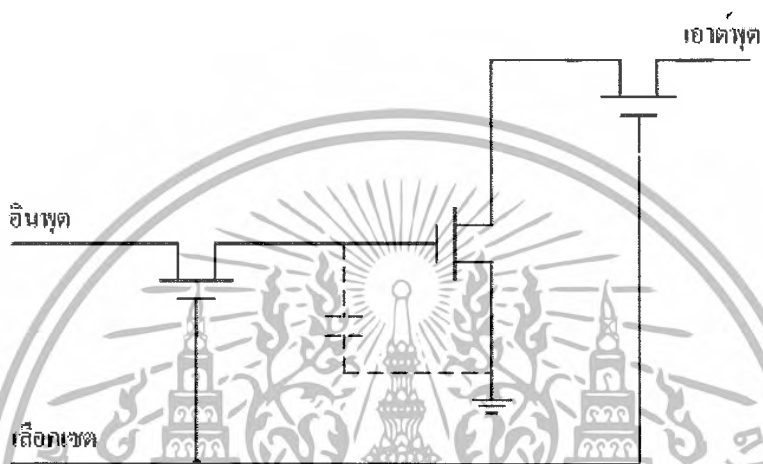


รูปที่ 2.12 แสดงการเก็บข้อมูลแต่ละบิตของสแตติกแรม

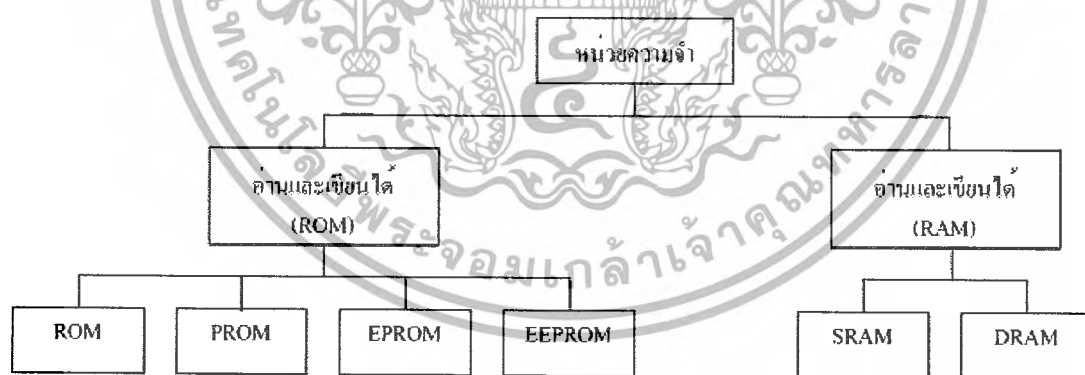
ไดนามิกแรม(Dynamic RAM)

เป็นหน่วยความจำที่ข้อมูลจะคงอยู่ตลอดไปถ้ามีไฟเลี้ยง แต่จะต้องมีการกระตุ้นหรือการเขียนข้อมูลซ้ำ (Refresh) ตลอดเวลาด้วยวงจรพิเศษ เนื่องจากโครงสร้างภายในของไดนามิกแรมจะสร้างเป็น

ตัวเก็บประจุ ดังนั้นในการเก็บข้อมูล 1 บิต จึงต้องมีการเขียนข้อมูลซ้ำเพื่อให้ประจุคงอยู่ และเนื่องจากโครงสร้างภายในเป็นแบบตัวเก็บประจุทำให้ในการเก็บข้อมูล 1 บิตจะสร้างทรานซิสเตอร์ไม่กึ่งตัว ซึ่งจะให้ความจุของข้อมูลต่อชิพสูงกว่าหน่วยความจำแบบสแตติกแรม โครงสร้างภายในแสดงได้ดังรูปที่ 2.13



รูป 2.13 แสดงโครงสร้างภายในของสแตติกแรม



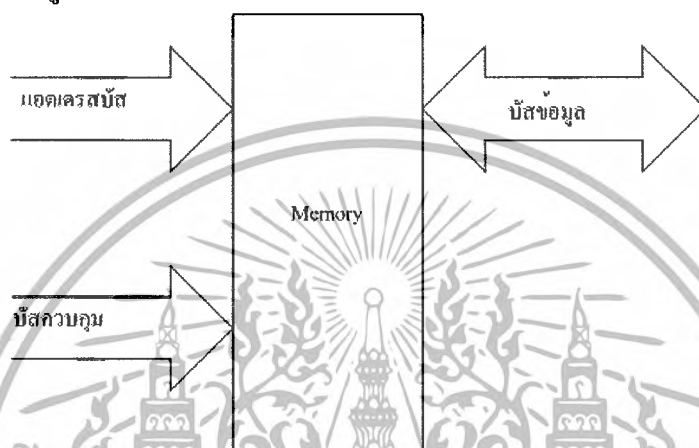
รูปที่ 2.14 แผนภาพของหน่วยความจำต่างๆ

2.6 การอ่านและเขียนข้อมูลกับหน่วยความจำ

ไอซีหน่วยความจำโดยทั่วไปจะประกอบด้วยกลุ่มสัญญาณ 3 กลุ่ม คือแอดเดรสบัส, บัสข้อมูล และบัสควบคุม โดยแอดเดรสบัสจะใช้ในการอ้างตำแหน่งถ้ามีแอดเดรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัส 10 เส้น คือ A0 – A9 สามารถอ้างตำแหน่งได้ 2^{10} หรือ 1 กิโลไบต์ บัสข้อมูลจะเป็นตัวบอกว่าในแต่ละตำแหน่งจะเก็บข้อมูลได้ที่บิต เช่นถ้ามีบัสข้อมูล 8 เส้น คือ D0 – D7 สามารถเก็บข้อมูลได้ตำแหน่งละ 1 ไบต์ ส่วนบัสควบคุมจะใช้ในการอ่านเขียนหน่วยความจำ ระบบบัสทั้ง 3 ระบบแสดงได้ดังรูปที่ 2.15



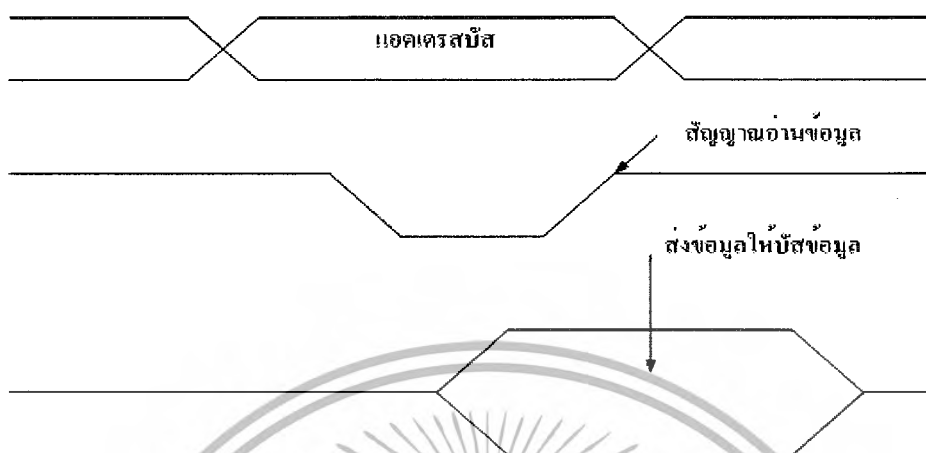
รูป 2.15 แสดงกลุ่มสัญญาณที่ใช้ในการติดต่อกับหน่วยความจำ

โดยทั่วไปไอซีหน่วยความจำจะมีขา CE (Chip Enable) ไว้สำหรับเลือกให้ไอซีตัวที่ต้องการใช้ทำงาน ในกรณีที่มีไอซีหน่วยความจำหลาย ๆ ตัว ขา OE (Output Enable) จะใช้ต่อกับไอซีถอดรหัสสำหรับอ่านข้อมูล ถ้าขานี้แอกทีฟ ข้อมูลที่อยู่ในไอซีหน่วยความจำจะส่งออกมาทางบัสข้อมูลได้

การอ่านข้อมูลจากหน่วยความจำ มีลำดับขั้นดังนี้

1. ส่งตำแหน่งที่จะอ่านไปก่อนทางแอดเดรสบัส
2. ส่งสัญญาณควบคุมว่าต้องการจะอ่าน
3. ข้อมูลภายในไอซีหน่วยความจำถูกส่งออกมาทางบัสข้อมูล

การอ่านข้อมูลจากหน่วยความจำสามารถเขียน ไตอะแกรมเวลา ได้ดังรูปที่ 2.16



รูปที่ 2.16 แสดงสัญญาณในการอ่านข้อมูล

การเขียนข้อมูลลงหน่วยความจำ มีลำดับขั้นดังนี้

1. ส่งตำแหน่งที่จะเขียนข้อมูลออกไปก่อนทางแอดเดรสบีต
2. ส่งข้อมูลที่จะเขียน ไปทางบัสข้อมูล
3. ส่งสัญญาณเขียนข้อมูล

การเขียนข้อมูลให้กับหน่วยความจำสามารถเขียนเป็น ไคอะแกรมเวลาได้ดังที่ 2.17



รูปที่ 2.17 แสดงสัญญาณในการเขียนข้อมูล

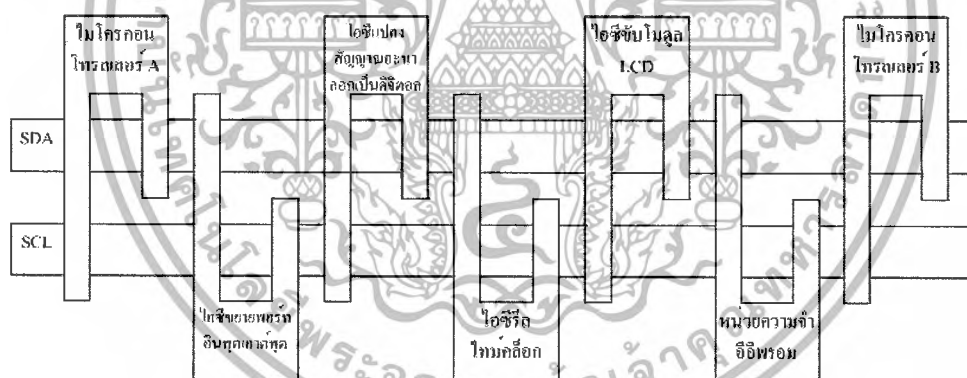
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 การขยายพอร์ตอินพุตเอาต์ด้วยอุปกรณ์ระบบบัส I^2C

ความรู้เบื้องต้นเกี่ยวกับ I^2C

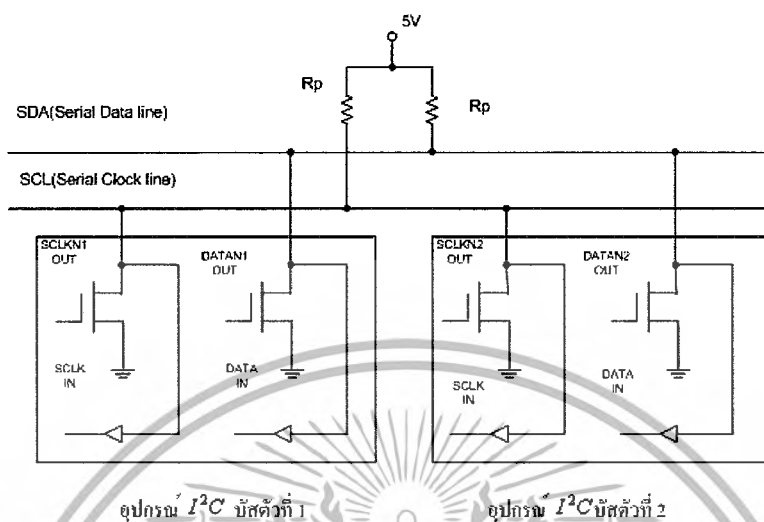
I^2C ย่อมาจาก Inter-IC Communication หมายถึง การติดต่อสื่อสารระหว่างไอซี โดยบัส I^2C ได้รับการพัฒนาขึ้นโดยฟิลิปส์ (Philips) ด้วยจุดมุ่งหมายหลักคือต้องการให้ไอซีหรือโมดูลสามารถติดต่อ สั่งงาน และควบคุมภายใต้สายสัญญาณเพียง 2 เส้น เส้นหนึ่งคือ สายข้อมูล อีกเส้นหนึ่งคือ สายสัญญาณนาฬิกาที่ใช้ในการกำหนดจังหวะการทำงาน การต่อร่วมกันของอุปกรณ์บนบัส I^2C ทำได้ง่ายมาก เพียงต่อสายข้อมูลและสายสัญญาณนาฬิกาของอุปกรณ์แต่ละตัวขนานหรือพ่วงกันไปส่วน การกำหนดแอดเดรสหรือตำแหน่งสำหรับติดต่ออุปกรณ์แต่ละตัว จะใช้รหัสข้อมูลและการกำหนดสภาวะลอจิกที่ขาแอดเดรสของอุปกรณ์แต่ละตัว

สายข้อมูลบนบัส I^2C มีชื่อเรียกอย่างเป็นทางการว่า สายข้อมูลอนุกรมหรือ SDA (Serial Data line) ส่วนสายสัญญาณนาฬิกามีชื่อว่า สายสัญญาณนาฬิกาอนุกรมหรือ SCL (Serial Clock line) ในการอธิบายต่อไปนี้จะเรียกสายสัญญาณทั้งสองว่าสาย SDA และ SCL



รูปที่ 2.18 ฟังแสดงการเชื่อมต่อของอุปกรณ์ต่างๆ บนระบบบัส IC

ในรูปที่ 2.17 แสดงผังของการเชื่อมต่ออุปกรณ์ต่างๆ บนบัส I^2C จะเห็นได้ว่า อุปกรณ์ที่ทำการเชื่อมต่อบนบัส I^2C มีหลากหลาย ไม่ว่าจะเป็นไอซีขยายพอร์ตอินพุต (I/O Expander), ไอซีแปลงสัญญาณอนาล็อกเป็นดิจิทัล (ADC) และแปลงสัญญาณดิจิทัลเป็นอนาล็อก (DAC), ไอซีรีลไทม์คล็อก (RTC), ไอซีรับโมดูล LCD, หน่วยความจำอีอีพรอม และไมโครคอนโทรลเลอร์

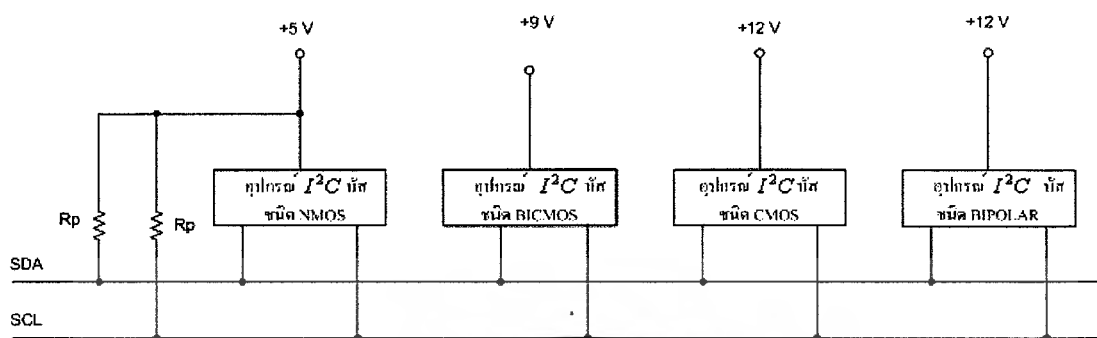


รูปที่ 2.19 แสดงวงจรเอาต์พุตของอุปกรณ์ในระบบ I^2C

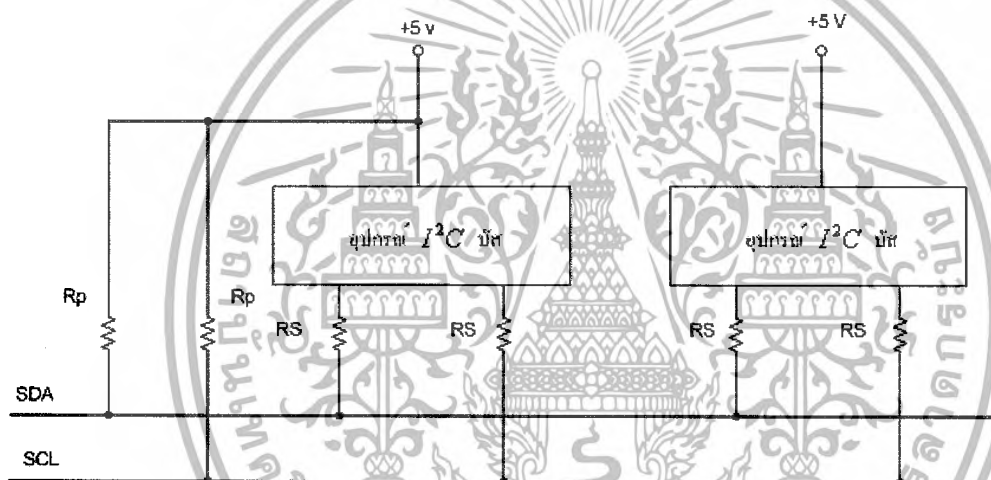
คุณสมบัติโดยทั่วไปของบัส I^2C

สาย SDA และ SCL เป็นสายสัญญาณ 2 ทิศทาง (bi-direction line) ต้องมีการต่อตัวต้านทาน पुलอัปกับแรงดัน +5 V ไว้ตลอดเวลาเพื่อให้สายมีสถานะลอจิกสูงในขณะที่ไม่มีการติดต่อใช้งาน ทั้งช่วยในการป้องกันสัญญาณรบกวนที่อาจมีเข้ามาในสายสัญญาณทั้งสอง วงจรเอาต์พุตของอุปกรณ์ที่ต่ออยู่บนบัส I^2C ต้องมีลักษณะเป็นวงจรเดรนเปิด (open-drain) หรือคอลเล็กเตอร์เปิด (open-collector) ดังแสดงรายละเอียดในรูปที่ 2.19

อัตราการถ่ายทอข้อมูลบนบัส IC สูงถึง 100 กิโลบิตต่อวินาทีในโหมดปกติ (standard mode) และสูงถึง 400 กิโลบิตต่อวินาทีใน โหมดความเร็วสูง (fast mode) อุปกรณ์ที่ต่อรวมอยู่บนบัส I^2C จะต้องมีค่าความจุไฟฟ้ารวมที่เกิดขึ้นระหว่างสาย SDA และ SCL ไม่เกิน 400 pF การเข้าถึงอุปกรณ์บนบัส I^2C ใช้ข้อมูลสำหรับการเข้าถึง 2 คำคือ 7บิต (7-bit addressing) หรือ 10 บิต (10-bit addressing) ข้อเด่นอีกประการหนึ่งของบัส I^2C ก็คือสามารถเชื่อมต่ออุปกรณ์ที่ใช้ไฟเลี้ยงไม่เท่ากันให้สามารถติดต่อสื่อสารกันได้ โดยอุปกรณ์บนบัส I^2C ตัวหนึ่งอาจใช้ไฟเลี้ยง +5V ในขณะที่อีกตัวหนึ่งใช้ไฟเลี้ยง +12 V การต่อร่วมกันบนบัส I^2C สามารถกระทำได้ในลักษณะเดียวกับกรณีที่อยู่บนบัสที่ใช้ไฟเลี้ยงเท่ากัน กล่าวคือ ให้ต่อสาย SDA และ SCL ของอุปกรณ์แต่ละตัวเข้าด้วยกัน และต้องต่อตัวต้านทาน पुलอัป (R_p) เข้ากับแรงดัน +5V ไว้ด้วยเสมอ ดังแสดงในรูปที่ 2.20



รูปที่ 2.20 การต่อตัวต้านทานพูลอัปบนสายสัญญาณในระบบบัส I²C



รูปที่ 2.21 การต่อตัวต้านทาน Rs เพื่อลดสัญญาณรบกวนขนาดใหญ่ที่อาจเข้ามาในบัส

I²C ในกรณีที่อาจมีแรงดันไฟฟ้ากระชากขนาดใหญ่ปะปนเข้ามาในบัส I²C ที่ขา SDA และ SCL ของอุปกรณ์แต่ละตัวต้องต่อตัวต้านทานอนุกรมกับขา SDA และ SCL เรียกว่า Rs ก่อนต่อเข้าสู่บัส I²C ดังแสดงในรูปที่ 2.21

หลักการของบัส I²C

บัส I²C ประกอบด้วยสายสัญญาณ 2 เส้น ดังที่ได้กล่าวมาแล้วคือ SDA และ SCL อุปกรณ์ที่ต่อพ่วงบนบัสสามารถมีได้มากมาย ดังนั้นจึงต้องมีการกำหนดรูปแบบของการติดต่อบนบัส หรือเรียกว่า

โปรโตคอล (protocol) เพื่อให้ผู้ใช้งานทราบว่า ขณะนี้อุปกรณ์ใดติดต่อกันอยู่ และอุปกรณ์ตัวใดเป็นตัวรับหรือตัวส่ง ต่อไปนี้จะขออธิบายลักษณะ หน้าที่ และนิยามของอุปกรณ์ที่ต่ออยู่บน บัส I^2C เพื่อเป็นข้อตกลงพื้นฐานก่อนที่จะอธิบายการทำงานของบัส I^2C ต่อไป

อุปกรณ์ที่ เป็นผู้สร้างข้อมูลหรือส่งข้อมูล เรียกว่า ตัวส่ง (Transmitter)

อุปกรณ์ที่ เป็นผู้รับข้อมูล เรียกว่า ตัวรับ (Receiver) ในอุปกรณ์บนบัส I^2C สามารถเป็นได้ทั้ง ตัวรับและตัวส่ง บางอุปกรณ์ทำหน้าที่เป็นตัวรับเพียงอย่างเดียว จะไม่มีอุปกรณ์ใดบนบัส I^2C ที่ทำหน้าที่เป็นตัวส่งเพียงอย่างเดียว

อุปกรณ์ที่ทำหน้าที่ควบคุมจังหวะการติดต่อบนบัส I^2C เรียกว่า มาสเตอร์ (Master)

อุปกรณ์ที่ถูกควบคุมหรืออุปกรณ์ที่ต่อพ่วงเข้าไปบนบัส I^2C เรียกว่า สเลฟ (Slave)

ข้อกำหนด 2 ประการสำคัญของการติดต่อบนบัส I^2C คือ

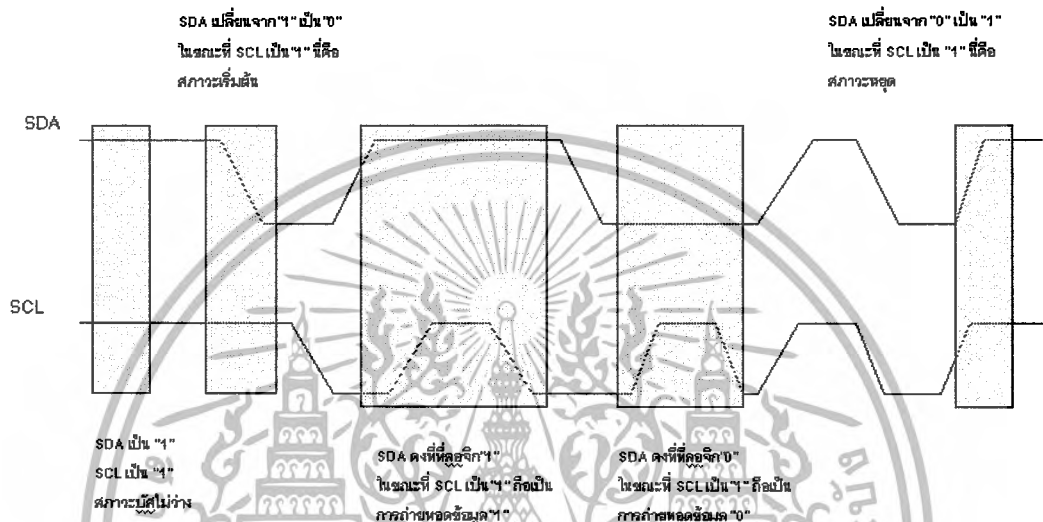
- (1) การถ่ายทอดข้อมูลขณะเกิดขึ้นได้เมื่อบัสว่างเท่านั้น
- (2) ในระหว่างการถ่ายทอดข้อมูล เมื่อใดก็ตามที่สาย SCL มีสถานะเป็นลอจิกสูง สายข้อมูลต้องรักษาข้อมูลไว้ อย่าให้เกิดการเปลี่ยนแปลงขึ้นเด็ดขาด มิฉะนั้น สัญญาณที่เกิดขึ้นจะได้รับการแปลความหมายเป็นสัญญาณควบคุมแทน

สถานะที่เกิดขึ้นบนบัส I^2C

มีด้วยกัน 5 สถานะดังนี้

- (1) บัสว่าง (Bus not busy) สถานะนี้เกิดขึ้นเมื่อสถานะลอจิกบนสาย SDA และ SCL เป็นลอจิกสูงทั้งคู่ นั่นหมายความว่า การถ่ายทอดข้อมูลสามารถเริ่มต้นขึ้นได้
- (2) เริ่มต้นการถ่ายทอดข้อมูล (start data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากสูงไปต่ำ ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสถานะที่เกิดขึ้นนี้ว่า สถานะเริ่มต้น (START)
- (3) หยุดการถ่ายทอดข้อมูล (Stop data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากต่ำไปสูง ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสถานะที่เกิดขึ้นว่า สถานะหยุด (STOP)
- (4) ข้อมูลดำรงอยู่บนบัส (data valid) สถานะนี้เกิดขึ้นถัดจากสถานะเริ่มต้น โดยสถานะลอจิกที่เกิดขึ้นบนสาย SDA ก็คือข้อมูลที่ทำการถ่ายทอด เมื่อสาย SCL เป็นลอจิกสูงสถานะที่สาย SDA ต้องคงที่ เพื่อให้อุปกรณ์รับรู้ข้อมูลในจังหวะนั้นว่าเป็น “0” หรือ “1” ข้อมูลอาจเกิดการเปลี่ยนแปลงได้ในขณะที่สาย SCL เป็นลอจิกต่ำ แต่เมื่อใดก็ตามที่ต้องการให้เกิดการถ่ายทอดข้อมูลอย่างสมบูรณ์ สถานะลอจิกที่ขา SDA ต้องคงที่ตลอดช่วงเวลาที่สาย SCL มีสถานะลอจิกสูง หากเกิดการเปลี่ยนแปลงสถานะลอจิกในขณะที่สาย SCL มีลอจิกสูงอยู่นั้นอุปกรณ์มาสเตอร์ที่ทำการควบคุม

การถ่ายทอดข้อมูลจะแปลความหมายเป็นสถานะหยุดหรือสถานะเริ่มต้นก็ได้ ทำให้ข้อมูลที่ทำการถ่ายทอดนั้นเกิดความผิดพลาดขึ้น



รูปที่ 2.22 ไตอะแกรมเวลาแสดงสถานะต่างๆในบัส I²C

(5) รับรู้ข้อมูล (acknowledge)เกิดขึ้นหลังจากที่การถ่ายทอดข้อมูลจากตัวส่งมายังตัวรับเกิดขึ้น อย่างสมบูรณ์ โดยตัวส่งจะทำการส่งข้อมูลมา 1บิตเรียกว่า บิตรับรู้(acknowledge bit) มีสถานะ เป็นลอจิกสูง หลังจากส่งข้อมูลมาครบถ้วน ส่วนอุปกรณ์มาสเตอร์จะทำการส่งสัญญาณรับรู้พิเศษซึ่งสัมพันธ์กับสัญญาณนาฬิกา เพื่อตอบสนองบิตรับรู้ที่ส่งมาจากตัวส่ง ทางด้านตัวรับจะส่งบิตรับรู้ที่มีสถานะลอจิกต่ำลงบนบัส อุปกรณ์สเลฟที่ถูกอ้างถึงในการติดต่อหรือกำลังติดต่ออยู่ในขณะนั้นก็จะกำเนิดบิตรับรู้เพื่อตอบสนองให้ทราบว่าได้รับข้อมูลในแต่ละไบต์เรียบร้อยแล้ว

ในรูปที่ 2.22 เป็นไตอะแกรมเวลาที่แสดงถึงการเกิดสถานะต่างๆบนบัส I²C ไม่ว่าจะเป็นสถานะบัสว่าง, เริ่มต้น, ถ่ายทอดข้อมูล, รับรู้ และหยุดการถ่ายทอดข้อมูล

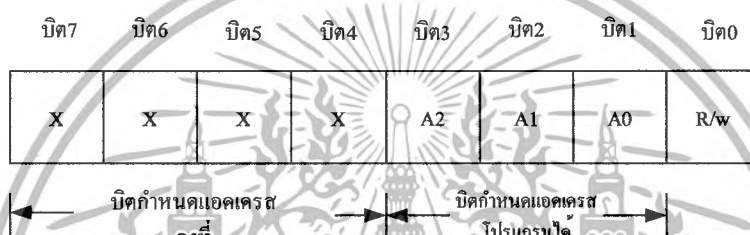
การทำงานบนบัส I²C

ก่อนที่จะเริ่มต้นการถ่ายทอดข้อมูลระหว่างอุปกรณ์ต่างๆ ที่ต่ออยู่บนบัส ต้องมีการอ้างถึงเสียก่อน โดยการอ้างถึงอุปกรณ์บนบัส I²C นั้นจะใช้การอ้างถึงแบบ 7 บิตหรือ 10 บิต ในกรณีที่มีอุปกรณ์ต่ออยู่บนบัสไม่มาก ใช้การอ้างถึงแบบ 7 บิตก็เพียงพอแต่ถ้ามีอุปกรณ์ต่ออยู่บนบัสมากกว่า 127

แอดเดรส จำเป็นต้องใช้การอ้างถึงแบบ 10 บิต หลังจากที่ได้ติดต่ออุปกรณ์แต่ละตัวได้เรียบร้อยแล้วก็จะเริ่มต้นการถ่ายทอดข้อมูลกันต่อไป

ดังนั้นหัวใจสำคัญในอันดับแรกของการทำงานบนบัส I^2C คือการอ้างถึงอุปกรณ์แต่ละตัวซึ่งในที่นี้จะอธิบายรายละเอียดของการอ้างถึงทั้ง 2 รูปแบบ

การอ้างถึงแบบ 7 บิต (7-bit addressing)



รูปที่ 2.23 รูปแบบของข้อมูลกำหนดแอดเดรสที่ใช้ในการอ้างถึงแบบ 7 บิต

ข้อมูลไบต์แรกที่เกิดขึ้นหลังจากสถานะเริ่มต้นคือ ข้อมูลที่ใช้ในการอ้างถึงอุปกรณ์ที่ต้องการติดต่อหรือ ข้อมูลกำหนดแอดเดรส โดยมีรูปแบบแสดงในรูปที่ 2.23 ใน 7 บิตบนรวมทั้งบิต MSB ด้วย จะเป็นข้อมูลแอดเดรสของอุปกรณ์สเลฟที่ต้องการติดต่อ โดยแบ่งเป็นบิตกำหนดแอดเดรสคงที่ (fixed address bit) จำนวน 4 บิต ซึ่งข้อมูลนี้ที่อุปกรณ์แต่ละตัวจะถูกกำหนดจากผู้ผลิต ไม่สามารถเปลี่ยนแปลงแก้ไขได้ถัดมาอีก 3 บิตเป็นบิตกำหนดแอดเดรสที่สามารถโปรแกรมได้ (programmable address bit) โดยผู้ใช้งานต้องกำหนดสถานะลอจิกให้แก่ว่า A0-A2 ของอุปกรณ์ที่มีการเชื่อมต่อแบบบัส I^2C ส่วนในบิต LSB เป็นบิตที่ใช้กำหนดการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวนั้นๆ หากบิต LSB เป็น "0" หมายถึงต้องการเขียนข้อมูลไปยังอุปกรณ์นั้นถ้าเป็น "1" จะเป็นการอ่านข้อมูลจากอุปกรณ์สเลฟ

ข้อมูลในไบต์ต่อมาคือ ข้อมูลควบคุม (Control byte) ในอุปกรณ์แต่ละตัวมีการกำหนดข้อมูลควบคุมที่แตกต่างกันไป ยกตัวอย่างไอซีขยายพอร์ตมีข้อมูลควบคุมที่ใช้กำหนดว่า บิตใดเป็นอินพุตบิตใดเป็นเอาต์พุต ในขณะที่ไอซี ADC/DAC ต้องการข้อมูลควบคุมเพื่อกำหนดให้ทำงานเป็นวงจรถอดค่า ADC หรือ DAC เป็นต้น

ข้อมูลในไบต์ต่อมาคือ ข้อมูลที่ทำการถ่ายทอดจริง (Data)

หลังจากที่มีการถ่ายทอดข้อมูลในแต่ละไบต์ อุปกรณ์เลขที่ที่ได้รับการติดต่อต้องส่งสัญญาณรับรู้ตอบกลับมาด้วยทุกครั้ง เพื่อให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้ ในรูป 7 แสดงรูปแบบข้อมูลอนุกรมที่เกิดขึ้นในการติดต่อบนบัส I^2C ของการอ้างถึงแบบ 7 บิต การอ้างถึงแบบ 10 บิต

ในการอ้างถึงแบบนี้ยังคงใช้รูปแบบข้อมูลอนุกรมที่เหมือนกับแบบ 7 บิต หากแต่จะมีข้อมูลเพิ่มเติมขึ้นมาเล็กน้อย โดยในข้อมูลไบต์แรกหลังจากเกิดสถานะเริ่มต้น ต้องกำหนดให้ 5 บิตบนมีข้อมูลเป็น 11110 ส่วนอีก 2 บิตถัดมาเป็นบิตแอดเดรสของอุปกรณ์ที่ต้องการติดต่อ ในบิต LSB ของข้อมูลไบต์แรกยังคงเป็นการกำหนดว่า ต้องการอ่านหรือเขียนข้อมูลกับอุปกรณ์เลขที่ที่ต้องการติดต่อด้วย ข้อมูลไบต์ต่อมาเป็นข้อมูลแอดเดรสในไบต์ที่ 2 ของอุปกรณ์ที่ต้องการติดต่อด้วย ข้อมูลไบต์ถัดไปจึงเป็นข้อมูลควบคุม ข้อมูลหลังจากนั้นก็จะจะเป็นข้อมูลจริงที่ใช้ในการติดต่อเช่นเดียวกับการอ้างถึงแบบ 7 บิต หลังจากถ่ายทอดข้อมูลครบทุกไบต์ ต้องมีสถานะรับรู้เกิดขึ้น เพื่อให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้ในรูปที่ 8 แสดงรูปแบบข้อมูลอนุกรมของการอ้างถึงแบบ 10 บิต อุปกรณ์ที่ใช้การเชื่อมต่อแบบบัส I^2C

ในปัจจุบัน บัส I^2C ได้รับความนิยมเพิ่มมากขึ้นเรื่อยๆ ด้วยข้อดีที่ชัดเจนคือ ใช้สายสัญญาณเพียง 2 เส้นเท่านั้นและการขยายระบบไมโครคอนโทรลเลอร์ ที่มีจำนวนอินพุตเอาต์และหน่วยความจำกักจำกัดสามารถทำได้ง่ายขึ้นด้วยระบบบัส I^2C เมื่อเป็นเช่นนี้จึงมีอุปกรณ์เพอร์IPHERAL ที่ใช้การเชื่อมต่อแบบบัส I^2C มากมายจากหลายผู้ผลิตออกมาให้ใช้งานกันดังมีตัวอย่างต่อไปนี้

ไอซีขยายพอร์ตอินพุตเอาต์พุต(I/O expander) PCF8574, PCF8582, PCF8584

ไอซีหน่วยความจำอีอีพรอมอนุกรม(Serial EEPROM):24Cxx, PCF8570, PCF72/73, PCF8582

ไอซี ADC/DAC: PCF8591

ไอซีรีไทม์คล็อก(Real-time clock: RTC):PCF8583, PCF8593, PCF8598, 41T56C

ไอซี LCD โมดูล (LCD driver): PCF8466, PCF8576, PCF8577/78, PCF8579, SAA1064

ไอซีกำหนดสัญญาณ DTMF (DTMF generator): PCD3311/12

การต่ออุปกรณ์ระบบบัส I^2C กับไมโครคอนโทรลเลอร์ MCS-51

สามารถทำได้ง่ายมาก เพียงใช้ขาพอร์ต 2 ขาโดยกำหนดให้ขาหนึ่งเป็น SDA อีกขาหนึ่งเป็น SCL และต่อตัวต้านทานค่าประมาณ 4.7k พูลอัปที่ขาพอร์ตทั้งสองขา เพียงเท่านี้ก็สามารถติดต่อกับอุปกรณ์ระบบบัส I^2C ได้แล้ว

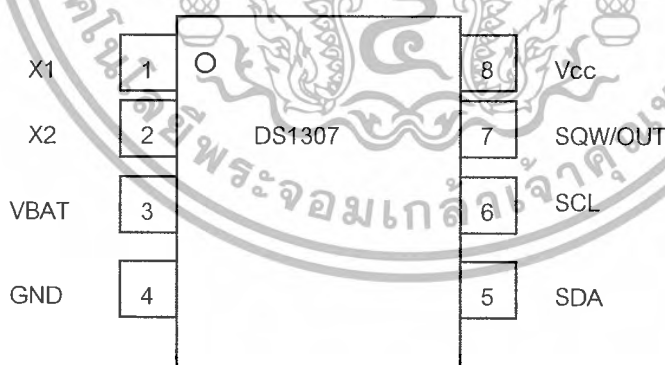
2.8 การใช้งานไอซีสร้างฐานเวลาจริงหรือรีลไทม์คล็อก (RTC)

DS1307 ไอซีสร้างฐานเวลาจริงหรือรีลไทม์คล็อก (RTC) ผู้ผลิตคือ ดัลลัสเซมิคอนดักเตอร์ (Dallas semiconductor) มีหน้าที่สร้างฐานเวลาจริงให้แก่ระบบไมโครคอนโทรลเลอร์โดย DS1307 จะให้ข้อมูลเกี่ยวกับเวลาทั้งหมด ไม่ว่าจะเป็นค่าของเวลาที่ละเอียดถึงหลักวินาที, นาที, ชั่วโมง, วันที่ (date), วันในสัปดาห์ (day), เดือน และ ปี โดยสามารถปรับวันเดือนปีให้ตรงตามปฏิทินได้อย่างถูกต้อง รวมถึงการกำหนดวันในปีก่อนศักราชด้วย คุณสมบัติทางเทคนิคที่สำคัญมีดังนี้

- เป็นไอซีรีลไทม์คล็อกให้ข้อมูลตั้งแต่วินาทีจนถึงปี รวมถึงการปรับวันในปีก่อนศักราชด้วยสามารถให้ข้อมูลเวลาได้อย่างเที่ยงตรงถึงปีคริสตศักราช 2100
- มีหน่วยความจำอนโวลตาไทล์แรม 56 ไบต์อยู่ภายใน สามารถใช้เก็บข้อมูลทั่วไปได้
- ใช้การเชื่อมต่อแบบระบบบัส I^2C
- มีวงจรตรวจจับไฟเลี้ยงต่ำหรือหายไปอย่างอัตโนมัติและสามารถรักษาข้อมูลเวลาไว้ได้แม้ไม่มีไฟเลี้ยงไอซี

รายละเอียดขาต่อใช้งานของ DS1307

ในรูปที่ 2.24 แสดงการจัดขาของ DS1307 แต่ละขามีหน้าที่และการใช้งานดังนี้ Vcc,GND(ขา 8,4) ต่อไฟเลี้ยง +5V



รูป 2.24 การจัดขาของไอซี

VBAT (ขา3) ใช้ต่อกับแบตเตอรี่ 3V เพื่อรักษาการทำงานของวงจรสร้างฐานเวลาของ DS1307 ให้คงอยู่ต่อไป แม้ว่าไม่มีไฟเลี้ยงจ่ายให้แก่ DS1307 ชนิดของแบตเตอรี่ที่เหมาะสมคือ แบตเตอรี่แบบลิเธียม ซึ่งมีความจุ 40mAh หรือมากกว่า จะสามารถรักษาข้อมูลได้นาน 10 ปีที่อุณหภูมิ 25 องศาเซลเซียส

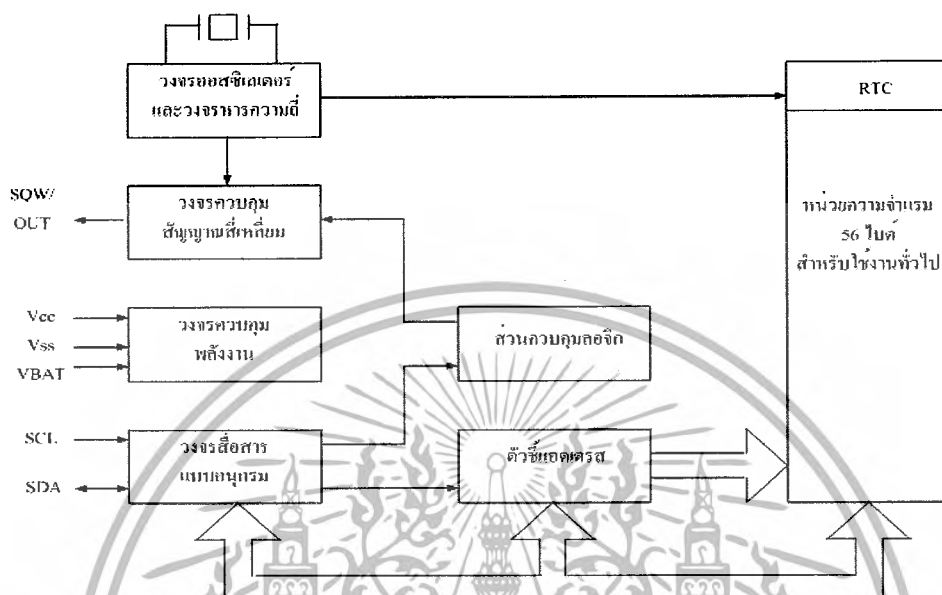
SDA, SCL (ขา 5 และ 6) เป็นขาสำหรับเชื่อมต่อกับไมโครคอนโทรลเลอร์บนระบบ I^2C SQW/OUT (ขา 7) ที่ขา 7 นี้จะมีสัญญาณรูปสี่เหลี่ยมส่งออกมา โดยสามารถเลือกความถี่ได้ 1Hz, 4.096 kHz, 8.192kHz และ 32kHz ในการใช้งานต้องต่อตัวต้านทาน 1k พูลอัพที่ขา 7 นี้ด้วย X1, X2 (ขา 1 และ 2) ใ้ต่อกับคริสตอลความถี่มาตรฐาน 32.768 kHz เพื่อให้เป็นฐานเวลาในการสร้างค่าเวลาจริงในการใช้งานต้องต่อคริสตอลเข้ากับขาทั้งสองนี้และที่แต่ละขาต้องต่อตัวเก็บประจุค่าต่างๆ ประมาณ 15pf ครอบงับขากราวด์ด้วย

การทำงานของ DS1307

ไอซี DS1307 จัดการเชื่อมต่อในแบบบัส I^2C โดยจะทำงานเป็นอุปกรณ์เลฟเฟอเมอ ดังนั้นการติดต่อเพื่อใช้งานจึงต้องกำหนดรูปแบบตามที่กำหนดไว้ในกรณีการติดต่อแบบ I^2C ในรูปที่ 2-2 แสดงส่วนประกอบหลักที่สำคัญและไดอะแกรมการทำงานของ DS1307 วงจรออสซิลเลเตอร์ถือเป็นหัวใจหลักของไอซี เนื่องจากเป็นจุดเริ่มต้นของการสร้างข้อมูลเวลาจริง ในขณะที่ DS1307 ทำงานที่ขา SQW/OUT จะมีสัญญาณพัลส์สี่เหลี่ยมส่งออกมาตลอดเวลาในกรณีที่การอินเวสจอร์กำหนดสัญญาณพัลส์ที่รีจิสเตอร์ควบคุม ค่าความถี่ของสัญญาณนี้สามารถเลือกได้ 4 ค่าคือ 1 Hz, 4.096kHz, 8.192kHz และ 32kHz พร้อมกันนั้นก็จะมีกรเก็บค่าของเวลาไว้ในหน่วยความจำอนโวลตาไทล์แรม ซึ่งมีขนาดรวม 64 ไบต์แต่จัดสรรให้ใช้เก็บข้อมูลเวลา 3 ไบต์ และเป็นหน่วยความจำสำหรับเก็บข้อมูลทั่วไปสำหรับผู้ใช้งานอีก 56 ไบต์

วงจรควบคุมพลังงานไฟฟ้าจะคอยตรวจสอบสถานะของไฟเลี้ยงไอซี หากไฟเลี้ยงต่ำกว่า $1.25 \times V_{BAT}$ ก็จะควบคุมให้ DS1307 หยุดการทำงานรีเซตค่าตัวนับแอดเดรสภายในทำให้ไม่สามารถติดต่อกับ DS1307 ได้ ดังนั้นในการใช้งาน DS1307 ต้องมีตัวระงับอย่าให้ไฟเลี้ยงตกต่ำกว่า $1.25 \times V_{BAT}$ หรือประมาณ 3.75V ในกรณีที่ใช้ VBAT เท่ากับ 3V

ถ้าหากไฟเลี้ยงมีค่าต่ำกว่า VBAT ไอซี DS1307 จะเข้าสู่โหมดสำรองข้อมูลกระแสต่ำทันที จะไม่มีการส่งสัญญาณพัลส์ออกมาที่ขา SQW/OUT แต่วงจรสร้างฐานเวลายังคงทำงานเพื่อให้ค่าของเวลาเดินไปอย่างไม่มีผิดพลาด เมื่อมีไฟเลี้ยงปรากฏขึ้นอีกครั้ง DS1307 ก็จะสามารถให้ค่าของเวลาที่เป็จริงแก่ผู้ใช้งานได้ต่อไป



รูป 2.25 โครงสร้างภายในของไอซีรีลไทม์คล็อกเบอร์ DS1307

วงจรสื่อสารอนุกรมภายใน DS1307 ได้รับการกำหนดให้ทำงานตามรูปแบบของบัส I^2C เป็นช่องทางการสื่อสารระหว่าง DS1307 กับอุปกรณ์มาสเตอร์ ผู้ใช้สามารถเข้าถึงหน่วยความจำที่ใช้เก็บค่าเวลาและหน่วยความจำใช้งานทั่วไปได้โดยการเขียนข้อมูลตามรูปแบบที่กำหนดในระบบบัส I^2C

การจัดสรรหน่วยความจำใน DS1307

ในรูปที่ 2.26 แสดงการจัดสรรพื้นที่ของหน่วยความจำภายใน DS1307 พื้นที่ 7 ไบต์แรกตั้งแต่แอดเดรส 00H-06H เป็นพื้นที่ของรีจิสเตอร์ค่าเวลาใช้ในเก็บข้อมูลเกี่ยวกับเวลา ไบต์ต่อมาที่แอดเดรส 07 เป็นพื้นที่ของรีจิสเตอร์ควบคุมการทำงานของ DS1307 ในรูปที่ 2.26 แสดงรายละเอียดของรีจิสเตอร์ค่าเวลาและรีจิสเตอร์ควบคุมของ DS1307

ด้วยการจัดสรรพื้นที่แบบนี้ ทำให้ผู้ใช้งานสามารถเรียกข้อมูลเวลาออกมาได้ตามที่ต้องการโดยไม่จำเป็นต้องอ่านออกมาทั้งหมดก็ได้ ค่าของเวลาทั้งหมดจะอยู่ในรูปของเลขฐานสิบ สำหรับการแสดงเวลาในรูปของชั่วโมง สามารถเลือกได้ว่าต้องการแบบ 12 หรือ 24 ชั่วโมง โดยกำหนดที่บิต 6 ของแอดเดรส 02H และเมื่อเลือกแบบ 12 ชั่วโมง ที่บิต 5 ในแอดเดรสเดียวกันจะใช้ในการแสดงค่า AM/PM

โดยถ้าบิตนี้เป็น “1” หมายถึง ค่าชั่วโมงในขณะนี้เป็นเวลาหลังเที่ยงวัน ในกรณีที่ เป็นแบบ 24 ชั่วโมง บิตนี้จะใช้ในการแสดงค่า 2 ของหลักสิบในหน่วยชั่วโมง

00H	วินาที																			
	นาที																			
	ชั่วโมง																			
	วัน																			
	วันที่																			
	เดือน																			
	ปี																			
07H	วิเศษลอว์ควบคุม																			
08H	แรม 56 ไบต์	CH	ข้อมูลวันที่เทลลีสปี			ข้อมูลวินาที(หลักหน่วย)			00-59											
		X	ข้อมูลวันที่เทลลีสปี			ข้อมูลวินาที(หลักหน่วย)			00-59											
		X	เลขจำนวน 24 ชั่วโมง	จำนวนหลักสิบ AMPM	ห้วงชั่วโมง เทลลีสปี	ข้อมูลชั่วโมง(หลักหน่วย)			01-12 00-25											
		X	X	X	X	X	ข้อมูลวันในสัปดาห์			1-7										
		X	X	ข้อมูลวันที่หลักสิบ		ข้อมูลวินาที(หลักหน่วย)			01-28,29 01-30											
		X	X	X	ข้อมูลสิบ (หลักสิบ)	ข้อมูลเดือน(หลักหน่วย)			01-12											
			ข้อมูลปีหลักสิบ			ข้อมูลปี(หลักหน่วย)			00-99											
3FH			OUT	X	X	SQWE	X	X	RS1	RS0										

(ก)

(ข)

รูปที่ 2.26 จัดสรรหน่วยความจำแรมภายใน DS1307

2.26 รายละเอียดของรีจิสเตอร์เก็บค่าเวลาและรีจิสเตอร์ควบคุมของ DS1307

รีจิสเตอร์ควบคุม

มีแอดเดรสอยู่ที่ 07 H มีรายละเอียดของแต่ละบิตดังนี้

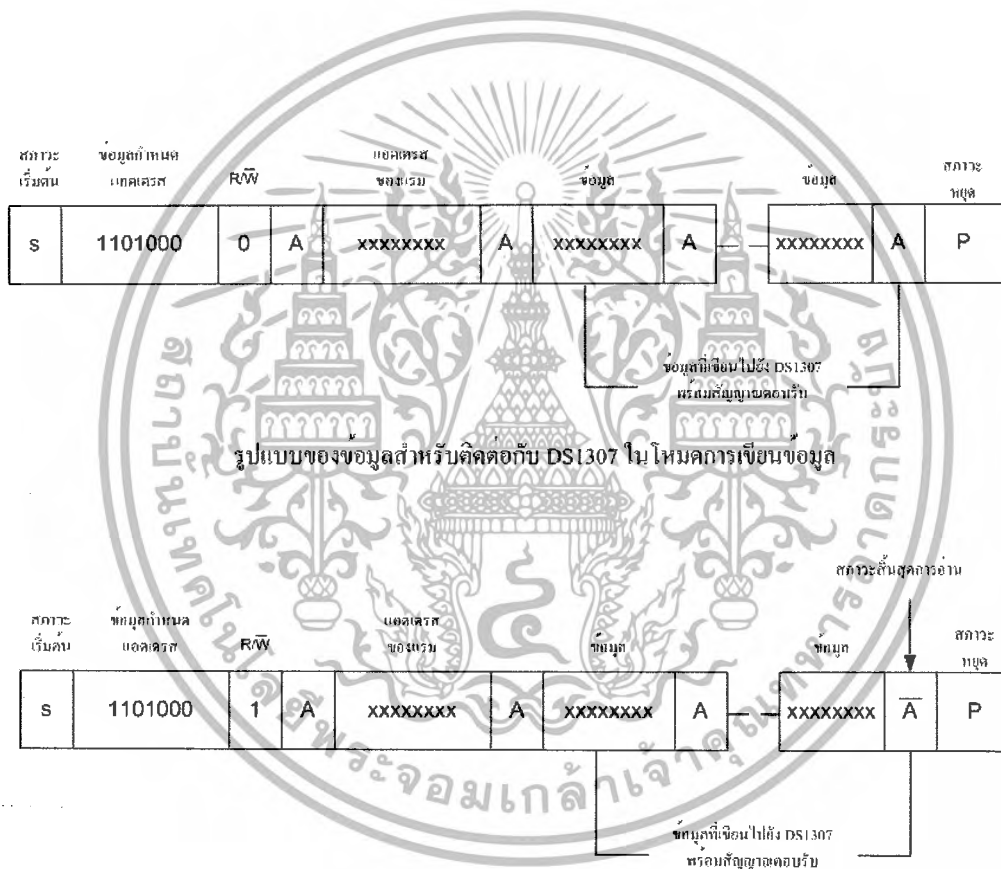
OUT (output control): ใช้ในการควบคุมระดับลอจิกที่ขา SQW/OUT ในกรณีที่คิสเซเบิลการกำหนดสัญญาณสี่เหลี่ยม โดยถ้าบิตนี้เป็น “1” ที่ขา SQW/OUT ก็จะเป็น”1” ถ้าบิตนี้เป็น”0” ที่ขา SQW/OUT ก็จะเป็น”0”

SQWE (Square Wave Enable): ใช้ในการอินาเบิลวงจรถูกกำหนดสัญญาณสี่เหลี่ยมที่ขา SQW/OUT ถ้าต้องการให้มีสัญญาณสี่เหลี่ยมออกให้กำหนดบิตนี้เป็น”1”

RS1, RS0 (Rate Select): ใช้ในการเลือกความถี่ของสัญญาณสี่เหลี่ยมที่ออกจากขา SQW/OUT ดังมีรายละเอียดต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RS1	RS0	ค่าความถี่ของสัญญาณสี่เหลี่ยม
0	0	1 Hz
0	1	4.096 kHz
1	0	8.192 kHz
1	1	32.768 kHz



รูปแบบของข้อมูลสำหรับติดต่อ DS1307 ในโหมดการเขียนข้อมูล

รูปที่ 2.27 รูปแบบของข้อมูลสำหรับติดต่อ DS1307 ในโหมดการเขียนและอ่านข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมดการทำงานของ DS1307

มีด้วยกัน 2 โหมดคือ โหมดเขียนข้อมูลและโหมดอ่านข้อมูลในการใช้งาน DS1307 ตามปกติ จะใช้งานเฉพาะโหมดอ่านข้อมูลเท่านั้น เนื่องจากไมโครคอนโทรลเลอร์จะติดต่อกับ DS1307 เพื่ออ่านข้อมูลของเวลาไปใช้งาน โหมดการเขียนข้อมูลจะถูกใช้งานก็ต่อเมื่อต้องการตั้งค่าเวลาใหม่และต้องการเขียนข้อมูลลงในหน่วยความจำใช้งานทั่วไป อย่างไรก็ตามเมื่อเริ่มต้นติดต่อกับ DS1307 จำเป็นอย่างยิ่งที่จะต้องเข้าสู่โหมดการเขียนข้อมูลก่อนเพื่อกำหนดแอดเดรสที่ต้องการอ่านข้อมูล จากนั้นจึงเปลี่ยนโหมดการทำงานมาเป็นโหมดการอ่านข้อมูลต่อไป

โหมดการเขียนข้อมูล

มีรูปแบบดังในรูปที่ 2.27 เริ่มต้นเมื่อไมโครคอนโทรลเลอร์ ทำการกำหนดสถานะเริ่มต้น (START: S) จากนั้นส่งข้อมูลกำหนดแอดเดรส 1101000 ตามด้วยข้อมูลเลือกการเขียน นั่นคือค่า 0 จากนั้นจะรอการตอบรับจาก DS1307 ขั้นตอนต่อมาคือ ส่งข้อมูลเพื่อเลือกแอดเดรสที่ต้องการเขียน จากนั้นรอการตอบรับจาก DS1307 เมื่อมีการตอบรับมาเรียบร้อย ก็เริ่มทยอยเขียนข้อมูลลงไป

ครั้งละแอดเดรส หลังจากเขียนข้อมูลในแต่ละแอดเดรส จะต้องหยุดรอการตอบรับจาก DS1307 ทุกครั้ง จึงจะสามารถเขียนข้อมูลต่อไปได้ เมื่อเขียนเรียบร้อยแล้วให้ส่งสถานะหยุด (STOP: P) เป็นอันสิ้นสุดกระบวนการเขียนข้อมูล

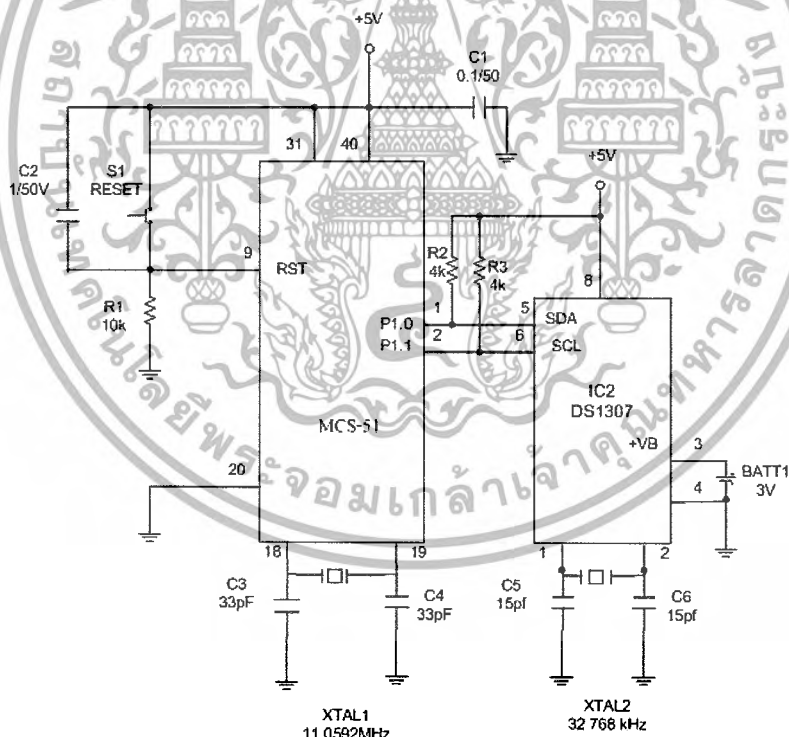
โหมดการอ่านข้อมูล

มีรูปแบบแสดงในรูปที่ 2.27 เริ่มต้นการทำงานเหมือนกับโหมดการเขียนข้อมูลคือ ไมโครคอนโทรลเลอร์กำหนดสถานะเริ่มต้นแล้วส่งข้อมูลกำหนดแอดเดรสตามด้วยข้อมูลเลือกการอ่าน ซึ่งเท่ากับ 1 จากนั้นรอการตอบรับจาก DS1307 เมื่อตอบรับเรียบร้อย DS1307 จะทยอยส่งข้อมูลออกมาให้ไมโครคอนโทรลเลอร์คราวละ 1 แอดเดรสหรือ 1 ไบต์โดยแอดเดรสที่เลือกอ่านข้อมูลจะต้องมีการกำหนดมาก่อนล่วงหน้าด้วยโหมดการเขียนข้อมูล วิธีการง่ายๆคือ เข้าสู่โหมดการเขียนข้อมูลก่อน เมื่อถึงจังหวะที่ต้องเขียนข้อมูล ให้ทำการสร้างสถานะเริ่มต้นและส่งข้อมูลกำหนดแอดเดรสใหม่อีกครั้งตามด้วยเลือกโหมดการอ่านข้อมูล ข้อมูลที่ออกมาจาก DS1307 ก็จะเป็นข้อมูลจากแอดเดรสที่กำหนดไว้ก่อนหน้านี้

การเชื่อมต่อ DS13071 กับไมโครคอนโทรลเลอร์ MCS-51

มีตัวอย่างวงจรแสดงในรูปที่ 2-6 จะเห็นได้ว่ามีลักษณะการต่อเหมือนกับอุปกรณ์บนระบบบัส I^2C ตัวอื่นๆ ทุกประการ และสามารถที่จะต่อไอซีทั้งหมดร่วมกันบนสาย SDA และ SCL ได้ เป็นการย่ำให้เห็นถึงความสามารถพิเศษของระบบบัส I^2C ที่ผู้ใช้งานสามารถเชื่อมต่ออุปกรณ์ที่มีความต่างกัน ในหน้าที่การทำงานบนสายสัญญาณเดียวกันได้ ถึงการทดลองนี้ผู้ใช้งานสามารถเชื่อมต่ออุปกรณ์ระบบบัส I^2C ได้ถึง 3 ตัว 3 ลักษณะการทำงาน โดยใช้สายสัญญาณเพียง 2 เส้น

จากวงจรในรูปที่ 2.29 ไอซี DS1307 จำเป็นจะต้องต่อแบตเตอรี่ไว้ตลอดเวลาไม่ว่าจะใช้งานหรือไม่ ทั้งนี้เพื่อรักษาการทำงานของวงจรภายใน DS1307 ให้ยังคงทำงานต่อเนื่องไป เมื่อไอซี ไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูล ก็จะได้ข้อมูลเวลาที่เป็จริงตลอดเวลา



รูปที่ 2.28 แสดงการเชื่อมต่อไมโครคอนโทรลเลอร์ MCS-51 กับไอซีรีลไทม์คล็อก DS1307

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานไอซีสร้างฐานเวลาจริงหรือรีลไทม์คล็อก(RTC) ในการติดต่อสื่อสารแบบนี้อาจใช้ อาร์เอส 232C ซึ่งเป็นมาตรฐานที่นิยมมากในการติดต่อสื่อสารข้อมูลระหว่างคอมพิวเตอร์ โดยมีสายในการติดต่อสื่อสาร 25 เส้นระหว่างการติดต่อสื่อสารคือทีอี(DTE)และ(DCE)ซึ่งสายแต่ละเส้นมีหน้าที่ติดต่อดังต่อไปนี้จากทีอีเข้าเครื่องไมโครคอมพิวเตอร์(PC) และดีซีอี เช่น โมเด็ม(MODEM) ซึ่งอุปกรณ์ทั้งสองจะไม่ส่งข้อมูลถ้าอุปกรณ์อีกเครื่องไม่ต้องการรับข้อมูลเมื่อดีซีอีเชื่อมต่อเข้ากับโครงข่าย (NETWORK)ซึ่งมี ดีทีอี เชื่อมต่ออยู่กับดีซีอี ต้องรู้ว่าเมื่อไหร่ที่ ดีทีอี พร้อมในการส่งข้อมูลประกอบเมื่อ ดีทีอี ส่งสัญญาณ ดีทีอาร์ (DTR: Data Terminal Ready ที่เวลา t1)ต่อมา ดีซีอี จะตอบสนองต่อสัญญาณโดยเชื่อมต่อเข้ากับโครงข่าย ต่อมา ดีซีอี จะส่งสัญญาณดีเอสอาร์ (DSR : Data Set Ready) เพื่อตอบกลับไปว่าพร้อมสำหรับรับข้อมูลที่ตามเวลา2ตอนนี้อุปกรณ์ทั้งสองพร้อมในการติดต่อสื่อสารกันแล้วหากดีทีอีขออนุญาตในการส่งข้อมูลให้แก่ ดีซีอี มันก็จะส่ง สัญญาณ อาร์ทีเอส(RTS : Request To Send) ช่วงนี้ระบบจะมีทิศทางการสื่อสารเป็นแบบ ฮาร์ฟดูเพล็กซ์(Half-duplex) ในการตอบรับสัญญาณอาร์ทีดี นั้น ดีซีอี จะเข้าสู่โหมด ทรานสมิต (Transmit) คือจะเป็นการบอกว่าพร้อมที่จะส่งข้อมูลออกไปตามสายสัญญาณ(TD : Transmitted Data) เมื่อ ดีซีอี ตรวจพบสัญญาณเข้ามาจากโครงข่ายมันจะส่งสัญญาณ ดีซีดี (DCD: incoming signal) ดีซีอีจะส่งสัญญาณเหล่านี้ไปยัง ดีทีอี โดยใช้สายสัญญาณอาร์ดี (RD: Received Data) โดยทั่วไปตัวเชื่อมต่อ กับ อาร์เอส232 จะมักใช้ไม่ถึง 25 เส้น ซึ่งอาจใช้เพียง 8-9 เส้นหรือน้อยกว่านั้น ซึ่งเพียงพอที่จะนำไปใช้งานแล้ว

มาตรฐานอาร์เอส232 สามารถส่งข้อมูลได้ 20,000 บิตต่อวินาที โดยระยะทางสายประมาณ 50 ฟุต

2.9 มาตรฐานการสื่อสารข้อมูลอนุกรมแบบอาร์เอส 485

อาร์เอส485เป็นมาตรฐานการสื่อสารข้อมูลแบบสมดุลพัฒนามาจากมาตรฐานอาร์เอส 422A(RS422A) เพื่อให้ตัวรับและตัวส่งจำนวนมากคู่ สามารถใช้คู่สายในการรับส่งสัญญาณร่วมกันได้ (multipoint multiple drivers and receivers) ซึ่งในกรณีของอาร์เอส 422A คู่สายสัญญาณสามารถรับส่งได้หนึ่งคู่ จะมีตัวรับได้ไม่เกิน 10 ชุดและมีตัวส่งเพียงหนึ่งชุด แต่ในกรณีอาร์เอส 485 สามารถใช้ตัวรับ 32 ชุด และตัวส่ง 32 ชุด ร่วมกันได้ภายในคู่สายสัญญาณ 1 คู่โดยทั่วไปอาร์เอส 485 มีคุณลักษณะเฉพาะทางไฟฟ้าของตัวรับและตัวส่งคล้ายตัวรับและตัวส่งของอาร์เอส 422A และไม่จำกัดรูปแบบของโปรโตคอลที่จะนำไปใช้งานกับระบบที่พัฒนาขึ้น โดยขึ้นอยู่กับผู้พัฒนาระบบเองว่าจะเลือกใช้

โปรโตคอลแบบไหนมาใช้ นอกจากนี้ตัวรับและตัวส่งมีราคาไม่สูง ทำให้ อาร์เอส 485 ถูกนำมาประยุกต์ใช้งานกันอย่างแพร่หลายสำหรับการเชื่อมต่อ

คุณลักษณะเฉพาะของอาร์เอส 485

คุณลักษณะของตัวส่ง

- ตัวส่ง 1 ตัวสามารถขับโหลดได้ถึง 32 ชุด (ตัวรับ 1 ตัว ตัวส่ง 1 ตัว)และค่าความต้านทานรวมระหว่างคู่สายมากกว่า 60 โอห์ม
 - เอาต์พุตของตัวส่งในสภาวะออฟ มีกระแสรั่วไหลไม่เกิน 100 ไมโครแอมป์ในช่วงแรงดันไฟฟ้าโหมคร่วมระหว่าง -7 โวลต์ ถึง 7 โวลต์
 - เอาต์พุตของตัวส่ง ให้ แรงดันไฟฟ้าเอาต์พุต 1.5 โวลต์ ถึง 5 โวลต์ ในช่วงแรงดันไฟฟ้าโหมคร่วมระหว่าง ค่า -7 โวลต์ถึง 12 โวลต์
 - ตัวส่งมีวงจรป้องกันตนเองที่ส่วนเอาต์พุต ในกรณีที่ตัวส่งหลายๆตัวส่งข้อมูลออกมาพร้อมๆกัน
- #### คุณลักษณะของตัวรับ
- ค่าความต้านทานที่อินพุตมีค่าสูง โดยมีค่าไม่น้อยกว่า 12 กิโลโอห์ม
 - ตัวรับมีค่าแรงดันไฟฟ้าอินพุตโหมคร่วม ระหว่าง -7 โวลต์ถึง 12 โวลต์
 - ตัวรับสามารถตอบสนองต่อสัญญาณที่แตกต่างจากสัญญาณ โหมคร่วม ได้ ± 200 มิลลิโวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

โครงสร้าง และการออกแบบระบบ

โครงสร้างของระบบควบคุมการเข้าออกโดยใช้การสแกนลายนิ้วมือนั้นจะนำไปใช้ในการควบคุมการใช้งานของเครื่องคอมพิวเตอร์ โดยในการออกแบบระบบของเรานั้นได้ทำการแยกเป็นส่วนๆคือ

3.1 ส่วนประมวลผลกลางและจัดเก็บข้อมูล

ในการทำงานของระบบที่ได้ออกแบบไว้นั้นซึ่งมีทั้ง หน่วยตรวจสอบและแปลงสัญญาณจากการกดรหัส หน่วยตรวจสอบและแปลงสัญญาณจากลายนิ้วมือ หน่วยตัด-จ่ายกำลังไฟฟ้านั้นทั้งหมดเหล่านี้จะถูกควบคุมโดยหน่วยประมวลผลกลาง โดยในการทำงานของหน่วยประมวลผล กลางนั้นจะสามารถแบ่งออกได้เป็น 2 โหมดการทำงานสำคัญคือ

โหมดการประมวลผล

โดยในโหมดของการประมวลผลนี้ตัวประมวลผลของเรา (ซึ่งในที่นี้คือ ไอซีไมโครคอนโทรลเลอร์ตระกูล MCS-51) จะทำงานร่วมกันกับหน่วยตรวจสอบและแปลงสัญญาณจากการกดรหัส หน่วยตรวจสอบและแปลงสัญญาณจากลายนิ้วมือ หน่วยตัด-จ่ายกำลังไฟฟ้า โดยจะทำงานเริ่มจากการรอรับสัญญาณจาก หน่วยตรวจสอบและแปลงสัญญาณจากการกดรหัส เพื่อตรวจสอบว่าเป็นรหัสของใครใช้ของผู้ที่ลงทะเบียนไว้หรือไม่ หากตรวจสอบแล้วว่าใช่ก็จะส่งสัญญาณต่อไปที่หน่วยตรวจสอบและแปลงสัญญาณจากลายนิ้วมือ เพื่อเป็นการส่งค่าของรหัสตำแหน่งของลายนิ้วมือของเจ้าของรหัส (ค่าของตำแหน่งได้มาจากการประมวลผลจากข้อมูลการกดรหัส) ซึ่งหน่วยตรวจสอบและแปลงสัญญาณจากลายนิ้วมือก็จะทำการอ่านค่าของลายนิ้วมือ แล้วนำมาเปรียบเทียบกับลายนิ้วมือที่ได้บันทึกไว้ของเจ้าของรหัส ว่าเป็นคนคนเดียวกันหรือไม่จากนั้นก็ส่งสัญญาณกลับมาที่โดยหน่วยประมวลผลกลางหากเป็นคนคนเดียวกันหน่วยประมวลผลกลางก็จะรอรับสัญญาณจากคีย์แพด ซึ่งจะเป็นการระบุว่าผู้ใช้ต้องการที่จะใช้เครื่องคอมพิวเตอร์เครื่องใดเมื่อ โดยหน่วยประมวลผลกลาง ได้รับข้อมูลแล้วก็จะนำมาประมวลผลแล้วจึงส่งสัญญาณเปิดประตูและสัญญาณเพื่อทำการจ่ายกำลังไฟฟ้าให้แก่หน่วยตัด-จ่ายกำลังไฟฟ้า เพื่อควบคุมการเปิดประตู และเปิดเครื่องคอมพิวเตอร์เครื่องที่ต้องการใช้งานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมดการจับเก็บข้อมูล

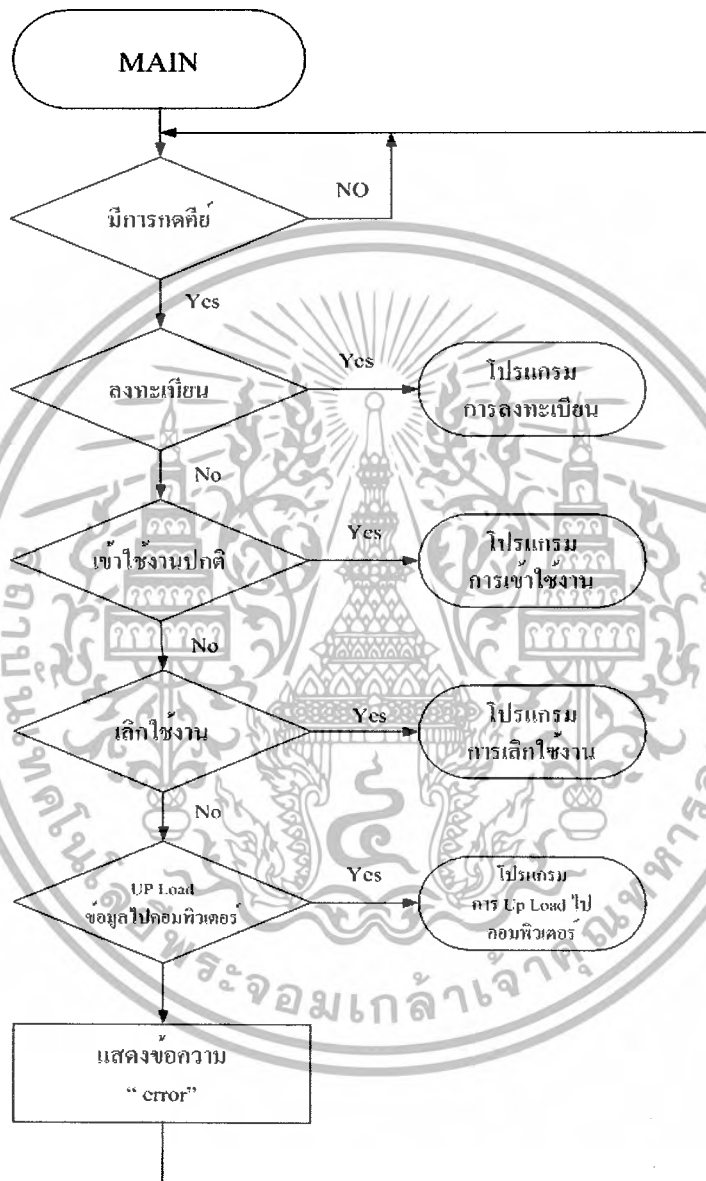
โดยในโหมดของการจับเก็บข้อมูลนั้นหน่วยประมวลผลกลางจะทำการบันทึกข้อมูลในแต่ละครั้งว่ามีผู้ใดเข้ามาใช้เครื่องคอมพิวเตอร์เครื่องใด เวลาใดโดยจะทำการบันทึกข้อมูลไว้ในหน่วยความจำชั่วคราวภายนอก (RAM) โดยข้อมูลที่จะนำมาบันทึกนั้นก็จะได้มาจากการประมวลผลของข้อมูลที่มาจากหน่วยตรวจสอบและแปลงสัญญาณจากบัตรแม่เหล็ก หน่วยตรวจสอบและแปลงสัญญาณจากลายนิ้วมือ และไอซีสร้างฐานเวลาจริง หรือ ไอซี เรียลไทม์คล็อก (RTC) เบอร์ DS1307 โดยในหน่วยประมวลผลกลางและจับเก็บข้อมูลนี้เราจะใช้อุปกรณ์คือ ไอซีไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ AT89S8252 ซึ่งเป็นไมโครคอนโทรลเลอร์ ขนาด 8 บิต เป็นอุปกรณ์ประมวลผล โดยมีอุปกรณ์ต่อรวมอื่นๆเช่น ไอซีบัฟเฟอร์ ไอซีสร้างฐานเวลาจริง หรือ ไอซี เรียลไทม์คล็อก (RTC) เบอร์ DS1307 หน่วยความจำชั่วคราวภายนอก (RAM) และในการใช้งานนั้นจะต้องมีส่วนสำคัญอีกส่วนคือ ส่วนของการแสดงผล เพื่อแสดงผลของข้อมูลต่างๆที่ถูกบันทึกไว้ โดยจะต้องมีการติดต่อกันระหว่าง เครื่องคอมพิวเตอร์ที่จะแสดงผลกับ หน่วยประมวลผลกลางและจับเก็บข้อมูล ของเรา โดยจะทำการติดต่อกันแบบอนุกรมโดยใช้มาตรฐาน RS-232 หรือมาตรฐาน RS-485 (โดยจะพิจารณาเลือกจากระยะทางระหว่างเครื่องคอมพิวเตอร์ที่จะแสดงผลกับ หน่วยประมวลผลกลางและจับเก็บข้อมูล) โดยในส่วนของคอมพิวเตอร์ที่จะแสดงผลนั้นต้องสามารถดึงข้อมูลเข้ามาแล้วทำการแปลงข้อมูลนั้นออกมาได้ โดยจะใช้โปรแกรม วิชาการเบสิก

3.2 ส่วนควบคุมการปิด-เปิด ประตูและเครื่องคอมพิวเตอร์หรือหน่วยตัด-จ่ายกำลังไฟฟ้า

ส่วนควบคุมการปิด-เปิด ประตูและเครื่องคอมพิวเตอร์หรือหน่วยตัด-จ่ายกำลังไฟฟ้า จะมีหน้าที่ในการรับคำสั่งจากหน่วยประมวลผลกลาง เพื่อทำการตัด-จ่ายไฟฟ้า ให้แก่อุปกรณ์ไฟฟ้า ซึ่งในที่นี้ หมายถึงเครื่องคอมพิวเตอร์ และโซลินอยด์ ที่ใช้ในการปิดเปิดประตู ซึ่งในส่วนของหน่วยตัด-จ่ายกำลังไฟฟ้าจะประกอบไปด้วยอุปกรณ์หลัก ๆ คือ

-ไมโครคอนโทรลเลอร์ ตระกูล MCS-51 เบอร์ 2051 เพื่อใช้ในการรับและประมวลผลสัญญาณที่ส่ง มาจากหน่วยประมวลผลกลาง

-ทรานซิสเตอร์ และรีเลย์ เพื่อใช้ในการควบคุมแหล่งจ่ายไฟฟ้า AC 220 V. ด้วยกระแสไฟฟ้า DC 5 V.



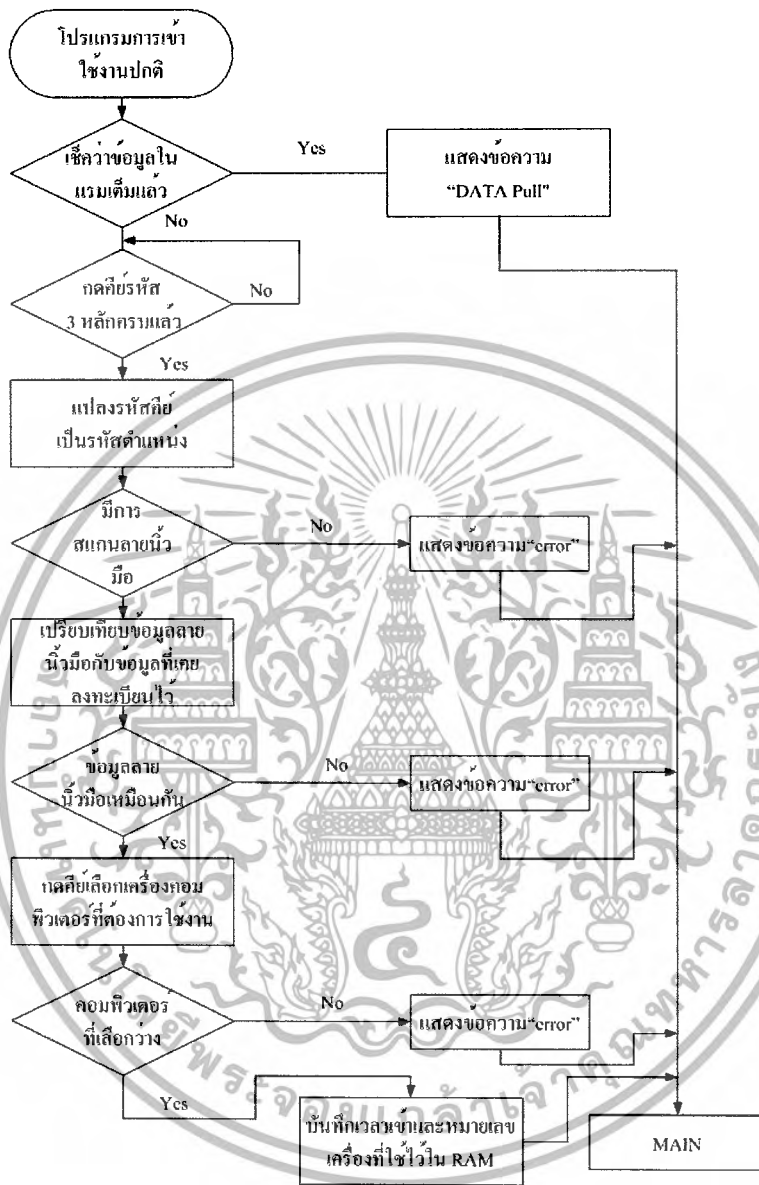
รูปที่ 3.1 MAINโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



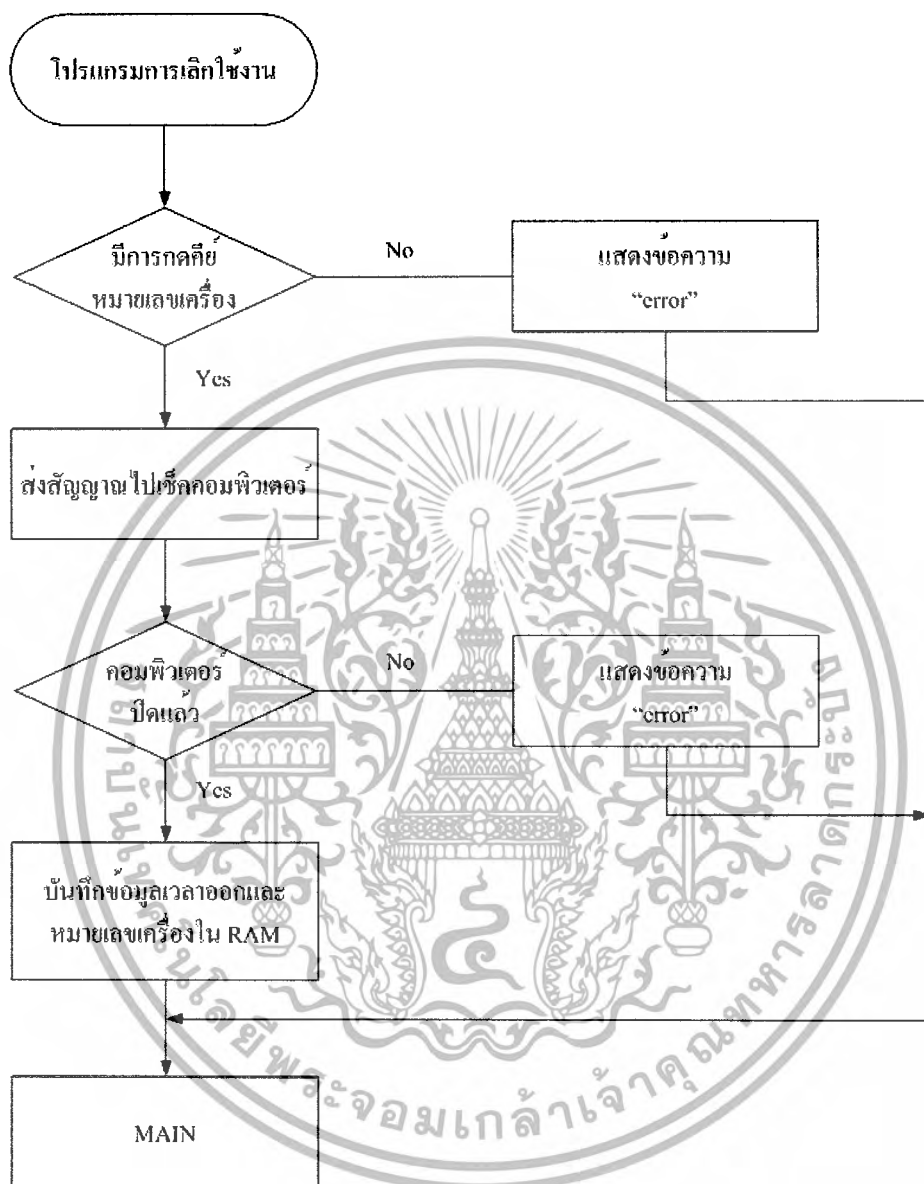
รูปที่ 3.2 โปรแกรมการลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



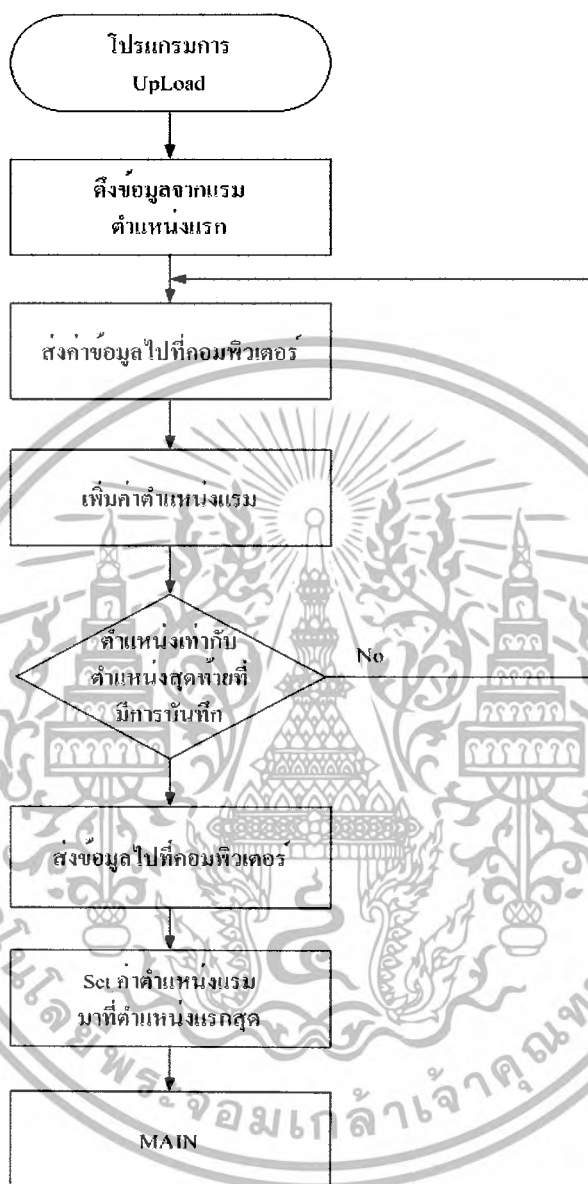
รูปที่ 3.3โปรแกรมการเข้าใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



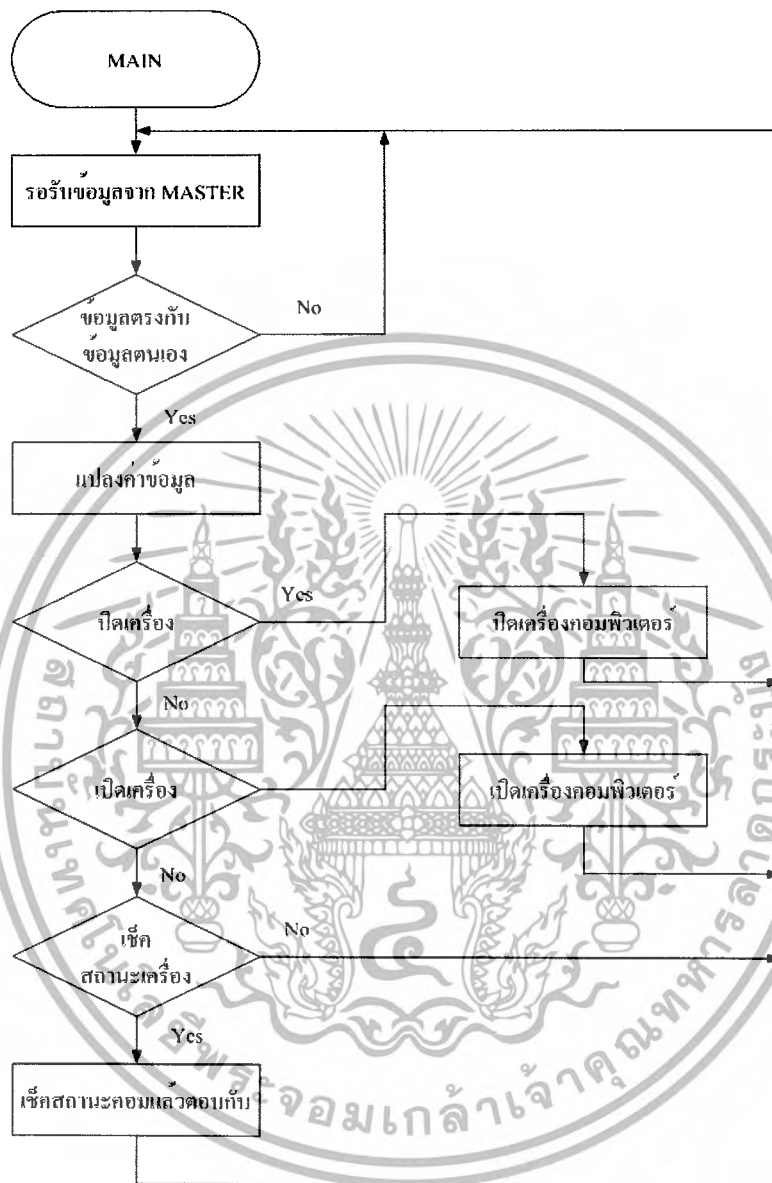
รูปที่ 3.4 โปรแกรมการเลิกใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



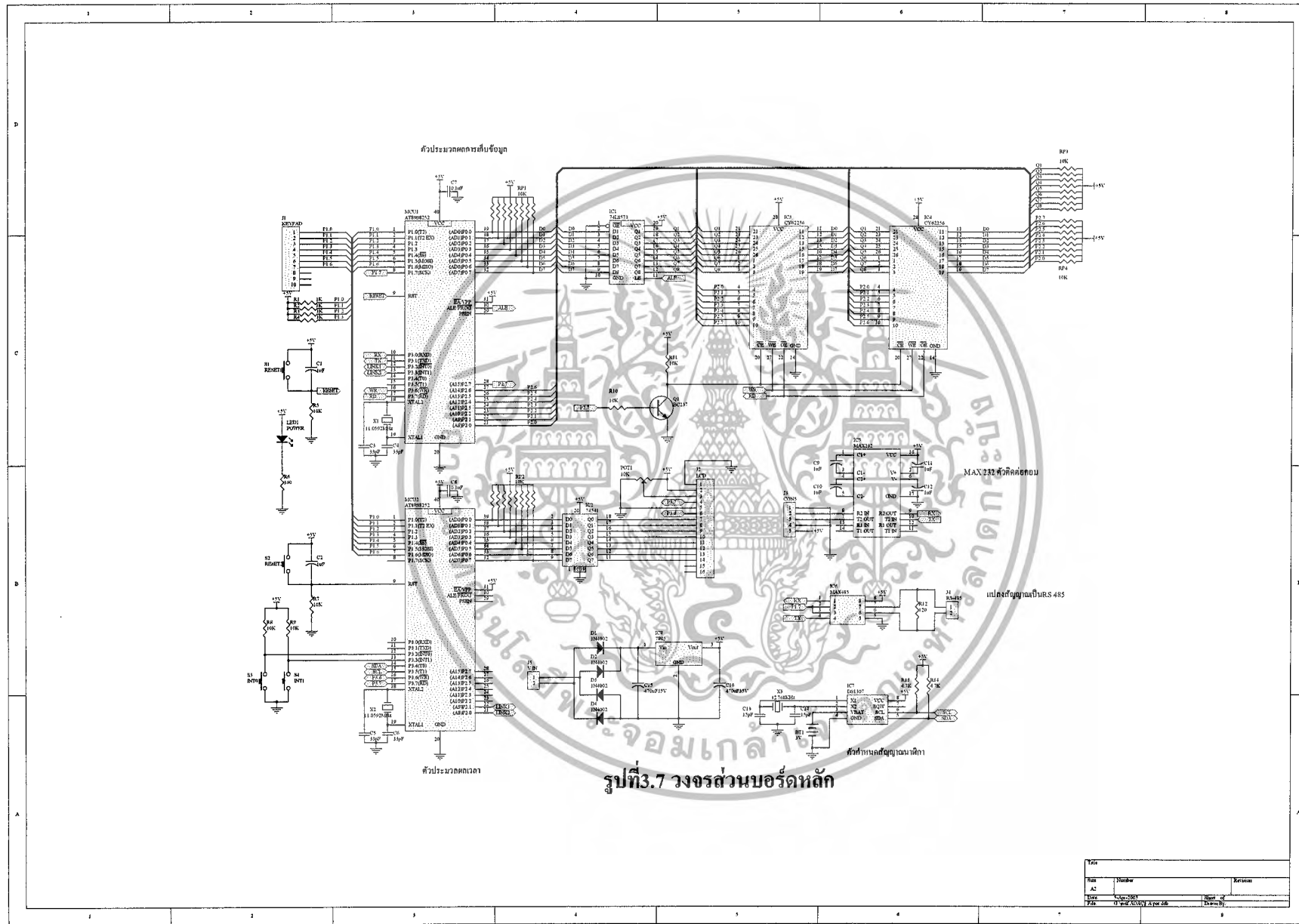
รูปที่ 3.5 โปรแกรมการ Up-load ข้อมูลไปคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



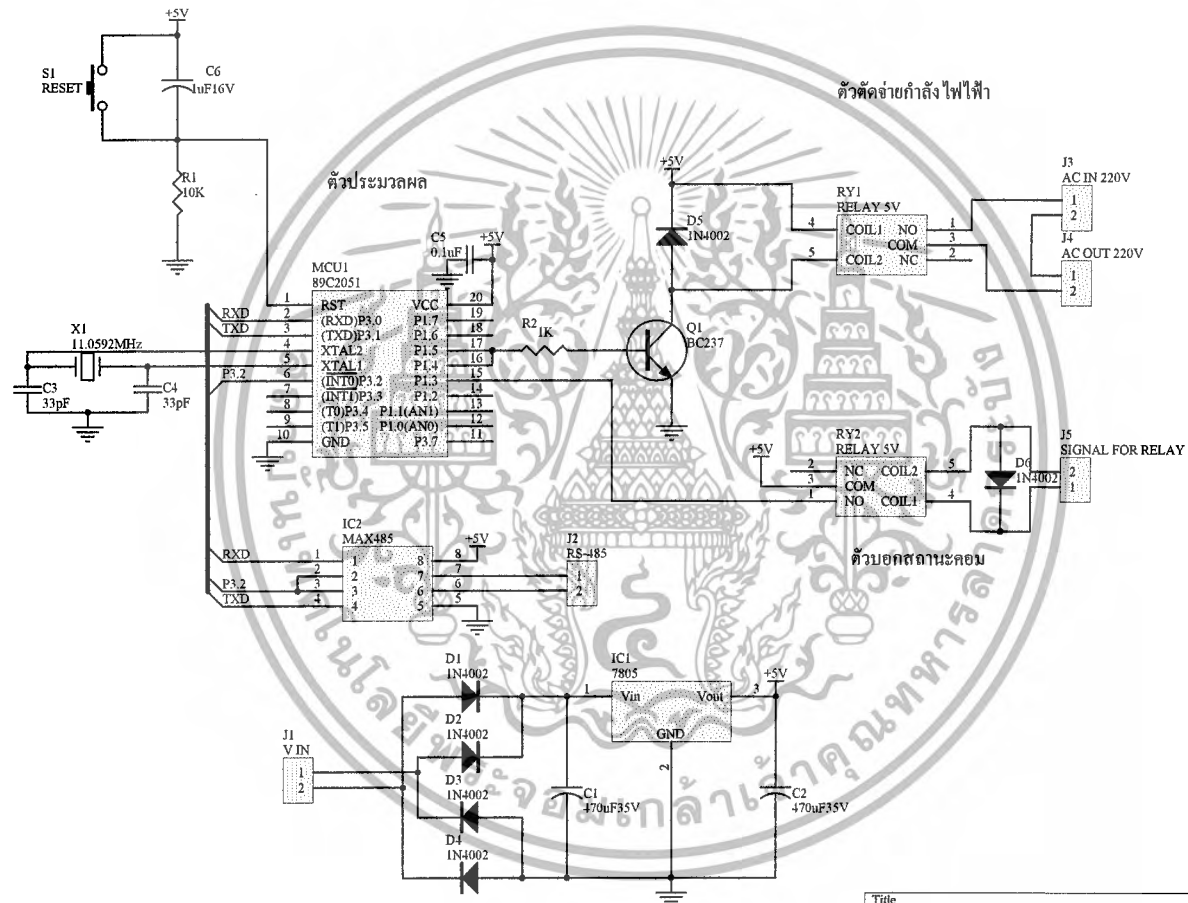
รูปที่ 3.6 โปรแกรมการทำงานส่วนตัดจ่ายกำลังไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 วงจรผ่านบอร์ดหลัก

Title		
Date	Number	Version
Author	Author	Author
Page	Page	Page



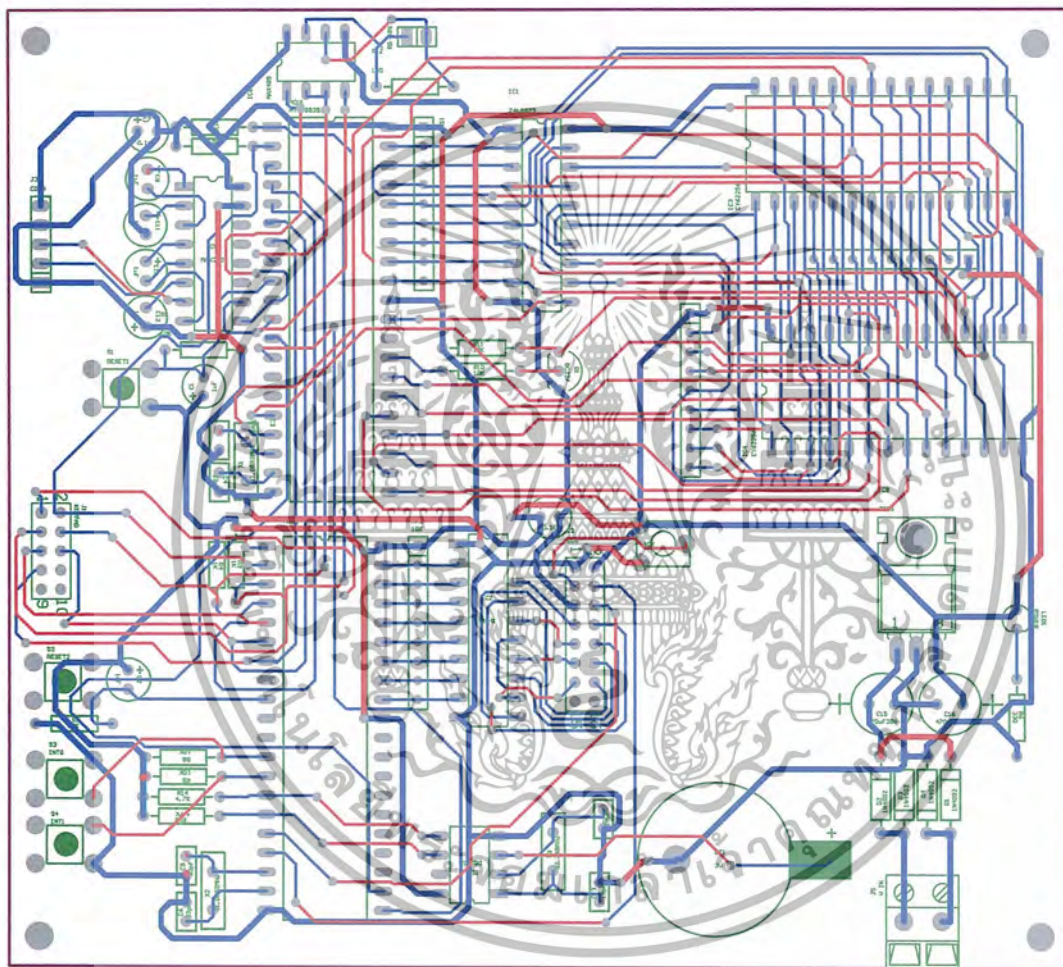
ตัวตัดจ่ายกำลัง ไฟฟ้า

ตัวประมวลผล

ตัวบอกสถานะคอม

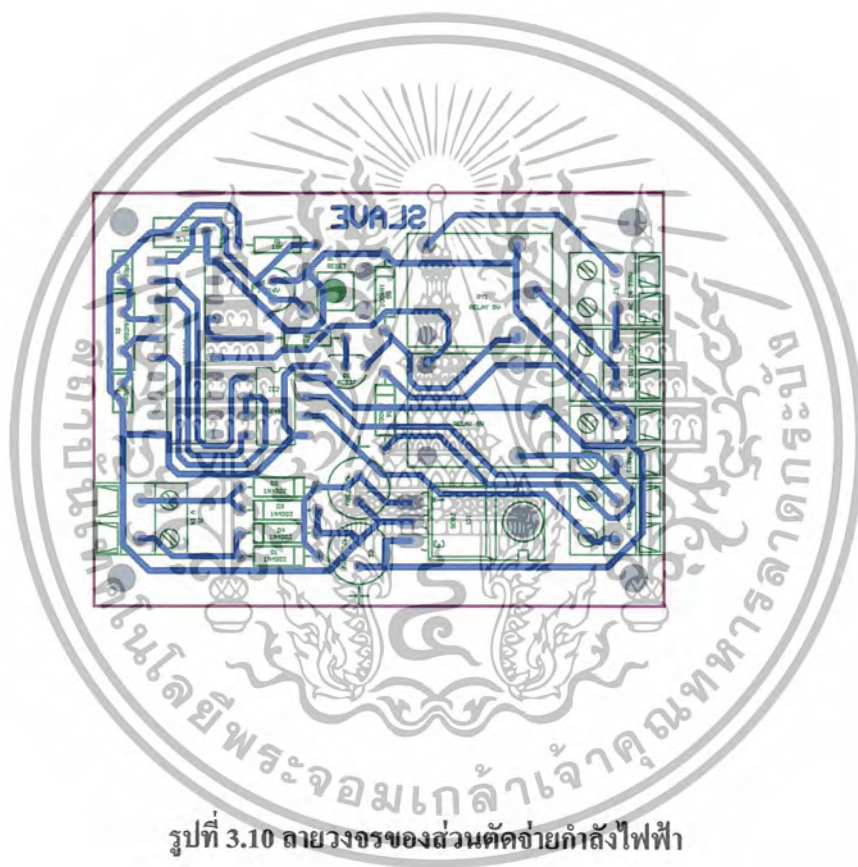
รูปที่ 3.8 วงจรส่วนตัดจ่ายกำลังไฟฟ้า

Title		
Size	Number	Revision
A4		
Date:	5-Apr-2005	Sheet of
File:	G:\gol\AOACS\A\por.ddb	Drawn By:



รูปที่ 3.9ลายวงจรของบอร์ดหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 ลายวงจรของส่วนตัดจ่ายกำลังไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

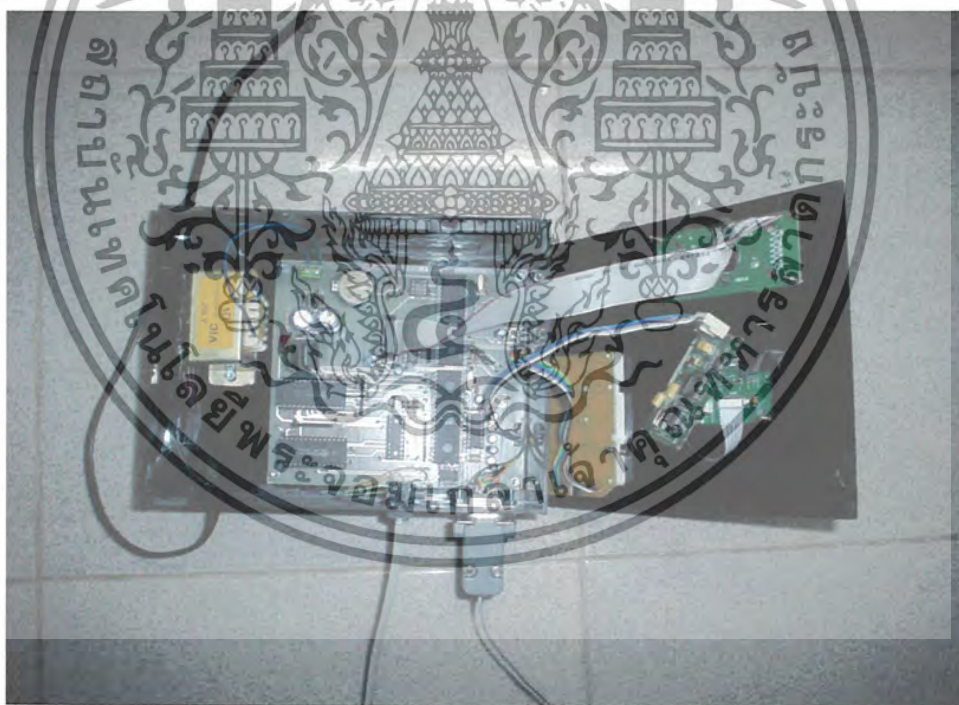
การทดลองได้ทำการทดลองโดยแยกเป็นส่วนของ ภาคจ่ายไฟ วงจรบอร์ดหลัก ส่วนตัดจ่าย กำลังงาน และในส่วนของภาคแสดงผล

4.1 การทดลองส่วนของภาคจ่ายไฟ

ทำการต่อวงจรแล้วทำการจ่ายแรงดัน 220 VAC เข้าไปแล้ววัดแรงดันเอาต์พุต ได้ 4.96 VDC

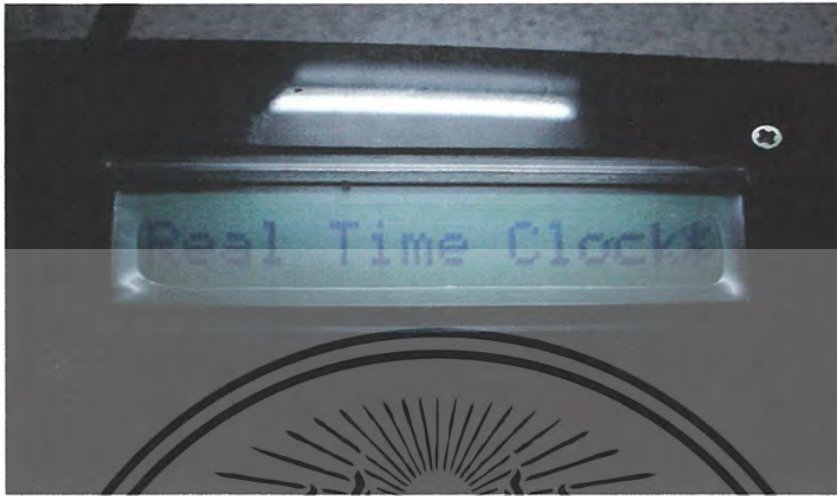
4.2 การทดลองส่วนของวงจรบอร์ดหลัก

ทำการต่อวงจรและทดลองเขียน โปรแกรมเพื่อส่งข้อมูลออกมาทางพอร์ตอนุกรมของ ไมโครคอนโทรลเลอร์แล้วมาแสดงผลที่หน้าจอดีไซน์เพื่อทำการแปลรหัสที่ได้แล้วแสดงผลออกมาทางจอภาพ



รูปที่ 4.1 วงจรส่วนของบอร์ดหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 การแสดงผลทางจอ LCD

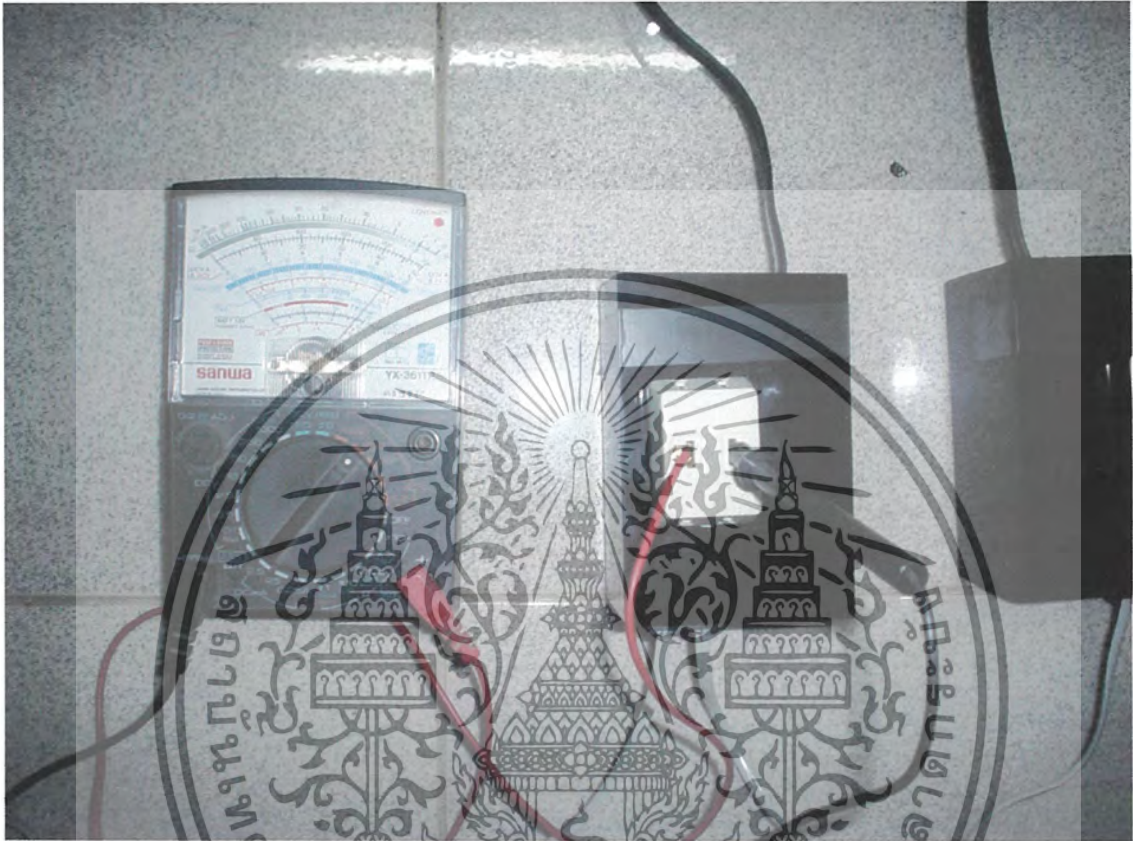
4.3 การทดลองส่วนของวงจรส่วนตัดจ่ายกำลังงานไฟฟ้า

โดยทำการทดลองเขียน โปรแกรมเพื่อรับค่าจากพอร์ตอนุกรมแล้วทำการเช็คข้อมูล ถ้าข้อมูลตรงกับข้อมูลของตัวเองก็จะทำการตัด-ต่อหน้าสัมผัสของรีเลย์ ตามคำสั่งที่ได้รับ



รูปที่ 4.3 วงจรส่วนตัด-จ่ายกำลังงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 การตัด-จ่ายกำลังงาน

4.4 การทดลองในส่วนของภาคแสดงผล

ทำการรับข้อมูลผ่านทางพอร์ตอนุกรมจากไมโครคอนโทรลเลอร์ แล้วทำการแปลงข้อมูลที่ได้ เพื่อแสดงผลที่ได้ทางจอคอมพิวเตอร์ โดยใช้โปรแกรม VISUAL BASIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการประเมิน

ผลการประเมิน

ผลการประเมิน

๔

รูป 4.5 การแสดงผลทางจอคอมพิวเตอร์

ชื่อผู้ศึกษา	ผลการเรียน	วันที่แก้ไข	เวลา
ผู้สอน ชัยรัตน์	1	22 3 05 dd / mm / yyyy	19 00
รายชื่อผู้ศึกษาที่เลือกได้คะแนนเฉลี่ยในแบบทดสอบ	ผลการเรียนทั้งหมด	ผลสัมฤทธิ์งาน	เวลาที่เลือกได้
ผู้สอน ชัยรัตน์	1	22 3 05	19 00
ผู้สอน ชัยรัตน์	2	21 3 05	17 52
ผู้สอน ชัยรัตน์	2	20 3 05	14 40
ผู้สอน ชัยรัตน์	2	19 3 05	13 01
ผู้สอน ชัยรัตน์	1	19 3 05	12 09
ผู้สอน ชัยรัตน์	1	17 3 05	10 00
ผู้สอน ชัยรัตน์	1	17 3 05	10 02

Com 1
Com 2

ผลการแก้ไขงาน

ผลการเรียน

รูป 4.6 การแสดงผลทางจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุป วิจัย ปัญหาที่พบ และการพัฒนา

บทสรุป

ในโครงการ “ระบบควบคุมการเข้าออกด้วยลายนิ้วมือ” นี้ นั้น ไม่ใช่แค่การควบคุมการเปิด-ปิด ประตูเท่านั้นแต่เราได้พัฒนาเพื่อใช้ในการควบคุมการเปิด-ปิดเครื่องคอมพิวเตอร์ที่อยู่ในโครงข่ายของเราได้ด้วย โดยวัตถุประสงค์หลักคือเพื่อความปลอดภัยของทรัพย์สินและข้อมูลที่อยู่ในเครื่องคอมพิวเตอร์นั้นๆ โดยผู้ที่มีสิทธิ์เข้าใช้งานเครื่องคอมพิวเตอร์นั้นต้องเป็นบุคคลที่มีสิทธิ์และได้รับอนุญาตเท่านั้น ซึ่งหากเกิดปัญหาในการใช้งานหรือมีปัญหาเกี่ยวกับทรัพย์สินเราก็สามารถตรวจสอบได้โดยง่าย เพราะจะมีการบันทึกว่าใครเข้ามาใช้เวลาใดและใช้เครื่องคอมพิวเตอร์เครื่องใด ซึ่งจะทำให้ระบบของเรามีมาตรฐานในด้านของการรักษาความปลอดภัยมากขึ้นเพราะความปลอดภัยเป็นสิ่งที่สำคัญสำหรับโลกปัจจุบันนี้

บทวิจารณ์

ในระบบของเรานั้นได้นำไอซีฐานเวลาจริง (Real time clock) มาใช้งานด้วยทำให้ระบบของเราสามารถทราบ และตรวจสอบได้ในภายหลังว่ามีใครเข้ามาใช้งานคอมพิวเตอร์เมื่อใด และยังทราบอีกว่าใช้เครื่องไหน ทำให้ระบบของเรามีความสามารถที่จะตรวจเช็คและรู้ตัวผู้ที่เข้ามาใช้งานแล้วไม่ทำตามกฎระเบียบ อันพึงจะทำให้เกิดความเสียหายแก่ทรัพย์สินได้

ปัญหาที่พบ

ระบบของเราจะเกิดปัญหาขึ้นได้ ถ้ามีการใช้งานในฟังก์ชันหลัก พร้อมกัน โดยจะทำให้ไม่สามารถตอบสนองผู้ที่มาขอใช้บริการที่หลังได้ทันทีและจะทำให้เกิดความเสียหายแก่ข้อมูลได้ ยกตัวอย่างเช่น หากมีการใช้งานฟังก์ชันการอัปโหลดข้อมูลจากบอร์ดหลักไปยังคอมพิวเตอร์อยู่ แล้วมีการมาขอใช้งานการเข้า หรือออก จะทำให้มีการชนกันของข้อมูลในช่วงของการรับ-ส่งแบบอนุกรมเกิดขึ้นและในส่วนของการแสดงผลทางจอคอมพิวเตอร์นั้น ควรเพิ่มความสามารถในการจัดเก็บข้อมูล เพราะในขณะนี้ระบบของเราไม่สามารถจัดเก็บข้อมูลได้ หากมีการเรียกมาดูแล้วข้อมูลที่ถูกรบกวนมาดูก็จะถูกลบหายไป

แนวทางการพัฒนาต่อ

ควรจะมีการจัดเก็บข้อมูลไว้ในฐานข้อมูลซึ่งต้องมีการพัฒนาต่อไป และในส่วนของวงจรมันก็อาจจะทำการเพิ่มขนาดของหน่วยความจำเพื่อที่จะสามารถรองรับจำนวนผู้ใช้งานได้มากขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ก.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานเครื่องสแกนลายนิ้วมือ

การใช้งานเครื่องสแกนลายนิ้วมือ แบ่งเป็น 4 โหมดการใช้งานคือ

1. โหมดการลงทะเบียน
2. โหมดการเข้าใช้งานห้องคอมพิวเตอร์
3. โหมดการออกจากห้องคอมพิวเตอร์
4. โหมดการอัปเดตข้อมูลไปแสดงผลที่คอมพิวเตอร์

ในการใช้งานนั้นหากทำการติดตั้งใหม่แล้วทำการต่อ ไฟเข้าเครื่องครั้งแรก ต้องมีการตั้งวันที่ และเวลา ให้แก่เครื่องก่อน โดยทำการกดปุ่มการตั้งเวลาด้าน ในเครื่องแล้วกดเวลาปัจจุบันที่คีย์สวิตซ์ที่ด้านหน้าของเครื่อง โดยกด ชั่วโมง/นาฬิกา/วินาที จากนั้นกดปุ่มการตั้งวันที่ด้าน ในเครื่องแล้วกดเวลาปัจจุบันที่คีย์สวิตซ์ที่ด้านหน้าของเครื่อง โดยกด วันที่/เดือน/ปี เมื่อกดครบแล้ว ที่หน้าจอ LCD ก็จะแสดงเวลาปัจจุบันไปเรื่อยๆหากไม่มีการใช้งานในโหมดอื่นๆ

โหมดการลงทะเบียน

ในโหมดนี้เป็นการเริ่มแรกของการ ใช้งานเครื่องสแกนลายนิ้วมือ โดยปกติแล้วต้องมีการเก็บบันทึกลายนิ้วมือของสมาชิกไว้ก่อนเสมอ โดยจะต้องมาเริ่มที่โหมดนี้โดยการเก็บบันทึกลายนิ้วมือมีขั้นตอนดังนี้

- กดรหัสสำหรับการลงทะเบียน โดยรหัสนี้จะเป็นรหัสลับซึ่งควรจะเป็นเจ้าหน้าที่ที่เกี่ยวข้องเท่านั้นที่ทราบรหัสนี้ (โดยในที่นี้ได้ตั้งรหัสนี้ไว้คือ #123) หากกดรหัสผิดพลาดหน้าจอ LCD จะแสดงข้อความ ******ERROR******
- จากนั้นทำการกดรหัสประจำตัว 2 หลัก แล้วที่จอ LCD จะแสดงผลว่า “PRESS FINGER” จากนั้นทำการวางนิ้วมือลงที่แป้นสแกน หากการสแกนถูกต้องหรือสำเร็จ ที่หน้าจอ LCD จะแสดงผล ว่า **“OK*OK*OK*OK”** เป็นอันว่าสำเร็จแล้วระบบก็จะออกจากโหมดนี้ แต่หากว่าการสแกนผิดพลาดที่หน้าจอ LCD ก็จะแสดงผลว่า ******ERROR****** แล้วระบบก็จะออกจากโหมดนี้

โหมดการเข้าใช้งานห้องคอมพิวเตอร์

ในโหมดนี้จะเป็นการใช้งานตามปกติหลังจากที่ได้มีการลงทะเบียนสมาชิกแล้ว โดยโหมดนี้มีขั้นตอนการใช้งานดังนี้

- ทำการกดรหัสประจำตัวโดยกด *0xx (xx คือรหัสประจำตัว2หลักที่ลงทะเบียนไว้) หากไม่มีการผิดพลาดที่หน้าจอ LCD จะแสดงข้อความ **“PRESS FINGER”**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำการวางนิ้วมือลงที่เป็นสแกน หากการสแกนถูกต้องหรือสำเร็จและมีการเปรียบเทียบลายนิ้วมือครั้งนี้กับลายนิ้วมือที่ได้มีการลงทะเบียนไว้ หากถูกต้อง ที่หน้าจอ LCD จะแสดงผล “SELECT COMP” แต่หากผิดพลาด หรือเปรียบเทียบแล้วไม่ตรงกันหน้าจอ LCD ก็จะแสดงผลว่า “*****ERROR*****” แล้วระบบก็จะออกจากโหมดนี้
- ทำการกดรหัสเพื่อเลือกเครื่องคอมพิวเตอร์ที่ต้องการใช้งานหากเครื่องนั้นว่าง หน้าจอ LCD จะแสดงผลว่า “*****OK*****” เราก็จะสามารถเข้าใช้งานได้ต่อไป

โหมดการออกจากห้องคอมพิวเตอร์

ในโหมดนี้จะใช้เมื่อเราใช้งานคอมพิวเตอร์เสร็จเรียบร้อยแล้ว โดยมีขั้นตอนดังนี้

- กดรหัสเครื่องคอมพิวเตอร์ที่เลิกใช้งาน โดยกด 0xx (xx คือรหัสของเครื่องคอมพิวเตอร์)
- หากมีการปิดเครื่องแล้วที่หน้าจอ LCD จะแสดงผลว่า “*****OK*****” เป็นอันเสร็จเรียบร้อย

โหมดการอัปเดตข้อมูลไปแสดงผลที่คอมพิวเตอร์

ในการใช้งาน โหมดนี้แบ่งเป็น 2 ลักษณะคือ

- การอัปเดตข้อมูลการเข้าใช้งานห้องคอมพิวเตอร์
- การอัปเดตข้อมูลการเลิกใช้งานห้องคอมพิวเตอร์

โดยทั้งสองแบบมีการทำงานที่เหมือนกัน ต่างกันที่รหัสที่ต้องกดต่างกัน ซึ่งการทำงาน โหมดนี้มีจ็องระวังคือ ก่อนกดรหัส ต้องให้แน่ใจว่าที่คอมพิวเตอร์ที่ใช้แสดงผลนั้นเปิดโปรแกรมเพื่อรอรับข้อมูลอยู่หรือไม่ แล้วจึงทำการกดรหัส โดยในที่นี้ได้กำหนดรหัสไว้คือ

การอัปเดตข้อมูลการเข้าใช้งานห้องคอมพิวเตอร์ กด #987

การอัปเดตข้อมูลการเลิกใช้งานห้องคอมพิวเตอร์ กด #654

โดยข้อมูลก็จะไปปรากฏที่จอคอมพิวเตอร์ และเมื่อมีการอัปเดตข้อมูลไปเรียบร้อยแล้วนั้น ที่หน่วยความจำในเครื่องสแกนลายนิ้วมือก็จะทำการลบข้อมูลการเข้าใช้งานและการเลิกใช้งานโดยอัตโนมัติ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUPIENT



Supident Science & Technology Co., LTD.

Monday, April 04, 2005

1. Description

Commands hereof apply to TSFID-335416EB embedded suite without card and voice support. In order to strengthen the operating feasibility, in each real-time fingerprint capture operation, TSFID-335416EB will prompt “place/withdraw the finger” to the host and the application system may determine the methods of prompt (voice or text) at its own option.

2. Conventions

Command messages hereof contain HEX digits, with or without suffix of H. For instance, 1B equals 1BH (0x1B) and 05 equals 05H (0x05).

The word “host” refers to application system controlling machine (card-issuing system controlling machine or main control SCM); “TSFID” or “equipment” refers to TSFID-335416EB.

“Command” refers to a message sent from the host to TSFID to direct or inform the verification equipment how to act. “Response” refers to a message sent from TSFID to the host, and is mostly the feedback execution result after receiving a command. The command 84H is special. It is sent from the equipment to the host, prompting an operation requiring no response from the host and without leader command. It can be considered as a response just in the category of transmission direction.

In the following explanation, if multi-bytes are involved, the order of the bytes is determined by their order of being sent or received. The first sent or received byte is named the first byte, and the next the second byte, and so on.

The proposed method of implementation is that: the host sends a command message to TSFID, and TSFID returns a response message to the host after processing the command accordingly.

2.1 Command Message

	Header	Length	Command ID	Parameter	Checksum
Length	3	2	1	n	1
Value	1B 72 73				

The format of a data packet is as described in the above table. It is composed of five parts, and the length unit is byte. Data in the message is in HEX digits.

(i) Message Header

Message header marks the start of a command. It has 3 bytes. Its value is 1B 72 73 HEX.

(ii) Length

Length refers to the bytes number, include the command ID, the parameter and the checksum. It has 2 bytes, and the first byte reflects the high order of the length. For instance, length 01 00 means that the length of the message is 0100H (256).

(iii) Command ID

Command ID shows the denote which command to be executed currently. It includes one byte, describing what kind of operation to be executed.

(iv) Parameter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงแก้ไข หรือทำซ้ำโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Parameter is optional. It describes data requirements of the command.

(v) Checksum

Checksum equals to the unsigned carry-free sum from the first byte to the last byte near the checksum byte itself, one byte.

For Example:

1B 72 73 00 02 61 63

Explanation: This is a plain text command message. The first three bytes form the message header 1B 72 73, marking the start of the command. The length is 0002H, which means that there are 2 bytes in total of command, parameter and checksum. The Command ID is 61. And there is no parameter. The checksum is 1B+72+73+00+02+61=63.

2.2 Response Message

	Header	Length	Command ID	Response	Checksum
Length	3	2	1	n	1
value	1B 72 73				

The message header, length, Command ID and checksum of the response message is similar with those of the command message. The most difference lies in the response. The response usually includes the following parts:

- (i) Return code (1 byte) 00 representing operation successful
- (ii) Operation result whose existence rests with the command, and when Return code is 00, it has a practical meaning

For example:

1B 72 73 00 05 61 00 04 FF 69

Explanation: The first three bytes of the response message make the message header 1b 72 73. The length is 0005H. There are 5 bytes in total of the Command ID, response parameter and checksum. The Command ID is 61. The return code is 00, which means the operation is successful. The operation result is 04 FF. And the checksum is 69.

2.3 Data Structure

Among the provided interface commands, there are several commands that require data transmission. Therefore, we define the structures of some frequently used data parameters. The whole structure information must be offered no matter being used in command parameters or information saving.

2.3.1 Fingerprint feature Structure

In some feature extracting or template matching commands, there are usually requirements for feature transmission. Its structure is as follow:

Byte Order	Definition	description
1、 2	Length of the template, indicating the byte number of the next valid characteristic bytes.	The length does not include the length byte itself.
Rest	Minutiae characteristic value contained in the template	

The size of the whole structure is the first byte*256+the second byte+2. A template (include the length byte) is not more than 263 bytes.

For example: 00 09 02 11 12 13 14 15 16 17 18 as a feature, the valid characteristic byte number is

09H (00H*256+09H) , the minutiae characteristic value is 021112131415161718.

The size of the whole template is 11 bytes.

3. Command Set

3.1 Summary List

Command ID	Function	Note
Basic Operation Commands		
61H	Set up Connection	
62H	Reset the baud rate used in communication	
Fingerprint Operation Commands		
80H	Unconditioned scan of fingerprint images	
81H	Conditioned scan of fingerprint images	
82H	Extract features from the saved image data and upload to the host	
83H	Capture a Real-time sample, upload the features to the host.	
84* 31/32	Finger action prompts: play/withdraw the finger	One-way
85H	The features extraction during enrollment stage	
86H	The template construction during enrollment by the command 85H and upload to the host	
89H	Capture a real-time sample, and verify it against the templates downloaded as the command parameter.	
8AH	Compare the two templates download and response the compared result.	

*84H is a one-way command, sent from TSFID device to the controlling machine to indicate an operation and with no follow-up processing. When the host sends 81H, 83H, 85H and 89H commands to TSFID, it will receive a response with the Command ID 84H, prompting placing/withdrawing the finger.

3.2 Return code

In the process of command operation, different value will be returned according to different operation status of the command.

Order	Value(Hex)	Description
1	00h	The command executed successfully
2	01h	Invalid Command ID
3	02h	The parameter does not match its definition.
4	03h	Checksum error
5	30h	Failure to scan valid fingerprint image during desired time
6	31h	Enrollment failure
7	32h	Verification, the captured fingerprint is not matched with the template download.
8	33h	Failure to capture, Not enough features captured.

3.3 Command Details

3.3.1 Connect: 61H

Command ID: 61H

Function: Connect to TSFID device

The host send this command to try to connect with TSFID. Once received this command, TSFID will return a response with the parameters used in the communication to the host.

Parameter: none

Command (the host→TSFID):

	Header	Length	Command ID	Checksum
Length	3	2	1	1
Value	1B 72 73		61	

Response (TSFID→the host):

	Header	Length	Command ID	Response Parameter	Checksum
Length	3	2	1	1+[2]	1
Value	1B 72 73		61	Return code + [communication parameter]	

When Return code is 00, the communication parameter in the response parameter is valid. The communication parameter includes two bytes, and is mainly designed for serial ports. The definition is as follow:

The first byte: baud rate

Value (Hex)	Baud Rate
04	9600
06	14400
08	19200
10	38400

The second byte: reserved, without practical meaning.

Return code:

Value(Hex)	Description
00	Correct connection.
01	Invalid Command ID
02	The baud rate in this communication will adopt the default value 9600 due to incorrect parameters.
03	Checksum error

For example:

Command -- 1B 72 73 00 02 61 xx

Successful Response --1B 72 73 00 05 61 00 04 FF xx

Failed Response -- 1B 72 73 00 03 61 03 xx

3.3.2 Modify Baud Rate: 62H

Command ID: 62H

Function: Reset the baud rate of serial communication

The host uses this command to modify the baud rate used in the serial communication. After receiving the command, TSFID will return a response with the current communication setting and reset the baud rate. And later communication will adopt the new setting.

Parameter: communication parameter

Command (the host→TSFID):

	Header	Length	Command ID	Parameter	Checksum
Length	3	2	1	2	1
value	1B 72 73		62	new baud rate	

The length of the new baud rate is two byte. Refer to Command 61H for its definition.

Response (TSFID→the host):

	Header	Length	Command ID	Response	Checksum
Length	3	2	1	1	1
value	1B 72 73		62	Return code	

Return code:

Value(Hex)	Description
00	The command executed successfully
01	Invalid Command ID
02	The parameter does not match its definition
03	Checksum error

For example:

Command –1B 72 73 00 04 62 04 FF xx

Successful Response –1B 72 73 00 03 62 00 xx

Failed Response -- 1B 72 73 00 03 62 03 xx

3.3.3 Unconditioned Scan of Fingerprint Images: 80H

Command ID: 80H

Function: unconditioned scan of fingerprint images

The host uses this command to inform TSFID to scan fingerprint image once immediately after it receiving the command, no matter whether the finger has placed or not, or whether the fingerprint image quality achieved the system requirements.

Parameter: none

Command (the host→TSFID):

	Header	Length	Command ID	Checksum
Length	3	2	1	1
Value	1B 72 73		80	

Response (TSFID→the host):

	Header	Length	Command ID	Response	Checksum
Length	3	2	1	1	1
value	1B 72 73		80	Return code	

Return code:

Value(Hex)	Description
00	Command executed successful.
01	Invalid Command ID
03	Checksum error

For example:

Command –1B 72 73 00 02 80 xx

Successful Response –1B 72 73 00 03 80 00 xx

Failed Response -- 1B 72 73 00 03 80 01 xx

3.3.4 Conditioned scan of Fingerprint Images: 81H

Command ID: 81H

Function: conditioned scan of fingerprint images

The host sends this command to start one conditioned scan. After receiving this command, TSFID will:

Step 1: Send Command 84 31 to prompt placing on the finger, and stand by for the finger to press;

Step 2: scan the image and judge the image quality; and if having scanned qualified images,

proceed to Step 3; or else, send response 81 30 to the host prompting capture

overtime and terminate the operation;

Step 3: Send Command 84 32 to prompt raising the finger, and then send response 81 00 to the host prompting scan OK, and exit the operation.

Response hereafter shall refer to the response of Command 81 itself.

Parameter: none

Command (the host→TSFID) :

	Header	Length	Command ID	Checksum
Length	3	2	1	1
Value	1B 72 73		81	

Response (TSFID→the host):

	Header	Length	Command ID	Response	Checksum
Length	3	2	1	1	1
value	1B 72 73		81	Return code	

Return code:

Value(Hex)	Description
00	Command executed successfully.
01	Invalid Command ID
03	Checksum error
30	Fail to capture valid fingerprint images during the desired time

For example:

Command –1B 72 73 00 02 81 xx

Successful Response –1B 72 73 00 03 81 00 xx

Failed Response -- 1B 72 73 00 03 80 30 xx

3.3.5 Extract Features From Saved Images: 82H

Command ID: 82H

Function: extract features from saved images

The host sends the command to inform TSFID to extract characteristics from the saved image of the latest scan, and upload the features to the host.

Parameter: 00

Command (the host→TSFID):

	Header	Length	Command ID	Parameter	Checksum
Length	3	2	1	1	1
Value	1B 72 73		82	00	

Response (TSFID→the host):

	Header	Length	Command ID	Response	Checksum
Length	3	2	1	1+[n]	1
Value	1B 72 73		82	Return code + [Fingerprint feature]	

When Return code is 00, the **fingerprint feature** is valid. Refer to 2.3.1 for its structure definition.

Return code:

Value(Hex)	Description
00	Command executed successfully. Feature extracted;
01	Invalid Command ID
02	The parameter does not match its definition.
03	Checksum error
33	Not enough features extracted

For example:

Command –1B 72 73 00 03 82 00 xx;

Successful Response –1B 72 73 xx xx 82 00 nn xx xx ... xx

Failed Response -- 1B 72 73 00 03 82 33 xx: *fail to extract valid characteristics*

3.3.6 Capture a Real-time sample, upload the features to the host.: 83H

Command ID: 83H

Function: Capture a Real-time sample, upload the features to the host

The host sends this command to inform TSFID to execute one real-time conditioned image capture and extract fingerprint characteristics. After receiving this command, TSFID will:

Step 1: Send Command 84 31 to prompt placing the finger, and stand by for the finger to press;

Step 2: scan the image and judge the image quality; and if collected qualified images , proceed to Step 3; or else, send response 83 30 to the host prompting capture overtime and terminate the operation;

Step 3: Send Command 84 32 to prompt removing the finger, and extract characteristics according to the captured images; if the extraction is successful, send response 83 00 to the host, or else, send response 83 33 prompting extraction failure and exit the operation.

Response in the following text shall refer to the response of Command 83 itself.

Parameter: 00

Command (the host→TSFID):

	Header	Length	Command ID	Parameter	Checksum
Length	3	2	1	1	1
Value	1B 72 73		83	00	

Response (TSFID→the host):

	Header	Length	Command ID	Response	Checksum
Length	3	2	1	1+[n]	1
Value	1B 72 73		83	Return code+ [fingerprint feature]	

When Return code is 00, the **fingerprint feature** is valid. Refer to 2.3.1 for its structure definition.

Return code:

Value(Hex)	Description
00	Command executed successfully; characteristics successfully extracted
01	Invalid Command ID
02	The parameter does not match its definition
03	Checksum error
30	Fail to capture invalid fingerprint images during the desired time
33	Not enough features extracted

For example:

Command –1B 72 73 00 03 83 00 xx;

Successful Response –1B 72 73 xx xx 83 00 nn xx xx ... xx

Failed Response -- 1B 72 73 00 03 83 xx B9; *fail to extract valid characteristics*

3.3.7 Finger Action Prompts: 84H

Command ID: 84H

Function: finger action prompts

During the some operation demanded to real-time scan, TSFID sends this command to inform the host that the equipment is waiting for the finger to place or withdraw.

Parameter: operation

Response (TSFID→the host):

	Header	Length	Command ID	Parameter	Checksum
Length	3	2	1	1	1
Value	1B 72 73		84	Operation	

Definition of the operation:

Value	Description
0x31	Place the finger
0x32	Remove the finger

For example:

Command -- 1B 72 73 00 03 84 31 *xx*; *place the finger*
 1B 72 73 00 03 84 32 *xx*; *remove the finger*

3.3.8 Fingerprint Template Enrollment

Fingerprint Template Enrollment is to register the fingerprint template for later recognition. First, capture a number of samples by using the specific device, such as TSFID, then extract the salient features from these samples and construct the base template.

TSFID use such method to enroll a template: first, take three-time samples, and then generate template .

For enrolling a template, The host should send:

- Command 85 01 to capture the first sample;
- Command 85 02 to capture the second sample;
- Command 85 03 to capture the third sample;
- Command 86 to construct the fingerprint.

Each time TSFID receives Command 85, it will respond step by step:

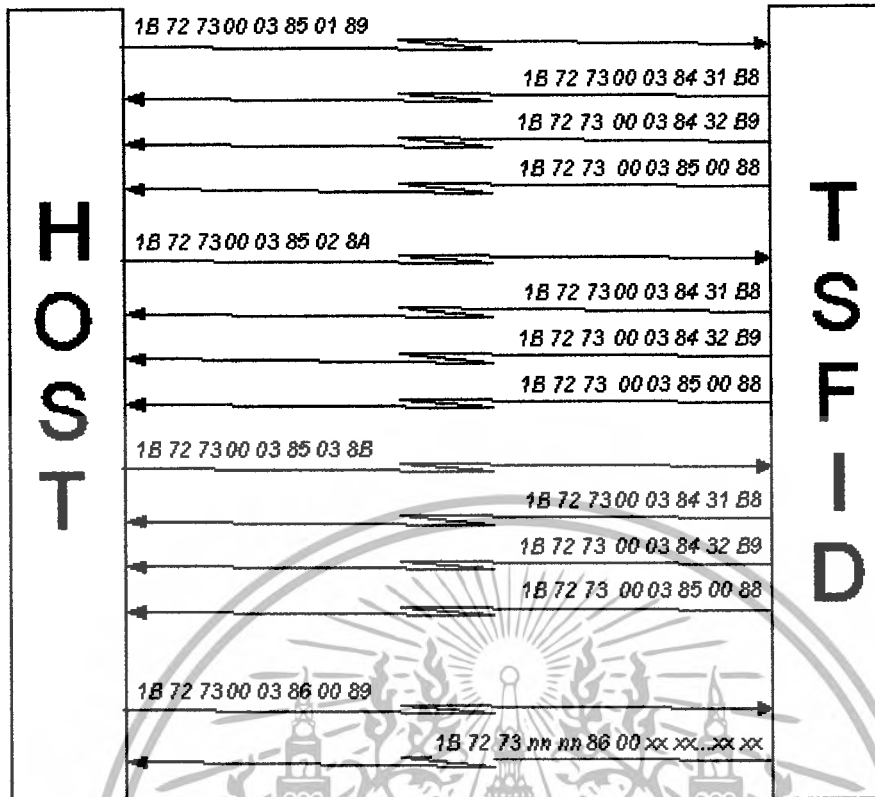
Step 1: Send Command 84 31 to prompt placing the finger, and stand by for the finger to press;

Step 2: scan the image and judge the image quality; and if collected qualified images , proceed to Step 3; or else, send response 85 30 to the host prompting capture overtime and terminate the operation;

Step 3: Send Command 84 32 to prompt removing the finger, and extract characteristics according to the captured images; if the extraction is successful, send response 83 00 to the host, or else, send response 85 33 prompting extraction failure and exit the operation.

Response in the following text shall refer to the response of Command 85 itself.

The communication process of a complete and correct fingerprint template enrollment is as follow:



1) Capture samples : 85H

Command ID: 85H

Function: Sample capturing for template enrollment

The host sends the command to inform TSFID to start enrolling template. Capturing fingerprint images in real time, extracting characteristics and save them at a temporary space pursuant to sampling serial number.

The response of the Command is complicated. Please refer to the text above for response process. Response in the follow-up text refers to the final response of the command.

Parameter: sample order

Command (the host→TSFID):

	Header	Length	Command ID	Parameter	Checksum
Length	3	2	1	1	1
Value	1B 72 73		85	sample order	

Valid value of sample order: 01 (first), 02 (second), 03 (third times)

Response (TSFID→the host):

	Header	Length	Command ID	Response	Checksum
Length	3	2	1	1	1
Value	1B 72 73		85	Return code	

Return code:

Value(Hex)	Description
00	Command executed successfully
01	Invalid Command ID
02	The parameter does not match its definition.
03	Checksum error
30	Fail to capture valid fingerprint image during the desired time
33	Not enough features extracted

For example:

Command –1B 72 73 00 03 85 01 xx

Successful Response -- 1B 72 73 00 03 85 00 xx

Failed Response -- 1B 72 73 00 03 85 33 xx: *fail to extract valid characteristics*

2) Template Construction: 86H

Command ID: 86H

Function: construct template

The host sends this command to inform TSFID to combine templates using the samples extracted through Command 85, and upload the template to the host or save it temporarily based on the saving conditions .

Parameter: 00

Command (the host→TSFID):

	Header	Length	Command ID	Parameter	Checksum
Length	3	2	1	1	1
Value	1B 72 73		86	00	

Response (TSFID→the host):

	Header	Length	Command ID	Parameter	Checksum
Length	3	2	1	1+[n]	1
Value	1B 72 73		86	Return code+ [fingerprint features]	

When Return code is 00, the **fingerprint features** is valid. Refer to 2.3.1 for its structure definition.

Return code:

Value(Hex)	Description
00	Command executed successfully
01	Invalid Command ID
02	The parameter does not match its definition.
03	Checksum error
31	Template enrollment fail.

For example:

Command –1B 72 73 00 03 86 00 xx: *Combine the templates and output*

Successful Response –1B 72 73 xx xx 86 00 mm xx xx ... xx

Failed Response -- 1B 72 73 00 03 86 31 xx: *template enrollment failure*

3.3.9 Capture a real-time sample, and verify it against the templates downloaded as the command parameters.: 89H

Command ID: 89H

Function: Real-time sampling and comparing with the templates downloaded along with the command. After receiving the command, TSFID will:

Step 1: Send Command 84 31 prompting placing the finger, and wait for the finger to press;

Step 2: Scan images and judge image quality; if having collected qualified images in due course, proceed to Step 3; or else, send response 89 30 to the host prompting capture overtime and terminate the operation;

Step 3: Send Command 89 32 to prompt raising the finger; extract characteristics according to captured fingerprints, and if succeed, proceed to Step 4, or else, send response 89 33 prompting extraction failure and terminate the operation;

Step 4: Compare the extracted characteristics with templates downloaded along with the command; in case of matching, send response 89 00; in case of mismatching, send response 89 32 and terminate the operation.

Response hereafter shall refer to response of Command 89 itself.

Parameter: template number (1) +fingerprint features

Command (the host→TSFID) :

	Header	Length	Command ID	Parameter	Checksum
Length	3	2	1	1+n	1
Value	1B 72 73		89	template number+ fingerprint feature	

The template number means how many templates there are in the command, and the valid value is 1-3.

Each template is saved one by one. Refer to 2.3.1 for the definition of **fingerprint features structure**.

For example: 02 00 20 xx xx00 16 xx xx. It means there are two templates. The length of the first template is 32 bytes. The length of the second template is 22 bytes. And length n shall be 1+20H+2+16H+2.

Response (TSFID→the host):

	Header	Length	Command ID	Response	Checksum
Length	3	2	1	1	1
Value	1B 72 73		89	Return code	

Return code:

Value(Hex)	Description
00h	The command executed successfully
01h	Invalid Command ID
02h	The parameter does not match its definition.
03h	Checksum error
30h	Failure to scan valid fingerprint image during desired time
32h	Verification was rejected or not match
33h	Not enough features extracted

For example:

Command –1B 72 73 xx xx 89 nn xx xx ... xx

Successful Response –1B 72 73 00 03 89 00 xx

Failed Response -- 1B 72 73 00 03 89 32 xx

3.3.10 Download two templates to match: 8AH

Command ID: 8AH

Function: Comparing the two templates downloaded along with the command.

The host sends this command to inform TSFID not to conduct real-time capture, but to compare the two downloaded templates and return the result to the host.

Parameter: 00+ fingerprint feature

Command (the host→TSFID) :

	Header	Length	Command ID	Parameter	Checksum
Length	3	2	1	1+n	1
Value	1B 72 73		8A	00+ fingerprint features	

Refer to 2.3.1 for definition for **fingerprint feature structure**.

For example: 00 20 xx xx00 16 xx xx. It means that the length of the first template to be

กิตติกรรมประกาศ

ขอขอบคุณอาจารย์เกษรติวรรณ ทรงสัจย์ เป็นอย่างสูงที่ชี้แนะและเป็นທີ່ปรึกษาให้กับ
โครงการ รวมทั้งความเอื้อเฟื้อ อุปกรณ์การทำงานเป็นอย่างดี

ขอขอบคุณท่านอาจารย์ทุกท่านที่ภาควิชาวิศวกรรมระบบควบคุม ที่ประสิทธิประสาท
วิชาการแก่นักศึกษาอย่างเต็มกำลัง รวมทั้งความรักความเมตตาที่มอบให้

ขอขอบคุณคุณพ่อ คุณแม่ผู้ให้ความดูแลเลี้ยงดู อบรมสั่งสอนตลอดมา

ขอขอบคุณ เพื่อนๆทุกคนที่คอยเป็นกำลังใจ ให้ความช่วยเหลือ ให้ยืมอุปกรณ์และให้
คำแนะนำที่ดีๆ มีประโยชน์ยิ่ง

และสุดท้ายขอขอบคุณ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร
ลาดกระบังที่เป็นสถานศึกษาที่เป็นแหล่งสร้างทักษะความรู้และประสบการณ์ต่างๆที่สำคัญต่อผู้จัดทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้