

เครื่องเก็บข้อมูลโดยผ่านบัตรแถบแม่เหล็ก
DATA STORAGE BY MAGNETIC CARD



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2541

เลขหม.....
เลขทะเบียน..... 32576
วัน, เดือน, ปี 18 พ.ค. 2542

13
3 f

เครื่องเก็บข้อมูลโดยผ่านบัตรแถบแม่เหล็ก
DATA STORAGE BY MAGNETIC CARD



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2541

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องเก็บข้อมูลโดยผ่านบัตรแถบแม่เหล็ก

DATA STORAGE BY MAGNETIC CARD

ผู้จัดทำ

1. นายชาลยุทธ ดิษฐศิริ 39013047

2. นายสุริยะ ฉายคุณรัฐ 39013063

3. นายแสนศักดิ์ ภาคกล้าเจียก 39013065


(ดร. สุทธิชัย นพนาถพงษ์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องเก็บข้อมูลโดยผ่านบัตรแถบแม่เหล็ก

DATA STORAGE BY MAGNETIC CARD

โดย นายชาญยุทธ คิชฐศิริ 39013047

นายสุริยะ ฉายคุณรัฐ 39013063

นายแสนศักดิ์ ภาคคำเจียก 39013065

อาจารย์ที่ปรึกษา ดร.สุทธิชัย นพนาถิพงษ์

บทคัดย่อ

โครงการนี้ เป็นการออกแบบและสร้างเครื่องอ่านและเก็บบันทึกข้อมูล โดยผ่านบัตรแถบแม่เหล็ก ซึ่ง ได้ใช้ไมโครคอนโทรลเลอร์ในการควบคุมและตรวจสอบ ตลอดจนทำหน้าที่ติดต่อสื่อสารข้อมูลแบบอนุกรมตามมาตรฐานการสื่อสาร RS-232C เพื่อเชื่อมต่อกับคอมพิวเตอร์ในการเก็บบันทึกและแสดงข้อมูลต่างๆ เช่น ข้อมูลประจำตัวของผู้ถือบัตร ฯ อีกทั้งสามารถนำไปใช้ในระบบรักษาความปลอดภัยและสามารถตรวจสอบการใช้บริการได้

ABSTRACT

This project is a design and construction of read and write data by magnetic card. The microcontroller is used for controlling and checking all data via the serial data communication Standard port RS-232C for interfacing with computer. The computer is used to show the stored data of card holder, as well as for the security system.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีหรือหลักการ	3
2.1 ความรู้เบื้องต้นเกี่ยวกับบัตรแม่เหล็ก	3
2.1.1 มาตรฐานของบัตรแม่เหล็ก	3
2.1.2 ชุดรหัสข้อมูลในแทร็คที่ 2 ของบัตรแม่เหล็ก	4
2.1.3 รูปแบบข้อมูลที่บันทึกในแทร็คที่ 2 บนบัตรแม่เหล็ก	4
2.2 การตรวจสอบความผิดพลาด	5
2.2.1 การตรวจสอบแบบพาริตี	6
2.2.2 การตรวจซ้ำในแนวตั้งและแนวนอน	7
2.3 การบันทึกเทปแม่เหล็กในระบบดิจิทัล	8
2.4 การกลับคืนสัญญาณของการบันทึก	10
2.5 หลักการเบื้องต้นของเครื่องเก็บข้อมูลโดยผ่านบัตรแถบแม่เหล็ก	11
2.5.1 การอ่านและบันทึกข้อมูลผ่านบัตรแม่เหล็ก	11
2.5.1.1 ส่วนอ่านรหัสข้อมูลของบัตร	12
2.5.1.2 ส่วนบันทึกข้อมูลลงบัตร	13
2.5.1.3 ส่วนควบคุมมอเตอร์	14
2.5.1.4 ส่วนตรวจจับบัตร	14
2.5.2 ส่วนควบคุม	15
2.5.3 ส่วนแสดงผล	16
2.6 โฟโต้ไดโอด	17
2.7 อุปกรณ์แปลงแสง	18
2.8 สถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบชิพตระกูลเด็วตระกูล MCS-51	19
2.8.1 โครงสร้างของ AT89S8252	21
2.8.2 การจัดการกับหน่วยความจำ	22
2.8.3 การจัดวางขาและหน้าที่ต่าง ๆ	26
2.8.4 การทำงานของ AT89S8252	29
2.8.5 ไคอะแกรมการติดต่อกับหน่วยความจำ	30
2.8.6 การรีเซ็ต	33
2.8.7 การโปรแกรมร็อตค็อกไทม์เมอร์	34
2.8.8 การเชื่อมต่ออุปกรณ์แบบอนุกรม	35
2.9 คีอัทเมทริกซ์แอลซีดีโมดูล	36
2.9.1 รายละเอียดของคำสั่ง ของ HD 44780	37

2.10 DS1202 ชิพสัญญาณนาฬิกาแบบอนุกรม	42
2.10.1 การรับ – ส่งข้อมูลของ DS1202	43
2.10.2 การรับ – ส่งข้อมูลกับ DS 202	44
2.10.3 การจัดแอดเดรสใน DS1202	46
2.10.4 DS1202 กับการใช้งานจริง	47
2.11 MAX691 ชิพผู้ช่วยงาน CPU	48
2.11.1 การจัดวางขาและหน้าที่	49
2.11.2 การใช้งาน MAX691	49
2.12 การเชื่อมต่อระหว่างคอมพิวเตอร์กับพริ้นเตอร์	52
2.12.1 ลักษณะการสื่อสารข้อมูล	52
2.12.2 การอินเตอร์เฟสพริ้นเตอร์แบบขนานและขาการเชื่อมต่อแบบเซนโทรนิกส์	54
2.13 การติดต่อสื่อสารข้อมูลผ่านพอร์ทอนุกรม RS-232C	60
2.13.1 พอร์ทอนุกรม	60
2.13.2 รูปแบบของเฟรมข้อมูล	63
2.13.3 ซอฟต์แวร์ที่ใช้ในการติดต่อสื่อสารผ่านพอร์ทอนุกรม	64
บทที่ 3 การคำนวณและการสร้าง	84
3.1 ส่วนของการอ่านข้อมูลจากบัตรแม่เหล็ก	84
3.1.1 ภาคขยายสัญญาณจากหัวเทป	84
3.1.2 ภาคตรวจจับแรงดันสูงสุด	85
3.1.3 ภาคปรับแต่งรูปคลื่น	86
3.1.4 ภาคการถอดรหัสข้อมูล	87
3.2 ส่วนของการบันทึกข้อมูลลงบัตร	88
3.2.1 ภาคกำเนิดสัญญาณนาฬิกา	88
3.2.2 ภาคหารความถี่	88
3.2.3 ภาคการบันทึกข้อมูล	89
3.2.4 ภาคขยายสัญญาณข้อมูล	89
3.3 ส่วนควบคุมมอเตอร์	90
3.3.1 ภาคควบคุมการเดินหน้าและถอยหลัง	90
3.3.2 ภาคขับมอเตอร์	91
3.4 ส่วนตรวจจับบัตร	91
3.4.1 ภาคกำเนิดความถี่	92
3.4.2 ภาคขับแอลอีดีอินฟราเรด	92

เอกสารนี้เป็นเอกสาร 3.4.3 ภาคขยายสัญญาณใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านอื่น 92
 3.4.4 ภาคตรวจจับระดับสัญญาณ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำ 92

3.5 ส่วนควบคุม	93
3.5.1 ส่วนประมวลผลกลาง	93
3.5.2 หน่วยความจำสำหรับเก็บข้อมูลภายนอก	93
3.5.3 พอร์ท 8255	94
3.5.4 การเชื่อมโยง MCS-51 กับหน่วยความจำ	95
3.5.5 เป็นคีย์บอร์ด	97
3.6 ส่วนติดต่อและแสดงผลทางคอมพิวเตอร์	110
บทที่ 4 การทดลองและผลการทดลอง	118
4.1 ส่วนของการอ่านข้อมูลจากบัตร	118
4.2 ส่วนของการบันทึกข้อมูลลงบัตร	120
4.3 ส่วนของการควบคุมมอเตอร์	124
4.4 ส่วนของการตรวจจับบัตร	124
4.5 ส่วนควบคุม	126
4.6 ส่วนแสดงผลทางคอมพิวเตอร์	129
บทที่ 5 บทวิจารณ์และบทสรุป	136
5.1 สรุปผลการทดลอง	136
5.2 วิจารณ์ผลการทดลอง	136
ภาคผนวก	
กิตติกรรมประกาศ	
หนังสืออ้างอิง	



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ

หน้า

รูปที่ 2.1	แสดงมาตรฐานของบัตรแม่เหล็ก	3
รูปที่ 2.2	แสดงรูปแบบของข้อมูลที่บันทึกในแทร็คที่ 2 บนบัตรแม่เหล็ก	4
รูปที่ 2.3	แสดงหลักการสร้างพาริตีบิท	6
รูปที่ 2.4	แสดงวงจรการสร้างพาริตีบิท	6
รูปที่ 2.5	แสดงความเข้มสนามแม่เหล็กที่เกิดขึ้นบริเวณหัวเทป	8
รูปที่ 2.6	แสดงฮิสเทอรีซิสที่เกิดขึ้นในหัวเทป	8
รูปที่ 2.7	แสดงการบันทึกในช่วงอิมพัลส์	9
รูปที่ 2.8	สัญญาณที่อ่านได้จากหัวเทป	10
รูปที่ 2.9	บล็อกไดอะแกรมเบื้องต้นของเครื่องเก็บข้อมูลโดยผ่านบัตรแถบแม่เหล็ก	11
รูปที่ 2.10	บล็อกไดอะแกรมของเครื่องอ่านและบันทึกข้อมูลของบัตรแม่เหล็ก	12
รูปที่ 2.11	สัญญาณการอ่านข้อมูล	12
รูปที่ 2.12	แสดงลักษณะการบันทึกข้อมูลแบบF2F	13
รูปที่ 2.13	การทำงานของเครื่องอ่านเขียนบัตรแม่เหล็กแบบป้อนบัตรด้วยมอเตอร์	14
รูปที่ 2.14	โครงสร้างของระบบควบคุม	15
รูปที่ 2.15	แสดง พี-เอ็น จังก์ชัน โฟโตไดโอด	17
รูปที่ 2.16	แสดง พี-ไอ-เอ็น จังก์ชัน โฟโตไดโอด	17
รูปที่ 2.17	โครงสร้างภายนอกของโฟโตไดโอด	18
รูปที่ 2.18	โครงสร้างของไดโอดเปล่งแสง	19
รูปที่ 2.19	ไดอะแกรมโครงสร้างของ AT89S8252	21
รูปที่ 2.20	การจัดวางขา AT89S8252	26
รูปที่ 2.21	วงจรออสซิลเลเตอร์ภายใน AT89S8252	29
รูปที่ 2.22	ผังการทำงานแต่ละคำสั่ง	30
รูปที่ 2.23	ผังเวลาการติดต่อกับหน่วยความจำภายนอก	31
รูปที่ 2.24	ผังเวลาการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก	31
รูปที่ 2.25	ผังเวลาการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก	31
รูปที่ 2.26	ผังเวลาการรีเซ็ต	33
รูปที่ 2.27	แสดงการจำขาของDS1202	43
รูปที่ 2.28	บล็อกไดอะแกรมส่วนต่าง ๆ ภายใน DS1202	44
รูปที่ 2.29	แสดงรูปแบบของคำสั่ง	44

เอกสารรูปที่ 2.30 แสดงรูปคลื่นของคำสั่งรับส่งข้อมูลเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านอื่น

ไม่ว่ารูปที่ 2.31 นี้แสดงผังเวลาขณะเขียนและส่งข้อมูลและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.32	วงจรใช้งานจริงของ DS1202	48
รูปที่ 2.33	การจัดวางขาของ MAX 691	48
รูปที่ 2.34	แสดงวงจรใช้งานจริงของ MAX691	50
รูปที่ 2.35	การต่อ MAX691 เพื่อกำหนด time period	51
รูปที่ 2.36	แสดงการติดต่อระหว่างคอมพิวเตอรืและพรีนเตอร์แบบขนาน	52
รูปที่ 2.37	แสดงการติดต่อระหว่างคอมพิวเตอรืและพรีนเตอร์แบบอนุกรม	53
รูปที่ 2.38	แสดงขาของคอนเนคเตอร์แบบอิมพินอล	55
รูปที่ 2.39	แสดงช่วงเวลาของรูปคลื่นที่จะส่งข้อมูลอักขระไปยังพรีนเตอร์แบบขนาน	58
รูปที่ 2.40	แสดงขาของคอนเนคเตอร์แบบ DB25	58
รูปที่ 2.41	การเชื่อมต่อสายสัญญาณระหว่าง CPU ของเครื่องคอมพิวเตอรืกับพอร์ทอนุกรม	61
รูปที่ 2.42	อ็อบเจกต์โปรแกรมย่อยในฐานข้อมูล	72
รูปที่ 3.1	บล็อกไดอะแกรมของเครื่องอ่านและบันทึกข้อมูลของบัตรแม่เหล็ก	83
รูปที่ 3.2	วงจรขยายความแตกต่าง	84
รูปที่ 3.3	วงจรขยายสัญญาณแบบกลับเฟส	84
รูปที่ 3.4	อัตรการขยายรวมของวงจรขยายสัญญาณแบบคาสเคด	85
รูปที่ 3.5	วงจรตรวจจับแรงดันสูงสุด	85
รูปที่ 3.6	วงจรปรับแต่งรูปคลื่น	86
รูปที่ 3.7	วงจรสร้างสัญญาณนาฬิกาในการถอดรหัส	87
รูปที่ 3.8	วงจรกำเนิดสัญญาณนาฬิกา	88
รูปที่ 3.9	วงจรหารความถี่	88
รูปที่ 3.10	วงจรบันทึกข้อมูล	89
รูปที่ 3.11	วงจรขยายสัญญาณการบันทึก	89
รูปที่ 3.12	วงจรควบคุมมอเตอร์	90
รูปที่ 3.13	วงจรขับมอเตอร์	91
รูปที่ 3.14	วงจรตรวจจับบัตร	91
รูปที่ 3.15	วงจรตรวจระดับสัญญาณ	92
รูปที่ 3.16	การต่อรีเซ็ทและออสซิลเลเตอร์ของ AT89S8252	93
รูปที่ 3.17	แผนผังหน่วยความจำภายนอกในการเก็บข้อมูล	94
รูปที่ 3.18	การจัดวางขาและตารางความจริงของ 74LS138	95
รูปที่ 3.19	การต่อและจัดวางเป็นคีย์บอร์ด	97
รูปที่ 3.20	การจัดวางตำแหน่งเป็นคีย์บอร์ด	97
รูปที่ 3.21	แผนผังโปรแกรมหลัก	98

เอกสารรูปที่ 3.22 แผนผังโปรแกรมในส่วนหลักและการแก้ไขข้อมูลนั้น ไม่นุญาตให้นำไปใช้ประโยชน์ใด ๆ การค้า
ไม่ว่ากรรูปที่ 3.23 แผนผังโปรแกรมการตรวจสอบและแก้ไขรหัสผ่านของระบบของเอกสารทุกครั้งที่มีการนั 102 ใช้

รูปที่ 3.24	แผนผังโปรแกรมลบรหัสสมาชิกออกจากระบบและแสดงการป้อนค่าผ่านแอลซีดี	103
รูปที่ 3.25	แผนผังโปรแกรมแก้ไขรหัสผ่านของผู้ใช้และลบข้อมูลทั้งหมด	104
รูปที่ 3.26	แผนผังโปรแกรมตรวจสอบรหัสสมาชิกและตรวจสอบรหัสผ่าน	105
รูปที่ 3.27	แผนผังโปรแกรมการบันทึกรหัสข้อมูลลงบนบัตรแม่เหล็ก	106
รูปที่ 3.28	แผนผังโปรแกรมการอ่านรหัสบนบัตรแม่เหล็ก	107
รูปที่ 3.29	แผนผังโปรแกรมการติดต่อกับเครื่องพิมพ์และบันทึกการใช้บริการ	108
รูปที่ 3.30	แผนผังโปรแกรมการติดต่อกับเครื่องพิมพ์และการบันทึกข้อมูลการใช้บริการ	109
รูปที่ 3.31	แผนผังโปรแกรมหลัก	110
รูปที่ 3.32	แผนผังโปรแกรมเมนูฟอร์ม	111
รูปที่ 3.33	แผนผังโปรแกรมการเข้าสู่ระบบ	112
รูปที่ 3.34	แผนผังโปรแกรมผลการลงทะเบียน	113
รูปที่ 3.35	แผนผังโปรแกรมผลการศึกษา	114
รูปที่ 3.36	แผนผังโปรแกรมการค้นหาประวัติบุคคล	115
รูปที่ 3.37	แผนผังโปรแกรมการแก้ไขรหัสผ่าน	115
รูปที่ 3.38	แสดงเครื่องต้นแบบของเครื่องเก็บข้อมูลโดยผ่านบัตรแม่เหล็ก	116
รูปที่ 3.39	แสดงส่วนหน้าของเครื่องต้นแบบ	116
รูปที่ 3.40	แสดงส่วนหลังของเครื่องต้นแบบ	117
รูปที่ 4.1	แสดงผลการทดลองวงจรขยายสัญญาณและวงจรตรวจจับแรงดันสูงสุด	118
รูปที่ 4.2	แสดงผลการทดลองวงจรปรับแต่งรูปคลื่นและสร้างฐานเวลา	119
รูปที่ 4.3	แสดงผลการทดลองวงจรสร้างสัญญาณนาฬิกาและหารความถี่	120
รูปที่ 4.4	แสดงผลการบันทึกข้อมูลดิจิทัลที่ลอจิก 1	121
รูปที่ 4.5	แสดงผลการบันทึกข้อมูลที่ลอจิก 0	121
รูปที่ 4.6	แสดงผลการขยายสัญญาณการบันทึกข้อมูลดิจิทัลที่ลอจิก 1	122
รูปที่ 4.7	แสดงผลการขยายสัญญาณบันทึกข้อมูลที่ลอจิก 0	122
รูปที่ 4.8	แสดงผลการอ่านสัญญาณบันทึกข้อมูลที่ลอจิก 1	123
รูปที่ 4.9	แสดงผลการอ่านสัญญาณบันทึกข้อมูลที่ลอจิก 0	123
รูปที่ 4.10	แสดงผลการทดลองวงจร โมโนสเตเบิล	124
รูปที่ 4.11	แสดงผลการทดลองวงจรผลิตความถี่และขับแอลอีดีอินฟาเรด	125
รูปที่ 4.12	แสดงผลการทดลองวงจรขยายสัญญาณจากโฟโต้ไดโอด	125
รูปที่ 4.13	แสดงผลการทดลองวงจรตรวจจับบัตร	126
รูปที่ 4.14	แสดงชื่อโปรเจกต์ผ่านทางแอลซีดี	126
รูปที่ 4.15	แสดงผลเมนูหลักในการเข้าสู่ระบบ	127

เอกสารนี้ได้รับการสงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านอื่น

รูปที่ 4.16 แสดงผลการแก้ไขข้อมูล

รูปที่ 4.17 แสดงเมนูการเลือกใช้บริการ

รูปที่ 4.18 สัญลักษณ์ที่ใช้ในการนับจำนวนกระดาษ	129
รูปที่ 4.19 แสดงหน้าจอหลักสำหรับการใช้บริการ	130
รูปที่ 4.20 แสดงหน้าจอสำหรับการเข้าสู่ระบบ	130
รูปที่ 4.21 แสดงรายงานผลการลงทะเบียน	131
รูปที่ 4.22 แสดงรายงานผลการศึกษา	132
รูปที่ 4.23 แสดงหน้าจอการค้นหาประวัติบุคคลแบบป้อนรหัสประจำตัว	133
รูปที่ 4.24 แสดงหน้าจอการค้นหาประวัติบุคคลแบบป้อนชื่อ	134
รูปที่ 4.25 แสดงหน้าจอการแก้ไขรหัสผ่าน	135
รูปที่ 4.26 ตัวอย่างการพิมพ์แสดงผลข้อมูลการใช้บริการ	135



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงรหัสข้อมูลแต่ละตัวของเทอร์คที่ 2	4
ตารางที่ 2.2 แสดงการตรวจซ้ำในแนวตั้งและแนวนอน	7
ตารางที่ 2.3 แผนผังSFRและรีจิสเตอร์เมื่อเกิดการรีเซ็ต	23
ตารางที่ 2.4 วอร์ทดีค็อกและรีจิสเตอร์ควบคุมหน่วยความจำ	24
ตารางที่ 2.5 รีจิสเตอร์ควบคุมSPI	24
ตารางที่ 2.6 รีจิสเตอร์แสดงสถานะของSPI	24
ตารางที่ 2.7 แสดงหน้าที่ขาS/CและR/L	25
ตารางที่ 2.8 แสดงการกำหนดค่าตัวอักษร	39
ตารางที่ 2.9 ตำแหน่ง ADDRESS ของจอ	40
ตารางที่ 2.10 แสดงขาของLCD MODULE	42
ตารางที่ 2.11 การจัดแอดเดรสของ DS1202	47
ตารางที่ 2.12 แสดงการเชื่อมต่อขาและคำอธิบายสำหรับการเชื่อมต่อแบบเซน โทริคส์ขนาน	55
ตารางที่ 2.13 แสดงการเชื่อมต่อขาสำหรับการต่อขาแบบDB25	59
ตารางที่ 2.14 แสดงการปรับต่อขาDB 25 และแอมทินอล	59
ตารางที่ 2.15 ค่าคงที่ใช้ในการระบุถึงประเภทอ็อปเจ็คข้อมูล	76
ตารางที่ 2.16 วิธีการเคลื่อนตำแหน่งตัวชี้ข้อมูล	77
ตารางที่ 2.17 ฟังก์ชันการทำงานสำหรับการค้นหาข้อมูล	79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในปัจจุบัน การอ่านและการบันทึกข้อมูลลงบัตรพลาสติกที่ใช้งานทั่วไป ไม่ว่าจะเป็นบัตรเครดิต บัตรธนาคาร บัตรประจำตัวนักศึกษา หรือแม้กระทั่งบัตรการใช้โทรศัพท์ ก็ยังมีการใช้บัตรแบบแถบแม่เหล็ก ในการบันทึกรหัสข้อมูลส่วนตัวของผู้ถือบัตร ซึ่งผู้ถือบัตรจึงจะมีสิทธิในการใช้บริการต่างๆ ได้ ถึงแม้ว่าจะมีการนำการอ่านและการบันทึกข้อมูลแบบบาร์โค้ดหรือส്മาร์ทการ์ด มาใช้แล้วก็ตาม จึงจะเห็นได้ว่าบัตรแม่เหล็กหรือบัตรแถบแม่เหล็กก็ยังมีข้อดีอยู่คือ ในการปลอมแปลงเลียนแบบบัตรเพื่อลักลอบนำไปใช้งานนั้นทำได้ยากเมื่อเทียบกับบัตรบาร์โค้ด โดยในบัตรแบบบาร์โค้ดนั้นจะสามารถทำการลอกเลียนแบบได้ง่ายมากเพียงแต่นำรหัสแถบบาร์โค้ดไปถ่ายเอกสารก็จะได้บัตรแบบบาร์โค้ดเช่นเดิม ดังนั้นความปลอดภัยในสิทธิการใช้สิทธิการใช้ประโยชน์จากบัตรจะน้อยกว่าบัตรแบบแถบแม่เหล็ก

ในโครงการนี้จะใช้บัตรแม่เหล็ก ซึ่งเป็นมาตรฐานเดียวกับบัตร เอทีเอ็ม ของธนาคารในการตรวจสอบก่อนเข้าสู่ระบบ หรือสามารถกระทำกรอื่นๆ ได้ โดยที่เมื่อมีการเข้าสู่ระบบจะมีการบันทึกข้อมูลการใช้บริการในด้านต่างๆ

1. เวลาเข้า – ออกของการใช้บริการ
2. การใช้จำนวนกระดาษถ่ายเอกสาร
3. แสดงผลข้อมูลต่างๆ ผ่านทางคอมพิวเตอร์
 - 3.1 ประวัติส่วนบุคคล
 - 3.2 ผลการศึกษา
 - 3.2 การลงทะเบียน
4. ติดต่อผ่านเครื่องพิมพ์

เครื่องเก็บข้อมูลโดยผ่านบัตรแถบแม่เหล็กนี้ จะเป็นการอ่านและบันทึกข้อมูลลงบนบัตรแม่เหล็ก ซึ่งรหัสข้อมูลนี้อาจเป็นรหัสประจำตัวนักศึกษาหรือรหัสบัตรก็ได้ โดยในส่วนของข้อมูลการใช้บริการต่างๆ จะใช้บัตรแม่เหล็กเป็นส่วนหลักในการนำไปสู่การบันทึกข้อมูลต่างๆ เปรียบเสมือนว่าผู้ถือบัตรเป็นผู้มีสิทธิในการใช้บริการ นอกจากนั้นระบบยังรักษาความปลอดภัยโดยให้ผู้ใช้บริการต้องกรอกรหัสผ่านของผู้ถือบัตรเสมอทุกครั้งที่ใช้บริการ ส่วนในการแก้ไขข้อมูลต่างๆ ภายในระบบ เจ้าของระบบเท่านั้นจึงจะเป็นผู้ที่สามารถแก้ไขข้อมูลได้ เนื่องจากระบบจะมีการสอบถามรหัสผ่านประจำระบบก่อนทุกครั้งที่จะมีการแก้ไขข้อมูลเสมอ

ในส่วนของการอ่านและบันทึกข้อมูลลงบัตรจะเป็นลักษณะการป้อนบัตรด้วยมอเตอร์ ซึ่งในการอ่านและบันทึกข้อมูลจะใช้หัวเทปคาสเซ็ทเป็นตัวอ่านและบันทึกข้อมูลลงบัตร โดยถูกควบคุมด้วยไมโครคอนโทรลเลอร์ตระกูล MCS – 51 โครงการนี้ใช้เบอร์ AT89S8252 เนื่องจากสามารถบันทึกโปรแกรมมอนิเตอร์ได้ภายในตัวไอซี ดังนั้นจึงสามารถใช้งานพื้นที่แอดเดรสภายนอกได้มากกว่าเมื่อเทียบกับการใช้รอมภายนอก ทำให้สามารถนำพื้นที่แอดเดรสมาใช้งานในด้านฐานข้อมูลได้เต็มที่ แล้วทำการแสดงผลผ่านทางแอลซีดีหรือทางคอมพิวเตอร์

ส่วนคอมพิวเตอร์ที่ใช้ในการรับข้อมูลและแสดงผลข้อมูลต่างๆจะใช้โปรแกรม Visual Basic เป็นส่วนหลักในการแสดงผลและ Microsoft Access ช่วยในการเก็บข้อมูลต่างๆที่รับข้อมูลเข้ามาไว้ในฐานข้อมูล โดยในการรับข้อมูลระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์จะติดต่อสื่อสารข้อมูลผ่านพอร์ตอนุกรม RS-232C ซึ่งในการรับข้อมูลการใช้บริการทั้งหมดของระบบผ่านมายังคอมพิวเตอร์ จะให้บันทึกข้อมูลในเวลาที่กำหนด

การติดต่อใช้งานกับเครื่องถ่ายภาพเอกสาร จะใช้สำหรับการตรวจสอบการใช้จำนวนกระดาษถ่ายภาพเอกสาร โดยที่เมื่อมีการถ่ายภาพเอกสารขึ้นนั้น เครื่องถ่ายภาพเอกสารจะมีการนับจำนวนการใช้กระดาษอยู่แล้วดังนั้นจะนำส่วนนี้มาเชื่อมต่อกับไมโครคอนโทรลเลอร์ ในการควบคุมการใช้กระดาษของสมาชิกแต่ละคน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 ทฤษฎีหรือหลักการ

2.1 ความรู้เบื้องต้นเกี่ยวกับบัตรแม่เหล็ก

บัตรแม่เหล็กที่ใช้ในโครงการนี้ จะใช้มาตรฐานเดียวกันกับมาตรฐานบัตรแม่เหล็กเอทีเอ็มที่ใช้ในธนาคารทั่วไป โดยรูปแบบของข้อมูลที่บันทึกบนแถบแม่เหล็กและคุณลักษณะต่างๆของบัตรแม่เหล็กและคุณลักษณะต่างๆของบัตรจะเป็นไปตามมาตรฐานสากล ISO 7810-7813 เพื่อที่จะสามารถพัฒนาระบบต่อไปได้

2.1.1 มาตรฐานของบัตรแม่เหล็ก



รูปที่ 2.1 แสดงขนาดมาตรฐานของบัตรแม่เหล็ก

จากรูปที่ 2.1 ตามมาตรฐานของ ISO เราสามารถพิจารณา แทร็คต่างๆที่ใช้งานได้เป็น 3 แทร็ค โดยมีรายละเอียดในการใช้งานแต่ละแทร็คดังนี้

แทร็คที่ 1 เป็นแทร็คที่สามารถทำการอ่านข้อมูลได้อย่างเดียว เรียกว่า Read only โดยข้อมูลประกอบด้วยตัวอักษรและตัวเลข อยู่ในช่วงบริเวณเส้นขนาน a และเส้นขนาน b

แทร็คที่ 2 เป็นแทร็คที่สามารถอ่านข้อมูลได้อย่างเดียวเช่นกัน แต่ข้อมูลจะเป็นลักษณะตัวเลขเพียงอย่างเดียว อยู่ในช่วงเส้นขนาน b และเส้นขนาน c ซึ่งเป็นแทร็คที่ใช้งานโดยทั่วไป

แทร็คที่ 3 เป็นแทร็คที่สามารถทำการอ่านและเขียนข้อมูลลงไปได้ ซึ่งเก็บข้อมูลที่เป็นตัวเลข ซึ่งอยู่ในช่วงเส้นขนาน c และเส้นขนาน d

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 ชุดรหัสข้อมูลในแตร็คที่ 2 ของบัตรแม่เหล็ก

ในการใช้งานโดยทั่วไปของบัตรแม่เหล็ก จะใช้งานในส่วนของแตร็คที่ 2 ถ้าเป็นบัตรเอทีเอ็มของธนาคารต่างๆ ในส่วนของแตร็คนี้จะเป็นการเก็บข้อมูลเกี่ยวกับข้อมูลของธนาคารและเลขที่บัญชี ดังนั้นแต่ละธนาคารจึงสามารถใช้บริการร่วมกันได้

ข้อมูลที่บันทึกในแตร็คที่ 2 ของบัตรแม่เหล็กจะเป็นตัวเลขเพียงอย่างเดียว โดยที่ตัวเลขหนึ่งตัวจะประกอบด้วยบิตข้อมูลแบบ BCD 4 บิตและบิตพาริตี 1 บิต ซึ่งใช้ในการตรวจสอบข้อมูลแต่ละตัวเลข โดยตรวจสอบแบบพาริตีคู่ ตามมาตรฐาน ISO ระบุไว้ว่าจำนวนข้อมูลสูงสุดในแตร็คที่ 2 มีได้ไม่เกิน 40 ตัว (รวมสัญลักษณ์การเริ่มต้นและสิ้นสุด) ในส่วนของชุดข้อมูลตัวเลขแต่ละตัวสำหรับแตร็คที่ 2 แสดงไว้ในตารางที่ 2.1

2.1.3 รูปแบบข้อมูลที่บันทึกในแตร็คที่ 2 บนบัตรแม่เหล็ก

SYN	START	DATA	SEP	DATA	----	STOP	LRC	SYNC
-----	-------	------	-----	------	------	------	-----	------

รูปที่ 2.2 แสดงรูปแบบของข้อมูลที่บันทึกในแตร็คที่ 2 บนบัตรแม่เหล็ก

จากรูปที่ 2.2 เป็นรูปแบบการบันทึกข้อมูลในบัตรแม่เหล็กที่ใช้กันโดยทั่วไป ซึ่งแต่ละสัญลักษณ์มีความหมายดังนี้

- SYN (Synchronization character) เป็นการแสดงการเริ่มต้นและปิดท้ายข้อมูล ใช้รหัส 00H
- START (Start bit) เป็นการแสดงการเริ่มต้นในการอ่านหรือบันทึกข้อมูล ใช้รหัส 0BH
- DATA เป็นส่วนของข้อมูลที่เป็นตัวเลข 0-9
- SEP (Separate) เป็นตัวเชื่อมหรือกั้นระหว่างข้อมูล ใช้รหัส 0DH
- STOP (Stop bit) เป็นการแสดงการสิ้นสุดของข้อมูล ใช้รหัส 0FH
- LRC (Longitude Redundancy Check) เป็นการตรวจสอบชุดข้อมูลในแนวนอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P	b4	b3	b2	b1	รหัส
1	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	2
1	0	0	1	1	3
0	0	1	0	0	4
1	0	1	0	1	5
1	0	1	1	0	6
0	0	1	1	1	7
0	0	1	0	0	8
1	1	0	0	1	9
1	1	0	1	0	A
0	1	0	1	1	B1
1	1	1	0	0	A
0	1	1	0	1	B2
0	1	1	1	0	A
1	1	1	1	1	B3

ตารางที่ 2.1 แสดงรหัสข้อมูลแต่ละตัวของเทร็คที่ 2

2.2 การตรวจสอบความผิดพลาด (Error detection)

ในระบบการสื่อสารข้อมูลในทางอุดมคติจะไม่ให้เกิดความผิดพลาดในการรับ-ส่งข้อมูลแต่ในความเป็นจริง ทางฝ่ายรับข้อมูลจะเกิดความผิดพลาดขึ้น เนื่องจากข้อมูลที่ได้รับไม่ตรงกับข้อมูลที่ส่งมา ดังนั้นทางด้านฝ่ายรับข้อมูลจะต้องมีวิธีการตรวจสอบข้อมูล โดยในระบบการสื่อสารข้อมูลจะมีวิธีการตรวจสอบ 3 วิธี

- ตรวจสอบบิตพาริตี (Parity bit checking)
- การตรวจซ้ำในแนวตั้งและแนวนอน (Vertical and Longitude Redundancy Checking)
- การตรวจซ้ำแบบวงกลม (Cyclic Redundancy Checking)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น เมื่อผู้ยัดเห็นไปใช้ประโยชน์ในการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 การตรวจสอบแบบพาริตีบิต

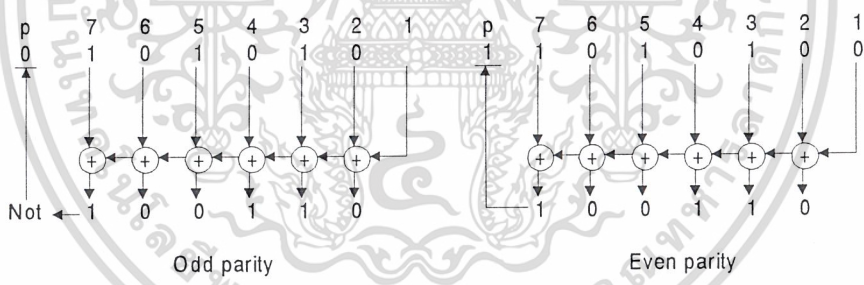
การตรวจสอบบิตพาริตีเป็นวิธีที่ง่ายที่สุดในการตรวจสอบความผิดพลาดของข้อมูล โดยพาริตีบิต 1 บิต จะรวมกับตัวอักษรแต่ละตัว จึงเป็นการเพิ่มบิตตัวอักษรอีก 1 บิต ดังนั้นบิตที่เพิ่มขึ้น เรียกว่า พาริตีบิต (Parity bit) ซึ่งมี 2 แบบ คือ

1. พาริตีคู่ (Even parity) คือ จำนวน “1” รวมทั้งพาริตีบิต เป็นจำนวนคู่
2. พาริตีคี่ (Odd parity) คือ จำนวน “1” รวมทั้งพาริตีบิต เป็นจำนวนคี่

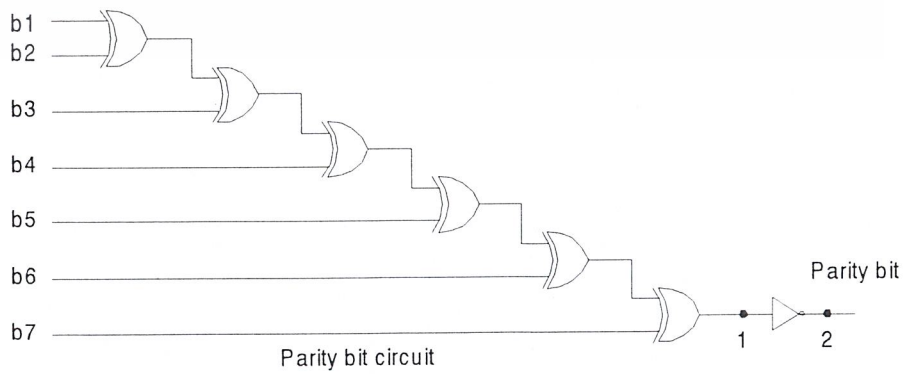
เพื่อความเข้าใจจึงได้แสดงตัวอย่างการหาพาริตีบิต ของข้อมูลตัวอักษร T ซึ่งในรหัสแอสกีมีค่าเท่ากับ 54H (1010100)

-เมื่อเป็นพาริตีคี่								-เมื่อเป็นพาริตีคู่								
P	7	6	5	4	3	2	1	ตำแหน่งบิต	P	7	6	5	4	3	2	1
0	1	0	1	0	1	0	0	บิตข้อมูล	1	1	0	1	0	1	0	0

จะเห็นได้ว่าบิตพาริตีจะเกิดขึ้นจากการเปรียบเทียบบิตข้อมูลแต่ละบิตโดยใช้การเอ็กคลูซีฟ ออร์ (Exclusive or) ในการเปรียบเทียบข้อมูลแต่ละบิต ซึ่งในการเอ็กคลูซีฟกันจะได้ เอาท์พุทเท่ากับ “0” เมื่อ อินพุทเหมือนกัน ดังแสดงไว้ในรูปที่ 2.3 และ ในรูปที่ 2.4



รูปที่ 2.3 แสดงหลักการสร้างพาริตีบิต



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดรูปที่ 2.4 แสดงวงจรการสร้างพาริตีบิต เอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.4 เป็นการสร้างพริตตี้บิตโดยใช้ฮาร์ดแวร์ (Hardware) ณ.จุดเอาต์พุต 1 จะได้พริตตี้คู่ ส่วนเอาต์พุต 2 จะได้พริตตี้คี่ โดยพริตตี้บิตจะใช้ทั้งฝ่ายรับ-ส่งข้อมูล เพื่อตรวจสอบความผิดพลาดข้อมูลแต่ละตัวอักษร

ข้อดีของการตรวจสอบแบบพริตตี้บิต คือ เป็นวิธีการที่ง่าย (Simplicity) แต่จะเกิดข้อเสียเมื่อจำนวนบิตเป็นจำนวนคู่ ถ้าด้านรับเกิดความผิดพลาด การตรวจสอบพริตตี้จะไม่สามารถตรวจสอบได้ถูกต้อง เช่น ถ้าเกิด 2 บิต เกิดการเปลี่ยนแปลงค่าพริตตี้บิตจะยังคงเดิม

2.2.2 การตรวจซ้ำในแนวตั้งและแนวนอน

การตรวจซ้ำในแนวตั้ง (Vertical Redundancy Checking : VCR) เป็นการตรวจสอบความผิดพลาดข้อมูลโดยใช้พริตตี้ในการกำหนด ถ้าการส่งเกิดความผิดพลาดในตัวอักษรระ ดังนั้นการตรวจซ้ำในแนวตั้งบางครั้ง เรียกว่า พริตตี้ตัวอักษร (Character parity)

การตรวจซ้ำในแนวนอน (Horizontal or Longitude Redundancy checking : HRC or LRC) เป็นการตรวจสอบความผิดพลาดโดยใช้พริตตี้ในการกำหนด ถ้าการส่งเกิดความผิดพลาดในข้อความ ดังนั้น LRC บางครั้งจึงเรียกว่า พริตตี้ข้อความ (Message parity)

จากตารางที่ 2.2 ค่าของการตรวจซ้ำในแนวนอน LRC จะได้เท่ากับ 2FH หรือ (00101111) ส่วนค่าการตรวจซ้ำในแนวตั้ง VRC แต่ละตัวอักษรจะคิดในแนวตั้ง ส่วนบิต LRC จะคำนวณในแนวนอนในกลุ่มของตัวอักษรที่ทำเป็นข้อความ (เช่น THE CAT) เรียกว่า กลุ่มของข้อมูล (Block of data) ดังนั้นลำดับบิตของ LRC เรียกว่า การตรวจสอบกลุ่มตัวอักษร (Block Check Character : BCC) หรือกลุ่มการตรวจสอบลำดับกลุ่ม (Block Check Sequence: BCS)

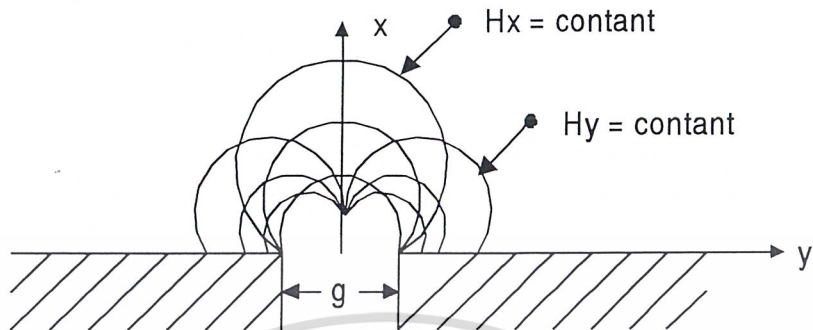
Character		T	H	E	SP	C	A	T	LRC(→)
Hex		54	48	45	20	43	41	54	2F
LSB	b0	0	0	1	0	1	1	0	1
	b1	0	0	0	0	1	0	0	1
	b2	1	0	1	0	0	0	1	1
	b3	0	1	0	0	0	0	0	1
	b4	1	0	0	0	0	0	1	0
	b5	0	0	0	1	0	0	0	1
MSB	b6	1	1	1	0	1	1	1	0
VRC(↓)	b7	0	1	0	0	0	1	0	0

ตารางที่ 2.2 แสดงการตรวจซ้ำในแนวตั้งและแนวนอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การบันทึกเทปแม่เหล็กในระบบดิจิตอล (Magnetic tape recording)

ในการบันทึกสัญญาณลงบนเทปแม่เหล็กและการนำสัญญาณคืนกลับหรือเล่นกลับ (Playback) จะเกี่ยวข้องกับสนามแม่เหล็ก (Magnetic field) ดังแสดงในรูปที่ 2.5

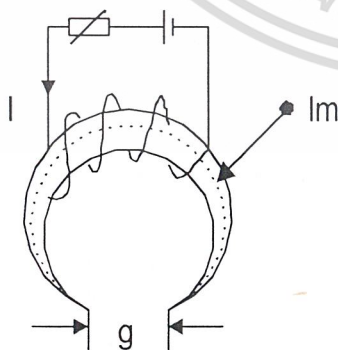


รูปที่ 2.5 แสดงความเข้มสนามแม่เหล็กที่เกิดขึ้นบริเวณหัวเทป

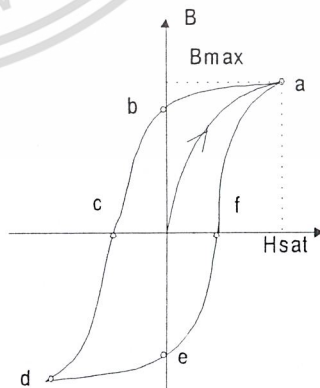
การอธิบายเบื้องต้นของการบันทึกลงบนพื้นผิวแม่เหล็ก จะเกี่ยวข้องกับกระบวนการทำให้เกิดความเป็นแม่เหล็ก (Magnetization) ในช่วงพื้นที่นั้นของส่วนที่จะถูกบันทึก (Recording medium) บริเวณช่องว่าง (Air gap) ของหัวบันทึก จะเกิดฟลักซ์แม่เหล็กรั่วในบริเวณความอึดตัวของแก๊ป (Gap) ผ่านชั้นบางๆของวัสดุที่มีความเป็นแม่เหล็กเมื่อหัวบันทึกได้รับแรงเคลื่อนแม่เหล็กไฟฟ้า

$$\int H \cdot dl = N \cdot I = F = \text{mmf.}$$

เมื่อบันทึกถูกบันทึกถึงจุดอึดตัว (Saturation) ในความเป็นแม่เหล็กของวัสดุแต่ละชนิดจะเกิดความสัมพันธ์ของความหนาแน่นของเส้นแรงแม่เหล็ก (B) กับความเข้มแม่เหล็ก (H) ซึ่งความสัมพันธ์นี้จะเกิดฮิสเทอรีซิสลูป (Hysteresis loop)



(a) magnetic head



(b) hysteresis loop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 2.6 แสดงฮิสเทอรีซิสที่เกิดขึ้นในหัวเทป
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.6(a) แสดงโครงสร้างเบื้องต้นของหัวเทปในลักษณะวงจรแม่เหล็กเมื่อคิดเป็นวงจรแม่เหล็ก จากทฤษฎีของเคอร์ซอร์ฟกล่าวไว้ว่า “ผลรวมของกระแสทั้งหมดของวงจรเท่ากับศูนย์”

$$\sum I = 0$$

$$\int H \cdot dl = I = 0$$

$$H_m \cdot l_m + H_g \cdot g = 0$$

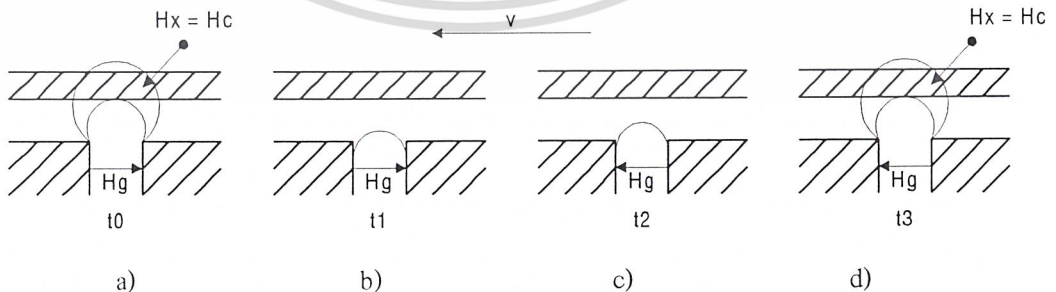
$$H \approx H_m = -H_g \cdot (g/l_m)$$

เมื่อ H_g เป็นความเข้มสนามแม่เหล็กที่เกิดในช่องว่าง (Gap)

g เป็นความยาวของช่องว่าง (Gap) เมื่อ $g \ll l_a$ สามารถตัดทิ้งได้

l_a เป็นเส้นทางเดินของเส้นแรงแม่เหล็กในอากาศในระนาบทางครึ่งวงกลม

จากรูป 2.6 (b) แสดงความสัมพันธ์ระหว่างความหนาแน่นของสนามแม่เหล็ก (B) กับความเข้มของสนามแม่เหล็ก (H) ในสารเฟอร์โรแมกเนติกที่เดิมไม่มีอำนาจแม่เหล็กอยู่เลยจะพบว่าเมื่อเพิ่มค่า H ขึ้นเรื่อยๆ ในช่วงแรกๆ B จะเพิ่มขึ้นอย่างรวดเร็วแต่เมื่อ H มีค่าสูงขึ้น อัตราการเพิ่มของ B จะมีค่าน้อยและแทบจะไม่เพิ่มเลยคือ H เริ่มมีค่าอิ่มตัว (Saturate) ณ . จุด a ให้ค่า B มีค่าสูงสุด (Bmax) จากนั้นถ้าลดค่า H ลงจนกระทั่ง $H = 0$ ณ . จุด b ช่วงนี้ค่า B จะยังไม่เท่ากับศูนย์เนื่องจากขนาดของความหนาแน่นสนามแม่เหล็กยังคงมีเหลืออยู่ (Retentivity or Remanance) ในสารแม่เหล็ก เรียกว่า B_r ต่อมาเปลี่ยนทิศทางการป้อนกระแสไฟฟ้า คือ เพิ่มค่า H ในทางลบโดยค่อยๆเพิ่ม H ไปเรื่อยๆจน $H = -H_c$ ณ . จุด c ทำให้ B มีค่าเท่ากับศูนย์ จากนั้นเพิ่มค่า H ในทางลบไปเรื่อยๆ อีก ($H < -H_c$) จนกระทั่งเกิดการอิ่มตัวในทางลบอีกครั้งหนึ่ง ($-H_{sat}$) ณ . จุด d ทำให้ B มีค่าสูงสุดทางลบอีกครั้ง ($-B_{max}$) ต่อจากนั้นเพิ่มค่า H จนกระทั่งถึงจุด e ทำให้ $B = -B_r$ และเพิ่ม H ขึ้นอีกเรื่อยๆจนกระทั่งถึงจุด f จึงจะได้ $H = H_c$ โดย $B = 0$ จากนั้นถ้าต้องการให้ $B = B_{max}$ ก็เพิ่ม H ให้เท่ากับ H_{sat} อีกครั้งจะเป็นวงรอบของความสัมพันธ์เช่นนี้เรื่อยๆ วงรอบของความสัมพันธ์ของ B-H นี้เรียกว่าฮิสเทอรีซิสลูป (Hysteresis loop)



รูปที่ 2.7แสดงการบันทึกในช่วงอิ่มตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 2.7 a) ในช่วง $t(0)$ เมื่อให้สนามที่หัวเทปอยู่ที่ระดับสูงสุด (Peak level) ในช่วงนี้เกิดความเข้มสูงสุด (High field) ของตัวกลางซึ่งได้แก่เส้นเทป ขณะที่เส้นเทปเคลื่อนไปด้วยความเร็ว (v) ในทิศทางที่แสดง ที่ $t(1)$ Hg มีค่าใกล้ศูนย์จะทำให้เส้นเทปคงความเป็นแม่เหล็กในช่วง Br ที่ $t(2)$ กระแสและ Hg ที่ช่องว่างจะเริ่มเพิ่มขึ้นในทิศทางตรงกันข้ามและที่ $t(3)$ จะเกิดความเข้มสูงสุดอีกครั้งในทิศทางตรงกันข้ามกับช่วง $t(1)$ ดังแสดงในรูปที่ 2.7 d)

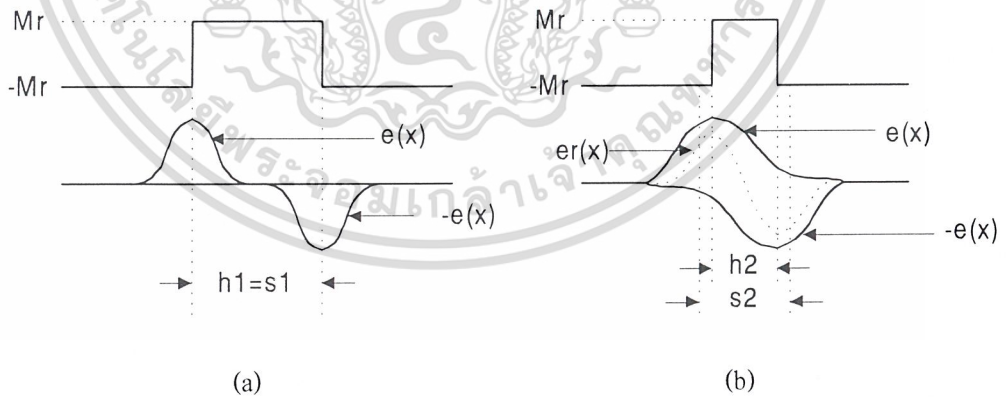
2.4 การกลับคืนสัญญาณของการบันทึก (Reproduction process)

จากการบันทึกสัญญาณผ่านหัวเทปลงบนเส้นเทป ทำให้เส้นเทปมีคุณสมบัติความเป็นแม่เหล็ก (Magnetization) ในการอ่านกลับจะได้แรงดัน ณ จุดหัวอ่านแปรผันกับเส้นแรงแม่เหล็กที่ผ่านมายังหัวอ่าน

$$e = -n \cdot (d\phi/dt) = -n \cdot (d\phi/d\lambda) \cdot (dx/dt) = -n \cdot V \cdot (d\phi/dx)$$

- เมื่อ V เป็นความเร็วของเส้นเทป
- n เป็นจำนวนรอบขดลวดของหัวอ่าน
- x เป็นระยะทางของเส้นเทป

จุดประสงค์หลักในการสร้างความกว้างและขนาดของพัลส์ จะเกิดขึ้นจากการอ่านค่าในช่วงการเปลี่ยนแปลงจากจุดอิมตัวหนึ่งไปยังอีกจุดหนึ่ง

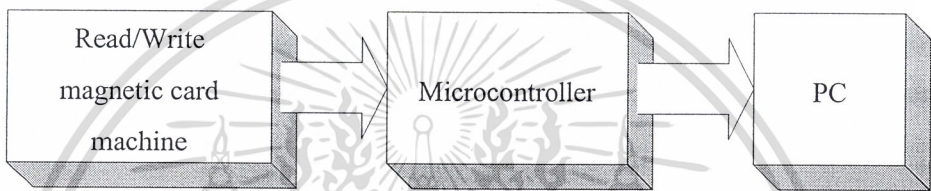


รูปที่ 2.8 แสดงสัญญาณที่อ่านได้จากหัวอ่าน

จากรูปที่ 2.8 การได้ค่าแรงดันซึ่งเป็นสัญญาณที่อ่านได้จากเส้นเทปที่ถูกบันทึกด้วยดิจิทัลเป็นลักษณะพัลส์หรือขบวนการเปลี่ยนแปลงของฟังก์ชันขั้นบันได (Step-function) จะได้รับแรงดันที่เกิดขึ้น เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า จากการวางซ้อนเชิงเส้น (Linear super position) ในรูป 2.8 a) จุดอิมตัวที่มีการเปลี่ยนแปลงจากช่วงบวกไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปช่วงลบ การเปลี่ยนแปลงในช่วงถัดมาอยู่ห่างกัน จึงไม่เกิดการกระทำร่วมกันของการเปลี่ยนแปลงทั้งสอง แต่ในรูป 2.8 b) เกิดการกระทำร่วมกันระหว่าง 2 พัลส์ เพราะว่าการเปลี่ยนแปลงของขนาดและการแบ่งแยะระหว่าง 2 ระดับค่าสูงสุด (s) จะมีค่ามากกว่าการแบ่งแยะระหว่างกระแสที่อ่านได้จากกรกกลับขั้ว (Reversal) (h) โดยในรูป 2.8 b) ค่า e(x) แสดงแรงดัน Output จากการเปลี่ยนแปลงไปช่วงบวก ส่วน -e(x) แสดงแรงดันเอาท์พุทจากการเปลี่ยนแปลงไปช่วงลบ และ er(x) เป็นผลลัพธ์ที่ได้จากการวางซ้อน (Superposition) การวางซ้อนของขบวนช่วงการเปลี่ยนแปลงใกล้เคียงกันจะเกิดการกระทำร่วมกันเป็นผลทำให้ขนาดของสัญญาณที่ได้ลดลงและเกิดการเลื่อนเฟสของสัญญาณในสัญญาณ er(x)

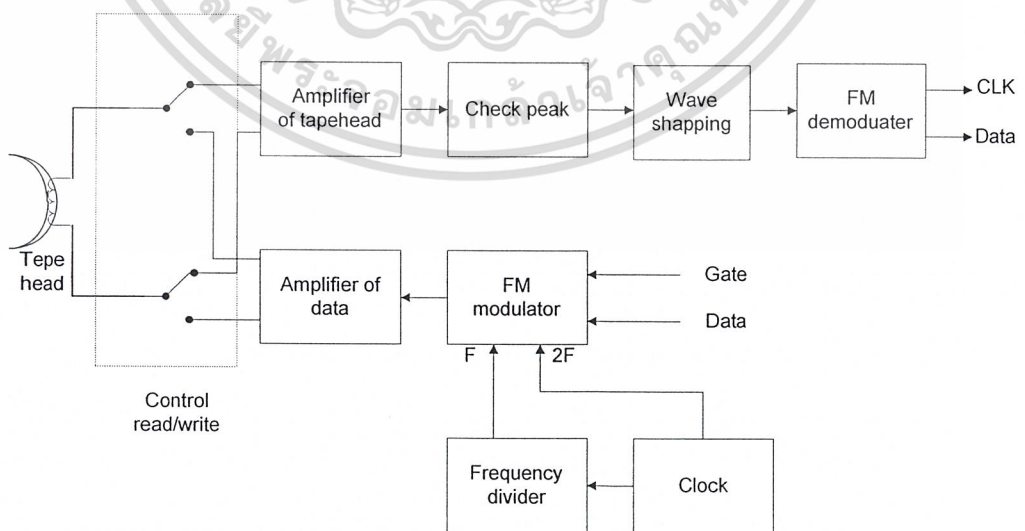
2.5 หลักการเบื้องต้นของเครื่องเก็บข้อมูลโดยผ่านบัตรแถบแม่เหล็ก



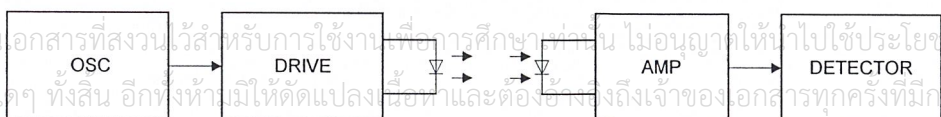
รูปที่ 2.9 บล็อกไดอะแกรมเบื้องต้นของเครื่องเก็บข้อมูลโดยผ่านบัตรแถบแม่เหล็ก

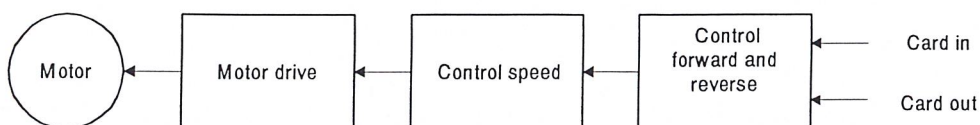
จากรูปที่ 2. 9 ได้แสดงบล็อกไดอะแกรมเบื้องต้นของเครื่องเก็บข้อมูลโดยผ่านบัตรแถบแม่เหล็ก ซึ่งจะประกอบไปด้วย 3 ส่วนหลักๆในการทำงาน โดยแต่ละส่วนจะมีหลักการทำงานดังนี้

2.5.1 การอ่านและบันทึกข้อมูลผ่านบัตรแม่เหล็ก



เอกสารนี้เป็น เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลง เนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



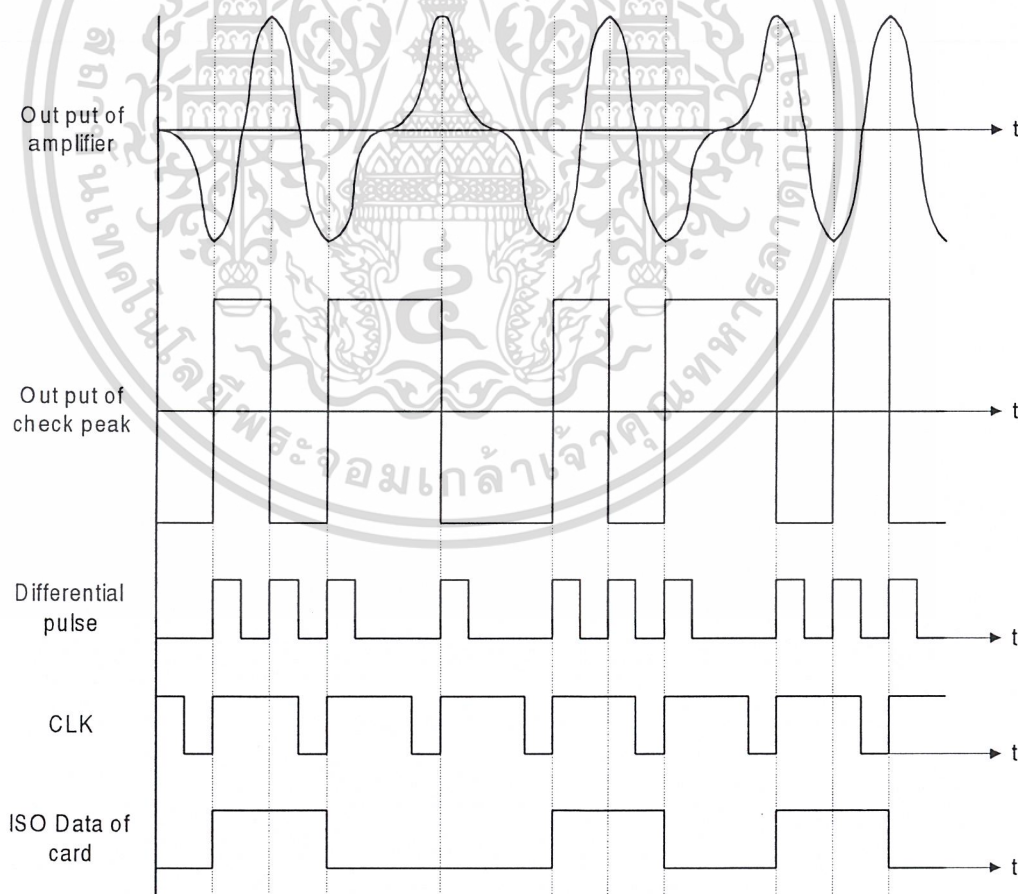


รูปที่ 2.10 บล็อกไดอะแกรมของเครื่องและบันทึกข้อมูลของบัตร

2.5.1.1 ส่วนอ่านรหัสข้อมูลของบัตร

การอ่านข้อมูลจากบัตรแม่เหล็ก ทำได้โดยให้หัวอ่านสัมผัสกับแถบแม่เหล็กซึ่งเคลื่อนที่ด้วยความเร็วคงที่ในรูปที่ 2.11 ฟลักซ์ (FLUX) ที่เกิดจากแม่เหล็กถาวรขนาดเล็กบนแถบแม่เหล็กจะผ่านจากแกนของหัวอ่านไปยังแกน (CORE) การเปลี่ยนแปลงของฟลักซ์ตามข้อมูลนั้น

จุดสูงสุด (PEAK) ของแรงดันที่อ่านได้นั้นจะตรงกับจุดที่สนามแม่เหล็กบนแถบแม่เหล็กกลับทิศทางพอดี ดังนั้นถ้าขยายแรงดันนี้ขึ้นและตรวจหาจุดสูงสุด (PEAK) ด้วยวิธีตรวจจับความแตกต่าง (DIFFERENTIAL) แล้วเปลี่ยนเป็นสัญญาณพัลส์ก็จะได้ข้อมูลที่บันทึกอยู่ในบัตรแม่เหล็ก (รูปที่ 2.6)

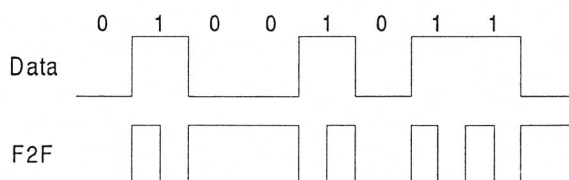


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จาก Block diagram รูปที่ 2.10 เป็นหลักการเบื้องต้นของส่วนอ่านและบันทึกข้อมูลในบัตรแม่เหล็ก
- Amplifier of tapehead ทำหน้าที่ขยายสัญญาณของหัวเทปที่อ่านข้อมูลบนบัตรแม่เหล็ก
 - Check peak ทำหน้าที่ตรวจจบบรรทัดสัญญาณสูงสุดและต่ำสุดของสัญญาณที่ผ่านการขยายสัญญาณแล้ว เพื่อที่จะได้สัญญาณพัลส์ในช่วงเวลาต่างๆกัน (Differential pulse)
 - Wave shaping ทำหน้าที่ปรับเปลี่ยนรูปสัญญาณให้มีขนาดเพียงพอ ในการนำไปถอดรหัสข้อมูล
 - FM demodulator ทำหน้าที่ถอดรหัสข้อมูลจากสัญญาณพัลส์ที่เป็นแบบ Differential pulse ซึ่งผ่านการปรับระดับสัญญาณให้เพียงพอที่จะให้ Microcontroller ถอดรหัสได้ โดยส่วนนี้จะมีส่วนที่เป็น Data และ Clk กลับคืนมา

2.5.1.2 ส่วนการบันทึกข้อมูล

การบันทึกข้อมูลลงบนบัตรแม่เหล็กจะใช้วิธีการบันทึกแบบคิวิตอล ในลักษณะเช่นเดียวกับที่ใช้ในแผ่น ฟลอปปีดิสก์ หรือ เทปแม่เหล็กสำหรับเครื่องคอมพิวเตอร์ทั่วไป การบันทึกข้อมูลลงบนแถบแม่เหล็กนั้นจะต้องป้อนกระแสพัลส์ซึ่งมีทั้งด้าน + และด้าน - พร้อมทั้งมีขนาดเพียงพอเข้าที่ขดลวดของหัวบันทึกซึ่งคอยูนกับแถบแม่เหล็กที่เคลื่อนด้วยความเร็วคงที่ แถบแม่เหล็กจะถูกเปลี่ยนให้มีรูปแบบของขั้วแม่เหล็กตาม พัลส์แม่เหล็กที่ไหลจาก แก๊ปของหัวบันทึก แถบแม่เหล็กจะเกิดเป็นแม่เหล็กถาวรขนาดเล็กเรียงตัวกันตามขั้ว + หรือ - ของพัลส์และความกว้างของพัลส์สัญญาณที่บันทึกเนื่องจากกระแสพัลส์ที่ใช้ในการบันทึกมีขนาดเพียงพอที่หัวบันทึก จะทำให้แถบแม่เหล็กมีสนามแม่เหล็กอ้อมตัวได้ดังนั้นเมื่อทำการบันทึกข้อมูล ตัวข้อมูลที่เคยมีอยู่จะถูกเขียนทับและหายไป เหลือเพียงข้อมูลใหม่เท่านั้น รูปแบบการบันทึกข้อมูลลงบนบัตรแม่เหล็กส่วนใหญ่ จะใช้รูปแบบ F2F หมายถึง การเข้ารหัสแบบสองความถี่ (Two Frequency Coherent Encoding) และ แบบการมอดูเลตเชิงความถี่ (Frequency Modulation : FM) เนื่องจากการบันทึกข้อมูลจะเป็นลักษณะเปลี่ยนแปลงทางความถี่ โดยการบันทึกข้อมูลในลักษณะเช่นนี้จะบันทึกข้อมูล (Data) และสัญญาณนาฬิกา (Clock) เข้าไว้ในแทร็คเดียวกันนอกเหนือจากนี้ยังมีการบันทึกที่ข้อมูลและแทร็ค แยกคนละแทร็คเช่นแบบ NRZI (Non Return To Zero Invented Recoding) ซึ่งมีความจุในการบันทึกค่า



รูปที่ 2.12 แสดงลักษณะการบันทึกแบบ F2F

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาด้านนี้ เมื่อนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

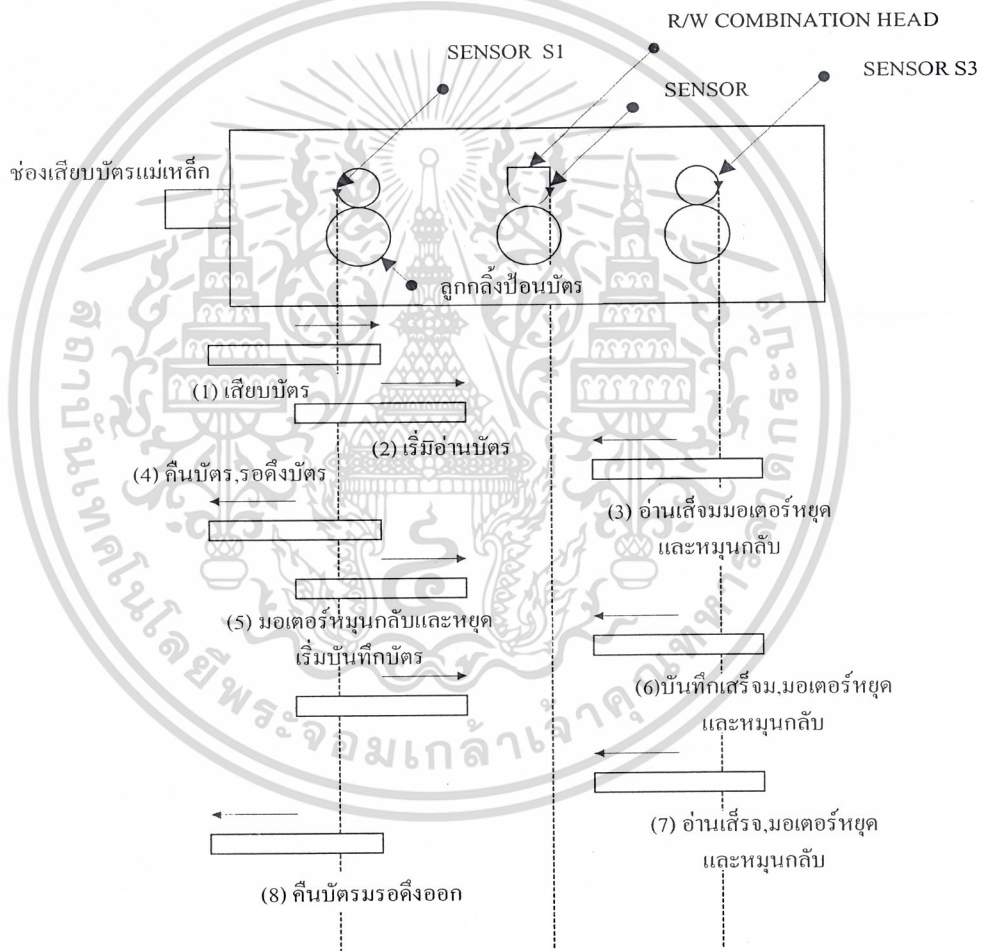
2.5.1.3 ส่วนควบคุมมอเตอร์

ในส่วนการควบคุมมอเตอร์ จะประกอบด้วยวงจรถอดจิกสำหรับควบคุมทิศทางการเดินหน้าและถอยหลัง และวงจรวอร์เตอร์สำหรับขับมอเตอร์

วงจรถควบคุมความเร็วจะต้องควบคุมไม่ให้ความเร็วผิดพลาดแม้แต่น้อย ดังนั้นการตรวจวัดความเร็วจึงต้องใช้พัลส์ ในการควบคุมความเร็ว

2.5.1.4 ส่วนตรวจจับบัตร

ตัวตรวจจับบัตรต้องใช้โฟโตไดโอดที่ให้แสงในย่านอินฟราเรด และโฟโตทรานซิสเตอร์ หากใช้งานเครื่องในสถานที่ที่มีฝุ่นผงมากจะต้องทำความสะอาดตัวตรวจจับบัตรเป็นประจำ

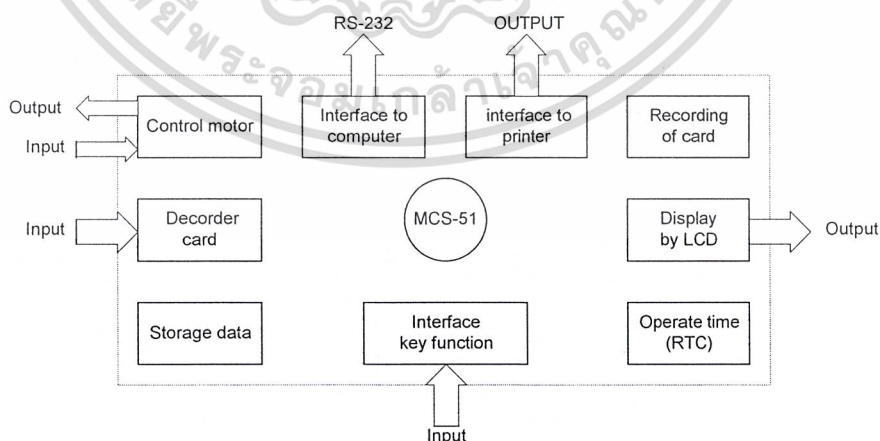


รูปที่ 2.13 การทำงานของเครื่องอ่าน - เขียนบัตรแม่เหล็กแบบป้อนบัตรด้วยมอเตอร์

ตัวอย่างการทำงานของเครื่องอ่าน - เขียนบัตรแม่เหล็กแบบป้อนบัตรด้วยมอเตอร์จะอยู่ในรูปที่ 2.13 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า และมีลำดับการทำงานดังนี้ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เมื่อเสียบบัตรแม่เหล็กเข้าที่ช่องเสียบ ตัวตรวจจับที่ทางเข้าของช่องเสียบจะทำงานทำให้มอเตอร์เดินหน้าป้อนบัตรเข้าไปข้างในเครื่อง
 2. เมื่อบัตรเคลื่อนที่มาตำแหน่งหัวอ่าน - เขียนจะมีสัญญาณ READ GATE ทำให้หัวจรอ่านทำงานข้อมูลที่บันทึกอยู่บนบัตรจะถูกอ่านด้วยหัวอ่านของ COMBINATION HEAD
 3. เมื่อบัตรเคลื่อนผ่านหัวอ่านจนหมดสัญญาณ READ GATE จะหายไปและมอเตอร์จะหยุดหมุน
 4. ข้อมูลที่อ่านได้จะถูกเช็คด้วยอุปกรณ์ภายนอกเครื่องหรือคอมพิวเตอร์ และจะนำไปประมวลผล กรณีที่เป็นเครื่องที่อ่านบัตรอย่างเดียว มอเตอร์จะหมุนกลับทางเพื่อส่งบัตรแม่เหล็กกลับไปยังช่องเสียบ รอการดึงออก
 5. กรณีที่ต้องการเปลี่ยนแปลงข้อมูลบนบัตร เมื่อบัตรเคลื่อนกลับผ่านหัวอ่าน - เขียนจนหมด มอเตอร์จะหยุดก่อนหน้าจะถึงช่องเสียบบัตร จากนั้นมอเตอร์จะเดินหน้าป้อนบัตรเข้าไปอีกจนถึงหัวอ่าน - เขียนจึงเริ่มบันทึกข้อมูลใหม่ลงบนบัตร
 6. เมื่อบันทึกข้อมูลเสร็จมอเตอร์จะหมุนกลับและหยุดจากนั้นจะเดินขึ้นตอน [2] [3] และ [4] ซ้ำเพื่อเช็คข้อมูลซ้ำอีกครั้ง
 7. ขั้นสุดท้ายจะทำการคืนบัตรออกมาช่องเสียบและรอการดึงออก
- ดังที่กล่าวไว้ข้างต้นว่า เครื่องอ่าน - เขียนบัตรแม่เหล็กมีอยู่หลายชนิด ผู้ที่จะนำเครื่องมาใช้งานจะต้องทำความเข้าใจลักษณะของเครื่องที่เหมาะสมต่องานที่ใช้ให้ดีที่สุด โดยเฉพาะอย่างยิ่งถ้าเป็นบัตรแบบพิเศษ หากไม่ระมัดระวังอาจทำให้ความเชื่อถือของระบบลดลงได้

2.5.2 ส่วนควบคุมระบบ (Control system)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการรูปที่ 2.14 โครงสร้างของระบบควบคุมให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของการควบคุมการทำงานของระบบทั้งหมดจะใช้ไมโครคอนโทรลเลอร์ในการควบคุมการทำงาน ดังรูปที่ 2.14 โดยมีหลักการดังนี้

- ควบคุมมอเตอร์ (Control motor) ส่วนนี้จะรับผลของการตรวจจับบัตร (Sensor) เข้ามาทาง อินพุทแล้วจะประมวลผลออกทางเอาท์พุทในการควบคุมมอเตอร์ ตามผลที่ได้รับจาก ส่วนของการตรวจจับบัตร
- การถอดรหัสบัตร (Decode I.D. number of card) ส่วนนี้จะรับผลของส่วน เครื่องอ่านบัตรแถบแม่เหล็ก (Read magnetic card) ซึ่งสัญญาณที่ได้รับเข้ามาจะเป็นสัญญาณ ไบนารี (Binary signal) ดังนั้นส่วนนี้จะประมวลผลได้เป็นรหัสข้อมูลทีละ 5 บิต
- การเก็บข้อมูล (Data storage) ส่วนนี้จะเป็นส่วนที่เก็บข้อมูลของรหัสบัตรที่สามารถจะใช้บริการกับระบบได้ ซึ่งถ้าบัตรใดไม่มีรหัสบัตรอยู่ในส่วนนี้ก็จะไม่สามารถใช้บริการได้
- การตรวจสอบบัตร (Check card) ส่วนนี้จะเป็นส่วนที่นำส่วนการถอดรหัสบัตรมาเปรียบเทียบกับส่วนการเก็บข้อมูล เมื่อบัตรใดมีรหัสบัตรในส่วนการเก็บข้อมูลก็จะให้สามารถใช้บริการกับระบบได้
- การจัดการบันทึกบัตร (Record to card) ส่วนนี้จะรับข้อมูลรหัสบัตรจากส่วนของการจัดการ Key function โดยข้อมูลที่รับจาก Key function เป็น 8 บิต ใน 1 ตัวอักษร ส่วนการบันทึกบัตรจะเปลี่ยนรหัสตัวอักษรเป็น 5 บิต พร้อมทั้งให้เป็นไปตามรูปแบบ (Format) ของการบันทึกบัตรแถบแม่เหล็ก
- การจัดการ Key function ส่วนนี้จะรับข้อมูลจากการกดแป้นคีย์บอร์ด (Keyboard)แล้วนำมาประมวลผลเก็บไว้เป็นค่ารหัส เพื่อนำค่ารหัสนี้ไปใช้ในส่วนอื่นๆ
- การแสดงผล LCD ส่วนนี้จะนำข้อมูลใน CENTER แสดงผลที่จอ LCD
- การจัดการทางเวลา (RTC) ส่วนนี้จะทำหน้าที่เป็นฐานเวลาจริงให้กับระบบ โดยจะแสดงวัน เดือนปี ชั่วโมง นาที และวินาที
- การบันทึกการใช้บริการ ส่วนนี้จะทำการบันทึกการใช้บริการหลังจากบัตรได้รับการตรวจสอบแล้วให้ใช้บริการได้ ซึ่งจะเริ่มบันทึกเวลาเลิกใช้บริการ จนกระทั่งเลิกติดต่อกับระบบนำบัตรคืน จะบันทึกเวลาเลิกใช้บริการพร้อมทั้งบันทึกจำนวนเวลาที่ใช้บริการไว้

2.5.3 ส่วนแสดงผล

ในส่วนของการแสดงผลสามารถแสดงผลผ่านทางแอลซีดี (LCD) และผ่านทางคอมพิวเตอร์ได้ โดยในส่วนของคอมพิวเตอร์จะเป็นส่วนเก็บและแสดงผลข้อมูลส่วนบุคคลของผู้ถือบัตร ซึ่งในโครงการนี้จะประกอบด้วยข้อมูลต่างๆ คือ

1. ประวัติประจำตัวผู้ถือบัตร
2. เวลาเข้า – ออก ของการใช้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามแก้ไขตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การลงทะเบียน
4. ผลการศึกษา

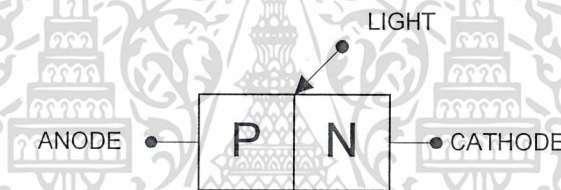
ข้อมูลเหล่านี้จะถูกเก็บในลักษณะเป็นฐานข้อมูล (Database) โดยใช้โปรแกรม Microsoft Access ในการเก็บข้อมูลต่างๆ แล้วแสดงผลด้วยโปรแกรม Visual Basic ซึ่งโปรแกรมนี้จะเป็นโปรแกรมหลักในการอินเตอร์เฟสระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ ผ่านทางพอร์ตอนุกรม RS-232C

2.6 โฟโตไดโอด (Photodiode)

แสงคือ การแผ่รังสีของสนามแม่เหล็กไฟฟ้า ซึ่งมองเห็นด้วยตาเปล่า และแสงจะเดินทางในรูปของคลื่นวิทยุ ช่วงความถี่ของแสงคือ 300 จิกะเฮิรตซ์ ถึง 300,000,000 จิกะเฮิรตซ์ ซึ่งจะความถี่ดังนี้

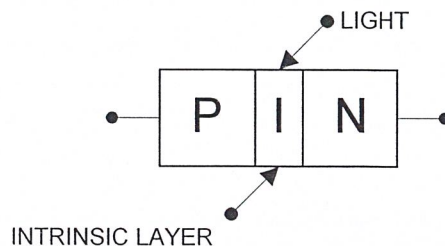
แสงที่มองเห็นด้วยตาเปล่า อยู่ในช่วงความถี่ 400,000 จิกะเฮิรตซ์ ถึง 750,000 จิกะเฮิรตซ์ โดยที่แสงอินฟราเรดอยู่ต่ำกว่า 400,000 จิกะเฮิรตซ์ แสงอุลตราไวโอเล็ตสูงกว่า 750,000 จิกะเฮิรตซ์

โฟโตไดโอดคือ อุปกรณ์สารกึ่งตัวนำที่ทำจากซิลิกอน เป็นอุปกรณ์ตรวจจับแสง และมันสามารถถูกสร้างได้สองวิธี ซึ่งวิธีแรกจะทำการเป็นแบบต่อรวมกันของสารพี-เอ็น ดังแสดงในรูปที่ 2.15 และอีกวิธีหนึ่งคือการใช้อินเทนซิก ระหว่างสารพีกับเอ็น ดังแสดงในรูปที่ 2.15 ซึ่งจะอยู่ในรูปของพี-ไอ-เอ็น โฟโตไดโอด (PIN photodiode)



รูปที่ 2.15 แสดง พี-เอ็น จังก์ชันโฟโตไดโอด

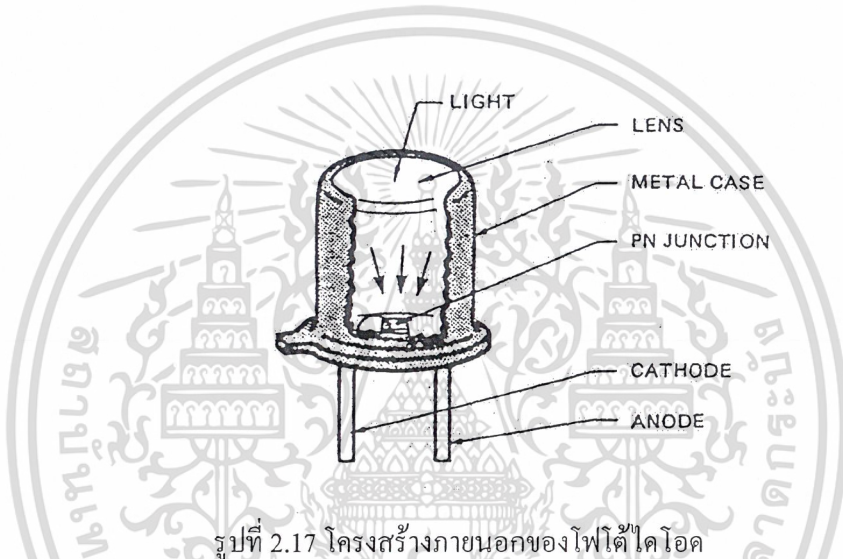
โฟโตไดโอดแบบรอยต่อพี-เอ็น (PN Junction Photodiode) จะมีหลักการการทำงานเหมือนโซลาร์เซลล์ แต่แตกต่างกันตรงที่มันถูกใช้ในการควบคุมไหลของกระแส ไม่ได้เป็นตัวกำเนิดและเมื่อมีแรงเคลื่อนไฟฟ้าถูกจ่ายคร่อมโฟโตไดโอดแบบรีเวิร์ดไบอัส ซึ่งทำให้ขอบเขต (Depletion Region) ถูกขยายให้กว้างขึ้น และเมื่อพลังแสงตกกระทบที่ตัวโฟโตไดโอดในทิศทางรีเวิร์ดและเมื่อพลังงานแสงเพิ่มขึ้นมากเท่าไร อิเล็กตรอนอิสระก็จะยิ่งขึ้นตาม ก็จะให้มีกระแสไหลผ่านมากขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 2.16 แสดง พี-ไอ-เอ็น จังก์ชันโฟโตไดโอดไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พี-ไอ-เอ็น โฟโอดีไดโอด จะมีชั้นอินเทนซิติระหว่างสารพีกับสารเอ็นซึ่งจะส่งผลให้ขอบเขต ศีลพิษันถูกขยาย การที่ขอบเขตศีลพิษันกว้างขึ้นจะทำให้พี-ไอ-เอ็น โฟโอดีไดโอด ตอบสนองกับความถี่ ต่ำ ซึ่งแสงความถี่ต่ำจะมีพลังงานต่ำ และจะต้องเจาะเข้าไปในขอบเขตศีลพิษันลึกขึ้น เพื่อจะทำให้เกิด อิเล็กตรอนอิสระและขอบเขตศีลพิษันกว้างมากขึ้นยังทำให้หนทางที่เกิดอิเล็กตรอนอิสระมาก ส่วนการ ตอบสนองต่อการเปลี่ยนแปลงของความเข้มแสงจะเร็วกว่าโฟโอดีไดโอดชนิดรอยต่อพี-เอ็น ซึ่งข้อดีของ โฟโอดีไดโอด คือมีความไวในการตอบสนองต่อความเปลี่ยนแปลงของแสงและยังเป็นอุปกรณ์ตรวจจับ แสงที่ไวที่สุดด้วย

ข้อเสียของโฟโอดีไดโอด คือจะให้เอาที่พุดำเมื่อเทียบกับอุปกรณ์ตรวจจับแสงอื่นๆจากรูปที่ 2.17 แสดงโครงสร้างภายนอกของโฟโอดีไดโอด



รูปที่ 2.17 โครงสร้างภายนอกของโฟโอดีไดโอด

2.7 อุปกรณ์เปล่งแสง (Light – Emitting Devices)

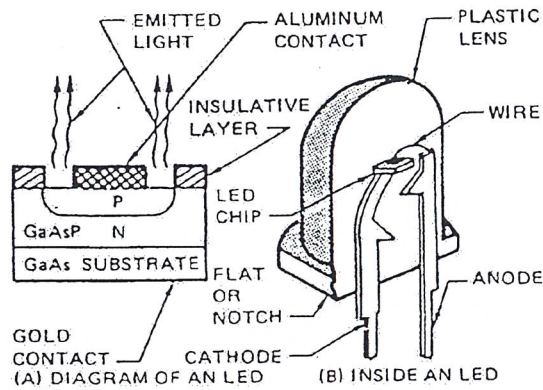
อุปกรณ์เปล่งแสงจะเปล่งแสงออกมาเมื่อมีกระแสไหลผ่านอุปกรณ์นั้นๆ โดยมันจะเปลี่ยนพลังงาน ไฟฟ้าให้เป็นพลังงานแสง

ไดโอดเปล่งแสง เป็นอุปกรณ์เปล่งแสงสารกึ่งตัวนำที่ใช้กันมากที่สุด โดยส่วนที่เป็นสารกึ่งตัวนำจะมีอายุการใช้งานไม่จำกัด ตลอดจนกว่าไส้จะขาดไป

ไดโอดแบบรอยต่อพีเอ็น (PN Junction Diode) สามารถเปล่งแสงเมื่อมีกระแสไหลผ่าน โดยที่จะเกิดเมื่ออิเล็กตรอนรวมตัวกับโฮล และจะมีการปล่อยพลังงานที่สูงออกมาในรูปของแสง ซึ่งความถี่ของแสงที่ปล่อยออกมาพิจารณาจากชนิดของสารกึ่งตัวนำที่ใช้สร้างไดโอดและไดโอดทั่วไปจะไม่สามารถเปล่งแสงได้ เพราะว่ามันจะใช้วัสดุที่ทึบในการสร้างโครงสร้างภายนอก

ไดโอดเปล่งแสง ที่ใช้สารแกลเลียมอาร์เซไนด์ ฟอสเฟต (GaAs - P) มันจะผลิตแสงสีแดง ซึ่งสามารถมองเห็นได้ด้วยตาเปล่า และถ้าเปลี่ยนฟอสเฟตสามารถผลิตแสงที่มีความถี่แตกต่างออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 แสดง โครงสร้างของไดโอดเปล่งแสง

ในวงจรไฟฟ้า ไดโอดเปล่งแสง จะถูกต่อเป็นฟอรัเวิร์ดไบอัสถึงจะสามารถเปล่งแสงได้โดยที่แรงเคลื่อนไฟฟ้าที่ป้อนจะต้องมากกว่า 1-2 V และไดโอดเปล่งแสงจะพังได้ง่ายถ้าจ่ายแรงเคลื่อนไฟฟ้าหรือกระแสไฟฟ้าสูงเกินไป วิธีแก้ไขคือต่อตัวต้านทานอนุกรมเข้าไป เพื่อลดกระแสที่ไหลผ่าน LED เข้าไป

2.8 สถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบชิปเดี่ยวตระกูล MCS-51

ไมโครคอนโทรลเลอร์แบบชิปเดี่ยว (Single Chip Microcontroller) คือ ไมโครคอมพิวเตอร์แบบที่มีขนาดเล็กโดยบรรจุไว้ในแผงวงจรรวม (Integrated Circuit) เพียงชิปเดียวเหมาะสำหรับงานควบคุมอุปกรณ์อื่น ๆ แบบอัตโนมัติ เพราะผู้ใช้สามารถเขียนโปรแกรมควบคุมการทำงานได้ตามต้องการ ไมโครคอนโทรลเลอร์แบบชิปเดี่ยว ตระกูล 51 หรือ MCS51 อันได้แก่ เบอร์ 8051, 8052 และอื่นๆ

MCS51 มีข้อดีดังนี้

- สามารถนำเอาข้อมูลมา AND , OR หรือทำ Complement ทั้งแบบทีละ 8 บิตและ 1 บิต
- สามารถใช้กับหน่วยความจำสำหรับโปรแกรม (Program Memory) ซึ่งเป็นหน่วยความจำที่ใช้สำหรับเก็บชุดคำสั่งที่จะให้ MCS - 51 ทำงานได้สูงสุด 64 กิโลไบต์ (Kilobyte) (64 x 1024 ไบต์) ทำให้เขียนโปรแกรมควบคุมการทำงานได้มาก
- สามารถต่อกับหน่วยความจำสำหรับข้อมูล (Data Memory) ซึ่งเป็นหน่วยความจำสำหรับเก็บข้อมูลในระหว่างการทำงานของโปรแกรมได้สูงสุด 64 กิโลไบต์
- ใน 8051 และ 8751 มีหน่วยความจำสำหรับโปรแกรมนับ 4 กิโลไบต์ (ใน 8052 และ 8752 มีหน่วยความจำสำหรับโปรแกรมจำนวน 8 กิโลไบต์) อยู่ในวงจรรวมทำให้ไม่ต้องต่อหน่วยความจำสำหรับโปรแกรมอยู่ภายนอก ระบบรวมทั้งหมดจึงมีขนาดเล็กและสัญญาณรบกวนจากภายนอกจะทำให้ MCS - 51 ทำงานผิดพลาดได้ยาก
- มีพอร์ทแบบขนาน (Parallel Port) สำหรับข้อมูลเข้าและออกจำนวน 32 บิต ที่ข้อมูลแต่ละบิตเป็นอิสระต่อกัน

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในการเรียนการสอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

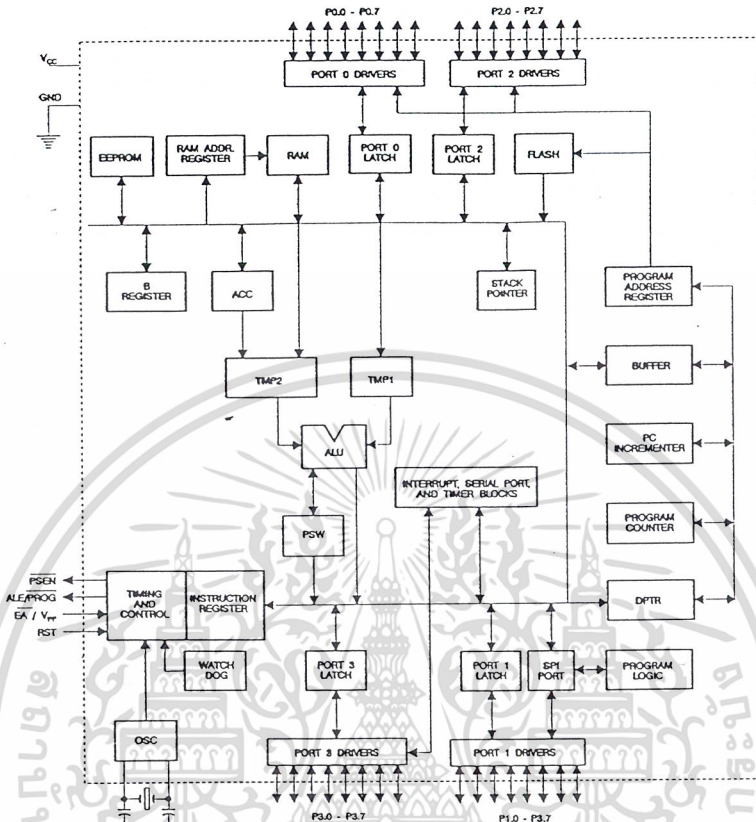
- มีวงจร Timer/Counter ขนาด 16 บิต 2 ชุด (8052 มี 3 ชุด) ที่ทำงานในโหมดต่าง ๆ ได้ถึง 4 โหมด
- มี Universal Asynchronous Receiver Transmitter (UART) สำหรับรับ-ส่งข้อมูลอนุกรม (Serial) แบบ Full Duplex ที่สามารถเลือกรูปแบบการรับ-ส่งข้อมูลได้ 4 แบบ
- มีแหล่งกำเนิดสัญญาณขอขัดจังหวะการทำงานของโปรแกรม (Interrupt Request Signal) 6 แหล่ง ซึ่งสามารถทำการโค๊ดไปทำงานตอบสนองการขัดจังหวะ (Interrupt Service Routine) ได้ต่างๆ กัน 5 ตำแหน่ง
- สามารถเลือกการทำงานให้อยู่ในโหมดของ Idle และ Power Down ซึ่งจะประหยัดการใช้กำลังไฟในการทำงาน

ในโครงงานนี้ ได้ใช้ไมโครคอนโทรลเลอร์เบอร์ AT89S8252 ของบริษัท ATMEL ซึ่งเป็นไมโครคอนโทรลเลอร์ในตระกูลเดียวกันกับ MCS-51 เช่นกัน แต่มีส่วนที่แตกต่างจาก 8051 ดังนี้

- 8 กิโลไบต์ สำหรับการโปรแกรมหน่วยความจำแบบแฟลช
- SPI การเชื่อมต่อแบบอนุกรม สำหรับการดาวน์โหลด(Down load) โปรแกรม
- อายุการใช้งาน : สามารถเขียนและลบได้ถึง 1,000 ครั้ง
- 2 กิโลไบต์ สำหรับ EEPROM
- อายุการใช้งาน : สามารถเขียนหรือลบได้ถึง 1,000 ครั้ง
- ช่วงการทำงานที่ 4.0 ถึง 6 โวลท์
- ช่วงทำงานเต็มที่ 0-24 MHz
- สามารถโปรแกรม memory Lock ได้ 3 ระดับ
- 256*8 bit ใน internal RAM
- สามารถ program ทางด้านอินพุท/เอาต์พุท(I/O) ได้ 32 เส้น
- Timer/Counter 16 bit มี 3 ตัว
- มีอินเตอร์รัพท์ 9 แหล่ง
- สามารถโปรแกรมผ่านช่องสัญญาณอนุกรมแบบ UART ได้
- มี SPI เชื่อมต่อแบบอนุกรม
- กินกำลังไฟต่ำในโหมด Idle และ Power Down
- อินเตอร์รัพท์ที่กลับคืนมาจากกำลังไฟฟ้าตกได้
- สามารถโปรแกรมเป็น watchdog timer ได้
- มีพอยน์เตอร์ข้อมูล 2 ชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.1 โครงสร้างของ AT89S8252



รูปที่ 2.19 ไคอะแกรมโครงสร้างของ AT89S8252

ภายใน AT89S8252 จะประกอบขึ้นด้วย GATE ต่างๆ กัน เช่น AND, OR, NOT ซึ่ง GATE เหล่านี้จะถูกนำมาออกแบบให้มีหน้าที่การทำงานต่างๆ เช่น วงจรถอดรหัสคำสั่ง (Instruction Decoder) , วงจรสร้างสัญญาณนาฬิกา (Clock Signal Generator) โครงสร้างภายในของ AT89S52 จะประกอบด้วยส่วนย่อยๆ ดังไคอะแกรมในรูปที่ 2.19

- CPU (Central Processing Unit) หรือตัวประมวลผล ส่วนนี้จะมีวงจรที่ทำหน้าที่สร้างสัญญาณควบคุมในการติดต่อกับส่วนอื่น ๆ เรียกว่าวงจรควบคุม (Control Unit) สัญญาณที่สร้างจากวงจรควบคุมได้แก่ สัญญาณสำหรับการติดต่อกับหน่วยความจำ , อุปกรณ์รับข้อมูลออกจากตัว AT89S8252 ซึ่งส่วนควบคุมการขัดจังหวะ (Interrupt Control) และส่วนควบคุมบัส (Bus Control) ก็เป็นส่วนหนึ่งของวงจรถอดรหัสคำสั่งด้วยที่สร้างสัญญาณควบคุมจากส่วน CPU นี้จะทำการสร้างสัญญาณโดยการถอดรหัสจากคำสั่งไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Instruction) ตามที่มีการกำหนดไว้ และสัญญาณที่สร้างขึ้นมาจะอ้างอิงกับสัญญาณนาฬิกา ที่สร้างจากวงจรรอสซิงเลเตอร์ เพื่อให้ทุกๆ ส่วนในวงจรทำงานประสานกัน (Synchronize) อย่างถูกต้อง

ใน CPU นี้ยังประกอบด้วยส่วนย่อยอีกส่วนที่เรียกว่าส่วนประมวลผล (Arithmetic Logic Unit) ส่วนนี้จะทำหน้าที่ประมวลข้อมูล เช่น การบวก , ลบ , คูณ หรือหารข้อมูล แล้วนำผลลัพธ์ไปเก็บไว้ในรีจิสเตอร์ หรือหน่วยความจำที่ต้องการ

- หน่วยความจำ (Memory) มีไว้สำหรับจัดจำข้อมูล การเอาข้อมูลไปเก็บในหน่วยความจำเรียกว่าการเขียน (Write) ข้อมูล และการนำข้อมูลออกจากหน่วยความจำจะเรียกว่าการเขียน (Read)

- อุปกรณ์อินพุตและเอาต์พุต (Input/Output Device) เป็นส่วนที่จะใช้ส่งข้อมูลเข้าหรือออก ทำให้

AT89S8252 ติดต่อกับภายนอกได้ ดังในไดอะแกรมรูปที่ 2.19 อุปกรณ์อินพุตและเอาต์พุตได้แก่ 4 I/O Port , Timer 0 , Timer 1 , Serial Port การทำงานของแต่ละส่วนมีดังนี้

- 4 I/O Port คำว่าพอร์ต หมายถึงจุดที่จะติดต่อกับส่วนที่อยู่ภายนอก 4 I/O Port ของ 8051 เป็นที่ใช้สำหรับรับ-ส่ง ข้อมูล ซึ่งเป็นสัญญาณดิจิทัลเข้าหรือออกจากตัว MCS51 พอร์ตมีทั้งหมด 4 พอร์ต โดยแต่ละพอร์ตจะรับ-ส่งข้อมูลได้ 8 บิต มีพอร์ต P0 , P1 , P2 และ P3 บางพอร์ตจะใช้ทำงานมากกว่า 1 อย่างก็ได้ เช่น พอร์ต P0 และ P2 จะใช้สำหรับการส่งค่าตำแหน่ง (Address) ของหน่วยความจำที่ต้องการติดต่อ และพอร์ต P0 จะใช้รับส่งข้อมูลเมื่อติดต่อกับหน่วยความจำได้ด้วย แต่สิ่งเหล่านี้ไม่ได้เกิดขึ้นในเวลาเดียวกัน แต่จะใช้วิธีทำงานตามลำดับ โดยควบคุมจากสัญญาณควบคุม (Control) ที่ถอดรหัสมาจากแต่ละคำสั่งที่ให้คอมพิวเตอร์ทำงานนั่นเอง และสัญญาณทั้งหมดจะอ้างอิงกับจากสัญญาณนาฬิกา

- Timer 0 และ Timer 1 เป็นวงจรนับที่สามารถกำหนดให้ทำการนับจำนวนไซเคิลของสัญญาณที่ต่อ

จากภายนอก 8051 หรือจำนวนไซเคิลของสัญญาณนาฬิกาภายใน ก็ได้ค่าจากการนับ จะพูดอ่านหรือตั้งค่าเริ่มต้นของการนับได้โดย CPU

- Serial Port หรือพอร์ตอนุกรม CPU จะอ่านและเขียนข้อมูลกับ Serial Port เป็นแบบ 8 บิต แต่ข้อมูลจะถูกส่งออกจาก 8051 เรียงไปทีละบิต ออกจากขา TXD และในการรับข้อมูลเข้าก็จะรับเข้ามาทีละบิตทางขา RXD แล้วจัดเรียงใหม่เป็น 8 บิตเพื่อให้ CPU อ่านไปใช้งานต่อไป

2.8.2 การจัดการหน่วยความจำ

หน่วยความจำของ 8051 แบ่งออกไว้เป็น 2 แบบตามลักษณะของการใช้งานคือ

1. Program Memory เป็นหน่วยความจำที่ใช้เก็บคำสั่งในรูปรหัสภาษาเครื่อง (Machine Language) ซึ่งถ้าต้องการให้ทำงานก็จะอ่านข้อมูลที่เก็บไว้ในหน่วยความจำประเภทนี้เข้าไปถอดรหัสแล้วสร้างสัญญาณควบคุมส่วนอื่น ๆ ตามการทำงานของแต่ละคำสั่งนั้น หน่วยความจำแบบนี้จะต้องเป็นแบบ Read Only Memory (ROM) และผู้ใช้ต้องเขียนข้อมูลในแต่ละตำแหน่งของหน่วยความจำเป็นรหัสภาษาเอกสารนี้เครื่องตามลำดับการทำงานที่ต้องการ (หน่วยความจำแบบ ROM เป็นแบบ Non volatile ซึ่งเมื่อปิดไฟแล้วไม่มีการสูญข้อมูลก็ไม่มีกรสูญเสีย) การเขียนข้อมูลลงไปบน ROM จะต้องใช้เครื่องมือพิเศษๆ ในระหว่างการทำงาน

ของผู้ใช้จะไม่สามารถใช้คำสั่งทำการเขียนข้อมูลลงในหน่วยความจำแบบนี้ได้ จำนวนตำแหน่งสูงสุดของหน่วยความจำแบบนี้ที่จะใช้งานได้คือ 65536 ตำแหน่ง ค่าของตำแหน่ง (Address) จะเขียนเป็นเลขฐาน 16 ได้ตั้งแต่ 0000H ถึง FFFFH ผู้ใช้จะเลือกได้ว่าเป็นตำแหน่งของ ROM ที่อยู่ภายในหรือภายนอก ถ้าต้องการให้ ทำงานตามคำสั่งที่เก็บไว้ใน ROM ภายใน ก็ให้ป้อนสัญญาณสถานะลอจิก High (1) เข้าที่ขา EA แต่ถ้าต้องการให้ทำงานในโปรแกรมที่เก็บไว้ใน ROM ภายนอกก็ให้ต่อลอจิก Low (0) เข้าที่ขา EA

2. Data Memory เป็นหน่วยความจำที่ จะใช้สำหรับพัก , เก็บข้อมูล แล้วเรียกมาใช้ใหม่ในระหว่างการทำงานของการทำงานหรือเขียนข้อมูลจากหน่วยความจำจะกระทำโดยคำสั่งที่เก็บไว้ใน Program Memory หน่วยความจำแบบนี้เป็นประเภท Random Access Memory (RAM) ถ้ามีไฟเลี้ยงอยู่ข้อมูลที่เก็บไว้จะไม่สูญหาย แต่ถ้าปิดเครื่องหรือไม่จ่ายไฟให้แก่ RAM แล้ว ข้อมูลใน RAM ก็จะสูญหายไป การสูญหายของข้อมูลไม่ได้หมายความว่าไม่มีอะไรอยู่เลย แต่เป็นการที่มีข้อมูลใหม่ ซึ่งไม่ใช่ข้อมูลที่เก็บไว้เดิมเข้ามาอยู่แทนที่ เช่นเดิมเก็บข้อมูล 18H ไว้ตำแหน่ง 1900H เมื่อเปิดไฟแล้วเปิดใหม่ ข้อมูลที่ตำแหน่ง 1900H จะไม่ใช่ 18H อาจเป็นค่าอะไรก็ได้ ซึ่งเรียกการเกิดลักษณะแบบนี้ว่าข้อมูลสูญหายไป

Table 1. AT89S8252 SFR Map and Reset Values

0FH										0FFH
0FH	B									0F7H
0E8H										0EFH
0E0H	ACC									0E7H
0D8H										0DFH
0D0H	PSW							SPCR		0D7H
0C8H	T2CON	T2MOD	RCAP2L	RCAP2H	TL2			TH2		0CFH
0C0H										0C7H
0B8H	IP									0BFH
0B0H	P3									0B7H
0A8H	IE							SPSR		0AFH
0A0H	P2									0A7H
98H	SCON	SBUF								9FH
90H	P1							WMCON		97H
88H	TCON	TMOD	TL0	TL1	TH0	TH1				8FH
80H	P0	SP	DP0L	DP0H	DP1L	DP1H	SPDR	PCON		87H

ตารางที่ 2.3 แผนผัง SFR และค่ารีจิสเตอร์เมื่อเกิดการรีเซ็ต

จากตารางที่ 2.3 เป็นการแสดงแผนผังของรีจิสเตอร์พิเศษ (SFR : Special Function Register) และค่าของรีจิสเตอร์เมื่อเกิดการรีเซ็ต โดยใน AT89S8252 จะมีรีจิสเตอร์ที่เพิ่มขึ้นมาคือ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Timer2 Register เป็นส่วนควบคุมและบอกสถานะบิต ประกอบไปด้วย T2CON(0C814),T2MOD (009H) คูรีจิสเตอร์ RCAP2H(0CAH) และ RCAP2L(0CBH) ใช้ในการจับและโหลดรีจิสเตอร์สำหรับ Timer2 ในโหมด 16 บิต auto-reload
- Watchdog and Memory Control Register (WMCON) ประกอบด้วยบิตควบคุมสำหรับ watchdog timer อยู่ในแอดเดรส 96H

WMCON(96H)

PS2	PS1	PS0	EEMWE	EEMEN	DPS	WDTRST	WDTEN
7	6	5	4	3	2	1	0

ตารางที่ 2.4 วอร์ทดอกและรีจิสเตอร์ควบคุมหน่วยความจำ

- PS2,PS1,PS2 บิตฟรีสเกลเลอร์สำหรับ watchdog timer เมื่อทั้ง 3 บิตเป็น “0” watchdog timer มีคาบเวลา 16 ms
- EEMWE Set bit นี้เป็น “1” ก่อนการเริ่มการเขียนลงบน EEPROM ด้วยคำสั่ง MOVX “0” เมื่อใช้ซอฟต์แวร์ที่เขียนลง EEPROM เรียบร้อยแล้ว
- EEMEN เป็น “1” เป็นการเข้าถึง EEPROM ภายในชิพชุดคำสั่ง MOVX พร้อมด้วย DPIR จะเข้าถึงภายในชิพ เป็น “0” ชุดคำสั่ง MOVX พร้อมด้วย DPTR จะเข้าถึงหน่วยความจำข้อมูลภายนอก
- DPS (Data Pointer Register Select) DPS = 0 เลือก DP0 DPS = 1 เลือก DP1
- WDTRST บิตนี้ set เป็น “1” โดยใช้ซอฟต์แวร์ ฟลัชจะถูกผลิต ไม่มีรีเซต watchdog timer
- RDY/BSY/ บิต WDTRST จะถูกรีเซตอัตโนมัติ RDY/BSY/ = “1” แสดงว่า EEPROM พร้อมโปรแกรม ขณะโปรแกรมทำงาน บิต BDY/BSY/ = “0” และเป็น “1” โดยอัตโนมัติ เมื่อโปรแกรมเสร็จสิ้น
- WDTEN (Watchdog Timer Enabl bit) ถ้ามีค่า = “1” เป็นการ Enable Watchdog Timer
- SPI (Serial Peripheral Interface) register ประกอบด้วย SPCR และ SPSR เป็นรีจิสเตอร์ควบคุม

SPCR (0D5H)

SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
7	6	5	4	3	2	1	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและตงยงอิงของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 รีจิสเตอร์ควบคุม SPI

- SPIE (SPI interrupt Enable) เชื่อมต่อกับ ES bit โดย SPIE = “1” SPI interrupt จะ enable
- SPE (SPI Enable) SPI = “1” enable ช่องสัญญาณ SPI และต่อกับ SS/,MOSI,MISO และ

SCK

- DORD (Data Oder) = “1” เลือก LSB ก่อนในการส่งข้อมูล
- MSTR (Master/Slave Select) MSTR = “1” เลือก Master SPI Mode
- CPOL (Clock Polarity) CPOL = “1” เป็น high เมื่อ Idle
- CPHA (Clock Phase) เป็นบิตควบคุม clock และข้อมูลระหว่าง Master กับ Slave

SPR0	SPR0	SPR1	SCK = FOSC
	0	0	4
SPR1	0	1	16
	1	0	64
	1	1	128

- SPSR - SPI Status Register

SPSR (0AAH)

SPIF	WCOL	-	-	-	-	-	-
7	6	5	4	3	2	1	0

ตารางที่ 2.6 รีจิสเตอร์แสดงสถานะของ SPI

- SPIF (SPI Interrupt Flag) เมื่อการโอนย้ายแบบอนุกรมสมบูรณ์ SPIF = “1” และ ES = “1”
- WCOL (Write Collision Flag) บิตนี้เป็น “1” ถ้า SPI ข้อมูลรีจิสเตอร์เขียนระหว่างการโอนย้าย

ข้อมูล ผลลัพธ์การอ่าน SPDR register จะ ไม่ถูกต้อง

- SPDR - SPI Data Register = 86H

Data Memory – EEPROM and RAM

ประกอบไปด้วย 2Kbyte ของ EEPROM ภายในชิพ สำหรับเก็บข้อมูล และ 256 byte ของ RAM ช่วง 128 byte ขึ้นไปของ RAM จะเป็นพื้นที่ชนานไปกับ SFR

เมื่อคำสั่งที่เข้าถึงตำแหน่งภายในหน่วย 7FH แอดเดรสโหมคจะใช้ชุดคำสั่งเฉพาะให้ CPU เข้าถึง

ช่วงเหนือ 128 byte ขึ้นไปของRAM หรือ SFR ชุดคำสั่งที่ใช้ direct addressing เข้าถึงพื้นที่ SFR

ภายในชิพ EEPROM หน่วยความจำข้อมูลจะถูกเซตค่าที่บิต EEMEN ในรีจิสเตอร์ WMCON ที่ SFR แอดเดรสที่ 96H EEPROM จะมีแอดเดรสในช่วง 000H ไปถึง 77FH ชุดคำสั่ง MOVX จะใช้เข้าถึง EEPROM การเข้าถึงหน่วยความจำข้อมูลภายนอกชิพ ด้วยชุดคำสั่ง MOVX บิต EEMEN ต้องเซตค่าเป็น “0”

บิต EEMWE ใน WMCON ถ้าเซตให้เป็น “1” จะทำให้ EEPROM สามารถถูกเขียนได้ การใช้ซอฟต์แวร์ต้องรีเซต EEMWE เป็น “0” ไซเคิลการเขียนข้อมูลลง EEPROM ในการโปรแกรมแบบโหมดอนุกรมจะมีเวลาของตัวเองโดยประมาณ 2.5 ms การเขียนลง EEPROM สามารถดูได้จาก การอ่านบิต RDY/BSR/ = “0” หมายความว่ากำลังทำการโปรแกรม ถ้า RDY/BSY/ = “1” หมายความว่า EEPROM มีไซเคิลการเขียนสมบูรณ์แล้ว

2.8.3 การจัดวางขาและหน้าที่ต่างๆ



รูปที่ 2.20 การจัดวางขา AT89S8252

ไมโครคอนโทรลเลอร์ที่บรรจุอยู่ในวงจรรวมแบบ Dual Inline Package (DIP) ซึ่งแต่ละข้างของ 8051 มีขาอยู่ข้างละ 20 ขารวมทั้งหมด 40 ขานั้นจะใช้ขาต่าง ๆ กันดังนี้ คือ

- Vcc ขา 40 เป็นขาที่ต้องป้อนไฟเลี้ยง +5 โวลต์เข้าไปเพื่อให้วงจรรวมทำงานได้ ระดับโวลเตจของลอจิก 0 และ 1 ของขา 8051 จึงต่อเข้ากับอุปกรณ์ลอจิกแบบ TTL ได้โดยตรง
- Vss ขา 20 เป็นขาที่ต้องต่อกับกราวด์ (Ground) ของแหล่งจ่ายไฟ การต่ออุปกรณ์ทั้งหมดจะต้องมีกราวด์ของอุปกรณ์ต่อเข้าด้วยกัน
- Port 0 เป็นพอร์ตขนานขนาด 8 บิต อยู่ที่ขา 39 ถึง 32 เริ่มจากบิต 0 ถึงบิต 7 ตามลำดับดัง

ในรูปที่ 1.6 แต่ละขาจะเขียนว่า P0.0, P0.1, ... , P0.7 หมายถึงบิต 7 ของพอร์ต 0 ซึ่งเป็นบิตที่มีนัยสำคัญสูงสุด (Most Significant) และ P0.0 คือบิต 0 ของพอร์ต 0 เป็นบิตที่มีนัยสำคัญต่ำสุด (Least Significant) และ พอร์ต 0 นี้ใช้ได้ทั้งการรับ-ส่งตำแหน่งและข้อมูลกับหน่วยความจำหรือใช้เป็นพอร์ตรับ-ส่งข้อมูลก็ได้

ได้ ข้อมูลที่ส่งออกทางพอร์ต 0 จะถูก Latch ไว้ที่ขาของพอร์ต โครงสร้างแต่ละบิตของพอร์ต 0 เป็นแบบ Open Drain Bidirectional พอร์ต 0 จะใช้งานหลายอย่างดังนี้

1. ใช้สำหรับส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อกับ ตำแหน่งหน่วยความจำสูงสุดที่จะติดต่อก็ได้ก็คือ 64 kbyte จึงมีค่าตำแหน่งหน่วยความจำ 16 บิตของเลขฐาน 2 ค่าตำแหน่งหน่วยความจำ 8 บิตล่าง จะถูกส่งออกไปทางพอร์ต 0 และ 8 บิตบน จะส่งออกไปทางพอร์ต 2

2. ใช้รับ-ส่งข้อมูลกับ Data Memory หรือใช้รับข้อมูลจาก Program memory

3. ใช้รับ-ส่งข้อมูลผ่านทางพอร์ตโดยตรง ในกรณีที่ไม่มีการใช้หน่วยความจำของ Program Memory หรือ Data Memory ภายนอก

- Port 1 เป็นพอร์ตขนานขนาด 8 บิต ที่ขา P1.0 ถึง P1.7 (ขา 1- 8) P1.0 หมายถึง บิต 0 ของพอร์ต 1 ซึ่งเป็นบิต Least Significant Bit และบิต P1.7 หมายถึงบิตที่ 7 ของพอร์ต 1 ซึ่งเป็นบิต Most significant bit

P1.0 T2(เป็นอินพุตรับการนับจากภายนอกไปยัง Timer/Counter2),เอาท์พุทของสัญญาณนาฬิกา

P1.1 T2EX(ทำหน้าที่จับ/กระตุ้นการ โหลดซ้ำและควบคุมทิศทางให้ Timer/Counter2)

P1.4 SS/(Slave port ในการเลือกอินพุท)

P1.5 MOSI(เอาท์พุทของข้อมูลตัวหลัก(Master) รับมาจากอินพุทของข้อมูลตัวรอง (Slave) สำหรับช่อง SPI)

P1.6 MISO(อินพุทของข้อมูลหลัก ไปยังเอาท์พุทที่ข้อมูลตัวรอง)

P1.7 SCK(สัญญาณนาฬิกาของเอาท์พุทที่ตัวหลัก โดยอินพุทมาจากขาสัญญาณ Clock ตัวรอง

P1.4 ถึง P1.7 ใช้สำหรับ SPI channel

- Port 2 เป็นพอร์ตขนานขนาด 8 บิต คือขา P2.0 ถึง P2.7 (บิต 0 ถึงบิต 7 ของพอร์ต 2) ลักษณะโครงสร้างจะเหมือนกับ Port 0 แตกต่างกันใน Port 2 นั้นภาค Driver จะใช้งานเพียง 2 ลักษณะคือ

1. ใช้ส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อกับ ค่าตำแหน่งนี้เป็น 8 บิตบนของค่าตำแหน่ง

2. ใช้เป็นพอร์ตรับและส่งข้อมูลกับภายนอก

- Port 3 คือขา P3.0 ถึง P3.7 หรือขา 10-17 ตามลำดับแต่ละบิตของพอร์ต 3 จะมีฟังก์ชันอื่นดังนี้

P3.0/RXD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม

P3.1/TXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม

P3.1/ $\overline{\text{INT0}}$ (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

P3.3/ $\overline{\text{INT1}}$ (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

P3.4/TO (Timer/Counter 0 External Input) ขารับสัญญาณเข้าไปยังวงจร Timer/Counter 0 ที่ทำหน้าที่นับจำนวนไซเคิลของสัญญาณ T0 นี้หรือสัญญาณนาฬิกาก็ได้

P3.5/T1 (Timer/Counter 1 External Input) ขารับสัญญาณเข้าไปยัง Timer/Counter 1 ซึ่งมีกรเอกสารนี้ทำงานเหมือนกับ T0 สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P3.6/ \overline{WR} (External Data Memory Write Strobe) ขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 8051

P3.7/ \overline{RD} (External Data Memory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก

- RST ขารีเซ็ตขานี้จะใช้ทำการรีเซ็ตการทำงาน ที่ขา RST ภายใน จะมีตัวต้านทานต่อระหว่างขาเข้ากับกราวด์(Ground) ถ้าป้อนสัญญาณที่มีสถานะลอจิก 1 เข้าไปที่ขานี้จะเป็นการรีเซ็ตการทำงาน ดังนั้นจึงสามารถต่อตัวเก็บประจุ (Capacitor) ภายนอกระหว่างขา RST กับไฟเลี้ยง +5 โวลต์ เพื่อให้เกิดการรีเซ็ต เมื่อเริ่มป้อนไฟเลี้ยง ซึ่งเรียกว่า Power on reset การรีเซ็ตจะทำให้ค่าในรีจิสเตอร์ต่างๆ เปลี่ยนไปเป็นค่าหนึ่งดังในตารางที่ 2.3

เมื่อสัญญาณที่ขา RST กลับเป็น 0 ก็จะออกจากการรีเซ็ต จะเริ่มทำงานจากคำสั่งที่อยู่ใน Program memory ตำแหน่ง 0000H เพราะค่าของรีจิสเตอร์ PC (Program Counter) ซึ่งใช้ชี้ตำแหน่งโปรแกรมที่จะทำงาน ถูกเปลี่ยนให้เป็น 0000H ดังนั้นผู้จะต้องเขียนโปรแกรมมาเก็บไว้ที่ตำแหน่ง 0000H ในเครื่องไมโครคอมพิวเตอร์ แบบบอร์ดเดี่ยว (Single Board Microcomputer) จะมีโปรแกรมที่เขียนเก็บไว้เริ่มจากตำแหน่ง 0000H นี้เรียกว่ามอนิเตอร์โปรแกรม (Monitor Program) ที่จะคอยรับการกดแป้นพิมพ์ (Keyboard) และแสดงผลทางตัวแสดงผล (Display) แบบ 7 Segment

- ALE (Address Latch enable) ขานี้จะส่งสัญญาณที่มีความถี่ 1/6 เท่าของสัญญาณนาฬิกาจากออสซิลเลเตอร์ สัญญาณนี้จะส่งออกมาตลอดเวลา ยกเว้นบางครั้งของการติดต่อกับหน่วยความจำสำหรับข้อมูลภายนอก 8051 สัญญาณนี้จะใช้บอกกับอุปกรณ์ภายนอก 8051 ว่าขณะนี้สัญญาณนี้ Active (เป็นลอจิก 1) จะมีการส่งข้อมูลที่เป็น 8 บิตล่างของตำแหน่งหน่วยความจำภายนอก 8051 ที่ต้องการติดต่อออกไปทางพอร์ท 0 อุปกรณ์ภายนอกจะใช้สัญญาณนี้ในการ Latch ข้อมูลไว้เพราะพอร์ท 0 จะส่งค่าตำแหน่งหน่วยความจำออกมาเพียงชั่วขณะเท่านั้น ซึ่งในเวลาต่อมาพอร์ท 0 จะใช้รับ-ส่งข้อมูลกับหน่วยความจำภายนอก สัญญาณ ALE จะสามารถต่อเข้ากับอุปกรณ์ TTL ชนิด LS ได้ถึง 8 อินพุท

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

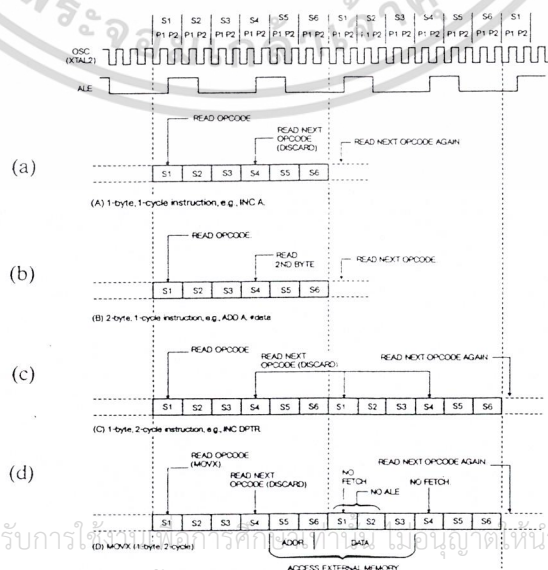
- EA (External Access) ขา 31 ขานี้เป็นขาอินพุทที่ต่อเข้าไปยังวงจร Timing and Control ในรูปที่ 1.5 เพื่อควบคุมการสร้างสัญญาณ PSEN ถ้าป้อนสัญญาณลอจิก 0 เข้าไปที่ขา EA นี้แสดงว่าโปรแกรมในตำแหน่ง 0000H ถึง 0FFFH ที่ต้องการให้ทำงานถูกเก็บไว้ภายนอก จะต้องสร้างสัญญาณออกไปยังภายนอก เพื่อทำการ FETCH คำสั่งเข้ามาทำงาน แต่ถ้าสัญญาณที่ป้อนให้ขา \overline{EA} เป็น 1 หมายความว่าโปรแกรมในตำแหน่ง 0000H ถึง 0FFFH ถูกเก็บไว้ภายใน 8051 การทำงานในตำแหน่งหน่วยความจำช่วงนี้จะอ่านคำสั่งต่างๆ จาก ROM

เอกสารนี้เป็นเอกสาร XTAL1 ขาที่ 19 ขานี้จะต่อเข้ากับขาของ Inverting Amplifier (วงจรขยายแบบป้อนกลับเฟส) (ไม่ว่าการสัญญาณ) ที่ประกอบเป็นวงจรออสซิลเลเตอร์ และเห็นวงจรภายในของออสซิลเลเตอร์ NAND GATE จะ

ถัดไป วงจร Timing and Control เมื่อถอดรหัสคำสั่งแรกก็จะทราบว่าการทำงานคำสั่งนี้ให้สิ้นสุดจะใช้คำสั่ง เพียง 1 ไบต์ ดังนั้นคำสั่งที่ถูกอ่านมาไบท์ที่ 2 จะไม่ถูกนำมาทำงาน เพียงแต่อ่านเข้ามาแล้วทิ้งไป (Discard) ดังในรูปที่ 2.22 a

คำสั่งประเภท 2 ไบต์ และใช้เวลา 1 ไชเคลของเครื่องได้แก่คำสั่ง ADD A , # data ในหนึ่งไชเคลของเครื่องนี้จะมีการอ่านคำสั่งเข้ามา 2 ไบต์ เหมือนกับคำสั่งประเภท 1 ไบต์ 1 ไชเคลของเครื่อง แตกต่างกันที่ไบท์ที่ 2 จะถูกนำมาใช้ด้วยไม่ได้ถูกทิ้งไปดังรูปที่ 2.22b ตัวอย่างของคำสั่ง ADD A , #33H จะเขียนเป็นภาษาเครื่องได้ 2 ไบท์คือ 24 33 เมื่ออ่านคำสั่งไบท์แรกคือ 24 เข้าไปไว้ที่ Instruction Register แล้ว Timing and Control จะถอดรหัสดพบว่า เป็นคำสั่งบวกเลข ก็จะส่งสัญญาณไปยัง Accumulator ให้เอาข้อมูลไปไว้ที่ TMP1 เมื่อคำสั่งที่ 2 ถูกอ่านเข้ามาที่ Instruction Register แล้ว Timing and Control จะสั่งให้เอาข้อมูลไบท์ที่ 2 ส่งลงไปยัง Internal Data Bus ไปเก็บยัง TMP1 จากนั้นวงจร ALU จะนำเอาข้อมูล TMP1 และ TMP2 มาบวกกัน ผลลัพธ์ที่ได้จะส่งออกจาก ALU ไปยัง Internal Data Bus แล้วไปเก็บไว้ที่ Accumulator

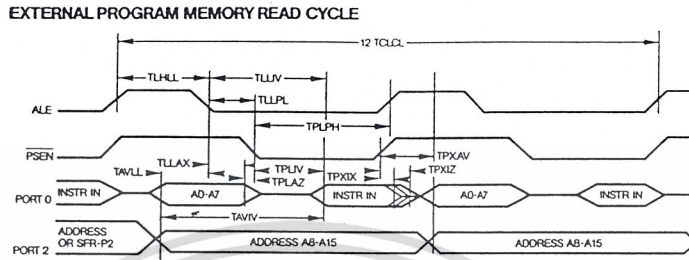
คำสั่งประเภท 1 , 2 หรือ 3 ไบท์ ที่ใช้เวลาทำงาน 2 ไชเคลของเครื่องเช่นคำสั่ง INC DPTR จะมีการอ่านคำสั่งเข้าไป 4 ครั้งทุกๆ ขอบขาขึ้นของสัญญาณ ALE ที่มี 2 ครั้งต่อ 1 ไชเคลของเครื่อง ถ้าเป็นคำสั่งประเภท 1 , 2 หรือ 3 ไบท์ วงจร Timing and Control จะเอาคำสั่ง 1 , 2 หรือ 3 ไบท์แรกเท่านั้นไปทำงานส่วนคำสั่งที่เหลือทิ้งไปดังในรูปที่ 2.22c คำสั่ง 1 ไบท์ที่ใช้เวลาทำงาน 2 ไชเคลของเครื่องที่กล่าวมาแล้วจะไม่รวมถึงคำสั่ง MOVX ซึ่งใช้ในการอ่านหรือเขียนข้อมูลกับหน่วยความจำ Data Memory ภายนอก การทำงานของคำสั่งนี้จะมีการ Fetch คำสั่งเข้าไป 2 ไบท์ในไชเคลของเครื่องแรก ในไชเคลของเครื่องที่ 2 จะไม่มีการ Fetch คำสั่งเข้าไป แต่ละช่วงเวลาของการอ่านหรือเขียนข้อมูลกับ Data Memory ภายนอกสัญญาณ ALE ซึ่งปกติจะเปลี่ยนเป็น 1 ที่ SIP2 ก็จะไม่เปลี่ยนเป็น 1 ในไชเคลของเครื่องที่ 2 โดยจะเป็น 0 อยู่จนกว่าจะถึงเวลา S4P2 ของไชเคลของเครื่องที่ 2 สัญญาณ ALE จะเปลี่ยนเป็น 1 เพื่อทำการอ่านหรือเขียนข้อมูลกับ Data Memory ภายนอก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในมหาวิทยาลัยเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้รูปที่ 2.22 ผังการทำงานแต่ละคำสั่ง

2.8.5 ไดอะแกรมเวลาของการติดต่อกับหน่วยความจำ

การอ่านข้อมูลจากหน่วยความจำสำหรับโปรแกรมภายนอก ลำดับสัญญาณตามเวลา (Timing Diagram) ของสัญญาณที่ทำการอ่านคำสั่งมีดังรูปที่ 2.23



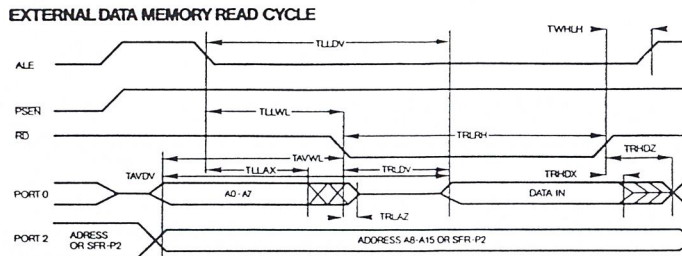
รูปที่ 2.23 ผังเวลาการติดต่อกับหน่วยความจำภายนอก

การอ่านคำสั่ง (Fetch) จาก Program area ภายนอกจะเริ่มจาก AT89S8252 ส่งสัญญาณลอจิก 1 ออกมาทางขา ALE ขณะนี้สัญญาณที่ขา PSEN จะเป็น 1 จากนั้น Port 0 จะส่งค่าตำแหน่งหน่วยความจำ 8 บิตล่างและพอร์ท 2 จะส่งตำแหน่งหน่วยความจำ 8 บิตบนออกมาแล้วสัญญาณ ALE จะกลับเป็น 0 อุปกรณ์ภายนอกจะสามารถใช้ขอบขาของสัญญาณ ALE เพื่อ Latch ตำแหน่งหน่วยความจำที่พอร์ท 0 ไว้ จากนั้นพอร์ท 0 ก็จะยกเลิกการส่งค่าตำแหน่งหน่วยความจำเข้าสู่สภาวะ High Impedance และสัญญาณ PSEN จะเป็น 0 เพื่อเตรียมรับคำสั่งที่ส่งออกจากหน่วยความจำภายนอกเข้าไปยัง 8051 เพื่อทำงานต่อไป เมื่อคำสั่งถูกอ่านเข้าไปเก็บใน Instruction Register (ดูรูปที่ 2.19) แล้วสัญญาณ PSEN จะกลับเป็น 1 พร้อมกับสัญญาณ ALE ก็จะกลับเป็น High เพื่อการอ่านคำสั่งต่อไปทำงาน ข้อมูลในพอร์ท 2 จะคงที่ตลอดเวลา ตั้งแต่สัญญาณ ALE เป็น 1 จนกระทั่งสัญญาณ ALE เปลี่ยนเป็น 0 และกลับเป็น 1 อีกครั้งหนึ่ง จากนั้นจะเริ่มลำดับการ Fetch ข้อมูลไปที่ 2 จากหน่วยความจำสำหรับโปรแกรม ซึ่งจะมีการเปลี่ยนแปลงของสัญญาณตามเวลาเหมือนกับการ Fetch ไบท์แรกนั่นเอง จาก Timing Diagram ดังกล่าวจะออกแบบวงจรที่มี Program Memory อยู่ภายนอก

การอ่าน-เขียนข้อมูลกับ Data Memory ภายใน 8051 นั้นจะมีสัญญาณสร้างมาจากส่วน Timing and Control โดยที่ผู้ใช้ไม่จำเป็นต้องทำความเข้าใจ แต่การอ่าน-เขียนข้อมูลกับ Data Memory ภายนอก อันเนื่องมาจากคำสั่ง MOVX นั้นเมื่อคำสั่งดังกล่าวถูกอ่านเข้ามายัง Instruction Register แล้ว Timing and Control จะทำการถอดรหัสแล้วสร้างสัญญาณควบคุมดังนี้

การอ่านข้อมูลจาก External Data Memory จะมีไดอะแกรมสัญญาณตามเวลาดังรูปที่ 2.24

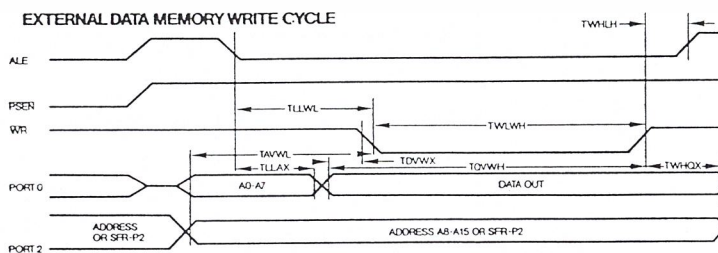
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 ฝั่งเวลาการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก

การทำงานจะเริ่มจากการส่งค่าตำแหน่งหน่วยความจำภายนอก 8 บิตล่างออกทางพอร์ต 0 และ 8 บิตบนออกทางพอร์ต 2 เมื่อส่งค่าตำแหน่งแล้ว สัญญาณ ALE ซึ่งเดิมมีลอจิกเป็น 1 จะกลับเป็น 0 เพื่อให้อุปกรณ์ภายนอกสามารถ Latch ค่าตำแหน่งหน่วยความจำไว้เหมือนกับในการอ่านข้อมูลจากหน่วยความจำสำหรับโปรแกรมภายนอก เพื่อส่งไปยังหน่วยความจำ แม้ว่าข้อมูลบนพอร์ต 0 จะเปลี่ยนแปลงไป ก็จะมีค่าตำแหน่งหน่วยความจำส่งไปยังหน่วยความจำในระหว่างการติดต่อกับ Data Memory นี้ สัญญาณ PSEN จะเป็น 1 ตลอดเพราะสัญญาณจะ Active (เป็น 0) ก็ต่อเมื่อเป็นการติดต่อกับหน่วยความจำสำหรับโปรแกรมภายนอก จะส่งสัญญาณลอจิก 0 ออกมาทางขา RD (P3.7) เพื่อบอกกับหน่วยความจำภายนอกว่าต้องการอ่านข้อมูลเข้าไปเมื่อ 8051 ส่งสัญญาณ RD เป็นลอจิก 0 จะทำให้พอร์ต 0 เข้าสู่สถานะ High Impedance พร้อมทั้งจะให้หน่วยความจำภายนอกส่งข้อมูลมาบนพอร์ต 0 ข้อมูลบนพอร์ต 0 ซึ่งส่งมาจากหน่วยความจำภายนอกจะถูกอ่านเข้าไปเก็บที่เวลาขอบขาขึ้นของสัญญาณ RD จากนั้นสัญญาณ ALE ก็จะกลับเป็น 1 เพื่อเริ่มการทำงานในคำสั่งต่อไปในระหว่างการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอกนี้ พอร์ต 2 จะส่งตำแหน่งหน่วยความจำ 8 บิตบนออกมาตลอดเวลา

การเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 8051 จะมีไคอะแกรมสัญญาณตามเวลาดังรูปที่ 2.25

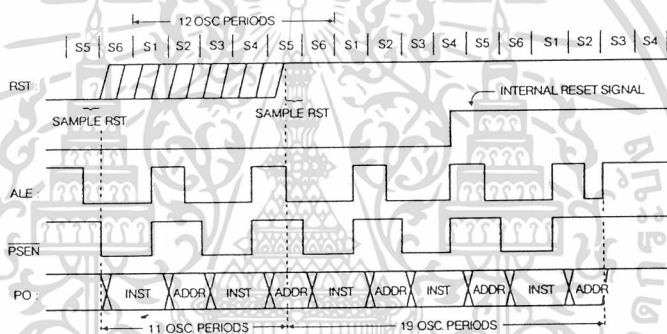


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น รูปที่ 2.25 ฝั่งเวลาการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอกก็มีการนำไปใช้

เมื่อ AT89S8252 ส่งค่าตำแหน่งหน่วยความจำ 8 บิตล่างไปทางพอร์ท 0 และ 8 บิตบนลงไปทางพอร์ท 2 แล้ว สัญญาณ ALE จะกลับเป็น 0 อุปกรณ์ภายนอกจะสามารถใช้สัญญาณนี้ในการ Latch ค่าตำแหน่งหน่วยความจำบนพอร์ท 0 เหมือนกับในการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก เมื่อสัญญาณ ALE เป็น 0 แล้ว AT89S8252 จะส่งข้อมูลที่ต้องการเขียนไปยังพอร์ท 0 แล้วจะให้สัญญาณ \overline{WR} เปลี่ยนสถานะลอจิกเป็น 0 ขณะนี้หน่วยความจำภายนอกจะต้องเขียนข้อมูลไปเก็บยังตำแหน่งที่กำหนด จากนั้นสัญญาณ \overline{WR} จะกลับเป็น 1 เพื่อเป็นการบอกสิ้นสุดการเขียนข้อมูลแล้วสัญญาณ ALE ก็กลับเป็น 1 เพื่อ Fetch คำสั่งต่อไปมาทำงาน

2.8.6 การรีเซ็ต

เมื่อป้อนสัญญาณที่มีสถานะลอจิก 1 เข้าไปทางขา RST จะไม่ได้เกิดการรีเซ็ตขึ้นทันทีทันใด แต่ลำดับการเกิดรีเซ็ตจะแสดงได้ดังไดอะแกรมตามเวลาในรูปที่ 2.26



รูปที่ 2.26 ผังเวลาการรีเซ็ต

ในรูป 2.26 เป็น Timing Diagram ของการรีเซ็ต สถานะลอจิกของสัญญาณที่ขา RST จะถูกอ่านเข้ามาที่เวลา S5P2 (เฟส 2 State 5) ของทุกๆ ไซเคิลของเครื่อง ในกรณีที่เป็นการคำสั่งซึ่งมีการทำงานเสร็จสิ้นใน 2 ไซเคิลของเครื่องก็จะตรวจสอบเฉพาะสัญญาณที่อ่านเข้ามาในไซเคิลที่ 2 ของการทำงาน ดังนั้นในการรีเซ็ตจะต้องป้อนสัญญาณที่มีสถานะลอจิก 1 เข้าไปที่ขา RST เป็นเวลาอย่างน้อย 2 ไซเคิลของเครื่องหรือ 24 ไซเคิลของสัญญาณนาฬิกาที่สร้างจากวงจรรอสซิลเลเตอร์ภายใน 8051 เพื่อให้แน่ใจว่าสัญญาณรีเซ็ตจะถูกอ่านเข้าไปตรวจสอบและทำงาน ขณะที่ทำการรีเซ็ต 8051 ออสซิลเลเตอร์จึงจะต้องทำงานอยู่ด้วย เมื่อ 8051 สุ่มข้อมูลที่ขา RST แล้วตรวจพบว่าเป็นสถานะลอจิก 1 ก็จะสร้างสัญญาณรีเซ็ตขึ้นภายใน ที่เวลา S2P4 ของไซเคิลเครื่องถัดไป ข้อมูลที่แต่ละพอร์ทส่งออกมาจะยังคงปรากฏที่พอร์ทจนกว่าจะเกิดการรีเซ็ตขึ้นซึ่งต้องใช้เวลา 19 ไซเคิลของสัญญาณนาฬิกาจากออสซิลเลเตอร์นับตั้งแต่เวลา S5P2 ในไซเคิลของเครื่องที่พบสัญญาณรีเซ็ต ในระหว่างเวลา 19 ไซเคิลนี้ก็ยังมีการ Fetch คำสั่งเข้าไปทำงานได้อยู่

เอกสารนี้เป็นเอกสารสถานะของสัญญาณลอจิกที่ขา RST จะถูกอ่านเข้าไปตรวจสอบที่เวลา S5P2 ของทุกๆ ไซเคิลของเครื่อง ดังนั้นถึงแม้ว่าสัญญาณที่ขา RST จะมีลอจิกเป็น 1 มาก่อนก็จะยังไม่เกิดการตรวจสอบสัญญาณใช้

รีเซ็ต ดังในรูปที่ 2.26 สัญญาณที่ขา RST อาจเป็น 1 มาตั้งแต่ State ที่ 6 ก็จะไม่เกิดอะไรขึ้นจนกระทั่ง 1 ไชเคล็ดของออสซิลเลเตอร์ต่อมาซึ่งเป็นเวลา S5P2 จึงจะเกิดการตรวจสอบสัญญาณที่ขา RST ถ้าคำสั่งนั้นมีการทำงานมากกว่า 1 ไชเคล็ดของเครื่อง 8051 ก็จะต้องทำงานในคำสั่งนั้นให้เสร็จสิ้นเสียก่อนจึงจะเริ่มการรีเซ็ตได้ โดย 8051 จะดูสถานะของสัญญาณที่ขา RST ของ S5P2 ในไชเคล็ดของเครื่องสุดท้ายเท่านั้น ดังนั้นใน S5P2 ของไชเคล็ดเครื่องแรกๆ ในคำสั่งอาจมีสถานะลอจิกที่ขา RST เป็น 1 แต่ที่ S5P2 ของไชเคล็ดของเครื่องสุดท้าย มีสถานะลอจิกที่ขา RST เป็น 0 ก็จะไม่เกิดการรีเซ็ต

ขึ้นที่เวลา S5P2 เมื่อตรวจสอบสถานะสัญญาณที่ขา RST แล้วพบว่า เป็น 1 จะต้องรอไปจนถึงเวลา S4P2 ของไชเคล็ดของเครื่องถัดไปจึงจะทำให้สัญญาณรีเซ็ตภายในเปลี่ยนสถานะลอจิกจาก 0 เป็น 1 ในระหว่างเวลา S5P2 ที่ตรวจพบสัญญาณ RST มีลอจิกเป็น 1 จนถึง S4P2 ของไชเคล็ดของเครื่องถัดไปจะยังคงมีการ Fetch คำสั่งเข้าไปทำงานอีก 2 คำสั่ง เมื่อสัญญาณรีเซ็ตภายในเปลี่ยนเป็น 1 ก็จะเริ่มการรีเซ็ต โดยการเขียนข้อมูล 0 ไปยัง Special Function Register ทุกตัวยกเว้นพอร์ท 0 ถึงพอร์ท 3 Stack Pointer และรีจิสเตอร์ SBUF ดังตารางในรูปที่ 1.11 ระหว่างนี้ข้อมูลใน RAM ภายใน 8051 จะไม่เปลี่ยนแปลงข้อมูลในระหว่างการเขียนข้อมูลลงไปยัง SFR จะยังมีการ Fetch คำสั่งเข้ามาทำงานอีก 1 คำสั่งจนกว่าจะถึง S3P1 ของไชเคล็ดของที่ 2 (นับแต่ไชเคล็ดของเครื่องที่ตรวจพบลอจิก 1 ที่ขา RST) ก็จะทำให้สถานะลอจิกที่ขา ALE และ $\overline{\text{PSEN}}$ ค้างอยู่ที่สถานะลอจิก 1 และจะเป็นอย่างนี้ไปจนกว่าสถานะลอจิกที่ขา RST เป็น 0 เวลานั้นนับตั้งแต่พบสัญญาณลอจิก 1 ที่ขา RST ที่เวลา S5P2 จนถึงเวลาที่ ALE และ $\overline{\text{PSEN}}$ ค้างอยู่ที่ 1 จะเท่ากับ 19 ไชเคล็ดของออสซิลเลเตอร์ เมื่อสัญญาณที่ขา RST ถูกเปลี่ยนกลับเป็นลอจิก 0 8051 จะรออีก 1 ถึง 2 ไชเคล็ดของเครื่องสัญญาณ ALE และ $\overline{\text{PSEN}}$ จะเริ่มเปลี่ยนแปลงเพื่อเริ่มกระบวนการ Fetch คำสั่งเข้าไปทำงานเริ่มจากคำสั่งในหน่วยความจำสำหรับโปรแกรมตำแหน่ง 0000H

2.8.7 การโปรแกรมวอร์ทด็อกไทม์เมอร์ (Watchdog Timer)

Watchdog Timer จะ Disable โดย Power on Reset และระหว่างที่คำสั่งไฟฟ้าตก มันจะ enable โดยเซทบิต WDTEN ใน SFR WMCON(96H) WDT จะรีเซทโดยตั้งค่าบิตWDTRSTใน WMCON เมื่อเวลาของ WDT เลยเวลากำหนดจะไม่ถูกรีเซท แต่ถ้าภายในช่วง RST Pulse จะรีเซทไปยัง CPU

PS2	SP1	PS0	Period (ms)
0	0	0	16
0	0	1	32
0	1	0	64
0	1	1	128
1	0	0	256
1	0	1	512
1	1	0	1024

2.8.8 การเชื่อมต่ออุปกรณ์แบบอนุกรม

เป็นลักษณะการโอนย้ายข้อมูลแบบซิงโครนัสความเร็วสูง ระหว่าง AT89S8252 และ อุปกรณ์เพอร์เฟอรัล โดย SPI ใน AT89S8252 มีคุณลักษณะดังนี้

- พูลคูลเฟล็กซ์ , สาย 3 เส้น ในการโอนย้ายข้อมูลซิงโครนัส
- การทำงานแบบ Master หรือ Slave
- 1.5 MHz ความถี่บิตสูงสุด
- LSB ก่อนหรือ MSB ก่อน ในการโอนย้ายข้อมูล
- การโปรแกรมมี 4 bite rate
- สิ้นสุดการส่งด้วย interrupt flag
- ป้องกันการเขียนซ้ำด้วย Write Collision Flag

การเชื่อมต่อระหว่าง Master และ Slave CPUS

ขา SCK จะเป็น output ในโหมด Master และจะเป็น Clock input ในโหมด Slave การเขียนลง SPI data register ของ Master เริ่มต้นด้วย SPI ผิด Clock และ ข้อมูลที่เขียนจะเลื่อนไปยังขา MOSI ของ CPU หลังจากเลื่อนไป 1 byte SPI Clock จะเซตเป็นสิ้นสุดการส่งด้วย SPIF ถ้า SPIE และ ES = "1" จะต้องการการ interrupt

SS/ (Slave Select Input) จะถูกเซตเป็น "0" ไปเลือก SPI device เป็น Slave แต่เมื่อให้ SS/ ="1" SPI Port จะไม่กระทำใดๆ และขา MOSI สามารถใช้เป็น input ได้

มีการรวม SCK เฟสและขั้ว (phase and polarity) 4 แบบใช้ตรงกับข้อมูลอนุกรม ซึ่งถูกควบคุมด้วยบิต CPHA และ CPOL

Program Memory Lock Bit

มีการล็อกบิต 3 แบบ สามารถใช้กับแบบไม่โปรแกรม (U) หรือใช้โปรแกรม (P) โดยต้องเป็นไปตามตาราง

เมื่อล็อกบิต 1 (LB1) เป็นโปรแกรมระดับลอจิกที่ขา EA/ จะเป็นการแลตซ์ระหว่างรีเซท ถ้าอุปกรณ์ไม่รีเซท การแลตซ์จะกำหนดให้เป็นค่าสุ่ม และถือว่าค่านั้นจนกระทั่งจะรีเซท ค่ารีแลตซ์ของ EA/ จะต้องได้รับกระแสที่ขา

	LB1	LB2	LB3	
1	U	U	U	หน่วยความจำภายในจะไม่ถูกล็อก
2	P	U	U	คำสั่ง MOVC ที่ execute จากโปรแกรมภายนอกจะถูก disable จากการ fetching code จากภายใน EA/ จะสุ่มค่าและแลตซ์ในการเซ็ทของการ Program Flash memory เป็น Disable

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3	P	P	U	เหมือน mode2 แต่การขนานหรืออนุกรม จะถูกตรวจสอบให้ disable
	P	P	P	เหมือน mode3 แต่ภายนอกจะถูก execute ทั้งหมดเป็น disable จากโปรแกรม Flash และ EEPROM

ภายในชิพ AT 89S8252 จะมี Flash code และ EEPROM data memory การเขียนโปรแกรมต้องมี แรงดันสูง 12 V สำหรับโหมดการโปรแกรมแบบขนาน และแรงดันต่ำ 5 V สำหรับโหมดแบบอนุกรม สำหรับพื้นที่ของ code memory เท่ากับ 0000H – 1FFFFH และพื้นที่ของ data memory เท่ากับ 2000H – 27FFH สำหรับการโปรแกรมแบบขนาน

ขั้นตอนการโปรแกรมแบบขนาน

1. ป้อนหรือจ่ายกำลังไฟระหว่าง Vcc กับ GND Set RST เป็น “H” การจ่าย 3 MHz – 24 MHz ของสัญญาณนาฬิกา ต้องรอน้อย 10 ms
2. เซท PSEN = “L” ALE เป็น “H”

2.9 คีอเมตริกซ์ แอลซีดีโมดูล (Dot matrix LCD module)

อุปกรณ์ในปัจจุบันนี้ ในส่วนแสดงผลนั้นจะใช้ LCD (Liquid Crytral Display) เสียเป็นส่วนใหญ่ ไม่ว่าจะเป็นเครื่องเล่นวีดีโอ, เครื่องถ่ายภาพเอกสาร, เครื่องมือวัดคุมต่าง ๆ เราพอจะแบ่งออกเป็นพวก ๆ ได้ดังนี้

1. CHARACTER LCD MODULE
2. GRAPHIC LCD MODULE
3. SEGMENT DISPLAY TYPE LCD MODULE

โดยในแต่ละแบบจะมีส่วนประกอบหลัก ๆ แบ่งได้เป็น

1. DOT MATRIX LCD เป็นตัวแสดงผลให้เรามองเห็นในลักษณะการปิดและเปิดตัวเองกับแสง คือ ส่วนที่เป็นตัวกระจกบรรจุผลึก

2. DRIVE เป็นตัวรับสัญญาณจากตัวควบคุมมาขับผลึก LCD อีกทีหนึ่ง โดยมีเบอร์ที่นิยมใช้ใน LCD MODULE เช่น HD44100H, MSM5259

3. CONTROLLER เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาและจัดการควบคุม LCD MODULE ให้ทำงานแสดงผลต่าง ๆ เช่น การลบจอภาพ, การเกิดตัวอักษร เป็นต้น โดยมีเบอร์ IC ที่นิยมใช้กันคือ HD44780 ซึ่งจะใช้ในแบบ CHARACTER SCD MODULE เป็นส่วนใหญ่ เบอร์ IC HD61830 จะใช้ในแบบ GRAPHIC LCD MODULE

HD44780 เป็น IC LSI ตัวหนึ่งใช้ควบคุม LCD โดยแสดงผลในรูปตัวอักษรหรือสัญลักษณ์ต่าง ๆ ตัวของ IC สามารถต่อใช้งานแบบ 4 BIT หรือ 8 BIT ก็ได้ โดยถ้าเราต่อแบบ 4 BIT จะต่อใช้งานที่ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้นำไปใช้ประโยชน์ด้านการค้า DB7-DB4

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.1 รายละเอียดคำสั่ง HD44780

1. CLEAR DISPLAY

CODE

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

คำสั่งจะเป็นการเขียนช่องว่างหรือ SPACE (ASCII 20H) เข้าไปใน DD RAM ทั้งหมดและทำการ SET DD REM ADDRESSER เป็นศูนย์ ตัว CURSOR จะกลับไปตำแหน่งบนสุดซ้ายมือของจอภาพ SET I/D = 1,S ไม่มีการเปลี่ยนแปลง

2. RETURN HOME

CODE

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	*

* NO EFFECT

คำสั่งนี้จะทำการ SET DD RAM ADDRESSER เป็นศูนย์ ตัว CURSOR จะกลับไปอยู่ตำแหน่งบนสุดซ้ายมือของจอภาพข้อมูลในจอภาพไม่เปลี่ยน

3. ENTRY MODE SET

CODE

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	I/D	S

BIT I/D : โดยจะเป็นตัวกำหนดให้ว่าเมื่อเขียนหรืออ่านข้อมูลแล้วจะทำให้ DD RAM ADDRESS เพิ่มขึ้นหนึ่งหรือลดลงหนึ่งโดย 1 = เพิ่ม 0 = ลดลงหนึ่ง

BIT S : เป็นตัวกำหนดแสดงผลโดยถ้า S = 1 จะเป็นการใส่ข้อมูลแล้วตัว CURSOR อยู่ที่บิต ข้อมูลจะถูกดันไปทางซ้าย ถ้า S = 0 ข้อมูลจะอยู่กับที่ ตัว CURSOR จะถูกดันไปทางขวามือ

4. DISPLAY ON /OFF CONTROL

CODE

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	0	1	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BIT D : เป็น BIT ให้เปิดปิดหน้าจอภาพโดยถ้า D = 1 จะ ON และ D = 0 จะ OFF

BIT C : จะให้แสดง CURSOR ให้ BIT C = 1 และถ้าไม่ต้องการแสดง CURSOR BIT C = 0 โดยตัว CURSOR จะอยู่ที่ LINE ที่ 8 ในแบบ 5*7 DOT และจะอยู่ LINE ที่ 11 ในแบบ 5*7 DOT

5. CURSOR OR DISPLAY SHIFT

CODE

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	*	*

* NO EFFECT

เป็นคำสั่งกำหนดให้ตำแหน่ง CURSOR หรือข้อมูลไปเกิดทางซ้ายหรือขวาโดยไม่ต้องใช้คำสั่งเขียนหรืออ่านโดยที่ ตาราง 2.10 แสดงหน้าที่ขา S/C และ R/L และตารางที่ 2.11 แสดงการกำหนดค่าตัวอักษร

S/C	R/L	FUNCTION
0	0	ทำการย้าย CURSOR ไปจากตำแหน่งเดิมไปซ้ายมือ 1 ตำแหน่ง
0	1	ทำการย้าย CURSOR ไปจากตำแหน่งเดิมไปขวามือ 1 ตำแหน่ง
1	0	เป็นการค้นตัวอักษรที่เกิดไปทางซ้าย
1	1	เป็นการค้นตัวอักษรที่เกิดไปทางขวา

ตารางที่ 2.7 แสดงหน้าที่ขา S/C และ R/L

6. FUNCTION SET

CODE

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	*	*

* NO EFFECT

BIT DL : เป็นการ SET การติดต่อว่าจะให้เป็นแบบ 8 BIT หรือ 4 BIT โดยถ้าต้องการ ติดต่อ 4

BIT DL = 0 และ 8 BIT DL = 1

BIT N : เป็นการ SET บรรทัดการแสดงผล N = 0 แสดง 1 บรรทัด N = 1 แสดง 2 บรรทัด
ไม่ว่ากรณีใดก็ตามก็ให้ SET N = 1 หากและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BIT F : เป็นการ SET ขนาด DOT การแสดงผล 5*7 หรือ 5*10 โดย F = 0 เป็นแบบ 5*7 และ F = 1 เป็นแบบ 5*10

N	S	NO. OF DISPLAY LINE	CHARACTER FONT	DUTY FACTOR	REMARK
0	0	1	5*7 DOT	1/8	
0	1	1	5*7 DOT	1/11	
1	*	2	5*7 DOT	1/16	CANNOT DISPLAY 2 YEAR 2 LINES WITH 5*10 DOT CHARACTER FONT

ตารางที่ 2.8 แสดงกำหนดค่าตัวอักษร

7. SET CG RAM ADDRESS

CODE

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	A	A	A	A	A	A

ใน HD44780 นั้นจะมีหน่วยความจำอยู่ 2 ชุด คือ DISPLAY DATA RAM จำนวน 80*8 BIT และ CHARACTER GENERATOR RAM CG RAM คำสั่งนี้จะเป็นการ SET ADDRESS ใน CG RAM โดยต้องทำการ SET ADDRESS ก่อนเขียนหรืออ่านข้อมูลจาก CG RAM ด้วย

8. SET DD RAM ADDRESS

CODE

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	A	A	A	A	A	A	A

เป็นคำสั่ง SET ค่า ADDRESS ใน DD RAM ในการเขียนหรืออ่านค่าจาก DD RAM (DD RAM คือ ส่วนที่จะแสดงผล LCD หน้าจอ) โดยจำนวน ADDRESS ที่เกิดขึ้นบนจอ LCD จะอยู่กับ SET ค่า N ด้วย

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้า N = 0 (2 บรทัด) ADDRESS จะอยู่ OOH-27FH สำหรับบรทัดที่ 1 40H-67H สำหรับบรทัดที่ 2

ตัวอย่างการจัด ADDRESS ของ DD RAM หน้าจอ LCD แบบ 16 ตัวอักษร 4 บรทัด HDM16416H แสดงในตารางที่ 2.9

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF

HDM-16416H

ตารางที่ 2.9 ตำแหน่ง ADDRESS ของจอ

9. READ BUSY FLAG AND ADDRESS

CODE

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	A	A	A	A	A	A	A

เป็นคำสั่งอ่านค่า BUSY FLAG ซึ่งจะเป็นตัวบอกว่า GD44780 นี้อยู่ในขบวนการทำงานภายในอยู่หรืออยู่ในสภาพพร้อมจะรับข้อมูล โดย

BF = 1 อยู่ในขบวนการทำงานภายในไม่พร้อมจะรับข้อมูลหรือคำสั่ง

BF = 0 พร้อมจะรับข้อมูลหรือคำสั่งได้

และนอกจากนี้ยังเป็นคำสั่งอ่านค่าข้อมูล ADDRESS ของ CG RAM หรือ DD RAM ด้วย

10. WRITE DATA TO CG หรือ DD RAM

CODE

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D	D	D	D	D	D	D	D

เป็นคำสั่งเขียนข้อมูลเข้าไปใน CG หรือ DD RAM โดยเมื่อเขียนข้อมูลและ ADDRESS จะเพิ่มหรือลดโดยอัตโนมัติตามคำสั่งที่ SET ใน ENTRY MODE ข้อกำหนดที่จี้จะรู้ว่าเป็นการเขียนข้อมูลของ CG

RAM หรือ DD RAM ทำได้โดยการ SET ADDRESS ของ CG RAM หรือ DD RAM ขึ้นมาก่อนจะเขียนข้อมูล

11. READ DATA FROM CG OR DD RAM

CODE

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	D	D	D	D	D	D	D	D

เป็นคำสั่งอ่านข้อมูลจาก CG RAM หรือ DD RAM โดยก่อนอ่านค่าจาก DD RAM หรือ CG RAM นี้ควรจะใช้คำสั่ง SET ADDRESS ก่อนเพื่อให้รู้ว่าข้อมูลที่อ่านได้นั้นเป็น DD หรือ CG RAM จากตารางการทำงานจะเห็นว่าการใช้งาน LCD MODULE นั้นง่ายเพียงแค่เราส่งคำสั่งเริ่มแรกและ SET ความต้องการในขนาดตัวอักษร, CURSOR หลังจากนั้นเราก็สามารถเขียนตัวอักษรเข้าไปใน DD RAM ตามตารางตัวอักษรที่ให้นั้นก็จะเกิดตัวอักษรในจอภาพ LCD เรายังสามารถกำหนดตำแหน่งตัวอักษรที่จะให้เกิดบนจอได้โดยการ SET DD RAM ADDRESS ตามตารางที่ให้มาในหัวข้อ SET DD RAM ADDRESS จะมีส่วนเริ่มต้นก็คือ ส่วนการ INITIAL LCD เพื่อกำหนดหน้าที่การทำงานต่างๆ

ขาสัญญาณของ LCD MODULE

PIN	SYMBOL	LEVEL	FUNCTION
1	Vss	---	GND
2	Vcc	---	+5 V POWER SUPPLY
3	Vo	---	+ V FOR LIQUID CRYSTAL DREVE
4	Rs	H/L	REGISTER SELECT
5	R/W	H/L	H = READ L = WRITE
6	E	H	DATA BUS BIT 0
7	DB0	H/L	DATA BUS BIT 1
8	DB1	H/L	DATA BUS BIT 2
9	DB2	H/L	DATA BUS BIT 3
10	DB3	H/L	DATA BUS BIT 4
11	DB4	H/L	DATA BUS BIT 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามแก้ไขเปลี่ยนแปลงเนื้อหาใดๆ โดยไม่ต้องแจ้งให้ทราบล่วงหน้า

12	DB5	H/L	DATA BUS BIT 6
13	DB6	H/L	DATA BUS BIT 7
14	DB7	H/L	DATA BUS BIT 8

ตารางที่ 2.10 แสดงขาของ LCD MODULE

1. RS (REGISTER SELECTION) จะเป็นขาเลือก REGISTER ภายในซึ่งมีอยู่ 2 ตัวคือ INSTRUCTION REGISTER (IR) และ DATA REGISTER (DR) โดยถ้าเป็น 1 จะเป็นการเลือก DATA และถ้าเป็น 0 จะเป็นการเลือก INSTRUCTION

2. R/W (READ/WRITE) เป็นตัวเลือกว่าจะเขียนหรืออ่านข้อมูลจากตัว IC โดยอ่านข้อมูล = 1 เขียนข้อมูล = 0

3. E (ENABLE) เป็นขาคำหนดสภาพการรับเขียนอ่านข้อมูล

4. DBO-DB7 เป็นขารับส่งข้อมูลจากตัว IC

5. V_{dd} ไฟเลี้ยงตัววงจร

6. V_{ss} เป็นขา GROUND

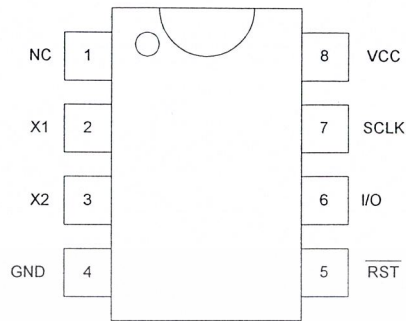
7. V_o เป็นขารับ VOLTAGE ในการขับ LCD ให้มีคหรือสว่าง

2.10 DS1202 ชิพสัญญาณนาฬิกาแบบอนุกรม

DS1202 เป็นชิพไอซีของบริษัท DALLAS SEMICONDUCTOR ที่ให้เวลาเป็นจริง Real Time Clock) กับระบบไมโครโปรเซสเซอร์ โดยระบบไมโครโปรเซสเซอร์ สามารถเซตค่าเวลาหรืออ่านค่าเวลาออกมาได้เป็นวินาที นาที ชั่วโมง วันและปี และมีการปรับวันสุดท้ายของเดือนรวมทั้งปรับเดือนกุมภาพันธ์ที่มี 29 วันวัน (leap year) ให้โดยอัตโนมัติ แล้วยังมีหน่วยความจำแรมให้ใช้งานอีก 24 ไบต์

DS1202 เชื่อมต่อกับระบบไมโครโปรเซสเซอร์หรือต่อโดยตรงกับไอซีทีทีแอล โดยการรับส่งข้อมูลแบบอนุกรม จึงใช้สายเพียง 3 เส้นคือ RESET, SCLK, และ I/O ให้การอินเตอร์เฟสทำได้โดยง่ายและระบบ RTC มีขนาดเล็กลงมาก DS1201 ได้ถูกออกแบบมาให้ทำงานที่กัลดังต่างๆ โดยจะเก็บรักษาข้อมูลนาฬิกา และข้อมูลในแรมได้ที่ไฟเลี้ยง 3 โวลต์และกินกระแสเพียง 1 ไมโครแอมป์เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.27 แสดงการจัดขาของ DS1202

หน้าที่และการใช้งานของแต่ละขาคือ

X1,X2 ต่อกับคริสตอลความถี่ 32.768 กิโลเฮิร์ตซ์ เพื่อเป็นฐานเวลาให้กับวงจรมติกาภายในตัว DS1202

RST เป็นขารีเซ็ตเพื่อทำให้วงจรมติกาบางส่วนเริ่มต้นทำงานใหม่

I/O DATA in/out สำหรับการเขียนและการอ่านข้อมูลกับระบบไมโครโปรเซสเซอร์

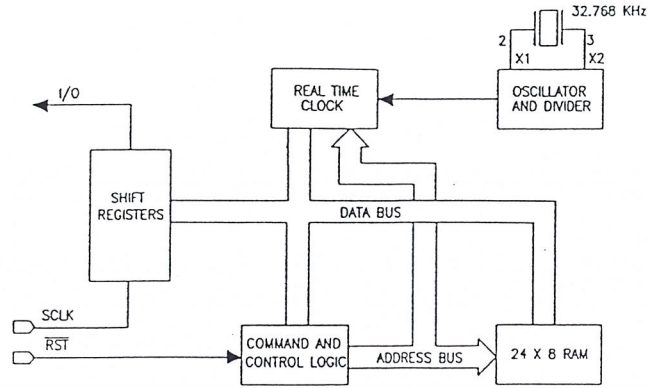
SCLK Serial Clock กำหนดการเคลื่อนย้ายข้อมูลแต่ละบิตในการรับส่งข้อมูล

Vcc ไฟเลี้ยง 2.0-5.5 โวลต์โดยกินกระแสต่ำกว่า 300 นาโนแอมป์ที่ 2 โวลต์

2.10.1 การทำงานของ DS1202

รูปที่ 2.28 แสดงส่วนต่างภายใน DS1202 ที่ประกอบด้วยวงจรมติกา, วงจรมติกาความถี่, วงจรมติกาหน่วยความจำแรม และส่วนควบคุมต่างๆ การรับการส่งจะทำเมื่อสัญญาณรีเซ็ตเป็น “1” เท่านั้นและข้อมูลไบต์แรก จะกำหนดลักษณะคำสั่งและแอดเดรสที่ต้องการติดต่อด้วยการรับส่งข้อมูลแต่ละบิตจะถูกควบคุมจังหวะด้วยสัญญาณ SCLK (Serial Clock) โดย DS1202 จะรับข้อมูลอินพุตจาก INT ที่ขอบขาขึ้น (Rising edge) ของสัญญาณ SCLK จะให้ข้อมูลที่ถูกต้องออกจากขา I/O ที่ขอบขาลง (falling edge) ของสัญญาณ SCLK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.28 บล็อกโคอะแกรมภายใน DS1202

การติดต่อกับ DS1202 จะต้องใช้ข้อมูลอย่างน้อย 2 ไบต์แรกถือว่าเป็นคำสั่งและ ไบต์ที่สองคือข้อมูลที่จะรับเข้าไปหรือส่งออกมา จากรูปที่ 3 แสดงถึงรูปแบบของคำสั่ง โดยบิตที่ 0 จะบอกว่าเป็นการเขียนหรืออ่านข้อมูล, บิตที่ 1-5 จะกำหนดแอดเดรสของนาฬิกาหรือแรมที่จะติดต่อกับ, บิตที่ 6 จะติดต่อกับนาฬิกาหรือแรม และบิตที่ 7 เป็น "1" แสดงว่าไบต์นี้คือ คำสั่ง

เมื่อคำสั่งขนาด 1 ไบต์ถูกโหลดเข้าไปใน DS1202 แล้วจึงตามด้วยข้อมูลอีก 1 ไบต์ในกรณีการเขียนข้อมูลหรือรับข้อมูล 1 ไบต์ออกมาในการอ่าน โดยทั้งคำสั่งและข้อมูลแต่ละบิต จะถูกควบคุมจังหวะด้วยสัญญาณ SCLK รูปคลื่นของสัญญาณต่างๆ เมื่อเริ่มให้ DS1202งานจะต้องกำหนดให้ขา RESET และ SCLK เป็น "0" เสียก่อนเพื่อให้วงจรบางส่วนเริ่มต้นทำงาน (initial) และให้วงจรฟลิปฟล็อปรับรู้ว่ามีข้อมูลไบต์แรกที่ได้รับเข้ามาคือคำสั่ง และเมื่อคำสั่งรีเซตเป็น "0" แล้วจะทำให้ขา I/O เป็น high impedance อีกด้วยเมื่อสิ้นสุดการรับส่งข้อมูลแล้วจะต้องทำให้สัญญาณรีเซตเป็น "0" อีกครั้งในขณะที่สัญญาณ SCLK เป็น "1" อยู่

7	6	5	4	3	2	1	0
1	RAM CK	A4	A3	A2	A1	A0	RD W

รูปที่ 2.29 แสดงรูปแบบของคำสั่ง

2.10.2 การรับส่งข้อมูลกับ DS1202

การรับส่งข้อมูลจากระบบไมโครโปรเซสเซอร์กับ DS1202 จะกระทำโดยผ่านทางพอร์ตและจะเริ่มต้นรับส่งด้วยบิตที่ 0 จบที่บิต 7 โดยสัญญาณรีเซตจะต้องเป็น "1" ตลอดช่วงการรับและส่งออกสารเป็นเอกสารที่ส่งมานี้ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับว่าเห็นหน้าไปเซประโยชน์ด้านการค้าส่งข้อมูลในแต่ละครั้ง (ในแต่ละครั้งรับส่งข้อมูลได้ 2 ไบต์หรือมากกว่า) ไม่ว่ากรณีใดๆ ทั้งสน อีกทั้งห้ามมีเหตุดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขั้นตอนการส่งข้อมูลให้ DS1202

1. กำหนดให้ขา RESET และ SCLK เป็น “0”
2. ให้ขา RESET เป็น “1” เพื่อเริ่มต้นการส่งข้อมูล
3. กำหนดข้อมูลบิตแรกให้กับขา I/O
4. เปลี่ยนระดับสัญญาณ SCLK จาก “0” เป็น “1” ข้อมูลจะถูกโหลดเข้า DS1202
5. ถ้าสิ้นสุดของข้อมูลทั้งหมดแล้วให้ขา RESET เป็น “0” อีกครั้ง
6. ถ้ายังมีข้อมูลที่จะส่งอีก ให้เปลี่ยนระดับสัญญาณ SCLK จาก “1” เป็น “0” แล้วทำตามขั้นตอน

ที่ 3 ใหม่จนกระทั่งหมดข้อมูล

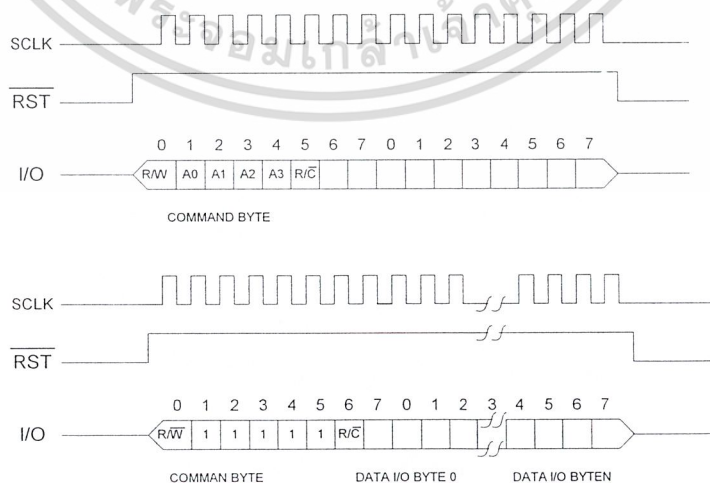
- ขั้นตอนการรับข้อมูลจาก DS1202

1. กำหนดให้ขา RESET และ SCLK เป็น “0”
2. ให้ขา RESET เป็น “1” เพื่อเริ่มต้นการรับข้อมูล
3. ให้ขา SCLK เป็น “1”
4. เปลี่ยนระดับสัญญาณ SCLK จาก “1” เป็น “0”
5. อ่านข้อมูลบิตแรกจากขา I/O
6. เปลี่ยนระดับสัญญาณ SCLK จาก “0” เป็น “1”
7. ถ้าสิ้นสุดของข้อมูลทั้งหมดแล้วให้ขา RESET เป็น “0” อีกครั้ง
8. ถ้ายังมีข้อมูลที่จะรับอีกให้กลับไปทำตามขั้นตอนที่ 4 ใหม่จนกระทั่งหมดข้อมูล

- รูปแบบการรับส่งข้อมูล

การรับส่งข้อมูลนาฬิกาและแรมสามารถทำได้ 2 แบบ คือ ข้อมูลไบต์เดี่ยว (Single Byte) และแบบข้อมูลเป็นชุด (Burst Mode) ความแตกต่างของการรับส่งทั้งสองแบบมีดังนี้

- Single Byte เป็นการรับส่งที่จะต้องมียไบต์คำสั่ง (command byte) นำหน้าทุกครั้งเช่น ถ้าต้องการส่งข้อมูล 8 ไบต์ ก็จะต้องส่งข้อมูลทั้งหมด 16 ไบต์ (ดูรูปที่ 2.30 ประกอบ)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่เนื้อหาของเอกสารทุกครั้งที่มีการนำไปใช้

- Burst Mode เป็นการรับส่งที่กำหนดไปบิตคำสั่ง (Command byte) นำหน้าเพียงบิตเดียว ในการรับส่งข้อมูลแต่ละครั้ง และบิตต่อไปจะเป็นข้อมูลทั้งหมด โดยบิตที่ 1-5 ในไปบิตคำสั่งจะต้องเป็น 1 ทั้งหมด การรับส่งข้อมูลในโหมดนี้ทำให้การติดต่อกับ DS1202 สะดวกมาก จะเห็นว่าถึงแม้ DS1202 จะติดต่อกับลักษณะอนุกรม ก็ไม่ใช่เรื่องยากหรือเกิดความซ้ำในแง่การใช้งาน

2.10.3 การจัดแอดเดรสใน DS1202

จากตารางที่ 2.11 แสดงถึงการจัดตำแหน่งของรีจิสเตอร์ที่เก็บค่าเวลาและหน่วยความจำแรม จะเห็นว่าข้อมูลที่เก็บค่าเวลาจะเป็นแบบ BCD และเก็บเท่าที่จำเป็นเท่านั้น เช่น ที่แอดเดรส 82 บิตที่ 0-3 จะเป็นหลักหน่วยของนาฬิกา และบิตที่ 4-6 จะเป็นหลักสิบของนาฬิกา (เพราะค่านาฬิกาสูงสุดคือ 59) ส่วนบิตที่เหลือก็จะเป็น 0 ไป ส่วนของรีจิสเตอร์นาฬิกาจะอยู่ที่แอดเดรส 80-8D และหน่วยความจำแรมทั้งหมด 24 บิต ที่สามารถใช้เก็บข้อมูลได้อย่างอิสระจะอยู่ที่แอดเดรส ส่วนแอดเดรสที่ BF,BF,FE ,FF จะถูกอ้างอิงเมื่อทำงานใน MODE เท่านั้น การป้องกันการเปลี่ยนแปลงข้อมูลที่แอดเดรส 8E, 8F จะเป็นรีจิสเตอร์ป้องกันการเขียน (Write Protect Register) สำหรับควบคุมการเขียนข้อมูลลงในรีจิสเตอร์นาฬิกาหรือแรม โดยถ้าบิตที่ 7 นี้เป็น “1” จะไม่สามารถเขียนข้อมูลลงในรีจิสเตอร์นาฬิกาหรือแรมได้ต้องเมื่อจ่ายไฟ (Power up) บิตนี้จะถูกเซตให้เป็น “1” อัตโนมัติ ก่อนการเปลี่ยนแปลงข้อมูลใดๆ จึงต้องทำให้บิต 7 เป็น “0” เสียก่อน

REGISTER	FUNCTION	COMMAND ADDRESS (HEX)	WRITE=W READ=R	RANGE DATA (BCD)	REGISTER DEFINITION							
					7	6	5	4	3	2	1	0
0	SECONDS	80	W	00-59	CH	10 SEC			SEC			
		81	R									
1	MINUTES	82	W	00-59	0	10 MIN			MIN			
		83	R									
2	12HRS	84	W	01-12	12\	0	AP	HR	HOUR			
	24HRS	85	R	00-23	24	0	10	HR				
3	DATE	86	W	01-31	0	0	10 DATE			DATE		
		87	R									
4	MONTH	88	W	01-12	0	0	0	10M	MONTH			
		89	R									
5	DAY	8A	W	01-07	0	0	0	0	DAY			
		8B	R									

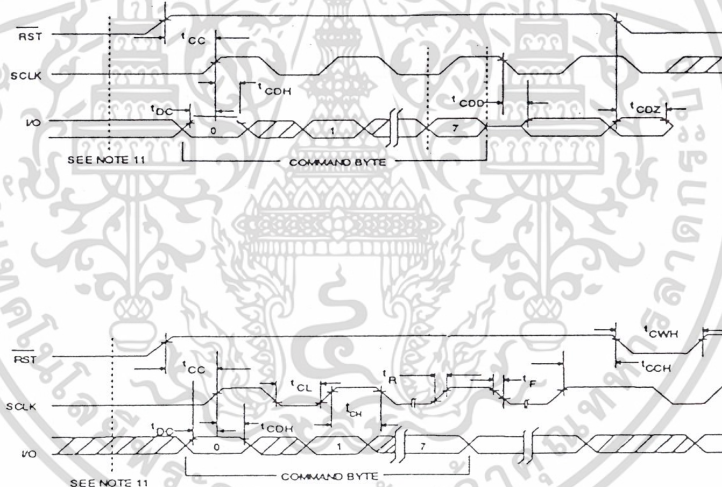
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6	YEAR	8C 8D	W R	00-99	10 YEAR	YEAR
7	WRITE PROTECT	8E 8F	W R	00-80	WP	ALWAYS ZERO

ตารางที่ 2.11 การจัดแอดเดรสใน DS1202

หน่วยความจำขนาด 24 ไบต์ (ไม่เกี่ยวกับหน่วยความจำของระบบไมโครโปรเซสเซอร์) ใช้สำหรับเก็บข้อมูลที่ต้องการถาวร เพราะตรงส่วนนี้ได้รับการแบคอัพเช่นเดียวกับส่วนนาฬิกา การเขียนอ่านแรมก็สามารถทำได้เช่นเดียวกับข้อมูลนาฬิกา รับส่งข้อมูลได้ทั้ง Single byte และ Burst Mode และค่าในรีจิสเตอร์ป้องกันการเขียนทับจะมีผลกับการเขียนข้อมูลในแรมส่วนนี้ด้วย

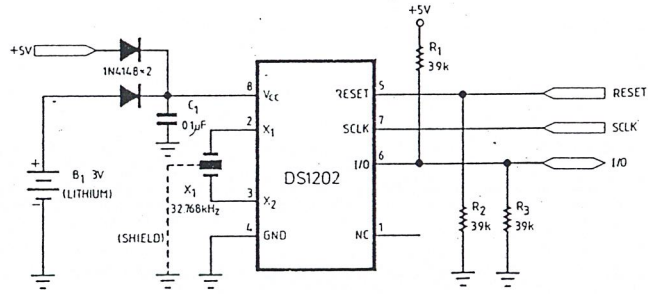


รูปที่ 2.31 แสดงผังเวลาขณะเขียนและอ่านข้อมูล

2.10.4 DS1202 กับการใช้งานจริง

ขา RESEET, SCLK จะเป็นอินพุตอย่างเดียว ส่วนขา I/O เป็นทั้งอินพุตและเอาต์พุตด้วย จากรูปที่ 2.32 ขานี้อาจใช้ตามรูปหรือมาจากวงจรแบคอัพเช่นจากขา Vout ของ MAX691, ขาX1,X2 ต่อกับคริสตอลความถี่ 32.768 กิโลเฮิร์ตซ์ได้โดยตรง จะเห็นว่าไม่มีส่วนปรับความถี่ จึงทำให้วงจรนาฬิกาเดินเที่ยงตรงและถ้าต่อตัวถังของคริสตอลลงกราวด์ก็ทำให้มีเสถียรภาพ (stability) ดีขึ้น

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



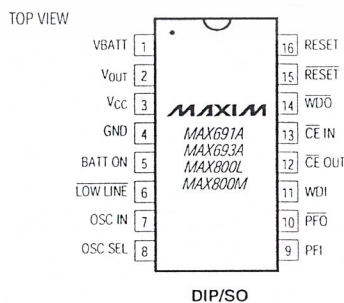
รูปที่ 2.32 วงจรใช้งานจริงของ DS1202

DS1202 สามารถต่อกับพอร์ต 1 ของ CPU ตระกูล MCS-51 ได้โดยตรงถ้าเป็นระบบ Z-80 ก็ต้องผ่านพอร์ต เช่นพอร์ต C ของ 8255 และเนื่องจาก DS1202 ไม่มีขาอินเตอร์รัพต์ (interrupt) ดังนั้นชิพนี้จะต้องทำการอ่านข้อมูลจาก DS1202 เอง

2.11 MAX691 ชิพผู้ช่วยงาน CPU

MAX691 เป็นชิพไอซีในตระกูล MAX690 จากบริษัท MAXIM ทำหน้าที่เป็นผู้ช่วยงานที่จำเป็นหลาย ๆ อย่างในระบบไมโคร ๓ เพื่อลดอุปกรณ์ต่างๆ ลง และทำให้ระบบมีประสิทธิภาพและความเชื่อถือได้ในการทำงาน MAX691 สามารถทำงานได้โดยต่อกับอุปกรณ์ภายนอกเพียงเล็กน้อยเท่านั้น หน้าที่และการใช้งานต่างๆ มีดังนี้

- รีเซต CPU เมื่อเริ่มจ่ายไฟให้กับระบบ (power up) และเมื่อแรงดัน ไฟเลี้ยงตกลงกว่าที่กำหนด
- รีเซตเมื่อ CPU ไม่กระตุ้น (trig) วงจรวอตช์ด็อกในช่วงเวลาที่กำหนด
- ตัดต่อไฟเลี้ยงกับแบตเตอรี่ เพื่อจ่ายให้กับซีมอสแรม, วงจรนาฬิกาหรืออุปกรณ์ลอจิกกำลังต่ำ เพื่อให้เก็บรักษาข้อมูลไว้ได้
- ให้อสัญญาณ low batt เมื่อแรงดันจากแบตเตอรี่ตกลงกว่าที่กำหนด
- ป้องกันการเขียนข้อมูลในแรมหรือ EEPROM ในสภาวะไฟเลี้ยงไม่พร้อม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในองค์กรหรือหน่วยงานที่ซื้อไปจากภาคนี้หน้าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11.1 การจัดวางขาและหน้าที่

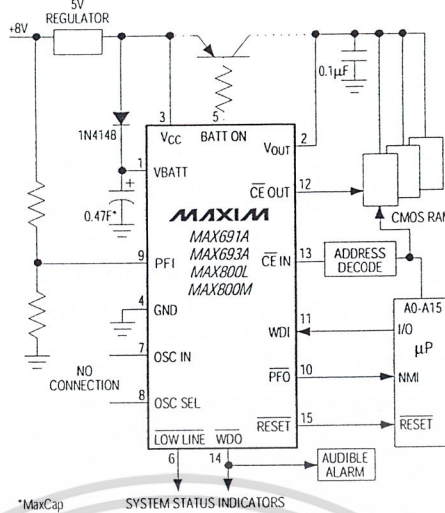
MAX691 เป็นไอซีแบบคิพมีจำนวนขา 16 ขา ด้วยกัน โดยหน้าที่และการใช้งานของแต่ละขา ดังนี้

- V(bat) ต่อกับขั้วบวกของแบตเตอรี่ เมื่อต้องการแบคอัพซิมอสแรมหรือ RTC ฟังก์ชันนี้จะไม่ให้ต่อลงกราวด์
- V_{OUT} ให้ไฟบวกที่สูงกว่าจากไฟเลี้ยงของอุปกรณ์ที่ต้องการแบคอัพถ้าไม่ใช้ให้ต่อขาที่เข้าขั้วบวกของระบบ
- GND กราวด์ของระบบ
- BATT ON เป็น high เมื่อขา V_{OUT} ได้แรงดันจากแบตเตอรี่ ขานี้จ่ายกระแสซิงค์ได้ 25 mA เพื่อขับทรานซิสเตอร์ pnp ภายนอก เพื่อเพิ่มกระแสที่ขา V_{OUT} ให้สูงขึ้น
- LOW LINE เป็น low เมื่อไฟเลี้ยงตกลงต่ำกว่า 4.65 โวลต์
- OSC_{in} เมื่อกำหนดให้ขา OSC_{scr} เป็น low ขานี้จะรับสัญญาณนาฬิกาจากภายนอกเพื่อกำหนดช่วงเวลาในการรีเซ็ตและกำหนดช่วงเวลาสูงสุดของวงจรวอตช์ดีออก
- PFI ขาอินพุตของวงจรเปรียบเทียบแรงดันตก เมื่อแรงดันที่ขาที่ตกลงต่ำกว่า 1.3 โวลต์ ทำให้มีสัญญาณ low ที่ขา PFO เพื่อใช้ในการอินเทอร์รัพต์ CPU ได้
- PFO เป็น low เมื่อแรงดันที่ขา PFI ต่ำกว่า 1.3 โวลต์
- WDI อินพุตสำหรับการกระตุ้นวงจรวอตช์ดีออก โดยการเปลี่ยนระดับของสัญญาณจากเอาต์พุตพอร์ต หรือสัญญาณ Port select ก็ได้แต่จะต้องกระตุ้นภายในช่วงเวลาที่กำหนด
- CE_{out} เป็น low ตามขา CE_{in} และในสถานะที่ไฟเลี้ยงปกติ คือ สูงกว่า 4.65 โวลต์ เท่านั้น
- CE_{in} เป็นสัญญาณ chip enable ของซิมอสแรมหรือ RTC ที่ต้องการแบคอัพ โดยขาที่นี้จะต่อมาจากส่วนที่ถอดรหัสหน่วยความจำ
- WDO เป็น low ถ้า CPU ไม่กระตุ้นวงจรวอตช์ดีออก ภายในเวลาที่กำหนดและในสถานะที่ไฟเลี้ยงต่ำกว่า 4.65 V จะไม่มีการกระตุ้นวงจรวอตช์ดีออกทำให้ขาที่เป็น high ด้วย
- RESET LOW เป็นขาที่ให้สัญญาณรีเซ็ต high เพื่อรีเซ็ต CPU เช่น MCS-51

2.11.2 การใช้งาน MAX691

จากวงจรรูปที่ 2.34 ขาไฟเลี้ยงของซิมอสแรมหรือ RTC จะต่อกับขา V_{out} ของ MAX691 โดยขาที่นี้จะได้แรงดันจากไฟเลี้ยงของระบบ แต่เมื่อไม่มีไฟเลี้ยงหรือไฟเลี้ยงตกลงต่ำกว่า 4.65 โวลต์ ขา V_{out} จะได้รับแรงดันจากแบตเตอรี่ จะเห็นว่าแรงดันที่ขา V_{out} จ่ายกระแสได้ 50 มิลลิแอมป์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.34 แสดงวงจรใช้งานจริงของ MAX691

แต่ถ้าต้องการกระแสมากขึ้น ก็สามารถต่อทรานซิสเตอร์จากภายนอกเข้าไปได้รายละเอียดการใช้งานอื่น ๆ มีดังต่อไปนี้

- สัญญาณรีเซต

วงจรเปรียบเทียบแรงดันจะให้สัญญาณรีเซต เมื่อไฟเลี้ยงตกลงต่ำกว่า 4.65 โวลต์หรือเมื่อ CPU ไม่กระตุ้นวงจรวอตซ์ดีค็อก ภายในเวลาที่กำหนดสัญญาณรีเซตจะมีช่วงเวลา 50 มิลลิวินาที หรือกำหนดจากสัญญาณที่ป้อนให้กับขา OSC_{in} จะเห็นว่าไม่จำเป็นต้องต่อตัวเก็บประจุที่ขารีเซตของ CPU เพราะภายใน MAX691 ได้กำหนดช่วงเวลาในการรีเซตให้แล้ว สำหรับการรีเซต high จะต้องต่อตัวต้านทานพูลอัพเพื่อเพิ่มกระแสให้สามารถรีเซตได้ด้วย

- การตรวจจับแรงดันตก

ไฟเลี้ยงของระบบหรือแรงดันจากแบตเตอรี่ จะถูกแบ่งด้วยตัวต้านทานภายนอก (ดูวงจรรูปที่ 2) แล้วป้อนให้กับขา PFI (power fail input) เพื่อเป็นระดับในการเปรียบเทียบ การตกของไฟเลี้ยงหรือแบตเตอรี่ เช่น ถ้าต้องการตรวจจับไฟเลี้ยงที่ต่ำกว่า 4.65 โวลต์ จะเกิดสัญญาณ low ที่ขา PFO และให้เป็นการอนมาสเคเบิลอินเทอร์รัพต์ให้แก่ CPU เพื่อทำการเก็บข้อมูลที่สำคัญไว้

- การป้องกันการเขียนข้อมูลในแรม

สัญญาณ CE_{out} ซึ่งใช้เป็นชิพอีนาเบิลให้กับชิพอสแรมที่ต้องการแบคอัพจะเกิดตามสัญญาณที่ขา CE_{in} และในสภาวะที่ไฟเลี้ยงปกติคือ สูงกว่า 4.65 โวลต์เท่านั้น เมื่อไฟเลี้ยงตกลงต่ำกว่านี้จะทำให้ขา CE_{out} เป็น high เป็นการป้องกันการเขียนข้อมูลได้ และทำให้ขา LOW LINE เป็น low เพื่อให้ CPU ตรวจสอบสถานะของระบบได้ด้วย

- การตัดต่อแรงกัน V_{out}

แรงดันที่ขา V_{out} จะได้จากแรงดันที่สูงกว่า ระหว่างไฟเลี้ยงกับแบตเตอรี่และจ่ายกระแสได้ 50 มิลลิแอมป์ ซึ่งก็เพียงพอกับชิพอสแรม 1-2 ตัว แต่ถ้าต้องการกระแสที่สูงขึ้นก็ต่อทรานซิสเตอร์ pnp ขนานกับไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ V_{out} จากภายนอกได้ โดยใช้สัญญาณจากขา BATTON กระแส 25 มิลลิแอมป์

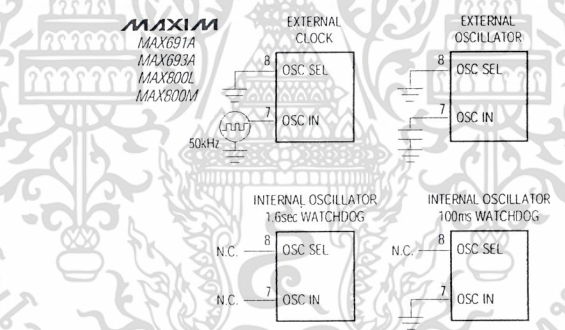
- วงจรวอตช์ด็อกและการใช้งาน

ในสภาพการใช้งานจริงของระบบไมโครฯ อาจจะมีสัญญาณรบกวนจากภายนอก เช่น สาย AC หรือสนามแม่เหล็ก ทำให้ CPU ทำงานผิดพลาดได้ วอตช์ด็อก (watchdog) เป็นวงจรที่คอยติดตามการทำงานของ CPU เพื่อให้ระบบทำงานถูกต้องอยู่ตลอดเวลาโดย CPU จะทำการกระตุ้นวงจรวอตช์ด็อกในช่วงเวลาที่กำหนดอย่างต่อเนื่อง แต่ถ้า CPU เกิดการ hang หรือทำงานผิดพลาดไปก็จะไม่กระตุ้นวงจรวอตช์ด็อกในเวลาที่กำหนดได้ วงจรวอตช์ด็อกก็จะส่งสัญญาณรีเซ็ต CPU เพื่อให้กลับมาทำงานอีกครั้ง การกระตุ้นวงจรวอตช์ด็อกทำได้โดยการเปลี่ยนระดับสัญญาณที่ขา WDI โดยสัญญาณนี้อาจจะมาจากเอาต์พุตพอร์ตหรือจากสัญญาณ port select ก็ได้

- การกำหนดช่วงเวลาของวงจรวอตช์ด็อก

การให้วงจรวอตช์ด็อกภายใน MAX691 ทำงานได้ โดยการเปลี่ยนลอจิกที่ขา WDI จาก 0 ไปเป็น 1 หรือ 1 ไปเป็น 0 ในช่วงเวลาที่กำหนด บางระบบเราอาจจะกำหนดให้กระตุ้นภายในช่วงสั้น ๆ เพื่อว่าถ้าเกิดความผิดพลาดขึ้นจะได้รีเซ็ต CPU ในระยะเวลาสั้น ๆ ด้วย จะได้ไม่เกิดความเสียหายขึ้นกับอุปกรณ์บางตัว

เราสามารถกำหนดช่วงเวลาของวงจรวอตช์ด็อกและการรีเซ็ตของ MAX691 ได้ 4 แบบ โดยที่รายละเอียดของแต่ละแบบมีดังนี้



รูปที่ 2.35 การต่อ MAX691 เพื่อกำหนด time period

- ใช้สัญญาณนาฬิกาจากภายนอก โดยการต่อ OSC (sel) กับกราวด์ และป้อนสัญญาณความถี่ไม่เกิน 250 กิโลเฮิร์ตซ์ เข้าที่ขา OSC (in) ได้ช่วงเวลาวอตช์ด็อกเท่ากับสัญญาณนาฬิกาจำนวน 1,024 ลูก ของสัญญาณที่ขา OSC(in) และเวลารีเซ็ตเท่ากับสัญญาณนาฬิกาจำนวน 2,048 ลูก
- ใช้ฮอสซิลเลเตอร์จากภายนอกโดยการต่อขา OSC_{sel} กับกราวด์ และจำนวน 2,048ลูกเตอร์จากภายนอก โดยต่อขา OSC_{sel} กับกราวด์ และต่อตัวเก็บประจุเข้ากับขา OSC_{in} จะได้เวลาของวอตช์ด็อก และเวลารีเซ็ต
- ใช้ฮอสซิลเลเตอร์ภายใน ถ้าปล่อยขา OSC_{sel} และ OSC_{in} ลอยไว้ได้ช่วงเวลาวอตช์ด็อก 1.6 วินาที และเวลารีเซ็ต 50 มิลลิวินาที

เอกสารนี้เป็นเอกสาร 4. ที่ใช้ฮอสซิลเลเตอร์ภายใน ถ้าปล่อยขา OSC_{sel} ลอยไว้ และต่อขา OSC_{in} กับกราวด์ ได้ช่วงเวลาที่ไม่ว่ากรวอตช์ด็อก 100 มิลลิวินาที และเวลารีเซ็ต 50 มิลลิวินาทีข้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAX691 เป็นไอซีที่ทำหน้าที่ได้หลายอย่างในตัวเดียวกัน โดยไม่ต้องไปเพิ่มเติมวงจรเหล่านั้นให้ยุ่งยาก และยังทำให้ขนาดของระบบไมโคร ๗ มีขนาดเล็ก

2.12 การเชื่อมต่อระหว่างคอมพิวเตอร์กับพริ้นเตอร์

การสื่อสารข้อมูลกับพริ้นเตอร์ คอมพิวเตอร์สามารถส่งข้อมูล 3 ชนิด ไปที่พริ้นเตอร์ คือข้อมูลตัวอักษร รหัสควบคุม และข้อมูลกราฟฟิก ข้อมูลอักขระแสดงได้ในรูปของตัวอักษร ตัวเลข เครื่องหมายวรรคตอน สัญลักษณ์ต่างๆ รหัสควบคุมจะถูกใช้เพื่อส่งคำสั่งไปยังพริ้นเตอร์ รหัสเหล่านี้จะกำหนดโหมดการทำงาน เช่น รูปแบบตัวพิมพ์ (Font Style) ขนาดตัวอักษร หรือการควบคุมที่สามารถกระทำได้โดยตรง เช่น การเลื่อนกระดาษทีละบรรทัด หรือทีละหน้า การใช้รหัสควบคุมแยกแยะมีความจำเป็นต่อการทำงาน ขณะที่พิมพ์เอกสาร ซึ่งรหัสควบคุมอื่นๆ จะทำให้พริ้นเตอร์ทำงานในโหมดกราฟฟิกต่างๆ ได้

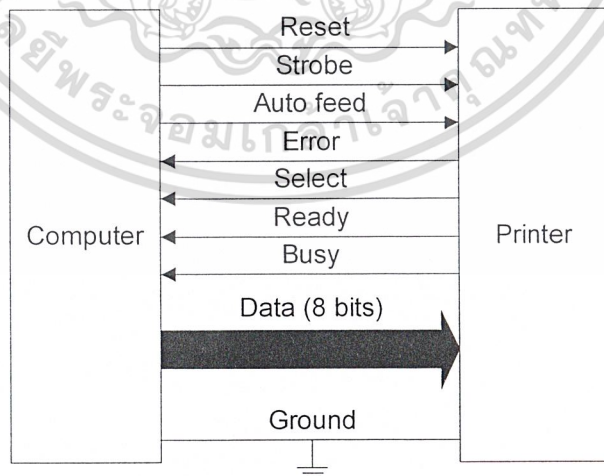
2.12.1 ลักษณะของการสื่อสารข้อมูล

ข้อมูลที่อยู่ในรูปแบบไบนารี (Binary) ที่จะใช้ถ่ายเทระหว่างคอมพิวเตอร์ และพริ้นเตอร์ต้องใช้ตัวกลางในการสื่อสารคือ สายเคเบิล การผลิตและออกแบบคุณสมบัติของสายเคเบิลจะต้องขึ้นอยู่กับวิธีที่เราใช้ติดต่อ ซึ่งมักใช้อยู่ 2 แบบ คือ

1. แบบอนุกรม
2. แบบขนาน

การติดต่อแบบขนาน (Parallel) เป็นวิธีที่ง่ายที่สุด เนื่องจาก เราส่งข้อมูลตรงๆ เลย ดังแสดงในรูป

ที่ 2.36



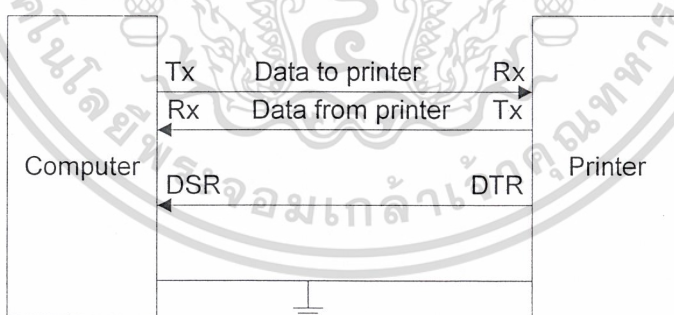
รูปที่ 2.36 แสดงการติดต่อระหว่างคอมพิวเตอร์และพริ้นเตอร์แบบขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สังเกตได้ว่า ข้อมูล 8 บิต ส่งแยก บิต D_0 ถึง บิต D_7 พร้อม กัน แต่ถ้าจะให้ส่งได้อย่างถูกต้อง คอมพิวเตอร์ และพริ้นเตอร์ต้องซิงโครไนส์ (Synchronous) กัน คือ พริ้นเตอร์จะรับข้อมูลเมื่อมันขอ หรือ พริ้นเตอร์สั่งให้คอมพิวเตอร์รอจนกว่าพริ้นเตอร์จะพร้อมรับข้อมูล ดังนั้นการซิงโครไนส์ของการสื่อสาร ข้อมูลแบบขนานจำเป็นต้องมีสายควบคุมจำนวนมากสำหรับสัญญาณพริ้นเตอร์ และ สัญญาณคอมพิวเตอร์การทำงานอย่างสัมพันธ์กันดังกล่าวจะเรียกว่า การแฮนด์เชค (Handshaking)

การส่งแบบขนานมีความเร็วพอใช้ โดยพริ้นเตอร์จะรับข้อมูลเร็วเท่าๆ กับที่คอมพิวเตอร์ส่งไป ซึ่งมีความเร็วประมาณ 1,000 cps. เมื่อข้อมูลหนึ่งตัวอักษรจะมี 8 บิต ความเร็วจะมากกว่า 8,000 บิต ต่อ วินาที บัส (Bus) ข้อมูลภายในพริ้นเตอร์มี 8 บิต จะสามารถรับข้อมูลในบัฟเฟอร์ (Buffer) ได้โดยเพิ่มวงจร เพียงเล็กน้อย การส่งข้อมูลแบบขนานเป็นการติดต่ออย่างง่าย แต่มีข้อเสียจุดใหญ่คือ ใช้สายเคเบิลมากแต่ จะมีความเร็วในการส่งสูง และมีสัญญาณรบกวนทางอิเล็กทรอนิกส์ ซึ่งทำให้เกิดความผิดพลาดของข้อมูล ได้

การติดต่อแบบอนุกรม สามารถทำได้ง่าย เพราะมีจำนวนสายน้อยลง ดังรูปที่ 2.37 แต่การทำงาน จะยุ่งยากกว่า เราจะพบว่า จากรูปที่ 2.2 มีสายสองเส้นที่ใช้ในการส่งข่าวสาร เส้นหนึ่งสำหรับส่งข้อมูลจาก คอมพิวเตอร์ไปยังพริ้นเตอร์ และอีกหนึ่งเส้นสำหรับส่งข้อมูลจากพริ้นเตอร์ไปยังคอมพิวเตอร์ เพราะว่า ข้อมูลสามารถส่งไปมาได้ทั้งสองทิศทาง จึงเรียก การติดต่อสื่อสารแบบนี้ว่า การสื่อสารข้อมูลแบบสอง ทิศทาง (Bidirectional Data Link) ถ้ามีสายเส้นเดียวจะสามารถส่งหรือรับข้อมูลได้เพียงอย่างเดียวอย่างหนึ่ง ใน เวลาหนึ่งเท่านั้น ข้อมูลแบบอนุกรมต้องซิงโครไนส์กันระหว่างคอมพิวเตอร์ และพริ้นเตอร์ ดังนั้นเราต้อง เพิ่มซิงโครไนซ์บิต ที่จุดเริ่มต้น และจุดสิ้นสุด ของแต่ละข้อมูล อักขระที่ส่ง และ ยังมีพาริตีบิต (Parity Bit) ที่ต้องรวมอยู่ด้วย สำหรับตรวจสอบผิดพลาด



รูปที่ 2.37 แสดงการติดต่อระหว่างคอมพิวเตอร์และพริ้นเตอร์แบบอนุกรม

การทำแฮนด์เชคแบบอนุกรมนั้น แบ่งเป็นทั้ง ซอฟต์แวร์ และฮาร์ดแวร์ โดยซอฟต์แวร์ใช้ในการติดต่ออนุกรมแบบสองทิศทาง เช่น ใช้สั่งพริ้นเตอร์ให้ส่งรหัสควบคุมยังคอมพิวเตอร์ เช่น รหัส สำหรับการแฮนด์เชค “XON” และ “XOFF” ในส่วนของฮาร์ดแวร์ของการแฮนด์เชคนั้น ไม่ได้ใช้ในการส่ง ข้อมูลจากพริ้นเตอร์ไปยังคอมพิวเตอร์ สัญญาณแฮนด์เชคที่แจ้งให้คอมพิวเตอร์สัญญาณพริ้นเตอร์ไม่ว่าจะ ดังนั้นการติดต่อจึงจะมีมากกว่าหนึ่งสายแฮนด์เชค โดยทั่วไปเรามักจะให้มีการแฮนด์เชคระหว่าง

พริ้นเตอร์ค่อนข้างมาก แม้ว่าการทำงานของ การสื่อสารข้อมูลแบบอนุกรมจะยุ่งยากแต่การติดต่อแบบอนุกรมก็เป็นที่ยอมรับกันมาก เนื่องจากความสามารถในการติดต่อแบบสองทิศทาง และสามารถใช้งานในระยะไกล

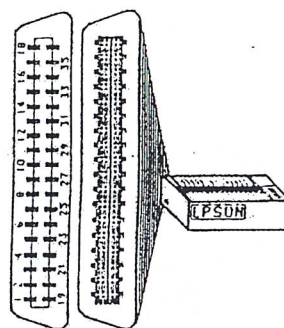
2.12..2 การอินเตอร์เฟสกับพริ้นเตอร์แบบขนานและขาการเชื่อมต่อแบบเซ็นโทรนิคส์

ข้อมูลที่จะส่งไปพิมพ์ที่พริ้นเตอร์จะส่งในลักษณะของรหัสแอสกี (ASCII) ผ่านสายแบบขนาน 8 สาย พริ้นเตอร์รับตัวอักษรเพื่อจะพิมพ์ และ เก็บข้อมูลไว้ในบัฟเฟอร์แรกภายใน เมื่อพริ้นเตอร์ตรวจพบอักขระแคริเอจ รีเทิร์น (Carriage Return : ODH) มันจะพิมพ์อักขระแถวแรกจากบัฟเฟอร์ เมื่อพริ้นเตอร์ตรวจพบแคริเอจ รีเทิร์น ตัวที่สอง มันจะพิมพ์อักขระแถวที่สอง ออกมา ขบวนการต่างๆ จะดำเนินต่อไปจนกระทั่งตัวอักษรที่ต้องการทั้งหมดถูกพิมพ์

การแปลงรหัสแอสกีจากไมโครคอมพิวเตอร์ส่งไปยังพริ้นเตอร์จะต้องทำบนพื้นฐานการแฮนด์เชด เพราะ ไมโครคอมพิวเตอร์สามารถส่งตัวอักษรเร็วกว่าความสามารถในการพิมพ์ของพริ้นเตอร์มาก พริ้นเตอร์จะต้องมีแนวทางเพื่อที่จะบอกให้ไมโครคอมพิวเตอร์รู้ว่าบัฟเฟอร์ของมันเต็มแล้ว และไม่สามารถส่งข้อมูล อักขระได้อีกจนกระทั่งมันพิมพ์ออกมาแล้ว มารตราฐานสำหรับการอินเตอร์เฟสกับพริ้นเตอร์ของพริ้นเตอร์เป็นมาตรฐานของ เซ็นโทรนิคส์ (Centronics Parallels Standard)

พริ้นเตอร์ชนิดเซ็นโทรนิคส์ โดยปกติแล้วจะมี 36 ขา ในการอินเตอร์เฟส (Interface) ตารางที่ 2.11 แสดงการกำหนดขา และ คำอธิบายสำหรับขาคอนเนคเตอร์ (Connector) การที่มีขาสำหรับการเชื่อมต่อกับพริ้นเตอร์มากถึง 36 ขา นั้นเนื่องจากสายของแต่ละสัญญาณข้อมูลจะมีสายกราวด์ (Ground) ของมันแยกออกจากกัน เช่น จากตารางที่ 2.11 ขา 2 คือ บิทข้อมูลที่มีนัยสำคัญต่ำสุด (LSB) ของข้อมูลที่จะส่งไปยังพริ้นเตอร์ และขา 20 จะเป็นกราวด์สำหรับสายสัญญาณนี้ เหตุผลสำหรับการแยกสัญญาณกราวด์นี้เพื่อลดสัญญาณรบกวนในสาย ถ้าจะทำการต่อสายเคเบิลแบบขนานกับพริ้นเตอร์ สายกราวด์ทั้งหมดจะต้องถูกต่อเข้าด้วยกันเพื่อต่อกับกราวด์ของไมโครคอมพิวเตอร์ที่ปลายของสายเคเบิล โดยขา 16 เรียกว่า ลอจิกกราวด์ (Logic Ground) และ ขา 17 เรียกว่า เซสลิส กราวด์ (Chassis Ground) เพื่อป้องกันกระแสสัญญาณรบกวนที่มาจากสายของลอจิกกราวด์ เราจึงต้องเชื่อมต่อสายเหล่านี้เข้าด้วยกันกับไมโครคอมพิวเตอร์ ส่วนขาอื่นๆ ในจำนวน 36 ขา ของคอนเนคเตอร์นั้น แบ่งออกได้เป็น 2 ประเภท คือ สัญญาณที่ส่งไปยังพริ้นเตอร์ เพื่อที่จะบอกว่าจะไรกำลังทำงาน และ สัญญาณจากพริ้นเตอร์เพื่อที่จะแสดงสถานะของมัน สัญญาณควบคุมหลักไปยังพริ้นเตอร์ คือ INIT ที่ขา 31 ซึ่งจะบอกพริ้นเตอร์เพื่อให้ทำการกำหนดค่าเริ่มต้นภายในของลำดับ และ STROBE ที่ขา 1 ซึ่งจะบอกพริ้นเตอร์ว่า ขณะนี้มีข้อมูลส่งมาแล้ว ส่วนอีก 2 ขา อินพุท (Input) คือ ขา 14 และ ขา 36 โดยปกติจะใช้สำหรับคูแลกภายในพริ้นเตอร์ โดยตัวคอนเนคเตอร์จะมีลักษณะดังรูปที่ 2.38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.38 แสดงขาของคอนเน็คเตอร์แบบอิมพีเนอล (Amphenol)

ตารางที่ 2.12 แสดงการเชื่อมต่อขาและคำอธิบายสำหรับการเชื่อมต่อแบบเซนโทรนิคส์ขนาน

ขา สัญญาณ ที่	ขา สัญญาณ ตอกลับ	สัญญาณ	ทิศทาง	คำอธิบาย
1	19	STROBE	เข้า	พัลส์ (PULSE) STROBE เพื่อจะอ่านข้อมูลเข้ามา ความกว้างของพัลส์จะต้องมากกว่า 0.5 μ s ที่เทอร์มินัลด้านรับ ระดับสัญญาณปกติจะอยู่ในสภาวะ “สูง” การอ่านข้อมูลเข้ามากระทำที่ระดับสัญญาณ “ต่ำ”
2	20	DATA 1	เข้า	สัญญาณเหล่านี้ จะแสดงข่าวสารข้อมูลของ บิตที่ 1-8 ของข้อมูลขนานตามลำดับ แต่ละสัญญาณจะมีระดับ “สูง” เมื่อมีข้อมูลเป็นลอคจิก “0”
3	21	DATA 2	เข้า	
4	22	DATA 3	เข้า	
5	23	DATA 4	เข้า	
6	24	DATA 5	เข้า	
7	25	DATA 6	เข้า	
8	26	DATA 7	เข้า	
9	27	DATA 8	เข้า	
10	28	ACKNLG	ออก	พัลส์ประมาณ 5 μ s สถานะ “ต่ำ” แสดงว่าข้อมูลนั้นถูกรับโดยพรีนเตอร์แล้วและพร้อมที่จะรับข้อมูลอื่นต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ควรนำเอกสารนี้ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11	29	BUSY Or READY	ออก	สัญญาณ “สูง” แสดงว่าพริ้นเตอร์ไม่สามารถรับข้อมูล ถ้าสัญญาณกลายเป็นสถานะ “สูง” จะมีกรณีต่างๆ ดังนี้ 1. อยู่ในระหว่างข้อมูลเข้า 2. อยู่ในระหว่างกระทำการพิมพ์ 2. อยู่ในสภาวะออฟไลน์ (offline) 4. อยู่ในสภาวะเกิดเหตุผิดพลาดที่พริ้นเตอร์
12	30	PE	ออก	สัญญาณ “สูง” แสดงว่าพริ้นเตอร์อยู่ในสภาวะไม่มีกระดาษ
13	-	SLCT	ออก	สัญญาณนี้แสดงว่าพริ้นเตอร์อยู่ในสภาวะถูกเลือก
ขา สัญญาณ ที่	ขา สัญญาณ ตอบกลับ	สัญญาณ	ทิศทาง	คำอธิบาย
14		AUTO FEEDXT	เข้า	ถ้าสัญญาณนี้อยู่ที่ระดับ “ต่ำ” กระดาษจะถูกเลื่อนโดยอัตโนมัติ 1 บรรทัด หลังจากการพิมพ์ (ระดับสัญญาณสามารถกำหนดที่ระดับ “ต่ำ” กับดิฟ- สวิตช์ ขา 2-3 เตรียมไว้สำหรับควบคุมแผงวงจร)
15	-	NC		ไม่ใช่
16	-	OV		ระดับลอจิกกราวน์
17	-	CHASSIS- GND		พริ้นเตอร์ แชนลิสกราวน์ ในพริ้นเตอร์ แชนลิสกราวน์และลิจิกกราวน์จะแยกออกจากกันและกัน
18	-	NC	-	ไม่ใช่
19-30	-	GND	-	สัญญาณทวิสต์แพร์รีเทิร์น (Twilted Pair Return)ระดับกราวน์
31	-	INIT	เข้า	เมื่อระดับของสัญญาณนี้กลายเป็นสถานะ “ต่ำ” ส่วนควบคุมพริ้นเตอร์จะถูกรีเซตเพื่อไปที่ค่าสถานะเริ่มต้นของมัน (Initial State) และพริ้นท์บัฟเฟอร์ (Print Buffer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามแก้ไขเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

32	-	_____	ออก	ระดับของสัญญาณนี้จะกลายเป็นสถานะ “ต่ำ” เมื่อพริเตอร์อยู่ในสถานะเปเปอร์เอนด์ (Paper End) ออฟไลน์ หรือ เออเรอร์ (Error)
33	-	GND	-	เหมือนกับขาที่ 19-30
34	-	NC	-	ไม่ใช่
35	-			ต่อกับ 5 VDC ผ่านตัวต้านทาน 4.7 k-ohms
36	-	_____	เข้า	ข้อมูลเข้ามาที่พริเตอร์อาจเป็นไปได้เมื่อระดับของสัญญาณนี้อยู่ในสถานะ “ต่ำ” (การกำหนดภายในสามารถทำได้ที่ ดิฟสวิทช์ 1-8 เงื่อนไขที่เวลาของการส่งจะถูกกำหนดให้ให้อยู่ในสถานะ “ต่ำ” สำหรับสัญญาณนี้)

หมายเหตุ

1. ทิศทาง หมายถึง ทิศทางการไหลของสัญญาณที่มองจากพริเตอร์
2. สัญญาณตอบกลับหรือทวิสท์แพร์ที่เทอร์มินัลจะถูกต่อกับสัญญาณที่ระดับกราวด์ เมื่อมีการอินเตอร์เฟส ต้องแน่ใจว่าใช้สายเคเบิลแบบทวิสท์แพร์ สำหรับแต่ละสัญญาณ และต้องเชื่อมต่ออย่างสมบูรณ์ที่ด้านตอบกลับ
3. เงื่อนไขทั้งหมดของการอินเตอร์เฟสมีพื้นฐานบนระดับของ TTL ทั้งเวลาขึ้น และ ขาลงของแต่ละสัญญาณต้องน้อยกว่า 0.2 μ s
4. ข้อมูลที่ส่งต้องมีสัญญาณ ACKNLG หรือ สัญญาณ BUSY ด้วย (ข้อมูลที่ส่งไปที่พริเตอร์นี้สามารถส่งเดี่ยว ๆ หลังจากที่มีการยืนยันสัญญาณ ACDNLG หรือ เมื่อระดับของสัญญาณ BUSY เป็นสถานะ “ต่ำ”)

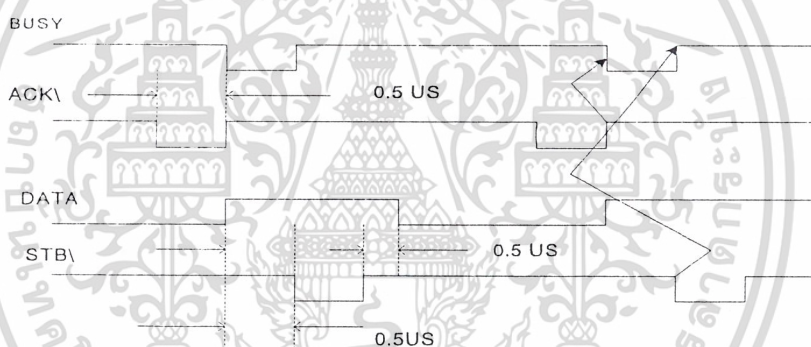
สัญญาณแสดงสถานะเอาต์พุต (Output) จากพริเตอร์มีดังนี้

1. สัญญาณ ACKNLG ที่ขา 10 ซึ่งจะมีสถานะ “ต่ำ” (Low) จะแสดงว่าข้อมูลอักขระได้ถูกรับโดยพริเตอร์แล้วและพร้อมที่จะรับตัวอักขระต่อไป
2. สัญญาณ BUSY ที่ขา 11 ซึ่งจะมีสถานะ “สูง” (High) ในกรณีที่ไม่มีการคาย หรือพริเตอร์ยังไม่พร้อมที่จะรับข้อมูลอักขระ
3. สัญญาณ PE ที่ขา 12 ซึ่งจะมีสถานะ “สูง” ถ้าสวิทช์ของการไม่มีการคายในพริเตอร์ถูกกระตุ้น
4. สัญญาณ SLCT ที่ขา 13 ซึ่งจะมีสถานะ “สูง” ถ้าพริเตอร์ถูกเลือกสำหรับการรับข้อมูล
5. สัญญาณ ERROR ที่ขา 32 ซึ่งจะมีสถานะ “ต่ำ” เมื่อเกิดปัญหาในเงื่อนไขต่างๆ ของพริเตอร์

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่สามารถนำไปใช้เพื่อวัตถุประสงค์อื่นใดได้โดยไม่ได้รับอนุญาต
ไม่ว่าในรูปแบบใดก็ตาม สัญญาณที่แสดงโดยสัญญาณและตัวอักษรที่ปรากฏในเอกสารฉบับนี้ได้รับการกำหนดค่าเริ่มต้นแล้ว ขึ้น

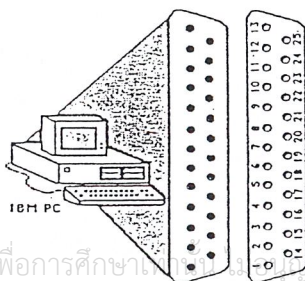
แรกต้องทำการตรวจสอบสัญญาณที่ขา BUSY เพื่อจะดูว่าพรินเตอร์พร้อมที่จะรับข้อมูลหรือไม่ ถ้าสัญญาณอยู่ในสถานะ “ต่ำ” แสดงว่าพรินเตอร์พร้อมจะรับข้อมูล เราจะสามารถส่งรหัสแอสกีบนสายข้อมูลขนานทั้ง 8 สายได้ หลังจากอย่างน้อย 0.5 μ s เราแสดงสัญญาณ STROBE ที่สถานะ “ต่ำ” เพื่อจะบอกพรินเตอร์ว่าข้อมูลตัวอักษรถูกส่งไปแล้ว สัญญาณ STROBE จะเป็นสถานะ “ต่ำ” เพราะพรินเตอร์แสดงสัญญาณ BUSY ของมันเป็นสถานะ “สูง” หลังจากเวลาอย่างน้อยที่สุด 0.5 μ s แล้วสัญญาณ STROBE สามารถขึ้นเป็นสถานะ “สูง” ได้อีกครั้งหนึ่ง โดยข้อมูลข่าวสารต้องคงสภาพการใช้งานได้บนสายข้อมูลอย่างน้อย 0.5 μ s หลังจากสัญญาณ STROBE ถูกทำให้อยู่ในสถานะ “สูง”

เมื่อพรินเตอร์พร้อมจะรับข้อมูลอีกขระถัดไป มันจะแสดงสัญญาณ ACKNLG ในสถานะ “ต่ำ” เป็นเวลา 5 μ s ขอบขาขึ้นของสัญญาณ ACKNLG จะบอกไมโครคอมพิวเตอร์ว่ามันสามารถส่งข้อมูลอีกขระถัดไปได้แล้ว ขอบขาขึ้นของสัญญาณ ACKNLG จะทำการรีเซ็ตสัญญาณ BUSY จากพรินเตอร์ สัญญาณ BUSY จะเป็นสถานะ “ต่ำ” เพื่อแสดงว่าพรินเตอร์พร้อมจะรับข้อมูลอีกขระถัดไป บางระบบจะใช้สัญญาณ ACKNLG สำหรับการแฮนด์เชค และ ระบบจะใช้สัญญาณ BUSY



รูปที่ 2.39 แสดงช่วงเวลาของรูปคลื่นที่จะส่งข้อมูลอีกขระไปยังพรินเตอร์แบบขนาน

ส่วนทางฝั่งคอมพิวเตอร์ โดยปกติแล้วจะใช้คอนเนคเตอร์ของการต่อขาแบบ DB-25 ซึ่งเป็นตัวผู้ และพอร์ตของคอมพิวเตอร์จะเป็นตัวเมีย ดังนั้นในการต่อออกจากคอมพิวเตอร์จะต้องดูด้วยว่าจะต่อแบบใด ดังรูปที่ 2.39



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาและเป็นสมบัติของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงชื่อของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.40 แสดงขาของคอนเนคเตอร์แบบ DB-25

โดยรายละเอียดของขาของ DB-25 แสดงไว้ในตารางที่ 2.13 และตารางที่ 2.14 แสดงการปรับให้ การต่อของ DB-25 กับแอมฟินอล สัมพันธ์กัน

ตารางที่ 2.13 แสดงการเชื่อมต่อสำหรับการต่อขาแบบ DB-25

ขา DB-25	หน้าที่
1	STROBE
2	DATA BIT 0
3	DATA BIT 1
4	DATA BIT 2
5	DATA BIT 3
6	DATA BIT 4
7	DATA BIT 5
8	DATA BIT 6
9	DATA BIT 7
10	ACKNLG
11	BUSY
12	PAPER END (OUT OF PAPER)
13	SELECT
14	AUTOFEED
15	ERROR

ตารางที่ 2.14 แสดงการปรับต่อขาของ DB-25 และ แอมฟินอล

ขา DB-25	Amphenol
1	1
2	2
3	3
4	4
5	5
6	6
7	7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	32
16	31
17	36
18	33
19	19
20	21
22	25
23	27
24	29
25	30

2.13 การติดต่อสื่อสารข้อมูลผ่านพอร์ตอนุกรม RS-232C

2.13.1 พอร์ตอนุกรม (SERIAL PORT)

เนื่องจากในปัจจุบันมีการใช้งานตามมาตรฐานการเชื่อมต่อแบบ RS-232-C กันอย่างแพร่หลายดังนั้นผู้เขียนจึงขอกล่าวรายละเอียดเฉพาะรายละเอียดเฉพาะมาตรฐานการเชื่อมต่อแบบ RS-232-C เท่านั้นซึ่งเป็นมาตรฐานที่กำหนดโดย EIA ซึ่งเป็นองค์กรอุตสาหกรรมอิเล็กทรอนิกส์ของอเมริกาโดยแบ่งการเชื่อมต่อออกเป็น 2 ลักษณะคือ DTE (Data Terminal Equipment) และ DCE (Data communication Equipment) ซึ่งโดยปกติ DTE จะต้องต่อเข้ากับ DCE เสมอ เช่นการต่อเครื่องคอมพิวเตอร์ (อุปกรณ์ DTE) เข้ากับโมเด็ม (อุปกรณ์ DCE) เป็นต้น

พอร์ตอนุกรม RS-232-C จะเป็นพอร์ตของเครื่องคอมพิวเตอร์ที่มีขาต่อ (connector) ทั้งประเภท 9 และ 25 ขาและเราเรียกกันว่าพอร์ต COM1: และพอร์ต COM2: นั่นเองในความเป็นจริงพอร์ตอนุกรมไม่ได้ถูกควบคุมโดยตรงจาก CPU บนเมนบอร์ด แต่การสื่อสารทั้งหมดจะถูกจัดการโดยชิป UART (Universal Asynchronous Receiver/Transmitter) อีกทีหนึ่งซึ่งในปัจจุบันเบอร์ที่ใช้กันมากที่สุดก็คือเบอร์

16550C ซึ่งเป็นเวอร์ชันที่ได้รับการแก้ไขข้อผิดพลาดแล้วซึ่งชิป UATR นี้จะทำหน้าที่ในการรับและส่งข้อมูลเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า

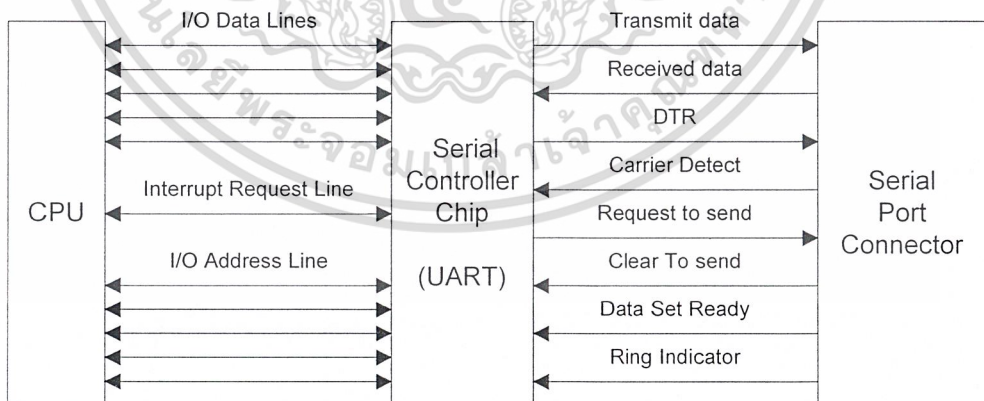
มุลคงต่อไป
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรับส่งข้อมูล (Data transmission)

- รับตัวอักษรจากเครื่องคอมพิวเตอร์
- แปลงตัวอักษรให้เป็นสายข้อมูลแบบบิต (เราเรียกว่าขบวนการ serialization)
- สร้งแฟรมข้อมูล โดยการเพิ่มบิตที่จำเป็นสำหรับการสื่อสารและการตรวจสอบ เช่น บิต START , STOP , และ PARITY เป็นต้น
- ส่งผ่านแฟรมข้อมูลที่สร้างขึ้นมา จากชั้นตอนที่ผ่านมาด้วยความเร็วของโมเด็มหรือพอร์ตอนุกรม (baud rate)
- แสดงสถานะความพร้อมที่จะรับข้อมูลตัวอักษรถัดไปให้กับเครื่องคอมพิวเตอร์

การรับข้อมูล (Data Receiver)

- รับตัวอักษรจากอินเตอร์เฟซ
 - ตรวจสอบความถูกต้องของแฟรมข้อมูลตามมาตรฐานแฟรมที่กำหนด โดยถ้าหากแฟรมข้อมูลมีรูปแบบที่ไม่ถูกต้องก็จะมีแจ้งเตือนผิดพลาดทันที
 - ตรวจสอบความถูกต้องของพาริตี
 - แปลงสายข้อมูลแบบบิตให้เป็นตัวอักษร
 - ส่งตัวอักษรให้กับเครื่องคอมพิวเตอร์
 - แสดงสถานะความพร้อมที่จะรับข้อมูลตัวอักษรถัดไปให้กับอินเตอร์เฟซ
- สำหรับการเชื่อมต่อสายสัญญาณต่างๆ ระหว่าง CPU ของเมนบอร์ดของเครื่องคอมพิวเตอร์กับพอร์ตอนุกรมนั้น จะต้องกระทำผ่านทางชิป UART ซึ่งจะมีวิธีการเชื่อมต่อดังรูปที่ 12.1



รูปที่ 2.41 การเชื่อมต่อสายสัญญาณระหว่าง CPU ของเครื่องคอมพิวเตอร์กับพอร์ตอนุกรม เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.40 จะเห็นว่าการเชื่อมต่อระหว่าง UART กับพอร์ตอนุกรมนั้น มีสายสัญญาณมากมายที่ช่วยให้การสื่อสารมีความถูกต้องมากขึ้น โดยที่สายสัญญาณแต่ละเส้นมีความหมายดังต่อไปนี้

- Transmit Data (TD)

เป็นวงจรที่สร้างสัญญาณ Transmitted data ซึ่งถูกส่งผ่านทาง DTE ไปยัง DCE ซึ่งสัญญาณที่ส่งออกมาจะเป็นโค้ดคำสั่งของโมเด็มหรือข้อมูลก็ได้

- Received Data (RD)

เป็นวงจรที่สร้างสัญญาณ received data ซึ่งถูกส่งจาก DCE ไปยัง DTE ซึ่งสัญญาณที่ส่งออกมาจะเป็นโค้ดคำสั่งของโมเด็มหรือข้อมูลก็ได้ ซึ่งเป็นสัญญาณที่มีทิศทางตรงข้ามกับสัญญาณ transmitted data

- Data Terminal Ready (DTR)

สัญญาณ DTR จะถูกส่งจาก DTE ไปยัง DCE เพื่อเป็นการแจ้งความพร้อมในการสื่อสารให้โมเด็มได้ทราบ โดยถ้าหากโมเด็ม (อุปกรณ์ DCE) มีความสามารถในการตอบรับแบบอัตโนมัติ (automatically answer) โมเด็มก็จะสามารถตอบรับได้เฉพาะเมื่อสัญญาณ DTR อยู่ในสถานะ on เท่านั้น

- Carrier Detect (CD)

สัญญาณ CD จะถูกส่งผ่าน DCE ไปยัง DTE เพื่อเป็นการแจ้งว่าโมเด็มอยู่ในสถานะกำลังติดต่อกับโมเด็มตัวอื่น หรือโมเด็มกำลังได้รับสัญญาณที่พร้อมสำหรับการติดต่อสื่อสาร สำหรับสัญญาณ Carrier Detect นี้ผู้อ่านสามารถเรียกอีกชื่อได้ว่า Received Line Signal Detector

- Request to send (RTS)

สัญญาณ RTS จะถูกส่งจาก DTE ไปยัง DCE โดยเมื่อสัญญาณอยู่ในสถานะ on ก็หมายถึงเครื่องคอมพิวเตอร์พร้อมที่จะรับข้อมูลจากโมเด็ม และในทางกลับกันถ้าหากสัญญาณ RTS อยู่สถานะ off ก็หมายถึงเครื่องคอมพิวเตอร์ไม่พร้อมที่จะรับข้อมูลจากโมเด็ม

- Data Set Ready (DSR)

สัญญาณ DSR จะถูกส่งจาก DCE ไปยัง DTE เพื่อเป็นการแจ้งความพร้อมในการสื่อสารจากโมเด็มให้กับเครื่องคอมพิวเตอร์ได้ทราบโดยสัญญาณ DSR จะอยู่ในสถานะ on ก็ต่อเมื่อโมเด็มได้รับสัญญาณ DTR เท่านั้น

- Ring indicator (RI)

สัญญาณ RI จะถูกส่งจาก DCE ไปยัง DTE เพื่อเป็นการแจ้งให้เครื่องคอมพิวเตอร์ทราบว่าโมเด็มกำลังได้รับสัญญาณกระดิ่ง (Ring Signal) จากโมเด็มตัวอื่น โดยที่สัญญาณ RI จะอยู่ในสถานะ on โมเด็มได้รับสัญญาณกระดิ่งและ off เมื่อโมเด็มไม่ได้รับสัญญาณกระดิ่ง เนื่องจากโมเด็มรุ่นใหม่ๆ ในปัจจุบันมักมีความสามารถในการตอบรับแบบอัตโนมัติ (automatically answer) ดังนั้นจึงไม่มีความจำเป็นต้องใช้สัญญาณ RI อีกต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13.2 รูปแบบของเฟรมข้อมูล (Data Format)

ในการสื่อสารระหว่างโมเด็ม 2 ตัวนั้น โมเด็มทั้งสองจะต้องมีความเข้าใจถึงรูปแบบเฟรมข้อมูลที่ตรงกัน ซึ่งการส่งข้อมูลแบบ asynchronous นั้นนอกจากข้อมูลที่จะต้องส่งผ่านแล็งโมเด็มยังต้องจัดการช่วงจังหวะในการส่งหรือรับข้อมูลระหว่างกันอีกด้วย โดยที่เฟรมข้อมูลในการส่งหรือรับจะประกอบด้วยบิตข้อมูลที่มีความหมายดังต่อไปนี้

- บิตข้อมูล (Data bit)

เมื่อชิป UART ได้รับตัวอักษรที่ส่งมาจากเครื่องคอมพิวเตอร์แล้ว ก็ต้องทำการแปลงตัวอักษรดังกล่าวให้เป็นสายข้อมูลชนิดบิตที่มีความยาวตั้งแต่ 5 ถึง 8 บิต ซึ่งเราเรียกระบบการแปลงตัวอักษรให้เป็นสายข้อมูลชนิดบิตนี้ว่ากระบวนการ Serialization จากนั้นโมเด็มก็ทำการส่งแต่ละบิตไปยังโมเด็มปลายทาง โดยจะเริ่มส่งจากบิตที่มีนัยสำคัญต่ำสุด (least significant bit) ไปยังบิตที่มีนัยสำคัญสูงสุด (most significant)

- บิตเริ่มต้นข้อมูล (START bit)

ในการส่งข้อมูลแบบ asynchronous นั้นเราจะต้องมีวิธีการบอกโมเด็มให้ทราบถึงจุดเริ่มต้นของข้อมูลที่ต้องการส่ง ดังนั้นก่อนหน้าข้อมูลในทุกๆ เฟรมจะต้องถูกนำหน้าด้วยบิตเริ่มต้นข้อมูล (START bit) เสมอ

- บิตสิ้นสุดข้อมูล (STOP bit)

ในการส่งข้อมูลแบบ asynchronous นั้นในกรณีที่บิตเริ่มต้นข้อมูลเกิดการสูญหายในระหว่างการส่งโมเด็มก็จะไม่สามารถทราบถึงจุดสิ้นสุดของสายข้อมูลบิตได้เลย นอกเสียจากโมเด็มจะตรวจพบบิตเริ่มต้นใหม่อีกครั้งเท่านั้น ดังนั้นจึงมีการเพิ่มบิตสิ้นสุดข้อมูลสายข้อมูลบิต และบิตสิ้นสุดข้อมูลนี้เราเรียกว่า เฟรมของข้อมูล (data frame)

- บิตพาริตี (Parity)

เนื่องจากการส่งผ่านข้อมูลผ่านสายโทรศัพท์นั้น สามารถเกิดสัญญาณรบกวนได้ง่าย ด้วยเหตุนี้จึงเป็นไปได้มากที่สถานะของแต่ละบิตของข้อมูลที่ถูกส่งจะมีการเปลี่ยนแปลง เช่น จากบิต 0 เป็นบิต 1 เป็นต้น ในระหว่างการส่งข้อมูลด้วยเหตุนี้จึงมีการคิดค้นวิธีที่ง่ายที่สุดที่ช่วยในการตรวจสอบความถูกต้องของสถานะของบิตข้อมูล ซึ่งเรียกกันว่า การตรวจสอบค่าพาริตี (parity check) โดยในการส่งผ่านข้อมูลด้วยโปรโตคอล START/STOP นั้นในการส่งตัวอักษรข้อความทั่วไปจะใช้เพียง 7 บิตข้อมูลเท่านั้น ดังนั้นจึงมีการเพิ่มบิตพาริตีต่อสายเฟรมข้อมูลของแต่ละตัวอักษรสำหรับหลักการในการคำนวณค่าของพาริตีมีดังนี้

1. ถ้าหากจำนวนของบิตที่ค่าเท่ากับ 1 ของสายบิตข้อมูลมีค่าเป็นเลขคู่ (even number) บิตพาริตีจะมีค่าเท่ากับ 0
2. ถ้าหากจำนวนของบิตที่ค่าเท่ากับ 1 ของสายบิตข้อมูลมีค่าเป็นเลขคู่ (odd number) บิตพาริตีจะมีค่าเท่ากับ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13.3 ซอฟต์แวร์ที่ใช้ในการติดต่อสื่อสารผ่านพอร์ตอนุกรม

คอนโทรล MSCComm (Communication) เป็นคอนโทรลตัวหนึ่งซึ่งช่วยในการติดต่อกับพอร์ตอนุกรม (Serial Port) ซึ่งผู้อ่านสามารถทำการรับ-ส่งข้อมูลผ่านทางพอร์ตอนุกรมได้ด้วยคอนโทรลนี้ เช่น การติดต่อผ่านโมเด็ม (Modem) หรือติดต่อโดยตรงกับบอร์ดอิเล็กทรอนิกส์ เป็นต้น ซึ่งคอนโทรล MSCComm ที่มากับ Visual Basic จะเป็นคอนโทรลที่ทำงานโดยมีการตอบสนองต่อเหตุการณ์แบบ Event-Driven นั่นก็คือคอนโทรลจะทำหน้าที่ตรวจสอบการเกิดขึ้นหรือการร้องขอให้เกิดเหตุการณ์ต่าง ๆ กับพอร์ตอนุกรมโดยอัตโนมัติ และจะมีการแจ้งเตือนให้ผู้อ่านได้ทราบโดยผ่านทางโพสิซีเยอร์เหตุการณ์ เช่นเดียวกับคอนโทรลทั่วไปของ VBA ดังนั้นในการเขียนโค้ดผู้อ่านจึงไม่จำเป็นต้องสร้างโพสิซีเยอร์ที่ทำหน้าที่คอยตรวจสอบเหตุการณ์ต่าง ๆ ของพอร์ตอนุกรม ซึ่งจะทำได้ง่ายต่อการใช้งานเป็นอย่างมาก

คอนโทรล MSCComm จะมีหน้าที่มาตรฐานหลัก ๆ สำหรับการสื่อสารผ่านพอร์ตอนุกรม 3 ประการ ดังต่อไปนี้

หมุนหมายเลขติดต่อกับโทรศัพท์ปลายทางที่กำหนด
ตรวจสอบการเข้ามาของข้อมูลยังพอร์ตอนุกรมโดยอัตโนมัติ
ส่งข้อมูลตามที่กำหนดจากโปรแกรมไปยังพอร์ตอนุกรม

ในความเป็นจริงคอนโทรล MSCComm ไม่ได้ทำหน้าที่ติดต่อกับพอร์ตอนุกรมโดยตรง แต่มันจะทำหน้าที่เรียกใช้ฟังก์ชันวินโดวส์ API ซึ่งวินโดวส์จะทำการส่งหรือรับข้อมูลผ่านทางพอร์ตอนุกรมโดยอาศัยไดรเวอร์ Comm.drv อีกทอดหนึ่ง ดังนั้นจึงสามารถสรุปสั้น ๆ ได้ว่าทุกครั้งที่มีผู้อ่านมีการเรียกใช้คอนโทรล MSCComm ก็หมายถึงการเรียกใช้ฟังก์ชันวินโดวส์ API ซึ่งจะถูกตีความอีกทอดหนึ่งโดยไดรเวอร์ Comm.drv จากนั้นก็จะส่งผ่านข้อมูลที่ถูกรูปแบบตามมาตรฐานการสื่อสาร (ทั้งนี้ก็ขึ้นกับอุปกรณ์ที่ต่อเข้ากับพอร์ตอนุกรม) ให้กับดีไวซ์ไดรเวอร์อีกทอดหนึ่งนั่นเองสำหรับฟอร์มหนึ่ง ๆ ผู้อ่านสามารถเพิ่มได้หลาย ๆ คอนโทรล MSCComm ทั้งนี้ขึ้นกับความต้องการของผู้อ่านในการติดต่อกับพอร์ตอนุกรมใดบ้าง สำหรับวินโดวส์ 95 และ NT 4.0 ผู้อ่านสามารถที่จะติดตั้งพอร์ตอนุกรมได้มากกว่า 4 พอร์ต โดยเมื่อผู้อ่านเพิ่มคอนโทรล MSCComm ลงในฟอร์ม ซึ่งจะสนับสนุนคุณสมบัติ และโพสิซีเยอร์เหตุการณ์ที่สำคัญ ดังต่อไปนี้

- CommEvent

รายงานเหตุการณ์ทุกครั้งที่เกิดข้อผิดพลาดหรือมีการสื่อสาร ซึ่งผู้อ่านสามารถอ่านค่าได้เฉพาะในขณะที่ทำงานเท่านั้น

รูปแบบการใช้งาน

object.CommEvent

คอนโทรล MSCComm จะมีการเรียกโพสิซีเยอร์เหตุการณ์ OnComm ทุกครั้งที่เกิดข้อผิดพลาดหรือมีการสื่อสารเกิดขึ้น ซึ่งค่าตัวเลขที่จำนวนเต็ม แสดงถึงข้อผิดพลาดหรือเหตุการณ์ที่มีการสื่อสารดังกล่าว ก็จะถูกจัดเก็บเอาไว้ที่คุณสมบัติ CommEvent เสมอ ดังนั้นถ้าหากผู้อ่านต้องการตรวจสอบข้อผิดพลาดหรือเหตุการณ์ที่มีการสื่อสารภายในโพสิซีเยอร์เหตุการณ์ OnComm ก็ควรที่จะใช้ค่าตัวเลขจากคุณ

สมบัติ CommEvent ในการตรวจสอบเสมอ ซึ่งค่าตัวเลขที่รายงานโดยคุณสมบัติ CommEvent มีดังต่อไปนี้

ข้อผิดพลาด (Error)

ค่าคงที่	ค่าตัวเลข	รายละเอียด
comEventBreak	1001	ได้รับสัญญาณการหยุด (Break Signal)
comEventCTSTO	1002	สายสัญญาณ Clear To Send อยู่ในสถานะ Low(Timeout)ในขณะที่จะพยายามส่งออกตัวอักษร
ComEventDSRTO	1003	สายสัญญาณ Data Set Ready จะอยู่ในสถานะ Low(Timeout)ในขณะที่จะพยายามส่งออกตัวอักษร
ComEventFrame	1004	เฟรม ของข้อมูลไม่ถูกต้อง ซึ่งถูกตรวจพบโดยฮาร์ดแวร์
ComEventOverrun	1006	เกิด Port Overrun หมายถึง มีการรับตัวอักษรตัวใหม่เข้ามาในขณะที่ตัวอักษรก่อนหน้านี้ยังไม่ถูกอ่านจากฮาร์ดแวร์ ดังนั้นจึงเกิดการสูญหายของตัวอักษร
comEventCDTO	1007	สายสัญญาณ Carrier Detect อยู่ในสถานะ Low (Timeout) ในขณะที่พยายามจะส่งออกตัวอักษร
comEventRxOver	1008	Receive Buffer Overflow หมายถึง ขนาดของบัฟเฟอร์ด้านรับเข้าข้อมูล(Receive Buffer) ไม่เพียงพอกับขนาดของข้อมูลที่รับเข้ามา
comEventRxParity	1009	Parity Error ซึ่งถูกตรวจพบโดยฮาร์ดแวร์
comEventTxFull	1010	Transmit Bffer Full หมายถึง บัฟเฟอร์ ด้านส่งออกข้อมูล (Transmit Buffer) เต็ม ในขณะที่พยายามจัดเก็บข้อมูลใหม่ลงในบัฟเฟอร์
comEventDCB	1011	Unexpected Error หมายถึง เกิดข้อ ผิดพลาด ที่ไม่ได้ถูกนิยามเอาไว้ในขณะที่อ่าน Device Control Block (DCB)จากพอร์ตอนุกรมเหตุการณ์ (Event)
comEvSend	1	มีจำนวนตัวอักษร ในบัฟเฟอร์ด้านส่งออกข้อมูล น้อยกว่าจำนวนตัวอักษรที่กำหนดในคุณสมบัติ Sthreshold
comEvReceive	2	การรับเข้าจำนวนตัวอักษรที่ถูกกำหนดในคุณสมบัติ Rthreshold ซึ่งจะเกิดขึ้นอย่างต่อเนื่องจนกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อผู้อ่านจะใช้คุณสมบัตินี้ Input ในการอ่านข้อมูลด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาเอกสารนี้เพื่อใช้ในการรับเข้าข้อมูลทุกครั้งที่มีการนำไปใช้

comEvCTS	3	มีการเปลี่ยนสถานะของสายสัญญาณ Clear To Send
comEvDSR	4	มีการเปลี่ยนสถานะของสายสัญญาณ Data Set Ready ซึ่งเหตุการณ์นี้จะเกิดขึ้นเมื่อมีการเปลี่ยนสถานะของสายสัญญาณ Data Set Ready จาก 1 เป็น 0 เท่านั้น
comEvCD	5	มีการเปลี่ยนสถานะของสายสัญญาณ Carrier Detect
comEvRing	6	มีการตรวจพบการเรียกหมายเลข (สัญญาณเสียงกริ่ง) ซึ่ง UART บางตัวอาจจะไม่สนับสนุนคุณสมบัตินี้
comEvEOF	7	มีการรับตัวอักษรรหัสสุดท้ายสิ้นสุดของไฟล์ (EOF, ASCII 26)

- CommPort

รายงานหรือกำหนดหมายเลขของพอร์ตอนุกรมของเครื่องคอมพิวเตอร์ที่ต้องการติดต่อรูปแบบการใช้งาน

```
object.CommPort [=Value]
```

Value หมายถึง ข้อมูลชนิดเลขจำนวนเต็ม ที่กำหนดหมายเลขของพอร์ตอนุกรม

สำหรับหมายเลขของพอร์ตอนุกรมสามารถมีค่าได้ตั้งแต่ 1 ถึง 16 (ค่าปกติจะเท่ากับ 1)ซึ่งก่อนที่ผู้อ่านจะเปิดพอร์ตด้วยคุณสมบัติ PortOpen ผู้อ่านต้องกำหนดหมายเลขของพอร์ตอนุกรมให้กับคุณสมบัติ CommPort เสียก่อน โดยถ้าหากหมายเลขของพอร์ตอนุกรมที่กำหนดให้กับคุณสมบัติ CommPort ไม่เป็นความจริง ก็จะเกิดข้อผิดพลาด 68 (Device Unavailable) ทันที ซึ่งในกรณีนี้ผู้อ่านสามารถแก้ไขได้โดยการกำหนดหมายเลขของพอร์ตอนุกรมที่ถูกต้องเสียใหม่ แล้วจึงทำการเปิดพอร์ตอนุกรมอีกครั้งด้วยคุณสมบัติ PortOpen

- Handshaking

รายงานหรือกำหนดการใช้โปรโตคอลการตอบรับการติดต่อสื่อสารฮาร์ดแวร์ (Hardware Handshaking Protocol)

รูปแบบการใช้งาน

```
object..Handshaking [=Value]
```

Value หมายถึงข้อมูลชนิดเลขจำนวนเต็มที่กำหนดโปรโตคอลการตอบรับการติดต่อสื่อสารดังต่อไปนี้

ค่าคงที่	ค่าตัวเลข	รายละเอียด
----------	-----------	------------

เอกสารนี้เป็นเอกสาร comNone ไว้สำหรับการใช้งานเพื่อการไม่มีการตอบรับการติดต่อสื่อสาร (Default) บนด้านการค้าไม่ว่ากรณีใดๆ comXOnXOff มีให้ดัดแปลงเนื้อหาการตอบรับการติดต่อสื่อสารแบบ XON/XOFF นำไปใช้

comRTS	2	การตอบรับการติดต่อสื่อสารแบบ RTS/CTS (Request To Send/Clear To Send)
comRTSXOnXOff	3	การตอบรับการติดต่อสื่อสารแบบทั้ง Request To Send และ XON/XOFF

ในด้านการสื่อสารด้วยโมเด็ม Handshaking หมายถึงโปรโตคอลการสื่อสารภายในที่ซอฟต์แวร์ใช้ในการส่งข้อมูลจากพอร์ตอนุกรมของเครื่องคอมพิวเตอร์ไปยังบัพเฟอร์ด้านรับเข้าข้อมูล โดยทุกครั้งที่มีการรับข้อมูลมายังพอร์ตอนุกรมของเครื่องคอมพิวเตอร์ มันก็จะถูกจัดส่งต่อมายังบัพเฟอร์ด้านรับเข้าข้อมูลทันที ทั้งนี้เพื่อให้ซอฟต์แวร์สามารถอ่านข้อมูลดังกล่าวได้ต่อไป ซึ่งโปรโตคอลการสื่อสารภายในดังกล่าวจะเป็นการช่วยในการตรวจสอบ เพื่อป้องกันการสูญหายของข้อมูล เมื่อเกิดปัญหาขนาดของบัพเฟอร์ไม่เพียงพอกับจำนวนข้อมูลที่ส่งมา (Buffer Overrun)

- Input

รายงานพร้อมทั้งทำการลบข้อมูลในบัพเฟอร์ด้านรับเข้าที่ ซึ่งผู้อ่านสามารถอ่านค่าได้เฉพาะในขณะที่ทำงานเท่านั้น

รูปแบบการใช้งาน

```
object.Input
```

ทุกครั้งที่มีการใช้คุณสมบัติ Input ในการอ่านข้อมูลจากบัพเฟอร์ด้านรับเข้านั้น จำนวนของตัวอักษรที่อ่านได้จะถูกกำหนดลงในคุณสมบัติ InputLen ทั้งนี้ ซึ่งถ้าหากผู้อ่านกำหนดให้คุณสมบัติ InputLen มีค่าเท่ากับ 0 ก็จะมีผลถึงการกำหนดให้คุณสมบัติ Input อ่านข้อมูลทั้งหมดจากบัพเฟอร์ด้านรับเข้านั่นเอง ซึ่งชนิดของข้อมูลที่อ่านโดยคุณสมบัติ Input จะเป็นข้อมูลแบบข้อความหรือไบนารีก็ขึ้นกับการกำหนดค่าของคุณสมบัติ InputMode ดังจะกล่าวในรายละเอียดต่อไป

- InputLen

รายงานหรือกำหนดจำนวนของตัวอักษรที่อ่านโดยคุณสมบัติ Input จากบัพเฟอร์ด้านรับเข้าซึ่งมีหน่วยเป็น ไบต์

รูปแบบการใช้งาน

```
object.InputLen [=Value]
```

Value หมายถึงข้อมูลชนิดเลขจำนวนเต็มที่กำหนดจำนวนของตัวอักษรที่อ่านโดยคุณสมบัติ

Input

โดยปกติผู้เขียนมักจะกำหนดให้คุณสมบัติ InputLen มีค่าเท่ากับ 0 เพื่อให้คุณสมบัติ Input มีการอ่านตัวอักษรทั้งหมดจากบัพเฟอร์ด้านรับเข้าทั้งหมด แต่ถ้าหากผู้เขียนติดต่อกับบอร์ดควบคุมต่าง ๆ (Single Control Board) ก็จะกำหนดค่าของคุณสมบัติ InputLen ให้เท่ากับขนาดของเฟรมข้อมูลที่จะมีการรับ - ส่ง ในแต่ละครั้งซึ่งจะมีค่าคงที่

- Output

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของบริษัทฯ ขอสงวนสิทธิ์ในเนื้อหาและข้อมูลผู้จัดทำไว้เพื่อใช้ในการทำงานเท่านั้น ไม่ควรนำออกเผยแพร่โดยไม่ได้รับอนุญาต หากต้องการนำออกเผยแพร่ กรุณาแจ้งบริษัทฯ เพื่อขออนุญาตก่อน

ทำการส่งข้อมูลไปยังบัพเฟอร์ด้านส่งออก ซึ่งผู้อ่านสามารถกำหนดค่าได้เฉพาะในขณะที่ทำงานเท่านั้น

ไม่ว่าการแก้ไขใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบการทำงาน

object.Output [= Value]

Value หมายถึงข้อมูลชนิดสตริงหรือชนิด Variant ที่ต้องการส่งไปยังบัพเฟอร์ด้านส่งออก สำหรับชนิดของข้อมูลที่ถูกส่งโดยคุณสมบัติ Output จะเป็นข้อมูลแบบข้อความหรือไบนารี ก็ขึ้นกับการกำหนดค่าของคุณสมบัติ InputMode ดังที่กล่าวมาแล้วข้างต้น

- PortOpen

กำหนดสถานะการเปิด (Open) หรือปิด (Close) ของพอร์ตอนุกรมของเครื่องคอมพิวเตอร์
รูปแบบการใช้งาน

object.PortOpen [=Boolean]

Boolean หมายถึง ข้อมูลชนิดบิตที่กำหนดสถานะของพอร์ตอนุกรมดังต่อไปนี้

True หมายถึง พอร์ตอนุกรมถูกเปิด

False หมายถึง พอร์ตอนุกรมถูกปิด (Default)

พอร์ตอนุกรมถูกปิดโดยอัตโนมัติโดยคอนโทรล MSComm เมื่อแอปพลิเคชันสิ้นสุดการทำงาน โดยถ้าหากหมายเลขของพอร์ตอนุกรมที่กำหนดให้เปิดไม่มีการติดตั้งอยู่จริง ก็จะมีข้อผิดพลาดหมายเลข 68 (Device Unavailable) ทั้งนี้

- Rthreshold

รายงาน หรือกำหนดจำนวนตัวอักษรที่จะรับเข้าก่อนที่จะคอนโทรล MSComm จะกำหนดให้คุณสมบัติ CommEvent มีค่าเท่ากับ comEvReceive และมีการเรียกโพรซีเจอร์เหตุการณ์ OnComm

รูปแบบการใช้งาน

object.Rthreshold [=Value]

Value หมายถึง ข้อมูลชนิดเลขจำนวนเต็ม ที่กำหนดจำนวนตัวอักษรที่จะรับเข้าก่อนที่จะคอนโทรล MSComm ตามรายละเอียดข้างต้น

ถ้าหากคุณสมบัติ Rthreshold มีค่าเท่ากับ 0 (Default) ก็จะเป็นการยกเลิกการเรียกโพรซีเจอร์เหตุการณ์ OnComm เมื่อมีการรับเข้าตัวอักษรมายังบัพเฟอร์ด้านรับเข้า และในทางกลับกันถ้าหากคุณสมบัติ Rthreshold มีค่าเท่ากับ 1 ก็จะมีการเรียกโพรซีเจอร์เหตุการณ์ OnComm ทุกครั้งที่มีการรับเข้าตัวอักษรมายังบัพเฟอร์ด้านรับเข้า

- RTSEnable

กำหนดให้มีการใช้งานสาย Request To Send (RTS) ซึ่งโดยปกติสัญญาณ Request To Send จะถูกส่งจากเครื่องคอมพิวเตอร์ไปยังโมเด็ม เพื่อเป็นการแจ้งขอส่งข้อมูลจากคอมพิวเตอร์
รูปแบบการใช้งาน

object.RTSEnable [=Boolean]

Boolean หมายถึง ข้อมูลชนิดบิตที่กำหนดให้มีการใช้งานสาย Request To Send ดังต่อไปนี้
True หมายถึง มีการใช้งานสายสัญญาณ RTS (Request To Send) ที่มีการนำไปใช้

False หมายถึง ไม่มีการใช้งานสายสัญญาณ RTS (Default)

ถ้าหากคุณสมบัติ RTSEnable มีค่าเท่ากับ True สายสัญญาณ Request To Send จะอยู่ในสถานะ High เมื่อพอร์ตอนุกรมถูกเปิด และอยู่ในสถานะ Low เมื่อพอร์ตอนุกรมถูกปิด โดยปกติสายสัญญาณ Request To Send จะถูกใช้ในการตอบรับการติดต่อ

สื่อสารแบบฮาร์ดแวร์ RTS/CTS เท่านั้น

- Settings

รายงานหรือกำหนดพารามิเตอร์ในการสื่อสารผ่านพอร์ตอนุกรม

รูปแบบการใช้งาน

object.Settings [= Value]

Value หมายถึง ข้อมูลชนิดสตริงที่กำหนดพารามิเตอร์ในการสื่อสารผ่านพอร์ตอนุกรม ดังต่อไปนี้

รูปแบบของการกำหนดลำดับของพารามิเตอร์ในการสื่อสารผ่านพอร์ตอนุกรม สำหรับคุณสมบัติ Settings จะต้องเรียงลำดับดังนี้ “BBBB,P,D,S” (โดยปกติทั่ว ๆ ไปจะมีค่าเท่ากับ “9600,N,8,1”) เพราะถ้าหากลำดับไม่ถูกต้องก็จะทำให้เกิดข้อผิดพลาดหมายเลข 380 (Invalid Property Value) ทั้งนี้ซึ่งสัญลักษณ์แต่ละตัวมีความหมายดังนี้

- BBBB

ความเร็วของการส่งถ่ายข้อมูลในหน่วยของ Baud Rate ซึ่งในทางปฏิบัติ 1 Baud Rate อาจจะมีค่าเท่ากับ 1 BPS (Bits Per Second) หรือมากกว่าก็ได้ สำหรับค่าของBaud Rate ที่คอนโทรล MCommm สามารถรับได้มีค่าดังต่อไปนี้ 110, 300, 600, 1200, 2400, 9600 (Default), 14400, 19200, 28800, 38400 (Reserved), 56000, (Reserved), 128000, 256000 (Reserved)

- P

บิตพาริตี (Parity Bit) สำหรับใช้ในการตรวจสอบความถูกต้องของข้อมูล ซึ่งสามารถมีค่าได้ดังต่อไปนี้

พารามิเตอร์	ความหมาย
E	Even
M	Mark
N	None (Default)
O	Odd
S	Space

- D

ขนาดของบิตข้อมูล ซึ่งสามารถมีค่าได้ดังต่อไปนี้ 4, 5, 6, 7 หรือ 8 (Default)

- S

ขนาดของบิตหยุด (Stop Bit) ซึ่งสามารถมีค่าได้ดังต่อไปนี้ 1 (Default), 1.5 หรือ 2

เอกสารนี้เป็นเอกสารที่บริษัทได้จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่สามารถนำข้อมูลไปใช้ในการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Sthreshold

รายงานหรือกำหนดจำนวนตัวอักษรที่น้อยที่สุดที่ถูกจัดเก็บในบัฟเฟอร์ด้านส่งออก ก่อนที่คอนโทรล MSComm จะกำหนดให้คุณสมบัติ CommEvent มีค่าเท่ากับ comEvSend และมีการเรียกเหตุการณ์ OnComm

รูปแบบการใช้งาน

object.Sthreshold [= Value]

Value หมายถึงข้อมูลชนิดตัวเลขจำนวนเต็ม ที่กำหนดจำนวนตัวอักษรที่น้อยที่สุดที่ถูกจัดเก็บในบัฟเฟอร์ด้านส่งออกตามรายละเอียดข้างต้น

ถ้าหากคุณสมบัติ Sthreshold มีค่าเท่ากับ 0 (Default) ก็จะเป็นการยกเลิกการเรียกโปรซีเยอร์เหตุการณ์ OnComm เมื่อมีการส่งออกตัวอักษรไปยังบัฟเฟอร์ด้านส่งออก และในทางกลับกันถ้าหากคุณสมบัติ Sthreshold มีค่าเท่ากับ 1 ก็จะมีการเรียกโปรซีเยอร์เหตุการณ์ OnComm เมื่อบัฟเฟอร์ด้านส่งออกว่าง

โดยถ้าหากจำนวนของตัวอักษรในบัฟเฟอร์ด้านส่งออกน้อยกว่าค่าตัวเลขที่กำหนด คุณสมบัติ CommEvent ก็จะมีค่าเท่ากับ comEvSend และพร้อมทั้งเกิดการเรียกโปรซีเยอร์เหตุการณ์ OnComm ทันที เช่น สมมติให้คุณสมบัติ Sthreshold มีค่าเท่ากับ 10 และถ้าหากจำนวนตัวอักษรในบัฟเฟอร์ด้านส่งออกลดลงจาก 10 เป็น 9 ก็จะเกิดเหตุการณ์ OnComm ทันที เป็นต้น

โปรซีเยอร์เหตุการณ์ OnComm

- OnComm

เกิดขึ้นเมื่อมีการเปลี่ยนแปลงค่าของคุณสมบัติ CommEvent ซึ่งเป็นการบอกถึงการเกิดข้อผิดพลาดหรือมีการสื่อสารเกิดขึ้นก็ได้

รูปแบบโปรซีเยอร์เหตุการณ์

Private Sub object_OnComm ()

โดยปกติเมื่อคอนโทรล MSComm มีการเรียกโปรซีเยอร์เหตุการณ์ OnComm เรามักจะมีการเขียนโค้ดภายในโปรซีเยอร์เหตุการณ์นี้ เพื่อทำการตรวจสอบค่าของคุณสมบัติ CommEvent ทั้งนี้เพื่อตรวจสอบสถานะของการสื่อสารหรือข้อผิดพลาดที่เกิดขึ้นนั่นเอง

2.14 การจัดการโครงสร้างฐานข้อมูล

2.14.1 อ็อบเจ็กต์ทางด้านฐานข้อมูล (Database Related Objects)

ในการทำงานหรือจัดการกับฐานข้อมูลโดยใช้อ็อบเจ็กต์ทางด้านฐานข้อมูลของวิซวลเบสิก มีอ็อบเจ็กต์ต่างๆ ที่เกี่ยวข้องที่ใช้หมายถึงองค์ประกอบต่างๆ ภายในฐานข้อมูล ซึ่งจะอธิบายในรายละเอียดเรียงตามลำดับของโครงสร้างฐานข้อมูลได้ดังนี้

1. อ็อบเจ็กต์ระบบจัดการฐานข้อมูล (DBEngine Object) คือ อ็อบเจ็กต์ที่หมายถึงระบบจัดเอกสารนี้การฐานข้อมูลของวิซวลเบสิก หรือที่เรียกว่า JET นั้น ซึ่งมีเพียงอ็อบเจ็กต์เดียวเท่านั้น แต่อ็อบเจ็กต์ทางด้านฐานข้อมูลอื่นๆ จะถูกกำหนดควบคุมผ่านทาง DBEngine นี้ทั้งหมดทุกได้จะมีคุณลักษณะ

(Property) เช่น เวอร์ชันของ JET (Version) ชื่อผู้ใช้ที่จะใช้ Login เข้าระบบจัดการฐานข้อมูล (Default User) เป็นต้น

ในอ็อบเจกต์ระบบจัดการฐานข้อมูล จะเก็บรายละเอียดหรือประกอบด้วยอ็อบเจกต์พื้นที่ทำงานกับระบบจัดการฐานข้อมูล (Workspace Object) ซึ่งหากมี Workspace Object มากกว่าหนึ่งอ็อบเจกต์ก็หมายความว่าโปรแกรมนั้น มีการติดต่อกับระบบจัดการฐานข้อมูล (JET) มากกว่าหนึ่งการเชื่อมต่อ (Connection)

2. อ็อบเจกต์พื้นที่ทำงานกับระบบจัดการฐานข้อมูล (Workspace Object) โปรแกรมของวิซวลเบสิกที่ต้องการทำงานกับฐานข้อมูล จะต้องทำงานโดยการเรียกใช้บริการจากอ็อบเจกต์ระบบจัดการฐานข้อมูล (DBEngine Object) ดังนั้นทุกๆ โปรแกรมดังกล่าวจะต้องมีการติดต่อกับระบบจัดการฐานข้อมูล ซึ่งในที่นี้วิซวลเบสิกจะเรียกว่าเป็นอ็อบเจกต์พื้นที่ทำงานกับระบบจัดการฐานข้อมูล หรือ (Workspace Object) อ็อบเจกต์นี้ทำหน้าที่เป็นตัวกลางเชื่อมต่อโปรแกรมวิซวลเบสิกกับอ็อบเจกต์ระบบจัดการฐานข้อมูล (DBEnging Object) ทุกโปรแกรมที่ทำงานกับฐานข้อมูลจะต้องมีอ็อบเจกต์พื้นที่ทำงานกับระบบจัดการฐานข้อมูล (Workspace Object) อย่างน้อยหนึ่งอ็อบเจกต์หรือมากกว่าหนึ่งอ็อบเจกต์ก็ได้ในแต่ละอ็อบเจกต์ชนิดนี้ จะประกอบด้วยอ็อบเจกต์ฐานข้อมูลที่โปรแกรมทำงานด้วยผ่านทางอ็อบเจกต์พื้นที่การทำงานกับระบบจัดการฐานข้อมูลนั้นๆ

3. อ็อบเจกต์ฐานข้อมูล (Database Object) คืออ็อบเจกต์ที่หมายถึงตัวฐานข้อมูลเอง โดยจะมีคุณลักษณะ (Property) เช่น ฐานข้อมูลชื่ออะไร (Name) อนุญาตให้แก้ไขฐานข้อมูลได้หรือไม่ (Updatable) วิธีการเรียงลำดับข้อมูลเป็นอย่างไร (Collating Order) เป็นต้น ทุกโปรแกรมที่ทำงานกับฐานข้อมูลจะต้องมีการกำหนดอ็อบเจกต์ฐานข้อมูล (Database Object) อย่างน้อยเท่ากับจำนวนฐานข้อมูลที่ใช้ในโปรแกรมนั้น

ในแต่ละอ็อบเจกต์ฐานข้อมูลจะประกอบด้วยอ็อบเจกต์ทางด้านโครงสร้างอื่นๆ ที่ประกอบกันเป็นฐานข้อมูลนั้น เช่น อ็อบเจกต์ข้อกำหนดตาราง (TableDef Object) อ็อบเจกต์ข้อกำหนดความสัมพันธ์ (Relation Object) และอ็อบเจกต์โปรแกรมย่อยในฐานข้อมูล (ซึ่งในวิซวลเบสิกเรียกว่า QueryDef) เป็นต้น

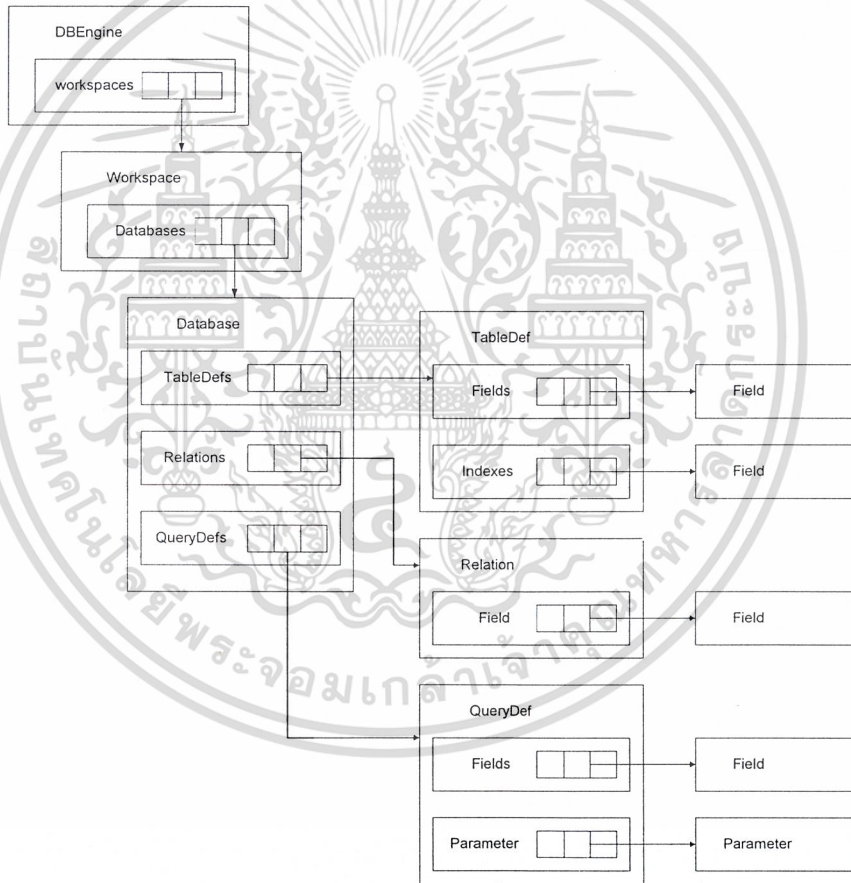
4. อ็อบเจกต์ข้อกำหนดตาราง (TableDef Object) คือ อ็อบเจกต์ที่หมายถึงโครงสร้างของตารางนั้น (ไม่ได้หมายถึงข้อมูลที่เก็บในตาราง) เช่น ประกอบด้วยฟิลด์อะไรบ้าง หรือมีดรรชนีอะไรบ้าง เป็นต้น ในแต่ละอ็อบเจกต์ข้อกำหนดตารางจะประกอบด้วยอ็อบเจกต์ที่แสดงถึงองค์ประกอบของตารางนั้นๆ เช่น อ็อบเจกต์ข้อกำหนดฟิลด์ (Field Object) อ็อบเจกต์ข้อกำหนดดรรชนี (Index Object)

5. อ็อบเจกต์ข้อกำหนดฟิลด์ (Field Object) อ็อบเจกต์นี้เป็นองค์ประกอบหนึ่งของอ็อบเจกต์ข้อกำหนดตาราง เพื่อใช้อธิบายถึงคุณลักษณะของฟิลด์ที่ประกอบกันเป็นตารางนั้น

6. อ็อบเจกต์ข้อกำหนดดรรชนี (Index Object) ตัวอ็อบเจกต์ข้อกำหนดดรรชนีก็เช่นเดียวกันกับอ็อบเจกต์ข้อกำหนดฟิลด์ ก็จะช่วยในการอธิบายให้ทราบว่าตารางนั้นมีดรรชนีในลักษณะใดประกอบอยู่ เอกสารนี้ด้วยเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. อ็อบเจ็กต์ข้อกำหนดความสัมพันธ์ (Relation Object) ฐานข้อมูลโดยทั่วไปมักจะมี ความสัมพันธ์ระหว่างตารางเป็นอีกองค์ประกอบหนึ่งที่สำคัญ อ็อบเจ็กต์ข้อกำหนดความสัมพันธ์นี้จะเป็นส่วนที่ใช้อธิบายความสัมพันธ์ต่าง ๆ ที่มีอยู่ในฐานข้อมูลนั้น ๆ จัดเป็นองค์ประกอบย่อยภายในอ็อบเจ็กต์ฐานข้อมูลเช่นเดียวกับอ็อบเจ็กต์ข้อกำหนดตาราง

8. อ็อบเจ็กต์โปรแกรมย่อยในฐานข้อมูล (QueryDef Object) โปรแกรมย่อยในฐานข้อมูลหรือที่เรียกว่า Stored Procedure ในระบบจัดการฐานข้อมูลโดยทั่วไป แต่สำหรับระบบจัดการฐานข้อมูลเจ็ด (JET) ที่ใช้ในวิซวลเบสิก และ MsAccess จะเรียกว่า QueryDef เป็นชุดคำสั่งทำงานจัดการฐานข้อมูล ที่ประกอบด้วยคำสั่ง SQL ต่าง ๆ โดยจะถูกสร้างและจัดเก็บรวมไว้กับฐานข้อมูล ประโยชน์การใช้โปรแกรมย่อยในฐานข้อมูลจะช่วยในเรื่องของความเร็ว ลดความซับซ้อนของโปรแกรมและช่วยให้โปรแกรมเป็นอิสระจากโครงสร้างของฐานข้อมูลมากขึ้น



รูปที่ 2.42 อ็อบเจ็กต์โปรแกรมย่อยในฐานข้อมูล

2.14.2 โครงสร้างฐานข้อมูลแบบเจ็ด (JET)

เอกสารนี้เป็นเอกสารที่สงวนการตรวจสอบถึงโครงสร้างและองค์ประกอบต่างๆ ในฐานข้อมูล จะทำได้โดยการท่องไม่ว่าการไปในอ็อบเจ็กต์ต่างๆ ที่เกี่ยวข้องกับฐานข้อมูล แล้วตรวจสอบค่าของคุณสมบัติในอ็อบเจ็กต์ต่างๆ นั้นนั่นเอง

โดยมีขั้นตอนต่าง ๆ ดังนี้

1. กำหนดวิธีการติดต่อกับระบบจัดการฐานข้อมูลและการเปิดฐานข้อมูล

การทำงานกับฐานข้อมูลทุก ๆ อย่างในวิซวลเบสิก จะถูกจัดการโดยระบบจัดการฐานข้อมูล หรือ JET ดังนั้นโปรแกรมที่ต้องการทำงานกับฐานข้อมูลจะต้องกำหนดรายละเอียดและวิธีการติดต่อกับระบบจัดการฐานข้อมูลเสียก่อน ซึ่งทำได้โดยการสร้างอ็อบเจ็กต์พื้นที่ทำงานกับระบบจัดการฐานข้อมูล (Workspace Object) โดยปกติแล้ววิซวลเบสิกจะสร้างอ็อบเจ็กต์พื้นที่ทำงานกับระบบจัดการฐานข้อมูลขึ้นมาให้โดยอัตโนมัติแล้วหนึ่งอ็อบเจ็กต์ ที่เป็นมาตรฐานสำหรับการทำงานกับฐานข้อมูลทั่วไปเราสามารถเรียกอ็อบเจ็กต์พื้นที่ทำงานกับระบบจัดการฐานข้อมูล ซึ่งเป็นมาตรฐานนี้ขึ้นมาใช้ได้โดยอ้างอิงถึงอ็อบเจ็กต์ลำดับแรก ในอ็อบเจ็กต์กลุ่มของอ็อบเจ็กต์พื้นที่ทำงานกับระบบจัดการฐานข้อมูล (Workspace Collection) ดังในตัวอย่างข้างล่างนี้

```
Dim wksDefault AS Workspace
Set wksDefault = DBEngine.Workspaces(0)
```

การเปิด (Open) ฐานข้อมูลที่ต้องการ หลังจากที่ได้อ็อบเจ็กต์พื้นที่ทำงานกับระบบจัดการฐานข้อมูลแล้ว ก่อนจะสามารถอ่านโครงสร้างหรือจัดการใด ๆ กับข้อมูลที่อยู่ภายในฐานข้อมูลนั้นได้ ฐานข้อมูลที่ต้องการนั้นจะต้องถูกเปิด (Open) ก่อน ในลักษณะเช่นเดียวกับการเปิดแฟ้มข้อมูลปกติทั่วไปก่อนที่จะทำการอ่านหรือเขียนแฟ้มข้อมูลนั้น

การเปิดฐานข้อมูลขึ้นมาเพื่อใช้งาน ทำได้โดยการเรียกฟังก์ชันการทำงาน Open Database จากอ็อบเจ็กต์พื้นที่ทำงานกับฐานข้อมูล (Workspace Object) แล้วกำหนดค่าที่ได้ให้กับตัวแปรอ็อบเจ็กต์ฐานข้อมูล โดยมีรูปของการเรียกใช้ดังนี้

```
Set db_object = wks_object.OpenDatabase(db_filename[,exclusive[,readonly[,connect[]])]
```

โดยที่

- db_object ตัวแปรประเภทอ็อบเจ็กต์ฐานข้อมูล
- db_filename เป็นตัวแปรหรือค่าคงที่ประเภทตัวอักษร(String) ที่แสดงชื่อแฟ้มข้อมูลของฐานข้อมูล การระบุชื่อแฟ้มข้อมูลทำในลักษณะเช่นเดียวกับการระบุถึงแฟ้มข้อมูลทั่วไป จะมีหรือไม่มีส่วนที่บอกว่าจะอยู่ที่ดิสก์ใดหรือ ไดรฟ์อะไรก็ได้

- exclusive ค่าของตัวแปรหรือค่าคงที่มีค่าเป็น True หรือ False เท่านั้น คือ ถ้ามีค่าเป็น True หมายความว่าต้องการจะใช้ฐานข้อมูลนี้เพียงคนเดียว แต่หากมีค่าเป็น False หรือละไว้ไม่กำหนด จะเป็นการใช้ฐานข้อมูลร่วมกับผู้อื่น

- readonly ค่าของตัวแปรหรือค่าคงที่มีค่าเป็น True หรือ False เท่านั้น เพื่อบอกว่าต้องการจะใช้ฐานข้อมูลนี้ในลักษณะอย่างไร จะเป็นการอ่านข้อมูลอย่างเดียว หรือว่าต้องการจะแก้ไขข้อมูลได้ด้วย ซึ่งถ้าเป็น False หรือไม่ได้ระบุ จะหมายถึงว่าต้องการจะใช้ฐานข้อมูลในลักษณะที่มีการแก้ไขข้อมูล

- connect เป็นตัวแปรหรือค่าคงที่ประเภทตัวอักษร จะใช้เมื่อต้องการจะเปิด ฐานข้อมูลภายนอกอื่น ๆ (External Database) ที่ไม่ใช่ Ms-Access เช่น SQL-Server หรือ Oracle เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในวงจำกัดเท่านั้น ไม่สามารถเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นที่เห็นเหตุเห็นผลโดยชัดแจ้ง และต้องขออนุญาตจากเจ้าของลิขสิทธิ์ก่อนนำไปใช้

2.14.3 อ็อบเจ็กต์ข้อมูล (Recordset) และการจัดการกับข้อมูล

การใช้วิซวลเบสิกจัดการกับข้อมูลในฐานข้อมูลทำได้ 2 วิธี คือ

วิธีที่ 1 ทำโดยการใช้คอนโทรลข้อมูล (Data Control) ซึ่งวิธีนี้มีข้อดีในเรื่องของความสะดวกและง่ายต่อการใช้งานเป็นอย่างมาก ช่วยให้สร้างโปรแกรมขึ้นมาได้อย่างรวดเร็ว แต่ก็มีข้อเสีย คือ ความยืดหยุ่นในการใช้งาน และวิธีการควบคุมการใช้งานทำได้ไม่ค่อยสะดวกนัก ไม่สามารถเรียกใช้ประโยชน์จากโครงสร้างของฐานข้อมูลได้มากนัก ทำให้การจัดการหรือค้นหาข้อมูลไม่มีประสิทธิภาพและยืดหยุ่นอย่างเพียงพอ

วิธีที่ 2 เป็นการจัดการกับข้อมูลโดยอาศัย อ็อบเจ็กต์ข้อมูล (Recordset Object) ด้วยวิธีการนี้ผู้พัฒนาจะจัดการข้อมูลได้อย่างมีประสิทธิภาพ และมีความยืดหยุ่นต่อการใช้งานในลักษณะต่าง ๆ มากกว่าการใช้คอนโทรลข้อมูล (Data Control) สำหรับวิธีการนี้จะมีเฉพาะในวิซวลเบสิกชุดโปรแกรมพัฒนาเว็บและอินเทอร์เน็ตเท่านั้น จะไม่มีในเวอร์ชันมาตรฐาน

2.14.3.1 อ็อบเจ็กต์ข้อมูล (Recordset Object)

อ็อบเจ็กต์ข้อมูล หมายถึง กลุ่มของข้อมูลในฐานข้อมูลที่สนใจหรือต้องการจะใช้งานอ็อบเจ็กต์ข้อมูล จะแบ่งออกเป็นประเภทที่แตกต่างกันได้ 3 ประเภทหลัก ๆ แต่ละประเภทก็จะเหมาะสมกับการใช้งานในลักษณะที่ต่างกัน ซึ่งพอจะสรุปได้ดังนี้

1. อ็อบเจ็กต์ข้อมูลประเภทตาราง (Table Type Recordset Object) อ็อบเจ็กต์ข้อมูลประเภทตารางนี้จะเก็บข้อมูลทั้งหมดทุก ๆ เรคอร์ด (Record) และทุก ๆ ฟิลด์ (Field) ของตารางที่สนใจการกำหนดข้อมูลให้กับอ็อบเจ็กต์ข้อมูลประเภทตารางนี้ ทำได้โดยการระบุชื่อตารางที่ต้องการในขณะที่มีการกำหนดสร้างอ็อบเจ็กต์นี้ขึ้นมา

2. อ็อบเจ็กต์ข้อมูลประเภทไดนาเซต (Dynaset Type Recordset Object) เป็นอ็อบเจ็กต์ข้อมูลที่สามารถมีข้อมูลทั้งหมดจากตาราง ข้อมูลบางส่วนของตาราง หรือข้อมูลจากตารางหลายตารางรวมกันก็ได้ และนำเสนอให้ผู้ใช้งานเข้าใจเสมือนว่าเป็นอีกตารางหนึ่ง คำว่า ไดนาเซต (Dynaset) ในวิซวลเบสิกจะมีลักษณะเช่นเดียวกัน View ที่พบในระบบจัดการฐานข้อมูลขนาดใหญ่ทั่วไป วิธีการกำหนดข้อมูลให้กับอ็อบเจ็กต์ข้อมูลประเภทนี้ ทำได้โดยการใช้คำสั่ง Select ของ SQL ในขณะที่มีการกำหนดโครงสร้างอ็อบเจ็กต์นี้ขึ้นมา เพื่อเลือกเอาข้อมูลที่มีลักษณะที่ต้องการจากตาราง (Table) ตารางหนึ่ง หรือเลือกข้อมูลจากหลายตารางมารวมกัน(จากการเชื่อม Join ตารางต่าง ๆ เข้าด้วยกัน) ซึ่งจะได้อธิบายในรายละเอียดอีกครั้งในวิธีการสร้างอีกทีหนึ่ง

2. อ็อบเจ็กต์ข้อมูลประเภทสแน็ปช็อต (Snapshot Type Recordset Object) สแน็ปช็อตจะมีลักษณะเหมือนกับอ็อบเจ็กต์ข้อมูลประเภทไดนาเซตทุกประการทั้งในเรื่องของการสร้างและการนำไปใช้งาน ยกเว้นแต่เพียงว่าสามารถอ่านหรือค้นหาข้อมูลที่อยู่ในอ็อบเจ็กต์ข้อมูลประเภทสแน็ปช็อตได้เพียงอย่างเดียวเท่านั้น ไม่สามารถเพิ่มเติม ลบหรือแก้ไขได้ และข้อมูลในอ็อบเจ็กต์ประเภทนี้เมื่อมีการกำหนดสร้างขึ้นมาแล้วจะ ไม่มีการเปลี่ยนแปลง ไม่ว่าข้อมูลในฐานข้อมูลได้ถูกเปลี่ยนแปลงโดยโปรแกรมหรือ

ผู้ใช้รายอื่น ๆ แล้วก็ตาม สำหรับวิธีการกำหนดสร้างก็ทำได้ในลักษณะเดียวกันกับการกำหนดสร้างอ็อบเจ็กต์ข้อมูลประเภทไดนาเจ็ท เพียงแต่ระบุประเภทที่แตกต่างกันเท่านั้น

ขั้นตอนและวิธีการต่าง ๆ ที่จะใช้งานอ็อบเจ็กต์ข้อมูลมีดังนี้

1. กำหนดตัวแปร

- อ็อบเจ็กต์ข้อมูล (Recordset Object) จัดประเภทของตัวแปร (Variable) หนึ่ง มีวิธีการกำหนดและการอ้างอิงเหมือนกับตัวแปรชนิดอื่น ๆ ดังในตัวอย่างข้างล่างนี้

```
Dim recordset_object As Recordset
```

โดยที่

recordset_object เป็นชื่อของตัวแปรที่ต้องการกำหนดให้มีประเภทเป็นอ็อบเจ็กต์ข้อมูล

2. กำหนดค่าให้กับตัวแปรอ็อบเจ็กต์ข้อมูล

ก่อนที่จะใช้ข้อมูลที่อยู่ในอ็อบเจ็กต์ข้อมูลได้ จะต้องมีกำหนดค่าให้กับตัวแปรอ็อบเจ็กต์ข้อมูลนั้นเสียก่อน เพื่อเป็นการบอกว่าอ็อบเจ็กต์ข้อมูลที่ต้องการ จะประกอบไปด้วยข้อมูลหรือเรคคอร์ดใดบ้าง และเป็นอ็อบเจ็กต์ข้อมูลประเภทใด เป็นต้น

สำหรับวิธีการกำหนดค่าให้กับตัวแปรอ็อบเจ็กต์ข้อมูล ทำโดยการเรียกใช้ฟังก์ชันการทำงาน (Method) ชื่อ OpenRecordset บนตัวแปรอ็อบเจ็กต์ฐานข้อมูลที่ได้เปิดไว้แล้ว โดยมีรูปแบบและวิธีการเรียกใช้ ดังต่อไปนี้

```
Set recordset_object = db_object.OpenRecordset (source [,type[,options  
[,lockedits]])
```

โดยที่

recordset_object เป็นตัวแปรประเภทอ็อบเจ็กต์ข้อมูลที่ต้องการจะกำหนดค่าให้

db_object เป็นตัวแปรประเภทอ็อบเจ็กต์ฐานข้อมูลที่ได้ถูกกำหนดค่าหรือเปิด (Open) ไว้แล้ว

source เป็นตัวแปรหรือค่าคงที่ที่เป็นตัวอักษร เพื่อใช้ในการระบุถึงแหล่งของข้อมูลสำหรับกำหนดให้อ็อบเจ็กต์ข้อมูลนี้มีค่าเป็นได้ 3 กรณีคือดังนี้

1. เป็นชื่อของตาราง ในกรณีที่ต้องการจะให้อ็อบเจ็กต์ข้อมูลนี้จะกำหนดค่านี้หมายถึงข้อมูลทั้งหมดในตารางนั้น ๆ
2. เป็นคำสั่ง Select ของ SQL ในกรณีที่ต้องการจะให้อ็อบเจ็กต์ข้อมูลนี้มีข้อมูลตามผลลัพธ์ที่ได้จากคำสั่ง Select นี้
3. เป็นชื่อของโปรแกรมย่อยในฐานข้อมูล (QueryDef) ในฐานข้อมูลที่มีการกำหนดคำสั่ง Select เก็บไว้ในฐานข้อมูลแล้ว ซึ่งจะได้อธิบายในรายละเอียดอีกครั้งภายหลัง

เอกสารนี้เป็นเอกสาร typevar เป็นค่าคงที่ที่ใช้ในการระบุถึงประเภทของอ็อบเจ็กต์ข้อมูลที่ต้องการจะโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งกำหนดขึ้นด้วยรายละเอียดดังนี้ ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าคงที่	ความหมาย
DbOpenTable	ระบุเมื่อต้องการจะกำหนดสร้างอ็อบเจ็กต์ชื่อประเภทตาราง
DbOpenDynaset	ระบุเมื่อต้องการจะกำหนดสร้างอ็อบเจ็กต์ข้อมูลประเภทไดนาเซต
DbOpenSnapshot	ระบุเมื่อต้องการจะกำหนดสร้างอ็อบเจ็กต์ข้อมูลประเภทสแน็บช็อต
DbOpenDynamic	สำหรับประเภทนี้จะมีลักษณะเช่นเดียวกับอ็อบเจ็กต์ข้อมูลประเภทไดนาเซต แต่แตกต่างกันที่จำนวนของข้อมูลที่อยู่ในอ็อบเจ็กต์จะมีการเปลี่ยนแปลงได้เสมอ จะใช้ได้กับอ็อบเจ็กต์พื้นที่ทำงานกับระบบจัดการฐานข้อมูลโดยตรง (ODBC Direct Workspace) เท่านั้น
DbOpenForwardOnly	อ็อบเจ็กต์ข้อมูลประเภทนี้จะมีลักษณะเช่นเดียวกับอ็อบเจ็กต์ข้อมูลประเภทสแน็บช็อต แตกต่างตรงที่อ็อบเจ็กต์ประเภทนี้จะอ่านเรคอร์ดถัดไปได้เท่านั้น ไม่สามารถจะอ่านข้อมูลในเรคอร์ดก่อนหน้าเรคอร์ดปัจจุบันไม่ได้

ตารางที่ 2.15 ค่าคงที่ใช้ในการระบุถึงประเภทอ็อบเจ็กต์ข้อมูล

2.14.3.2 การเคลื่อนตำแหน่งตัวชี้ตำแหน่งข้อมูล (Navigation)

การควบคุมตำแหน่งของตัวชี้ตำแหน่งข้อมูล ทำโดยการเรียกใช้ฟังก์ชันการทำงาน (Method) ของอ็อบเจ็กต์ข้อมูลที่เกี่ยวข้องกับการเคลื่อนตำแหน่งตัวชี้ตำแหน่งข้อมูล พร้อมทั้งทำการตรวจสอบค่าคุณสมบัติบางตัวที่เกี่ยวข้องคุณสมบัติ (Property) ของอ็อบเจ็กต์ข้อมูลที่เกี่ยวข้องกับตัวชี้ตำแหน่งข้อมูล คุณสมบัติของอ็อบเจ็กต์ข้อมูลที่สำคัญและเกี่ยวข้องกับตัวชี้ตำแหน่งข้อมูล มีเพียง

2 คุณสมบัติที่สำคัญคือ

1. BOF เป็นค่าของคุณสมบัติ ซึ่งถ้ามีค่าเป็น True หมายความว่า ตอนนี้ตัวชี้ตำแหน่งของข้อมูลของอ็อบเจ็กต์นี้อยู่ก่อนหน้าเรคอร์ดแรกของอ็อบเจ็กต์ข้อมูลนี้ (Begin Of File) หมายความว่า ไม่ได้ชี้ข้อมูลที่ตำแหน่งของข้อมูลที่ถูกต้อง

2. EOF มีลักษณะเช่นเดียวกับ BOF หากมีค่าเป็น True หมายความว่า ตำแหน่งของตัวชี้ข้อมูลจะอยู่เลยถัดไปจากเรคอร์ดสุดท้ายในอ็อบเจ็กต์ข้อมูลนี้อยู่แล้ว (End Of File)

ฟังก์ชันการทำงานสำหรับการเคลื่อนตำแหน่งตัวชี้ตำแหน่งข้อมูล มีวิธีการเคลื่อนตำแหน่งตัวชี้เอกสารนี้ ข้อมูล มีอยู่ 4 ฟังก์ชัน ดังนี้ การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน	คำอธิบายการทำงาน	คุณลักษณะที่เกี่ยวข้อง
MoveFirst	เป็นการเลื่อนตัวชี้ตำแหน่งในอ็อปเจ็ทข้อมูลนั้น ให้ไปที่ตำแหน่งของเรคอร์ดแรก	EOF,BOF
MoveNext	เพื่อเคลื่อนตัวชี้ตำแหน่งไปยังเรคอร์ดถัดไปจากตำแหน่งปัจจุบัน	จะมีค่าเป็น False ทั้งคู่
MovePrevious	เพื่อเคลื่อนตัวชี้ตำแหน่งไปยังเรคอร์ดก่อนหน้าตำแหน่งปัจจุบัน	EOF: หากเรคอร์ดปัจจุบันก่อนทำงาน เป็นเรคอร์ดสุดท้ายในอ็อปเจ็ทข้อมูล ค่าของ EOF นี้จะเป็น True EOF: หากเป็นการ เคลื่อนถอยหลังไป จากเรคอร์ดแรกสุด BOF จะเป็น True
MoveLast	เมื่อเคลื่อนไปยังตำแหน่งเรคอร์ดสุดท้ายในอ็อปเจ็ทข้อมูลนั้น	EOF,BOF จะมีค่าเป็น False ทั้งคู่
Move n	เคลื่อนไปยังตำแหน่งเรคอร์ดถัดไปจากตำแหน่งปัจจุบันนี้ไปอีก n เรคอร์ด หาก n เป็นบวก หมายถึงจะเคลื่อนไปทางตำแหน่งสุดท้าย ถ้า n เป็นลบ จะเคลื่อนไปทางตำแหน่งแรก	EOF และ BOF จะมีค่าในลักษณะเดียวกันกับการ MoveNext และ MovePrevious แล้วแต่ค่าของ n

ตารางที่ 2.16 วิธีการเคลื่อนตำแหน่งตัวชี้ข้อมูล

วิธีการเรียกใช้ฟังก์ชันการทำงานเคลื่อนตัวชี้ตำแหน่งข้อมูล มีรูปแบบ และ ตัวอย่างดังนี้

recordset_object.MoveFirst หรือ

recordset_object.MoveNext หรือ

recordset_object.MovePrevious หรือ

recordset_object.MoveLast หรือ

recordset_object.Move N

หมายเหตุ : หากมีการเรียกฟังก์ชันการทำงาน MoveNext ในขณะที่คุณสมบัติ EOF

มีค่าเป็น True อยู่แล้ว (ตำแหน่งของตัวชี้ตำแหน่งข้อมูลอยู่ถัดจากเรคอร์ดสุดท้ายในอ็อปเจ็ทข้อมูลแล้ว)

จะเกิดข้อผิดพลาดในขณะที่ทำงานขึ้น (Run-Time Error) ทำนองเดียวกันหากมีการเรียกฟังก์ชันการ

เอกสารนี้ใช้กับ MovePrevious ในขณะที่คุณสมบัติ BOF มีค่าเป็น True ก็จะมีข้อผิดพลาดขณะทำงานได้

ไม่ต่างกันทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.14.3.3 การอ่านและกำหนดค่าของฟิลด์ในเรคอร์ด

เมื่อระบุตำแหน่งของเรคอร์ดที่ต้องการ ได้แล้วจะสามารถทำการอ่านค่าของฟิลด์ต่าง ๆ ในเรคอร์ดที่ตำแหน่งนั้นหรือทำการเปลี่ยนแปลงแก้ไขค่าของฟิลด์ในตำแหน่งนั้นให้มีค่าตามที่ต้องการได้ (ในกรณีที่ต้องการจะเปลี่ยนแปลงค่าของข้อมูลในตำแหน่งนั้นของอ็อบเจ็กต์ของข้อมูล)

- การอ้างอิงค่าในฟิลด์และการอ่านค่า

วิธีที่ง่ายต่อการเข้าใจ สำหรับการอ้างอิงค่าในฟิลด์ทำได้ด้วยการอ้างอิงชื่อฟิลด์ที่ต้องการ ทำได้ตามรูปแบบดังนี้

```
recordset_object.Fields(field_name).Value
```

โดยที่

recordset_object เป็นตัวแปรประเภทอ็อบเจ็กต์ข้อมูล
field_name เป็นตัวแปรหรือค่าคงที่ประเภทตัวอักษรที่หมายถึงชื่อฟิลด์ที่ต้องการ

อีกวิธีจะเป็นการระบุถึงลำดับของฟิลด์ในอ็อบเจ็กต์ข้อมูลนั้นแทนที่จะเป็นชื่อของฟิลด์ ด้วยรูปแบบ

```
recordset_object.Fields(field_ordinal_number).Value
```

โดยที่

recordset_object เป็นตัวแปรประเภทอ็อบเจ็กต์ข้อมูล
field_ordinal_name เป็นตัวแปรหรือค่าคงที่ที่เป็นตัวเลข ใช้ระบุลำดับของฟิลด์ที่ต้องการในเรคอร์ดนั้น โดยฟิลด์แรกจะมีลำดับที่เป็น 0

- วิธีกำหนดค่าให้กับฟิลด์

หากมองว่าฟิลด์ในอ็อบเจ็กต์ข้อมูล เป็นเหมือนกับตัวแปรใด ๆ ตัวหนึ่งแล้ว วิธีการกำหนดค่าจะสามารถทำได้ในลักษณะเดียวกับตัวแปรอื่น ๆ ต่างเพียงแต่วิธีการอ้างอิงฟิลด์ค่อนข้างจะยาวสักหน่อยเท่านั้น ดังรูปแบบ

```
recordset_object.Fields(field_name).Value = value
```

โดยที่

recordset_object เป็นตัวแปรประเภทอ็อบเจ็กต์ข้อมูล
field_name เป็นตัวแปรหรือค่าคงที่ประเภทตัวอักษรที่หมายถึงชื่อฟิลด์ที่ต้องการ
value เป็นตัวแปรหรือค่าคงที่ที่ต้องการจะกำหนดให้กับฟิลด์

- การจัดการค้นหาข้อมูลที่ต้องการ

วิธีการที่สะดวกกว่านี้ในการค้นหาข้อมูลในอ็อบเจ็กต์ที่ต้องการ คือ การเรียกใช้ฟังก์ชันการทำงานสำหรับการค้นหาข้อมูล ซึ่งมีอยู่ 5 ฟังก์ชัน พอสรุปได้พร้อมทั้งคุณสมบัติที่เกี่ยวข้อง ดังตาราง

ประกอบข้างล่างนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน	คำอธิบายการทำงาน	
FineFirst	การค้นหาข้อมูลที่มีลักษณะตามที่ต้องการ โดยเริ่มค้นหา จากเรคอร์ดแรกสุดไปยังเรคอร์ดสุดท้าย ในอ็อบเจ็กต์ข้อมูล หากพบจะทำการเคลื่อนตัวชี้ตำแหน่งข้อมูลไปยังตำแหน่งของเรคอร์ดที่มีข้อมูล จะอยู่ที่เดิมก่อนเริ่มค้นหา	NoMatch - หากพบเรคอร์ดที่มีข้อมูลในลักษณะที่กำหนด ค่าของคุณสมบัติจะมีค่าเป็น Fase
FineNext	มีวิธีการทำงานเช่นเดียวกันกับ FineFirst แต่จะเริ่มการค้นหาข้อมูลที่ต้องการจากเรคอร์ดในตำแหน่งถัดจากตัวชี้ตำแหน่งข้อมูลไปยังเรคอร์ดสุดท้าย ในอ็อบเจ็กต์ข้อมูล	เช่นเดียวกันกับ FineFirst
FinePrevious	เหมือนกับฟังก์ชัน FineFirst แต่เป็นการค้นหาในทิศทางย้อนกลับ คือเริ่มจากเรคอร์ดที่อยู่ในตำแหน่งก่อนหน้าตัวชี้ข้อมูลย้อนหลัง ไปยังเรคอร์ดแรกสุด ในอ็อบเจ็กต์ข้อมูล	เช่นเดียวกันกับ FineFirst
FineLast	เหมือนกับฟังก์ชัน FineFirst แต่ในทิศทางตรงกันข้าม คือ เริ่มจากเรคอร์ดสุดท้ายก่อน	เช่นเดียวกันกับ FineFirst
	ฟังก์ชันนี้จะใช้ได้กับเรคอร์ดเซตประเภทตารางเพียงอย่างเดียว โดยจะทำการค้นหาข้อมูลที่ต้องการจากดรรชนี (index) ที่กำหนดในคุณสมบัติ index วิธีการนี้จะค้นหาข้อมูลได้รวดเร็วกว่าการใช้ฟังก์ชัน Find ต่างๆ	Index: จะกำหนดชื่อดรรชนีที่จะใช้ในการค้นหาและต้องเป็นดรรชนีของตารางที่ใช้เท่านั้น NoMatch: เช่นเดียวกันกับ Fine ต่าง ๆ

ตารางที่ 2.17 ฟังก์ชันการทำงานสำหรับการค้นหาข้อมูล

1. การค้นหาด้วยฟังก์ชัน Find ในลักษณะต่าง ๆ

การค้นหาข้อมูลที่ต้องการด้วยฟังก์ชัน Find ต่าง ๆ สามารถจะใช้ได้กับอ็อบเจ็กต์ข้อมูล

ประเภทไคนาเซตและสแน็ปช็อต ไม่สามารถจะใช้กับอ็อบเจ็กต์ข้อมูลประเภทตารางได้

รูปแบบของคำสั่งที่ใช้กับฟังก์ชันการทำงานค้นหาข้อมูลต่าง ๆ มีดังนี้

เอกสารนี้โดยที่เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ที่recordset_object เป็นตัวแปรประเภทอ็อบเจ็กต์ข้อมูลประเภทไคนาเซตหรือสแน็ปช็อตนำไปใช้

criteria เป็นตัวแปรหรือค่าคงที่ประเภทตัวอักษรที่กำหนดเงื่อนไขของการค้นหาข้อมูล รูปแบบและการกำหนดค่าเงื่อนไขการค้นหา (criteria)

criteria เป็นตัวแปรหรือค่าคงที่ประเภทตัวอักษร (String) เพื่อกำหนดว่าต้องการจะค้นหาข้อมูล ในลักษณะอย่างไร วิธีการกำหนดค่าให้กับ criteria จะมีหลักการเหมือนกับการกำหนดค่าในส่วนของ WHERE ในคำสั่ง Select ของ SQL

เพียงแต่ไม่มีคำว่า Where ประกอบด้วยเท่านั้น รูปแบบมีดังนี้

ฟิลด์ที่มีค่าเป็นตัวเลข เช่น Integer,Long,Currency และอื่น ๆ

field_name [=, <>, >, <, >=, <=] value หรือ

field_name In (value 1,value 2, ...)

โดยที่

field_name เป็นชื่อของฟิลด์ที่ต้องการจะค้นหาข้อมูลที่อยู่ในฟิลด์นี้

value,value1,... เป็นค่าที่ต้องการจะค้นในฟิลด์ field_name

ฟิลด์ที่มีชนิดเป็นตัวอักษร เช่น Text

field_name [=, <>, >, <, >=, <=] value หรือ

field_name In (value 1, value 2,...)

field_name Like pattern_string

โดยที่

field_name เป็นชื่อของฟิลด์ที่ต้องการจะค้นหา

ข้อมูลที่อยู่ในฟิลด์นี้

value,value 1,... เป็นค่าที่ต้องการจะค้นหาในฟิลด์ field_name

pattern_string เป็นรูปแบบชุดตัวอักษร (Pattern) ที่ต้องการจะค้นหา

2. การค้นหาโดยใช้ฟังก์ชัน Seek

การค้นหาโดยใช้ฟังก์ชันการทำงาน Seek เป็นการค้นหาข้อมูลหรือเรคอร์ดที่ต้องการโดยการค้นหาจาก ครรชนี การค้นหาโดยใช้ฟังก์ชันการทำงาน Seek จะเร็วมาก แต่ข้อจำกัดของการใช้ Seek ก็คือจะ ใช้ได้กับอ็อบเจ็กต์ข้อมูลประเภทตารางเท่านั้น ไม่สามารถจะใช้กับอ็อบเจ็กต์ข้อมูลประเภทไดนามิคเซตหรือส เน็บช็อดได้ และมีวิธีการใช้ที่ค่อนข้างยุ่งยากคือ ต้องกำหนดว่าจะให้ค้นหาจากครรชนีตัวใดของตาราง นั้น ดังนั้นการระบุครรชนีที่ใช้สำหรับการ Seek ต้องใช้ครรชนีที่กำหนดจากฟิลด์ที่ต้องการจะค้นหาข้อมูล

รูปแบบของคำสั่งที่ใช้กับฟังก์ชันการทำงาน Seek มีดังนี้

recordset_object.Index = index_name

recordset_object.Seek comparison_operator,value

โดยที่

recordset_object คือตัวแปรอ็อบเจ็กต์ข้อมูลประเภทตาราง ถูกอนุญาตให้นำไปใช้ประโยชน์ด้านการคำนวณได้ทั้งนี้ indexname เป็นตัวแปรหรือค่าคงที่ชนิดตัวอักษร มีค่าเป็นชื่อของครรชนีที่จะใช้ในการค้นหา

comparison_coperator เป็นตัวแปรหรือค่าคงที่ชนิดตัวอักษร ระบุวิธีการเปรียบเทียบ จะกำหนดได้เพียง =, <, >, >=, <=

comparison_value เป็นตัวแปรที่มีชนิดตามชนิดของฟิลด์ ที่มีค่าที่ต้องการค้นหาข้อความระวางในการค้นหาข้อมูลโดยใช้ฟังก์ชัน Seek มีดังนี้

1. โดยปกติการ Seek จะเริ่มค้นหาจากเรคคอร์ดแรกตามครรชนที่ใช้ แต่หากเราได้ทำการกำหนด comparison_operator เป็น < หรือ <= จะเป็นการเริ่มหาเรคคอร์ดจากเรคคอร์ดสุดท้ายตามครรชนก่อนแล้วจึงย้อนกลับ

2. ผลของการค้นหาโดย Seek ว่าพบหรือไม่ จะแสดงที่ค่าของคุณสมบัติ NoMatch ของอ็อบเจ็กต์ข้อมูลที่ค้นหา เหมือนกับการค้นหาโดยใช้ฟังก์ชันต่าง ๆ ของ Find

- การเพิ่มข้อมูลใหม่

วิธีการเพิ่มข้อมูลใหม่เข้าไปในอ็อบเจ็กต์ข้อมูลนั้น จะต้องเรียกใช้ฟังก์ชันการทำงานกับอ็อบเจ็กต์ข้อมูลนั้น 2 ฟังก์ชันด้วยกัน คือ AddNew และ Update โดยมีรูปแบบของการเรียกใช้ดังนี้

recordset_object.AddNew ‘เป็นการสั่งให้เพิ่มเรคคอร์ดว่าง ๆ หนึ่งเรคคอร์ด

‘แล้วจึงตามด้วยคำสั่งสำหรับกำหนดค่าให้กับฟิลด์ขีต่าง ๆ ตามต้องการ

recordset_object.Fields(“FIELD_NAME”) = value

...

...

recordset_object.Update ‘เป็นการสั่งให้แก้ไขค่าในฐานข้อมูลตามที่กำหนด

โดยที่

recordset_object คือตัวแปรอ็อบเจ็กต์ข้อมูล

field_name เป็นชื่อของฟิลด์ที่ต้องการจะกำหนดค่า

value เป็นตัวแปรหรือค่าคงที่ที่ต้องการจะกำหนดให้กับฟิลด์

- การแก้ไขข้อมูลเดิม

มีลักษณะการใช้งานที่เหมือนกับการเพิ่มข้อมูลใหม่ คือต้องเรียกฟังก์ชันการทำงาน 2 ฟังก์ชันเหมือนกัน แต่จะต้องเคลื่อนตัวชี้ตำแหน่งข้อมูลของอ็อบเจ็กต์ข้อมูลไปยังตำแหน่งของเรคคอร์ดที่ต้องการจะแก้ไขเสียก่อนจะด้วยวิธีการเคลื่อนตัวชี้ตำแหน่ง (การ Move ต่าง ๆ) การค้นหา (การ Find ต่าง ๆ หรือ Seek) หรือการระบุตำแหน่งโดยกำหนดค่าให้กับคุณสมบัติ Bookmark โดยตรงก็ได้

รูปแบบ และตัวอย่างที่แสดงจะสมมติว่า ได้เคลื่อนตัวชี้ตำแหน่งมายังตำแหน่งของข้อมูลที่ต้องการแล้ว

recordset_object.Edit ‘ ประกาศว่าจะเริ่มแก้ไขข้อมูลตำแหน่งของตัวชี้ ‘กำหนดค่าให้กับฟิลด์ที่ต้องการจะเปลี่ยนค่า

recordset_object.Fields(“FIELD_NAME”) = value

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

recordset_object.Update ‘สั่งให้เปลี่ยนข้อมูลในฐานข้อมูลได้แล้ว

- การลบข้อมูล

เมื่อต้องการจะลบเรคอร์ดหนึ่งออกจากอ็อบเจ็กต์ข้อมูล ให้เรียกฟังก์ชันการทำงาน Delete บนอ็อบเจ็กต์ข้อมูลนั้น โดยที่จะต้องเคลื่อนตัวชี้ตำแหน่งข้อมูลไปยังเรคอร์ดที่ต้องการจะลบเสียก่อน รูปแบบของฟังก์ชันที่เรียกใช้ มีดังนี้

recordset_object.Delete

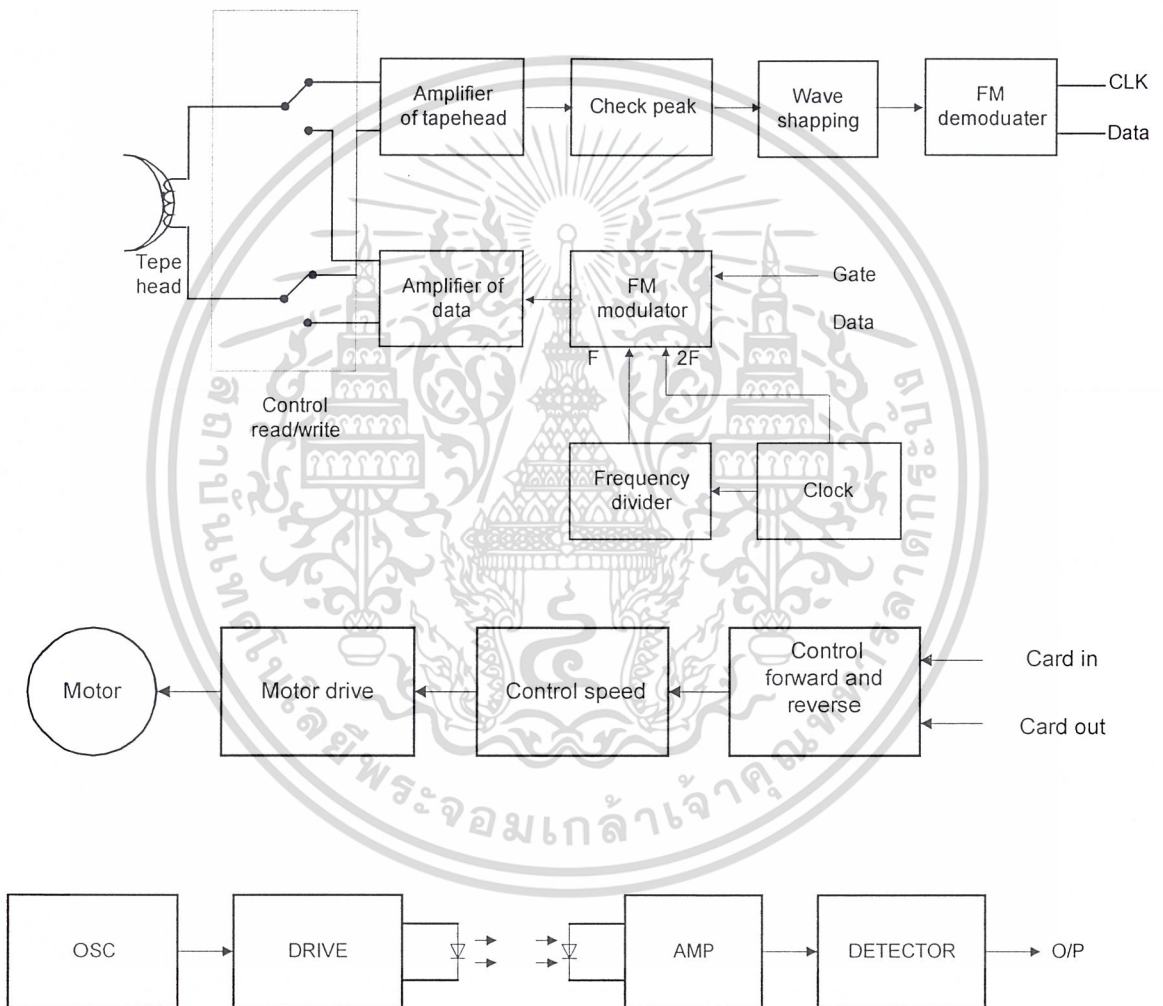


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การคำนวณและการสร้าง

โครงการนี้แบ่งออกได้เป็น 4 ส่วนใหญ่ๆ คือ ส่วนของการอ่านรหัสข้อมูลจากบัตร ส่วนของการบันทึกรหัสข้อมูลลงบัตร ส่วนของการควบคุมมอเตอร์ และส่วนของการตรวจจับบัตร ดังแสดงในรูปที่ 3.1



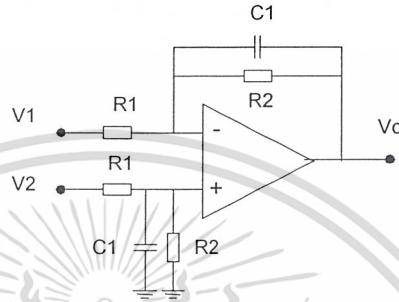
รูปที่ 3.1 บล็อก ไดอะแกรมของเครื่องอ่านและบันทึกข้อมูลของบัตรแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 ส่วนของการอ่านข้อมูลจากบัตรแม่เหล็ก

3.1.1 ภาขยายสัญญาณจากหัวเทป (Amplifier of tape head)

เมื่อหัวเทปทำการอ่านรหัสข้อมูลบนแถบแม่เหล็ก ซึ่งสัญญาณที่อ่านได้จะมีขนาดเป็นมิลลิโวลต์ จึงต้องขยายสัญญาณที่อ่านได้ให้มีขนาดเพียงพอในการที่จะนำไปถอดรหัสข้อมูลของบัตรได้ โดยในภาคนี้ได้ใช้ออปแอมป์ในการสร้างวงจรขยายสัญญาณเป็นลักษณะวงจรขยายความแตกต่าง (Differential Amplifier) เพื่อให้ได้ขนาดสัญญาณทั้งด้านบวกและลบ



รูปที่ 3.2 วงจรขยายความแตกต่าง

$$V(+)= [R2/(R1+R2)].V2$$

$$\{(V(-)-V1)/R1 + (V(-)-Vo)/R2\} = 0$$

$$Vo = (R2/R1).V(-) - (R2/R1).V1 + V(-)$$

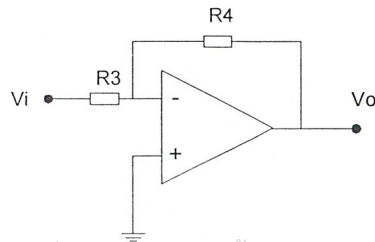
เมื่อ $V(+)= V(-)$

$$Vo = (R2/R1).[R2/(R1+R2)].V2 - (R2/R1).V1 + [R2/(R1+R2)].V2$$

ให้ $R2 > R1$ ดังนั้น $[R2/(R1+R2)] \approx 1$

ดังนั้น $Vo \approx -(R2/R1).[V1-V2]$

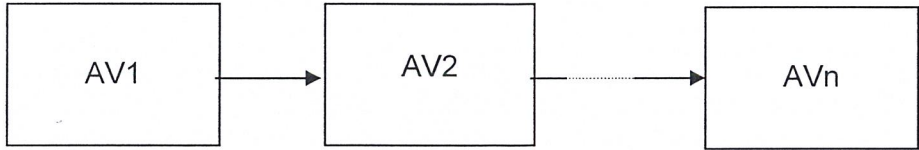
ในการออกแบบ เมื่อต้องการอัตราขยายเท่ากับ 33 เท่า โดยให้ $R2=33K$ ก็จะได้ $R1 = 1K$ ในส่วนของวงจรขยายความแตกต่างจะเป็นวงจรที่เกิดการขยายผลต่างที่มีการกลับเฟสของสัญญาณจึงสร้างวงจรขยายสัญญาณแบบกลับเฟส (Inverting Amplifier) เพื่อเป็นการเพิ่มอัตราขยาย และได้สัญญาณที่เป็นเฟสเดียวกับสัญญาณที่อ่านได้จากหัวเทป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 3.3 วงจรขยายสัญญาณแบบกลับเฟส
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.2 จะได้แรงดันเอาต์พุท $V_o = -(R_4/R_3) \cdot V_i$

ในการออกแบบ เมื่อต้องการอัตราขยายรวมประมาณ 1650 เท่า ที่ทำให้ได้สัญญาณขนาดแรงที่สุดที่ทำให้สัญญาณที่ขยายได้ไม่ผิดเพี้ยน



รูปที่ 3.4 อัตราขยายรวมของวงจรขยายสัญญาณแบบคาสเคด

ดังนั้น $A_{vt} = A_{v1} \cdot A_{v2} \dots A_{vn}$

ในการออกแบบ เมื่อต้องการออกแบบให้วงจรมีอัตราขยายรวมเท่ากับ 1650 เท่า โดยอัตราการขยายของวงจรที่ 1 เท่ากับ 33 เท่า

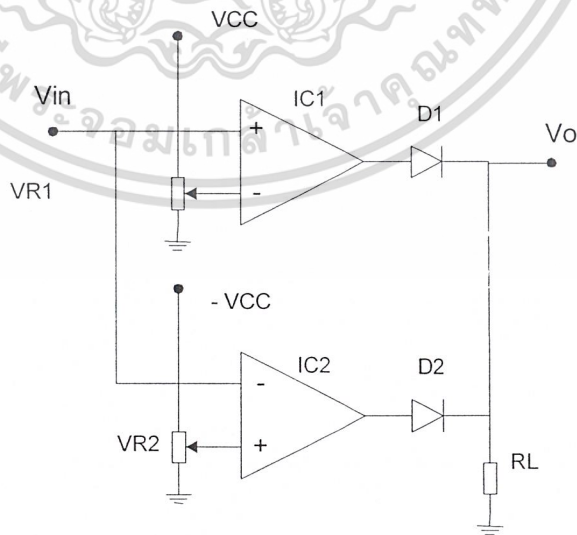
$$A_{v2} = A_{vt} / A_{v1}$$

$$= 1650 / 33 = 50$$

$A_{v2} = 50$

3.1.2 ภาควัดจับแรงดันสูงสุด (Peak detector)

เนื่องจากสัญญาณที่ได้รับขยายจากหัวเทป จะเป็นลักษณะสัญญาณที่มีระดับสัญญาณสูงสุดในช่วงบวกและลบ เมื่อข้อมูลซึ่งเป็นแบบดิจิตอลมีการเปลี่ยนแปลงสถานะของระดับสัญญาณซึ่งผลที่ได้จากส่วนนี้จะทำให้ได้ขบวนพัลส์ในช่วงระดับเกิดความแตกต่าง (Differential Pulse)

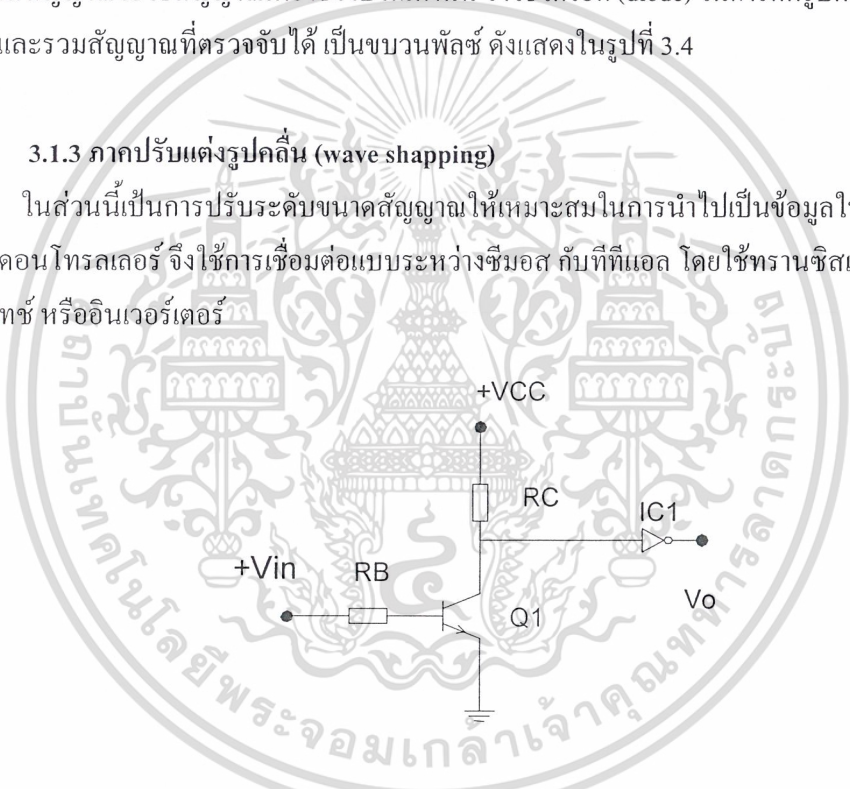


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 3.5 วงจรตรวจจับแรงดันสูงสุด
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากหลักการทำงานของออปแอมป์ ผลต่างของแรงดันอินพุต เท่ากับแรงดันที่ขาอินเวอร์สดีง
 ลบกับแรงดันที่ขาอินเวอร์สดีง $[V_d = V(+)- V(-)]$ โดยอัตราขยายของวงจรรูปเปิดมีค่าสูงมาก (ดูได้
 จากเอกสารข้อมูลออปแอมป์ที่ภาคผนวก) ทำให้ผลของสัญญาณเอาต์พุตที่ได้จะมีค่าประมาณเท่ากับแรง
 ดันที่จ่ายให้กับวงจร จากรูปที่ 3.4 เมื่อให้แรงดันอ้างอิงค่าบวกที่ขาอินเวอร์สดีงแอมป์ที่ IC1 ผลต่างของ
 แรงดันอินพุต จะมีค่าเป็นบวก เมื่อแรงดันอินพุตจะมีค่ามากกว่าแรงดันอ้างอิง ดังนั้น IC1 จึงเป็นวงจร
 เปรียบเทียบแรงดันสัญญาณอินพุตในช่วงสัญญาณเฟสบวก แบบไม่กลับเฟส เมื่อให้แรงดันอ้างอิงค่าลบที่
 ขาอินเวอร์สดีงแอมป์ของ IC2 ผลต่างของแรงดันอินพุตจะมีค่าบวก เมื่อแรงดันอินพุตมีค่าเป็นลบมาก
 กว่าแรงดันอ้างอิง ดังนั้น IC2 จึงเป็นวงจรเปรียบเทียบแรงดันสัญญาณอินพุตในช่วงสัญญาณเฟสลบ แบบ
 กลับเฟส สัญญาณที่ได้จากเอาต์พุตของ IC1 และ IC2 จะมีขนาดสัญญาณทั้งช่วงบวกและลบ ซึ่งในการ
 ถอดรหัสสัญญาณจะใช้สัญญาณเพียงช่วงบวกเท่านั้น จึงใช้ไดโอด (diode) ในการคัดรูปคลื่นสัญญาณช่วง
 ลบทิ้ง และรวมสัญญาณที่ตรวจจับได้ เป็นขบวนพัลส์ ดังแสดงในรูปที่ 3.4

3.1.3 ภาคปรับแต่งรูปคลื่น (wave shapping)

ในส่วนนี้เป็นการปรับระดับขนาดสัญญาณให้เหมาะสมในการนำไปเป็นข้อมูลให้กับ
 ไมโครคอนโทรลเลอร์ จึงใช้การเชื่อมต่อแบบระหว่างซิมอส กับทีทีแอล โดยใช้ทรานซิสเตอร์ ทำหน้าที่
 เป็นสวิตช์ หรืออินเวอร์เตอร์



รูปที่ 3.6 วงจรปรับแต่งรูปคลื่น

จากรูป 3.6 แสดงการนำทรานซิสเตอร์มาทำงานเป็นสวิตช์ (Switch)

LOOP INPUT :

$$-V_{in} + I_B \cdot R_B + V_{BE} = 0$$

$$I_B = (V_{in} - V_{BE}) / R_B$$

LOOP OUTPUT :

$$-V_{cc} + I_C \cdot R_C + V_{CE} = 0$$

$$I_C = (V_{cc} - V_{CE}) / R_C = \beta \cdot I_B$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการออกแบบให้ทรานซิสเตอร์ทำงานเป็นสวิตช์ ต้องให้กระแสคอลเลกเตอร์ ณ จุดทำงานมากกว่า กระแสคอลเลกเตอร์อิ่มตัว (เมื่อคิดให้แรงดันตกคร่อมระหว่างคอลเลกเตอร์-อิมิตเตอร์เท่ากับศูนย์) จะเห็นได้ว่าสัญญาณเอาต์พุตที่ได้จะมีลักษณะกลับเฟสกับสัญญาณข้อมูลที่เข้ามา จึงต้องผ่านการกลับเฟสสัญญาณอีกครั้งหนึ่ง โดยใช้อินเวอร์สเตอร์เกต (inverter gate)

3.1.4 ภาคการถอดรหัสข้อมูล (FM Demodulator)

จะเป็นส่วนการสร้างวงจรมอดูเลชัน เพื่อทำหน้าที่ควบคุมช่วงเวลาของข้อมูลแต่ละบิตให้ถูกต้อง โดยใช้ไอซีเบอร์ 555 ทำงานเป็นวงจรมอนอสเตเบิล ให้มีการหน่วงเวลามากกว่า ครึ่งหนึ่งของช่วงเวลาแต่ละบิตของข้อมูล จะทำให้สามารถถอดรหัสข้อมูลได้โดยอาศัยช่วงเวลาครึ่งหนึ่งของแต่ละบิตข้อมูล



รูปที่ 3.7 วงจรสร้างสัญญาณนาฬิกาในการถอดรหัส

จากรูปที่ 3.7 แสดงการนำวงจรมอนอสเตเบิลมาควบคุมการถอดรหัสข้อมูล โดยให้สัญญาณถูกหน่วงเวลาไปมากกว่าครึ่งหนึ่งของช่วงเวลา 1 บิตข้อมูล

$$T = RC \cdot \ln 3$$

$$\cong 1.1RC$$

เมื่อให้ $T > (1/2)T_b$

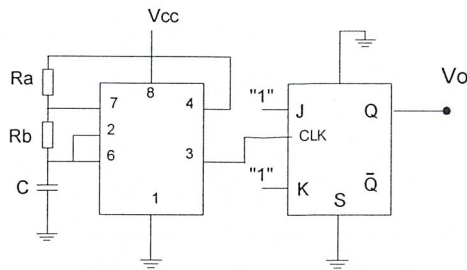
โดย T คือช่วงเวลาของสัญญาณที่ถูกหน่วงเวลา

T_b คือช่วงเวลา 1 บิตข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ส่วนของการบันทึกข้อมูลลงบัตร

3.2.1 ภาคกำเนิดสัญญาณนาฬิกา (Clock)



รูปที่ 3.8 วงจรกำเนิดสัญญาณนาฬิกา

เป็นส่วนผลิตความถี่ เพื่อใช้เป็นความถี่อ้างอิงในการบันทึกข้อมูลแบบดิจิทัล โดยใช้ไอซีเบอร์ 555 สร้างเป็นวงจระะสเทเบิล จากรูปที่ 3.8 แสดงวงจระะสเทเบิล มาทำงานเป็นวงจระะกำเนิดสัญญาณนาฬิกาสามารถคำนวณได้ดังนี้

$$t1 = Rb * C * \ln 2 \cong 0.693 * Rb * C$$

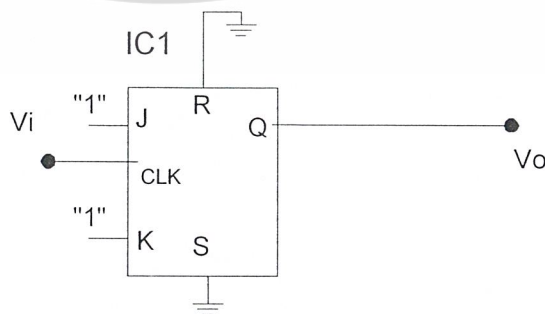
$$t2 = Rab * C * \ln 2 \cong 0.693 * (Ra + Rb) * C$$

$$T = t1 + t2 \cong 0.693 * (Ra + 2Rb) * C$$

จะเห็นว่าช่วงเวลาในแต่ละคาบเวลาจะไม่สามารถเท่ากันได้ ($t1 \neq t2$) ทำให้จะได้สัญญาณที่ไม่สมมาตรกัน ดังนั้นจึงใช้ ที ฟลิปฟลอป (T Flip Flop) เข้าช่วยในการปรับแต่งรูปคลื่นให้ได้สัญญาณที่สมมาตรกัน

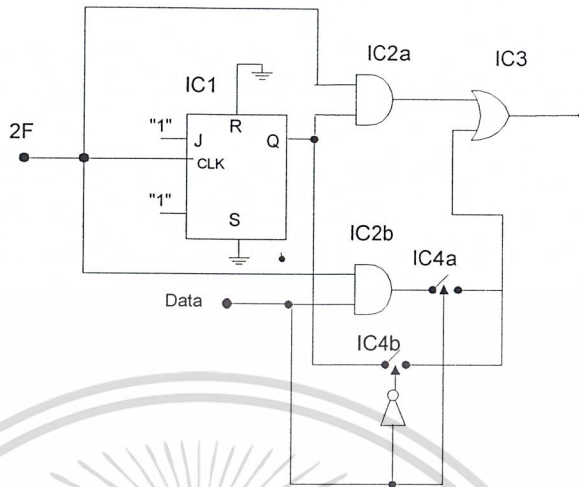
3.2.2 ภาคหารความถี่ (Frequency Divider)

เป็นส่วนที่ทำ การลดความถี่ลงครึ่งหนึ่งของความถี่หลัก โดยใช้วงจระะหาร 2 หรือ โมดูลัส 2 ซึ่งใช้ ที ฟลิปฟลอปในการสร้างดังรูปที่ 3.8



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานรูปที่ 3.9 วงจระะหารความถี่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 ภาคการบันทึกข้อมูล(Record data or FM modulation)



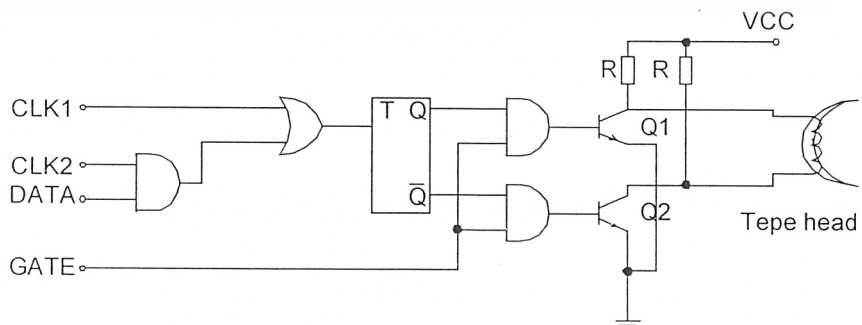
รูปที่ 3.10 วงจรบันทึกข้อมูล

จากรูปที่ 3.10 เมื่อข้อมูลที่ทำการบันทึกเป็นลอจิก 1 สัญญาณเอาต์พุตที่ IC2b จะเท่ากับลอจิก 1 ในขณะที่ IC4b ขาควควบคุมจะ ได้รับลอจิก 0 ดังนั้นสวิตช์จะเปิดวงจรในขณะที่ IC4a ขาควควบคุมจะ ได้รับลอจิก 1 ดังนั้นสวิตช์จะปิดวงจรทำให้ได้พัลส์ความถี่ 2F ออกที่ออร์เกท

เมื่อข้อมูลทำการบันทึกเป็นลอจิก 0 สัญญาณเอาต์พุตที่ IC2b จะเท่ากับลอจิก 0 ทำให้สวิตช์ของ IC4a เปิดวงจรสวิตช์ของ IC4b ปิดวงจรทำให้มีสัญญาณพัลส์ความถี่ F ออรักับพัลส์ช่วงเวลาบิทข้อมูล ได้เอาต์พุตเป็นพัลส์ความถี่ F

3.2.4 ภาคขยายสัญญาณข้อมูล(Amplifier of data)

เนื่องจากข้อมูลที่ทำการบันทึกเป็นลักษณะดิจิทัล จึงใช้การปรับระดับสัญญาณให้สูงขึ้นได้ เพื่อให้ออกสร้างกระแสป้อนให้กับขดลวดของหัวเทป สามารถบันทึกข้อมูลลงบนแถบแม่เหล็กได้



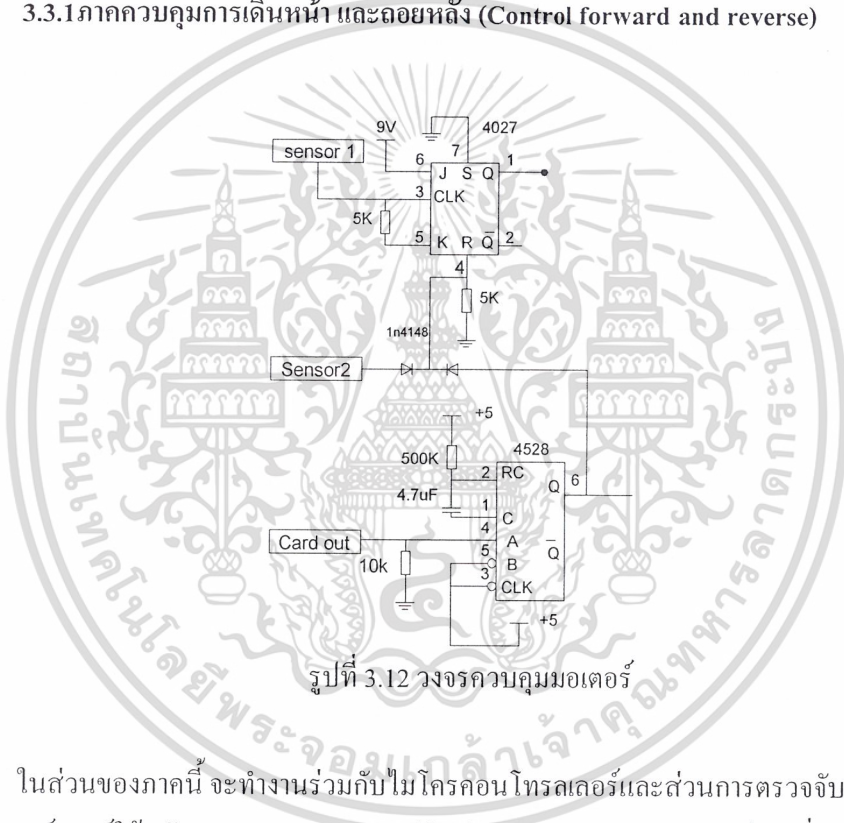
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดและแก้ไขเนื้อหาและข้อมูลของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.11 วงจรขยายสัญญาณบันทึก

จากรูปที่ 3.11 ฟลิปฟลอป (T Flip Flop) จะสร้างข้อมูลที่ใช้ในการป้อนกระแสพัลส์ด้านบวกและด้านลบ ให้กับขดลวดของหัวเทปซึ่งจะมีการควบคุมการบันทึกที่ขาเกต ถ้าขาเกตได้รับลอจิก 1 จึงจะสามารถทำการบันทึกได้ดังนั้นเอาท์พุท 1 (Q) เป็นลอจิก 1 ทำให้เอาท์พุทแอนด์เกต 1 เป็นลอจิก 1 ทรานซิสเตอร์ (Q1) จึงสามารถทำงานเป็นสวิตช์ได้ ขณะที่เอาท์พุท 2 เป็นลอจิก 0 ทำให้แอนด์เกต 2 เป็นลอจิก 0 ทรานซิสเตอร์ (Q2)ไม่ทำงาน ผลของกระแสที่เป็นเข้าขดลวดจึงไหลผ่านทิศทางจาก b ไป a เสมือนป้อนกระแสพัลส์ทางด้านลบเข้าขดลวด โดยถ้าเอาท์พุท 1 (a) เป็นลอจิก 0 ก็จะทำให้กระแสพัลส์ที่ป้อนเข้าขดลวดจะกลับทิศทางจาก a ไป b เป็นการป้อนพัลส์ทางด้านบวกเข้าขดลวด

3.3 ส่วนควบคุมมอเตอร์

3.3.1ภาคควบคุมการเดินหน้า และถอยหลัง (Control forward and reverse)



รูปที่ 3.12 วงจรควบคุมมอเตอร์

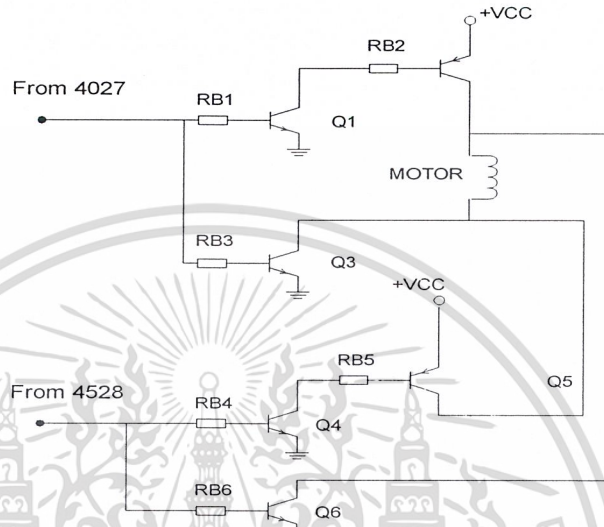
ในส่วนของภาคนี้ จะทำงานร่วมกับไมโครคอนโทรลเลอร์และส่วนการตรวจจับบัตรด้วย ในส่วนของฮาร์ดแวร์ใช้หลักการการทำงานของ เจค ฟลิปฟลอป(JK FlipFlop) ร่วมกับไอซีที่ทำหน้าที่เป็นโมโนสเตเบิล ดังแสดงในรูปที่ 3.12

จากรูปที่ 3.12 ในสภาวะไม่มีบัตรผ่าน ขา 3 ของ IC 4027 จะได้รับลอจิก 0 ทำให้เอาท์พุท Q เป็นลอจิก 0 แต่เมื่อส่วนตรวจจับบัตรได้ตรวจจับการผ่านเข้ามาของบัตร จะทำให้ขา 3 ของ IC 4027 ได้รับลอจิก 1 ทำให้ Q เป็นลอจิก 1 ตลอด ถ้าต้องการให้ Q กลับมาเป็นลอจิก 0 ต้องป้อน ลอจิก 1 ที่ขา 4 ของ IC #4027 และเมื่อต้องการนำบัตรออก ก็ใช้ไมโครคอนโทรลเลอร์ควบคุมส่งสัญญาณลอจิก 1 ไปยังขา 4 ของ IC # 4528 (a) ทำให้เอาท์พุท (Q) เป็นลอจิก 1 และไปควบคุมให้ชุดควบคุมส่วนบนมีเอาท์พุทเป็นลอจิก 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 ภาคขับมอเตอร์ (Control motor)

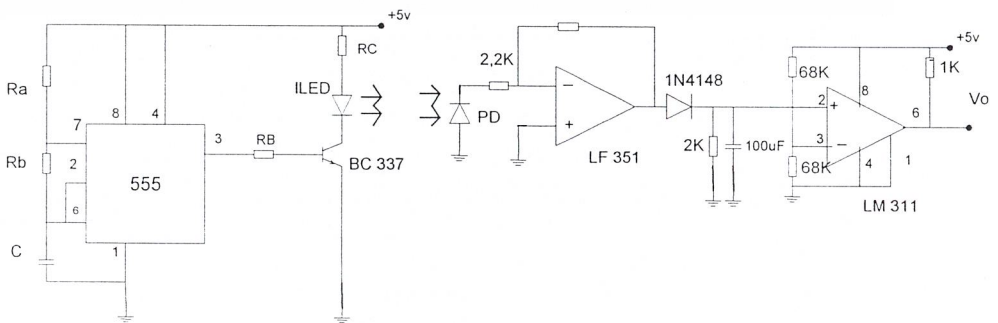
ใช้ทรานซิสเตอร์ในการเชื่อมต่อระหว่าง ไอซีดิจิตอลกับมอเตอร์ในส่วนของภาคนี้จะควบคุมความเร็วของมอเตอร์ด้วย เนื่องจากมอเตอร์จะมีความแปรผันกับค่าของกระแสที่ป้อนให้กับตัวมอเตอร์ ดังนั้นการออกแบบต้องคำนึงถึงกระแสที่ป้อนมอเตอร์ด้วย



รูปที่ 3.13 วงจรขับมอเตอร์

จากรูปที่ 3.12 ถ้าขา B ของชุดทรานซิสเตอร์ที่ได้รับสัญญาณจาก IC 4027 เป็น “1” โดยชุดที่ได้รับสัญญาณจาก IC 4528 เป็น “0” จะทำให้ Q1, Q2, Q3 ON กระแสจะไหลผ่านมอเตอร์จาก Q2 ลงผ่านไปยัง Q3 ทำให้มอเตอร์หมุนนำบัตรเคลื่อนที่เข้าผ่านหัวเทปเพื่ออ่านข้อมูล เมื่อทรานซิสเตอร์ชุดล่างขา B ได้รับ “1” โดยชุดบนขา B ได้รับ “0” จะทำให้ Q4, Q5, Q6 ON เป็นเหตุให้กระแสไหลผ่านจาก Q5 ไป Q6 ในลักษณะย้อนกลับทิศทางการเคลื่อนที่นำบัตรออก

3.4 ส่วนตรวจจับบัตร (Sensor)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงรูปที่ 3.14 วงจรตรวจจับบัตรฯ ของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 ภาคกำเนิดความถี่ (Oscillator)

จะใช้ผลิตสัญญาณพัลส์ที่มีความถี่ที่ต่อเนื่องเพื่อนำไปใช้ในการส่งสัญญาณผ่านแอลอีดีอินฟราเรด (LED Infrared)

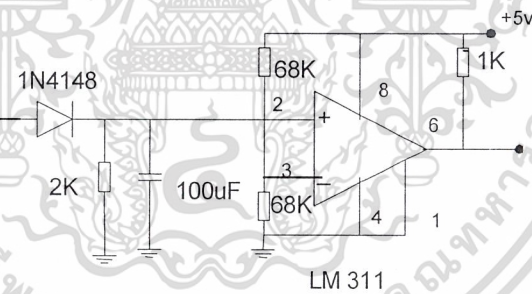
3.4.2 ภาคขับแอลอีดีอินฟราเรด (Drive infrared LED)

จะใช้ทรานซิสเตอร์ทำหน้าที่เชื่อมต่อควบคุมการทำงานให้แอลอีดีอินฟราเรดสามารถส่งข้อมูลเป็นไปตามสัญญาณที่เข้ามาทางอินพุตได้ โดยแอลอีดีอินฟราเรดจะส่งข้อมูลในลักษณะแสงไปยังส่วนของภาครับแสงอินฟราเรด

3.4.3 ภาคขยายสัญญาณ (Amplifier of PD)

ส่วนนี้จะใช้โฟโต้แอลอีดี (Photo LED) ในการรับแสงอินฟราเรดที่ส่งมาจากภาคส่งแล้วเปลี่ยนสัญญาณแสงเป็นกระแสไฟฟ้าที่มีความสัมพันธ์กับความเข้มแสงที่ได้รับ จากนั้นจึงนำสัญญาณกระแสไฟฟ้าที่มีขนาดต่ำ มาขยายเพื่อให้สามารถตรวจจับระดับสัญญาณได้

3.4.4 ภาคตรวจจับระดับสัญญาณ (Peak detector)



รูปที่ 3.15 วงจรตรวจจับระดับสัญญาณ

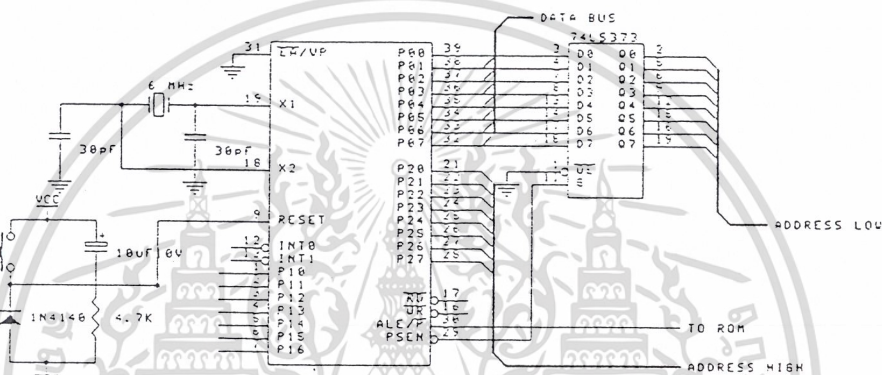
จากรูปที่ 3.14 ใช้ไอซีที่ทำหน้าที่ตรวจจับแรงดัน (voltage Comparator) ในการตรวจจับการเปลี่ยนแปลงของบิตที่ผ่านเข้า-ออก เครื่องอ่านและบันทึก โดยกรณีเมื่อบิตผ่านเข้าไปในเครื่อง ซึ่งจะผ่านระหว่างตัวรับกับตัวส่งแสงอินฟราเรด ดังนั้นบิตจะบ่งการเดินทางของแสงทำให้ระดับแรงดันของสัญญาณที่ขยายเมื่อเปรียบเทียบกับแรงดันอ้างอิง ได้เอาท์พุทระดับ “0”

กรณีไม่มีบิตผ่านระหว่างตัวรับ-ส่งแสงอินฟราเรด จำทำให้สัญญาณที่ขยายได้เอาท์พุทเป็นสัญญาณพัลส์ความถี่เดียวกับความถี่ที่ส่งมา จึงทำการกรองกระแส (Rectifier) เพื่อให้ได้สัญญาณเป็นระดับแรงดันไฟตรง ไปเปรียบเทียบกับแรงดันอ้างอิง ได้เอาท์พุทเป็น “1” จึงสรุปได้ว่า กรณีมีบิตผ่านสู่ระบบ จะได้เอาท์พุทเป็น “0” แต่ถ้ากรณีไม่มีบิตผ่านเข้าสู่ระบบจะได้เอาท์พุทเป็น “1” ที่มีการนำไปใช้

3.5 ส่วนควบคุม (Control)

3.5.1 ส่วนประมวลผลกลาง

จะใช้ไมโครคอนโทรลเลอร์ AT 89S8252 ซึ่งเป็นตระกูลเกี่ยวกับ MCS – 51 แต่มีคุณสมบัติแตกต่างไปคือมีหน่วยความจำสำหรับเก็บข้อมูลทั่วไปภายใน 256 ไบต์คือที่ตำแหน่ง 00H – FFH มีหน่วยความจำสำหรับเก็บโปรแกรมจึงไม่ต้องการต่อ EPROM ภายนอกเพิ่มเติมทำให้การต่อขา EA ของ CPU ต้องต่อ +VCC ส่วน CLOCK ของระบบจะต้องใช้ OSCILLATOR ที่มีความถี่ขนาด 11.059 MHZ การต่อขารีเซ็ตและขาอื่นๆจะกระทำเหมือนทั่วๆไปได้ตามรูปที่ 3.7 จะแสดงการต่อขารีเซ็ตและขา OSCILLATOR ของ AT89S8252 จากการต่อจะสังเกตว่าขา DATA BUS กับ ADDRESS BUS 8 ไบต์ค่าจะเป็นขาเดียวกันแต่แยกโดยขา ALE โดยจะใช้ IC เบอร์ 74LS373 เป็นตัว LATCH ค้าง ADDRESS ไว้



รูปที่ 3.16 การต่อรีเซ็ตและออสซิลเลเตอร์ของ AT89S8252

3.5.2 หน่วยความจำสำหรับเก็บข้อมูลภายนอก

ใช้ (RAM : Random Access Memmory) เบอร์ 27256 ขนาด 3 กิโลไบต์จำนวน 2 ตัว เพื่อที่จะสามารถอ่านและเขียนข้อมูลได้ถึง 64 กิโลไบต์แต่ในโครงการนี้จะกำหนด ADDRESS ให้กับ 8255 ไว้ด้วยจึงเสียพื้นที่ไป 8 กิโลไบต์ดังนั้นหน่วยความจำภายนอกจะมีพื้นที่ในการอ่านเขียนข้อมูลได้ 58 กิโลไบต์โดยได้กำหนดแผนผังพื้นที่หน่วยความจำ (Memory map) ไว้ดังนี้

- รหัสผ่านเข้าระบบ (System password) ขนาด 2 ไบต์
- ข้อมูลการอ่าน,บันทึกบัตร (ISO DATA) ขนาด 200 ไบต์ (5 * 40)
- ข้อมูลรหัสประจำตัวผู้ใช้บริการ (ID. BUFFER) ขนาด 5 ไบต์ : 1 คน
- รหัสผ่านส่วนบุคคล (Your password) ขนาด 3 ไบต์ : 1 คน
- ข้อมูลจำนวนสมาชิก (Member Buffer) ขนาด 3 ไบต์ : 1 คน
- ข้อมูลการใช้บริการ (User Buffer) จะมีการ เก็บข้อมูลการใช้บริการ 15 ไบต์ : 1 ครั้ง ให้ 1 วันสามารถใช้บริการได้ 10 ครั้ง : 1 คน

ดังนั้น เมื่อสามารถเก็บข้อมูลทั้งหมดได้ 58 กิโลไบต์ (59,392 ไบต์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในหน่วยงานเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งยังเป็นข้อมูลภายในของหน่วยงานซึ่งต้องรักษาไว้ถึงขั้นผู้ใช้บริการขอให้นำไปใช้
 เพราะฉะนั้นเครื่องเก็บข้อมูลนี้สามารถให้บริการกับผู้ใช้ได้ประมาณ 350 คน

E000H	MEMBER BUFFER
D920H	USER BUFFER
0C00H	YPASS
07E0H	ID BUFFER
0100H	ISO DATA
0008H	SPASS BUFFER
0004H	NOT USED
0000H	

รูปที่ 3.17 แผนผังหน่วยความจำภายนอกในการเก็บข้อมูล

3.5.3 พอร์ต 8255

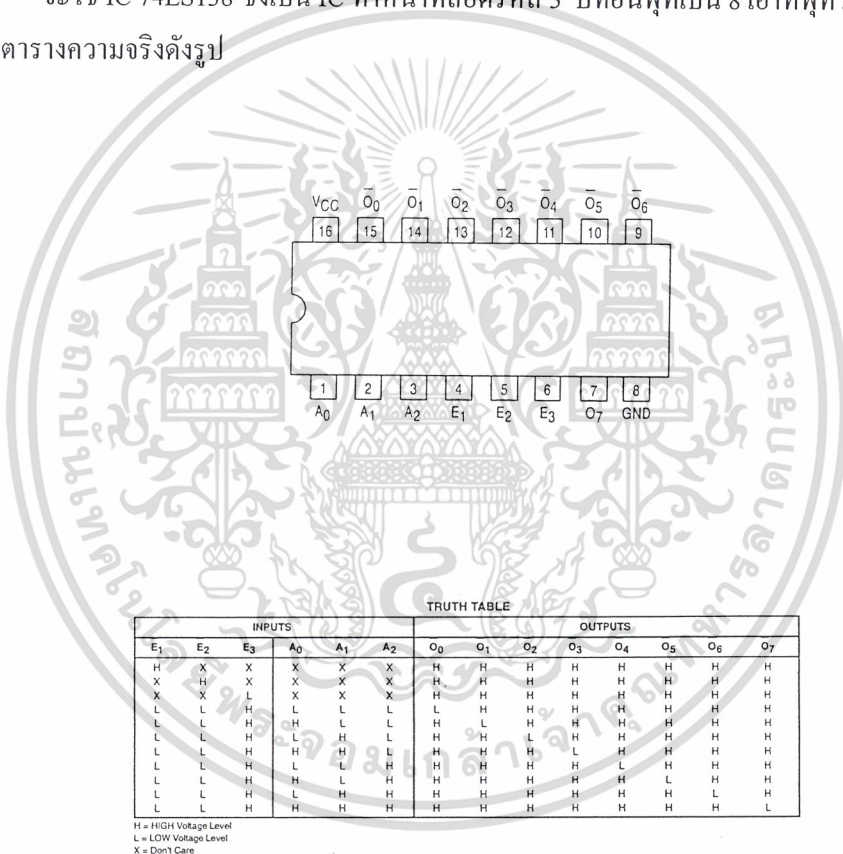
โครงการนี้ได้ใช้ 8255 จำนวน 2 ตัวเนื่องจากต้องติดต่อกับอุปกรณ์ภายนอกหลายประเภทซึ่งกำหนดแอดเดรสไว้ดังนี้คือ

- 8255 ตัวที่ 1 อยู่ที่แอดเดรส 0E000H – 0F000H โดยแต่ละพอร์ตมีหน้าที่ดังนี้
 - Port A0 - A7 เป็นบัสข้อมูล (DATA BUS) ของ LCD
 - Port B0 - B3 เป็นพอร์ตเอาต์พุตควบคุมการแสดงผลของ LCD
 - Port C0 – C3 เป็นพอร์ตเอาต์พุตของ Column Matrix Keyboard
 - Port C4 – C7 เป็นพอร์ตอินพุตของ Row Matrix Keyboard
- 8255 ตัวที่ 2 อยู่ที่แอดเดรส 0F00H – 0FFFH แต่ละพอร์ตมีหน้าที่ดังนี้
 - PA0 เป็นพอร์ตเอาต์พุต ควบคุมการหมุนบัตรเข้าของมอเตอร์
 - PA1 เป็นพอร์ตเอาต์พุต ควบคุมการหมุนบัตรออกของมอเตอร์
 - PA2 เป็นพอร์ตเอาต์พุต ควบคุมให้มอเตอร์หยุดหมุน
 - PA3 เป็นพอร์ตเอาต์พุต ควบคุมการบันทึกบัตร
 - PA4 เป็นพอร์ตเอาต์พุต ควบคุมการพิมพ์ของ Printer

- PBO เป็นพอร์ตอินพุท รับสัญญาณนาฬิกาอ้างอิง
- PB1 เป็นพอร์ตอินพุท รับสัญญาณการอ่านข้อมูลจากบัตร
- PB2 เป็นพอร์ตอินพุท รับสัญญาณการนำบัตรเข้า
- PB3 เป็นพอร์ตอินพุท รับสัญญาณการนำบัตรออก
- PB4 เป็นพอร์ตอินพุท รับสัญญาณความพร้อมของ Printer
- PB5 เป็นพอร์ตอินพุท รับสัญญาณการนับกระดาษจาก Xerox
- PC0 - PC7 เป็นพอร์ตเอาต์พุท ทำหน้าที่เป็น DATA BUS ของเครื่องพิมพ์

3.5.4 การเชื่อมโยง MCS – 51 กับหน่วยความจำ

จะใช้ IC 74LS138 ซึ่งเป็น IC ทำหน้าที่ถอดรหัส 3 บิตอินพุทเป็น 8 เอาต์พุทโดยมีการจัดวางขาและตารางความจริงดังรูป



รูปที่ 3.18 การจัดวางขาและตารางความจริงของ 74LS138

ดังนั้นในการออกแบบ จึงต้องเขียนตารางความจริงของแอดเดรสที่ต้องการถอดรหัสหรือไม่
โครคอนโทรลเลอร์ติดต่อกับแอดเดรสที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADDRESS	A15	A14	A13	A12	A0	OUTPUT
0000-1FFFH	0	0	0	X	X	Y0
2000-3FFFH	0	0	1	X	X	Y1
4000-5FFFH	0	1	0	X	X	Y2
6000-7FFFH	0	1	1	X	X	Y3
8000-9FFFH	1	0	0	X	X	Y4
A000-BFFFH	1	0	1	X	X	Y5
C000-DFFFH	1	1	0	X	X	Y6
E000-FFFFH	1	1	1	X	X	Y7

ตารางที่ 3.1 แสดงการถอดรหัสของหน่วยความจำกับMCS-51

จากรูปที่ 3.17 จะเห็นได้ว่าในส่วนของการเก็บข้อมูลจะใช้พื้นที่ในแอดเดรสตำแหน่งที่ 0000 – E000H ดังนั้นในการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับหน่วยความจำจะเป็นไปดังตารางที่ 3.1 สำหรับการเชื่อมต่อ MCS – 51 กับ 8255 จะใช้การถอดรหัสที่เอาท์พุท Y7 ของ IC 74LS138 มาถอดรหัสแบ่งแอดเดรสอีกทีหนึ่งเนื่องจากเราใช้ 8255 ถึง 2 ตัว

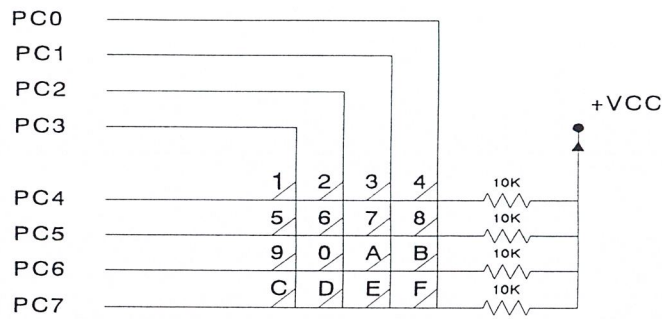
ADDRESS	A15	A14	A13	A12	A0	OUTPUT
E000-EFFFH	1	1	1	0	X	Y6
F000-FFFFH	1	1	1	1	X	Y7

ตารางที่ 3.2 แสดงการถอดรหัสของ 8255 กับMCS-51

จากตารางที่ 3.2 เห็นได้ว่า ได้แบ่งพื้นที่แอดเดรสให้กับ 8255 เพื่อติดต่อกับไมโครคอนโทรลเลอร์ โดย 8255 ตัวที่ 1 อยู่ที่แอดเดรส E000 – EFFFH และ 8255 ตัวที่ 2 อยู่ที่แอดเดรส F000 - FFFFH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.5 แป้นคีย์บอร์ด



รูปที่ 3.19 การต่อและจัดวางแป้นคีย์บอร์ด

จะเป็นบอร์ดที่มีจำนวน 16 คีย์ ดังรูปที่ 3.19 ซึ่งแสดงรูปของคีย์บอร์ดแบ่งเป็น 4 แถว 4 คอลัมน์ มีการต่อความต้านทานไว้ที่แหล่งจ่ายไฟขั้วบวก (ลอจิก 1) ซึ่งเรียกว่าพูลอัพ (R-PULL UP) ไว้ให้ซอฟต์แวร์ตรวจสอบการกดปุ่ม

1	2	3	4
5	6	7	8
9	0	INC	DEC
RTC	ESC	CLR	ENT

รูปที่ 3.20 การจัดวางตำแหน่งแป้นคีย์บอร์ด

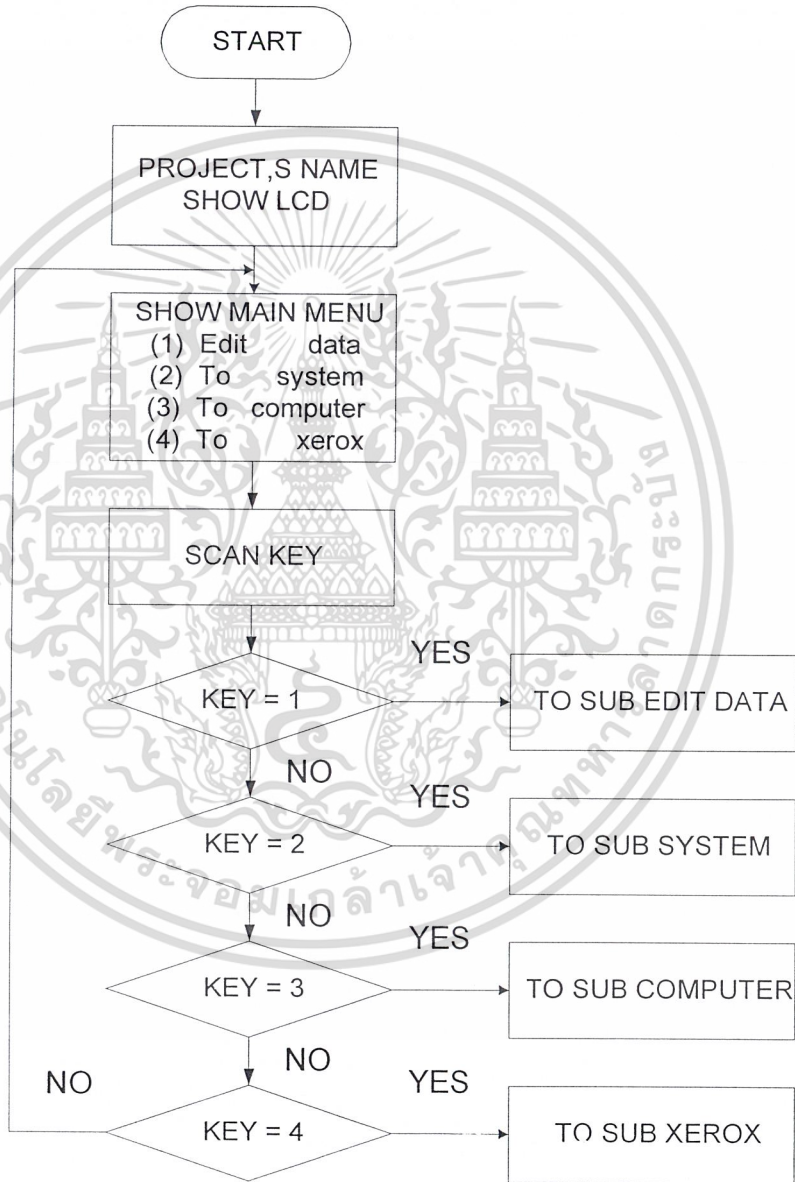
จากรูปที่ 3.20 แสดงการจัดวางตำแหน่งหน้าที่ของแป้นคีย์บอร์ด โดยแต่ละตำแหน่งมีหน้าที่ต่างๆ ดังนี้

- 0 - 9 ใช้สำหรับกดเลือกเมนูที่ต้องการ และ ค่ารหัสหมายเลขที่ต้องการ
- INC ใช้สำหรับเพิ่มค่าแอดเดรสที่ต้องการตรวจสอบข้อมูล
- DEC ใช้สำหรับลดค่าแอดเดรสที่ต้องการตรวจสอบข้อมูลและกลับไปสู่เมนูก่อนหน้า
- RTC ใช้สำหรับตรวจสอบคู่วันเวลา
- ESC ใช้สำหรับออกจากเมนูที่แสดงขณะนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- CLR ใช้สำหรับยกเลิกการทำงานที่ไม่ต้องการกระทำ
- ENT ใช้สำหรับกดเข้าสู่การทำงานที่ต้องการ

เนื่องจากเครื่องเก็บข้อมูลโดยผ่านบัตรแถบแม่เหล็กนี้ จะใช้ไมโครคอนโทรลเลอร์ในการควบคุมติดต่ออุปกรณ์ภายนอก,เกี่ยวกับฐานข้อมูล และทำหน้าที่ในลักษณะต่างๆ ซึ่งสามารถเขียนการทำงานเป็นโฟลว์ชาร์ท (flow chart) หลัก ๆ ดังนี้



รูปที่ 3.21 แผนผัง โปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก Flowchart จะเห็นได้ว่าจะมี mode การทำงานอยู่ 4 mode ซึ่งสามารถแบ่งเป็นย่อยๆ ได้ดังนี้

Function 1 Edit data

เป็นส่วนที่ทำหน้าที่แก้ไขข้อมูลทุกอย่างของระบบโดยส่วนนี้จะเป็นส่วนที่ผู้ควบคุมรับผิดชอบ ระบบเครื่องจะสามารถเข้ามาแก้ไขได้เพียงผู้เดียวเนื่องจากจะมีการถามรหัสผ่านของเครื่องก่อนทำการแก้ไขซึ่งการแก้ไขประกอบไปด้วย

- Edit SPASS แก้ไขรหัสผ่านของระบบ
- Edit ID แก้ไขหมายเลขประจำตัวผู้ถือบัตรซึ่งมีรายละเอียดต่างๆ ดังนี้
 - Insert ID เป็นการใส่รหัสเข้าสู่ระบบเครื่อง
 - Delete ID ลบรหัสที่ไม่ต้องการออกจากระบบ
 - Record ID บันทึกรหัสประจำตัวลงบัตร
 - Clear ID รหัสทั้งหมดออกจากเครื่อง
- Edit YPASS แก้ไขรหัสผ่านส่วนบุคคล
- Edit RTC แก้ไขเปลี่ยนแปลงวันเวลา มีเมนูให้เลือกแก้ไข ดังนี้
 - Present time แก้ไขเวลาให้ตรงเวลากับปัจจุบัน
 - Record time แก้ไขเวลาในการบันทึกข้อมูลการใช้บริการของสมาชิกทั้งหมดต่อคอมพิวเตอร์
- Check data ตรวจสอบข้อมูลทั้งหมดที่อยู่ในหน่วยความจำ
- Delete data ลบข้อมูลทั้งหมดของการใช้บริการ
- Print พิมพ์รายงานผลออกทางเครื่องพิมพ์

เป็นส่วนที่เจ้าของระบบสามารถขอพิมพ์รายงานผลข้อมูลของผู้ใช้บริการในลักษณะเลือกการพิมพ์รายงานได้ คือ

- ALL MEMBER เป็นการสั่งพิมพ์รายงานข้อมูลของสมาชิกทุกคน
- SELECT ID. เป็นการสั่งพิมพ์รายงานข้อมูลของสมาชิกที่ต้องการดูผลข้อมูล

Function 2 To system

เป็นส่วนที่ติดต่อระหว่างระบบซึ่งก็คือไมโครคอนโทรลเลอร์กับเครื่องอ่านและบันทึกข้อมูลของบัตรแม่เหล็กซึ่งจะมีการตรวจสอบการนำบัตรเข้าหรือออกจากเครื่องอ่าน / บันทึก เมื่อสามารถเข้าสู่ระบบได้จะมีเมนูย่อยดังนี้

- Your Record แสดงผลข้อมูลที่ถูกบันทึกไว้ในเครื่องเก็บข้อมูลนี้
- To Computer สามารถติดต่อไปยัง Computer ได้
- To Printer พิมพ์รายงานผลข้อมูลประจำวันของผู้ใช้บริการคนนั้นได้
- To Xerox ติดต่อใช้งานกับเครื่องถ่ายเอกสารได้

Function 3 To Computer

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สงวนไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้นหากท่านเป็นต้นตอของเรื่องร้องเรียนจะต้องแจ้งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
ระบบ ซึ่งจะไปตรวจสอบข้อมูลต่างๆ ในส่วนของคอมพิวเตอร์

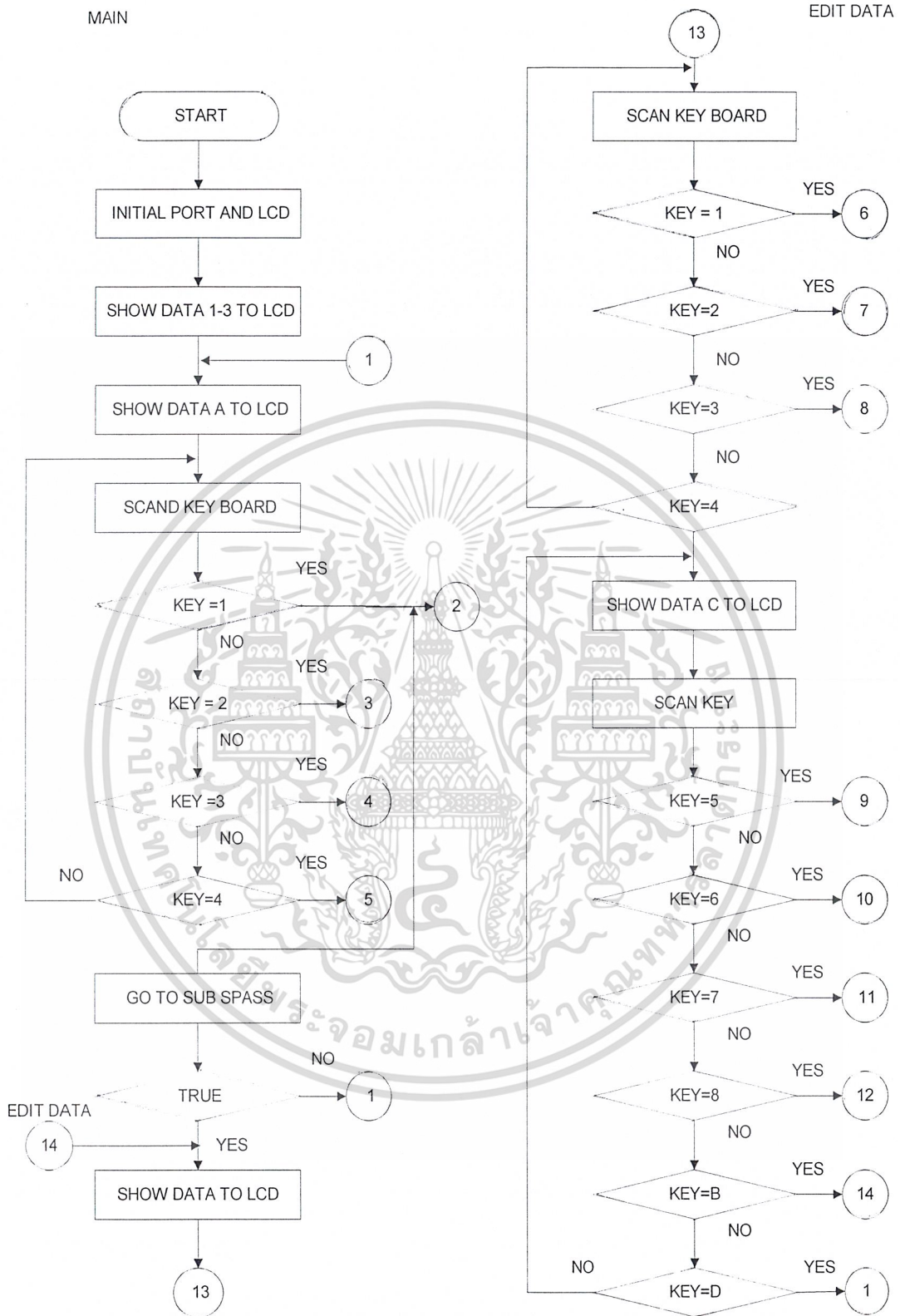
Function 4 To Xerox

เป็นส่วนที่ทำหน้าที่ติดต่อกับเครื่องถ่ายเอกสาร โดยจะทำการตรวจสอบการใช้กระดาษในการถ่ายเอกสาร ซึ่งจะมีการจำกัดกระดาษที่ใช้ ดังนั้นถ้าผู้ใช้บริการมีการใช้กระดาษเกินจะถูกระงับการใช้กระดาษจนกว่าจะถึงรอบการใช้งานใหม่

ดังนั้น จาก function ต่างๆ ที่ได้อธิบายไว้ข้างต้น เขียนเป็น flow chart เพื่อนำไปสู่การอธิบายโปรแกรมทั้งหมดได้ดังนี้

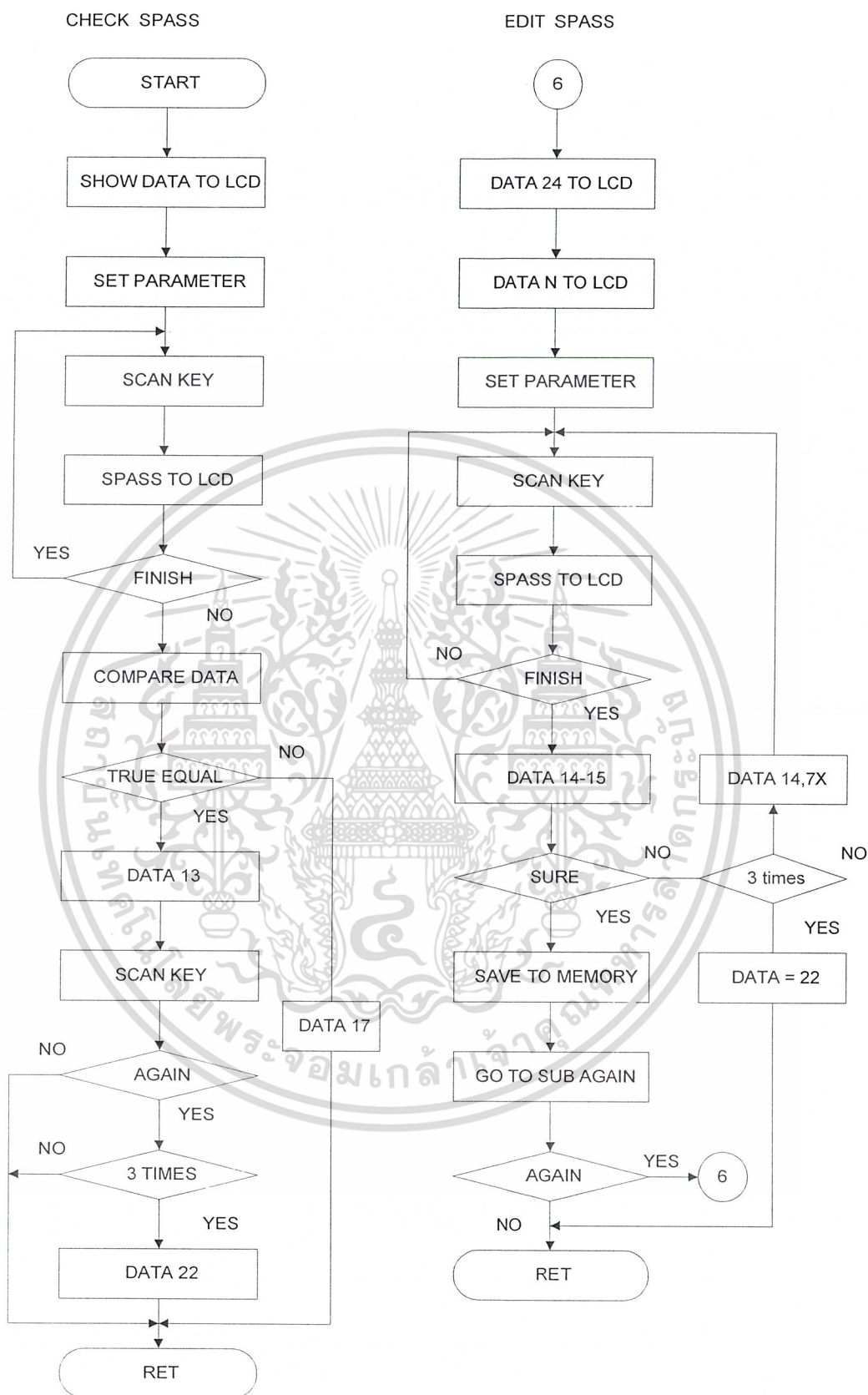


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

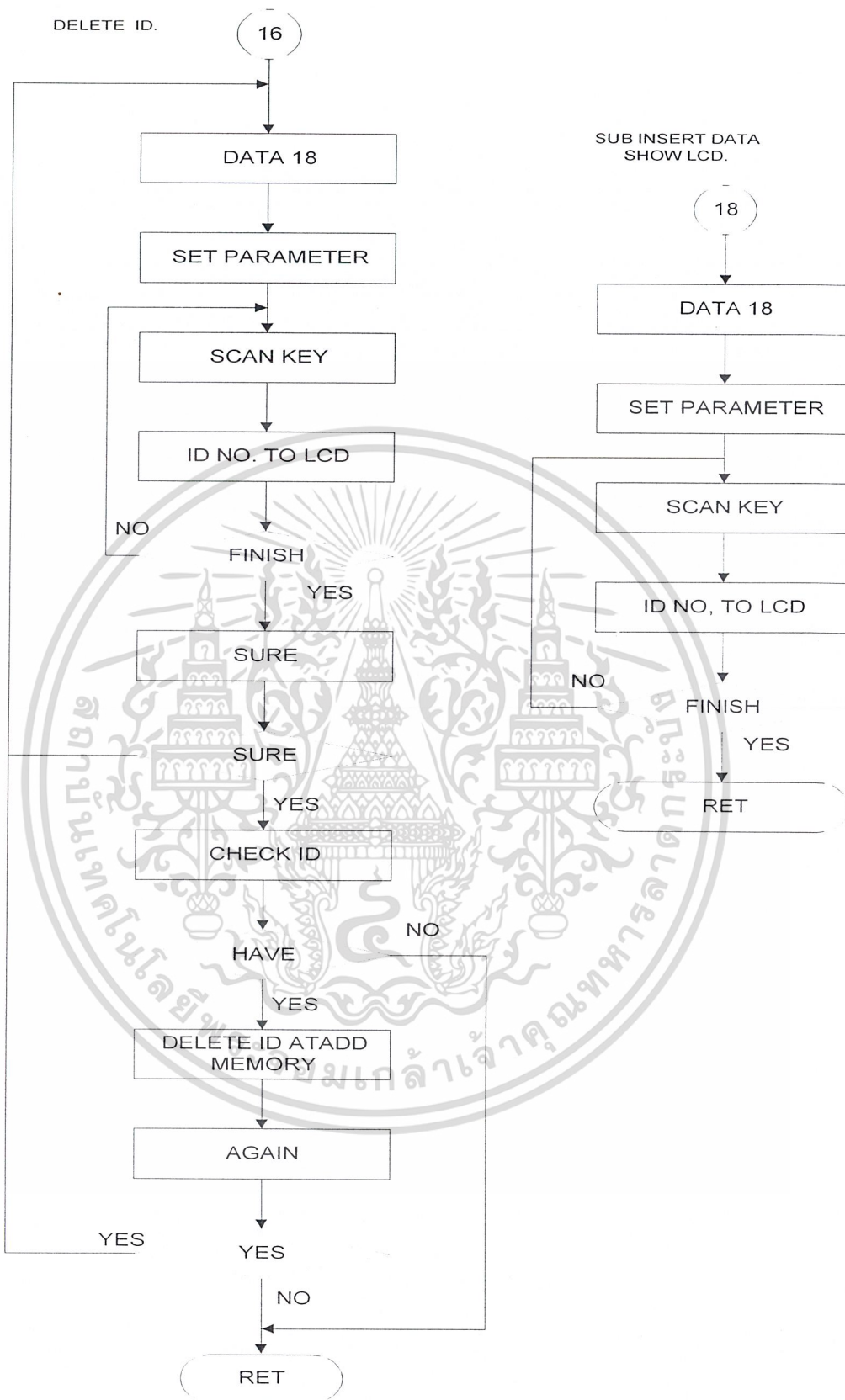


รูปที่ 3.22 แผนผังโปรแกรมในส่วนหลักและกรแก้ไขข้อมูล

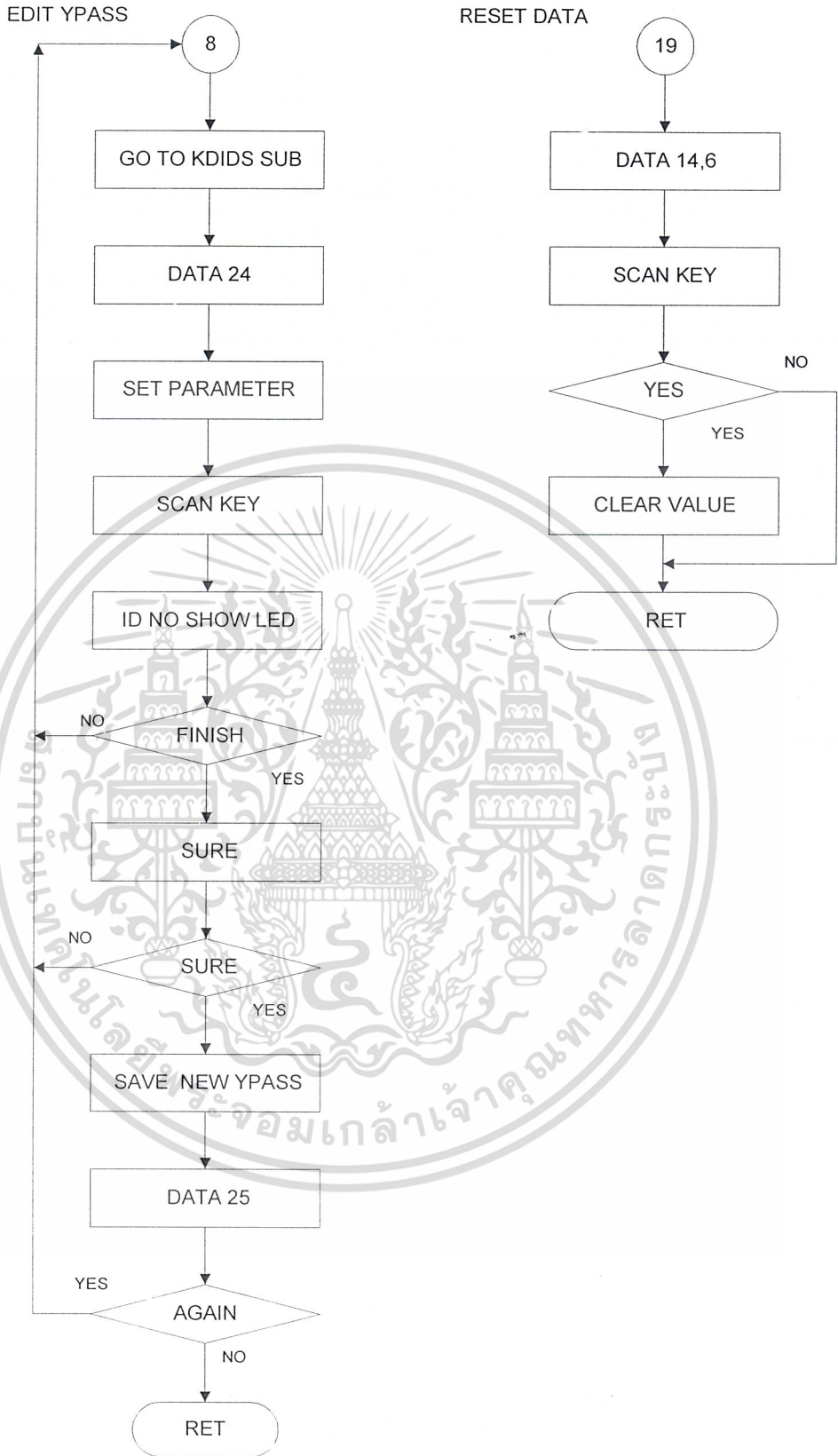
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



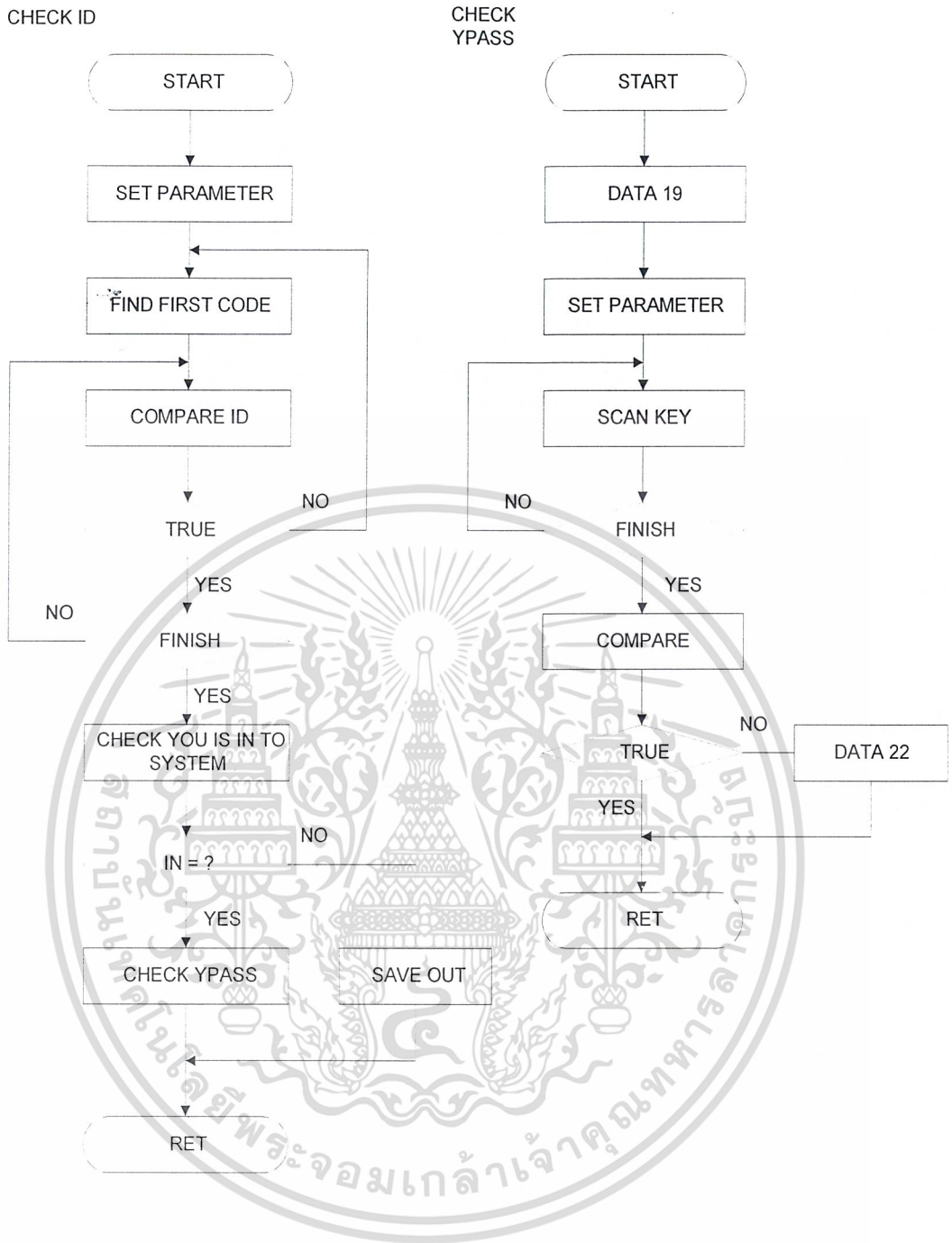
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีเนื้อหาและข้อมูลของเอกสารที่สงวนไว้ที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 3.24 แผนผังโปรแกรมลบรหัสสมาชิกออกจากระบบและการป้อนคีย์ผ่านแอลซีดี
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อแหล่งอื่นและต้องอ้างอิงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



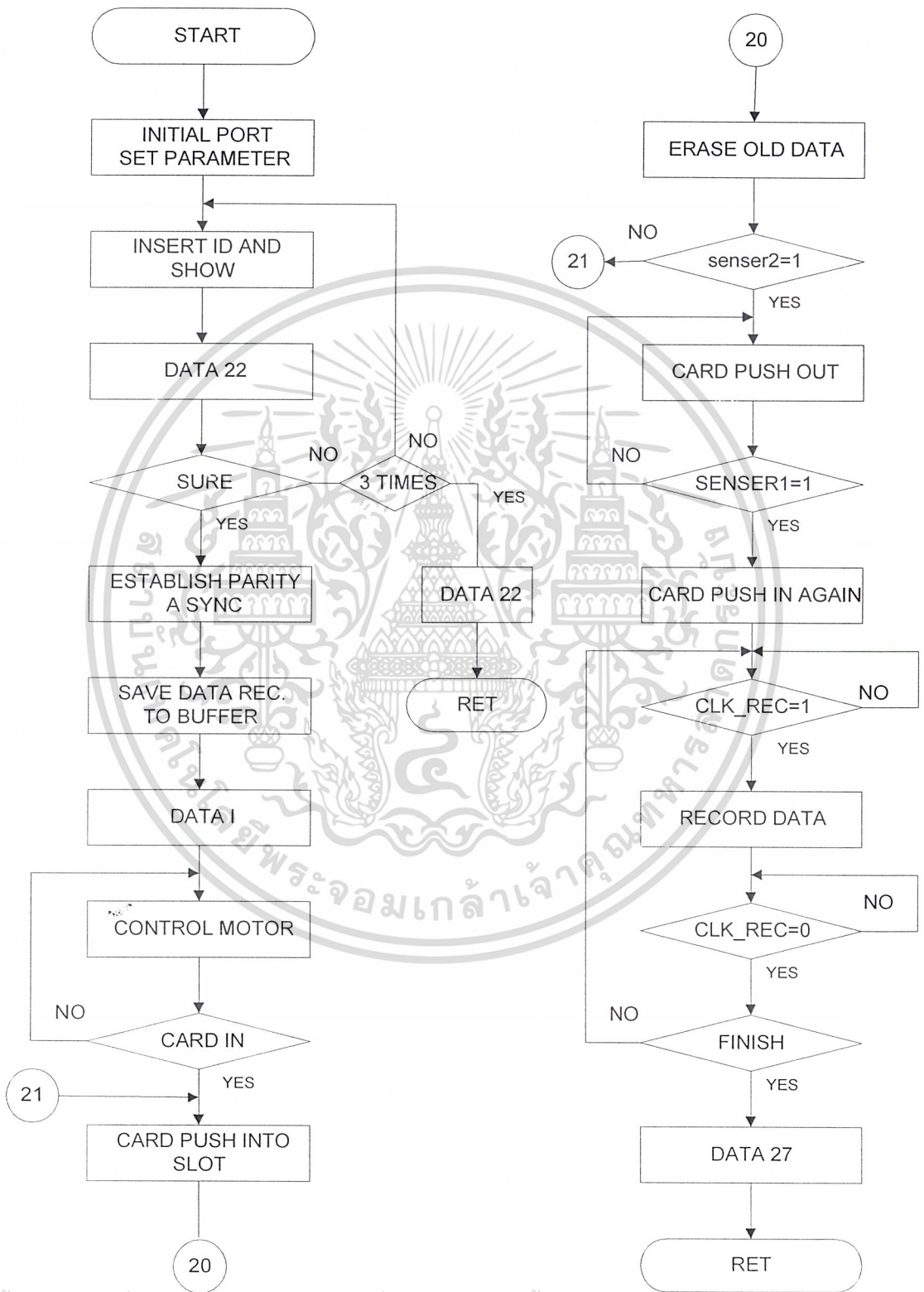
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีพัสส์ ยานบี ให้ต้นแบบเนื้อหาและสิ่งพิมพ์อ้างอิงนี้ของเอกสารชุดนี้ที่มีการนำไปใช้
รูปที่ 3.25 แผนผังโปรแกรมแก้ไขรหัสผ่านของผู้ใช้และลบข้อมูลทั้งหมด



รูปที่ 3.26 แผนผังโปรแกรมการตรวจสอบรหัสสมาชิกและตรวจสอบรหัสผ่าน

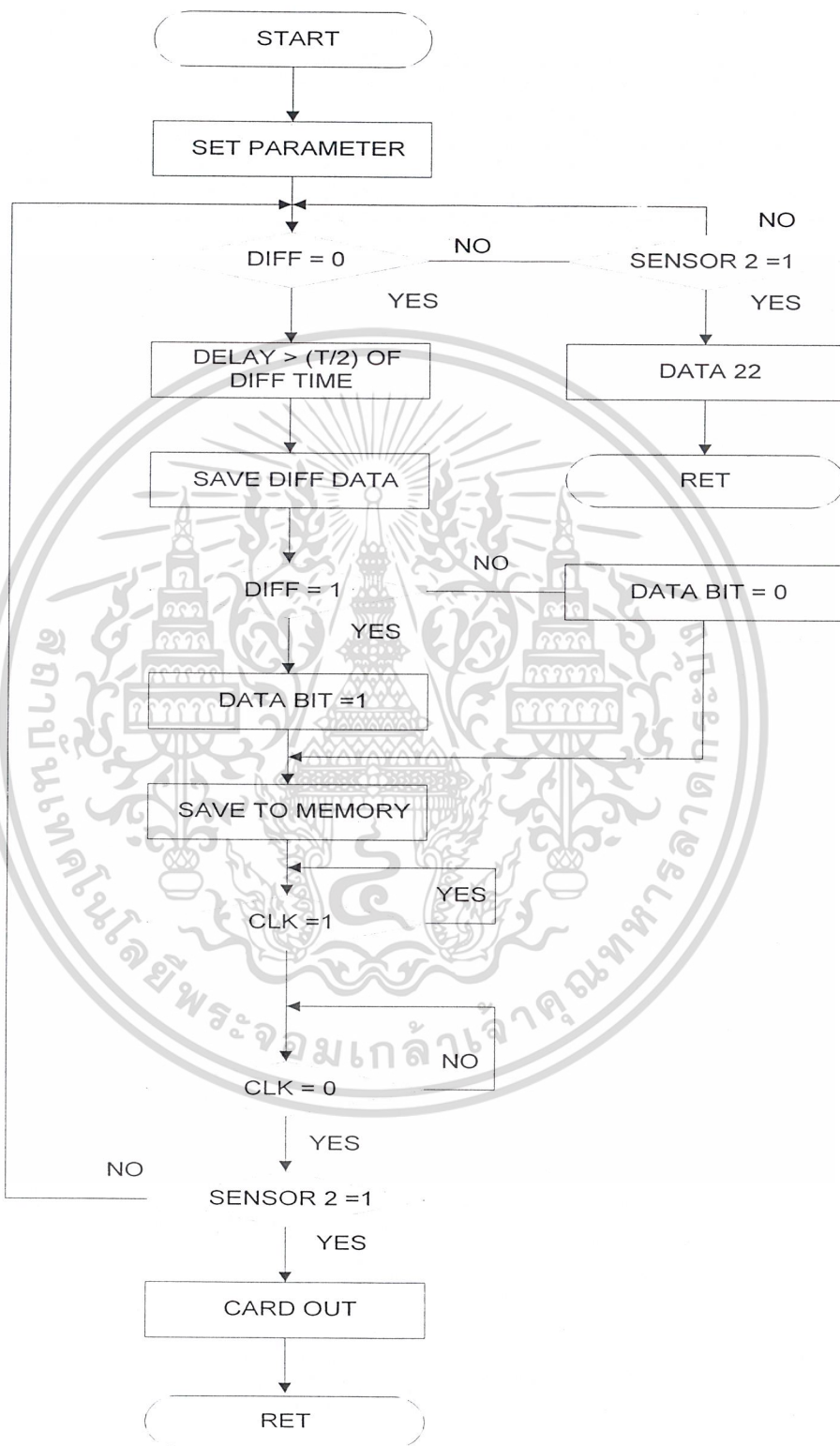
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RECORD ID



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกรูปที่ 3.27 แผ่นผังโปรแกรมการบันทึกรหัสข้อมูลลงบนบัตรแม่เหล็ก

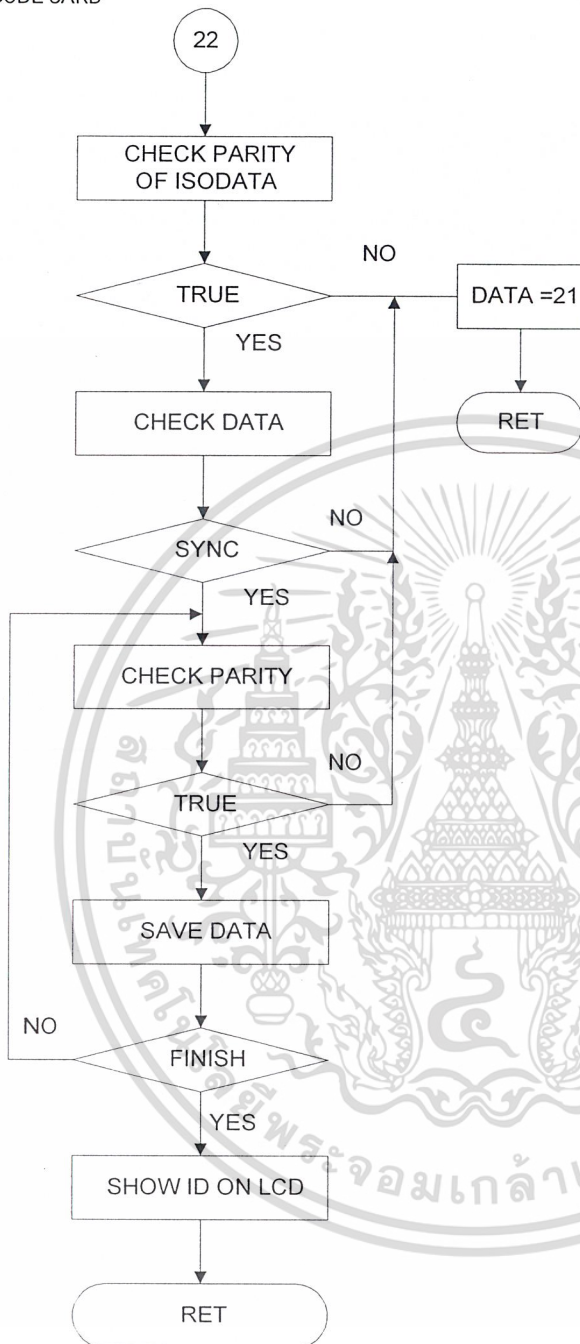
READ CARD



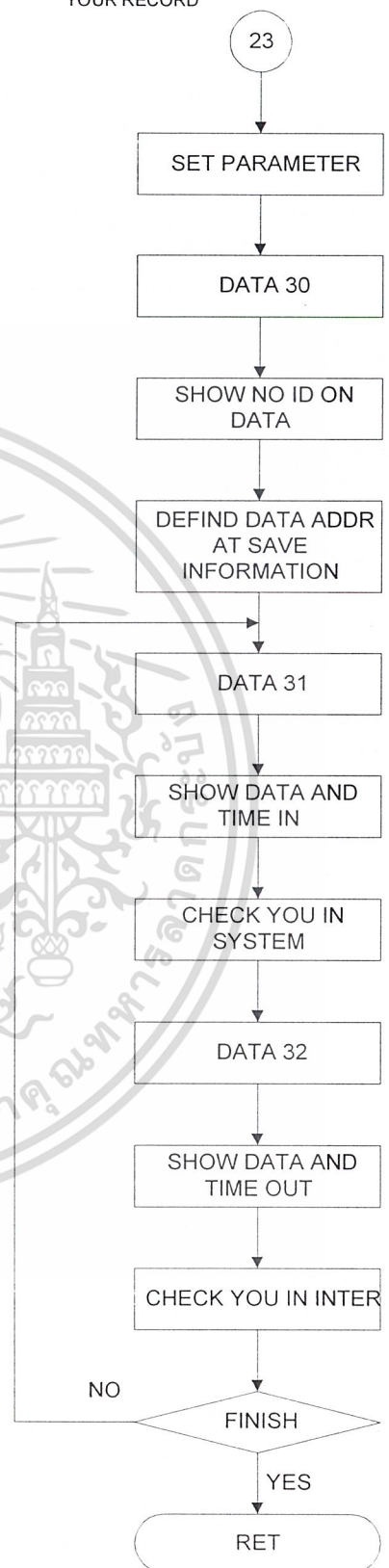
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงทั้งรูปและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.28 แผนผังโปรแกรมการอ่านรหัสบนบัตรแม่เหล็ก

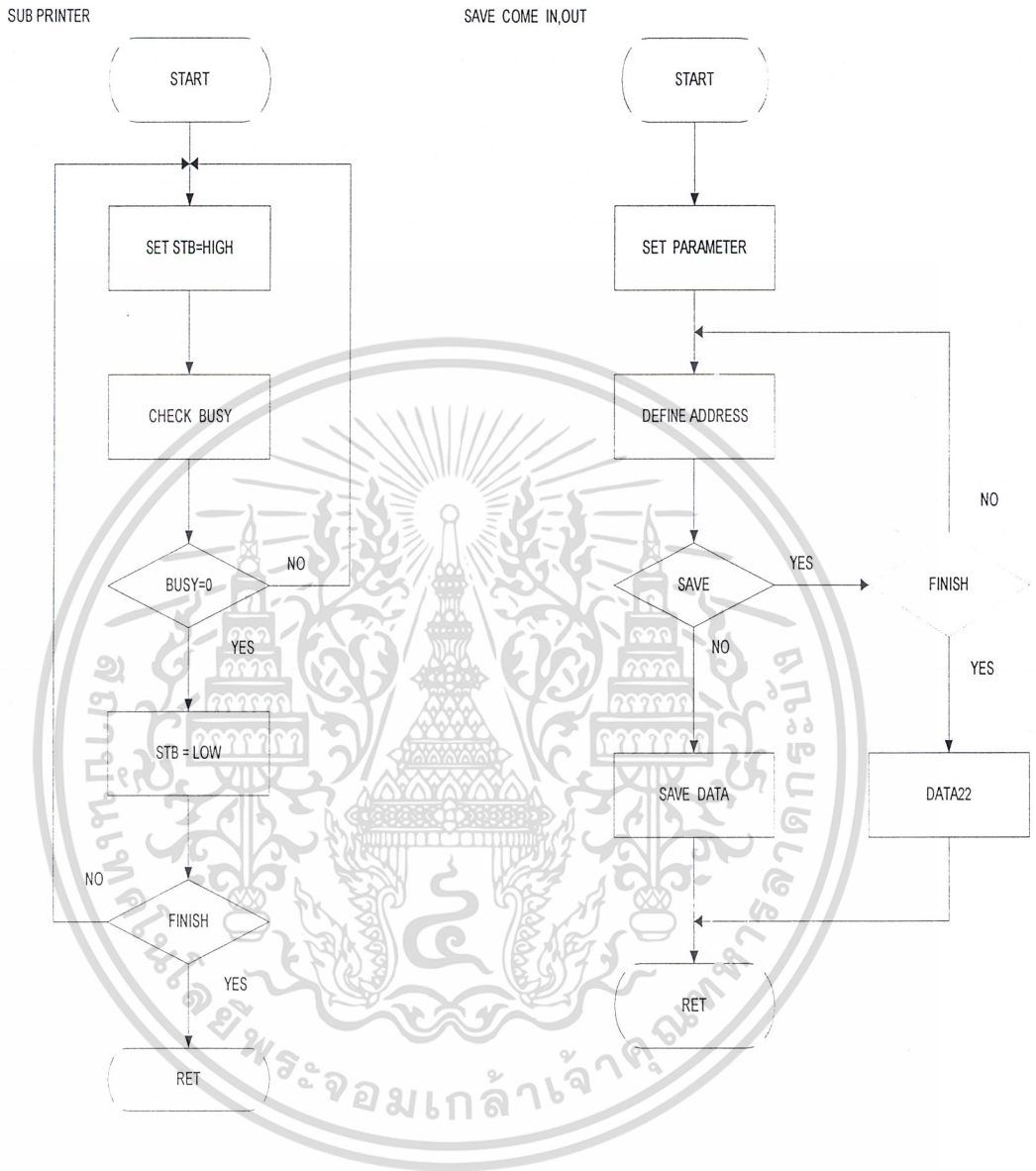
DECODE CARD



YOUR RECORD



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 3.29 แผนผังโปรแกรมการถอดรหัสบัตรและแสดงผลข้อมูลที่ให้บริการ
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ลงเนื้อหาและต้องอยู่ใต้อาณัติของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

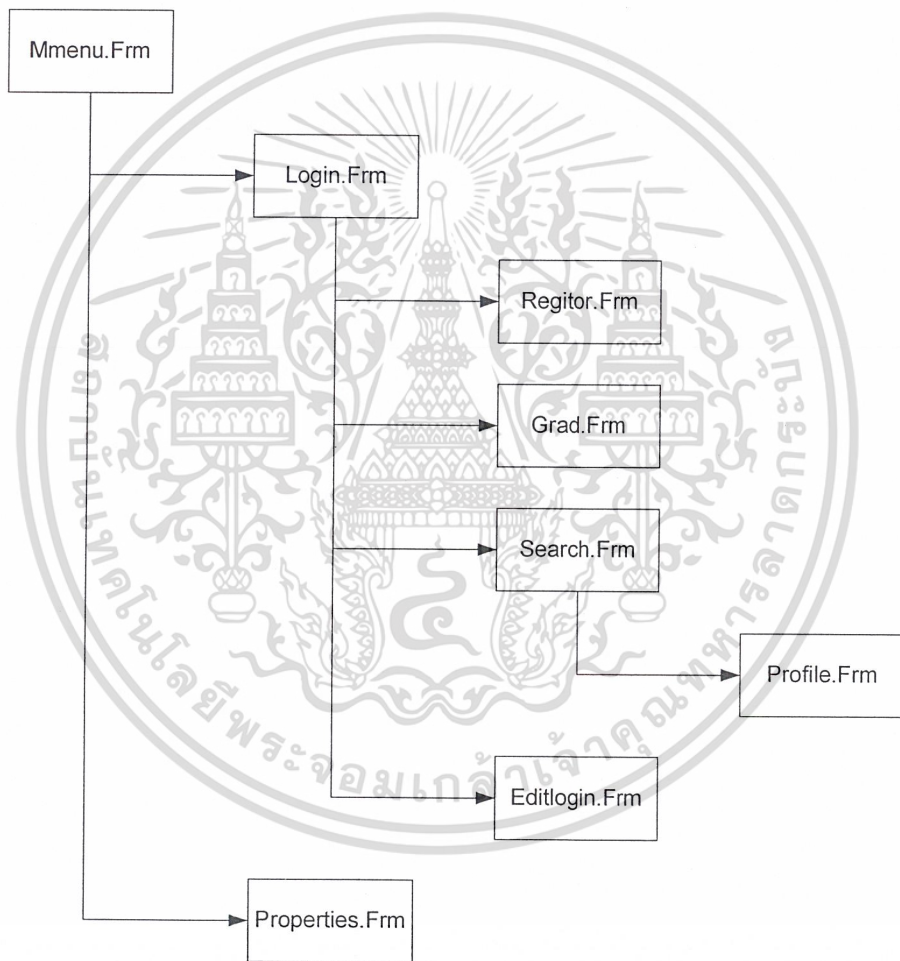


รูปที่ 3.30 แผนผังโปรแกรมกรติดต่อกับเครื่องพิมพ์และการบันทึกข้อมูลการใช้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

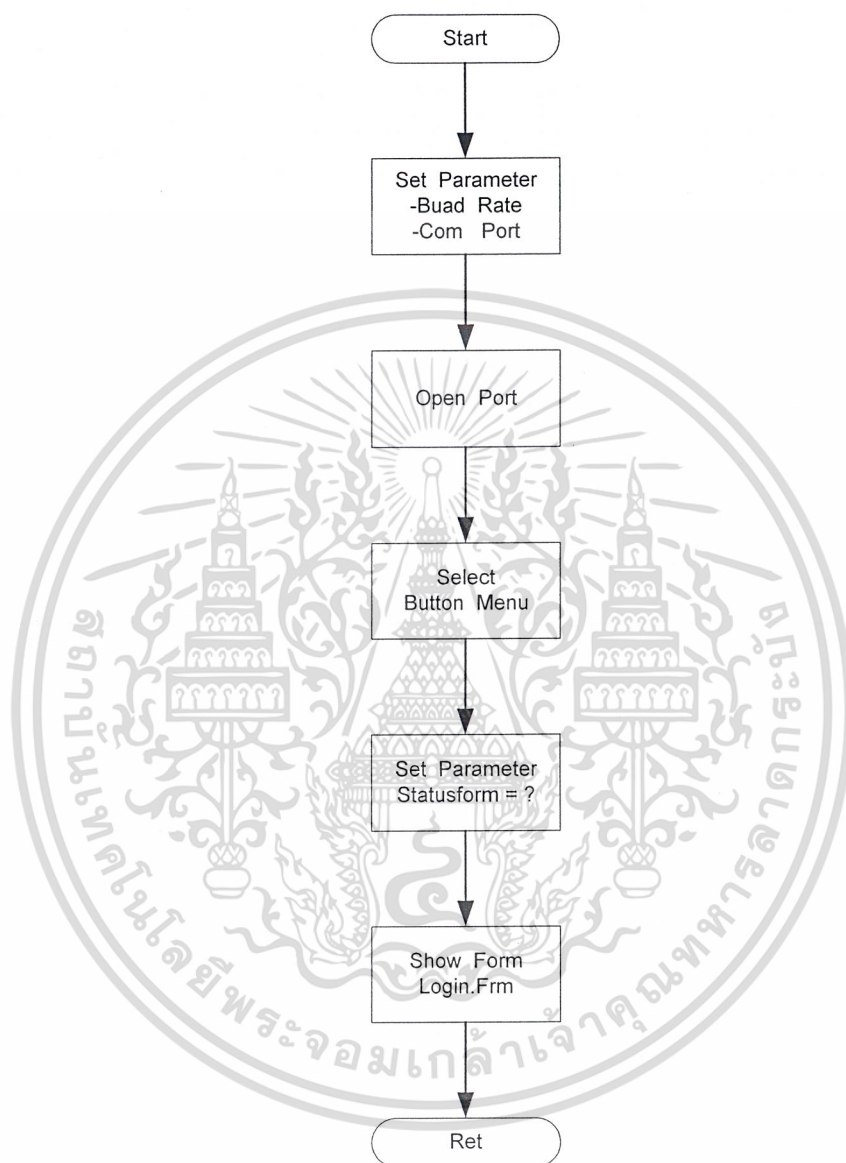
3.6 ส่วนติดต่อและแสดงผลทางคอมพิวเตอร์

สำหรับการติดต่อสื่อสารข้อมูลระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์ จะติดต่อสื่อสารผ่านทางพอร์ตอนุกรม ซึ่งได้ใช้โปรแกรม Visual Basic เป็นโปรแกรมหลักในการติดต่อและแสดงผลผ่านทางมอนิเตอร์ เพื่อแสดงข้อมูลต่างๆ ของผู้ใช้บริการ โดยส่วนของข้อมูลที่จะนำมาเก็บลงในคอมพิวเตอร์จะใช้ในลักษณะฐานข้อมูลโดยใช้โปรแกรม Microsoft Access ช่วยในการเก็บข้อมูลแต่ละบุคคล ดังนั้นสามารถอธิบายเป็นแผนผังโปรแกรมได้ดังนี้



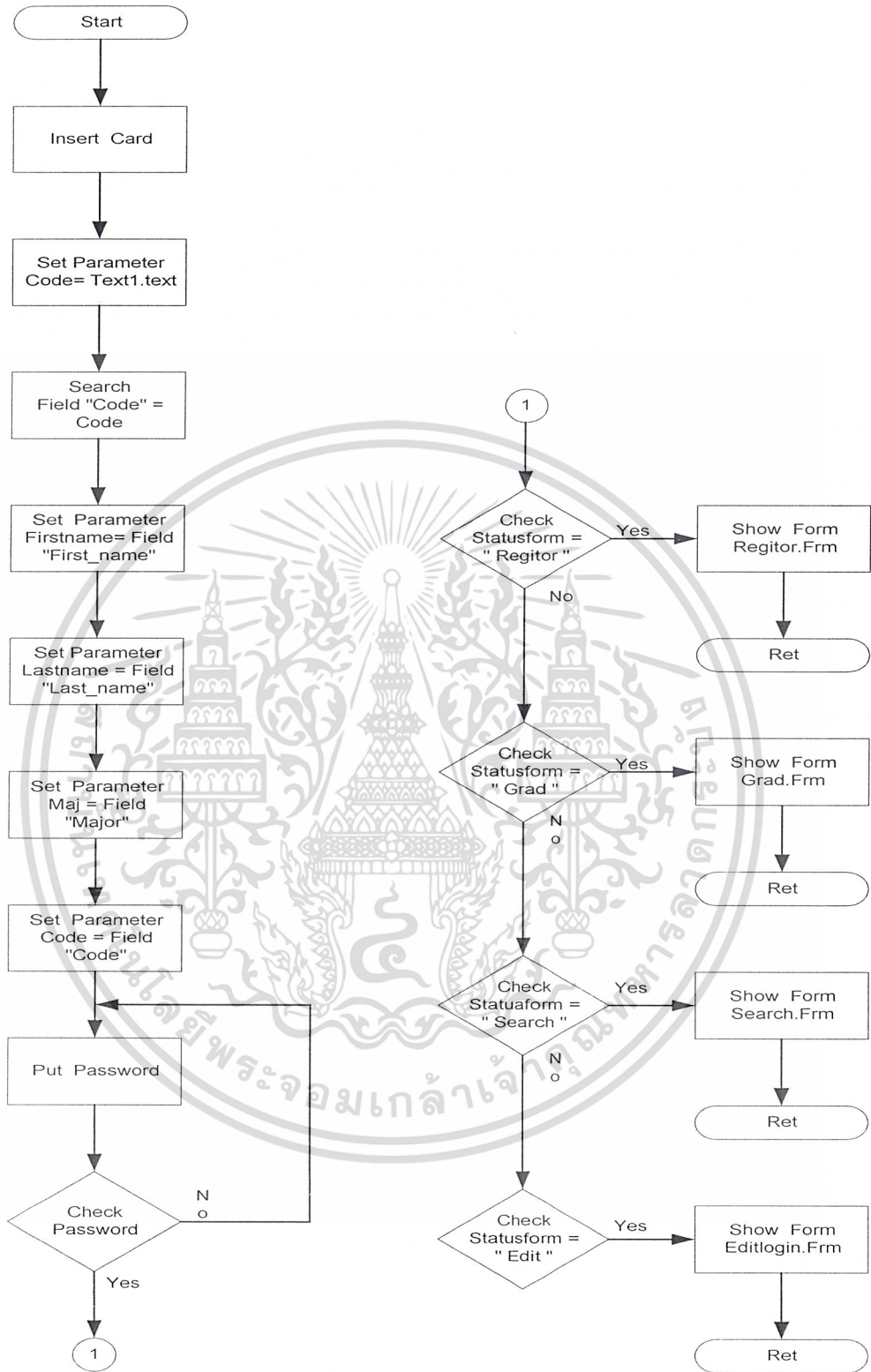
รูปที่ 3.31 แผนผัง โปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

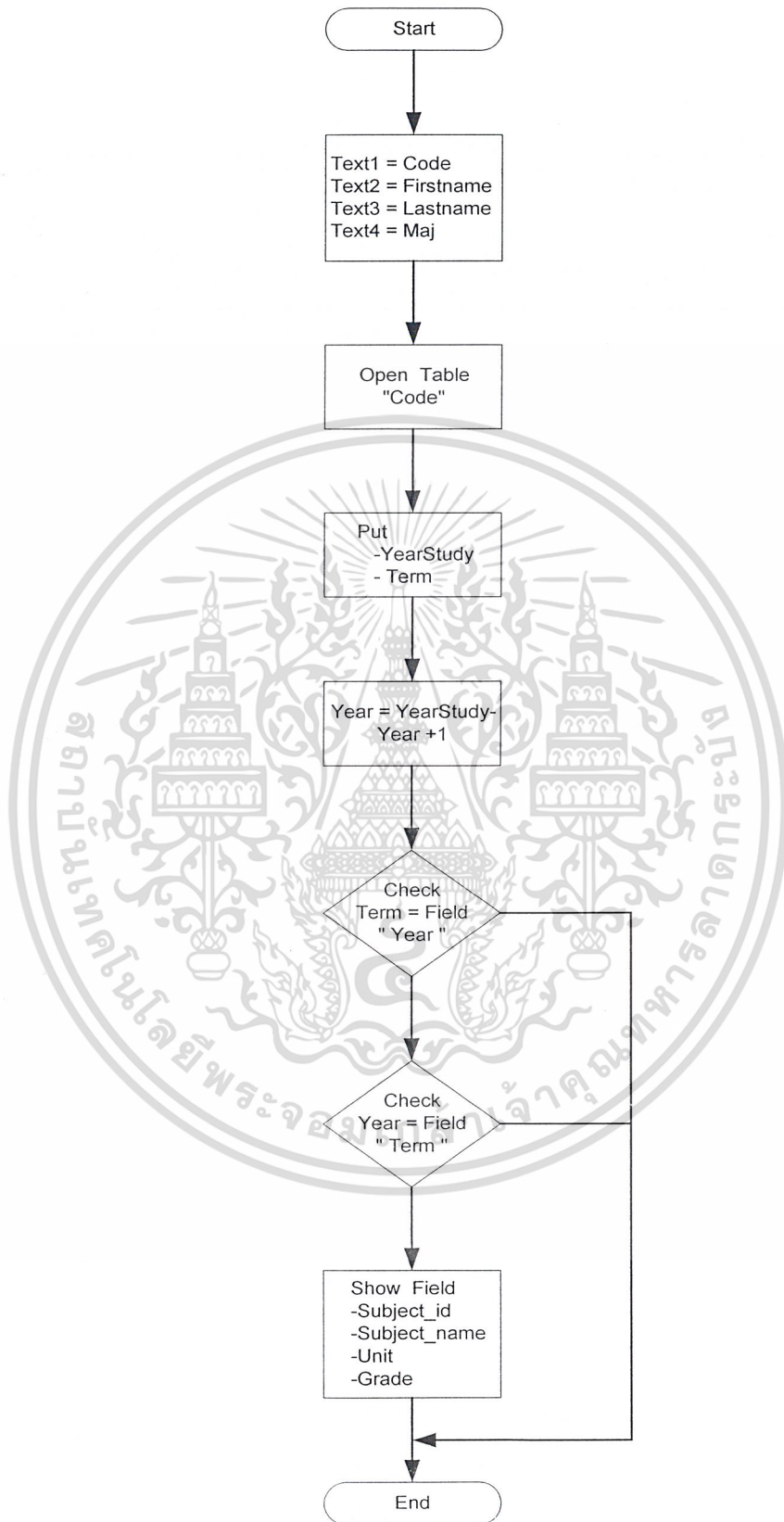


รูปที่ 3.32 แผนผัง โปรแกรมเมนูฟอร์ม

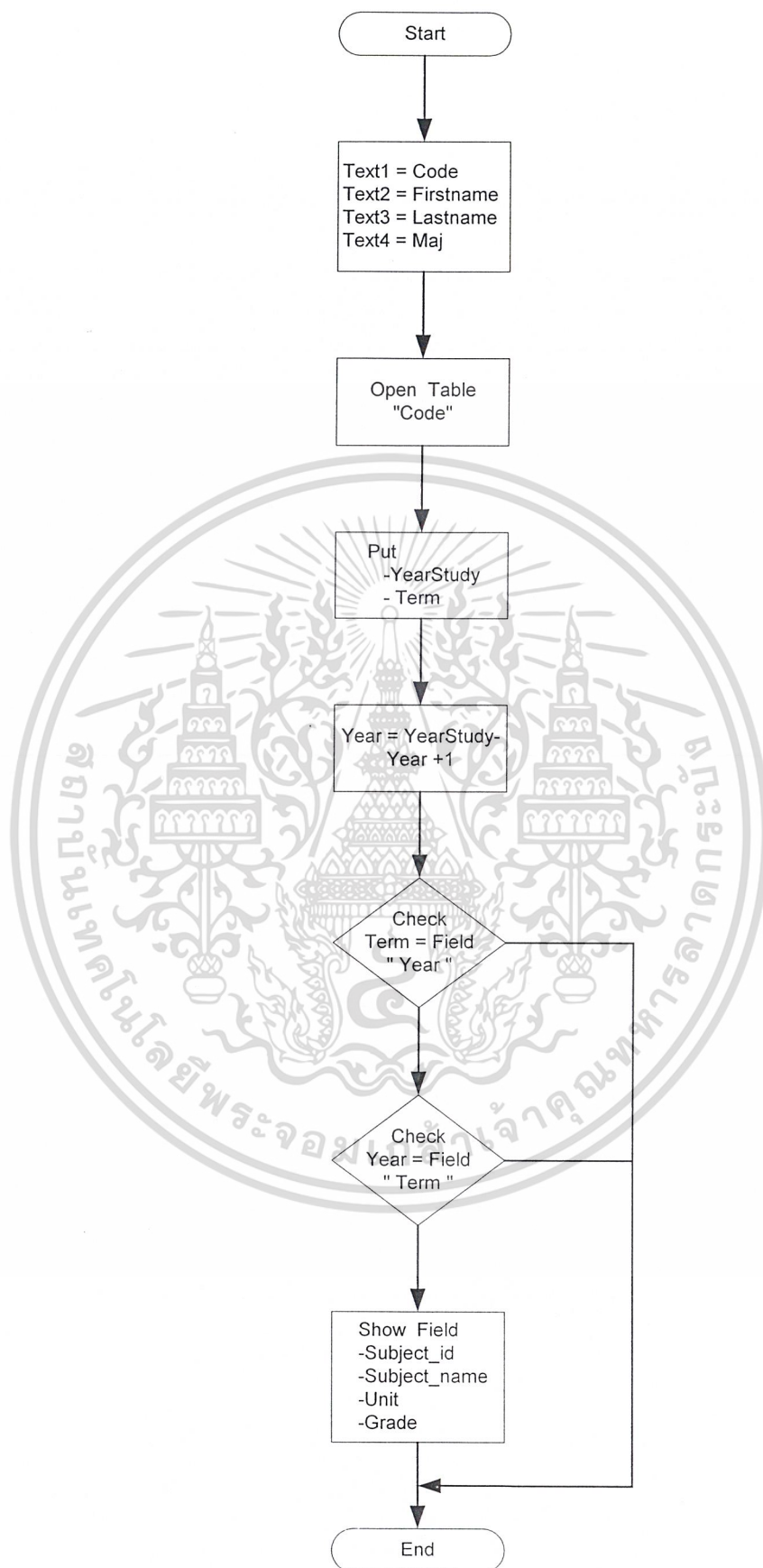
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบุคลากรในหน่วยงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 3.33 แผนผังโปรแกรมการเข้าสู่ระบบ (Login)
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

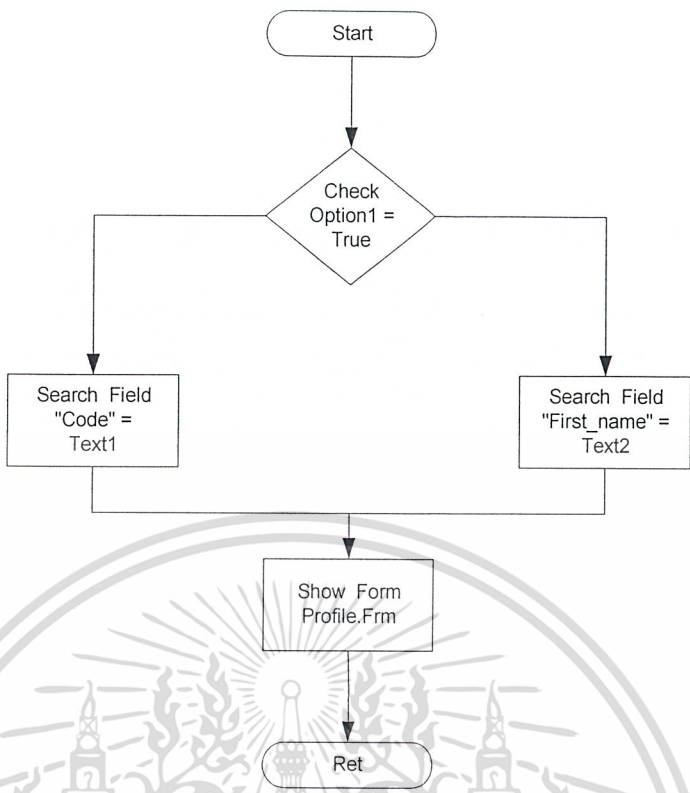


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 3.34 แผนผังโปรแกรมผลการลงทะเบียน
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

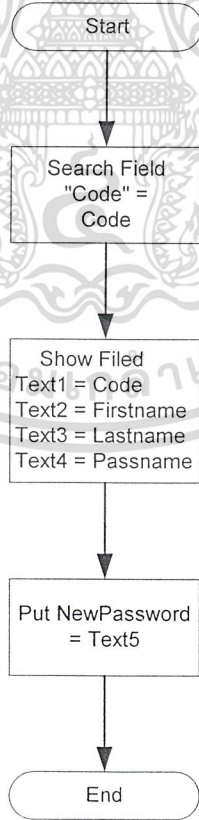


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงข้อมูลหรือทำซ้ำของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.35 แผนผังโปรแกรมผลการศึกษา

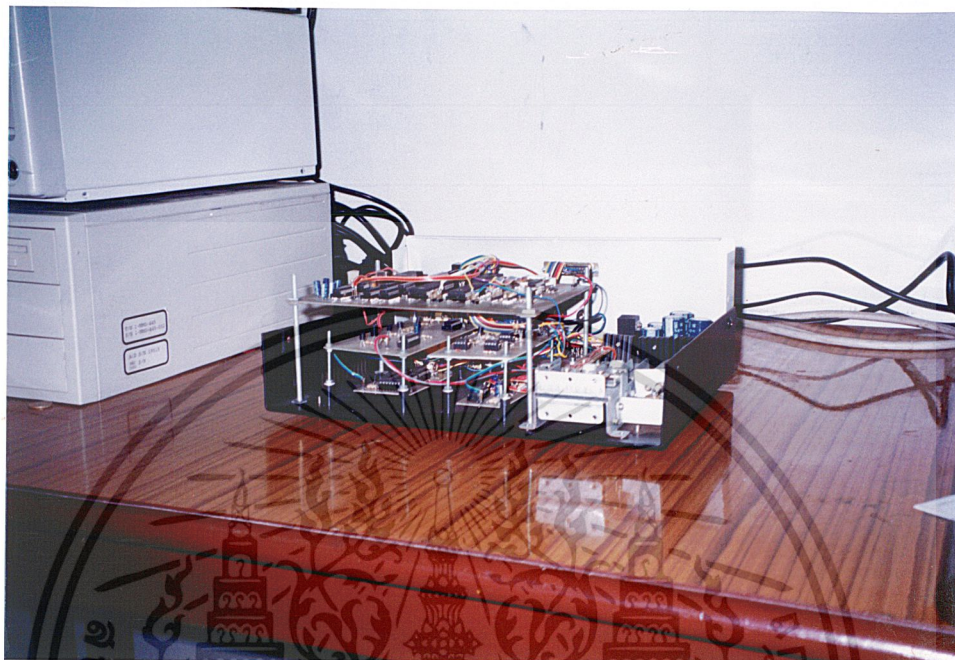


รูปที่ 3.36 แผนผังโปรแกรมการค้นหาประวัติบุคคล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 3.37 แผนผังโปรแกรมการแก้ไขรหัสผ่าน
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

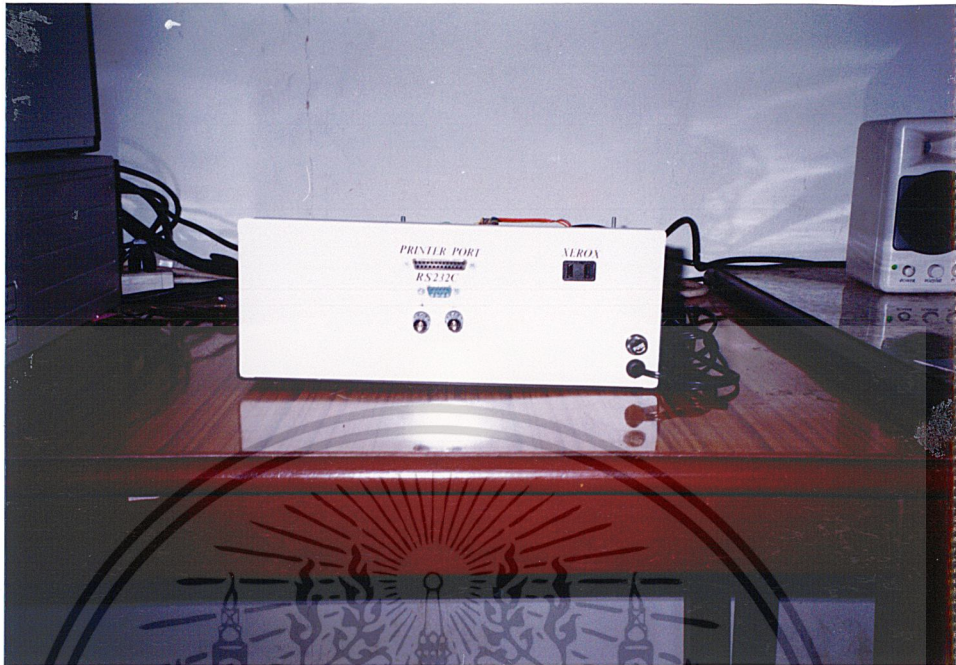
จากการออกแบบในส่วนต่าง ๆ ได้ทำการประกอบวงจรบนแผ่นปริ้นท์เอนกประสงค์ก่อนเพื่อทดลองทดสอบแก้ไขปรับปรุงวงจรก่อนประกอบทำเครื่องต้นแบบ ซึ่งวงจรและลายวงจรพิมพ์ต่าง ๆ ได้แทรกไว้ในภาคผนวกแล้ว



รูปที่ 3.38 แสดงเครื่องต้นแบบของเครื่องเก็บข้อมูล โดยผ่านบัตรแถบแม่เหล็ก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้รูปที่ 3.39 นี้ แสดงส่วนหน้าของเครื่องต้นแบบเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.40 แสดงส่วนหลังของเครื่องต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

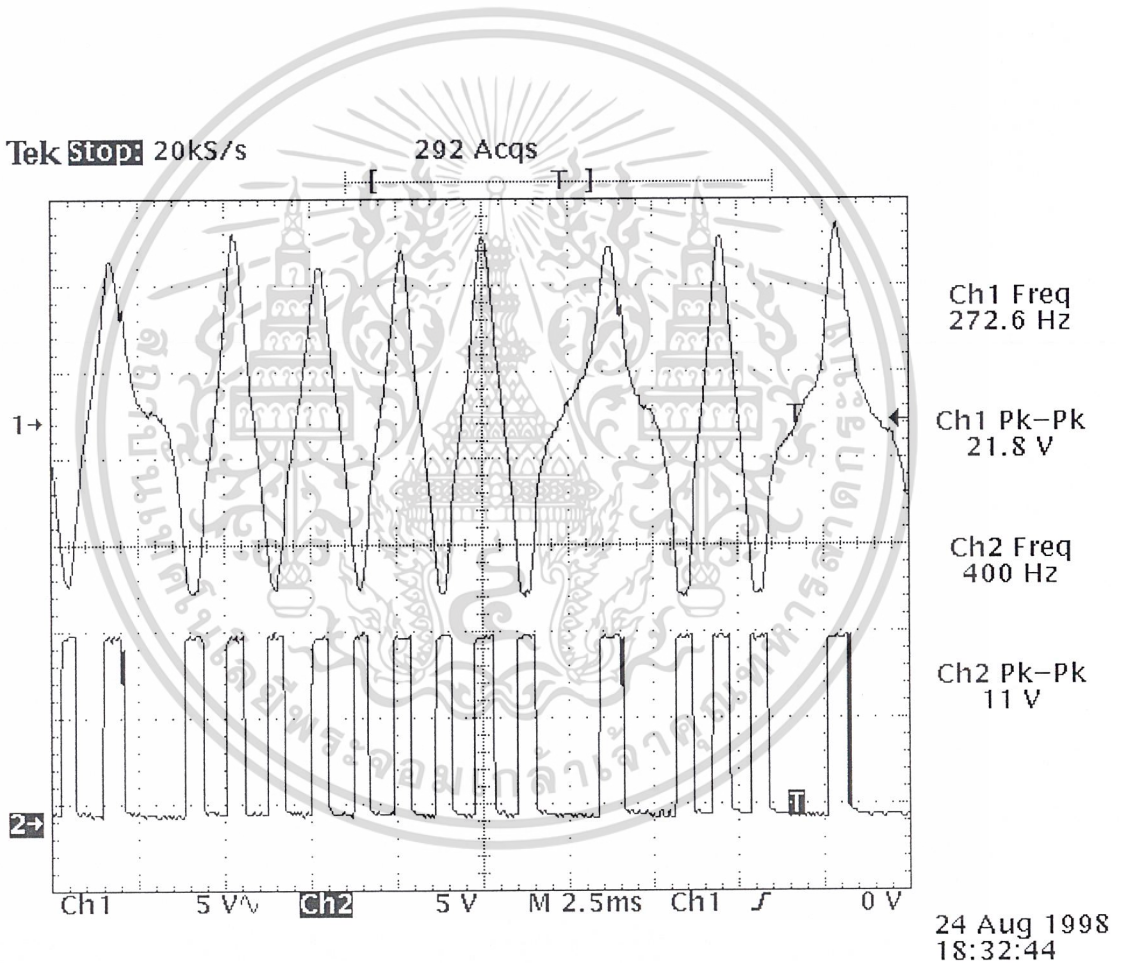
การทดลองและผลการทดลอง

ในการทดลองได้ใช้บัตรแม่เหล็กเอทีเอ็มของธนาคารกรุงเทพและกรุงไทย ในการตรวจสอบวัดผลการทดลอง เนื่องจากบัตรแม่เหล็กที่จะใช้ในโครงการนี้เป็นมาตรฐานเดียวกันกับบัตรแม่เหล็กของธนาคารต่างๆ

4.1 ส่วนของการอ่านข้อมูลจากบัตร

4.1.1 วัดผลการทดลองวงจรขยายสัญญาณจากหัวเทป

4.1.2 วัดผลการทดลองวงจรตรวจจับแรงดันสูงสุด

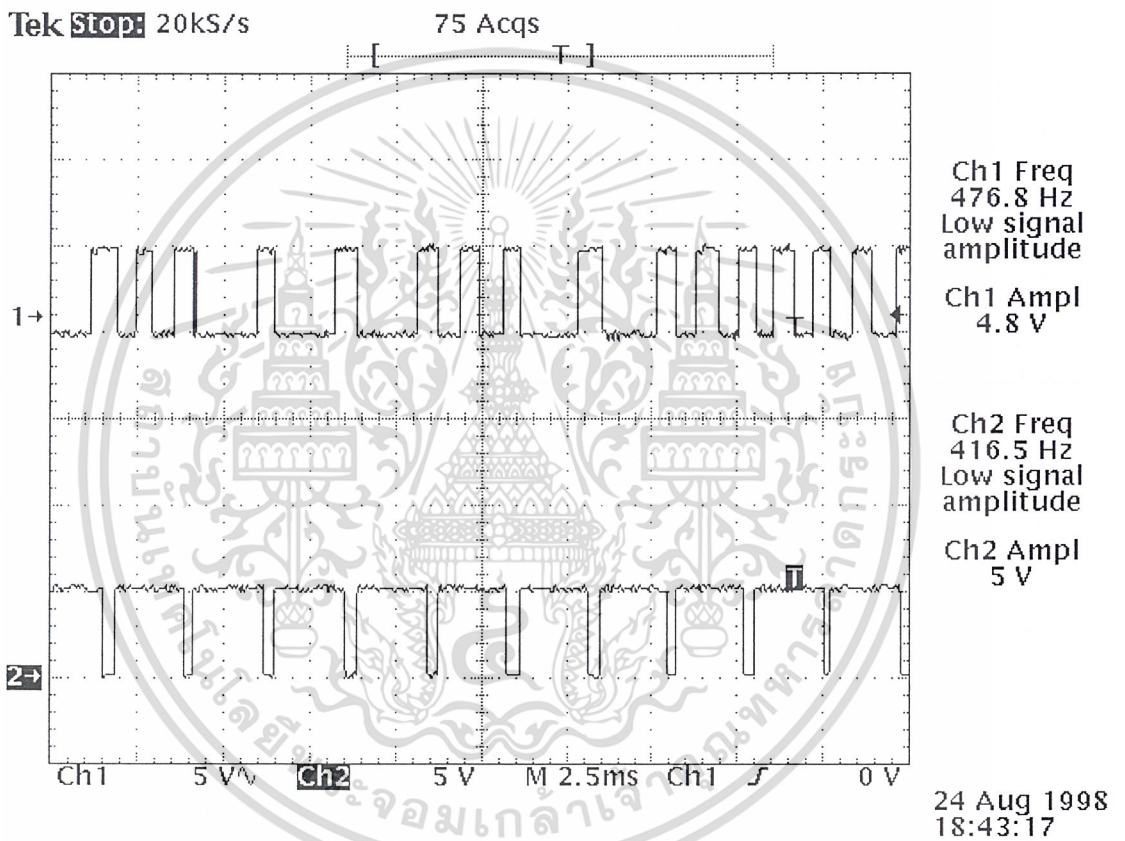


รูปที่ 4.1 แสดงผลการทดลองวงจรขยายสัญญาณและวงจรตรวจจับแรงดันสูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 วัดผลการทดลองวงจรปรับแต่งรูปคลื่น

4.1.4 วัดผลการทดลองวงจรสร้างฐานเวลาในการถอดรหัส



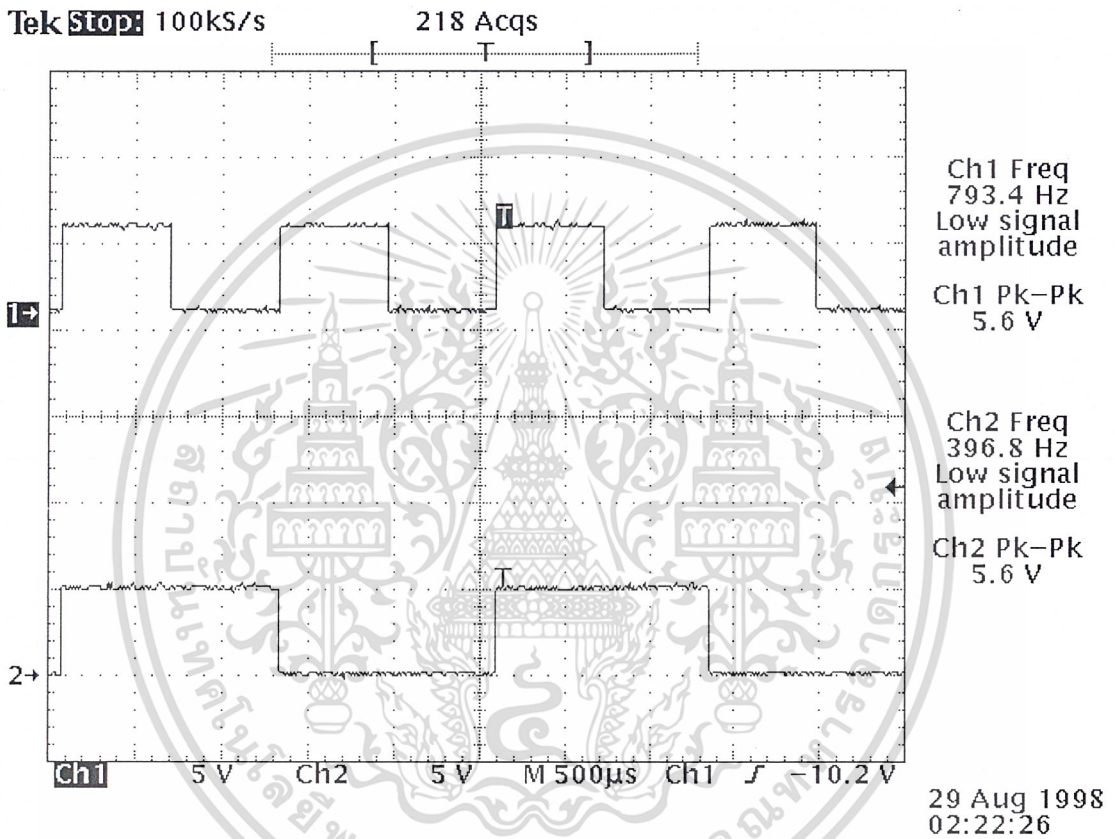
รูปที่ 4.2 แสดงผลการทดลองวงจรปรับแต่งรูปคลื่นและสร้างฐานเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ส่วนของการบันทึกข้อมูลลงบัตร

4.2.1 วัดผลการทดลองวงจรสร้างสัญญาณพิก้า

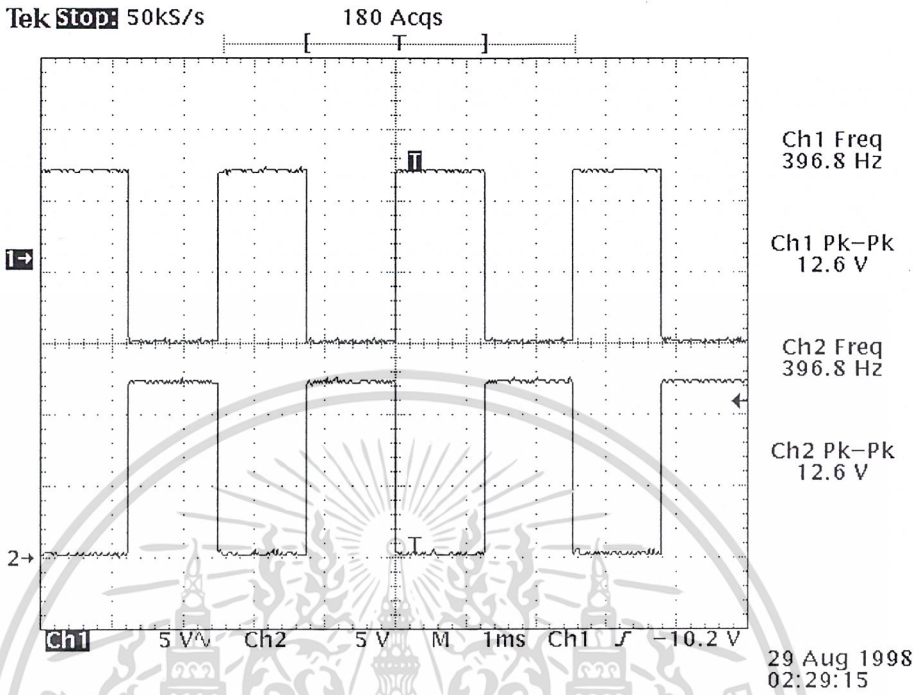
4.2.2 วัดผลการทดลองวงจรหารความถี่



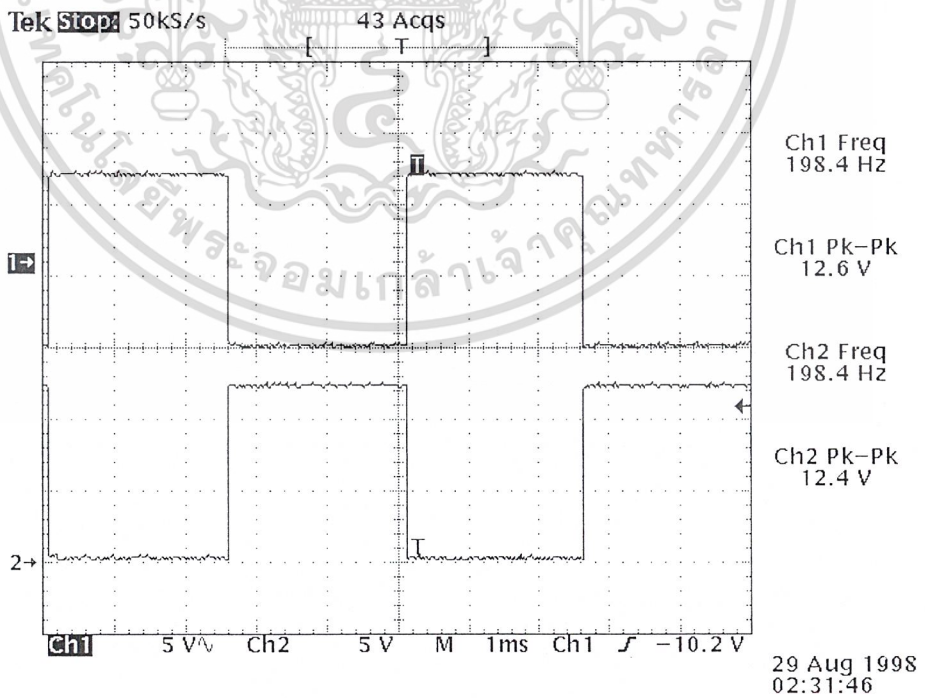
รูปที่ 4.3 แสดงผลการทดลองวงจรสร้างสัญญาณพิก้าและหารความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 วัดผลการทดลองวงจรขยายสัญญาณข้อมูล

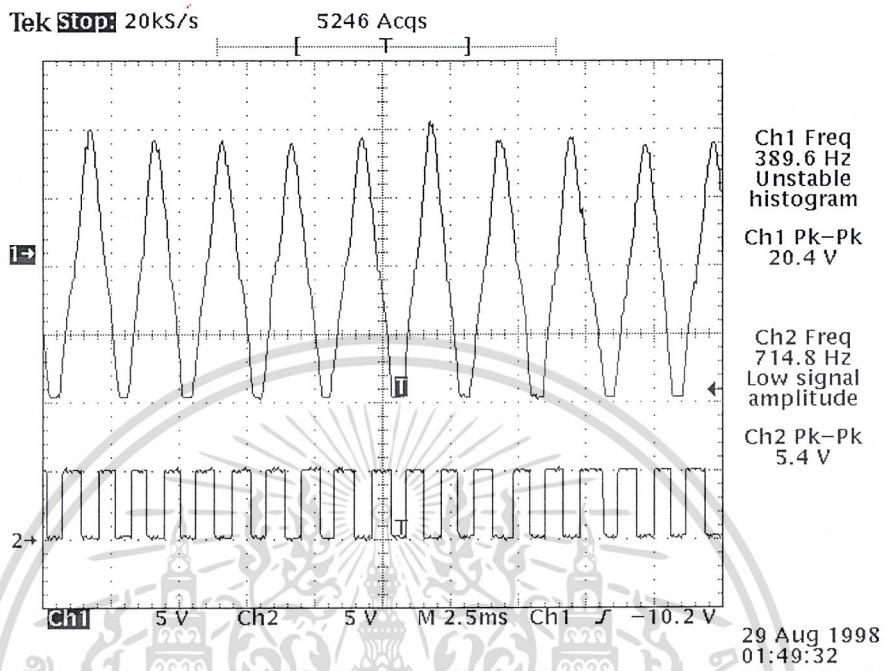


รูปที่ 4.6 แสดงผลการขยายสัญญาณบันทึกข้อมูลดิจิทัลที่ลอจิก 1

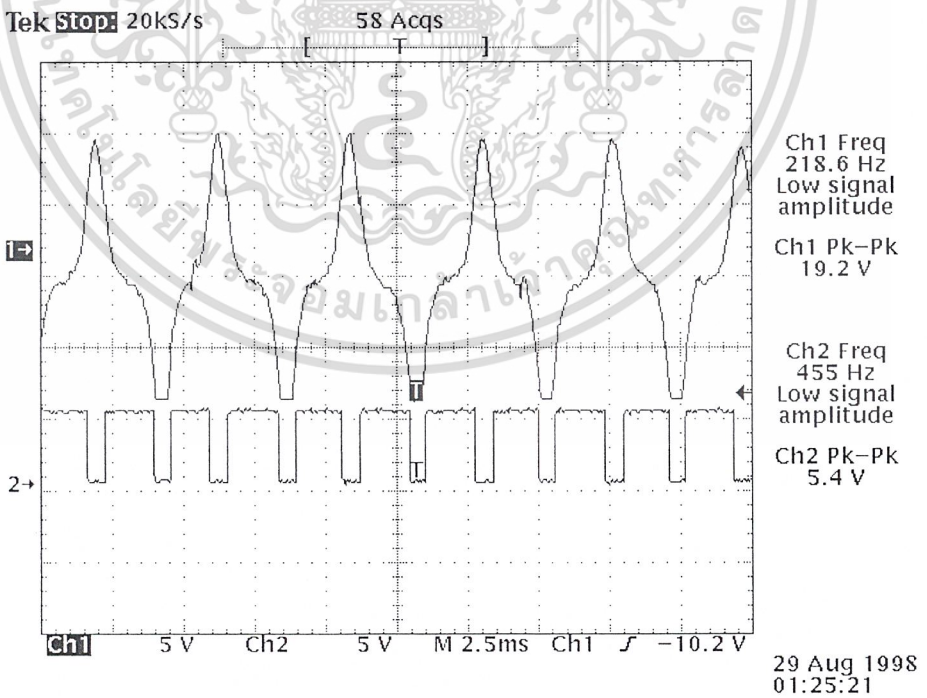


เอกสารนี้เป็นเอกสารที่สงวนไว้ที่ห้ามเผยแพร่หรือใช้ข้อมูลดิจิทัลที่ลอจิก 0 ระเบียบด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.5 นำบัตรที่ถูกบันทึกข้อมูลไปเข้าเครื่องอ่านข้อมูลจากบัตร บันทึกผลการทดลอง



รูปที่ 4.8 แสดงผลการอ่านสัญญาณบันทึกข้อมูลดิจิทัลที่ลอจิก 1

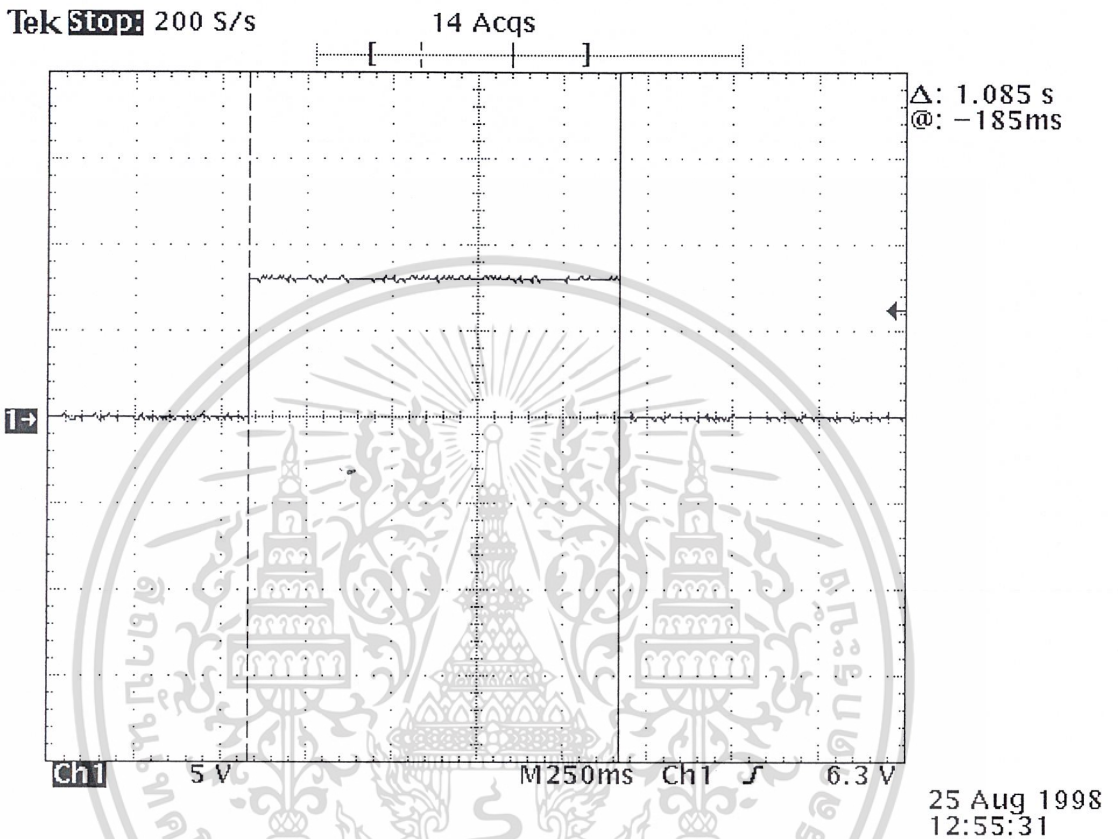


รูปที่ 4.9 แสดงผลการอ่านสัญญาณบันทึกข้อมูลดิจิทัลที่ลอจิก 0

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการเป็นเจ้าของเพื่อการค้าเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาตด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ส่วนของการควบคุมมอเตอร์

4.3.1 วัดผลการทดลองส่วนของวงจร โมโนสเตเบิล



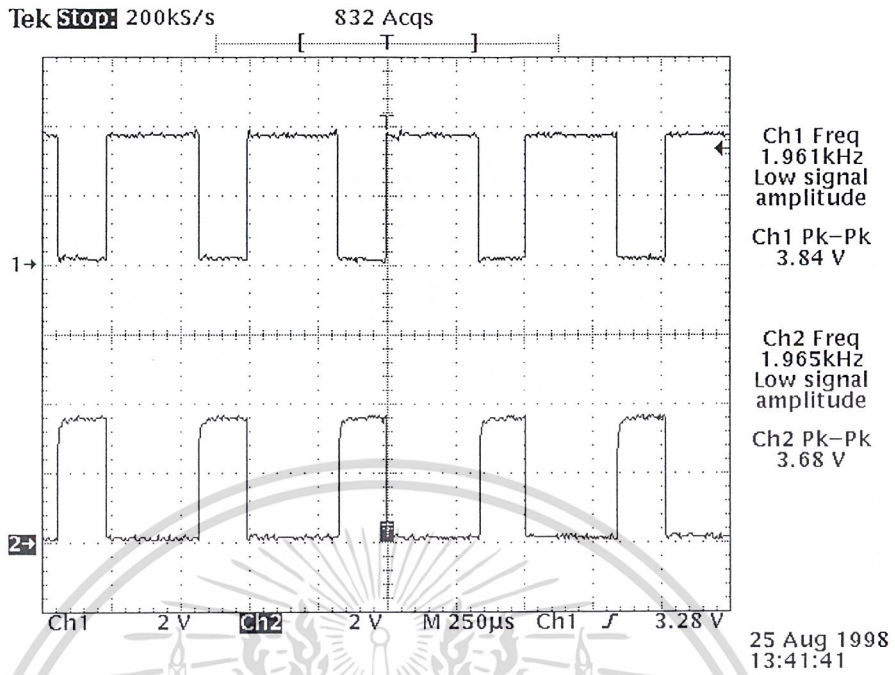
รูปที่ 4.10 แสดงผลการทดลองวงจร โมโนสเตเบิล

จากรูปที่ 4.10 จะเห็นได้ว่าผลการทดลองที่ได้คือ ช่วงเวลาที่ใช้ในการหน่วงเวลาในการนำบัตรออก จะใช้เวลาประมาณ 1.085 วินาที

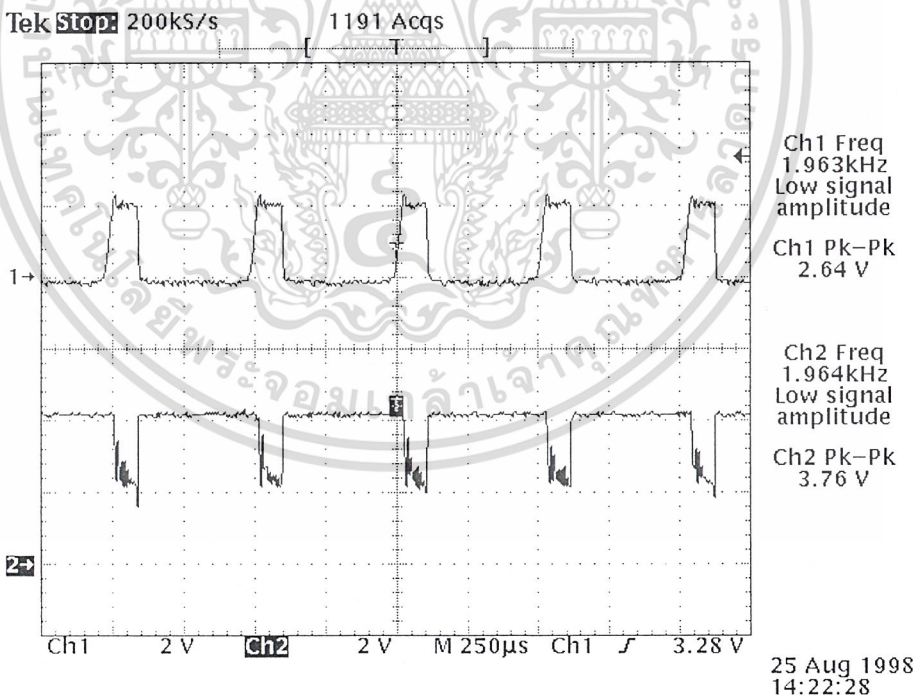
4.4 ส่วนของการตรวจจับบัตร

- 4.4.1 วัดผลการทดลองวงจรผลิตความถี่
- 4.4.2 วัดผลการทดลองวงจรขับแอลอีดีอินฟาเรด
- 4.4.3 วัดผลการทดลองการตรวจจับสัญญาณแสงของโฟโต้ไดโอด
- 4.4.4 วัดผลการทดลองวงจรขยายสัญญาณจากโฟโต้ไดโอด
- 4.4.5 วัดผลการทดลองวงจรตรวจจับบัตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

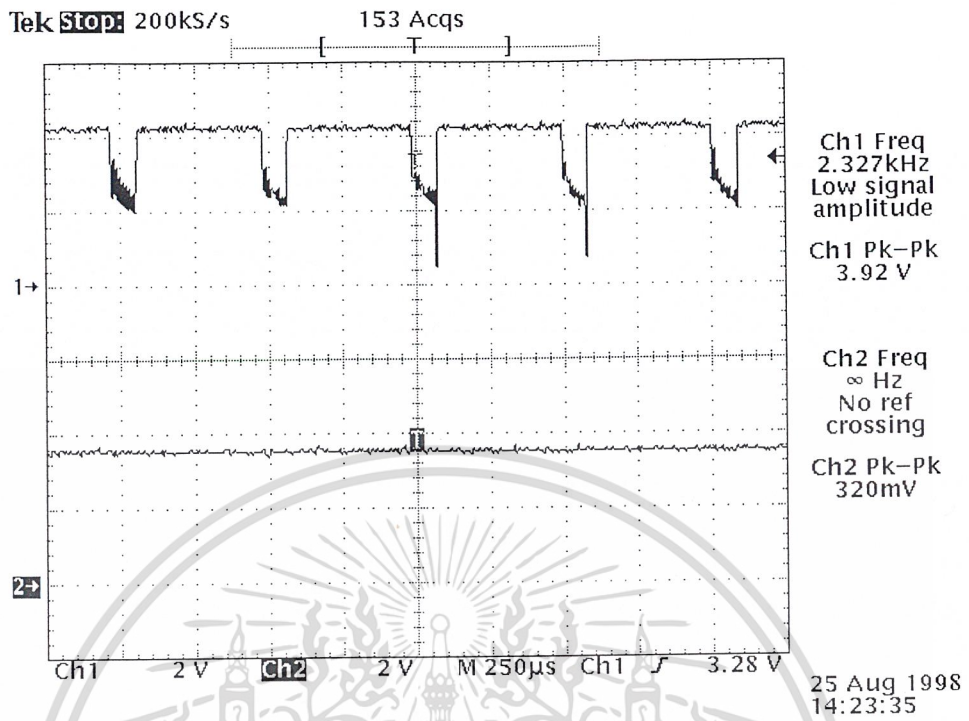


รูปที่ 4.11 แสดงผลการทดลองวงจรผลิตความถี่และขับแอลอีดีอินฟราเรด



รูปที่ 4.12 แสดงผลการทดลองวงจรขยายสัญญาณจากโฟโต้ไดโอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 แสดงผลการทดลองวงจรตรวจจับบัตร

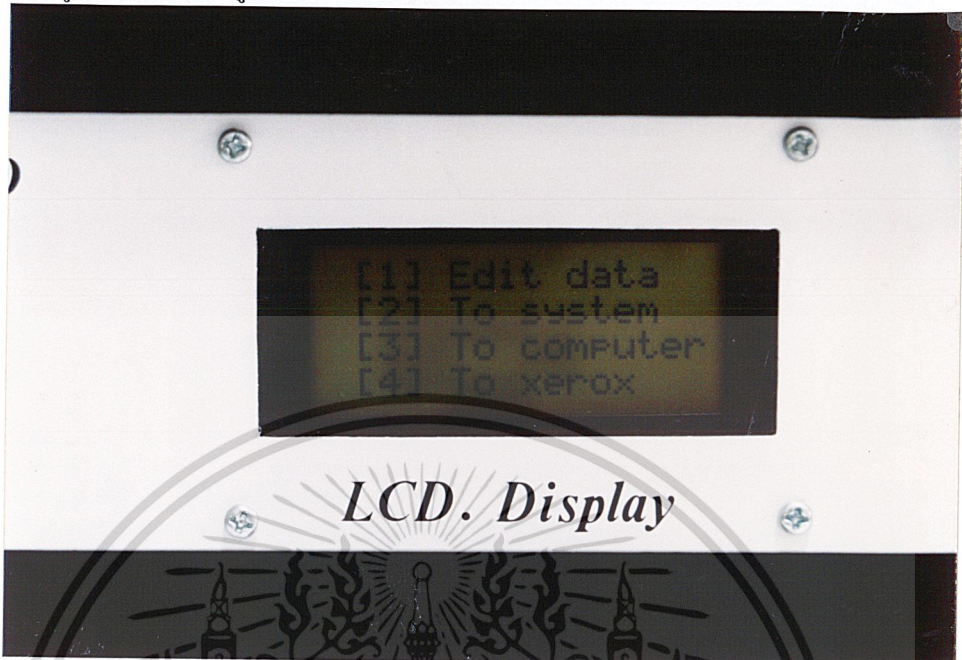
4.5 ส่วนควบคุม

4.5.1 ผลการแสดงชื่อโปรเจ็กต์ผ่านทางจอแอลซีดี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 4.14 แสดงชื่อโปรเจ็กต์ผ่านทางจอแอลซีดี
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.2 คู่มือการแสดงผลเมนูหลักผ่านทางจอแอลซีดี



รูปที่ 4.15 แสดงผลเมนูหลักในการเข้าระบบ

4.5.3 คู่มือการกดคีย์เลือกเมนู (1) จะผ่านการกดรหัสผ่านของระบบ

4.5.4 หลังจากนั้นระบบจะให้ใส่จำนวนกระดาษที่ต้องการในจำนวนจำกัด ถ้าไม่ต้องการให้กดให้กด ESC ออกสู่เมนูถัดไป

4.5.5 หลังใส่จำนวนกระดาษแล้วจะเข้าสู่ระบบย่อยของการแก้ไขข้อมูล

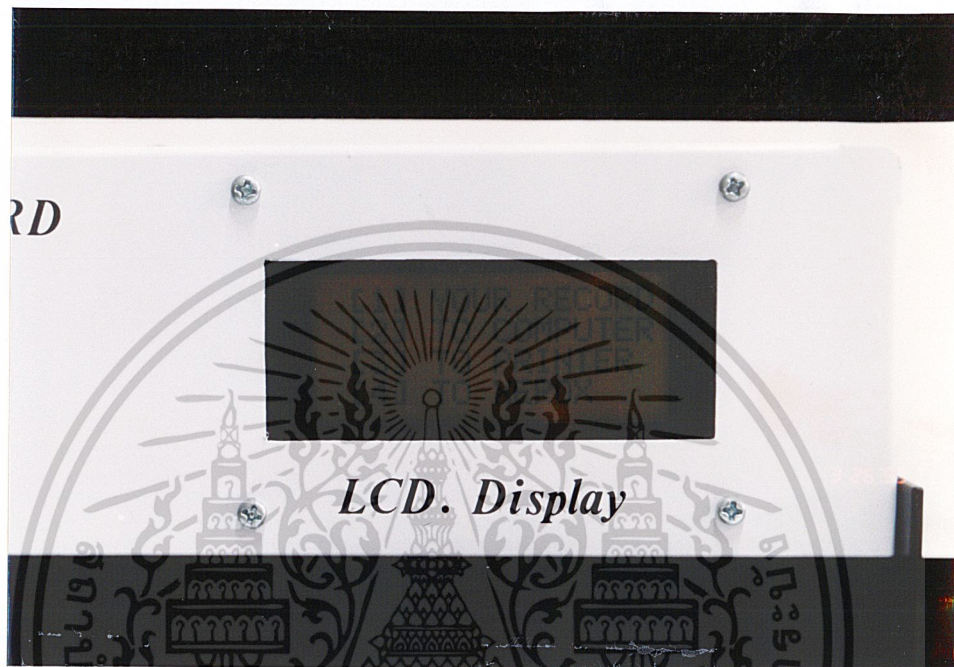


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการรูปที่ 4.16 แสดงผลเมนูการแก้ไขข้อมูลให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.6 ถ้าเลือกเมนู (2) ของเมนูหลักจะแสดงข้อความ “Please insert your card .” ให้นำบัตรสอดเข้าเครื่องเพื่อเข้าสู่ระบบ

4.5.7 ถ้าการอ่านข้อมูลของบัตรถูกต้องจะให้ใส่รหัสผ่านของผู้ถือบัตร

4.5.8 เมื่อใส่รหัสผ่านถูกต้อง จึงจะสามารถเข้าสู่ระบบเลือกใช้บริการต่างๆได้

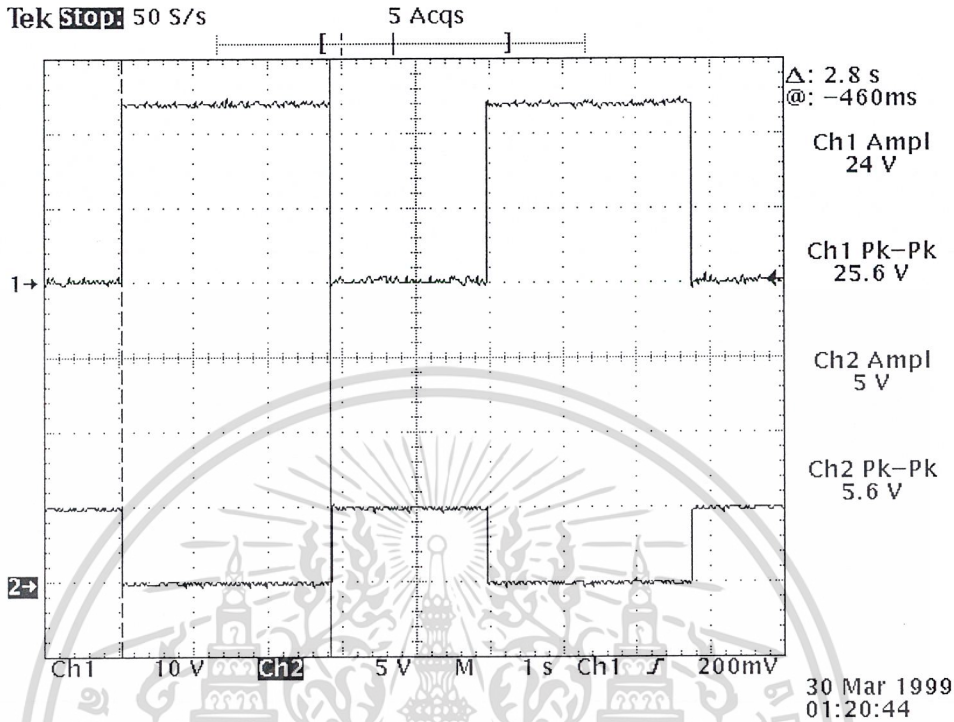


รูปที่ 4.17 แสดงเมนูเลือกการให้บริการ

4.5.9 ถ้าเลือกเมนู (3) ของเมนูหลักจะแสดงการส่งข้อมูลไปยังคอมพิวเตอร์ทุกครั้งที่เครื่องอ่านรหัสบัตรได้ถูกต้องแล้ว ระบบจะแสดงข้อความ “ Send data COMPLETE ”

4.5.10 ถ้าเลือกเมนู (4) ของเมนูหลักจะแสดงข้อความ “ WELCOME TO XEROX MACHINES ” ซึ่งในขณะนี้สัญญาณในการนับกระดาษของเครื่องถ่ายเอกสารจะมีสถานะลोजิก “ 0 ” เมื่อมีการถ่ายเอกสารในทุก 1 ครั้ง สัญญาณในการนับกระดาษจะมีค่าเป็นลोजิก “1 ” ระยะเวลาประมาณ 2.8 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



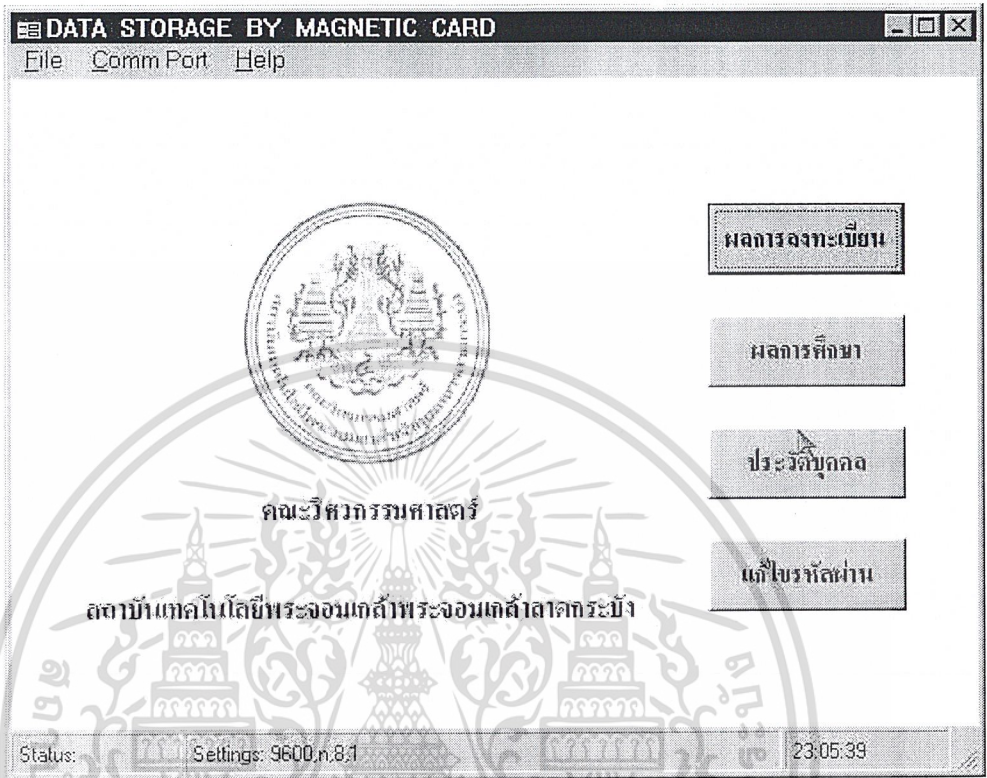
รูปที่ 4.18 สัญญาณที่ใช้ในการนับจำนวนกระดาษ

จากรูปที่ 4.18 จะเห็นได้ว่าสัญญาณที่เครื่องถ่ายเอกสารใช้ในการนับจำนวนกระดาษนั้นจะมีค่าแรงดันสูงประมาณ 24 โวลต์ ดังนั้นยังไม่สามารถให้ไมโครคอนโทรลเลอร์ควบคุมได้ จึงต้องปรับแรงดันให้เป็นระดับ TTL ประมาณ 5 โวลต์ แล้วใช้ซอฟต์แวร์ในการควบคุมอีกทีหนึ่ง

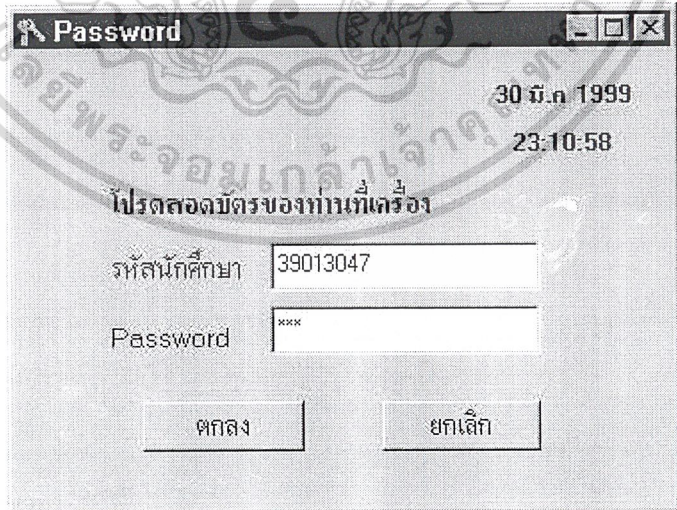
4.6 ส่วนแสดงผลทางคอมพิวเตอร์

- 4.6.1 แสดงการทำงานในส่วนของหน้าจอหลัก
- 4.6.2 แสดงการทำงานในส่วนการเข้าสู่ระบบ
- 4.6.3 แสดงการทำงานในส่วนแสดงผลการลงทะเบียน
- 4.6.4 แสดงการทำงานในส่วนแสดงผลการศึกษา
- 4.6.5 แสดงการทำงานในส่วนการค้นหาประวัติบุคคล
- 4.6.6 แสดงการทำงานในการแก้ไขรหัสผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 แสดงหน้าจอหลักสำหรับการให้บริการ



รูปที่ 4.20 แสดงหน้าจอสำหรับการเข้าสู่ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการลงทะเบียน

รหัสนักศึกษา 39013047

ชื่อ นามสกุล ดิษสุตธิ์

ภาควิชา วิศวกรรมโทรคมนาคม

ปีการศึกษา 41 เทอม 1

รายวิชาที่ลงทะเบียน

รหัสวิชา	รายวิชา	หน่วยกิต	สถานะ
01173103	Satellite Communication	3	N
01173104	Optical Communication	3	Y
01173105	Data Communication	3	Y
01173108	Telecommunication Project I	3	Y
03100002	Industrial Economics	2	Y

หมายเหตุ

Y = ลงทะเบียน	หน่วยกิตที่ลงทะเบียน	11	68
N = ไม่ได้ลงทะเบียน	หน่วยกิตที่ไม่ได้ลงทะเบียน	3	3
	หน่วยกิตรวมทั้งหมด	14	71

← ตกลง พิมพ์

รูปที่ 4.21 แสดงรายงานผลการลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการศึกษา

รหัสนักศึกษา 39013065

ชื่อ นศณศักดิ์ นามสกุล ภาคชาเจียก

ภาควิชา วิศวกรรมโทรคมนาคม

ปีการศึกษา 40 เทอม 2

ผลการศึกษา

รายวิชา	รายวิชา	หน่วยกิต	เกรด
01171102	Telex & Telephone Engineering (3)	3	B
01171103	Broadcasting Engineering (3)	3	A
01172104	Communication Network & Transmission Lines (3)	3	A
01172106	Telecommunication Laboratory II (2)	2	B+
01172107	Microprocessor Application (3)	3	A

เกรดเฉลี่ย หน่วยกิต

ต่อเทอม 3.71 14

ทุกเทอม 3.49 57

ตกลง พิมพ์

รูปที่ 4.22 แสดงรายงานผลการศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค้นหา

โปรดเลือกหัวข้อที่ต้องการค้นหา

หัวข้อ

รหัสนักศึกษา

ชื่อ

ใส่ข้อมูลที่ต้องการค้นหา

39013063

ตกลง ยกเลิก

ประวัติบุคคล

ประวัติบุคคล

รหัสนักศึกษา 39013063

ชื่อ สุริยะ นามสกุล ฉายคุณรัมย์

คณะ วิศวกรรมศาสตร์

ภาควิชา วิศวกรรมโทรคมนาคม

วันเกิด 15 มิถุนายน 2518

ที่อยู่ 82 ต. กาญจนวานิช ม. 8 ต. บ้านพรุ

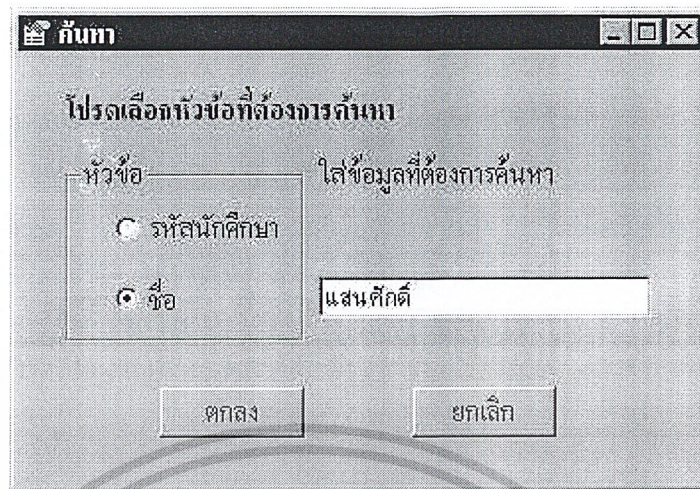
เขตอำเภอ หาดใหญ่ จังหวัด สงขลา

โทรศัพท์ (074)291-217

ค้นหา พิมพ์

รูปที่ 4.23 แสดงหน้าจอการค้นหาประวัติบุคคลแบบป้อนรหัสประจำตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ค้นหา

โปรดเลือกหัวข้อที่ต้องการค้นหา

หัวข้อ

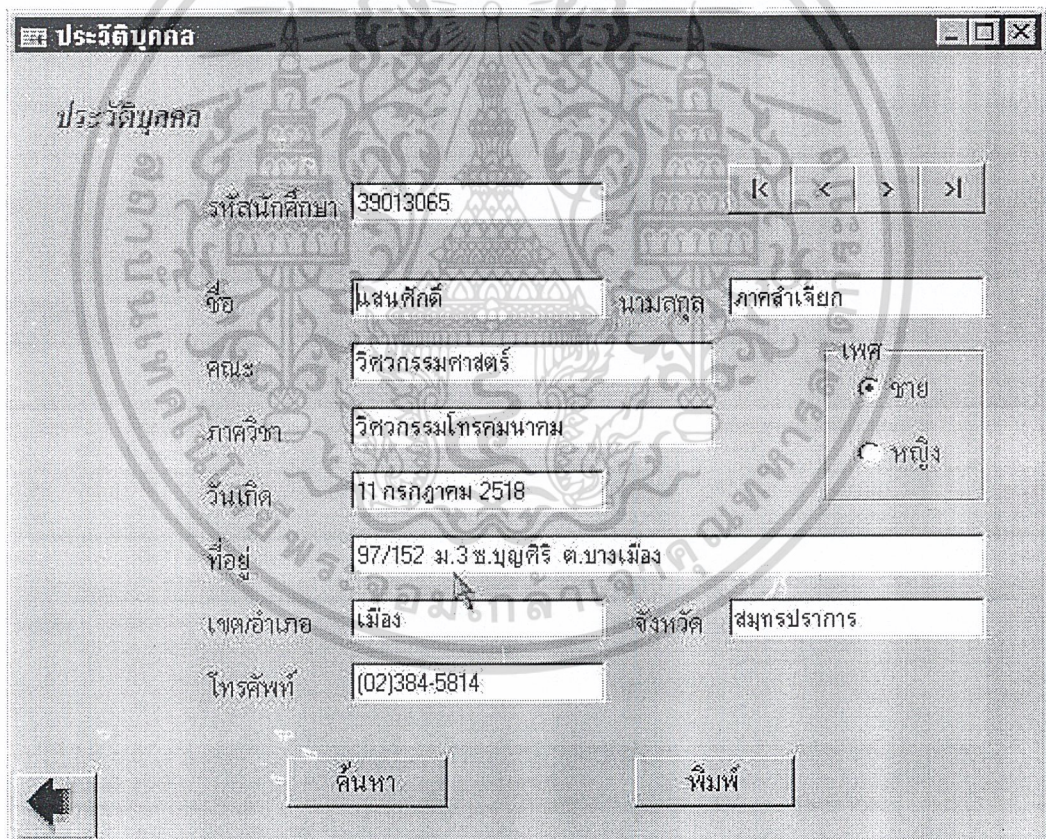
รหัสนักศึกษา

ชื่อ

ใส่ข้อมูลที่ต้องการค้นหา

แสนศักดิ์

ตกลง ยกเลิก



ประวัติบุคคล

ประวัติบุคคล

รหัสนักศึกษา 39013065

ชื่อ แสนศักดิ์ นามสกุล ภาคสาเจียง

คณะ วิศวกรรมศาสตร์

ภาควิชา วิศวกรรมโทรคมนาคม

วันเกิด 11 กรกฎาคม 2518

ที่อยู่ 97/152 ม.3 ซ.บุญศิริ ต.บางเมือง

เขตอำเภอ เมือง จังหวัด สมุทรปราการ

โทรศัพท์ (02)384-5814

← ค้นหา พิมพ์

รูปที่ 4.24 แสดงหน้าจอการค้นหาประวัติบุคคลแบบป้อนชื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.25 แสดงหน้าจอการแก้ไขรหัสผ่าน

สำหรับการติดต่อกับเครื่องพิมพ์ ในส่วนของระบบซึ่งใช้ไมโครคอนโทรลเลอร์จะติดต่อกับเครื่องพิมพ์แบบดอตเมตริกซ์ จากรูปที่ 4.25 เป็นการแสดงตัวอย่างการพิมพ์รายงานข้อมูลการใช้บริการ

*** DATA STORAGE BY MAGNETIC CARD ***

DESCRIPTION.

ID. NO. 10119404 AMOUNT OF SERVICE 02 TIMES

DATE IN	TIME IN	DATE OUT	TIME OUT	PAPER USED
07/02/99	23:02	07/02/99	23:19	0005
07/02/99	23:20	07/02/99	23:23	0005

AMOUNT OF PAPER USED 0010 SHEETS

รูปที่ 4.26 ตัวอย่างการพิมพ์แสดงผลข้อมูลการใช้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 บทวิจารณ์และบทสรุป

5.1 สรุปผลการทดลอง

-เครื่องอ่านข้อมูลจากบัตรแม่เหล็ก

ในส่วนของเครื่องอ่านข้อมูลจากบัตรที่ได้ทำการออกแบบและทำการสร้างเมื่อนำมาทดลองและทดสอบเครื่อง โดยนำบัตรแม่เหล็กเอทีเอ็มของธนาคารมาทำการทดสอบ ซึ่งได้ผลของการอ่านรหัสข้อมูลได้ถูกต้องทำให้สามารถนำบัตรแม่เหล็กมาประยุกต์ใช้งานได้

-เครื่องบันทึกข้อมูลลงบัตรแม่เหล็ก

ในส่วนของเครื่องบันทึกข้อมูลลงบัตร ได้ทำการทดลองนำแถบแม่เหล็กของม้วนวีดีโอ มาทดสอบในการทดลองบันทึกข้อมูลลงบนแถบแม่เหล็ก ซึ่งได้ทดลองบันทึกข้อมูลที่เป็นลอจิก 1 และ 0 ทั้งหมด สามารถบันทึกข้อมูลลงบนแถบแม่เหล็กได้ แต่เมื่อทำการทดสอบลองอ่านข้อมูลจากบัตรที่ทำการทดลองบันทึกผลของสัญญาณที่ได้จะมีขนาดต่ำกว่าบัตรของธนาคาร จึงต้องเพิ่มอัตราการขยายของเครื่องอ่าน จึงจะอ่านข้อมูลจากบัตรบันทึกได้ถูกต้องและเมื่อนำบัตรแถบชนิดพลาสติกมาทดลองปรากฏว่าสามารถบันทึกรหัสข้อมูลได้ แต่ยังมีบางโอกาสที่การบันทึกมีความผิดพลาดเกิดขึ้น

-ส่วนควบคุม

ได้ใช้ไมโครคอนโทรลเลอร์ เบอร์ AT89S8252 ของบริษัทแอตเมล (ATMEL) ซึ่งเป็นไอซีในตระกูลเดียวกับ MCS-51 ซึ่งการโปรแกรมใช้ชุดคำสั่งเดียวกันกับตระกูล MCS-51 ผลการทำงานของส่วนควบคุมสามารถติดต่อควบคุมกับอุปกรณ์ต่างๆ ได้ โดยมีการแสดงผลการทำงานผ่านทางแอลซีดี

-ส่วนแสดงผลทางคอมพิวเตอร์

ในการแสดงผลทางคอมพิวเตอร์ได้ใช้โปรแกรม Visual Basic ในการติดต่อสื่อสารกับส่วนควบคุมผ่านทางพอร์ทอนุกรม RS-232C โดยเป็นส่วนที่รับข้อมูลจากส่วนควบคุมแล้วนำมาตรวจสอบหรือบันทึกข้อมูลไว้ในคอมพิวเตอร์ โดยจะแบ่งเป็นช่วงเวลาปกติกับช่วงเวลาในการบันทึกข้อมูลซึ่งในช่วงเวลาในการบันทึกจะเป็นช่วงที่ทางด้านระบบจะหยุดให้บริการจนกว่าจะทำการบันทึกข้อมูลเสร็จสิ้นแล้ว

5.2 วิจารณ์ผลการทดลอง

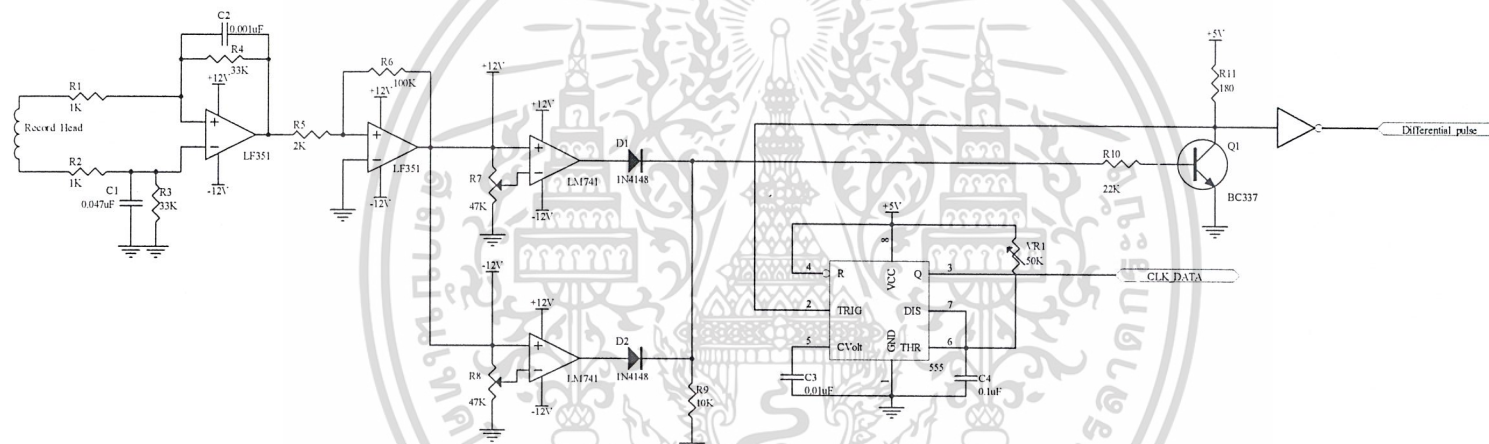
เครื่องอ่านและบันทึกข้อมูลบัตรแม่เหล็กที่ได้ทำนี้สามารถที่จะนำบัตรแม่เหล็กมาประยุกต์ใช้งานได้ ในส่วนหนึ่งซึ่งสามารถที่จะพัฒนาต่อไปควบคุมสิ่งต่างๆซึ่งในเครื่องต้นแบบที่ได้สร้างขึ้น สามารถเก็บบันทึกข้อมูลที่ส่วนควบคุมเพียงอย่างเดียวได้โดยไม่ต้องติดต่อกับคอมพิวเตอร์ตลอดก็ได้ และยังสามารถที่จะนำไปใช้งานในที่ใด ๆ ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

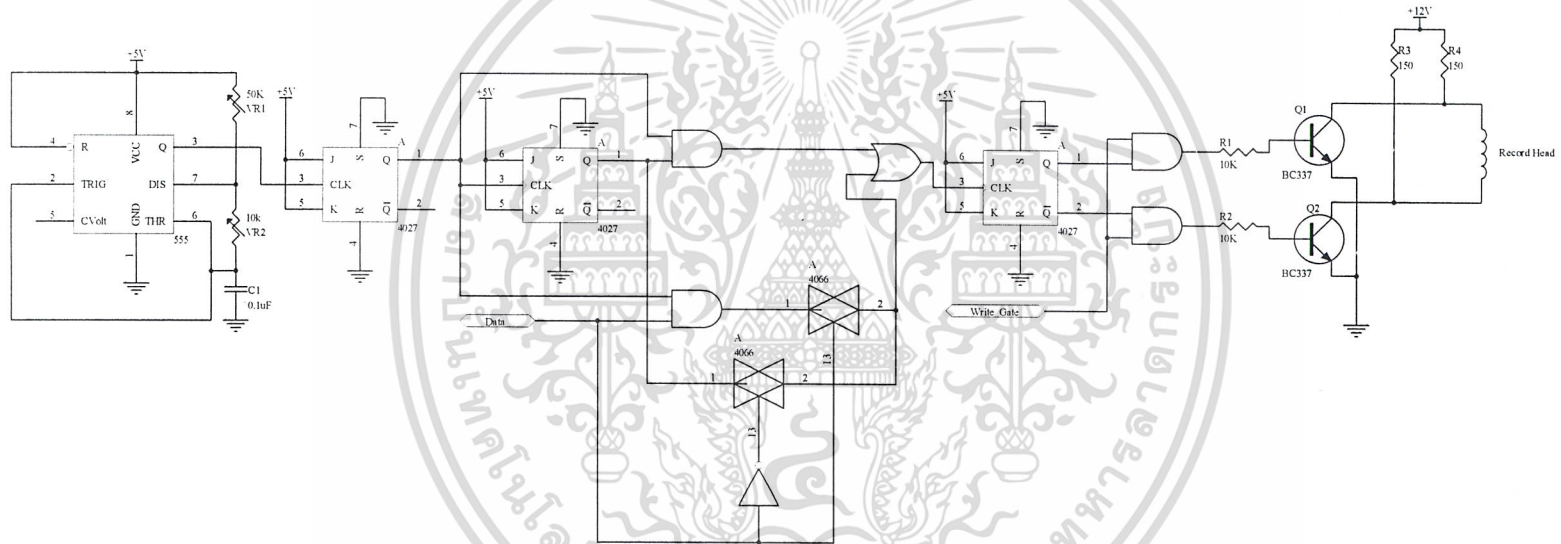


ภาคผนวก

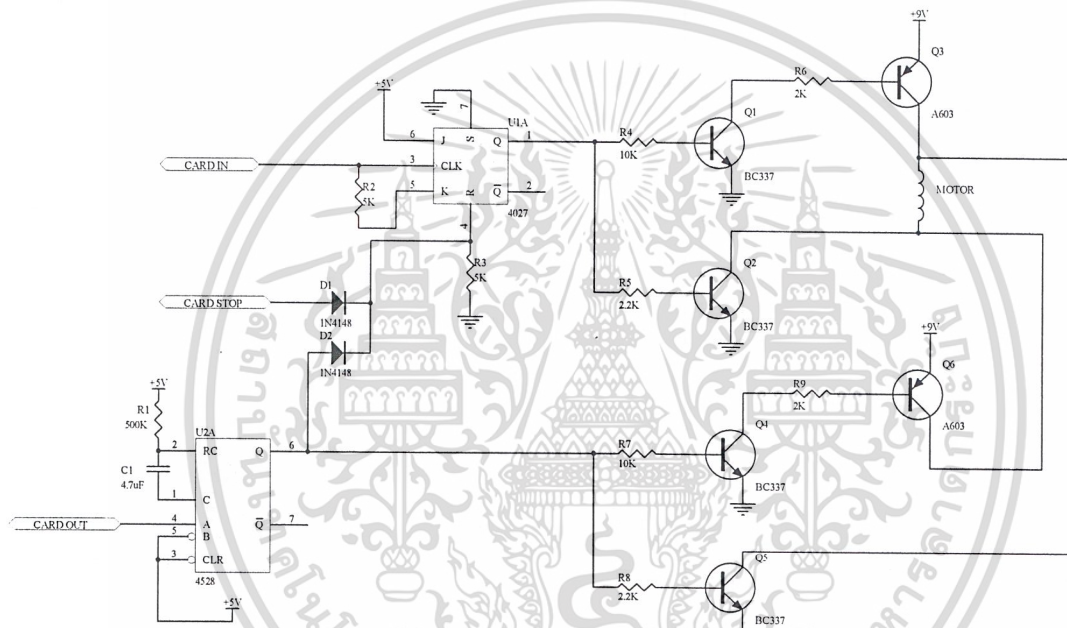
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



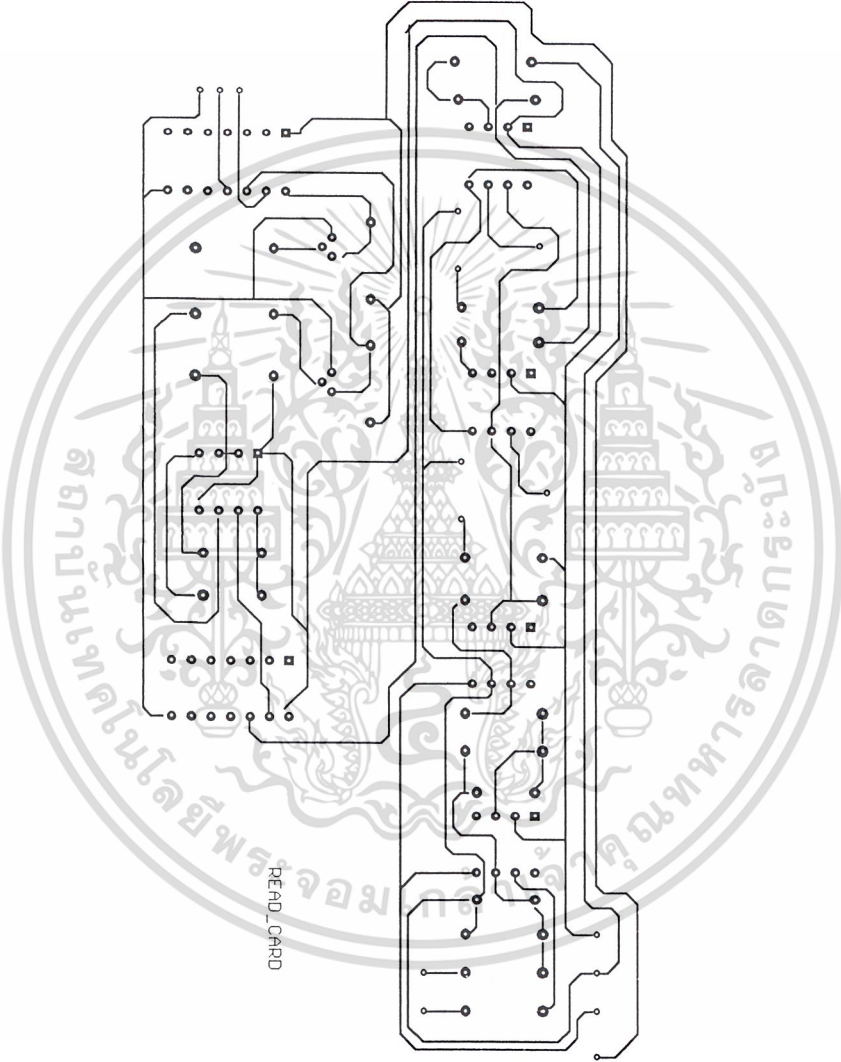
Title		
READ CARD.		
Size	Number	Revision
B		
Date:	L-Apr-1999	Sheet of
File:	A:PROJECTSCH	Drawn By:



Title		
WRITE DATA TO CARD.		
Size	Number	Revision
B		
Date:	1-Apr-1999	Sheet of
File:	A:PROJECT2SCH	Drawn By:

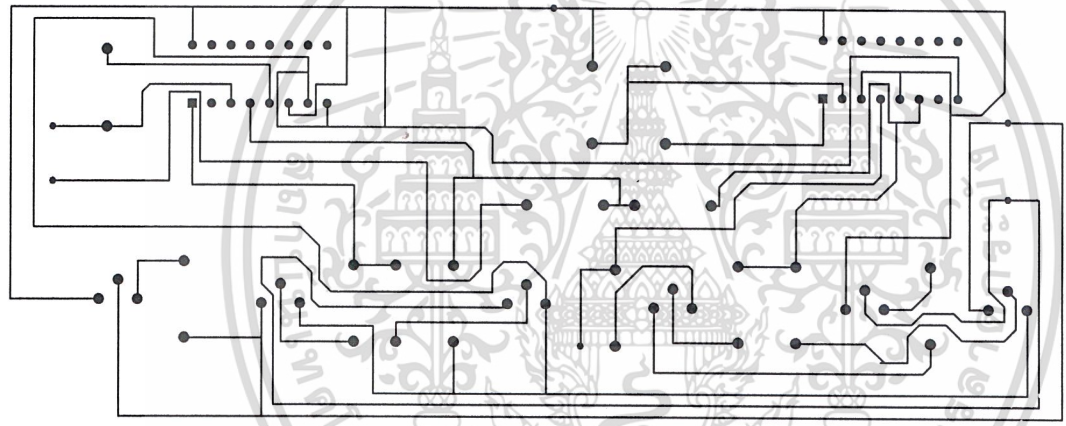


Title		
CONTROL MOTOR.		
Size	Number	Revision
B		
Date:	1-Apr-1999	Sheet of
File:	A:PROJECTS\SCH	Drawn By:

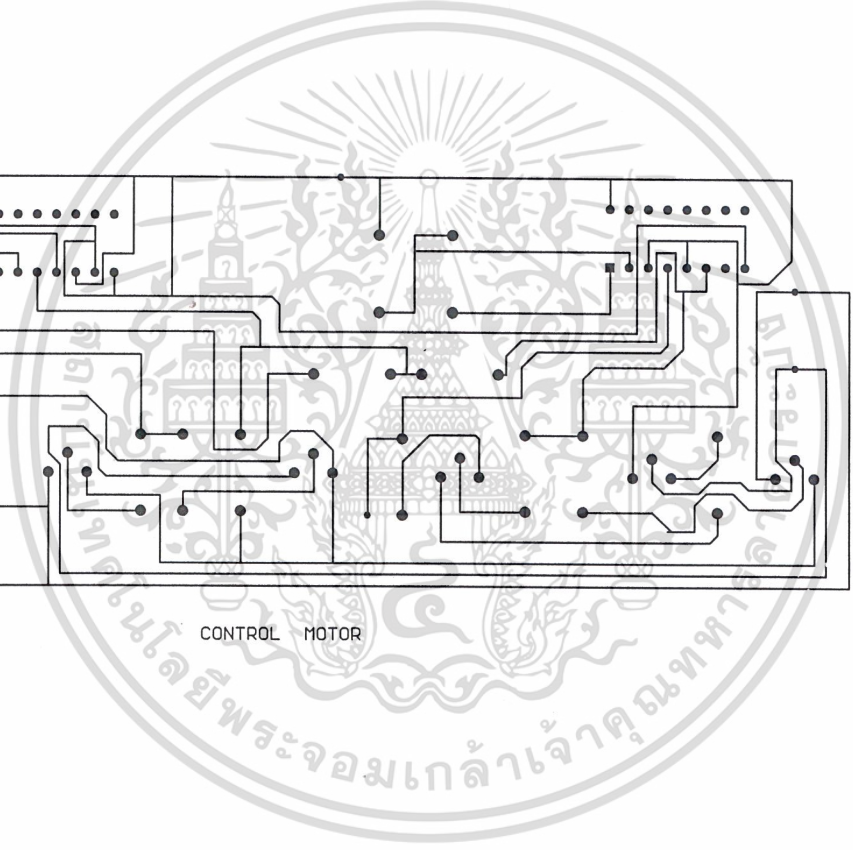


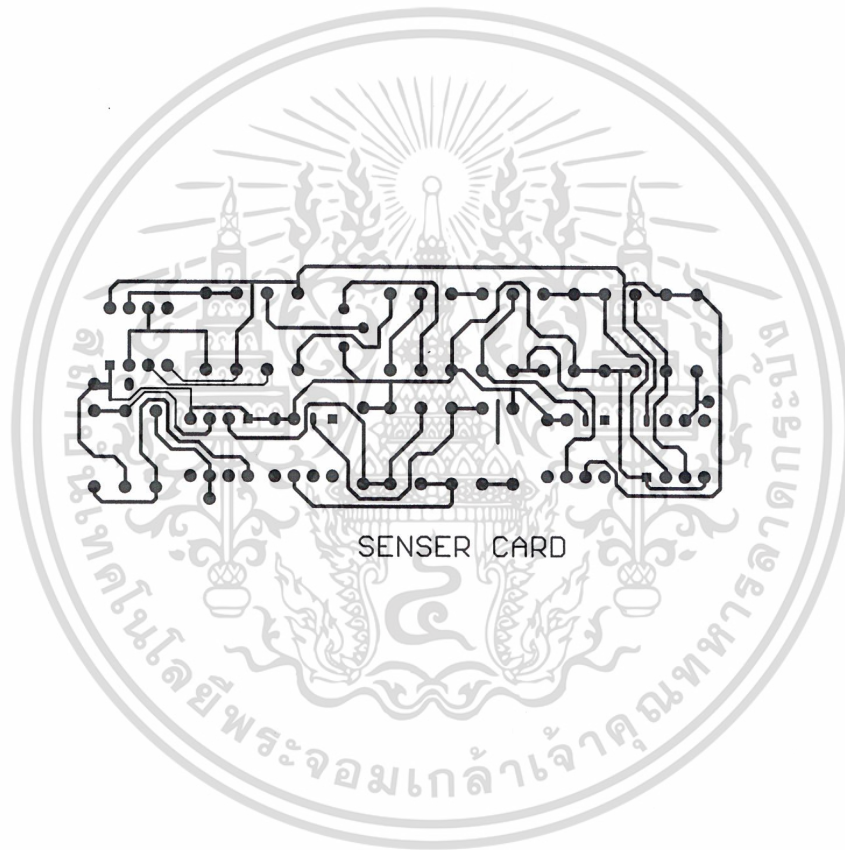


WR_CARD

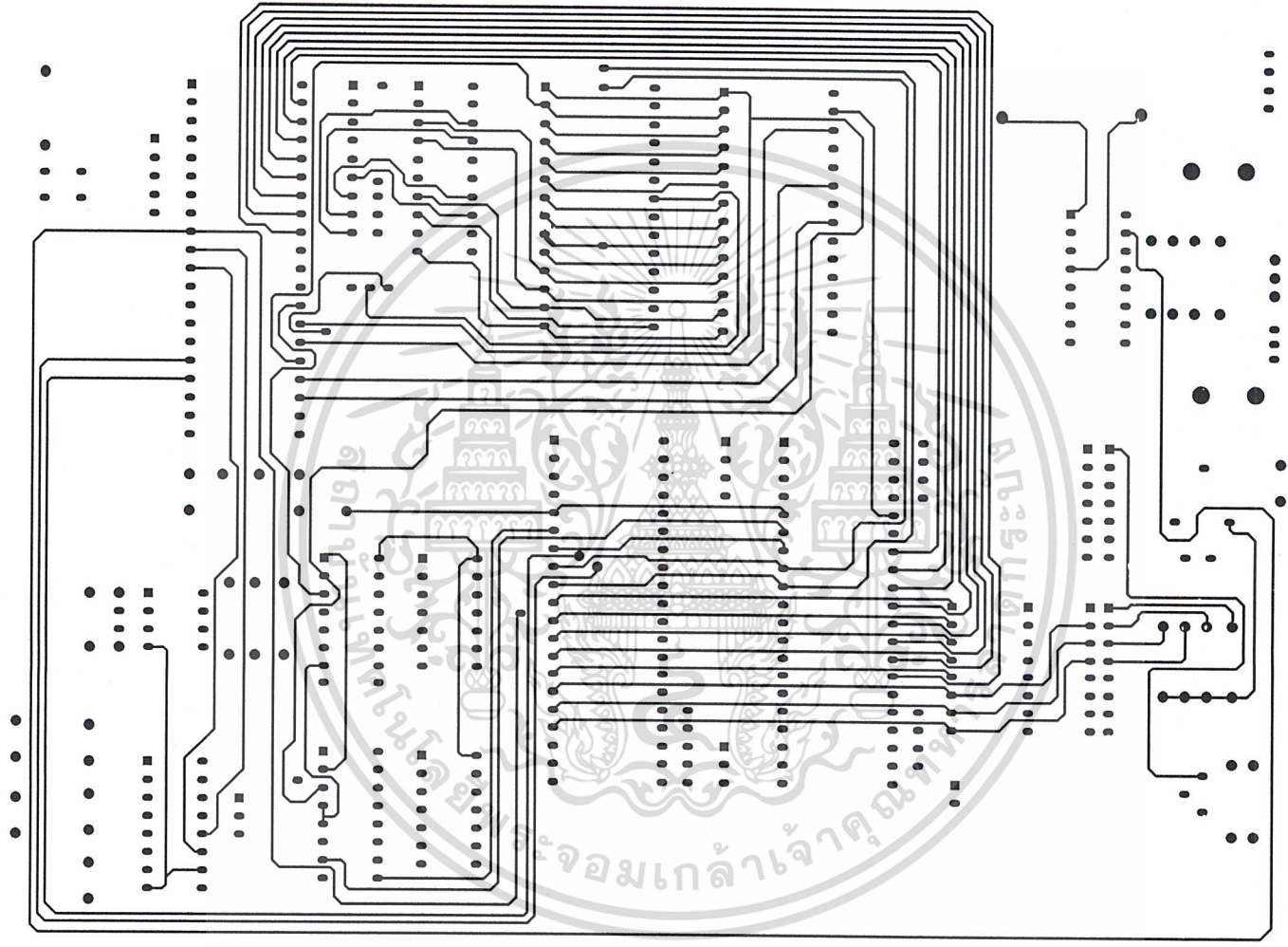


CONTROL MOTOR

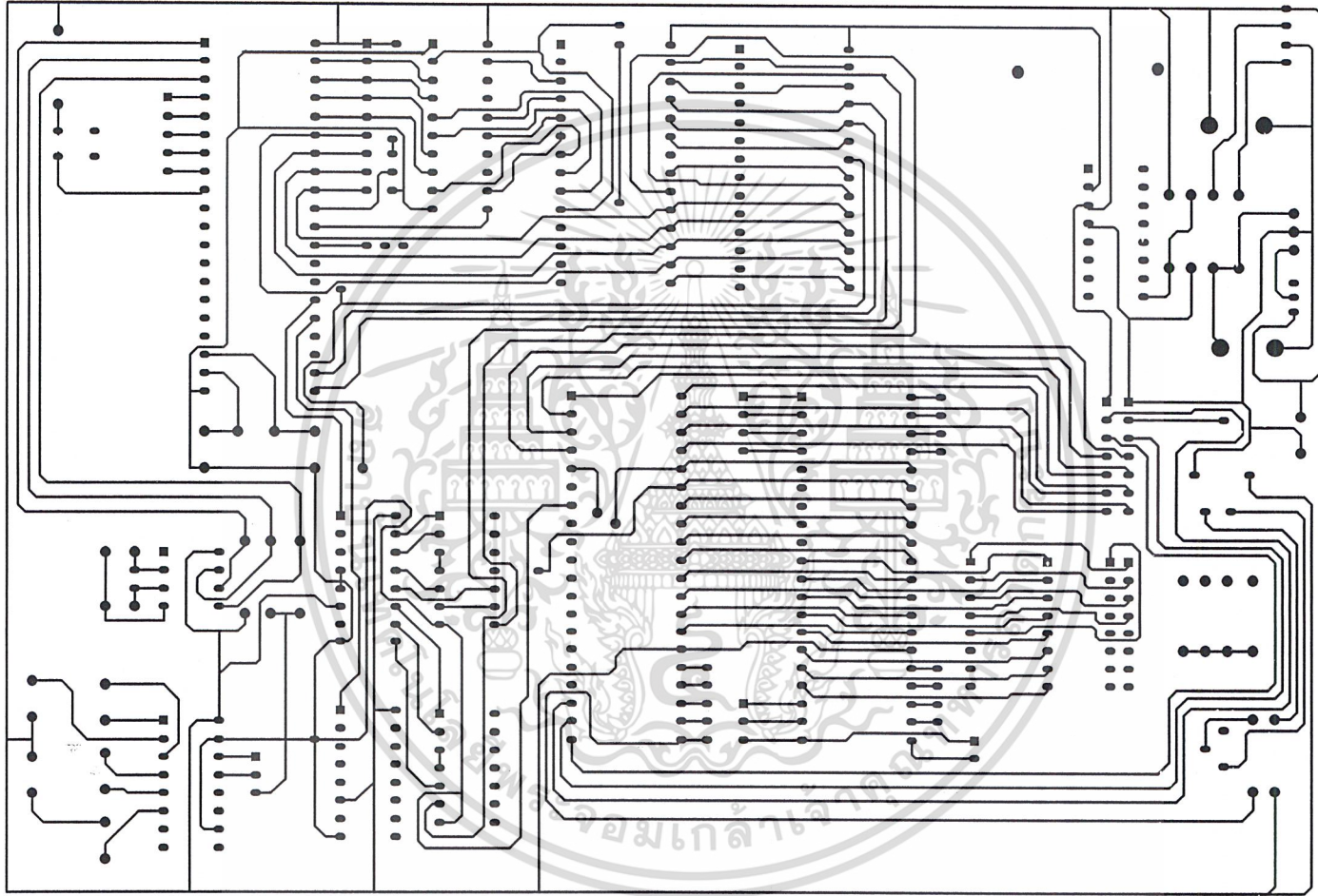




SENSOR CARD



CONTROL SYSTEM.



Features

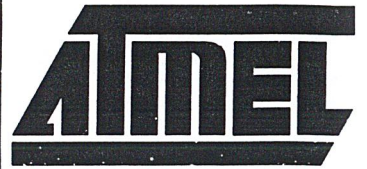
- Compatible with MCS-51™ Products
- 8K Bytes of In-System Reprogrammable Downloadable Flash Memory
 - SPI Serial Interface for Program Downloading
 - Endurance: 1,000 Write/Erase Cycles
- 2K Bytes EEPROM
 - Endurance: 100,000 Write/Erase Cycles
- 4.0V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Nine Interrupt Sources
- Programmable UART Serial Channel
- SPI Serial Interface
- Low Power Idle and Power Down Modes
- Interrupt Recovery From Power Down
- Programmable Watchdog Timer
- Dual Data Pointer
- Power Off Flag

Description

The AT89S8252 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of Downloadable Flash programmable and erasable read only memory and 2K bytes of EEPROM. The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard 80C51 instruction set and pinout. The on-chip Downloadable Flash allows the program memory to be reprogrammed in-system through an SPI serial interface or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Downloadable Flash on a monolithic chip, the Atmel AT89S8252 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

The AT89S8252 provides the following standard features: 8K bytes of Downloadable Flash, 2K bytes of EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog timer, two Data Pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S8252 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

The Downloadable Flash can be changed a single byte at a time and is accessible through the SPI serial interface. Holding RESET active forces the SPI bus into a serial programming interface and allows the program memory to be written to or read from unless Lock Bit 2 has been activated.



**8-Bit
Microcontroller
with 8K Bytes
Flash**

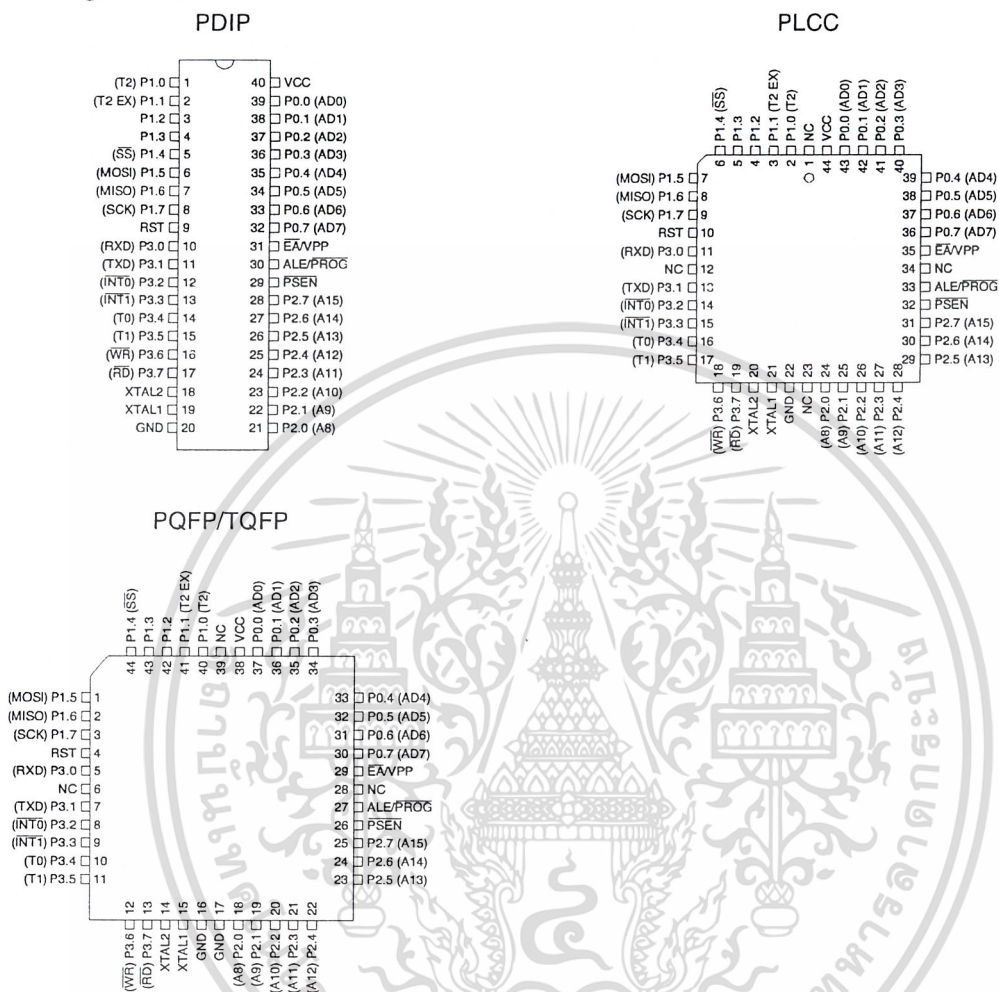
AT89S8252



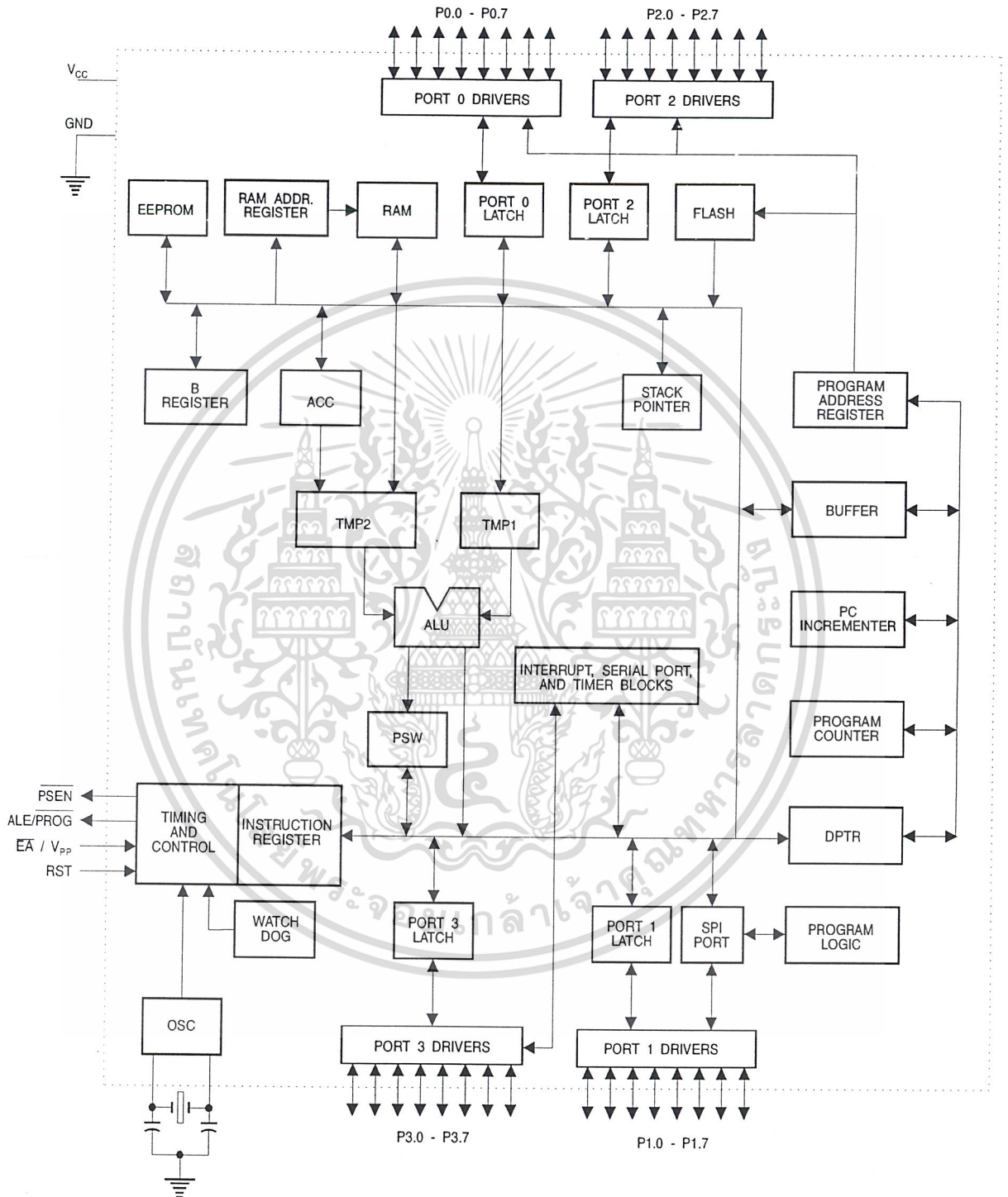
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Pin Configurations



Block Diagram



Furthermore, P1.4, P1.5, P1.6, and P1.7 can be configured as the SPI slave port select, data input/output and shift clock input/output pins as shown in the following table.

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
P1.4	\overline{SS} (Slave port select input)
P1.5	MOSI (Master data output, slave data input pin for SPI channel)
P1.6	MISO (Master data input, slave data output pin for SPI channel)
P1.7	SCK (Master clock output, slave clock input pin for SPI channel)

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8 bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89S8252, as shown in the following table.

Port 3 also receives some control signals for Flash programming and verification.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	\overline{WR} (external data memory write strobe)
P3.7	\overline{RD} (external data memory read strobe)

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/PROG

Address Latch Enable is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (\overline{PROG}) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

PSEN

Program Store Enable is the read strobe to external program memory.

When the AT89S8252 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

\overline{EA}/V_{PP}

External Access Enable. \overline{EA} must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, \overline{EA} will be internally latched on reset.

\overline{EA} should be strapped to V_{CC} for internal program executions. This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming when 12-volt programming is selected.

AT89S8252

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

Table 1. AT89S8252 SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000					SPCR 000001XX			0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			0CFH
0C0H									0C7H
0B8H	IP XX000000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0X000000		SPSR 00XXXXXX						0AFH
0A0H	P2 11111111								0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111						WMCON 00000010		97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000			8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	SPDR XXXXXXXX	PCON 0XXX0000	87H





User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Timer 2 Registers Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 9) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16 bit capture mode or 16-bit auto-reload mode.

Watchdog and Memory Control Register The WMCON register contains control bits for the Watchdog Timer (shown in Table 3). The EEMEN and EEMWE bits are used to select the 2K bytes on-chip EEPROM, and to enable byte-write. The DPS bit selects one of two DPTR registers available.

SPI Registers Control and status bits for the Serial Peripheral Interface are contained in registers SPCR (shown in Table 4) and SPSR (shown in Table 5). The SPI data bits are contained in the SPDR register. Writing the SPI data register during serial data transfer sets the Write Collision bit, WCOL, in the SPSR register. The SPDR is double buffered for writing and the values in SPDR are not changed by Reset.

Interrupt Registers The global interrupt enable bit and the individual interrupt enable bits are in the IE register. In addition, the individual interrupt enable bit for the SPI is in the SPCR register. Two priorities can be set for each of the six interrupt sources in the IP register.

Table 2. T2CON—Timer/Counter 2 Control Register

T2CON Address = 0C8H				Reset Value = 0000 0000B				
Bit Addressable								
	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
Bit	7	6	5	4	3	2	1	0

Symbol	Function
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
RCLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflows to be used for the receive clock.
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
C/T2	Timer or counter select for Timer 2. C/T2 = 0 for timer function. C/T2 = 1 for external event counter (falling edge triggered).
CP/RL2	Capture/Reload select. CP/RL2 = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/RL2 = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

Dual Data Pointer Registers To facilitate accessing both internal EEPROM and external data memory, two banks of 16 bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H-85H. Bit DPS = 0 in SFR WCON selects DP0 and DPS = 1 selects DP1. The user should always initialize the DPS bit to the

appropriate value before accessing the respective Data Pointer Register.

Power Off Flag The Power Off Flag (POF) is located at bit_4 (PCON.4) in the PCON SFR. POF is set to "1" during power up. It can be set and reset under software control and is not affected by RESET.

Table 3. WCON—Watchdog and Memory Control Register

WCON Address = 96H						Reset Value = 0000 0010B		
Bit	PS2	PS1	PS0	EEMWE	EEMEN	DPS	WDTRST	WDTEN
	7	6	5	4	3	2	1	0

Symbol	Function
PS2 PS1 PS0	Prescaler Bits for the Watchdog Timer. When all three bits are set to "0", the watchdog timer has a nominal period of 16 ms. When all three bits are set to "1", the nominal period is 2048 ms.
EEMWE	EEPROM Data Memory Write Enable Bit. Set this bit to "1" before initiating byte write to on-chip EEPROM with the MOVX instruction. User software should set this bit to "0" after EEPROM write is completed.
EEMEN	Internal EEPROM Access Enable. When EEMEN = 1, the MOVX instruction with DPTR will access on-chip EEPROM instead of external data memory. When EEMEN = 0, MOVX with DPTR accesses external data memory.
DPS	Data Pointer Register Select. DPS = 0 selects the first bank of Data Pointer Register, DP0, and DPS = 1 selects the second bank, DP1
WDTRST RDY/BSY	Watchdog Timer Reset and EEPROM Ready/Busy Flag. Each time this bit is set to "1" by user software, a pulse is generated to reset the watchdog timer. The WDTRST bit is then automatically reset to "0" in the next instruction cycle. The <u>WDTRST</u> bit is Write-Only. This bit also serves as the RDY/BSY flag in a Read-Only mode during EEPROM write. RDY/BSY = 1 means that the EEPROM is ready to be programmed. While programming operations are being executed, the RDY/BSY bit equals "0" and is automatically reset to "1" when programming is completed.
WDTEN	Watchdog Timer Enable Bit. WDTEN = 1 enables the watchdog timer and WDTEN = 0 disables the watchdog timer.





Table 4. SPCR—SPI Control Register

SPCR Address = D5H					Reset Value = 0000 01XXB			
	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
Bit	7	6	5	4	3	2	1	0

Symbol	Function															
SPIE	SPI Interrupt Enable. This bit, in conjunction with the ES bit in the IE register, enables SPI interrupts: SPIE = 1 and ES = 1 enable SPI interrupts. SPIE = 0 disables SPI interrupts.															
SPE	SPI Enable. SPI = 1 enables the SPI channel and connects \overline{SS} , MOSI, MISO and SCK to pins P1.4, P1.5, P1.6, and P1.7. SPI = 0 disables the SPI channel.															
DORD	Data Order. DORD = 1 selects LSB first data transmission. DORD = 0 selects MSB first data transmission.															
MSTR	Master/Slave Select. MSTR = 1 selects Master SPI mode. MSTR = 0 selects Slave SPI mode.															
CPOL	Clock Polarity. When CPOL = 1, SCK is high when idle. When CPOL = 0, SCK of the master device is low when not transmitting. Please refer to figure on SPI Clock Phase and Polarity Control.															
CPHA	Clock Phase. The CPHA bit together with the CPOL bit controls the clock and data relationship between master and slave. Please refer to figure on SPI Clock Phase and Polarity Control.															
SPR0 SPR1	SPI Clock Rate Select. These two bits control the SCK rate of the device configured as master. SPR1 and SPR0 have no effect on the slave. The relationship between SCK and the oscillator frequency, F_{OSC} , is as follows: <table border="1" style="margin-left: 20px;"> <tr> <td>SPR1</td> <td>SPR0</td> <td>SCK = F_{OSC} divided by</td> </tr> <tr> <td>0</td> <td>0</td> <td>4</td> </tr> <tr> <td>0</td> <td>1</td> <td>16</td> </tr> <tr> <td>1</td> <td>0</td> <td>64</td> </tr> <tr> <td>1</td> <td>1</td> <td>128</td> </tr> </table>	SPR1	SPR0	SCK = F_{OSC} divided by	0	0	4	0	1	16	1	0	64	1	1	128
SPR1	SPR0	SCK = F_{OSC} divided by														
0	0	4														
0	1	16														
1	0	64														
1	1	128														

Table 5. SPSR—SPI Status Register

SPSR Address = AAH					Reset Value = 00XX XXXXB			
	SPIF	WCOL	—	—	—	—	—	—
Bit	7	6	5	4	3	2	1	0

Symbol	Function
SPIF	SPI Interrupt Flag. When a serial transfer is complete, the SPIF bit is set and an interrupt is generated if SPIE = 1 and ES = 1. The SPIF bit is cleared by reading the SPI status register with SPIF and WCOL bits set, and then accessing the SPI data register.
WCOL	Write Collision Flag. The WCOL bit is set if the SPI data register is written during a data transfer. During data transfer, the result of reading the SPDR register may be incorrect, and writing to it has no effect. The WCOL bit (and the SPIF bit) are cleared by reading the SPI status register with SPIF and WCOL set, and then accessing the SPI data register.

Table 6. SPDR—SPI Data Register

SPDR Address = 86H					Reset Value = unchanged			
	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
Bit	7	6	5	4	3	2	1	0

Data Memory—EEPROM and RAM

The AT89S8252 implements 2K bytes of on-chip EEPROM for data storage and 256 bytes of RAM. The upper 128 bytes of RAM occupy a parallel space to the Special Function Registers. That means the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the address mode used in the instruction specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions that use direct addressing access SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

```
MOV 0A0H, #data
```

Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

```
MOV @R0, #data
```

Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

The on-chip EEPROM data memory is selected by setting the EEMEN bit in the WMCON register at SFR address location 96H. The EEPROM address range is from 000H to 7FFH. The MOVX instructions are used to access the EEPROM. To access off-chip data memory with the MOVX instructions, the EEMEN bit needs to be set to "0".

The EEMWE bit in the WMCON register needs to be set to "1" before any byte location in the EEPROM can be written. User software should reset EEMWE bit to "0" if no further EEPROM write is required. EEPROM write cycles in the serial programming mode are self-timed and typically take 2.5 ms. The progress of EEPROM write can be monitored by reading the RDY/BSY bit (read-only) in SFR WMCON. RDY/BSY = 0 means programming is still in progress and RDY/BSY = 1 means EEPROM write cycle is completed and another write cycle can be initiated.

In addition, during EEPROM programming, an attempted read from the EEPROM will fetch the byte being written with the MSB complemented. Once the write cycle is completed, true data are valid at all bit locations.

Programmable Watchdog Timer

The programmable Watchdog Timer (WDT) operates from an independent oscillator. The prescaler bits, PS0, PS1 and PS2 in SFR WMCON are used to set the period of the Watchdog Timer from 16 ms to 2048 ms. The available timer periods are shown in the following table and the actual timer periods (at V_{CC} = 5V) are within ±30% of the nominal.

The WDT is disabled by Power-on Reset and during Power Down. It is enabled by setting the WDTEN bit in SFR WMCON (address = 96H). The WDT is reset by setting the WDTRST bit in WMCON. When the WDT times out without being reset or disabled, an internal RST pulse is generated to reset the CPU.

Table 7. Watchdog Timer Period Selection

WDT Prescaler Bits			Period (nominal)
PS2	PS1	PS0	
0	0	0	16 ms
0	0	1	32 ms
0	1	0	64 ms
0	1	1	128 ms
1	0	0	256 ms
1	0	1	512 ms
1	1	0	1024 ms
1	1	1	2048 ms



Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 5. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency.

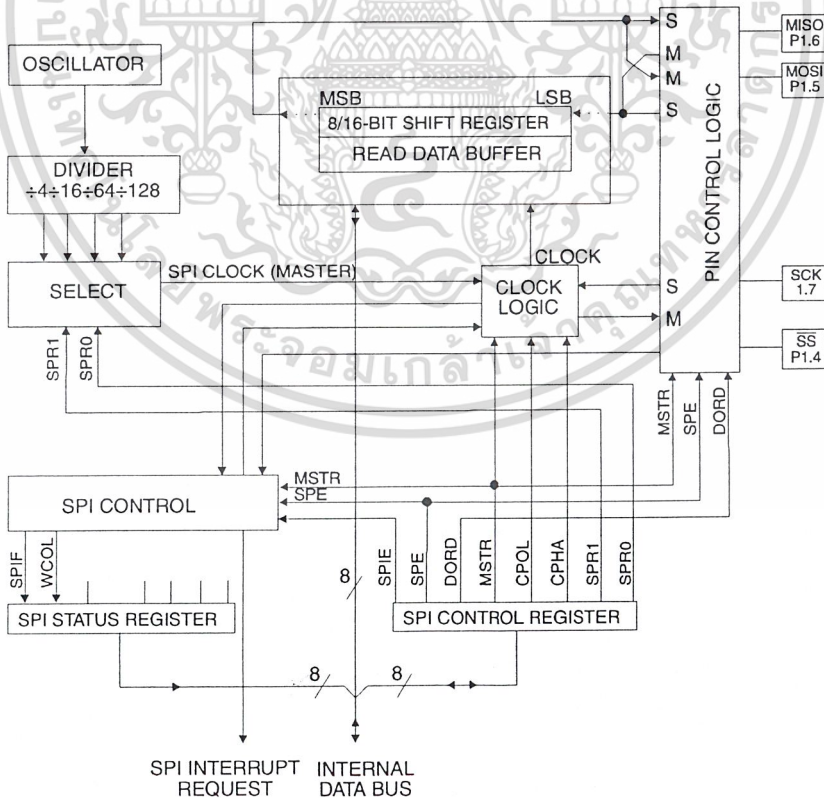
To configure the Timer/Counter 2 as a clock generator, bit C/T2 (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

The clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L), as shown in the following equation.

$$\text{Clock Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]}$$

In the clock-out mode, Timer 2 rollovers will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and clock-out frequencies cannot be determined independently from one another since they both use RCAP2H and RCAP2L.

Figure 6. SPI Block Diagram



UART

The UART in the AT89S8252 operates the same way as the UART in the AT89C51, AT89C52 and AT89C55. For further information, see the October 1995 Microcontroller Data Book, page 2-49, section titled, "Serial Interface."

Serial Peripheral Interface

The serial peripheral interface (SPI) allows high-speed synchronous data transfer between the AT89S8252 and peripheral devices or between several AT89S8252 devices. The AT89S8252 SPI features include the following:

- Full-Duplex, 3-Wire Synchronous Data Transfer
- Master or Slave Operation
- 1.5-MHz Bit Frequency (max.)
- LSB First or MSB First Data Transfer
- Four Programmable Bit Rates
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Wakeup from Idle Mode (Slave Mode Only)

The interconnection between master and slave CPUs with SPI is shown in the following figure. The SCK pin is the clock output in the master mode but is the clock input in the slave mode. Writing to the SPI data register of the master CPU starts the SPI clock generator, and the data written shifts out of the MOSI pin and into the MOSI pin of the slave CPU. After shifting one byte, the SPI clock generator stops, setting the end of transmission flag (SPIF). If both the SPI interrupt enable bit (SPIE) and the serial port interrupt enable bit (ES) are set, an interrupt is requested.

The Slave Select input, $\overline{SS}/P1.4$, is set low to select an individual SPI device as a slave. When $\overline{SS}/P1.4$ is set high, the SPI port is deactivated and the MOSI/P1.5 pin can be used as an input.

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figures 8 and 9.

Figure 7. SPI Master-Slave Interconnection

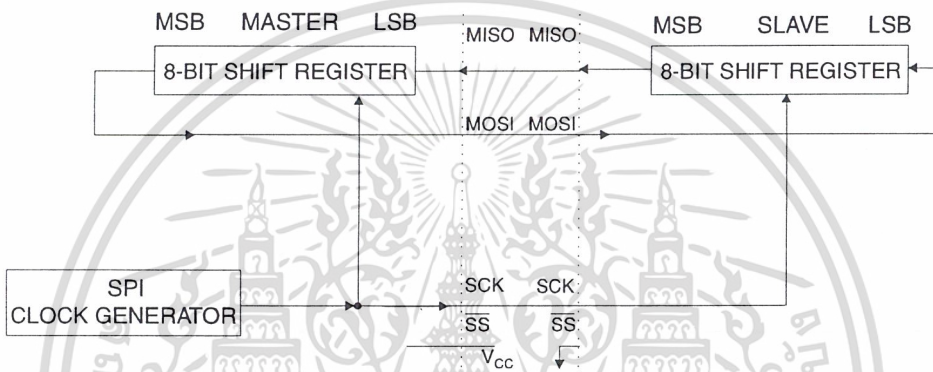
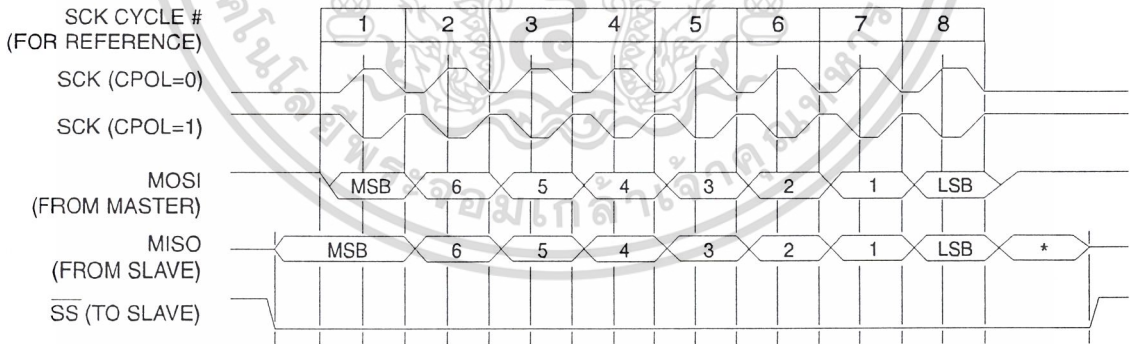


Figure 8. SPI transfer Format with CPHA = 0



*Not defined but normally MSB of character just received





Program Memory Lock Bits

The AT89S8252 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

When lock bit 1 is programmed, the logic level at the \overline{EA} pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random

value and holds that value until reset is activated. The latched value of \overline{EA} must agree with the current logic level at that pin in order for the device to function properly.

Once programmed, the lock bits can only be unprogrammed with the Chip Erase operations in either the parallel or serial modes.

Lock Bit Protection Modes⁽¹⁾⁽²⁾

Program Lock Bits				Protection Type
	LB1	LB2	LB3	
1	U	U	U	No internal memory lock feature.
2	P	U	U	MOV _C instructions executed from external program memory are disabled from fetching code bytes from internal memory. \overline{EA} is sampled and latched on reset and further programming of the Flash memory (parallel or serial mode) is disabled.
3	P	P	U	Same as Mode 2, but parallel or serial verify are also disabled.
4	P	P	P	Same as Mode 3, but external execution is also disabled.

- Notes: 1. U = Unprogrammed
2. P = Programmed

Programming the Flash and EEPROM

Atmel's AT89S8252 Flash Microcontroller offers 8K bytes of in-system reprogrammable Flash Code memory and 2K bytes of EEPROM Data memory.

The AT89S8252 is normally shipped with the on-chip Flash Code and EEPROM Data memory arrays in the erased state (i.e. contents = FFH) and ready to be programmed. This device supports a High-Voltage (12V) Parallel programming mode and a Low-Voltage (5V) Serial programming mode. The serial programming mode provides a convenient way to download the AT89S8252 inside the user's system. The parallel programming mode is compatible with conventional third party Flash or EPROM programmers.

The Code and Data memory arrays are mapped via separate address spaces in the serial programming mode. In the parallel programming mode, the two arrays occupy one contiguous address space: 0000H to 1FFFH for the Code array and 2000H to 27FFH for the Data array.

The Code and Data memory arrays on the AT89S8252 are programmed byte-by-byte in either programming mode. An auto-erase cycle is provided with the self-timed programming operation in the serial programming mode. There is no need to perform the Chip Erase operation to reprogram any memory location in the serial programming mode unless any of the lock bits have been programmed.

In the parallel programming mode, there is no auto-erase cycle. To reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

Parallel Programming Algorithm

To program and verify the AT89S8252 in the parallel programming mode, the following sequence is recommended:

- Power-up sequence:
Apply power between V_{CC} and GND pins.
Set RST pin to "H".
Apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
- Set \overline{PSEN} pin to "L"
ALE pin to "H"
 \overline{EA} pin to "H" and all other pins to "H".
- Apply the appropriate combination of "H" or "L" logic levels to pins P2.6, P2.7, P3.6, P3.7 to select one of the programming operations shown in the Flash Programming Modes table.
- Apply the desired byte address to pins P1.0 to P1.7 and P2.0 to P2.5.
Apply data to pins P0.0 to P0.7 for Write Code operation.
- Raise \overline{EA}/V_{PP} to 12V to enable Flash programming, erase or verification.
- Pulse ALE/ \overline{PROG} once to program a byte in the Code memory array, the Data memory array or the lock bits. The byte-write cycle is self-timed and typically takes 1.5 ms.
- To verify the byte just programmed, bring pin P2.7 to "L" and read the programmed data at pins P0.0 to P0.7.

เอกสารนี้เป็นเอกสารที่... **AT89S8252** ...

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. Repeat steps 3 through 7 changing the address and data for the entire 2K or 8K bytes array or until the end of the object file is reached.
9. Power-off sequence:
 - Set XTAL1 to "L".
 - Set RST and \overline{EA} pins to "L".
 - Turn V_{CC} power off.

In the parallel programming mode, there is no auto-erase cycle and to reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

DATA Polling

The AT89S8252 features \overline{DATA} Polling to indicate the end of a write cycle. During a write cycle in the parallel or serial programming mode, an attempted read of the last byte written will result in the complement of the written datum on P0.7 (parallel mode), and on the MSB of the serial output byte on MISO (serial mode). Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. \overline{DATA} Polling may begin any time after a write cycle has been initiated.

Ready/Busy

The progress of byte programming in the parallel programming mode can also be monitored by the RDY/BSY output signal. Pin P3.4 is pulled Low after ALE goes High during programming to indicate \overline{BUSY} . P3.4 is pulled High again when programming is done to indicate \overline{READY} .

Program Verify

If lock bits LB1 and LB2 have not been programmed, the programmed Code or Data byte can be read back via the address and data lines for verification. The state of the lock bits can also be verified directly in the parallel programming mode. In the serial programming mode, the state of the lock bits can only be verified indirectly by observing that the lock bit features are enabled.

Chip Erase

Both Flash and EEPROM arrays are erased electrically at the same time. In the parallel programming mode, chip erase is initiated by using the proper combination of control signals and by holding ALE/\overline{PROG} low for 10 ms. The Code and Data arrays are written with all "1"s in the Chip Erase operation.

In the serial programming mode, a chip erase operation is initiated by issuing the Chip Erase instruction. In this mode, chip erase is self-timed and takes about 16 ms.

During chip erase, a serial read from any address location will return 00H at the data outputs.

Serial Programming Fuse

A programmable fuse is available to disable Serial Programming if the user needs maximum system security. The Serial Programming Fuse can only be programmed or erased in the Parallel Programming Mode.

The AT89S8252 is shipped with the Serial Programming Mode enabled.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows:

(030H) = 1EH indicates manufactured by Atmel

(031H) = 72H indicates 89S8252

Programming Interface

Every code byte in the Flash and EEPROM arrays can be written, and the entire array can be erased, by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Serial Downloading

Both the Code and Data memory arrays can be programmed using the serial SPI bus while RST is pulled to V_{CC} . The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming Enable instruction needs to be executed first before program/erase operations can be executed.

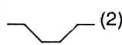



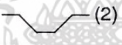
An auto-erase cycle is built into the self-timed programming operation (in the serial mode ONLY) and there is no need to first execute the Chip Erase instruction unless any of the lock bits have been programmed. The Chip Erase operation turns the content of every memory location in both the Code and Data arrays into FFH.

The Code and Data memory arrays have separate address spaces:

0000H to 1FFFFH for Code memory and 000H to 7FFFH for Data memory.

Either an external system clock is supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/40 of the crystal frequency. With a 24 MHz oscillator clock, the maximum SCK frequency is 600 kHz.

Flash and EEPROM Parallel Programming Modes

Mode	RST	PSEN	ALE/PROG	EA/V _{PP}	P2.6	P2.7	P3.6	P3.7	Data I/O P0.7:0	Address P2.5:0 P1.7:0
Serial Prog. Modes	H	h ⁽¹⁾	h ⁽¹⁾	x						
Chip Erase	H	L	 ⁽²⁾	12V	H	L	L	L	X	X
Write (10K bytes) Memory	H	L		12V	L	H	H	H	DIN	ADDR
Read (10K bytes) Memory	H	L	H	12V	L	L	H	H	DOUT	ADDR
Write Lock Bits:	H	L		12V	H	L	H	L	DIN	X
Bit - 1									P0.7 = 0	X
Bit - 2									P0.6 = 0	X
Bit - 3									P0.5 = 0	X
Read Lock Bits:	H	L	H	12V	H	H	L	L	DOUT	X
Bit - 1									@P0.2	X
Bit - 2									@P0.1	X
Bit - 3									@P0.0	X
Read Atmel Code	H	L	H	12V	L	L	L	L	DOUT	30H
Read Device Code	H	L	H	12V	L	L	L	L	DOUT	31H
Serial Prog. Enable	H	L	 ⁽²⁾	12V	L	H	L	H	P0.0 = 0	X
Serial Prog. Disable	H	L	 ⁽²⁾	12V	L	H	L	H	P0.0 = 1	X
Read Serial Prog. Fuse	H	L	H	12V	H	H	L	H	@P0.0	X

- Notes:
1. "h" = weakly pulled "High" internally.
 2. Chip Erase and Serial Programming Fuse require a 10-ms $\overline{\text{PROG}}$ pulse. Chip Erase needs to be performed first before reprogramming any byte with a content other than FFH.
 3. P3.4 is pulled Low during programming to indicate RDY/BSY.
 4. "X" = don't care





Serial Programming Algorithm

To program and verify the AT89S8252 in the serial programming mode, the following sequence is recommended:

- Power-up sequence:
Apply power between V_{CC} and GND pins.
Set RST pin to "H".
If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
- Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 40.
- The Code or Data array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. The selected memory location is first automatically erased before new data is

written. The write cycle is self-timed and typically takes less than 2.5 ms at 5V.

- Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/P1.6.
- At the end of a programming session, RST can be set low to commence normal operation.

Power-off sequence (if needed):

Set XTAL1 to "L" (if a crystal is not used).

Set RST to "L".

Turn V_{CC} power off.

Serial Programming Instruction

The Instruction Set for Serial Programming follows a 3-byte protocol and is shown in the following table:

Instruction Set

Instruction	Input Format			Operation
	Byte 1	Byte 2	Byte 3	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	Enable serial programming interface after RST goes high.
Chip Erase	1010 1100	xxxx x100	xxxx xxxx	Chip erase both 8K & 2K memory arrays.
Read Code Memory	aaaa a001	low addr	xxxx xxxx	Read data from Code memory array at the selected address. The 5 MSBs of the first byte are the high order address bits. The low order address bits are in the second byte. Data are available at pin MISO during the third byte.
Write Code Memory	aaaa a010	low addr	data in	Write data to Code memory location at selected address. The address bits are the 5 MSBs of the first byte together with the second byte.
Read Data Memory	00aa a101	low addr	xxxx xxxx	Read data from Data memory array at selected address. Data are available at pin MISO during the third byte.
Write Data Memory	00aa a110	low addr	data in	Write data to Data memory location at selected address.
Write Lock Bits	1010 1100	$\begin{matrix} \square & \square & \square & \square \\ \square & \square & \square & \square \end{matrix} \times x111$	xxxx xxxx	Write lock bits. Set LB1, LB2 or LB3 = "0" to program lock bits.

- Notes:
- \overline{DATA} polling is used to indicate the end of a write cycle which typically takes less than 2.5 ms at 5V.
 - "aaaaa" = high order address.
 - "x" = don't care.

AT89S8252

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของไมโครอิเล็กทรอนิกส์แห่งหนึ่ง และอยู่ภายใต้เงื่อนไขการใช้งานที่

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 14. Programming the Flash/EEPROM Memory

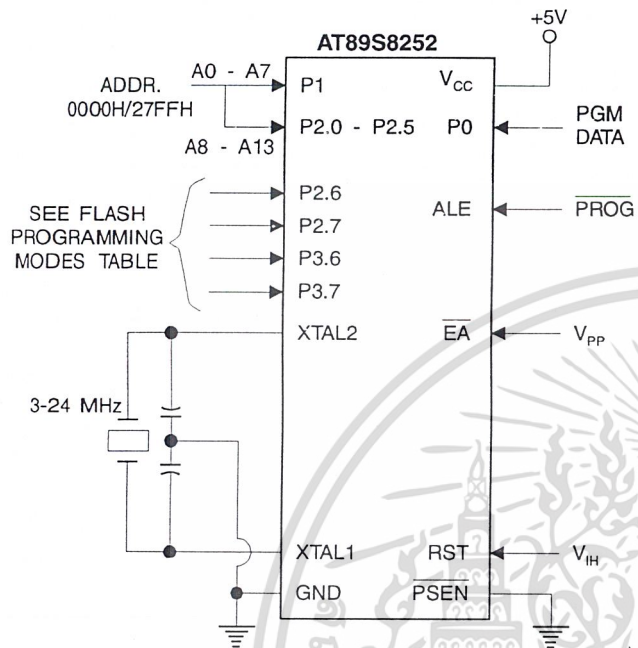


Figure 15. Flash/EEPROM Serial Downloading

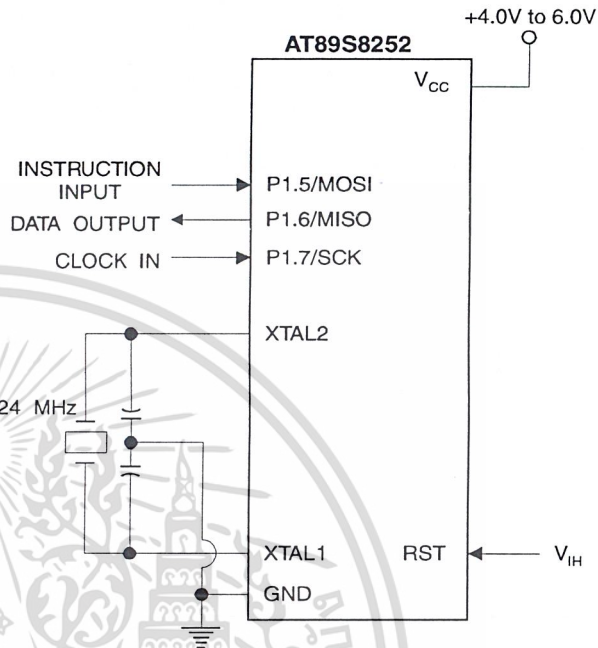
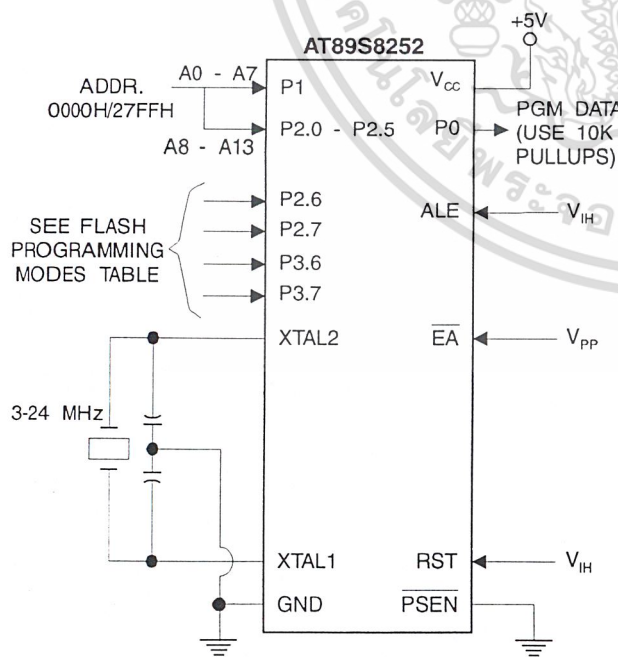


Figure 16. Verifying the Flash/EEPROM Memory



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DALLAS
SEMICONDUCTOR

DS1202, DS1202S Serial Timekeeping Chip

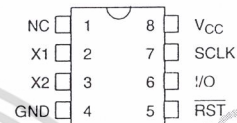
FEATURES

- Real time clock counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap year compensation valid up to 2100
- 24 x 8 RAM for scratchpad data storage
- Serial I/O for minimum pin count
- 2.0–5.5 volt full operation
- Uses less than 300 nA at 2 volts
- Single-byte or multiple-byte (burst mode) data transfer for read or write of clock or RAM data
- 8-pin DIP or optional 16-pin SOIC for surface mount
- Simple 3-wire interface
- TTL-compatible ($V_{CC} = 5V$)
- Optional industrial temperature range $-40^{\circ}C$ to $+85^{\circ}C$ (IND)

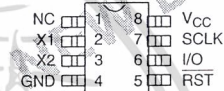
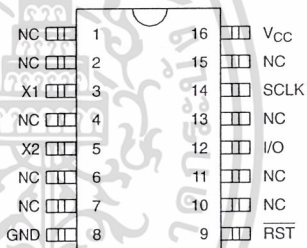
ORDERING INFORMATION

DS1202	8-pin DIP
DS1202S	16-pin SOIC
DS1202S-8	8-pin SOIC
DS1202N	8-pin DIP (IND)
DS1202SN	16-pin SOIC (IND)
DS1202SN-8	8-pin SOIC (IND)

PIN ASSIGNMENT



8-PIN DIP

8-PIN SOIC
(208 mil)

16-PIN SOIC

PIN DESCRIPTION

NC	– No Connection
X1, X2	– 32.768 KHz Crystal Input
GND	– Ground
RST	– Reset
I/O	– Data Input/Output
SCLK	– Serial Clock
V _{CC}	– Power Supply Pin

DESCRIPTION

The DS1202 Serial Timekeeping Chip contains a real time clock/calendar and 24 bytes of static RAM. It communicates with a microprocessor via a simple serial interface. The real time clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with less than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator. Interfacing the

DS1202 with a microprocessor is simplified by using synchronous serial communication. Only three wires are required to communicate with the clock/RAM: (1) RST (Reset), (2) I/O (Data line), and (3) SCLK (Serial clock). Data can be transferred to and from the clock/RAM one byte at a time or in a burst of up to 24 bytes. The DS1202 is designed to operate on very low power and retain data and clock information on less than 1 microwatt.

OPERATION

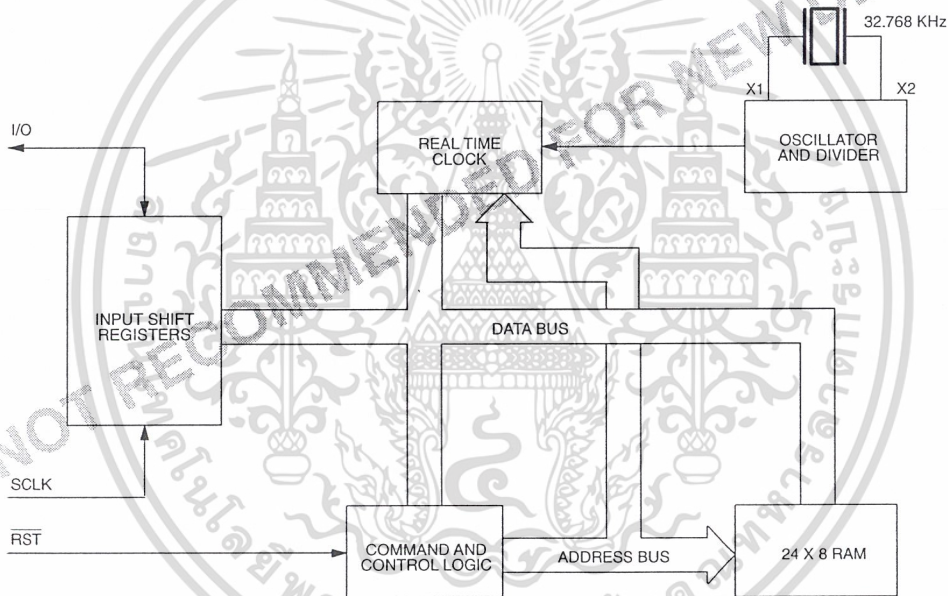
The main elements of the Serial Timekeeper are shown in Figure 1: shift register, control logic, oscillator, real time clock, and RAM. To initiate any transfer of data, \overline{RST} is taken high and eight bits are loaded into the shift register providing both address and command information. Data is serially input on the rising edge of the SCLK. The first eight bits specify which of 32 bytes will be accessed, whether a read or write cycle will take place, and whether a byte or burst mode transfer is to occur. After the first eight clock cycles have occurred which load the command word into the shift register, additional clocks will output data for a read or input data for a write.

The number of clock pulses equals eight plus eight for byte mode or eight plus up to 192 for burst mode.

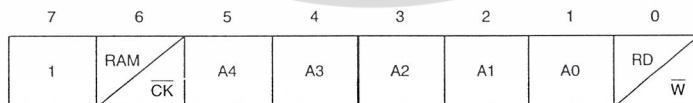
COMMAND BYTE

The command byte is shown in Figure 2. Each data transfer is initiated by a command byte. The MSB (Bit 7) must be a logic 1. If it is zero, further action will be terminated. Bit 6 specifies clock/calendar data if logic 0 or RAM data if logic 1. Bits one through five specify the designated registers to be input or output, and the LSB (Bit 0) specifies a write operation (input) if logic 0 or read operation (output) if logic 1. The command byte is always input starting with the LSB (bit 0).

DS1202 BLOCK DIAGRAM Figure 1



ADDRESS/COMMAND BYTE Figure 2



RESET AND CLOCK CONTROL

All data transfers are initiated by driving the $\overline{\text{RST}}$ input high. The $\overline{\text{RST}}$ input serves two functions. First, $\overline{\text{RST}}$ turns on the control logic which allows access to the shift register for the address/command sequence. Second, the $\overline{\text{RST}}$ signal provides a method of terminating either single byte or multiple byte data transfer. A clock cycle is a sequence of a falling edge followed by a rising edge. For data inputs, data must be valid during the rising edge of the clock and data bits are output on the falling edge of clock. All data transfer terminates if the $\overline{\text{RST}}$ input is low and the I/O pin goes to a high impedance state. Data transfer is illustrated in Figure 3.

DATA INPUT

Following the eight SCLK cycles that input a write command byte, a data byte is input on the rising edge of the next eight SCLK cycles. Additional SCLK cycles are ignored should they inadvertently occur. Data is input starting with bit 0. Due to the inherent nature of the logic state machine, writing times containing an absolute value of "59" seconds should be avoided.

DATA OUTPUT

Following the eight SCLK cycles that input a read command byte, a data byte is output on the falling edge of the next eight SCLK cycles. Note that the first data bit to be transmitted occurs on the first falling edge after the last bit of the command byte is written. Additional SCLK cycles retransmit the data bytes should they inadvertently occur so long as $\overline{\text{RST}}$ remains high. This operation permits continuous burst mode read capability. Data is output starting with bit 0.

BURST MODE

Burst mode may be specified for either the clock/calendar or the RAM registers by addressing location 31 decimal (address/command bits one through five = logical one). As before, bit six specified clock or RAM and bit 0 specifies read or write. There is no data storage capacity at locations 8 through 31 in the Clock/Calendar Registers or locations 24 through 31 in the RAM registers. When writing to the clock registers in the burst mode, the first eight registers must be written in order for the data to be transferred.

However, when writing to RAM in burst mode it is not necessary to write all 24 bytes for the data to transfer.

Each byte that is written to will be transferred to RAM regardless of whether all 24 bytes are written or not.

CLOCK/CALENDAR

The clock/calendar is contained in eight write/read registers as shown in Figure 4. Data contained in the clock/calendar registers is in binary coded decimal format (BCD).

CLOCK HALT FLAG

Bit 7 of the seconds register is defined as the clock halt flag. When this bit is set to logic 1, the clock oscillator is stopped and the DS1202 is placed into a low-power standby mode with a current drain of not more than 100 nanoamps. When this bit is written to logic 0, the clock will start.

AM-PM/12-24 MODE

Bit 7 of the hours register is defined as the 12- or 24-hour mode select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24-hour mode, bit 5 is the second 10 hour bit (20–23 hours).

WRITE PROTECT BIT

Bit 7 of the control register is the write protect bit. The first seven bits (bits 0–6) are forced to zero and will always read a zero when read. Before any write operation to the clock or RAM, bit 7 must be zero. When high, the write protect bit prevents a write operation to any other register.

CLOCK/CALENDAR BURST MODE

The clock/calendar command byte specifies burst mode operation. In this mode the eight clock/calendar registers can be consecutively read or written (see Figure 4) starting with bit 0 of address 0.

RAM

The static RAM is 24 x 8 bytes addressed consecutively in the RAM address space.

RAM BURST MODE

The RAM command byte specifies burst mode operation. In this mode, the 24 RAM registers can be consecutively read or written (see Figure 4) starting with bit 0 of address 0.

REGISTER SUMMARY

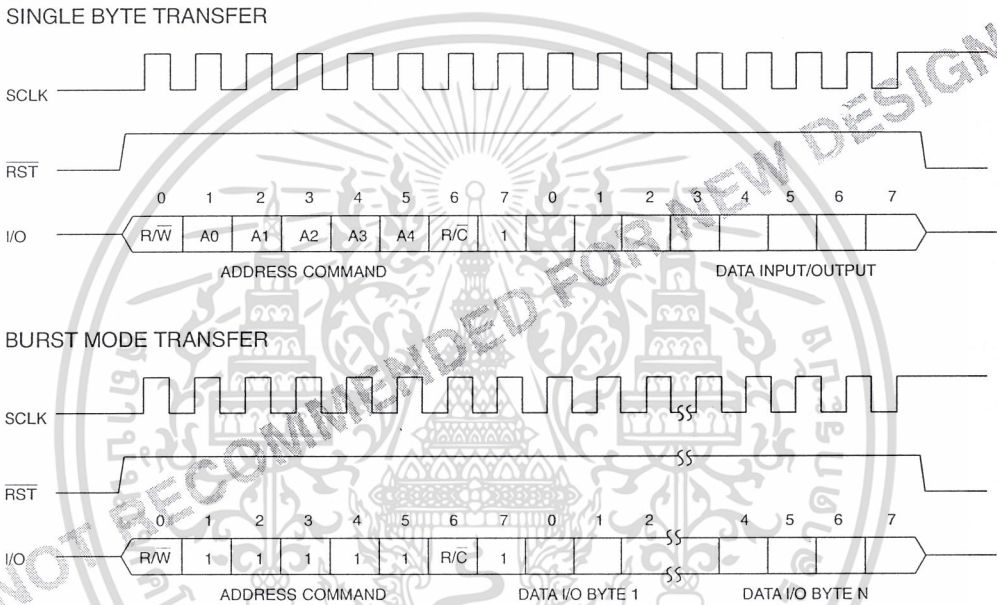
A register data format summary is shown in Figure 4.

CRYSTAL SELECTION

A 32.768 KHz crystal, can be directly connected to the DS1202 via pins 2 and 3 (X1, X2). The crystal selected for use should have a specified load capacitance (CL) of 6 pF. The crystal is connected directly to the X1 and X2

pins. There is no need for external capacitors or resistors. Note: X1 and X2 are very high impedance nodes. It is recommended that they and the crystal be guard-ringed with ground and that high frequency signals be kept away from the crystal area. For more information on crystal selection and crystal layout considerations, please consult Application Note 58, "Crystal Considerations with Dallas Real Time Clocks".

DATA TRANSFER SUMMARY Figure 3



FUNCTION	BYTE N	SCLK n
CLOCK	8	72
RAM	24	200

REGISTER ADDRESS/DEFINITION Figure 4

REGISTER ADDRESS
A. CLOCK

REGISTER DEFINITION

7	6	5	4	3	2	1	0		00-59	CH	10 SEC	SEC		
SEC	1	0	0	0	0	0	0	RD/W						
MIN	1	0	0	0	0	0	1	RD/W	00-59	0	10 MIN	MIN		
HR	1	0	0	0	0	1	0	RD/W	01-12	12/24	0	10 A/P	HR	HR
DATE	1	0	0	0	0	1	1	RD/W	01-28/29	0	0	10 DATE	DATE	
MONTH	1	0	0	0	1	0	0	RD/W	01-12	0	0	0	10 M	MONTH
DAY	1	0	0	0	1	0	1	RD/W	01-07	0	0	0	0	DAY
YEAR	1	0	0	0	1	1	0	RD/W	0-99	10 YEAR	YEAR			
CONTROL	1	0	0	0	1	1	1	RD/W	WP	FORCED TO ZERO				
CLOCK BURST	1	0	1	1	1	1	1	RD/W						

B. RAM

RAM 0	1	1	0	0	0	0	0	RD/W	RAM DATA 0							
RAM 23	1	1	1	0	1	1	1	RD/W	RAM DATA 23							
RAM BURST	1	1	1	1	1	1	1	RD/W								

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS*

Voltage on Any Pin Relative to Ground	-0.3V to +7.0V
Operating Temperature	0°C to 70°C
Storage Temperature	-55°C to +125°C
Soldering Temperature	260°C for 10 seconds

* This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

RECOMMENDED DC OPERATING CONDITIONS

(0°C to 70°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V_{CC}	2.0		5.5	V	1
Logic 1 Input	V_{IH}	2.0		$V_{CC}+0.3$	V	1
Logic 0 Input	V_{IL}	$V_{CC}=2.0V$	-0.3	+0.3	V	1
		$V_{CC}=5V$	-0.3	+0.8		

DC ELECTRICAL CHARACTERISTICS(0°C to 70°C; $V_{CC} = 2.0$ to 5.5V*)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Input Leakage	I_{LI}			+500	μA	6
I/O Leakage	I_{LO}			+500	μA	6
Logic 1 Output	V_{OH}	$V_{CC}=2V$	1.6		V	2
		$V_{CC}=5V$	2.4			
Logic 0 Output	V_{OL}	$V_{CC}=2V$		0.4	V	3
		$V_{CC}=5V$		0.4		
Active Supply Current	I_{CC}	$V_{CC}=2V$		0.4	mA	5
		$V_{CC}=5V$		1.2		
Timekeeping Current	I_{CC1}	$V_{CC}=2V$		0.3	μA	4
		$V_{CC}=5V$		1		
Leakage Current	I_{CC2}	$V_{CC}=2V$		100	nA	10
		$V_{CC}=5V$		100		

*Unless otherwise noted.

CAPACITANCE(t_A = 25°C)

PARAMETER	SYMBOL	CONDITION	TYP	MAX	UNITS	NOTES
Input Capacitance	C_I		5		pF	
I/O Capacitance	$C_{I/O}$		10		pF	
Crystal Capacitance	C_X		6		pF	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC ELECTRICAL CHARACTERISTICS

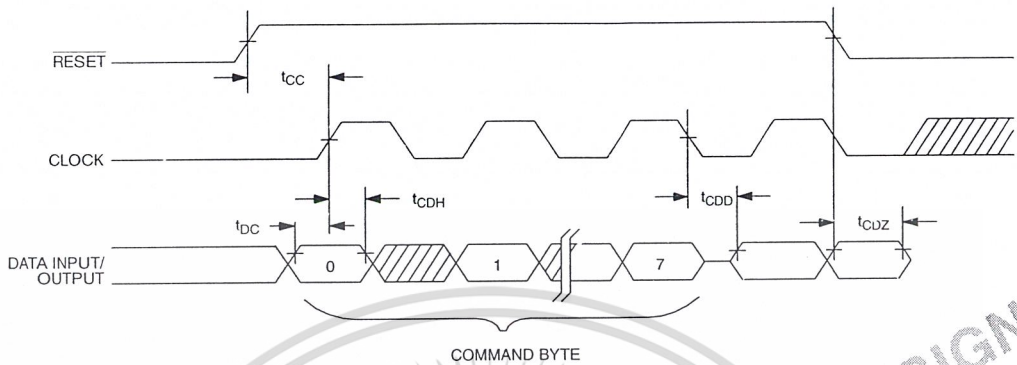
(0°C to 70°C; $V_{CC} = 2.0$ to $5.5V^*$)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Data to CLK Setup	t_{DC}	$V_{CC}=2V$	200			ns 7
		$V_{CC}=5V$	50			
CLK to Data Hold	t_{CDH}	$V_{CC}=2V$	280			ns 7
		$V_{CC}=5V$	70			
CLK to Data Delay	t_{CDD}	$V_{CC}=2V$			800	ns 7, 8, 9
		$V_{CC}=5V$			200	
CLK Low Time	t_{CL}	$V_{CC}=2V$	1000			ns 7
		$V_{CC}=5V$	250			
CLK High Time	t_{CH}	$V_{CC}=2V$	1000			ns 7, 12
		$V_{CC}=5V$	250			
CLK Frequency	f_{CLK}	$V_{CC}=2V$			0.5	MHz 7, 12
		$V_{CC}=5V$			DC 2.0	
CLK Rise and Fall	t_R, t_F	$V_{CC}=2V$			2000	ns
		$V_{CC}=5V$			500	
\overline{RST} to CLK Setup	t_{CC}	$V_{CC}=2V$	4			μs 7
		$V_{CC}=5V$	1			
CLK to \overline{RST} Hold	t_{CCH}	$V_{CC}=2V$	1000			ns 7
		$V_{CC}=5V$	250			
\overline{RST} Inactive Time	t_{CWH}	$V_{CC}=2V$	4			μs 7
		$V_{CC}=5V$	1			
\overline{RST} to I/O High Z	t_{CDZ}	$V_{CC}=2V$			280	ns 7
		$V_{CC}=5V$			70	

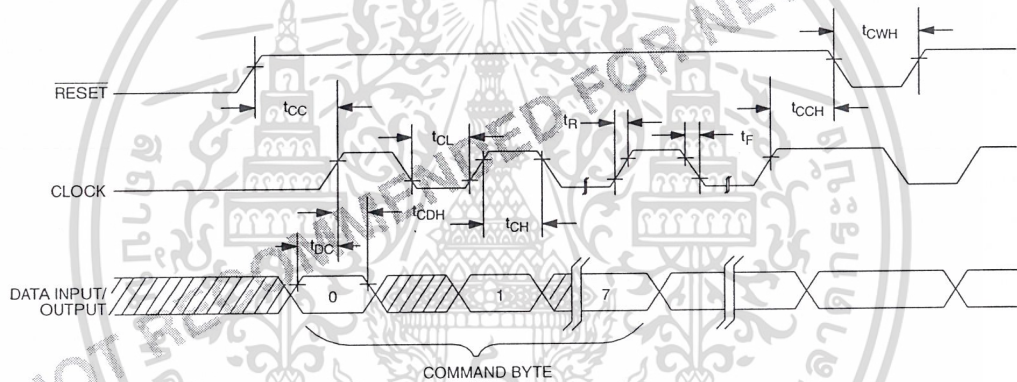
*Unless otherwise noted.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TIMING DIAGRAM: READ DATA TRANSFER Figure 5



TIMING DIAGRAM: WRITE DATA TRANSFER Figure 6

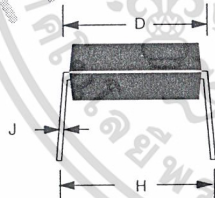
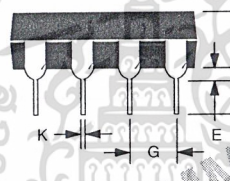
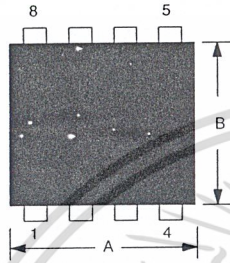


NOTES:

1. All voltages are referenced to ground.
2. Logic one voltages are specified at a source current of 1 mA at $V_{CC}=5V$ and 0.4 mA at $V_{CC}=2V$, $V_{OH}=V_{CC}$ for capacitive loads.
3. Logic zero voltages are specified at a sink current of 4 mA at $V_{CC}=5V$ and 1.5 mA at $V_{CC}=2V$.
4. I_{CC1} is specified with I/O open, \overline{RST} set to a logic 0, and clock halt flag=0 (oscillator enabled).
5. I_{CC} is specified with the I/O pin open, \overline{RST} high, $SCLK=2\text{ MHz}$ at $V_{CC}=5V$; $SCLK=500\text{ KHz}$, $V_{CC}=2V$ and clock halt flag=0 (oscillator enabled).
6. \overline{RST} , $SCLK$, and I/O all have 40K Ω pull-down resistors to ground.
7. Measured at $V_{IH}=2.0V$ or $V_{IL}=0.8V$ and 10 ms maximum rise and fall time.
8. Measured at $V_{OH}=2.4V$ or $V_{OL}=0.4V$.
9. Load capacitance = 50 pF.

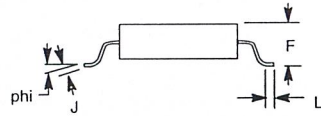
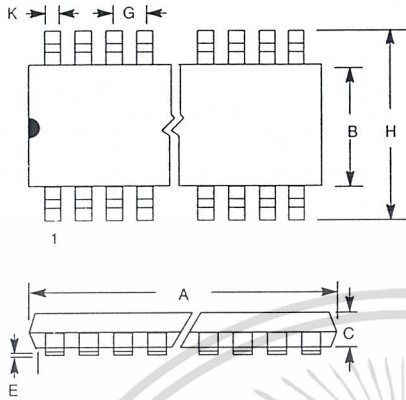
10. I_{CC2} is specified with \overline{RST} , I/O, and SCLK open. The clock halt flag must be set to logic one (oscillator disabled).
11. At power-up, \overline{RST} must be at a logic 0 until $V_{CC} \geq 2$ volts. Also, SCLK must be at a logic 0 when \overline{RST} is driven to a logic one state.
12. If t_{CH} exceeds 100 ms with \overline{RST} in a logic one state, then I_{CC} may briefly exceed I_{CC} specification.

DS1202 SERIAL TIMEKEEPER 8-PIN DIP



PKG	8-PIN	
	MIN	MAX
A IN. MM	0.360	0.400
B IN. MM	0.240	0.260
C IN. MM	0.120	0.140
D IN. MM	0.300	0.325
E IN. MM	0.015	0.040
F IN. MM	0.110	0.140
G IN. MM	0.090	0.110
H IN. MM	0.320	0.370
J IN. MM	0.008	0.012
K IN. MM	0.015	0.021

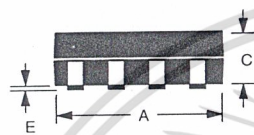
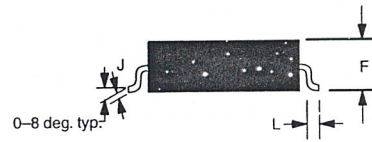
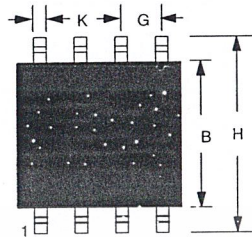
DS1202S SERIAL TIMEKEEPER 16-PIN SOIC



PKG	16-PIN	
DIM	MIN	MAX
A IN.	0.500	0.511
MM	12.70	12.99
B IN.	0.290	0.300
MM	7.37	7.65
C IN.	0.089	0.095
MM	2.26	2.41
E IN.	0.004	0.012
MM	0.102	0.30
F IN.	0.094	0.105
MM	2.38	2.68
G IN.	0.050 BSC	1.27 BSC
MM		
H IN.	0.398	0.416
MM	10.11	10.57
J IN.	0.009	0.013
MM	0.229	0.33
K IN.	0.013	0.019
MM	0.33	0.48
L IN.	0.016	0.040
MM	0.406	1.20
phi	0°	8°

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DS1202S8 8-PIN SOIC 200 MIL



PKG	8-PIN	
	DIM	MIN
A IN.	0.203	0.213
MM	5.16	5.41
B IN.	0.203	0.213
MM	5.16	5.41
C IN.	0.070	0.074
MM	1.78	1.88
E IN.	0.004	0.010
MM	0.102	0.390
F IN.	0.074	0.84
MM	1.88	2.13
G IN.	0.050 BSC	
MM	1.27 BSC	
H IN.	0.302	0.318
MM	7.67	8.07
J IN.	0.006	0.010
MM	0.152	0.254
K IN.	0.013	0.020
MM	0.33	0.508
L IN.	0.19	0.030
MM	4.83	0.762

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Microprocessor Supervisory Circuits

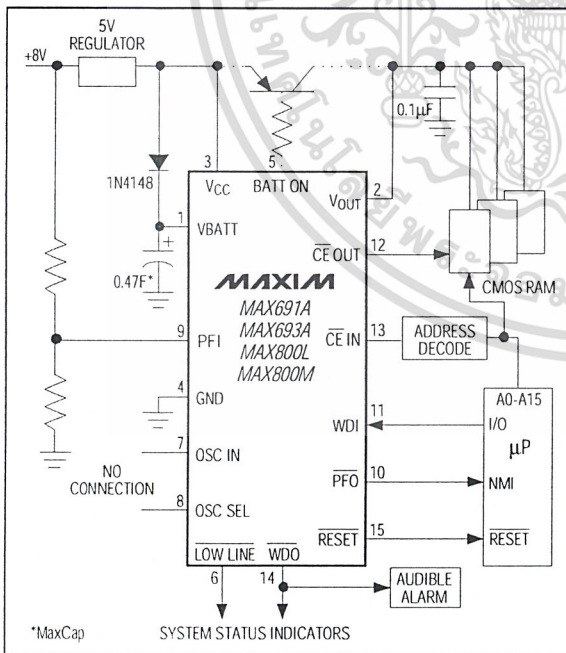
General Description

The MAX691A/MAX693A/MAX800L/MAX800M microprocessor (μ P) supervisory circuits are pin-compatible upgrades to the MAX691, MAX693, and MAX695. They improve performance with 30 μ A supply current, 200ms typ reset active delay on power-up, and 6ns chip-enable propagation delay. Features include write protection of CMOS RAM or EEPROM, separate watchdog outputs, backup-battery switchover, and a **RESET** output that is valid with V_{CC} down to 1V. The MAX691A/MAX800L have a 4.65V typical reset-threshold voltage, and the MAX693A/MAX800M's reset threshold is 4.4V typical. The MAX800L/MAX800M guarantee power-fail accuracies to $\pm 2\%$.

Applications

- Computers
- Controllers
- Intelligent Instruments
- Automotive Systems
- Critical μ P Power Monitoring

Typical Operating Circuit



Features

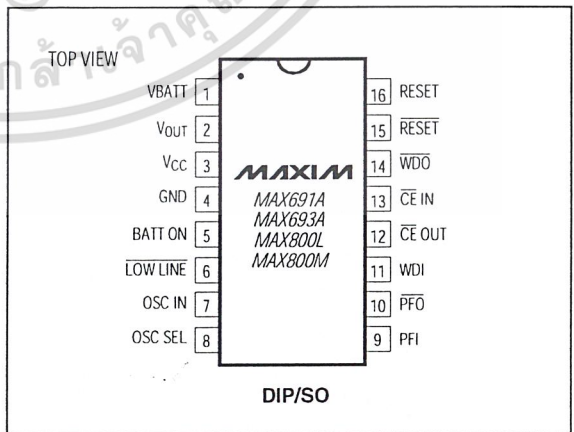
- ◆ 200ms Power-OK/Reset Timeout Period
- ◆ 1 μ A Standby Current, 30 μ A Operating Current
- ◆ On-Board Gating of Chip-Enable Signals, 10ns Max Delay
- ◆ MaxCap™ or SuperCap™ Compatible
- ◆ Guaranteed RESET Assertion to $V_{CC} = +1V$
- ◆ Voltage Monitor for Power-Fail or Low-Battery Warning
- ◆ Power-Fail Accuracy Guaranteed to $\pm 2\%$ (MAX800L/M)
- ◆ Available in 16-Pin Narrow SO and Plastic DIP Packages

Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX691ACPE	0°C to +70°C	16 Plastic DIP
MAX691ACSE	0°C to +70°C	16 Narrow SO
MAX691ACWE	0°C to +70°C	16 Wide SO
MAX691AC/D	0°C to +70°C	Dice*
MAX691AEPE	-40°C to +85°C	16 Plastic DIP
MAX691AESE	-40°C to +85°C	16 Narrow SO
MAX691AEWE	-40°C to +85°C	16 Wide SO
MAX691AEJE	-40°C to +85°C	16 CERDIP
MAX691AMJE	-55°C to +125°C	16 CERDIP

Ordering Information continued on last page.
* Dice are specified at $T_A = +25^\circ C$, DC parameters only.

Pin Configuration



SuperCap is a registered trademark of Baknor Industries. MaxCap is a registered trademark of The Carborundum Corp.



Maxim Integrated Products

For free samples & the latest literature: <http://www.maxim-ic.com>, or phone 1-800-998-8800.
For small orders, phone 408-737-7600 ext. 3468.

MAX691A/MAX693A/MAX800L/MAX800M

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Microprocessor Supervisory Circuits

Pin Description

PIN	NAME	FUNCTION
1	VBATT	Battery-Backup Input. Connect to external battery or capacitor and charging circuit. If backup battery is not used, connect to GND.
2	V _{OUT}	Output Supply Voltage. When V _{CC} is greater than VBATT and above the reset threshold, V _{OUT} connects to V _{CC} . When V _{CC} falls below VBATT and is below the reset threshold, V _{OUT} connects to VBATT. Connect a 0.1μF capacitor from V _{OUT} to GND. Connect V _{OUT} to V _{CC} if no backup battery is used.
3	V _{CC}	Input Supply Voltage, 5V input.
4	GND	Ground, 0V reference for all signals.
5	BATT ON	Battery On Output. When V _{OUT} switches to VBATT, BATT ON goes high. When V _{OUT} switches to V _{CC} , BATT ON goes low. Connect the base of a PNP through a current-limiting resistor to BATT ON for V _{OUT} current requirements greater than 250mA.
6	LOW LINE	LOW LINE output goes low when V _{CC} falls below the reset threshold. It returns high as soon as V _{CC} rises above the reset threshold.
7	OSC IN	External Oscillator Input. When OSC SEL is unconnected or driven high, a 10μA pull-up connects from V _{OUT} to OSC IN, the internal oscillator sets the reset and watchdog timeout periods, and OSC IN selects between fast and slow watchdog timeout periods. When OSC SEL is driven low, the reset and watchdog timeout periods may be set either by a capacitor from OSC IN to ground or by an external clock at OSC IN (Figure 3).
8	OSC SEL	Oscillator Select. When OSC SEL is unconnected or driven high, the internal oscillator sets the reset delay and watchdog timeout period. When OSC SEL is low, the external oscillator input (OSC IN) is enabled (Table 1). OSC SEL has a 10μA internal pull-up.
9	PFI	Power-Fail Input. This is the noninverting input to the power-fail comparator. When PFI is less than 1.25V, PFO goes low. When PFI is not used, connect PFI to GND or V _{OUT} .
10	PFO	Power-Fail Output. This is the output of the power-fail comparator. PFO goes low when PFI is less than 1.25V. This is an uncommitted comparator, and has no effect on any other internal circuitry.
11	WDI	Watchdog Input. WDI is a three-level input. If WDI remains either high or low for longer than the watchdog timeout period, WDO goes low and reset is asserted for the reset timeout period. WDO remains low until the next transition at WDI. Leaving WDI unconnected disables the watchdog function. WDI connects to an internal voltage divider between V _{OUT} and GND, which sets it to mid-supply when left unconnected.
12	CE OUT	Chip-Enable Output. CE OUT goes low only when CE IN is low and V _{CC} is above the reset threshold. If CE IN is low when reset is asserted, CE OUT will stay low for 15μs or until CE IN goes high, whichever occurs first.
13	CE IN	Chip-Enable Input. The input to chip-enable gating circuit. If CE IN is not used, connect CE IN to GND or V _{OUT} .
14	WDO	Watchdog Output. If WDI remains high or low longer than the watchdog timeout period, WDO goes low and reset is asserted for the reset timeout period. WDO returns high on the next transition at WDI. WDO remains high if WDI is unconnected.
15	RESET	RESET Output goes low whenever V _{CC} falls below the reset threshold. RESET will remain low typically for 200ms after V _{CC} crosses the reset threshold on power-up.
16	RESET	RESET is an active-high output. It is open drain, and the inverse of RESET.

Detailed Description

RESET and RESET Outputs

The MAX691A/MAX693A/MAX800L/MAX800M's RESET and RESET outputs ensure that the μP (with reset inputs asserted either high or low) powers up in a known state, and prevents code-execution errors during power-down or brownout conditions.

The RESET output is active low, and typically sinks 3.2mA at 0.1V saturation voltage in its active state. When deasserted, RESET sources 1.6mA at typically V_{OUT} - 0.5V. RESET output is open drain, active high, and typically sinks 3.2mA with a saturation voltage of 0.1V. When no backup battery is used, RESET output is

guaranteed to be valid down to V_{CC} = 1V, and an external 10kΩ pull-down resistor on RESET insures that it will be valid with V_{CC} down to GND (Figure 1). As V_{CC} goes below 1V, the gate drive to the RESET output switch reduces accordingly, increasing the R_{DS(ON)} and the saturation voltage. The 10kΩ pull-down resistor insures the parallel combination of switch plus resistor is around 10kΩ and the output saturation voltage is below 0.4V while sinking 40μA. When using a 10kΩ external pull-down resistor, the high state for RESET output with V_{CC} = 4.75V will be 4.5V typical. For battery voltages ≥ 2V connected to VBATT, RESET and RESET remain valid for V_{CC} from 0V to 5.5V.

MAXIM

MAX691A/MAX693A/MAX800L/MAX800M

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Microprocessor Supervisory Circuits

MAX691A/MAX693A/MAX800L/MAX800M

Table 1. Reset Pulse Width and Watchdog Timeout Selections

OSC SEL	OSC IN	Watchdog Timeout Period		Reset Timeout Period
		Normal	Immediately After Reset	
Low	External Clock Input	1024 clks	4096 clks	2048 clks
Low	External Capacitor	(600/47pF x C)ms	(2.4/47pF x C)sec	(1200/47pF x C)ms
Floating	Low	100ms	1.6sec	200ms
Floating	Floating	1.6sec	1.6sec	200ms

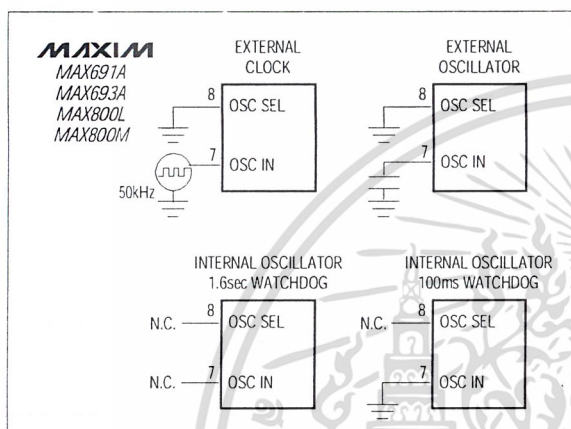


Figure 3. Oscillator Circuits

Chip-Enable Signal Gating

The MAX691A/MAX693A/MAX800L/MAX800M provide internal gating of chip-enable (CE) signals to prevent erroneous data from being written to CMOS RAM in the event of a power failure. During normal operation, the CE gate is enabled and passes all CE transitions. When reset is asserted, this path becomes disabled, preventing erroneous data from corrupting the CMOS RAM. All these parts use a series transmission gate from $\overline{\text{CE}} \text{ IN}$ to $\overline{\text{CE}} \text{ OUT}$ (Figure 4).

The 10ns max CE propagation delay from $\overline{\text{CE}} \text{ IN}$ to $\overline{\text{CE}} \text{ OUT}$ enables the parts to be used with most μPs .

Chip-Enable Input

The Chip-Enable Input ($\overline{\text{CE}} \text{ IN}$) is high impedance (disabled mode) while RESET and $\overline{\text{RESET}}$ are asserted.

During a power-down sequence where V_{CC} falls below the reset threshold or a watchdog fault, $\overline{\text{CE}} \text{ IN}$ assumes a high-impedance state when the voltage at $\overline{\text{CE}} \text{ IN}$ goes high or 15 μs after reset is asserted, whichever occurs first (Figure 5).

During a power-up sequence, $\overline{\text{CE}} \text{ IN}$ remains high impedance, regardless of $\overline{\text{CE}} \text{ IN}$ activity, until reset is deasserted following the reset timeout period.

In the high-impedance mode, the leakage currents into this terminal are $\pm 1\mu\text{A}$ max over temperature. In the low-impedance mode, the impedance of $\overline{\text{CE}} \text{ IN}$ appears as a 75 Ω resistor in series with the load at $\overline{\text{CE}} \text{ OUT}$.

The propagation delay through the CE transmission gate depends on both the source impedance of the drive to $\overline{\text{CE}} \text{ IN}$ and the capacitive loading on the Chip-Enable Output ($\overline{\text{CE}} \text{ OUT}$) (see Chip-Enable Propagation Delay vs. $\overline{\text{CE}} \text{ OUT}$ Load Capacitance in the *Typical Operating Characteristics*). The CE propagation delay is production tested from the 50% point of $\overline{\text{CE}} \text{ IN}$ to the 50% point of $\overline{\text{CE}} \text{ OUT}$ using a 50 Ω driver and 50pF of load capacitance (Figure 6). For minimum propagation delay, minimize the capacitive load at $\overline{\text{CE}} \text{ OUT}$, and use a low output-impedance driver.

Chip-Enable Output

In the enabled mode, the impedance of $\overline{\text{CE}} \text{ OUT}$ is equivalent to 75 Ω in series with the source driving $\overline{\text{CE}} \text{ IN}$. In the disabled mode, the 75 Ω transmission gate is off and $\overline{\text{CE}} \text{ OUT}$ is actively pulled to V_{OUT} . This source turns off when the transmission gate is enabled.

LOW LINE Output

LOW LINE is the buffered output of the reset threshold comparator. LOW LINE typically sinks 3.2mA at 0.1V. For normal operation (V_{CC} above the LOW LINE threshold), LOW LINE is pulled to V_{OUT} .

Power-Fail Comparator

The power-fail comparator is an uncommitted comparator that has no effect on the other functions of the IC. Common uses include low-battery indication (Figure 7), and early power-fail warning (see *Typical Operating Circuit*).

Power-Fail Input

Power Fail Input (PFI) is the input to the power-fail comparator. It has a guaranteed input leakage of $\pm 25\text{nA}$ max over temperature. The typical comparator delay is 25 μs from V_{IL} to V_{OL} (power failing), and 60 μs from V_{IH} to V_{OH} (power being restored). If PFI is not used, connect it to ground.

MAXIM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Microprocessor Supervisory Circuits

Battery-Backup Input

The Battery-Backup Input (VBATT) is similar to the V_{CC} input except the PMOS switch and parallel diode are much smaller. Accordingly, the on-resistances of the diode and the switch are each approximately 10Ω . Continuous current should be limited to 25mA and peak currents (only during power-up) limited to 250mA. The reverse leakage of this input is less than $1\mu A$ over temperature and supply voltage (Figure 8).

Output Supply Voltage

The Output Supply Voltage (V_{OUT}) pin is internally connected to the substrate of the IC and supplies current to the external system and internal circuitry. All open-circuit outputs will, for example, assume the V_{OUT} voltage in their high states rather than the V_{CC} voltage. At the maximum source current of 250mA, V_{OUT} will typically be 200mV below V_{CC} . Decouple this terminal with a $0.1\mu F$ capacitor.

Applications Information

The MAX691A/MAX693A/MAX800L/MAX800M are not short-circuit protected. Shorting V_{OUT} to ground, other than power-up transients such as charging a decoupling capacitor, destroys the device.

All open-circuit outputs swing between V_{OUT} and GND rather than V_{CC} and GND.

If long leads connect to the chip inputs, insure that these leads are free from ringing and other conditions that would forward bias the chip's protection diodes.

There are three distinct modes of operation:

- 1) Normal operating mode with all circuitry powered up. Typical supply current from V_{CC} is $35\mu A$ while only leakage currents flow from the battery.
- 2) Battery-backup mode where V_{CC} is typically within 0.7V below VBATT. All circuitry is powered up and the supply current from the battery is typically less than $60\mu A$.
- 3) Battery-backup mode where V_{CC} is less than VBATT by at least 0.7V. VBATT supply current is $1\mu A$ max.

Using SuperCap™ or MaxCap™ with the MAX691A/MAX693A/MAX800L/MAX800M

VBATT has the same operating voltage range as V_{CC} , and the battery switchover threshold voltages are typically $\pm 30mV$ centered at VBATT, allowing use of a SuperCap and a simple charging circuit as a backup source (Figure 9).

If V_{CC} is above the reset threshold and VBATT is 0.5V above V_{CC} , current flows to V_{OUT} and V_{CC} from VBATT until the voltage at VBATT is less than 0.5V above V_{CC} . For example, with a SuperCap connected to VBATT and through a diode to V_{CC} , if V_{CC} quickly changes from 5.4V to 4.9V, the capacitor discharges through V_{OUT} and V_{CC} until VBATT reaches 5.1V typ. Leakage current through the SuperCap charging diode and the internal power diode eventually discharges the SuperCap to V_{CC} . Also, if V_{CC} and VBATT start from 0.1V above the reset threshold and power is lost at V_{CC} , the SuperCap on VBATT discharges through V_{CC} until VBATT reaches the reset threshold; then the battery-backup mode is initiated and the current through V_{CC} goes to zero.

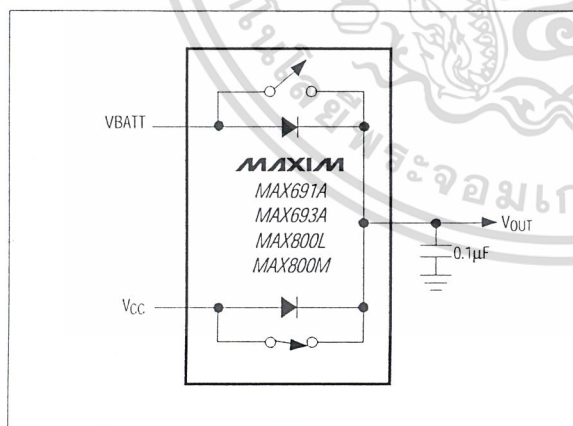


Figure 8. V_{CC} and VBATT to V_{OUT} Switch

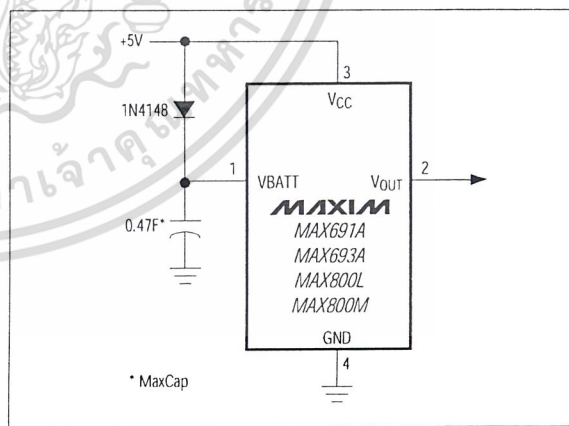


Figure 9. SuperCap or MaxCap on VBATT

MAXIM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Microprocessor Supervisory Circuits

MAX691A/MAX693A/MAX800L/MAX800M

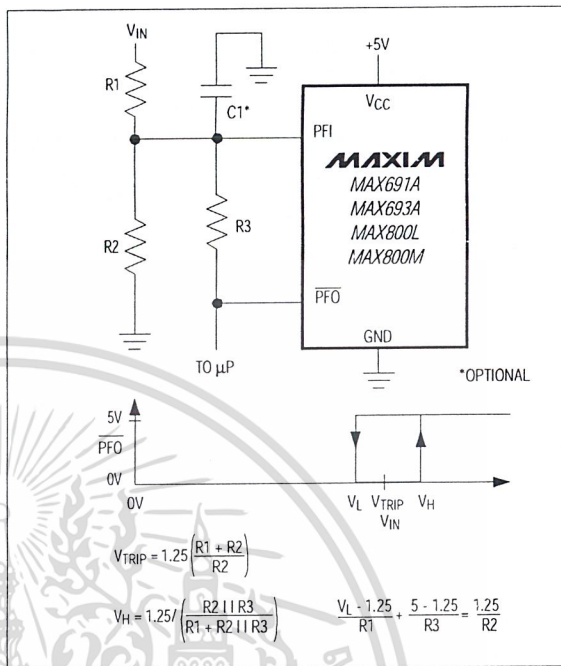
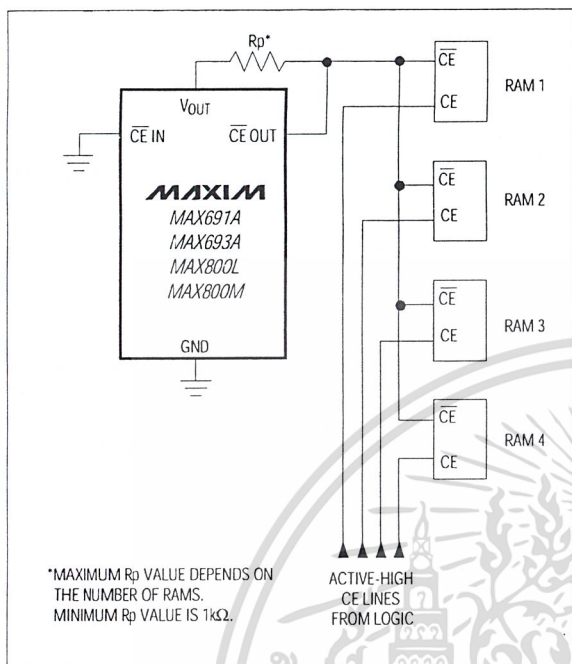


Figure 10. Alternate CE Gating

Figure 11. Adding Hysteresis to the Power-Fail Comparator

Using Separate Power Supplies for VBATT and VCC

If using separate power supplies for V_{CC} and V_{BATT} , V_{BATT} must be less than 0.3V above V_{CC} when V_{CC} is above the reset threshold. As described in the previous section, if V_{BATT} exceeds this limit and power is lost at V_{CC} , current flows continuously from V_{BATT} to V_{CC} via the V_{BATT} -to- V_{OUT} diode and the V_{OUT} -to- V_{CC} switch until the circuit is broken (Figure 8).

Alternate Chip-Enable Gating

Using memory devices with both CE and \overline{CE} inputs allows the CE loop to be bypassed. To do this, connect \overline{CE} IN to ground, pull up \overline{CE} OUT to V_{OUT} , and connect \overline{CE} OUT to the \overline{CE} input of each memory device (Figure 10). The CE input of each part then connects directly to the chip-select logic, which does not have to be gated.

Adding Hysteresis to the Power-Fail Comparator

Hysteresis adds a noise margin to the power-fail comparator and prevents repeated triggering of \overline{PFO} when V_{IN} is near the power-fail comparator trip point. Figure 11 shows how to add hysteresis to the power-fail com-

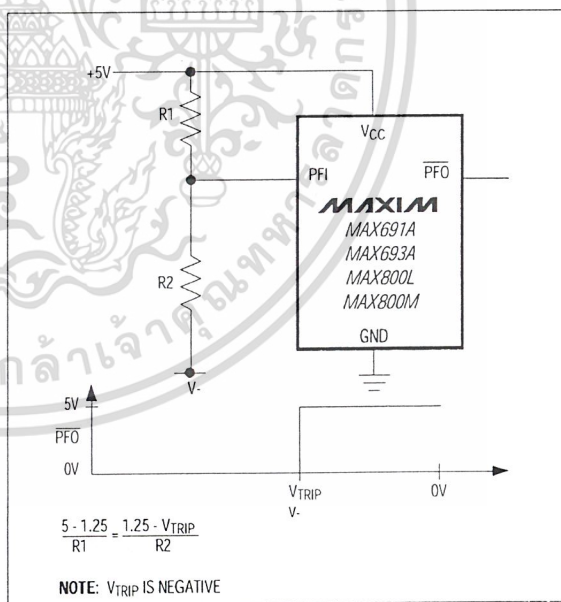


Figure 12. Monitoring a Negative Voltage

MAXIM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Microprocessor Supervisory Circuits

MAX691A/MAX693A/MAX800L/MAX800M

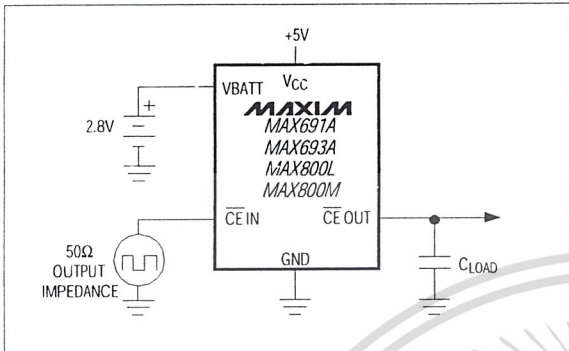


Figure 6. CE Propagation Delay Test Circuit

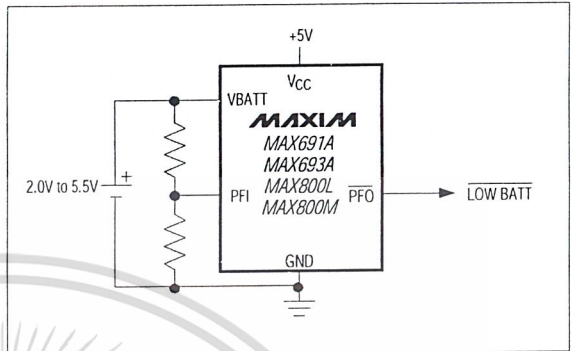


Figure 7. Low-Battery Indicator

Table 2. Input and Output Status in Battery-Backup Mode

PIN	NAME	STATUS
1	VBATT	Supply current is 1μA max.
2	V _{OUT}	V _{OUT} is connected to VBATT through an internal PMOS switch.
3	V _{CC}	Battery switchover comparator monitors V _{CC} for active switchover.
4	GND	GND 0V, 0V reference for all signals.
5	BATT ON	Logic high. The open-circuit output is equal to V _{OUT} .
6	LOWLINE	Logic low*
7	OSC IN	OSC IN is ignored.
8	OSC SEL	OSC SEL is ignored.
9	PFI	The power-fail comparator remains active in the battery-backup mode for V _{CC} ≥ VBATT - 1.2V typ.
10	PFO	The power-fail comparator remains active in the battery-backup mode for V _{CC} ≥ VBATT - 1.2V typ. Below this voltage, PFO is forced low.
11	WDI	Watchdog is ignored.
12	CE OUT	Logic high. The open-circuit voltage is equal to V _{OUT} .
13	CE IN	High impedance
14	WDO	Logic high. The open-circuit voltage is equal to V _{OUT} .
15	RESET	Logic low*
16	RESET	High impedance*

* V_{CC} must be below the reset threshold to enter battery-backup mode.

Power-Fail Output

The Power-Fail Output (PFO) goes low when PFI goes below 1.25V. It typically sinks 3.2mA with a saturation voltage of 0.1V. With PFI above 1.25V, PFO is actively pulled to V_{OUT}.

Battery-Backup Mode

Two conditions are required to switch to battery-backup mode: 1) V_{CC} must be below the reset threshold, and 2) V_{CC} must be below VBATT. Table 2 lists the status of the inputs and outputs in battery-backup mode.

Battery On Output

The Battery On (BATT ON) output indicates the status of the internal V_{CC}/battery switchover comparator, which controls the internal V_{CC} and VBATT switches. For V_{CC} greater than VBATT (ignoring the small hysteresis effect), BATT ON typically sinks 3.2mA at 0.1V saturation voltage. In battery-backup mode, this terminal sources approximately 10μA from V_{OUT}. Use BATT ON to indicate battery switchover status or to supply base drive to an external pass transistor for higher-current applications (see *Typical Operating Circuit*).

Input Supply Voltage

The Input Supply Voltage (V_{CC}) should be a regulated 5V. V_{CC} connects to V_{OUT} via a parallel diode and a large PMOS switch. The switch carries the entire current load for currents less than 250mA. The parallel diode carries any current in excess of 250mA. Both the switch and the diode have impedances less than 1Ω each. The maximum continuous current is 250mA, but power-on transients may reach a maximum of 1A.

MAXIM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Microprocessor Supervisory Circuits

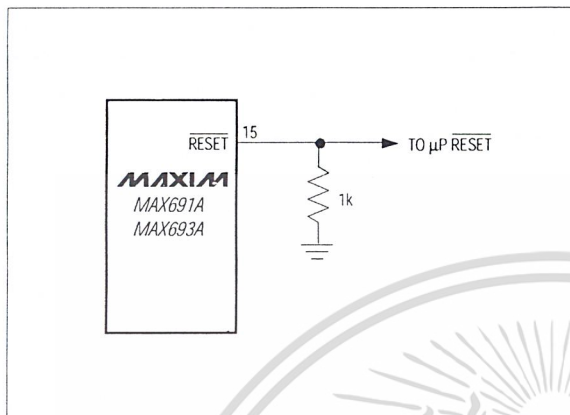


Figure 1. Adding an external pull-down resistor ensures \overline{RESET} is valid with V_{CC} down to GND.

\overline{RESET} and \overline{RESET} are asserted when V_{CC} falls below the reset threshold (4.65V for the MAX691A/MAX800L, 4.4V for the MAX693A/MAX800M) and remain asserted for 200ms typ after V_{CC} rises above the reset threshold on power-up (Figure 5). The devices' battery-switchover comparator does not affect reset assertion. However, both reset outputs are asserted in battery-backup mode since V_{CC} must be below the reset threshold to enter this mode.

Watchdog Function

The watchdog monitors μP activity via the Watchdog Input (WDI). If the μP becomes inactive, \overline{RESET} and \overline{RESET} are asserted. To use the watchdog function, connect WDI to a bus line or μP I/O line. If WDI remains high or low for longer than the watchdog timeout period (1.6sec nominal), \overline{WDO} , \overline{RESET} , and \overline{RESET} are asserted (see *RESET and RESET Outputs* section, and the *Watchdog Output* discussion on this page).

Watchdog Input

A change of state (high to low, low to high, or a minimum 100ns pulse) at the WDI during the watchdog period resets the watchdog timer. The watchdog default timeout is 1.6sec.

To disable the watchdog function, leave WDI floating. An internal resistor network (100k Ω equivalent impedance at WDI) biases WDI to approximately 1.6V. Internal comparators detect this level and disable the watchdog timer. When V_{CC} is below the reset threshold, the watchdog function is disabled and WDI is disconnected from its internal resistor network, thus becoming high impedance.

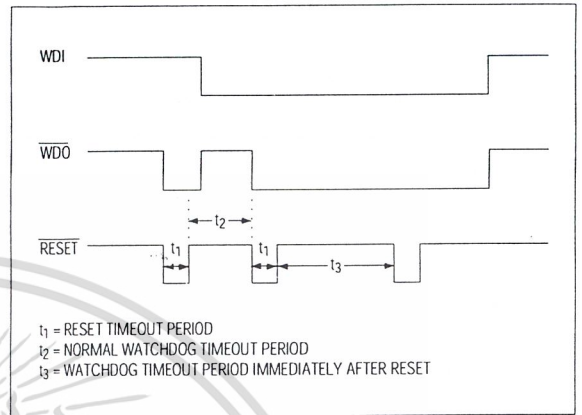


Figure 2. Watchdog Timeout Period and Reset Active Time

t_1 = RESET TIMEOUT PERIOD
 t_2 = NORMAL WATCHDOG TIMEOUT PERIOD
 t_3 = WATCHDOG TIMEOUT PERIOD IMMEDIATELY AFTER RESET

Watchdog Output

The Watchdog Output (\overline{WDO}) remains high if there is a transition or pulse at WDI during the watchdog timeout period. The watchdog function is disabled and \overline{WDO} is a logic high when V_{CC} is below the reset threshold, battery-backup mode is enabled, or WDI is an open circuit. In watchdog mode, if no transition occurs at WDI during the watchdog timeout period, \overline{RESET} and \overline{RESET} are asserted for the reset timeout period (200ms typical). \overline{WDO} goes low and remains low until the next transition at WDI (Figure 2). If WDI is held high or low indefinitely, \overline{RESET} and \overline{RESET} will generate 200ms pulses every 1.6sec. \overline{WDO} has a 2 x TTL output characteristic.

Selecting an Alternative Watchdog and Reset Timeout Period

The OSC SEL and OSC IN inputs control the watchdog and reset timeout periods. Floating OSC SEL and OSC IN or tying them both to V_{OUT} selects the nominal 1.6sec watchdog timeout period and 200ms reset timeout period. Connecting OSC IN to GND and floating or connecting OSC SEL to V_{OUT} selects the 100ms normal watchdog timeout delay and 1.6sec delay immediately after reset. The reset timeout delay remains 200ms (Figure 2). Select alternative timeout periods by connecting OSC SEL to GND and connecting a capacitor between OSC IN and GND, or by externally driving OSC IN (Table 1 and Figure 3). OSC IN is internally connected to a ± 100 nA (typ) current source that charges and discharges the timing capacitor to create the oscillator frequency, which sets the reset and watchdog timeout periods (see *Connecting a Timing Capacitor at OSC IN* in the *Applications Information* section).

MAXIM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Microprocessor Supervisory Circuits

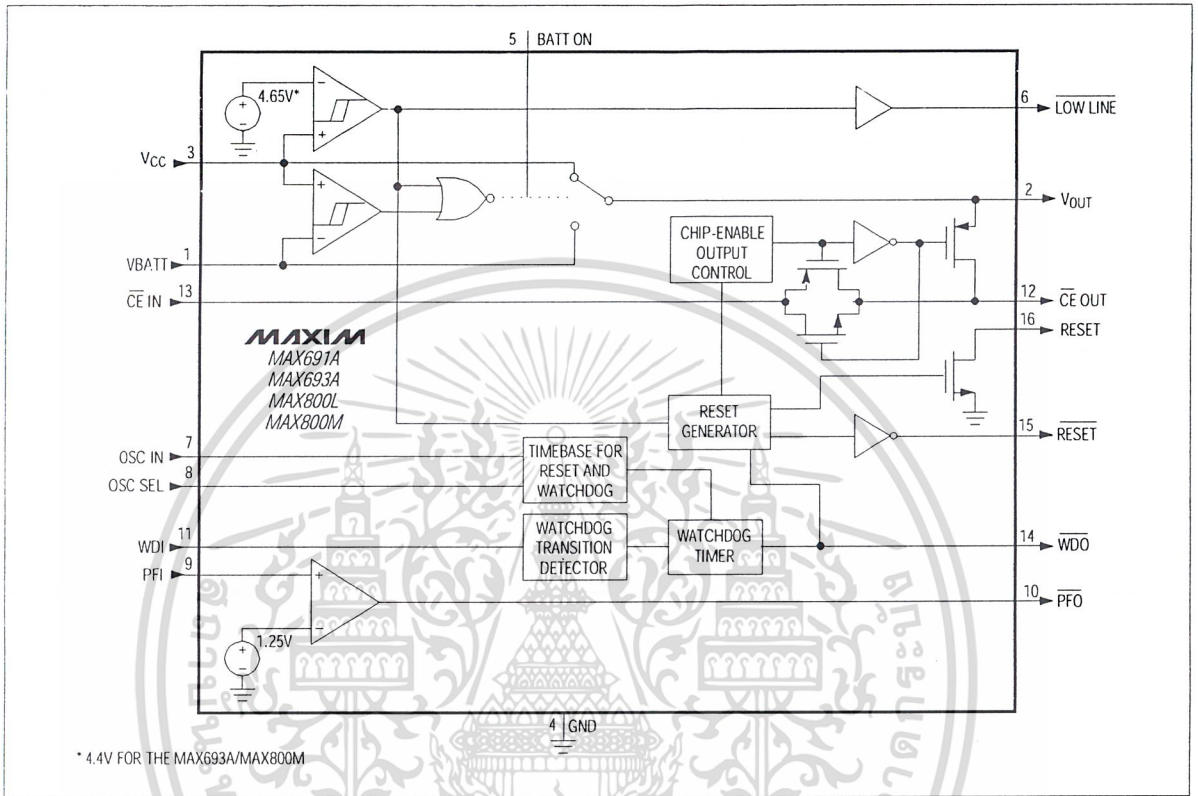


Figure 4. MAX691A/MAX693A/MAX800L/MAX800M Black Diagram

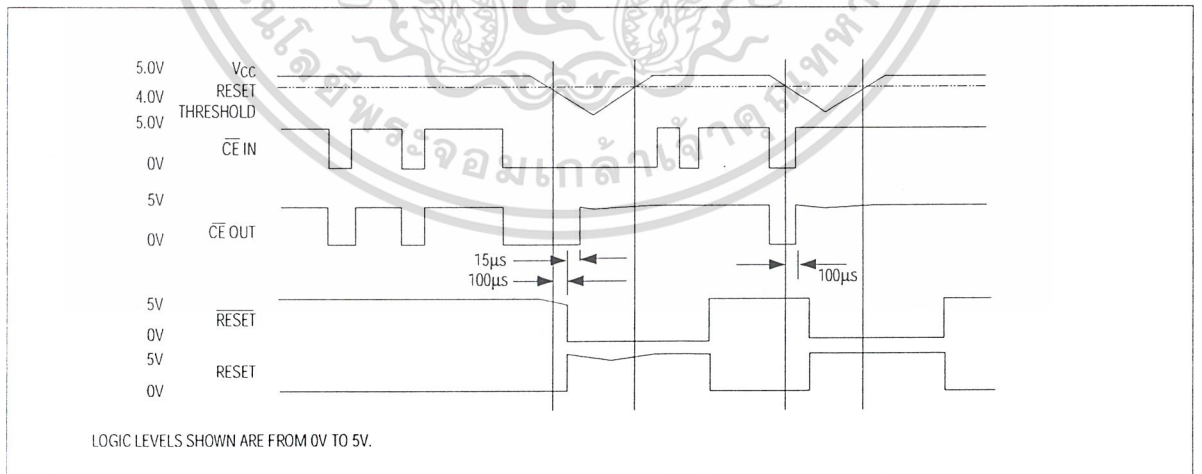


Figure 5. Reset and Chip-Enable Timing

Microprocessor Supervisory Circuits

Maximum V_{CC} Fall Time

The V_{CC} fall time is limited by the propagation delay of the battery switchover comparator and should not exceed 0.03V/μs. A standard rule of thumb for filter capacitance on most regulators is on the order of 100μF per amp of current. When the power supply is shut off or the main battery is disconnected, the associated initial V_{CC} fall rate is just the inverse or 1A/100μF = 0.01V/μs. The V_{CC} fall rate decreases with time as V_{CC} falls exponentially, which more than satisfies the maximum fall-time requirement.

Watchdog Software Considerations

A way to help the watchdog timer keep a closer watch on software execution involves setting and resetting the watchdog input at different points in the program, rather than "pulsing" the watchdog input high-low-high or low-high-low. This technique avoids a "stuck" loop where the watchdog timer continues to be reset within the loop, keeping the watchdog from timing out. Figure 14 shows an example flow diagram where the I/O driving the watchdog input is set high at the beginning of the program, set low at the beginning of every subroutine or loop, then set high again when the program returns to the beginning. If the program should "hang" in any subroutine, the I/O is continually set low and the watchdog timer is allowed to time out, causing a reset or interrupt to be issued.

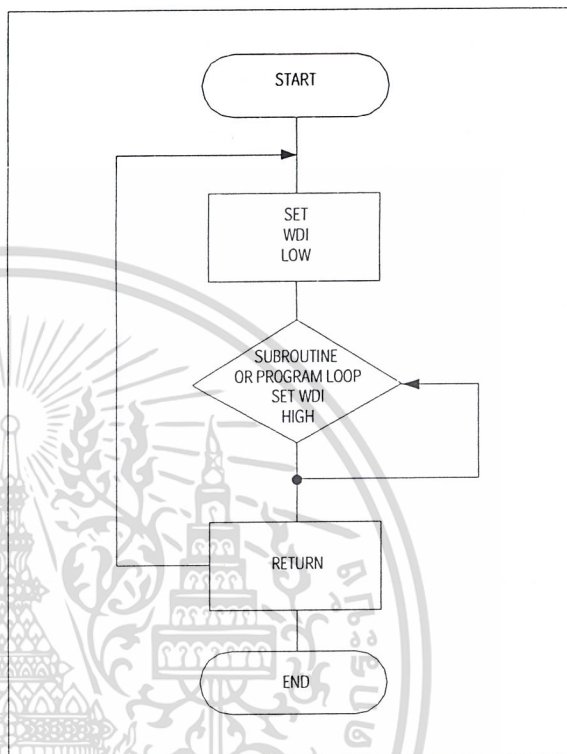


Figure 14. Watchdog Flow Diagram



82C55A CHMOS PROGRAMMABLE PERIPHERAL INTERFACE

- Compatible with all Intel and Most Other Microprocessors
- High Speed, "Zero Wait State" Operation with 8 MHz 8086/88 and 80186/188
- 24 Programmable I/O Pins
- Low Power CHMOS
- Completely TTL Compatible
- Control Word Read-Back Capability
- Direct Bit Set/Reset Capability
- 2.5 mA DC Drive Capability on all I/O Port Outputs
- Available in 40-Pin DIP and 44-Pin PLCC
- Available in EXPRESS
 - Standard Temperature Range
 - Extended Temperature Range

The Intel 82C55A is a high-performance, CHMOS version of the industry standard 8255A general purpose programmable I/O device which is designed for use with all Intel and most other microprocessors. It provides 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. The 82C55A is pin compatible with the NMOS 8255A and 8255A-5.

In MODE 0, each group of 12 I/O pins may be programmed in sets of 4 and 8 to be inputs or outputs. In MODE 1, each group may be programmed to have 8 lines of input or output. 3 of the remaining 4 pins are used for handshaking and interrupt control signals. MODE 2 is a strobed bi-directional bus configuration.

The 82C55A is fabricated on Intel's advanced CHMOS III technology which provides low power consumption with performance equal to or greater than the equivalent NMOS product. The 82C55A is available in 40-pin DIP and 44-pin plastic leaded chip carrier (PLCC) packages.

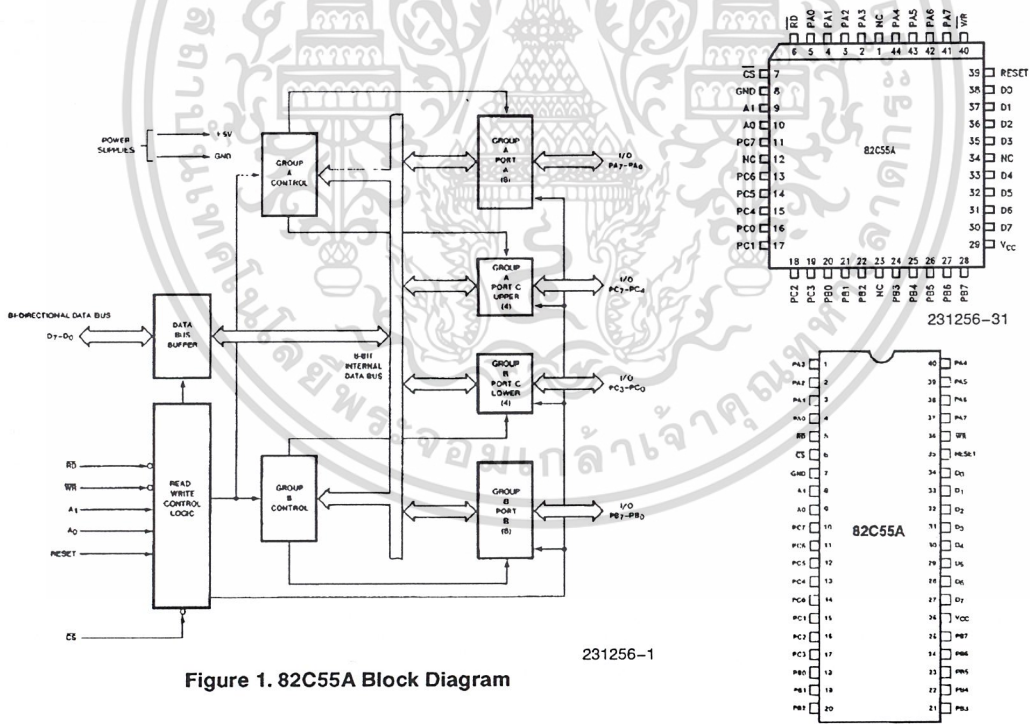


Table 1. Pin Description

Symbol	Pin Number Dip PLCC		Type	Name and Function					
PA ₃₋₀	1-4	2-5	I/O	PORT A, PINS 0-3: Lower nibble of an 8-bit data output latch/buffer and an 8-bit data input latch.					
RD	5	6	I	READ CONTROL: This input is low during CPU read operations.					
CS	6	7	I	CHIP SELECT: A low on this input enables the 82C55A to respond to RD and WR signals. RD and WR are ignored otherwise.					
GND	7	8		System Ground					
A ₁₋₀	8-9	9-10	I	ADDRESS: These input signals, in conjunction RD and WR, control the selection of one of the three ports or the control word registers.					
				Input Operation (Read)					
				A ₁	A ₀	RD	WR	CS	Input Operation (Read)
				0	0	0	1	0	Port A - Data Bus
				0	1	0	1	0	Port B - Data Bus
				1	0	0	1	0	Port C - Data Bus
				1	1	0	1	0	Control Word - Data Bus
				Output Operation (Write)					
				0	0	1	0	0	Data Bus - Port A
				0	1	1	0	0	Data Bus - Port B
				1	0	1	0	0	Data Bus - Port C
1	1	1	0	0	Data Bus - Control				
Disable Function									
X	X	X	X	1	Data Bus - 3 - State				
X	X	1	1	0	Data Bus - 3 - State				
PC ₇₋₄	10-13	11,13-15	I/O	PORT C, PINS 4-7: Upper nibble of an 8-bit data output latch/buffer and an 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.					
PC ₀₋₃	14-17	16-19	I/O	PORT C, PINS 0-3: Lower nibble of Port C.					
PB ₀₋₇	18-25	20-22, 24-28	I/O	PORT B, PINS 0-7: An 8-bit data output latch/buffer and an 8-bit data input buffer.					
V _{CC}	26	29		SYSTEM POWER: + 5V Power Supply.					
D ₇₋₀	27-34	30-33, 35-38	I/O	DATA BUS: Bi-directional, tri-state data bus lines, connected to system data bus.					
RESET	35	39	I	RESET: A high on this input clears the control register and all ports are set to the input mode.					
WR	36	40	I	WRITE CONTROL: This input is low during CPU write operations.					
PA ₇₋₄	37-40	41-44	I/O	PORT A, PINS 4-7: Upper nibble of an 8-bit data output latch/buffer and an 8-bit data input latch.					
NC		1, 12, 23, 34		No Connect					

82C55A FUNCTIONAL DESCRIPTION

General

The 82C55A is a programmable peripheral interface device designed for use in Intel microcomputer systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 82C55A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 82C55A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 82C55A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 82C55A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A - Port A and Port C upper (C7-C4)
Control Group B - Port B and Port C lower (C3-C0)

The control word register can be both written and read as shown in the address decode table in the pin descriptions. Figure 6 shows the control word format for both Read and Write operations. When the control word is read, bit D7 will always be a logic "1", as this implies control word mode information.

Ports A, B, and C

The 82C55A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 82C55A.

Port A. One 8-bit data output latch/buffer and one 8-bit input latch buffer. Both "pull-up" and "pull-down" bus hold devices are present on Port A.

Port B. One 8-bit data input/output latch/buffer. Only "pull-up" bus hold devices are present on Port B.

Port C. One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B. Only "pull-up" bus hold devices are present on Port C.

See Figure 4 for the bus-hold circuit configuration for Port A, B, and C.

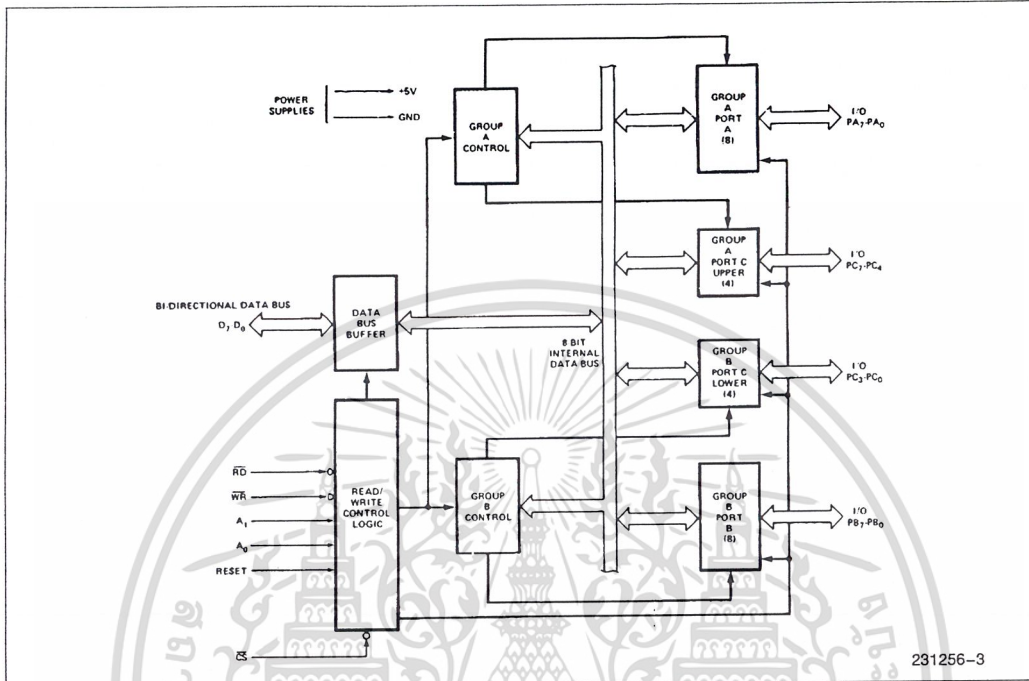


Figure 3. 82C55A Block Diagram Showing Data Bus Buffer and Read/Write Control Logic Functions

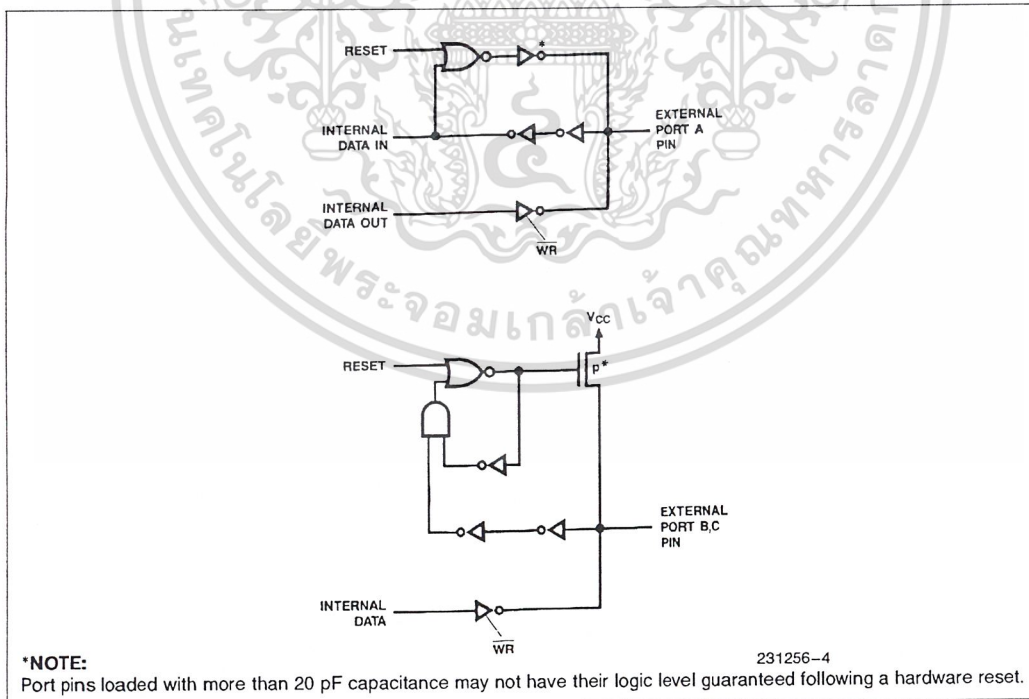


Figure 4. Port A, B, C, Bus-hold Configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

82C55A OPERATIONAL DESCRIPTION

Mode Selection

There are three basic modes of operation that can be selected by the system software:

- Mode 0 — Basic input/output
- Mode 1 — Strobed Input/output
- Mode 2 — Bi-directional Bus

When the reset input goes "high" all ports will be set to the input mode with all 24 port lines held at a logic "one" level by the internal bus hold devices (see Figure 4 Note). After the reset is removed the 82C55A can remain in the input mode with no additional initialization required. This eliminates the need for pullup or pulldown devices in "all CMOS" designs. During the execution of the system program, any of the other modes may be selected by using a single output instruction. This allows a single 82C55A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

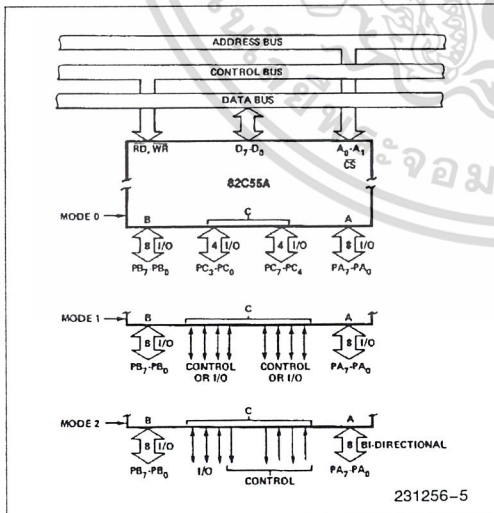


Figure 5. Basic Mode Definitions and Bus Interface

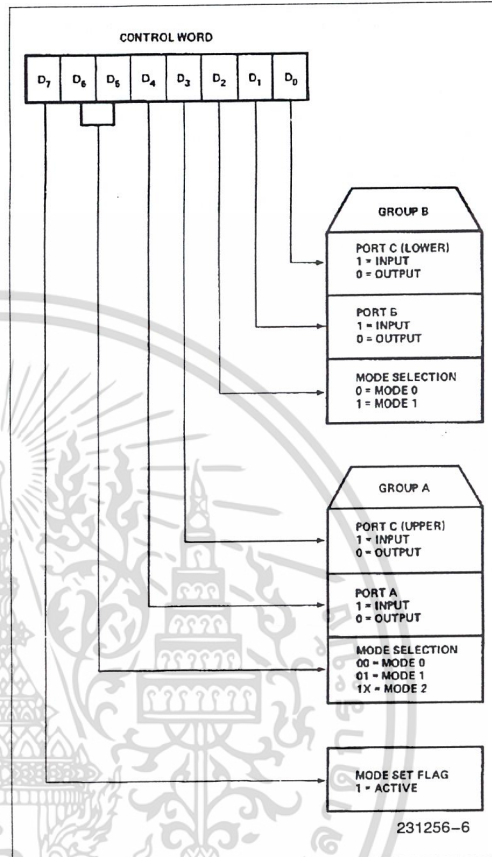


Figure 6. Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 82C55A has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in Control-based applications.

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

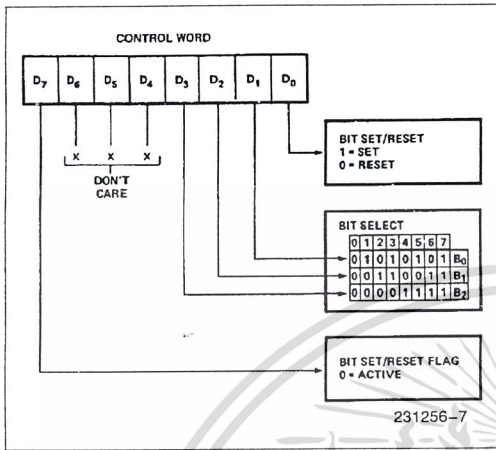


Figure 7. Bit Set/Reset Format

Interrupt Control Functions

When the 82C55A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

INTE flip-flop definition:

- (BIT-SET)—INTE is SET—Interrupt enable
- (BIT-RESET)—INTE is RESET—Interrupt disable

Note:

All Mask flip-flops are automatically reset during mode selection and device Reset.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

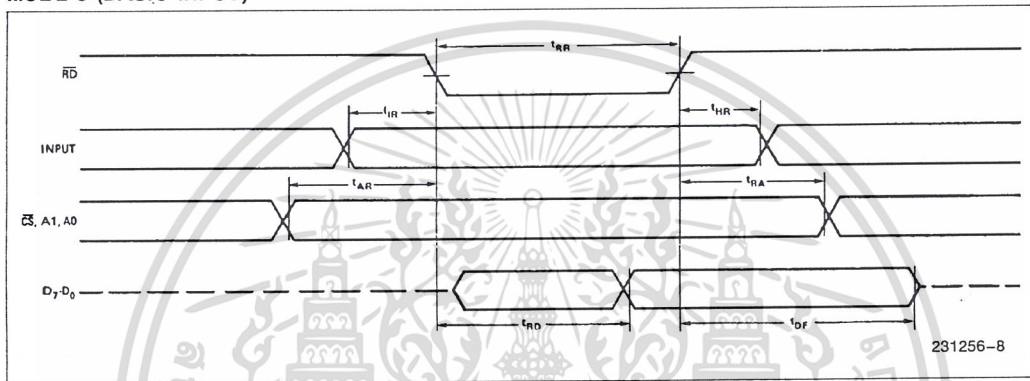
Operating Modes

Mode 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

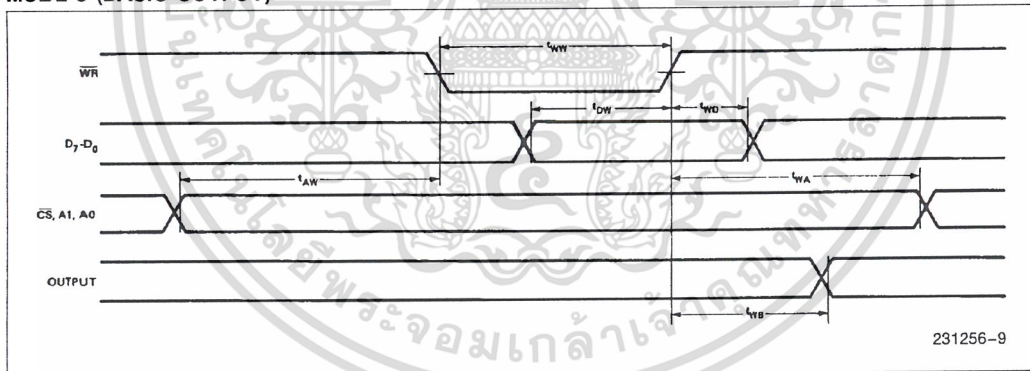
Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.

MODE 0 (BASIC INPUT)



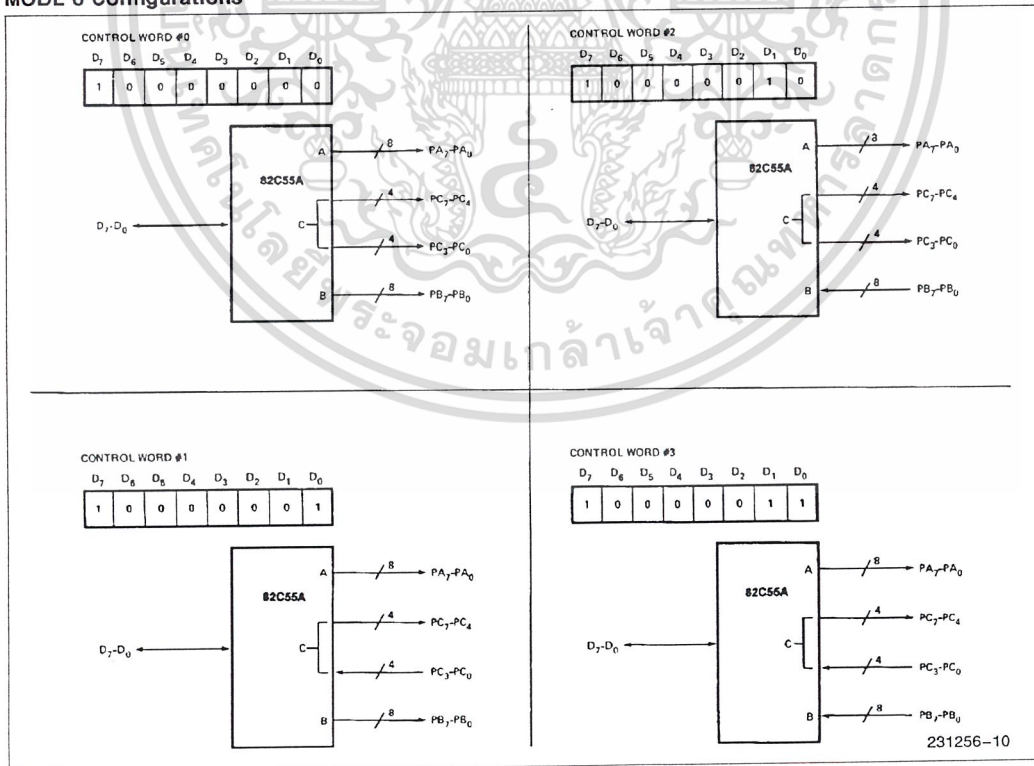
MODE 0 (BASIC OUTPUT)



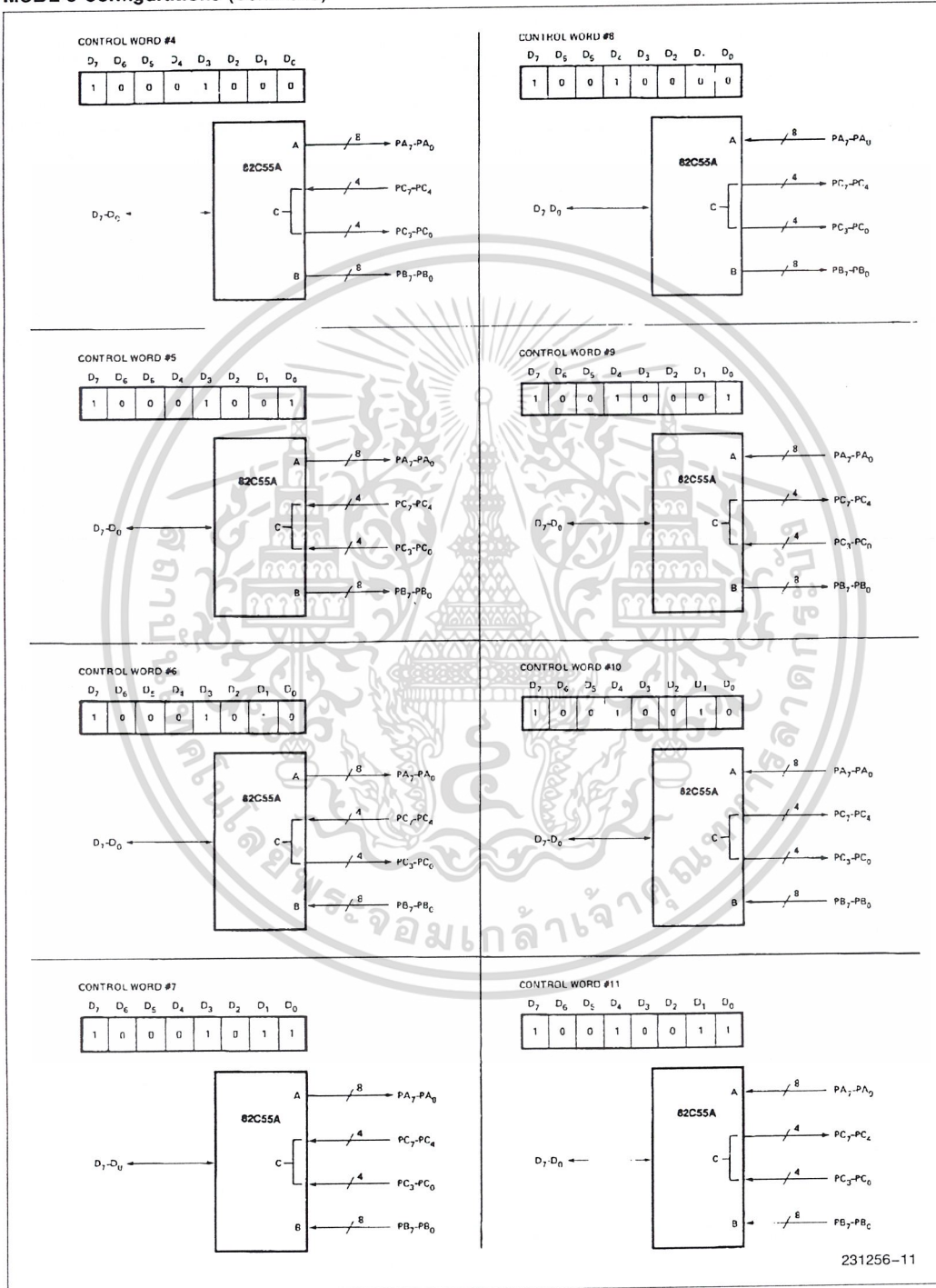
MODE 0 Port Definition

A		B		GROUP A			GROUP B	
D ₄	D ₃	D ₁	D ₀	PORT A	PORT C (UPPER)	#	PORT B	PORT C (LOWER)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

MODE 0 Configurations

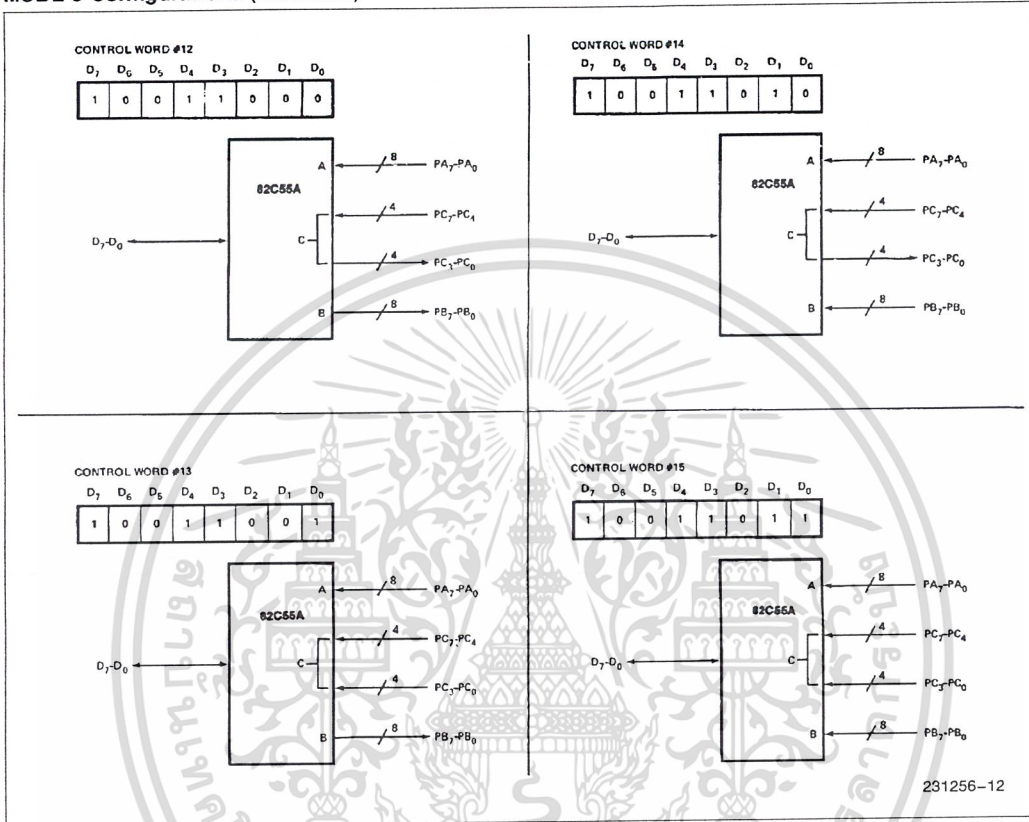


MODE 0 Configurations (Continued)



231256-11

MODE 0 Configurations (Continued)



Operating Modes

MODE 1 (Strobed Input/Output). This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In mode 1, Port A and Port B use the lines on Port C to generate or accept these "handshaking" signals.

Mode 1 Basic functional Definitions:

- Two Groups (Group A and Group B).
- Each group contains one 8-bit data port and one 4-bit control/data port.
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit data port.

Input Control Signal Definition

STB (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch; in essence, an acknowledgement. IBF is set by STB input being low and is reset by the rising edge of the RD input.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the STB is a "one", IBF is a "one" and INTE is a "one". It is reset by the falling edge of RD. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

INTE A

Controlled by bit set/reset of PC4.

INTE B

Controlled by bit set/reset of PC2.

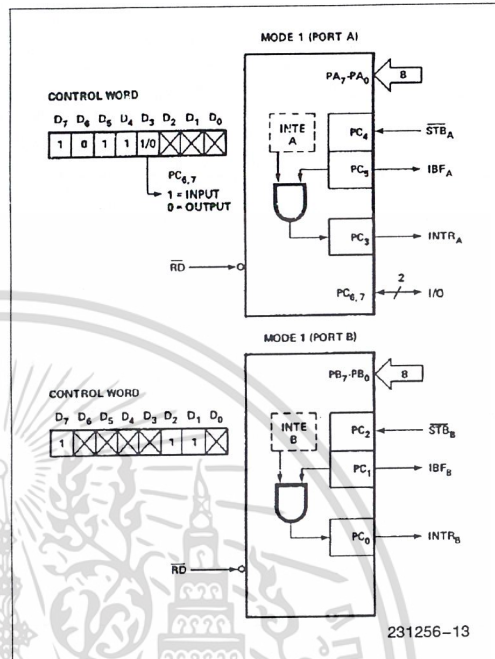


Figure 8. MODE 1 Input

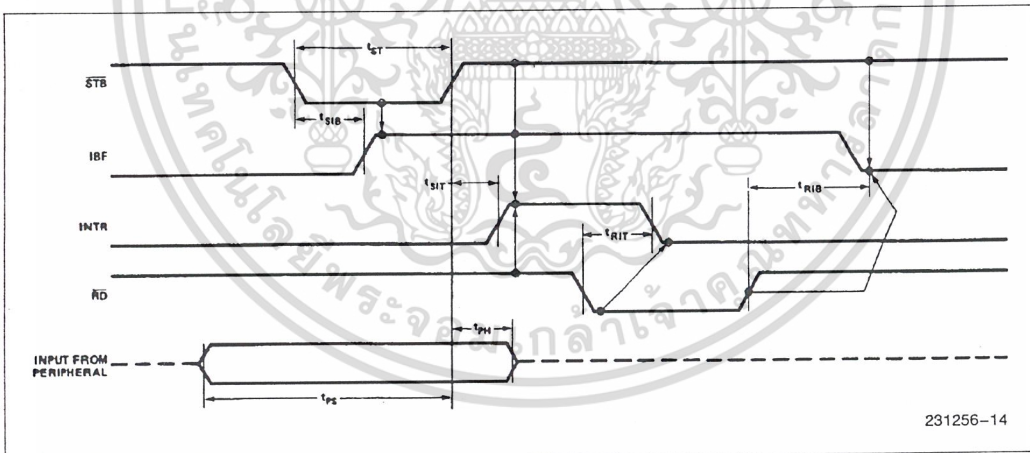


Figure 9. MODE 1 (Strobed Input)

Output Control Signal Definition

OBF (Output Buffer Full F/F). The OBF output will go "low" to indicate that the CPU has written data out to the specified port. The OBF F/F will be set by the rising edge of the WR input and reset by ACK Input being low.

ACK (Acknowledge Input). A "low" on this input informs the 82C55A that the data from Port A or Port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the CPU.

INTR (Interrupt Request). A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when ACK is a "one", OBF is a "one" and INTE is a "one". It is reset by the falling edge of WR.

- INTE A**
Controlled by bit set/reset of PC₆.
- INTE B**
Controlled by bit set/reset of PC₂.

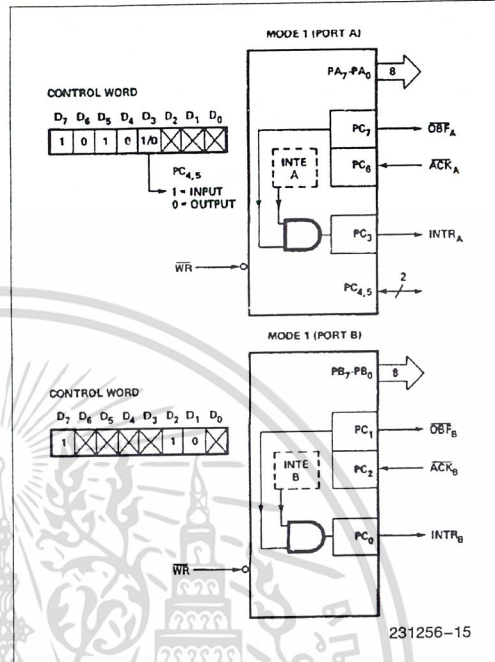


Figure 10. MODE 1 Output

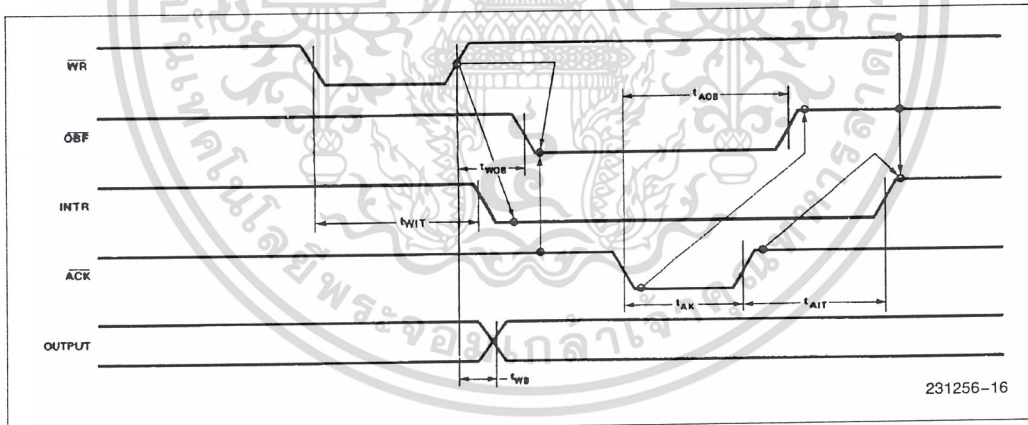


Figure 11. MODE 1 (Strobed Output)

Combinations of MODE 1

Port A and Port B can be individually defined as input or output in Mode 1 to support a wide variety of strobed I/O applications.

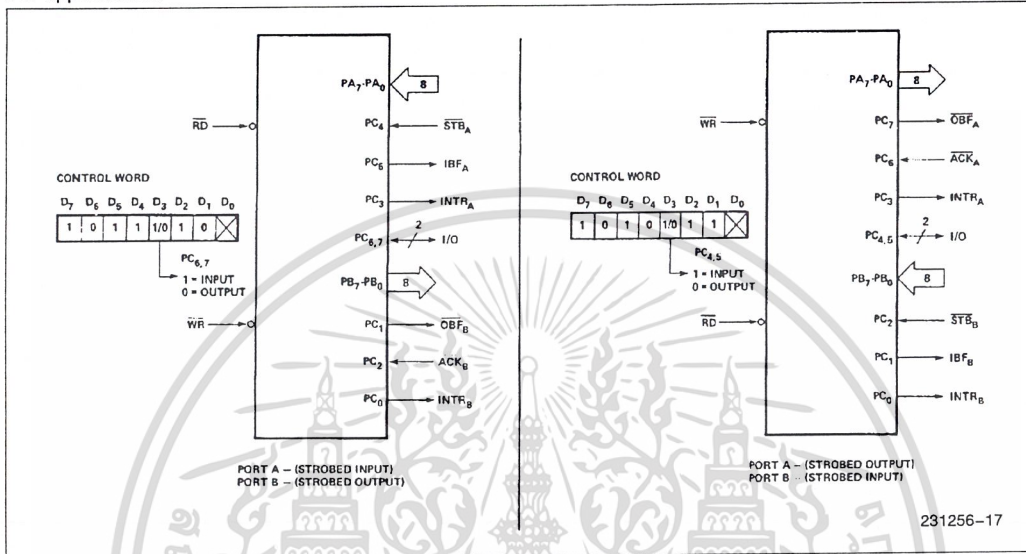


Figure 12. Combinations of MODE 1

Operating Modes

MODE 2 (Strobed Bidirectional Bus I/O). This functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bidirectional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline in a similar manner to MODE 1. Interrupt generation and enable/disable functions are also available.

MODE 2 Basic Functional Definitions:

- Used in Group A only.
- One 8-bit, bi-directional bus port (Port A) and a 5-bit control port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A).

Bidirectional Bus I/O Control Signal Definition

INTR (Interrupt Request). A high on this output can be used to interrupt the CPU for input or output operations.

Output Operations

OBF (Output Buffer Full). The OBF output will go "low" to indicate that the CPU has written data out to port A.

ACK (Acknowledge). A "low" on this input enables the tri-state output buffer of Port A to send out the data. Otherwise, the output buffer will be in the high impedance state.

INTE 1 (The INTE Flip-Flop Associated with OBF). Controlled by bit set/reset of PC₆.

Input Operations

STB (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F). A "high" on this output indicates that data has been loaded into the input latch.

INTE 2 (The INTE Flip-Flop Associated with IBF). Controlled by bit set/reset of PC₄.

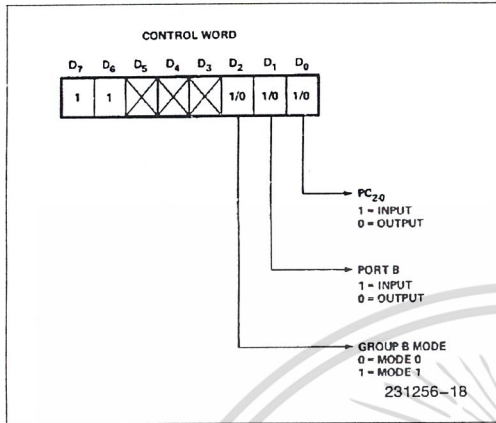


Figure 13. MODE Control Word

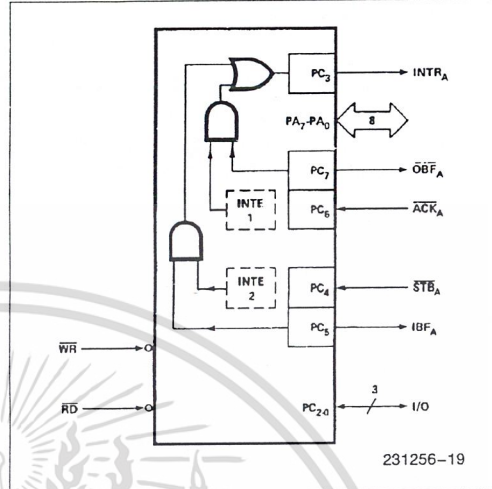


Figure 14. MODE 2

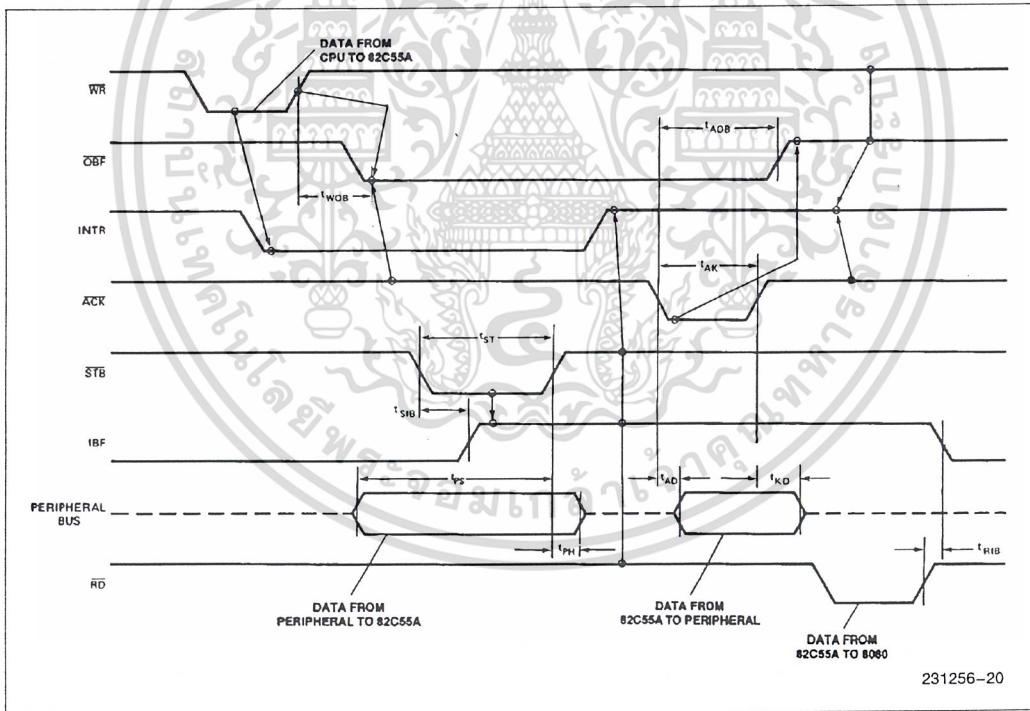


Figure 15. MODE 2 (Bidirectional)

NOTE:

Any sequence where \overline{WR} occurs before \overline{ACK} , and \overline{STB} occurs before \overline{RD} is permissible.
 (INTR = IBF • MASK • STB • RD + OBF • MASK • ACK • WR)

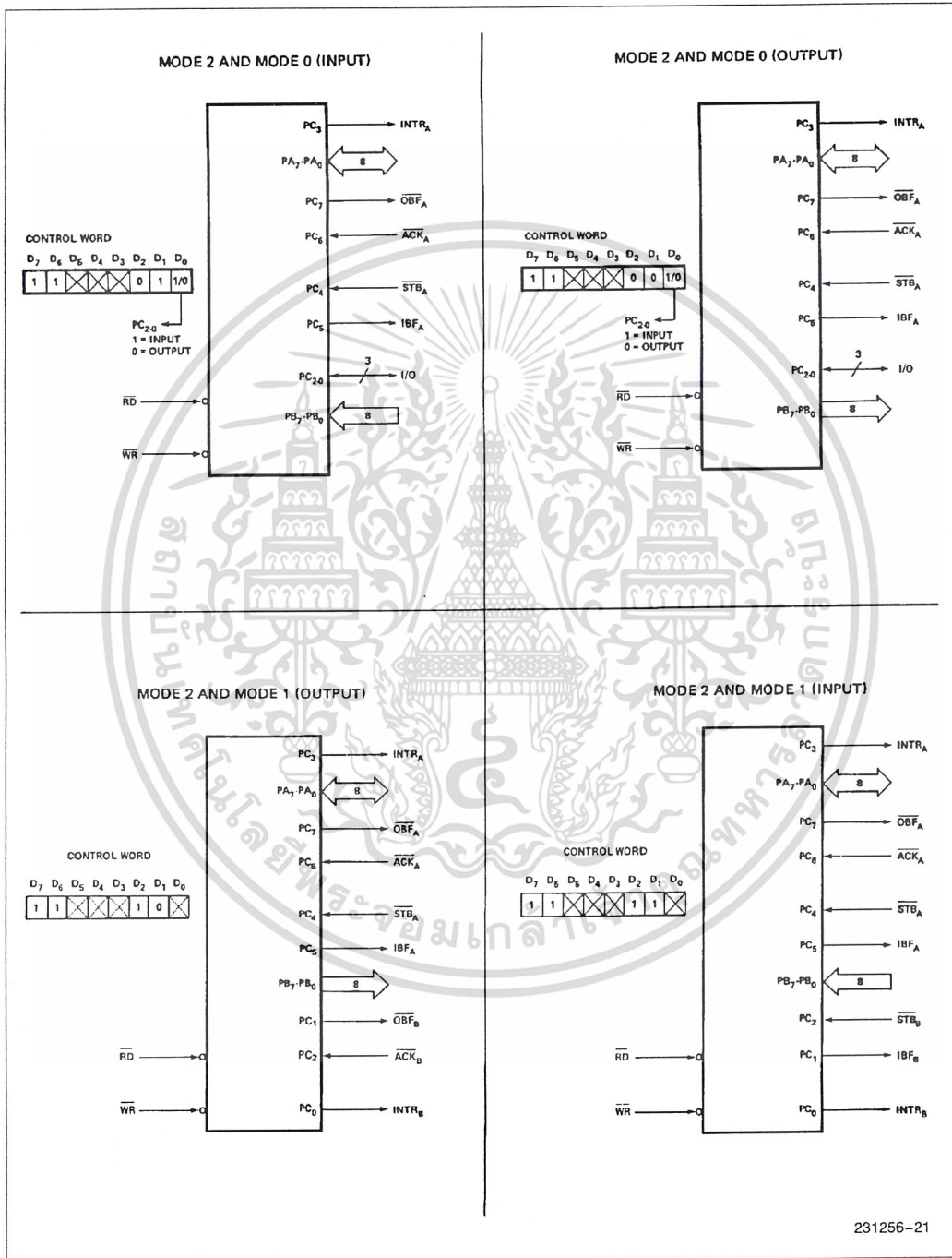


Figure 16. MODE 1/4 Combinations

Mode Definition Summary

	MODE 0		MODE 1		MODE 2
	IN	OUT	IN	OUT	GROUP A ONLY
PA ₀	IN	OUT	IN	OUT	↔
PA ₁	IN	OUT	IN	OUT	↔
PA ₂	IN	OUT	IN	OUT	↔
PA ₃	IN	OUT	IN	OUT	↔
PA ₄	IN	OUT	IN	OUT	↔
PA ₅	IN	OUT	IN	OUT	↔
PA ₆	IN	OUT	IN	OUT	↔
PA ₇	IN	OUT	IN	OUT	↔
PB ₀	IN	OUT	IN	OUT	—
PB ₁	IN	OUT	IN	OUT	—
PB ₂	IN	OUT	IN	OUT	—
PB ₃	IN	OUT	IN	OUT	—
PB ₄	IN	OUT	IN	OUT	—
PB ₅	IN	OUT	IN	OUT	—
PB ₆	IN	OUT	IN	OUT	—
PB ₇	IN	OUT	IN	OUT	—
PC ₀	IN	OUT	INTR _B	INTR _B	I/O
PC ₁	IN	OUT	IBF _B	OBF _B	I/O
PC ₂	IN	OUT	STB _B	ACK _B	I/O
PC ₃	IN	OUT	INTR _A	INTR _A	INTR _A
PC ₄	IN	OUT	STB _A	I/O	STB _A
PC ₅	IN	OUT	IBF _A	I/O	IBF _A
PC ₆	IN	OUT	I/O	ACK _A	ACK _A
PC ₇	IN	OUT	I/O	OBF _A	OBF _A

MODE 0 OR MODE 1 ONLY

Special Mode Combination Considerations

There are several combinations of modes possible. For any combination, some or all of the Port C lines are used for control or status. The remaining bits are either inputs or outputs as defined by a "Set Mode" command.

During a read of Port C, the state of all the Port C lines, except the \overline{ACK} and \overline{STB} lines, will be placed on the data bus. In place of the \overline{ACK} and \overline{STB} line states, flag status will appear on the data bus in the PC₂, PC₄, and PC₆ bit positions as illustrated by Figure 18.

Through a "Write Port C" command, only the Port C pins programmed as outputs in a Mode 0 group can be written. No other pins can be affected by a "Write Port C" command, nor can the interrupt enable flags be accessed. To write to any Port C output programmed as an output in a Mode 1 group or to

change an interrupt enable flag, the "Set/Reset Port C Bit" command must be used.

With a "Set/Reset Port C Bit" command, any Port C line programmed as an output (including INTR, IBF and OBF) can be written, or an interrupt enable flag can be either set or reset. Port C lines programmed as inputs, including \overline{ACK} and \overline{STB} lines, associated with Port C are not affected by a "Set/Reset Port C Bit" command. Writing to the corresponding Port C bit positions of the \overline{ACK} and \overline{STB} lines with the "Set/Reset Port C Bit" command will affect the Group A and Group B interrupt enable flags, as illustrated in Figure 18.

Current Drive Capability

Any output on Port A, B or C can sink or source 2.5 mA. This feature allows the 82C55A to directly drive Darlington type drivers and high-voltage displays that require such sink or source current.

Reading Port C Status

In Mode 0, Port C transfers data to or from the peripheral device. When the 82C55A is programmed to function in Modes 1 or 2, Port C generates or accepts "hand-shaking" signals with the peripheral device. Reading the contents of Port C allows the programmer to test or verify the "status" of each peripheral device and change the program flow accordingly.

There is no special instruction to read the status information from Port C. A normal read operation of Port C is executed to perform this function.

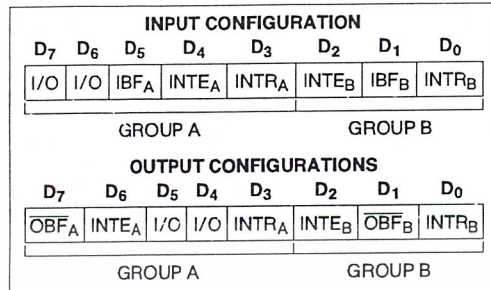


Figure 17a. MODE 1 Status Word Format

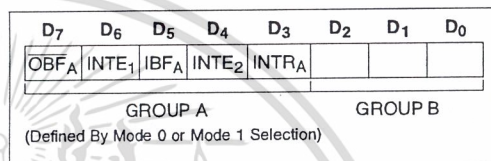
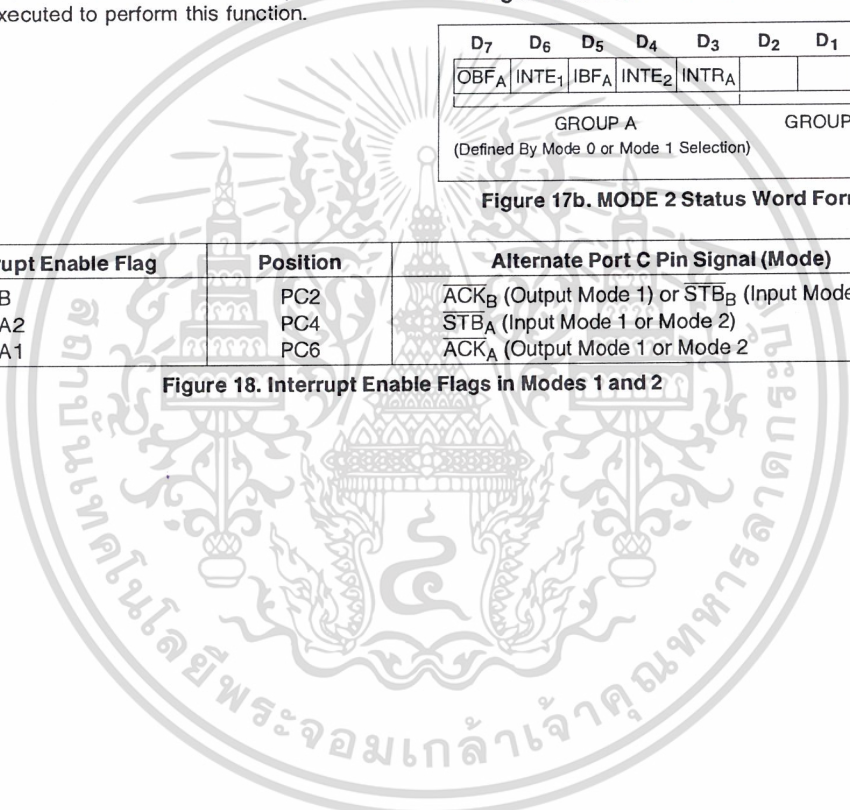


Figure 17b. MODE 2 Status Word Format

Interrupt Enable Flag	Position	Alternate Port C Pin Signal (Mode)
INTE B	PC2	ACK _B (Output Mode 1) or STB _B (Input Mode 1)
INTE A2	PC4	STB _A (Input Mode 1 or Mode 2)
INTE A1	PC6	ACK _A (Output Mode 1 or Mode 2)

Figure 18. Interrupt Enable Flags in Modes 1 and 2



กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ขอขอบพระคุณทุกท่านที่ให้ความช่วยเหลือให้คำปรึกษาในเรื่องต่างๆ แก่คณะผู้จัดทำ โดยเฉพาะอย่างยิ่ง ดร.สุทธิชัย นพนาถพิงษ์ ที่ได้ให้คำแนะนำ ความอนุเคราะห์และความไว้วางใจในการให้ใช้อุปกรณ์และเครื่องมือในการสร้างผลงานนี้อย่างดียิ่ง และที่สำคัญยิ่งขอขอบคุณคุณพ่อและคุณแม่ที่ได้เป็นกำลังใจอันสำคัญทำให้ผลงานชิ้นนี้สามารถสำเร็จเสร็จสิ้นได้อย่างสมบูรณ์



คณะผู้จัดทำ

1. นายชาญยุทธ คิชฐศิริ
2. นายสุริยะ ฉายคุณรัฐ
3. นายแสนศักดิ์ ภาคลำเจียก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

- [1] Albert S. Hogland and Jame E. Monson ., *Digital magnetic recording.* ,Second edition., John willey & Sons,Inc.
- [2] L.G.Sebestyen ., *Digital magnetic tape recording for computer applications.*,Chapman and London.
- [3] Norbert R. Malk., *Electronic circuit analysis,simulatoin and design .* ,Prentice-hall International,Inc.
- [4] Roger Jennings,*Database developer's guide with visual basic3*,Sams publishing,1994.
- [5] Zane Thomas,*Visual Basic 4 How to*,Waite Group Press.
- [6] กิตติ ภักดีวัฒนะกุล,จำลอง ครูอุตสาหะ,*Visual Basic5 ฉบับโปรแกรมเมอร์*, หจก. ไทยเจริญ การพิมพ์ .
- [7] สุทธิศักดิ์ พงษ์ธนาพานิชน์,*Visual Basic5.0 Professional การใช้คำสั่งและคอนโทรล ActiveX*, บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน).
- [8] คณิดา นามประสิทธิ์ และคณะ,*style Microsoft ACCESS 97 Visual Guide*, บริษัท เฟิสท์ แปซิฟิก มีเดีย (ไทยแลนด์) จำกัด.
- [9] รศ. สมยศ จุณณะปิยะ, *การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51*,สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, กรุงเทพมหานคร.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้