

ระบบตรวจสอบการเข้าออกโดยบัตรแม่เหล็ก

Magnetic Card Access Control System



โดย
นายภูษชาติ เดชศิริ รหัส 41012040
นายจิตติพันธุ์ มุกดา รหัส 41012048

เลขหม.....
เลขทะเบียน.....42182
วัน, เดือน, ปี..... 14 พ.ค. 2545

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต
สาขาเทคโนโลยีโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มี **611205878**

หัวข้อปริญญานิพนธ์ ระบบตรวจสอบการเข้าออกโดยบัตรแม่เหล็ก
Magnetic Card Access Control System

ชื่อนักศึกษา นายกุลชาติ เดชศิริ
นายจิตติพันธุ์ มุกดา
อาจารย์ที่ปรึกษา รองศาสตราจารย์นิกร สุขุมตันติ
ผู้ช่วยศาสตราจารย์ไพศาล สิทธิโยภาสกุล
ภาควิชา เทคนิคอุตสาหกรรม
ปีการศึกษา 2542

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
อนุมัติให้นำปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตร
บัณฑิต

คณะกรรมการสอบปริญญานิพนธ์

..... ประธานกรรมการ
()
..... กรรมการ
()
..... กรรมการ
()
..... กรรมการ
()
..... กรรมการ
()
..... กรรมการ
()

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบตรวจสอบการเข้าออกโดยบัตรแม่เหล็ก

โดย นายกุลละชาติ เศษศิริ
นายจิตติพันธ์ มุกดา

อาจารย์ที่ปรึกษา รองศาสตราจารย์นิกร สุขุมตันติ
ผู้ช่วยศาสตราจารย์ไพศาล สิทธิโยภาสกุล

ปีการศึกษา 2542

บทคัดย่อ

เครื่องบันทึกการเข้าออกโดยบัตรแม่เหล็กนี้ จะทำหน้าที่เป็นระบบรักษาความปลอดภัยแก่องค์กรที่ติดตั้ง สามารถเก็บข้อมูลของบุคคลที่เข้าออก แต่บุคคลนั้นจะต้องเป็นบุคคลที่มีบัตรประจำตัว ซึ่งเป็นบัตรสำหรับบังคับตัวผู้ถือบัตรที่ทางองค์กรออกให้ บัตรนั้นจึงจะสามารถรูดผ่านได้ เพราะถ้าเป็นบัตรที่ไม่ใช่บัตรประจำตัวที่ทางองค์กรกำหนดให้ก็ไม่สามารถผ่านไปได้ บัตรที่นำมาใช้นี้เป็นบัตรแบบแถบแม่เหล็ก ซึ่งจะมีข้อมูลที่ Track 2 โครงการนี้ถูกออกแบบให้มีฟังก์ชันที่อำนวยความสะดวกแก่เจ้าของบริษัทห้างร้าน ที่ติดตั้งระบบนี้ ฟังก์ชันต่างๆ ดังได้แก่ ส่วนค้นหาข้อมูลผู้เข้าออก ส่วนเพิ่มรายละเอียดสมาชิก ที่สำคัญโครงการนี้สามารถมีเสียงพูดเมื่อผู้ผ่านเข้าออกรูดผ่านโปรแกรมควบคุมถูกพัฒนาโดย Visual Basic เพราะติดต่อกับอุปกรณ์ภายนอกได้ง่ายโดยผ่านพอร์ตอนุกรม RS-232

ส่วนที่เป็น Interface ก็ใช้ไมโครคอนโทรลเลอร์เบอร์ AT89C2051 เป็นคอนโทรลเลอร์ตระกูล MCS-51 แบบ flash สามารถโปรแกรมได้ 1000 ครั้ง นอกจากจะทำหน้าที่ Interface ระหว่างคอมพิวเตอร์กับเครื่องอ่านบัตรแล้ว คอนโทรลเลอร์ยังมีหน้าที่ควบคุมโซลินอยด์สำหรับเปิด-ปิดประตู เมื่อผู้ผ่านเข้าออกใช้บัตรแม่เหล็กที่เป็นบัตรประจำตัว

Magnetic Card Access Control System

BY MR.Kulachat Detsiri
MR.Chittiphun Mukda

ADVISOR Associate.Prof.Nikorn Sukutamantanti
Assistant.Prof.Paisan Sityopasakul

YEAR 1999

ABSTRACT

Magnetic Card Access Control System will be security system part for organization and keep data of member, who has ID card. Which it uses for identify a person. Only ID cards are added data to Database can slide to pass other card can not to pass. Magnetic cards have data at track2. This project is designed to have convenience function for user Online function, Add data function and Find data are example of a function. Human voice is a highlight of this project. It is like a receptionist for a member to pass. It is speck when a member sliding the card into slot. Software is program by Visual Basic. So easily to interfere with other device by series port (RS-232) interfacing

Microcontroller number AT89C2051 used to interfacing between computer and magnetic card reader and control Solinoid to close or open the door when member sliding the card. It is MCS-51 flash type and it can burn for 1000 times.

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้ได้สำเร็จลุล่วงไปได้ด้วยดี ก็เพราะอาศัยความรู้จากบูรพาจารย์ทั้งหลาย คำแนะนำจากอาจารย์ที่ปรึกษาที่คอยให้คำปรึกษาและข้อมูลที่เป็นแหล่งได้ค้นคว้าหาความรู้ทั้งจากเอกสารวารสารอ้างอิงต่างๆ และที่บรรณารักษ์ที่ให้ข้อมูลและเอื้อเพื่อความสะดวกในการยืมหนังสือ

แม้ผู้จัดทำจะพยายามจัดทำปริญญาบัตรฉบับนี้ด้วยความตั้งใจเป็นอย่างดีและพยายามที่จะทำให้ดีที่สุด แต่ก็อาจจะมีจุดบกพร่องและไม่สมบูรณ์ ทางผู้จัดทำยินดีรับคำแนะนำเพื่อนำไปแก้ไขปรับปรุงให้สมบูรณ์ดียิ่งขึ้นไป

สุดท้ายขอขอบคุณท่านอาจารย์ที่ปรึกษาที่ได้แนะนำและให้คำปรึกษาที่ดี อาจารย์ทุกท่านที่ให้ความรู้มาจนถึงวันนี้ พี่ๆ ทุกคนให้ความช่วยเหลือและให้คำแนะนำด้านการเขียนโปรแกรม ขอขอบคุณบิดา - มารดา ที่สนับสนุนด้านต่างๆ และเป็นกำลังใจให้ลูกเสมอ

นายกุลชาติ เชนศิริ

นายจิตติพันธ์ มุกดา

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูปภาพ	ฉ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 เป้าหมายของโครงการ	1
1.4 วิธีดำเนินงาน	2
1.5 ประโยชน์ที่ได้รับ	2
บทที่ 2 ทฤษฎี	3
2.1 คำนำ	3
2.2 คุณสมบัติของบัตรแม่เหล็ก	3
2.3 การบันทึกข้อมูลลงบนบัตรแม่เหล็ก	6
2.4 การอ่านข้อมูลจากบัตรแม่เหล็ก	10
2.5 การอินเตอร์รัปต์	17
2.6 พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51	24
2.7 วงจรมัลติไวเบรเตอร์	40
2.8 Visual Basic เบื้องต้น	47
บทที่ 3 การออกแบบ	56
3.1 บล็อกไดอะแกรม	56
3.2 ส่วนเครื่องอ่านบัตรแม่เหล็ก	57
3.3 ส่วนบอร์ดควบคุม	62
3.4 ส่วนไมโครคอมพิวเตอร์หรือโปรแกรม	64

สารบัญ (ต่อ)

หน้า

บทที่ 4 ผลการทดลอง

74

บทที่ 5 สรุปผลการทดลอง

82

บรรณานุกรม

ภาคผนวก ก

ภาคผนวก ข

ภาคผนวก ค



สารบัญรูปลูกภาพ

	หน้า
บทที่ 2 ทฤษฎี	3
รูปที่ 2.1 แสดงตำแหน่งแตรีกที่สองของบัตรแม่เหล็ก (ISO 7811/5)	4
รูปที่ 2.2 แสดงหัวบันทึกและแถบแม่เหล็กในกระบวนการบันทึก ข้อมูลลงบัตรแม่เหล็ก	6
รูปที่ 2.3 แสดงสนามแม่เหล็กรอบช่องว่างของหัวบันทึก	7
รูปที่ 2.4 แสดงกระแสพัลส์, รูปแบบของแตรีกที่ถูกบันทึก, ฟลักซ์แม่เหล็ก ที่พื้นผิวแตรีกข้อมูลและลักษณะแรงดัน ไฟฟ้าของการบันทึก	7
รูปที่ 2.5 แสดงรูปแบบของการบันทึก โดยใช้สัญญาณความถี่สอง ความถี่ที่มีเฟสตรงกัน	8
รูปที่ 2.6 แสดงรูปวงจรพื้นฐานสำหรับการบันทึกข้อมูลบนบัตรแม่เหล็ก	9
รูปที่ 2.7 แสดงรูปสัญญาณการทำงานของวงจรพื้นฐานสำหรับการบันทึก ข้อมูลบนบัตรแม่เหล็ก	9
รูปที่ 2.8 กราฟแสดงคุณสมบัติการอิมิตัวของแม่เหล็กบนบัตรแม่เหล็ก มาตรฐาน	10
รูปที่ 2.9 แสดงสัญญาณแรงดัน ไฟฟ้าจากหัวอ่านและถูกขยายสัญญาณ ประมาณ 100 เท่า	10
รูปที่ 2.10 แสดงส่วนประกอบของวงจรพื้นฐานสร้างสัญญาณรูปแบบ ISO จากบัตรแม่เหล็ก	11
รูปที่ 2.11 แสดงวงจรขยายสัญญาณแรงดัน ไฟฟ้าสำหรับอ่านข้อมูล บัตรแม่เหล็ก	12
รูปที่ 2.12 แสดงการเปรียบเทียบสัญญาณและลักษณะสัญญาณที่ต้องการ	12
รูปที่ 2.13 แสดงวงจรเลื่อนเฟสและวงจรเปรียบเทียบแรงดัน ไฟฟ้า	13
รูปที่ 2.14 แสดงวงจร Differentiator และวงจรเปรียบเทียบ แรงดัน ไฟฟ้า	13
รูปที่ 2.15 แสดงแผนภาพของวงจรอ่านบัตรแม่เหล็ก	14

สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 2.16 แสดงรูปคลื่นสัญญาณของ ISO ที่ถูกมอดูเลต	15
รูปที่ 2.17 แสดงวงจรสร้างสัญญาณข้อมูลและสัญญาณนาฬิกาอ่านข้อมูล โดยใช้ 8031	16
รูปที่ 2.18 แสดงรูปสัญญาณเอาต์พุตของ 8031 สำหรับสร้างสัญญาณ อ่านบัตรแม่เหล็ก	16
รูปที่ 2.19 แสดงโครงสร้างของกลไกการเกิดอินเตอร์รัปต์จากภายนอก	20
รูปที่ 2.20 แสดงเงื่อนไขในการเกิดอินเตอร์รัปต์จากพอร์ตอนุกรมด้วย การออร์กันของบิต RI และ TI	21
รูปที่ 2.21 แสดงเงื่อนไขในการเกิดอินเตอร์รัปต์ของไทเมอร์/คาน์เตอร์ 2 ด้วยการออร์กันของบิต TF2 และ EXF2	21
รูปที่ 2.22 โครงสร้างของระบบอินเตอร์รัปต์ภายในไมโครคอนโทรลเลอร์ MCS-51	23
รูปที่ 2.23 แบบข้อมูลอนุกรมแบบอะซิงโครนัส	24
รูปที่ 2.24 ไดอะแกรมการทำงานในโหมด 0 ของพอร์ตอนุกรมภายใน ไมโครคอนโทรลเลอร์ MCS - 51	28
รูปที่ 2.25 ไดอะแกรมการทำงานในโหมด 1 ของพอร์ตอนุกรมภายใน ไมโครคอนโทรลเลอร์ MCS - 51	30
รูปที่ 2.26 ไดอะแกรมการทำงานในโหมด 2 ของพอร์ตอนุกรมภายใน ไมโครคอนโทรลเลอร์ MCS - 51	32
รูปที่ 2.27 ไดอะแกรมการทำงานในโหมด 3 ของพอร์ตอนุกรมภายใน ไมโครคอนโทรลเลอร์ MCS - 51	33
รูปที่ 2.28 รายละเอียดเบื้องต้นของไอซีแปลงสัญญาณเพื่อเชื่อมต่อกับ	38
รูปที่ 2.29 วงจรเชื่อมต่อ MAX 232 หรือ ICL232 เข้ากับพอร์ตอนุกรม ของคอมพิวเตอร์และไมโคร	39
รูปที่ 2.30 วงจรไบสเตเบิล	40
รูปที่ 2.31 วงจรโมนอสเตเบิล	42
รูปที่ 2.32 วงจรอะสเตเบิล	42

สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 2.33 การตอบสนองต่อสัญญาณกระตุ้นของวงจรโมโนสเตเบิล ทั้งสองแบบ	43
รูปที่ 2.34 วงจรกระตุ้นสำหรับ โมโนสเตเบิล	43
รูปที่ 2.35 วงจรโมโนสเตเบิลที่สร้างจากออปแอมป์	44
รูปที่ 2.36 ไอซีเบอร์ 555 เป็นโมโนสเตเบิลแบบหนึ่ง	45
รูปที่ 2.37 โมโนสเตเบิลที่เป็นไอซีทีทีแอล	46
บทที่ 3 การออกแบบ	56
รูปที่ 3.1 บล็อกแสดงหลักการของเครื่องบันทึกการเข้าออกด้วย บัตรแม่เหล็ก	56
รูปที่ 3.2 แสดงการต่อบอร์ดกับคอมพิวเตอร์	57
รูปที่ 3.3 เครื่องอ่านแถบแม่เหล็ก	57
รูปที่ 3.4 ภาพแสดงความสัมพันธ์ของสัญญาณ DATA ,CLOCK และ PRESENT ของเครื่องอ่านแถบแม่เหล็ก	59
บทที่ 4 ผลการทดลอง	74
รูป 4.1 รูปแสดงการรันซึ่งเป็นเสมือน Main Menu ที่รอรับ การเลือกฟังก์ชันการใช้งาน	74
รูปที่ 4.2 เมื่อClickที่ Menu “เปิดระบบ On line” จะ Load Form ใหม่ขึ้นมา	75
รูปที่ 4.3 เมื่อรูคบัตรที่มีข้อมูลใน Database ก็จะแสดงข้อมูลของผู้ถือบัตร	76
รูปที่ 4.4 เมื่อทำการคิดที่ Menu “เพิ่มข้อมูลสมาชิก” ก็จะ Load Form ใหม่ขึ้นมา	76
รูปที่ 4.5 เมื่อClickที่ Menu “แสดงข้อมูล” จะทำการ Load Form ที่แสดง ข้อมูลการเข้า-ออก	77
รูปที่ 4.6 แสดงเมื่อมีการClickที่ Menu “หมายเหตุ” ทำหน้าที่เป็น Help	78

บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงการ

ในปัจจุบันบ้านเราได้มีการนำเทคโนโลยีของบัตรแถบแม่เหล็กได้ถูกนำมาใช้อย่างแพร่หลายไม่ว่าธุรกิจทางการเงินการธนาคาร ที่นำมาใช้ในรูปแบบของ ATM , VISA และบัตรเครดิตเป็นต้น ธุรกิจการเงินการธนาคารได้มีการนำเทคโนโลยีด้านนี้มาใช้อย่างชัดเจนไม่เฉพาะด้านนี้อย่างเดียวที่ได้มีการนำบัตรแถบแม่เหล็กมาใช้ บัตรแถบแม่เหล็กยังได้นิยมนำไปใช้อย่างมากมายหลากหลายรูปแบบ เพราะคุณสมบัติของบัตรแถบแม่เหล็กที่สามารถเก็บข้อมูลในลักษณะของสนามแม่เหล็กทำให้ข้อมูลสูญหายยาก ปลอดภัยทางข้อมูล ขนาดเล็กกระทัดรัด สะดวกสบาย ค่าใช้จ่ายก็ไม่มากนัก รูปแบบหนึ่งคือใช้เป็นบัตรผ่านประตู ใช้เป็นบัตรสมาชิก (Membercard)

หนึ่งในรูปแบบของบัตรสมาชิกเห็นจะเป็น บัตรประจำตัวประชาชน (ID Card) ปัจจุบันในไทยเราได้มีการนำเทคโนโลยีบัตรแม่เหล็กมาใช้กับบัตรประจำตัวประชาชน ทำให้เราสะดวกสบายรวดเร็ว และสามารถนำไปใช้ใน ธุรกรรมได้หลายประเภท อาทิเช่นใช้เป็นบัตร ATM ได้ในบัตรเดียวกัน บัตรนักศึกษาที่เช่นเดียวกันได้นำรูปแบบของบัตรแม่เหล็กมาใช้เหมือนกัน ดังนั้นเราจะพบว่า บัตรแม่เหล็กได้มีการใช้อย่างแพร่หลายหลากหลายรูปแบบใช้งาน

1.2 วัตถุประสงค์ของโครงการ

1. ศึกษาการเก็บข้อมูลของบัตรแถบแม่เหล็ก
2. ศึกษาการอ่านข้อมูลของบัตรแถบแม่เหล็ก
3. ศึกษาโปรโตคอลของข้อมูลที่อ่านได้จากเครื่องอ่านแถบแม่เหล็ก
4. ออกแบบบอร์ดควบคุมเพื่อใช้ Interface ตัวอ่านแถบแม่เหล็กกับเครื่องคอมพิวเตอร์
5. ศึกษาการพัฒนาโปรแกรมโดยใช้ Visual Basic เพื่อ นำข้อมูลที่ได้จากบัตรแม่เหล็กมาประยุกต์ใช้กับระบบฐานข้อมูล
6. เขียนโปรแกรมควบคุมและบันทึกรายการเข้า-ออก อาคาร

1.3 เป้าหมายของโครงการ

จะใช้บัตรแม่เหล็ก ร่วมกับโปรแกรมควบคุมและบันทึก รายชื่อและเวลาการเข้าอาคาร โดยใช้โปรแกรม Visual Basic Version 6 เป็นตัวเขียนโปรแกรม และสามารถเก็บข้อมูลรายชื่อผู้ที่ต้องเข้าอาคาร ชื่อของผู้ที่ต้องใช้อาคารและยังควบคุม การเปิด-ปิดประตูเข้าออกอาคารบันทึกรายชื่อและเวลาการเข้าออกของผู้ที่เข้าอาคาร

1.4 วิธีการดำเนินงาน

1. ศึกษาการอ่านข้อมูลเครื่องอ่านบัตรแม่เหล็ก
2. ศึกษาการเชื่อมโยง (Interface) ข้อมูลระหว่างเครื่องอ่านบัตรแม่เหล็กกับคอมพิวเตอร์
3. เขียนโปรแกรมควบคุมและบันทึก รายการเวลาการเข้าออกอาคาร

1.5 ประโยชน์ที่ได้รับ

1. ช่วยลดขั้นตอนการลงเวลาการทำงาน
2. เพิ่มประสิทธิภาพในการรักษาความปลอดภัย
3. สามารถนำความรู้เกี่ยวกับการอ่านข้อมูลของบัตรแม่เหล็กไปประยุกต์ด้านอื่นๆ ได้



บทที่ 2

ทฤษฎี

2.1 คำนำ

ในการนำบัตรแม่เหล็กมาใช้งานเป็นบัตรตรวจสอบการเข้าออก โดยใช้ข้อมูลบนบัตรแม่เหล็กเป็นข้อมูลประจำตัวผู้ถือบัตร สำหรับการพัฒนาระบบในขั้นตอนนี้ ได้นำบัตรแม่เหล็กของธนาคารพาณิชย์ต่างๆ มาประยุกต์ใช้งานกับระบบ ซึ่งบัตรแม่เหล็กดังกล่าวมีรูปแบบของข้อมูลที่บันทึกอยู่บนแถบแม่เหล็กและคุณลักษณะต่างๆ ของบัตรแม่เหล็ก เป็นไปตามมาตรฐานสากล (International Organization for Standardization : ISO) โดยรายละเอียดของบัตรแม่เหล็กจะนำเสนอในหัวข้อ 2.2 สำหรับรูปแบบการอ่านข้อมูลจากแถบแม่เหล็กได้นำเสนอการอ่านข้อมูลจากบัตรแม่เหล็กเพื่อเชื่อมต่อกับคอมพิวเตอร์ ซึ่งรายละเอียดจะนำเสนอในหัวข้อ 2.4

2.2 คุณสมบัติของบัตรแม่เหล็ก

แถบแม่เหล็กของบัตรแม่เหล็กของบัตรประชาชนชนิดใหม่ประเทศไทย, จะมีแทร็ก (Track) บันทึกข้อมูลจำนวนสามแทร็ก โดยแทร็กที่สองของแถบแม่เหล็กบันทึกตัวเลขที่อ้างอิงกับหมายเลขบัญชีของผู้ถือบัตร เพื่อใช้เป็นข้อมูลดิบ (Raw Data) สำหรับการคำนวณรหัส PIN (Personal Identification Number) ของบัตรแม่เหล็กใบนั้นๆ โดยข้อมูลในแทร็กที่สองนี้ จะเป็นตัวเลขชุดเดียวกันกับตัวเลขที่ปั๊มอยู่บนบัตรแม่เหล็ก อาทิ บัตรแม่เหล็กของธนาคารกรุงเทพจำกัด, บัตรเครดิตและวีซ่า (Visa) เป็นต้น จากข้อมูลดังกล่าวบนบัตรจะบันทึกข้อมูลและรายละเอียดต่างๆ ของบัตรไว้ในรูปของเส้นแรงแม่เหล็ก (Flux) โดยแถบแม่เหล็กบนบัตรซึ่งเรียกว่าแทร็กนั้นจะถูกแบ่งออกเป็น 3 ส่วน ซึ่งแต่ละส่วนจะใช้เก็บข้อมูลซึ่งมีความหนาแน่น และลักษณะข้อมูลที่มีความแตกต่างกัน

ตารางที่ 2.1 แสดงลักษณะของส่วนที่ใช้เก็บข้อมูลที่มีความแตกต่างกัน

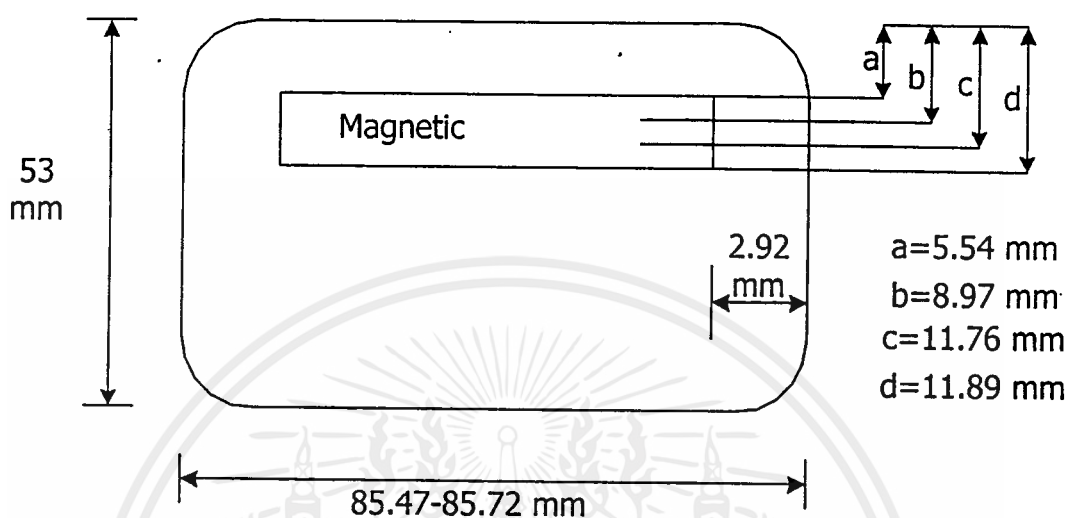
แถบแม่เหล็ก	ความหนาแน่น	การเข้ารหัส	จำนวนตัวอักษร	ลักษณะของข้อมูลที่เก็บ
TRACK 1	210 BPI	ALPHA	79	ชื่อเจ้าของบัตรและหมายเลขบัตร
TRACK 2	75 BPI	BCD	40	หมายเลขบัตรและวันหมดอายุ
TRACK 3	210 BPI	BCD	107	หมายเลขบัตรและรหัสพิเศษ

หมายเหตุ ความหนาแน่นของการบันทึกข้อมูลมีหน่วยเป็น BPI (Bit Per Inch)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 ตำแหน่งแทร็กที่สองของบัตรแม่เหล็ก

ตำแหน่งแทร็กที่สองของบัตรแม่เหล็ก เป็นไปตามมาตรฐาน ISO 7811/5 ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 แสดงตำแหน่งแทร็กที่สองของบัตรแม่เหล็ก (ISO 7811/5)

จากภาพที่ 2.1 บริเวณของแทร็กที่สอง จะอยู่ระหว่างเส้นขนาน (b) ค่า 8.97 มิลลิเมตร (0.353 นิ้ว) (ค่าสูงสุด) หรือค่า 8.46 มิลลิเมตร (0.333 นิ้ว) (ค่าต่ำสุด) กับเส้นขนาน c ค่า 11.76 มิลลิเมตร (0.463 นิ้ว)

2.2.2 ชุดรหัสข้อมูลในแทร็กที่สอง

ข้อมูลที่บันทึกในแทร็กที่สองของแถบแม่เหล็กเป็นตัวเลขอย่างเดียวโดยที่ตัวเลขหนึ่งตัวจะประกอบด้วยบิต (Bit) ข้อมูลแบบ BCD 4 บิต และบิตภาวะเสมอมูล (Parity Bit) 1 บิต ซึ่งใช้ในการตรวจสอบข้อมูลของแต่ละตัวเลข โดยตรวจสอบแบบภาวะเสมอมูลคือ ISO ได้ระบุจำนวนข้อมูลสูงสุดที่สามารถบันทึกในแทร็กที่สองไว้ไม่เกิน 40 ตัว (รวมสัญลักษณ์เริ่มและสิ้นสุด) แสดงในตารางที่ 2.2 ส่วนชุดรหัสข้อมูลตัวเลขแต่ละตัวสำหรับแทร็กที่สอง แสดงในตารางที่ 2.3

2.2.3 รูปแบบของข้อมูลที่บันทึกแทร็กที่สองบนบัตรแม่เหล็ก

SS	PAN	FS	Addition Data	ES	LRC
----	-----	----	---------------	----	-----

ตารางที่ 2.2 รูปแบบข้อมูลที่บันทึกแทร็กที่สองของบัตรแม่เหล็กทั่วไป

SS = รหัสข้อมูลแสดงการเริ่มต้นของข้อมูลในบิต (Start Sentinel) มีรหัสข้อมูลเป็น 0BH (;)

PAN= ข้อมูลแสดงหมายเลขของบิต (มีจำนวนข้อมูลส่วนนี้ไม่เกิน 19 หลัก)

FS = รหัสข้อมูลแสดงการแบ่งแยกข้อมูล (Field Separator) มีรหัสข้อมูลเป็น 0DH (=)

Additional Data = เป็นข้อมูลเพิ่มเติมต่างๆของบิตเช่น วัน/เดือน/ปี ที่ออกบิตและหมดอายุ

ES = รหัสข้อมูลแสดงการสิ้นสุดของข้อมูลในบิต (End Sentinel) มีรหัสเป็น 0FH (?)

LRC = ข้อมูลตรวจสอบความผิดพลาด (Longitudinal Redundancy Check)

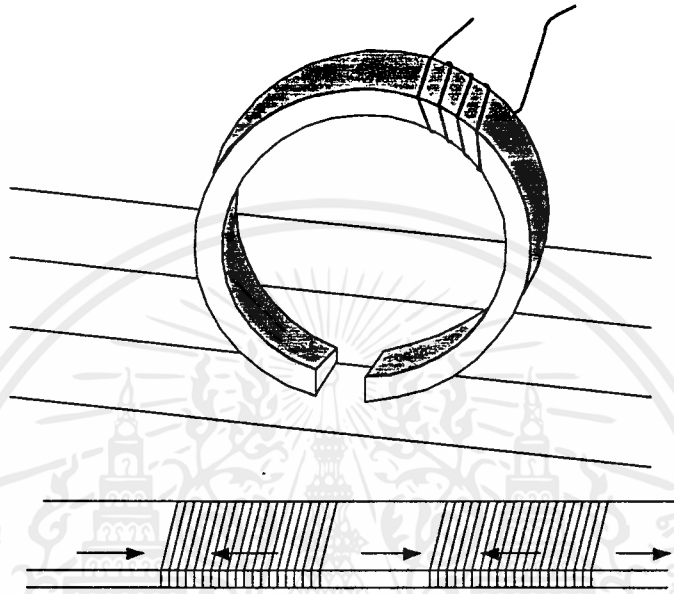
Parity	D3	D2	D1	D0	Character	Function
1	0	0	0	0	0 (0H)	Data
0	0	0	0	1	1(1H)	Data
0	0	0	1	0	2(2H)	Data
1	0	0	1	1	3(3H)	Data
0	0	1	0	0	4(4H)	Data
1	0	1	0	1	5(5H)	Data
1	0	1	1	0	6(6H)	Data
0	0	1	1	1	7(7H)	Data
0	1	0	0	0	8(8H)	Data
1	1	0	0	1	9(9H)	Data
1	1	0	1	0	:(AH)	Control
0	1	0	1	1	;(BH)	Start Sentinel
1	1	1	0	0	<(CH)	Control
0	1	1	0	1	=(DH)	Field Separator
0	1	1	1	0	>(EH)	Control
1	1	1	1	1	? (FH)	End Sentinel

ตารางที่ 2.3 แสดงข้อมูลของรหัส BCD สำหรับเครื่องอ่านแถบแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

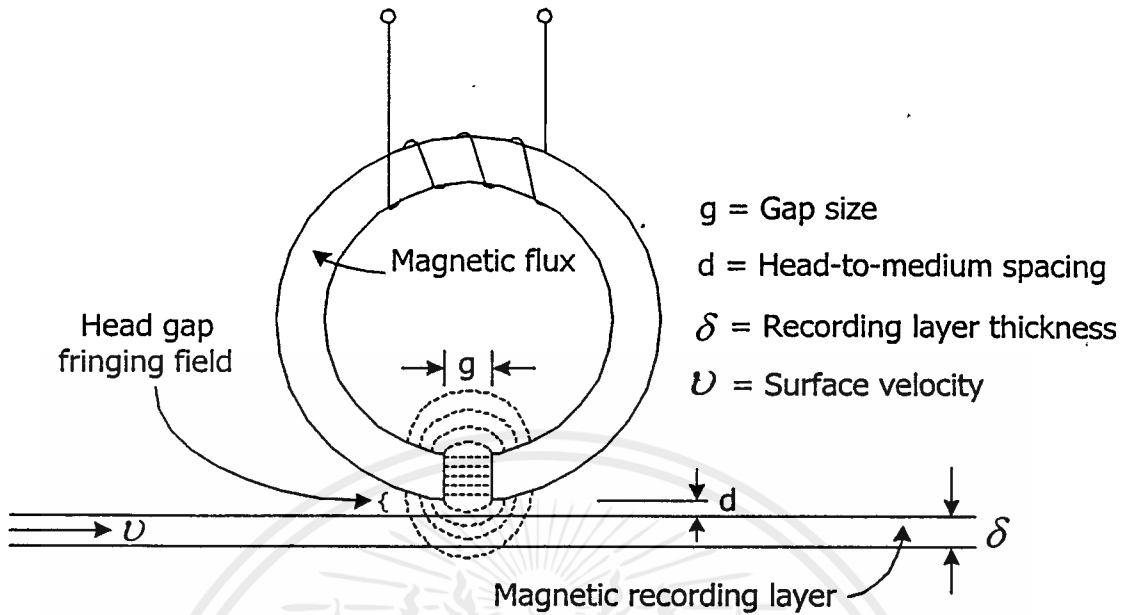
2.3 การบันทึกข้อมูลลงบนบัตรแม่เหล็ก

การบันทึกข้อมูลลงบนบัตรแม่เหล็กประกอบด้วย แถบแม่เหล็กที่ใช้เป็นตัวกลางและหัวแม่เหล็กที่ใช้เป็นหัวบันทึก ซึ่งเคลื่อนที่ด้วยความเร็วคงที่ ดังแสดงในรูปที่ 2.2



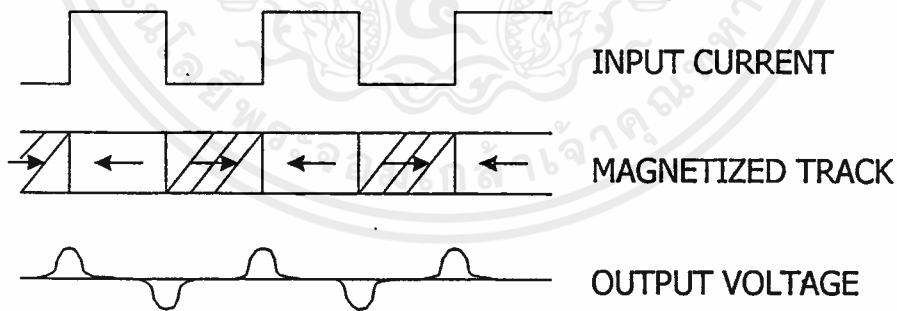
รูปที่ 2.2 แสดงหัวบันทึกและแถบแม่เหล็กในกระบวนการบันทึกข้อมูลลงบัตรแม่เหล็ก

จากรูปที่ 2.2 การบันทึกข้อมูลลงบนบัตรแม่เหล็ก จะขึ้นอยู่กับคุณสมบัติของสารแม่เหล็กในแถบแม่เหล็กและหัวแม่เหล็กที่นำมาใช้เป็นหัวบันทึก โดยที่แกนของหัวบันทึกทั่วไปจะเป็นแกนที่มีคุณสมบัติทางแม่เหล็กอย่างอ่อนๆ พันด้วยขดลวดและที่แกนมีช่องว่าง (Grasp) อยู่บริเวณของสารแม่เหล็กที่จะถูกบันทึกข้อมูลเรียกว่าแทร์ก โดยแต่ละแทร์กจะเรียงขนานกันบนแถบแม่เหล็ก สัญญาณเอาต์พุตจากขดลวดที่พันรอบแกน จะเป็นสัดส่วนกับอัตราการเปลี่ยนแปลงฟลักซ์ (Flux) ของหัวบันทึก ความกว้างของหัวบันทึกและความกว้างของแทร์ก การบันทึกข้อมูลลงบนบัตรแม่เหล็กจะใช้วิธีการป้อนพัลส์ (Pulse) ทั้งด้านบวกและด้านลบ พร้อมทั้งมีขนาดเพียงพอเข้าที่ขดลวดของหัวบันทึกที่วางอยู่ใกล้กับแถบแม่เหล็ก เมื่อป้อนกระแสพัลส์จะก่อให้เกิดสนามแม่เหล็กรอบบริเวณช่องว่างของแกนหัวบันทึกที่วางอยู่ใกล้กับแถบแม่เหล็ก ซึ่งสนามแม่เหล็กนี้ใช้ในการบันทึกข้อมูล ดังแสดงในรูปที่ 2.3 ส่วนรูปแบบของกระแสพัลส์ แรงดันไฟฟ้า และหัวแม่เหล็กในแถบแม่เหล็กเมื่อมีการบันทึกแล้ว แสดงในรูปที่ 2.4



รูปที่ 2.3 แสดงสนามแม่เหล็กรอบช่องว่างของหัวบันทึก

จากรูปที่ 2.3 ค่าความกว้างของช่องว่างต้องมีค่าน้อยกว่าความกว้างของแทร็กข้อมูลเสมอ โดย ISO ได้ระบุความกว้างของช่องว่างของหัวบันทึก บันทึกไว้มีค่าประมาณ 0.00625 มิลลิเมตร (0.00025 นิ้ว) หรือน้อยกว่าและค่าช่องว่างของหัวอ่านมีค่าประมาณ 0.025 มิลลิเมตร (0.001 นิ้ว) หรือน้อยกว่าสำหรับค่าความหนาของเนื้อแถบแม่เหล็ก (δ) ISO ระบุไว้มีค่าไม่เกิน 0.038 มิลลิเมตร



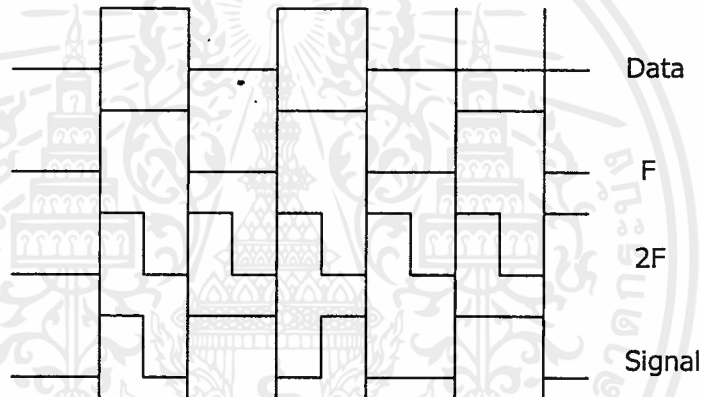
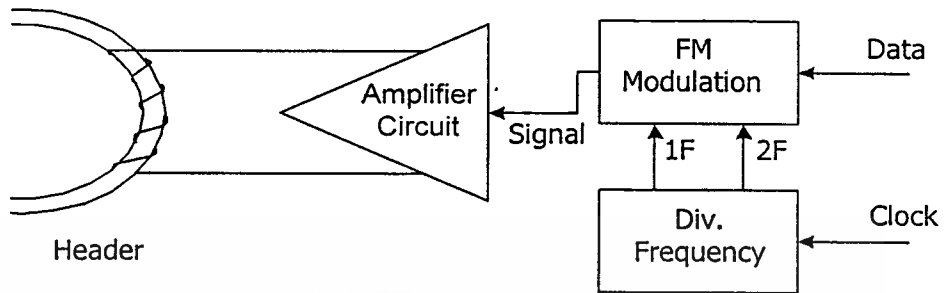
รูปที่ 2.4 แสดงกระแสพัลส์, รูปแบบของแทร็กที่ถูกบันทึก, พลักซ์แม่เหล็กที่พื้นผิวแทร็กข้อมูลและลักษณะแรงดันไฟฟ้าของการบันทึก

2.3.1 รูปแบบของการบันทึกข้อมูลบนบัตรแม่เหล็ก

รูปแบบพื้นฐานสำหรับการบันทึกข้อมูลบนบัตรแม่เหล็ก จะใช้สัญญาณความถี่สองความถี่ ซึ่งมีเฟส (Phase) ตรงกันมาใช้ในการบันทึกข้อมูล ดังแสดงไว้ในรูปที่ 2.5 ซึ่งการบันทึกในลักษณะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่นนี้จะบันทึกข้อมูล และสัญญาณนาฬิกาที่ใช้ในการบันทึกเข้าไว้ในแทร็กเดียวกัน ซึ่งคิดค้นโดย Aiken (1954)

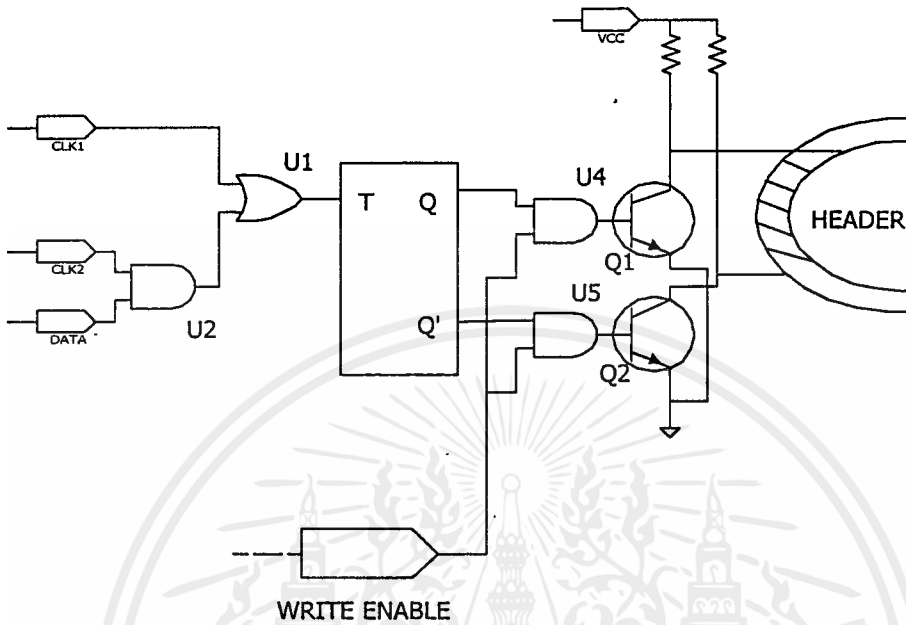


รูปที่ 2.5 แสดงรูปแบบของการบันทึกโดยใช้สัญญาณความถี่สองความถี่ที่มีเฟสตรงกัน

จากรูปที่ 2.5 สัญญาณบันทึกจะประกอบด้วยบิตของข้อมูลที่บันทึก และบิตของสัญญาณนาฬิกา ($2F$) สำหรับการเปลี่ยนแปลงของฟลักซ์แม่เหล็กของสัญญาณบันทึก จะเกิดขึ้นในระหว่างที่สัญญาณข้อมูลมีสถานะ “1” ทางตรรก (Logic) หรือระหว่างสัญญาณบิตของข้อมูลติดกันและจะไม่มีการเปลี่ยนแปลงของฟลักซ์แม่เหล็กในระหว่างที่สัญญาณข้อมูลมีสถานะ “0” ทางตรรก สำหรับการบันทึกบิตข้อมูลของแต่ละตัวเลขหรือตัวอักษรบนบัตรแม่เหล็ก จะเป็นการบันทึกข้อมูลที่มีค่านัยสำคัญต่ำสุด (b1) ก่อนและบิตสถานะเสมอมูล (p) เป็นบิตสุดท้ายที่ถูกบันทึก ส่วนค่าความหนาแน่นในการบันทึกข้อมูลในแทร็กที่สอง ISO ได้กำหนดให้บันทึกด้วยความหนาแน่น 75 ± 20 บิตต่อนิ้ว

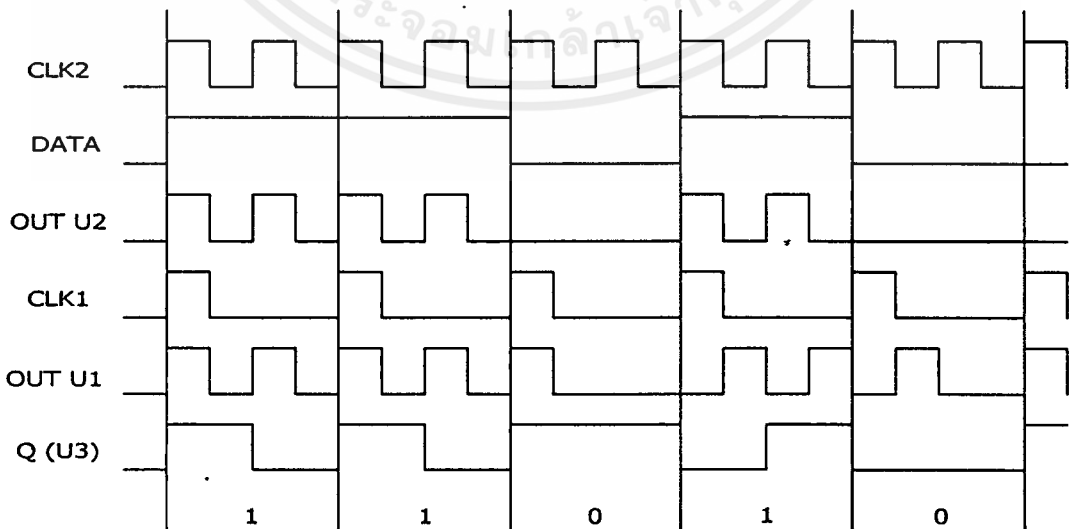
2.3.2 วงจรพื้นฐานสำหรับการบันทึกข้อมูลบนบัตรแม่เหล็ก

วงจรพื้นฐานสำหรับการบันทึกข้อมูลบนบัตรแม่เหล็ก แสดงในรูปที่ 2.6



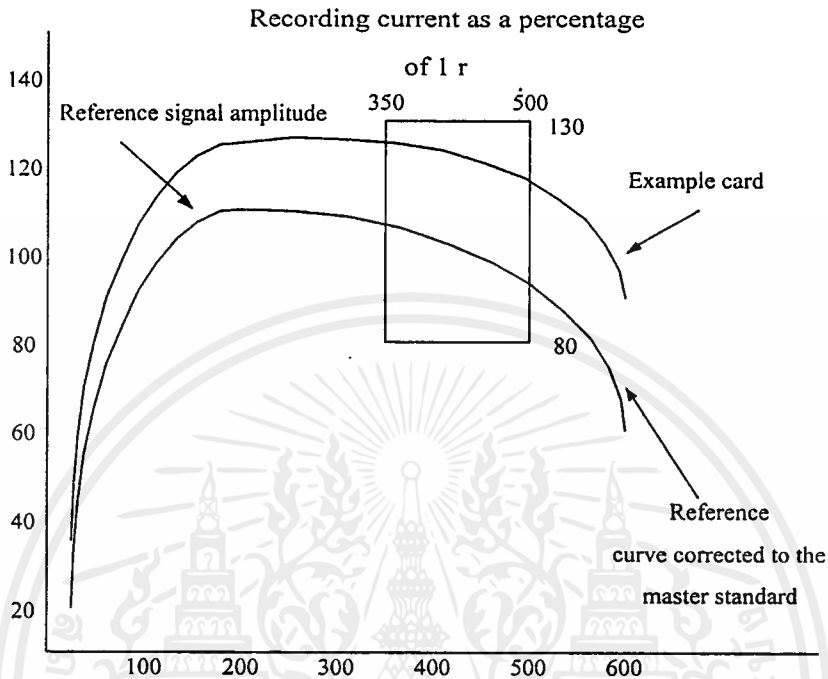
รูปที่ 2.6 แสดงรูปวงจรวจรพื้นฐานสำหรับการบันทึกข้อมูลบนบัตรแม่เหล็ก

จากรูปที่ 2.6 ไอซี U1, U2, U3, U4, และ U5 จัดรูปวงจรมอดูเลท (Modulation Circuit) สัญญาณข้อมูลและสัญญาณนาฬิกา ให้อยู่ในรูปแบบของสัญญาณเอพเอ็ม แล้วป้อนสัญญาณดังกล่าวให้กับวงจรจ่ายกระแสไฟฟ้าให้กับขดลวดที่พันรอบแกนของหัวบันทึก เมื่อทำการบันทึกข้อมูลสัญญาณ Write Enable มีตรรกเป็น “1” พร้อมทั้งทำให้ทรานซิสเตอร์ (Transistor) Q1 และ Q2 ทำงาน ซึ่งทรานซิสเตอร์ทั้งคู่จะสลับกันทำงานเมื่อป้อนกระแสพัลส์ด้านบวกและด้านลบให้แก่ขดลวดตามสัญญาณบันทึก โดยรูปสัญญาณของวงจรวจรพื้นฐานนี้แสดงดังรูปที่ 2.7



รูปที่ 2.7 แสดงรูปสัญญาณการทำงานของวงจรวจรพื้นฐานสำหรับการบันทึกข้อมูลบนบัตรแม่เหล็ก

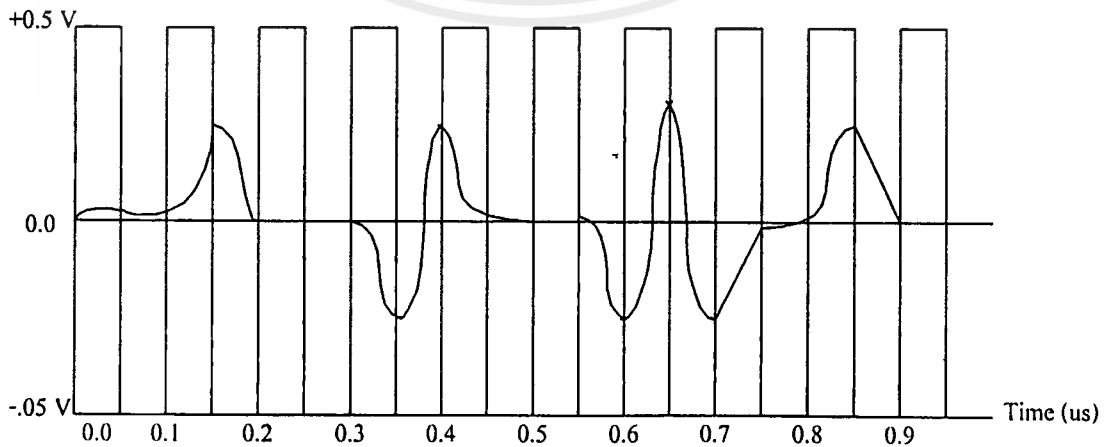
สำหรับค่าของกระแสพัลส์ที่ใช้ในการบันทึกข้อมูลสามารถหาได้จากกราฟแสดงคุณสมบัติการอิมิตัวของแม่เหล็กบนบัตรแม่เหล็ก ดังแสดงในรูปที่ 2.8



รูปที่ 2.8 กราฟแสดงคุณสมบัติการอิมิตัวของแม่เหล็กบนบัตรแม่เหล็กมาตรฐาน

2.4 การอ่านข้อมูลจากบัตรแม่เหล็ก

การอ่านข้อมูลจากบัตรแม่เหล็ก สามารถทำได้โดยให้แถบแม่เหล็กสัมผัสกับหัวอ่าน ซึ่งเคลื่อนที่ด้วยความเร็วคงที่ ดังแสดงในรูปที่ 2.2 พัลส์แม่เหล็กที่เกิดขึ้นจะผ่านจากช่องว่างของแกนหัวอ่าน ไปยังขดลวดที่พันรอบแกนอยู่ ซึ่งการเปลี่ยนแปลงของพัลส์แม่เหล็กตามข้อมูลที่บันทึกจะทำให้เกิดแรงดันไฟฟ้าที่ขดลวดของหัวอ่านตามข้อมูลนั้น ดังแสดงในรูปที่ 2.9



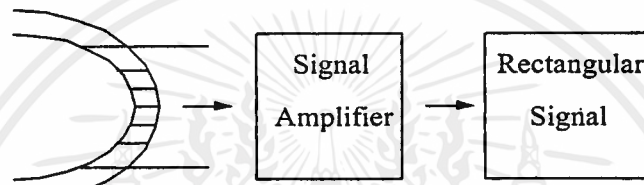
รูปที่ 2.9 แสดงสัญญาณแรงดันไฟฟ้าจากหัวอ่านและถูกขยายสัญญาณประมาณ 100 เท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.9 ตำแหน่งสูงสุดของแรงดันไฟฟ้าที่อ่านได้จะตรงกับตำแหน่งสนามแม่เหล็กบนบัตรแม่เหล็กมีการกลับทิศทาง ทำให้สามารถนำตำแหน่งนี้มาสร้างสัญญาณพัลส์ เพื่ออ่านข้อมูลที่บันทึกอยู่บนหลักแม่เหล็กได้

1.4.1 วงจรพื้นฐานสำหรับสร้างสัญญาณรูปแบบ ISO จากบัตรแม่เหล็ก
ส่วนประกอบของวงจรพื้นฐาน สร้างสัญญาณรูปแบบ ISO จากบัตรแม่เหล็ก แสดงดังรูปที่

2.10



รูปที่ 2.10 แสดงส่วนประกอบของวงจรพื้นฐานสร้างสัญญาณรูปแบบ ISO จากบัตรแม่เหล็ก

1) วงจรขยายสัญญาณแรงดันไฟฟ้า สัญญาณแรงดันไฟฟ้าจากหัวอ่านข้อมูลมีค่าค่อนข้างต่ำ (5-50 มิลลิโวลต์) จึงจำเป็นต้องขยายสัญญาณเพื่อสะดวกต่อการนำไปสร้างสัญญาณรูปสาม สี่ เหลี่ยม โดยการใช้วงจรขยายสัญญาณสองชุดและกำหนดค่ากำลังของสัญญาณรวมมีค่าประมาณ 1000 เท่า (ไม่ให้สัญญาณอิ่มตัว) ดังแสดงไว้ในรูปที่ 2.11

ก) วงจรขยายสัญญาณความแตกต่าง (Differential Amplifier)

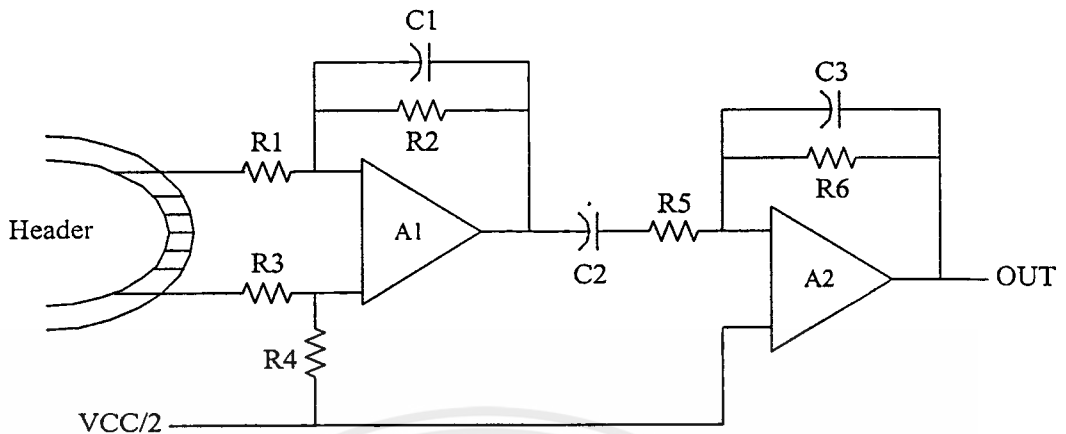
เป็นวงจรขยายสัญญาณชุดแรกที่รับสัญญาณข้อมูลจากหัวทั้งสองของหัวอ่านข้อมูล ซึ่งมีค่าความต้านทานต่ำ และใช้ IC A1 ทำหน้าที่ขยายสัญญาณความแตกต่าง มีค่าอัตราขยาย

$$AV = -R2/R1$$

ข) วงจรขยายสัญญาณแบบกลับเฟส (Inverting Amplifier)

เพื่อให้ได้สัญญาณข้อมูลที่มีค่าแรงดันไฟฟ้าสูงขึ้นพอที่จะป้อนให้กับ วงจรสร้างสัญญาณรูปสี่เหลี่ยมจึงนำสัญญาณข้อมูลมาผ่านวงจรกลับเฟส ซึ่งอัตราขยายแรงดันไฟฟ้ามีค่า

$$AV = -R6/R7$$



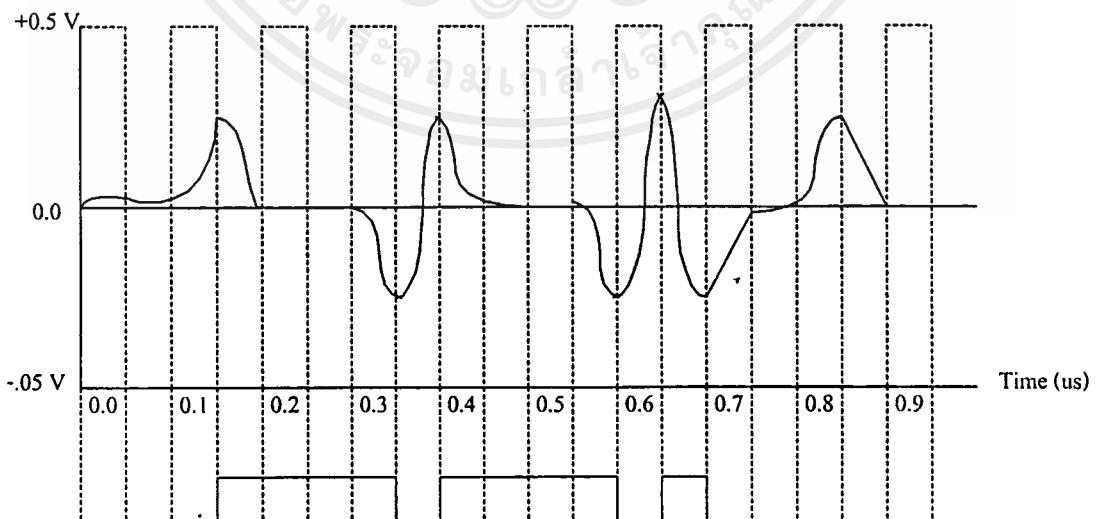
รูปที่ 2.11 แสดงวงจรขยายสัญญาณแรงดันไฟฟ้าสำหรับอ่านข้อมูลบัตรแม่เหล็ก

2) วงจรสร้างสัญญาณรูปสี่เหลี่ยม (รูปแบบ ISO)

จากสัญญาณข้อมูลที่ขยายแล้ว (จากรูปที่ 2.9) ทำให้สามารถพิจารณารูปแบบการสร้างสัญญาณรูปสี่เหลี่ยมได้หลายรูปแบบ โดยเฉพาะการนำตำแหน่งจุดยอดของสัญญาณข้อมูลทั้งด้านบนและด้านล่าง มาใช้ในการสร้างสัญญาณรูปสี่เหลี่ยม จะทำให้ได้สัญญาณพัลส์ที่เหมือนกับสัญญาณที่ใช้การบันทึกข้อมูลบนบัตรแม่เหล็ก การสร้างสัญญาณรูปสี่เหลี่ยม สามารถสร้างได้ 2 วิธี

ก) วงจรเลื่อนเฟสและวงจรเปรียบเทียบแรงดันไฟฟ้า

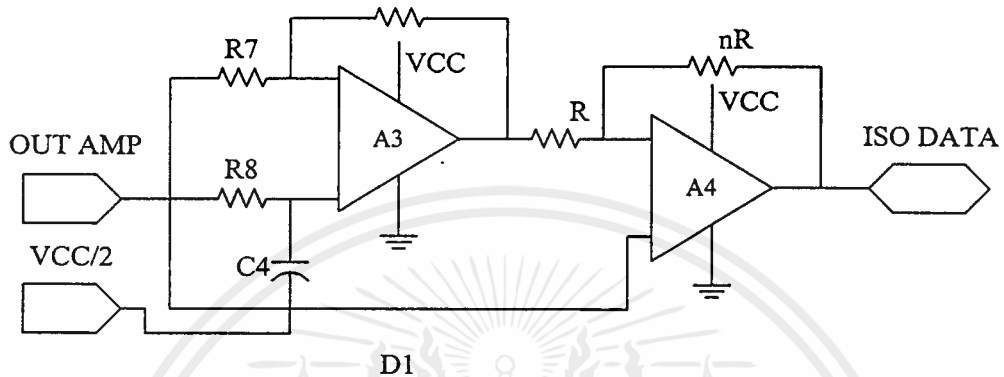
ใช้หลักการหน่วงสัญญาณข้อมูลในช่วงเวลาสั้นมากๆ แล้วนำสัญญาณข้อมูลจริง (เส้นทึบ) เปรียบเทียบกับสัญญาณข้อมูลที่ถูกระงับ (เส้นประ) ซึ่งสัญญาณเอาท์พุทจากการเปรียบเทียบนี้ แสดงไว้ในรูปที่ 2.12



รูปที่ 2.12 แสดงการเปรียบเทียบสัญญาณและลักษณะสัญญาณที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

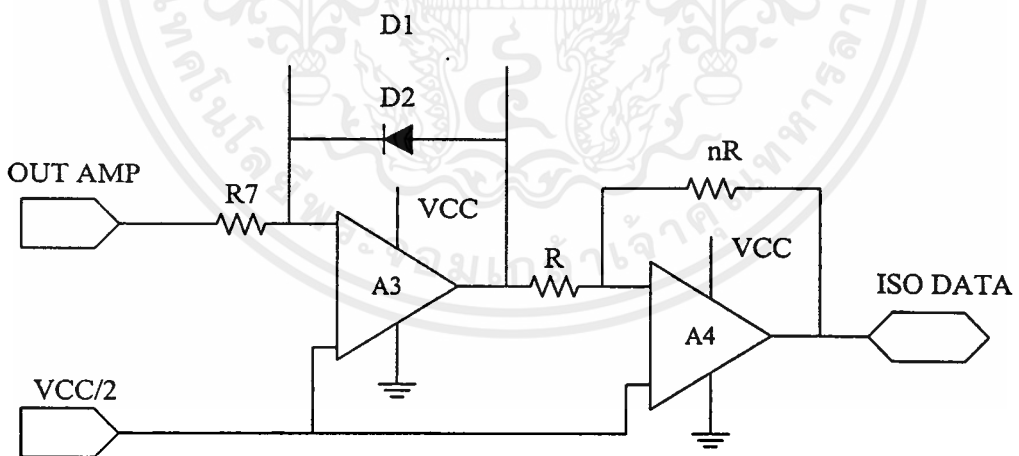
จากวงจรรูปที่ 2.12 วงจรที่ใช้งานจะใช้โอปแอมป์ (Op-Amp) สองตัว โดยที่โอปแอมป์ตัวแรกจัดรูปวงจรเป็นวงจรเลื่อนเฟส เพื่อให้ได้สัญญาณที่ถูกหน่วง (เอาต์พุตเป็นสัญญาณเส้นประ) และโอปแอมป์ตัวที่สองจัดรูปวงจรเป็นวงจรเปรียบเทียบแรงดันระหว่างสัญญาณข้อมูลกับสัญญาณที่ถูกหน่วง ดังแสดงไว้ในรูปที่ 2.13



รูปที่ 2.13 แสดงวงจรเลื่อนเฟสและวงจรเปรียบเทียบแรงดันไฟฟ้า

ข) วงจร Differentiator และวงจรเปรียบเทียบแรงดันไฟฟ้า

แนวทางที่สองจะใช้วงจร Differentiator ซึ่งมีไดโอด D1 และ D2 ประพทิตัวเสมือนตัวต้านทานป้อนกลับด้านลบของวงจร ทำงานร่วมกับวงจรเปรียบเทียบแรงดันไฟฟ้า ดังแสดงในรูปที่ 2.14



รูปที่ 2.14 แสดงวงจร Differentiator และวงจรเปรียบเทียบแรงดันไฟฟ้า

จากรูปที่ 2.14 เมื่อพิจารณาสัญญาณข้อมูลจากหัวอ่านข้อมูล ซึ่งมีลักษณะของสัญญาณคล้ายรูปคลื่นสามเหลี่ยม เมื่อป้อนสัญญาณข้อมูลให้กับวงจร Differentiator สัญญาณเอาต์พุตที่ได้จะเป็นสัดส่วนกับค่าอนุพันธ์ของสัญญาณเอาต์พุต เมื่อสัญญาณข้อมูลเป็นรูปสามเหลี่ยมที่มีค่า

สโลป (Slope) เป็นค่าบวกและค่าลบที่แกว่งบนแรงดันอ้างอิง ($V_{cc}/2$) สัญญาณเอาต์พุตของวงจร Differentiator จะเป็นสัญญาณรูปคลื่นสี่เหลี่ยม มีค่าแรงดันไฟฟ้าเอาต์พุต ตามสมการที่ 2.1

$$V_o = -RCK_n \quad \text{--- (2.1)}$$

เมื่อค่า k_n เป็นค่าสโลปของสัญญาณเอาต์พุต (โวลต์ต่อวินาที) และใช้วงจรเปรียบเทียบสัญญาณแรงดันไฟฟ้าเป็นส่วนสร้างสัญญาณรูปสี่เหลี่ยมให้ดียิ่งขึ้น

สำหรับวงจรเปรียบเทียบแรงดันไฟฟ้าของทั้งสองแนวทาง จะใช้ตัวต้านทาน R และ nR จัดรูปวงจรแบบป้อนกลับทางด้านบวก เพื่อป้องกันให้วงจรเกิดการออสซิลเลท โดยมีค่าแรงดันขีดเริ่มเปลี่ยนด้านสูง (Upper Threshold Voltage : V_{UT}), ค่าแรงดันขีดเริ่มเปลี่ยนด้านต่ำ (Lower Threshold Voltage : V_{LT}) และค่าศักดาไฟฟ้าฮิสเตอร์ซิส (V_T) ตามสมการที่ 2.2, 2.3 และ 2.4 ตามลำดับ

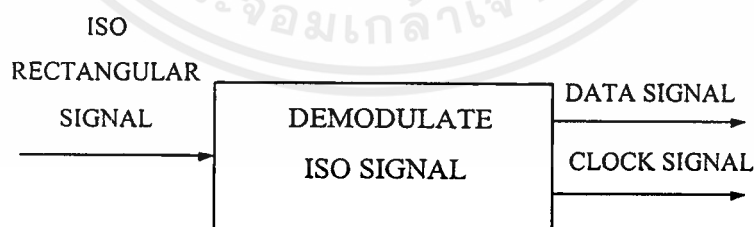
$$V_{UT} = V_{ref} \cdot [1 + (1/n)] - (V_{sat}/n) \quad \text{--- (2.2)}$$

$$V_{LT} = V_{ref} \cdot [1 + (1/n)] - (+V_{sat}/n) \quad \text{--- (2.3)}$$

$$V_T = [(+V_{sat}) - (V_{sat})]/n \quad \text{--- (2.4)}$$

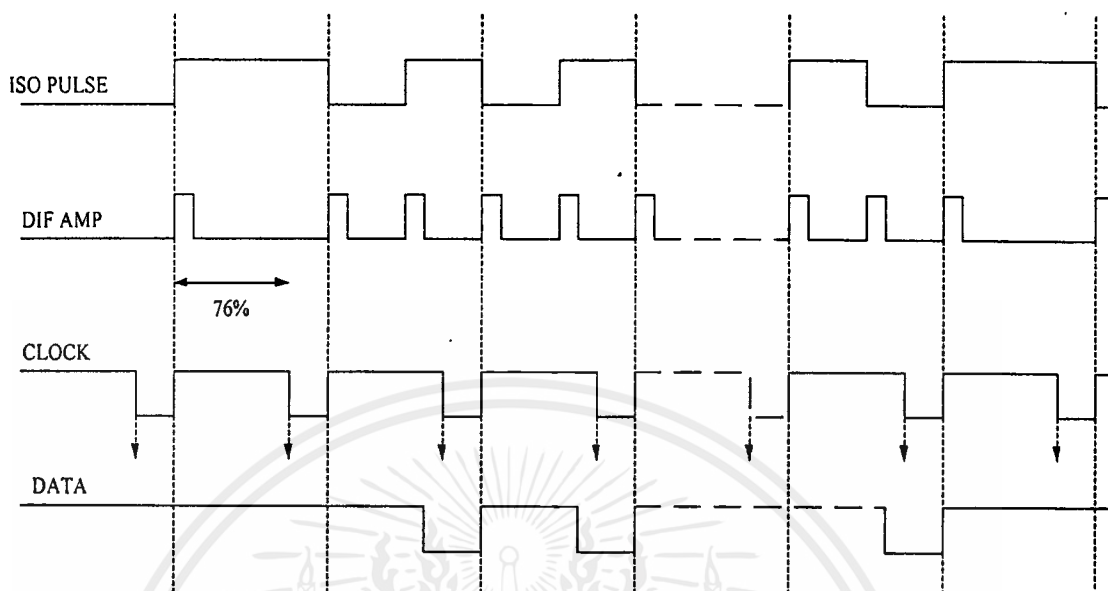
2.4.2 วงจรอ่านข้อมูลบัตรแม่เหล็ก

วงจรพื้นฐานในหัวข้อ 2.4.1 สามารถนำสัญญาณดังกล่าวมาประยุกต์ใช้งานต่อเนื่องได้ เพื่อให้อ่านข้อมูลจากบัตรแม่เหล็กได้สะดวกขึ้น โดยการดิมอดูเลท (Demodulate) สัญญาณรูปสามเหลี่ยม ISO เพื่อให้ได้สัญญาณนาฬิกาสำหรับการอ่านและสัญญาณข้อมูล ดังแสดงรูปที่ 2.15



รูปที่ 2.15 แสดงแผนภาพของวงจรอ่านบัตรแม่เหล็ก

การแยกสัญญาณข้อมูลออกจากสัญญาณนาฬิกา โดยใช้หลักการดิมอดูเลทของสัญญาณ ISO สามารถพิจารณาได้จากรูปคลื่นสัญญาณ ดังแสดงในรูปที่ 2.16



0	0	1	1	0	1	0
SYN	SYN	b1	b2	b3	b4	p

รูปที่ 2.16 แสดงรูปคลื่นสัญญาณของ ISO ที่ถูกมอดูเลท

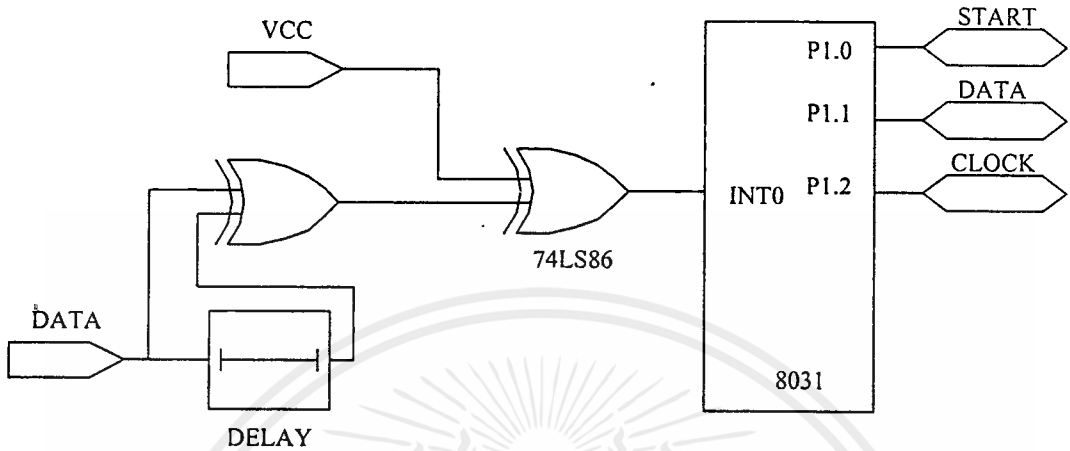
จากรูปที่ 2.16 การแยกสัญญาณข้อมูลออกจากสัญญาณรูปสี่เหลี่ยม ISO ได้นั้น จะต้องพิจารณาการกลับขั้วของสัญญาณรูปสี่เหลี่ยม ISO ในช่วงเวลามาตรฐาน (Standard Time) หรือไม่ ถ้ามีการกลับขั้วของสัญญาณรูปสี่เหลี่ยม ISO แสดงว่าข้อมูลมีค่าทางตรรกเป็น “1” และถ้าไม่มีการกลับขั้วของสัญญาณรูปสี่เหลี่ยม ISO แสดงว่าข้อมูลมีค่าทางตรรกเป็น “0” โดยมีค่าเวลามาตรฐานมีค่าประมาณ 75% ของคาบเวลา 1 บิต ส่วนสัญญาณที่ได้จากการดีมอดูเลทจะมีสัญญาณข้อมูล และสัญญาณนาฬิกาสำหรับการอ่านข้อมูลโดยที่ทุกๆ ของขาลงของสัญญาณนาฬิกา บิตข้อมูลจริงกับตรรกกับสัญญาณข้อมูลที่อ่านได้ สำหรับการสร้างสัญญาณเวลาเพื่อใช้สำหรับแยกสัญญาณข้อมูลออกมานั้นมี 2 วิธี คือแบบความยาวคงที่และแบบความยาวเปลี่ยนแปลงได้

1) แบบความยาวคงที่ สามารถกำหนดเวลามาตรฐานให้มีค่าคงที่และเปรียบเทียบกับทุกบิต วิธีนี้เหมาะกับชุดอ่านบัตรแบบที่ใช้มอเตอร์ดึงผ่านหัวอ่านด้วยความเร็วคงที่

2) แบบความยาวเปลี่ยนแปลงได้ ค่าเวลามาตรฐานมีค่าไม่คงที่ สัญญาณเวลามาตรฐานจะถูกสร้างจากความยาวของบิตก่อนหน้า 1 บิตและสัญญาณเวลามาตรฐาน ถูกลดไปจะสร้างจากความยาวของบิตปัจจุบัน โดยจะเปรียบเทียบกับไปอย่างนี้จนหมดข้อมูล ซึ่งวิธีนี้ใช้กับการอ่านบัตรแบบบัตรที่ใช้การรูดบัตรผ่านหัวอ่านด้วยมือ เนื่องจากความเร็วในการรูดมีค่าไม่แน่นอนและจาก

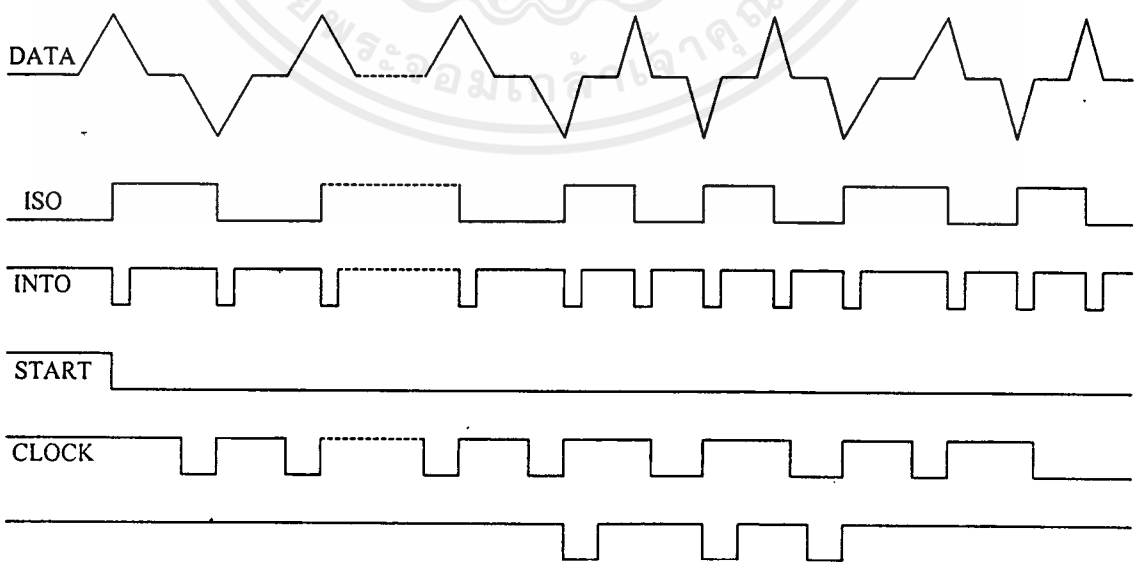
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปคลื่นสัญญาณที่แสดงในรูปที่ 2.16 ได้นำไมโครคอนโทรลเลอร์ 8031 มาประยุกต์ในการสร้างข้อมูลและสัญญาณเวลามาตรฐาน จากสัญญาณรูปสี่เหลี่ยม ISO โดยจัดวงจรแสดงในรูปที่ 2.17



รูปที่ 2.17 แสดงวงจรสร้างสัญญาณข้อมูลและสัญญาณนาฬิกาอ่านข้อมูล โดยใช้ 8031

จากรูปที่ 2.17 หลักการทำงานจะใช้ไอซี 74LS86 เป็นตัวสร้างสัญญาณ Difference pulse แล้วป้อนให้กับ INTO การทำงานในส่วนของ 8031 จะใช้ Time 0 ของ 8031 เป็นตัวจับคาบเวลาแต่ละช่วงเวลาของ Difference pulse ถ้าช่วงเวลามีค่ามากกว่าเวลามาตรฐาน (5% ของช่วงเวลาระหว่าง Difference pulse ชุดก่อน) จะสร้างสัญญาณนาฬิกาทันที แต่จะไปสร้างสัญญาณนาฬิกา ณ ที่เวลา 75% ของช่วงเวลาคู่ก่อน พร้อมกับสร้างสัญญาณข้อมูล "0" โดยใช้พอร์ต P1.0 เป็นขาสัญญาณการเริ่มต้นของการรูดบิตพอร์ต P1.1 เป็นขาสัญญาณข้อมูลและพอร์ต P1.2 เป็นขาสัญญาณนาฬิกาสำหรับการอ่านข้อมูล โดยรูปสัญญาณเอาต์พุตทั้งสามแสดงในรูปที่ 2.18



รูปที่ 2.18 แสดงรูปสัญญาณเอาต์พุตของ 8031 สำหรับสร้างสัญญาณอ่านบิตแม่เหล็ก

2.5 การอินเตอรืรืปต์ (interrupt)

เป็นชื่อเรียกกระบวนการที่เข้ามาขัดจังหวะการทำงาน โดยปกติของไมโครคอนโทรลเลอร์ ในไมโครคอนโทรลเลอร์ MCS-51 สามารถตอบสนองการอินเตอรืรืปต์ที่เกิดขึ้นได้จาก 5 แหล่งกำเนิดสำหรับเบอร์ AT89C51 ประกอบด้วย การรับสัญญาณอินเตอรืรืปต์จากภายนอกผ่านทางขา INT0 และ INT1 ,สัญญาณอินเตอรืรืปต์จากไทมเมอร์/เคาน์เตอร์ T0 , T1 และสัญญาณอินเตอรืรืปต์จากพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ ในขณะที่ไมโครคอนโทรลเลอร์เบอร์ AT89C52 และในอนุกรม AT89Sxx สามารถตอบสนองการอินเตอรืรืปต์ได้จาก 6 แหล่งกำเนิด โดยเพิ่มการรับสัญญาณอินเตอรืรืปต์จากไทมเมอร์/เคาน์เตอร์ 2 อีกหนึ่งแหล่งกำเนิด

การจัดการอินเตอรืรืปต์ในไมโครคอนโทรลเลอร์ MCS-51

เมื่อมีการอินเตอรืรืปต์ในไมโครคอนโทรลเลอร์ MCS-51 เกิดขึ้นและมีการอินาเบิลการตอบสนองการอินเตอรืรืปต์ไว้ กระบวนการหลังจากนั้นซีพียูจะทำการกระโดดไปยังแอดเดรสในหน่วยความจำที่กำหนดไว้ เรียกตำแหน่งแอดเดรสนี้ว่า แอดเดรสอินเตอรืรืปต์เวกเตอร์ (interrupt vector address) ดังนั้นจะต้องมีการเขียนโปรแกรมข้อยการบริการอินเตอรืรืปต์ไว้ที่แอดเดรสอินเตอรืรืปต์เวกเตอร์ โดยค่าของแอดเดรสอินเตอรืรืปต์เวกเตอร์จะแตกต่างกันไปในการอินเตอรืรืปต์แบบต่างๆ ดังมีรายละเอียดดังต่อไปนี้

การอินเตอรืรืปต์ภายนอกที่ขา INT0 มีค่าแอดเดรสอินเตอรืรืปต์เวกเตอร์อยู่ที่ 0003H

การอินเตอรืรืปต์จากไทมเมอร์ 0 มีค่าแอดเดรสอินเตอรืรืปต์เวกเตอร์อยู่ที่ 000BH

การอินเตอรืรืปต์ภายนอกที่ขา INT1 มีค่าแอดเดรสอินเตอรืรืปต์เวกเตอร์อยู่ที่ 0013H

การอินเตอรืรืปต์จากไทมเมอร์ 1 มีค่าแอดเดรสอินเตอรืรืปต์เวกเตอร์อยู่ที่ 0018H

การอินเตอรืรืปต์จากพอร์ตอนุกรม มีค่าแอดเดรสอินเตอรืรืปต์เวกเตอร์อยู่ที่ 0023H

การอินเตอรืรืปต์จากไทมเมอร์ 2 มีค่าแอดเดรสอินเตอรืรืปต์เวกเตอร์อยู่ที่ 002BH

รีจิสเตอร์ที่เกี่ยวข้องกับการอินเตอรืรืปต์ในไมโครคอนโทรลเลอร์ MCS-51

การอินเตอรืรืปต์ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช มีรีจิสเตอร์ที่เกี่ยวข้องอยู่ 2 ตัว ดังมีรายละเอียดดังต่อไปนี้

รีจิสเตอร์อินาเบิลการอินเตอรืรืปต์หรือ IE (Interrupt Enable register)

มีแอดเดรสอยู่ที่ A8H ในพื้นที่ของรีจิสเตอร์ฟังก์ชันพิเศษหรือ SFR มีขนาด 8 บิต สามารถเข้าถึงได้ในระดับบิต ใช้ในการอินาเบิลการตอบสนองการอินเตอรืรืปต์ในแบบต่างๆ มีรายละเอียดการทำงานดังต่อไปนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
EA	-	ET2	ES	ET1	EX1	ET0	EX0

EA (Global enable/disable interrupt) : ใช้ในการอีนาเบิลและดิสเอเบิลการตอบสนองการอินเทอร์รัปต์ทั้งหมด

“0” ดิสเอเบิลการอินเทอร์รัปต์ นั่นคือ กำหนดให้ไมโครคอนโทรลเลอร์ไม่ตอบสนองการอินเทอร์รัปต์ ไม่ว่าจะมาจากแหล่งกำเนิดใดก็ตาม

“1” อีนาเบิลการอินเทอร์รัปต์ นั่นคือ กำหนดให้ไมโครคอนโทรลเลอร์สามารถตอบสนองการอินเทอร์รัปต์จากแหล่งกำเนิดต่างๆ

นั่นคือ ถ้าต้องการให้ไมโครคอนโทรลเลอร์ตอบสนองการอินเทอร์รัปต์ไม่ว่าแหล่งกำเนิดใด จะต้องเซตบิตนี้ก่อนเสมอ สามารถเคลียร์ด้วยระบบการทางซอฟต์แวร์

ET2 (Timer 2 interrupt enable) : ใช้ในการอีนาเบิลการอินเทอร์รัปต์อันเนื่องมาจากโอเวอร์โฟลวหรือการแคปเจอร์ในไทเมอร์/คาน์เตอร์ 2 จะมีเฉพาะในเบอร์ AT89C52 และในอนุกรม AT89Sxx เท่านั้น บิตนี้สามารถเซตและเคลียร์ด้วยระบบการทางซอฟต์แวร์

ES (Serial port interrupt enable bit) : ใช้ในการอีนาเบิลการอินเทอร์รัปต์อันเนื่องมาจากการรับหรือส่งข้อมูลทางพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51 สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

ET1 (Timer 1 interrupt enable) : ใช้ในการอีนาเบิลการอินเทอร์รัปต์อันเนื่องมาจากการโอเวอร์โฟลวในไทเมอร์/คาน์เตอร์ 1 บิตนี้สามารถเซตและเคลียร์ด้วยระบบการทางซอฟต์แวร์

EX1 (External interrupt 1 enable bit) : ใช้ในการอีนาเบิลการอินเทอร์รัปต์อันเนื่องมาจากสัญญาณภายนอกที่ป้อนเข้ามายังขา INT1 บิตนี้สามารถเซตและเคลียร์ด้วยระบบการทางซอฟต์แวร์

ET0 (Timer 0 interrupt enable) : ใช้ในการอีนาเบิลการอินเทอร์รัปต์อันเนื่องมาจากการโอเวอร์โฟลวในไทเมอร์/คาน์เตอร์ 0 บิตนี้สามารถเซตและเคลียร์ด้วยระบบการทางซอฟต์แวร์

EX0 (External interrupt 0 enable bit) : ใช้ในการอีนาเบิลการอินเทอร์รัปต์อันเนื่องมาจากสัญญาณภายนอกที่ป้อนเข้ามายังขา INT0 บิตนี้สามารถเซตและเคลียร์ด้วยระบบการทางซอฟต์แวร์

สำหรับบิต 6 ของรีจิสเตอร์ IE ไม่มีการใช้งาน ต้องกำหนดให้เป็น “0” เสมอ

รีจิสเตอร์จัดลำดับความสำคัญการตอบสนองการอินเทอร์รัปต์หรือIP(Interrupt Priority register)

มีแอดเดรสอยู่ที่ B8H ในพื้นที่ของรีจิสเตอร์ฟังก์ชันพิเศษหรือ SFR มีขนาด 8 บิต สามารถเข้าถึงได้ในระดับบิต ใช้ในการเลือกจัดลำดับความสำคัญในการตอบสนองการอินเทอร์รัปต์ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องการให้ตอบสนองสัญญาณอินเทอร์รัปต์จากแหล่งกำเนิดใดเป็นลำดับก่อนหลัง ถ้าต้องการให้การอินเทอร์รัปต์จากแหล่งกำเนิดใดมีความสำคัญมากที่สุด ให้กำหนดที่บิตนั้นเป็น “1” มีรายละเอียดของรีจิสเตอร์ IP ดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
-	-	PT2	PS	PT1	PX1	PT0	PX0

PT2 (Timer 2 interrupt priority bit) : ใช้ในการกำหนดความสำคัญของการอินเทอร์รัปต์อันเนื่องมาจากการโอเวอร์โฟลวหรือการแคปเจอร์ในไทเมอร์/เคาน์เตอร์ 2 จะมีเฉพาะในเบอร์ AT89C52 และในอนุกรม AT89Sxx เท่านั้น บิตนี้สามารถเซตและเคลียร์ด้วยระบบการทางซอฟต์แวร์

PS (Serial port interrupt priority bit) : ใช้ในการกำหนดความสำคัญของการอินเทอร์รัปต์อันเนื่องมาจากการรับส่งข้อมูลทางพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51 สามารถเซตและเคลียร์ด้วยระบบการทางซอฟต์แวร์

PT1 (Timer 1 interrupt priority bit) : ใช้ในการกำหนดความสำคัญของการอินเทอร์รัปต์อันเนื่องมาจากการโอเวอร์โฟลวหรือการแคปเจอร์ในไทเมอร์/เคาน์เตอร์ 1 บิตนี้สามารถเซตและเคลียร์ด้วยระบบการทางซอฟต์แวร์

PX1 (External interrupt 1 priority bit) : ใช้ในการกำหนดความสำคัญของการอินเทอร์รัปต์อันเนื่องมาสัญญาณจากภายนอกที่ป้อนเข้ามายังขา INT1 บิตนี้สามารถเซตและเคลียร์ด้วยระบบการทางซอฟต์แวร์

PT0 (Timer 0 interrupt priority bit) : ใช้ในการกำหนดความสำคัญของการอินเทอร์รัปต์อันเนื่องมาจากการโอเวอร์โฟลวหรือการแคปเจอร์ในไทเมอร์/เคาน์เตอร์ 0 บิตนี้สามารถเซตและเคลียร์ด้วยระบบการทางซอฟต์แวร์

PX0 (External interrupt 0 priority bit) : ใช้ในการกำหนดความสำคัญของการอินเทอร์รัปต์อันเนื่องมาสัญญาณจากภายนอกที่ป้อนเข้ามายังขา INTO บิตนี้สามารถเซตและเคลียร์ด้วยระบบการทางซอฟต์แวร์

สำหรับบิต 6 และ 7 ของรีจิสเตอร์ IP ไม่มีการใช้งาน ต้องกำหนดให้เป็น “0” เสมอ แหล่งกำเนิดสัญญาณอินเทอร์รัปต์ในไมโครคอนโทรลเลอร์ MCS-51 สัญญาณอินเทอร์รัปต์จากภายนอก

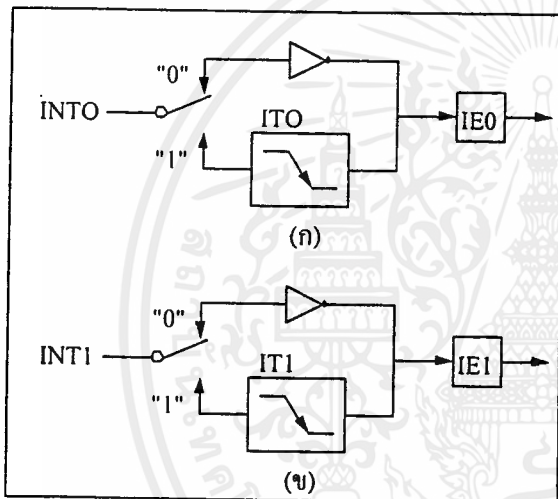
เป็นการตรวจสอบสัญญาณที่ส่งเข้ามายังขา INTO และ INT1 หากตรงตามเงื่อนไขที่กำหนดก็จะทำให้เกิดการอินเทอร์รัปต์ โดยการอินเทอร์รัปต์แบบนี้สามารถกระทำได้โดยการกำหนดค่าในรีจิสเตอร์ IE ที่บิต EX0 สำหรับสัญญาณอินเทอร์รัปต์ที่ขา INTO และ EX1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับสัญญาณอินเทอร์รัปต์ที่ขา INT1 และการทำเงื่อนไขของการตรวจสอบสัญญาณในรีจิสเตอร์ TCON ที่บิต IE0 สำหรับสัญญาณอินเทอร์รัปต์ที่ขา INT0 และ IE1 สำหรับสัญญาณอินเทอร์รัปต์ที่ขา INT1

เงื่อนไขการตรวจสอบสัญญาณอินเทอร์รัปต์ที่ขา INT0 และ INT1 มีด้วยกัน 2 ลักษณะคือ

1. ตรวจสอบระดับลอจิก ถ้าหากบิต IEx ในรีจิสเตอร์ TCON เป็น "0" จะเกิดการอินเทอร์รัปต์จากภายนอกที่ขา INT0 หรือ INT1 ได้ก็ต่อเมื่อตรวจสอบพบระดับลอจิกต่ำหรือ "0" เมื่อเกิดการอินเทอร์รัปต์แล้ว ให้ดำเนินการทำให้สัญญาณที่ขาที่กลับสู่ระดับลอจิก "1" ก่อนที่จะทำการบริการอินเทอร์รัปต์เสร็จสิ้น เพื่อป้องกันการเกิดอินเทอร์รัปต์ซ้ำซ้อน



รูปที่ 2.19 แสดงโครงสร้างของกลไกการเกิดอินเทอร์รัปต์จากภายนอก

(ก) ที่ขา INT0

(ข) ที่ขา INT1

2. ตรวจสอบขอบขาของสัญญาณ ถ้าหากบิต IEx ในรีจิสเตอร์ TCON เป็น "1" จะเกิดการอินเทอร์รัปต์จากภายนอกที่ขา INT0 และ INT1 ได้ก็ต่อเมื่อตรวจพบการเปลี่ยนแปลง

ของสัญญาณที่ขา INT0 หรือ INT1 จาก "1" เป็น "0" หรือตรวจสอบพบขอบขาลงของสัญญาณที่ป้อนมายังขา INT0 หรือ INT1 และต้องมีการรักษาสถานะลอจิก "0" นี้เป็นเวลาอย่างน้อย 1 แมกซีนไซเคิล จึงถือว่า เกิดการอินเทอร์รัปต์อย่างสมบูรณ์

เมื่อเกิดการอินเทอร์รัปต์ขึ้นซีพียูภายในไมโครคอนโทรลเลอร์ จะกระโดดไปยังแอดเดรส 0003H สำหรับการอินเทอร์รัปต์ที่ขา INT0 และ 0013H สำหรับการอินเทอร์รัปต์ที่ขา INT1

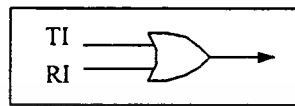
ในรูปที่ 2.19 เป็นไดอะแกรมแสดงการทำงานของอินเทอร์รัปต์ในลักษณะนี้ โดยรูปที่ 2.19 (ก)เป็นการอินเทอร์รัปต์ที่ขา INT0 ส่วนในรูปที่ 2.19 (ข) เป็นการอินเทอร์รัปต์ที่ขา INT1 การอินเทอร์รัปต์จากไทเมอร์/คาน์เตอร์ 0 และ 1

แหล่งกำเนิดอินเทอร์รัปต์นี้จัดเป็นแหล่งกำเนิดอินเทอร์รัปต์ภายในแบบหนึ่ง โดยใช้การเกิดโอเวอร์โพลจากการนับค่าในไทเมอร์/คาน์เตอร์ภายในไมโครคอนโทรลเลอร์ MCS-51 เมื่อไทเมอร์ 0 เกิดการโอเวอร์โพลก็จะทำการเซตบิต TFO ในรีจิสเตอร์ TCON และถ้าไทเมอร์ 1 เกิดการโอเวอร์โพลบิต TF1 ในรีจิสเตอร์ TCON จะได้รับการเซตเช่นเดียวกัน

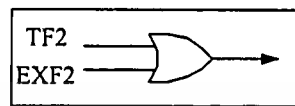
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าแอดเดรสอินเตอร์รัปต์เวกเตอร์ของการอินเตอร์รัปต์แบบนี้อยู่ที่ 000BH สำหรับไทมเมอร์ 0 และ 001BH สำหรับไทมเมอร์ 1

อย่างไรก็ตามการอินเตอร์รัปต์แบบนี้จะเกิดขึ้นหรือมีการตอบสนองก็ต่อเมื่อมีการอินาเบิลการอินเตอร์รัปต์ โดยการเซตบิต EA, ET0 และ ET1 ในรีจิสเตอร์ IE



รูปที่ 2.20



รูปที่ 2.21

รูปที่ 2.20 แสดงเงื่อนไขในการเกิดอินเตอร์รัปต์จากพอร์ตอนุกรมด้วยการ OR กันของบิต RI และ TI
รูปที่ 2.21 แสดงเงื่อนไขในการเกิดอินเตอร์รัปต์ของไทมเมอร์/เคาน์เตอร์ 2 ด้วยการ OR กันของบิต TF2 และ EXF2

การอินเตอร์รัปต์จากพอร์ตอนุกรม

แหล่งกำเนิดอินเตอร์รัปต์จัดเป็นแหล่งกำเนิดอินเตอร์รัปต์ภายในแบบหนึ่งเมื่อวงจรพอร์ตอนุกรมสูงหรือรับส่งข้อมูลเสร็จสมบูรณ์ก็จะกำเนิดสัญญาณอินเตอร์รัปต์ขึ้น โดยการเซตบิต RI ในกรณีรับข้อมูล และบิต TI ในกรณีส่งข้อมูล บิต RI และ TI อยู่ในรีจิสเตอร์ SCON

ค่าแอดเดรสอินเตอร์รัปต์เวกเตอร์ของการอินเตอร์รัปต์แบบนี้อยู่ที่ 0023H การอินเตอร์รัปต์ในแบบนี้สามารถแทนได้ด้วยการ OR กันของบิต RI และ TI ดังแสดงในรูปที่ 2.20

อย่างไรก็ตามการอินเตอร์รัปต์แบบนี้จะเกิดขึ้นหรือมีการตอบสนองก็ต่อเมื่อมีการอินาเบิลการอินเตอร์รัปต์ โดยการเซตบิต EA และ ES ในรีจิสเตอร์ IE หลังการตอบสนองการอินเตอร์รัปต์ต้องทำการเคลียร์บิต RI และ TI เสมอ

การอินเตอร์รัปต์จากไทมเมอร์/เคาน์เตอร์ 2

แหล่งกำเนิดอินเตอร์รัปต์นี้จัดเห็นแหล่งกำเนิดอินเตอร์รัปต์ภายในแบบหนึ่งโดยใช้การเกิดโอเวอร์โฟลวจากการนับค่าในไทมเมอร์/เคาน์เตอร์ 2 หรือการแคปเจอร์ หรือการตรวจจับสัญญาณที่ขา T2EX ภายในไมโครคอนโทรลเลอร์ MCS-51 เบอร์ AT89C52 และในอนุกรม AT89Sxx เมื่อไทมเมอร์ 2 เกิดการโอเวอร์โฟลวก็จะทำการเซตบิต TF2 ในรีจิสเตอร์ TCON และถ้าไทมเมอร์ 2 สามารถตรวจจับการเปลี่ยนแปลงจากระดับ “1” เป็น “0” ที่ขา T2EX ได้และบิต EXEN2 ในรีจิสเตอร์ T2CON ได้รับการเซตไว้ ก็จะทำให้บิต EXF2 ในรีจิสเตอร์ T2CON เซต เป็นการแจ้งว่าเกิดการอินเตอร์รัปต์เนื่องจากการแคปเจอร์ที่ไทมเมอร์ 2 ค่าแอดเดรสอินเตอร์รัปต์เวกเตอร์ของการอินเตอร์รัปต์แบบนี้อยู่ที่ 002BH การอินเตอร์รัปต์แบบนี้สามารถแทนได้ด้วยการ OR กันของบิต TF2 และ EXF2 ดังแสดงในรูปที่ 2.21

อย่างไรก็ตามการอินเทอร์รัปต์แบบนี้จะเกิดขึ้นหรือมีการตอบสนองก็ต่อเมื่อมีการอินาเบิลการอินเทอร์รัปต์ โดยการเซตบิต EA ในรีจิสเตอร์ IE จะตอบสนองในกรณีเกิดโอเวอร์โฟลวเมื่อบิต EXEN2 ในรีจิสเตอร์ T2CON เป็น “0” และตอบสนองในกรณีเกิดแคปเจอร์เมื่อบิต EXEN2 ในรีจิสเตอร์ T2CON เป็น “1”

ลำดับความสำคัญของการอินเทอร์รัปต์ในไมโครคอนโทรลเลอร์ MCS-51

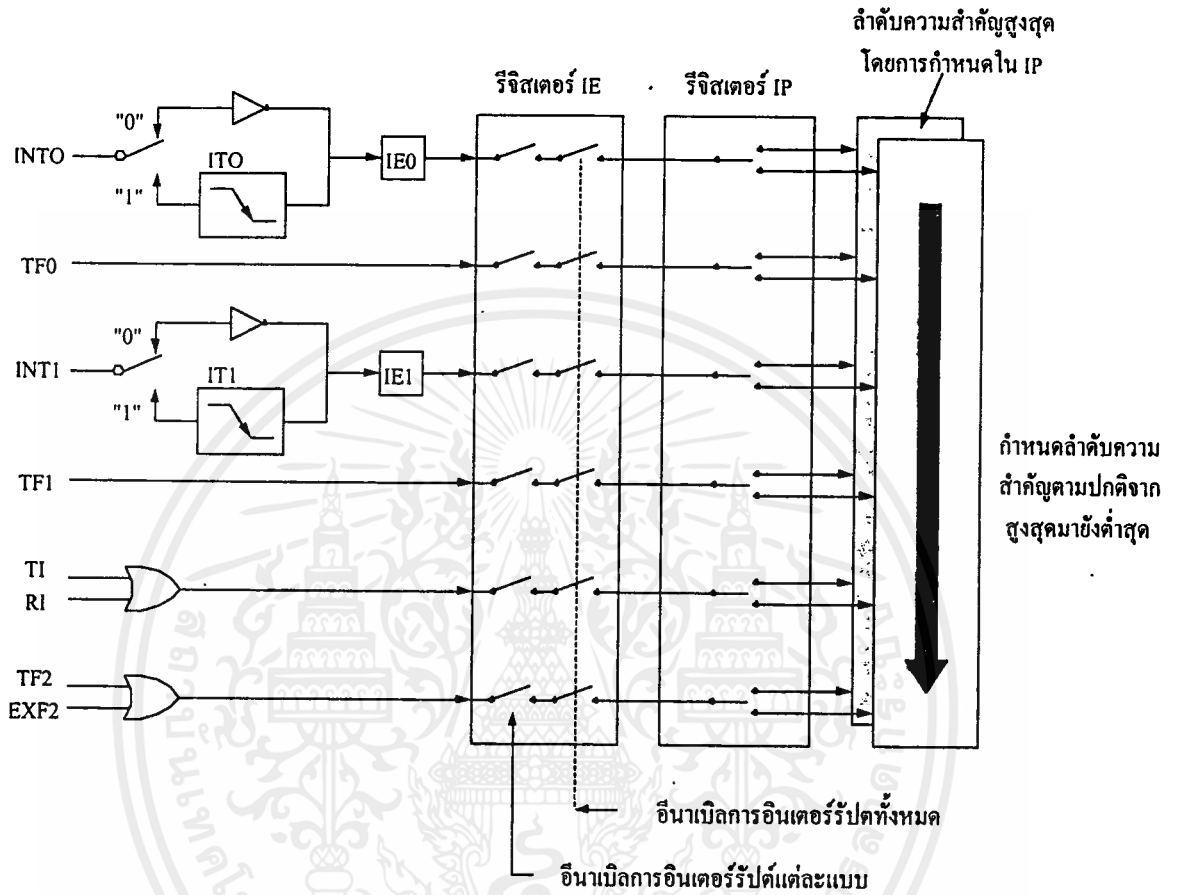
การกำเนิดสัญญาณอินเทอร์รัปต์ในไมโครคอนโทรลเลอร์ MCS-51 มีได้จาก 5-6 แหล่ง ดังนั้นจึงต้องมีการจัดลำดับความสำคัญ ในกรณีที่เกิดการอินเทอร์รัปต์ขึ้นพร้อมๆ กันจากหลายแหล่งกำเนิด โดยสามารถกำหนดความสำคัญได้ที่รีจิสเตอร์ IP

อย่างไรก็ตาม ลำดับความสำคัญของการอินเทอร์รัปต์โดยในปกติหรือในกรณีกำหนดข้อมูลในรีจิสเตอร์ IP ให้เป็น “1” ทุกบิต (ยกเว้นบิต 6 และ 7) จะเรียงลำดับความสำคัญสูงสุดไปจนถึงต่ำสุดดังนี้

1. อินเทอร์รัปต์ภายนอกที่ขา INTO หรือการเซตของบิต IE0
2. อินเทอร์รัปต์จากไทเมอร์ 0 หรือการเซตของบิต TF0
3. อินเทอร์รัปต์ภายนอกที่ขา INT หรือการเซตของบิต IE1
4. อินเทอร์รัปต์จากไทเมอร์ 1 หรือการเซตของบิต TF1
5. อินเทอร์รัปต์จากพอร์ตอนุกรม หรือการเซตบิต RI หรือ TI
6. อินเทอร์รัปต์จากไทเมอร์ 2 หรือการเซตของบิต TF2 หรือ EXF2

โครงสร้างระบบการอินเทอร์รัปต์ของไมโครคอนโทรลเลอร์ MCS-51

จากรายละเอียดทั้งหมดของการอินเทอร์รัปต์ในไมโครคอนโทรลเลอร์ MCS-51 สามารถแสดงเป็นโครงสร้างของระบบอินเทอร์รัปต์โดยรวมดังแสดงในรูปที่ 2.22 โดยรีจิสเตอร์ IP แทนด้วยสวิทช์สองทางเพื่อกำหนดนัยสำคัญของการตอบสนองการอินเทอร์รัปต์ ในขณะที่รีจิสเตอร์ IE และบิตอินาเบิลการอินเทอร์รัปต์แบบต่างๆ จะแทนด้วยสวิทช์ต่ออนุกรมกัน นั่นคือถ้าไม่ต่อวงจรทั้งสองตัวก็จะไม่มีการอินเทอร์รัปต์แบบนั้นๆ เกิดขึ้น นั่นหมายความว่า การอินเทอร์รัปต์จะเกิดขึ้นได้ก็ ต่อเมื่อเงื่อนไขในการเกิดเหตุการณ์เป็นจริง มีการอินาเบิลการอินเทอร์รัปต์ไว้อย่างถูกต้อง และกำหนดระดับความสำคัญไว้เหมาะสม



รูปที่ 2.22 โครงสร้างของระบบอินเตอร์ร็ปต์ภายในไมโครคอนโทรลเลอร์ MCS-51

2.6 พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51

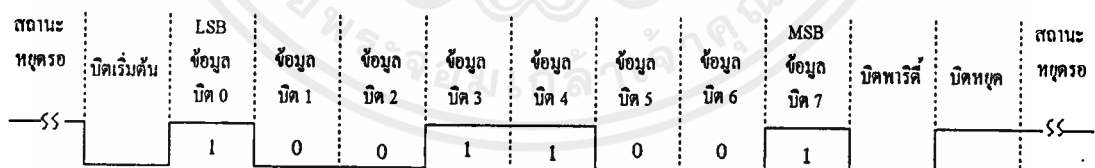
ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีวงจรสื่อสารอนุกรมแบบ Full Duplex 1 ชุด(วงจรสื่อสารแบบ Full Duplex หมายถึง วงจรสื่อสารที่สามารถทำการรับและส่งข้อมูลในลักษณะ 2 ทิศทางได้ในเวลาเดียวกัน) โดยใช้ขั้วสัญญาณของพอร์ต 3 คือขา P3.0 เป็นขารับข้อมูลเข้าหรือ RxD และขา P3.1 เป็นขาส่งข้อมูลออกหรือ TxD โดยวงจรสื่อสารข้อมูลแบบอนุกรมของไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบบแฟลชเป็นแบบอะซิงโครนัส ปกติแล้วพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 จะใช้ในการติดต่อสื่อสารกับพอร์ตอนุกรมของคอมพิวเตอร์ โดยใช้มาตรฐาน RS-232 แต่ในปัจจุบันสามารถติดต่อกันในมาตรฐาน RS-422 หรือ RS-485 ได้แล้ว โดยใช้ไอซีพิเศษที่ทำหน้าที่ในการแปลงสัญญาณการสื่อสารดังกล่าว

การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัสคือ การรับและส่งข้อมูลโดยไม่จำเป็นต้องมีสัญญาณนาฬิกาพร้อมด้วย แต่จะใช้การกำหนดค่าอัตราเร็วในการรับและส่งข้อมูลให้มีค่าเท่ากัน ซึ่งเรียกอัตราเร็วนี้ว่า อัตราบอดหรือบอดเรต (baud rate) มีหน่วยเป็น บิตต่อวินาที (bit per second : bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกันคือ

1. บิตเริ่มต้น (start bit) มีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรม มีขนาด 8 บิต
3. บิตตรวจสอบพาริตี (parity bit) มีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้ายหรือบิตหยุด (stop bit) มีขนาด 1 บิต



รูปที่ 2.23 แบบข้อมูลอนุกรมแบบอะซิงโครนัส

รูปที่ 2.23 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัสเมื่อไม่มีการส่งข้อมูลขา DATA จะมีสถานะลอจิก “1” เรียกสถานะนี้ว่าสถานะหยุดรอ (waiting stage) การเริ่มส่งข้อมูลจะเริ่มจากการให้ขา DATA มีลอจิก “0” ด้วยช่วงระยะเวลา 1 บิต เรียกบิตนี้ว่าบิตเริ่มต้น (start bit) จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุดหรือบิต LSB ก่อน ซึ่งข้อมูลที่ต้องการส่งมีจำนวน 8 บิต จากนั้นตามด้วยบิตพาริตี (parity bit) ซึ่งใช้ในการตรวจสอบความผิดพลาดที่เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการส่งข้อมูล บิตสุดท้ายที่จะส่งคือ บิตปิดท้ายหรือบิตหยุด (stop bit) โดยจะเป็นการทำให้ขา DATA มีสถานะลอจิก “1” อีกครั้งด้วยเวลาอย่างน้อย 1 บิต, 1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัส หรือ อัตราบอด หรือบอดเรตที่ใช้สำหรับพอร์ตอนุกรม RS-232 มีด้วยกันหลายค่าได้แก่ 110, 150, 300, 600, 1,200, 2,400, 4,800, 9,600 และ 19,200 บิตต่อวินาที โดยมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์เนื่องจากอัตราบอดคือค่าของจำนวนบิตที่สามารถส่งได้ใน 1 วินาที สมมติว่าข้อมูลอนุกรมมีขนาด 8 บิต ไม่มีการตรวจสอบพาริตี มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูล 1 ไบต์จะมีความยาวเท่ากับ 10 บิต ถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9,600 บิตต่อวินาที ก็สามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที

การตรวจสอบพาริตีสามารถกำหนดให้เป็นแบบคี่ (odd), แบบคู่ (even) หรือไม่มีการตรวจสอบพาริตีก็ได้ พาริตีคี่หรือพาริตีคู่แสดงถึงจำนวนลอจิก “1” ทั้งหมดภายในข้อมูลที่ส่งไป 1 ไบต์ รวมบิตพาริตีว่ามีจำนวนเป็นเลขคู่หรือเลขคี่ ยกตัวอย่าง ข้อมูลที่จะทำการส่งมีขนาด 8 บิต มีค่าเท่ากับ 99H หรือ 10011001B จะเห็นว่าข้อมูลในไบต์นี้มีจำนวนลอจิก “1” จำนวน 4 ตัวที่เป็นเลขคู่ ดังนั้นถ้ากำหนดค่าพาริตีเป็นคู่ ค่าของพาริตีจะต้องมีลอจิกเป็น “0” แต่ถ้ากำหนดพาริตีเป็นคี่ ค่าของบิตพาริตีจะต้องเป็น “1” เพื่อให้ข้อมูล 1 ไบต์รวมทั้งบิตพาริตีเป็นคี่

บิตพาริตีถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART (Universal Asynchronous Receiver Transmitter : เป็นอุปกรณ์ที่ใช้ในการรับและส่งข้อมูลอนุกรม) ซึ่งทางภาครับจะต้องกำหนดคุณสมบัติการตรวจสอบพาริตีที่ตรงกันเอาไว้ว่าจะตรวจสอบพาริตีคี่หรือพาริตีคู่ จากนั้นภาครับของ UART จะทำการตรวจสอบค่าพาริตีที่เกิดขึ้นว่าเป็นคู่หรือเป็นคี่ โดยการนับจำนวนลอจิก “1” ทั้งหมดรวมทั้งบิตพาริตีด้วย ถ้ากำหนดค่าพาริตีไว้เป็นคู่แต่อ่านค่าตัวเลขในการนับออกมาได้ตัวเลขเป็นคี่ ทางภาครับจะแสดงข้อผิดพลาดออกมาให้ผู้ใช้งานทราบ กระบวนการดังกล่าวเป็นวิธีการตรวจสอบความผิดพลาดที่เกิดขึ้นในการรับส่งข้อมูลที่ง่ายที่สุด แต่มันสามารถตรวจสอบได้เมื่อมีบิตข้อมูลที่ทำการรับส่งข้อมูลที่ผิดพลาดเพียงบิตเดียวเท่านั้น ถ้าข้อมูลที่ทำการส่งมีบิตที่ผิดพลาดมากกว่า 1 บิต การตรวจสอบด้วยวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตีบิตเป็น NONE นั้นทั้งภาครับและภาคส่ง จะไม่มีการตรวจสอบพาริตี

รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานของพอร์ตอนุกรมใน MCS-51

ในการทำงานของวงจรพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 มีรีจิสเตอร์ที่ต้องเกี่ยวข้องอยู่ 2 ตัว ดังมีรายละเอียดดังนี้

รีจิสเตอร์บัฟเฟอร์ของพอร์ตอนุกรมหรือ SBUF (Serial data buffer register)

มีแอดเดรสอยู่ที่ 99H ในพื้นที่ของรีจิสเตอร์ฟังก์ชันพิเศษหรือ SFR มีขนาด 8 บิต มีการแบ่งเป็น 2 ส่วนคือ รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล (transmit buffer register) และรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูล (receive buffer register) เมื่อมีการเขียนข้อมูลมายังรีจิสเตอร์ SBUF ข้อมูลนั้นจะถูกส่งต่อไปยังบัฟเฟอร์สำหรับส่งข้อมูล เพื่อส่งออกจากไมโครคอนโทรลเลอร์ผ่านทางขา TxD หรือขา P3.1 ในกรณีที่มีการอ่านข้อมูลจากรีจิสเตอร์ SBUF ข้อมูลจะถูกส่งผ่านไปยังรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูลเพื่อส่งต่อไปยังไมโครคอนโทรลเลอร์ต่อไป สำหรับการรับข้อมูลอนุกรมจากภายนอกนั้นจะผ่านมาทางขา RxD หรือ P3.0 ของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช รีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรมหรือ SCON (Serial port Control Register)

เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ 98H ในพื้นที่ของรีจิสเตอร์ SFR สามารถเข้าถึงได้ในระดับบิต มีรายละเอียดการทำงานดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0-SM1 (serial port mode bit 0-1) : ใช้ในการเลือกโหมดการทำงานของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51 ดังมีรายละเอียดต่อไปนี้

SM2 : ใช้ในการอินทิเกรตการสื่อสารในแบบมัลติโพรเซสเซอร์ (multiprocessor) ในการทำงานของโหมด 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 ในโหมด 2 และ 3 ถ้าบิตนี้เป็น "1" บิต RI จะไม่แอกติฟถ้าบิตที่ 9 ที่รับเข้ามาเป็น "0" (ข้อมูลบิตที่ 9 เก็บไว้ที่บิต RB8) ในการทำงานโหมด 1 ถ้าบิตนี้เซต บิต RI จะไม่แอกติฟถ้ายังไม่ได้รับบิตหยุด ส่วนในโหมด 0 บิตนี้ไม่มีการใช้งาน

REN (Enable serial reception) : ใช้ในการอินทิเกรตการรับข้อมูลของพอร์ตอนุกรม ทำการเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์ ถ้าต้องการให้มีการรับข้อมูลต้องเซตบิตนี้ให้เป็น "1"

TB8 : ใช้สำหรับเก็บข้อมูลบิตที่ 9 ที่ต้องการส่งออกไปในการทำงานโหมด 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 เซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

RB2 : ใช้สำหรับเก็บข้อมูลบิตที่ 9 ที่เข้ามาในการทำงานโหมด 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 แต่ถ้าหากพอร์ตอนุกรมทำงานอยู่ในโหมด 1 และบิต SM2 เป็น "0" ข้อมูลที่บิต RB8 คือข้อมูลของบิตหยุด (STOP bit) สำหรับในการทำงานโหมด 0 บิตนี้จะไม่ใช้งาน บิต RB8 นี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

TI (Transmit Interrupt flag) : ใช้ในการแสดงการเกิดอินเทอร์รัปต์เมื่อมีการส่งข้อมูลออกทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 สามารถเซตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อทำการส่งข้อมูลบิตที่ 8 ไปเรียบร้อยแล้วในการทำงานโหมด 0 ส่วนในการทำงานของโหมดอื่น บิตนี้จะเซตเมื่อมีการเริ่มส่งบิตชุดออกไป การเคลียร์บิตนี้ต้องใช้กระบวนการทางซอฟต์แวร์

RI (Receive Interrupt flag) : ใช้ในการแสดงการเกิดอินเทอร์รัปต์เมื่อมีการรับข้อมูลเข้าสู่พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 สามารถเซตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อทำการส่งข้อมูลบิตที่ 8 ไปเรียบร้อยแล้วในการทำงานโหมด 0 ส่วนในการทำงานของโหมดอื่น บิตนี้จะเซตเมื่อสามารถรับบิตชุดของข้อมูลอนุกรมไปได้ครึ่งทางแล้ว ยกเว้นในกรณีที่บิต SM2 มีการเซต บิตนี้จะเซตได้ก็ต่อเมื่อมีการรับบิตชุดหรือบิตที่ 9 เกิดขึ้นอย่างสมบูรณ์แล้ว การเคลียร์บิตนี้ต้องใช้กระบวนการทางซอฟต์แวร์เท่านั้น

โหมดการทำงานของใน พอร์ตอนุกรมMCS-51

พอร์ตอนุกรมไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชสามารถเลือกโหมดการทำงานได้ถึง 4 โหมดคือ

1. โหมด 0 เป็นการกำหนดให้พอร์ตอนุกรมทำงานในลักษณะชิฟต์รีจิสเตอร์
2. โหมด 1 เป็นการกำหนดให้เป็น UART ขนาด 8 บิต สามารถเลือกอัตราบอดได้
3. โหมด 2 เป็นการกำหนดให้เป็น UART ขนาด 9 บิต โดยมีอัตราบอดคงที่
4. โหมด 3 เป็นการกำหนดให้เป็น UART ขนาด 9 บิต สามารถเลือกอัตราบอดได้

การเลือกโหมดการทำงานของวงจรพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51

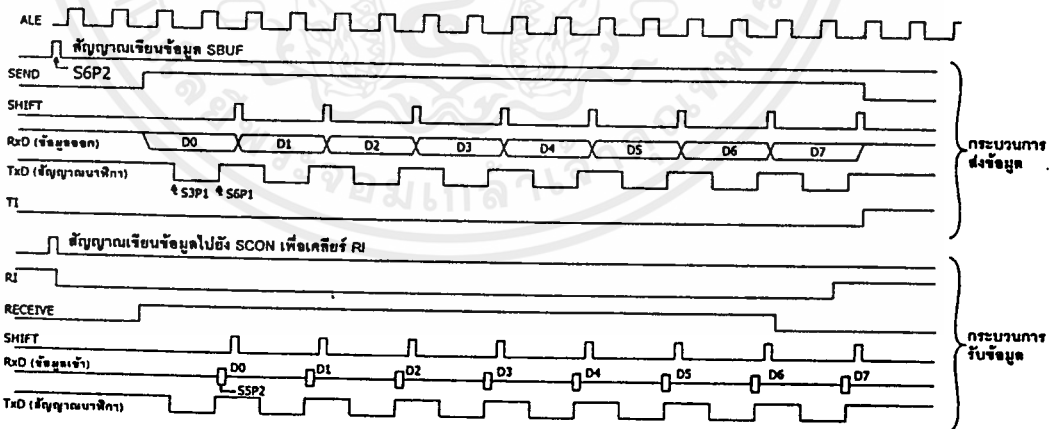
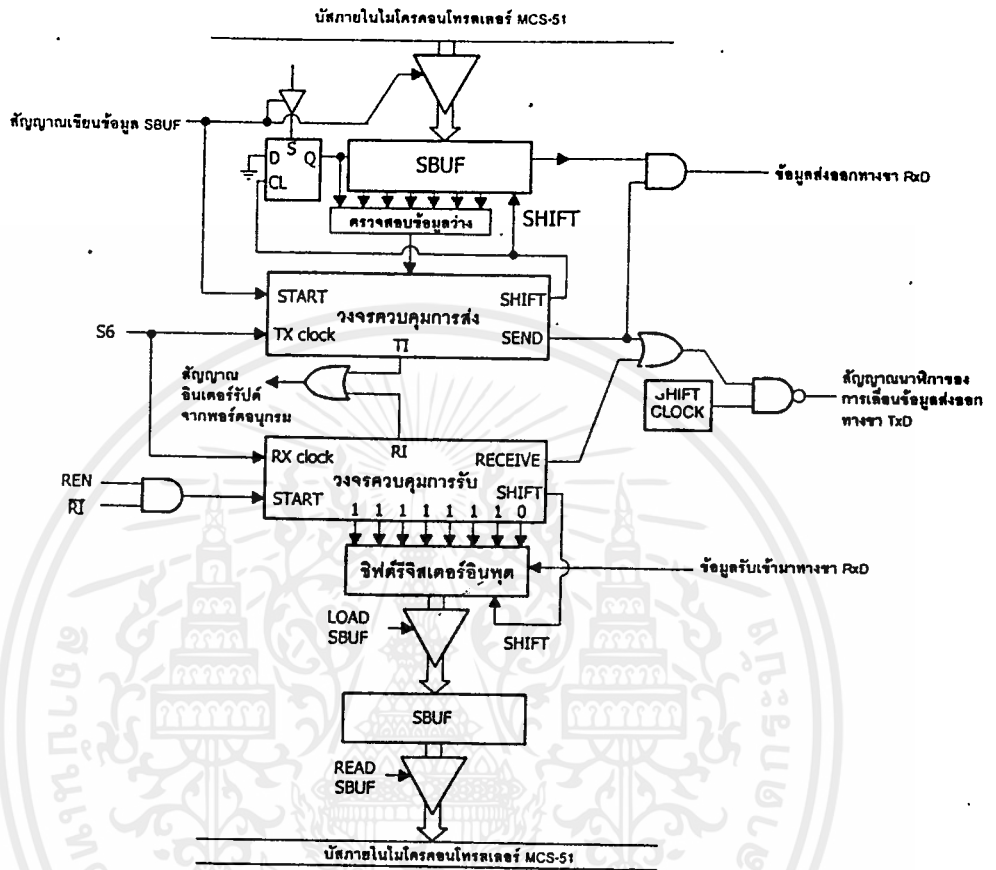
กระทำได้โดยการกำหนดข้อมูลให้แก่บิต SM0 และ SM1 ในรีจิสเตอร์ SCON

การทำงานในโหมด 0 ของวงจรพอร์ตอนุกรม

มีไคอะแกรมการทำงานและไคอะแกรมเวลาแสดงในรูปที่ 2.24 ข้อมูลอนุกรมจะผ่านเข้าและออกทางขา RxD ส่วนขา TxD ทำหน้าที่เป็นสัญญาณนาฬิกาของการเลื่อนข้อมูล (shift clock) ในโหมดนี้มีจำนวนข้อมูล 8 บิต โดยทำการรับและส่งข้อมูลในบิต LSB ก่อน อัตราในการรับส่งข้อมูลหรืออัตราบอดถูกกำหนดไว้ครั้งที่ 1/12 ของความถี่ของสัญญาณนาฬิกา

เริ่มต้นการส่งข้อมูลด้วยการเขียนข้อมูลที่ต้องการส่งมายังรีจิสเตอร์ SBUF สัญญาณเขียนข้อมูล SBUF แยกที่พินเป็น "1" ที่สแตต 6 เฟส 2 (S6P2) ของเมซินไซเกิล ส่งมายังวงจรควบคุมการส่ง (TX control) ทำให่วงจรควบคุมการส่งเริ่มต้นการส่งข้อมูล สัญญาณ SEND จะแยกที่พินเป็น "1" ตลอดกระบวนการส่งข้อมูล

ข้อมูลจากรีจิสเตอร์ SBUF จะถูกเลื่อนออกที่ขา P3.0 หรือขา RxD ครั้งละบิต ตามจังหวะของสัญญาณนาฬิกาที่ส่งออกมาทางขา P3.1 หรือขา TxD โดยสัญญาณนาฬิกาของการเลื่อนข้อมูล



รูปที่ 2.24 ไตอะแกรมการทำงานในโหมด 0 ของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS - 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะมีขอบขาลงของสัญญาณที่สเตต 3 เฟส และมีขอบขาขึ้นของสัญญาณที่สเตต 6 เฟส 1 ของแต่ละแมชชีนไจเกิดในกระบวนการส่งข้อมูล จนกระทั่งเมื่อส่งข้อมูลครบ 8 บิตแล้ว บิต TI ในรีจิสเตอร์ SCON จะเกิดการเซตเป็นการแจ้ง ให้ทราบว่าส่งข้อมูลครบแล้ว หากการอินเตอร์รัปต์จากพอร์ตอนุกรมได้รับการอินาเบิลไว้ ก็จะมีการอินเตอร์รัปต์ขึ้นในระบบ เมื่อเสร็จสิ้นกระบวนการรับข้อมูล สัญญาณ SEND จะกลายเป็น “0” จนกว่าจะเริ่มกระบวนการรับข้อมูลใหม่

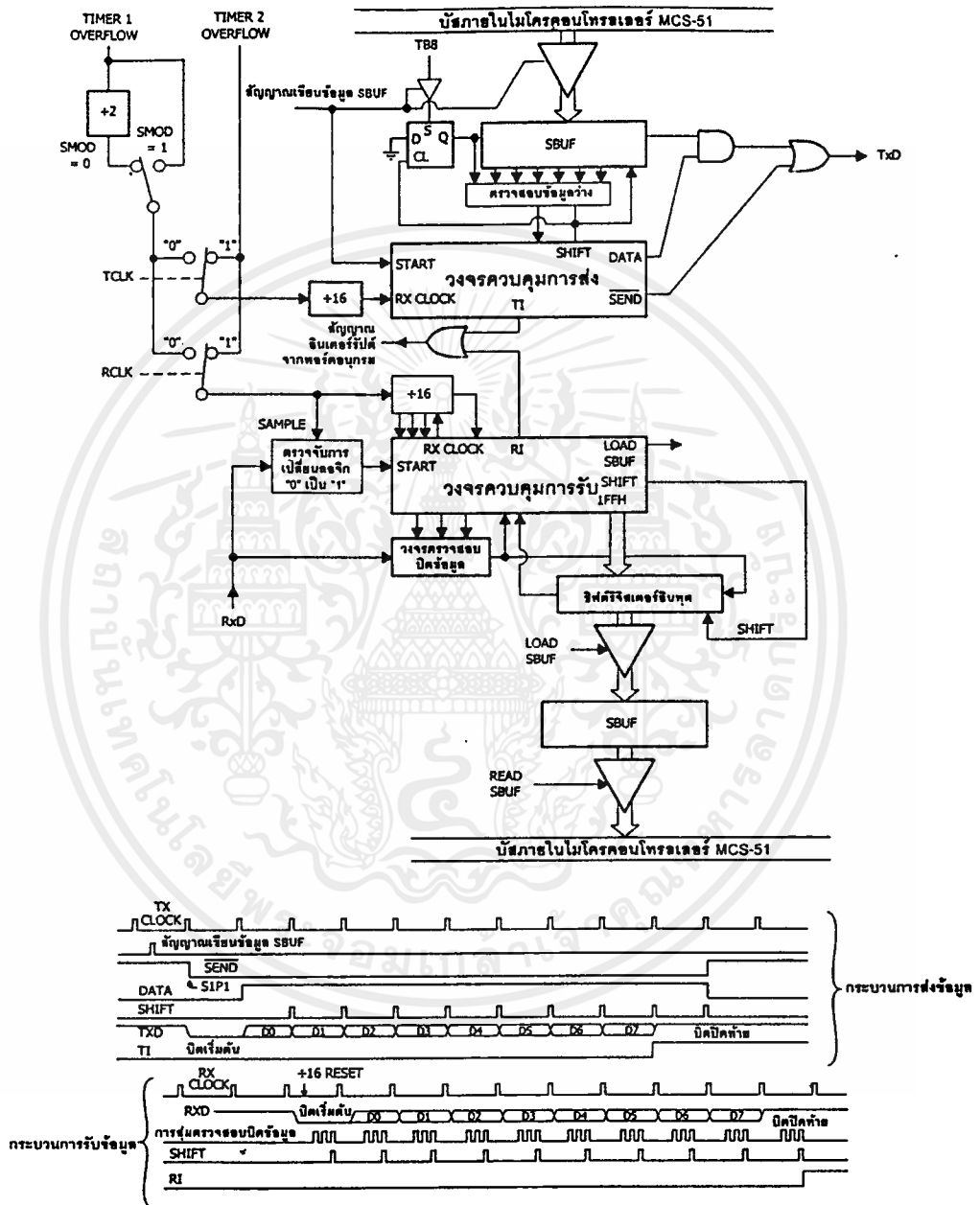
ในกระบวนการรับข้อมูล เริ่มต้นด้วยการเซต REN ให้เป็น “1” และเคลียร์บิต RI ในรีจิสเตอร์ SCON ก่อน ที่สเตต 6 เฟส 2 ของแมชชีนไจเกิดถัดไป วงจรควบคุมการรับ (RX control) จะทำการเขียนข้อมูล 11111110 ไปยังชิพรีจิสเตอร์สำหรับรับข้อมูลและทำการแอกทีฟสัญญาณ RECEIVE ให้เป็น “1” ในสัญญาณนาฬิกาถูกล็อคไป

เมื่อสัญญาณ RECEIVE แอกทีฟ ก็จะมีการส่งสัญญาณนาฬิกาของการเลื่อนข้อมูล (Shift clock) ขึ้นผ่านทางขา P3.1 หรือ TxD เพื่อทำการกำหนดจังหวะการรับข้อมูลครั้งละบิต โดยสัญญาณนาฬิกาจะเกิดขึ้นในช่วงสเตต 3 เฟส 1 ถึงสเตต 6 เฟส 1 ของแต่ละแมชชีนไจเกิด การรับข้อมูลเข้ามาทางขา P3.0 หรือ RxD จะเกิดขึ้นที่สเตต 5 เฟส 2 ในแมชชีนไจเกิดเดียวกับสัญญาณนาฬิกาของการเลื่อนข้อมูล จนกระทั่งรับข้อมูลครบทั้ง 8 บิต บิต RI จะได้รับการเซตเพื่อแจ้งการเสร็จสิ้นกระบวนการรับข้อมูล หากการอินเตอร์รัปต์จากพอร์ตอนุกรมได้รับการอินาเบิลไว้ ก็จะมีการอินเตอร์รัปต์ขึ้นในระบบ เมื่อเสร็จสิ้นกระบวนการรับข้อมูล สัญญาณ RECEIVE จะกลายเป็น “0” จนกว่าจะเริ่มกระบวนการรับข้อมูลใหม่

การทำงานในโหมดนี้ของพอร์ตอนุกรมไมโครคอนโทรลเลอร์ MCS-51 จะใช้ประโยชน์ในการเชื่อมต่อกับไอซีรีจิสเตอร์ภายนอกเพื่อทำการขยายจำนวนพอร์ตอินพุตหรือเอาต์พุต แต่ไม่เป็นที่นิยมใช้งานมากนัก เนื่องจากไมโครคอนโทรลเลอร์ MCS-51 เองมีพอร์ตอยู่ค่อนข้างมาก และสามารถติดต่อกับพอร์ตเหล่านั้นได้ง่ายและเร็วกว่ามาก

การทำงานในโหมด 1 ของวงจรพอร์ตอนุกรม

มีไคอะแกรมการทำงานและไคอะแกรมเวลาแสดงในรูปที่ 2.25 ในโหมดนี้จะใช้ในการรับส่งข้อมูลรวม 10 บิต โดยส่งข้อมูลออกทางขา P3.1 หรือ TxD และรับข้อมูลเข้าทางขา P3.0 หรือ RxD ข้อมูลรวมทั้ง 10 บิตประกอบด้วย บิตเริ่มต้น (มีค่าเป็น “0”) 1 บิต บิตข้อมูล 8 บิต โดยรับหรือส่งข้อมูลในบิต LSB ก่อน และบิตหยุดหรือบิตปิดท้าย (มีค่าเป็น “1”) ในการรับข้อมูล บิตหยุดจะถูกเก็บไว้ในบิต RB8 ในรีจิสเตอร์ SCON อัตราบอดในโหมดนี้ได้รับการกำหนดโดยอัตราการเกิดโอเวอร์โพลวของไทมเมอร์ 1 ใน AT89C51 ส่วนในไมโครคอนโทรลเลอร์เบอร์ AT89C52 และในอนุกรม AT89Sxx สามารถเลือกใช้ อัตราการเกิดโอเวอร์โพลวของไทมเมอร์ 1 หรือไทมเมอร์ 2 ในการกำหนดเป็นอัตราบอดได้



รูปที่ 2.25 โค้ดแกรมการทำงานในโหมด 1 ของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS - 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

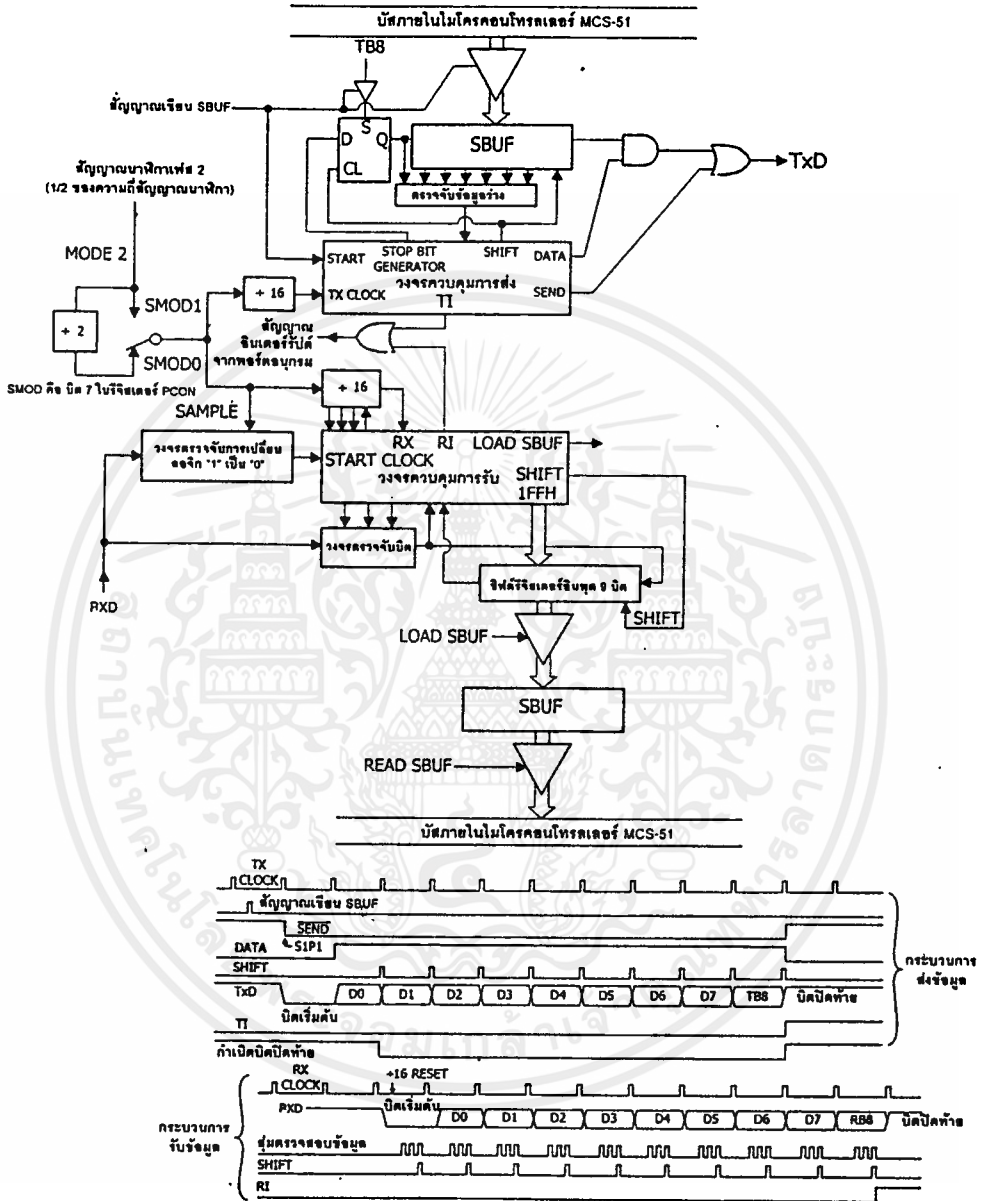
กระบวนการส่งข้อมูลเริ่มต้นด้วยการแอกทีฟสัญญาณเขียนข้อมูลมายังรีจิสเตอร์ SBUF ส่งมายังวงจรควบคุมการส่ง (TX control) จากนั้นวงจรควบคุมการส่งจะทำการแอกทีฟสัญญาณ SEND ที่สแตต 1 เฟส 1 ของแมชีนไซเกิดต่อมา โดยสัญญาณ SEND จะเป็น "0" ตลอดกระบวนการส่งข้อมูล เมื่อสัญญาณ SEND แอกทีฟจะทำการส่งบิตเริ่มต้นก่อนเป็นบิตแรกโดยมีคาบเวลาของบิตเริ่มต้นเท่ากับ 1 แมชีนไซเกิดจากนั้นตามด้วยการส่งข้อมูล 8 บิต เรียงลำดับจากบิต LSB โดยข้อมูล que ทำการส่งนี้จะถูกเรียกออกมาจากรีจิสเตอร์บัฟเฟอร์สำหรับการส่งข้อมูล ในทุกๆ บิตข้อมูลที่ทำ การส่งออกไป จะเกิดสัญญาณพัลส์ SHIFT ขึ้น เพื่อให้เรียกข้อมูลในแต่ละบิตจากรีจิสเตอร์บัฟเฟอร์ การกำหนดจังหวะการส่งข้อมูลจะใช้สัญญาณนาฬิกาการส่ง (TX clock) เป็นตัวกำหนด โดย สัญญาณนาฬิกานี้ได้มาจากการหารสัญญาณ TCLK จากไทมเมอร์ 1 ด้วย 16 หลังจากการส่งบิตข้อมูล ก็จะทำการส่งบิตหยุดหรือบิตปิดท้ายขนาด 1 บิต ดังนั้นการส่งข้อมูลจะใช้สัญญาณนาฬิกาทั้งหมด 10 ลูก เมื่อทำการส่งข้อมูลครบเรียบร้อยแล้ว จะทำการเซตบิต TI ในรีจิสเตอร์ SCON หากการอิน เตอร์รัปต์จากพอร์ตอนุกรมได้รับการอินเอบิลไว้ ก็จะเกิดการอินเตอร์รัปต์ขึ้นในระบบ

หลังจากที่ทำการบริการอินเตอร์รัปต์หรือส่งข้อมูลเรียบร้อยแล้ว ต้องทำการเคลียร์บิต TI ก่อนเป็นอันดับแรก เพื่อให้สามารถเริ่มต้นกระบวนการรับส่งข้อมูลทางพอร์ตอนุกรมดำเนินต่อไป ได้

ด้านการรับข้อมูล จะทำการตรวจจับการเปลี่ยนแปลงระดับลอจิกจาก "1" เป็น "0" ที่ขา RxD โดยใช้อัตราการสุ่มเท่ากับ 1/16 เท่าของอัตราบอด เมื่อตรวจจับพบ ไทมเมอร์/คาน์เตอร์ที่ใช้ใน การกำหนดอัตราบอดจะรีเซต และทำการเขียนข้อมูล IFFH ไปยังชิพรีจิสเตอร์อินพุต

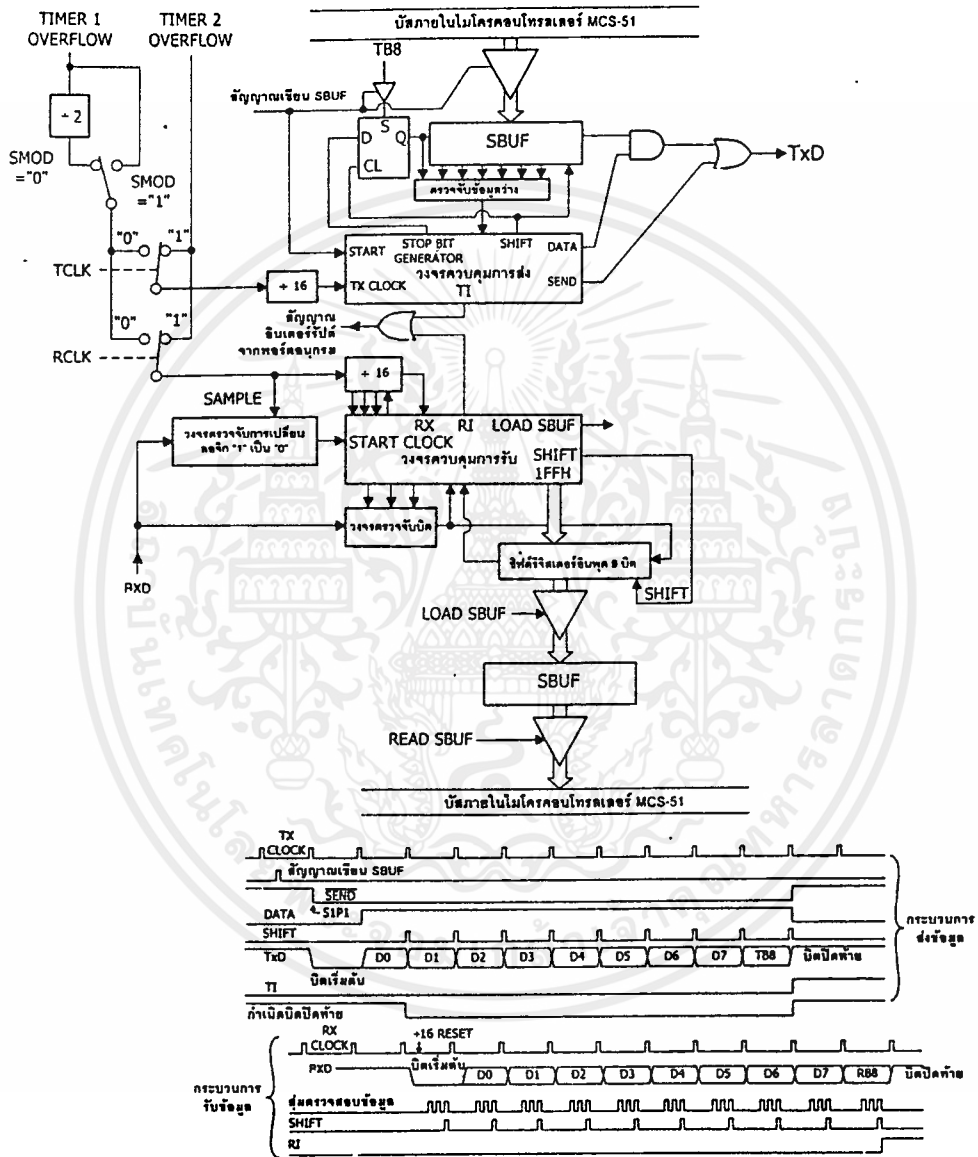
ข้อมูลจะเริ่มเดินทางเข้าสู่พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ผ่านทางขา RxD ในการ ตีความว่าบิตที่เข้ามาเป็น "0" หรือ "1" จะใช้ผลการสุ่มข้างมาก โดยบิตขิงข้อมูลที่เข้ามาได้รับการ แบ่งออกเป็น 16 สแตต การสุ่มข้อมูลจะทำการสุ่มสแตตที่ 7, 8 และ 9 หาก 2 ใน 3 ของการสุ่มพบว่า ข้อมูลเป็นลอจิกใด จะตีความข้อมูลในบิตนั้นเป็นตามเสียงข้างมาก ยกตัวอย่างสุ่มพบ ลอจิก "1" 2 ใน 3 ครั้ง จะตีความว่าบิตของข้อมูลที่รับได้นั้นเป็น "1"

ลำดับการรับข้อมูลมีลักษณะเกี่ยวกับการส่งข้อมูลคือ เริ่มด้วยบิตเริ่มต้นก่อน ตามด้วยบิต ข้อมูล และบิตปิดท้าย ในทุกๆ การรับข้อมูลได้ 1 บิต จะมีพัลส์ SHIFT เกิดขึ้น เพื่อทำการเลื่อนข้อมูลเข้าสู่รีจิสเตอร์บัฟเฟอร์การรับข้อมูล การกำหนดจังหวะการรับข้อมูลใช้สัญญาณนาฬิกาการรับ ข้อมูล (RX clock) หลังจากสัญญาณนาฬิกาถูกสุดท้าย อันหมายถึงสามารถรับข้อมูลได้ครบแล้ววง จรควบคุมการรับข้อมูลจากรีจิสเตอร์บัฟเฟอร์ไปยังรีจิสเตอร์ SBUF และบิต RB8 ในรีจิสเตอร์ SCON โดยข้อมูลในบิต RB8 ก็คือข้อมูลของบิตหยุดนั่นเอง พร้อมกันนั้นยังทำการเซตบิต RI ใน



รูปที่ 2.26 โค้ดแอมแกรมการทำงานในโหมด 2 ของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS - 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.27 ไลอองแกรมการทำงานในโหมด 3 ของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS - 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ SCON ด้วย หากการอินเตอร์รัปต์จากพอร์ตอนุกรมได้รับการอินทิเกรตไว้ ก็จะเกิดการอินเตอร์รัปต์ขึ้นในระบบ

หลังจากที่ทำการบริการอินเตอร์รัปต์หรือรับข้อมูลเรียบร้อยแล้ว ต้องทำการเคลียร์บิต RI ก่อนเป็นอันดับแรก เพื่อให้สามารถเริ่มต้นกระบวนการรับส่งข้อมูลทางพอร์ตอนุกรมดำเนินต่อไปได้

การทำงานในโหมดนี้ได้รับความนิยมสูงสุด เนื่องจากมีกระบวนการที่ไม่ซับซ้อนและสามารถทำการรับส่งข้อมูลกับคอมพิวเตอร์ได้อย่างมีประสิทธิภาพ

การทำงานในโหมด 2 และ 3 ของวงจรถ่ายโอนข้อมูล

ในทั้งสองโหมดนี้จะใช้รูปแบบข้อมูลรวม 11 บิต ประกอบด้วยบิตเริ่มต้น มีค่าเป็น "0" จำนวน 1 บิต, บิตข้อมูล 8 บิต โดยทำการรับและส่งบิต LSB ก่อน, บิตข้อมูลบิตที่ 9 และบิตปิดท้าย มีค่าเป็น "1" จำนวน 1 บิต ในการส่งข้อมูล ข้อมูลบิตที่ 9 จะได้รับการเก็บไว้ในบิต TB8 ในรีจิสเตอร์ TCON และในการรับข้อมูล ข้อมูลบิตที่ 9 จะนำไปเก็บไว้ในบิต RB8 ในรีจิสเตอร์ SCON สำหรับอัตราบอดในโหมด 2 จะคงที่โดยเลือกได้ 2 ค่าคือ 1/32 หรือ 1/64 ของความถี่สัญญาณนาฬิกา สำหรับในโหมด 3 อัตราบอดสามารถปรับได้เหมือนกับในโหมด 1

ในรูป 2.26 และ 2.27 เป็นไดอะแกรมการทำงานในโหมด 2 และไดอะแกรมการทำงานในโหมด 3 ของพอร์ตอนุกรม การทำงานโดยรวมจะคล้ายกับการทำงานในโหมด 1 ส่วนที่แตกต่างกันคือจำนวนบิตของข้อมูลในโหมด 2 และ 3 จะมีเพิ่มมาอีก 1 บิต โดยส่วนใหญ่จะใช้เป็นบิตตรวจสอบพาริตี

อัตราบอดของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51

โหมด 0

อัตราบอดของโหมดนี้มีค่าคงที่ โดยสามารถคำนวณได้จากสูตรต่อไปนี้

อัตราบอดในโหมด 0 = ความถี่ของสัญญาณนาฬิกา/12 หน่วยเป็น บิตต่อวินาที

โหมด 1 และ 3

เนื่องจากทั้งสองโหมดนี้ สามารถเลือกแหล่งกำเนิดอัตราบอดได้ 2 แหล่งคือ จากอัตราโอเวอร์โพลวของไทมเมอร์ 1 และ 2 สำหรับอัตราบอดเมื่อใช้การโอเวอร์โพลวของไทมเมอร์ 1 จะใช้ค่าของบิต SMOD ในรีจิสเตอร์ PCON มาพิจารณาประกอบด้วย โดยสามารถคำนวณหาค่าอัตราบอดได้จาก

$$\text{อัตราบอด} = (2^{\text{ค่าของบิต SMOD}} / 32) * \text{อัตราโอเวอร์โพลวของไทมเมอร์ 1}$$

ถ้าหากในไทมเมอร์ 1 ไม่ได้อินทิเกรตอินเตอร์รัปต์ไว้ สามารถคำนวณหาค่าอัตราบอดได้จาก

$$\text{อัตราบอด} = (2^{\text{ค่าของบิต SMOD}} / 32) * (\text{ความถี่สัญญาณนาฬิกา} / \{12 * [256 - (\text{TH1})]\})$$

ในตารางที่ 2.4 แสดงการกำหนดอัตราบอดโดยใช้ไทมเมอร์ 1

ในกรณีที่ใช้ไทมเมอร์ 2 ในการกำหนดอัตราบอด โดยกำหนดให้ไทมเมอร์ 2 ทำงานในโหมดกำเนิดอัตราบอด (baud rate generator) สามารถคำนวณหาค่าอัตราบอดได้จาก

$$\text{อัตราบอด} = \text{อัตราโอเวอร์โฟลวของไทมเมอร์ 2} / 16 \text{ หน่วยเป็น บิตต่อวินาที}$$

ถ้าหากกำหนดไทมเมอร์ 2 ทำงานในโหมดไทมเมอร์หรือเคาน์เตอร์ตามปกติ สามารถคำนวณหาค่าอัตราบอดได้จาก

$$\text{อัตราบอด} = (\text{ความถี่ของสัญญาณนาฬิกา} / (32 * (65536 - (\text{RCAP2H}, \text{RCAP2L}))))$$

อัตราบอด (บิตต่อวินาที : bps)	ความถี่สัญญาณ นาฬิกา	SMOD	ไทมเมอร์ 1		
			C/T	โหมด	ค่ารี โหลด
โหมด 0 : สูงสุด 1 MHz	12 MHz	X	X	X	X
โหมด 2 : สูงสุด 375K	12 MHz	1	X	X	X
โหมด 1,3 : 62.5K	12 MHz	1	0	2	FFH
19.2K (19,200)	11.0592 MHz	1	0	2	FDH
9.6K (9,600)	11.0592 MHz	0	0	2	FDH
4.8K (4,800)	11.0592 MHz	0	0	2	FAH
2.4K (2,400)	11.0592 MHz	0	0	2	F4H
1.2K (1,200)	11.0592 MHz	0	0	2	E8H
137.5	11.0592 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FEEBH

ตารางที่ 2.4 การเลือกอัตราบอดของวงจรถ่ายรูปอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51

โดยที่ (RCAP2H, RCAP2L) เป็นค่าของรีจิสเตอร์ RCAP2H และ RCAP2L มีขนาด 16 บิต
ไม่คิดเครื่องหมาย

โหมด 2

ในโหมดนี้การกำหนดอัตราบอดจะขึ้นอยู่กับค่าของบิต SMOD ในรีจิสเตอร์ PCON ถ้า SMOD เป็น "0" อัตราบอดจะเท่ากับ 1/64 ของความถี่สัญญาณนาฬิกา ในกรณีที่ SMOD เป็น "1"

อัตราบอดจะเท่ากับ $1/32$ ของความถี่สัญญาณนาฬิกา สามารถแสดงเป็นสูตรคำนวณทางคณิตศาสตร์ได้ดังนี้

$$\text{อัตราบอด} = (2^{\text{ค่าของบิต SMOD}} / 64) * \text{ความถี่สัญญาณนาฬิกา}$$

การกำหนดค่าของไทมเมอร์เพื่อเลือกอัตราบอด

ในการใช้งานพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 สิ่งที่ต้องให้ความสนใจมากที่สุดประการหนึ่งคือ อัตราการถ่ายเทข้อมูล หรืออัตราบอด ซึ่งการกำหนดอัตราบอดนั้นจะขึ้นอยู่กับค่าความถี่ของสัญญาณนาฬิกาเป็นหลัก สำหรับโหมดการทำงานของพอร์ตอนุกรมที่สามารถเลือกอัตราบอดได้อย่างอิสระคือในโหมด 1 และ 3 โดยกำหนดได้จากอัตราการเกิดโอเวอร์โฟลวของไทมเมอร์ 1 ถ้าหากไทมเมอร์ 1 มีการโอเวอร์โฟลวในอัตราที่สูงมากเท่าใด อัตราบอดก็จะมีค่าสูงมากขึ้นตาม นั่นหมายความว่า อัตราในการถ่ายเทข้อมูลจะสูงมาก สามารถถ่ายเทข้อมูลได้อย่างรวดเร็ว ดังนั้นการกำหนดค่าและโหมดการทำงานของไทมเมอร์ 1 จึงเป็นสิ่งที่ส่งผลต่อการเปลี่ยนแปลงของอัตราบอดด้วย

ในการใช้งานไทมเมอร์ 1 เพื่อกำหนดอัตราบอดในโหมด 1 และ 3 ของพอร์ตอนุกรมจะต้องกำหนดให้ไทมเมอร์ 1 ทำงานในโหมด 2 หรือโหมด 8 บิตแบบตั้งค่าการนับอัตโนมัติ การกำหนดค่ารีโหลดให้แก่วีลิสเตอร์ TH1 จึงเป็นตัวแปรหลักที่ใช้ในการกำหนดอัตราบอดให้แก่พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51

เริ่มต้นด้วยการเคลียร์บิต SMOD ซึ่งเป็นบิต 7 ของรีจิสเตอร์ PCON ให้เป็น "0" ค่าของการรีโหลดให้แก่ TH1 สามารถคำนวณได้จาก

$$TH1 = 256 - ((\text{ค่าความถี่ของคริสตอล} / 384) / \text{อัตราบอด})$$

แต่ถ้าบิต SMOD เกิดการเซต จะเป็นการอินวาเบิลการทวิคูณของอัตราบอด ดังนั้นการกำหนดค่าให้แก่ TH1 จึงต้องคำนวณจาก

$$TH1 = 256 - ((\text{ค่าความถี่ของคริสตอล} / 192) / \text{อัตราบอด})$$

ยกตัวอย่าง ถ้าหากในไมโครคอนโทรลเลอร์ AT89C51 ใช้คริสตอล 11.0592 MHz ต้องการกำหนดอัตราบอดของพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ไว้ที่ 19,200 บิตต่อวินาที ในกรณีที่ไมอินวาเบิลการทวิคูณของอัตราบอด ค่ารีโหลดของไมโครคอนโทรลเลอร์จะเท่ากับ

$$\begin{aligned} TH1 &= 256 - ((\text{ค่าความถี่ของคริสตอล} / 384) / \text{อัตราบอด}) \\ &= 256 - ((11059200 / 384) / 19200) \\ &= 256 - (28800 / 19200) \\ &= 256 - 1.5 = 254.5 \end{aligned}$$

เนื่องจากผลลัพธ์ที่ได้เป็นค่าที่ไม่ใช่จำนวนเต็มถ้าหากกำหนดค่าของ THI เป็น 254 เมื่อทำการแทนค่าเพื่อคำนวณหาค่าอัตราบอด จะได้อัตราบอดเท่ากับ 14,400 บิตต่อวินาที และถ้าหากกำหนดค่าของ THI เป็น 255 อัตราบอดจะมีค่าเท่ากับ 28,800 บิตต่อวินาที ดังนั้นจะเห็นได้ว่าค่าของ THI ที่ไม่เป็นจำนวนเต็มจะไม่สามารถทำให้เกิดอัตราบอดตามที่ต้องการได้

ทางแก้ไขคือ ให้ทำการอื่นาเปิดการทวีคูณอัตราบอด โดยการเซตบิต SMOD ในรีจิสเตอร์ PCON ให้เป็น “1” จากนั้นทำการแทนค่าลงในสมการหาค่า THI เมื่อมีการเซตบิต SMOD ได้ผลดังนี้

$$\begin{aligned} \text{TH1} &= 256 - ((\text{ค่าความถี่ของคริสตอล}/192)/\text{อัตราบอด}) \\ &= 256 - ((11059200/192)/19200) \\ &= 256 - (57600/19200) \\ &= 256 - 3 = 253 \end{aligned}$$

นำค่าของ THI ที่ได้ทำการแทนค่าคำนวณหาอัตราบอดจะได้เท่ากับ 19,200 บิตต่อวินาที สามารถสรุปขั้นตอนในการเลือกอัตราบอด โดยการกำหนดค่าของ ไทเมอร์ 1 ได้ดังนี้

1. กำหนดให้ พอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 ทำงานในโหมด 1 หรือ 3
2. กำหนดให้ ไทเมอร์ 1 ทำงานในโหมด 2 หรือ โหมด 8 บิตตั้งค่าอัตราบอด
3. กำหนดข้อมูลให้แก่ THI เท่ากับ 253 เพื่อให้สามารถกำหนดอัตราบอดได้ 19,200 บิตต่อวินาทีตามที่ต้องการ
4. ทำการเซตบิต SMOD ซึ่งเป็นบิต 7 ของรีจิสเตอร์ PCON เพื่ออื่นาเปิดการทวีคูณอัตราบอด

การเขียนหรือส่งข้อมูลออกจากพอร์ตอนุกรม

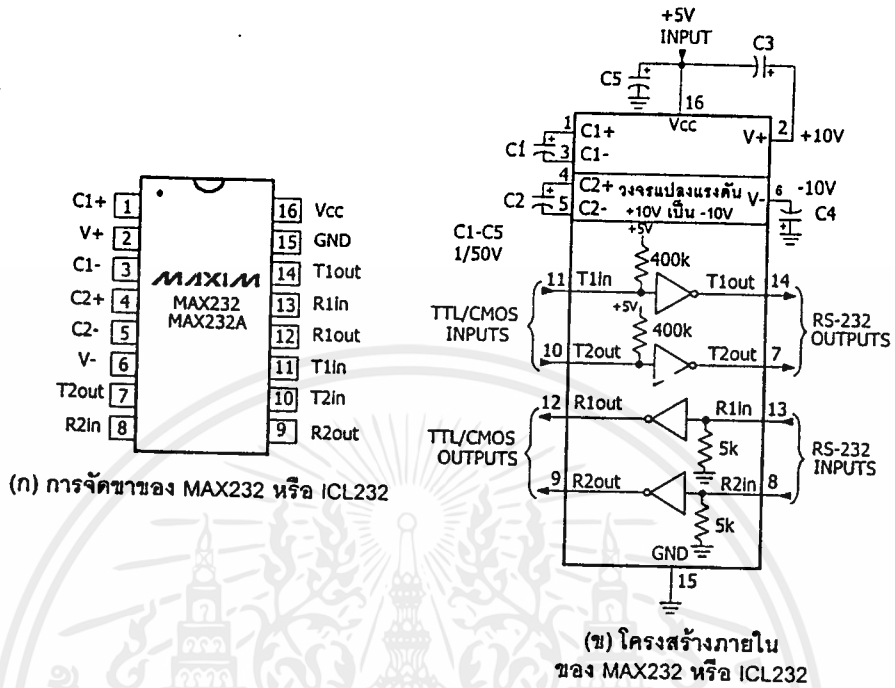
ข้อมูลที่ต้องการส่งออกทุกค่าต้องนำไปเก็บไว้ในรีจิสเตอร์บัฟเฟอร์ของพอร์ตอนุกรม ซึ่งก็คือรีจิสเตอร์ SBUF ตามตัวอย่าง

```
MOV SBUF,#'A'
```

จากคำสั่งข้างต้นเป็นการส่งข้อมูลของตัวอักษร A ออกไปยังพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ อย่างไรก็ตามก่อนทำการส่งข้อมูลทุกครั้ง ต้องแน่ใจว่าบิต TI เคลียร์หรือมีค่าเป็น “0” และเมื่อทำการส่งข้อมูลเรียบร้อยแล้ว ก็จะเกิดการเซตบิต TI เพื่อแจ้งให้ทราบ ดังตัวอย่างโปรแกรมต่อไปนี้

```
CLR TI ; เคลียร์บิต TI เพื่อเตรียมส่งข้อมูลออก
MOV SBUF,#'A' ; ส่งข้อมูลตัวอักษร A ไปยังพอร์ตอนุกรม
JNB TI,$ ; รอการเซตบิต TI เพื่อแจ้งการส่งข้อมูลที่เสร็จสมบูรณ์
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

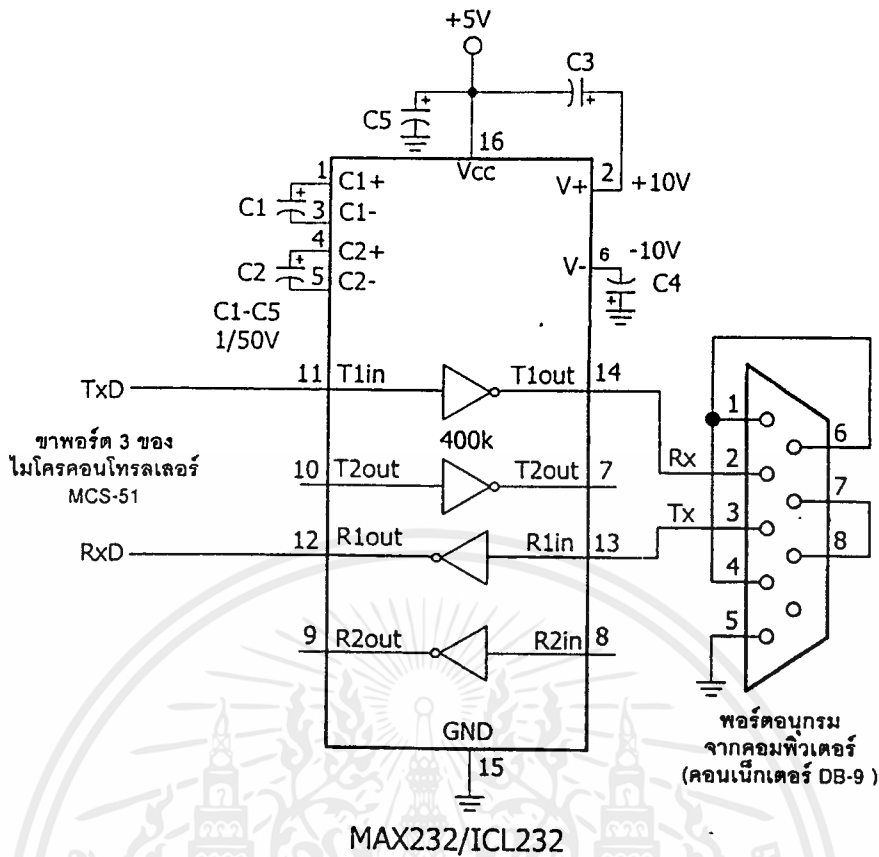


รูปที่ 2.28 รายละเอียดเบื้องต้นของ ไอซีแปลงสัญญาณเพื่อเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์

การอ่านหรือรับข้อมูลจากพอร์ตอนุกรม

การรับข้อมูลจากพอร์ตอนุกรมสามารถกระทำได้ง่ายมากเพียงทำการตรวจสอบว่าบิต RI เกิดการเซตขึ้นหรือไม่ ถ้าพบว่ามีอาการเซตเกิดขึ้นแล้ว ให้ทำการอ่านค่าจากรีจิสเตอร์ SBUF โดยต้องทำการโอนย้ายข้อมูลผ่านทางแอกคิวมูลเตอร์หรือรีจิสเตอร์ A ดังตัวอย่าง

- CLR RI ; เคลียร์บิต RI เพื่อเตรียมรับข้อมูลเข้า
- JNB RI,\$; รอการเซตบิต RI เพื่อแจ้งว่าการรับข้อมูลเสร็จสมบูรณ์ ; และมีข้อมูลเกิดขึ้นที่รีจิสเตอร์ SBUF
- MOV A,SBUF ; อ่านค่าจากรีจิสเตอร์ โดยการโอนย้ายข้อมูลผ่านทาง ; รีจิสเตอร์ A
- CLR RI ; หลังจากทำการอ่านข้อมูลเรียบร้อยแล้ว ต้องทำการ ; เคลียร์บิต RI เสมอ



รูปที่ 2.29 วงจรเชื่อมต่อ MAX 232 หรือ ICL232 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์และไมโครคอนโทรลเลอร์ MCS - 51

การเชื่อมต่อพอร์ตอนุกรมของคอมพิวเตอร์

การใช้งานวงจรพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 มักนิยมใช้ในการติดต่อเพื่อแลกเปลี่ยนข้อมูลกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรมในมาตรฐาน RS-232 เป็นส่วนใหญ่ แต่เนื่องจากระดับสัญญาณของพอร์ตอนุกรม RS-232 มีระดับตั้งแต่ ± 3 ถึง ± 12 ในขณะที่ระดับสัญญาณของไมโครคอนโทรลเลอร์ MCS-51 อยู่ในระดับที่ทีแอล ดังนั้นจึงไม่สามารถเชื่อมต่อพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์ได้โดยตรง จึงต้องอาศัยการเชื่อมต่อผ่านไอซีพิเศษที่ทำหน้าที่ในการแปลงระดับสัญญาณ

ไอซีที่ทำหน้าที่ ในการแปลงระดับสัญญาณนี้ต้องทำการแปลงข้อมูลส่งของไมโครคอนโทรลเลอร์ MCS-51 จากระดับที่ทีแอลไปเป็นระดับของ RS-232 และจะทำการแปลงข้อมูลรับจากคอมพิวเตอร์จากระดับของ RS-232 เป็นระดับที่ทีแอลเพื่อให้สามารถถ่ายทอดไปยังไมโครคอนโทรลเลอร์ MCS-51 ได้อย่างสมบูรณ์ ไอซีดังกล่าวด้วยกันหลายเบอร์จากหลายผู้ผลิตอย่างเช่น MAX232 จาก MAXIM หรือ ICL232 จาก HARRIS เป็นต้น ในรูปที่ 2.28 แสดงการจัดขาของไอซี ICL232 ซึ่งใช้ในการแปลงสัญญาณ RS-232 ส่วนวงจรของการต่อกับไมโครคอนโทรลเลอร์ MCS-51 แสดงในรูป 2.29

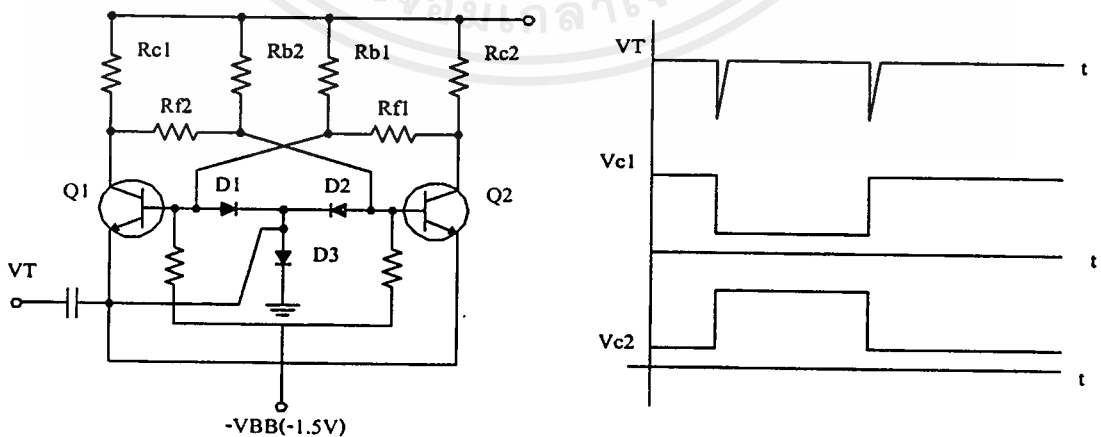
2.7 วงจรมัลติไวเบรเตอร์

หลักการของมัลติไวเบรเตอร์

มัลติไวเบรเตอร์(Multivibrators) เป็นวงจรที่ให้เอาต์พุตสองสถานะคือ “0” กับ “1” โดยอาจจะมีสถานะเสถียรเพียงสถานะเดียวหรือสองสถานะหรือไม่มีสถานะเสถียรเลยก็ได้ เราจึงอาจแบ่งมัลติไวเบรเตอร์ตามสถานะเสถียรของเอาต์พุต ออกเป็น 3 แบบคือ

1. โมโนสเตเบิล (Monostable Multivibrator)
2. ไบสเตเบิล (Bistable Multivibrator)
3. อะสเตเบิล (Astable Multivibrator)

รูปแบบดั้งเดิมของวงจรมัลติไวเบรเตอร์ ประกอบด้วยทรานซิสเตอร์ 2 ตัว ต่อสัญญาณป้อนกลับแบบบวกซึ่งกันและกัน กล่าวคือ สัญญาณป้อนกลับจากคอลเล็กเตอร์ของทรานซิสเตอร์ตัวที่ 2 ต่อเข้าขาเบสของตัวที่ 1 และสัญญาณป้อนกลับจากคอลเล็กเตอร์ของตัวที่ 1 ต่อเข้าขาเบสของตัวที่ 2 รูปแบบที่ง่ายที่สุดคือวงจรไบสเตเบิล ดังแสดงในรูป 2.30 ในกรณีนี้วงจรป้อนกลับสัญญาณทั้งสอง เป็น R (R_{n1} และ R_{n2}) การทำงานของวงจรสามารถอธิบายได้ง่ายๆ ดังนี้ สมมติขณะนั้น Q_2 นำกระแส คอลเล็กเตอร์ของ Q_2 จึงมีสถานะ LO การป้อนกลับของสัญญาณผ่าน R_{n1} มาที่ขา Q_1 จึงทำให้ Q_1 ไม่นำกระแส วงจรจึงอยู่ในสถานะเสถียร โดยมี Q_2 นำกระแส แต่ถ้าเราสามารถทำให้ Q_2 หยุดนำกระแสได้เพียงชั่วขณะ เช่น โดยการป้อนพัลส์ลบเข้าทางเบสของ Q_2 ผลจากการหยุดนำกระแสของ Q_2 คือคอนเล็กเตอร์ของ Q_2 มีค่าแรงดันสูงขึ้น ซึ่งสัญญาณนี้ถูกป้อนกลับมาที่ขาเบสของ Q_1 ทำให้ Q_1 เริ่มนำกระแสอย่างรวดเร็ว วงจรจะเข้าสู่สถานะเสถียรอันใหม่ โดยมี Q_1 นำกระแส ไดโอด D_1 , D_2 และ D_3 ทำหน้าที่ควบคุมทิศทางของสัญญาณกระตุ้น V_T ให้ไปสู่เบสของทรานซิสเตอร์ตัวที่กำลังนำกระแสอยู่ ณ ขณะนั้น เพื่อให้เกิดการเปลี่ยนสถานะขึ้น



รูปที่ 2.30 วงจรไบสเตเบิล

ถ้าเราแทน RF ตัวใดตัวหนึ่งในวงจรรูป 2.30 ด้วย C วงจรจะมีสถานะเสถียรเพียงสถานะเดียวกลายเป็นวงจรโมนอสเตเบิล วงจรในรูป 2.31 ได้รับการออกแบบให้มีสถานะเสถียรคือเมื่อ Q2 นำกระแสเราอาจกระตุ้นให้วงจรเปลี่ยนสถานะได้ชั่วคราว โดยการทำให้ Q2 หยุดนำกระแสชั่วคราว เช่น ป้อนพัลส์ลบที่ B2 เป็นต้น ซึ่งจะส่งผลให้แรงดันที่คอลเล็กเตอร์ของ Q2 เปลี่ยนสถานะจาก LO เป็น HI การเปลี่ยนแปลงนี้จะป้อนกลับผ่าน RF1 ทำให้ Q1 นำกระแส แรงดันที่คอลเล็กเตอร์ Q1 จึงลดลง แต่เนื่องจากแรงดันตกคร่อม C2 จะเปลี่ยนทันทีไม่ได้ แรงดันที่ B2 จึงลดลงด้วยการลดลงของแรงดันที่ B2 นี้เองทำให้ Q2 หยุดนำกระแสต่างๆที่พัลส์ลบที่นำกระตุ้นได้ผ่านพ้นไปแล้ว อย่างไรก็ตามสถานะที่ Q2 หยุดนำกระแสและไม่อาจคงอยู่ตลอดไป เพราะ C1 ต้องผ่าน RB2 ขึ้นสู่ VCC ดังนั้น ในขณะที่ Q1 นำกระแส และแรงดันที่คอลเล็กเตอร์ของ Q1 มีค่าต่ำอยู่นั้น C2 จะเริ่มประจุผ่าน RB2 ทำให้แรงดันที่ B2 สูงขึ้นตามลำดับ จนในที่สุดเมื่อแรงดันที่ B2 สูงพอก็จะทำให้ Q2 นำกระแสขึ้นมาอีก และการป้อนกลับผ่าน RF1 ก็บังคับให้ Q1 นำกระแส วงจรก็จะคืนสู่สถานะเสถียรตามเดิม สถานะที่ Q1 นำกระแสและ Q2 ไม่นำกระแสนั้น เรียกว่าสถานะกึ่งเสถียร (Quasi-stable) ระยะเวลาของการอยู่สถานะกึ่งเสถียรขึ้นอยู่กับเวลาของการประจุ C2 ซึ่งอาจคำนวณได้จากสมการ

$$T = \ln 2 = 0.69 RC$$

$$T = \text{ระยะเวลาที่วงจรอยู่ในสถานะกึ่งเสถียร}$$

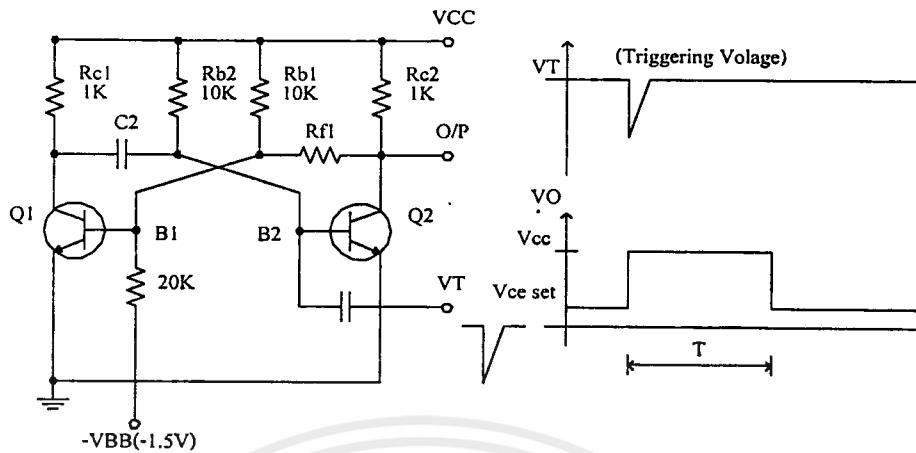
$$\ln 2 = \text{ค่าลอการิทึม ฐาน e ของ 2} = 0.69$$

$$R = \text{ความต้านทานที่จ่ายกระแสให้ C ในที่นี้คือ RB2}$$

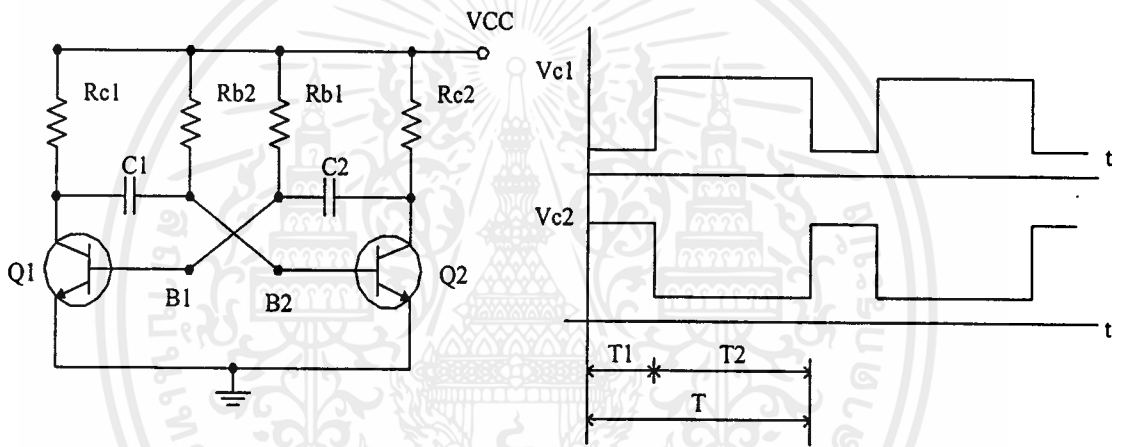
$$C = \text{ความจุไฟฟ้าของตัวเก็บประจุที่ทำการประจุในที่นี้คือ C2}$$

$$T = \text{ค่านิจเวลาของวงจร RC}$$

ถ้าเราเปลี่ยน RF1 เป็นตัวเก็บประจุ C1 ด้วยเราจะได้วงจรอะสแตเบิล ดังที่แสดงในรูปที่ 2.32 วงจรนี้จะไม่มีการเสถียรเลย เพราะทั้งสองซีกของวงจรจะทำงานแบบเดียวกัน และการเปลี่ยนสถานะแต่ละครั้งจะเป็นการเปลี่ยนสถานะเพียงชั่วคราว จนกว่า C จะประจุมากพอที่จะทำให้ทรานซิสเตอร์อีกตัวหนึ่งนำกระแส และเป็นเช่นนี้สลับไปสลับมา อนึ่งการที่วงจรนี้ไม่มีสถานะเสถียรเลยย่อมหมายความว่าวงจรนี้สามารถเปลี่ยนสถานะกลับไปกลับมาได้ โดยไม่ต้องอาศัยสัญญาณอินพุต หรือสัญญาณกระตุ้นใดๆ การทำงานของวงจรอะสแตเบิลจึงเป็นลักษณะการทำงานของวงจรกำเนิดคลื่นหรือออสซิลเลเตอร์นั่นเอง แต่คลื่นที่ได้ออกมาเป็นคลื่นสี่เหลี่ยม



รูปที่ 2.31 วงจรโมโนสเตเบิล



รูปที่ 2.32 วงจรอะสเตเบิล

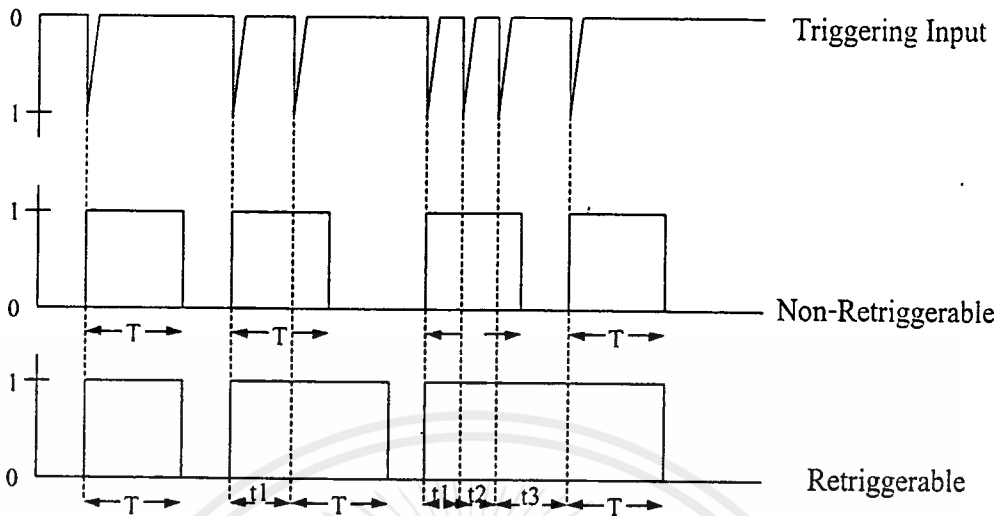
วงจรมอนอสเตเบิล

ปกติ เอาต์พุตของวงจระจะคงสถานะอยู่สถานะหนึ่ง อาจเป็น 0 หรือ 1 ขึ้นอยู่กับการออกแบบ วงจร เอาต์พุตของวงจระจะเปลี่ยนเป็นสถานะตรงข้ามทันทีเมื่อได้รับการกระตุ้น (Triggering) และจะเปลี่ยนกลับคืนสู่สถานะปกติเมื่อเวลาผ่านไปเป็นคาบเวลา (T) ที่แน่นอนค่าหนึ่ง ซึ่งสามารถกำหนดได้ด้วยอุปกรณ์ในวงจร

วงจรมอนอสเตเบิลนี้อาจแบ่งออกเป็น 2 แบบคือ

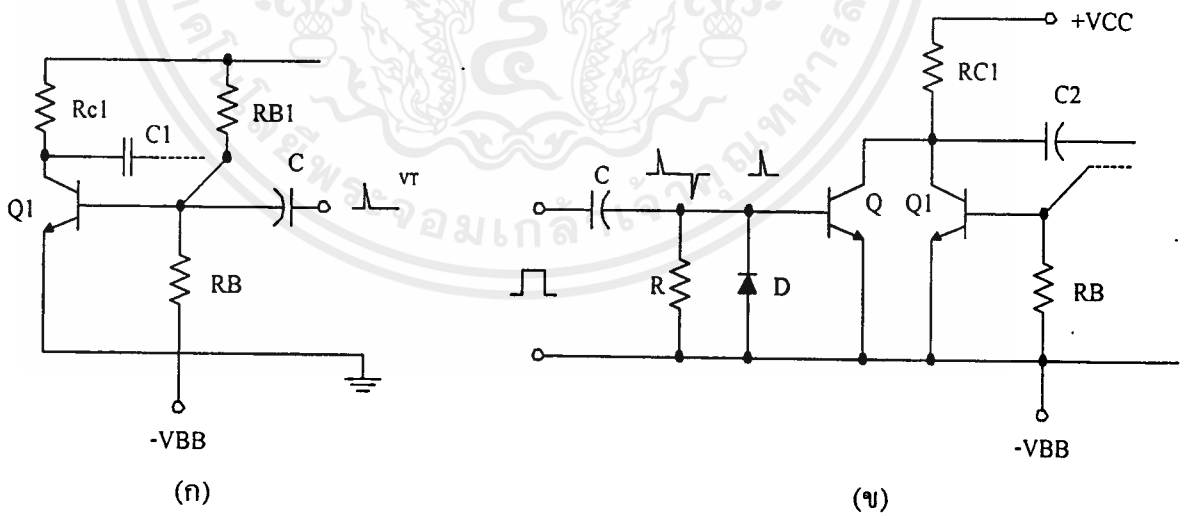
2. แบบไม่รับการกระตุ้นซ้ำ (Non-retriggerable)
3. แบบยอมรับการกระตุ้นซ้ำ (Retriggerable)

ทั้งสองแบบมีหลักการทำงานคล้ายกัน ต่างกันที่มียอมรับการกระตุ้นของอินพุตซ้ำอีกหรือไม่ดังในรูปที่ 2.33



รูปที่ 2.33 การตอบสนองต่อสัญญาณกระตุ้นของวงจรโมโนสเตเบิลทั้งสองแบบ

วงจรที่ไม่รับการกระตุ้นซ้ำจะเริ่มทำงานตั้งแต่สัญญาณกระตุ้นอันดับแรก และขณะที่อยู่ในสถานะกึ่งเสถียรนั้น จะไม่สนใจการกระตุ้นอีก จนกว่าจะกลับเข้าสู่สถานะปกติแล้ว จึงยอมรับการกระตุ้นใหม่ ส่วนวงจรที่ยอมรับการกระตุ้นซ้ำนั้น คาบเวลา T จะเริ่มนับใหม่ทุกครั้งที่ได้รับสัญญาณการกระตุ้น จะเห็นได้ว่าคาบเวลาทั้งหมดของสถานะกึ่งเสถียรแตกต่างกันในกรณีทั้งสอง กล่าวคือ ในกรณีของวงจรโมโนสเตเบิลแบบไม่รับการกระตุ้นซ้ำ คาบเวลาจะคงที่ตลอด แต่ในกรณีของวงจรโมโนสเตเบิลแบบยอมรับการกระตุ้นซ้ำ จะไม่คงที่ โดยขึ้นอยู่กับสัญญาณกระตุ้น



รูปที่ 2.34 วงจรกระตุ้นสำหรับโมโนสเตเบิล

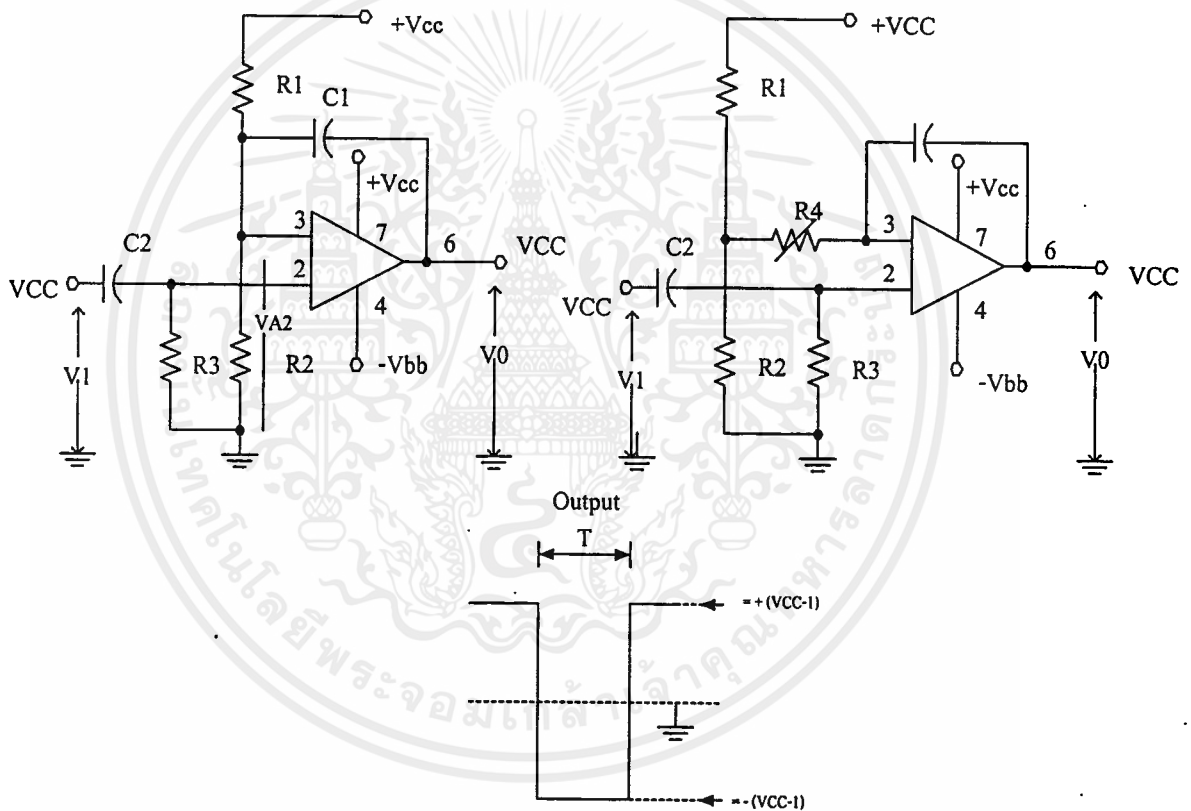
- (ก) กระตุ้นด้วยพัลส์บวก
- (ข) กระตุ้นด้วยขอบขาขึ้นของสัญญาณกระตุ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการกระตุ้นให้เกิดการเปลี่ยนแปลงสถานะนั้นทำได้หลายวิธี นอกจากจะกระตุ้นโดยป้อนพัลส์ลบเข้า B_2 ดังแสดงในรูปที่ 2.31 แล้ว อาจใช้วิธีป้อนพัลส์บวกเข้าทาง B_1 ดังที่แสดงในรูปที่ 2.34 (ก) หรือถ้าเพิ่มเติมวงจรตามรูปที่ 2.34 (ข) แล้ววงจรจะตอบสนองต่อขอบขาขึ้นของสัญญาณกระตุ้น

การกระตุ้นด้วยพัลส์ลบตามรูปที่ 2.31 เป็นแบบไม่รับการกระตุ้นซ้ำ ส่วนการกระตุ้นด้วยพัลส์บวกและขอบขาขึ้นตามรูปที่ 2.34 เป็นแบบยอมรับการกระตุ้นซ้ำ

นอกจากวงจรโมนอสเตเบิลที่สร้างด้วยทรานซิสเตอร์ดังกล่าวแล้ว ในทางปฏิบัติเราอาจใช้ออปแอมป์แทนทรานซิสเตอร์ก็ได้ ดังแสดงในรูปที่ 7.6



รูปที่ 2.35 วงจร โมนอสเตเบิลที่สร้างจากออปแอมป์

ตามรูปที่ 2.35 (ก) ขา Inverting ของออปแอมป์ถูกต่อลงดินผ่าน R_3 ส่วนขา Noninverting ถูกไบอัสโดย R_1, R_2 จึงมีแรงดันสูงกว่าขา Inverting ทำให้เอาต์พุต (V_0) ของ Op-amp เป็นบวก (ประมาณ $V_{CC} - 1$ โวลต์) ในขณะที่ C_1 จะเก็บประจุไว้ในลักษณะที่มีขั้วบวกลบตามรูป และจะคงสถานะนี้ต่อไป จนเมื่อมีสัญญาณพัลส์บวกที่อินพุต (V_2) ที่มีความสูงพอที่จะทำให้ขา Inverting มีแรงดันสูงกว่าขา Non-inverting ก็จะทำให้เอาต์พุต (V_0) มีค่าเป็นลบ (ประมาณ $(V_{EE}-1)$ โวลต์) C_1 ก็จะคายประจุและเกิดการประจุกลับซ้ำโดยผ่าน R_1 ซึ่งในขณะนี้มีแรงดันที่ขา Non-inverting ยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

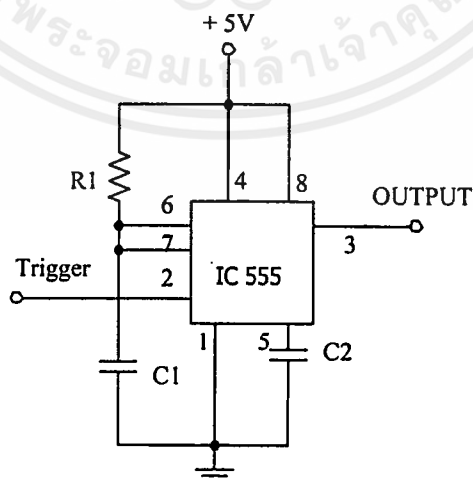
คงมีค่าต่ำกว่าแรงดันที่ขา Inverting เอาต์พุตจะยังคงเป็นลบอยู่จนเมื่อ C1 ประจุมากขึ้น แรงดันที่ขา Non-inverting สูงขึ้นกว่าขา Inverting เอาต์พุตจึงเปลี่ยนกลับเป็นบวกเหมือนเดิมอีกครั้ง สัญญาณเอาต์พุตเป็นดังรูปที่ 2.35 (ข) คาบเวลา (T) ของสัญญาณเอาต์พุตขึ้นอยู่กับค่า C1, R1, R2 และแรงดันไบอัสที่ขา Non-inverting เราอาจทำการปรับปรุงวงจรให้สามารถเปลี่ยนคาบเวลาได้ง่ายๆ โดยเพิ่ม R4 ดังวงจรในรูป 2.35 (ค) โดยที่ R4 ต้องมากกว่า R1, R2 มากๆ

วงจรโมโนสเตเบิลที่เป็นไอซีสำเร็จรูปมีให้เลือกใช้หลายแบบ เช่น ไอซีเบอร์ 555, ไอซีทีทีแอลและซีมอสเป็นต้น

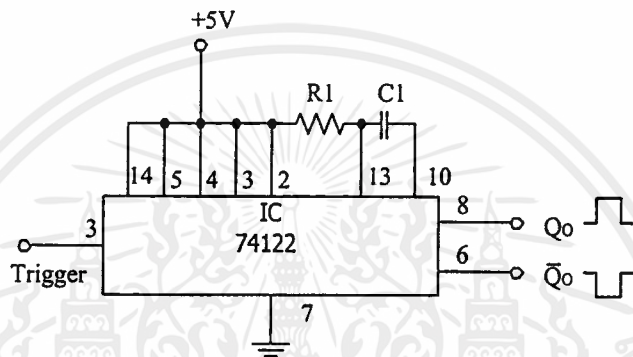
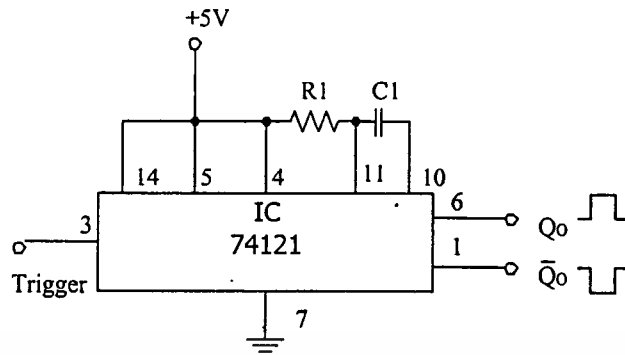
ไอซีเบอร์ 555 จัดอยู่ในจำพวก Linear IC (คำว่า Linear ในที่นี้หมายถึงอนาล็อก) ออกแบบมาสำหรับใช้เป็นวงจรจับเวลา (Timer) มีลักษณะการทำงานเป็นวงจรโมโนสเตเบิลแบบไม่รับการกระตุ้นซ้ำดังรูปที่ 2.36 คาบเวลา (T) ของเอาต์พุตที่เป็น “1” ขึ้นอยู่กับค่า C1, R1 โดยคำนวณได้จากสมการ $T = 1.1 (R1C1)$ (วินาที)

โมโนสเตเบิลที่เป็นไอซีทีทีแอล มีทั้งแบบขอมรับการกระตุ้น (74123) และแบบไม่รับการกระตุ้น (74121)

วงจรตามรูปที่ 2.37 เป็นไอซีทีทีแอล ซึ่งทำหน้าที่เป็นโมโนสเตเบิลแบบขอมรับการกระตุ้นซ้ำ สัญญาณกระตุ้นทางลบซึ่งสามารถเลือกใช้สัญญาณกระตุ้นทางบวกแทนได้ โดยการเปลี่ยนแปลงขาอินพุต และเราสามารถเลือกได้ว่าจะให้เอาต์พุตเป็น “1” หรือเป็น “0” ในสถานะเสถียร คาบเวลา (T) คำนวณได้จาก $T = 0.69 (R1C1)$ (วินาที) นอกจากนี้ยังมีวงจรโมโนสเตเบิลที่เป็นไอซีเบอร์อื่นทั้งทีทีแอลและซีมอสให้เลือกใช้ได้อีกหลายเบอร์ รายละเอียดดูได้จากคู่มือของไอซีนั่นๆ โดยทั่วหลักการทำงานก็คล้ายๆ กันที่กล่าวมาแล้ว



รูปที่ 2.36 ไอซีเบอร์ 555 เป็น โมโนสเตเบิลแบบหนึ่ง



รูปที่ 2.37 โมโนสเตเบิลที่เป็นไอซีทีทีแอล

ประโยชน์ของวงจรโมโนสเตเบิลคือ ใช้ควบคุมความกว้างของพัลส์ให้มีค่าแน่นอน และใช้ปรับสภาพสัญญาณดิจิทัลในเชิงเวลา

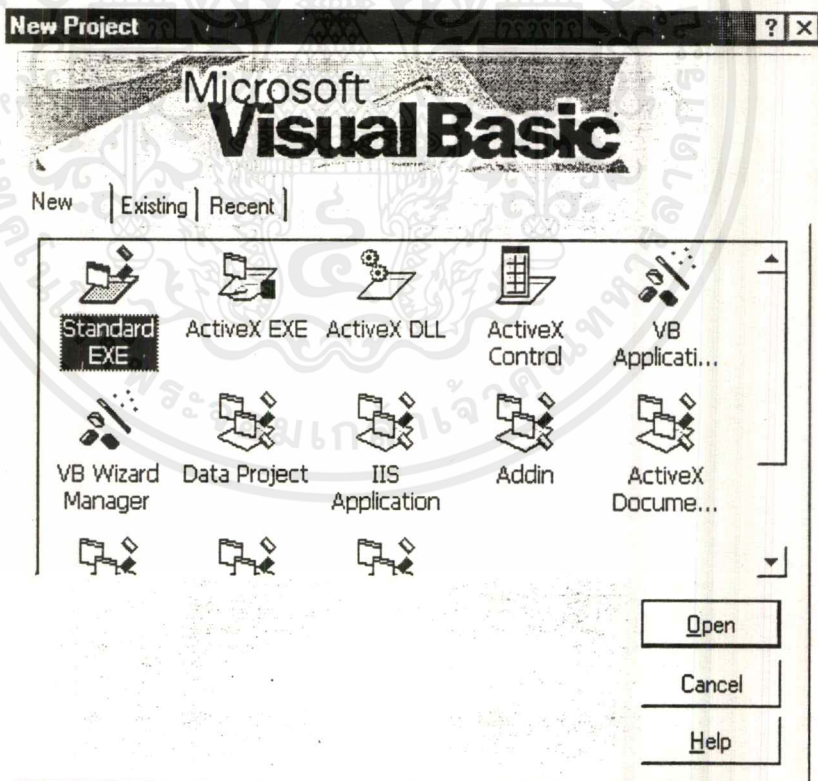
2.8 Visual Basic เบื้องต้น

เราจะพบว่าในปัจจุบันนี้ windows เป็นระบบปฏิบัติการที่นิยมมาก และการใช้ windows ก็แพร่หลายไปทั่วโลก ไม่ว่าใครก็สามารถใช้ windows ได้อย่างง่ายดายค่านั้นเพราะ windows สามารถใช้งานและเรียนรู้ได้ง่าย และจุดเด่นที่สำคัญอีกอย่างนั้นก็คือมันสามารถที่จะทำงานหลายอย่างพร้อมๆ กันได้(Multitasking) ดังนั้น Software ที่เป็นที่ยอมรับปัจจุบันนี้จึงเป็นโปรแกรมที่สามารถทำงานบน windows ได้

Visual Basic เป็นโปรแกรมภาษาของคอมพิวเตอร์ที่พัฒนาให้สามารถใช้งานได้ง่ายและสามารถพัฒนาโปรแกรมบน windows แต่ก่อนจะพบว่าการเขียนโปรแกรมบน windows การที่จะเขียนหน้าต่างของ windows ขึ้นมานั้นจะต้องใช้คำสั่งต่างๆมากมาย แต่ถ้าเป็น Visual Basic แล้วสามารถสร้างได้ง่ายคายต่อไปจะได้อีกกล่าวถึงการใช้งาน Visual Basic 6.0

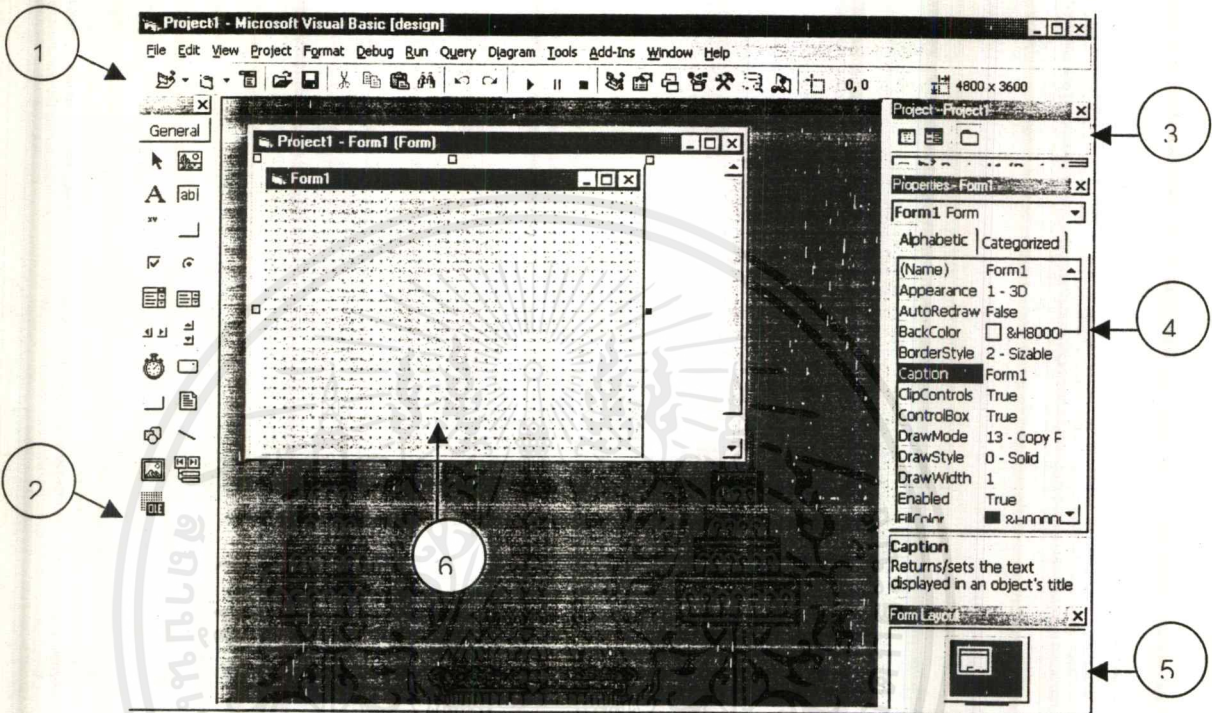
การใช้งาน Visual Basic เบื้องต้น

สิ่งแรกที่จะพบเมื่อเข้าสู่โปรแกรมก็คือหน้าจอสำหรับเปิด project ดังรูป



Don't show this dialog in the future

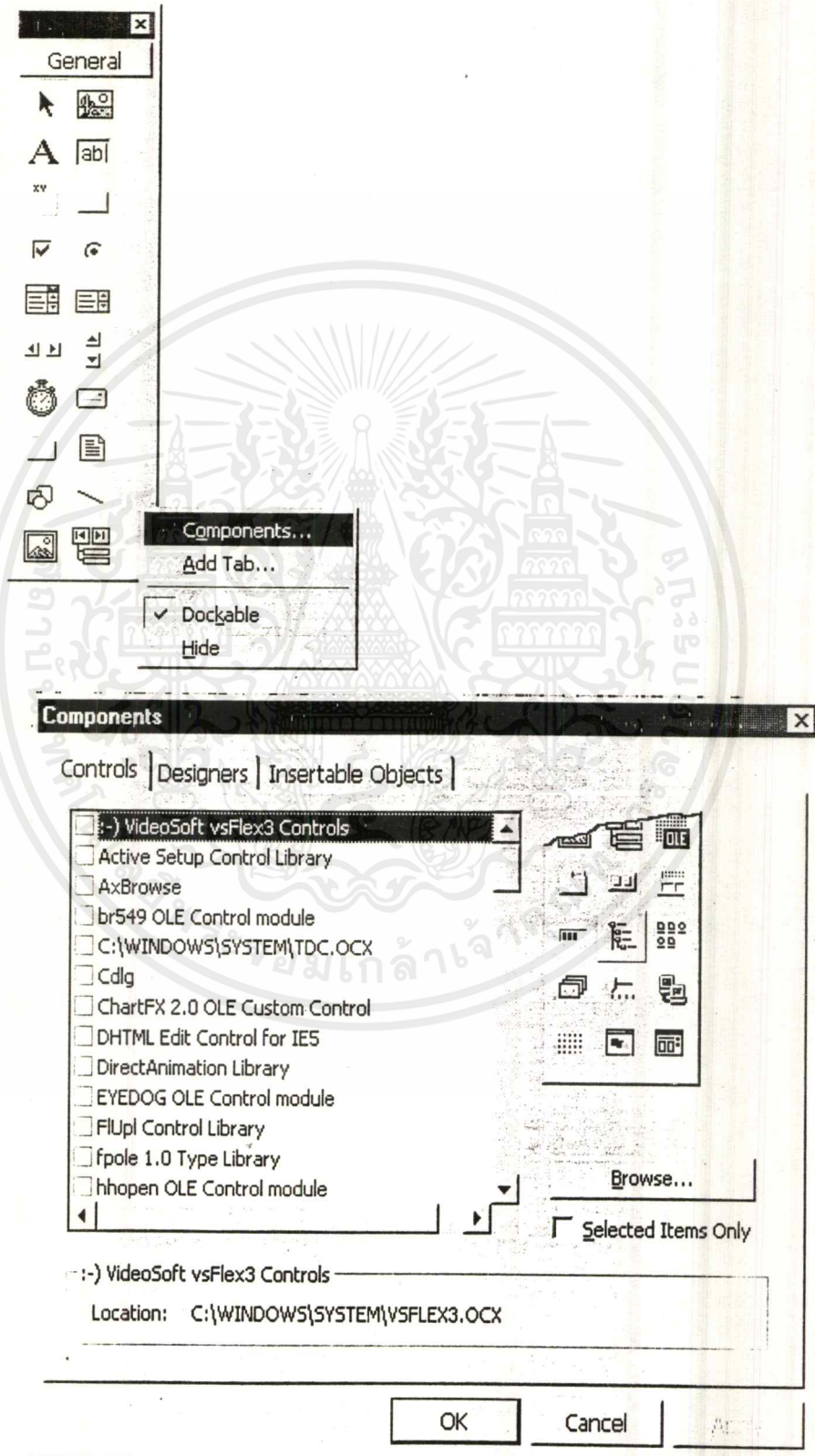
ในนี้จะเลือก “ Standard EXE “ เพื่อเข้าสู่การใช้ Visual Basic ที่ใช้ในการเขียนโปรแกรม
ต่อไปดังรูปข้างล่าง



1. Toolbar จะประกอบด้วย icon ที่ใช้ในการพัฒนา โปรแกรม เหมือนกับในโปรแกรมอื่นๆ
2. Toolbox จะประกอบด้วย icon ต่างๆ ซึ่งใน Visual Basic เราเรียกว่า control
3. Project Explorer Windows จะแสดงรายชื่อ Form ที่พร้อมจะถูกแก้ไข
4. Properties Windows ใช้สำหรับกำหนดคุณสมบัติ (Property) ให้กับ Form และ Object ต่างๆ ที่อยู่บน Form
5. Form Layout Windows จะแสดงตำแหน่งของ Form บนจอคอมพิวเตอร์เมื่อทำการ Run
6. Form. เป็นส่วนที่ใช้ในการสร้างจอเมื่อทำการรัน แล้วจะทำให้มีหน้าต่างอย่างไรซึ่งเราสามารถที่จะตกแต่งให้สวยงามได้ และเป็นส่วนที่เราสามารถเลือกเอา Control ต่างๆ มาวางไว้และสามารถจัดตำแหน่งได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

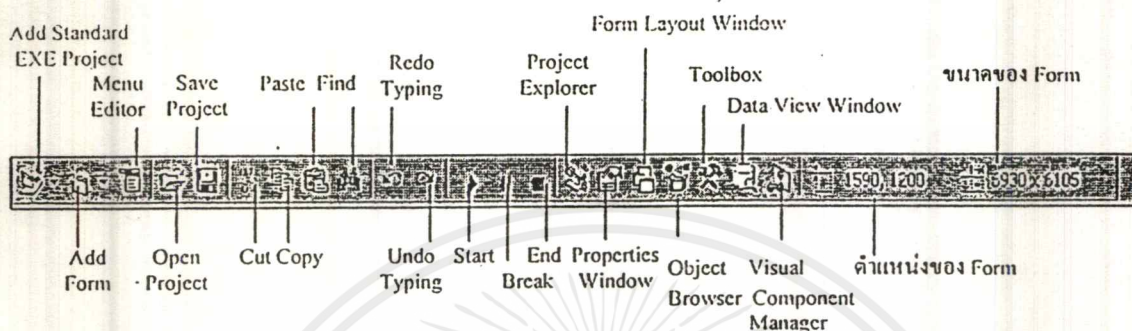
ส่วนของ Control ต่างๆ ที่สามารถดึงมาใช้ได้นอกจากนี้ยังมีอีกส่วน โดยการ Click ขวา ที่บริเวณสีเทาแล้วเลือก Component แล้วทำการเลือก Control ที่ต้องการ ดังรูปข้างล่าง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของ Toolbar












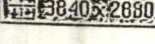
เป็นแถบเครื่องมือที่ประกอบด้วย Icon ต่างๆ คั่งรูป จะมีลักษณะที่คล้ายกับ Toolbar ของ MS-Word และมีหน้าที่ที่คล้ายๆ กัน



Toolbar ทำหน้าที่เป็นผู้ช่วยในการพัฒนาโปรแกรม ซึ่งเมื่อเลื่อนเมาส์ไปชี้ยัง Icon ใด ก็จะปรากฏชื่ออยู่ที่ Icon นั้น แต่ละ Icon จะมีหน้าที่ต่างกันไปดังนี้

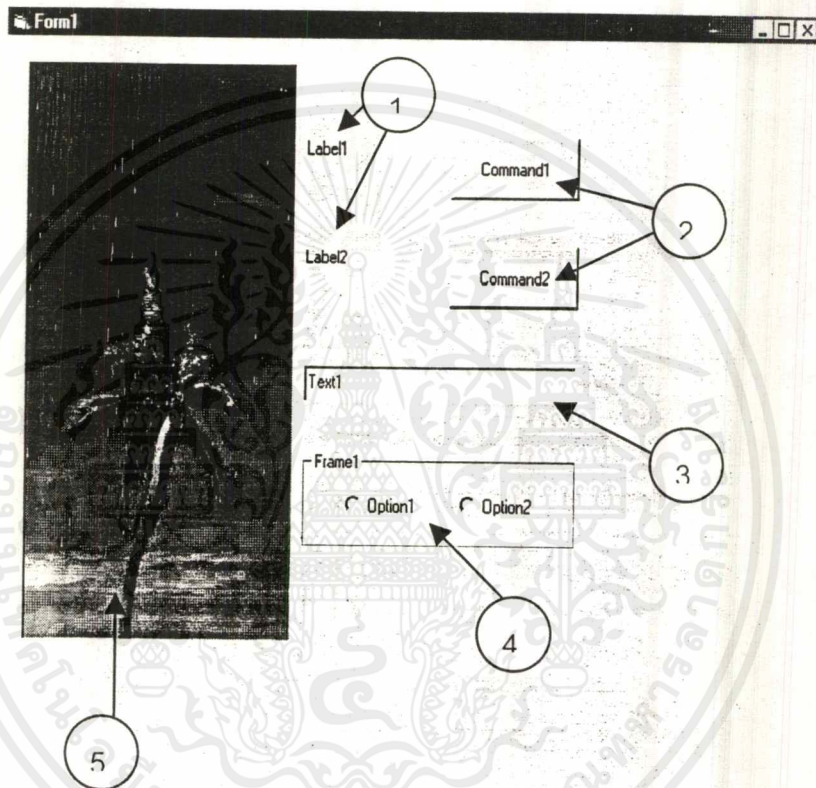
รูป Icon	ชื่อเรียก	รายละเอียดการใช้งาน
	Add Standard EXE-Project	ใช้สำหรับเปิด Project ใหม่ ในกรณีที่มีหลาย Project อยู่บนจอภาพ สามารถสลับไปมาระหว่าง Project ได้โดยเข้าไปเลือกในเมนู Windows
	Add Form	ใช้ในการเพิ่ม Form ให้กับ Project ซึ่งอาจเรียกจากเมนู Project และ Add Form ตามลำดับ
	Menu Editor	ใช้เรียก Menu Editor ซึ่งเป็น Tool สำหรับสร้างเมนูให้กับ Form ซึ่งอาจใช้การกดปุ่ม Ctrl+E หรือเลือกจากเมนู Tools และ Editor ตามลำดับ
	Open Project	ใช้สำหรับเปิด Project ซึ่งอาจใช้ Hot Keys Ctrl+O หรือเรียกจากเมนู File และ Open Project ตามลำดับ
	Save Project	ใช้สำหรับบันทึก Project และ Form ซึ่งอาจเลือกจากเมนู File และ Save Project หรือ Save Project As ตามลำดับ ถ้าต้องการระบุชื่อของ Project ใหม่
	Cut	ใช้สำหรับตัด Object ต่าง ๆ บน Form ซึ่งอาจใช้ Hot Keys Ctrl+X หรือเลือกจากเมนู Edit และ Cut ตามลำดับ
	Copy	ใช้สำหรับ Copy Object บน Form ซึ่งอาจใช้ Hot Keys Ctrl+C หรือเลือกจากเมนู Edit และ Copy ตามลำดับ สำหรับ Object ที่ถูก Copy จะเรียกว่า Control Array ซึ่งจะกล่าวถึงในส่วนต่อไป
	Paste	ใช้สำหรับ Paste Object ที่ Cut หรือ Copy ไว้ ซึ่งอาจใช้ Hot Keys Ctrl+V หรือเลือกจากเมนู Edit และ Paste ตามลำดับ
	Find	ใช้สำหรับค้นหาคำใน Editor ซึ่งเป็น Tool ที่ใช้ในการเขียนโปรแกรม โดยอาจใช้ Hot Keys Ctrl+F หรือเลือกจากเมนู Edit และ Find ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Icon	คำสั่ง	รายละเอียดการใช้งาน
	Undo Typing	ใช้สำหรับยกเลิกคำที่พิมพ์ใน Editor ซึ่งอาจใช้ Hot Keys Ctrl+Z หรือเลือกจากเมนู Edit และ Undo Typing ตามลำดับ ในกรณีที่ไม่ได้อยู่ในจอภาพ Editor จะปรากฏข้อความ Can't Undo แทนทั้งในส่วนของชื่อไอคอน และในเมนู Edit
	Redo Typing	ใช้สำหรับทำซ้ำคำที่พิมพ์ใน Editor ซึ่งอาจเลือกจากเมนู Edit และ Redo Typing ตามลำดับ และก็เช่นเดียวกับ Undo Typing ในกรณีที่ไม่ได้อยู่ในจอภาพ Editor ปรากฏข้อความ Can't Redo แทน
	Start	ใช้สำหรับ Run Project ที่จัดทำขึ้น ซึ่งอาจใช้ Hot Keys F5 หรือเลือกจากเมนู Run และ Start ตามลำดับ
	Break	ใช้สำหรับหยุดการทำงานของ Project ชั่วคราว ซึ่งอาจใช้ Hot Keys Ctrl+Break หรือเลือกจากเมนู Run และ Break ตามลำดับ
	End	ใช้สำหรับหยุดการทำงานของ Project ซึ่งอาจเลือกจากเมนู Run และ End ตามลำดับ
	Project Explorer	ใช้แสดงว่า Project นั้นประกอบไปด้วย Form และ Module ใดบ้าง ซึ่งอาจใช้ Hot Keys Ctrl-R หรือเลือกจากเมนู View และ Project Explorer ตามลำดับ
	Properties Window	ใช้สำหรับกำหนดคุณสมบัติ (Property) ของ Object และ Form ซึ่งอาจใช้ Hot Keys F4 หรือเลือกจากเมนู View และ Properties Window ตามลำดับ
	Form layout - Window	ใช้สำหรับเรียกจอภาพ Form Layout ซึ่งใช้แสดงตำแหน่งของ Form บนจอภาพ โดยอาจเลือกจากเมนู View และ Form Layout Window ตามลำดับ
	Object Browser	ใช้สำหรับเรียกจอภาพ Object Browser ซึ่งใช้แสดงถึง Class และสมาชิกของแต่ละ Class อาจใช้ Hot Keys F2 หรือเลือกจากเมนู View และ Object Browser ตามลำดับ
	Toolbox	ใช้สำหรับเรียก Toolbox ขึ้นมาบนจอภาพ ซึ่งอาจเลือกจากเมนู View และ Toolbar ตามลำดับ
	ตำแหน่งของ Form	ใช้บอกตำแหน่งในแกน X และ Y ของ Form
	ขนาดของ Form	ใช้บอกขนาดของ Form ตามแนวแกน X และ Y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Visual Basic เป็นโปรแกรมที่ง่ายต่อการเรียนรู้ และมีขั้นตอนที่ง่ายด้วยอาทิเช่นว่า เมื่อเราต้องการที่จะต้องการให้หน้าจอของเราเป็นอย่างไรก็แต่เพียงดึงเอา Control นั้นๆ มาวางไว้บน Form จึงทำให้เราไม่ต้องเขียน คำสั่งต่างๆ มากมายจึงทำให้ผู้ใช้สะดวกมาก บนหน้าจอของ Form ก็คือส่วนที่เราจะแสดงเมื่อมีการ Run เราสามารถที่จะตกแต่งให้ดูสวยงามได้ซึ่งจะยกตัวอย่างการเขียน Form และการดึง Control มาใช้งาน



หมายเลข 1 เป็นการใช้ Control ชื่อ Label เมื่อเราดึงมาวางบน Form เราก็จะสามารถที่พิมพ์ข้อความใส่รูปแบบตัวอักษรหรือใส่สีให้ข้อความได้

หมายเลข 2 เป็นการใช้ Control ชื่อ Command Button เป็นลักษณะของปุ่มให้เรา Click ซึ่งจะพบว่าเราจะคุ้นเคยกับ Control ตัวนี้มากในการใช้งาน Windows ซึ่งเราสามารถที่จะเขียนโปรแกรมให้กับตัวมัน โดยมองลักษณะที่ว่าเมื่อเกิดการ Click แล้วให้ทำงานใดๆ ก็ได้

หมายเลข 3 เป็นการใช้ Control ชื่อ TextBox เป็นลักษณะ กล่องข้อความใช้แสดงข้อความ

หมายเลข 4 เป็นการใช้ Control Frame และ Control Option Button ร่วมกันลักษณะเช่นนี้ก็จะพบบ่อยคือเราอาจจะเขียนโปรแกรมให้ผู้ใช้เลือกอย่างใดอย่างหนึ่งจึงเอามารวมเป็น Frame เดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเลข 5 เป็นการใส่ Control Picture Box ก็เป็นการใส่รูปภาพเพื่อเป็นการจัดให้โปรแกรมของเรา มี Presentation ที่สวยงาม

Form ตัวอย่างข้างบนทำให้มองเห็นภาพการใช้งานของ Visual Basic ได้ชัดเจนขึ้นจะพบว่าเราสามารถที่จะจัดรูปแบบตามที่เราร้องการ ดังนั้น Visual Basic จะเป็น presentation ที่ใช้ได้ทีเดียวแถมยังสามารถที่จะเขียนคำนวณ และพัฒนาโปรแกรมที่ซับซ้อน อาทิเช่น พวกโปรแกรมที่ติดต่อกับอุปกรณ์ Input / Output , งานทางด้านฐานข้อมูล , Application ทางด้าน Internet และยังมีด้านต่างๆมากมาย ดังนั้นจึงพบว่า Visual Basic เป็นโปรแกรมภาษาที่น่าสนใจทีเดียว ที่สามารถรวมเอา Presentation ที่สวยงามแถมง่ายและรวดเร็ว กับ ความสามารถที่สูงในการคำนวณและการพัฒนาโปรแกรม

ที่ยกตัวอย่างการใช้งานของ Control บาง Control เท่านั้นเองยังมี Control ที่ไม่ได้กล่าวถึงอีกมากมายและแต่ละ Control ก็มีเหตุการณ์ที่จะเกิดขึ้นกับตัวมันได้หลายเหตุการณ์เช่นกันที่กล่าวไว้เป็นเพียงการแนะนำเท่านั้นเอง ปัจจุบัน Visual Basic เป็นโปรแกรมที่น่าสนใจและมีหนังสือที่อธิบายไว้ค่อนข้างละเอียดสามารถที่จะทำตามเป็นขั้นตอนได้ แต่หนังสือมีในท้องตลาดนั้นจะกล่าวไว้เพียงส่วนที่เป็นการใช้งาน และพัฒนาด้านสำนักงาน ที่มี Presentation ที่สวยงาม และการคำนวณเบื้องต้น ท่านที่สนใจในงานวิศวกรรมก็ควรที่จะอ่านส่วนที่เป็นพื้นฐานให้เข้าใจ และยังมีอีกแหล่งข้อมูลที่รวบรวมการใช้งานและยังมีโปรแกรมตัวอย่างไว้ค่อนข้างละเอียดก็คือ Library ของ Visual Basic ที่มีขายตามท้องตลาด

ตัวอย่างโปรแกรมติดต่อกับบอร์ดอินเตอร์เฟส เพื่อรับและส่งค่าข้อมูล

เพื่อเปรียบเทียบให้เห็นถึงความแตกต่างระหว่างการใช้โปรแกรมที่เขียนด้วยภาษา C กับการใช้ Visual Basic

```
#include<stdio.h>
#include<dos.h>
#include<conio.h>
#define PORT1 0x2F8
void main(void)
{
int c;
int ch;
clrscr();
outportb(PORT1+1,0x00);
outportb(PORT1+3,0x80);
outportb(PORT1+0,0x0C);
outportb(PORT1+1,0x00);
outportb(PORT1+3,0x03);
outportb(PORT1+2,0xC7);
outportb(PORT1+4,0x0B);
outportb(PORT1+0,0xFF);
delay(5000);
```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

outportb(PORT1+0,0x00);

do{
c=inportb(PORT1+5);
if (c&1)
{
ch=inportb(PORT1);clrscr();
printf("press ESC to quit. input data is %X==: ",ch);
outportb(PORT1,ch);
}
if(kbhit())
{
ch=getch();
outportb(PORT1,0x00);
}
}while(ch!=27);
}

```

• ใช้ Visual Basic

Private Sub Command1_Click()

• เมื่อมีการ Click ที่ปุ่ม ส่งข้อมูล

Command2 = False

• ให้ปุ่มรับข้อมูลไม่ทำงานหรือไม่สามารถ Click

ได้

MSComm1.CommPort = 2

• ติดต่อกับ port 2

MSComm1.Settings = "9600,N,8,1"

• เซ็ตค่าการรับส่งข้อมูลของ port 2

MSComm1.InputLen = 0

MSComm1.PortOpen = True

• เปิดพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MSComm1.Output = Chr(Text1.Text) ' เอาข้อมูลที่เก็บไว้ใน textbox ที่เราป้อนเข้ามาออก
                               ไปที่พอร์ต
MSComm1.PortOpen = False      ' ปิดพอร์ต
End Sub

Private Sub Command2_Click()   ' Click ที่ปุ่มรับข้อมูล
Dim InString As String
Command1 = False
MSComm1.CommPort = 2
MSComm1.Settings = "9600,N,8,1"
MSComm1.InputLen = 0
MSComm1.PortOpen = True
InString = MSComm1.Input
Text2.Text = Asc(InString)
MSComm1.PortOpen = False
End Sub

Private Sub Command3_Click() ' Click ที่ปุ่ม End
End
End Sub

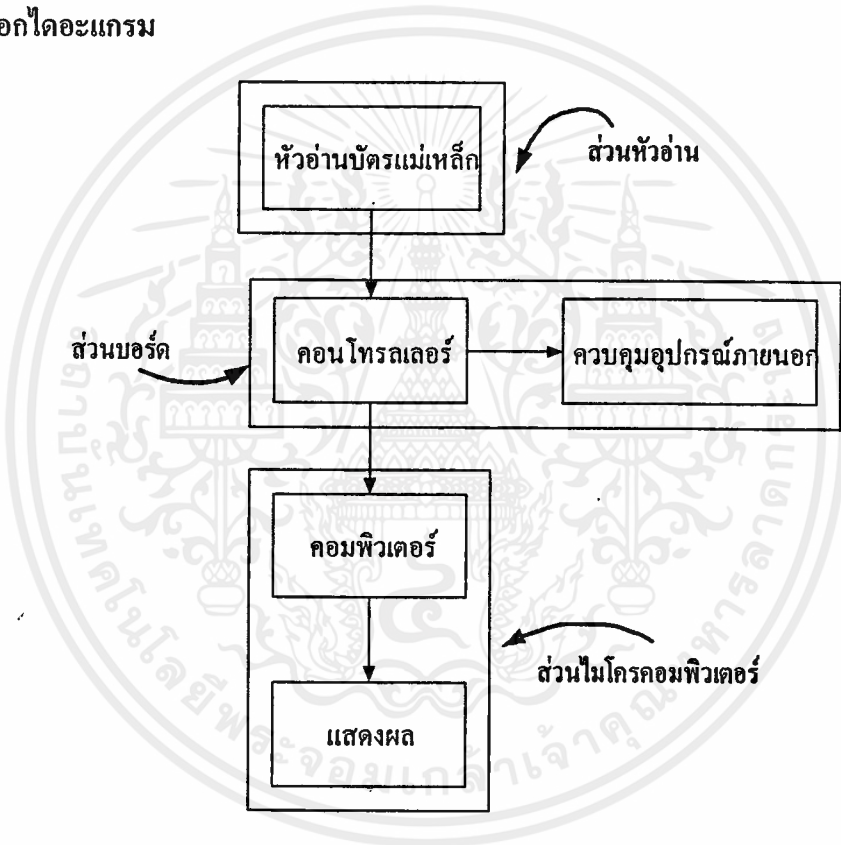
```

จะเห็นว่า Visual Basic จะมีการแสดงผลที่สวยงามและยังง่ายต่อการใช้งานทำให้ User ที่ใช้โปรแกรมสะดวกสบายและตัว Source code ก็ศึกษาทำความเข้าใจง่าย จากที่ได้กล่าวไว้ตั้งแต่ตอนต้นแล้วว่า Visual Basic จะมองเป็น วัตถุ (Object) แต่ละ วัตถุก็แตกต่างกันออกไปแล้วแต่เราจะเลือกใช้ Control อะไร แล้ว วัตถุแต่ละวัตถุก็มีคุณสมบัติหลายอย่างซึ่งเราสามารถกำหนดให้มันได้

บทที่ 3 การออกแบบ

เครื่องบันทึกการเข้าออกโดยบัตรแม่เหล็กมีส่วนสำคัญอยู่ 3 ส่วนใหญ่ๆ คือ ส่วนหัวอ่านบัตรแม่เหล็ก ,ส่วนบอร์ดควบคุมโดยมีคอนโทรลเลอร์เป็นตัวควบคุม และส่วนไมโครคอมพิวเตอร์เป็นตัวแสดงผลและโปรแกรมควบคุม

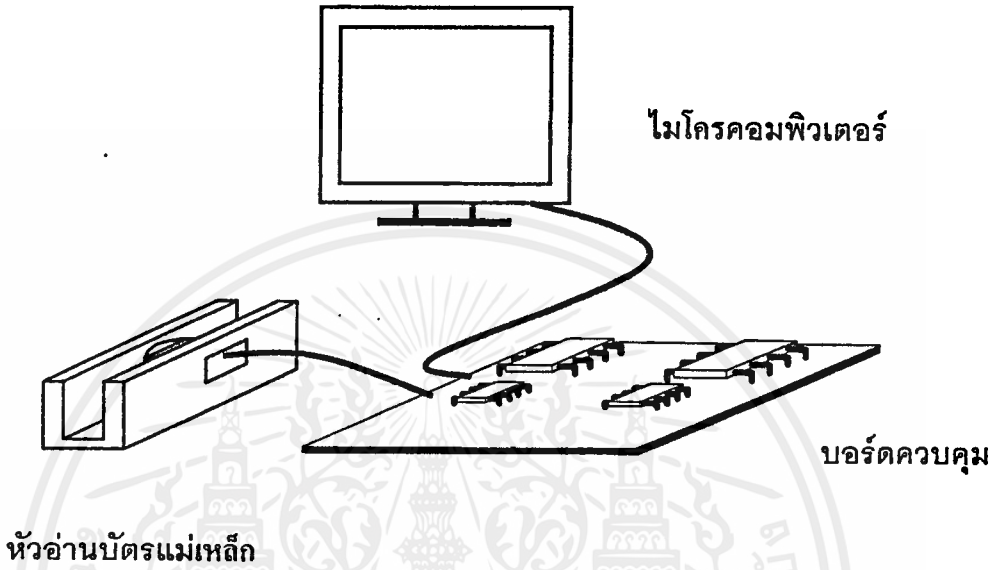
3.1 บล็อกไดอะแกรม



รูปที่ 3.1 บล็อกแสดงหลักการของเครื่องบันทึกการเข้าออกด้วยบัตรแม่เหล็ก

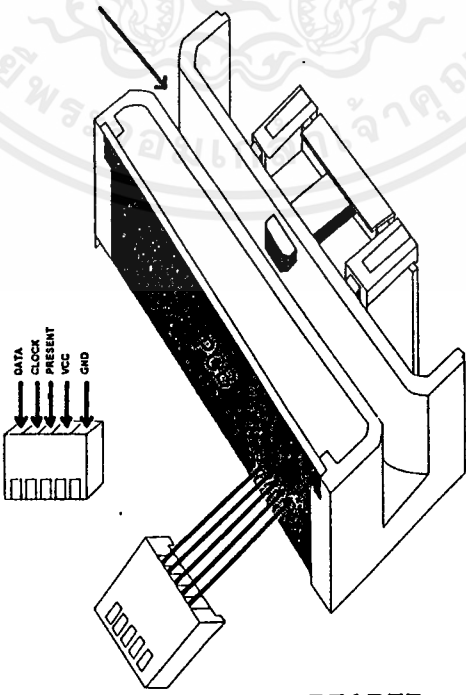
จากรูปที่ 3.1 เป็นบล็อกไดอะแกรมการแสดงผลการทำงานของเครื่องบันทึกการเข้าออกโดยบัตรแม่เหล็กนี้ จะมีส่วนหัวอ่านบัตรแม่เหล็กเป็นตัวอ่านข้อมูลมาจากบัตรและส่งสัญญาณมายังบอร์ดควบคุม โดยส่วนของบอร์ดควบคุมนี้จะมี ไมโครคอนโทรลเลอร์ เบอร์ AT89C2051 เป็นตัวประเมินผล เมื่อข้อมูลที่อ่านมาจากบัตรแม่เหล็กถูกอ่านจากบัตรแถบแม่เหล็กจะถูกตี-ไลต์และจัดเฟรมข้อมูลแล้วส่งไปยังคอมพิวเตอร์ ฉะนั้นหน้าที่หลักอย่างหนึ่งของบอร์ดก็คือการ Interface กับ คอมพิวเตอร์นั่นเอง โดยผ่านทางพอร์ตอนุกรม และบอร์ดควบคุมก็จะรอรับข้อมูลจากเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์เพื่อส่งข้อมูลออกพอร์ตเพื่อไปควบคุมอุปกรณ์ต่อไป RS-232 ส่วนคอมพิวเตอร์ก็จะ
รอรับข้อมูลแล้วนำมาตรวจสอบกับ Database ว่ามีข้อมูลอยู่ใน Database หรือไม่ ถ้าตรงก็จะส่งข้อมูล
ออกไปที่ บอร์ดเพื่อที่จะจับอุปกรณ์จับให้ประตูเปิดต่อไป



รูปที่ 3.2 แสดงการต่อบอร์ดกับคอมพิวเตอร์

3.2 ส่วนเครื่องอ่านบัตรแม่เหล็ก (Magnetic Card Reader)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Magnetic Card หรือ บัตรแถบแม่เหล็ก ที่เรารู้จักดีในรูปแบบของบัตรเงินสด บัตร ATM บัตรเครดิต บัตร VISA และบัตรอื่นๆ อีกมากมาย ฯลฯ เป็นบัตรแม่เหล็กซึ่งบนตัวบัตรเองจะบันทึกข้อมูลและรายละเอียดต่างๆของบัตรไว้ในรูปแบบของเส้นแรงแม่เหล็กภายในส่วนที่เป็นแถบแม่เหล็กในการ์ด โดยที่แถบแม่เหล็กที่อยู่ในการ์ด ซึ่งเรียกว่า “Track “ นั้น ตามปกติแล้ว จะถูกแบ่งออกเป็น 3 ส่วนด้วยกันซึ่งในแต่ละส่วนจะถูกเก็บข้อมูลซึ่งมีความหนาแน่นและลักษณะของข้อมูลที่มีความแตกต่างกันดังนี้คือ

แถบแม่เหล็กที่	ความหนาแน่นในการบันทึก	การเข้ารหัสข้อมูล	จำนวนตัวอักษรสูงสุด	ลักษณะของข้อมูลที่เก็บในบัตร
Track 1	210 Byte/Inch	ALPHA	79	ชื่อเจ้าของบัตรและหมายเลข
Track 2	75 Byte/Inch	BCD	40	หมายเลขและวันหมดอายุ
Track 3	210 Byte/Inch	BCD	107	หมายเลขบัตรและรหัสพิเศษ

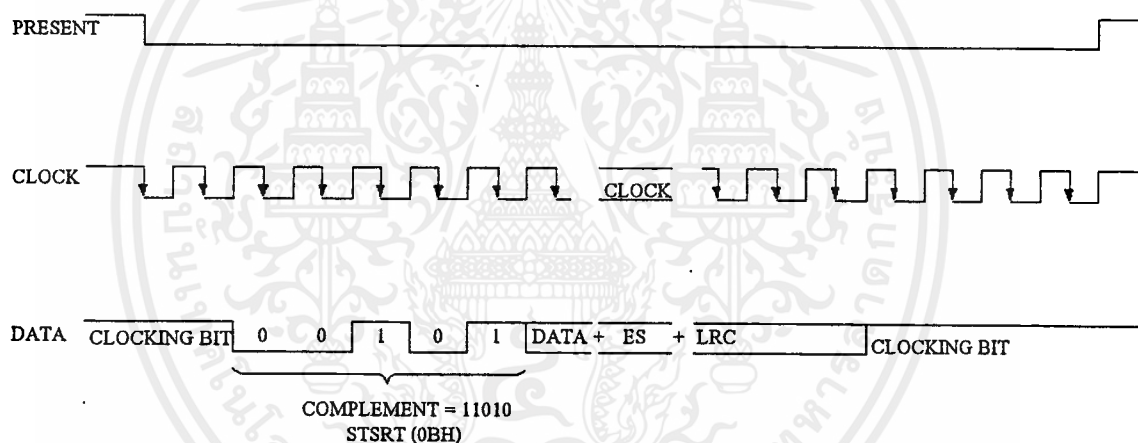
โดยเครื่องอ่านแถบแม่เหล็กที่ใช้ในโครงงานนี้ จะอ่านได้ เฉพาะข้อมูล ที่บันทึกที่ความละเอียด 75 Byte/Inch คือเฉพาะในส่วนของแถบแม่เหล็กแถบที่ 2 เท่านั้น โดยมีสัญญาณที่ใช้ในการเชื่อมต่อเพื่ออ่านข้อมูลจากเครื่องอ่านแถบแม่เหล็กทั้งหมด 5 สัญญาณ โดยเป็นสัญญาณเอาต์พุตที่ส่งออกมาจากเครื่องอ่านแถบแม่เหล็กทั้งหมด และมีขาไฟเลี้ยงวงจรที่ต้องต่อให้เครื่องอ่านอีกด้วย ดังนี้คือ

1. ไฟเลี้ยงวงจร (+VCC) ซึ่งต้องมีค่าเป็น +5 VDC ซึ่งต่อจากภายนอกให้เครื่องอ่านแถบแม่เหล็ก
2. Ground เป็นจุดอ้างอิงของสัญญาณต่างๆระหว่างเครื่องอ่านแถบแม่เหล็กกับบอร์ดไมโครคอนโทรลเลอร์
3. สัญญาณข้อมูล (Data) เป็นสัญญาณข้อมูลเอาต์พุตส่งออกจากเครื่องอ่านแถบแม่เหล็ก ซึ่งเป็นสัญญาณของข้อมูลที่อ่านได้จากบัตรแถบแม่เหล็ก โดยการอ่านของข้อมูลเป็นแบบอนุกรมทีละบิต โดยต้องให้สัมพันธ์สอดคล้องกับสัญญาณนาฬิกา ของบัตรด้วย ซึ่งสัญญาณข้อมูลที่อ่านได้นี้จะกลับสภาวะกับสัญญาณจริง คือมีสภาวะตรงข้ามกันอยู่ ดังนั้นเมื่ออ่านสัญญาณจากเครื่องอ่านแถบแม่เหล็ก ได้แล้วจะต้องกลับสภาวะของสัญญาณเสียก่อนจึงจะได้ข้อมูลที่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. สัญญาณนาฬิกา (Clock) เป็นสัญญาณ Clock เอาต์พุตส่งออกมาจากเครื่องอ่านแถบแม่เหล็ก ใช้เป็นสัญญาณอ้างอิงในการอ่านข้อมูลจากบิตร์ โดยการอ่านสัญญาณข้อมูลแต่ละบิตนั้น ต้องอ่านในขณะที่สัญญาณนาฬิกาเป็นขอบขาลง (Falling Edge) เสมอ ซึ่งสัญญาณข้อมูลที่อ่านได้นั้นจะเริ่มต้นจากบิตที่ต่ำสุด (LSB) ก่อนเป็นอันดับแรก

5. สัญญาณ Present เป็นสัญญาณสถานะเอาต์พุตส่งออกมาจากเครื่องอ่านแถบแม่เหล็กทำงานที่ลอจิก “0” เมื่อสัญญาณนี้เกิด จะบ่งบอกให้ทราบว่าเครื่องอ่านแถบแม่เหล็กเริ่มต้นส่งข้อมูลออกมาหรือมีการนำบิตร์แถบแม่เหล็กไปรูดผ่านหัวอ่านสัญญาณของเครื่อง เมื่อสัญญาณนี้หมดลง (กลับเป็น ลอจิก “1”) อีกครั้งหนึ่งแสดง ว่าข้อมูลจากการอ่านในครั้งนั้น ถูกส่งออกไปหมดสิ้นเรียบร้อยแล้ว ซึ่งเราใช้ประโยชน์จากสัญญาณนี้ ส่งเป็นสัญญาณอินเตอร์รัปต์ ให้ CPU เพื่อทำการอ่านข้อมูล



รูปที่ 3.4 ภาพแสดงความสัมพันธ์ของสัญญาณ DATA ,CLOCK และ PRESENT ของเครื่องอ่านแถบแม่เหล็ก

การอ่านข้อมูลจากบิตร์แถบแม่เหล็กของเครื่อง

การใช้เครื่องอ่านบิตร์แถบแม่เหล็กเพื่อให้ได้ข้อมูลจากบิตร์แถบแม่เหล็กนั้น จะใช้วิธีการนำแผ่นบิตร์ส่วนที่เป็นแถบแม่เหล็กไป Slide ผ่านหัวอ่านของเครื่องอ่านไปตามทิศทางและตำแหน่งที่กำหนดไว้เท่านั้น โดยให้สังเกตตำแหน่งของหัวอ่านกับส่วนที่เป็นแถบแม่เหล็กของบิตร์ต้องอยู่ในตำแหน่งเดียวกัน และเนื่องจากเครื่องอ่านแถบแม่เหล็กสามารถอ่านข้อมูลจากแถบแม่เหล็กได้เฉพาะข้อมูลของ Track 2 เท่านั้น โดย Format ของข้อมูลในแถบแม่เหล็กส่วนที่เป็น Track 2 นี้ได้

สูงสุดไม่เกิน 40 ตัวอักษร โดยนับรวมรหัสควบคุมและรหัสตรวจสอบด้วย ซึ่งมีรายละเอียดดังนี้คือ

SS	PAN	FS	Addition Data	ES	LRC
----	-----	----	---------------	----	-----

SS = รหัสข้อมูลแสดงการเริ่มต้นของข้อมูลในบัตร (Start Sentinel) มีรหัสข้อมูลเป็น 0BH(;

PAN= ข้อมูลแสดงหมายเลขของบัตร (มีจำนวนข้อมูลส่วนนี้ไม่เกิน 19 หลัก)

FS = รหัสข้อมูลแสดงการแบ่งแยกข้อมูล (Field Separator) มีรหัสข้อมูลเป็น 0DH (=)

Additional Data = เป็นข้อมูลเพิ่มเติมต่างๆของบัตรเช่น วัน/เดือน/ปี ที่ออกบัตรและหมดอายุ

ES = รหัสข้อมูลแสดงการสิ้นสุดของข้อมูลในบัตร (End Sentinel) มีรหัสเป็น 0FH (?)

LRC = ข้อมูลตรวจสอบความผิดพลาด (Longitudinal Redundancy Check)

0000000000	SS	Data,Data,Data.....,Data	ES	LRC	0000000000
------------	----	--------------------------	----	-----	------------

ลักษณะการจัดเรียงข้อมูลที่เก็บไว้ในบัตรแม่เหล็กที่ในส่วนที่เป็น Track 2 ซึ่งจัดเรียงความสำคัญจากซ้ายไปขวาแบบอนุกรมทีละบิต โดยสังเกตเห็นว่าข้อมูลที่เป็น ลอจิก 0 หรือที่เรียกว่า "Clocking bit" นำหน้าและปิดท้ายข้อมูลจริงอยู่ ซึ่งลักษณะการเก็บข้อมูลของ Track 2 จะเก็บข้อมูลด้วยการเข้ารหัสแบบ Modulo 5 ซึ่งแต่ละชุดข้อมูลจะประกอบไปด้วยข้อมูลที่ละ 5บิต โดย 4 บิตแรก (D0-D3) เป็นรหัสข้อมูลแบบ BCD ส่วนบิตที่ 5 (D4) เป็นพาริตีบิตแบบคี่ เพื่อใช้ตรวจสอบความถูกต้องของข้อมูลแต่ละbyte (5 bit) ที่อ่านได้ ซึ่งหากว่าค่าของพาริตีที่ผิดพลาดแสดงว่าการอ่านข้อมูลนั้นล้มเหลว โดยจะเริ่มอ่านข้อมูลเมื่อสัญญาณ Present เริ่มเป็น 0 ก่อน และทำการอ่านข้อมูลในทุกๆขอบขาลงของสัญญาณนาฬิกาเสมอ ซึ่งข้อมูลในส่วนก่อนเริ่มต้นและหลังจากสิ้นสุดของการอ่านนี้จะมีค่าเป็น 0 นำหน้าและปิดท้ายข้อมูลจริงอยู่ ซึ่งเรียกว่า "Clocking Bit" ซึ่งข้อมูลในส่วนนี้เราไม่ต้องสนใจแต่ให้ตรวจสอบและรองจนกว่าจะเริ่มเป็นข้อมูล Start บิต (0BH หรือ 11010 ในที่นี้เรียงลำดับความสำคัญจากซ้ายไปขวา ซึ่งบิตเริ่มต้นของรหัส Start หรือ 0BH ต้องเริ่มต้นด้วย 1 เป็น บิตแรกเสมอ) ดังนั้นในการอ่านเราต้องรองจนกว่าจะพบสัญญาณข้อมูลมีค่าเป็น 1 (สัญญาณ Data= "0" เพราะกลับสภาวะกันอยู่) จึงเริ่มเก็บข้อมูลชุดทีละ 5 บิต ไปเรื่อยๆ จนถึงรหัสจบ (0FH) เมื่อพบ รหัสจบแล้วจะมีข้อมูลตามมามาก 1 byte ซึ่งเป็นข้อมูลสำหรับตรวจสอบความผิดพลาดของการข้อมูลทั้งหมดของ Track 2 เรียกว่า LRC ซึ่งค่าของ LRC สามารถหาได้จากการนำเอาข้อมูลในแต่ละ byte (ไม่คิดพาริตีบิต) ตั้งแต่เริ่มต้นจนสิ้นสุดมาทำการ XOR กัน โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ครั้งแรกนำเอา byte แรก (0BH) ทำการ XOR กับ 0 แล้วนำผลลัพธ์ที่ได้ XOR กับ byte ถัดไปเรื่อยๆ ตามลำดับ ซึ่งหากผลลัพธ์ที่ได้สุดท้ายไม่เท่ากับค่าของ LRC ที่อ่านมาได้แสดงว่าการอ่านข้อมูลทั้งหมดคล้มเหลวสำหรับข้อมูลของรหัส BCD ที่ใช้สำหรับเครื่องอ่านแถบแม่เหล็กนี้จะเป็นข้อมูลชุดละ 5 บิต โดยเป็นข้อมูลจริง 4 บิต และเป็นรหัสตรวจสอบพาริตี้อีก 1 บิต ซึ่งมีจำนวนทั้งหมด 16 อักขระ ดังนี้คือ

Parity	D3	D2	D1	D0	Character	Function
1	0	0	0	0	0 (0H)	Data
0	0	0	0	1	1(1H)	Data
0	0	0	1	0	2(2H)	Data
1	0	0	1	1	3(3H)	Data
0	0	1	0	0	4(4H)	Data
1	0	1	0	1	5(5H)	Data
1	0	1	1	0	6(6H)	Data
0	0	1	1	1	7(7H)	Data
0	1	0	0	0	8(8H)	Data
1	1	0	0	1	9(9H)	Data
1	1	0	1	0	:(AH)	Control
0	1	0	1	1	;(BH)	Start Sentinel
1	1	1	0	0	<(CH)	Control
0	1	1	0	1	=(DH)	Field Separator
0	1	1	1	0	>(EH)	Control
1	1	1	1	1	?(FH)	End Sentinel

ตารางแสดงข้อมูลของรหัส BCD สำหรับเครื่องอ่านแถบแม่เหล็ก

3.3 ส่วนบอร์ดควบคุม

ใช้ชิพเบอร์ 89C2051 ของ บริษัท ATMEL เป็นหลัก ซึ่งมีหน่วยความจำภายในแบบ FLASH MEMORY ความจุ 2K Byte สามารถโปรแกรมได้ถึง 1000ครั้งแต่อย่างไรก็ตามเราสามารถเลือกใช้ชิพเบอร์ 89C1051(1K) หรือ 89C4051(4K) ก็ได้เพราะโครงสร้างโดยหลักๆ แล้วจะเหมือนกันจะแตกต่างกันที่ตรงของหน่วยความจำภายในของตัวชิพเท่านั้น และคุณสมบัติจะเป็นไปตามโครงสร้างของบอร์ดนั้นๆ

คุณสมบัติของบอร์ด

CPU	89C2051
CLOCK	11.0592 MHz
MEMORY	2K Byte (MCU FLASH MEMORY)
PORT	15 BIT I/O (MCU)
CONNECTER	3PIN RS-232 (MAX232) 2 PIN RS-485 (75176) 2 PIN 5VDC
ONB-OARD	1 POWER LED 1 USER LED 2 TEST POINT FOR CLIP (VCC, GND)
POWER	5VDC CURRENT 20 mA

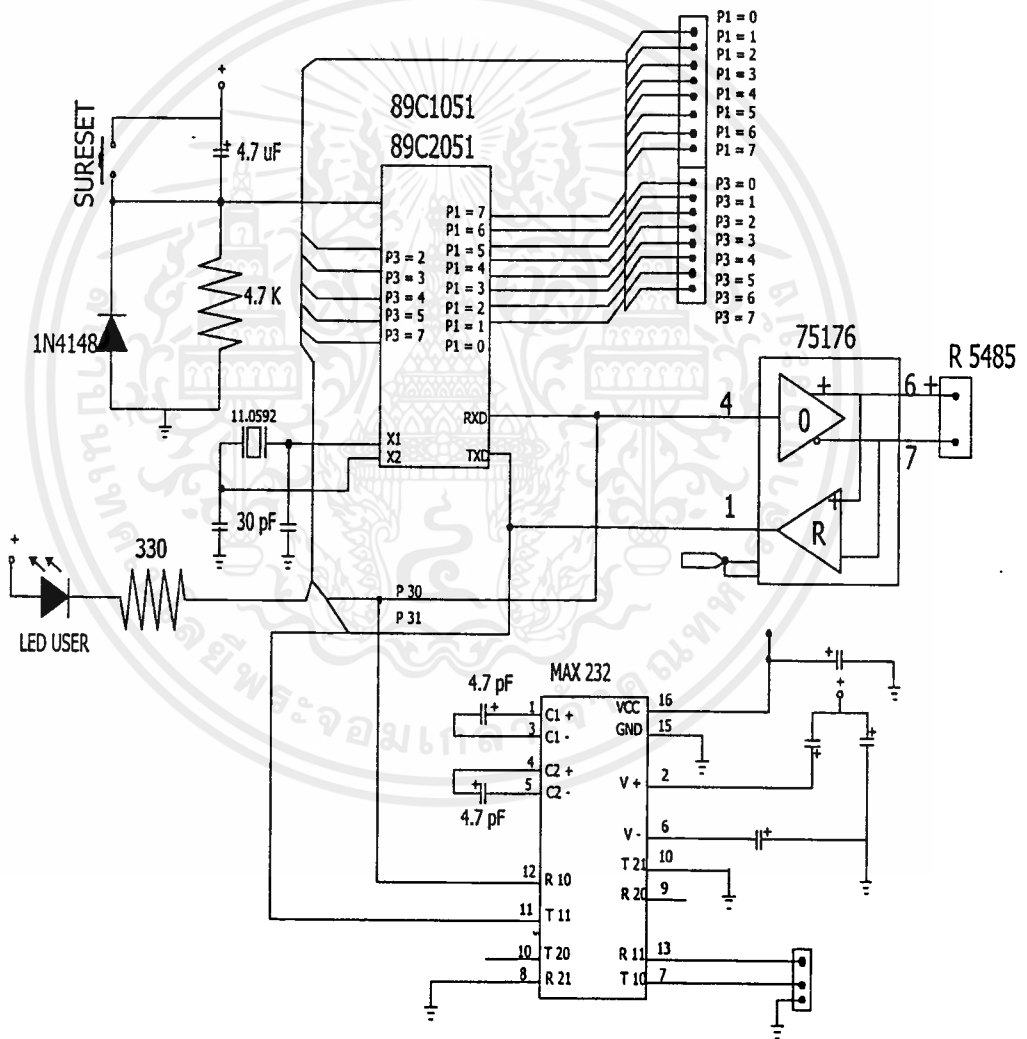
SERIAL PORT

พอร์ตสื่อสารอนุกรมของบอร์ด สามารถเลือกใช้งานได้ 2แบบ คือ RS-232 และ RS-485 โดยถ้าต้องการใช้งานเป็น RS-232 ก็ให้เสียบชิพเบอร์ MAX232 และใช้งานที่ขั้วต่อ 3 PIN และถ้าต้องการใช้งานเป็น RS-485 ก็ให้เสียบเบอร์ 75176 และใช้งานที่ ขั้วต่อแบบ 2 PIN โดยในกรณีของ RS-485 นี้ จะใช้ขา T0 ของ CPU เป็นตัวควบคุมทิศทางการรับส่ง โดยถ้า T0=0 จะเป็นการรับข้อมูล และ T0=1 จะเป็นการส่งข้อมูล โดยได้กล่าวไว้ในส่วนของบทที่ 2 อย่างละเอียดพอสมควรแล้ว RS-485 จะมีแนวทางการใช้คล้ายกับ RS-232 แต่ต่างกันตรงที่ระดับแรงดันไฟ คือจะใช้ที่ 5 V โดยจะเป็น บวกและลบแล้วแต่ค่าของบิต 0 ,1 ซึ่งยังผลให้ค่าลบล้างกันเอง ทำให้ลดสัญญาณรบกวนได้เป็นอย่างดี และ RS-485 ยังจะมีการสื่อสารในแบบ Half Duplex โดยจะต้องรับรับและส่งสัญญาณคนละจังหวะ ไม่สามารถสวนทางกันแบบ RS-232 ได้ จึงทำให้ต้องมีบิตเพื่อควบคุมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รับส่งนั่นเองและด้วยคุณสมบัตินี้เองจึงทำให้ RS-485 สามารถต่อเป็นระบบ Network ก็คือการทำโปรแกรมควบคุม (PROTOCOL) นั่นเองโดยจะต้องควบคุมทิศทางและการสื่อสารข้อมูลแต่ละจุดในระบบให้ทำงานได้เป็นระเบียบและมีประสิทธิภาพ

บนบอร์ดจะมี LED สำหรับผู้ใช้งานได้อย่างอิสระทั้งอาจจะใช้แสดงสถานะต่างๆนี้ จะต่อผ่าน P1.7 ของ CPU การใช้งานจะเป็นแบบ ACTIVE LOW คือถ้ากำหนดให้เป็น 1 จะดับ และจะติดเมื่อได้รับการกำหนดให้เป็น 0

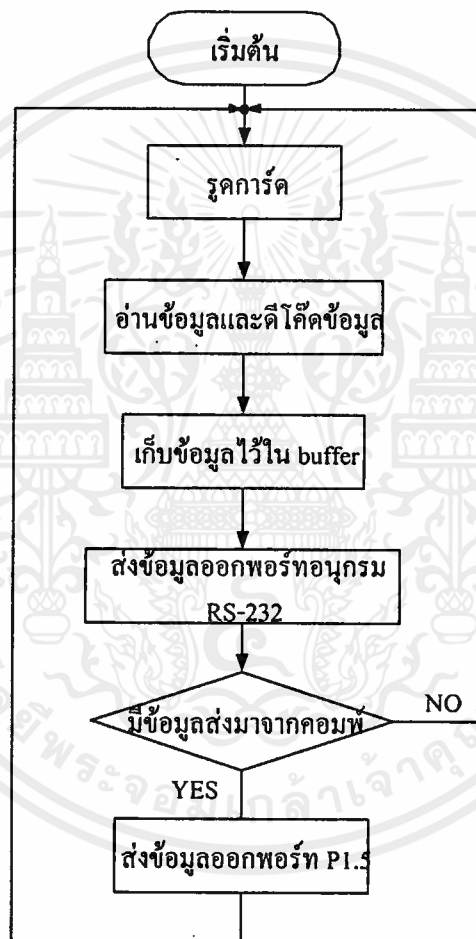


รูปที่ 3.5 วงจรส่วนของบอร์ดไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาโปรแกรมบนบอร์ดควบคุม

Software ที่ใช้กับบอร์ดควบคุมพัฒนาโดยภาษา Assembly ที่เป็นชุดคำสั่งของ ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ซึ่งสามารถใช้ได้กับ เบอร์ AT89C2051ที่เราใช้ในบอร์ด Software ส่วนนี้ จะต้องทำหน้าที่ในการนำข้อมูลที่อ่านมาได้จากหัวอ่านนำมาตีโค้ดแล้วจัดเฟรมข้อมูล (โปรโตคอล) แล้วส่งออกทาง RS-232 ซึ่งมี Flowchart



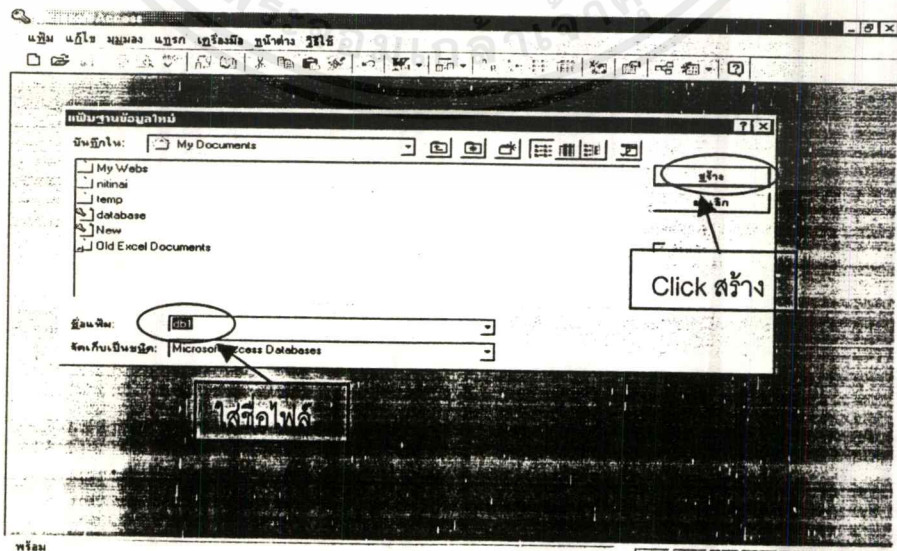
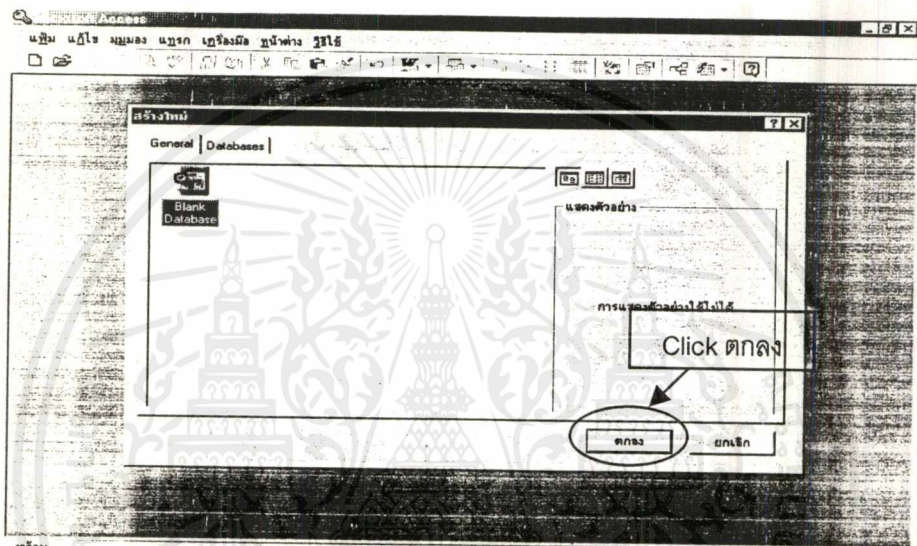
3.4 ส่วนไมโครคอมพิวเตอร์หรือโปรแกรม

ส่วนนี้จะเป็นส่วน Presentation และยังเป็นส่วนควบคุม ต้องติดต่อกับ Database และต้องยังติดต่อกับอุปกรณ์ภายนอกที่ต้องติดต่อกันทางพอร์ทเป็นแบบอนุกรม โปรแกรมที่เลือกใช้นี้จะสมควรจะเป็น โปรแกรมที่พัฒนาบน Window นั้นเพราะว่า Window เป็นระบบปฏิบัติการที่นิยมใช้กันอย่างแพร่หลาย จึงเลือกใช้ Visual Basic

ในส่วนของ Database ใช้ Access ในการสร้างซึ่งมีวิธีการสร้างที่ดังจะยกตัวอย่างการสร้างอย่างง่ายๆดังนี้

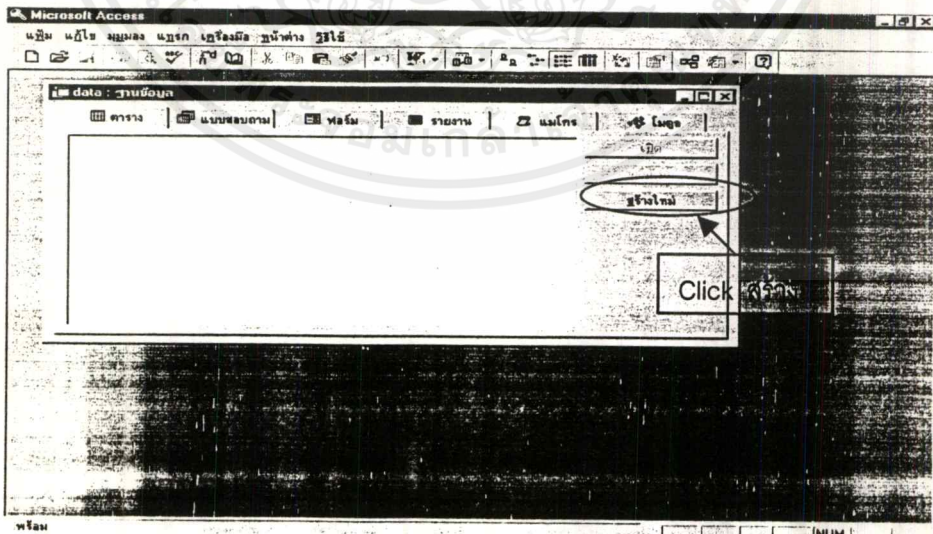
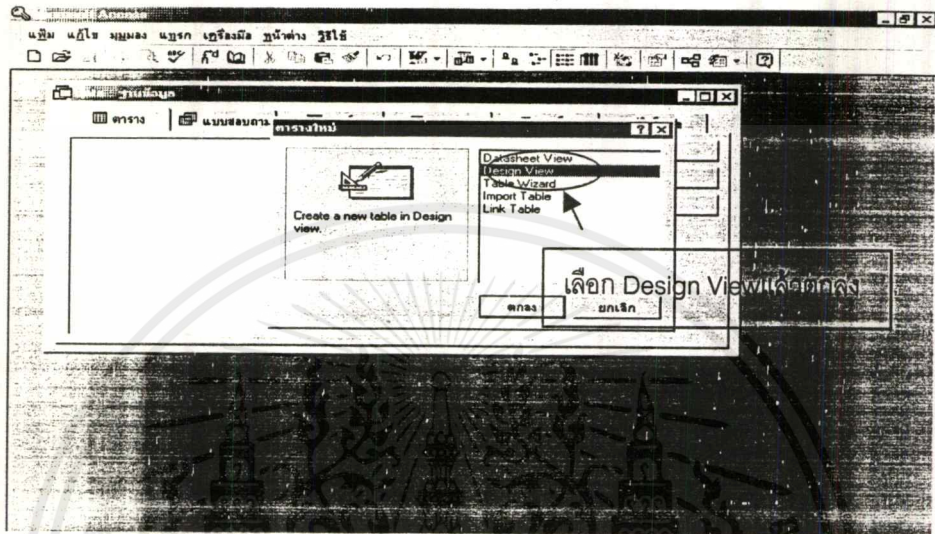
การสร้าง Database

ในการสร้าง Database ก็สามารถสร้างได้หลายวิธีแต่ในที่นี้จะกล่าวถึง Microsoft Access ซึ่งง่ายต่อการออกแบบและการสร้างการสร้าง Database โดยใช้ Access นี้ก็จะเก็บเป็น filed, record

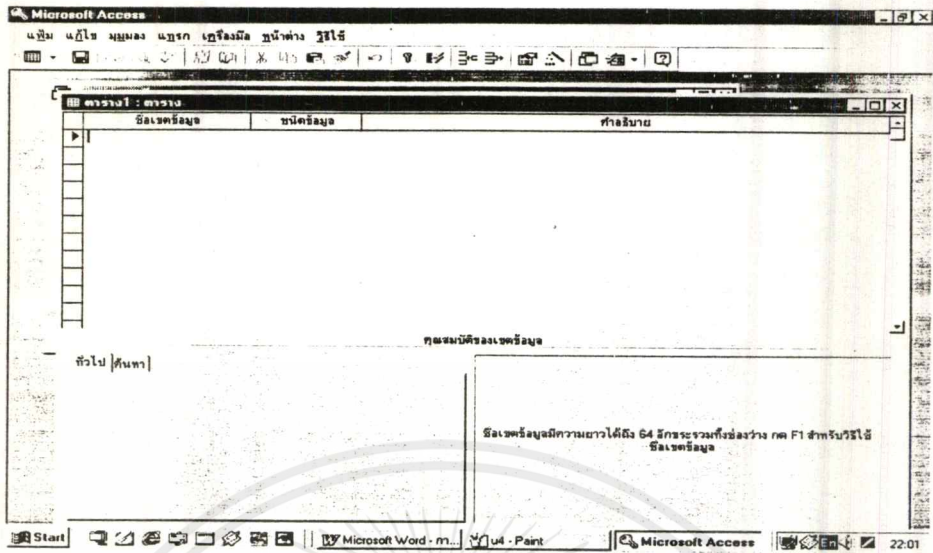


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มโดยการเปิด เข้าไปที่สร้างฐานข้อมูลก็จะเปิดเข้ามาเหมือนรูปข้างบน Click ตกลงแล้วใส่ชื่อแฟ้มและไดเรกทอรีแล้วเลือกสร้างแล้วเลือกสร้างใหม่ก็จะขึ้นหน้าจอสำหรับการสร้างตาราง เราสร้างตารางใหม่โดยเลือก Design view ตามรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ชื่อเขตข้อมูล	ชนิดข้อมูล
Index	Text
Name	Text
LastName	Text
Address	Text

บันทึกเป็น

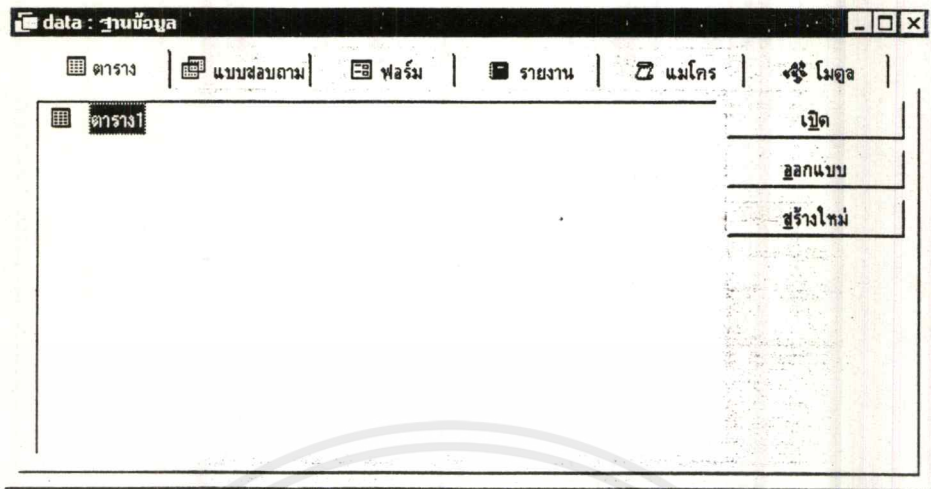
ชื่อตาราง: ตาราง1

ตกลง

ยกเลิก

ใส่ record และกำหนดชนิดของข้อมูลเป็นแบบ text (default) แล้วก็ทำการปิดหน้าต่างของการสร้างตารางแล้วใส่ชื่อ ตาราง ก็จะได้ตาราง จากตัวอย่างนี้ตั้งชื่อตามที่ Default ซึ่งก็คือ “ตาราง1” เมื่อเราตอบตกลงก็จะได้ ตารางใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



พอมาถึงตอนนี้ก็จะสามารถแก้ไขหรือเพิ่มข้อมูลให้แต่ละ Record ได้ดังตัวอย่าง จากที่กล่าวมาเป็นการสร้าง ตารางอย่างง่ายและเป็นพื้นฐานเบื้องต้นของการสร้าง Database ที่จะสามารถนำมาใช้งานเพื่อเก็บข้อมูลของลูกค้าหรือสมาชิก ในโครงการนี้ได้นำการสร้าง Database โดยวิธีการแบบ นี้มาใช้ร่วมกับการเขียนโปรแกรมโดยเลือกใช้ Visual Basic ในการพัฒนา โปรแกรมร่วมกันเพื่อเป็นตัวเก็บข้อมูลของผู้ผ่านเข้าออก

Microsoft Access - [table1 : ตาราง]

เพิ่ม แก้ไข มุมมอง แทรก รูปแบบ ระบุเงื่อนไข เครื่องมือ หน้าต่าง วิธีใช้

ID	name	lastname	Time_In	check
:0000021591615580219=12990002393?	MR Kulachat	detsiri	10/11/00 6:24:30 PM	0
:20000060004371119026=02000099991?	MR Jittipun	Mukda	10/11/00 6:24:38 PM	0
000012569888847556544=1255445450?	Mr Lacha	Nara	11/11/00 5:30:40 AM	0
*				

ที่กล่าวมาทั้งหมดนั้นเป็นวิธีการสร้าง Database ซึ่งสามารถสร้างขึ้นมาใช้ได้และข้อมูลแบบละเอียดผู้อ่านสามารถหาอ่านได้ตามหนังสือ Microsoft Access ทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาโปรแกรมโดย Visual Basic

ในการดำเนินการสร้างอุปกรณ์ที่เป็น Hardware เราคงตอบไม่ได้ว่าเราสร้างได้ถูกต้องหรือเปล่านั้นกว่าเราจะมี Software มารองรับข้อมูลเพื่อทดสอบดูว่า Hardware ที่เราสร้างนั้นมันถูกต้องตามที่ต้องการหรือแต่ก่อนที่ Software จะถูกเขียนขึ้นมาเราต้องรู้ว่า Hardware เราที่ถูกต้องแล้วมันต้องส่งอะไรออกมา ที่กล่าวมาเป็นหลักการเบื้องต้นของการออกแบบทาง Software อย่างที่กล่าวไว้ตอนต้นแล้วว่า โครงการนี้ได้ใช้ Visual Basic ในการพัฒนาโปรแกรม จึงจะขอยกตัวอย่างที่เป็นโปรแกรมในการค้นหาข้อมูลจาก Database ซึ่งเป็นหนึ่งฟังก์ชันในโครงการนี้แต่เป็นเพียงหน้าที่ที่เหมือนกันแต่การประยุกต์ใช้งานจะต่างกัน

โปรแกรมนี้เป็นโปรแกรมตัวอย่างที่สามารถค้นหาข้อมูลจาก Database จะให้เราใส่เบอร์แล้วก็จะทำการหา ชื่อและที่อยู่ ที่ตรงกับเบอร์ที่เราใส่เข้าไป โดย Database เขียนโดย Access และมีรูปแบบดังนี้

TelNumber	Name	Address	Pcode
7391931	กำจัด ดงบุญ ดต	357 ม จินดาภิเวศน์	
7391974	กาญจนา ระลึกมูล	70/8 ม ชลพฤกษา	
7391995	กวี ศวเสนี	240 ซ่อนนุช-ลาดกระบัง	
7392500	ไกรลักษณ์ ศิริอนันต์	111/39 ซ ไปรษณีย์เจ้าคุณทหารลาดกระบัง	
7392553	กิตติชัย ตติยสกุลเลิศ	40/96 ซ เปรมฤทัย	
7392572	กิตติกร พูลเกษม สดท	171/20 หมู่ 3 ซ่อนนุช-ลาดกระบัง	
7393044	กมล ไตทิพย์	100/74 หมู่ 3 ชุมทอง-ลำด้อยดิ่ง	
7393072	เกียงศักดิ์ กิจวิวัฒนาชัย	100/34 หมู่ 3 ชุมทอง-ลำด้อยดิ่ง	

เริ่มต้นด้วยการสร้าง Form ตามรูปและใส่ Properties ตามรูปเมื่อได้ตามรูปแล้วต่อไปเราจะใส่เหตุการณ์ (Even) ให้แต่ละ Control

Command1 เป็นปุ่มกด ใส่ชื่อ “Command1” ตามที่ default เปลี่ยน Caption เป็น “ค้นหา”
Command1_Click() ใส่ Even “Click” โดย ดับเบิลคลิก ที่ Command1 เขียน Code ตาม

ข้างล่าง

```
Private Sub Command1_Click()
    Dim a As String
    Dim dbsperson As Database
    Dim rstperson As Recordset
    Dim strperson As String

    Set dbsperson = OpenDatabase("C:\joop_c\copy_a\dbtel.mdb")
    Set rstperson = dbsperson.OpenRecordset("Address", dbOpenDynaset)
    strperson = "TelNumber = " & Text1.Text & " "

    With rstperson
        .MoveLast
        .FindFirst strperson
        Text2.Text = rstperson("Name")
        Text3.Text = rstperson("Address")
        Text4.Text = rstperson("TelNumber")
    End With
End Sub
```

Command2 ใส่ Caption = “End” ดับเบิลคลิก ที่ Command2 แล้วใส่ Code ตามข้างล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Command2_Click()
```

```
End
```

```
End Sub
```

สิ่งที่ต้องระวังในการเขียนคือส่วนติดต่อกับค่าตำแหน่งที่อยู่ว่าให้ถูกต้องและการเชื่อม Record ก็จะต้องตรงกับชื่อของแต่ละ Record ด้วย เมื่อเขียนทุกอย่างเสร็จแล้วลองรันดูแล้วใส่เบอร์ดีจะทำการหาและแสดงข้อมูลที่ตรงกันกับที่ป้อนเข้าไป

The screenshot shows a Windows application window titled "Form1". It contains a search form with the following elements:

- ชื่อ (Name):** กุระชาติ เตชศิริ
- ที่อยู่ (Address):** 257 หมู่ 1 อ่อนนุช ลาดกระบัง กรุงเทพฯ
- เบอร์โทร (Phone Number):** 7390278
- ใส่เบอร์ที่จะค้นหา (Enter number to search):** 7390278
- ค้นหา (Search):** A button to execute the search.
- Data1:** A label with navigation arrows (back, forward) and the text "Data1".
- End:** A button at the bottom of the form.

โปรแกรมทดสอบการอ่านข้อมูลจากพอร์ตอนุกรม

จะใช้เป็นโปรแกรม Simulator การส่งข้อมูลของพอร์ตอนุกรมใช้สำหรับการทดสอบการรับข้อมูลจากบอร์ดที่ส่งผ่านพอร์ตอนุกรม ซึ่งถูกออกแบบตามวัตถุประสงค์ คือเมื่อรับข้อมูลมาแล้ว จะทดสอบการส่งข้อมูลออกไปที่พอร์ตเพื่อให้บอร์ดทำงานครบรูปและบอร์ดจะได้ทำงานได้ถูกต้อง

```

Private Sub Command1_Click()
    Command1.Caption = "ok"
End Sub
Private Sub Command2_Click()
    Command2.Caption = "Clear"
End Sub
Private Sub Command3_Click()
    Command3.Caption = "ok"
End Sub
Private Sub Command4_Click()
    Command4.Caption = "ok"
End Sub
Private Sub Command5_Click()
    MSComm1.Output = Text2.Text
End Sub

DoEvents
buffer$ = buffer$ & MSComm1.Input
Text1.Text = buffer$
Me.Show
If Command2.Caption = "Clear" Then
    buffer$ = ""
    Command2.Caption = "กดเพื่อClear"
End If
If Command3.Caption = "ok" Then
    MSComm1.Output = Chr$(0)
    MSComm1.Output = Chr$(0)
    Command3.Caption = "ส่ง 0"
End If
If Command4.Caption = "ok" Then
    MSComm1.Output = Chr$(1)
    MSComm1.Output = Chr$(1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Form_Load()
    Dim Instring As String
    MSComm1.CommPort = 1
    MSComm1.Settings = "9600,N,8,1"
    MSComm1.InputLen = 0
    MSComm1.PortOpen = True
    Do
        Command4.Caption = "ส่ง 1"
    End If
    Loop Until Command1.Caption = "ok"
    MSComm1.PortOpen = False
End Sub

```

เมื่อเรารันโปรแกรมแล้วเมื่อทำการรูดบัตรถ้ามีข้อมูลก็จะแสดงว่ามีข้อความอะไรบ้าง แล้วทดสอบ ส่งข้อมูลออกที่พอร์ตเลือกที่จะส่ง 0 หรือ 1 ก็ได้แล้วดูที่บอร์ดว่าทำงานถูกต้องหรือไม่



บทที่ 4

ผลการทดลอง

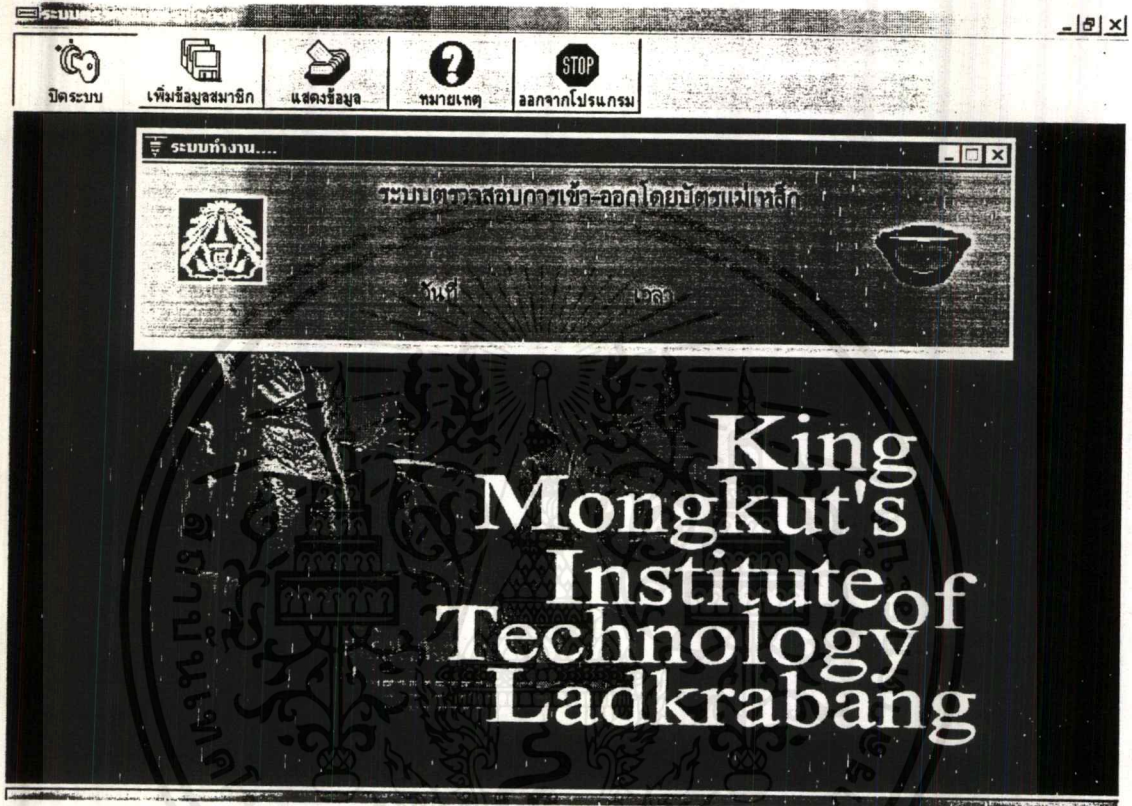
จากบทที่ผ่านมาเราได้กล่าวแต่เพียงการออกแบบโปรแกรมแต่ที่เป็นทฤษฎีและพื้นฐาน
 ทั้งนี้ แต่สำหรับบทนี้เป็นการรันโปรแกรมที่ใช้สำหรับ รับข้อมูลที่ได้มาจาก พอร์ตอนุกรมที่เรา
 ต่อมาจากบอร์ด เมื่อเราทำการรูดการ์ดผ่านตัวรูดการ์ดตัวรูดการ์ดก็จะส่งสัญญาณข้อมูลมาที่บอร์ด
 ควบคุมแล้วบอร์ดควบคุมก็จะทำการ Decode ว่าเป็นข้อมูลอะไร และก็จะทำการเฟรมข้อมูลแล้วก็
 ส่งออกทางพอร์ตอนุกรม การออกแบบ Software เราจึงได้ออกแบบตาม Hardware ซึ่งเรามี
 โปรแกรมที่เขียนขึ้นมาเป็น Simulate ทดสอบดูก่อนหน้านี้ ตอนนี้เมื่อเรารันก็จะได้ผลตามรูป



รูป 4.1 รูปแสดงการรันซึ่งเป็นเสมือน Main Menu ที่รอรับการเลือกฟังก์ชันการใช้งาน

จากรูปด้านบนเป็นการรันโปรแกรมที่เป็นการรอรับค่าข้อมูลผ่านมาทางพอร์ตอนุกรมตัวที่
 หนึ่ง (Com1) เมื่อมีข้อมูลมาจากบอร์ด เข้ามาที่คอมพิวเตอร์ส่วนที่เหลือทั้งหมดจะจัดการบน
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น เมื่อผู้ดูแลเห็นว่าเป็นประโยชน์ในการ
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

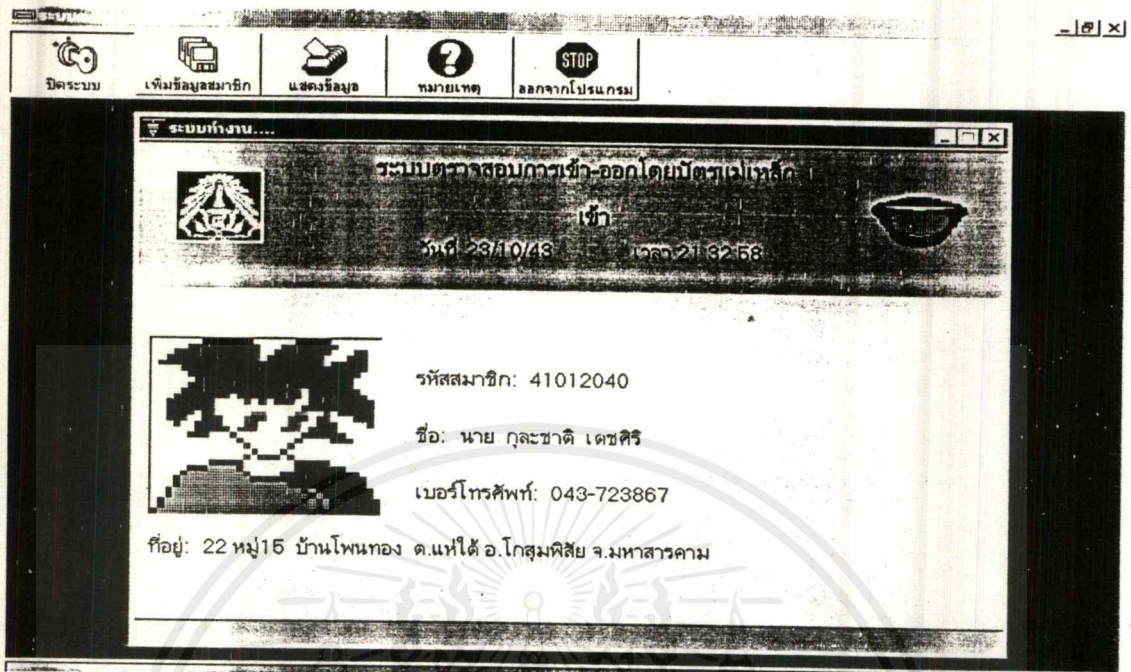
คอมพิวเตอร์ โดยการเอาข้อมูลมาประมวลผลและเก็บไว้ใน Database จากรูปด้านบนเป็น Main Menu เมื่อเรากดที่ Menu Bar “เปิดระบบ” ก็จะเป็นการ On line รอรับค่าข้อมูล โดยเมื่อกด “เปิดระบบ” โปรแกรมก็จะทำการโหลด Form ใหม่ขึ้นมาดังรูป



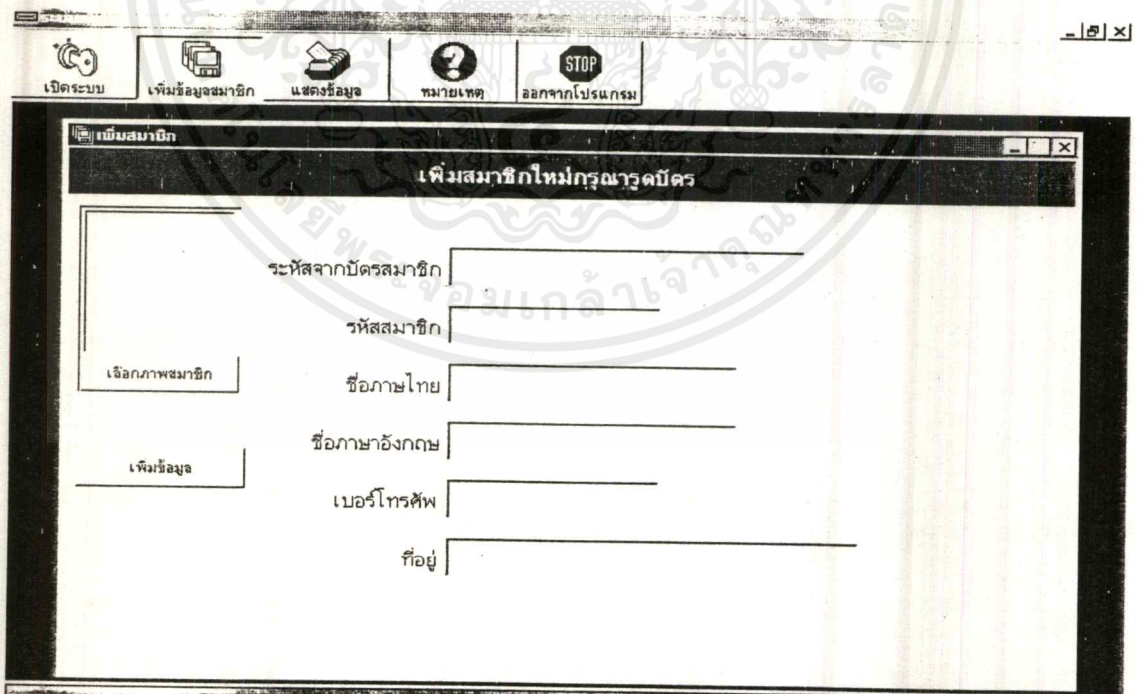
รูปที่ 4.2 เมื่อ Click ที่ Menu “เปิดระบบ On line” จะ Load Form ใหม่ขึ้นมา

จากรูปเมื่อเรากด “เปิดระบบ Online” ก็จะปรากฏ Form ข้างบนจะรอกันกว่าจะมีการรูดบัตรแล้วตัวโปรแกรมก็จะทำการเปรียบเทียบกับข้อมูลที่มีอยู่ในค้ำค้าเบสว่ามีอยู่หรือไม่ ถ้ามีก็จะส่งข้อมูลเพื่อไปจับอุปกรณ์ภายนอกต่อไป แต่ถ้าไม่มีก็จะส่ง 0 เพื่อให้บอร์ด ทำงานได้ถูกต้อง Form นี้ได้ฟังก์ชันพิเศษของ Visual Basic คือ มันจะสามารถพูดได้ เมื่อ ผู้ใช้กดสวิทช์เซ็ทที่บอร์ด บอร์ด จะส่งข้อมูล “START” มาที่พอร์ต เมื่อโปรแกรมตรวจเจอมันก็จะพูด(ปากก็จะขยับตาม) เพื่อเป็นการต้อนรับและบอกให้รู้ว่า ต่อไปต้อง รูดบัตรที่ตัวรูดบัตร เหมือนกับว่าเรามีพนักงานต้อนรับที่ประตู เมื่อเรารูดบัตรเสร็จถ้าข้อมูลไม่ตรงกับ DataBase ก็จะมีเสียงพูดออกมาเพื่อบอกให้ผู้รูดบัตรทราบ และถ้าตรงก็จะพูดชื่อเจ้าของบัตรต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 เมื่อรูดบัตรที่มีข้อมูลใน Database ก็จะแสดงข้อมูลของผู้ถือบัตร



รูปที่ 4.4 เมื่อทำการ Click ที่ Menu “เพิ่มข้อมูลสมาชิก” ก็จะ Load Form ใหม่ขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปด้านบนเป็นการ Load Form ที่ใช้สำหรับเพิ่มข้อมูลสมาชิก เมื่อเรารูดบัตรและใส่ข้อมูลของสมาชิกใส่รูปภาพโดยการเลือก File ใน เพิ่มข้อมูล เมื่อเสร็จเรียบร้อยแล้วก็ทำการClickที่เพิ่มข้อมูลดังรูปข้างล่างเมื่อเรา เพิ่มข้อมูลโดยการ Click ที่ปุ่มเพิ่มข้อมูลหน้าจอก็จะทำการเคลียร์ข้อมูลเก่าแล้วให้เพิ่มข้อมูลคนใหม่เข้าไปได้อีก

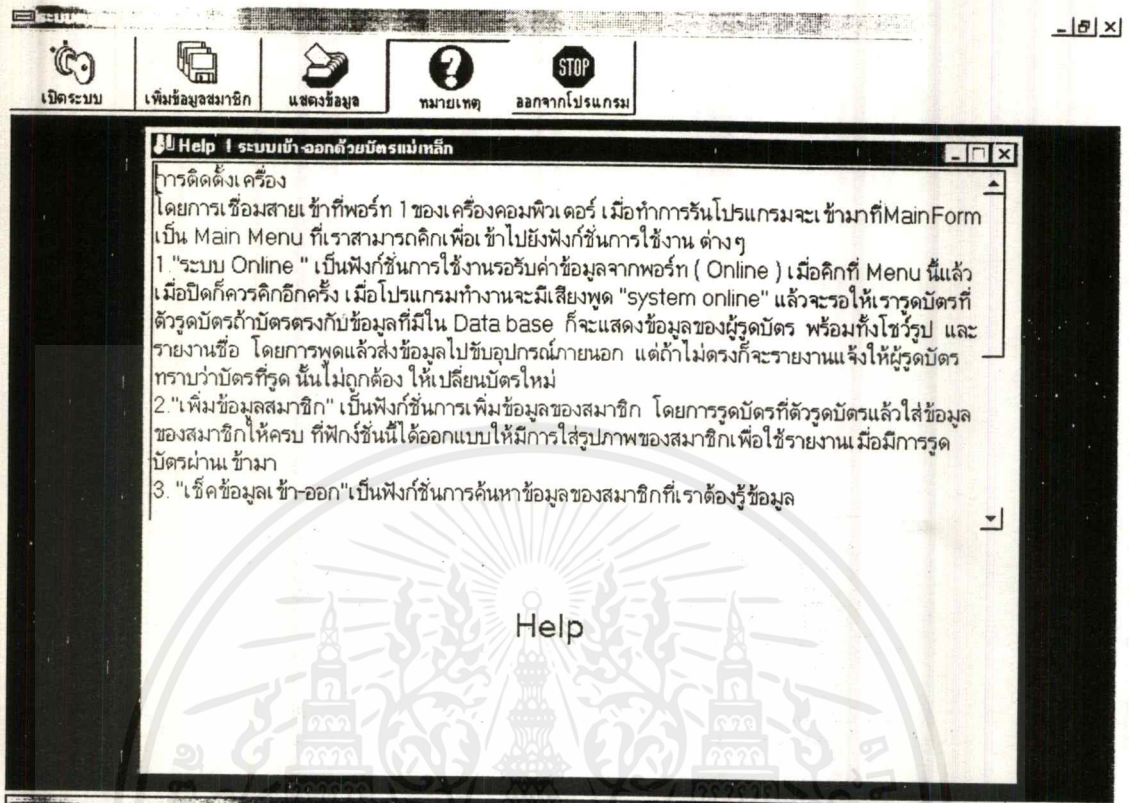
ID	Name	Thai name	Address	Tel	Time In	Tim
▶ 41012040	Mr. Kulachat Detsiri	นาย ภูษชาติ เดชศิริ	22 หมู่15 บ้านโพหนอง ต	043-723867	0:47:57	0:4
41012042	Mr. Suljarit Tungitrean	นาย สุจริต ตั้งจิตเรียน	300/125 ชาติกระมัง กทม.	7390278 ต่อ 517	0:47:35	0:4
41012040	Mr. Kulachat Detsiri	นาย ภูษชาติ เดชศิริ	22 หมู่15 บ้านโพหนอง ต	043-723867	2:30:50	2:3
41012040	Mr. Kulachat Detsiri	นาย ภูษชาติ เดชศิริ	22 หมู่15 บ้านโพหนอง ต	043-723867	2:36:51	2:3
41012042	Mr. Suljarit Tungitrean	นาย สุจริต ตั้งจิตเรียน	300/125 ชาติกระมัง กทม.	7390278 ต่อ 517	2:38:29	2:3
41012042	Mr. Suljarit Tungitrean	นาย สุจริต ตั้งจิตเรียน	300/125 ชาติกระมัง กทม.	7390278 ต่อ 517	2:41:35	2:4
41012040	Mr. Kulachat Detsiri	นาย ภูษชาติ เดชศิริ	22 หมู่15 บ้านโพหนอง ต	043-723867	2:41:41	2:4
41012045	Mr. John Horner	Mr. John Horner	100-300 Westside River	101-598-6633	3:03:37	3:0
41012045	Mr. John Horner	Mr. John Horner	100-300 Westside River	101-598-6633	3:03:59	3:0
41012045	Mr. John Horner	Mr. John Horner	100-300 Westside River	101-598-6633	3:04:12	3:0
41012040	Mr. Kulachat Detsiri	นาย ภูษชาติ เดชศิริ	22 หมู่15 บ้านโพหนอง ต	043-723867	3:08:11	3:2
41012042	Mr. Suljarit Tungitrean	นาย สุจริต ตั้งจิตเรียน	300/125 ชาติกระมัง กทม.	7390278 ต่อ 517	3:08:18	3:2
41012045	Mr. John Horner	Mr. John Horner	100-300 Westside River	101-598-6633	3:08:23	3:2
41012040	Mr. Kulachat Detsiri	นาย ภูษชาติ เดชศิริ	22 หมู่15 บ้านโพหนอง ต	043-723867	21:32:58	ถึงเ

รูปที่ 4.5 เมื่อ Click ที่ Menu “แสดงข้อมูล” จะทำการ Load Form ที่แสดงข้อมูลการเข้า-ออก

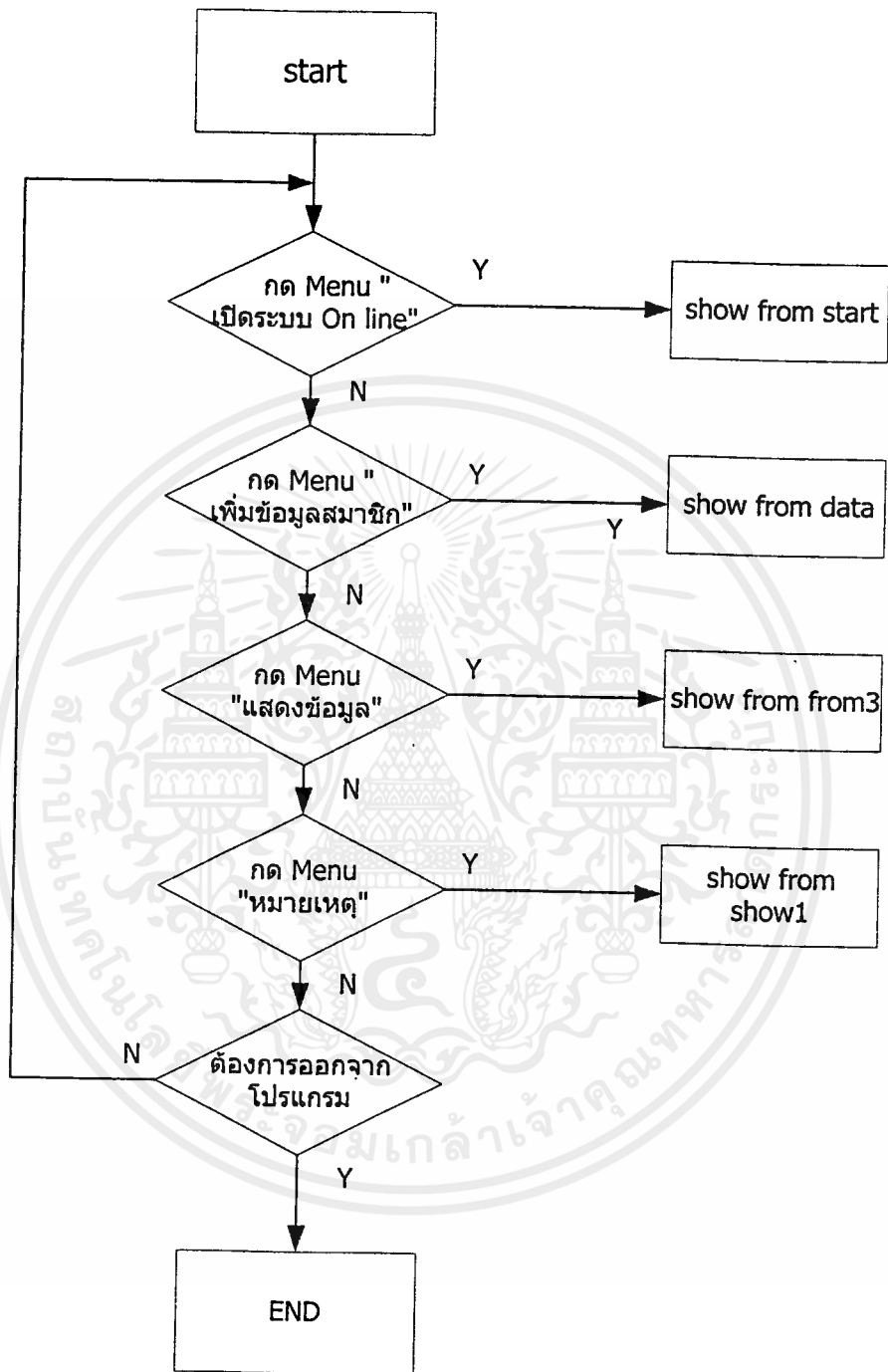
เมื่อเลือก Menu แสดงข้อมูล ก็จะ Load Form ที่ใช้แสดงข้อมูลการเข้าออกโดยจะแสดงข้อมูลที่มีอยู่ใน Database ซึ่งจากรูปข้อมูลจะดูไม่เต็มเราก็ทำการเลื่อนเมาท์เพื่อดูข้อมูลที่สามารถได้

ฟังก์ชันหมายเหตุใช้สำหรับเป็น Help ที่แนะนำการใช้โปรแกรม เมื่อต้องการออกจากโปรแกรมก็เลือก Click ที่ STOP โปรแกรมก็จะทำการปิด

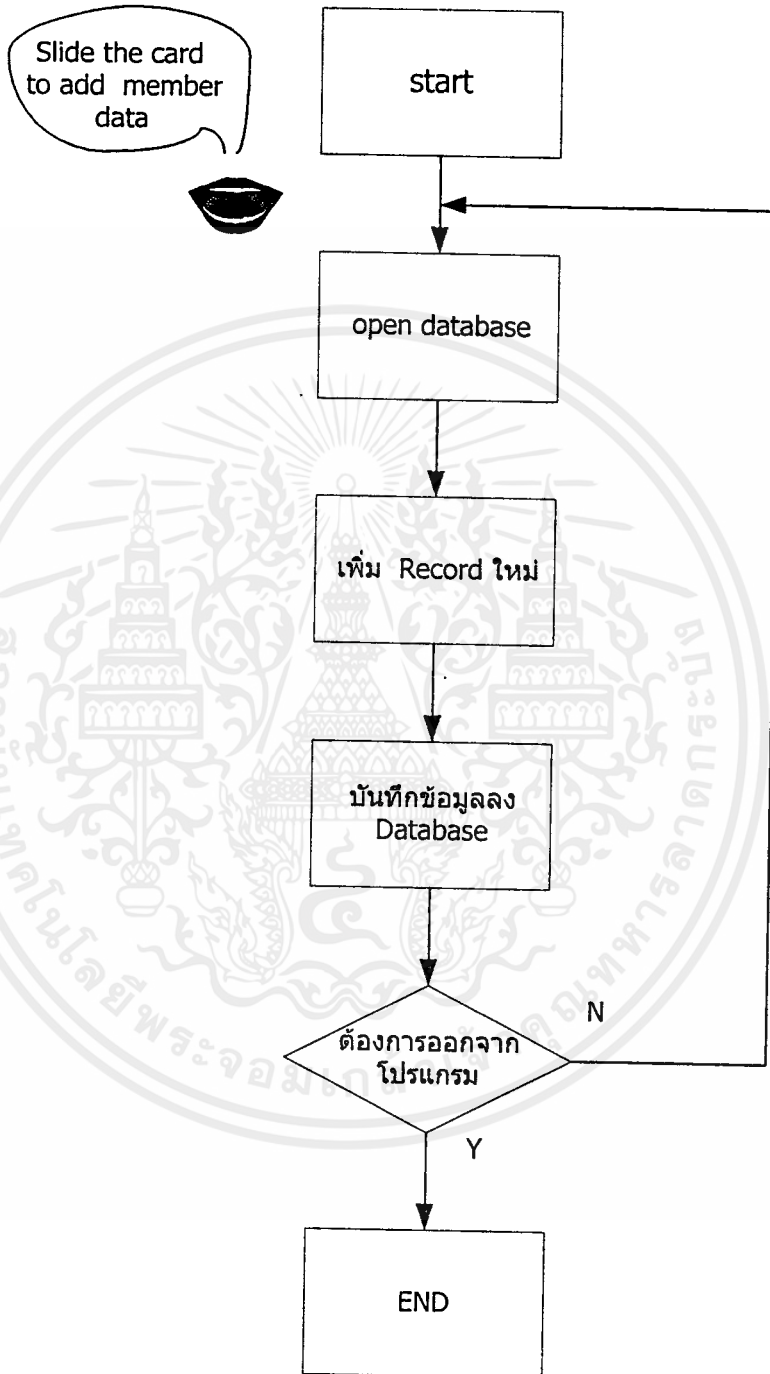
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



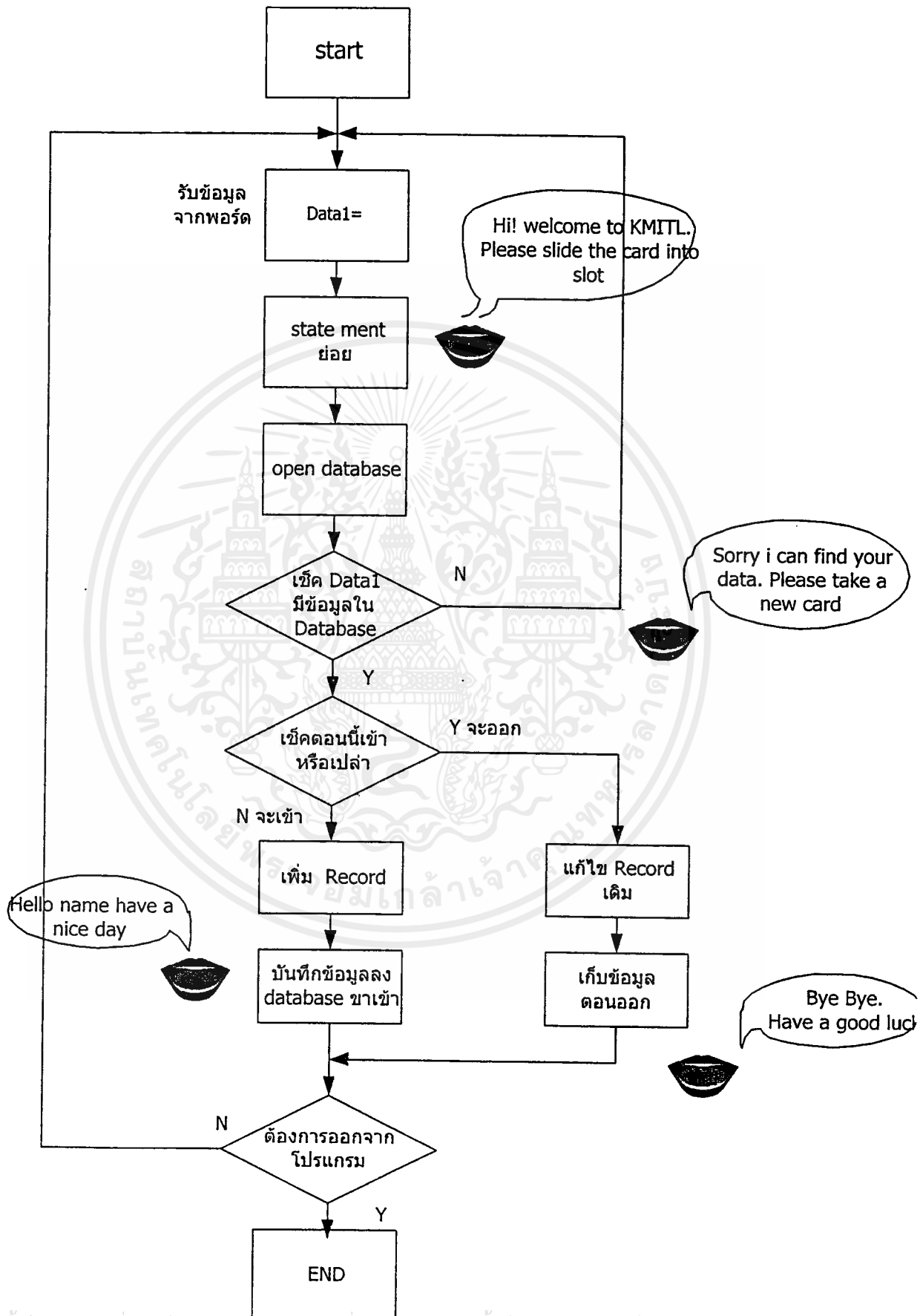
รูปที่ 4.6 แสดงเมื่อมีการ Click ที่ Menu “หมายเหตุ” ทำหน้าที่เป็น Help



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

ปริญญาบัตรฉบับนี้เสนอผลงานเกี่ยวกับระบบควบคุมการเข้าออกโดยใช้บัตรแม่เหล็กใช้สำหรับเป็นระบบรักษาความปลอดภัยแก่องค์กรทำหน้าที่เพื่อใช้ตรวจสอบข้อมูลของผู้ผ่านเข้าออกว่าผู้ที่ผ่านเข้าออกนั้นเป็นใครและทำให้ทราบข้อมูลที่ระบุไว้ใน Database ของบุคคลนั้น ก่อนที่จะสามารถผ่านเข้า-ออกได้ผู้ผ่านเข้าออกนั้นจะต้องมีบัตรที่ทำการบันทึกข้อมูลลงใน Database ก่อน ข้อมูลของสมาชิกก็จะประกอบไปด้วยข้อมูลที่สำคัญๆ อาทิเช่น ID ,รหัสการ์ด,ชื่อ,ที่อยู่,เบอร์โทร และข้อมูลต่างๆ

แล้วแต่ผู้ควบคุมระบบว่าต้องการข้อมูลได้ไว้ใช้ประโยชน์ในองค์กรบ้าง Software ควบคุม พัฒนาโดย Visual Basic และมีส่วน Interface ติดต่อกับอุปกรณ์ภายนอกและติดต่อกับหัวอ่านแถบแม่เหล็ก เมื่อผู้ผ่านเข้าออกเป็นสมาชิกก็จะทำการจับอุปกรณ์ให้เปิดประตู ซึ่งโดยสรุปแล้ว โครงการงานสามารถที่จะทำงานได้ดีเป็นที่น่าพอใจในระดับหนึ่ง โดยเฉพาะ Software ได้เพิ่ม ลูกเล่นต่างๆเพื่อให้เพิ่มประสิทธิภาพแก่ระบบมากขึ้น

5.2 ปัญหาและแนวทางแก้ไข

จากการลงมือเริ่มทำโครงการนี้พบว่าปัญหาที่เกิดขึ้นเมื่อลงมือปฏิบัติจะมีมากนอกเหนือจากที่ได้วางแผนไว้ อาทิเช่นว่า

1. Clock ไม่เสถียร เกิดการรบกวนของระบบกราวด์ของเครื่องคอมพิวเตอร์กับกราวด์ของอุปกรณ์ภายนอกต้องแก้ไขโดยการบล็อคด้วย อุปกรณ์ประเภท Opto ก็ทำให้การทำงานดีขึ้น
2. อุปกรณ์ที่เป็นหัวอ่านหายากต้องรอสั่งซื้อนาน

5.3 การพัฒนาโครงการ

โครงการนี้สามารถที่จะพัฒนาเพื่อที่จะรองรับความต้องการของผู้ใช้งานและเนื่องจากเทคโนโลยีได้พัฒนาขึ้นเรื่อยๆ จึงทำให้เราจำเป็นต้องปรับปรุงตามไปด้วยโครงการนี้สามารถที่จะปรับปรุงได้หลายส่วนด้วยกัน

1. ส่วนของ Software สามารถที่จะเพิ่มลูกเล่นได้หลากหลายรูปแบบเพื่อให้รองรับกับความต้องการของผู้ใช้งาน
2. ส่วนควบคุมอุปกรณ์หรือส่วนของคอนโทรลเลอร์สามารถที่จะเพิ่มการควบคุมได้

หลายอย่างเพราะคอนโทรลเลอร์ตัวที่ใช้นั้นมีพอร์ตถึง 3 พอร์ต และเมื่อจะปรับปรุงหรือแก้ไขก็
เพียงแต่ปรับปรุงที่ตัว Software เท่านั้นทำให้ง่ายในการแก้ไขหรือปรับปรุง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

กิตติ ภัคดีวัฒนกุล, จำลอง ทรูตสาหะ. Visual Basic 6 ฉบับโปรแกรมเมอร์. กรุงเทพมหานคร :
ไทยเจริญการพิมพ์, 2542

รศ.ดร.ถวัลย์วงศ์ ไกรโรจนานันท์. อิเล็กทรอนิกส์ระบบดิจิทัล. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น
พิมพ์ครั้งที่ 1,2539

ชัยวัฒน์ ลิ้มพรจิตรวิไล, วรพจน์ กรแก้ววัฒนกุล. เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์

(MCS – 51 Microcontroller Theory & Practical Approach). อินโนเวทีฟ เอ็กเพอริเมนต์

รศ.ดร.สมยศ จุณณะปิยะ. การประยุกต์ใช้งานไมโครคอนโทรลเลอร์(Microcontroller Applications

MCS – 51). พิมพ์ครั้งที่ 3, 2543

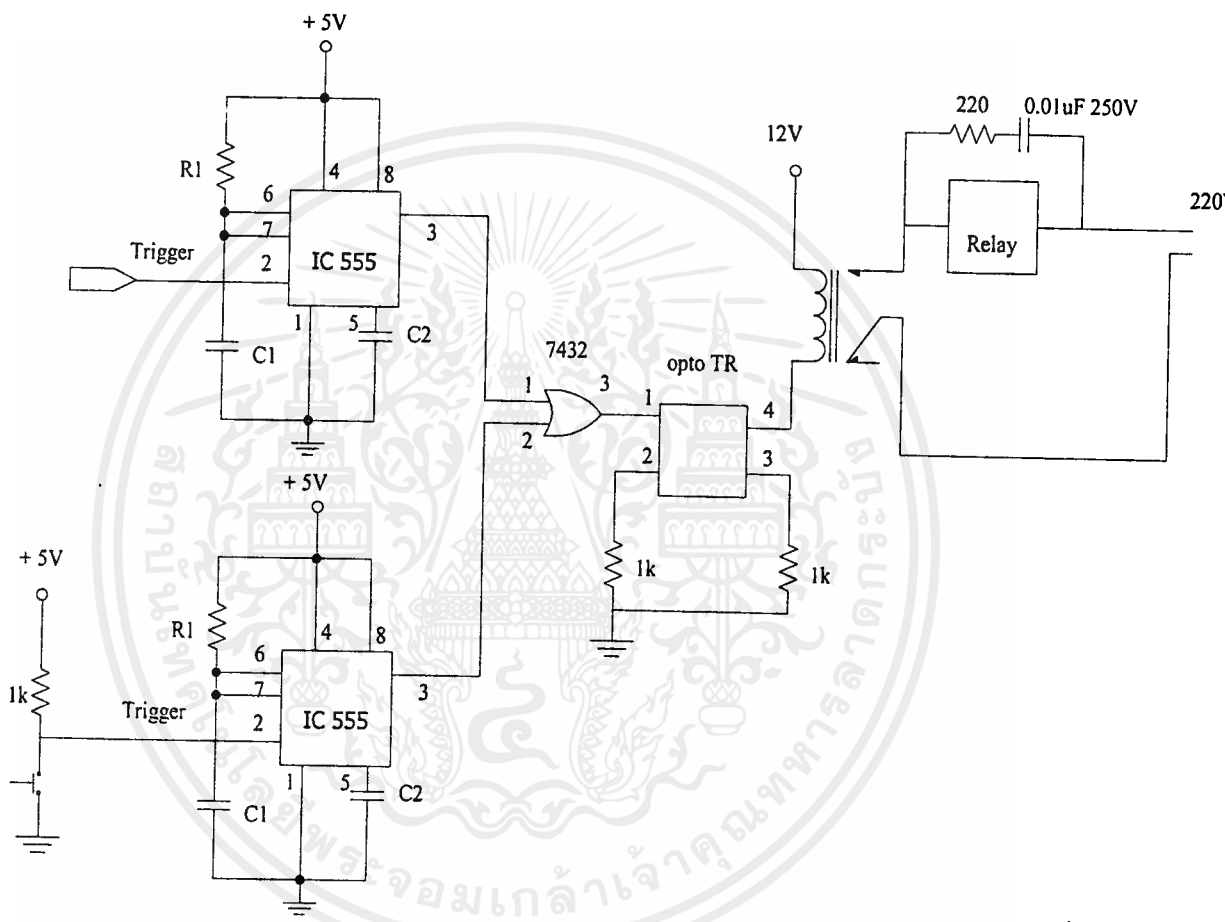
<http://www.microsoft.com>

<http://www.atmel.com>

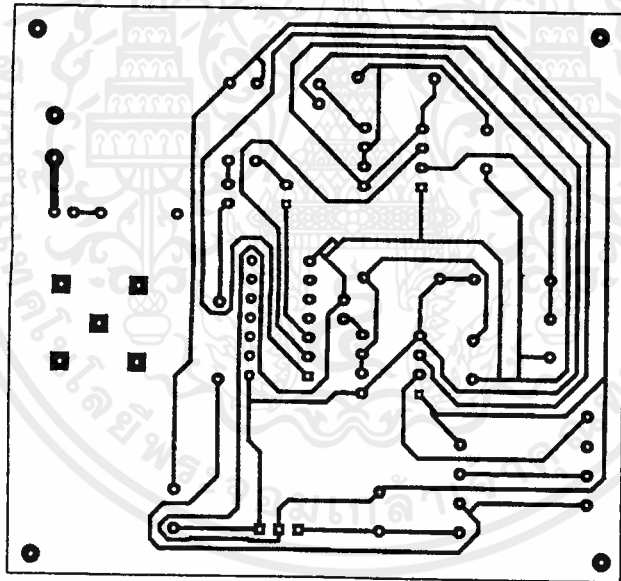
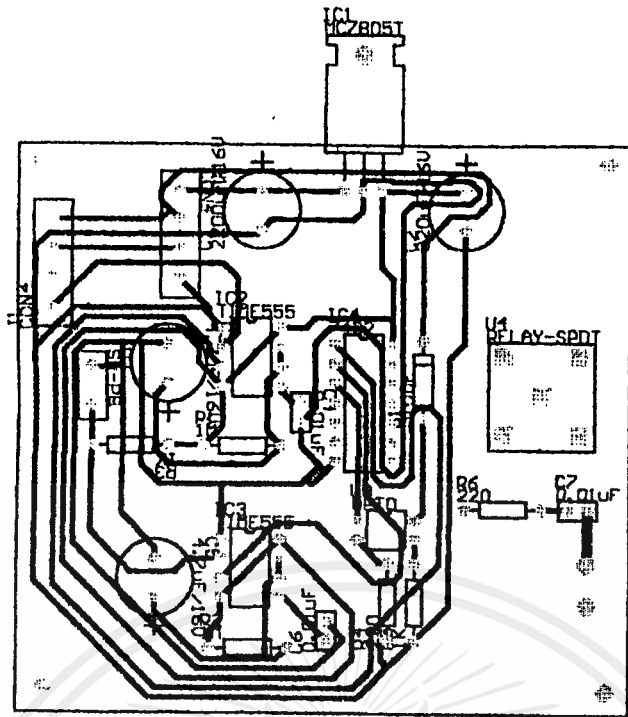
<http://www.tokin.com>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



๕๗

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*****
;* โปรแกรม Decode ข้อมูลจากหัวอ่านแถบแม่เหล็ก *
;* Read Track2 Modulo5 Format *
;* Display Data Code to RS232 *
;* CPU Control : AT89C2051 *
;* Complier : SXA51 *
;* (Used Interrupt INT0 Mode) *
;*****
;
CLK_MAG EQU P1.0 ; Clock Sync Magnetic Card
DATA_MAG EQU P1.1 ; Data From Magnetic Card
PRESENT EQU P3.2 ; Enable Magnetic Card (INT0)

ORG 20H
FLAG_BUF: DS 1 ; Flag Buffer
CHECKSUM: DS 1
BUFFER: DS 40 ; 40 Byte Buffer
;
START EQU FLAG_BUF.0 ; Start Flag Status
STOP EQU FLAG_BUF.1 ; Stop Flag Status
ERROR EQU FLAG_BUF.2 ; Error Flag Status
PARITY EQU FLAG_BUF.3 ; Parity Error Flag Status
UPDATE EQU FLAG_BUF.4

ORG 0000H
LJMP PROGRAM

ORG 0003H
LJMP GET_DATA
;
PROGRAM: MOV SP,#128-32 ; Stack 32 Byte
MOV FLAG_BUF,#0 ; Clear All Flag
SETB DATA_MAG ; Standy Signale
SETB PRESENT
SETB CLK_MAG
;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

INIT_SR: MOV  A,#0FDH      ; Set baud rate 9600
        MOV  TH1,A
        MOV  TL1,A
        MOV  TMOD,#00100000B ; TIMER1 MODE 1
        CLR  ES            ; Disable serial interrupt
        CLR  ET1          ; Disable timer1 interrupt
        SETB TR1          ; Set timer1 control
        MOV  SCON,#01010000B ; Serial mode 1
        ;
        SETB EX0          ; Enable EX0
        SETB IT0          ; Interrupt By Falling Edge
        SETB EA           ; Enable Global Interrupt
        ;
TEST_MAG: LCALL PRINT_SER
        DB  'START',0DH,0AH,00
        ;
        ;
DSP_NUM0: JNB  UPDATE,$    ; Wait to Display Data
        ;
        MOV  R0,#BUFFER
        JB   ERROR,SUM_ERR
        JB   PARITY,PAR_ERR
        SJMP DSP_NUM1
        ;
SUM_ERR: CLR  ERROR
        CLR  UPDATE
        SJMP DSP_NUM0
        ;
PAR_ERR: CLR  PARITY
        CLR  UPDATE
        SJMP DSP_NUM0
        ;
DSP_NUM1: MOV  A,@R0
        CJNE A,#0FH,DSP_NUM2
        ADD  A,#30H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        LCALL TX_BYTE      ; Display End
        LJMP  DSP_NUM3
DSP_NUM2: ADD  A,#30H
        LCALL TX_BYTE
        INC  R0
        SJMP DSP_NUM1

```

```

;*****
;
;* Clear Data Buffer *
;* Register : R0,ACC *
;*****
;

```

```

CLR_BUFF: PUSH  B
          MOV   B,#40
          MOV   R0,#BUFFER ; Clear Data Buffer
CLR_BUF1: CLR   A
          MOV   @R0,A
          INC  R0
          DJNZ B,CLR_BUF1
          POP  B
          RET

```

```

;*****
;
;* Read Data From Magnetic Card *
;* Get Number of Card (Modulo5) *
;* Output : BUFFER (BCD 40Byte) *
;* Reg.  : R1,R2,ACC,CHECKSUM *
;*****
;

```

```

GET_DATA: LCALL CLR_BUFF      ; Clear Buffer
          MOV   CHECKSUM,#0
          MOV   R1,#BUFFER-1 ; Pointer to Save Data
          CLR   START        ; Clear Any Flag
          CLR   STOP
          CLR   ERROR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;
GET_DAT0: LCALL GET_CLK      ; Get Start Sentinel
          MOV  C,DATA_MAG
          CPL  C
          JB  DATA_MAG,GET_DAT0 ; Loop Until Start Bit
          SETB START          ; Strat Bit OK
;
GET_DAT1: MOV  R2,#5          ; Modulo 5 Format Counter
          CLR  A
          CLR  PARITY
          INC  R1              ; Point to Next Byte Save
GET_DAT2: JB  START,GET_DAT3 ; Get 1-Byte Data
          LCALL GET_CLK
          MOV  C,DATA_MAG
          CPL  C
GET_DAT3: RRC  A
          CLR  START
          DJNZ R2,GET_DAT2
          RR  A
          RR  A
          RR  A
          JB  P,GET_DAT4      ; Parity Odd OK
          SETB PARITY         ; Parity Error
          SJMP GET_DAT8
;
GET_DAT4: ANL  A,#0FH         ; Ignore Parity
          JB  STOP,GET_DAT5
          PUSH ACC
          XRL  A,CHECKSUM      ; Checksum Data
          MOV  CHECKSUM,A
          POP  ACC
GET_DAT5: MOV  @R1,A          ; Save Data
          JB  STOP,GET_DAT6   ; Stop Operation
          CJNE A,#0FH,GET_DAT1
          SETB STOP

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    SJMP GET_DAT1
GET_DAT6: CJNE A,CHECKSUM,GET_DAT7
    CLR ERROR          ; Checksum OK
    SETB UPDATE
    SJMP GET_DAT8
GET_DAT7: SETB ERROR  ; Checksum Error
GET_DAT8: JNB P3.2,$
    RETI

```

```

;*****
;
;* Get Magnetic Clock *
;*****
;
GET_CLK: JNB CLK_MAG,$ ; Wait Rising Clock
    NOP
    NOP
    JB CLK_MAG,$ ; Wait Falling Clock
    NOP
    NOP
    RET

```

```

;*****
;
;* Send 1-Byte to RS-232 *
;* Input : ACC *
;* Output : Serial port *
;*****
;
TX_BYTE: CLR TI
    MOV SBUF,A
    JNB TI,$
    CLR TI
    RET

```

```

;*****
;
;* Receive 1-Byte from RS-232 *
;* Input : BUFFER      *
;* Output : Serial port *
;*****
;
;
RX_BYTE: JNB RI,$
        CLR RI
        MOV A,SBUF
        JNZ OUT
        JZ  ZEE
        RET
;
RX_BYTE1: JNB RI,$
         CLR RI
         MOV A,SBUF
         RET
;
;*****
;* Outport 1 *
;*ส่งข้อมูลออก port1.5 *
;*****
;
OUT:    CPL P1.5
        LCALL DELAY
        CPL P1.5
        RET
;
;*****
;* Outport 2 *
;*ส่งข้อมูลออก port1.4 *
;*
;*****
;
ZEE:    CPL P1.4

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        LCALL DELAY
        CPL  P1.4
        RET
;
;*****
;* Delay *
;* ฟังก์ชันย่อย Delay*
;*****
;
DELAY:  MOV  R6,#0FFH
DEL1:   MOV  R7,#0FFH
        DJNZ R7,$
        DJNZ R6,DEL1
        RET
DSP_NUM3: MOV  A,#0DH      ; Carrier Return
        LCALL TX_BYTE
        MOV  A,#0AH      ; Line Feed
        LCALL TX_BYTE
        LCALL RX_BYTE
        LCALL DELAY
        MOV  A,#0BH
        LCALL TX_BYTE
        LCALL RX_BYTE1
        CLR  UPDATE
        LJMP DSP_NUM0
        RET
;
PRINT_SER: POP  DPH
          POP  DPL
PRINT1:   CLR  A
          MOVC A,@A+DPTR
          CJNE A,#00H,PRINT2
          SJMP PRINT3
PRINT2:   LCALL TX_BYTE
          INC  DPTR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SJMP PRINT1

PRINT3: PUSH DPL

PUSH DPH

RET

END



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมตรวจสอบการเข้าออกด้วยบัตรแม่เหล็ก

```
'Form2
'start Control
'Option Explicit
Dim Loopimg As Byte
Private Sub Form_Load()
Dim D1 As Variant
Dim Instring As Variant
Loopimg = 0
MSComm1.CommPort = 1
MSComm1.RThreshold = 1
MSComm1.PortOpen = True
Me.Height = 2400
End Sub
Private Sub delay()
Dim x As Variant
For x = 0 To 150000
Next x
End Sub
Private Sub delay1()
Dim x As Variant
For x = 0 To 450000
Next x
End Sub
Private Sub MSComm1_OnComm()
Dim a, b, C As Integer
Dim buffer As Variant
Dim dbsperson As Database
Dim rstperson As Recordset
Dim strperson As String
Dim strinoutCheck As String
Dim Data1 As Variant
Dim Temp(1 To 10) As String
D1 = ""
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Match = ""
delay
D1 = MSComm1.Input
m = InStr(1, D1, "S", 0)
If m = 1 Then
TextToSpeech1.Speak "Hello Wellcome to K-M-I-T-L "
delay1
TextToSpeech1.Speak "Please slide the card into slot"
delay1
End If
a = InStr(1, D1, ":", 0)
If a <> 0 Then
b = InStr(1, D1, "?", 0)
Data1 = Mid(D1, a, (b - a) + 1)
C = InStr(1, Data1, ":", 0)
End If
Set dbsperson = OpenDatabase(App.Path & "\db1.mdb")
Set rstperson = dbsperson.OpenRecordset("table1", dbOpenDynaset)
strperson = "GardId = " & Data1 & " "
With rstperson
.MoveLast
.FindLast strperson
If .NoMatch = False Then
Me.Height = 6855
Match = "ok"
If rstperson("Time_out") = "ยังไม่ออก" Then
.Edit
rstperson("Time_out") = Time
rstperson("Check_in") = "1"
.Update
Label10.Caption = "ออก"
Label12.Caption = Date
Label14.Caption = "เวลาเข้า"
Label15.Caption = "เวลาออก"
Label13.Caption = rstperson("Time_out")

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Label16.Caption = rstperson("Time_in")
Label17.Caption = rstperson("Time_out")
Label6.Caption = rstperson("ID")
Label7.Caption = rstperson("Thai_name")
Label8.Caption = rstperson("Tel")
Label9.Caption = rstperson("Address")
Label12.Caption = Date
Image1.Picture = LoadPicture(rstperson("Image"))
TextToSpeech1.Speak " Bye Bye "
delay1
TextToSpeech1.Speak rstperson("Name")
delay1
TextToSpeech1.Speak "Have a good luck"
delay1
MSComm1.Output = Chr$(0)
MSComm1.Output = Chr$(0)
Else
If rstperson("Check_in") = "1" Then
Timer1.Enabled = False
D1 = Time
Temp(1) = rstperson("GardId")
Temp(2) = rstperson("ID")
Temp(3) = rstperson("Name")
Temp(4) = rstperson("Thai_name")
Temp(5) = rstperson("Address")
Temp(6) = rstperson("Tel")
Temp(7) = rstperson("Image")
.AddNew
!GardId = Temp(1)
!ID = Temp(2)
!Name = Temp(3)
!Thai_name = Temp(4)
!Address = Temp(5)
!Tel = Temp(6)
!Image = Temp(7)
```

```

!Time_In = D1
!Date = Date
!Time_out = "ยังไม่ออก"
!Check_in = "0"
.Update
End If
TextToSpeech1.Speak "hello"
delay1
TextToSpeech1.Speak rstperson("Name")
delay1
TextToSpeech1.Speak "Have a nice day "
delay1
Label10.Caption = "เข้า"
Label14.Caption = ""
Label15.Caption = ""
Label16.Caption = ""
Label17.Caption = ""
Label6.Caption = Temp(2)
Label7.Caption = Temp(4)
Label8.Caption = Temp(6)
Label9.Caption = Temp(5)
Label13.Caption = D1
Label12.Caption = Date
Image1.Picture = LoadPicture(rstperson("Image"))
MSComm1.Output = Chr$(1)
MSComm1.Output = Chr$(1)
End If
End If
If .NoMatch = True Then
If C = 1 Then
out0
End If
End If
End With
MSComm1.Output = Chr$(0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MSComm1.Output = Chr$(0)
rstperson.Close
dbsperson.Close
End Sub
Private Sub out0()
Dim Z As Variant
delay
If Match <> "ok" Then
If m <> 1 Then
MSComm1.Output = Chr$(0)
MSComm1.Output = Chr$(0)
TextToSpeech1.Speak "sorry i can't find your data"
delay1
TextToSpeech1.Speak "Please take a new card "
delay1
End If
End If
Match = ""
End Sub
Private Sub out1()
Dim R As Variant
MSComm1.Output = Chr$(1)
MSComm1.Output = Chr$(1)
delay
End Sub

Private Sub Timer2_Timer()
Label6.Caption = Format(Now, "dd/mm/yy hh:mm:ss AMPM")
End Sub

Private Sub Timer3_Timer()
Select Case looping
Case 1
Me.Caption = "ร"
Case 2
Me.Caption = "ระ"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Case 3

Me.Caption = "ระบบ"

Case 4

Me.Caption = "ระบบ"

Case 5

Me.Caption = "ระบบท"

Case 6

Me.Caption = "ระบบทำ"

Case 7

Me.Caption = "ระบบทำง"

Case 8

Me.Caption = "ระบบทำงา"

Case 9

Me.Caption = "ระบบทำงาน"

Case 10

Me.Caption = "ระบบทำงาน."

Case 11

Me.Caption = "ระบบทำงาน..."

Case 12

Me.Caption = "ระบบทำงาน...."

Case 13

Me.Caption = "ระบบทำงาน...."

Case 14

looping = 0

End Select

looping = looping + 1

End Sub

'Form3

Private Sub Timer1_Timer()

Select Case looping

Case 1

Me.Caption = "u"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Case 2
Me.Caption = "แสด"
- Case 3
Me.Caption = "แสด"
- Case 4
Me.Caption = "แสดง"
- Case 5
Me.Caption = "แสดงข"
- Case 6
Me.Caption = "แสดงขั"
- Case 7
Me.Caption = "แสดงขัอ"
- Case 8
Me.Caption = "แสดงขัอม"
- Case 9
Me.Caption = "แสดงขัอมู"
- Case 10
Me.Caption = "แสดงขัอมูล"
- Case 11
Me.Caption = "แสดงขัอมูลก"
- Case 12
Me.Caption = "แสดงขัอมูลกา"
- Case 13
Me.Caption = "แสดงขัอมูลการ"
- Case 14
Me.Caption = "แสดงขัอมูลการ"
- Case 15
Me.Caption = "แสดงขัอมูลการเข"
- Case 16
Me.Caption = "แสดงขัอมูลการเข้"
- Case 17
Me.Caption = "แสดงขัอมูลการเข้"
- Case 18
Me.Caption = "แสดงขัอมูลการเข้"
- Case 19



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Me.Caption = "แสดงข้อมูลการเข้า-อ"

Case 20

Me.Caption = "แสดงข้อมูลการเข้า-ออก"

Case 21

Me.Caption = "แสดงข้อมูลการเข้า-ออก"

Case 22

looping = 0

End Select

looping = looping + 1

End Sub

'Form4

Private Sub Form_Load()

TextToSpeech1.Speak " you wanna help "

looping = 0

End Sub

Private Sub Timer1_Timer()

Select Case looping

Case 1

Me.Caption = ""

Case 2

Me.Caption = "H"

Case 3

Me.Caption = "He"

Case 4

Me.Caption = "Hel"

Case 5

Me.Caption = "Help"

Case 6

Me.Caption = "Help "

Case 7

Me.Caption = "Help !"

Case 8

Me.Caption = "Help ! r"



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Case 9

Me.Caption = "Help ! ะ"

Case 10

Me.Caption = "Help ! ะบ"

Case 11

Me.Caption = "Help ! ะบบ"

Case 12

Me.Caption = "Help ! ะบบ"

Case 13

Me.Caption = "Help ! ะบบบ"

Case 14

Me.Caption = "Help ! ะบบบ"

Case 15

Me.Caption = "Help ! ะบบบบ"

Case 16

Me.Caption = "Help ! ะบบบบ"

Case 17

Me.Caption = "Help ! ะบบบบ-"

Case 18

Me.Caption = "Help ! ะบบบบ-"

Case 19

Me.Caption = "Help ! ะบบบบ-"

Case 20

Me.Caption = "Help ! ะบบบบ-"

Case 21

Me.Caption = "Help ! ะบบบบ-"

Case 22

Me.Caption = "Help ! ะบบบบ-"

Case 23

Me.Caption = "Help ! ะบบบบ-"

Case 24

Me.Caption = "Help ! ะบบบบ-"

Case 25

Me.Caption = "Help ! ะบบบบ-"

Case 26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Me.Caption = "Help ! ระบบเข้า-ออกด้วยบัตร"

Case 27

Me.Caption = "Help ! ระบบเข้า-ออกด้วยบัตร"

Case 28

Me.Caption = "Help ! ระบบเข้า-ออกด้วยบัตรแม่"

Case 29

Me.Caption = "Help ! ระบบเข้า-ออกด้วยบัตรแม่"

Case 30

Me.Caption = "Help ! ระบบเข้า-ออกด้วยบัตรแม่"

Case 31

Me.Caption = "Help ! ระบบเข้า-ออกด้วยบัตรแม่"

Case 32

Me.Caption = "Help ! ระบบเข้า-ออกด้วยบัตรแม่เห"

Case 33

Me.Caption = "Help ! ระบบเข้า-ออกด้วยบัตรแม่เหล็ก"

Case 34

Me.Caption = "Help ! ระบบเข้า-ออกด้วยบัตรแม่เหล็ก"

Case 35

Me.Caption = "Help ! ระบบเข้า-ออกด้วยบัตรแม่เหล็ก"

Case 36

looping = 0

End Select

looping = looping + 1

End Sub

Form5

Private Sub CmdGetpicture_Click()

On Error Resume Next

ComDIPic.ShowOpen

Image1.Picture = LoadPicture(ComDIPic.FileName)

End Sub

Private Sub Command1_Click()

Dim dbNameAdd As Database

Dim rsTable As Recordset

Set dbNameAdd = OpenDatabase(App.Path & "db1.mdb")

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Set rsTable = _
    dbNameAdd.OpenRecordset("table1", dbOpenDynaset)
With rsTable
    .AddNew
    !GardId = Text1.Text
    !ID = Text6.Text
    !Name = Text3.Text
    !Thai_name = Text2.Text
    !Address = Text5.Text
    !Tel = Text4.Text
    !Time_in = "-"
    !Time_out = "-"
    !Image = ComDIPic.FileName
    !Check_in = "1"
    .Update
    .Bookmark = .LastModified
End With
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Image1.Picture = LoadPicture(App.Path & "\NoPicture.bmp")
rsTable.Close
dbNameAdd.Close
End Sub

Private Sub Form_Load()
Dim D1 As Variant
Dim Instring As Variant
TextToSpeech1.Speak "Slide the gard to add member data "
MSComm1.CommPort = 1
MSComm1.RThreshold = 1
MSComm1.PortOpen = True
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub MSComm1_OnComm()
Dim a, b, C As Integer
Dim buffer As Variant
Dim Data1 As Variant
Dim dbNameAdd As Database
Dim rsTable As Recordset
Set dbNameAdd = OpenDatabase(App.Path & "\db1.mdb")
Set rsTable = _
    dbNameAdd.OpenRecordset("table1", dbOpenDynaset)
D1 = ""
delay
D1 = MSComm1.Input
m = InStr(1, D1, "S", 0)
a = InStr(1, D1, ";", 0)
If a <> 0 Then
b = InStr(1, D1, "?", 0)
Data1 = Mid(D1, a, (b - a) + 1)
C = InStr(1, Data1, ":", 0)
Text1.Text = Data1
MSComm1.Output = Chr$(0)
MSComm1.Output = Chr$(0)
End If
End Sub

Private Sub delay()
Dim x As Variant
For x = 0 To 150000
Next x
End Sub
*****

'Formdi
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
Case 1
If Button.Value = 1 Then
TextToSpeech1.Speak "Open System "

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Delay3
    Button.Caption = "ปิดระบบ"
    Form2.Show
Else
    TextToSpeech1.Speak "Close system "
    Delay3
    Button.Caption = "เปิดระบบ"
    Form2.MSComm1.PortOpen = False
    Unload Form2
End If
Case 2
If Button.Value = 1 Then
    Form5.Show
Else
    Form5.MSComm1.PortOpen = False
    Unload Form5
End If
Case 3
If Button.Value = 0 Then
    Form3.Show
Else
    Unload Form3
End If
Case 4
If Button.Value = 1 Then
    Form4.Show
Else
    Unload Form4
End If
Case 5
    If Button.Value = 1 Then
        End
    End If
End Select
End Sub

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

'Module1

Public Match As Variant

Public D1, check As Variant

Public m As Integer

Public looping As Variant

Sub main()

fmdi.Show

End Sub

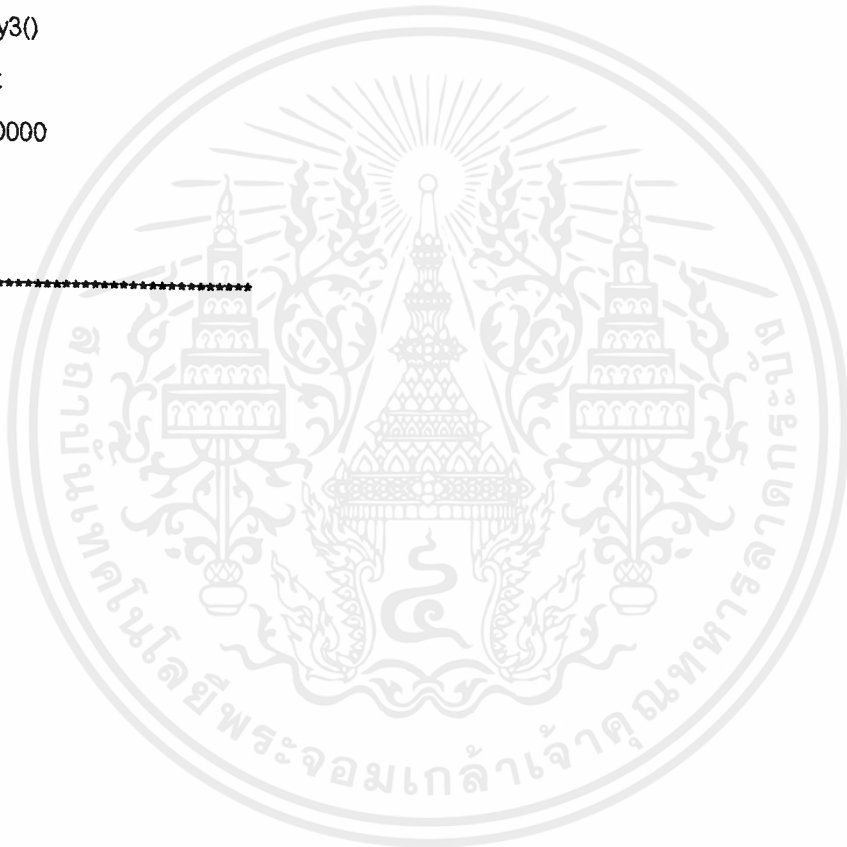
Public Sub Delay3()

Dim y As Variant

For y = 0 To 450000

Next y

End Sub



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

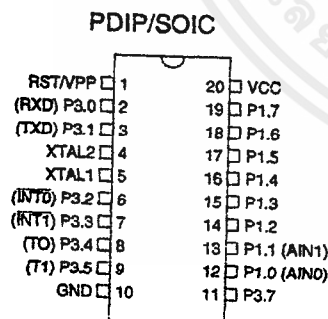
- Compatible with MCS-51™ Products
- 2K Bytes of Reprogrammable Flash Memory
 - Endurance: 1,000 Write/Erase Cycles
- 2.7V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Two-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 15 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial UART Channel
- Direct LED Drive Outputs
- On-chip Analog Comparator
- Low-power Idle and Power-down Modes

Description

The AT89C2051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 2K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C2051 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89C2051 provides the following standard features: 2K bytes of Flash, 128 bytes of RAM, 15 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, a precision analog comparator, on-chip oscillator and clock circuitry. In addition, the AT89C2051 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 hardware reset.

Pin Configuration



ATMEL®

8-bit Microcontroller with 2K Bytes Flash

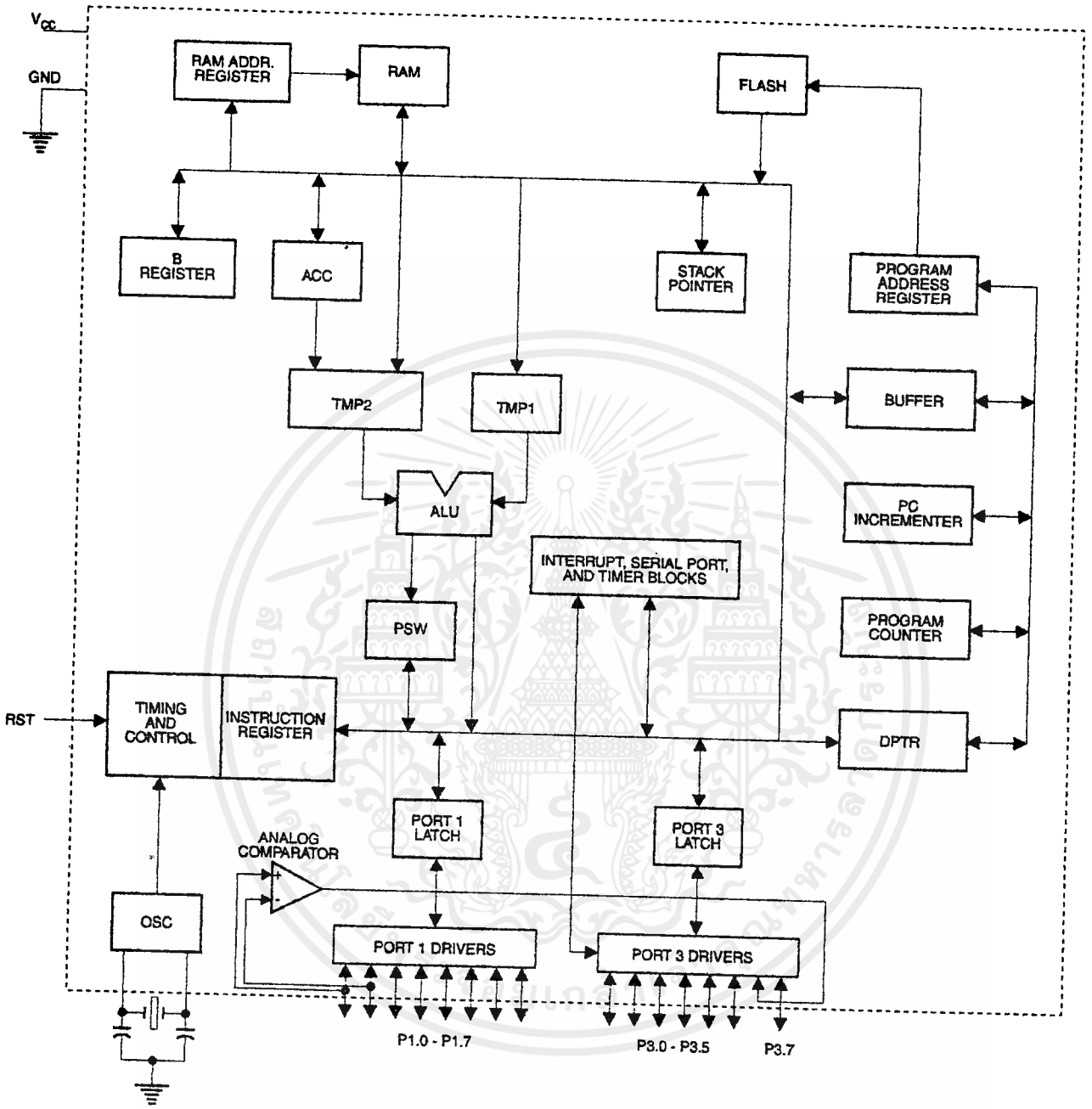
AT89C2051

Rev. 0368E-02/00

ATMEL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



AT89C2051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Description

VCC

Supply voltage.

GND

Ground.

Port 1

Port 1 is an 8-bit bi-directional I/O port. Port pins P1.2 to P1.7 provide internal pullups. P1.0 and P1.1 require external pullups. P1.0 and P1.1 also serve as the positive input (AIN0) and the negative input (AIN1), respectively, of the on-chip precision analog comparator. The Port 1 output buffers can sink 20 mA and can drive LED displays directly. When 1s are written to Port 1 pins, they can be used as inputs. When pins P1.2 to P1.7 are used as inputs and are externally pulled low, they will source current (I_{IL}) because of the internal pullups.

Port 1 also receives code data during Flash programming and verification.

Port 3

Port 3 pins P3.0 to P3.5, P3.7 are seven bi-directional I/O pins with internal pullups. P3.6 is hard-wired as an input to the output of the on-chip comparator and is not accessible as a general purpose I/O pin. The Port 3 output buffers can sink 20 mA. 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 AT89C2051 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)

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

RST

Reset input. All I/O pins are reset to 1s as soon as RST goes high. Holding the RST pin high for two machine cycles while the oscillator is running resets the device.

Each machine cycle takes 12 oscillator or clock cycles.

XTAL1

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

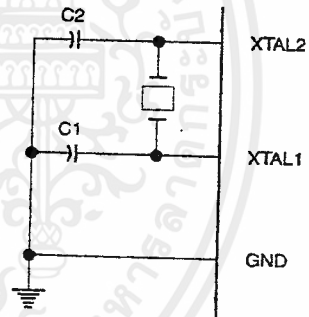
XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

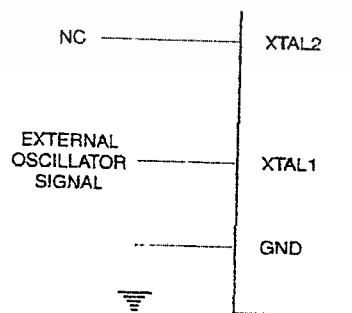
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 1. Oscillator Connections



Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 2. External Clock Drive Configuration





Special Function Registers

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

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.

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.

Table 1. AT89C2051 SFR Map and Reset Values

0F8H										0FFH
0F0H	B 00000000									0F7H
0E8H										0EFH
0E0H	ACC 00000000									0E7H
0D8H										0DFH
0D0H	PSW 00000000									0D7H
0C8H										0CFH
0C0H										0C7H
0B8H	IP XXX00000									0BFH
0B0H	P3 11111111									0B7H
0A8H	IE 0XX00000									0AFH
0A0H										0A7H
98H	SCON 00000000	SBUF XXXXXXXX								9FH
90H	P1 11111111									97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000				8FH
80H		SP 00000111	DPL 00000000	DPH 00000000					PCON 0XXX0000	87H

AT89C2051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Restrictions on Certain Instructions

The AT89C2051 and is an economical and cost-effective member of Atmel's growing family of microcontrollers. It contains 2K bytes of flash program memory. It is fully compatible with the MCS-51 architecture, and can be programmed using the MCS-51 instruction set. However, there are a few considerations one must keep in mind when utilizing certain instructions to program this device.

All the instructions related to jumping or branching should be restricted such that the destination address falls within the physical program memory space of the device, which is 2K for the AT89C2051. This should be the responsibility of the software programmer. For example, LJMP 7E0H would be a valid instruction for the AT89C2051 (with 2K of memory), whereas LJMP 900H would not.

1. Branching instructions:

LCALL, LJMP, ACALL, AJMP, SJMP, JMP @A+DPTR

These unconditional branching instructions will execute correctly as long as the programmer keeps in mind that the destination branching address must fall within the physical boundaries of the program memory size (locations 00H to 7FFH for the 89C2051). Violating the physical space limits may cause unknown program behavior.

CJNE [...], DJNZ [...], JB, JNB, JC, JNC, JBC, JZ, JNZ With these conditional branching instructions the same rule above applies. Again, violating the memory boundaries may cause erratic execution.

For applications involving interrupts the normal interrupt service routine address locations of the 80C51 family architecture have been preserved.

2. MOVX-related instructions, Data Memory:

The AT89C2051 contains 128 bytes of internal data memory. Thus, in the AT89C2051 the stack depth is limited to 128 bytes, the amount of available RAM. External DATA memory access is not supported in this device, nor is external PROGRAM memory execution. Therefore, no MOVX [...] instructions should be included in the program.

A typical 80C51 assembler will still assemble instructions, even if they are written in violation of the restrictions mentioned above. It is the responsibility of the controller user to know the physical features and limitations of the device being used and adjust the instructions used correspondingly.

Program Memory Lock Bits

On the chip are two lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

Lock Bit Protection Modes⁽¹⁾

Program Lock Bits			Protection Type
	LB1	LB2	
1	U	U	No program lock features.
2	P	U	Further programming of the Flash is disabled.
3	P	P	Same as mode 2, also verify is disabled.

Note: 1. The Lock Bits can only be erased with the Chip Erase operation.

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

P1.0 and P1.1 should be set to "0" if no external pullups are used, or set to "1" if external pullups are used.

It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Power-down Mode

In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

P1.0 and P1.1 should be set to "0" if no external pullups are used, or set to "1" if external pullups are used.



Programming The Flash

The AT89C2051 is shipped with the 2K bytes of on-chip PEROM code memory array in the erased state (i.e., contents = FFH) and ready to be programmed. The code memory array is programmed one byte at a time. *Once the array is programmed, to re-program any non-blank byte, the entire memory array needs to be erased electrically.*

Internal Address Counter: The AT89C2051 contains an internal PEROM address counter which is always reset to 000H on the rising edge of RST and is advanced by applying a positive going pulse to pin XTAL1.

Programming Algorithm: To program the AT89C2051, the following sequence is recommended.

1. Power-up sequence:
Apply power between V_{CC} and GND pins
Set RST and XTAL1 to GND
 2. Set pin RST to "H"
Set pin P3.2 to "H"
 3. Apply the appropriate combination of "H" or "L" logic levels to pins P3.3, P3.4, P3.5, P3.7 to select one of the programming operations shown in the PEROM Programming Modes table.
- To Program and Verify the Array:
4. Apply data for Code byte at location 000H to P1.0 to P1.7.
 5. Raise RST to 12V to enable programming.
 6. Pulse P3.2 once to program a byte in the PEROM array or the lock bits. The byte-write cycle is self-timed and typically takes 1.2 ms.
 7. To verify the programmed data, lower RST from 12V to logic "H" level and set pins P3.3 to P3.7 to the appropriate levels. Output data can be read at the port P1 pins.
 8. To program a byte at the next address location, pulse XTAL1 pin once to advance the internal address counter. Apply new data to the port P1 pins.
 9. Repeat steps 5 through 8, changing data and advancing the address counter for the entire 2K bytes array or until the end of the object file is reached.
 10. Power-off sequence:
set XTAL1 to "L"
set RST to "L"
Turn V_{CC} power off

Data Polling: The AT89C2051 features Data Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P1.7. Once the write cycle has been completed, true data is valid on all outputs, and

the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

Ready/Busy: The Progress of byte programming can also be monitored by the RDY/ \overline{BSY} output signal. Pin P3.1 is pulled low after P3.2 goes High during programming to indicate BUSY. P3.1 is pulled High again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed code data can be read back via the data lines for verification:

1. Reset the internal address counter to 000H by bringing RST from "L" to "H".
2. Apply the appropriate control signals for Read Code data and read the output data at the port P1 pins.
3. Pulse pin XTAL1 once to advance the internal address counter.
4. Read the next code data byte at the port P1 pins.
5. Repeat steps 3 and 4 until the entire array is read.

The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire PEROM array (2K bytes) and the two Lock Bits are erased electrically by using the proper combination of control signals and by holding P3.2 low for 10 ms. The code array is written with all "1"s in the Chip Erase operation and must be executed before any non-blank memory byte can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 000H, 001H, and 002H, except that P3.5 and P3.7 must be pulled to a logic low. The values returned are as follows.

(000H) = 1EH indicates manufactured by Atmel
(001H) = 21H indicates 89C2051

Programming Interface

Every code byte in the Flash array 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.

Flash Programming Modes

Mode		RST/VPP	P3.2/PROG	P3.3	P3.4	P3.5	P3.7
Write Code Data ⁽¹⁾⁽³⁾		12V		L	H	H	H
Read Code Data ⁽¹⁾		H	H	L	L	H	H
Write Lock	Bit - 1	12V		H	H	H	H
	Bit - 2	12V		H	H	L	L
Chip Erase		12V		H	L	L	L
Read Signature Byte		H	H	L	L	L	L

- Notes:
1. The internal PEROM address counter is reset to 000H on the rising edge of RST and is advanced by a positive pulse at XTAL 1 pin.
 2. Chip Erase requires a 10 ms $\overline{\text{PROG}}$ pulse.
 3. P3.1 is pulled Low during programming to indicate $\overline{\text{RDY/BSY}}$.

Figure 3. Programming the Flash Memory

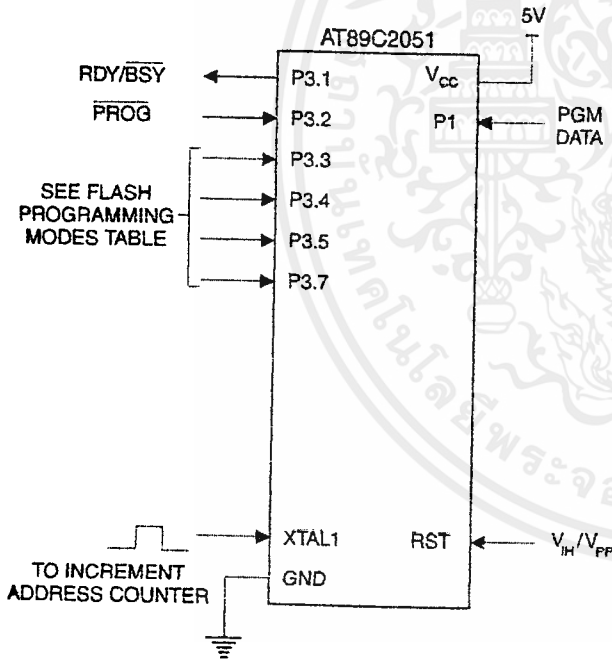
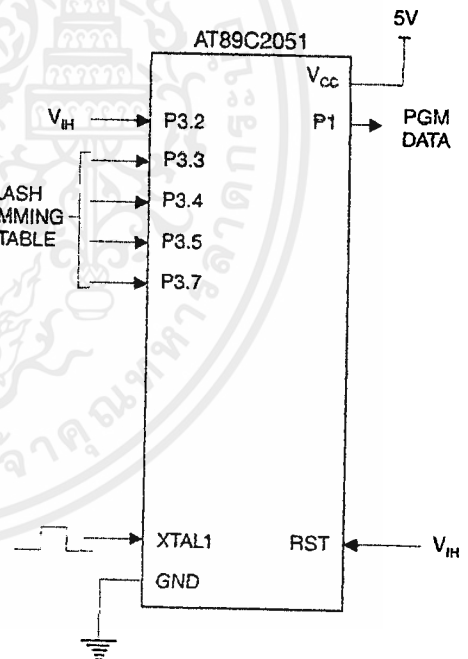


Figure 4. Verifying the Flash Memory





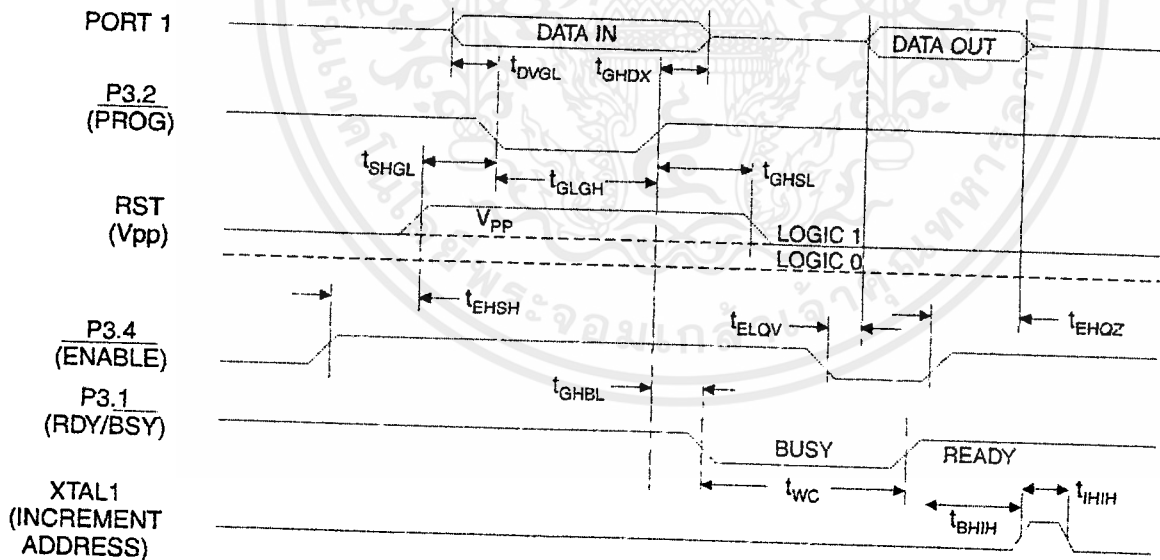
Flash Programming and Verification Characteristics

$T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5.0 \pm 10\%$

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current		250	μA
t_{DVGL}	Data Setup to $\overline{\text{PROG}}$ Low	1.0		μs
t_{GHDX}	Data Hold after $\overline{\text{PROG}}$	1.0		μs
t_{EHS}	P3.4 (ENABLE) High to V_{PP}	1.0		μs
t_{SHGL}	V_{PP} Setup to $\overline{\text{PROG}}$ Low	10		μs
t_{GHSL}	V_{PP} Hold after $\overline{\text{PROG}}$	10		μs
t_{GLGH}	$\overline{\text{PROG}}$ Width	1	110	μs
t_{ELQV}	ENABLE Low to Data Valid		1.0	μs
t_{EHQZ}	Data Float after ENABLE	0	1.0	μs
t_{GHBL}	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		50	ns
t_{WC}	Byte Write Cycle Time		2.0	ms
t_{BHIH}	RDY/BSY to Increment Clock Delay	1.0		μs
t_{IHIL}	Increment Clock High	200		ns

Note: 1. Only used in 12-volt programming mode.

Flash Programming and Verification Waveforms



AT89C2051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
DC Output Current.....	25.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

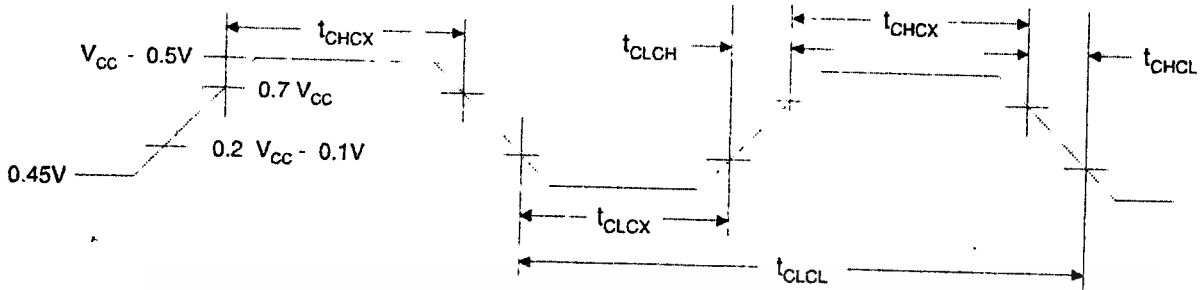
T_A = -40°C to 85°C, V_{CC} = 2.0V to 6.0V (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V _{IL}	Input Low-voltage		-0.5	0.2 V _{CC} - 0.1	V
V _{IH}	Input High-voltage	(Except XTAL1, RST)	0.2 V _{CC} + 0.9	V _{CC} + 0.5	V
V _{IH1}	Input High-voltage	(XTAL1, RST)	0.7 V _{CC}	V _{CC} + 0.5	V
V _{OL}	Output Low-voltage ⁽¹⁾ (Ports 1, 3)	I _{OL} = 20 mA, V _{CC} = 5V I _{OL} = 10 mA, V _{CC} = 2.7V		0.5	V
V _{OH}	Output High-voltage (Ports 1, 3)	I _{OH} = -80 μA, V _{CC} = 5V ± 10%	2.4		V
		I _{OH} = -30 μA	0.75 V _{CC}		V
		I _{OH} = -12 μA	0.9 V _{CC}		V
I _{IL}	Logical 0 Input Current (Ports 1, 3)	V _{IN} = 0.45V		-50	μA
I _{TL}	Logical 1 to 0 Transition Current (Ports 1, 3)	V _{IN} = 2V, V _{CC} = 5V ± 10%		-750	μA
I _{LI}	Input Leakage Current (Port P1.0, P1.1)	0 < V _{IN} < V _{CC}		±10	μA
V _{OS}	Comparator Input Offset Voltage	V _{CC} = 5V		20	mV
V _{CM}	Comparator Input Common Mode Voltage		0	V _{CC}	V
RRST	Reset Pull-down Resistor		50	300	KΩ
C _{IO}	Pin Capacitance	Test Freq. = 1 MHz, T _A = 25°C		10	pF
I _{CC}	Power Supply Current	Active Mode, 12 MHz, V _{CC} = 6V/3V		15/5.5	mA
		Idle Mode, 12 MHz, V _{CC} = 6V/3V P1.0 & P1.1 = 0V or V _{CC}		5/1	mA
		Power-down Mode ⁽²⁾	V _{CC} = 6V P1.0 & P1.1 = 0V or V _{CC}	100	μA
	V _{CC} = 3V P1.0 & P1.1 = 0V or V _{CC}	20	μA		

- Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
 Maximum I_{OL} per port pin: 20 mA
 Maximum total I_{OL} for all output pins: 80 mA
 If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
 2. Minimum V_{CC} for Power-down is 2V.



External Clock Drive Waveforms



External Clock Drive

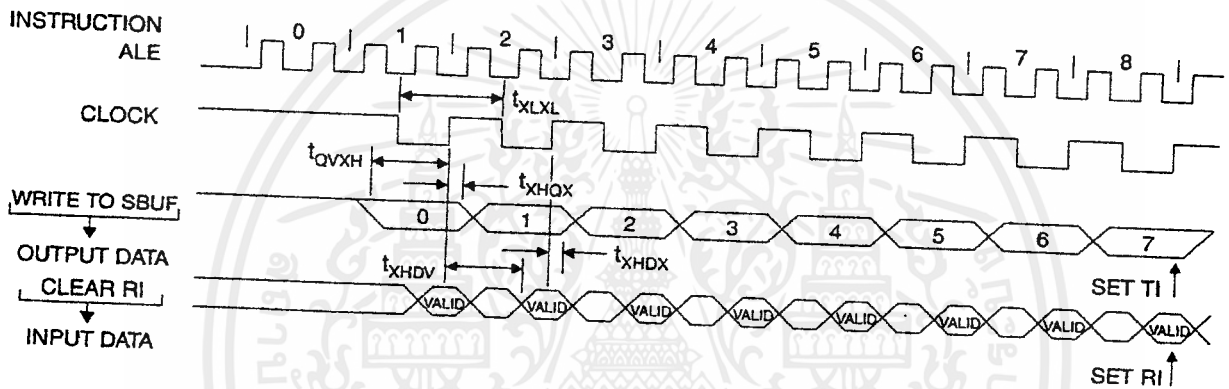
Symbol	Parameter	$V_{CC} = 2.7V \text{ to } 6.0V$		$V_{CC} = 4.0V \text{ to } 6.0V$		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency	0	12	0	24	MHz
t_{CLCL}	Clock Period	83.3		41.6		ns
t_{CHCX}	High Time	30		15		ns
t_{CLCX}	Low Time	30		15		ns
t_{CLCH}	Rise Time		20		20	ns
t_{CHCL}	Fall Time		20		20	ns

Serial Port Timing: Shift Register Mode Test Conditions

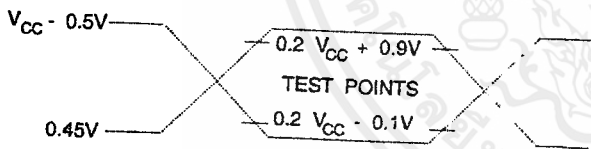
$V_{CC} = 5.0V \pm 20\%$; Load Capacitance = 80 pF

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		μs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}-133$		ns
t_{XHGX}	Output Data Hold after Clock Rising Edge	50		$2t_{CLCL}-117$		ns
t_{XHDX}	Input Data Hold after Clock Rising Edge	0		0		ns
t_{XHGV}	Clock Rising Edge to Input Data Valid		700			ns
					$10t_{CLCL}-133$	ns

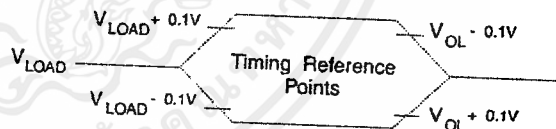
Shift Register Mode Timing Waveforms



AC Testing Input/Output Waveforms⁽¹⁾



Float Waveforms⁽¹⁾



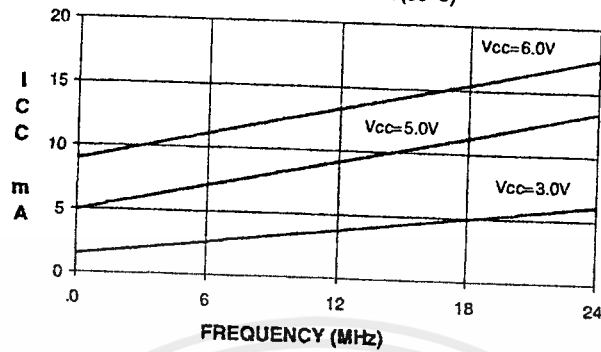
Note: 1. AC Inputs during testing are driven at $V_{CC} - 0.5V$ for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

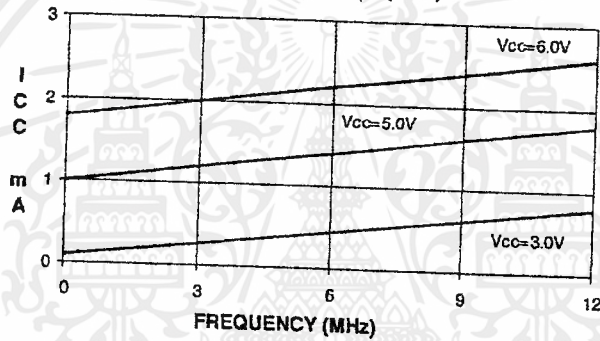




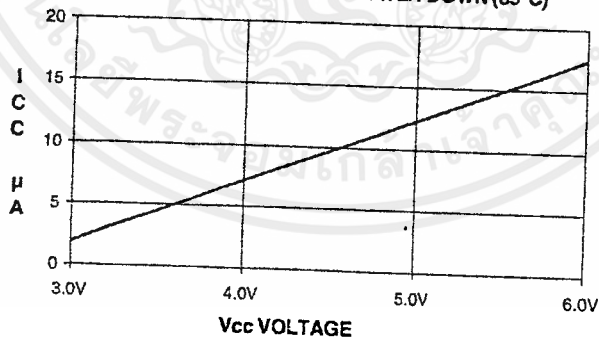
AT89C2051
TYPICAL I_{CC} - ACTIVE (85°C)



AT89C2051
TYPICAL I_{CC} - IDLE (85°C)



AT89C2051
TYPICAL I_{CC} vs. VOLTAGE - POWER DOWN (85°C)



- Notes:
1. XTAL1 tied to GND for I_{CC} (power-down)
 2. P1.0 and P1.1 = V_{CC} or GND
 3. Lock bits programmed

AT89C2051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	2.7V to 6.0V	AT89C2051-12PC	20P3	Commercial (0°C to 70°C)
		AT89C2051-12SC	20S	
24	4.0V to 6.0V	AT89C2051-12PI	20P3	Industrial (-40°C to 85°C)
		AT89C2051-12SI	20S	
24	4.0V to 6.0V	AT89C2051-24PC	20P3	Commercial (0°C to 70°C)
		AT89C2051-24SC	20S	
24	4.0V to 6.0V	AT89C2051-24PI	20P3	Industrial (-40°C to 85°C)
		AT89C2051-24SI	20S	



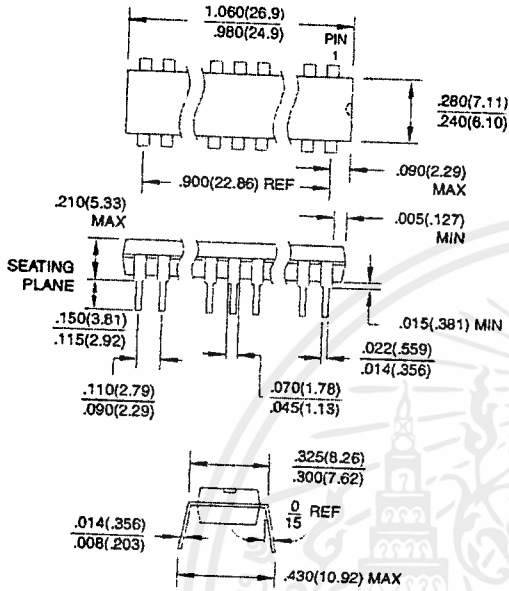
Package Type	
20P3	20-lead, 0.300" Wide, Plastic Dual In-line Package (PDIP)
20S	20-lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)



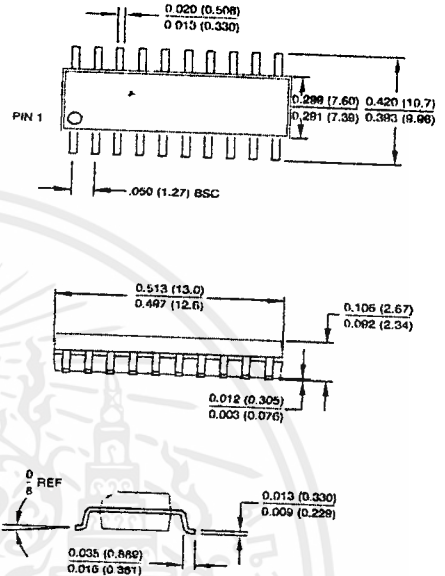
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Packaging Information

20P3, 20-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
 Dimensions in Inches and (Millimeters)
 JEDEC STANDARD MS-001 AD



20S, 20-lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)
 Dimensions in Inches and (Millimeters)



AT89C2051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Atmel Headquarters

Corporate Headquarters

2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe

Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686-677
FAX (44) 1276-686-697

Asia

Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan

Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs

1150 E. Cheyenne Min. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset

Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Fax-on-Demand

North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail

literature@atmel.com

Web Site

<http://www.atmel.com>

BBS

1-(408) 436-4309

© Atmel Corporation 2000.

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Marks bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

Terms and product names in this document may be trademarks of others.



Printed on recycled paper.

0368E-02/00/xM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAXIM

+5V-Powered, Multichannel RS-232 Drivers/Receivers

General Description

The MAX220-MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communication interfaces, particularly applications where $\pm 12V$ is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than $5\mu W$. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Applications

Portable Computers
Low-Power Modems
Interface Translation
Battery-Powered RS-232 Systems
Multidrop RS-232 Networks

Features

Superior to Bipolar

- ◆ Operate from Single +5V Power Supply (+5V and +12V—MAX231/MAX239)
- ◆ Low-Power Receive Mode in Shutdown (MAX223/MAX242)
- ◆ Meet All EIA/TIA-232E and V.28 Specifications
- ◆ Multiple Drivers and Receivers
- ◆ 3-State Driver and Receiver Outputs
- ◆ Open-Line Detection (MAX243)

Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

Ordering information continued at end of data sheet.

*Contact factory for dice specifications.

Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value (μF)	SHDN & Three-State	Rx Active in SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	4.7/10	No	—	120	Ultra-low-power, industry-standard pinout
MAX222	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	✓	120	MAX241 and receivers active in shutdown
MAX225	+5	5/5	0	—	Yes	✓	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes	—	120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	No	—	120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No	—	200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	—	No	—	120	No external caps
MAX233A	+5	2/2	0	—	No	—	200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	No	—	120	Replaces 1488
MAX235 (MAX205)	+5	5/5	0	—	Yes	—	120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes	—	120	Shutdown, three state
MAX237 (MAX207)	+5	5/3	4	1.0 (0.1)	No	—	120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	No	—	120	Replaces 1488 and 1489
MAX239 (MAX209)	+5 and +7.5 to +13.2	3/5	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; single-package solution for IBM PC serial port DIP or flatpack package
MAX240	+5	5/5	4	1.0	Yes	—	120	Complete IBM PC serial port
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes	—	120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✓	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No	—	200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	No	—	120	High slew rate
MAX245	+5	8/10	0	—	Yes	✓	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	8/10	0	—	Yes	✓	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/9	0	—	Yes	✓	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/8	4	1.0	Yes	✓	120	High slew rate, selective half-chip enables
MAX249	+5	6/10	4	1.0	Yes	✓	120	Available in quad flatpack package

MAXIM

Maxim Integrated Products 1

For free samples & the latest literature: <http://www.maxim-ic.com>, or phone 1-800-998-8800.
For small orders, phone 1-800-835-8769.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAX220-MAX249

+5V-Powered, Multichannel RS-232 Drivers/Receivers

ABSOLUTE MAXIMUM RATINGS—MAX220/222/232A/233A/242/243

Supply Voltage (V_{CC})	-0.3V to +6V	20-Pin Plastic DIP (derate 8.00mW/°C above +70°C)	..440mW
Input Voltages		16-Pin Narrow SO (derate 8.70mW/°C above +70°C)	...696mW
T_{IN}	-0.3V to ($V_{CC} - 0.3V$)	16-Pin Wide SO (derate 9.52mW/°C above +70°C)762mW
R_{IN} (Except MAX220)	±30V	18-Pin Wide SO (derate 9.52mW/°C above +70°C)762mW
R_{IN} (MAX220)	±25V	20-Pin Wide SO (derate 10.00mW/°C above +70°C)800mW
T_{OUT} (Except MAX220) (Note 1)	±15V	20-Pin SSOP (derate 8.00mW/°C above +70°C)640mW
T_{OUT} (MAX220)	±13.2V	16-Pin CERDIP (derate 10.00mW/°C above +70°C)800mW
Output Voltages		18-Pin CERDIP (derate 10.53mW/°C above +70°C)842mW
T_{OUT}	±15V	Operating Temperature Ranges	
R_{OUT}	-0.3V to ($V_{CC} + 0.3V$)	MAX2_AC_, MAX2_C_0°C to +70°C
Driver/Receiver Output Short Circuited to GND	Continuous	MAX2_AE_, MAX2_E_-40°C to +85°C
Continuous Power Dissipation ($T_A = +70°C$)		MAX2_AM_, MAX2_M_-55°C to +125°C
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C)842mW	Storage Temperature Range-65°C to +160°C
18-Pin Plastic DIP (derate 11.11mW/°C above +70°C)889mW	Lead Temperature (soldering, 10sec)+300°C

Note 1: Input voltage measured with T_{OUT} in high-impedance state, \overline{SHDN} or $V_{CC} = 0V$.

Note 2: For the MAX220, V+ and V- can have a maximum magnitude of 7V, but their absolute difference cannot exceed 13V.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243

($V_{CC} = +5V \pm 10\%$, C1-C4 = 0.1 μF , MAX220, C1 = 0.047 μF , C2-C4 = 0.33 μF , $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 TRANSMITTERS						
Output Voltage Swing	All transmitter outputs loaded with 3k Ω to GND		±5	±8		V
Input Logic Threshold Low				1.4	0.8	V
Input Logic Threshold High	All except MAX220		2	1.4		V
	MAX220: $V_{CC} = 5.0V$		2.4			
Logic Pull-Up/Input Current	All except MAX220, normal operation			5	40	μA
	$\overline{SHDN} = 0V$, MAX222/242, shutdown, MAX220			±0.01	±1	
Output Leakage Current	$V_{CC} = 5.5V$, $\overline{SHDN} = 0V$, $V_{OUT} = \pm 15V$, MAX222/242			±0.01	±10	μA
	$V_{CC} = \overline{SHDN} = 0V$, $V_{OUT} = \pm 15V$			±0.01	±10	
Data Rate	All except MAX220, normal operation			200	116	kb/s
Transmitter Output Resistance	$V_{CC} = V+ = V- = 0V$, $V_{OUT} = \pm 2V$		300	10M		Ω
Output Short-Circuit Current	$V_{OUT} = 0V$		±7	±22		mA
RS-232 RECEIVERS						
RS-232 Input Voltage Operating Range					±30	V
RS-232 Input Threshold Low	$V_{CC} = 5V$	All except MAX243 R_{2IN}	0.8	1.3		V
		MAX243 R_{2IN} (Note 2)	-3			
RS-232 Input Threshold High	$V_{CC} = 5V$	All except MAX243 R_{2IN}		1.8	2.4	V
		MAX243 R_{2IN} (Note 2)		-0.5	-0.1	
RS-232 Input Hysteresis	All except MAX243, $V_{CC} = 5V$, no hysteresis in shdn.		0.2	0.5	1	V
	MAX243			1		
RS-232 Input Resistance			3	5	7	k Ω
TTL/CMOS Output Voltage Low	$I_{OUT} = 3.2mA$			0.2	0.4	V
TTL/CMOS Output Voltage High	$I_{OUT} = -1.0mA$		3.5	$V_{CC} - 0.2$		V
TTL/CMOS Output Short-Circuit Current	Sourcing $V_{OUT} = GND$		-2	-10		mA
	Shrinking $V_{OUT} = V_{CC}$		10	30		

+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220-MAX249

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)

(V_{CC} = +5V ±10%, C₁-C₄ = 0.1µF, MAX220, C₁ = 0.047µF, C₂-C₄ = 0.33µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
TTL/CMOS Output Leakage Current	SHDN = V _{CC} or EN = V _{CC} (SHDN = 0V for MAX222), 0V ≤ V _{OUT} ≤ V _{CC}			±0.05	±10	µA
EN Input Threshold Low	MAX242			1.4	0.8	V
EN Input Threshold High	MAX242			2.0	1.4	V
Operating Supply Voltage			4.5		5.5	V
V _{CC} Supply Current (SHDN = V _{CC}), Figures 5, 6, 11, 19	No load	MAX220		0.5	2	mA
		MAX222/232A/233A/242/243		4	10	
	3kΩ load both inputs	MAX220		12		
		MAX222/232A/233A/242/243		15		
Shutdown Supply Current	MAX222/242	T _A = +25°C		0.1	10	µA
		T _A = 0°C to +70°C		2	50	
		T _A = -40°C to +85°C		2	50	
		T _A = -55°C to +125°C		35	100	
SHDN Input Leakage Current	MAX222/242				±1	µA
SHDN Threshold Low	MAX222/242			1.4	0.8	V
SHDN Threshold High	MAX222/242		2.0	1.4		V
Transition Slew Rate	C _L = 50pF to 2500pF, R _L = 3kΩ to 7kΩ, V _{CC} = 5V, T _A = +25°C, measured from +3V to -3V or -3V to +3V	MAX222/232A/233A/242/243	6	12	30	V/µs
		MAX220	1.5	3	30	
Transmitter Propagation Delay TLL to RS-232 (normal operation), Figure 1	t _{PHLT}	MAX222/232A/233A/242/243		1.3	3.5	µs
		MAX220		4	10	
	t _{PLHT}	MAX222/232A/233A/242/243		1.5	3.5	
		MAX220		5	10	
Receiver Propagation Delay RS-232 to TLL (normal operation), Figure 2	t _{PHLR}	MAX222/232A/233A/242/243		0.5	1	µs
		MAX220		0.6	3	
	t _{PLHR}	MAX222/232A/233A/242/243		0.6	1	
		MAX220		0.8	3	
Receiver Propagation Delay RS-232 to TLL (shutdown), Figure 2	t _{PHLS}	MAX242		0.5	10	µs
	t _{PLHS}	MAX242		2.5	10	
Receiver-Output Enable Time, Figure 3	t _{ER}	MAX242		125	500	ns
Receiver-Output Disable Time, Figure 3	t _{DR}	MAX242		160	500	ns
Transmitter-Output Enable Time (SHDN goes high), Figure 4	t _{ET}	MAX222/242, 0.1µF caps (includes charge-pump start-up)		250		µs
Transmitter-Output Disable Time (SHDN goes low), Figure 4	t _{DT}	MAX222/242, 0.1µF caps		600		ns
Transmitter + to - Propagation Delay Difference (normal operation)	t _{PHLT} - t _{PLHT}	MAX222/232A/233A/242/243		300		ns
		MAX220		2000		
Receiver + to - Propagation Delay Difference (normal operation)	t _{PHLR} - t _{PLHR}	MAX222/232A/233A/242/243		100		ns
		MAX220		225		

Note 3: MAX243 R_{2OUT} is guaranteed to be low when R_{2IN} is ≥ 0V or is floating.

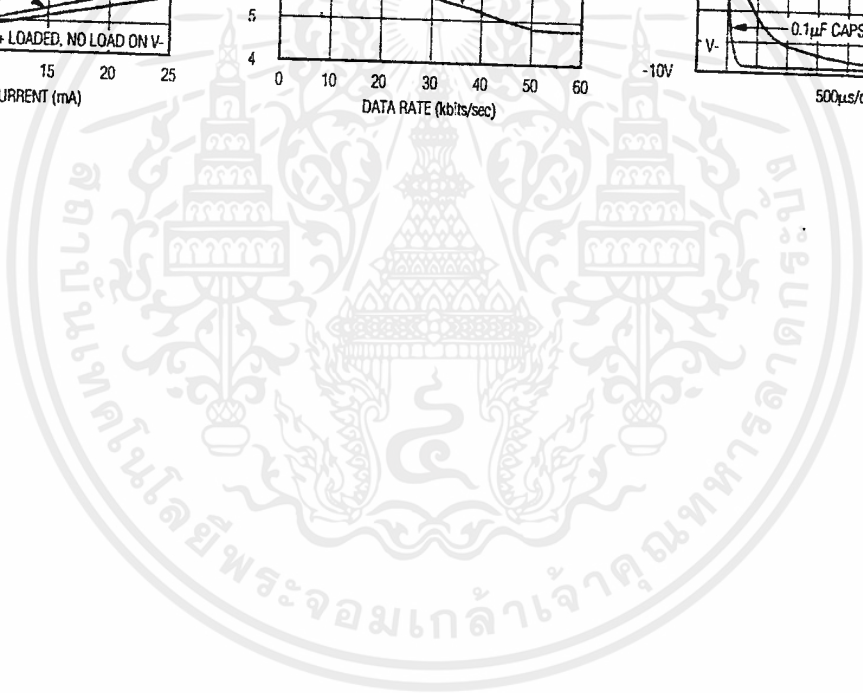
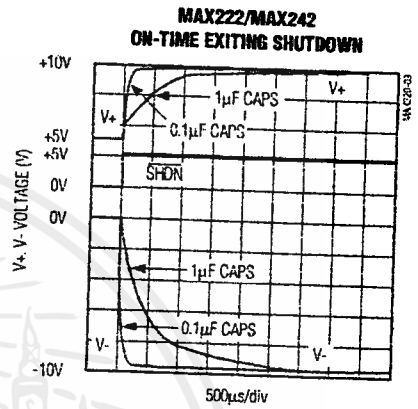
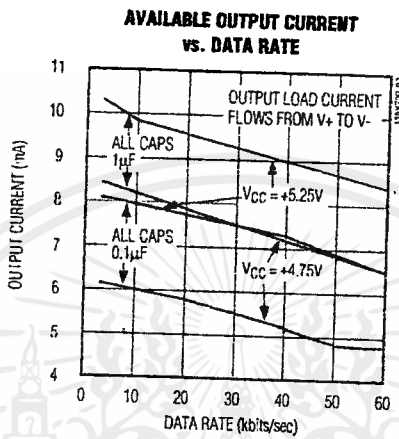
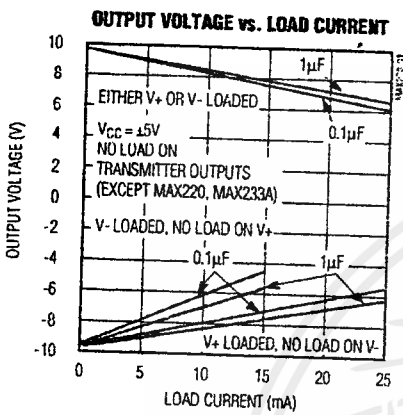
MAXIM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

+5V-Powered, Multichannel RS-232 Drivers/Receivers

Typical Operating Characteristics

MAX220/MAX222/MAX232A/MAX233A/MAX242/MAX243



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220-MAX249

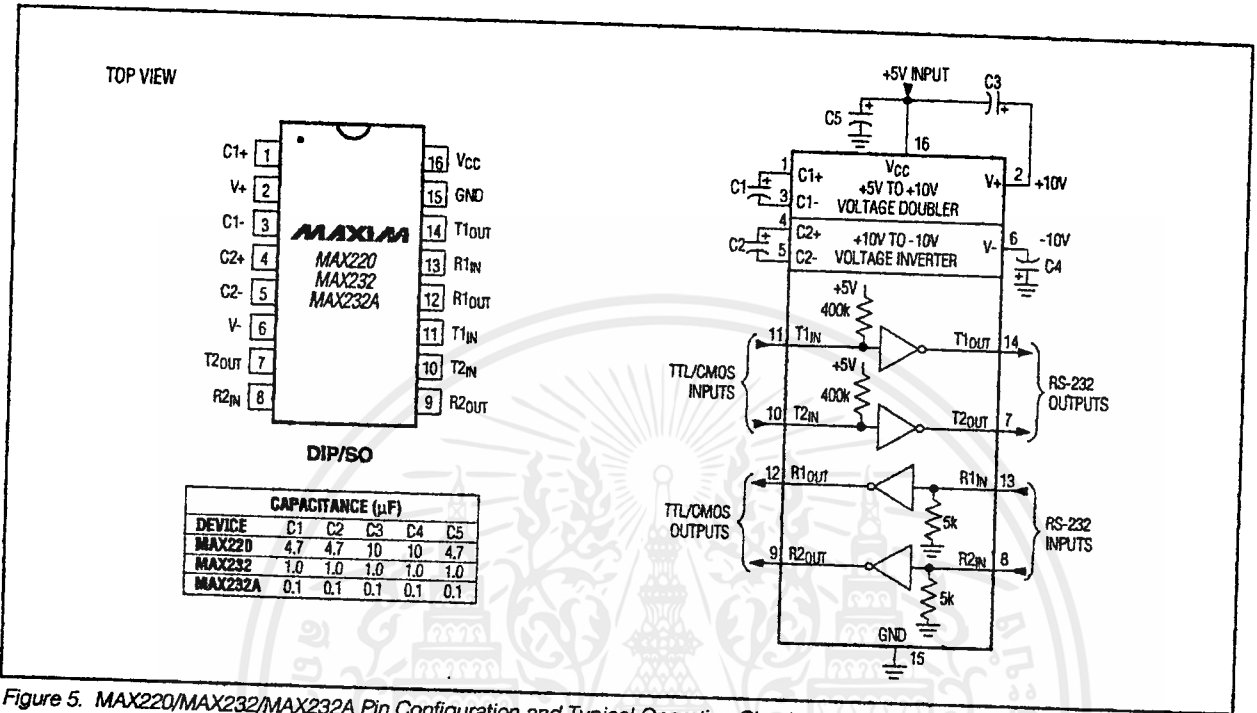


Figure 5. MAX220/MAX232/MAX232A Pin Configuration and Typical Operating Circuit

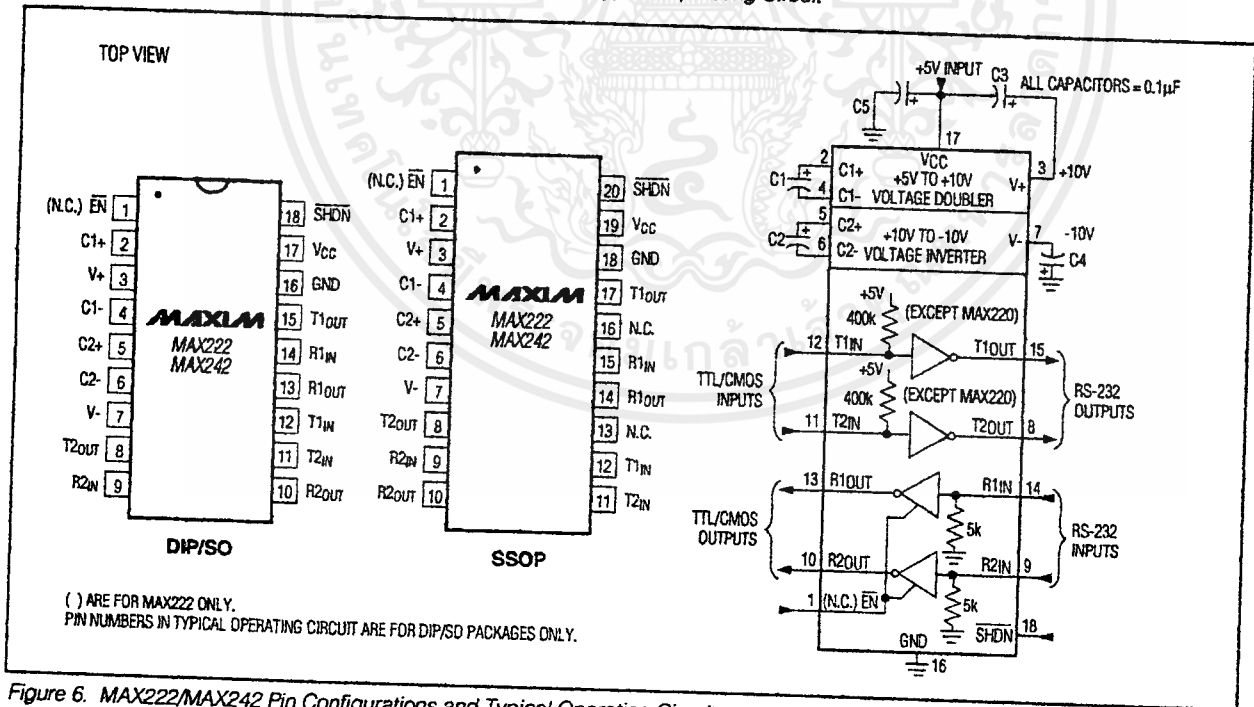


Figure 6. MAX222/MAX242 Pin Configurations and Typical Operating Circuit

MAXIM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

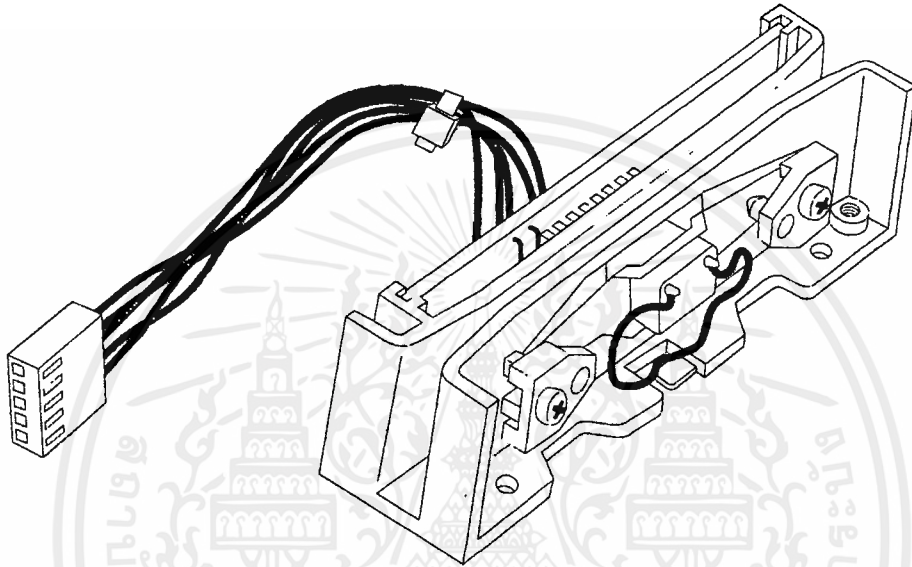


Figure 1-1. 90-millimeter Compatible Swipe Reader

SECTION 1. FEATURES AND SPECIFICATIONS

The 90-millimeter OEM Swipe Reader has a TTL level interface and is designed for use in retail, access control, and time and attendance environments. This Reader is in compliance with industry specifications, including ANSI/ISO Standards 7810, 7811-1 through -6, 7812, 7813, and AAMVA. The Reader can be customized. Bidirectional read capability is available.

CONFIGURATIONS

	Part Number	Read	Color
Integral Electronics, Single Track without Cover	21045001	Track 1	Black
	21045002	Track 2	Black
Integral Electronics, Dual Tracks without Cover	21045011	Tracks 1 & 2	Black
Integral Electronics, 3 Tracks without Cover	21045019	Tracks 1,2, & 3	Black

SPECIFICATIONS

IEC: Meets or Exceeds Requirements for:	IEC 1000-4-2 ESD (Electro Static Discharge) IEC 1000-4-3 Radiated EMC Field (2X requirement) IEC 1000-4-4 Electrical Fast Transient Burst requirement (transmission on I/O cable)
Flammability	Meets UL94V-0
Recording Method	Two-Frequency Coherent Phase (F2F)
Speed	Card speed through the unit may vary from: 2-125 ips at 75 bpi 2-60 ips at 210 bpi
Power Requirements	Single Track: 2.4 to 5.5VDC at 1mA, typical Dual Track: 2.4 to 5.5VDC at 2mA, typical 3 Tracks: 2.4 to 5.5VDC at 3mA, typical
Output Signal Levels	$V_{ol} = 0.4V$ at 2mA $V_{oh} = V_{cc} - 0.5V$ at -2mA
Operating Temperature	-30°C to 70°C
Operating Humidity	10% to 90% relative humidity
Life	300,000 passes Single Track 1,000,000 passes Multi-Track
Dimensions	Length: 3.54" (90.0mm) Height: 0.95" (24.13mm) Width: 0.88" (22.4mm)
Cable Length:	Single Track: 6" (150mm) Dual Track: 4" (101.6mm) 3 Track 5" (127mm)
Connector	See Section 2, Connectors
Colors available	Black, Standard

REFERENCE DOCUMENT

I/O Interface for TTL Swipe Readers, Technical Reference Manual, P/N 99875148



SECTION 2. INSTALLATION

This section consists of installation and checkout of the Reader.

MOUNTING

The dimensions for mounting without the cover are shown in Figure 2-1.

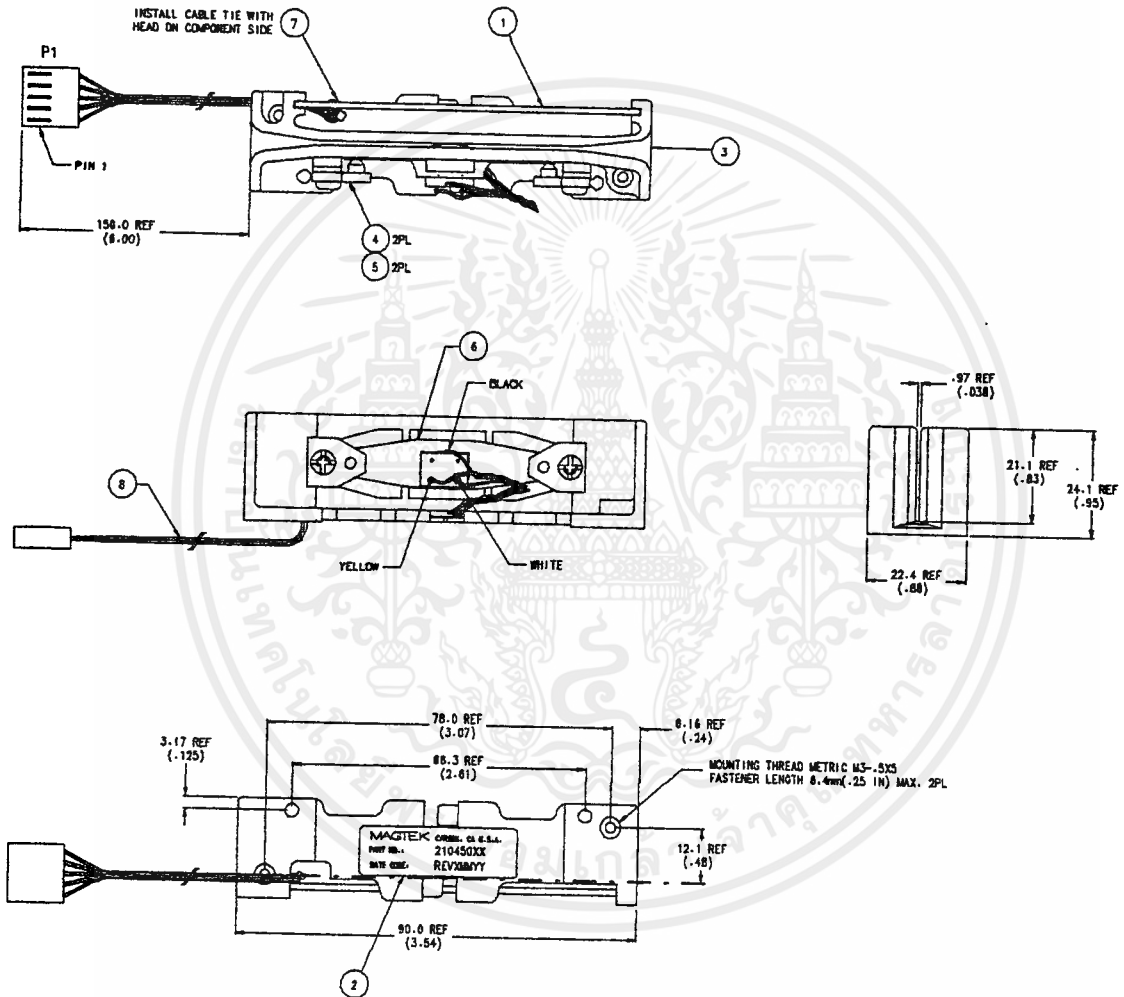


Figure 2-1. Reader Mounting Dimensions

CONNECTORS

Single Track I/O Connector is shown in Table 2-1, and the Dual Track I/O Connector is shown in Table 2-2.

Table 2-1. I/O Connector for Single Track, 5 Pin

	Pin Number	Color	Signal
Connector for Single Track, 5 Pin Molex 5 Pin 22-01-2951 0.100 inch Contact Spacing	1	Red	V_{cc}
	2	Black	GND
	3	Orange	$\overline{\text{DATA}}$
	4	Brown	$\overline{\text{STROBE}}$
	5	Green	$\overline{\text{CARD PRESENT}}$

Note: $V_{cc} = 2.4$ to 5.5 VDC

Table 2-2. I/O Connector for Dual Track, 7 Pin

	Pin Number	Color	Signal
Connector for Dual Track, 7 Pin Molex 7 Pin 22-01-2071 0.100 inch Contact Spacing	1	Red	V_{cc}
	2	Black	GND
	3	Orange	$\overline{\text{DATA (Tk 2)}}$
	4	Brown	$\overline{\text{STROBE (Tk 2)}}$
	5	Green	$\overline{\text{CARD PRESENT}}$
	6	Yellow	$\overline{\text{STROBE (Tk 1)}}$
	7	White	$\overline{\text{DATA (Tk 1)}}$

Note: $V_{cc} = 2.4$ to 5.5 VDC

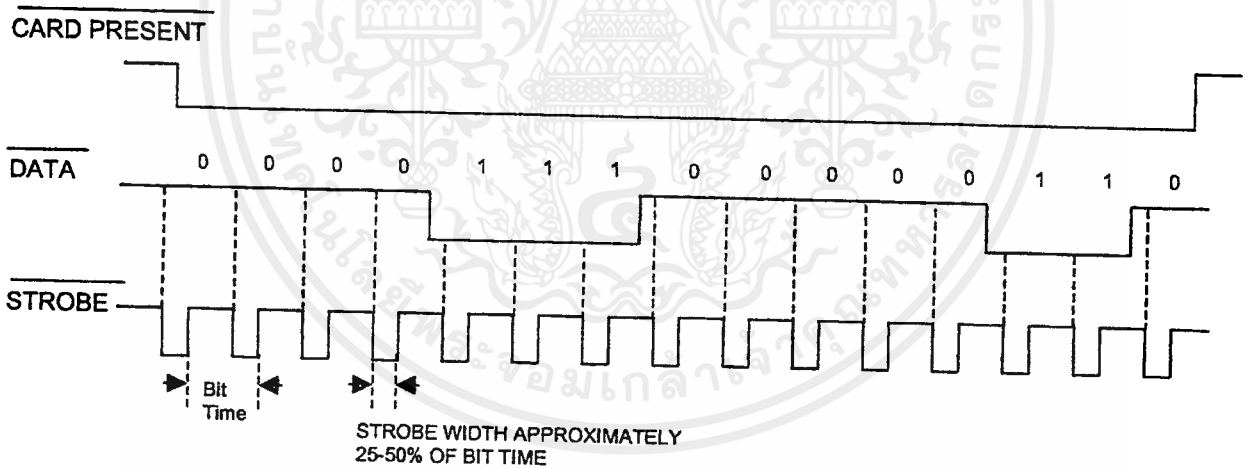
Table 2-3. I/O Connector for 3 Track, 9 Pin

Pin Number	Color	Signal
1	Red	V_{cc}
2	Black	GND
3	Yellow	DATA (Tk 2)
4	Green	STROBE (Tk 2)
5	White	CARD PRESENT
6	Blue	STROBE (Tk 1)
7	Brown	DATA (Tk 1)
8	Gray	STROBE (Tk 3)
9	Orange	DATA (Tk 3)

Connector for 3 Track, 9 Pin
Molex 9 Pin
22-01-2091
0.100 inch Contact Spacing

Note: $V_{cc} = 2.4$ to 5.5 VDC

TIMING



Notes:

1. Time out of the CARD PRESENT signal occurs approximately 150 ms after the last strobe transition.
2. DATA is valid 1.0μ sec before the negative edge of STROBE.
3. 16 or 17 head flux reversals for high density configuration.

Figure 2-2. Timing

DATA

The Data signal is valid while the strobe is low. If the Data signal is high, the bit is a zero. If the Data signal is low, the bit is a one.

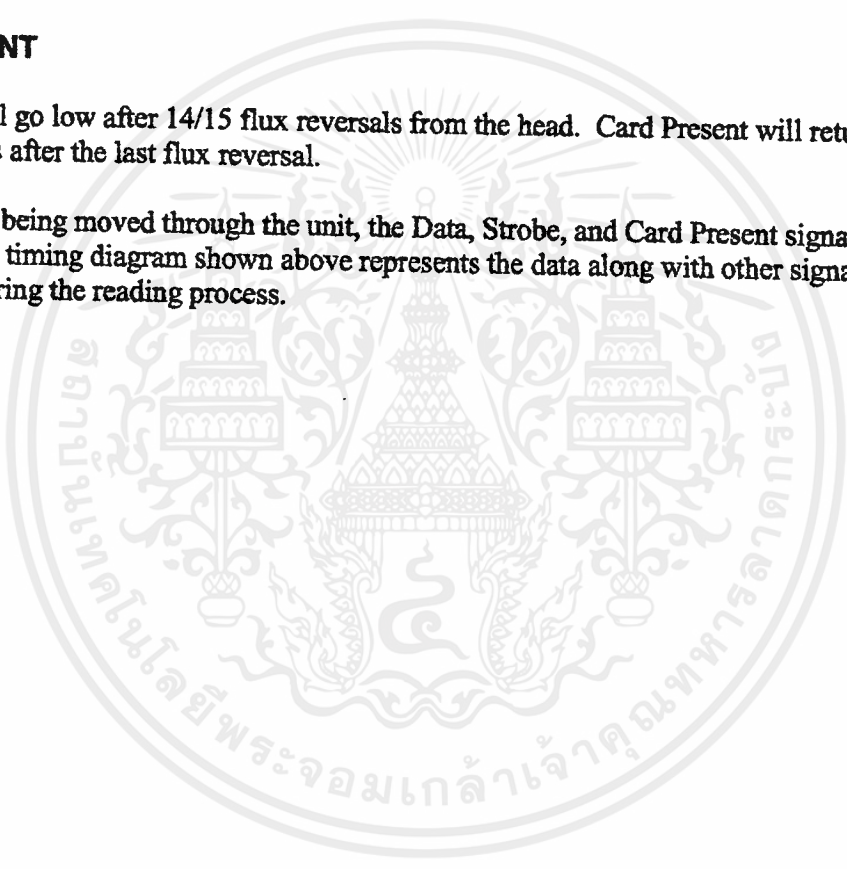
STROBE

The Strobe signal indicates when Data is valid. It is recommended that Data be loaded by the user with the leading edge (negative) of the Strobe.

CARD PRESENT

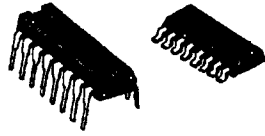
Card Present will go low after 14/15 flux reversals from the head. Card Present will return high 150 milliseconds after the last flux reversal.

When no card is being moved through the unit, the Data, Strobe, and Card Present signals are high. The signal timing diagram shown above represents the data along with other signals that are generated during the reading process.



MAGTEK

21006515 DIP
21006516 SMD (SOIC)



Magnetic Stripe F/2F Read/Decode Integrated Circuit

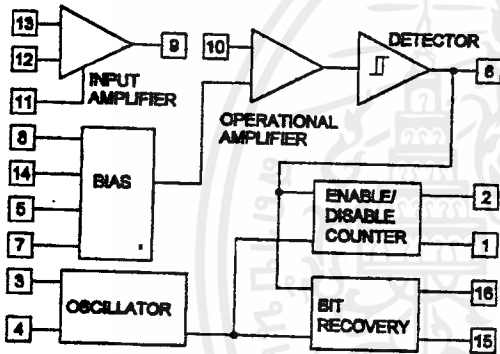
General Information

The F/2F Read/Decode Integrated Circuit is intended for use in recovering F/2F encoded data received from a magnetic head.

Features

- 150 to more than 12,000 F/2F bits per second
- Low power: Full operation from 2.4 to 5.5 volts. Current below 2.0 mA.
- Recovers Data with as much as 30% dropout of amplitude.
- Accepts amplitudes from below 20% of ISO reference voltage to more than 250% of ISO reference voltage.
- Meets or exceeds the requirement for:
 - IEC 1000-4-2 ESD (Electro Static Discharge)
 - IEC 1000-4-3 Radiated EMC Field (2x requirement)
 - IEC 1000-4-4 Electrical Fast Transient Burst requirement (transmissions on I/O cable)

Functional Block Diagram



Functional Description

The F/2F Read/Decode I.C. will recover clock and data signals from an F/2F data stream generated from a magnetic head. The I.C. will function for data rates from 150 to more than 12,000 bits per second. Acquisition and tracking of the data within this range is automatic. The F/2F Read/Decode I.C. is composed of three functional sections:

- Signal Conditioning and Detection
- Bit Recovery
- Enable/Disable Counters

The signal conditioning and detection section amplifies and filters the signal from the magnetic head, rejects common mode noise, and detects the signal peaks. Other features include protection against certain waveform distortions that may be present in the signal.

The enable/disable counters provide initialization for the recovery section. These counters initialize both the bit recovery and the signal conditioning and detection sections.

The Bit Recovery section locks onto the data rate and performs the recovery of individual bits from the F/2F data stream.

Absolute Maximum Ratings (Non-Operating)

Supply Voltage 7.0 Volts
Voltage Input Range 0 to VCC
Output Sink Current 10 mA
Internal Power Dissipation 100 mW
Storage Temperature Range -55 to 150°C
Lead Soldering (10 Sec.) 260°C

Electrical Characteristics

Unit operates from 2.4 to 5.5 Volts (VCC) and -30° to 70°C
Test conditions: VCC = 5.00 Volts, Ambient Temperature = 25°C

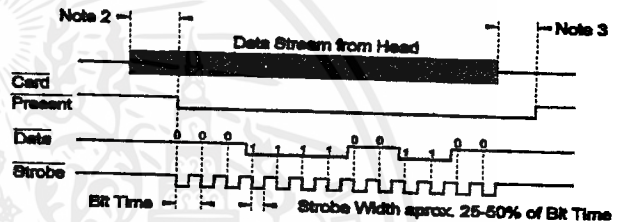
Parameter	Condition	Min.	Max.	Typ.	Units
Device Current			2		mA
Logic Low Out (VOL)	at +2.0 mA		0.4		Volts
Logic High Out (VOH)	at -2.0 mA	VCC - 0.5			Volts
Oscillator Frequency	(1)			1.2	MHZ

Table 1: Electrical Characteristics.

Notes:

1. TTL/CMOS compatible. Outputs covered include the following: Card Present, Read Data, Strobe and F/2F.
2. Frequency measured using R2 and C3 component values. Refer to recommended circuit for required values.

Signal Timing Diagram



Notes:

1. Card Present, Data, and Strobe are negative true logic.
2. Card Present goes low after 14-15 head flux reversals.
3. Card Present returns to high level approximately 150mS after the last flux transition.
4. Data is valid 1.0µS (min.) before the leading negative edge of strobe and remains valid until approximately 1.0µS before the next STROBE.

DATA

The DATA signal is valid while the STROBE is low. If the DATA signal is high, the bit is a zero. If the DATA signal is low, the bit is a one.

STROBE

The STROBE signal indicates when DATA is valid. It is recommended that DATA be loaded by the user with the leading negative edge of the STROBE.

CARD PRESENT

CARD PRESENT will go low after 14-15 flux reversals from the head. It will return high if Reset or about 150 milliseconds after the last flux reversal. The CARD PRESENT signal can be tied together with other card present signals from more than one IC, however use only one pull-up resistor (R4). E.g., if this is a 3 track reader, all 3 CARD PRESENTS would be tied together using a single 10K resistor.

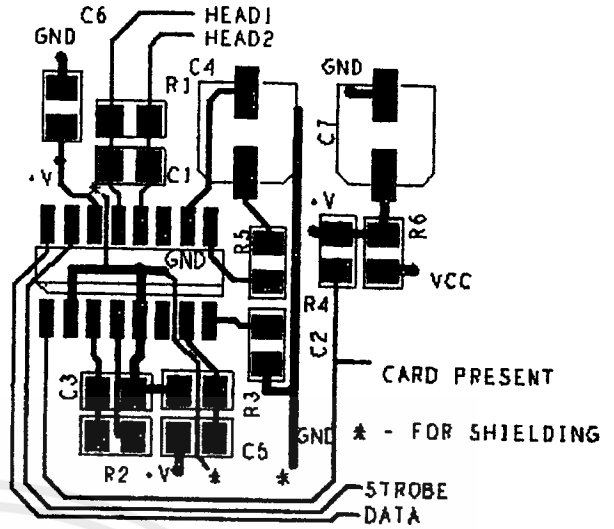
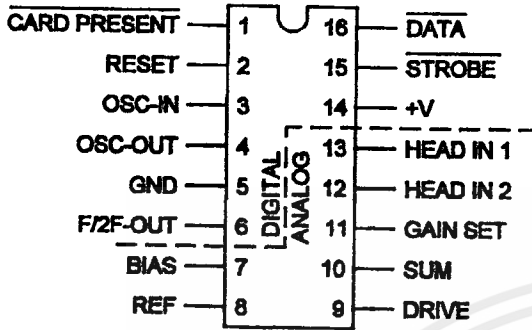
When no card is being moved through the unit, the DATA, STROBE and CARD PRESENT signals are high.

The signal timing diagram shown above represents the data along with other signals that are generated during the reading process.

Reset Feature

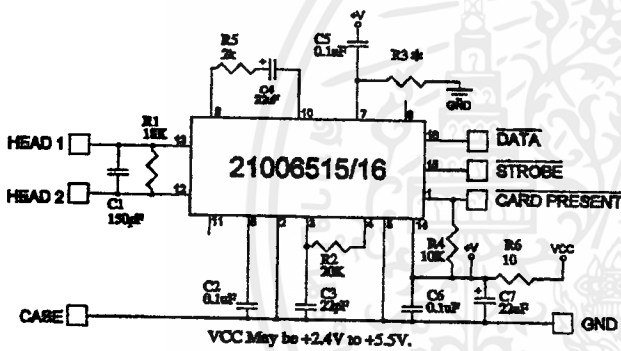
In normal operation, the I.C. resets itself approximately 150 milliseconds after the last flux reversal from the magnetic head. Reset may be forced by applying a pulse of 1.0 to 100 microseconds to the RESET pin. The positive edge of the pulse will reset the I.C.

Connection Diagram (Pin Outs)



Recommended Circuit

This circuit is intended for use in systems employing F/2F data such as ID cards that conform to ISO 7811-2 or 7811-6.



* R3 has different values for Rev B and Rev G. The table below shows the value of R3.

	Rev B	Rev G
R3	470K	1.5M

Table 2: Value of R3.

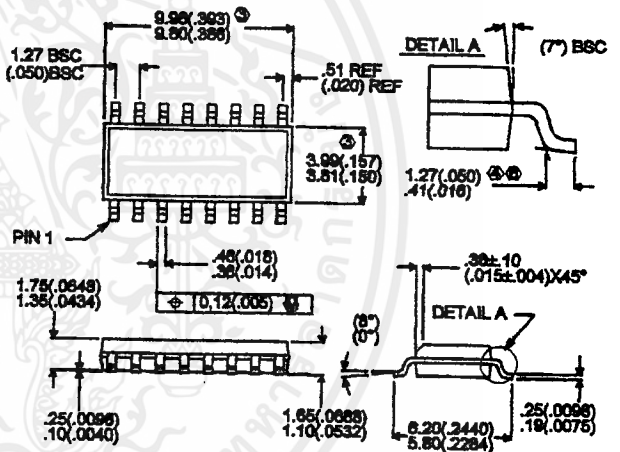
Printed Circuit Board Layout Requirements

The I.C. is a combined digital and analog system. The analog signal pins are very low level. Good layout practice requires that there be a separation of head and other analog signals from the digital outputs. The digital signals are: DATA, STROBE, CARD PRESENT, F/2F OUT, OSC-OUT and OSC-IN. The analog signals in order of sensitivity are HEAD in 1&2, SUM, DRIVE, GSR and BIAS. The digital signals must not pass near the first four analog signals. The layout below illustrates the requirements.

Caution

Keep Data, Strobe, and Card Present signals away from Head 1 and Head 2 signals.

SOIC Physical Dimensions



Notes: Unless otherwise specified

1. Dimensions in mm (in.)
2. Dimension and tolerancing per ANSI Y14.5m-1982
3. Reference Datum (Mold flash not included).
4. Length of terminal for soldering.
5. Flammability rating UL 94V-0
6. Thermal resistance JA=225, JC=190

MAGTEK

20725 South Annalee, Carson, CA. 90746
 Phone: (310) 631-8602, Help Line (888) 624-8350
 Fax (310) 631-3956
<http://www.magtek.com>

P/N 99821066-8, 9-29-99 © Copyright 1998 - 99, Mag-Tek, Inc.

Application Note - Character Conversion

Track 1 Character Set

Card Data ASCII

Character	P	B ₆	B ₅	B ₄	B ₃	B ₂	B ₁	P0	P1	EP	OP
Space	1	0	0	0	0	0	0	20	A0	A0	20
! (ND)	0	0	0	0	0	0	1	21	A1	21	A1
" (ND)	0	0	0	0	0	1	0	22	A2	22	A2
# (OG)	1	0	0	0	0	1	1	23	A3	A3	23
\$	0	0	0	0	1	0	0	24	A4	24	A4
% (SS)	1	0	0	0	1	0	1	25	A5	A5	25
& (ND)	1	0	0	0	1	1	0	26	A6	A6	26
' (ND)	0	0	0	0	1	1	1	27	A7	27	A7
(0	0	0	1	0	0	0	28	A8	28	A8
)	1	0	0	1	0	0	1	29	A9	A9	29
* (ND)	1	0	0	1	0	1	0	2A	AA	AA	2A
+ (ND)	0	0	0	1	0	1	1	2B	AB	2B	AB
, (ND)	1	0	0	1	1	0	0	2C	AC	AC	2C
-	0	0	0	1	1	0	1	2D	AD	2D	AD
.	0	0	0	1	1	1	0	2E	AE	2E	AE
/	1	0	0	1	1	1	1	2F	AF	AF	2F
0	0	0	1	0	0	0	0	30	B0	30	B0
1	1	0	1	0	0	0	1	31	B1	B1	31
2	1	0	1	0	0	1	0	32	B2	B2	32
3	0	0	1	0	0	1	1	33	B3	33	B3
4	1	0	1	0	1	0	0	34	B4	B4	34
5	0	0	1	0	1	0	1	35	B5	35	B5
6	0	0	1	0	1	1	0	36	B6	36	B6
7	1	0	1	0	1	1	1	37	B7	B7	37
8	1	0	1	1	0	0	0	38	B8	B8	38
9	0	0	1	1	0	0	1	39	B9	39	B9
: (ND)	0	0	1	1	0	1	0	3A	BA	3A	BA
; (ND)	1	0	1	1	0	1	1	3B	BB	BB	3B
< (ND)	0	0	1	1	1	0	0	3C	BC	3C	BC
= (ND)	1	0	1	1	1	0	1	3D	BD	BD	3D
> (ND)	1	0	1	1	1	1	0	3E	BE	BE	3E
? (ES)	0	0	1	1	1	1	1	3F	BF	3F	BF

Card Data ASCII

Character	P	B ₆	B ₅	B ₄	B ₃	B ₂	B ₁	P0	P1	EP	OP
@ (ND)	0	1	0	0	0	0	0	40	C0	C0	40
A	1	1	0	0	0	0	1	41	C1	41	C1
B	1	1	0	0	0	1	0	42	C2	42	C2
C	0	1	0	0	0	1	1	43	C3	C3	43
D	1	1	0	0	1	0	0	44	C4	44	C4
E	0	1	0	0	1	0	1	45	C5	C5	45
F	0	1	0	0	1	1	0	46	C6	C6	46
G	1	1	0	0	1	1	1	47	C7	47	C7
H	1	1	0	1	0	0	0	48	C8	48	C8
I	0	1	0	1	0	0	1	49	C9	C9	49
J	0	1	0	1	0	1	0	4A	CA	CA	4A
K	1	1	0	1	0	1	1	4B	CB	4B	CB
L	0	1	0	1	1	0	0	4C	CC	CC	4C
M	1	1	0	1	1	0	1	4D	CD	4D	CD
N	1	1	0	1	1	1	0	4E	CE	4E	CE
O	0	1	0	1	1	1	1	4F	CF	CF	4F
P	1	1	1	0	0	0	0	50	D0	50	D0
Q	0	1	1	0	0	0	1	51	D1	D1	51
R	0	1	1	0	0	1	0	52	D2	D2	52
S	1	1	1	0	0	1	1	53	D3	53	D3
T	0	1	1	0	1	0	0	54	D4	D4	54
U	1	1	1	0	1	0	1	55	D5	55	D5
V	1	1	1	0	1	1	0	56	D6	56	D6
W	0	1	1	0	1	1	1	57	D7	D7	57
X	0	1	1	1	0	0	0	58	D8	D8	58
Y	1	1	1	1	0	0	1	59	D9	59	D9
Z	1	1	1	1	0	1	0	5A	DA	5A	DA
[(ND)	0	1	1	1	0	1	1	5B	DB	DB	5B
\ (ND)	1	1	1	1	1	0	0	5C	DC	5C	DC
] (ND)	1	1	1	1	1	0	0	5D	DD	DD	5D
^ (FS)	0	1	1	1	1	1	0	5E	DE	DE	5E
_ (ND)	1	1	1	1	1	1	1	5F	DF	5F	DF

Track 2 and 3 Character Set

Card Data ASCII

Character	P	B ₄	B ₃	B ₂	B ₁	P0	P1	EP	OP
0	1	0	0	0	0	30	B0	30	B0
1	0	0	0	0	1	31	B1	B1	31
2	0	0	0	1	0	32	B2	B2	32
3	1	0	0	1	1	33	B3	33	B3
4	0	0	1	0	0	34	B4	B4	34
5	1	0	1	0	1	35	B5	35	B5
6	1	0	1	1	0	36	B6	36	B6
7	0	0	1	1	1	37	B7	B7	37

Card Data ASCII

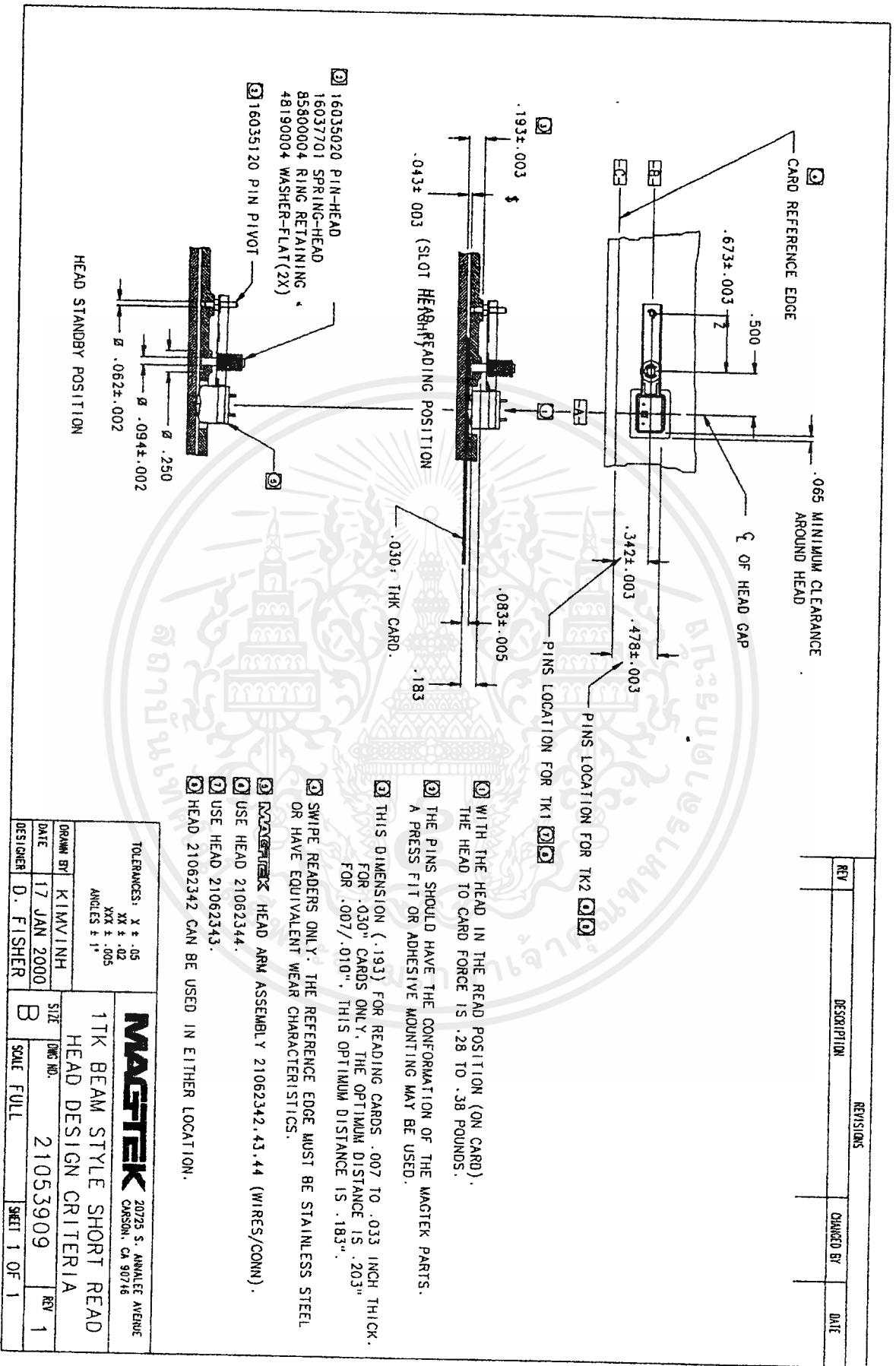
Character	P	B ₄	B ₃	B ₂	B ₁	P0	P1	EP	OP	Hex Character
8	0	1	0	0	0	38	B8	B8	38	
9	1	1	0	0	1	39	B9	39	B9	
: (AS)	1	1	0	1	0	3A	BA	3A	BA	A
; (SS)	0	1	0	1	1	3B	BB	BB	3B	B
< (ND)	1	1	1	0	0	3C	BC	3C	BC	C
= (FS)	0	1	1	0	1	3D	BD	BD	3D	D
> (ND)	0	1	1	1	0	3E	BE	BE	3E	E
? (ES)	1	1	1	1	1	3F	BF	3F	BF	F

P = Parity
 EP = Even Parity
 OP = Odd Parity
 P0 = Parity bit set to 0
 P1 = Parity bit set to 1
 SS = Start Sentinel

ES = End Sentinel
 FS = Field Separator
 AS = Account Separator (Track 3 only)
 ND = Character Not Defined by Credit Card Standards
 OG = Optional Graphic

MAGTEK

20725 South Annalee, Carson, CA. 90746
 Phone: (310) 631-8602, Help Line (888)624-8350
 Fax (310) 631-3956
<http://www.magtek.com>
 P/N 99875065-2, 1/00 © Copyright 2000, Mag-Tek, Inc.



- ① 16035020 PIN-HEAD
16037701 SPRING-HEAD
85800004 RING RETAINING
48190004 WASHER-FLAT (2X)
- ② 16035120 PIN PIVOT

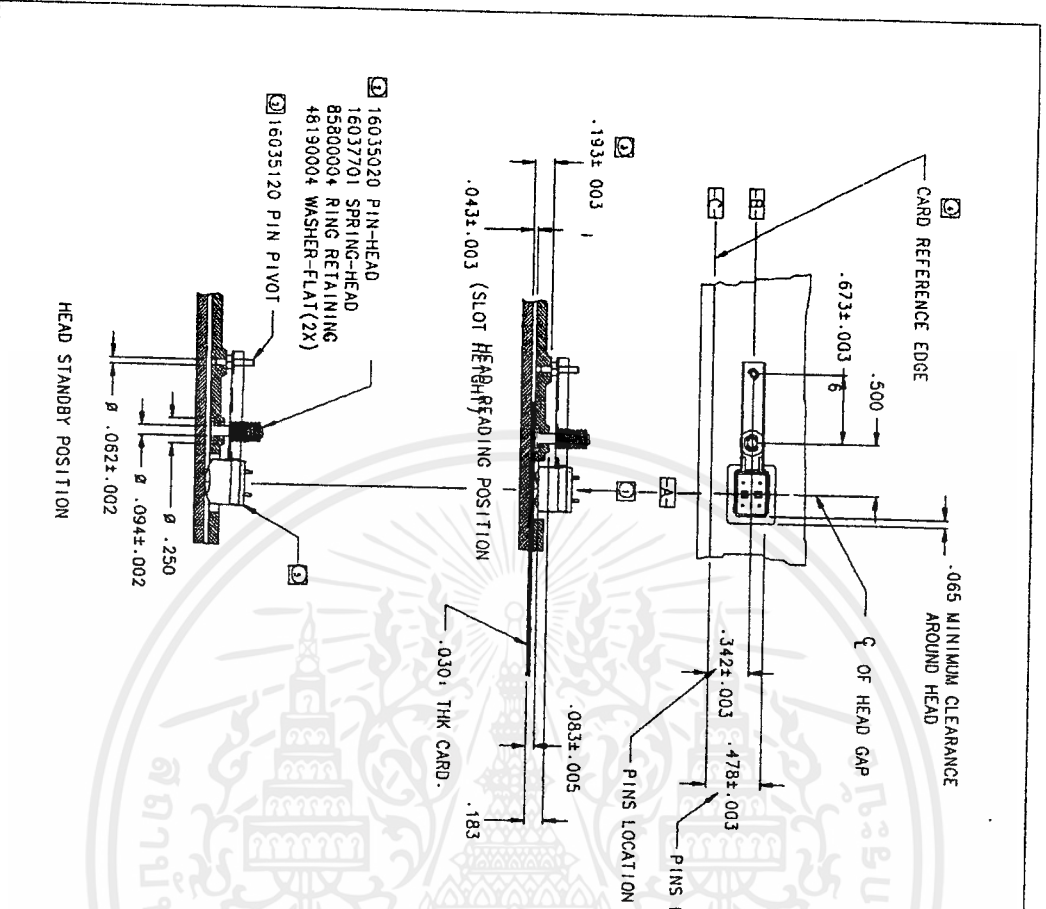
- ① WITH THE HEAD IN THE READ POSITION (ON CARD), THE HEAD TO CARD FORCE IS .28 TO .38 POUNDS.
- ② THE PINS SHOULD HAVE THE CONFORMATION OF THE MAGTEK PARTS. A PRESS FIT OR ADHESIVE MOUNTING MAY BE USED.
- ③ THIS DIMENSION (.193) FOR READING CARDS .007 TO .033 INCH THICK. FOR ".030" CARDS ONLY, THE OPTIMUM DISTANCE IS ".203" FOR ".007/.010", THIS OPTIMUM DISTANCE IS ".183".
- ④ SWIPE READERS ONLY. THE REFERENCE EDGE MUST BE STAINLESS STEEL OR HAVE EQUIVALENT WEAR CHARACTERISTICS.
- ⑤ ~~MAGTEK~~ HEAD ARM ASSEMBLY 21062342, 43, 44 (WIRES/CONN).
- ⑥ USE HEAD 21062344.
- ⑦ USE HEAD 21062343.
- ⑧ HEAD 21062342 CAN BE USED IN EITHER LOCATION.

REV		DESCRIPTION		REVISIONS		CHANGED BY	DATE

TOLERANCES: X ± .05 XX ± .02 XXX ± .005 ANGLES ± 1°		MAGSTEK 20729 S. ANNALEE AVENUE CARSON, CA 90716	
DRAWN BY	KIMVINH	SIZE	1TK BEAM STYLE SHORT READ
DATE	17 JAN 2000	DWG NO.	HEAD DESIGN CRITERIA
DESIGNER	D. FISHER	SCALE	FULL
			21053909
			REV 1
			SHEET 1 OF 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REV	DESCRIPTION	CHANGED BY	DATE

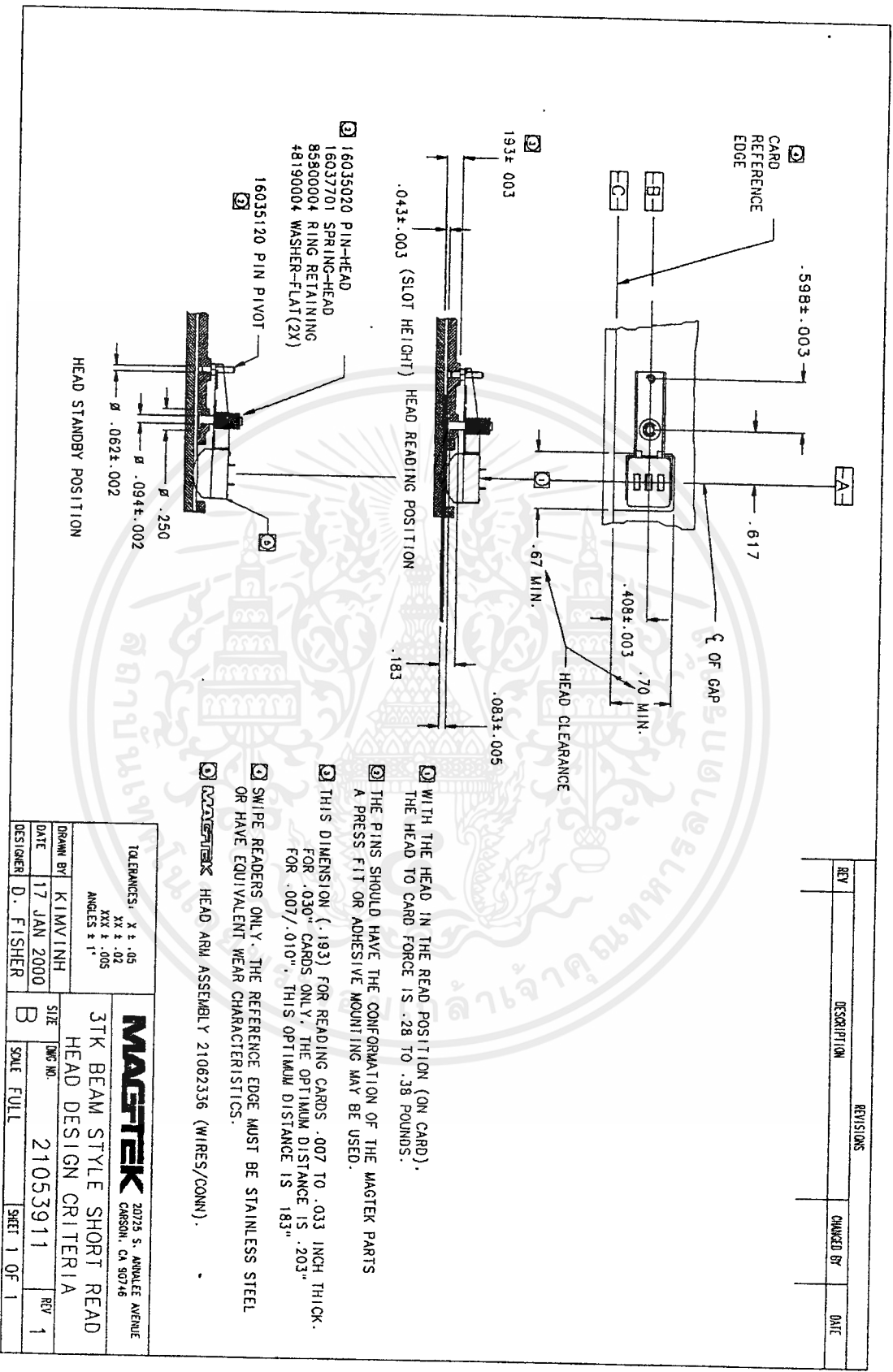


- 16035020 PIN-HEAD
16037701 SPRING-HEAD
85600004 RING RETAINING
48190004 WASHER-FLAT (2X)
- 16035120 PIN PIVOT

- WITH THE HEAD IN THE READ POSITION (ON CARD), THE HEAD TO CARD FORCE IS .28 TO .38 POUNDS.
- THE PINS SHOULD HAVE THE CONFORMATION OF THE MAGTEK PARTS. A PRESS FIT OR ADHESIVE MOUNTING MAY BE USED.
- THIS DIMENSION (.193) FOR READING CARDS .007 TO .033 INCH THICK. FOR .030" CARDS ONLY, THE OPTIMUM DISTANCE IS .203" FOR .007"/.010". THIS OPTIMUM DISTANCE IS .183".
- SWIPE READERS ONLY. THE REFERENCE EDGE MUST BE STAINLESS STEEL OR HAVE EQUIVALENT WEAR CHARACTERISTICS.
- MAXIMUM HEAD ARM ASSEMBLY 21062334 (WIRES/CONN).

TOLERANCES: X ± .05 XX ± .02 XXX ± .005 ANGLES ± 1°		MAGTEK 20725 S. ANNALEE AVENUE CARSON, CA 90716
DESIGNED BY D. FISHER	DATE 17 JAN 2000	
2TK BEAM STYLE SHORT READ HEAD DESIGN CRITERIA		SCALE FULL
DRWING BY KIM VINH	DATE 17 JAN 2000	SHEET 1 OF 1
SIZE B	REV NO. 21053910	REV 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



REV	DESCRIPTION	CHANGED BY	DATE

- ① WITH THE HEAD IN THE READ POSITION (ON CARD), THE HEAD TO CARD FORCE IS .28 TO .38 POUNDS.
- ② THE PINS SHOULD HAVE THE CONFORMATION OF THE MAGTEK PARTS A PRESS FIT OR ADHESIVE MOUNTING MAY BE USED.
- ③ THIS DIMENSION (.183) FOR READING CARDS .007 TO .033 INCH THICK. FOR .030" CARDS ONLY. THE OPTIMUM DISTANCE IS .203" FOR .007/.010". THIS OPTIMUM DISTANCE IS .183"
- ④ SWIPE READERS ONLY. THE REFERENCE EDGE MUST BE STAINLESS STEEL OR HAVE EQUIVALENT WEAR CHARACTERISTICS.
- ⑤ ~~DATA~~ HEAD ARM ASSEMBLY 21062336 (WIRES/CONN).

TOLERANCES: X ± .05
 XX ± .02
 XXX ± .005
 ANGLES ± 1°

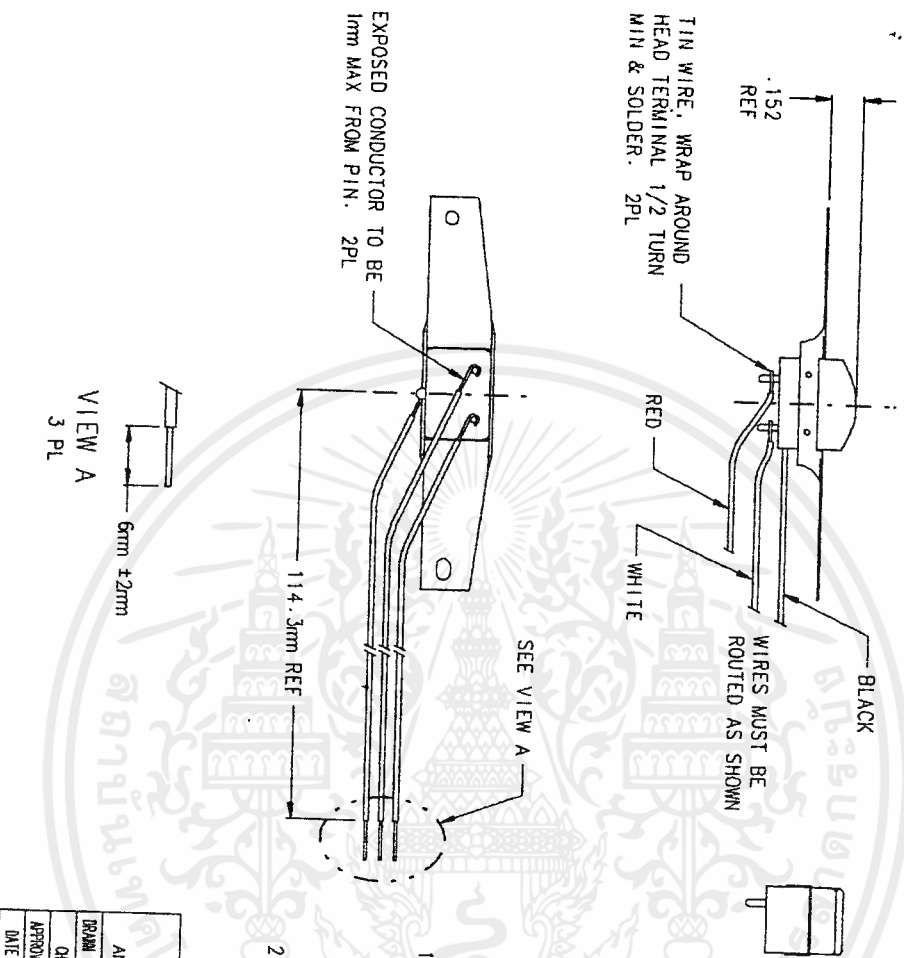
MAGTEK 2025 S. AMBLEE AVENUE
 CARSON, CA 90746

3TK BEAM STYLE SHORT READ
 HEAD DESIGN CRITERIA

DESIGNED BY	D. FISHER	DATE	17 JAN 2000
DRAWN BY	KIMVINH	DATE	17 JAN 2000
SIZE	B	SCALE	FULL
REV	1	SHEET	1 OF 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

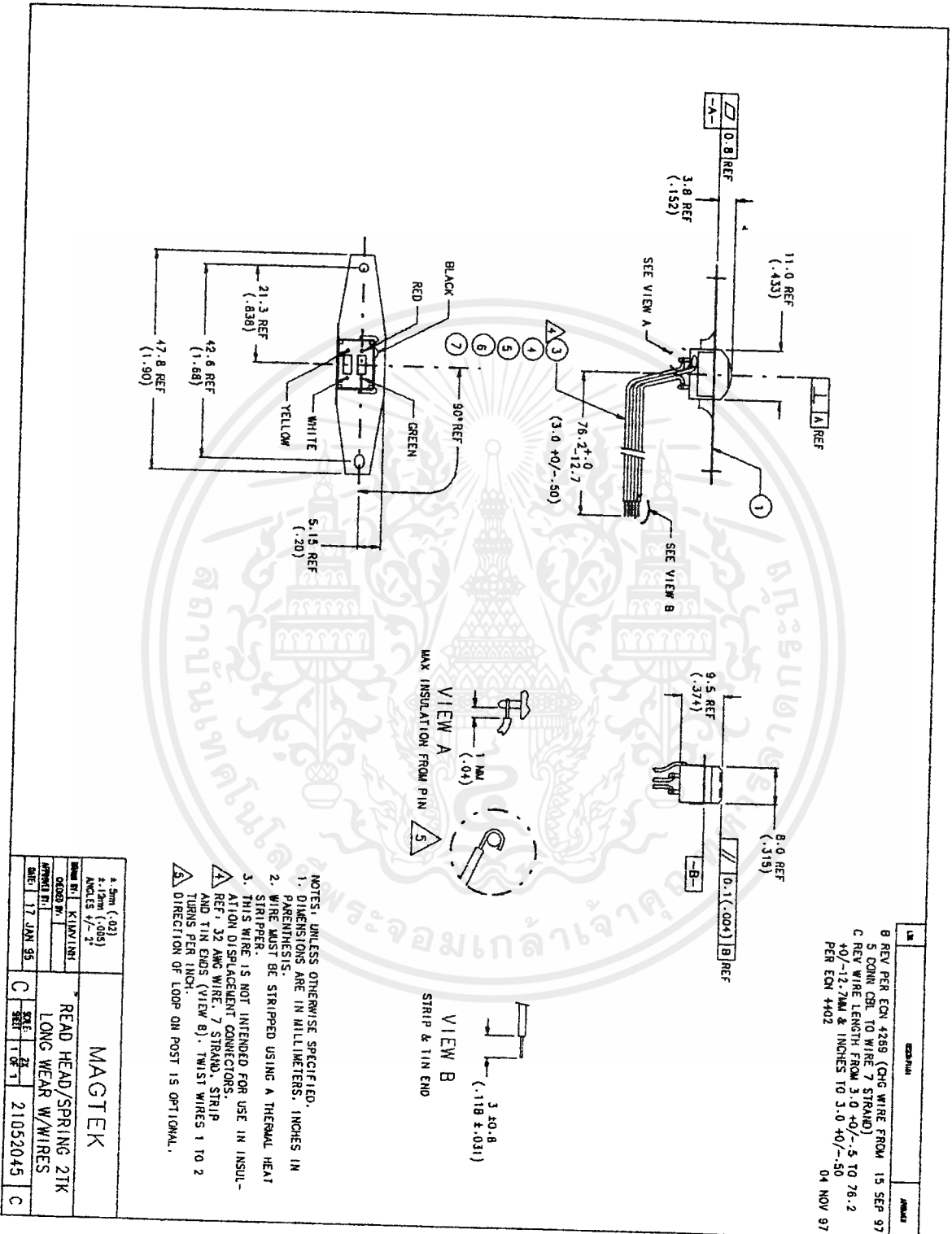
F REV PER ECN 3032 18 JAN 89 RB
 F DIM 114.3mm WAS 175mm 06 SEP 94



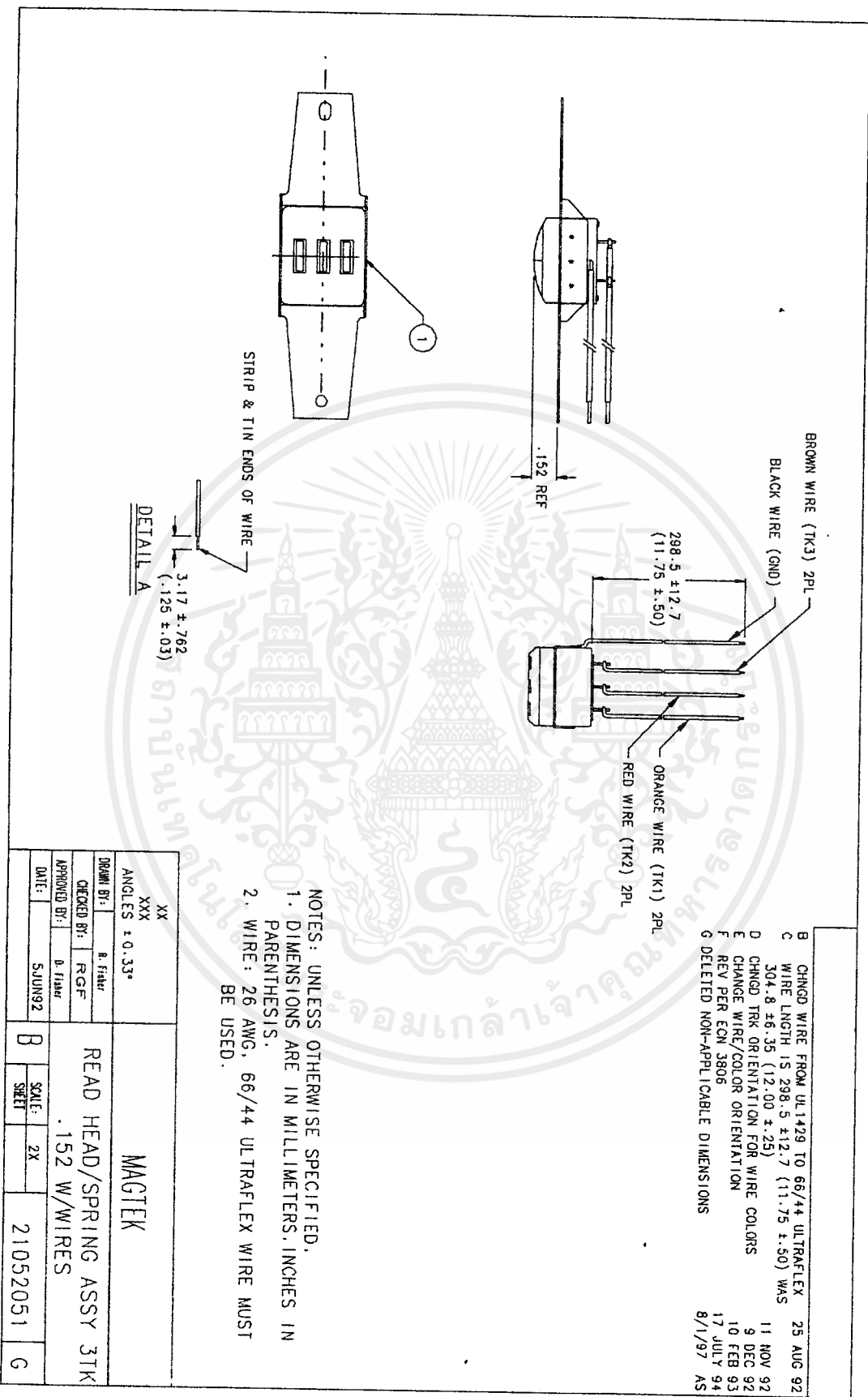
NOTES: UNLESS OTHERWISE SPECIFIED.

- 1. WIRE:
 - GAUGE: 28 AWG, SINGLE CONDUCTOR, STRANDED
 - STRAND: 7/.127mm
 - INSULATION: 0.34mm PVC; BLACK, RED, WHITE.
 - LENGTH: 114.3mm ±12.7mm (4.5" ±.5")
- 2. ALL WIRING TO BE UL STYLE 1095.

DRAWN BY: KIMVINH		MAGTEK	
CHECKED BY:		READ HEAD/SPRING TK2	
DATE: 08 MAR 94		W/WIRES	
B	SCALE: 2/1	1 OF 1	21052019
	SHEET		F



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



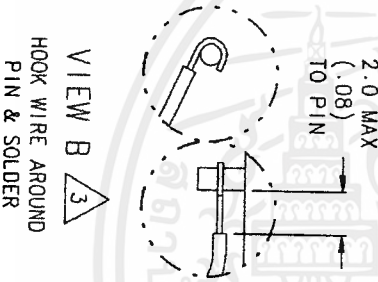
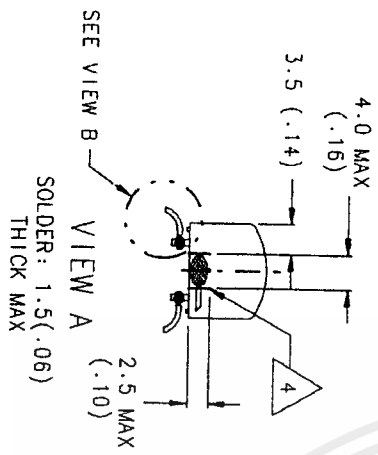
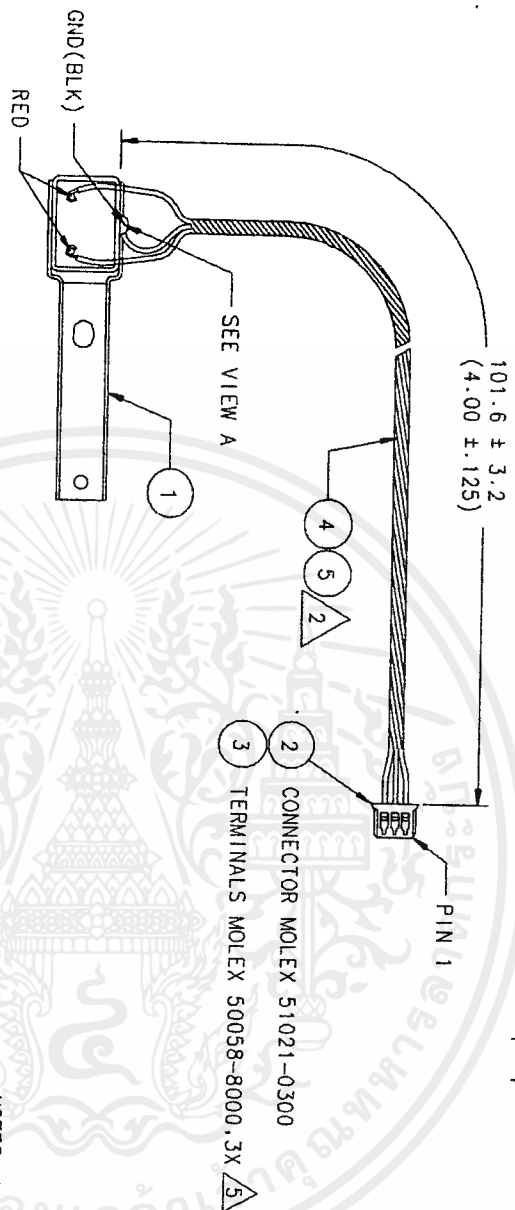
- B CHNGD WIRE FROM UL1429 TO 66/44 ULTRAFLEX 25 AUG 92
- C WIRE LENGTH IS 298.5 ±12.7 (11.75 ±.50) WAS 304.8 ±6.35 (12.00 ±.25) 11 NOV 92
- D CHANGE TRK ORIENTATION FOR WIRE COLORS 9 DEC 92
- E CHANGE WIRE/COLOR ORIENTATION 10 FEB 93
- F REV PER ECH 3806 17 JULY 94
- G DELETED NON-APPLICABLE DIMENSIONS 8/1/97 AS

NOTES: UNLESS OTHERWISE SPECIFIED,
 1. DIMENSIONS ARE IN MILLIMETERS, INCHES IN PARENTHESES.
 2. WIRE: 26 AWG, 66/44 ULTRAFLEX WIRE MUST BE USED.

XX	MAGTEK	
XXX	READ HEAD/SPRING ASSY 3TK	
ANGLES ±0.33°	.152 W/WIRES	
DRAWN BY: B. Fisher	SCALE: 2X	21052051
CHECKED BY: RGF	SHEET	
APPROVED BY: D. Fisher	DATE: 5/JUN/92	
DATE: 5/JUN/92	G	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REVISIONS			
R/V	DESCRIPTION	CHANGED BY	DATE

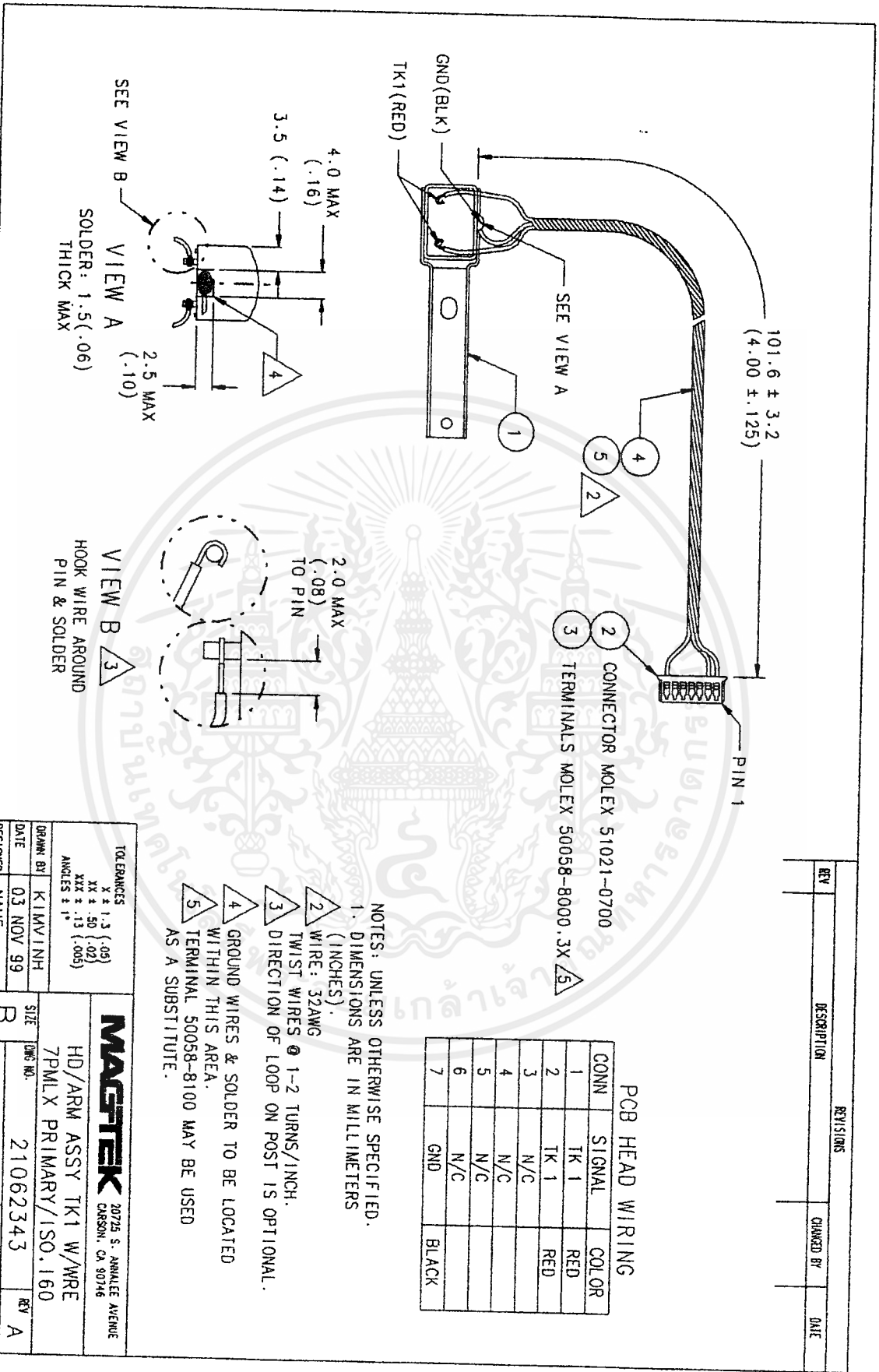


PCB HEAD WIRING

CONN	SIGNAL	COLOR
1	TK 1	RED
2	TK 1	RED
3	GND	BLACK

- NOTES: UNLESS OTHERWISE SPECIFIED,
1. DIMENSIONS ARE IN MILLIMETERS
(INCHES).
- 2 WIRE: 32AWG
 - 3 TWIST WIRES @ 1-2 TURNS/INCH.
 - 4 DIRECTION OF LOOP ON POST IS OPTIONAL.
 - 5 GROUND WIRES & SOLDER TO BE LOCATED WITHIN THIS AREA.
- TERMINAL 50058-8100 MAY BE USED AS A SUBSTITUTE.

TOLERANCES X ± .13 (.05) XX ± .50 (.02) XXX ± .13 (.005) ANGLES ± 1°		MAGTEK 20725 S. ANNULEE AVENUE CARSON, CA 90746
DRAWN BY: KIMVINH DATE: 03 NOV 99 DESIGNER:	SIZE: B SCALE: 2X SHEET: 1 OF 1	
HD/ARM ASSY 1TK W/WIRE 3PMLX PRIMARY/ISO.160		REV: A



CONNECTOR MOLEX 51021-0700
 TERMINALS MOLEX 50058-8000.3X

PCB HEAD WIRING

CONN	SIGNAL	COLOR
1	TK 1	RED
2	TK 1	RED
3	N/C	
4	N/C	
5	N/C	
6	N/C	
7	GND	BLACK

- NOTES: UNLESS OTHERWISE SPECIFIED,
 1. DIMENSIONS ARE IN MILLIMETERS
 2 WIRE: 32AWG
 TWIST WIRES @ 1-2 TURNS/INCH.
 3 DIRECTION OF LOOP ON POST IS OPTIONAL.
 4 GROUND WIRES & SOLDER TO BE LOCATED WITHIN THIS AREA.
 5 TERMINAL 50058-8100 MAY BE USED AS A SUBSTITUTE.

TOLERANCES

X ±	1.3 (.03)
XX ±	.50 (.02)
XXX ±	.13 (.005)
ANGLES ± 1°	

DRAWN BY	KIMVINH
DATE	03 NOV 99
DESIGNER	NAME

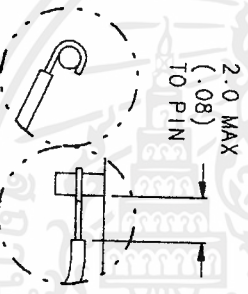
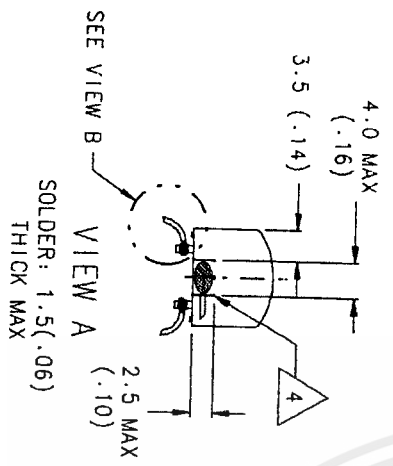
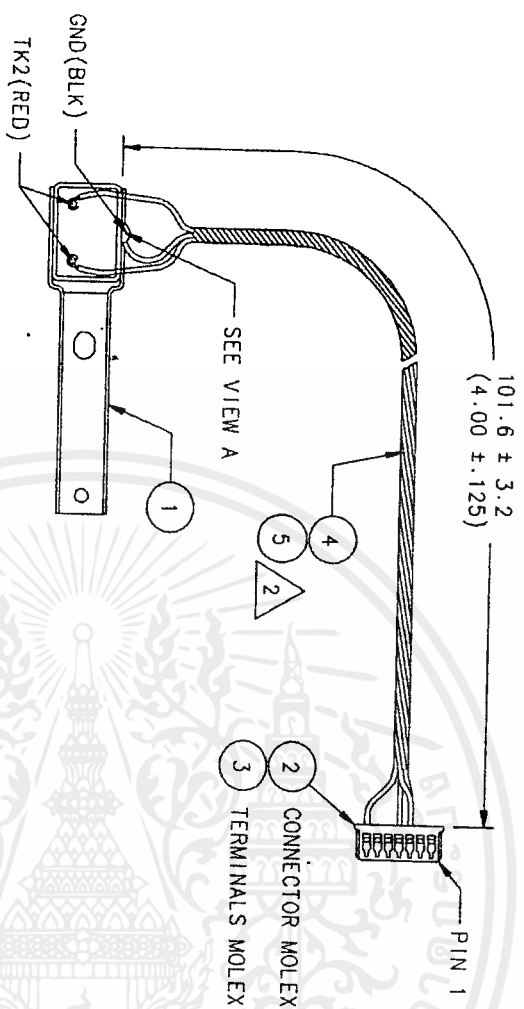
MAGSTEK 20725 S. ANNALIE AVENUE
 CARSON, CA 90746

HD/ARM ASSY TK1 W/WRE	SIZE	DWG NO.	REV
7PMLX PRIMARY/ISO.160	B	21062343	A
SCALE 2X	SHEET 1 OF 1		

REV	DESCRIPTION	CHANGED BY	DATE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REVISIONS			
REV	DESCRIPTION	CHANGED BY	DATE



PCB HEAD WIRING

CONN	SIGNAL	COLOR
1	N/C	
2	N/C	
3	TK 2	RED
4	TK 2	RED
5	N/C	
6	N/C	
7	GND	BLACK

- NOTES: UNLESS OTHERWISE SPECIFIED,
1. DIMENSIONS ARE IN MILLIMETERS.
- 2 WIRE: 32AWG
 - 3 TWIST WIRES @ 1-2 TURNS/INCH.
 - 4 DIRECTION OF LOOP ON POST IS OPTIONAL.
 - 5 GROUND WIRES & SOLDER TO BE LOCATED WITHIN THIS AREA.
 - 6 TERMINAL 50058-8100 MAY BE USED AS A SUBSTITUTE.

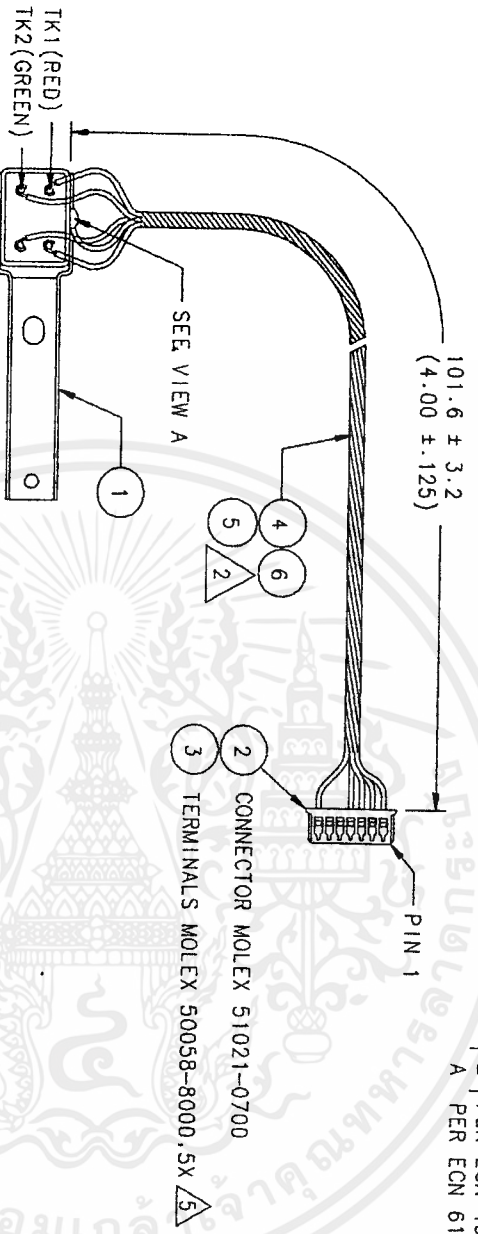
TOLERANCES		X ± 1.3 (.05)	
		XX ± .50 (.02)	
		XXX ± .13 (.005)	
		ANGLES ± 1°	
DRAWN BY	KIMVJNH	SIZE	B
DATE	03 NOV 88	DWG NO.	21062344
DESIGNER	NAME	SCALE	2X
		SHEET	1 OF 1

WAGSTEK 20725 S. ANNALIE AVENUE
ORSON, CA 90716

HD/ARM ASSY TK2 W/WRE
7PMLX PRIMARY/ISO.160

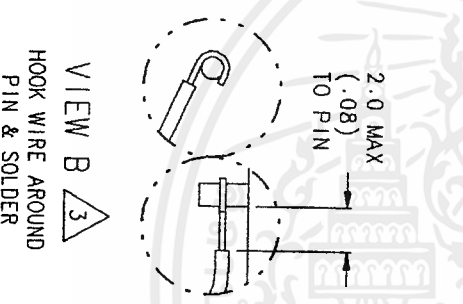
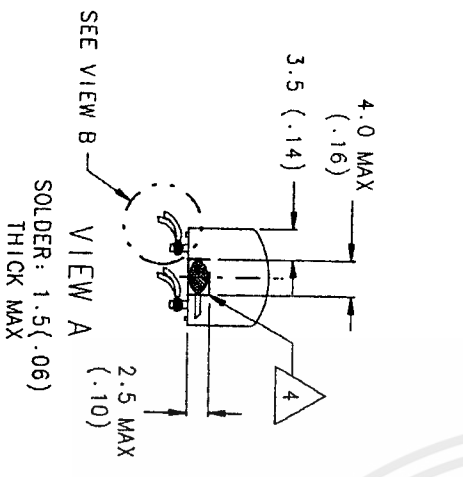
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REVISIONS			
REV	DESCRIPTION	CHANGED BY	DATE
2	PER ECN 4999, CHG TERMINAL.	KV	27MAY99
A	PER ECN 6179, PROD RELEASE.	KV	22OCT99



PCB HEAD WIRING

CONN	SIGNAL	COLOR
1	TK 1	RED
2	TK 1	RED
3	TK 2	GREEN
4	TK 2	GREEN
5	N/C	
6	N/C	
7	GND	BLACK



- NOTES: UNLESS OTHERWISE SPECIFIED,
1. DIMENSIONS ARE IN MILLIMETERS
- 2 WIRE: 32AWG
 - 3 TWIST WIRES @ 1-2 TURNS/INCH.
 - 4 DIRECTION OF LOOP ON POST IS OPTIONAL.
 - 5 GROUND WIRES & SOLDER TO BE LOCATED WITHIN THIS AREA.
 - 6 TERMINAL 50058-8100 MAY BE USED AS A SUBSTITUTE.

TOLERANCES		X ± .13 (.05)	
		XX ± .50 (.02)	
		XXX ± .13 (.005)	
ANGLES ± 1°			
DRAWN BY	KIMVINH	SIZE	DWG NO.
DATE	06 APR 99	B	21062334
DESIGNER	NAME	SCALE	2X
		SHEET	1 OF 1

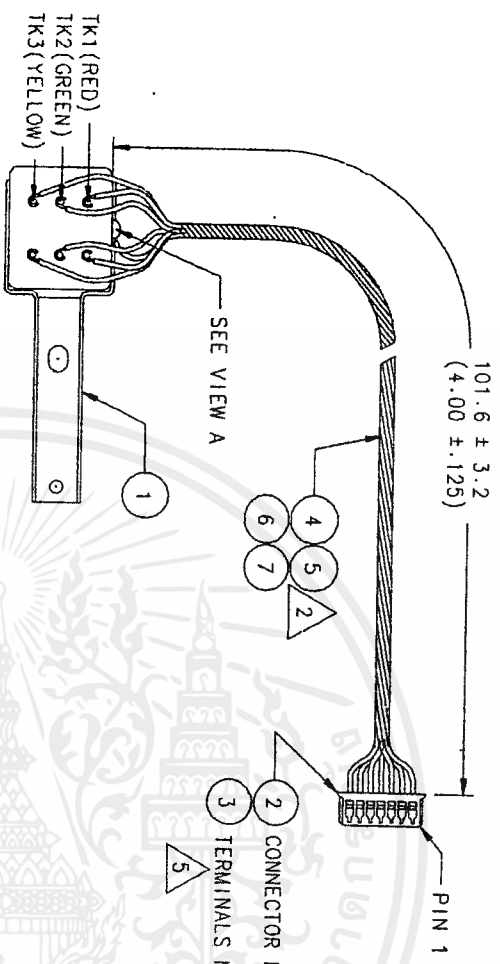
MAGSTEK 20725 S. ANAHEIM AVENUE
CARSON, CA 90746

HD/SPRG ASSY TK1, 2 W/WIRE,
PRIMARY/ISO, 160

REV A

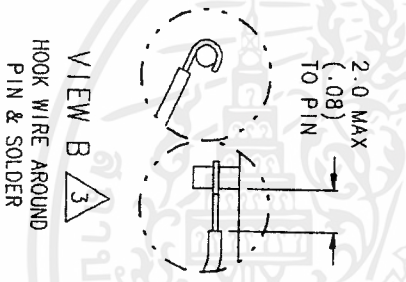
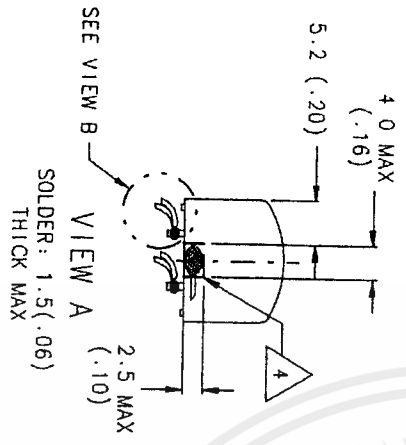
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REV	DESCRIPTION	CHANGED BY	DATE
2	PER ECN 4999, CHG TERMINAL.	KV	08JUN99



PCB HEAD WIRING

CONN	SIGNAL	COLOR
1	TK1	RED
2	TK1	RED
3	TK2	GREEN
4	TK2	GREEN
5	TK3	YELLOW
6	TK3	YELLOW
7	GND	BLACK



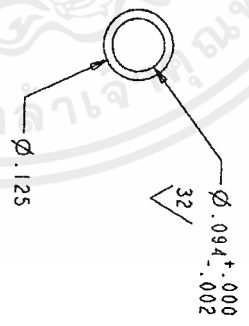
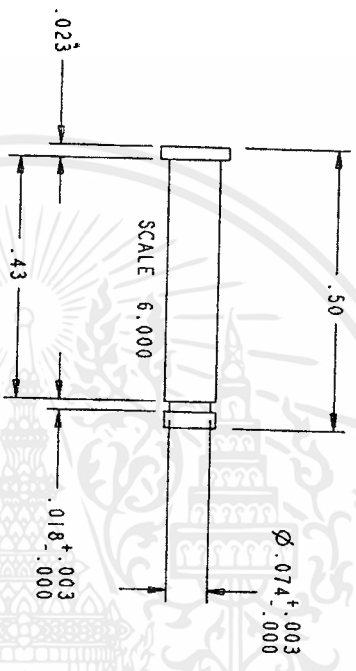
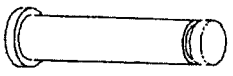
- NOTES: UNLESS OTHERWISE SPECIFIED,
1. DIMENSIONS ARE IN MILLIMETERS (INCHES).
2 WIRE: 32AWG
3 TWIST WIRES @ 1-2 TURNS/INCH.
4 DIRECTION OF LOOP ON POST IS OPTIONAL.
5 GROUND WIRES & SOLDER TO BE LOCATED WITHIN THIS AREA.
6 TERMINAL 50058-8100 MAY BE USED AS A SUBSTITUTE.

MAGSTEK 20725 S AVHILLE AVENUE CARSON, CA 90746

HD/SPRG ASSY TK1, 2, 3 W/WIRE, PRIMARY/ISO, 160

DESIGNER	DATE	SIZE	SCALE	SHEET
KIMVJNH	21 APR 99	B	2X	1 OF 1

REV 2



REV	DESCRIPTION	DATE	INITIAL
A	PRODUCTION RELEASE	22NOV94	
B	MATERIAL WAS $\phi .1875$	28FEB95	
C	REMOVED $\phi .125$ FROM NOTE 1, ADDED DIM. AND CALLOUT $\phi .125$. REPLACED BORDER.	16JUL97	CM
D	DWG CONVERTED TO PRO-E PER ECM4685	19AUG98	CM

NOTES: (UNLESS OTHERWISE SPECIFIED)
1. MATERIAL: 303S STAINLESS STEEL.

TOLERANCES:
.XX ± 0.02
.XXX ± 0.005
ANGLES ± 0.3

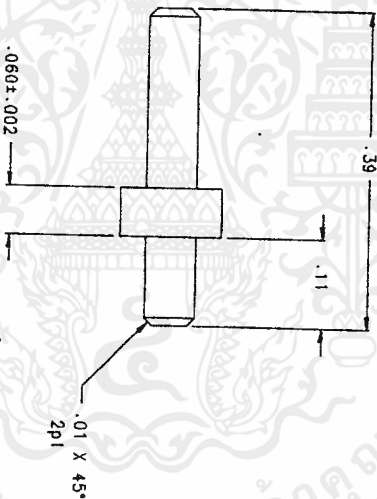
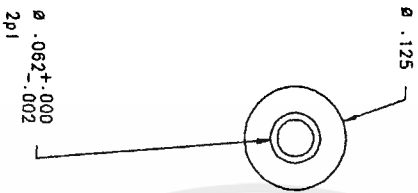
MAGTEK 20125 S. ANNALIE AVENUE
CARSON, CA 90746

DATE DRAWN: 25-MAY-94	SIZE: B	SCALE: 5.000	DWG NO.: 16035020	REV: D
DRAWN BY: R. FISHER	CHKD: B		DATE: 16035020	BY: D
DESIGNED BY:	INSTRUMENT: PART / 16035020		SHEET: 1	OF: 1

PIN - HEAD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REV	DESCRIPTION	DATE
A	PRODUCTION RELEASE	22NOV94
B	REMOVED #.125 FROM NOTE 1. ADDED DIM. CALLOUT #.125. REPLACED MAGTEK BORDER.	16JUL97 CM



NOTES: (UNLESS OTHERWISE SPECIFIED.)
 1. MATERIAL: 303S STAINLESS STEEL.

TOLERANCES
 .XX $\frac{+}{-}$.02
 .XXX $\frac{+}{-}$.005
 ANGLES $\frac{+}{-}$.33°

MAGTEK 20725 S. AMBULE AVENUE
 CARSON, CA 90746

DRAWN BY: R. FISHER
 DATE: 25 MAY 94

PIN - PIVOT

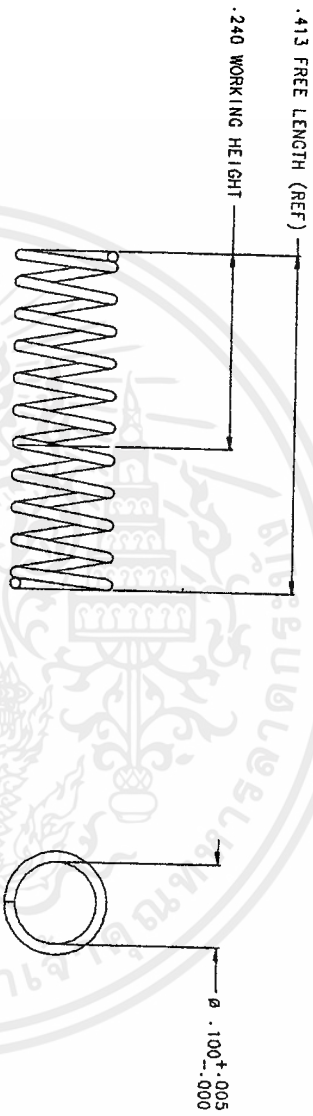
DO NOT SCALE DWG	SIZE	SCALE	FULL	REV
	B	SHEET	1 OF 1	B
			16035120	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NEXT ASSEM:
16037501

A. PRODUCTION RELEASE
B. ADDED GRADE OPTIONAL TO SST

9MAY95
17NOV97 RF



SPRING DATA
 MATERIAL: STAINLESS STEEL, GRADE OPTIONAL
 FORCE: 0.35lbs ±12% @ .240" WORKING HEIGHT.
 SPRING RATE: 2.67 lbs/inch
 HELIX DIRECTIONAL: OPTIONAL
 ENDS: CLOSED.

DRAWN BY: <i>R. S. S.</i>		CHECKED BY: <i>R. S. S.</i>		APPROVED BY: <i>R. S. S.</i>		DATE: 01MAR95	
ANGLES ±0.13°		±.005		±.02		MAGTEK	
SPRING-HEAD, MOTORIZED READER				SCALE: 8X		SHEET: 16037701	
B		B		B		B	

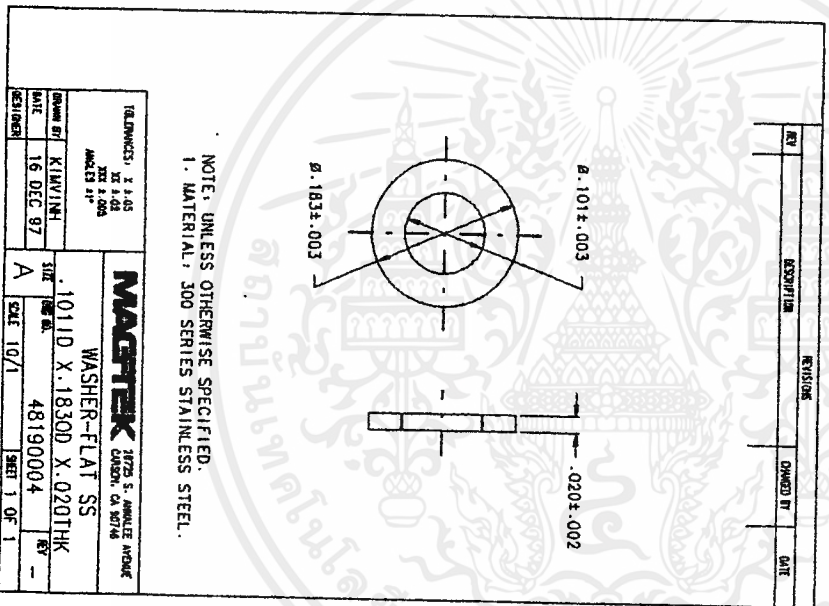
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AUTHORIZED VEN/MFG RETRIEVE - 'AVNEDT'

PART NUMBER - 85800004 RRG RETAINING RING, E TYPE 3/32 SHAFT VEN/MFG - DIST
VEN/MFG PART DESIGNATOR - 5133-9

VENDOR - WALDES TRUARC
UNIT OF MEASURE - EA

PART NUMBER - 48190004



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้