สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบรักษาความปลอดภัยแบบรหัสแถบ

BARCODE SECURITY SYSTEM

นาย อนุชิต สนิทใจ
MR.ANUCHIT SANITJAI
นาย อนุวัฒน์ ตั้งชีวินศิริกุล
MR.ANUWAT TUNGCHEEWINSIRIKUL

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอุตสาหการ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขหมู่.....
เลขทะเบียน 49875
วัน,เดือน,ปี - 2 เมษายน 2547
.b.....
.i.....

วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์	ระบบรักษาความปลอดภัยแบบรหัสแถบ			
ชื่อนักศึกษา	นาย อนุชิต	สนิทใจ	รหัสประจำตัว	43015711
	นาย อนุวัฒน์	ตั้งชีวินศิริกุล	รหัสประจำตัว	43015712
หลักสูตร	วิศวกรรมศาสตรบัณฑิต			
สาขาวิชา	วิศวกรรมอุตสาหการ			

อาจารย์ผู้ควบคุมปริญญานิพนธ์

(อ. พลชัย โชติปรายกุล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบรักษาความปลอดภัยแบบรหัสแถบ
ชื่อนักศึกษา	นาย อนุชิต สนิทใจ
	นาย อนุวัฒน์ ตั้งชีวินศิริกุล
หลักสูตร	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมอุตสาหการ

บทคัดย่อ

วัตถุประสงค์ของโครงงานเป็นการออกแบบและพัฒนาระบบรักษาความปลอดภัยแบบรหัสแถบโดยใช้หลักการควบคุมโดยไมโครคอนโทรลเลอร์ สามารถใช้ในการควบคุมการเข้าออกโดยการเปิด-ปิดประตูและสามารถจัดเก็บบันทึกข้อมูลการปฏิบัติงานด้วยฐานข้อมูล และสามารถนำไปประยุกต์ใช้งานรักษาความปลอดภัยของระบบต่างๆ ได้ในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	BARCODE SECURITY SYSTEM
Student	MR.ANUCHIT SANITJAI
	MR.ANUWAT TUNGCHEEWINSIRIKUL
Degree	Bachelor of Engineer in Industrial Engineering
Year	2002

ABSTRACT

The objective of this thesis is the design and develops of BARCODE SECURITY SYSTEM by using Micro Controller to control the security system. All operational data will be recorded in database and can be applied to other security system in future.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ เรื่องระบบรักษาความปลอดภัยแบบรหัสแถบสามารถสำเร็จลุล่วงไปด้วยดี ขอขอบคุณ ผศ. พรศักดิ์ อรรถวานิช ที่ให้คำปรึกษา ผศ.ดร. สรรพสิทธิ์ ลิ่มนรรัตน์ ช่วยแก้ไขบทคัดย่อภาษาอังกฤษ อาจารย์ พลชัย โชติปรายนกุล ที่ปรึกษาโครงงานและให้คำชี้แนะการดำเนินงาน ขอขอบคุณ คุณสุปรีชา จงทอง ที่ให้ความสะดวก ในการใช้เครื่องมือ และอุปกรณ์ต่างๆ ขอขอบคุณ อาจารย์ประจำภาควิชาวิศวกรรมอุตสาหการ

สุดท้ายนี้ ขอขอบพระคุณ คุณพ่อ คุณแม่ ผู้ให้กำเนิดให้ความรัก ความเมตตา และความห่วงใย ขอบคุณเพื่อนๆ ทุกคนที่คอยให้กำลังใจ

อนุชิต สนิทใจ

อนุวัฒน์ ตั้งชีวินศิริกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
ใบรับรองปริญญานิพนธ์	ก
บทคัดย่อภาษาไทย	ข
บทคัดย่อภาษาอังกฤษ	ค
กิตติกรรมประกาศ	ง
สารบัญ	จ
สารบัญตาราง	ช
สารบัญรูป	ซ
บทที่ 1 บทนำ	
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	1
1.3 ขอบเขตของปริญญานิพนธ์	1
1.4 ประโยชน์ที่คาดว่าจะได้รับจากการทำปริญญานิพนธ์	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	
2.1 รูปแบบการสื่อสารข้อมูลแบบอนุกรม	3
2.2 การสื่อสารข้อมูลแบบอนุกรมบนคอมพิวเตอร์	4
2.2.1 การติดต่อทางฮาร์ดแวร์	4
2.2.2 ปลั๊กและเต้ารับ (Socket)	4
2.2.3 มาตรฐาน RS-232-C	5
2.3 การส่งตัวอักษร	6
2.3.1 รูปแบบข้อมูลในคอมพิวเตอร์	6
2.3.2 การแปลงข้อมูลเป็นรูปแบบอนุกรม	6
2.3.3 การสื่อสารข้อมูลแบบซิงโครนัสและอะซิงโครนัส	7
2.4 ไมโครคอนโทรลเลอร์ AT89C52	12
2.4.1 การจัดขาใช้งานของ AT89C52	13
2.4.2 โครงสร้างของหน่วยความจำ	14
2.4.3 จิสเตอร์เฉพาะ (Special Function Register)	18
2.5 การทำงานของบาร์โค้ด	19
2.5.1 รูปแบบรหัสของบาร์โค้ด	19
บทที่ 3 ขั้นตอนและวิธีการดำเนินการ	
3.1 การวางแผนการดำเนินงาน	24
3.2 การออกแบบและส่วนประกอบ	24
3.2.1 ฮาร์ดแวร์	24
3.2.2 ซอฟต์แวร์	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
บทที่ 4 ผลการทดลอง	
4.1 การติดตั้งและการทำงานของฮาร์ดแวร์	30
4.2 การทำงานของโปรแกรม	32
บทที่ 5 บทสรุปและข้อเสนอแนะ	
5.1 บทสรุป	35
5.2 ปัญหาและแนวทางแก้ไข	35
5.3 ข้อเสนอแนะในการพัฒนา	35
บรรณานุกรม	36
ภาคผนวก	ผ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 จำนวนอักขระในแต่ละบิตข้อมูล	10
2.2 คุณสมบัติของไมโครคอนโทรลเลอร์ AT89C52	12
2.3 จิสเตอร์เฉพาะ (Special Function Register)	18
2.4 โครงสร้างตัวอักขระในรหัส USS-39	20
2.5 รูปแบบบาร์โค้ด Codabar และค่าของตัวเลขในระบบฐานสอง	22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 การสื่อสารข้อมูลแบบต่างๆ	3
2.2 หัวต่อชนิด D-type	4
2.3 บล็อกของข้อมูลอนุกรมแบบซิงโครนัส	7
2.4 บล็อกของข้อมูลอนุกรมแบบอะซิงโครนัส	8
2.5 ตัวอย่างของการส่งตัวอักษร A สองแบบ	8
2.6 การเพิ่มพาริตี้เข้าไปในข้อมูลแต่ละไบต์	9
2.7 รูปแบบของมูลในแต่ละไบต์ในการรับส่งข้อมูลอนุกรมแบบอะซิงโครนัส	11
2.8 ตำแหน่งขาของไอซี AT89C52และ AT89C2052	13
2.9 แสดงการจัดตำแหน่งแอดเดรสของหน่วยความจำโปรแกรม	15
2.10 แสดงการจัดตำแหน่งแอดเดรสของหน่วยความจำข้อมูล	16
2.11 แสดงการจัดตำแหน่งแอดเดรสของแรมส่วนล่าง	17
2.12 ระดับแรงดันทางเอาต์พุตของตัวอ่านบาร์โค้ด	19
2.13 รูปแบบของการเข้ารหัส "1A"	19
2.14 รูปแบบของบาร์โค้ดรหัสแทรก 3 ใน 9	20
2.15 รูปแบบของบาร์โค้ดรหัสแทรก 2 ใน 5	21
2.16 การเข้ารหัส Codabar "A37859B"	21
2.17 รูปแบบการเข้ารหัส UPC	22
2.18 รูปบาร์โค้ดรหัส EAN	23
3.1 การต่อใช้งานไมโครคอนโทรลเลอร์เบอร์ AT89C52	24
3.2 การติดต่อกับคอมพิวเตอร์จะใช้พอร์ตอนุกรม RS – 232 เป็น RS – 485	25
3.3 การติดต่อกับคอมพิวเตอร์จะใช้พอร์ตอนุกรม RS – 485	25
3.4 วงจรแป้นพิมพ์	26
3.5 วงจรการต่อพอร์ต PS/ 2ของเครื่องอ่านบาร์โค้ด	26
3.6 วงจรการทำงาน รีเลย์	27
3.7 การต่อสวิตช์เปิดประตู	27
3.8 การต่อไมโครสวิตช์	27
3.9 แสดงวงจรการต่อไมโครคอนโทรลเลอร์	28
3.10 Flowchart แสดงรูปแบบการทำงาน	29
3.11 การออกแบบติดตั้งอุปกรณ์การทำงาน	30
4.1 การติดตั้งอุปกรณ์การทำงาน	31
4.2 หน้าจอหลักโปรแกรม	32
4.3 หน้าจอการทำงานของโปรแกรม	33
4.4 การเพิ่มฐานข้อมูล	33
4.5 การดูข้อมูลที่จัดเก็บไว้	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

ปัจจุบันเทคโนโลยีคอมพิวเตอร์ได้เข้ามาช่วยในด้านการทำงานต่าง ๆ อย่างมากเช่น งานด้านการบริการ งานด้านอุตสาหกรรม งานด้านระบบควบคุมอัตโนมัติต่างๆ และอื่นๆ อีกมากมาย

ปริญญานิพนธ์นี้จะทำการศึกษาเกี่ยวกับการรับรหัสนักศึกษาจากเครื่องอ่านบาร์โค้ดหรือแป้นพิมพ์ เพื่อใช้ในการเปิด/ปิดประตูโดยโปรแกรมที่เขียนขึ้นให้คอมพิวเตอร์สามารถติดต่อผ่านพอร์ตอนุกรม RS-232 แล้วการส่งข้อมูลแบบอนุกรมซึ่งสามารถส่งได้ไกลประมาณ 50 ฟุต และสามารถส่งได้ไกลขึ้นอีกประมาณ 1,000 ฟุตในกรณีที่ส่งด้วยมาตรฐาน RS – 485 ซึ่งจะช่วยลดข้อจำกัดของ RS-232 แล้วทำการเก็บข้อมูลรหัสประจำตัว วันที่และเวลาการเข้าออกของแต่ละคนไว้เพื่อเป็นฐานข้อมูลในการรักษาความปลอดภัย

ด้วยขั้นตอนที่กล่าวมาข้างต้น การควบคุมการเปิด/ปิดประตูด้วยวิธีนี้ จะสามารถป้องกันนักศึกษาหรือบุคลภายนอกที่ไม่มีส่วนเกี่ยวข้องกับภาควิชาวิศวกรรมอุตสาหกรรมไม่ให้สามารถผ่านประตูเข้ามาได้

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1) ศึกษาหลักการตรวจสอบและจัดเก็บข้อมูลโดยใช้คอมพิวเตอร์ช่วยในการควบคุมปฏิบัติงาน

2) ออกแบบและพัฒนาระบบรักษาความปลอดภัยแบบรหัสแถบ

3) เพื่อเรียนรู้การใช้ไมโครคอนโทรลเลอร์ (Micro Controller) ในการติดต่อกับคอมพิวเตอร์

4) เพื่อสามารถเขียนโปรแกรม Microsoft Visual Basic 6 ในการติดต่อกับฮาร์ดแวร์

1.3 ขอบเขตของปริญญานิพนธ์

1) ควบคุมการเข้าออกประตูโดยไมโครคอนโทรลเลอร์ (Micro Controller) ณ ห้องคอมพิวเตอร์ของภาคอุตสาหการจำนวน 1 จุด

2) สามารถป้อนข้อมูลได้ทั้งจากบัตรประจำตัวนักศึกษาและทางแป้นคีย์รหัส

3) มีระบบป้องกันการปลอมแปลงข้อมูล โดยเพิ่มให้มีการใส่รหัสส่วนตัว (Password) เป็นตัวเลข 4 หลักหลังจากป้อนข้อมูลรหัสนักศึกษาทุกครั้ง

4) มีการบันทึกข้อมูลการเข้า – ออกในแต่ละวันโดยอัตโนมัติ และสามารถเรียกดูข้อมูลรวมทั้งสำเนาข้อมูลในแต่ละวันได้

5) สามารถส่งข้อมูลได้ไกลประมาณ 1,000 ฟุต โดยทำการส่งด้วย RS–485

6) ควบคุมการทำงานของฐานข้อมูลด้วยโปรแกรมที่เขียนจาก Microsoft Visual Basic 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ประโยชน์ที่คาดว่าจะได้รับจากการทำปริญญานิพนธ์

1) ได้รับความรู้เกี่ยวกับการออกแบบและพัฒนาระบบรักษาความปลอดภัยแบบรหัสแถบ

2) ได้สร้างระบบรักษาความปลอดภัยแบบรหัสแถบและใช้งานจริงที่ห้องคอมพิวเตอร์ ภาควิชาวิศวกรรมอุตสาหการ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

3) สามารถเป็นแนวทางในการศึกษาและพัฒนาด้านระบบรักษาความปลอดภัยได้ในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

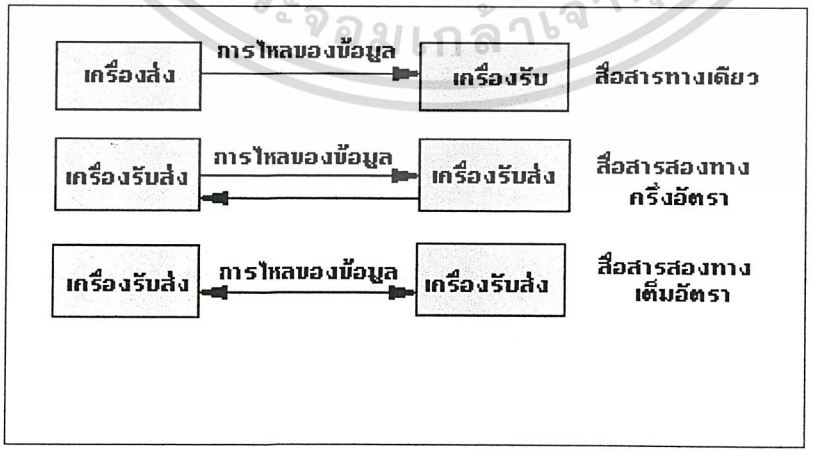
2.1 รูปแบบการสื่อสารข้อมูลแบบอนุกรม

ในการสื่อสารข้อมูลแบบอนุกรม ข้อมูลจะถูกทยอยส่งออกไปทีละบิตจนครบทั้งเวิร์ด (Word) ในสายส่งสัญญาณเพียงเส้นเดียว แต่ในการใช้งานจริงแล้ว ต้องมีสายสัญญาณอีกเส้นเป็นระดับอ้างอิง หรือกราวนด์ (Ground) ดังนั้น เมื่อมีการส่งข้อมูลแบบอนุกรม จะสามารถใช้สายสัญญาณอย่างน้อยที่สุดเพียง 2 เส้นในขณะที่การส่งข้อมูลแบบขนาน จะต้องใช้อย่างน้อยเท่ากับจำนวนบิตบวกกับสายสัญญาณระดับกราวนด์อีก 1 เส้นนอกจากนี้ การส่งข้อมูลแบบอนุกรม สามารถส่งข้อมูลออกไปได้ไกลกว่ามาก เช่น ถ้าส่งตามมาตรฐาน RS – 232 สามารถส่งได้ไกล 30 ถึง 40 ฟุต โดยไม่ต้องใช้อุปกรณ์ขับสัญญาณเพิ่มเติม อย่างไรก็ดีการส่งข้อมูลแบบอนุกรม จะต้องมีส่วนที่ทำหน้าที่แปลงข้อมูลจากข้อมูลแบบขนานมาเป็นข้อมูลแบบอนุกรม ซึ่งสามารถเป็นได้ทั้งฮาร์ดแวร์ (Hardware) และซอฟต์แวร์ (Software) และในการส่งข้อมูลนี้ยังมีข้อกำหนดบางประการเพื่อทำให้ข้อมูลมีความถูกต้อง และแน่นอนมากยิ่งขึ้น

ในการสื่อสารไม่ว่าจะเป็นการสื่อสารข้อมูลหรือการสื่อสารทั่วไปย่อมต้องประกอบด้วยผู้รับ และผู้ส่ง ผู้รับในขณะนี้อาจสามารถเป็นผู้ส่งในอนาคตได้ แต่มีบางกรณีสำหรับการสื่อสารข้อมูลที่ผู้รับ และผู้ส่งจะคงที่อยู่ตลอดเวลา เช่น การสื่อสารข้อมูลระหว่างเครื่องคอมพิวเตอร์กับเครื่องพิมพ์ เป็นต้น

การสื่อสารระหว่างอุปกรณ์ที่มี ผู้รับ และผู้ส่งที่ไม่เปลี่ยนแปลง เรียกว่า การสื่อสารแบบทางเดียว (Simplex)) กล่าวคือ การสื่อสารเป็นไปในลักษณะทิศทางเดียวตลอดเวลา ซึ่งจะมีการใช้งานน้อยมาก ดังรูปที่ 2.1 การสื่อสารโดยทั่วไปนั้น เป็นลักษณะการสื่อสารแบบดูเพกล้า (Duplex) คือ มีทิศทางการสื่อสาร 2 ทิศทาง ทั้งไปและกลับ

การสื่อแบบดูเพกล้ายังแบ่งออกได้เป็น 2 ชนิด คือ ครึ่งดูเพกล้า (Half Duplex) หรือ HDX ซึ่งจะมีทิศทางการสื่อสารในลักษณะที่ผลัดกันเป็นผู้ส่ง และผู้รับ กับการสื่อสารในลักษณะการสื่อสารแบบดูเพกล้าเต็ม (Full Duplex) หรือ FDX ซึ่งอุปกรณ์ปลายทางเป็นผู้ส่ง และผู้รับในเวลาเดียวกัน ดังรูปที่ 2.1 การสื่อสารระหว่างคอมพิวเตอร์มักจะอยู่ในลักษณะของ การสื่อสารแบบดูเพกล้าแบบใดแบบหนึ่ง [ชูชัย,2536]



รูปที่ 2.1 การสื่อสารข้อมูลแบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การสื่อสารข้อมูลแบบอนุกรมบนคอมพิวเตอร์

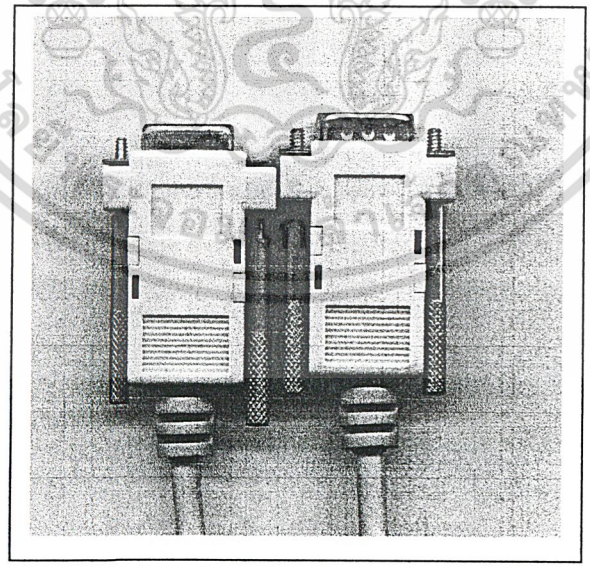
2.2.1 การติดต่อทางฮาร์ดแวร์

อุปกรณ์สองชนิดจะสื่อสารกันได้นั้น ต้องมีการเชื่อมต่อด้วยวิธีใดวิธีหนึ่ง เพื่อให้สัญญาณไฟฟ้าที่ถูกส่งโดยฝ่ายหนึ่ง สามารถถูกรับโดยอีกฝ่ายหนึ่งรับสัญญาณไฟฟ้า ที่ส่งมาจากอีกฝ่ายหนึ่งได้

การสื่อสารอาจเกิดขึ้นโดยตรง ด้วยการเชื่อมต่ออุปกรณ์สองตัวด้วยสายสัญญาณ หรือโดยอ้อมด้วยสื่อกลางที่สอดแทรกเข้ามา สื่อกลางนี้มักจะเป็นระบบโทรศัพท์สาธารณะ ซึ่งในกรณีนี้คือ โมเดม (Modem) เพื่อแปลงสัญญาณที่ปลายด้านหนึ่งให้เป็นสัญญาณที่เหมาะสมกับการส่งผ่านสายโทรศัพท์ และเพื่อแปลงกลับที่ปลายอีกด้านหนึ่งสื่อชนิดอื่น เช่น เส้นใยแสง และการส่งผ่านคลื่นวิทยุสามารถนำมาใช้ได้เช่นกัน อุปกรณ์การสื่อสารจะทำให้คอมพิวเตอร์สามารถ สื่อสารโดยใช้สื่อเหล่านี้ เสมือนถูกเชื่อมต่อด้วยวิธีเดียวกับอุปกรณ์แบบอนุกรม ดังนั้นหลักการสำหรับการสื่อสารโดยตรงจะประยุกต์ใช้กับการสื่อสารโดยอ้อมได้เช่นกัน ซึ่งกล่าวถึงการเชื่อมต่ออุปกรณ์สองตัวโดยตรง สายสัญญาณและหัวต่อ (Connector) ที่ต้องใช้และมาตรฐานที่ใช้กันทั่วไปในการเชื่อมต่อสาย[ชูชัย,2536]

2.2.2 ปลั๊กและเต้ารับ (Socket)

ปลั๊กและเต้ารับสำหรับการเชื่อมต่อสายสัญญาณเข้ากับอุปกรณ์อนุกรมมีอยู่มากมายหลายชนิด แต่นิยมใช้หัวต่อแบบ D-type ชนิด 9 ขา และ 25 ขา บางครั้งเรียกว่า DB-9 และ DB-25 นอกจากนี้ยังมีหัวต่อชนิดอื่นที่ใช้กัน เช่น หัวต่อแบบ DIN ที่ใช้ในคอมพิวเตอร์แอปเปิล (Apple Computer) หัวต่อแบบ D-type ที่ถูกตั้งชื่อแบบนี้ เพราะเมื่อมองจากด้านหน้าของหัวต่อมีรูปร่างคล้ายตัว D ประกอบด้วยขา และรูจำนวนหนึ่ง แบบที่มีขาเป็นหัวต่อตัวผู้ แบบที่มีรูเป็นหัวต่อตัวเมีย แต่ละขา หรือแต่ละรู จะมีหมายเลขซึ่งโดยทั่วไปจะถูกพิมพ์ข้าง ๆ หัวต่อแบบ D-type บางแบบแสดงไว้ในรูปที่ 2.2 [ชูชัย,2536]



รูปที่ 2.2 หัวต่อชนิด D-type

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 มาตรฐาน RS-232-C

เพื่อที่จะทำให้อุปกรณ์จากผู้ผลิตต่างบริษัทกันทำงานร่วมกันได้ มาตรฐานหลายชนิดจึงได้รับการออกแบบขึ้น มาตรฐานที่ใช้กันมากที่สุดคือ มาตรฐาน RS-232-C ถูกนำมาใช้ใน ปี ค.ศ. 1969 โดย Electronic Industrial Association มาตรฐาน RS-232-C ที่ร่างขึ้น ในตอนเริ่มแรก สำหรับการเชื่อมต่อระหว่างเทอร์มินัล และโมเดม ระบุคุณลักษณะทางไฟฟ้า ของวงจรระหว่างอุปกรณ์สองตัว กำหนดชื่อและหมายเลขแก่สายที่จำเป็นสำหรับการเชื่อมต่อวงจร ซึ่งชื่อวงจรตามมาตรฐาน RS-232-C เช่น AA,AB เป็นต้น จำได้ยากในทางปฏิบัติใช้ชื่อย่อแทน

เพื่อป้องกันไม่ให้อุปกรณ์ส่งข้อมูลบนสายเส้นเดียวกัน จึงถูกแบ่งออกเป็นสองชนิดอุปกรณ์ เช่น เทอร์มินอล ซึ่งใช้สายเส้นที่ 2 สำหรับเอาต์พุต เรียกว่าอุปกรณ์ปลายทาง (Data Terminal Equipment : DTE) เป็นต้น อุปกรณ์บางอย่าง เช่น โมเดม ซึ่งใช้สายเส้นที่ 2 สำหรับอินพุต เรียกว่าอุปกรณ์สื่อสาร (Data Communication Equipment : DCE) เป็นต้น

2.2.3.1 อุปกรณ์ปลายทางและอุปกรณ์สื่อสาร

ตามมาตรฐาน RS-232-C อุปกรณ์ปลายทางควรใช้หัวต่อตัวผู้และอุปกรณ์สื่อสารควรใช้หัวต่อตัวเมีย อย่างไรก็ตามผู้ผลิตไม่ได้ปฏิบัติตามกฎข้อนี้เสมอ ดังนั้น จึงไม่อาจแยกแยะอุปกรณ์ปลายทาง และอุปกรณ์สื่อสารโดยการมองได้เสมอไป

เมื่อทราบว่าอุปกรณ์ตัวหนึ่ง เป็นอุปกรณ์ปลายทาง และอีกตัวเป็นอุปกรณ์สื่อสาร ในทางทฤษฎีสามารถเชื่อมต่อได้อย่างง่ายดาย โดยการเชื่อมต่อสายที่มีหมายเลขตรงกัน เช่น เส้นที่ 2 กับเส้นที่ 2 และเส้นที่ 3 กับเส้นที่ 3 เป็นต้น เรียกว่าการเชื่อมต่อแบบตรง แต่มีผู้ผลิตบางรายที่ไม่ได้ทำตามมาตรฐานและทำให้เกิดปัญหาหลายอย่าง

2.2.3.2 สัญญาณทางไฟฟ้า

มาตรฐาน RS-232-C กำหนดคุณลักษณะของสัญญาณทางไฟฟ้าที่ใช้ใน การเชื่อมต่อแบบอนุกรมโดยตรงมีเพียงสองลักษณะคือ สเปซ (Space) แสดงถึงไบนารี 0 หรือแรงดันไฟบวกและมาร์ก (Mark) แสดงถึงไบนารี 1 หรือแรงดันไฟฟ้าลบบนสายข้อมูล เช่นสาย 2 และ 3 แรงดันไฟฟ้าบวกแสดงค่าลอจิก (Logic) 0 และแรงดันไฟฟ้าลบแสดงถึงค่าลอจิก 1 ส่วนบนสายตอบรับ เช่น ปลายทางพร้อม (Data Terminal Ready : DTR) และชุดข้อมูลพร้อม (Data Set Ready : DSR) แรงดันไฟฟ้าบวกแสดงว่าส่งข้อมูลได้ ส่วนแรงดันไฟฟ้าลบ หมายถึงหยุดส่งข้อมูล แรงดันไฟฟ้าบวก หรือ สเปน อยู่ระหว่าง +5 ถึง +15 โวลต์สำหรับเอาต์พุต และระหว่าง +3 ถึง +15 โวลต์ สำหรับอินพุต ความแตกต่างมีไว้เผื่อกรณีที่แรงดันไฟฟ้าสูญหาย เนื่องจากความยาวของสายสัญญาณ ในทำนองเดียวกันแรงดันไฟฟ้าลบ หรือมาร์ก ถูกกำหนดระหว่าง -5 ถึง -15 โวลต์ สำหรับเอาต์พุต และ -3 ถึง -15 โวลต์ สำหรับอินพุต สังเกตว่า ถ้าสายนำสัญญาณยาวเกินไป ระดับแรงดันไฟฟ้าจะตกลงไปเกินขอบเขตที่ยอมได้ นอกจากนี้ ความจุไฟฟ้าที่เกิดขึ้นนั้น มีผลกับคุณภาพของสัญญาณอย่างมาก โดยทำให้มีการเปลี่ยนสถานะ จากแรงดันไฟฟ้าบวกเป็นแรงดันไฟฟ้าลบไม่ชัดเจน ดังนั้น RS-232-C ไม่ได้มุ่งหวังให้ใช้กับระยะทางไกลและโดยทั่วไป 50 ฟุต เป็นระยะทางไกลที่สุดในการใช้สายสัญญาณปกติที่อัตรา การส่งข้อมูลปกติถ้าอุปกรณ์อยู่ห่างกันมากอาจจำเป็นต้องใช้โมเดม หรือวิธีการอื่น [ชูชัย,2536]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การส่งตัวอักษร

การเชื่อมต่อทางฮาร์ดแวร์ระหว่างอุปกรณ์สองตัว วิธีที่อักษรแต่ละตัวถูกเข้ารหัส และถูกส่งไปตามสายตัวนำ สามารถประยุกต์ใช้กับสัญญาณที่ถูกส่งไปตามสายโทรศัพท์ระหว่างโมเดมได้เช่นเดียวกับสัญญาณที่ถูกส่งไปตามสายสัญญาณระหว่างคอมพิวเตอร์

2.3.1 รูปแบบข้อมูลในคอมพิวเตอร์

การที่จะทำความเข้าใจการส่งผ่านข้อมูลสิ่งแรก คือ ต้องทำความเข้าใจกับวิธีที่ข้อมูลถูกเก็บไว้ภายในคอมพิวเตอร์ก่อน

2.3.1.1 บิตและไบต์

ในเลขฐานสิบ มีตัวเลขอยู่สิบตัวคือ 0 ถึง 9 การเพิ่มศูนย์หนึ่งตัวเข้าทางท้าย เป็นการคูณจำนวนด้วยสิบ โดยที่ในเลขฐานสองมีเพียงตัวเลขสองตัว คือ 0 กับ 1 การเพิ่มศูนย์เข้าทางท้ายจำนวนเป็นการคูณจำนวนด้วยสองตัวเลขศูนย์หรือหนึ่งแต่ละตัวในเลขฐานสองเรียกว่า บิต (Bit) 8 บิตจะเป็น 1 ไบต์ (Byte) ผลที่ตามมาคือ ค่าของหนึ่งไบต์จึงเป็นไปได้ตั้งแต่ 00000000 ถึง 11111111 หรือ 0 ถึง 255 ในฐานสิบบิตที่อยู่ทางขวาสุดของไบต์ เรียกว่า บิตศูนย์ บิตที่อยู่ทางซ้ายสุด เรียกว่า บิตเจ็ด โดยที่ บิตศูนย์ เรียกว่า บิตที่มีนัยสำคัญต่ำสุด (Least Significant Bit : LSB) และบิตเจ็ด เรียกว่า บิตที่มีนัยสำคัญสูงสุด (Most Significant Bit : MSB)

คอมพิวเตอร์ทำงานในระบบเลขฐานสอง เพราะว่าเป็นการง่ายที่จะแปลงรหัส 0 และ 1 เป็นแรงดันไฟฟ้าบวกและลบ ในคอมพิวเตอร์ส่วนใหญ่หน่วยที่เล็กที่สุดของหน่วยความจำ ที่อ้างถึงได้โดยการอ้างตำแหน่งคือ ไบต์ ดังนั้นเมื่อข้อมูลถูกเก็บ และจัดการในคอมพิวเตอร์ ตามปกติจึงถูกแปลงให้เป็นไบต์ที่เรียงลำดับกัน

2.3.1.2 การเข้ารหัสที่ไม่ใช่ข้อความ

ข้อมูลที่ถูกเก็บในคอมพิวเตอร์ไม่ได้อยู่ในรูปแบบของข้อความเสมอไป เช่น คำสั่งของโปรแกรม ข้อมูลตัวเลข และรูปภาพกราฟฟิก เป็นตัวอย่างข้อมูลที่ไม่ได้ถูกเก็บในรูปแบบ แอสกี (ASCII) ข้อมูลประเภทนี้โดยปกติถูกเข้ารหัส ให้ใช้ค่าทุกค่าที่เป็นไปได้ของหนึ่งไบต์ ซึ่งถูกเก็บในรูปไบนารี และสามารถขยายไปเป็นหลายไบต์ คำสั่งของโปรแกรมมักจะประกอบด้วยหนึ่งหรือ สองไบต์ เรียกข้อมูลประเภทนี้ว่าข้อมูลไบนารี (Binary Data) แม้ว่าข้อความจะถูกเก็บในรูปแบบไบนารีเช่นกัน

เนื่องจากไบต์ที่เก็บข้อมูลซึ่งไม่ใช่ข้อความสามารถเป็นค่าใด ๆ ก็ได้ในเวลาที่ข้อมูลตรงกับค่าที่มีความหมายพิเศษในตารางแอสกี ทำให้เกิดความยุ่งยากในการส่งข้อมูล ถ้าอุปกรณ์เกิดแปลงไบต์ที่ไม่ใช่ข้อความว่าหมายถึงสิ้นสุดข่าวสารในกรณีนี้ ข้อมูลไม่สามารถถูกส่งในรูปแบบข้อมูลที่ไม่ได้แปลง เพราะว่า ไบต์ที่กลางข่าวสารอาจตรงกับสัญลักษณ์สิ้นสุดข่าวสาร โดยบังเอิญ และทำให้อุปกรณ์ฝ่ายรับหยุดรับข้อมูล [ชูชัย,2536]

2.3.2 การแปลงข้อมูลเป็นรูปแบบอนุกรม

คอมพิวเตอร์เกือบทั้งหมดเก็บ และจัดการข้อมูลในแบบขนาน ซึ่งจะหมายความว่าเมื่อไบต์หนึ่งถูกส่งจากส่วนหนึ่งของคอมพิวเตอร์ไปยังส่วนอื่น ๆ ไม่ได้ถูกส่งไปครั้งละหนึ่งบิต แต่จะถูกส่งไปหลายบิตพร้อมกัน ผ่านตัวนำในแบบขนาน จำนวนบิตที่ถูกส่งในครั้งหนึ่งแปรผันไปตามเครื่อง แต่โดยปกติจะเป็นแปดหรือทวีคูณของแปด เพราะฉะนั้น คอมพิวเตอร์สามารถทำงานกับหนึ่งไบต์เป็นอย่างน้อยในครั้งหนึ่งๆ จากการสื่อสารจากคอมพิวเตอร์ไปยังอุปกรณ์อื่นๆ หลายชนิดเป็นแบบอนุกรม หมายความว่า ข้อมูลถูกส่งไปทีละหนึ่งบิต ตัวเชื่อมต่อการสื่อสารต้อง

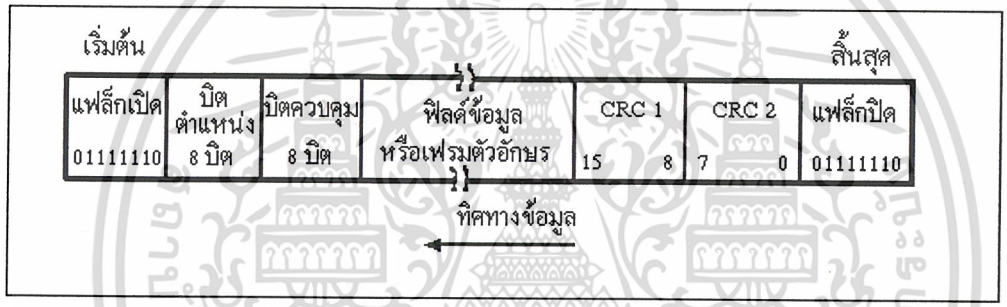
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถนำไบต์ที่รับมาและส่งออกไปทีละบิตได้สายข้อมูลในการสื่อสารแบบอนุกรมมีเพียงสภาวะ มาร์ก และสเปซ ซึ่งในกรณีของการเชื่อมต่อโดยตรงเท่ากับแรงดันไฟฟ้าลบ หรือบวกตามลำดับ ข้อมูลใดๆ ที่ถูกส่งต้องถูกแปลงให้เป็นลำดับของมาร์ก และสเปซ ก่อนสำหรับการส่งข้อมูล มาร์ก แทนค่า 1 และสเปซแทนค่า 0

2.3.3 การสื่อสารข้อมูลแบบซิงโครนัสและอะซิงโครนัส

2.3.3.1 การสื่อสารข้อมูลแบบซิงโครนัส

ในการส่งข้อมูลจะเป็นแบบขนาน หรือแบบอนุกรม การส่งข้อมูลแบบซิงโครนัส ก็คือระบบการส่งที่ข้อมูลแต่ละตัวอักษรถูกส่งออกไปตามเวลาที่แน่นอน ซึ่งหมายถึง ระยะเวลาข้อมูล แต่ละตัวอักษรที่ส่งออกไปมีค่าที่แน่นอน การสื่อสารข้อมูลแบบอนุกรมในแบบซิงโครนัสต้องมีสายสัญญาณเพิ่มเติม เพื่อกำกับการว่าควรจะส่งเมื่อใด และควรหยุดเมื่อใด ระบบที่เป็นซิงโครนัสจะเป็นระบบที่มีความเร็วสูง แต่ก็ยังต่ำกว่าการสื่อสารแบบขนาน การส่งข้อมูลอนุกรมแบบซิงโครนัสจะเพิ่มการจัดเฟรมตัวอักษร(I-field) เข้าไปร่วมกับบล็อกของข้อมูลแต่ละบล็อกดังแสดงในรูปที่ 2.3



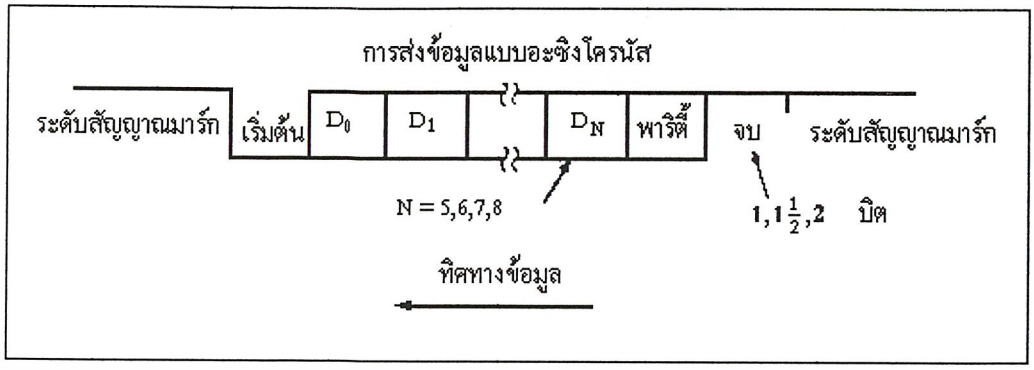
รูปที่ 2.3 บล็อกของข้อมูลอนุกรมแบบซิงโครนัส

2.3.3.2 การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัส คือระบบการรับส่งข้อมูลที่แต่ละคำถูกส่งออกไปอย่างไม่มีกำหนดเวลาที่แน่นอน คือระยะเวลาระหว่างข้อมูลแต่ละคำที่ส่งออกไป มีค่าไม่แน่นอน ตัวอย่าง เช่น เมื่อมีการส่งข้อมูลจากคอมพิวเตอร์ A ไปยังคอมพิวเตอร์ B ในการป้อนข้อมูลจากแป้นพิมพ์นั้น ผู้ที่พิมพ์ไม่สามารถพิมพ์ด้วยอัตราเร็วคงที่ได้ แต่ละครั้งที่กดจะห่างไม่เท่ากัน ดังนั้นสิ่งที่กำหนดเวลาในการส่งข้อมูล ก็คือความพร้อมเพรียงของเครื่องรับ และเครื่องส่ง

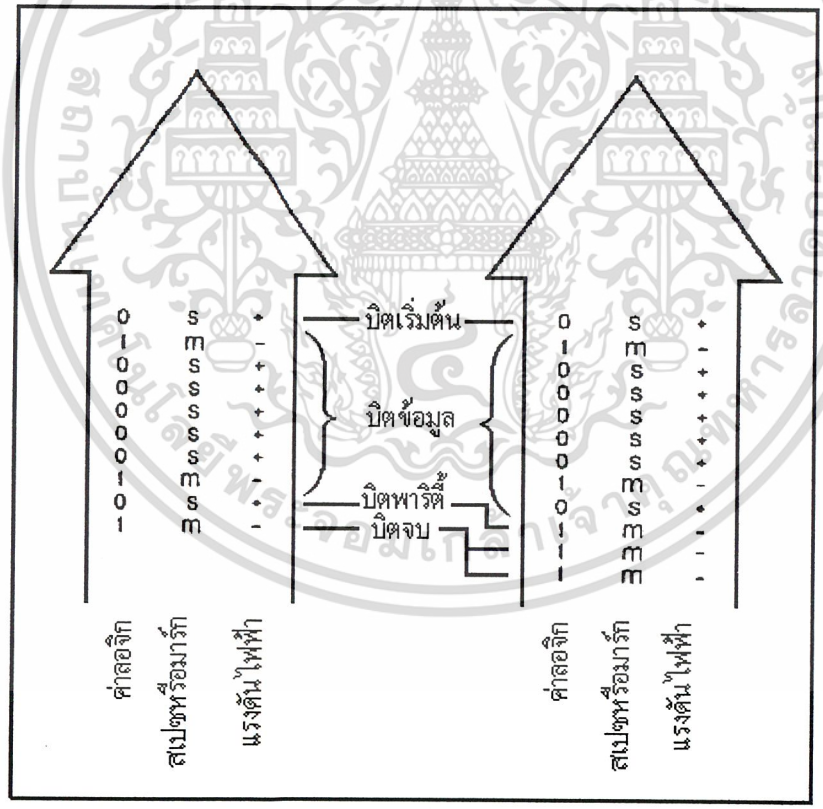
ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัสนั้น โครงสร้างของข้อมูลที่จะส่งมีลักษณะเป็นบล็อกๆ ซึ่งแต่ละบล็อกประกอบด้วยบิตเริ่มต้น (Start Bit) ส่วนของข้อมูล และบิตสุดท้าย คือบิตสิ้นสุดข้อมูล (Stop Bit) โดยบิตเริ่มต้นจะแสดงถึงการเริ่มต้นของกลุ่มข้อมูลแล้วตามด้วยส่วนของกลุ่มข้อมูล และบางกรณีอาจจะมีการเพิ่มบิตพาริตี้ (Parity Bit) เพื่อใช้ตรวจสอบ ความถูกต้องของข้อมูล และบิตสิ้นสุดข้อมูลก็จะเป็นการบอกว่าข้อมูลในบล็อกหมดลงเพียงแค่นี้ เพราะฉะนั้นถ้าเป็นการส่งข้อมูลอนุกรมแบบอะซิงโครนัส ก็จะเพิ่มการจัดเฟรมบิตข้อมูล D0-Dn รวมเข้าไปในแต่ละอักขระดังแสดงในรูปที่ 2.4 [ชูชัย,2536]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 บล็อกของข้อมูลอนุกรมแบบอะซิงโครนัส

การสื่อสารแบบอะซิงโครนัส บิตที่เป็นตัวแทนของหนึ่งไบต์ซึ่ง เรียกว่า บิตข้อมูล(Data Bit) จะถูกนำและตามด้วยบิตเริ่มต้น บิตจบ และบิตพาริตี้ กระบวนการนี้ เรียกว่าการจัดเฟรม (Framing)



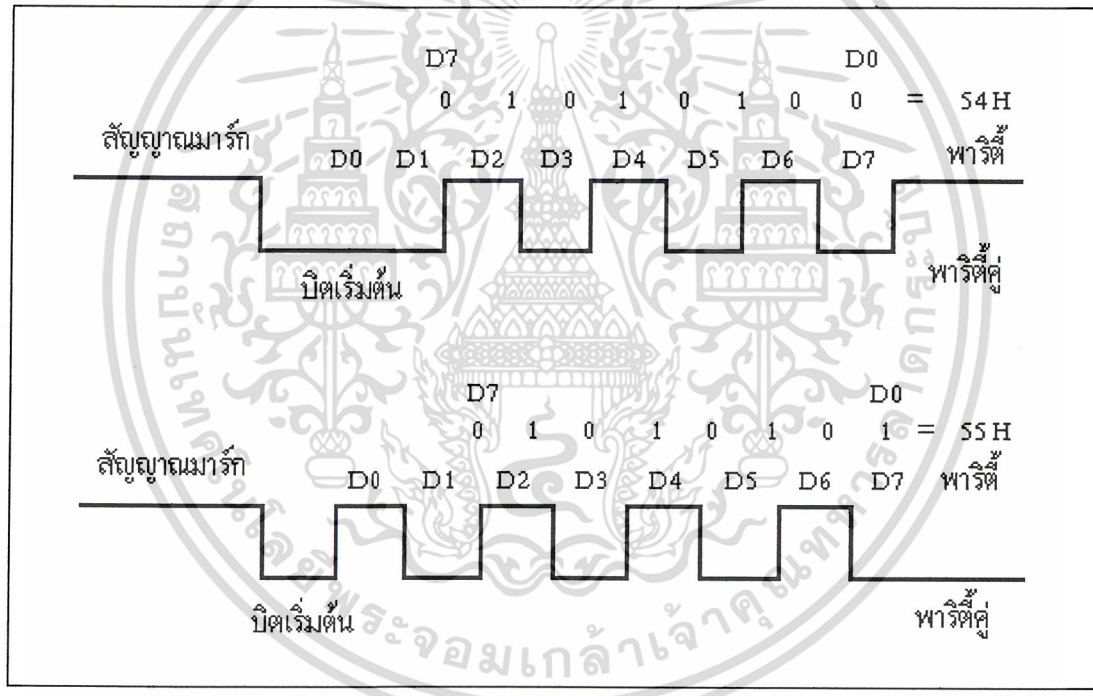
รูปที่ 2.5 ตัวอย่างของการส่งตัวอักษร A สองแบบ

จำนวนของบิตที่แทนหนึ่งตัวอักษรแปรผันไปตามโพรโตคอลสื่อสารที่ใช้ จำนวนบิตดังกล่าว หมายถึง จำนวนของบิตข้อมูล หรือความยาวเวิร์ด (Word Length) โดยปกติจะเป็นเจ็ด หรือแปดบิต ซึ่งแต่ละตัวอักษรจะถูกส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกไปเป็นกลุ่มที่ประกอบด้วย บิตเริ่มต้นตัวอักษรบิตข้อมูล บิตพาริตี้ ซึ่งสามารถเลือกได้ และบิตจบหนึ่งบิต หรือสองบิต เพื่อความชัดเจน จะเรียกคำของกลุ่มตัวอักษร และบิตเหล่านั้นว่า เฟรม (Frame) เพื่อหลีกเลี่ยงความสับสนกับคำว่า ตัวอักษรที่บางครั้งอ้างถึงบิตข้อมูล และบางครั้งอ้างถึงทั้งกลุ่มพร้อมด้วยบิตเริ่มต้นบิตจบ และบิตพาริตี้ ตัวอย่างของเฟรมที่ถูกส่งแสดงในรูปที่ 2.5

บิตเริ่มต้น (Start Bit) โพรโทคอลของการส่งข้อมูลอนุกรมแบบอะซิงโครนัส กำหนดให้สถานะ มาร์ก เป็นสัญญาณลอจิก 1 เมื่อทางด้านส่งจะทำการส่งข้อมูล จะต้องส่งบิตเริ่มต้นโดยแทนด้วยสถานะ สเปซ หรือสัญญาณลอจิก 0 จำนวนหนึ่งก่อน ซึ่งจะทำให้ด้านรับตรวจจับสถานะของสายส่งได้ว่า ขณะนั้นสายส่งกำลังมีข้อมูลส่งมา สำหรับปัญหาที่เกิดขึ้น และมีผลต่อสัญญาณข้อมูลก็คือ สัญญาณสไปก์ (Spike) ซึ่งทำให้สถานะลอจิกของสายส่งมีช่วงเวลาสั้นเกินไป ทำให้ทางด้านรับ ไม่สามารถตรวจจับสถานะของสายส่ง หรือสถานะของบิตเริ่มต้นได้ ดังนั้นส่วนใหญ่ทางด้านรับจะมีส่วนของวงจรตรวจจับสัญญาณสไปก์ ซึ่งจะทำหน้าที่สุ่มตรวจจับสัญญาณสถานะของสายส่งด้วยความถี่ของการสุ่มตัวอย่างค่าหนึ่งในระหว่างบิตต่อบิต ซึ่งอาจจะเป็น 2,4 หรือ 16 ครั้ง ในระหว่างหนึ่งบิตก็ได้



รูปที่ 2.6 การเพิ่มพาริตี้เข้าไปในข้อมูลแต่ละไบต์

หน้าที่ของบิตเริ่มต้น จะเป็นตัวบอกว่าข้อมูลเริ่มต้นตรงไหน และเมื่อร่วมใช้กับ บิตหยุด โดยความกว้างของบิตเริ่มต้นนี้จะมีความกว้างเท่ากับ 1 บิตข้อมูล ดังแสดงในรูปที่ 2.6

บิตข้อมูล (Data Bit) หลังจากด้านรับสามารถตรวจจับสัญญาณบิตเริ่มต้นได้แล้ว จะทำการเซตสถานะของชิฟรีจิสเตอร์ให้พร้อมที่จะรับบิตข้อมูลได้ โดยบิตข้อมูลจะมีจำนวนบิตเป็น 5,6,7 หรือ 8 บิต ขึ้นกับจำนวนอักขระที่ใช้ ดังแสดงตามตารางที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 จำนวนอักขระในแต่ละบิตข้อมูล

จำนวนบิตข้อมูล	จำนวนอักขระ
5 บิต	32
6 บิต	64
7 บิต	128
8 บิต	256

นอกจากนี้รหัสต่างๆที่ใช้อาจจะแทนด้วย 5 บิต ซึ่งเป็นมาตรฐานของรหัสโบดอต (Baudot) โดยประกอบไปด้วยกลุ่มของอักขระต่าง ๆ จำนวน 32 อักขระ และถ้าเป็นรหัสขนาด 7 บิต ประกอบไปด้วยกลุ่มของอักขระจำนวน 128 อักขระ ซึ่งเป็นมาตรฐานของรหัสแอสกี และใช้กันแพร่หลายมาก นอกจากนี้ยังมีรหัสขนาด 8 บิต หรือมาตรฐานของรหัส เอบซีดิก (EBCDIC) โดยมาตรฐานนี้ประกอบด้วยกลุ่มของอักขระ 256 อักขระเป็นต้น

บิตพาริตี้ การตรวจสอบพาริตี้เป็นวิธีหนึ่งในการทดสอบว่าข้อมูลที่ส่งได้ถูกรับอย่างถูกต้องหรือไม่ อุปกรณ์ฝ่ายส่งจะเพิ่มบิตพาริตี้อีกหนึ่งบิตเป็นค่า 0 หรือ 1 ขึ้นอยู่กับบิตข้อมูลทางด้านฝ่ายรับจะตรวจสอบว่า บิตพาริตี้มีความสัมพันธ์ที่ถูกต้องกับบิตอื่นหรือไม่ ถ้าไม่แสดงว่า บางสิ่งต้องผิดพลาดในระหว่างการส่ง พาริตี้สามารถคำนวณได้จากวิธีต่อไปนี้

พาริตี้คู่ (Even Parity) หมายความว่า จำนวนของบิตข้อมูลที่เป็น 1 และจะมีค่าของบิตพาริตี้รวมกันเป็นจำนวนคู่ เช่น ตัว A ในฐานสองคือ 01000001 เมื่อนับจำนวนของบิตที่เป็น 1 จะได้สองซึ่งเป็นเลขคู่ ดังนั้นบิตพาริตี้ต้องเป็น 0 ถ้าตัวอักษร A ที่รับได้มีพาริตี้เป็น 1 แสดงว่าเกิดความผิดพลาดในระหว่างการส่ง

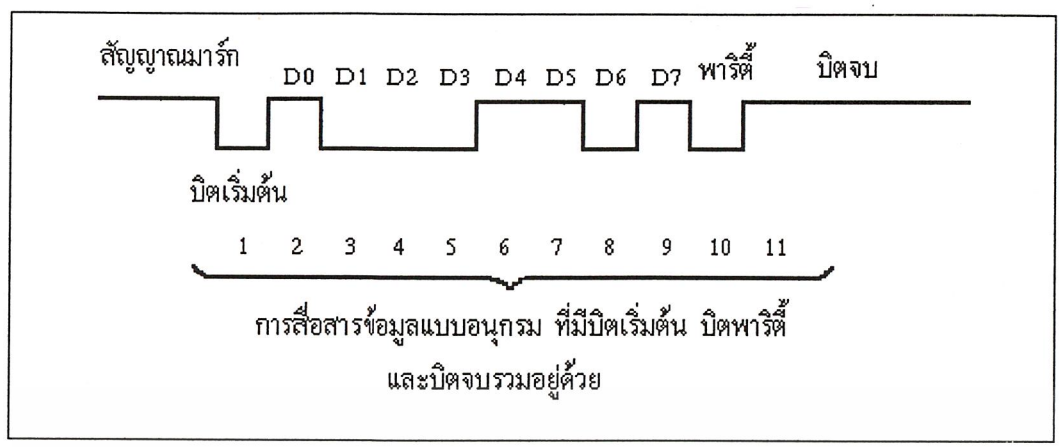
พาริตี้คี่ (Odd Parity) หมายความว่า จำนวนทั้งหมดของบิตข้อมูลที่เป็น 1 บวกกับค่าของบิตพาริตี้เป็นจำนวนคี่ ดังนั้นสำหรับตัวอักษร A บิตพาริตี้ควรถูกเซตเป็น 1 ทั้งหมดเป็นสามซึ่งเป็นจำนวนคี่ไม่มีพาริตี้ (Null Parity) หมายถึงไม่มีบิตพาริตี้ สเปซ บางครั้งเรียกว่า Bit Trimming คือ บิตพาริตี้ที่เป็น 0 เสมอ ซึ่งมีประโยชน์ในการตรวจสอบข้อผิดพลาดบางอย่าง เมื่อการส่งข้อมูลเป็นขยะมาก ในการส่งข้อมูลบางครั้งบิตพาริตี้อาจกลายเป็น 1 แสดงว่าเกิดข้อผิดพลาด พาริตี้แบบนี้สามารถใช้เพื่อส่งอักษรเจ็ดบิต ให้กับอุปกรณ์ที่ต้องการตัวอักษรแปดบิตได้ อุปกรณ์ฝ่ายรับจะถือว่าบิตพาริตี้ เป็นบิตสุดท้ายของข้อมูล

มาร์ก บางครั้งเรียกว่า Bit Forcing ทำงานเหมือนกับพาริตี้แบบสเปซ ยกเว้นแต่ พาริตี้จะเป็น 1 เสมอเนื่องจากหนึ่งในตำแหน่งนั้นสามารถที่จะถูกตีความรวมเข้ากับค่าของจำนวนบิตจบได้ อุปกรณ์หรือคอมพิวเตอร์ฝ่ายรับต้องถูกโปรแกรมไม่ให้สนบิตพาริตี้ที่เป็น 1

บิตจบ ที่ท้ายของแต่ละเฟรมบิตจบถูกส่งออกมา บิตจบมีทั้งแบบหนึ่งบิต หนึ่งบิตครึ่ง หรือสองบิตอย่างน้อยต้องมีหนึ่งบิต เพื่อประกันว่ามีแรงดันไฟฟ้าลบอย่างน้อยเป็นช่วงเวลาหนึ่งก่อนที่เฟรมถัดไปจะมาถึง เพื่อที่จะสามารถแยกแยะเฟรมถัดไปได้ จากบิตเริ่มต้นที่เป็นบวก บิตจบมากกว่าหนึ่งบิต โดยทั่วไปจะใช้เมื่ออุปกรณ์ฝ่ายรับต้องการเวลาเพิ่มขึ้น ก่อนที่อุปกรณ์ฝ่ายรับจะสามารถจัดการกับตัวอักษรที่เข้ามาตัวถัดไปได้ ดังแสดงในรูปที่ 2.7

หนึ่งบิตครึ่ง หมายความว่า ความยาวของบิตนั้น มากกว่าความยาวของบิตปกติ บิตจบบังคับให้มีช่องว่างอย่างน้อยระหว่างเฟรมถูกส่งเป็นไบนารีหนึ่ง ซึ่งในการเชื่อมต่อโดยตรงจะเป็นแรงดันไฟฟ้าลบ บิตจบสองบิตซึ่งมักจะใช้ที่อัตราบอด 110 บิตต่อวินาที ซึ่งเป็น อัตราการส่งข้อมูลต่ำสุดที่ใช้กันทั่วไปเพื่อสอดคล้องกับความต้องการของเครื่องพิมพ์รุ่นเก่า ซึ่งใช้อัตราบอดต่ำ และต้องการเวลาพิเศษเพื่อประมวลตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 รูปแบบของมูลในแต่ละไบต์ในการรับส่งข้อมูลอนุกรมแบบอะซิงโครนัส

อัตราบอด (Baud Rate) อัตราบอดแสดงจำนวนของสัญญาณแต่ละหน่วยในหนึ่งหน่วยวินาที ซึ่งถูกตั้งชื่อตาม Mr. Baudot ซึ่งเป็นผู้บุกเบิกการสื่อสารชาวฝรั่งเศส ในการส่งแบบไบนารีจะเป็นสิ่งเดียวกับ บิตต่อวินาที หรือจำนวนของเลขฐานสองที่ถูกส่งในหนึ่งวินาที ซึ่งคำที่ใช้ทั้งสองคำนี้ มีความแตกต่างกัน แต่มักจะทำให้สับสนในการเชื่อมต่อพอร์ต RS-232 โดยตรง สัญญาณจะเป็นหนึ่งในสองสถานะ ในเวลาขณะใดขณะหนึ่งอัตราบอด และบิตต่อวินาทีจึงเท่ากัน มีจุดน่าสังเกตคือทั้งอัตราบอด และบิตต่อวินาทีอ้างถึงอัตราที่บิตภายในหนึ่งเฟรมที่ถูกส่ง ช่องว่างระหว่างเฟรมอาจมีความยาวแปรเปลี่ยนได้ เช่น การพิมพ์ตัวอักษรด้วยอัตราแตกต่างกัน ดังนั้นทั้งอัตราบอด และบิตต่อวินาที จึงไม่ได้หมายถึงอัตราที่ข้อมูลถูกส่งไปจริง ๆ [ชูชัย,2536]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

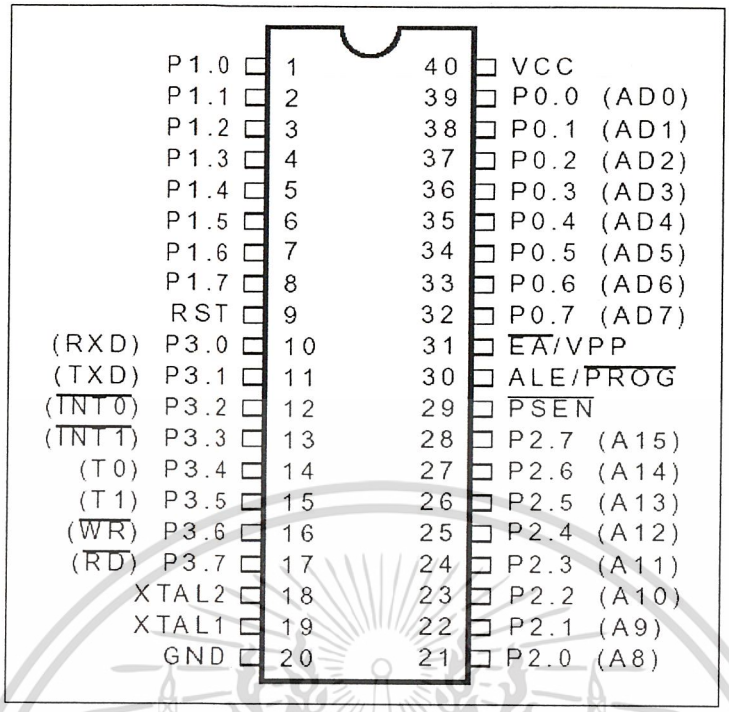
2.4 ไมโครคอนโทรลเลอร์ AT89C52

เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต (MCU) ที่สามารถใช้เครื่องคอมพิวเตอร์ PC ทำการลบและเขียนโปรแกรมได้โดยตรง ไม่ต้องถอด MCU ออกจากการ์ดหรือแผงวงจรในลักษณะที่เรียกว่า in system programming หรือจะใช้เครื่องโปรแกรม (universal programmer) โดยตรงก็ได้โดยไม่ต้องใช้การลบ ด้วยแสงอัลตร้าไวโอเลต (UV eraser) เพื่อลบโปรแกรมให้ยุ่งยากอีกต่อไป และมีหน่วยความจำ แบบแฟลตอยู่ ภายในตัวขนาด 4 กิโลไบต์ ที่สามารถโปรแกรมทับลงไปได้อีกนับเป็นพันครั้งเหมือนกับ การโปรแกรมในลักษณะเดียวกันกับ EEPROM

ตารางที่ 2.2 คุณสมบัติของไมโครคอนโทรลเลอร์ AT89C52

คุณสมบัติ	AT89C52
ขนาดหน่วยความจำแบบแฟลต	8 กิโลไบต์
ขนาดหน่วยความจำ RAM	256 ไบต์
จำนวนขาต่อใช้งาน	40 ขา
พอร์ตอินพุตเอาต์พุต (I/O)	32 I/O
รูปแบบตัวถังบรรจุ	PDIP, PLCC, QFP
คาบเวลาและชุดวงจรนับ	16 บิต, 2 ชุด
ย่านแรงดันไฟเลี้ยง	2.7 - 6.0 โวลต์
ย่านความถี่การทำงาน	24 เมกะเฮิรตซ์
วงจรเปรียบเทียบอนาลอก	ไม่มี
อินเตอร์รัพเอาต์พุต	5 เอาต์พุต
โหมดสภาวะสงบ, ประหยัดกำลังงาน	มี
ขับ LED ได้โดยตรง	ไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.8 ตำแหน่งขาของไอซี AT89C52

2.4.1 การจัดขาใช้งานของ AT89C52

ขา 1-8: P1.0-P1.7 เป็นขาอินพุต/เอาต์พุตพอร์ต 1 เป็นพอร์ตขนาด 8 บิต สามารถใช้เป็นอินพุต/เอาต์พุตพอร์ต

ขา 9 : RESET เป็นขาอินพุต จะทำการรีเซตระบบเมื่อมีสัญญาณอินพุตเป็นลอจิกต่ำอย่างน้อย 2 วงรอบการทำงาน (machine cycle)

ขา 10-17 : P3.0-P3.7 เป็นขาอินพุต/เอาต์พุตพอร์ต 3 เป็นพอร์ตขนาด 8 บิต ซึ่งมีลักษณะพิเศษคือ สามารถใช้เป็นอินพุต/เอาต์พุตพอร์ตทั่วไปหรือใช้ทำหน้าที่พิเศษ ก็ได้ ซึ่งถ้าใช้งานในหน้าที่พิเศษจะต้องทำการเซตค่าลอจิกไฮไปยังขาพอร์ตที่ต้องการเสียก่อนหน้าที่พิเศษต่างๆมีดังนี้

- เป็นขา RxD (P3.0) ใช้งานเป็นพอร์ตอนุกรม ขานี้จะทำหน้าที่รับข้อมูลที่ส่งผ่านแบบอนุกรม เข้าสู่ตัวไมโครคอนโทรลเลอร์ ถ้าการส่งนั้นเป็นแบบอะซิงโครนัส (asynchronous) และขานี้จะทำหน้าที่ทั้งรับและส่งข้อมูลถ้าการส่งนั้นเป็นแบบซิงโครนัส(synchronous)

- เป็นขา TxD (P3.1) ใช้งานเป็นพอร์ตอนุกรม ขานี้จะทำหน้าที่ส่งข้อมูลแบบอนุกรมออกจาก ไมโครคอนโทรลเลอร์ ถ้าการส่งนั้นเป็นแบบอะซิงโครนัส และขานี้จะทำหน้าที่ส่งสัญญาณนาฬิกา (clock)ถ้าการส่งนั้นเป็นแบบซิงโครนัส

- เป็นขา INT0 (P3.2) เป็นขารับสัญญาณอินเตอร์รัพต์ที่ 0 หรือรับสัญญาณอินพุตควบคุมตัว ไทเมอร์ที่0

- เป็นขา INT1 (P3.3) เป็นขาที่รับสัญญาณอินเตอร์รัพต์ที่ 1 หรือรับสัญญาณอินพุตควบคุมตัวไทเมอร์ที่1

- เป็นขา T0(P3.4)ขานี้ทำหน้าที่เป็นอินพุตรับสัญญาณของตัวเคาน์เตอร์ที่0

- เป็นขา T1(P3.5)ขานี้ทำหน้าที่เป็นอินพุตรับสัญญาณของตัวเคาน์เตอร์ที่1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เป็นขา WR (P3.6) ขานี้เป็นสัญญาณควบคุมการเขียนข้อมูลจากพอร์ต 0 ไปยังหน่วยความจำ ข้อมูลภายนอก ในที่นี้คือหน่วยความจำแรมภายนอกนั่นเอง

- เป็นขา RD (P3.7) ขานี้เป็นสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำข้อมูลภายนอกไปยังพอร์ต 0

ขา18-1:XTAL2,XTAL1เป็นขาเอาต์พุตจากออสซิลเลเตอร์

ขา 20 : GND เป็นขากราวด์

ขา 21-28 : P2.0-P2.7 เป็นขาอินพุต/เอาต์พุตพอร์ต 2 สามารถใช้เป็นพอร์ตอินพุต/เอาต์พุตแบบทั่วไปได้ โดยถ้าใช้งานเป็นอินพุตต้องทำการเซตค่าลอจิกไฮไปยังขาที่จะใช้งานเป็นอินพุตก่อน

ขา 29 : PSEN (Program Store Enable) เป็นขาเอาต์พุตสัญญาณควบคุมซึ่งจะไปอีนาเบิลหน่วยความจำโปรแกรมภายนอก สัญญาณที่ขานี้จะแอคทีฟทุกๆ 2 ครั้งในแต่ละแมชชีนไซเกิล ยกเว้นในระหว่างที่มีการเข้าถึง หน่วยความจำข้อมูลภายนอก และขานี้จะเป็นลอจิกไฮระหว่างที่โปรแกรมหน่วยความจำภายในกำลังทำงานอยู่

ขา 30 : ALE (Address Latch Enable) เป็นขาที่ใช้ควบคุมการแลตช์ของขา P0 เมื่อมีการใช้หน่วยความจำภายนอก

ขา 31 : EA (External Access enable)

ขา 32-39 : P0.0-P0.7 เป็นขาอินพุต/เอาต์พุตพอร์ต 0 สามารถใช้งานเป็นอินพุต/เอาต์พุตแบบทั่วไป และทำหน้าที่พิเศษคือ พอร์ต 0 สามารถทำงานแบบมัลติเพล็กซ์ได้คือ สามารถสลับการทำงาน โดยทำงานเป็นแบบส่งข้อมูล (data bus) กับแบบส่งตำแหน่งแอดเดรสค่าต่ำ (low address) เพื่อใช้ในการเข้าถึงหน่วยความจำข้อมูลและหน่วยความจำโปรแกรมได้

ขา 40 : Vcc เป็นขาไฟเลี้ยง +5 โวลต์

2.4.2 โครงสร้างของหน่วยความจำ

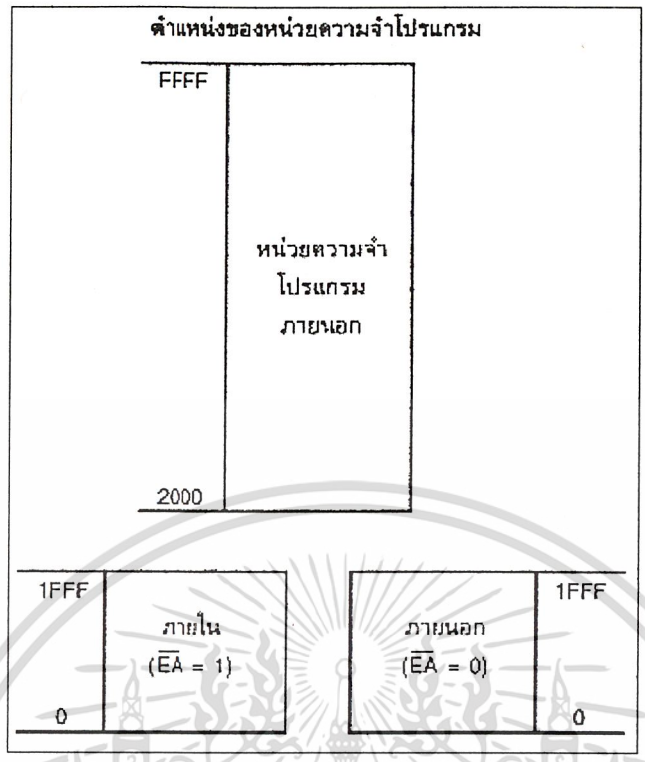
หน่วยความจำของไมโครคอนโทรลเลอร์ตระกูล MCS-51 สามารถแบ่งหน่วยความจำออกได้เป็น 2 ประเภท คือ หน่วยความจำโปรแกรมและหน่วยความจำข้อมูล

2.4.2.1 หน่วยความจำโปรแกรม

ไมโครคอนโทรลเลอร์เบอร์ AT89C52 มีรอมภายในขนาด 8K ถ้ากำหนดให้ขา EA เป็นลอจิกไฮ SAB80C515 จะทำงานตามคำสั่ง ที่อยู่ในรอมภายใน ซึ่งจะทำงานไปจนถึงตำแหน่งที่ 1FFFH ที่เหลือตั้งแต่ ตำแหน่งที่ 2000H ถึงตำแหน่ง FFFFH จะถูกอ่านมาจากหน่วยความจำโปรแกรม ภายนอกตามปกติ

ถ้ากำหนดให้ขา EA เป็นลอจิกโลว์ SAB80C515 จะ อ่านคำสั่งทั้งหมดจากหน่วยความจำโปรแกรมภายนอก ในกรณี เบอร์ SAB80C535 ไม่มีรอมอยู่ภายใน เพราะฉะนั้นขา EA จะต้อง กำหนดให้เป็นลอจิกโลว์ตลอดเวลารูปที่ 9 แสดงรายละเอียดการจัด ตำแหน่งของหน่วยความจำโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



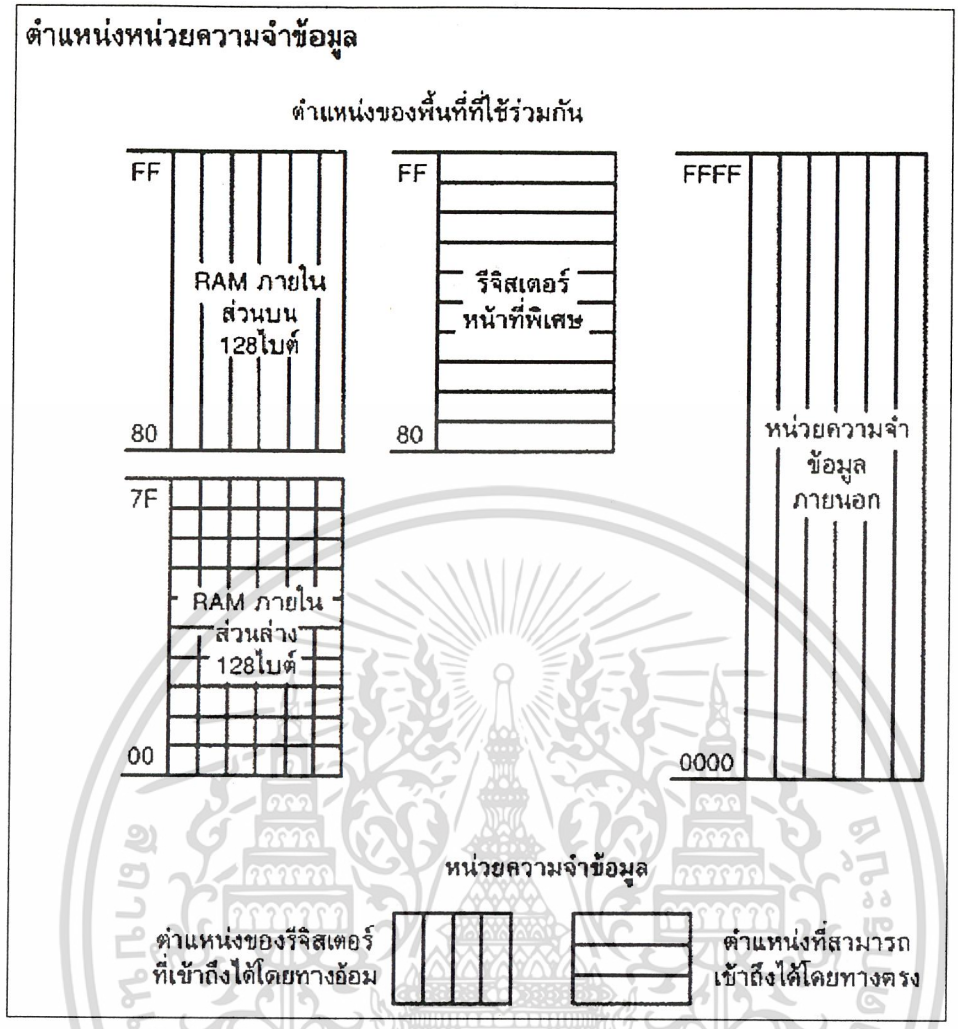
รูปที่2.9 แสดงการจัดตำแหน่งแอดเดรสของหน่วยความจำโปรแกรม

2.4.2.2 หน่วยความจำข้อมูล

หน่วยความจำข้อมูลประกอบด้วย หน่วยความจำข้อมูลภายในและหน่วยความจำข้อมูลภายนอก ในหน่วยความจำข้อมูลภายในนั้นสามารถแบ่งออกได้เป็น 3 ส่วนคือ แรมส่วนล่างจำนวน 128 ไบต์ (lower RAM), แรมส่วนบน 128ไบต์(upperRAM)และส่วนของรีจิสเตอร์หน้าที่พิเศษ(SFR)จำนวน128ไบต์

แรมส่วนบนจำนวน 128 ไบต์ และส่วนของรีจิสเตอร์หน้าที่พิเศษจะมีการอ้างตำแหน่งแอดเดรส บางส่วนร่วมกัน การเข้าถึงหน่วยความจำทั้ง 2 ส่วนทำได้โดยใช้วิธีการอ้างถึงตำแหน่งรูปแบบ ต่างกันซึ่งสามารถอธิบายได้ดังนี้ ในแรมส่วนล่างจำนวน 128 ไบต์ สามารถเข้าถึงได้ โดยการอ้างถึงตำแหน่งแบบโดยตรง (direct addressing) หรือสามารถอ้างถึง โดยใช้วิธีการกำหนดตำแหน่งโดยอ้อมโดย ใช้รีจิสเตอร์ (register indirect addressing ส่วนแรมส่วนบนจำนวน 128 ไบต์ สามารถอ้างถึงได้โดยใช้วิธี การกำหนดตำแหน่งโดยอ้อมโดยใช้รีจิสเตอร์ในส่วนของรีจิสเตอร์หน้าที่พิเศษ (SFR) สามารถ เข้าถึงได้โดยการอ้างถึงตำแหน่งแบบโดยตรง รูปที่ 2.10 แสดงรายละเอียดการจัดตำแหน่งของ หน่วยความจำข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

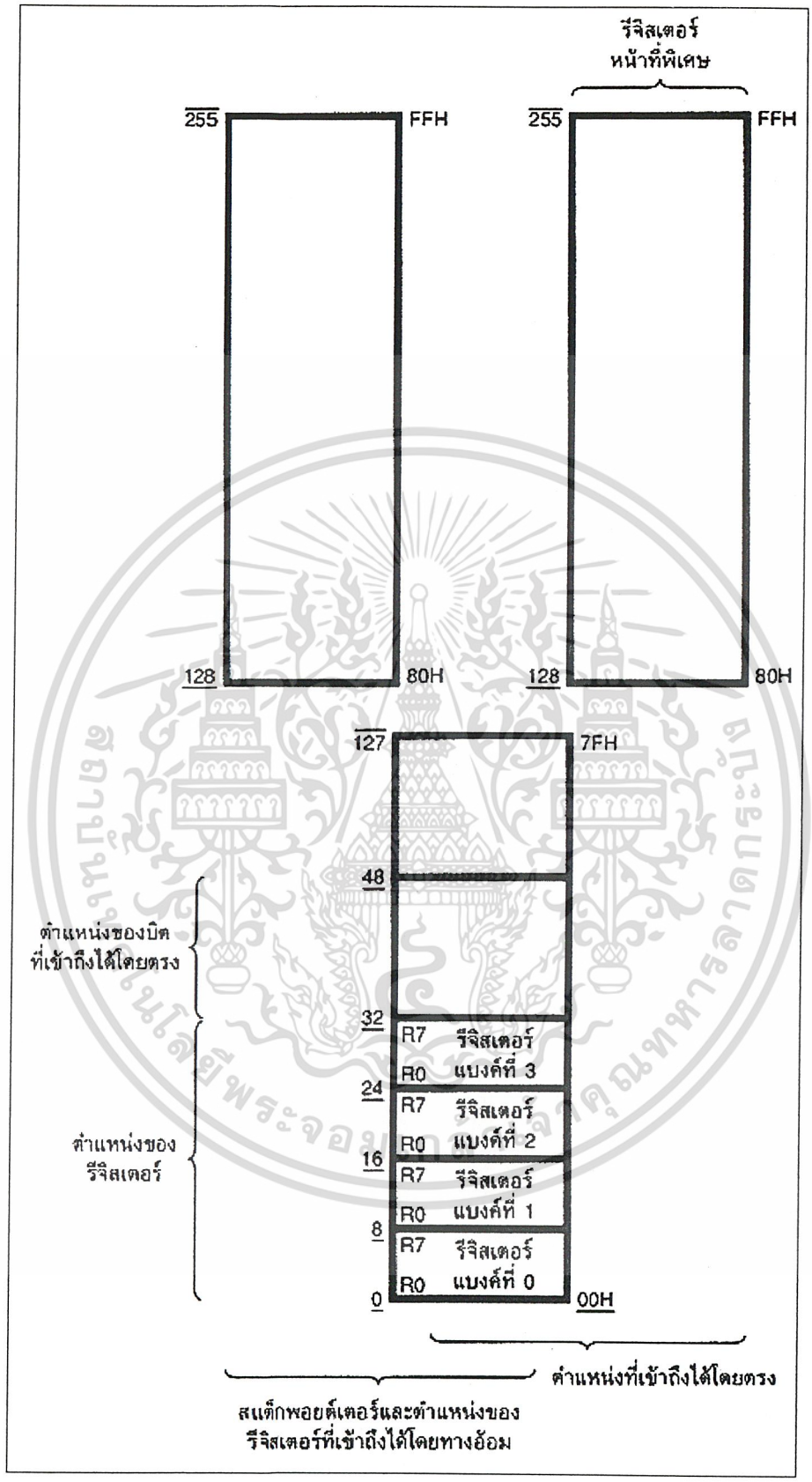


รูปที่2.10 แสดงการจัดตำแหน่งแอดเดรสของหน่วยความจำข้อมูล

แรมภายในส่วนล่างสุดแบ่งออกเป็นรีจิสเตอร์แบงค์ (register bank) อยู่ 4 แบงค์ ซึ่งแต่ละแบงค์จะประกอบด้วยรีจิสเตอร์ใช้งานแบบทั่วไปอยู่ 8 ตัว ซึ่งอยู่ที่ตำแหน่ง 0H ถึง 1FH ส่วนตำแหน่ง 16 ไบต์ถัดไปคือตำแหน่งที่ 20H ถึง 2FH เป็นตำแหน่งของหน่วยความจำ ที่สามารถอ้างถึงแบบบิตได้โดยตรงจำนวน 128 บิต ขอเน้นย้ำสักนิดว่าเป็นบิต ไม่ใช่ไบต์ รูปที่ 11 แสดงการจัดตำแหน่งแอดเดรสของแรมส่วนล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่2.11 แสดงการจัดตำแหน่งแอดเดรสของแรมส่วนล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

49875

2.4.3 รีจิสเตอร์เฉพาะ (Special Function Register)

ตารางที่ 2.3 จิสเตอร์เฉพาะ (Special Function Register)

สัญลักษณ์	ชื่อ	ตำแหน่ง
*ACC	Accumulator	E0H
*B	B Register	F0H
*PSW	Program Status Word	D0H
SP	Stack Pointer	81H
DPTR	Data Pointer	
DPL	Low Byte	82H
DPH	High Byte	83H
*PO	Port 0	80H
*P1	Port 1	90H
*P2	Port 2	A0H
*P3	Port 3	B0H
*IP	Interrupt Priority Control	B8H
*IE	Interrupt Enable	A8H
TMOD	Timer / Counter Mode Control	89H
*TCON	Timer / Counter Control	88H
*+T2CON	Timer / Counter 2 Control	C8H
TH0	Timer / Counter 0 High Byte	8CH
TL0	Timer / Counter 0 Low Byte	8AH
TH1	Timer / Counter 1 High Byte	8DH
TL1	Timer / Counter 1 Low Byte	8BH
+ TH2	Timer / Counter 2 High Byte	CDH
+ TL2	Timer / Counter 2 Low Byte	CCH
*SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H

* สามารถอ้างอิงตำแน่งข้อมูลแบบบิตได้

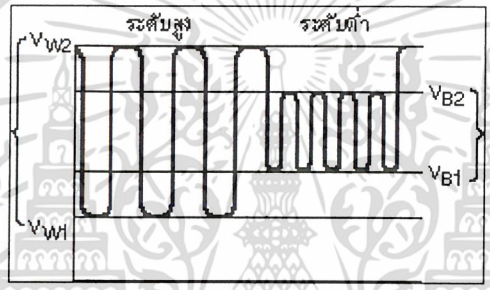
+ มีเฉพาะ ใน 8052 และ 8032

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในไมโครคอนโทรลเลอร์ MCS-51 มีรีจิสเตอร์ต่าง ๆ ที่ทำหน้าที่เฉพาะอย่างซึ่งเรียกว่า Special Function Register (SFR) อยู่หลายตัวซึ่งอยู่ในตำแหน่งของหน่วยความตั้งแต่ 80H ขึ้นไป การติดต่อกับรีจิสเตอร์เหล่านี้สามารถอ้างถึงตำแหน่งที่อยู่ได้โดยตรง (Direct Addressing) และบางตัวสามารถอ้างตำแหน่งข้อมูลแบบบิตได้ ตำแหน่งของรีจิสเตอร์ต่าง ๆ ดังแสดงในตารางที่ 2.3

2.5 การทำงานของบาร์โค้ด

ตัวอ่านบาร์โค้ดหรือสแกนเนอร์เป็นอุปกรณ์ที่ใช้ในการตรวจจับบาร์โค้ดซึ่งจะให้เอาต์พุตเป็นสถานะ ต่ำ เมื่อพบแถบสีดำ และมีสถานะสูงเมื่อพบสีขาว รูปแบบของสัญญาณ ดังรูปที่ 2.12 ซึ่งทำหน้าที่เหมือน ตาในระบบบาร์โค้ด โดยการเปลี่ยนแถบเส้นขาวดำที่เห็นให้เป็นสัญญาณทางไฟฟ้าคือแรงดันเมื่อผ่านแถบขาว และ V_{w1-2} คือแรงดันเมื่อผ่านแถบขาว และ V_{B1-2} คือแรงดันเมื่อผ่านแถบดำ



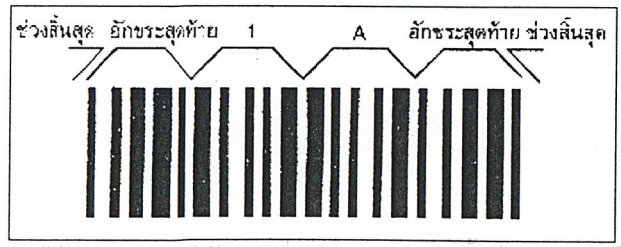
รูปที่ 2.12 ระดับแรงดันทางเอาต์พุตของตัวอ่านบาร์โค้ด

2.5.1 รูปแบบรหัสของบาร์โค้ด

2.5.1.1 รหัส 3ใน 9 หรือรหัส 39

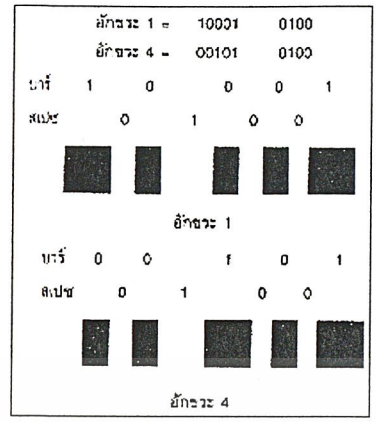
รหัส 39 ประกอบด้วยส่วนประกอบแถบกว้าง 3 ส่วน ซึ่งเป็นแถบทึบหรือบาร์ และแถบว่างหรือ สเปซ จากทั้งหมด 9 ส่วน ดังแสดงในรูปที่ 13 และรูปที่ 14 ซึ่งในบาร์โค้ดจะประกอบด้วย

1. ช่วงแถบว่างที่อยู่แต่ละด้านของบาร์โค้ด
2. ส่วนแสดงการเริ่มต้นและหยุดของตัวอักขระ
3. ข้อมูลของตัวอักขระ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.13 รูปแบบของการเข้ารหัส "1A"



รูปที่ 2.14 อักขระในบาร์โค้ด 3 ใน 9

ตัวอย่างของข้อมูลบาร์โค้ด ตัวอักษรทั้งหมด ของรหัส 39 แสดงไว้ในตารางที่ 2.4 แถบและช่องว่างหรือว่าบาร์และสเปซแต่ละอันสามารถเลือกได้ ซึ่งจะแคบ หรือกว้างขึ้นอยู่กับการแปลงรหัส ตัวอักขระแต่ละตัวประกอบด้วยแถบกว้าง 3 แถบ และแถบแคบ 6 แถบ ตัวเลข 1 ใช้แทนส่วนกว้าง และเลข 0 แทนส่วนแคบ ตัวอักษรอื่นๆ แบ่งแยกโดยช่องว่างระหว่างตัวอักษร

ตารางที่ 2.4 โครงสร้างตัวอักขระในรหัส USS-39

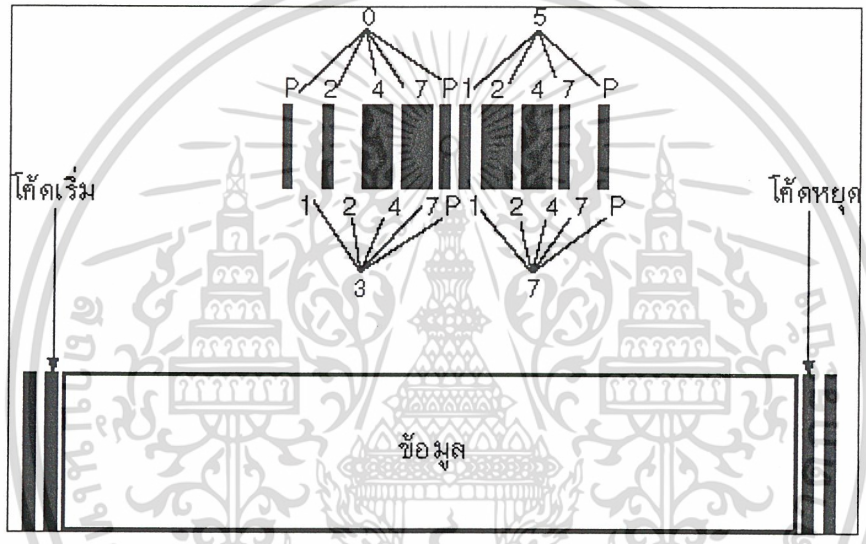
ลัวยักขระ	รูปแบบ	บาร์	สเปซ	ลัวยักขระ	รูปแบบ	บาร์	สเปซ
1		10001	0100	M		11000	0001
2		01001	0100	N		00101	0001
3		11000	0100	O		10100	0001
4		00101	0100	P		01100	0001
5		10100	0100	Q		00011	0001
6		01100	0100	R		10010	0001
7		00011	0100	S		01010	0001
8		10010	0100	T		00110	0001
9		01010	0100	U		10001	1000
0		00110	0100	V		01001	1000
A		10001	0010	W		11000	1000
B		01001	0010	X		00101	1000
C		11000	0010	Y		10100	1000
D		00101	0010	Z		01100	1000
E		10100	0010	.		00011	1000
F		01100	0010	-		10010	1000
G		00011	0010	SPACE		01010	1000
H		10010	0010	@		00110	1000
I		01010	0010	$		00000	1110
J		00110	0010	/		00000	1101
K		10001	0001	+		00000	1011
L		00001	0001	%		00000	0111

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1.2 รหัสแทรก 2 ใน 5

เฉพาะข้อมูลตัวเลขเท่านั้นที่สา-มารถเข้ารหัสแบบแทรก 2 ใน 5 ความหนาแน่นของข้อมูล สูงสุดคือ 18 ตัวอักขระต่อนิ้ว รหัสอาจผิดพลาดได้ หากไม่เข้ารหัสเป็นตัวเลขคู่ของตัวอักษรเข้ารหัส ในสัญลักษณ์ ตัวอักษรตัวแรกของคู่แทนโดยบาร์ และตัวอักษรตัวที่ 2 แทนโดยสเปซ รหัสแทรก 2 ใน 5 ในอุตสาหกรรมผลิตยา, ในร้านอาหารและอุตสาหกรรมผลิตสิ่งพิมพ์ ฯลฯ

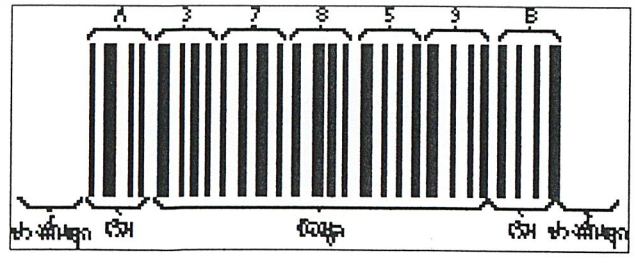
รูปแบบสำหรับตัวอักขระจะประกอบด้วยบาร์และสเปซสลับกันไป แทนค่าตัวเลขฐานสิบในรูปของ ตัวเลขฐานสอง 5 บิต (4 บิตแสดงค่าตัวเลข และ 1 บิตพาริตี้) รูปแบบของบาร์โค้ดจะประกอบด้วยส่วน เริ่มต้น, ส่วนของข้อมูลและส่วนแสดงการสิ้นสุดของรหัส การเข้ารหัสตัวเลขต้องประกอบด้วยจำนวนคู่เสมอ ตัวเลขที่เข้ารหัสแล้ว ดังแสดงในรูปที่ 2.15



รูปที่ 2.15 รูปแบบของบาร์โค้ดรหัสแทรก 2 ใน 5

2.5.1.3 รหัสแบบ Coda bar

Coda bar สามารถใช้กับข้อมูลตัวเลขและตัวอักขระพิเศษอีก 6 ตัวคือ $,-,:,/,., และ + และตัวอักษร 4 ตัวที่แสดงการเริ่มต้นและหยุด คือ A, B, C และ D สัญลักษณ์ของ Coda bar ใช้สำหรับเปลี่ยนแปลงความยาว ของข้อมูล ซึ่งคุณสมบัติต่างๆ ได้ให้ไว้ในตารางที่ 5 ส่วนประกอบของข้อมูลแต่ละตัวจะแสดงให้ในรูปที่ 2.16



รูปที่ 2.16 การเข้ารหัส Coda bar "A37859B"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

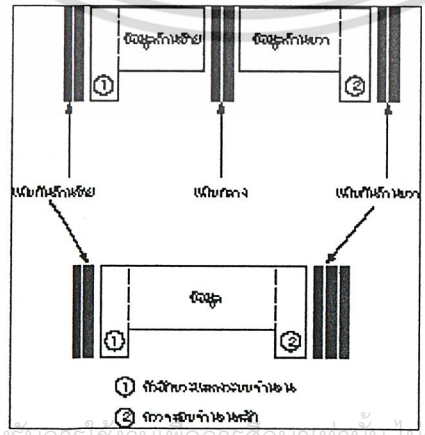
Coda bar แต่ละตัวประกอบด้วย ขอบเขตแสดงการสิ้นสุด, ส่วนแสดงการเริ่มต้นหรือหยุดและ ส่วนของข้อมูล ซึ่งข้อมูลสามารถเปลี่ยนแปลงความยาวได้ถึง 32 ตัวอักษร ตัวอย่างดังแสดงในรูปที่ 2.14 ตัวอักษร แต่ละตัวแทนโดยส่วนประกอบ 7 ส่วน มี 4 บาร์ และ 3 สเปซ ระหว่างแถบตัวอักขระแสดงว่า เริ่มต้นหรือหยุดมี 4 ตัว สามารถใช้เป็นตัวเริ่มต้นหรือหยุดได้ ส่วนประกอบการเพิ่มข้อมูลสัญลักษณ์ ภายในตัวอักขระแบ่งโดยช่องว่างระหว่างตัวอักขระประกอบด้วยส่วน ของช่องว่างแคบๆ 1 ช่อง ส่วนประกอบเบื้องต้นสำหรับตัวอักษร Coda bar ทั้งหมด แสดงไว้ในตารางที่ 2.5 ซึ่งส่วนประกอบกว้าง แทนไบนารี 1 และส่วนประกอบแคบ แทนไบนารี 0 แต่ละตัวอักขระสามารถแทนโดยไบนารีขนาด 7 บิตเท่านั้น

ตารางที่ **2.5** รูปแบบบาร์โค้ด Codabar และค่าของตัวเลขในระบบฐานสอง

ตัวอักขระที่เข้ารหัส	ค่าในรูปของเลขฐานสอง	แพตเทิร์นของบาร์และสเปซ
0	0000011	
1	0000110	
2	0001001	
3	1100000	
4	0010010	
5	1000010	
6	0100001	
7	0100100	
8	0110000	
9	1001000	
-	0001100	
$	0011000	
:	1000101	
/	1010001	
.	1010100	
+	0010101	
A	0011010	
B	0101001	
C	0001011	
D	0001110	

2.5.1.4 รหัสสากล UPC (Univer-sal Product Code)

บาร์โค้ด UPC ชุดอักขระประกอบด้วยตัวเลขและอีก 3 ส่วนพิเศษ คือส่วนเริ่มต้น, ส่วนหยุดและ ตัวอักขระตัวอักษรแต่ละตัวสร้างขึ้นโดย 4 ส่วน คือ 2 บาร์ และ 2 สเปซ มี 2 ตัวอย่างข้อมูลพื้นฐาน ดังแสดงในรูปที่ 2.17



รูปที่ 2.17 รูปแบบการเข้ารหัส UPC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1.5 รหัสตัวเลขของยุโรป EAN (European Article Numbering)

ลักษณะรหัสตัวเลขของยุโรป ที่ใช้ในทวีปยุโรป ซึ่งมันจะใช้คู่กับรหัส UPC เป็นรหัสที่ใช้ สำหรับข้อมูลตัวเลข ตัวอย่างดังแสดงในรูปที่ 2.18 ซึ่งเป็นรหัส EAN-13

รูปที่ 2.18 รูปบาร์โค้ดรหัส EAN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ขั้นตอนและวิธีการดำเนินการ

3.1 การวางแผนการดำเนินงาน

1) ศึกษาข้อมูลและแนวทาง
2) วิเคราะห์และรวบรวมข้อมูล
3) การจัดซื้ออุปกรณ์และการสร้าง
4) ทดสอบการทำงานและติดตั้ง
5) วิเคราะห์ สรุปผล และแก้ไขข้อผิดพลาด
6) จัดทำคู่มือการใช้งาน
7) จัดทำปริญญานิพนธ์

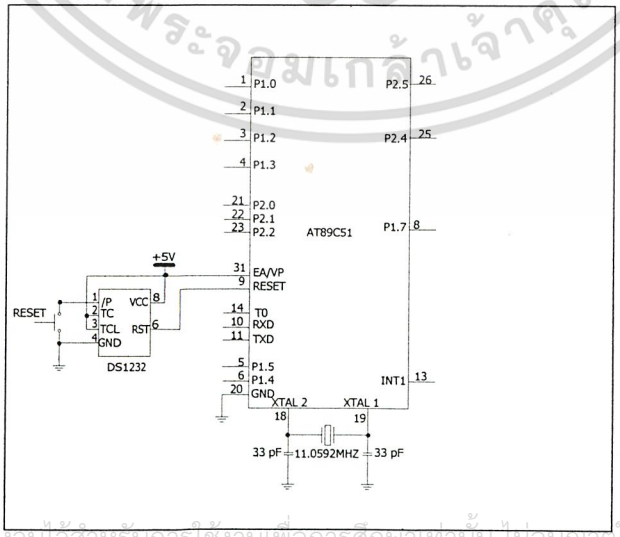
3.2 การออกแบบและส่วนประกอบ

ในการดำเนินออกแบบสร้างเครื่องตรวจสอบและเก็บข้อมูลการเข้าออกประตูด้วยรหัสแถบ สามารถแบ่งขั้นตอนการดำเนินงานได้เป็น 2 ส่วนดังนี้

3.2.1 ฮาร์ดแวร์

3.2.1.1 วงจรไมโครคอนโทรลเลอร์

ในการออกแบบปริญญานิพนธ์ชุดนี้ ได้เลือกใช้ไมโครคอนโทรลเลอร์ เบอร์ AT89C52 ของบริษัท Atmel เป็นตัวประมวลผลกลาง เนื่องจากเป็นไมโครคอนโทรลเลอร์แบบชิปเดี่ยว ทำให้ง่ายต่อการจัดพอร์ต และทำให้ขนาดของวงจรเล็กลงด้วยและไอซี AT89C52 ตัวนี้มีหน่วยความจำภายใน 12 กิโลไบต์ ซึ่งมากพอในการเขียนโปรแกรมควบคุม และไม่ทำให้ถูกจำกัดพื้นที่หน่วยความจำในการเขียนโปรแกรมมากเกินไป

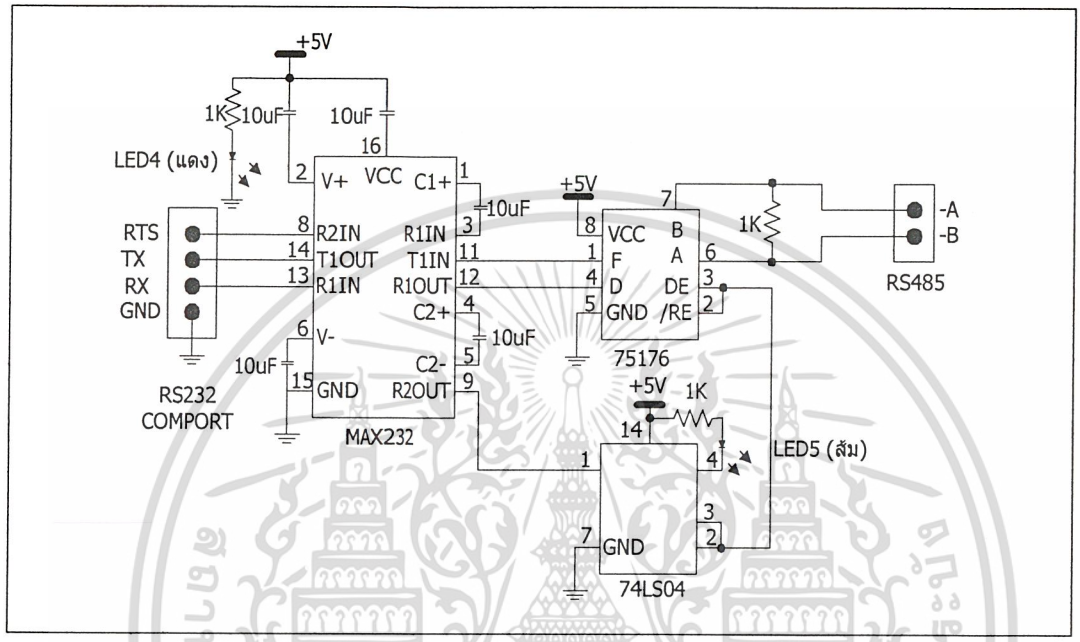


รูปที่ 3.1 การต่อใช้งานไมโครคอนโทรลเลอร์เบอร์ AT89C52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

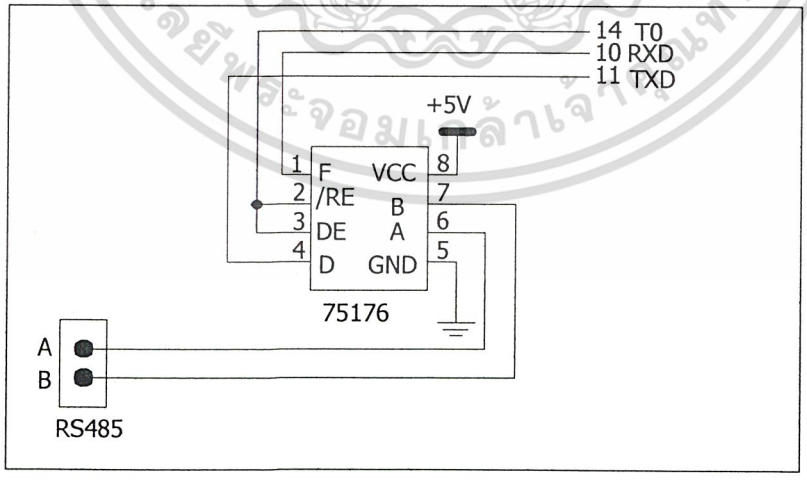
3.2.1.2 การออกแบบวงจรการพอร์ตอนุกรม RS- 232 เป็นRS – 485

วงจรสมบูรณ์ของวงจรจะแสดงดังรูปที่ 3.2 วงจรนี้จะใช้ไอซี MAX –2 คอยทำหน้าที่แปลงแรงดันในมาตรฐาน RS – 232 ให้อยู่ในรูปทีทีแอลและในทางกลับกันคือ แปลงแรงดันอินพุตที่ป้อนเข้า MAX 232 ให้อยู่ในรูปแบบมาตรฐาน RS – 232 เพื่อส่งให้คอมพิวเตอร์ ไอซี 75176 ทำหน้าที่แปลงสัญญาณให้เป็นมาตรฐาน RS – 485 โดยเป็นการส่งข้อมูลแบบฮาร์ฟดูเพล็กซ์ใช้สาย RTS มาทำหน้าที่ควบคุมทิศทางของการสื่อสาร



รูปที่ 3.2 การติดต่อกับคอมพิวเตอร์จะใช้พอร์ตอนุกรม RS – 232 เป็น RS – 485

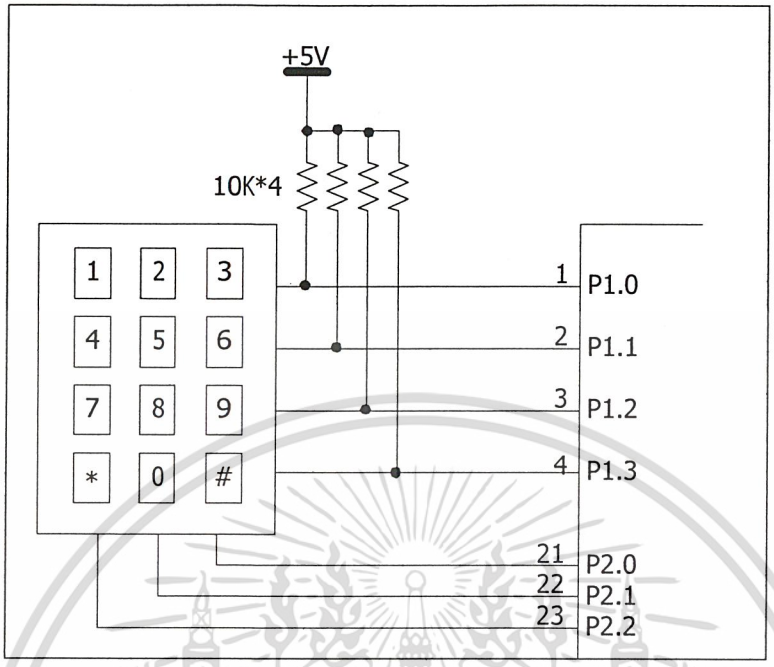
3.2.1.3 การออกแบบวงจรการติดต่อกับพอร์ตอนุกรม RS – 485



รูปที่ 3.3 การติดต่อกับคอมพิวเตอร์จะใช้พอร์ตอนุกรม RS – 485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

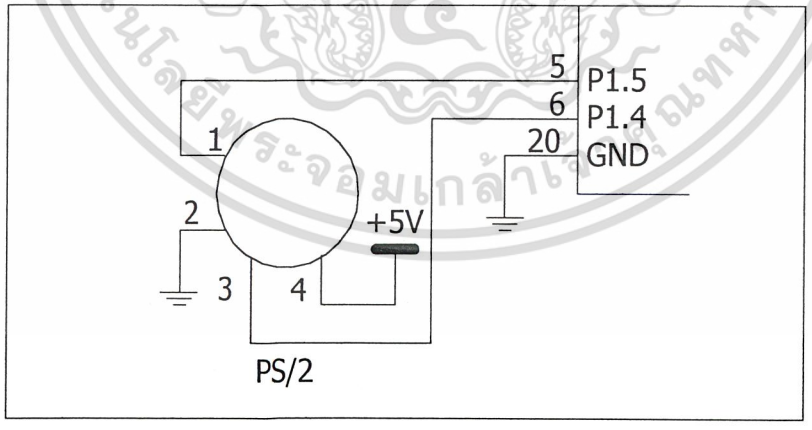
3.2.1.4 วงจรแป้นพิมพ์ 3*4



รูปที่ 3.4 วงจรแป้นพิมพ์

จากรูปที่ 3.4 อินพุตของวงจร จะต่อแบบเมตริกซ์ โดยมีสัญญาณจากแถว (Row) 3 หลัก และสดมภ์ (Column) 4 หลัก ซึ่งสามารถรวมสัญญาณออกมาได้ 12 ตัว ตั้งแต่ตัวเลข 0-9 และ * กับ #

3.2.1.5 การติดต่อพอร์ต PS/ 2 ของเครื่องอ่านบาร์โค้ด



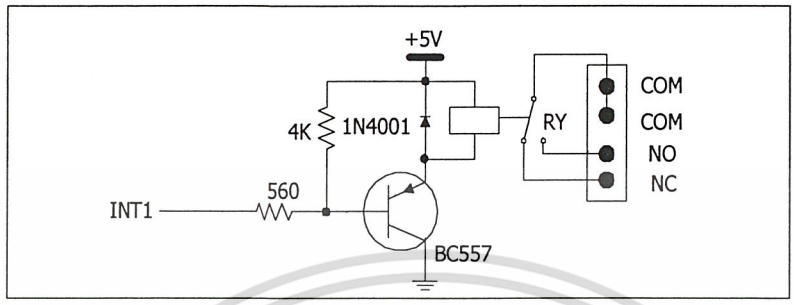
รูปที่ 3.5 วงจรการต่อพอร์ต PS/ 2ของเครื่องอ่านบาร์โค้ด

จากรูปที่3.5 ขา 1 ของพอร์ต PS/ 2 จะส่งข้อมูลที่อ่านได้พอร์ต P1.5 ของไมโครคอนโทรลเลอร์ ขา 2 จะต่อลงกราวด์กับวงจร ขา 3 จะรับสัญญาณนาฬิกา (Clock) จากพอร์ต P1.4 ของไมโครคอนโทรลเลอร์ ขา 4 จะรับไฟ +5 V จากวงจรมาจ่ายให้กับเครื่องอ่านบาร์โค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.6 วงจรการทำงานรีเลย์

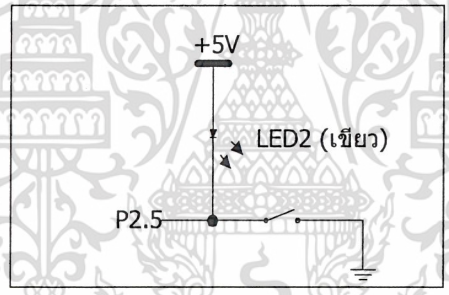
โดยจะทำงานเมื่อมีสัญญาณจากพอร์ต INT 1 ของ AT89C52 จะทำให้ทรานซิสเตอร์นำกระแส ทำให้แรงดันไฟฟ้า +5V สร้างสนามแม่เหล็กในรีเลย์ โดยมีไดโอดต่อขนานกับรีเลย์เพื่อป้องกันกระแสหมุนเวียนภายในเมื่อทรานซิสเตอร์นำกระแส ดังแสดงในรูปที่ 3.6



รูปที่ 3.6 วงจรการทำงาน รีเลย์

3.2.1.7 การต่อสวิตช์เปิดประตู

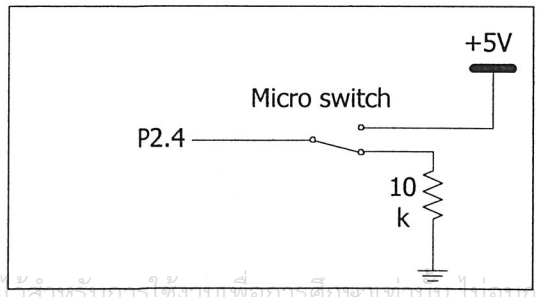
สวิตช์เปิดประตูจะติดตั้งอยู่ภายในห้องโดยจะทำหน้าที่สั่งงานเปิดประตูจากด้านใน ดังแสดงในรูปที่ 3.7



รูปที่ 3.7 การต่อสวิตช์เปิดประตู

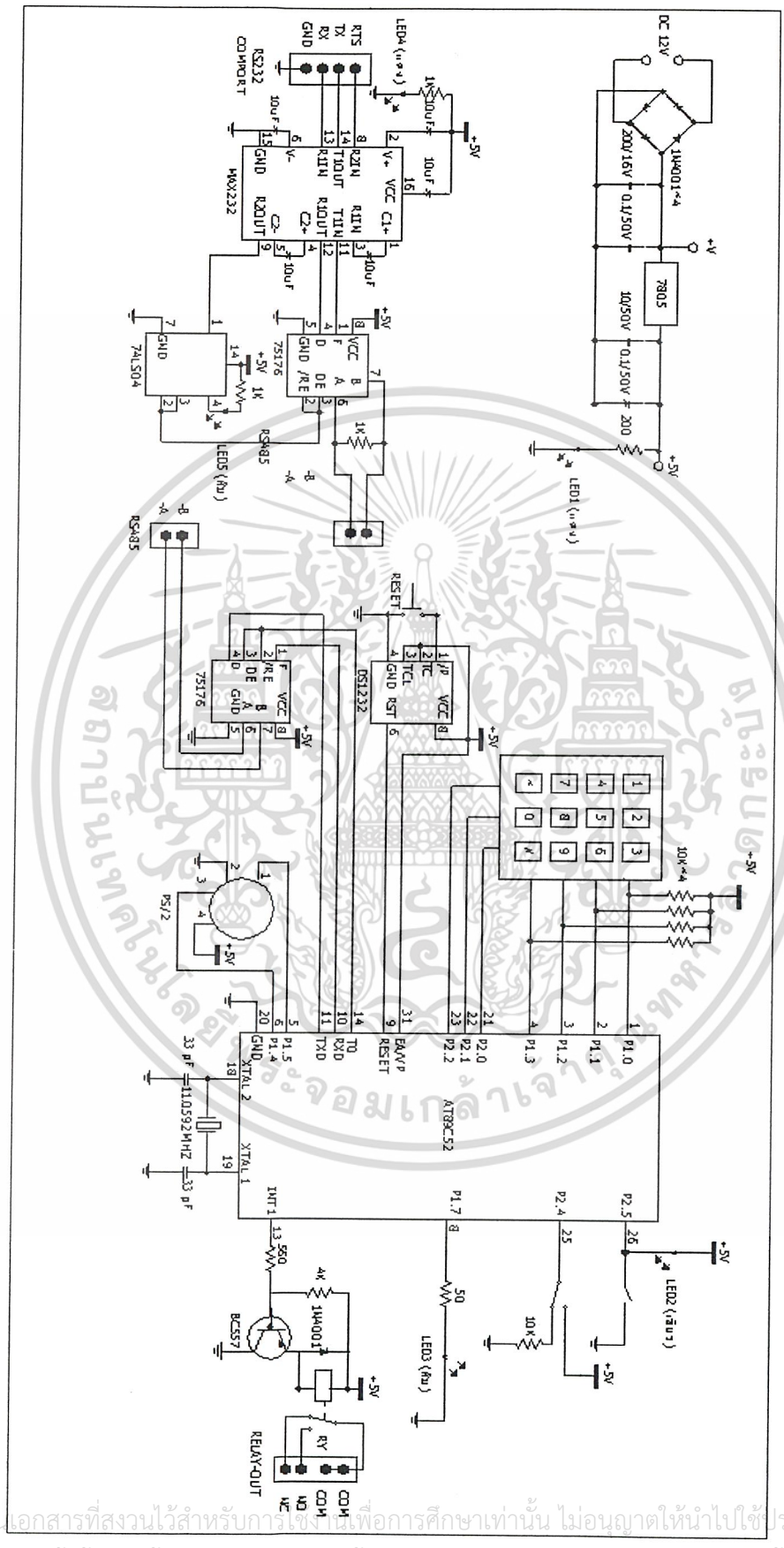
3.2.1.8 การต่อไมโครสวิตช์

ไมโครสวิตช์ จะติดตั้งอยู่กับอุปกรณ์ล๊อค โดยจะทำหน้าที่ตรวจสอบสถานะการเปิด-ปิดประตู ถ้าหากประตูปิด ไมโครสวิตช์ ก็จะต่อพอร์ต P2.4 ของไมโครคอนโทรลเลอร์ลงกราวด์ และเมื่อประตูเปิดออก ไมโครสวิตช์ จะทำให้พอร์ต P2.4 ต่อกับแหล่งจ่ายไฟ +5 V ดังแสดงในรูปที่ 3.8



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.8 การต่อไมโครสวิตช์

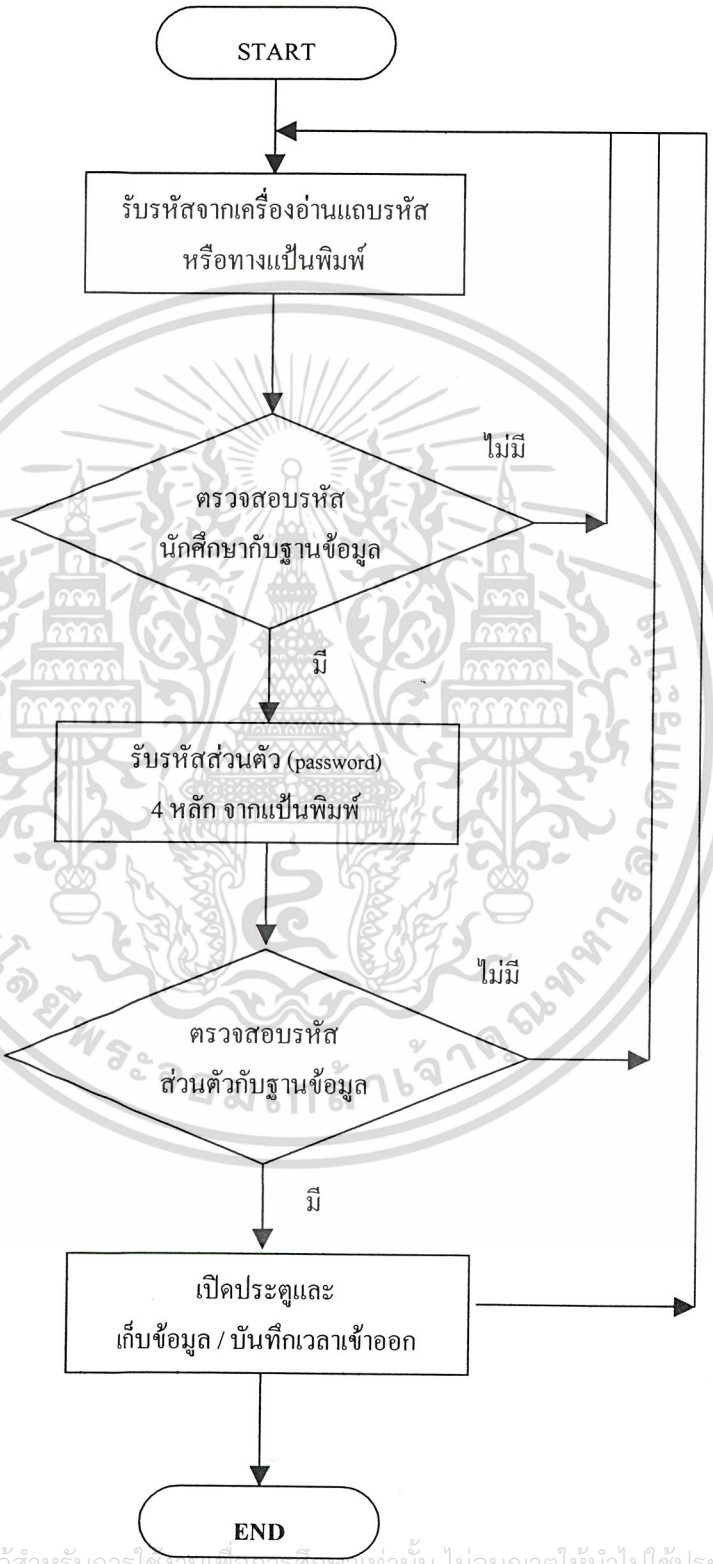


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.9 วงจรการต่อไมโครคอนโทรลเลอร์

3.2.2 ซอฟต์แวร์

3.2.2.1 แผนผังการทำงานของโปรแกรมไมโครซอฟต์วิชวลเบสิก ในคอมพิวเตอร์ส่วนกลาง

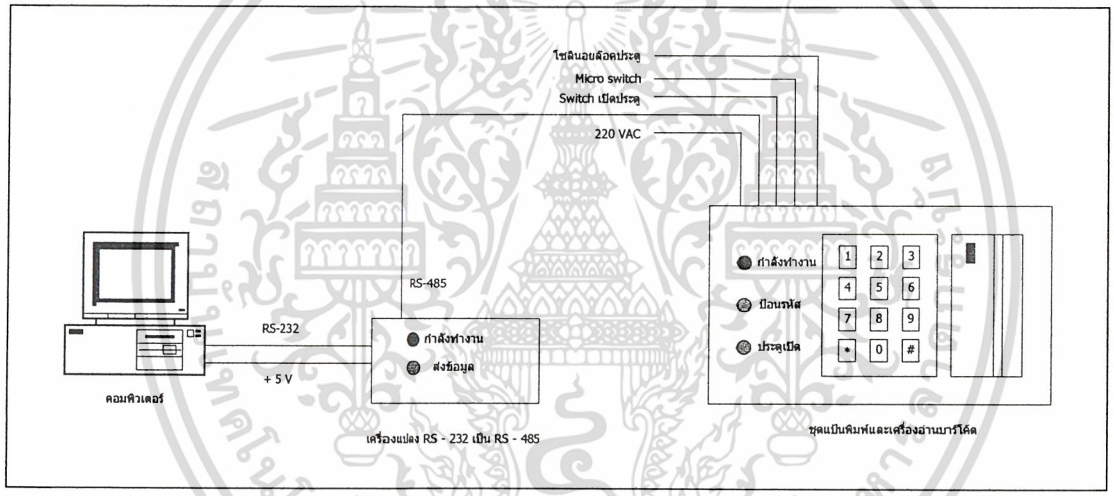


รูปที่ 3.10 Flowchart แสดงรูปแบบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเขียนโปรแกรมเพื่อรับ – ส่งข้อมูลระหว่าง เครื่องลูกข่าย และคอมพิวเตอร์ มีลำดับการทำงานซึ่งสามารถอธิบายได้จากรูปที่ 3.10 เมื่อเริ่มต้นโปรแกรมคอมพิวเตอร์ส่วนกลางจะรับข้อมูลจากชุดเครื่องอ่านบาร์โค้ดหรือแป้นพิมพ์ เมื่อมีข้อมูลส่งมาจากเครื่องอ่านบาร์โค้ด คอมพิวเตอร์ส่วนกลางจะทำการตรวจสอบรหัสที่ได้รับมาจากเครื่องอ่านบาร์โค้ด โดยนำไปเปรียบเทียบกับฐานข้อมูลภายในระบบฐานข้อมูล ถ้าไม่มีข้อมูลภายในระบบฐานข้อมูล คอมพิวเตอร์ส่วนกลางจะกลับไปรอรับข้อมูลใหม่อีกครั้ง แต่ถ้าคอมพิวเตอร์ส่วนกลางตรวจสอบพบรหัสที่ชุดเครื่องอ่านบาร์โค้ดส่งมาให้ จะทำการรอรับรหัสส่วนตัว(password) 4 หลักทางแป้นพิมพ์ หลังจากนั้นคอมพิวเตอร์ส่วนกลางจะทำการตรวจสอบรหัสส่วนตัวที่ได้รับมาจากทางแป้นพิมพ์ โดยนำไปเปรียบเทียบกับฐานข้อมูลภายในระบบฐานข้อมูล ถ้าไม่มีข้อมูลภายในระบบฐานข้อมูล คอมพิวเตอร์ส่วนกลางจะกลับไปรอรับข้อมูลใหม่ทั้งหมดอีกครั้ง แต่ถ้าคอมพิวเตอร์ส่วนกลางตรวจสอบพบรหัสส่วนตัวที่ได้รับมาจากทางแป้นพิมพ์ จึงทำการสั่งงานให้เปิดประตู

ในส่วนของการเปิดประตูออกจากทางด้านในห้อง ทำโดยการกดสวิตซ์ที่ได้ทำการติดตั้งไว้ภายในห้อง ที่ติดตั้งไว้โดยจะมีการหน่วงเวลาในการรอเปิดประตูไว้ที่ 5 วินาที หากไม่มีการเปิดประตูภายในเวลาที่กำหนดโปรแกรมก็จะทำการยกเลิกการเปิดประตูและจะกลับไปรับรหัสนักศึกษาต่อไป



รูปที่ 3.11 การออกแบบติดตั้งอุปกรณ์การทำงาน

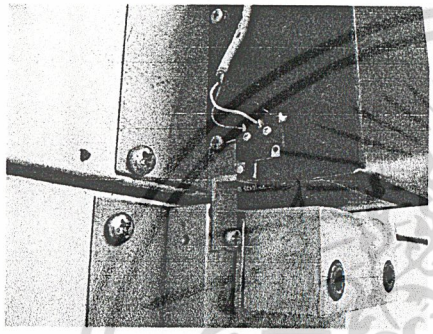
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

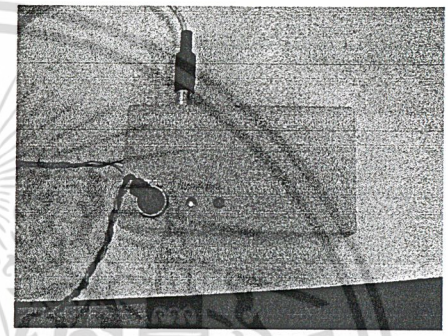
ผลการทดลอง

4.1 การติดตั้งและการทำงานของฮาร์ดแวร์

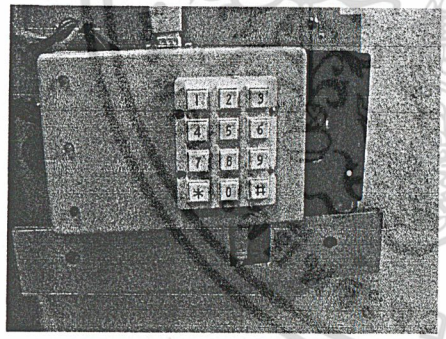
การติดตั้งฮาร์ดแวร์ผ่านทางสาย RS-232 แบบ 25 pin เข้ากับพอร์ต com2 ของคอมพิวเตอร์และแหล่งจ่ายไฟ +5V เข้ากับเครื่องแปลง RS - 232 เป็น RS - 485 ขั้นตอนต่อไปทำการต่อพอร์ต RS-485 ของเครื่องแปลง RS - 232 เป็น RS - 485 เข้ากับชุดแป้นพิมพ์และเครื่องอ่านบาร์โค้ด ดังรูปที่ 4.1



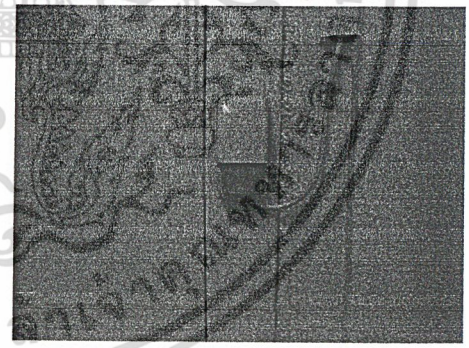
ก. ไมโครสวิตช์และอุปกรณ์ล็อค



ข. เครื่องแปลงพอร์ตอนุกรม RS – 232 เป็น RS – 485



ค. แป้นพิมพ์และเครื่องอ่านบาร์โค้ด



ง. สวิตช์เปิดประตูที่ติดตั้งด้านใน

รูปที่ 4.1 การติดตั้งอุปกรณ์การทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของฮาร์ดแวร์

1. นักศึกษาสามารถที่จะป้อนรหัสนักศึกษาจากเครื่องอ่านบาร์โค้ดโดยใช้บัตรนักศึกษาหรือจากแป้นพิมพ์โดยกดรหัสนักศึกษาแล้วตามด้วยปุ่ม # เพื่อทำการส่งข้อมูล

2. โปรแกรมจะทำการตรวจสอบรหัสนักศึกษากับฐานข้อมูล

3. ถ้าพบรหัสนักศึกษาตรงกับฐานข้อมูลโปรแกรมจะส่งสัญญาณให้ไฟสีส้มของชุดการทำงานแป้นพิมพ์และเครื่องอ่านบาร์โค้ด ทำงานเพื่อรอรับรหัสส่วนตัวเป็นเวลา 5 วินาที

4. ใส่รหัสส่วนตัวแล้วตามด้วยปุ่ม # เพื่อทำการส่งข้อมูล

5. โปรแกรมก็จะทำการตรวจสอบส่วนตัวกับฐานข้อมูล

6. ถ้ารหัสส่วนตัวถูกต้องโปรแกรมจะส่งสัญญาณให้ไฟสีเขียวของชุดการทำงานแป้นพิมพ์และเครื่องอ่านบาร์โค้ด ทำงานและปลดล็อคประตูเป็นเวลา 5 วินาที ถ้าหากไม่มีการเปิดประตูชุดล็อคก็จะทำการล็อคประตูโดยอัตโนมัติ

7. ในการออกจากห้องทำได้โดยกดสวิตช์ที่ติดตั้งด้านในจากนั้นชุดจะทำการปลดล็อคประตูเป็นเวลา 5 วินาที และถ้าหากไม่มีการเปิดประตูชุดล็อคก็จะทำการล็อคประตูโดยอัตโนมัติ

4.2 การทำงานของ โปรแกรม

โปรแกรมการทำงานจะเขียนด้วยโปรแกรม Microsoft Visual Basic 6 โดยติดตั้งไว้ที่ C:\Bar\Project2.exe โดยทำการกำหนด Baud rate ไว้ที่ 9600, N, 8, 1 เพื่อให้เป็นไปตามกำหนดของฮาร์ดแวร์

การใช้โปรแกรม

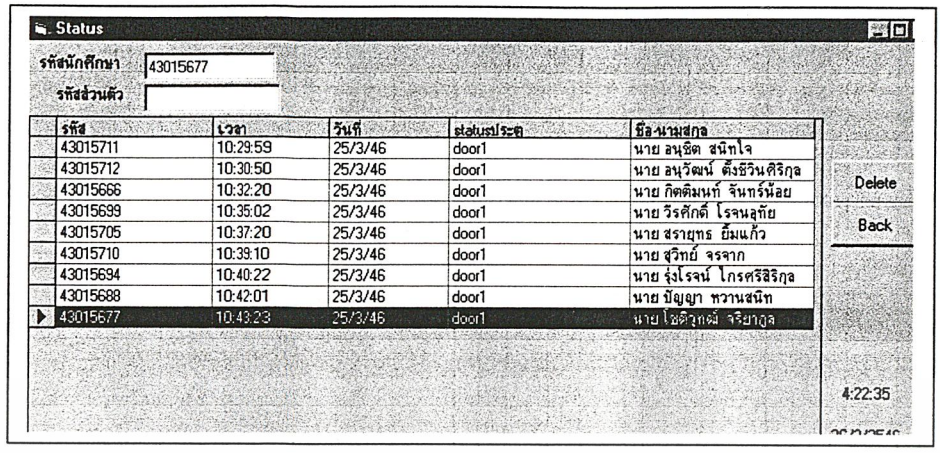
1. การเรียกโปรแกรมจาก C:\Bar\Project2.exe ก็จะปรากฏหน้าจอหลักขึ้นดังรูปที่ 4.2

รูปที่ 4.2 หน้าจอหลักโปรแกรม

2. กดปุ่ม Start เพื่อเข้าสู่โปรแกรม

ภายในโปรแกรมก็จะประกอบด้วย รหัสนักศึกษา เวลาการเข้าใช้งาน วันที่ ประตู และชื่อ-สกุลของผู้เข้าประตู และเมื่อมีการเปลี่ยนวันก็จะมีการจัดเก็บข้อมูลโดยอัตโนมัติและทำการลบข้อมูลเก่า แสดงดังรูปที่ 4.3

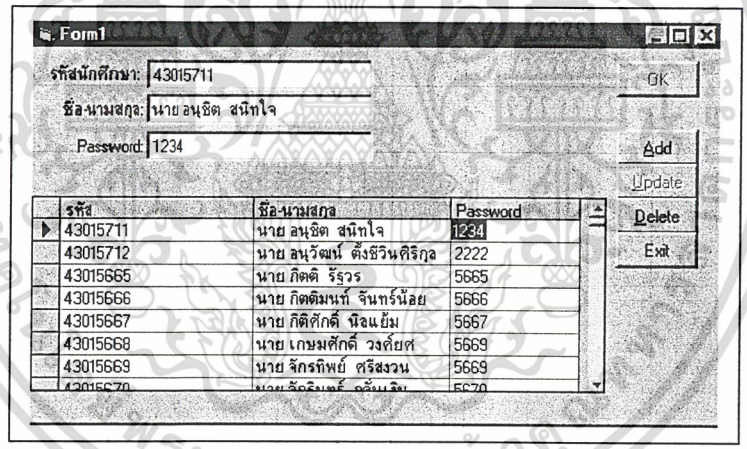
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 หน้าจอการทำงานของโปรแกรม

3. จากหน้าจอหลักกดปุ่ม Edit เพื่อแก้ไขหรือเพิ่มข้อมูลให้กับฐานข้อมูล

โดยกดปุ่ม Add จากนั้นใส่รหัสนักศึกษา ชื่อ-นามสกุล และPassword จนครบจากนั้น กดปุ่ม Update เพื่อจัดเก็บข้อมูล โดยแสดงดังรูปที่ 4.4

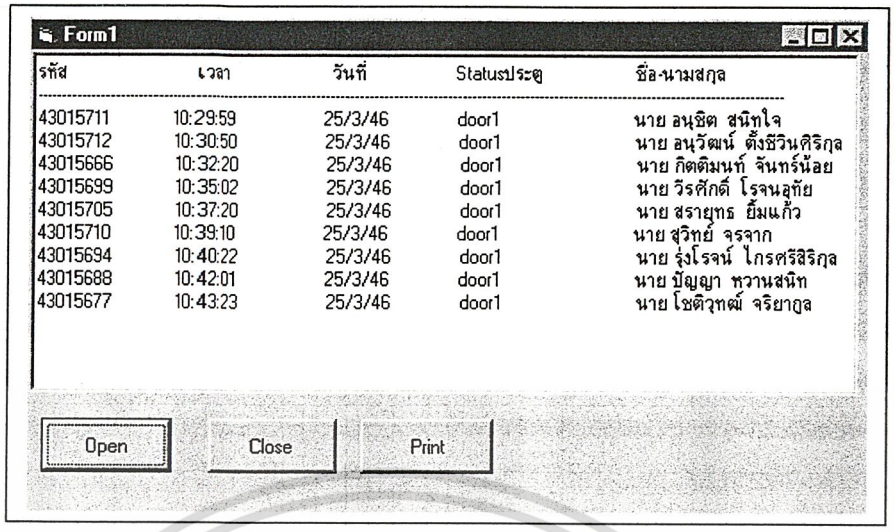


รูปที่ 4.4 การเพิ่มฐานข้อมูล

4. สามารถดูเข้ามูลการเข้าใช้ที่ได้ที่จัดเก็บไว้เป็นรายวันโดยกดปุ่ม View

แล้วกดปุ่ม Open ไปที่ c:\Bar\ โดยชื่อที่ใช้ในการจัดเก็บข้อมูลจะเป็นแบบ วันที่_เดือน_ปี ตัวอย่างเช่น ต้องการดูข้อมูลของวันที่ 25/3/46 ก็จะเป็น C:\Bar\25_3_46.doc และสามารถ Print ได้ ดังแสดงในรูปที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 การดูข้อมูลที่จัดเก็บไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

ปริญญานิพนธ์ฉบับนี้เป็นระบบรักษาความปลอดภัยแบบรหัสแถบโดยใช้หลักการควนคุมโดยไมครอนโทรลเลอร์ ซึ่งสามารถใช้ในการควบคุมการเข้าออกโดยการเปิด-ปิดประตูและสามารถจัดเก็บและบันทึกข้อมูลการปฏิบัติงานด้วยฐานข้อมูล และได้ทำการติดตั้งใช้งานจริงที่ห้องคอมพิวเตอร์ ภาควิชาวิศวกรรมอุตสาหการ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง พบว่าสามารถปฏิบัติงานได้ตามขั้นตอนที่ออกแบบไว้กล่าวคือเพื่อเพิ่มมาตรการรักษาความปลอดภัยภายในห้องคอมพิวเตอร์และความรวดเร็วในการค้นหาข้อมูลผู้เข้า – ออกห้องคอมพิวเตอร์และสร้างความมีระเบียบวินัยของนักศึกษาในการเข้าใช้ห้องคอมพิวเตอร์

5.2 ปัญหาและแนวทางแก้ไข

จากการติดตั้งใช้งานสามารถสรุปปัญหาและแนวทางการแก้ไขได้ 3 ประการ ได้แก่

1) ปัญหาจาก ไมโครสวิตต์ ประตู

โดยในขณะที่ประตูปิดในบางครั้งจะมีการสั่งงานปิดหน้าสัมผัสมากกว่า 1 ครั้ง ทำให้เกิดความผิดพลาดของโปรแกรมการทำงาน ซึ่งแก้ไขได้โดยการเปลี่ยนไมโครสวิตต์ และทำการแก้โปรแกรมการทำงาน

2) ปัญหาจากการกดแป้นพิมพ์

เมื่อยกนิ้วออกจากแป้นพิมพ์แล้วปุ่มของแป้นพิมพ์จะดีดตัวไปกดปุ่มอีกครั้งทำให้ข้อมูลที่ส่งไปเกิดการผิดพลาด

3) ปัญหาจากบาร์โค้ดของบัตรประจำตัวนักศึกษา

ชำรุดและลบจางลงไปทำให้เครื่องอ่านบาร์โค้ดไม่สามารถอ่านค่าได้หรืออ่านผิดพลาด แก้ไขโดยให้ป้อนรหัสประจำตัวนักศึกษาทางแป้นพิมพ์

5.3 ข้อเสนอแนะในการพัฒนา

ในการออกแบบเครื่องตรวจสอบและเก็บข้อมูลการเข้าออกประตูด้วยรหัสแถบจำเป็นต้องมีการปรับปรุงและพัฒนาในอีกหลายด้านซึ่งมีแนวทางพัฒนาโปรแกรมดังนี้

1) เขียนโปรแกรมการตรวจสอบการเข้าเรียนโดยให้มีการบันทึกการเข้าเรียนของนักศึกษาตามตารางเรียนและสามารถเรียกดูข้อมูลได้

2) การเชื่อมโยงกันทั้งหมดเข้ากับฐานข้อมูลของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังเพื่อให้ระบบรักษาความปลอดภัยที่สมบูรณ์และง่ายในตรวจสอบ

3) เพิ่มการแสดงผลด้วยหลอด LED 7 เซกแมนต์ (LED7 – segment)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. โปรแกรมไมโครคอนโทรลเลอร์ ภาษา C

```
#include <reg51.h>
//#include <absacc.h>

// Barcode Address //
sbit KB_CK = P1^5;
sbit KB_DA = P1^4;

sbit A = P1^0;
sbit BB = P1^1;
sbit C = P1^2;
sbit D = P1^3;
sbit Re=P1^7;
sbit Door = P1^6;
sbit Out1 = P2^0;
sbit Out2 = P2^1;
sbit Out3 = P2^2;
sbit SDoor = P2^5;
sbit SenDoor = P2^4;
sbit out   =P3^4;

unsigned char dat[20],i=0,co=0,tmp_dat[8];
unsigned char key[20],count=0;


void delay(unsigned int time_)
{
  unsigned int i;
  for (i = 0; i < time_; i++);
}
void init_serial ()
{
  IE = 0 ;
  SCON = 0x52 ;          // set RS232 parameter (9600)
//  SCON = 0x50 ;          // set RS232 parameter (9600)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    TMOD = 0x20 ;
    TH1  = 0xFD ;
    TL1 = 0XFD ;
//          TH1 = 0xFB ;
//   TL1 = 0XFB ;
    TCON = 0x40 ;
          out=0;
}
void send_rs232(char byte)
{
          TI=0;
          out=1;
          SBUF=byte;
          while(TI!=1) ;
          out=0;
}
void Opendoor()
{
                    Door=0;
                    INT1=0;
                    delay(10000);delay(10000);delay(10000);delay(10000);
                    delay(10000);delay(10000);delay(10000);delay(10000);
                    delay(10000);delay(10000);delay(10000);delay(10000);
                    delay(10000);delay(10000);delay(10000);delay(10000);
                    delay(10000);delay(10000);delay(10000);delay(10000);
                    delay(10000);delay(10000);delay(10000);delay(10000);
                    if(SenDoor==0){
                    while(SenDoor==0);
                    INT1=1;
                    }
                    else{INT1=1;}
                    Door=1;
}
void send_serial()
{
        unsigned char i;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if((key[0] == '1' )&&(key[1] == '1')&&(key[2] == '7')&&(key[3] == '5')&&(key[4] == '1')&&(key[5] ==
'0')&&(key[6] == '3')&&(key[7] == '4')){
        Opendoor();
        count=0;
        }
        else if((key[0] == '2' )&&(key[1] == '1')&&(key[2] == '7')&&(key[3] == '5')&&(key[4] == '1')&&(key[5]
== '0')&&(key[6] == '3')&&(key[7] == '4')){
        Opendoor();
        count=0;
        }
        else{
        if(count==8)send_rs232('Y');//select door
        delay(1000);
        for(i=0;i<count;i++)
        send_rs232(key[i]);
        }
        count=0;
}
unsigned char KB_Rx_Byte() { // receive byte from keyboard
        unsigned char i, buff, temp;
        buff = 0; temp = 0x01;
        while(KB_CK == 1) {
                        Out1=0;Out2=1;Out3=1;
        if((A==0)||(BB==0)||(C==0)||(D==0))
                {
                if(KB_CK == 0) break;
                if(A==0){ key[count]='1';count++;while(A==0){if(KB_CK == 0) break;}Out1=1;}
                else if(BB==0){ key[count]='4';count++;while(BB==0){if(KB_CK == 0) break;}Out1=1;}
                else if(C==0){ key[count]='7';count++;while(C==0){if(KB_CK == 0) break;}Out1=1;}
                else if(D==0){ key[count]='*';count++;while(D==0){if(KB_CK == 0) break;}Out1=1;}
                }
        else{
                if(KB_CK == 0) break;
                Out1=1;Out2=0;Out3=1;
                if((A==0)||(BB==0)||(C==0)||(D==0))
                        {
                        if(A==0){key[count]='2';count++;while(A==0){if(KB_CK == 0) break;}Out1=1;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                else if(BB==0){ key[count]='5';count++;while(BB==0){if(KB_CK == 0)
break;}Out1=1;}
                else if(C==0){ key[count]='8';count++;while(C==0){if(KB_CK == 0) break;}Out1=1;}
                else if(D==0){ key[count]='0';count++;while(D==0){if(KB_CK == 0) break;}Out1=1;}
                }
        else{
        if(KB_CK == 0) break;
                        Out1=1;Out2=1;Out3=0;
                        if((A==0)||(BB==0)||(C==0)||(D==0))
                        {
                        if(A==0){ key[count]='3';count++;while(A==0){if(KB_CK == 0)
break;}Out1=1;}
                        else if(BB==0){ key[count]='6';count++;while(BB==0){if(KB_CK == 0)
break;}Out1=1;}
                        else if(C==0){ key[count]='9';count++;while(C==0){if(KB_CK == 0)
break;}Out1=1;}
                        else if(D==0){send_serial();while(D==0){if(KB_CK == 0) break;}Out1=1;}
                        }
                }
        }
        if(RI)
        {
                if(SBUF=='A'){
                RI=0;
                delay(500);
                        if(RI){
                                if(SBUF=='2'){
                                Re=0;
                                Opendoor();
                                }
                                RI=0;
                        }
                }
        else if(SBUF=='B'){
                RI=0;
                delay(500);
                        if(RI){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    if(SBUF=='2'){
                    Re=1;
                    }
                RI=0;
                }
        }
        else if(SBUF=='r'){
                RI=0;
                delay(500);
                        if(RI){
                            if(SBUF=='2'){
                            Re=0;
                            }
                        RI=0;
                        }
                }
        }
        if(SDoor==0){
        while(SDoor==0);
        Door=0;
        INT1=0;
        delay(10000);delay(10000);delay(10000);delay(10000);
        delay(10000);delay(10000);delay(10000);delay(10000);
        delay(10000);delay(10000);delay(10000);delay(10000);
        delay(10000);delay(10000);delay(10000);delay(10000);
        delay(10000);delay(10000);delay(10000);delay(10000);
        delay(10000);delay(10000);delay(10000);delay(10000);
        if(SenDoor==0){
        while(SenDoor==0);
        INT1=1;
        }
        else{INT1=1;}
        Door=1;
        }
        Door=1;
        }
while(KB_CK == 0) {}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        while(KB_CK == 1) {}
        for (i = 1; i <= 8; i++) {
                while (KB_CK == 0) {}
                if (KB_DA == 1) buff |= temp;
                temp <<= 1;
                while (KB_CK == 1) {}
        }
        while(KB_CK == 0) {}
        while(KB_CK == 1) {}
        return(buff);
}
void send()
{
        unsigned char c;
        for(c=0;c<i;c++){
        if(dat[c]==0xF0){
        tmp_dat[co]=dat[c-1];
        co++;
        }
        }
        if(co==8){
        send_rs232('Y');//select door
        delay(1000);
        for(c=0;c<8;c++) send_rs232(tmp_dat[c]);
        }
        co=0;i=0;
}

void KB2ASCII(unsigned char key) { //!!!
        switch(key) {

        case 0x16 : dat[i]='1';i++;break;
        case 0x1E :dat[i]='2';i++;break;
        case '&' : dat[i]='3';i++;break;
        case '%' : dat[i]='4';i++;break;
        case '.' : dat[i]='5';i++;break;
        case '6' : dat[i]='6';i++;break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            case '=' : dat[i]='7';i++;break;
            case '>' : dat[i]='8';i++;break;
            case 'F' : dat[i]='9';i++;break;
            case 'E' : dat[i]='0';i++;break;
            case 0x5A: send();break;
            case 0xF0: dat[i]=0XF0;i++;;break;
            }
}
void main()
{
            unsigned char buff=0x00;
            delay(10000);
            init_serial ();
            i=0;Re=0;
            Door=1;
            INT1=1;
            while(1)
            {
                        buff = KB_Rx_Byte();
                        KB2ASCII(buff);
             }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โปรแกรมการทำงานโดย Microsoft Visual Basic 6

main.frm

```
Private Sub Closecomport_Click()
If Status.MSComm1.PortOpen Then
Status.MSComm1.PortOpen = False
MsgBox "Com " & Str(a) & " Is Close"
End If
End Sub

Private Sub Edit_Click()
Editform.Show
Unload Me
End Sub

Private Sub Exit_Click()
End
End Sub

Private Sub Oncom_Click()
On Error GoTo Errlable
Status.MSComm1.Settings = "9600,n,8,1"
Status.MSComm1.CommPort = 2 '2
Status.MSComm1.RThreshold = 1
Status.MSComm1.RTSEnable = False
If Status.MSComm1.PortOpen Then
Status.MSComm1.PortOpen = False
Else
Status.MSComm1.PortOpen = True
MsgBox ("ComPort " & 2 & "is Open")
End If
Status.Show
Unload main
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Exit Sub
Errlable:
If Err.Number = 8002 Then
MsgBox "Error on select com port"
ElseIf Err.Number = 8005 Then
   MsgBox "Com port in use"
Else
   MsgBox "Com Error"
End If
End Sub

Private Sub View_Click()
Viewdata.Show
Unload Me
End Sub
```

Status.frm

```
Dim door As String
Dim Count12 As Integer
Dim Count123 As Integer
Public Static Sub ShowData(Term As Control, Data As String)
   On Error GoTo Handler
   Const MAXTERMSIZE = 16000
   Dim TermSize As Long, i

   TermSize = Len(Term.Text)
   If TermSize > MAXTERMSIZE Then
     Term.Text = Mid$(Term.Text, 4097)
     TermSize = Len(Term.Text)
   End If

   Term.SelStart = TermSize
   Do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      i = InStr(Data, Chr$(8))
      If i Then
        If i = 1 Then
          Term.SelStart = TermSize - 1
          Term.SelLength = 1
          Data = Mid$(Data, i + 1)
        Else
          Data = Left$(Data, i - 2) & Mid$(Data, i + 1)
        End If
      End If
    Loop While i

    Do
      i = InStr(Data, Chr$(10))
      If i Then
        Data = Left$(Data, i - 1) & Mid$(Data, i + 1)
      End If
    Loop While i

    i = 1
    Do
      i = InStr(i, Data, Chr$(13))
      If i Then
        Data = Left$(Data, i) & Chr$(10) & Mid$(Data, i + 1)
        i = i + 1
      End If
    Loop While i

    Term.SelText = Data

    Term.SelStart = Len(Term.Text)

  Exit Sub
  Handler:
    MsgBox Error$
    Resume Next
  End Sub
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Command1_Click()
RichTextBox1.Text = "รหัส                เวลา            วันที่          Statusประตู          ชื่อ-นามสกุล"
ShowData RichTextBox1, "" & Chr(13)
RichTextBox1.Text = RichTextBox1.Text + "----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------"
ShowData RichTextBox1, "" & Chr(13)
With DataEnvironment1.rsCommand2
.MoveFirst
While (.EOF = False)
RichTextBox1.Text = RichTextBox1.Text + txtรหัส.Text + "          " + txtเวลา.Text + "              " + txtวันที่.Text
+ "          " + txtstatusประตู.Text + "                    " + Text2.Text
ShowData RichTextBox1, "" & Chr(13)
.MoveNext
Wend
End With
CommonDialog1.ShowSave
RichTextBox1.SaveFile CommonDialog1.FileName, rtfRTF
End Sub

Private Sub Command2_Click()
Dim t As Integer
With DataEnvironment1.rsCommand2
While (.EOF = False)
.MoveFirst
.Delete
.MoveNext
Wend
End With
End Sub

Private Sub Command3_Click()
main.Show
Unload Me
End Sub

Private Sub Form_Load()
Timer3.Enabled = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
door = "V"
D_Y = Year(Date)
D_M = Month(Date)
D_D = Day(Date)
Text6.Text = Str(D_D) & Str(D_M) & Str(D_Y)
End Sub

Private Sub MSComm1_OnComm()
 Dim userCriteria As String 'เงื่อนไขของผู้ใช้
Select Case MSComm1.CommEvent
   Case comEvReceive
   Dim Buffer As Variant
  Buffer = MSComm1.Input
  Text1.Text = ""
  Text1.Text = Text1.Text + Buffer
  Text7.Text = Str(Len(Text1.Text))
   End Select
  a = Len(Text1.Text)
  If (a = 1) Then
 door = Text1.Text
   ElseIf (a = 8) Then
      txtรหัส.Text = Text1.Text
      txtเวลา.Text = Label1.Caption
      txtวันที่.Text = Label2.Caption
      Text2.Text = "Not find"
      userCriteria = "รหัส  like '" & txtรหัส.Text & "'"
        If IsNumeric(txtรหัส.Text) Then
      With DataEnvironment1.rsCommand1
        .MoveLast 'ให้เคอร์เซอร์วิ่งไปที่เร็คคอร์ดแรกก่อน
       .Find userCriteria, , adSearchBackward
        End With
      End If
      If DataEnvironment1.rsCommand1.BOF Then 'ถ้าไม่พบรหัสที่ต้องการ
      Text4.Text = "off"
      Else
      Text4.Text = "On"
      Text2.Text = Text3.Text
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        If (door = "Z") Then
        Count123 = 20
        Timer4.Enabled = True
        ElseIf (door = "Y") Then
        Count123 = 21
        Timer4.Enabled = True
        End If
        Timer2.Enabled = True
        Count12 = 0
        End If
ElseIf (a = 4) And (Text4.Text = "On") Then
   Text4.Text = Text1.Text
   userCriteria = "Password  like '" & Text4.Text & "'"
   If IsNumeric(Text4.Text) Then
   With DataEnvironment1.rsCommand1
        .MoveLast 'ให้เคอร์เซอร์วิ่งไปที่เร็คคอร์ดแรกก่อน
        .Find userCriteria, , adSearchBackward
   End With
   End If
   If DataEnvironment1.rsCommand1.BOF Then 'ถ้าไม่พบรหัสที่ต้องการ
     txtstatusประตู = "Off"
   Else
   If (door = "Z") Then
   txtstatusประตู = "Door1"
   ElseIf (door = "Y") Then
   txtstatusประตู = "Door2"
   End If
  Timer2.Enabled = False
  If (door = "Z") Or (door = "Y") Then
  With DataEnvironment1.rsCommand2
  If (.EOF = False) Then
  .MoveLast
  If (txtรหัส.Text = "") Then
  .Delete
  End If
  End If
  .AddNew
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'.Fields("รหัส").Value = txtรหัส.Text
'.Fields("เวลา").Value = txtเวลา.Text
'.Fields("วันที่").Value = txtวันที่.Text
'.Fields("statusประตู").Value = txtstatusประตู.Text
.Fields("ชื่อ-นามสกุล").Value = Text2.Text
.Update
 End With
If (door = "Z") Then
Count123 = 10
ElseIf (door = "Y") Then
Count123 = 11
End If
Timer4.Enabled = True
End If
End If
End If
Errlebel:      Exit Sub
End Sub

Private Sub Timer1_Timer()
Label1.Caption = Format(Time, hh, mm, ss)
Label2.Caption = Date
End Sub

Private Sub Timer2_Timer()
If (Count12 = 3) Then
If (door = "Z") Then
Timer2.Enabled = False
MSComm1.RTSEnable = True
MSComm1.Output = "r1"
door = ""
Text4.Text = ""
Timer5.Enabled = True
ElseIf (door = "Y") Then
Timer2.Enabled = False
MSComm1.RTSEnable = True
MSComm1.Output = "r2"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
door = ""
Text4.Text = ""
Timer5.Enabled = True
End If
End If
Count12 = Count12 + 1
End Sub

Private Sub Timer3_Timer()
D_Y = Year(Date)
D_M = Month(Date)
D_D = Day(Date)
Text5.Text = Str(D_D) & Str(D_M) & Str(D_Y)
If (Text6.Text <> Text5.Text) Then
Text6.Text = Text5.Text
Text6.Text = Right(Text6.Text, 9)
Text6.Text = "c:\bar\" & Text6.Text + ".doc"
RichTextBox1.Text = "รหัส          เวลา          วันที่          Statusประตู          ชื่อ-นามสกุล"
ShowData RichTextBox1, "" & Chr(13)
RichTextBox1.Text = RichTextBox1.Text + "----------
----------"
ShowData RichTextBox1, "" & Chr(13)
With DataEnvironment1.rsCommand2
.MoveFirst
While (.EOF = False)
RichTextBox1.Text = RichTextBox1.Text + txtรหัส.Text + "      " + txtเวลา.Text + "          " + txtวันที่.Text
+ "          " + txtstatusประตู.Text + "                    " + Text2.Text
ShowData RichTextBox1, "" & Chr(13)
.MoveNext
Wend
End With
RichTextBox1.SaveFile Text6.Text, rtfRTF '+ Text6.Text + ".txt", rtfRTF
Text6.Text = Text5.Text
End If
End Sub
Private Sub Timer4_Timer()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If (Count123 = 10) Then
MSComm1.RTSEnable = True
MSComm1.Output = "A1"
Timer5.Enabled = True
Count123 = 0
Timer4.Enabled = False
ElseIf (Count123 = 11) Then
MSComm1.RTSEnable = True
MSComm1.Output = "A2"
Timer5.Enabled = True
Timer4.Enabled = False
Count123 = 0
ElseIf (Count123 = 20) Then
MSComm1.RTSEnable = True
MSComm1.Output = "B1"
Timer4.Enabled = False
Timer5.Enabled = True
Count123 = 0
ElseIf (Count123 = 21) Then
MSComm1.RTSEnable = True
MSComm1.Output = "B2"
Timer4.Enabled = False
Timer5.Enabled = True
Count123 = 0
End If
End Sub

Private Sub Timer5_Timer()
Timer5.Enabled = False
MSComm1.RTSEnable = False
End Sub
```

Editfromt.frm

```
Private Sub Command1_Click()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim tmp As Long
With DataEnvironment1.rsCommand1
.MoveLast
'tmp = .Fields("รหัส").Value + 1
.AddNew
'txtรหัส.Text = tmp
End With
Command1.Enabled = False
Command2.Enabled = False
Command3.Enabled = False
Command4.Enabled = False
Command6.Enabled = True
End Sub

Private Sub Command2_Click()
With DataEnvironment1.rsCommand1
.Fields("รหัส").Value = txtรหัส.Text
.Fields("ชื่อ-นามสกุล").Value = Text1.Text
.Fields("Password").Value = txtPassword.Text
.Update
End With
Command1.Enabled = True
Command2.Enabled = False
Command3.Enabled = True
Command4.Enabled = True
Command6.Enabled = False
End Sub

Private Sub Command3_Click()
main.Show
Unload Me
End Sub

Private Sub Command4_Click()
With DataEnvironment1.rsCommand1
If (.RecordCount <> 1) Then
.Delete
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
.MoveNext
End If
End With
End Sub

Private Sub Command6_Click()
If (txtรหัส.Text = "") Then
MsgBox "Error in fill data", vbOKOnly, "Please insert data"
ElseIf (Text1.Text = "") Then
MsgBox "Error in fill data", vbOKOnly, "Please insert data"
ElseIf (txtPassword.Text = "") Then
MsgBox "Error in fill data", vbOKOnly, "Please insert data"
Else
Command2.Enabled = True
End If
End Sub

Private Sub Form_Load()
Command2.Enabled = False
Command6.Enabled = False
End Sub
```

Viewdata.frm

```
Private Sub Command1_Click()
CommonDialog1.ShowOpen
RichTextBox1.LoadFile CommonDialog1.FileName, rtfRTF
Command3.Enabled = True
End Sub

Private Sub Command2_Click()
RichTextBox1.Text = ""
main.Show
Unload Me
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Sub PrintFooter()
   Printer.Print
   Printer.NewPage
End Sub
End Sub
Private Sub Command3_Click()
FileNumber = FreeFile
FileName = CommonDialog1.FileName
Open FileName For Input As #FileNumber
Do
Line Input #FileNumber, inputline
spaceleft = Printer.ScaleHeight - Printer.ScaleTop - Printer.CurrentY
Printer.Print inputline
If spaceleft < (3 * Printer.TextHeight(inputline)) Then PrintFooter
Loop Until EOF(FileNumber)
Close FileNumber
Printer.EndDoc
'CommonDialog1.Flags = cdlPDHidePrintToFile Or cdlPDNoSelection Or cdlnopagenums Or cdlPDCollate
'CommonDialog1.CancelError = True
'CommonDialog1.PrinterDefault = True
'CommonDialog1.Copies = 1
'CommonDialog1.ShowPrinter
End Sub

Private Sub Form_Load()
Command3.Enabled = False
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้