

บอร์ดแสดงผลแบบหมุน
ROTATING LED DISPLAY



นายนิรุต ตันชะโล

นายบุญยิ่ง อิ่มแย้ม

นายวรยุทธ แดงทอง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เลขหมู่.....
เลขทะเบียน.....55030
วัน,เดือน,ปี - 7 เม.ย. 2548

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

ROTATING LED DISPLAY

NARIS TANCHALO
BOONYING IMYAM
WORAYUTH DANGTHONG

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING
DEPARTMENT OF INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MUNGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2003

ภาควิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์

บอร์ดแสดงผลแบบหมุน

ROTATING LED DISPLAY

นักศึกษาผู้จัดทำ

นายนิศ ดันชะโล รหัสประจำตัว 44015519

นายบุญยิ่ง อิ่มเย็น รหัสประจำตัว 44015521

นายวรยุทธ แดงทอง รหัสประจำตัว 44015530

ปริญญา

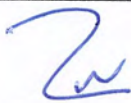
วิศวกรรมศาสตรบัณฑิต

สาขาวิชา

วิศวกรรมการวัดคุม

ปีการศึกษา

2546

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
ผศ. ไสว พงศ์สวัสดิ์	

วัน/เดือน/ปี ที่สอบ

วันอังคารที่ 23 มีนาคม พ.ศ. 2547

สถานที่สอบ

ณ ห้องสอบปริญญานิพนธ์ ภาควิชาวิศวกรรมการวัดคุม

ภาควิชารับรองแล้ว



(รศ.ประสิทธิ์ จุลเสรีวงศ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

หัวข้อปริญญานิพนธ์

บอร์ดแสดงผลแบบหมุน

ROTATING LED DISPLAY

นักศึกษาผู้จัดทำ

นายนิรศ ต้นชะโล

นายบุญยิ่ง อิ่มเข้ม

นายวรยุทธ แแดงทอง

อาจารย์ที่ปรึกษา

ผศ. ไสว พงศ์สวัสดิ์

ปีการศึกษา

2546

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ ได้เสนอวิธีการออกแบบและสร้างบอร์ดแสดงผลแบบหมุนโดยใช้ LED แบบ 3 สี เป็นตัวแสดงผลอักษรและข้อความต่างๆ ประกอบเป็นบอร์ดติดที่ปลายแขนที่สามารถหมุนได้โดยใช้มอเตอร์และมีการควบคุมการแสดงผลด้วยตัวประมวลผลไมโครคอนโทรลเลอร์ รูปแบบข้อความและอักษรที่แสดงสามารถโปรแกรมผ่านทางคอมพิวเตอร์และโหลดข้อมูลลงในตัวไมโครคอนโทรลเลอร์ซึ่งลักษณะการประยุกต์ใช้งานมีทั้ง ด้านธุรกิจการโฆษณา ด้านธุรกิจบันเทิง และในด้านการประชาสัมพันธ์ต่างๆ ได้ในหลายด้าน

ทางคณะผู้จัดทำหวังไว้ว่าจะมีผู้นำโครงงานนี้ไปพัฒนาให้ดียิ่ง ๆ ขึ้นไป

Thesis Title	Rotating LED Display
Student	Mr. Naris Tanchalo Mr. Boonying Imyam Mr. Worayuth Dangthong
Thesis Advisor	Asst.Prof. Sawai Pongswatd
Year	2003

ABSTRACT

This thesis proposed the method is design and Rotating Display Board by LED 3 colors. It can display and show several messages which consist a board on LED of the arm rotating display board .LED of the arm can rotate around by DC motor. Which control display with programmable use to the microcontroller. Data massage and word can program to move passing from computer or load data to microcontroller.This specific applies in advertisement, entertainment business and information etc.

Hope your will develop the project is an excellent.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จล่วงได้ด้วยดีเพราะได้รับความเมตตาจากผู้ช่วยศาสตราจารย์
ไสว พงศ์สวัสดิ์ ที่ได้ให้คำแนะนำแก่ผู้วิจัยตลอดมา อีกทั้งยังเอื้อเพื่ออุปกรณ์และเครื่องมือต่าง ๆ
ในการทำปริญญานิพนธ์นี้ ผู้วิจัยรู้สึกซาบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ อาจารย์ภาควิชาวิศวกรรมการวัดคุมทุกท่าน ที่ให้คำแนะนำอันเป็น
ประโยชน์ต่อการทำปริญญานิพนธ์ฉบับนี้

และที่ลืมเสียมิได้คือ ขอกราบขอบพระคุณคุณแม่ อันเป็นที่รักยิ่ง ที่สนับสนุนและเป็น
แรงบันดาลใจในการทำปริญญานิพนธ์ฉบับนี้

คุณค่าและประโยชน์อันพึงมีจากปริญญานิพนธ์ฉบับนี้ ผู้วิจัยขอบแต่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและเหตุจูงใจของการวิจัย.....	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	1
1.3 ขอบเขตของปริญญานิพนธ์.....	1
1.4 ขั้นตอนการศึกษา	2
บทที่ 2 หลักการและทฤษฎีเบื้องต้นที่เกี่ยวข้อง.....	3
2.1 ทฤษฎีเกี่ยวกับไดโอดเปล่งแสง	3
2.2 หลักการทำงานของคีมอเตอร์	6
2.3 ทฤษฎีเกี่ยวกับการใช้งาน MCS-51	13
2.4 การสื่อสารแบบอนุกรม	28
บทที่ 3 หลักการและทฤษฎีที่ใช้ในการออกแบบ	38
3.1 กล่าวนำ	38
3.2 การคำนวณและออกแบบเบื้องต้น.....	38
3.2.1 การคำนวณความเร็วรอบของมอเตอร์	38
3.2.2 การคำนวณความยาวรัศมี.....	38
3.2.3 การออกแบบวิธีการแสดงตัวอักษร.....	39
3.2.4 การออกแบบวิธีการส่งสัญญาณไฟฟ้าไปยังวงจร	39

สารบัญ(ต่อ)

	หน้า
บทที่ 4 การออกแบบและการสร้างบอร์ดแสดงผลแบบโปรแกรมได้.....	40
4.1 การออกแบบด้านฮาร์ดแวร์.....	40
4.1.1 การออกแบบวงจร	40
4.1.2 การออกแบบโครงยึดส่วนต่าง ๆ เข้าด้วยกัน	47
4.2 การออกแบบโปรแกรม	50
4.2.1 การออกแบบโปรแกรมการแสดงผลโดยไม่โครคอนโทรลเลอร์.....	51
4.2.2 การออกแบบโปรแกรมการส่งค่าจากคอมพิวเตอร์.....	62
 บทที่ 5 ผลการทดลอง.....	 64
5.1 กล่าวนำ.....	64
5.2 วิธีการใช้งาน.....	64
5.3 ผลการทดลองทางฮาร์ดแวร์.....	64
5.4 ผลการทดลองทางซอฟต์แวร์.....	65
 บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ.....	 66
6.1 บทสรุป	66
6.2 ข้อเสนอแนะ	66
 บรรณานุกรม.....	 67
 ภาคผนวก.....	 68

สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงแอดเดรสของรีจิสเตอร์เบงก์ต่างๆ.....	15
2.2 แสดงการกำหนดค่าของบิตที่อยู่ภายในรีจิสเตอร์.....	16
2.3 แสดงความสามารถในการอ้างถึงแบบบิตได้ของรีจิสเตอร์.....	19
2.4 แสดงชื่อ ตำแหน่ง และความหมายของรีจิสเตอร์ PCON.....	20
2.5 แสดงบิตต่าง ๆ ของ รีจิสเตอร์ SCON.....	24
2.6 รีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรม (serial port control register: SCON).....	25
2.7 แสดงบิตพาริตีของข้อมูล.....	30
2.8 การจัดขาของคอนเนคเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบ DB-9 และDB-25.....	34
4.1 แสดงตัวอย่างการเข้ารหัสอักษรเพื่อการแสดงผล.....	51

สารบัญรูป

รูปที่	หน้า
2.1 แสดงปรากฏการณ์ที่รอยต่อ(LED).....	3
2.2 สัญลักษณ์ไดโอดเปล่งแสง.....	4
2.3 แสดงการทำงานของไดโอดเปล่งแสง.....	4
2.4 แสดงกราฟความยาวของคลื่นแสงที่ไดโอดเปล่งแสงออกมา.....	5
2.5 แสดงกราฟการกระจายของไดโอดเปล่งแสง.....	5
2.6 แสดงลักษณะภายนอกของไดโอดเปล่งแสงแบบต่าง ๆ.....	6
2.7 แสดงถึงการเกิดแรงบิดในตัวดีซีมอเตอร์.....	7
2.8 เซอร์โวมอเตอร์แบบโรเตอร์เป็นแกนเหล็ก.....	8
2.9 เซอร์โวมอเตอร์แบบ Drag-Cup.....	8
2.10 ดีซีมอเตอร์แบบอาร์เมเจอร์ต่ออนุกรมกับขดลวดสนามแม่เหล็ก.....	8
2.11 คุณสมบัติระหว่างความเร็วและแรงบิดดีซีมอเตอร์อนุกรม.....	9
2.12 ดีซีมอเตอร์แบบแยกปรับสนามแม่เหล็กได้.....	9
2.13 แสดงคุณสมบัติระหว่างความเร็วและแรงบิดของซันท์มิเตอร์ ภายใต้ภาวะอาร์เมเจอร์โวลเตจคงที่และการกระตุ้นสนามแม่เหล็กคงที่.....	9
2.14 ซันท์มิเตอร์แบบแม่เหล็กถาวร.....	10
2.15 ดีซีมอเตอร์แบบฟิลด์.....	10
2.16 ดีซีมอเตอร์แบบแม่เหล็กถาวรและเกียร์ล้อยสำหรับลดความเร็ว.....	10
2.17 แสดงรูปหน้าตัดของดีซีมอเตอร์แบบอาร์เมเจอร์เป็นแกนเหล็กส่วนฟิลด์เป็นแม่เหล็ก.....	12
2.18 ดีซีมอเตอร์และเอนโคเดอร์แบบอาร์เมเจอร์เป็นแกนเหล็กขนาดต่างๆ.....	12
2.19 การจัดแบ่งพื้นที่ของหน่วยความจำโปรแกรม.....	14
2.20 ตารางตำแหน่งแอดเดรสของหน่วยความจำข้อมูลภายใน(Internal data memory).....	15
2.21 การจัดแบ่งพื้นที่ของหน่วยความจำข้อมูล.....	18
2.22 แสดงพอร์ตอินพุตและเอาต์พุตของ MCS-51.....	23
2.23 รูปแบบอย่างง่ายของข้อมูลอนุกรม.....	28
2.24 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส.....	29
2.25 แสดงการส่งข้อมูลแบบอนุกรมด้วยความเร็ว 9600 บิตต่อวินาที.....	31
2.26 แสดงการส่งข้อมูลขนาด 8 บิตแบบอนุกรมพร้อมด้วย บิตเริ่มต้น, บิตพาริตี, บิตหยุด ด้วย ความเร็ว 96บิตต่อวินาที.....	32

สารบัญรูป(ต่อ)

รูปที่	หน้า
2.27 แสดงระดับแรงดันสัญญาณของพอร์ตอนุกรม RS-232 กับ TTL ในสถานะลอจิก "1" และ "0"	33
2.28 คอนเนคเตอร์แบบอนุกรม 9 ขาและ 15 ขา.....	34
2.29 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่าง ๆ.....	35
3.1 ตัวอย่างการออกแบบตัวอักษร.....	38
4.1 แสดงส่วนประกอบของฮาร์ดแวร์.....	40
4.2 Block Diagram การทำงานของวงจร Microcontroller Board.....	41
4.3 แสดงวงจร Microcontroller Board	42
4.5 แสดงวงจร LED Display.....	43
4.6 แสดงลายปริ้นและการวางอุปกรณ์ของวงจร LED Display.....	44
4.7 Block Diagram การทำงานของวงจร RS232 Interface.....	44
4.8 แสดงวงจร RS232 Interface.....	45
4.9 แสดงลายปริ้นของวงจร RS232 Interface.....	45
4.10 แสดงวงจร Power Supply.....	45
4.12 แสดงวงจร Motor Driver.....	46
4.13 แสดงวงจร Infrared Sensor.....	47
4.14 แสดงภาพด้านหน้าของโครงขีคมอเตอร์.....	48
4.15 แสดงขนาดชิ้นส่วนของโครงขีคมอเตอร์.....	49
4.16 แสดงขนาดชิ้นส่วนของโครงขีคมอเตอร์.....	50
4.17 โฟล์วชาร์ตแสดงการรับค่าและควบคุมการแสดงผลของDisplay.....	60
4.18 โฟล์วชาร์ตการส่งค่ารูปแบบตัวอักษรจากคอมพิวเตอร์.....	62

บทที่ 1

บทนำ

1.1 ความเป็นมาและเหตุจูงใจของการวิจัย

ในยุคปัจจุบันถือเป็นยุคแห่งการสื่อสารและยุคแห่งเทคโนโลยี ดังนั้นเราจะพบเห็นสื่อรูปแบบใหม่ ๆ เกิดขึ้นอย่างต่อเนื่อง รวมทั้งบอร์ดโฆษณาด้วยเช่นกัน จากแรกเริ่มที่เป็นแผง LED ซึ่งได้รับความสนใจจากผู้พบเห็นจนเป็นที่นิยมกันอย่างแพร่หลาย มีการพัฒนาอย่างต่อเนื่อง เช่น เพิ่มขนาดให้ใหญ่ขึ้น เพิ่มสีสันของข้อความ แต่รูปแบบก็ยังคงเป็นรูปแบบเดิมอยู่ดี ทำให้ผู้คนคลายความสนใจลงไปเรื่อย ๆ จนในปัจจุบันได้มีผู้ประดิษฐ์ บอร์ดโฆษณารูปแบบใหม่ขึ้นมาโดยใช้ LED แค่อะเดียวนแต่สามารถแสดงข้อความหลายข้อความได้ไม่ต่างจากบอร์ดชนิดที่เป็นแผง LED ทำให้ขนาดของบอร์ดชนิดนี้มีขนาดเล็กกว่ามากทำให้เคลื่อนย้ายได้สะดวกกว่า ซึ่งเป็นข้อได้เปรียบกว่าบอร์ดชนิดแผง LED ข้อหนึ่ง ที่สำคัญเนื่องจากเป็นสิ่งประดิษฐ์ที่แปลกใหม่ ทำให้ดึงดูดความสนใจต่อผู้พบเห็นได้มากกว่าแบบเก่า

จึงเป็นที่น่าสนใจที่จะศึกษาและออกแบบให้เป็นรูปแบบของตนเอง เพื่อสร้างความแปลกใหม่และเป็นแนวทางสำหรับการพัฒนาต่อไปในอนาคต

1.2 วัตถุประสงค์ของปริญญานิพนธ์

ปริญญานิพนธ์นี้เป็นการศึกษาและพัฒนา ออกแบบ สร้างบอร์ดแสดงผลแบบโปรแกรมได้ เพื่อนำไปใช้ในการประชาสัมพันธ์ การโฆษณาทางธุรกิจ หรือเพื่อความบันเทิง เป็นการนำเสนอสื่อในอีกรูปแบบหนึ่งให้ดูน่าสนใจขึ้นโดยใช้เทคโนโลยีของไมโครคอนโทรลเลอร์ เพื่อเป็นแนวทางสำหรับพัฒนาต่อไปในอนาคต

1.3 ขอบเขตของปริญญานิพนธ์

ปริญญานิพนธ์เล่มนี้ จะกล่าวถึงการพัฒนาและออกแบบบอร์ดแสดงผลแบบโปรแกรมได้ ซึ่งการออกแบบจะมีทั้งในส่วนฮาร์ดแวร์และซอฟต์แวร์ โดยส่วนของฮาร์ดแวร์จะเป็นการออกแบบโครงสร้างของโครงงาน และในส่วนของวงจรอิเล็กทรอนิกส์ต่าง ๆ ในส่วนของซอฟต์แวร์จะเป็นการออกแบบการทำงานเพื่อให้สัมพันธ์กับฮาร์ดแวร์ที่ได้ออกแบบไว้แล้ว ให้ตรงกับวัตถุประสงค์ของโครงงาน คือการให้ความสามารถรับส่งค่าระหว่างบอร์ดแสดงผลกับคอมพิวเตอร์ได้ รวมทั้งไมโครคอนโทรลเลอร์สามารถควบคุมการแสดงผลได้ตรงตามตัวอักษรที่ได้เลือกไว้

1.4 ขั้นตอนการศึกษา

การทำโครงการวิจัยในปริญญานิพนธ์ฉบับนี้ มีขั้นตอนในการศึกษาการออกแบบบอร์ดแสดงผลแบบโปรแกรมได้ โดยเริ่มจาก ศึกษาทฤษฎีเกี่ยวกับการมองภาพของสายตามนุษย์เพื่อนำมาเป็นพื้นฐานในการออกแบบ วิธีการที่จะทำให้บอร์ดแสดงผลสามารถแสดงตัวอักษรให้อยู่ในระดับที่สายตามนุษย์มองเห็นได้ จากนั้นเป็นการศึกษาการออกแบบทางด้านฮาร์ดแวร์ให้ตรงตามข้อกำหนดต่าง ๆ เช่น น้ำหนักและขนาดของส่วนต่าง ๆ การศึกษาการออกแบบวงจรควบคุมโดยใช้ไมโครคอนโทรลเลอร์รวมทั้งวงจรอิเล็กทรอนิกส์ต่าง ๆ ที่ใช้ในโครงการ เมื่อออกแบบทางด้านฮาร์ดแวร์แล้ว จะเป็นการศึกษาทางด้านซอฟต์แวร์ ได้แก่ การทำงานของไมโครคอนโทรลเลอร์วิธีการติดต่อสื่อสารกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม เพื่อนำมาออกแบบโปรแกรมให้ทำงานได้สอดคล้องกับฮาร์ดแวร์ที่ได้ออกแบบไว้แล้ว

บทที่ 2

หลักการและทฤษฎีเบื้องต้นที่เกี่ยวข้อง

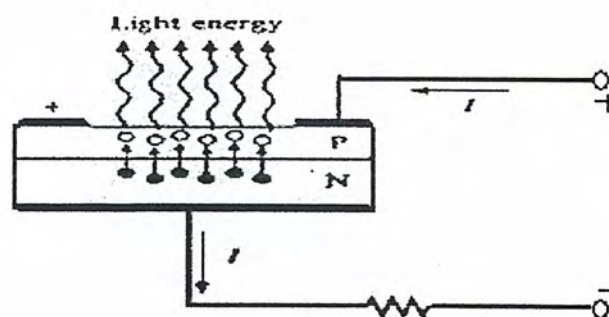
เครื่องแสดงผลแบบรอบทิศทางด้วยไดโอดเปล่งแสงจะแบ่งออกเป็น 2 ส่วนคือ ส่วนของ SOFTWARE และ HARDWARE ในส่วนของ SOFTWARE จะใช้เป็นส่วนประมวลผลควบคุมการทำงานของเครื่อง ส่วน HARDWARE จะประกอบด้วยวงจรรีเลย์ทรอนิกส์และส่วนประกอบอื่นเช่น มอเตอร์ และ โครงยึดส่วนประกอบ เป็นต้น

ทฤษฎีที่เกี่ยวข้องกับการใช้งานมีดังนี้

- 2.1 ทฤษฎีเกี่ยวกับไดโอดเปล่งแสง
- 2.2 ทฤษฎีเกี่ยวกับดีซีมอเตอร์
- 2.3 ทฤษฎีเกี่ยวกับการใช้งาน MCS-51
- 2.4 ทฤษฎีเกี่ยวกับการสื่อสารแบบอนุกรม

2.1 ทฤษฎีเกี่ยวกับไดโอดเปล่งแสง (The Light – Emitting Diode, LED)

เมื่อนำสารกึ่งตัวนำชนิดซิลิคอน หรือ เจอร์มาเนียมมาผสมกับสารบางชนิด จะทำให้เกิดปรากฏการณ์ที่เปลี่ยนแปลงไปจากรอยต่อพี-เอ็นปกติ เช่น เมื่อนำซิลิคอนผสมกับแกลเลียมอาร์เซไนด์และรอยต่อพี-เอ็นเมื่อนำรอยต่อพี-เอ็นดังกล่าวให้ไบอัสตรงจะทำให้กระแสไหลผ่านรอยต่อพี-เอ็นและอิเล็กตรอนอิสระที่รอยต่อจะหลุดออกจากวงโคจรเปลี่ยนเป็นพลังงานแสง(Light Energy) ดังรูป 2.1 ปรากฏการณ์อย่างนี้เกิดขึ้นกับไดโอดเปล่งแสงเท่านั้น(LED) เท่านั้นเรียกชื่อว่า Electroluminescence



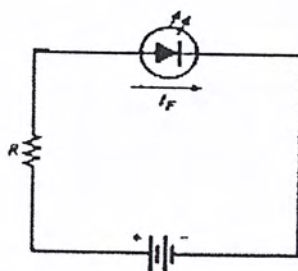
รูปที่ 2.1 แสดงปรากฏการณ์ที่รอยต่อ(LED)

แต่เมื่อให้ไบอัสกับไดโอดเปล่งแสง (LED) ไดโอดจะไม่กระแสและไม่เปล่งแสง ไดโอดเปล่งแสงจะเปล่งแสงไม่ได้เมื่อรับไบอัสตรงเท่านั้น ไดโอดเปล่งแสงมีสัญลักษณ์ดังรูป 2.2

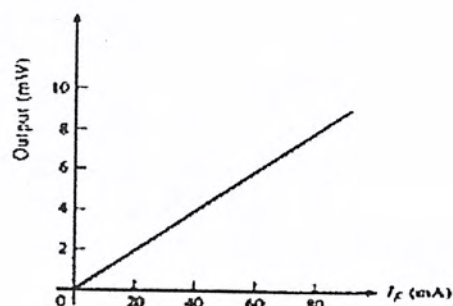


รูปที่ 2.2 สัญลักษณ์ไดโอดเปล่งแสง

แสงที่เปล่งออกมาจากไดโอดเปล่งแสงมีหลายสี เกิดจากสารที่ผสมเข้ากับรอยต่อพี-เอ็น เช่น แกลเลียมอาร์เซไนด์ (GaAs) ทำให้ไดโอดเปล่งแสงอินฟราเรด (I_R , infrared) แกลเลียมอาร์เซไนด์ฟอสไฟด์ (Gallium Arswiede Phosphide ,GaAsp) ทำให้เปล่งแสงสีแดงหรือสีเหลือง และ แกลเลียมฟอสไฟด์ (Gallium Phosphide ,GaP) ทำให้เปล่งแสงสีเขียวเป็นต้น เมื่อให้ไบอัสตรงกับ ไดโอดเปล่งแสงดังรูปที่ 2.3a จะทำให้เกิดพลังงานแสงที่ไดโอดเปล่งออกมามีหน่วยเป็นมิลลิวัตต์ ซึ่งกำลังไฟฟ้าที่เกิดขึ้นนี้จะแปรไปตามค่าของกระแสไบอัสตรงที่ผ่านไดโอดเปล่งแสง ดังกราฟ 2.3b



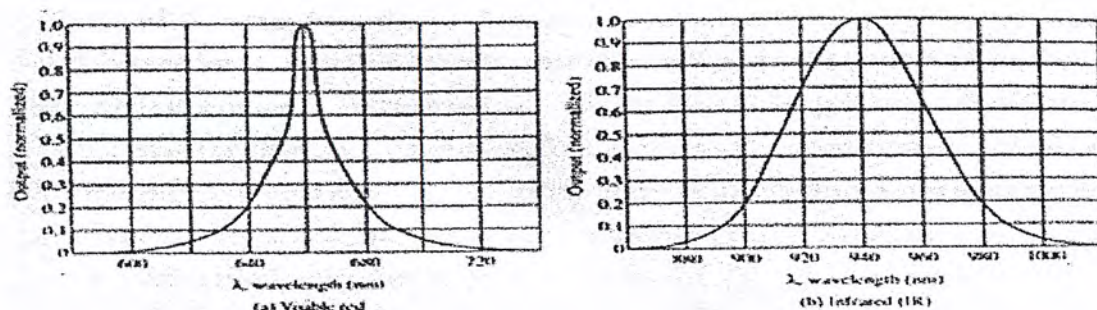
(a) Forward-bias operation



(b) Typical light output versus forward current

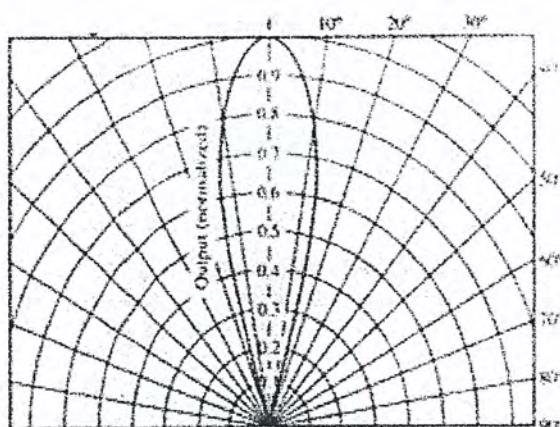
รูปที่ 2.3 แสดงการทำงานของไดโอดเปล่งแสง (LED)

ความยาวคลื่น (Wave Length) ไคโอดเปล่งแสงที่มองเห็นได้ด้วยตาเปล่า โดยมากจะเป็นสีแดง(Visible Red) ปกติจะมีความยาวคลื่นของแสงเท่ากับ 660 nm (นาโนเมตร) แต่ถ้าเป็นไคโอดเปล่งแสงอินฟราเรดซึ่งเป็นแสงที่มองเห็นด้วยตาเปล่าไม่เห็น (Unvisible) จะมีความยาวของคลื่นแสงเท่ากับ 940 nm ดังรูป 2.4



รูปที่ 2.4 แสดงกราฟความยาวของคลื่นแสงที่ไคโอดเปล่งแสงออกมา

การกระจาย (Radiation) ของแสงที่เปล่งออกมาจากไคโอดเปล่งแสง โดยทั่วไปการกระจายแสงของไคโอดเปล่งแสงจะเบี่ยงเบนจากแนวกลางของไคโอด (ของเลนส์) ไปทางซ้ายและทางขวาเท่า ๆ กันประมาณข้างละ 15 องศา ดังรูป 2.5



รูปที่ 2.5 แสดงกราฟการกระจายแสงของไคโอดเปล่งแสง



รูปที่ 2.6 แสดงลักษณะภายนอกของไดโอดเปล่งแสงแบบต่าง ๆ

2.2 หลักการทำงานของดีซีมอเตอร์

ดีซีมอเตอร์เป็นทรานซิสเตอร์แรงบิดซึ่งมีการออกแบบให้มีคุณลักษณะพิเศษคือ แรงบิดของเพลลาของดีซีมอเตอร์จะเป็นสัดส่วนโดยตรงกับกระแสอาร์เมเจอร์ แรงบิดของเพลลาของดีซีมอเตอร์จะได้จากผลระหว่างสนามแม่เหล็กและขดลวดตัวนำ หลักการนี้แสดงได้ดังรูปที่ 2.7 ในที่นี้ กระแสที่ไหลในขดลวดตัวนำจะสร้างฟิลด์ที่ประกอบด้วย เส้นแรงแม่เหล็กและขดลวดตัวนำเหล่านั้นอยู่ห่างจากจุดศูนย์กลางการหมุน r ความสัมพันธ์ระหว่างแรงบิดของเพลลาและกระแสเท่ากับ

$$T = K\phi I \quad (2-1)$$

เมื่อ T คือแรงบิดของเพลามีหน่วยเป็นนิวตัน – เมตร

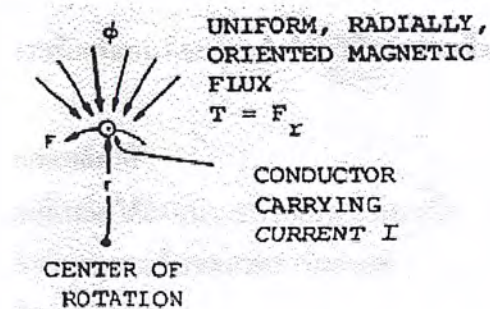
ϕ คือเส้นแรงแม่เหล็กมีหน่วยเป็นเวเบอร์

I คือกระแสแอมแปร์

K คือตัวคงที่

ดังนั้นแรงบิดของเพลลาจะเป็นสัดส่วนโดยตรงกับผลคูณของเส้นแรงแม่เหล็ก และกระแส เมื่อขดลวดตัวนำเคลื่อนที่ในสนามแม่เหล็กก็จะทำให้เกิด โวลต์เตจตกคร่อมตัวมันเอง โวลต์เตจนี้จะเป็นสัดส่วนกับความเร็วของเพลลาของมอเตอร์และด้านการไหลของกระแส ความสัมพันธ์ระหว่างโวลต์เตจย้อนกลับนี้และความเร็วของเพลลามอเตอร์คือ

$$E = K\phi\omega \quad (2-2)$$



รูปที่ 2.7 แสดงถึงการเกิดแรงบิดในตัวดีซีมอเตอร์

เมื่อ E คือโวลต์เตจย้อนกลับ emf มีหน่วยเป็นโวลต์

ϕ คือเส้นแรงแม่เหล็กมีหน่วยเป็นเวเบอร์

ω คือความเร็วของมอเตอร์มีหน่วยเป็นเรเดียน / วินาที

สมการ (2 - 1) และ (2 - 2) เป็นสมการที่แสดงหลักการทำงานพื้นฐานของดีซีมอเตอร์

การแยกประเภทของดีซีมอเตอร์

ดีซีมอเตอร์สามารถแบ่งออกได้เป็นหลายประเภท ขึ้นอยู่กับลักษณะตามแบบวิธีการสร้างสนามแม่เหล็กของตัวมอเตอร์และขึ้นอยู่กับพื้นฐานการออกแบบโครงสร้างของอาร์เมเจอร์การแบ่งประเภทลักษณะการจ่ายสนามแม่เหล็กได้เป็น 2 แบบ คือ

1. ดีซีมอเตอร์แบบปรับเส้นแรงแม่เหล็กได้
2. ดีซีมอเตอร์แบบเส้นแรงแม่เหล็กมีค่าคงที่

ถ้าพิจารณาแยกประเภทตามการออกแบบโครงสร้างอาร์เมเจอร์สามารถแยกออกได้เป็น 3 แบบคือ

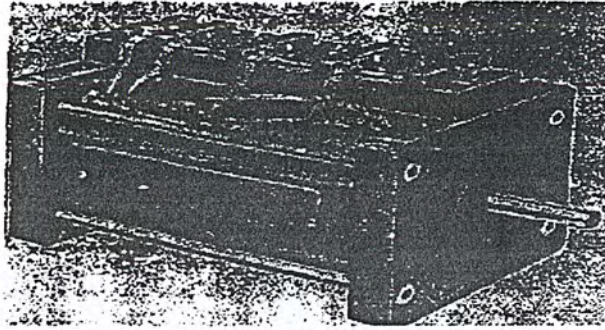
- ดีซีมอเตอร์แบบอาร์เมเจอร์เป็นแกนเหล็ก
- ดีซีมอเตอร์แบบอาร์เมเจอร์ที่มีขดลวดพันอยู่บนพื้นผิว
- ดีซีมอเตอร์แบบอาร์เมเจอร์เป็นขดลวดหมุน

นอกจากนี้ยังมีดีซีมอเตอร์ชนิดพิเศษอีกแบบหนึ่งแบบหนึ่งคือ แบบไม่แปรงถ่าน (Brushless dc - motor) ซึ่งมีหลักการทางเทคโนโลยีเหมือนกับดีซีมอเตอร์ชนิดแปรงถ่านยกเว้น การเชื่อมมีวเทชั่นกระทำโดยเทคนิคทางอิเล็กทรอนิกส์แทนที่จะกระทำโดยวิธีการทางเชิงกล

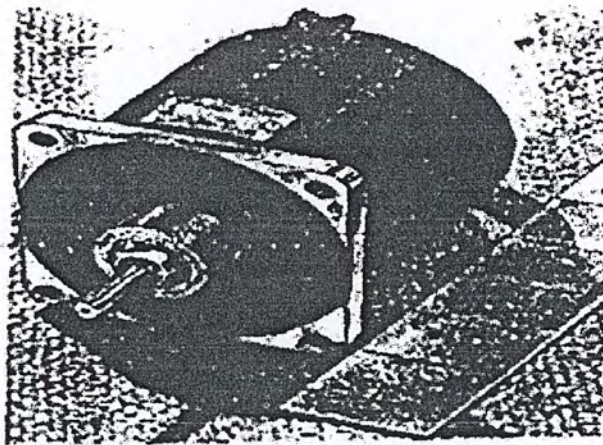
ดีซีมอเตอร์แบบปรับเส้นแรงแม่เหล็กได้

ดีซีมอเตอร์แบบปรับเส้นแรงแม่เหล็กได้ยังแบ่งแยกได้เป็น 2 แบบ คือ

- ก) แบบขดลวดสนามแม่เหล็กต่ออนุกรมกับขดลวดอาร์เมเจอร์
- ข) แบบขดลวดสนามแม่เหล็กแยกกระตุ้น

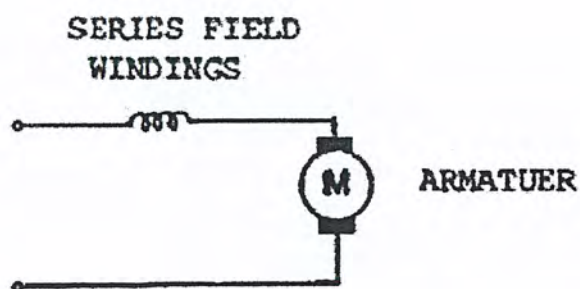


รูปที่ 2.8 เซอร์โวมอเตอร์แบบ โรเตอร์เป็นแกนเหล็ก

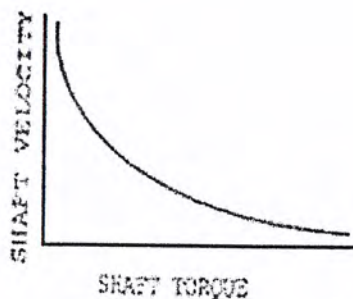


รูปที่ 2.9 เซอร์โวมอเตอร์แบบ Drag - cup

ตัวอย่างของมอเตอร์แบบขอลวดสนามแม่เหล็กต่ออนุกรมแสดงดังรูป 2.10 มอเตอร์แบบนี้จะมีแรงแม่เหล็กเป็นสัดส่วนกับกระแสตังนั้นเส้นแรงของสนามแม่เหล็กจึงสามารถปรับค่าได้ และเราได้ความสัมพันธ์ระหว่างความเร็วและแรงบิดเป็นนอนลิเนียร์ดังแสดงในรูป 2.11



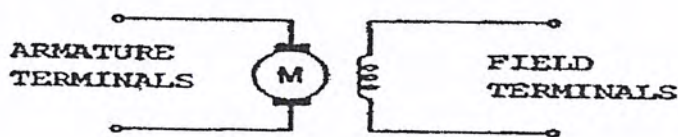
รูปที่ 2.10 คีซีมอเตอร์แบบอาร์มาเจอร์ต่ออนุกรมกับขลวดสนามแม่เหล็ก



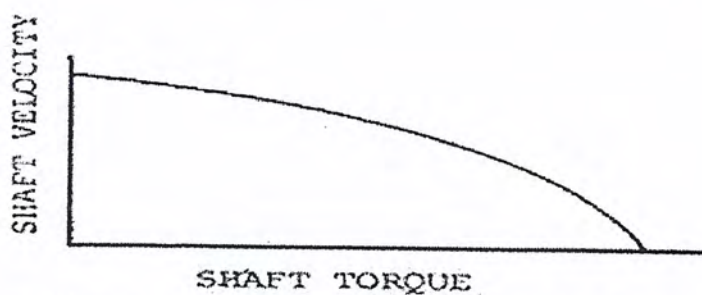
รูปที่ 2.11 คุณสมบัติระหว่างความเร็วและแรงบิดคิซีมอเตอร์อนุกรม

ภายใต้ภาวะโวลต์เตจคงที่มอเตอร์ดังกล่าวจะใช้งานในภาวะเฉพาะเมื่อต้องการแรงบิดสูงที่ความเร็วต่ำและแรงบิดต่ำที่ความเร็วสูงของการขับเคลื่อน

ตัวอย่างของมอเตอร์แบบขดลวดสนามแม่เหล็กแยกกระตุ้นแสดงในรูป 2.12 คิซีมอเตอร์แบบนี้มักนิยมเรียกกันว่ามอเตอร์ขนาน (Shunt motor) มอเตอร์แบบนี้สามารถปรับเส้นแรงแม่เหล็กได้อย่างอิสระต่อกระแสของอาร์มาเจอร์ยังผลให้สามารถควบคุมพารามิเตอร์ของมอเตอร์ ให้มีค่าคงที่ตลอดช่วงพิสัยที่กว้าง มอเตอร์นี้มักจะใช้งานในกรณีระบบการเคลื่อนที่ต้องการแรงบิดสูง ในรูป 2.13 แสดงถึงคุณสมบัติระหว่างแรงบิดกับความเร็วของขั้วตัมอเตอร์ภายใต้สภาวะการกระตุ้นสนามแม่เหล็กคงที่ที่อาร์มาเจอร์โวลต์เตจคงที่



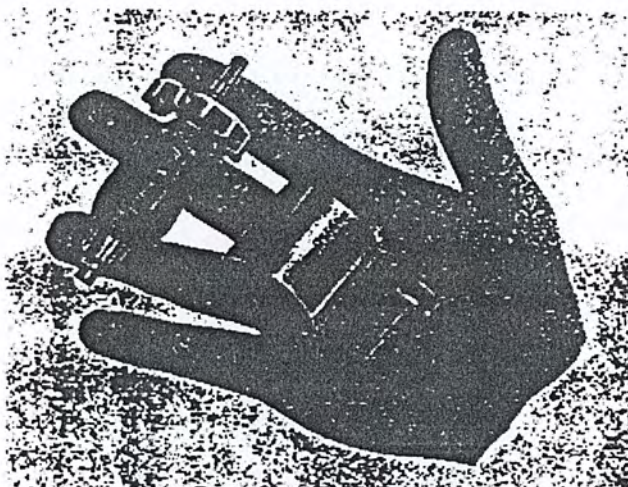
รูปที่ 2.12 คิซีมอเตอร์แบบแยกปรับสนามแม่เหล็กได้



รูปที่ 2.13 แสดงคุณสมบัติระหว่างความเร็วและแรงบิดของขั้วตัมอเตอร์ภายใต้ภาวะอาร์มาเจอร์โวลต์เตจคงที่และการกระตุ้นสนามแม่เหล็กคงที่

ดีซีมอเตอร์แบบเส้นแรงแม่เหล็กคงที่

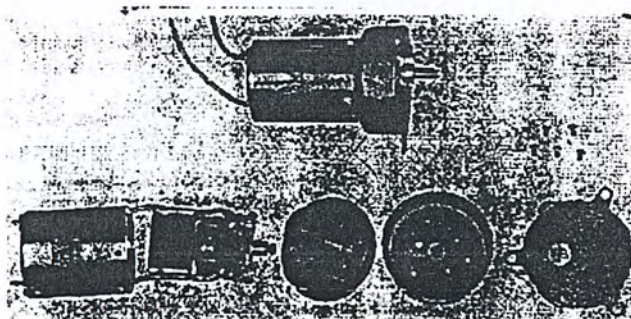
ระบบการกระตุ้นฟิวด์ของมอเตอร์โดยทั่วไปในปัจจุบันมักใช้แบบแม่เหล็กถาวรดังรูปที่-2.14 ในระบบนี้เส้นแรงของฟิวด์มีค่าคงที่ ดังนั้น ระหว่างกระแสอาร์เมเจอร์และแรงบิดจะมีค่าคงที่ ดังนั้นสมการ (2-1) เขียนให้ง่ายขึ้นได้เป็น



รูปที่ 2.14 ชั้้นท์มอเตอร์แบบแม่เหล็กถาวร



รูปที่ 2.15 ดีซีมอเตอร์แบบฟิวด์เป็นแม่เหล็กถาวร



รูปที่ 2.16 ดีซีมอเตอร์แบบแม่เหล็กถาวรและเกียร์ล็คสำหรับลดความเร็ว

$$T = K_I \quad (2-3)$$

และสมการ (2-2) ได้เป็น

$$E = K_e \omega \quad (2-4)$$

ระบบนี้จะให้ความสัมพันธ์ระหว่าง กระแสอาร์มาเจอร์ แรงบิดและความเร็วอยู่ในลักษณะลิเนียร์ สมการทางไฟฟ้าของดีซีมอเตอร์แบบนี้เขียนได้เป็น

$$V = K_e \omega + L \, di/dt + R i \quad (2-5)$$

- เมื่อ V คือ โวลต์เตจที่ป้อนให้กับมอเตอร์
 K_e คือ ค่าคงที่ของโวลต์เตจย้อนกลับ
 L คือ ค่าอินดักแตนซ์ของอาร์มาเจอร์
 R คือ ค่าความต้านทานที่ขั้วมอเตอร์

สมการไดนามิกของมอเตอร์คือ

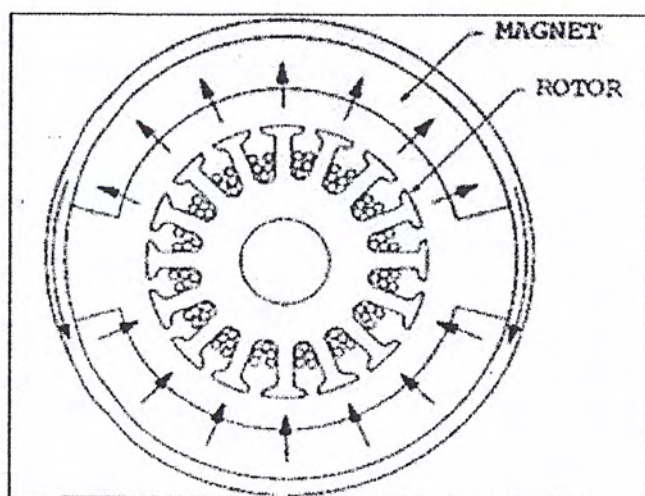
$$T_g = J \, d\omega/dt + B\omega + T_f + T_L$$

- เมื่อ T_g คือ แรงบิดที่กำเนิดโดยมอเตอร์
 J ผลรวมของโมเมนต์แรงเฉื่อยของมอเตอร์และโหลด
 B คือ สัมประสิทธิ์ของวิสคอสแดมป์ฟิง
 T_f คือแรงบิดเสียดทานภายใน
 T_L คือแรงบิดโหลด

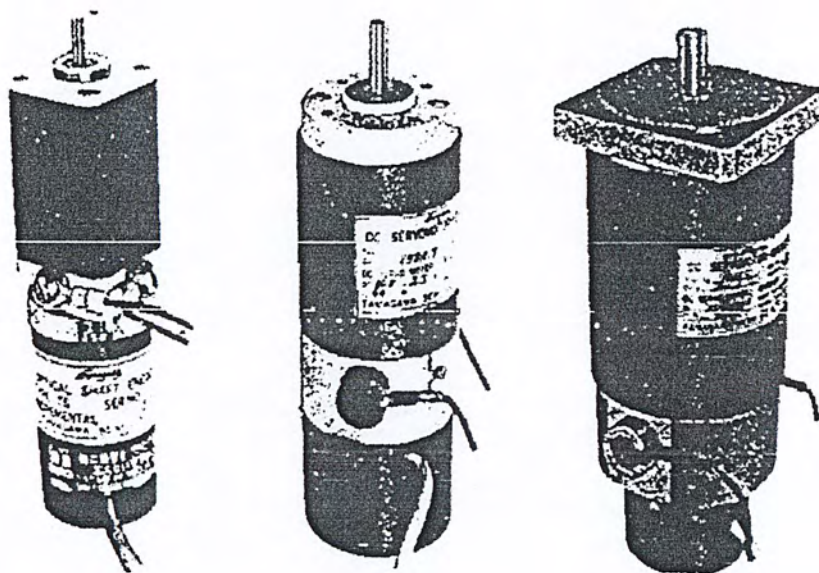
สมการต่างๆ ของดีซีมอเตอร์แบบ แยกกระตุ้นฟลด์จะเหมือนกับกระตุ้นอยู่กับที่อย่างไรก็ตามข้อดี ของมอเตอร์แบบฟลด์แม่เหล็กถาวรซึ่งเหนือกว่าแบบมีโครงสร้างของฟลด์ด้วยการพัน ขดลวดคือ ไม่มีกำลังสูญเสียในฟลด์ มีประสิทธิภาพสูงกว่าและมีขนาดเล็กกว่าเมื่อเทียบกับมอเตอร์ ที่มีขนาดของกำลังม้าเท่ากัน นอกจากนั้นความสัมพันธ์ในเชิงเส้นของสมการ (2-1) ยังให้ค่ากระแสอาร์มาเจอร์ที่สูงกว่าดีซีมอเตอร์แบบมีฟลด์ เป็นขดลวด การประยุกต์ใช้งานเหมาะสมกับระบบที่ต้องการแรงบิดของโหลดสูง

ดีซีมอเตอร์แบบอาร์มาเจอร์เป็นแกนหลัก

โครงสร้างของโรเตอร์และสเตเตอร์มอเตอร์ของมอเตอร์แบบแกนหลักแสดงในรูปที่ 2.17 โครงสร้างของมอเตอร์แบบนี้มีโมเมนต์ของแรงเฉื่อยสูงที่สุด และมีค่าอินดักแตนซ์ของโรเตอร์สูงที่สุดด้วยดังนั้นมอเตอร์จึงมีปริมาณการจุความร้อนได้สูง และสามารถจะทนโอเวอร์โหลดได้ในระยะเวลาที่ยาวนานโดยไม่ทำให้มอเตอร์เสียหาย



รูปที่ 2.17 แสดงรูปหน้าตัดของคีมอเตอร์แบบอาร์เมเจอร์เป็นแกนเหล็กส่วนฟิลด์เป็นแม่เหล็ก



รูปที่ 2.18 คีมอเตอร์และเอนโคเดอร์แบบอาร์เมเจอร์เป็นแกนเหล็กขนาดต่างๆ

2.3 ทฤษฎีเกี่ยวกับการใช้งาน MCS-51

คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ MCS-51

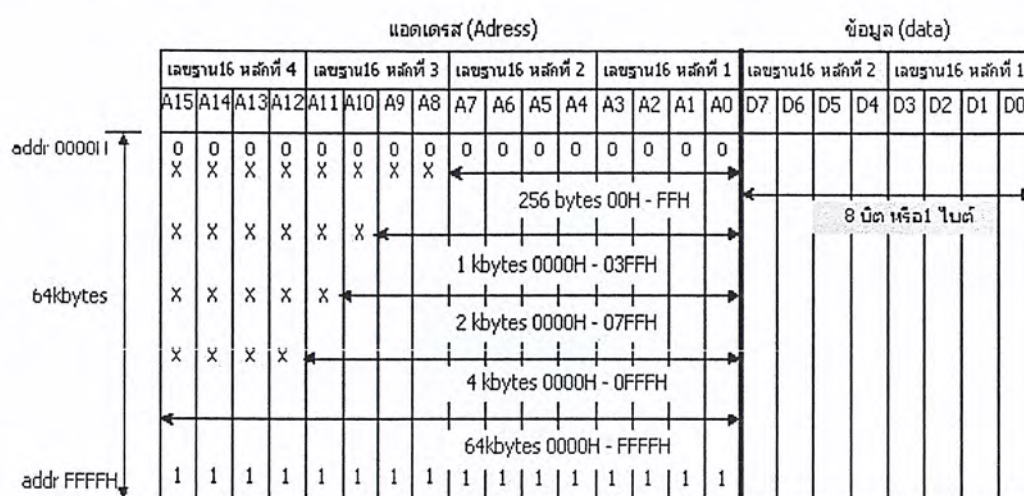
1. เป็นไมโครคอนโทรลเลอร์ ขนาด 8 บิต
2. มีวงจรออสซิลเลเตอร์ และวงจรผลิตสัญญาณนาฬิกาภายในไอซี
3. มีขาสัญญาณอินพุต-เอาต์พุต จำนวน 32 บิต
4. สามารถเชื่อมต่อหน่วยความจำข้อมูลภายนอก (External Data Memory) โดยอ้างตำแหน่ง แอดเดรสได้ถึง 64 กิโลไบต์
5. สามารถเชื่อมต่อหน่วยความจำโปรแกรมภายนอก (External Program Memory) โดยอ้างตำแหน่งแอดเดรสได้ถึง 64 กิโลไบต์
6. มีหน่วยความจำโปรแกรมภายในตัว (On Chip Program) ขนาด 4 กิโลไบต์ โดยเฉพาะ เบอร์ 8052 จะมีหน่วยความจำในส่วนนี้ถึง 8 กิโลไบต์ สำหรับเบอร์ 8031 และ M8032 จะไม่มีหน่วยความจำส่วนนี้
7. มีหน่วยความจำข้อมูลภายในตัว (On Chip Data Memory) ขนาด 128 ไบต์ โดยเฉพาะเบอร์-8032 และ 8052 มีถึง 256 ไบต์
8. หน่วยความจำข้อมูลภายในบางส่วนสามารถเข้าถึงข้อมูลระดับบิตได้ด้วย ทำให้การควบคุม หรือการตรวจสอบสถานะบิตทำได้ง่าย ส่งผลให้การเขียนโปรแกรมทำได้ง่ายมากขึ้น
9. มีไทม์เมอร์/เคาน์เตอร์ (Timer/Counters) ขนาด 16 บิต จำนวน 2 ตัว โดยเฉพาะ เบอร์ 8032 หรือ 8052 จะมีไทม์เมอร์/เคาน์เตอร์ จำนวน 3 ตัว
10. การอินเตอร์รัปต์ สามารถทำได้จาก 5 แหล่งกำเนิด โดยเฉพาะเบอร์ 8032 และ 8052 จะมีการทำอินเตอร์รัปต์ได้จาก 6 แหล่งกำเนิด โดยการอินเตอร์รัปต์ยังสามารถจัดระดับความสำคัญได้เป็นระดับ
11. มีพอร์ตสื่อสารอนุกรมภายในตัวเอง ซึ่งทำงานเป็นแบบสองทิศทาง (Full Duplex)
12. มีคำสั่งในการคำนวณทางคณิตศาสตร์ และทางตรรกศาสตร์ คำสั่งโดยส่วนใหญ่ใช้เวลาในการทำงานเพียง 1 ไมโครวินาที เมื่อใช้คริสตอลความถี่ 12 เมกกะเฮิร์ต
13. ต้องการแหล่งจ่ายไฟ 5 โวลต์ เพียงชุดเดียว

หน่วยความจำของไมโครคอนโทรลเลอร์ 8051

1. หน่วยความจำสำหรับเก็บโปรแกรม (Program Memory (Program Memory) หรือ (Code - Memory))
2. หน่วยความจำสำหรับเก็บข้อมูล (Data Memory)
3. รีจิสเตอร์ที่ทำหน้าที่เฉพาะ (Special Function Register)

หน่วยความจำสำหรับเก็บโปรแกรม (Program Memory) หรือหน่วยความจำรหัสคำสั่ง (Code - Memory)

หน่วยความจำสำหรับเก็บโปรแกรม(ทำหน้าที่เช่นเดียวกับรอม) หรือหน่วยความจำรหัสคำสั่ง (Code Memory) จะทำหน้าที่เก็บชุดคำสั่งเพื่อให้ไมโครคอนโทรลเลอร์ปฏิบัติตามคำสั่งนั้นๆ



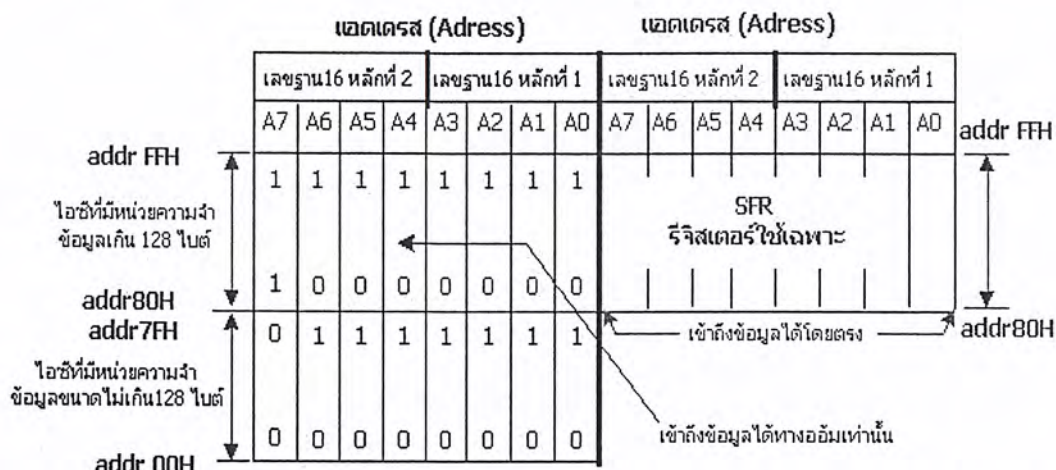
รูปที่ 2.19 การจัดแบ่งพื้นที่ของหน่วยความจำโปรแกรม

ข้อสังเกต ส่วนของแอดเดรส (ADDRESS) ไม่สามารถที่จะใช้ตำแหน่งเดียวกันได้แต่ข้อมูล (DATA) สามารถที่จะมีข้อมูลเหมือนกันได้

จากตารางหากต้องการเขียนข้อมูลใดๆลงในแต่ละบิต จะต้องมิตำแหน่งของบิตที่ไมซ้ำกัน และการที่จะเลือกจำนวนของบิต ขึ้นอยู่กับปริมาณของข้อมูลที่ต้องการจะเขียน

หน่วยความจำสำหรับเก็บข้อมูล (Data Memory)

หน่วยความจำข้อมูล (RAM) จะทำหน้าที่เก็บรักษาข้อมูล โดยข้อมูลอาจจะเก็บอยู่หลังจากไมโครคอนโทรลเลอร์ทำการประมวลผล หรือเก็บค่าข้อมูลที่จะให้กับไมโครคอนโทรลเลอร์ประมวลผลในขณะนั้น และจะทำหน้าที่เป็น สแต็ก (Stack) บางส่วน



รูปที่ 2.20 ตารางตำแหน่งแอดเดรสของหน่วยความจำข้อมูลภายใน (Internal data memory)

หน่วยความจำขนาด 128 ไบต์แรก

บริเวณแอดเดรส 00H - 1FH จำนวน 32 ไบต์จำแนกออกเป็นกลุ่ม(Blank) 8 ไบต์ จำนวน 4 กลุ่ม ซึ่งมีชื่อ เรียกว่า รีจิสเตอร์ R0 - R7 ดังตารางต่อไปนี้

ตารางที่ 2.1 แสดงแอดเดรสของรีจิสเตอร์เบงค์ต่าง ๆ

แอดเดรส	รีจิสเตอร์เบงค์	ชื่อรีจิสเตอร์ใช้งาน
00H-07H	0	R0-R7
08H-0FH	1	R0-R7
10H-17H	2	R0-R7
18H-1FH	3	R0-R7

จะเป็นได้ว่าชื่อของรีจิสเตอร์ไม่ว่าจะอยู่ในรีจิสเตอร์เบงค์ใด ก็จะมีชื่อ R0 ถึง R7 เหมือนกันทั้งสิ้น ดังนั้นในการใช้งานผู้ใช้จะต้องให้ความระมัดระวังว่าต้องการรีจิสเตอร์นั้นๆ จากเบงค์ใด กาสวิตช์เลือกแต่ละกลุ่มของรีจิสเตอร์นี้ก็ทำได้ง่าย เพียงการกำหนดค่าของบิตที่อยู่ภายในรีจิสเตอร์ PSW เท่านั้นตามตาราง ต่อไปนี้

ตารางที่ 2.2 แสดงการกำหนดค่าของบิตที่อยู่ภายในรีจิสเตอร์

รีจิสเตอร์แบงก์	บิต RS0	บิต RS1	ตำแหน่งหน่วยความจำ
0	0	0	0000H
1	0	1	0008H
2	1	0	0010H
3	1	1	0018H

อย่างไรก็ตามโดยทั่วไปก็มักจะมีการใช้งานรีจิสเตอร์ R0-R7 เฉพาะในแบงก์ 0 เท่านั้น ดังนั้นพื้นที่ของ แบงก์อื่นๆ ที่เหลือก็สามารถนำมาใช้ในลักษณะของหน่วยความจำข้อมูลภายในปกติด้วยการอ้างถึงหมายเลขของแอดเดรสอื่นๆ โดยตรง

บริเวณแอดเดรส 20H - 2FH จำนวน 16 ไบต์ บริเวณพื้นที่เป็นส่วนสำหรับผู้ใช้งานซึ่งจะมีความพิเศษต่างไปหน่วยความจำส่วนอื่นๆ เนื่องจากผู้ใช้งานสามารถอ้างถึงหน่วยความจำบริเวณนี้ได้ทั้งในลักษณะของ ไบต์ข้อมูล เช่น ปกติ หรืออาจจะเป็นบิตข้อมูลได้โดยตรง ดังนั้นหากเรามองในลักษณะบิตข้อมูลแล้ว ก็จะมีพื้นที่ตัวแปรแบบบิตให้ใช้งานได้มากถึง 128 บิต โดยตำแหน่งแรกของบิตจะเป็นบิตซึ่งเริ่มต้นนับจากบิตนัยสำคัญต่ำสุด (LSB) ของแอดเดรส 20H เรื่อยไปจนกระทั่งถึงบิตที่ 127 ซึ่งเป็นบิตนัยสำคัญสูงสุด (MSB) ของแอดเดรส 2FH 8051 ได้รับการออกแบบมาซึ่งจะเป็นเพียงการอ่านค่าสถานะลอจิก ของเส้นสัญญาณ หรือกรณีการส่งออกข้อมูลก็จะเป็นการกำหนดสถานะลอจิกให้กับวงจรภายนอกผ่านทางบิตไคบิต หนึ่งอยู่แล้ว ดังนั้นหากว่ามีการกำหนดบิตหรืออ่านค่าของบิตมาโดยตรง แทนที่จะต้องทำลอจิกขึ้นต้นกับข้อมูลทั้งไบต์ เพื่อต้องการทราบผลเพียงหนึ่งบิตเช่นที่กระทำกันในโปรเซสเซอร์โดยทั่วไป ก็จะเพิ่มความสะดวกและรวดเร็วในการ เขียนโปรแกรมควบคุมมาก

บริเวณแอดเดรส 30H - 7FH เป็นบริเวณที่สามารถนำไปใช้งานได้อย่างอิสระ โดยสามารถอ้างถึงได้เฉพาะ ในลักษณะของไบต์ข้อมูลตามปกติเท่านั้น

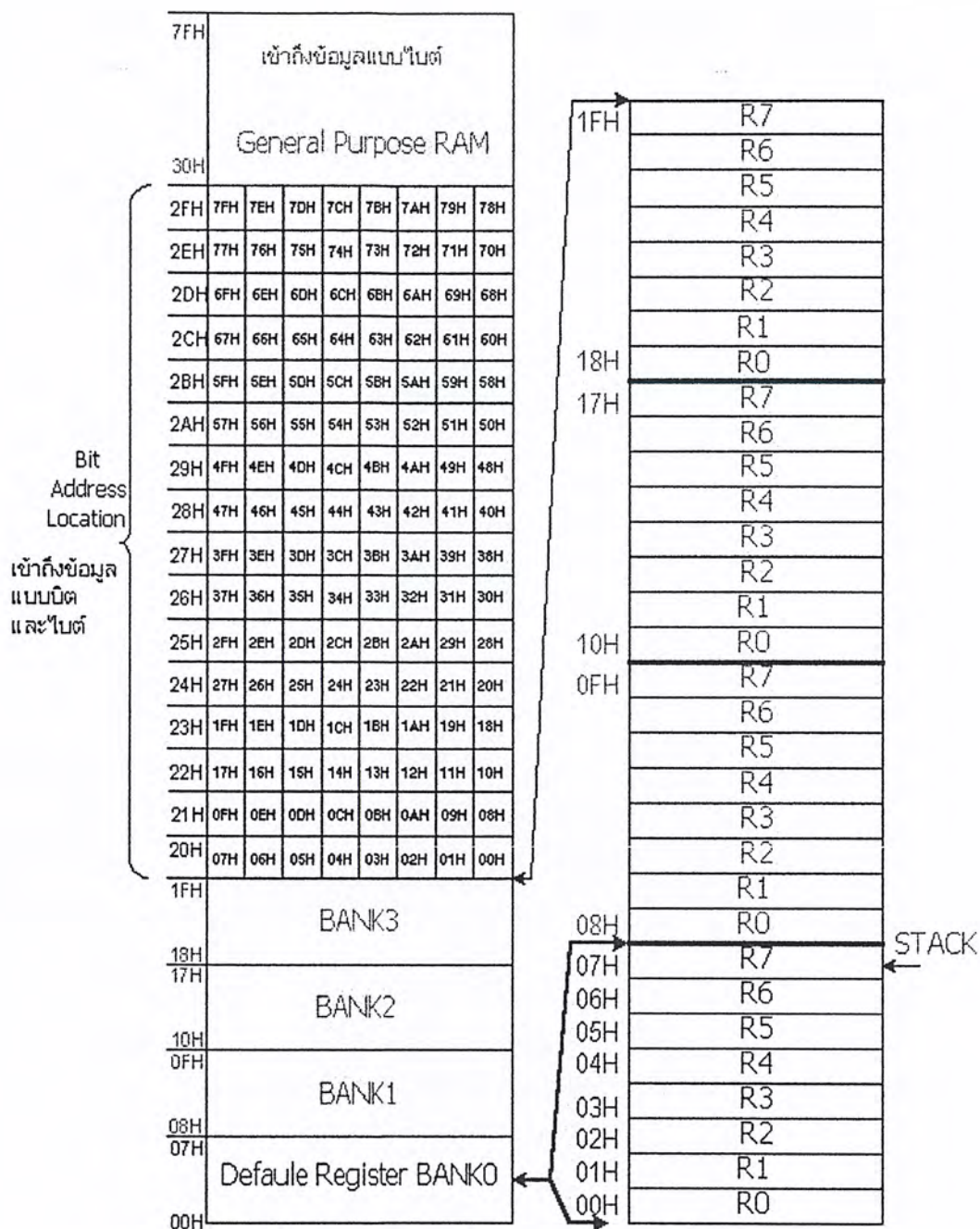
หน่วยความจำข้อมูลภายในยังแบ่งส่วนของการใช้งานได้อีกเป็นสองส่วนคือหน่วยความจำข้อมูลภายใน 128 ไบต์จะเป็นหน่วยความจำที่ใช้งานทั่วไปอยู่ที่ตำแหน่งแอดเดรส 00H-7FH และหน่วยความจำในตำแหน่งแอดเดรสที่ 80H-FFH ซึ่งจะเป็นส่วนของรีจิสเตอร์เฉพาะ (Special-Function Register) ในส่วนของหน่วยความจำที่ใช้งานทั่วไป จะแสดงได้ดังรูป โดยพื้นที่ของหน่วยความจำข้อมูลภายในที่ตำแหน่งแอดเดรส 00H-7FH ก็ยังสามารถที่จะแบ่งออกเป็นส่วนย่อยได้ดังนี้

พื้นที่ในหน่วยความจำข้อมูล (แรม) ตำแหน่งที่ 00H-1FH จำนวน 32 ไบต์ จะถูกแบ่งออกเป็น 4 กลุ่ม เรียกว่า แบงก์ (Bank) และในแต่ละแบงก์ จะมี 8 ไบต์ ดังแสดงในรูป 10 พื้นที่ในแต่ละแบงก์จะถูกใช้งานเป็นรีจิสเตอร์ที่ใช้งานทั่วไป (รีจิสเตอร์ R0-R7 เป็นรีจิสเตอร์ที่มีขนาด 8 บิต หรือ 1 ไบต์) โดยที่รีจิสเตอร์ R0 จะอยู่ในตำแหน่งแรกของแต่ละแบงก์ และ รีจิสเตอร์ R7 จะอยู่ในตำแหน่งสุดท้ายของแต่ละแบงก์ ในการนำไปใช้งานจะเลือกใช้รีจิสเตอร์ R0-R7 ได้เพียงแบงก์เดียว และเลือกใช้พื้นที่ของรีจิสเตอร์ R0-R7 ในแบงก์ใดๆก็ได้ โดยการกำหนดค่าข้อมูลที่รีจิสเตอร์ PSW ในส่วนของรีจิสเตอร์เฉพาะ (Special Function Register) หากไม่ได้กำหนดค่าใดๆเลย เมื่อทำการรีเซตให้กับ ไอซีไมโครคอนโทรลเลอร์ จะถูกกำหนดให้เริ่มต้นใช้งานที่รีจิสเตอร์ R0-R7 ในหน่วยความจำตำแหน่งแบงก์ 0 ให้เอง ดังนั้นในการทดลองเริ่มต้นในส่วนแรกๆ เราจะยังไม่กำหนดค่าใด ๆ ในการเลือกใช้งานรีจิสเตอร์แบงก์อื่นๆ(จะใช้เพียงรีจิสเตอร์R0-R7ในหน่วยความจำตำแหน่งแบงก์0ที่ถูกกำหนดมาให้เท่านั้นก่อน)

พื้นที่ในหน่วยความจำข้อมูลภายใน (แรม) ตำแหน่งแอดเดรสที่ 20H-2FH จำนวน 1 ไบต์ เป็นส่วนที่สามารถใช้งานในลักษณะการเข้าข้อมูลแบบไบต์หรือแบบบิตได้ และสามารถอ้างตำแหน่งแบบบิตได้โดยตรง เพียงแค่ระบุตำแหน่งหรือชื่อของบิตนั้นๆได้ ซึ่งจะมีด้วยกันอยู่จำนวนทั้งหมด 128 บิต แต่ละบิตจะมีหมายเลขตำแหน่งของบิตคือ 00H-7FH โดยตำแหน่งบิตที่ 00H ก็คือข้อมูลของบิตต่ำสุดในตำแหน่งแอดเดรสที่ 20H หรือเราอาจเรียกว่า (20H.1) และตำแหน่งของบิตที่ 7FH คือข้อมูลบิตสูงสุดในตำแหน่งแอดเดรสที่ 2FH หรือเราอาจเรียกว่า (20H.7) การอ้างตำแหน่งแบบบิตจะทำให้โปรแกรมทำงานได้รวดเร็วขึ้น

พื้นที่บริเวณหน่วยความจำข้อมูลในตำแหน่งที่ 30H-7FH จะเป็นพื้นที่ของหน่วยความจำใช้งานทั่วไป และการติดต่อกับข้อมูลในตำแหน่งต่างๆ ของหน่วยความจำส่วนนี้จะอ้างตำแหน่งข้อมูลได้ในลักษณะของแบบไบต์เท่านั้น และพื้นที่ส่วนนี้เราอาจจะใช้เป็นสแตคได้ (รายละเอียดของสแตคอยู่ในเรื่องหน่วยความจำแบบ SFR)

หน่วยความจำข้อมูลภายในของ 8051 มีจำนวนทั้งหมด 256 ไบต์ โดยจำแนกออกได้เป็นสองลักษณะ คือพื้นที่เฉพาะสำหรับตัวประมวลผลกลาง หรือเรียกว่า รีจิสเตอร์ R0-R7 และพื้นที่ใช้งานทั่วไปสำหรับโปรแกรมใช้งานที่ผู้ใช้สร้างขึ้นมา



รูปที่ 2.21 การจัดแบ่งพื้นที่ของหน่วยความจำข้อมูล

รีจิสเตอร์ที่ทำหน้าที่เฉพาะ (Special Function Register)

หน่วยความจำขนาด 128 ไบต์ถัดไป

พื้นที่ตั้งแต่บริเวณตั้งแต่แอดเดรส 80H-FFH เป็นบริเวณของหน่วยความจำที่มีการใช้งานเฉพาะจาก 8051 เท่านั้น โดยจะนำมาใช้เป็นตำแหน่งของ รีจิสเตอร์หน้าที่พิเศษ (Special-Function-Register หรือ SFR) จำนวน 20 ตำแหน่ง ดังแสดงในแผนภาพสำหรับไมโครคอนโทรลเลอร์เบอร์ 8051 จะมีหน่วยความจำข้อมูลภายในสำหรับการใช้งานเพิ่มมากขึ้น กว่าเบอร์อื่นๆ ซึ่งแม้ว่า จะเป็น

พื้นที่ที่มีหมายเลขแอดเดรสเดียวกับส่วนของรีจิสเตอร์หน้าที่พิเศษ แต่ในความเป็นจริงแล้วจะเป็นพื้นที่หน่วย ความจำอีกบริเวณหนึ่ง ซึ่งมีการซ้อนเกย (Overlap) กันให้อยู่ในบริเวณแอดเดรสส่วนนี้ ซึ่งหากว่าผู้ใช้งานต้องการจะเก็บข้อมูลในพื้นที่บริเวณนี้ก็จะต้องใช้การอ้างถึงหน่วยความจำ แบบ โดยอ้อม (Indirect Addressing) เท่านั้น

ตารางที่ 2.3 แสดงความสามารถในการอ้างถึงแบบบิตได้ของรีจิสเตอร์

ชื่อรีจิสเตอร์	ความสามารถในการอ้างถึงแบบบิตได้
ACC	*
B	*
PSW	*
SP	*
DPTR	*
P0	*
P1	*
P2	*
P3	*
IP	*
IE	
TMOD	
TCON	
TH0	
TL0	
TH1	
TL1	
SCON	
SBUF	*
PCON	

รีจิสเตอร์ที่เกี่ยวข้องกับพอร์ต (Port Register)

รีจิสเตอร์เหล่านี้จะมีความเกี่ยวข้องกับการทำงานของพอร์ตอินพุต/เอาต์พุตโดยตรง ซึ่งแต่ละตัวจะเป็นรีจิสเตอร์ขนาด 8 บิต สามารถใช้งานได้ทั้งในลักษณะของการอินพุต หรือการเอาต์พุตข้อมูลได้ การดำเนินการใดๆ ที่เกี่ยวกับพอร์ตทั้งสี่นี้จะมีผลทำให้ข้อมูลที่ตำแหน่งของพอร์ตเหล่านี้เปลี่ยนแปลงไปเช่นกัน นอกจากนี้พอร์ต P0 และ P2 ยังสามารถนำมาใช้ในการติดต่อกับหน่วยความจำโปรแกรมหรือหน่วยความจำข้อมูลภายนอกได้ โดยพอร์ต P2 จะเป็นค่าของแอดเดรส 8 บิตบนของหน่วยความจำ ส่วนพอร์ต P0 นั้นในช่วงเริ่มแรกจะเป็นค่าของแอดเดรส 8 บิตล่างของหน่วยความจำช่วงเวลาที่ต่อมาจึงจะนำพอร์ต P0 ไปใช้เป็นบัสสำหรับการรับหรือส่งข้อมูลกับหน่วยอุปกรณ์ ภายนอก สำหรับพอร์ต P3 นั้นนอกเหนือจากจะใช้ในสถานะของพอร์ตอินพุต/เอาต์พุตเช่นปกติแล้ว ยังนำมาใช้ในสถานะ บัสควบคุมเกี่ยวกับสัญญาณอินเทอร์รัปต์ได้อีกด้วย

รีจิสเตอร์ SBUF

เป็นบัฟเฟอร์ขนาด 8 บิต สำหรับการสื่อสารข้อมูลแบบอนุกรมทั้งการรับและส่งข้อมูล ซึ่งตามความเป็นจริงแล้วบัฟเฟอร์นี้มีอยู่ด้วยกันสองชุดและแยกจากกันอย่างชัดเจน สำหรับการส่งและการรับ โดยซีพียูจะทำการจัด การเลือกบัฟเฟอร์ที่เหมาะสมให้โดยอัตโนมัติ

รีจิสเตอร์ PCON

เป็นรีจิสเตอร์ที่ใช้ในการควบคุมหน้าที่การทำงานในสามลักษณะ ซึ่งได้แก่ การควบคุมการทำงานของโปรไซสเซอร์ (บิต IDL และ PD) การกำหนดอัตราการทำงานของอัตราเร็วในการสื่อสารข้อมูลอนุกรม (บิต SMOD) และแฟล็กสภาวะสำหรับการใช้งานทั่วไป (บิต GR0 และ GR1)

ตารางที่ 2.4 แสดงชื่อ ตำแหน่ง และความหมายของรีจิสเตอร์ PCON

ชื่อ	ตำแหน่ง	ความหมาย
SMOD	PCON.7	บิตทวิคูณของอัตราบอดเรตปกติ
-	PCON.6	
-	PCON.5	
-	PCON.4	
GF1	PCON.3	แฟล็กสำหรับให้ผู้ใช้ ใช้งานทั่วไป Flag 0
GF0	PCON.2	แฟล็กสำหรับให้ผู้ใช้ ใช้งานทั่วไป Flag 1
PD	PCON.1	บิตสำหรับการกำหนด Power down
IDL	PCON.0	บิตสำหรับการกำหนด Idle mode

บิต PD (Power down) เป็นการกำหนดให้ลดกำลังไฟฟ้าที่จ่ายให้กับส่วนของโปรเซสเซอร์ภายในลง โดยยังคงมีกำลังไฟฟ้าจ่ายให้กับส่วนหน่วยความจำข้อมูลภายในผ่านทางขาสัญญาณ RST วิธีการนี้มักนำมาใช้ในกรณีที่มีการตรวจสอบการไม่มีกำลังไฟฟ้า (Power failure) โดยวงจรตรวจสอบภายนอกจะต้องมีการอินเทอร์รัปต์เข้ามา เพื่อทำการเก็บข้อมูลที่กำลังประมวลผลอยู่ก่อน และเมื่อมีกระแสไฟฟ้าจ่ายให้เป็นปกติแล้ว จึงค่อยนำข้อมูลนั้นมาประมวลผลต่อไป

บิต IDL (Idle Mode) เป็นการกำหนดให้โปรเซสเซอร์หยุดการทำงานชั่วคราว (Sleep) และจะกลับมาอยู่ในสภาพปกติอีกครั้งเมื่อ ทำการรีเซตทางฮาร์ดแวร์ หรือมีการอินเทอร์รัปต์อย่างใดอย่างหนึ่งเกิดขึ้นการทำงานในลักษณะนี้สามารถเกิดขึ้นได้ ก็เนื่องจากว่าสภาวะการหยุดการทำงานชั่วคราวนั้น เป็นเพียงการห้ามไม่ให้มีสัญญาณนาฬิกาจ่ายให้ส่วนของโปรเซสเซอร์เท่านั้น ส่วนของวงจรการอินเทอร์รัปต์ฟอร์คอนุกรมและวงจรรับ/จับเวลา ยังคงมีสัญญาณนาฬิกาอยู่เป็นปกติ

รีจิสเตอร์ IP, IE, TMOD, TMOD, SCON

เป็นกลุ่มของรีจิสเตอร์ที่ทำหน้าที่กำหนดการควบคุม และการทำงานของอินเทอร์รัปต์ต่างๆ ของ 8051

ระบบ I/O ของ 8051

จากองค์ประกอบของชิป 8051 เราจะพบว่า มีพอร์ต I/O ที่รับส่งข้อมูลได้ 2 ทิศทางขนาด 8 บิตอยู่ 4 พอร์ต ซึ่งพอร์ตแต่ละตัวนี้จะมีคุณสมบัติพิเศษเฉพาะตัว ซึ่งแสดงแผนภาพทางตรรกะสำหรับบิตต่างๆ ในพอร์ตแต่ละตัว เราจะเห็นได้ว่าพอร์ตเหล่านี้ (P0-P3) มีคุณสมบัติทาง logic และทางไฟฟ้าที่แตกต่างกันเล็กน้อย พอร์ตแต่ละตัวจะมีแลตซ์ข้อมูล ซึ่งจะทำหน้าที่เก็บข้อมูลที่เข้าหรือออกจากพอร์ต โดยแลตซ์ข้อมูลนี้สามารถนำข้อมูลจากขาของพอร์ตหรือจากบัสข้อมูลของไมโครโปรเซสเซอร์เข้ามาเก็บได้ และแลตซ์ข้อมูลนี้ยังสามารถทกการส่งข้อมูลไปยังบัสข้อมูลของ microprocessor หรือไปยังขาของพอร์ตได้ เราจะเห็นได้จากรูป บิตในพอร์ตเบอร์ 0 และพอร์ตเบอร์-2 มี controlled pull-up ซึ่งรูปแบบการทำงานของมันจะขึ้นอยู่กับโหมดการทำงานของพอร์ต

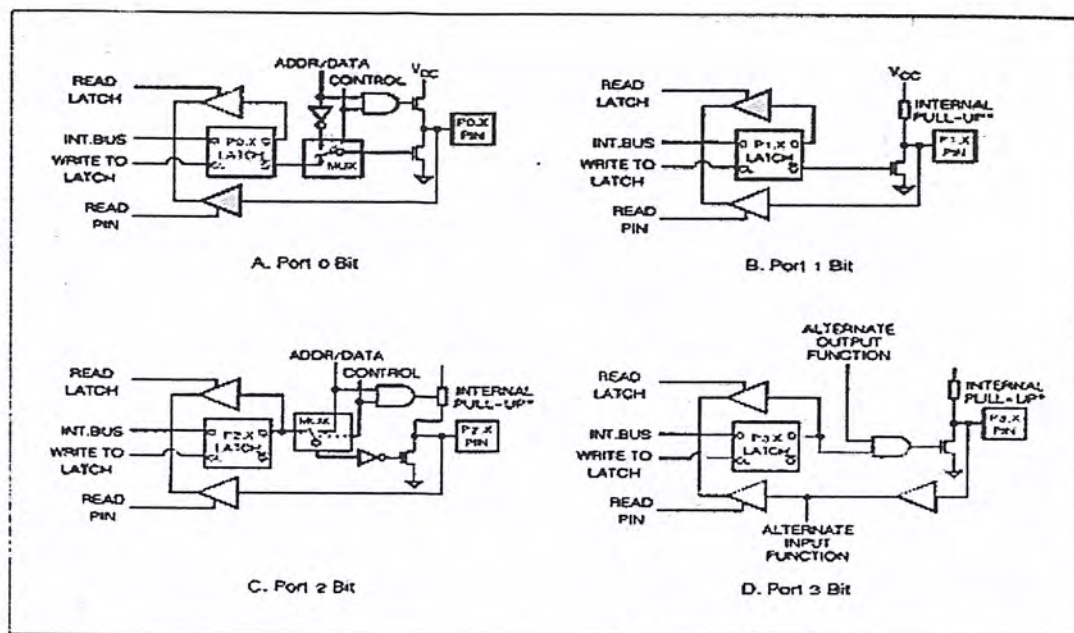
การใช้งานพอร์ตเป็นการอินพุต

การใช้งานพอร์ตเป็นการอินพุตข้อมูลจะต้องเริ่มด้วยการส่งข้อมูลที่มีค่าเป็น 1 ออกมาทางบิตของพอร์ตนั้นก่อนเป็นลำดับแรก เพื่อหยุดการทำงานของทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาต์พุตของบิตนั้น ทำให้ขาสัญญาณของบิตถูกต่อเข้ากับตัวต้านทานซึ่งทำหน้าที่ Pull-up ภายในซึ่งมีผลให้บิตนั้นๆ ของพอร์ต 1,2 และ 3 เป็น สภาวะของลอจิกสูง ตัวต้านทานนี้มีค่าประมาณ 50 K โอห์ม ซึ่งเป็นค่าที่สูงมาก และทำให้อุปกรณ์ภายนอกสามารถขับสัญญาณของพอร์ตเหล่านี้เป็นลอจิกต่ำได้ง่าย สำหรับบิตของพอร์ต 0 นั้น แม้ว่าจะมีหลักการทำงานที่คล้ายคลึง

กัน กับบิตของพอร์ตอื่นๆ แต่เนื่องจากการที่ไม่มีตัวต้านทานทำหน้าที่ Pull-up ภายในไว้ ทำให้เมื่อทรานซิสเตอร์ที่ทำหน้าที่ ขั้วสัญญาณเอาต์พุตนั้นหยุดการทำงาน ก็จะเป็นผลให้ขาสัญญาณนี้อยู่ในสถานะอิมพีแดนซ์สูงแทน

การใช้งานพอร์ตเป็นการเอาต์พุต

เมื่อมีการส่งข้อมูลที่มีค่าเป็น 0 ให้กับแต่ละบิตของพอร์ตทุกพอร์ต ข้อมูลนี้จะถูกส่งให้กับฟลิปฟล็อปซึ่งจะค้างค่านี้ไว้ และมีผลทำให้ทรานซิสเตอร์ที่ทำหน้าที่ขั้วสัญญาณเอาต์พุตนั้นทำงาน ดังนั้นขาสัญญาณก็จะมีสถานะ ลอจิกเป็นลอจิกต่ำส่วนการส่งข้อมูลที่มีค่าเป็น 1 ออกมานั้น ในกรณีที่เป็นการทำงานในแต่ละบิตของพอร์ต 1,2 หรือ 3 จะทำ ให้ทรานซิสเตอร์ที่ทำหน้าที่ขั้วสัญญาณเอาต์พุตนั้นหยุดการทำงาน มีผลทำให้ขาของสัญญาณเป็นลอจิกสูงด้วยตัว ต้านทานที่ Pull-up อยู่ภายในนั้น แต่สำหรับการทำงานในแต่ละบิตทางพอร์ต 0 นั้นจะมีผลที่แตกต่างออกไป โดยขาสัญญาณจะเป็นสถานะอิมพีแดนซ์สูงแทน เนื่องจากไม่มีตัวต้านทานภายในเชื่อมต่ออยู่นั่นเอง ดังนั้นในการใช้งานพอร์ต 0 เป็นการเอาต์พุตข้อมูล จึงจำเป็นต้องใช้ตัวต้านทานภายนอก Pull-up สัญญาณไว้กับลอจิกสูงแทน ความสามารถอีกประการหนึ่งเกี่ยวกับพอร์ตอินพุต/เอาต์พุตของ 8051 เป็นวิธีการอ่านลอจิกจากพอร์ตซึ่งมีได้สองวิธีคือ การอ่านค่าลอจิกที่ขาสัญญาณ (Port pin) และวิธีการอ่านลอจิกของการแลตช์ที่พอร์ต (Port latch) ดังจะสังเกตได้จากรูป วิธีการอ่านค่าจากพอร์ตทั้งสองแบบนี้จะช่วยให้ระบบทำงานได้ด้วยความสะดวกอย่างมาก ยิ่งขึ้น ยกตัวอย่างเช่น หากว่าพอร์ตถูกนำไปต่อกับขาเบสของทรานซิสเตอร์แบบ NPN และขาอิมิตเตอร์ต่อกับกราวด์ ของระบบ เมื่อมีการส่งค่า 1 ออกไปจะมีผลทำให้ทรานซิสเตอร์ทำงานในขณะนั้นถ้าซีพียูมีการอ่านค่าลอจิกจากขาสัญญาณของพอร์ตนี้ ก็จะได้ค่าลอจิกต่ำเนื่องจากมองเห็นค่าศักย์ไฟฟ้าระหว่างขาเบสและขาอิมิตเตอร์ซึ่งมีค่าประมาณ 0.6 โวลต์แทนดังนั้นในกรณีเช่นนี้หากว่าเป็นการอ่านค่าจากลอจิกของการแลตช์ ก็จะได้รับค่าระดับลอจิกสูงซึ่งเป็นค่า ที่ถูกต้องสภาพที่เป็นจริง



รูปที่ 2.22 แสดงพอร์ตอินพุตและเอาต์พุตของ MCS-51

พอร์ตแบบขนานของ 8051

8051 มีโครงสร้างของพอร์ตที่สามารถใช้งานแบบขนานได้จำนวนทั้งหมดสี่พอร์ตเรียกชื่อเรียงตามลำดับว่าพอร์ต 0, 1, 2 และ 3 และเป็นพอร์ตขนาด 8 บิตทั้งหมด การใช้งานพอร์ตสามารถทำได้ทั้งในลักษณะของเส้น สัญญาณเดี่ยว ๆ หรือกลุ่มของสัญญาณได้ นอกจากนี้พอร์ต 0, 2 และ 3 ยังสามารถนำไปใช้งานอื่นๆ ที่ไม่ใช่เป็นพอร์ต อินพุต/เอาต์พุตได้โดยพอร์ต 0 จะทำหน้าที่มัลติเพล็กซ์ ระหว่างบัสแอดเดรสไบต์ต่ำและบัสข้อมูลสำหรับการติดต่อ กับวงจรประกอบรวมข้อมูลบัสแอดเดรสไบต์สูงซึ่งจะส่งออกมาทางพอร์ต 2 สำหรับพอร์ต 3 นั้น นอกเหนือไปจากความสามารถเช่นพอร์ตปกติแล้วสามารถนำไปเป็นขาสัญญาณของการอินเตอร์รัปต์ต่างๆ ซึ่งรวมทั้งการสร้างสัญญาณ ควบคุม RD\ และ WR\ เพื่อทำหน้าที่อ่านหรือเขียนหน่วยความจำข้อมูลภายนอกด้วย การใช้งานพอร์ตลักษณะงานแบบ อื่นๆ ที่ไม่ใช่เป็นพอร์ต/เอาต์พุตนี้จะดำเนินการ โดย 8051 เองโดยอัตโนมัติ

พอร์ตอนุกรมของ 8051

คุณสมบัติที่สำคัญข้อหนึ่งของ 8051 คือ พอร์ตแบบอนุกรมที่มีอยู่ในตัวชิป 8051 ในไมโครโปรเซสเซอร์ทั่วไป การเพิ่มพอร์ตอนุกรมจะต้องเพิ่ม UART และวงจรควบคุมไอซีที่สนับสนุนการเชื่อมต่อแบบอนุกรม ซึ่งอุปกรณ์เหล่านี้มีความซับซ้อนและมีราคาค่อนข้างสูง การเชื่อมต่อแบบอนุกรมของ 8051 นั้นเป็นแบบ full duplex ซึ่งหมายความว่ามันสามารถทำการรับและส่งข้อมูลได้ในเวลาเดียวกัน การเชื่อมต่อแบบอนุกรมนี้อาจจะมีการพักข้อมูลที่รับเข้ามา (receive-

buffered) ซึ่งหมายความว่าก่อนที่ข้อมูลชิ้นแรกจะถูกส่งไปยังไมโครโปรเซสเซอร์ พอร์ตอนุกรมจะสามารถรองรับข้อมูลอนุกรมชิ้นที่สองได้ อย่างไรก็ตามข้อมูลชิ้นแรกจะต้องถูกส่งไปยังไมโครโปรเซสเซอร์ก่อนที่ข้อมูลชิ้นที่สองที่รับเข้ามาจะถูกนำไปเก็บในแลตช์ข้อมูล มิฉะนั้นข้อมูลชิ้นแรกที่อยู่在那นั้นจะถูกเขียนทับ การเชื่อมต่อแบบอนุกรมนี้มีการใช้รีจิสเตอร์ 2 ตัว ที่ทำหน้าที่รับและส่งข้อมูล เราติดต่อกับรีจิสเตอร์ทั้งสองตัวนี้ได้โดยการอ้างอิงรีจิสเตอร์พิเศษตัวหนึ่งที่มีชื่อว่า SBUF ถ้าเราทำการเขียนข้อมูลลง SBUF แสดงว่าเราได้เขียนข้อมูลลงรีจิสเตอร์ส่งข้อมูล แต่ถ้าเราทำการอ่านข้อมูลจาก SBUF แสดงว่าเราอ่านข้อมูลที่อยู่ในรีจิสเตอร์รับข้อมูล จะเห็นได้ว่ารีจิสเตอร์ที่ทำหน้าที่รับและส่งข้อมูลทั้งสองนี้มีค่าแอดเดรสค่าเดียวกันรีจิสเตอร์ควบคุมพอร์ตอนุกรม(serial-port control register :SCON) จะเก็บข้อมูลที่ควบคุมการทำงานของพอร์ตอนุกรม

Serial port control register: SCON

SM1	SM2	SM3	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

ตารางที่ 2.5 แสดงบิตต่าง ๆ ของ รีจิสเตอร์ SCON

SM0 SCON.7	กำหนดโหมดการทำงานของพอร์ตอนุกรม
SM1 SCON.6	กำหนดโหมดการทำงานของพอร์ตอนุกรม
SM2 SCON.5	บิตนี้จะถูกนำมาใช้ในการสื่อสารระหว่างไมโครโปรเซสเซอร์หลายตัวในโหมดการทำงานที่ 2 และ 3 สำหรับในโหมด 2 หรือโหมด 3 นี้ถ้าบิต SM2 มีค่าเป็น 1 บิต RI จะไม่เกิดการเปลี่ยนแปลง หากค่าในบิตที่ 9 ที่รับเข้ามามีค่าเป็น 0 สำหรับในโหมด 1 ถ้าบิต SM2 มีค่าเป็น 1 บิต RI จะไม่เปลี่ยนแปลง ถ้าเกิดการรับบิตสตอปผิด สำหรับในโหมด 0 บิต SM2 จะมีค่าเป็น 0
REN SCON.4	บิตนี้จะถูกเซตหรือเคลียร์โดยโปรแกรมเพื่อทำการรับ/ไม่รับข้อมูล
TB8 SCON.3	บิตที่ 9 ในข้อมูลอนุกรมที่จะถูกส่งไป
RB8 SCON.2	บิตนี้จะมีค่าเท่ากับบิตที่ 9 ในข้อมูลอนุกรมที่รับเข้ามาในโหมดที่ 2 และโหมดที่ 3 สำหรับในโหมดที่ 1 ถ้าบิต SM2 มีค่าเป็น 0 บิต RB8 จะมีค่าเท่ากับค่าในบิตสตอปของข้อมูลที่รับเข้ามา สำหรับในโหมด 0 บิต RB8 จะ

	ไม่ถูกนำมาใช้งาน
TI SCON.1	แฟล็ก transmit interrupt จะถูกเซตโดยฮาร์ดแวร์ หลังจากได้มีการเลื่อนข้อมูลขนาด 8 บิตออกในโหมด 0 หรือเมื่อได้ทำการส่งข้อมูลจนพบบิตสตอปในโหมดอื่น หลังการส่งข้อมูลแบบอนุกรมค่าในบิตนี้จะถูกเคลียร์โดยซอฟต์แวร์
RI SCON.0	แฟล็ก receive interrupt จะถูกเซตโดยฮาร์ดแวร์ หลังจากได้มีการเลื่อนข้อมูลขนาด 8 บิตเข้ามาในโหมด 0 หรือที่จุดครึ่งทางของช่องบิตสตอปในโหมดอื่น (ยกเว้นกรณี SM2 มีค่าเป็น 1) หลังการรับข้อมูลแบบอนุกรมค่าในบิตนี้จะถูกเคลียร์โดยซอฟต์แวร์

ตารางโหมดการทำงาน

ตารางที่ 2.6 รีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรม (serial port control register: SCON)

SM0	SM1	โหมดการทำงาน ของพอร์ตอนุกรม	รายละเอียด	baud rate
0	0	0	รีจิสเตอร์ที่เลื่อน ค่าได้	ความถี่ออสซิลเลเตอร์ /12 เปลี่ยนแปลงได้
0	1	1	UART ขนาด 8 บิต	ความถี่ออสซิลเลเตอร์ /64
1	0	2	UART ขนาด 9 บิต	หรือความถี่ออสซิลเลเตอร์ /32
1	1	3	UART ขนาด 9 บิต	เปลี่ยนแปลงได้

เราสามารถทำการอ้างอิงข้อมูลในรีจิสเตอร์นี้ทีละบิตได้ บิต SM0 และ SM1 จะใช้ในการกำหนดโหมดการทำงานของพอร์ตอนุกรม (ซึ่งมีทั้งหมด 4 โหมดคือ โหมด 0,1,2 และ 3) ต่อไปเราจะกล่าวถึงการทำงานของพอร์ตอนุกรมในแต่ละโหมด ในโหมด 0 พอร์ตอนุกรมจะเป็นเหมือนรีจิสเตอร์ที่เลื่อนค่าได้ (shift register) ซึ่งจะเลื่อนค่าตามสัญญาณนาฬิกาที่มีอัตราความถี่เท่ากับ 1/12 ของความถี่ออสซิลเลเตอร์ใน 8051 สัญญาณนาฬิกานี้เป็นตัวกำหนดอัตราความเร็วในการเคลื่อนย้ายข้อมูลแบบอนุกรมที่เราเรียกว่า อัตราบอด (baud rate) ในโหมดที่ 1 พอร์ตอนุกรมจะรับ

และส่งข้อมูลขนาด 10 บิต ซึ่งข้อมูลขนาด 10 บิตนี้ประกอบด้วยบิตสตาร์ท บิตข้อมูลขนาด 8 บิต และบิตสตอป 1 บิต ในโหมดการทำงานนี้ timer ตัวหนึ่งใน 8051 จะถูกนำมาใช้กำหนดอัตราเร็วในการเคลื่อนย้ายข้อมูลของพอร์ตอนุกรม ในโหมดที่ 2 พอร์ตอนุกรมจะทำการรับและส่งข้อมูลที่ละ 11 บิต ซึ่งข้อมูลขนาด 11 บิตนี้ประกอบด้วยข้อมูลขนาด 8 บิต บิตที่ 9 (เป็นบิตที่เราสามารถกำหนดค่าได้) บิตสตาร์ท และ บิตสตอป โดยบิตที่ 9 นี้จะมีค่าตรงกับบิต TB8 ของรีจิสเตอร์ควบคุมพอร์ตแบบอนุกรม (SCON) ในโหมดการทำงานนี้เราสามารถเลือก baud rate ให้มีค่าเท่ากับ $1/32$ หรือ $1/64$ ของความถี่ออสซิลเลเตอร์ใน 8051 ได้ การทำงานของพอร์ตอนุกรมในโหมดที่ 3 จะทำการรับและส่งข้อมูลที่ละ 11 บิตและมีบิตที่ 9 ที่กำหนดค่าได้ โดยบิตนี้จะมีค่าตรงกับบิต RB8 ในรีจิสเตอร์ควบคุมพอร์ตแบบอนุกรม (SCON) โหมดที่ 3 นี้จะทำหน้าที่เช่นเดียวกับโหมดที่ 2 ยกเว้นแต่ว่าอัตราเร็วในการเคลื่อนย้ายข้อมูลสามารถเปลี่ยนแปลงได้

ไมโครโปรเซสเซอร์ 8051 หลายตัวสามารถติดต่อกันได้โดยการเชื่อมต่อกันแบบอนุกรม เมื่อมีข้อมูลส่งมายังไมโครคอนโทรลเลอร์ บิตที่ 9 ในข้อมูลที่รับเข้ามาจะทำให้เกิดการอินเตอร์รัปต์ ซึ่งจะบอกให้ไมโครคอนโทรลเลอร์รู้ว่าพอร์ตอนุกรมของมันได้ทำการรับข้อมูลเข้ามาเก็บ ในที่พักข้อมูล และมันควรทำการโอนย้ายข้อมูลนี้เพื่อนำไปใช้ต่อไป ใน 8051 baud rate ในโหมด 1 และโหมด 3 จะถูกกำหนดโดยอัตราการเกิดโอเวอร์โฟลว์ของไทมเมอร์เบอร์ 1 และเราสามารถเพิ่ม baud-rate ให้เป็นสองเท่าได้โดยการกำหนดค่าของบิต SMOD ในรีจิสเตอร์ควบคุมกำลังให้มีค่าเป็น

การ Interrupt ใน 8051

ประเภทของการ Interrupt

External interrupt การตรวจสอบสัญญาณที่มา interrupt นี้ จะสามารถกำหนดให้มีการตรวจสอบในลักษณะเมื่อได้มีการเปลี่ยนแปลงระดับสัญญาณ (Level-sensitive) ไปแล้ว หรือในช่วงเวลาขณะเริ่มมี การเปลี่ยนแปลงสัญญาณจาก logic สูงไปต่ำ (Edge-sensitive)

Internal interrupt แหล่งกำเนิดสัญญาณนี้จะเป็นวงจรภายใน Microcontroller เอง เช่น วงจรนับ จับเวลา วงจรเชื่อมต่อสัญญาณอนุกรม เป็นต้น

โครงสร้างการ Interrupt เกิดได้ 5 ลักษณะ คือ

1. INT0 สัญญาณ interrupt จากภายนอก ทางขาสัญญาณ P3.2 โดย 8051 จะทำการ
สุ่มตัวอย่างสัญญาณเมื่อสิ้นสุดทุก Machine Cycle
2. INT1 สัญญาณ interrupt จากภายนอกทางขาสัญญาณ P3.3 โดย 8051 จะทำการสุ่ม
ตัวอย่าง สัญญาณเมื่อสิ้นสุดทุก Machine Cycle
3. Timer0 สัญญาณการเกิด Overflow ของ Timer0
4. Timer1 สัญญาณการเกิด Overflow ของ Timer1
5. Serial Port การเกิด interrupt ที่เกิดขึ้นจากการรับ/ส่งข้อมูลอนุกรม ทำให้มีผลต่อ flag-
interrupt RI และ TI

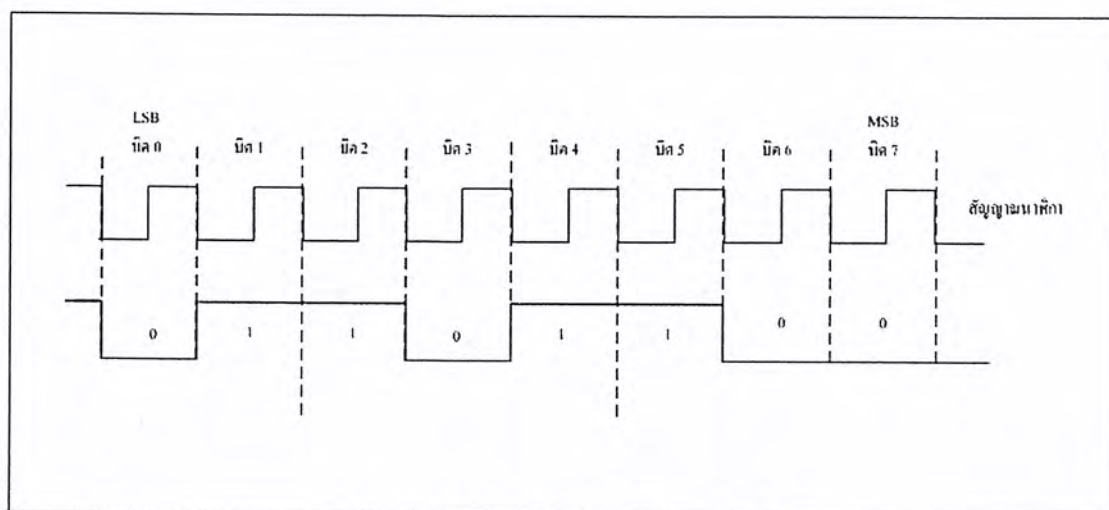
เมื่อเกิดการ interrupt สัญญาณ ต่างๆขึ้น จะส่งผลให้มีการควบคุมเพื่อสั่งให้ processor –
กระโดดไปทำงานที่ตำแหน่ง Address ต่างๆ ตามประเภทของแหล่งกำเนิดสัญญาณ interrupt ที่
เกิดขึ้น ซึ่งปรกติควรมีการสร้าง program เหล่านี้ไว้ เพื่อทำหน้าที่ขอยบริการ interrupt การ
กำหนดให้ 8051 สามารถตอบรับการ interrupt แต่ละประเภท ทำได้โดยกำหนด bit ของ ข้อมูลที่
เกี่ยวข้อง ซึ่งมักอยู่ภายใน register TCON และ SCON หากได้ว่าการกำหนด ค่าของ bit ซึ่งอยู่
ภายใน register IE (Intereupt Enable Register) ด้วยแล้ว ก็สามารถตอบรับการ interrupt ของ
สัญญาณนั้นๆ ได้ นอกจากนี้ตามแผนภาพในรูป ยังแสดงให้เห็นว่าสัญญาณ interrupt แต่ละ
ประเภท ยังสามารถกำหนด priority ของการ interrupt ได้ 2 ลักษณะ คือ High Low priority กล่าว คือ
ขณะที่ประมวลผลอยู่ภายในส่วนของ program ขอย บริการ interrupt ของสัญญาณที่มีระดับความ
สำคัญต่ำอยู่ ก็สามารถถูก interrupt ที่มี priority สูงกว่าได้ แต่หากว่าเป็นสัญญาณ interrupt ที่มี
priority เดียวกันหรือต่ำกว่าแล้วก็ต้องรอให้เสร็จสิ้นการประมวลผลที่ดำเนินอยู่ก่อน

การควบคุม Interrupt

ตามโครงสร้างที่ด้านการจัดการ Interrupt ของ 8051 สามารถกำหนดเรียกเพื่อยินยอม
หรือไม่ยินยอม (Enable/Disable) ให้มีการ interrupt แต่ละสัญญาณได้โดยใช้วิธีการกำหนด ค่าของ
bit ภายใน register IE

2.4 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมนั้นสามารถแบ่งออกได้เป็น 2 แบบคือการสื่อสารแบบซิงโครนัส และ อะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิกาเกี่ยวข้องกับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัส คือ คีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายสัญญาณของสัญญาณนาฬิกา ส่วนสายอีกเส้นจะเป็นสายสัญญาณของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสจะต้องใช้สายเชื่อมสัญญาณอย่างน้อยที่สุด 3 เส้น คือ สัญญาณนาฬิกา ข้อมูล และกราวด์



รูปที่ 2.23 รูปแบบอย่างง่ายของข้อมูลอนุกรม

การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบ อะซิงโครนัส คือการรับและส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิกาเกี่ยวข้องเหมือนกับการส่งข้อมูลแบบซิงโครนัส แต่จะการใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งเรียกว่า อัตราการถ่ายทอดข้อมูล หรือ บอดเรต (baudrate) มีหน่วยเป็น บิตต่อวินาที (bit per second : bps)

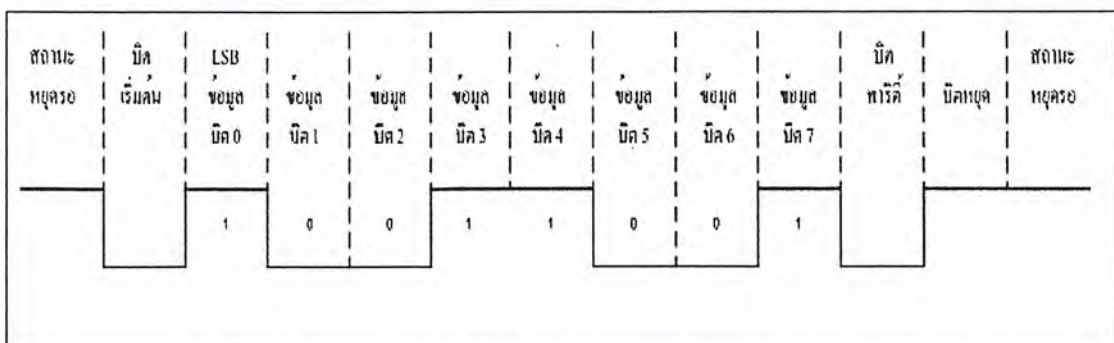
รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัส ประกอบด้วย 4 ส่วนด้วยกันคือ

1. บิตเริ่มต้น(Start Bit) ซึ่งมีขนาด 1 บิต
2. บิตข้อมูลอนุกรมจะมีขนาด 5, 6, 7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี (Parity Bit) จะมีขนาด 1 บิตหรือไม่
4. บิตปิดท้าย (Stop Bit) จะมีขนาด 1, 1.5, หรือ 2 บิต

รูปที่ 2.24 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส เมื่อไม่มีข้อมูลที่จะส่ง ขา DATA จะมีสถานะลอจิก “1” ซึ่งจะเรียกสถานะนี้ว่า สถานะหยุดรอ (waiting stage) การเริ่มต้นส่ง

ข้อมูลจะเริ่มจากการให้ขา DATA มีสถานะลอจิก “0” ด้วยช่วงระยะเวลา 1 บิต เรียกบิตนี้ว่าบิตเริ่มต้น จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อนซึ่งข้อมูลในไบต์ที่จะส่งอาจมีจำนวนบิต 5, 6, 7 หรือ 8 บิตก็ได้ จากนั้นตามด้วย บิตพาริตี ซึ่งใช้เพื่อตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่ส่งคือ บิตปิดท้าย ซึ่งจะให้ขา DATA มีสถานะลอจิก “1” อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต 1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าเป็นการสิ้นสุดข้อมูลแล้ว

อุปกรณ์พิเศษที่ออกแบบมาสำหรับการรับส่งข้อมูลแบบอะซิงโครนัสเรียกว่า Universal-Asynchronous Receiver / Transmitter หรือ UART อัตราเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัส คือ ค่าบอดเรต ซึ่งก็คือค่าจำนวนบิตต่อวินาทีที่ใช้ในการรับและส่งข้อมูล บอดเรตมาตรฐานสำหรับพอร์ตอนุกรม RS-232 ได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600, และ 19200 บิตต่อวินาทีและมีค่าเพิ่มมากขึ้นตามเทคโนโลยีคอมพิวเตอร์ซึ่งการรับส่งข้อมูลแบบอนุกรมโดยไม่ผ่านโมเด็มอาจจะกำหนดบอดเรตได้สูงถึง 115200 บิตต่อวินาที เนื่องจากบอดเรตคือจำนวนบิตของข้อมูลที่สามารถถ่ายทอดได้ภายใน 1 วินาที ยกตัวอย่างข้อมูลที่ถูกในลักษณะ 8 บิต ไม่มีการตรวจสอบพาริตี มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูลที่รับส่งนี้มีขนาด 10 บิตถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาที ก็สามารถรับส่งข้อมูลด้วยความเร็ว 960 ไบต์ต่อวินาที และถ้ามีการใช้พาริตีความเร็วในการรับส่งข้อมูลจะเหลือเป็น 872 ไบต์ต่อวินาที



รูปที่ 2.24 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส

การตรวจสอบพาริตีสามารถกำหนดให้เป็นแบบคี่หรือแบบคู่หรือไม่มีการตรวจสอบพาริตีก็ได้ การตรวจสอบพาริตีเป็นการตรวจสอบจำนวนรวมของบิตที่เป็นลอจิก “1” ภายใน ข้อมูลที่ส่งไป 1 ไบต์ว่ามีจำนวนรวมเป็นเลขคู่หรือเลขคี่โดยต้องรวมพาริตีบิตเข้าไปด้วย ยกตัวอย่างข้อมูลที่ทำการส่งมีขนาด 8 บิต และมีค่าเท่ากับ 99 ฐานสิบหก หรือ 1001001 ฐานสองจะเห็นว่าข้อมูลในไบต์มีจำนวนลอจิก “1” จำนวนสี่ตัวซึ่งเป็นเลขคู่ ดังนั้นถ้าตรวจสอบพาริตีเป็นคู่ ค่าในบิตพาริตีจะต้องมี

ลอจิกเป็น “0” แต่ถ้าพาริตีเป็นคี่ค่าที่พิตพาริตีจะต้องเป็น “1” เพื่อให้ข้อมูล 1 ไบต์รวมทั้งพิตพาริตีมีจำนวนที่เป็นลอจิก “1” รวมกันเป็นเลขคี่ ดังในตารางที่ 1.1 แสดงตัวอย่างของพิตพาริตีในการรับส่งข้อมูลอนุกรม

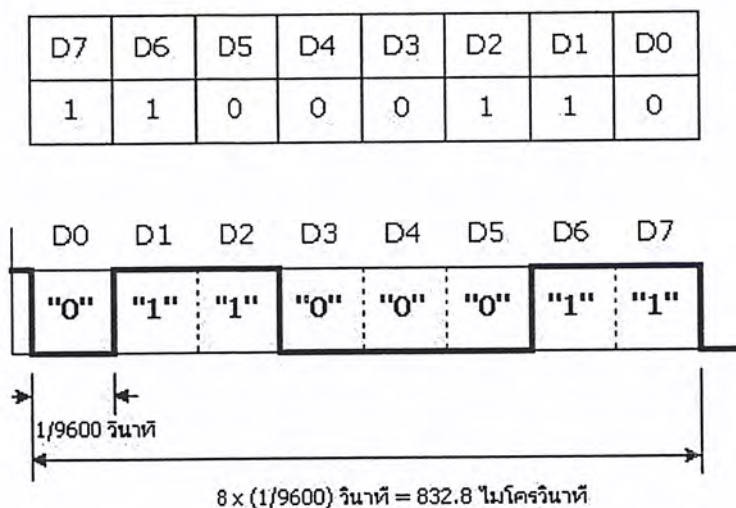
ตารางที่ 2.7 แสดงพิตพาริตีของข้อมูล

ข้อมูล	พิตพาริตีคู่	พิตพาริตีคี่
00000000	0	1
00000001	1	0
00000010	1	0
00000011	0	1
00000100	1	0
11111110	1	0
11111111	0	1

พิตพาริตีถูกสร้างจากภาคส่งข้อมูลของ UART โดยภาครับจะต้องกำหนดคุณสมบัติการตรวจสอบพาริตีให้ตรงกันว่า จะตรวจสอบพาริตีคี่หรือคู่ จากนั้นภาคส่งของ UART จะตรวจสอบพาริตีที่เกิดขึ้นว่าเป็นคี่หรือคู่โดยการนับจำนวนลอจิก “1” ทั้งหมดรวมทั้งพิตพาริตีด้วยถ้ากำหนดพาริตีไว้เป็นคู่แต่อ่านตัวเลขออกมาในการนับเป็นคี่ ทางภาครับจะแจ้งข้อมูลผิดพลาดให้ผู้ใช้งานนับเป็นการตรวจสอบความผิดพลาดของการถ่ายทอดข้อมูลที่ง่ายที่สุดแต่จะเชื่อถือได้เมื่อมีข้อมูล ที่ทำการส่งผิดพลาดมีเพียงบิตเดียวเท่านั้นถ้าข้อมูลที่ส่งผิดพลาดมีมากกว่า 1 บิตการตรวจสอบด้วยวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตีบิตเป็น NONE นั้นทั้งภาครับและภาคส่งจะไม่มีการตรวจสอบพาริตี คอมพิวเตอร์ในรุ่น AT เกือบทั้งหมดจะใช้ UART เบอร์ 16450 และ 16550 ส่วนคอมพิวเตอร์ในรุ่น XT จะใช้ UART เบอร์ 8250 UART ชิพเหล่านี้มีระดับแรงดันเป็นแบบ ทีทีแอล (0 และ +5 V) แต่เพื่อให้แรงดันเป็นไปตามมาตรฐาน RS-232 และเพื่อให้การสื่อสารข้อมูลทำได้ทั้งระยะทางไกลขึ้นระดับแรงดันทีทีแอลจะถูกแปลงเป็นระดับแรงดันที่สูงขึ้นโดยลอจิก “0 ” มีระดับแรงดัน +3 V ถึง +12 V ในขณะที่ลอจิก “1” มีระดับแรงดัน -3 V ถึง -12 V

จังหวะเวลาของการสื่อสารข้อมูลอนุกรม

ในการสื่อสารข้อมูลแบบอนุกรมเพื่อรับหรือส่งข้อมูลจะเป็นลักษณะของกลุ่มข้อมูล ดังนั้น อัตราความเร็วจะต้องมีค่าเท่ากันระหว่างการรับและการส่งโดยทั่วไปเราจะระบุความเร็วของจำนวน บิตในการรับและส่งข้อมูล เป็นจำนวนของบิตที่จะส่งใน 1 วินาที โดยเรียกความเร็วในการส่งข้อมูล ว่า อัตราบอด (Baud Rate) ซึ่งมีหน่วยเป็นบิตต่อวินาที เช่น 300, 1,200, 2,400, 4,800 และ 9,600 บิตต่อวินาที ถ้าหากมีการส่งข้อมูลด้วยความเร็ว 9600 บิตต่อวินาที จะใช้เวลาในการรับส่งข้อมูลหนึ่ง บิตมีค่าเท่ากับ $1/9600$ หรือ 104.1 ไมโครวินาที และเวลาในการรับส่งข้อมูลทั้ง 8 บิตจะมีค่าเท่ากับ 8×104.1 หรือ 832.8 ไมโครวินาที



รูปที่ 2.25 แสดงการส่งข้อมูลแบบอนุกรมด้วยความเร็ว 9600 บิตต่อวินาที

รูปแบบของการสื่อสารข้อมูลอนุกรม

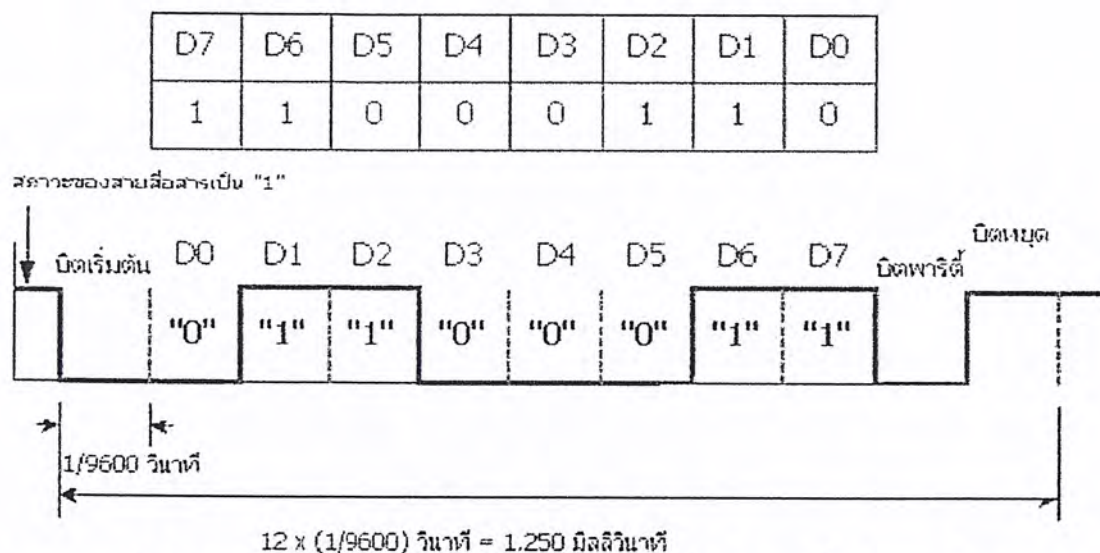
การสื่อสารข้อมูลอนุกรมแบบอะซิงโครนัส เป็นวิธีการรับและส่งข้อมูลโดยไม่ต้องอาศัย สัญญาณนาฬิกาส่งร่วมไปด้วย แต่จะใช้อัตราความเร็วของจำนวนข้อมูลต่อวินาที และจะทำการเพิ่ม บิตข้อมูลบางอย่างร่วมไปกับการส่งข้อมูลจริง เพื่อจะได้ทำการตรวจสอบข้อมูลได้อย่างถูกต้องมาก ยิ่งขึ้นซึ่งประกอบด้วยกัน 3 ส่วนคือ

1 บิตเริ่มต้น (Start bit) จะมีขนาด 1 บิต จะเป็นระดับลอจิกตรงกันข้ามกับระดับลอจิกของ สถานะสายสื่อสารขณะที่ยังไม่มีการส่งข้อมูล

2 บิตข้อมูล (Data bit) จะเริ่มจากบิตที่มีนัยสำคัญต่ำสุดก่อนหรือ บิต LSB ก่อน โดยข้อมูลที่จะส่งอาจจะมีขนาด 5, 6, 7 หรือ 8 บิตก็ได้

3 บิตแสดงสถานะเลขคู่หรือเลขคี่ (Parity bit) มีขนาด 1 บิตโดยบิตนี้จะนำไปต่อท้ายกับบิต ข้อมูล ค่าของบิตนี้ขึ้นอยู่กับจำนวนค่าของข้อมูลที่เป็น "1" โดยเลือกการส่งข้อมูลเป็นแบบ พาริตีคู่

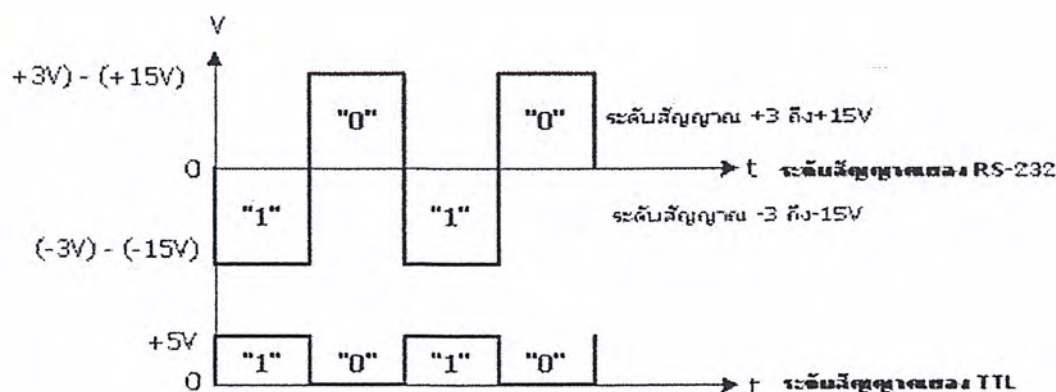
หรือ พาริตี ตัวอย่าง ถ้ากำหนดให้มีการส่งข้อมูลแบบพาริตีคู่ แต่ข้อมูลมีเลข 1 เป็นจำนวนคี่ ก็จะทำให้พาริตีนี้เป็น "1" เพื่อจะได้จำนวนเลข "1" เป็นคู่แน่นอน ทำนองเดียวกันทางด้านรับเองก็ต้องมีการตรวจสอบจำนวนข้อมูลที่รับเข้ามาเป็น "1" รวมทั้งพาริตี 1 บิต ถ้ามีค่า "1" เป็นจำนวนคู่ แสดงว่าข้อมูลที่รับเข้ามาถูกต้อง หรือสามารถกำหนดการรับและส่งข้อมูลเป็นแบบ NONE โดยไม่ต้องมีการตรวจสอบพาริตีบิตก็ได้ 4 บิตสุดท้ายหรือบิตหยุด (Stop bit) เป็นการระบุถึงขอบเขตของการสิ้นสุดข้อมูล โดยจะทำให้ขาข้อมูลมีสถานะ ลอจิกเป็น "1" ซึ่งอาจมีจำนวนมากกว่าหนึ่งบิตก็ได้ เช่น 1 บิต 1.5 บิต หรือ 2 บิต



รูป 2.26 แสดงการส่งข้อมูลขนาด 8 บิตแบบอนุกรมพร้อมด้วย บิตเริ่มต้น, บิตพาริตี, บิตหยุด ด้วยความเร็ว 9600 บิตต่อวินาที

การเชื่อมต่อพอร์ตอนุกรมมาตรฐาน RS-232

การกำหนดมาตรฐานการเชื่อมต่อแบบอนุกรม EIA RS-232(x) เป็นมาตรฐานอุตสาหกรรม โดยคณะกรรมการสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries -Association) ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบ อะซิงโครนัส 2 ทิศทาง เพื่อให้มีการใช้งาน ในการเชื่อมต่อที่สอดคล้องกัน ระหว่างอุปกรณ์คอมพิวเตอร์ต่างๆ การรับส่งสัญญาณจะกำหนดความยาวสูงสุดไว้ที่ไม่เกิน 50 ฟุตโดยมีระดับ สัญญาณตั้งแต่ 3 โวลต์ จนถึง 15 โวลต์ สำหรับลอจิก "0" และมีระดับแรงดันที่ -3 โวลต์ จนถึง -15 โวลต์ สำหรับลอจิก "1" ดังแสดงในรูปดังนั้นสังเกตได้ว่าจะมีระดับแรงดันที่ใช้ในสถานะลอจิก "0" และ ลอจิก "1" แตกต่างออกไปจากระบบไอซีดิจิทัลทั่วไป การต่อใช้งานจึงต้องมีอุปกรณ์ที่ทำหน้าที่เปลี่ยนระดับแรงดัน 0-5 โวลต์ จากไมโครคอนโทรลเลอร์ ให้เป็นระดับแรงดันที่สูงกว่า +3 หรือต่ำกว่า -3 โดยจะมีไอซีสำเร็จรูปพร้อมใช้งาน หรืออาจจะต่อวงจรจากทรานซิสเตอร์ได้



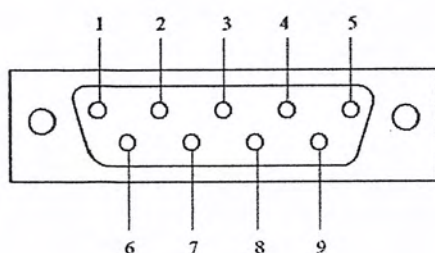
รูปที่ 2.27 แสดงระดับแรงดันสัญญาณของพอร์ตอนุกรม RS-232 กับ TTL ในสถานะลอจิก "1" และ "0"

ไอซี MAX232, L232

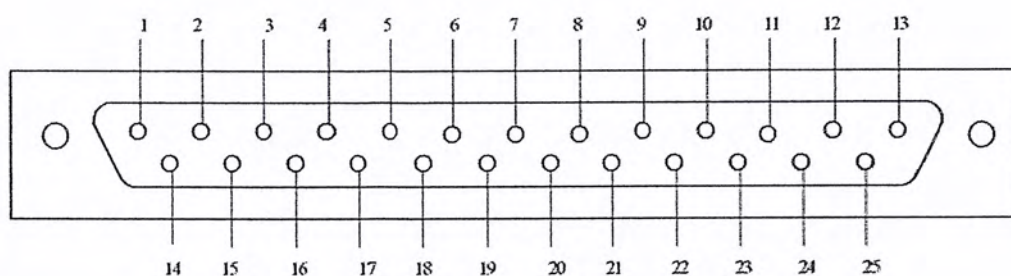
ไอซี MAX232, L232 เป็นไอซีที่แปลงระดับสัญญาณจากระดับ TTL ไปเป็นระดับของ RS-232 และในทำนองเดียวกันก็รับระดับสัญญาณจาก RS-232 เพื่อแปลงเป็นระดับสัญญาณจากระดับ TTL ให้กับไมโครคอนโทรลเลอร์ได้

มาตรฐานพอร์ตอนุกรมแบบ RS – 232

มาตรฐานการเชื่อมต่อแบบ RS – 232 จะใช้คอนเนคเตอร์แบบ DB 25 ตัวผู้หรือ DB-9 ตัวผู้ ซึ่งเป็นคอนเนคเตอร์แบบ DB-25 จะมีข้อต่อใช้งานเพียง 9 เส้น เช่นเดียวกับคอนเนคเตอร์แบบ DB - 9 เนื่องจากขาอื่น ๆ ที่เคยใช้งานในอดีตปัจจุบันมีการใช้งานไม่มากนักจึงถูกยกเลิกไป โดยแสดงในรูปและตำแหน่งของขาในรูป 2.4.5 สำหรับการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกแสดงในรูปที่ 2.28 ลูกศรในรูปแสดงถึงทิศทางของข้อมูลในรูปที่ 2.28 (ก) เป็นการเชื่อมต่อแบบ Null-modem หรือการเชื่อมต่อโดยตรงโดยไม่ต้องผ่านโมเด็มโดยมีการตรวจสอบหรือแฮนด์เช็กเต็มรูปแบบ ส่วนในรูปที่ 2.28 (ข) เป็นการต่อแบบ Null modem ในลักษณะที่ใช้สายเพียง 3 เส้น โดยเส้นหนึ่งสำหรับส่งข้อมูล อีกเส้นหนึ่งสำหรับรับ ข้อมูล และเส้นสุดท้ายเป็นกราวด์ สำหรับรายละเอียดหน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS – 232 มีดังนี้



(ก) คอนเนคเตอร์อนุกรม 9 ขา หรือแบบ DB-9 (มองจากด้านหลังคอมพิวเตอร์)

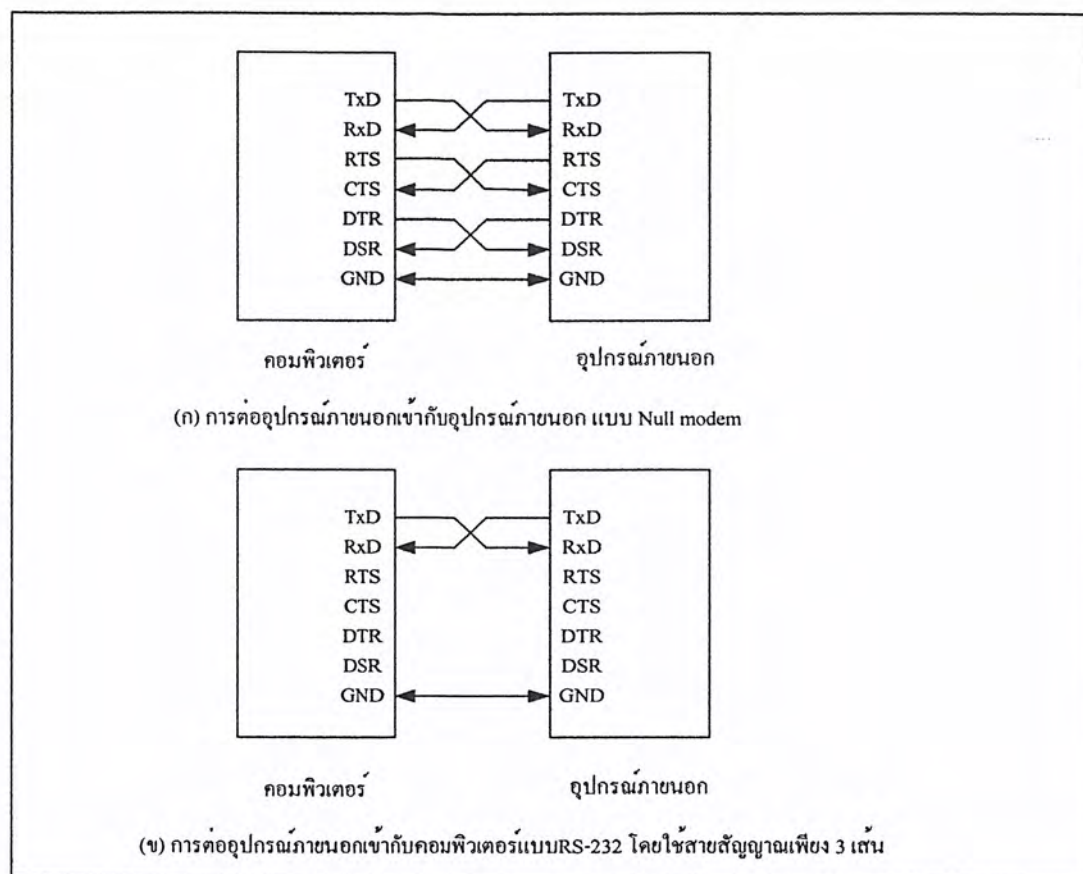


(ข) คอนเนคเตอร์อนุกรม 25 ขา หรือแบบ DB-25 (มองจากด้านหลังคอมพิวเตอร์)

รูปที่ 2.28 ลักษณะของพอร์ตอนุกรม

ตารางที่ 2.8 การจัดขาของคอนเนคเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบ DB-9 และ DB-25

คอนเนคเตอร์ DB-9	คอนเนคเตอร์ DB-25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data Carrier Detect : DCD	อินพุต
2	3	receive Data : RxD	อินพุต
3	2	Transmitted Data : TxD	เอาต์พุต
4	20	Data Terminal Ready :DTR	เอาต์พุต
5	7	Signal Ground : GND	
6	6	Data Set Ready : DSR	อินพุต
7	4	Request To Send : RTS	เอาต์พุต
8	5	Clear To Send : CTS	อินพุต
9	22	Ring Indicator : RI	อินพุต



รูปที่ 2.29 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่าง ๆ

Data Carrier Detect : DCD หรืออาจเรียกว่า Carrier Detect : CD ขานี้จะแอกทีฟเมื่อมีการส่งสัญญาณพาหะจากอุปกรณ์สื่อสารข้อมูลเช่น โมเด็ม สำหรับการใช้งานปกติ ขานี้จะไม่ได้ถูกใช้งานมากนัก

Receive Data : RD หรือ RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์ บัฟเฟอร์

Transmitted Data : TD หรือ TxD ใช้ส่งข้อมูลออกจากคอมพิวเตอร์โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูล ส่งออกไป

Data Terminal Ready : DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่า ต้องการติดต่อด้วย โดยขา DTR นี้ต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทางและขา DTR ของอุปกรณ์ปลายทางต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อเป็นแบบ Null modem ซึ่งใช้สายในการเชื่อมต่อเพียงสามเส้นจะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกันและต้องต่อเข้ากับขา DCD ด้วยในกรณีโปรแกรมสื่อสารที่ใช้มีการตรวจสอบสัญญาณพาหะ

Signal Ground : GND กราวด์ระบบ

Data Set Ready : DSR ขานี้ใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR

Request To Send : RTS เป็นขาสำหรับส่งสัญญาณร้องขอให้อุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบ Null - modem 3 สายจะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อให้การรับส่งข้อมูลเกิดขึ้นตลอดเวลา

Clear To Send : CTS ขานี้จะคอยรับสัญญาณจากขา CTS เมื่อรับสัญญาณได้ข้อมูลที่ขา Tx/D จะถูกส่งออกไป ดังนั้นขานี้จึงใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่

Ring Indicator : RI ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ปกติในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็มและโปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้น

UART

UART ย่อมาจากคำว่า Universal Asynchronous Receiver Transmitter ซึ่งหมายถึงอุปกรณ์ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัสนั่นเอง สำหรับการสื่อสารบนคอมพิวเตอร์แล้ว UART ถือว่าเป็นหัวใจหลักของการสื่อสารแบบอนุกรม

หน้าที่หลักของ UART ก็ทำหน้าที่แปลงข้อมูลที่อยู่ในรูปแบบขนานจากคอมพิวเตอร์ให้อยู่ในรูปแบบอนุกรมแบบอะซิงโครนัสแล้วส่งออกไปและทำหน้าที่แปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนานก่อนที่จะส่งเข้าสู่คอมพิวเตอร์ซึ่งนอกจาก UART จะส่งข้อมูลไปยังคอมพิวเตอร์แล้ว ยังแจ้งข้อมูลอื่น ๆ ให้คอมพิวเตอร์ทราบด้วย เช่น อัตราเร็วในการรับส่งข้อมูล (บอดเรต) รูปแบบการส่งข้อมูล ความผิดพลาดที่เกิดขึ้นระหว่างการถ่ายทอดข้อมูล (ผิดพลาดพาริตี,เฟรมข้อมูล, โอเวอร์รัน) เป็นต้น

ภายใน UART จะมีส่วนของวงจรสร้างบอดเรตแบบโปรแกรมได้(programmable-buadrate generator) โดยการกำหนดตัวหารค่าให้กับสัญญาณนาฬิกาของ UART โดยตัวหารนี้มีขนาด 16 บิตดังนั้นจึงสามารถกำหนดตัวหารอยู่ในช่วง 1- 65,535 UART

สามารถรับส่งข้อมูลได้ทั้งแบบฮาล์ฟดูเพล็กซ์ (Half duplex) และฟูลดูเพล็กซ์ (Full duplex) โดยการส่งแบบฮาล์ฟดูเพล็กซ์เป็นการส่งแบบทิศทางเดียว ส่วนการส่งแบบฟูลดูเพล็กซ์นั้นสามารถรับและส่งข้อมูลได้ในคราวเดียวกัน

ชนิดของ UART

ในเครื่องคอมพิวเตอร์โดยทั่วไปมี UART ที่ใช้งานอยู่ 2 เบอร์คือ 8250 ซึ่งเป็น UART มาตรฐานที่มีใช้กันมาช้านาน UART แบบนี้จะมีบัฟเฟอร์สำหรับรับและส่งข้อมูลตำแหน่งเดียวกัน ทำให้การรับและส่งข้อมูลถูกจำกัดความเร็วที่ 57.6 กิโลบิตต่อวินาทีเท่านั้น แต่ UART เบอร์นี้ถือว่าเป็นต้นแบบของ UART ที่ใช้ในคอมพิวเตอร์

UART อีกเบอร์หนึ่งคือ 16450 มีความสามารถรับส่งข้อมูลได้ด้วยความเร็ว 1,152,000 บิตต่อวินาทีและเพิ่มรีจิสเตอร์สำหรับพักข้อมูลสำหรับ UART นอกนั้นยังเพิ่มส่วนของชิพรีจิสเตอร์แบบ FIFO (First In First Out) ขนาด 16 ไบต์เข้าไป ทำให้สามารถสนับสนุนความเร็วในการรับส่งข้อมูลที่ 256 กิโลบิตต่อวินาทีได้โดยคอมพิวเตอร์ในปัจจุบันใช้ UART เบอร์นี้หรือใหม่กว่าเช่น เบอร์ TL16C750 ซึ่งมีรีจิสเตอร์แบบ FIFO ขนาด 64 ไบต์ทำงานได้ที่ระดับแรงดัน +5 V และ +3V มีโหมดประหยัดพลังงาน สามารถรับส่งข้อมูลได้ด้วยความเร็ว 1 เมกะบิตต่อวินาทีเมื่อใช้สัญญาณนาฬิกา 16 MHz เท่านั้น

อย่างไรก็ตามความเร็วในการส่งข้อมูลของ UART เบอร์ใหม่ ๆ ก็ไม่ได้ช่วยให้การรับส่งข้อมูลของคอมพิวเตอร์เร็วขึ้น เนื่องจากว่าคอมพิวเตอร์ยังใช้ความถี่ของสัญญาณนาฬิกาในการส่งข้อมูลเพียง 1.8432 MHz เท่านั้น

บทที่ 3

หลักการและทฤษฎีที่ใช้ในการออกแบบ

3.1 กล่าวนำ

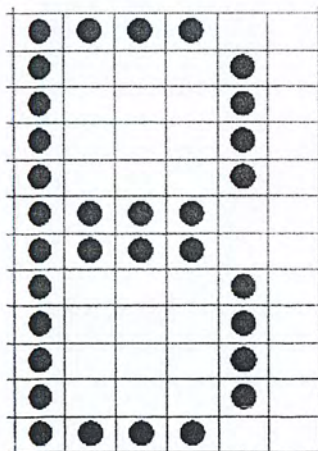
ทฤษฎีหลักที่ใช้ในการออกแบบโครงงานชิ้นนี้ คือทฤษฎีเกี่ยวกับการมองเห็นของมนุษย์ ที่ว่าสายตามนุษย์ไม่สามารถจับการเปลี่ยนแปลงของภาพได้ทันที ที่เวลา $1/25$ วินาที หรืออีกนัยหนึ่ง คือ ถ้ามีการเปลี่ยนแปลงภาพอย่างน้อย 25 ภาพ ในหนึ่งวินาที ซึ่งหลักการนี้ก็นำมาใช้กับโทรทัศน์เหมือนกัน ดังนั้นหากเราใช้ LED เพียงแค่แถวเดียวแต่ถ้าสามารถกระพริบติดดับที่ตำแหน่งเดิมตลอดจนสายตามนุษย์จับการกระพริบนั้นไม่ทัน ก็สามารถมองเห็นข้อความที่ดู เหมือนว่า LED นั้นติดอยู่ตลอดเวลาได้ และจากทฤษฎีนี้สามารถนำมาคำนวณเพื่อหาความเร็วรอบของมอเตอร์ เพื่อที่จะนำไปเลือกใช้ได้อีก รวมทั้งส่วนประกอบอื่นๆของโครงงานอีกด้วย

3.2 การคำนวณและออกแบบเบื้องต้น

3.2.1 การคำนวณความเร็วรอบของมอเตอร์

ใน $1/25$ วินาที ต้องหมุนหนึ่งรอบ
ดังนั้นในหนึ่งรอบจะใช้เวลา 0.04 วินาที หรือ $0.04/60$ นาที
ดังนั้น ใน 1 นาที จะหมุน $60/0.04$ เท่ากับ 1500 รอบ

3.2.2 การคำนวณความยาวรัศมี



รูปที่ 3.1 ตัวอย่างการออกแบบตัวอักษร

- ใน 1 ตัวอักษรใช้พื้นที่ 6 ช่อง ถ้าต้องการ แสดงอักษร 12 ตัว จะใช้พื้นที่ 72 ช่อง
- เลือกใช้ LED 2 สี ขนาด 5 มิลลิเมตร แต่เผื่อระยะการมองอีก 3 มิลลิเมตร
- จะได้เส้นรอบวงในการแสดงข้อความจะได้ 72×8 เท่ากับ 576 เซนติเมตร
- ดังนั้นระยะของรัศมีจะได้ $576/2\pi$ เท่ากับ 91.7 เซนติเมตร

3.2.3 การออกแบบวิธีการแสดงตัวอักษร

วิธีการแสดงอักษรถือเป็นหัวใจหลักของโครงงานชิ้นนี้ เป็นความสัมพันธ์ระหว่างการหมุนของมอเตอร์กับช่วงเวลาการติดดับของ LED ซึ่งก็คือการออกแบบโปรแกรมนั่นเอง โปรแกรมจำเป็นต้องรู้ตำแหน่งที่จะส่งข้อมูลชุดเดิมออกไปทุก ๆ รอบการหมุนของมอเตอร์เพื่อรักษาการกะปริบอย่างต่อเนื่องของ LED ในข้อมูลชุดเดิมและตำแหน่งเดิม ทำให้มองดูเหมือนว่า LED แต่ละตำแหน่งติดอยู่ตลอดเวลา ดังนั้นการที่จะทำให้โปรแกรมรับรู้ตำแหน่งเดิมในทุกรอบของการหมุน จำเป็นต้องมีตัว sensor เพื่อคอยตรวจสอบตำแหน่งแล้วส่งสัญญาณให้โปรแกรมได้รับรู้ สำหรับ sensor ที่เลือกใช้ คือ Infrared Sensor เพราะความถี่ของ Infrared อยู่ในย่านที่สายตามนุษย์มองไม่เห็น ทำให้ไม่รบกวนการแสดงผลของ LED ที่ใช้แสดงตัวอักษรโดยตัวส่ง (TX) จะเป็นตัวที่อยู่กับที่ส่วนตัวรับ (RX) จะเป็นตัวที่ติดอยู่กับบอร์ดเมื่อตำแหน่งของตัวส่งและตัวรับตรงกันก็จะทำให้โปรแกรมรับรู้ตำแหน่งที่จะส่งข้อมูลชุดเดิมออกไป เพื่อให้ LED ดูเหมือนติดอยู่ตำแหน่งเดิมตลอดเวลา

3.2.4 การออกแบบวิธีการส่งสัญญาณไฟฟ้าไปยังวงจร

ปัญหาที่สำคัญอย่างหนึ่งของโครงงานชิ้นนี้คือ การส่งสัญญาณไฟฟ้าไปสู่วงจรซึ่งจะต้องหมุนตลอดเวลาในขณะที่เราใช้งานทำให้ไม่สามารถใช้สายสัญญาณได้เพราะจะทำให้พันกัน จึงได้ค้นพบวิธีการอยู่ 2 วิธีด้วยกัน คือ

1. ใช้ Battery วางไว้บนวงจรด้านบน
2. ใช้แปรงถ่านเป็นตัวนำกระแสขึ้นไปสู่วงจรด้านบน

หลังจากที่ได้ทำการศึกษา และเปรียบเทียบข้อดีข้อเสีย ของแต่ละวิธีแล้วจึงเลือกวิธีใช้แปรงถ่านเป็นตัวนำกระแสไฟฟ้าไปสู่วงจรด้านบนเพราะจะสะดวกกว่าการใช้ Battery มาก ถึงแม้ว่าจะออกแบบและติดตั้งยากกว่าก็ตาม

การคำนวณและออกแบบที่ผ่านมาเป็นการคำนวณและออกแบบคร่าวๆ เพื่อที่จะนำไปเป็นแนวทางสู่การออกแบบที่ละเอียดขึ้นในบทต่อไป ซึ่งจะกล่าวถึงโครงสร้างทั้งหมดของโครงงานรวมทั้งการออกแบบในแต่ละส่วนด้วย

บทที่ 4

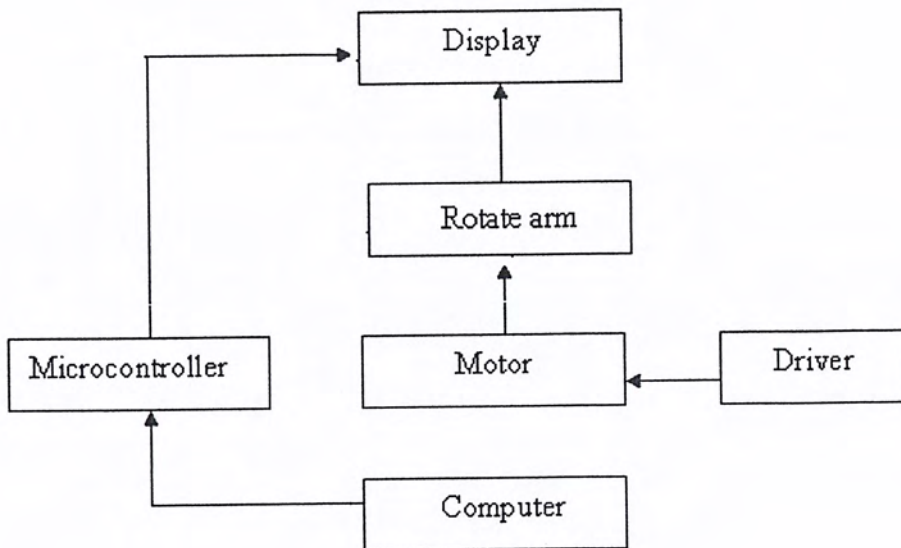
การออกแบบและการสร้างบอร์ดแสดงผลแบบหมุน

ส่วนประกอบของโครงการ

แบ่งส่วนประกอบของโครงการออกตามโครงสร้างได้เป็น 2 ส่วนหลักคือ

1. ฮาร์ดแวร์
2. ซอฟต์แวร์

4.1 การออกแบบด้านฮาร์ดแวร์



รูปที่ 4.1 แสดงส่วนประกอบของฮาร์ดแวร์

4.1.1 การออกแบบวงจร

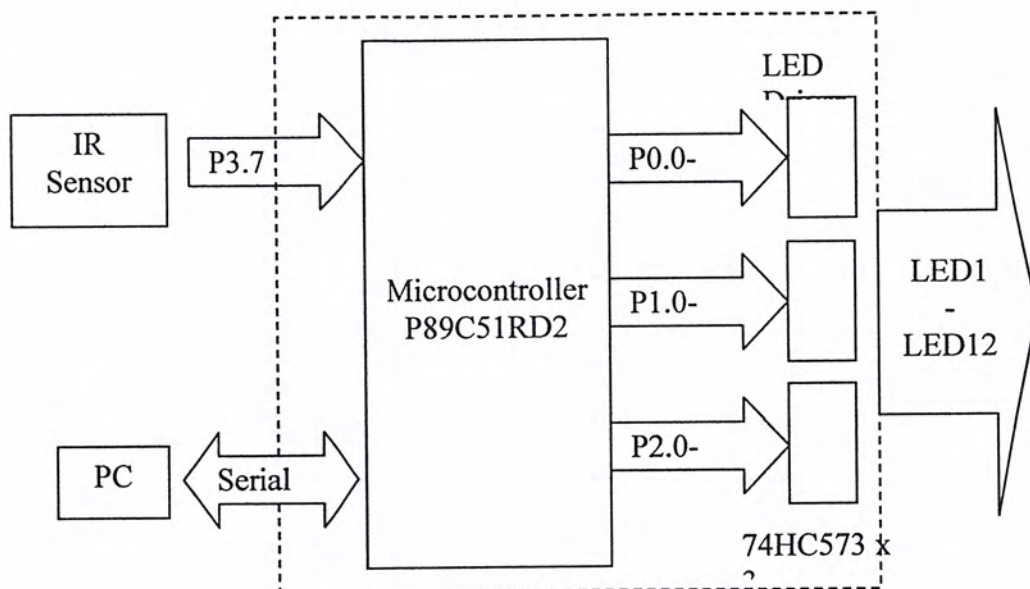
ข้อจำกัดของการออกแบบวงจร

1. วงจรที่อยู่บนบอร์ดต้องน้อยที่สุดเท่าที่จะเป็นไปได้ เพื่อที่จะทำให้อายุการใช้งานของบอร์ดเล็กที่สุดเท่าที่จะเป็นไปได้ เป็นการลด Torque ของ Motor และลดการสั่นสะเทือน
2. การวางอุปกรณ์บนบอร์ดต้องให้สมดุลกัน ทั้ง 2 ฝั่งเพื่อลดการสั่นสะเทือน

แบ่งวงจรออกเป็น 4 ส่วน คือ

1. วงจร Microcontroller Board
2. วงจร RS232 Interface
3. วงจร Motor Driver & Power Supply
4. วงจร Infrared Sensor

วงจร Microcontroller Board

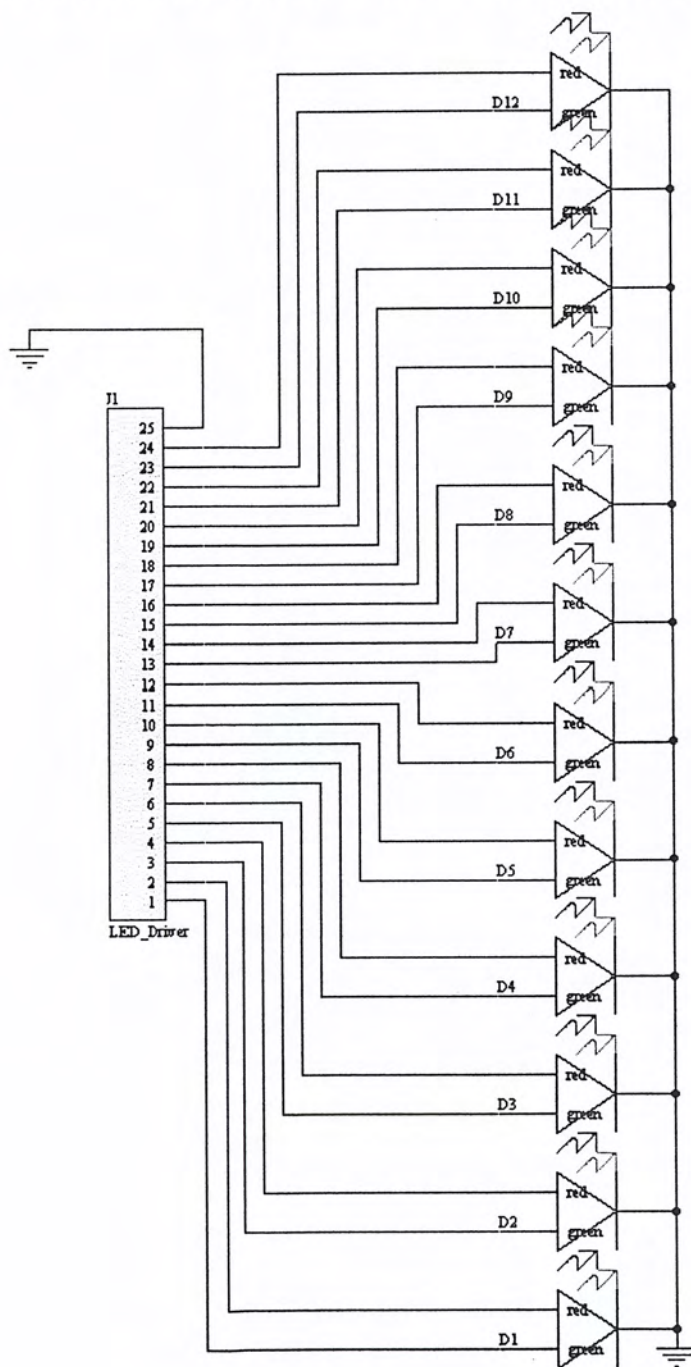


รูปที่ 4.2 Block Diagram การทำงานของวงจร Microcontroller Board

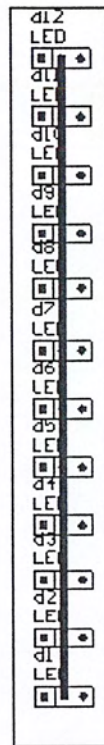
จาก Block Diagram ใช้ไมโครคอนโทรลเลอร์บอร์ด P89C51RD2 เป็นตัวควบคุมการทำงานทั้งหมดโดยจะทำหน้าที่รับค่าตัวอักษรจากคอมพิวเตอร์ผ่านพอร์ตอนุกรมมาประมวลผลแล้วส่งออกทางพอร์ต 0 ถึง พอร์ต 2 ไปที่ ไอซี 74HC573 ซึ่งทำหน้าที่ขับกระแสส่งต่อไปยังวงจร Display คือ LED1 ถึง LED12 นั่นเอง ส่วนพอร์ต 3.7 ใช้เป็นพอร์ตอินพุตรับค่าจากวงจรอินฟราเรดเซ็นเซอร์ เพื่อคอยเช็คตำแหน่งการหมุนครบรอบของมอเตอร์ทำให้ตัวอักษรที่แสดงออกทาง Display ไม่คลาดเคลื่อน

วงจร LED Display

LED ทั้ง 12 ดวง ซึ่งเป็น LED 2 สี รับสัญญาณจาก IC Driver เพื่อแสดงตัวอักษรตามคำสั่งจากไมโครคอนโทรลเลอร์ โดยจะสามารถแสดงผลได้ถึงสามสี คือจากการป้อนลอจิกหนึ่งให้แต่ละขาจะเป็นสีเขียวและสีแดง และถ้าป้อนลอจิกหนึ่งพร้อมกันทั้งสองขา ก็จะเป็นสีส้ม

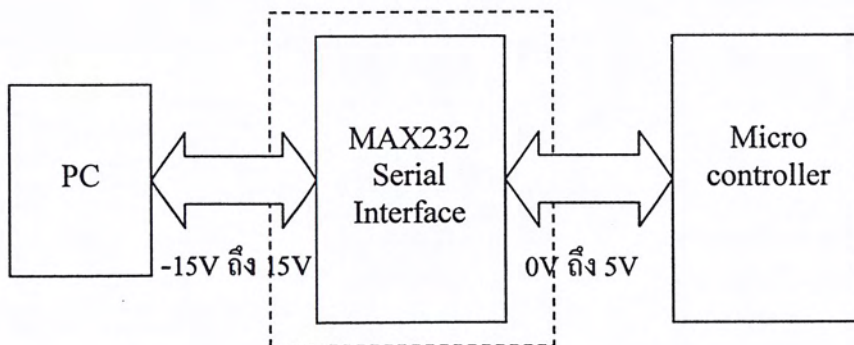


รูปที่ 4.4 แสดงวงจร LED Display



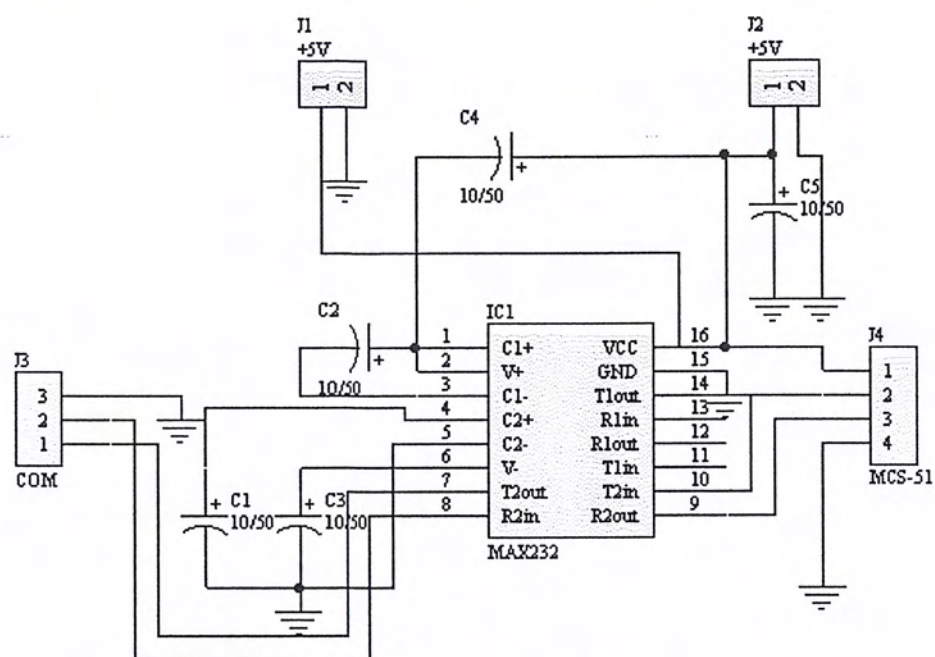
รูปที่ 4.5 แสดงลายปรีนและการวางอุปกรณ์ของวงจร LED Display

วงจร RS232 Interface



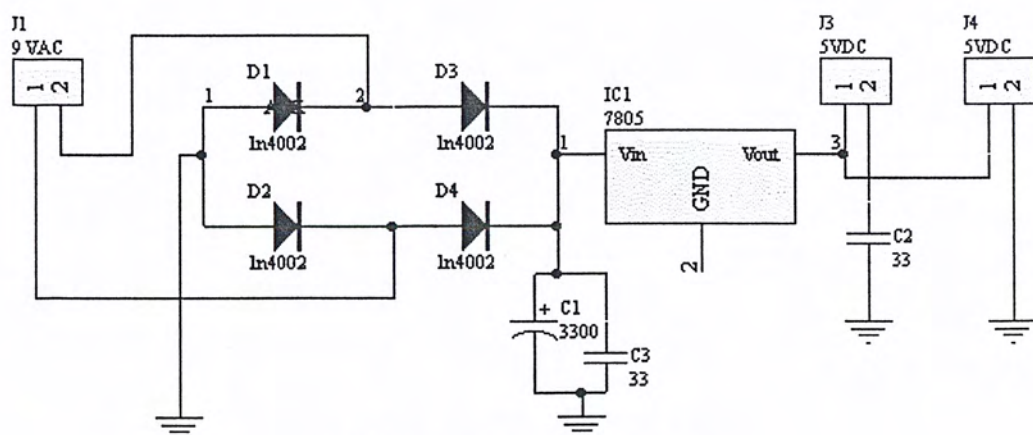
รูปที่ 4.6 Block Diagram การทำงานของวงจร RS232 Interface

จาก Block Diagram ไอซี MAX232 เป็นไอซีที่แปลงระดับสัญญาณจากระดับ TTL ไปเป็นระดับของ RS-232 และในทำนองเดียวกันก็รับระดับสัญญาณจาก RS-232 เพื่อแปลงเป็นระดับสัญญาณจากระดับ TTL ให้กับไมโครคอนโทรลเลอร์ได้



รูปที่ 4.7 แสดงวงจร RS232 Interface

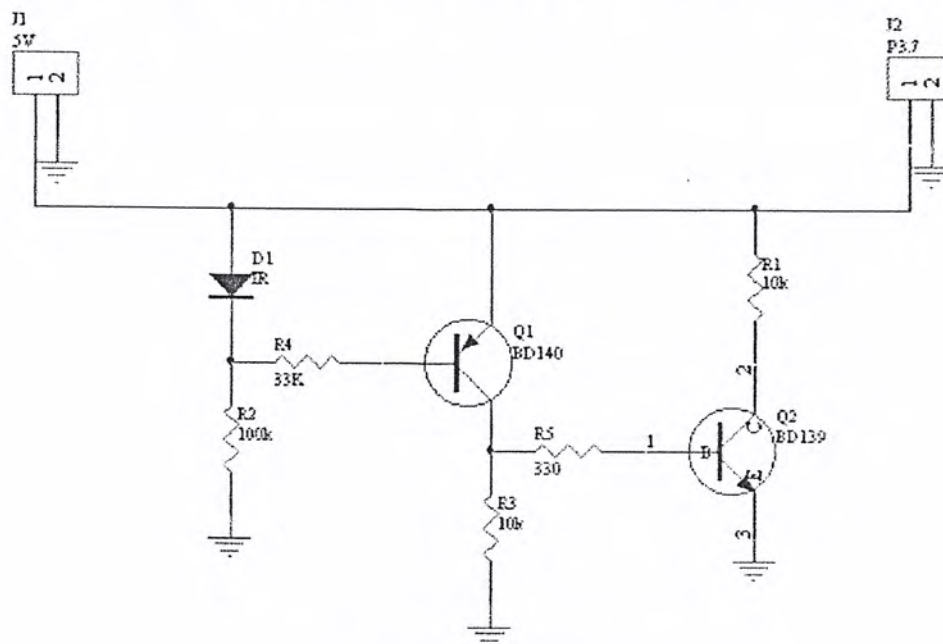
วงจร Power Supply



รูปที่ 4.8 แสดงวงจร Power Supply

วงจร Infrared Sensor

วงจร Infrared Sensor ทำหน้าที่ตรวจเช็คตำแหน่งการหมุนของมอเตอร์เพื่อเป็นการรับรู้ตำแหน่งที่จะส่งข้อมูลชุดเดิมทุกรอบการหมุน โดยอินฟราเรดเซนเซอร์ D1 ซึ่งเป็นส่วนที่หมุนติดอยู่กับบอร์ดไมโครคอนโทรลเลอร์ และรับสัญญาณจากตัวส่งที่ติดอยู่กับที่ ส่วน Q1 และ Q2 ทำหน้าที่เป็นสวิตช์เพื่อตัดต่อสัญญาณจากตัว เพื่อส่งให้ไมโครคอนโทรลเลอร์ต่อไป

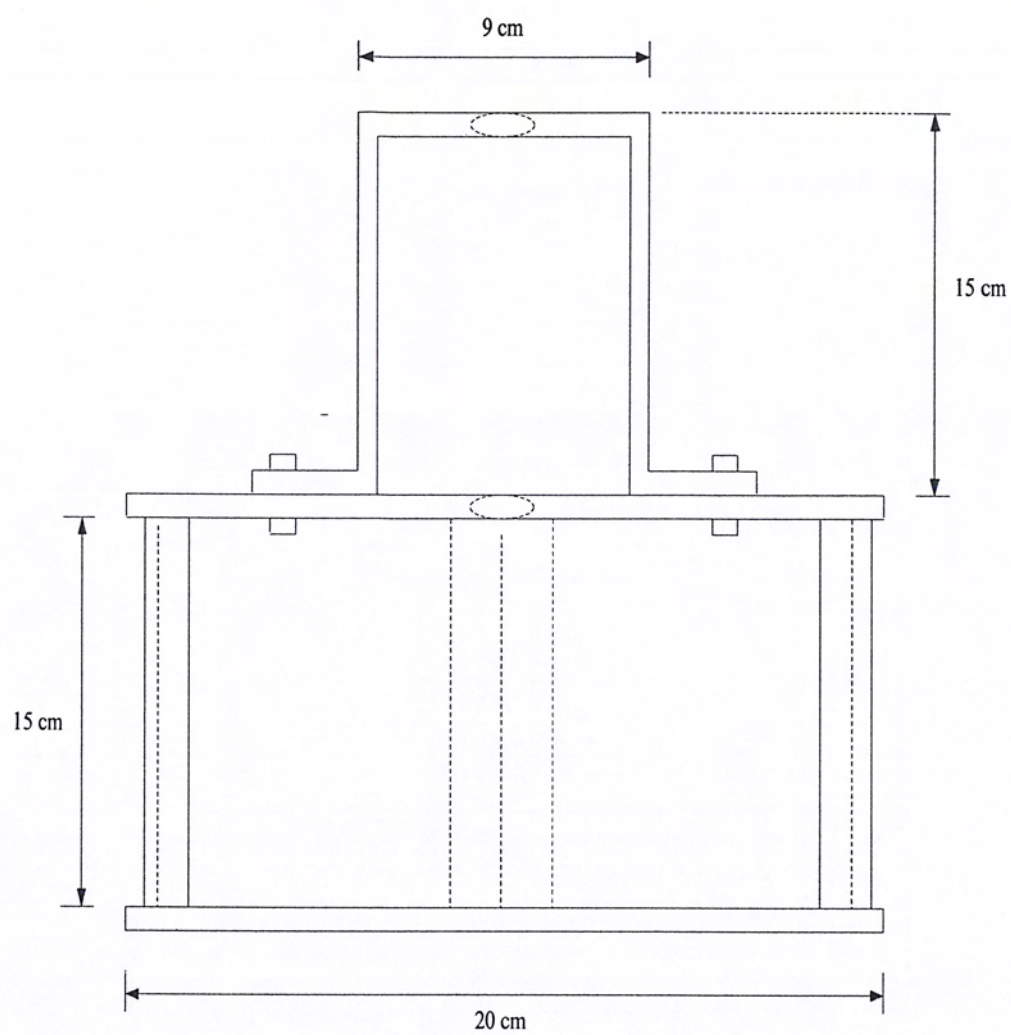


รูปที่ 4.10 แสดงวงจร Infrared Sensor

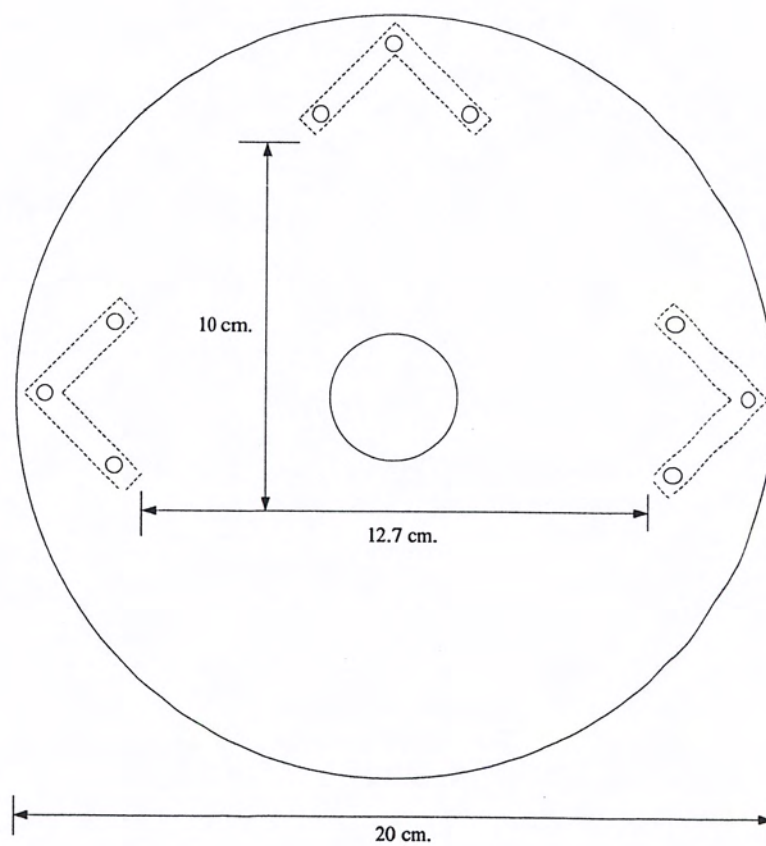
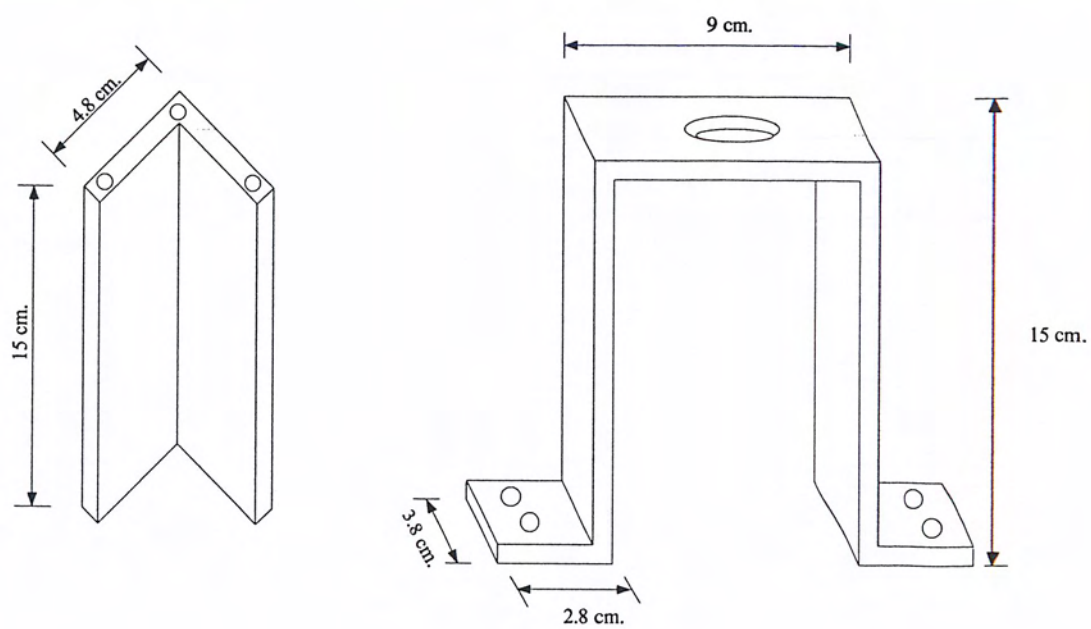
4.1.2 การออกแบบโครงยึดส่วนต่างๆเข้าด้วยกัน

ข้อจำกัดในการออกแบบ

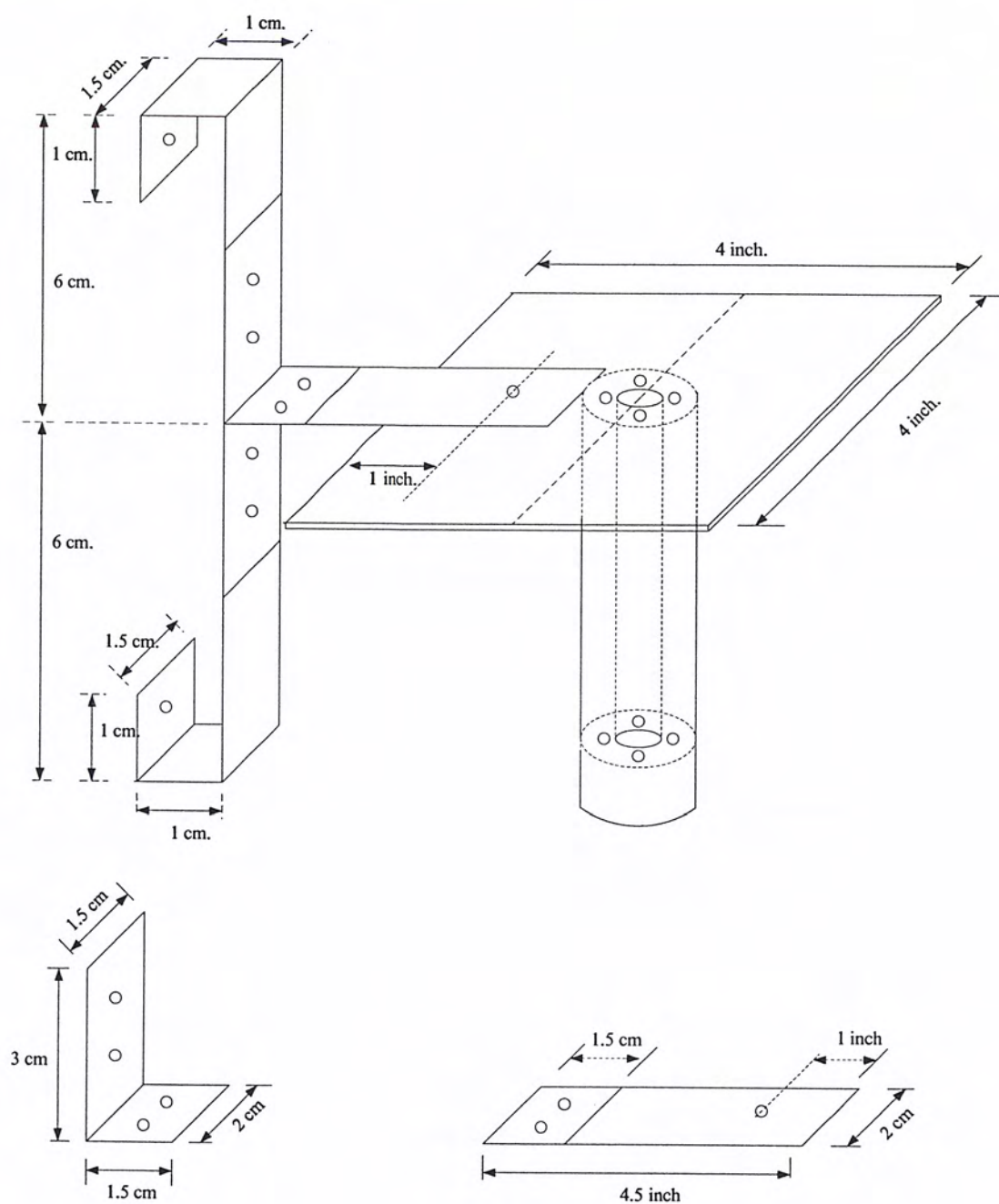
1. เพื่อความสะดวกในการเคลื่อนย้ายขนาดของโครงสร้างไม่ควรจะมีขนาดใหญ่จนเกินไป และต้องมีความแข็งแรงไม่บอบบางจนเกินไปรวมทั้งน้ำหนักต้องไม่มากเกินไปเหมาะแก่การเคลื่อนย้าย
2. เนื่องจากในขณะที่ใช้งานแรงเหวี่ยงมอเตอร์อาจทำให้โครงสร้างสั่นสะเทือนทำให้เกิดเสียงหรือการเคลื่อนที่ได้ ดังนั้น โครงสร้างต้องมีน้ำหนักมากพอที่จะรับแรงเหวี่ยงนี้ได้ รวมทั้งการออกแบบแขนต้องให้มีพื้นที่ที่ด้านลมน้อยที่สุดหรือให้แขนสั้นที่สุดเพื่อลดแรงเหวี่ยงให้น้อยที่สุด



รูปที่ 4.11 แสดงภาพด้านหน้าของโครงยึดมอเตอร์



รูปที่ 4.12 แสดงขนาดชิ้นส่วนของ โครงขีคมอเตอร์



รูปที่ 4.13 แสดงขนาดชิ้นส่วนของ โครงยึดมอเตอร์

4.2 การออกแบบโปรแกรม

แบ่งโปรแกรมออกเป็น 2 ส่วน คือ

1. โปรแกรมการรับค่าและแสดงผลของ Microcontroller โดยใช้ภาษา C
2. โปรแกรมการส่งค่าจาก Computer โดยใช้ภาษา Visual Basic

4.2.1 การออกแบบโปรแกรมการแสดงผลโดยไมโครคอนโทรลเลอร์

ตารางที่ 4.1 แสดงตัวอย่างการเข้ารหัสตัวอักษรเพื่อแสดงผล

Port Color	Red						Green						Orange						Port Color
0.0																			1.0
0.1																			1.1
0.2																			1.2
0.3																			1.3
0.4																			1.4
0.5																			1.5
0.6																			1.6
0.7																			1.7
2.0																			2.4
2.1																			2.5
2.2																			2.6
2.3																			2.7
Hex 0	E0	1E	01	1E	E0	00	00	00	00	00	00	00	E0	1E	01	1E	E0	00	Hex 0
Hex 1	00	00	00	00	00	00	E0	1E	01	1E	E0	00	E0	1E	01	1E	E0	00	Hex 1
Hex 2	0F	01	01	01	0F	00	F0	10	10	10	F0	00	FF	11	11	11	FF	00	Hex 2

Port Color	Red						Green						Orange						Port Color
0.0																			1.0
0.1																			1.1
0.2																			1.2
0.3																			1.3
0.4																			1.4
0.5																			1.5
0.6																			1.6
0.7																			1.7
2.0																			2.4
2.1																			2.5
2.2																			2.6
2.3																			2.7
Hex 0	FF	61	61	61	9E	00	00	00	00	00	00	00	FF	61	61	61	9E	00	Hex 0
Hex 1	00	00	00	00	00	00	FF	61	61	61	9E	00	FF	61	61	61	9E	00	Hex 1
Hex 2	0F	08	08	08	07	00	F0	80	80	80	70	00	FF	88	88	88	77	00	Hex 2

Port Color	Red						Green						Orange						Port Color
0.0																			1.0
0.1																			1.1
0.2																			1.2
0.3																			1.3
0.4																			1.4
0.5																			1.5
0.6																			1.6
0.7																			1.7
2.0																			2.4
2.1																			2.5
2.2																			2.6
2.3																			2.7
Hex 0	E0	1E	01	1E	E0	00	00	00	00	00	00	00	E0	1E	01	1E	E0	00	Hex 0
Hex 1	00	00	00	00	00	00	E0	1E	01	1E	E0	00	E0	1E	01	1E	E0	00	Hex 1
Hex 2	0F	01	01	01	0F	00	F0	10	10	10	F0	00	FF	11	11	11	FF	00	Hex 2

Port Color	Red						Green						Orange						Port Color
0.0																			1.0
0.1																			1.1
0.2																			1.2
0.3																			1.3
0.4																			1.4
0.5																			1.5
0.6																			1.6
0.7																			1.7
2.0																			2.4
2.1																			2.5
2.2																			2.6
2.3																			2.7
Hex 0	FF	01	01	02	F8	00	00	00	00	00	00	00	FF	01	01	02	F8	00	Hex 0
Hex 1	00	00	00	00	00	00	FF	01	01	02	F8	00	FF	01	01	02	F8	00	Hex 1
Hex 2	0F	08	08	04	01	00	F0	80	80	40	10	00	FF	88	88	44	11	00	Hex 2

Port Color	Red						Green						Orange						Port Color
0.0	●	●	●	●	●		●	●	●	●	●		●	●	●	●	●		1.0
0.1	●						●						●						1.1
0.2	●						●						●						1.2
0.3	●						●						●						1.3
0.4	●						●						●						1.4
0.5	●	●	●	●	●		●	●	●	●	●		●	●	●	●	●		1.5
0.6	●	●	●	●	●		●	●	●	●	●		●	●	●	●	●		1.6
0.7	●						●						●						1.7
2.0	●						●						●						2.4
2.1	●						●						●						2.5
2.2	●						●						●						2.6
2.3	●	●	●	●	●		●	●	●	●	●		●	●	●	●	●		2.7
Hex 0	FF	61	61	61	61	00	00	00	00	00	00	00	FF	61	61	61	61	00	Hex 0
Hex 1	00	00	00	00	00	00	FF	61	61	61	61	00	FF	61	61	61	61	00	Hex 1
Hex 2	0F	08	08	08	08	00	F0	80	80	80	80	00	FF	88	88	88	88	00	Hex 2

Port Color	Red						Green						Orange						Port Color
0.0	●	●	●	●	●		●	●	●	●	●		●	●	●	●	●		1.0
0.1	●						●						●						1.1
0.2	●						●						●						1.2
0.3	●						●						●						1.3
0.4	●	●	●	●	●		●	●	●	●	●		●	●	●	●	●		1.4
0.5	●						●						●						1.5
0.6	●						●						●						1.6
0.7	●						●						●						1.7
2.0	●						●						●						2.4
2.1	●						●						●						2.5
2.2	●						●						●						2.6
2.3	●						●						●						2.7
Hex 0	FF	11	11	11	11	00	00	00	00	00	00	00	FF	11	11	11	11	00	Hex 0
Hex 1	00	00	00	00	00	00	FF	11	11	11	11	00	FF	11	11	11	11	00	Hex 1
Hex 2	0F	00	00	00	00	00	F0	00	00	00	00	00	FF	00	00	00	00	00	Hex 2

Port Color	Red						Green						Orange						Port Color
0.0																			1.0
0.1																			1.1
0.2																			1.2
0.3																			1.3
0.4																			1.4
0.5																			1.5
0.6																			1.6
0.7																			1.7
2.0																			2.4
2.1																			2.5
2.2																			2.6
2.3																			2.7
Hex 0	EE	01	61	21	E2	00	00	00	00	00	00	00	EE	01	61	21	E2	00	Hex 0
Hex 1	00	00	00	00	00	00	EE	01	61	21	E2	00	EE	01	61	21	E2	00	Hex 1
Hex 2	07	08	08	08	07	00	70	80	80	80	70	00	77	88	88	88	77	00	Hex 2

Port Color	Red						Green						Orange						Port Color
0.0																			1.0
0.1																			1.1
0.2																			1.2
0.3																			1.3
0.4																			1.4
0.5																			1.5
0.6																			1.6
0.7																			1.7
2.0																			2.4
2.1																			2.5
2.2																			2.6
2.3																			2.7
Hex 0	FF	60	60	60	FF	00	00	00	00	00	00	00	FF	60	60	60	FF	00	Hex 0
Hex 1	00	00	00	00	00	00	FF	60	60	60	FF	00	FF	60	60	60	FF	00	Hex 1
Hex 2	0F	00	00	00	0F	00	F0	00	00	00	F0	00	FF	00	00	00	FF	00	Hex 2

Port Color	Red						Green						Orange						Port Color
0.0																			1.0
0.1																			1.1
0.2																			1.2
0.3																			1.3
0.4		●	●	●				●	●	●				●	●	●			1.4
0.5	●				●		●				●		●				●		1.5
0.6	●				●		●				●		●				●		1.6
0.7	●				●		●				●		●				●		1.7
2.0	●				●		●				●		●				●		2.4
2.1	●				●		●				●		●				●		2.5
2.2	●			●	●		●			●	●		●			●	●		2.6
2.3		●	●		●			●	●		●			●	●		●		2.7
Hex 0	E0	10	10	10	F0	00	00	00	00	00	00	00	E0	10	10	10	F0	00	Hex 0
Hex 1	00	00	00	00	00	00	E0	10	10	10	F0	00	E0	10	10	10	F0	00	Hex 1
Hex 2	07	08	08	04	0F	00	70	80	80	40	F0	00	77	88	88	44	FF	00	Hex 2

Port Color	Red						Green						Orange						Port Color
0.0																			1.0
0.1																			1.1
0.2	●						●						●						1.2
0.3	●						●						●						1.3
0.4	●						●						●						1.4
0.5	●						●						●						1.5
0.6	●						●						●						1.6
0.7	●	●	●	●			●	●	●	●			●	●	●	●			1.7
2.0	●				●		●				●		●				●		2.4
2.1	●				●		●				●		●				●		2.5
2.2	●				●		●				●		●				●		2.6
2.3	●	●	●	●			●	●	●	●			●	●	●	●			2.7
Hex 0	FE	80	80	80	00	00	00	00	00	00	00	00	FE	80	80	80	00	FE	Hex 0
Hex 1	00	00	00	00	00	00	FE	80	80	80	00	FE	FE	80	80	80	00	FE	Hex 1
Hex 2	0F	08	08	08	07	00	F0	80	80	80	70	00	FF	88	88	88	77	00	Hex 2

Port Color	Red						Green						Orange						Port Color
0.0																			1.0
0.1																			1.1
0.2																			1.2
0.3																			1.3
0.4		●	●	●				●	●	●				●	●	●			1.4
0.5	●				●		●				●		●				●		1.5
0.6	●						●						●						1.6
0.7	●						●						●						1.7
2.0	●						●						●						2.4
2.1	●						●						●						2.5
2.2	●				●		●				●		●				●		2.6
2.3		●	●	●				●	●	●				●	●	●			2.7
Hex 0	E0	10	10	10	20	00	00	00	00	00	00	00	E0	10	10	10	20	00	Hex 0
Hex 1	00	00	00	00	00	00	E0	10	10	10	20	00	E0	10	10	10	20	00	Hex 1
Hex 2	07	08	08	08	04	00	70	80	80	80	40	00	77	88	88	88	44	00	Hex 2

Port Color	Red						Green						Orange						Port Color
0.0																			1.0
0.1																			1.1
0.2					●						●						●		1.2
0.3					●						●						●		1.3
0.4					●						●						●		1.4
0.5					●						●						●		1.5
0.6					●						●						●		1.6
0.7		●	●	●	●			●	●	●	●			●	●	●	●		1.7
2.0	●				●		●				●		●				●		2.4
2.1	●				●		●				●		●				●		2.5
2.2	●				●		●				●		●				●		2.6
2.3		●	●	●	●			●	●	●	●			●	●	●	●		2.7
Hex 0	00	80	80	80	FC	00	00	00	00	00	00	00	00	80	80	80	FC	00	Hex 0
Hex 1	00	00	00	00	00	00	00	80	80	80	FC	00	00	80	80	80	FC	00	Hex 1
Hex 2	07	08	08	08	0F	00	70	80	80	80	F0	00	77	88	88	88	FF	00	Hex 2

Port Color	Red						Green						Orange						Port Color
0.0																			1.0
0.1																			1.1
0.2																			1.2
0.3																			1.3
0.4		●	●	●				●	●	●				●	●	●			1.4
0.5	●				●		●				●		●				●		1.5
0.6	●				●		●				●		●				●		1.6
0.7	●				●		●				●		●				●		1.7
2.0	●	●	●	●			●	●	●	●			●	●	●	●			2.4
2.1	●						●						●						2.5
2.2	●				●		●				●		●				●		2.6
2.3		●	●	●				●	●	●				●	●	●			2.7
Hex 0	E0	10	10	10	E0	00	00	00	00	00	00	00	E0	10	10	10	E0	00	Hex 0
Hex 1	00	00	00	00	00	00	E0	10	10	10	E0	00	E0	10	10	10	E0	00	Hex 1
Hex 2	07	09	09	09	04	00	70	90	90	90	40	00	77	99	99	99	44	00	Hex 2

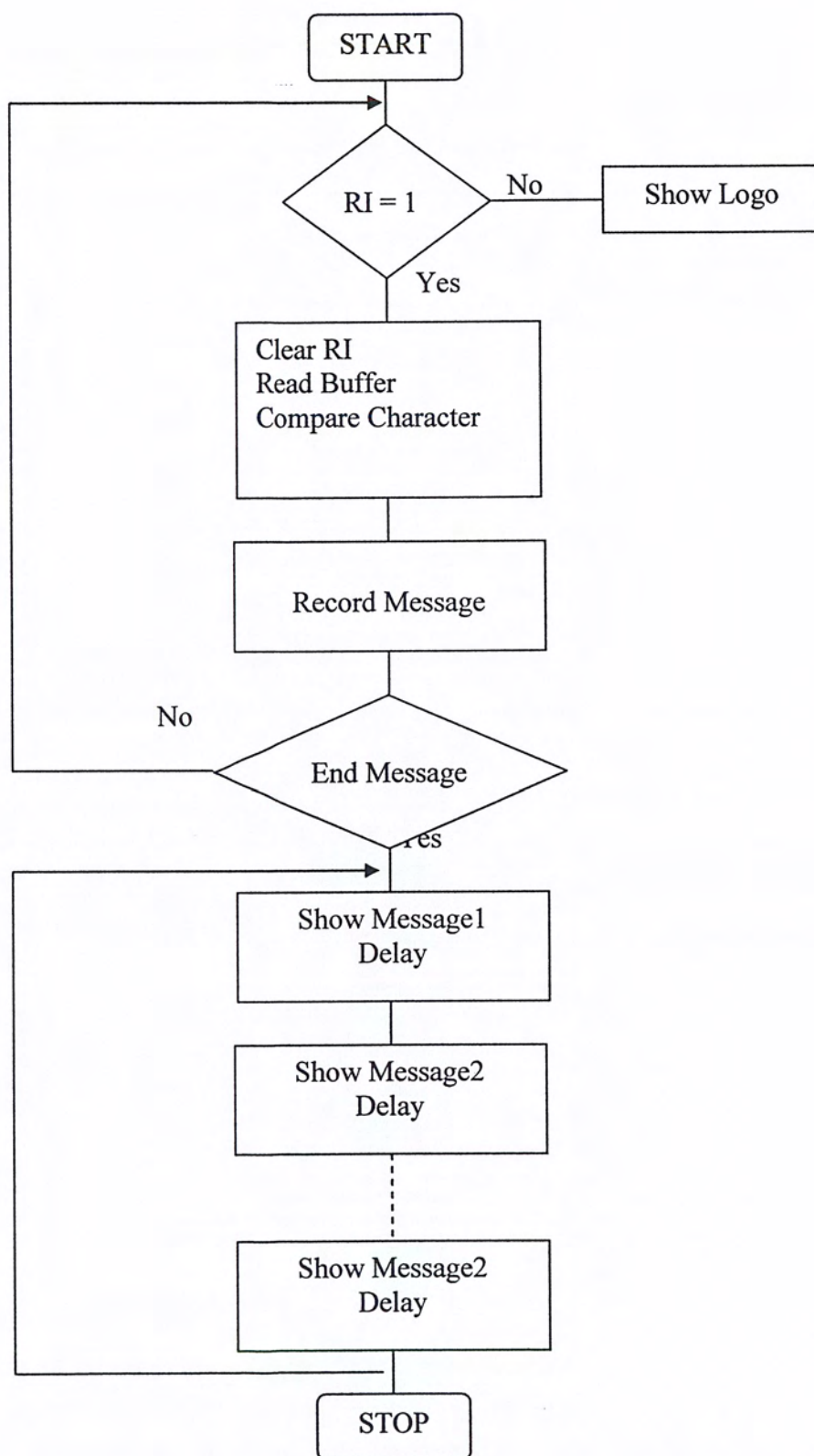
Port Color	Red						Green						Orange						Port Color
0.0																			1.0
0.1																			1.1
0.2			●	●				●	●					●	●				1.2
0.3		●			●			●			●			●			●		1.3
0.4		●						●						●					1.4
0.5	●	●	●	●			●	●	●	●			●	●	●	●			1.5
0.6		●						●						●					1.6
0.7		●						●						●					1.7
2.0		●						●						●					2.4
2.1		●						●						●					2.5
2.2		●						●						●					2.6
2.3		●						●						●					2.7
Hex 0	20	F8	24	24	08	00	00	00	00	00	00	00	20	F8	24	24	08	00	Hex 0
Hex 1	00	00	00	00	00	00	20	F8	24	24	08	00	20	F8	24	24	08	00	Hex 1
Hex 2	00	0F	00	00	00	00	00	F0	00	00	00	00	00	FF	00	00	00	00	Hex 2

Port Color	Red						Green						Orange						Port Color
0.0			●						●						●				1.0
0.1		●	●					●	●					●	●				1.1
0.2	●		●					●					●		●				1.2
0.3			●						●						●				1.3
0.4			●						●						●				1.4
0.5			●						●						●				1.5
0.6			●						●						●				1.6
0.7			●						●						●				1.7
2.0			●						●						●				2.4
2.1			●						●						●				2.5
2.2			●						●						●				2.6
2.3	●	●	●	●	●		●	●	●	●	●		●	●	●	●	●		2.7
Hex 0	04	02	FF	00	00	00	00	00	00	00	00	00	04	02	FF	00	00	00	Hex 0
Hex 1	00	00	00	00	00	00	04	02	FF	00	00	00	04	02	FF	00	00	00	Hex 1
Hex 2	08	08	0F	08	08	00	80	80	F0	80	80	00	88	88	FF	88	88	00	Hex 2

Port Color	Red						Green						Orange						Port Color
0.0		●	●	●				●	●	●				●	●	●			1.0
0.1	●				●		●				●		●				●		1.1
0.2					●						●						●		1.2
0.3					●						●						●		1.3
0.4				●						●						●			1.4
0.5				●						●						●			1.5
0.6			●						●						●				1.6
0.7			●						●						●				1.7
2.0		●						●						●					2.4
2.1		●						●						●					2.5
2.2	●						●						●						2.6
2.3	●	●	●	●	●		●	●	●	●	●		●	●	●	●	●		2.7
Hex 0	02	01	C1	31	0C	00	00	00	00	00	00	00	02	01	C1	31	0C	00	Hex 0
Hex 1	00	00	00	00	00	00	02	01	C1	31	0C	00	02	01	C1	31	0C	00	Hex 1
Hex 2	08	0B	08	08	08	00	80	B0	80	80	80	00	88	BB	88	88	88	00	Hex 2

Port Color	Red						Green						Orange						Port Color
0.0																			1.0
0.1																			1.1
0.2																			1.2
0.3																			1.3
0.4																			1.4
0.5																			1.5
0.6																			1.6
0.7																			1.7
2.0																			2.4
2.1																			2.5
2.2																			2.6
2.3																			2.7
Hex 0	02	61	61	91	0E	00	00	00	00	00	00	00	02	61	61	91	0E	00	Hex 0
Hex 1	00	00	00	00	00	00	02	61	61	91	0E	00	02	61	61	91	0E	00	Hex 1
Hex 2	04	08	08	08	07	00	40	80	80	80	70	00	44	88	88	88	77	00	Hex 2

Port Color	Red						Green						Orange						Port Color
0.0																			1.0
0.1																			1.1
0.2																			1.2
0.3																			1.3
0.4																			1.4
0.5																			1.5
0.6																			1.6
0.7																			1.7
2.0																			2.4
2.1																			2.5
2.2																			2.6
2.3																			2.7
Hex 0	FF	80	80	80	FF	00	00	00	00	00	00	00	FF	80	80	80	FF	00	Hex 0
Hex 1	00	00	00	00	00	00	FF	80	80	80	FF	00	FF	80	80	80	FF	00	Hex 1
Hex 2	00	00	00	00	0F	00	00	00	00	00	F0	00	00	00	00	00	FF	00	Hex 2



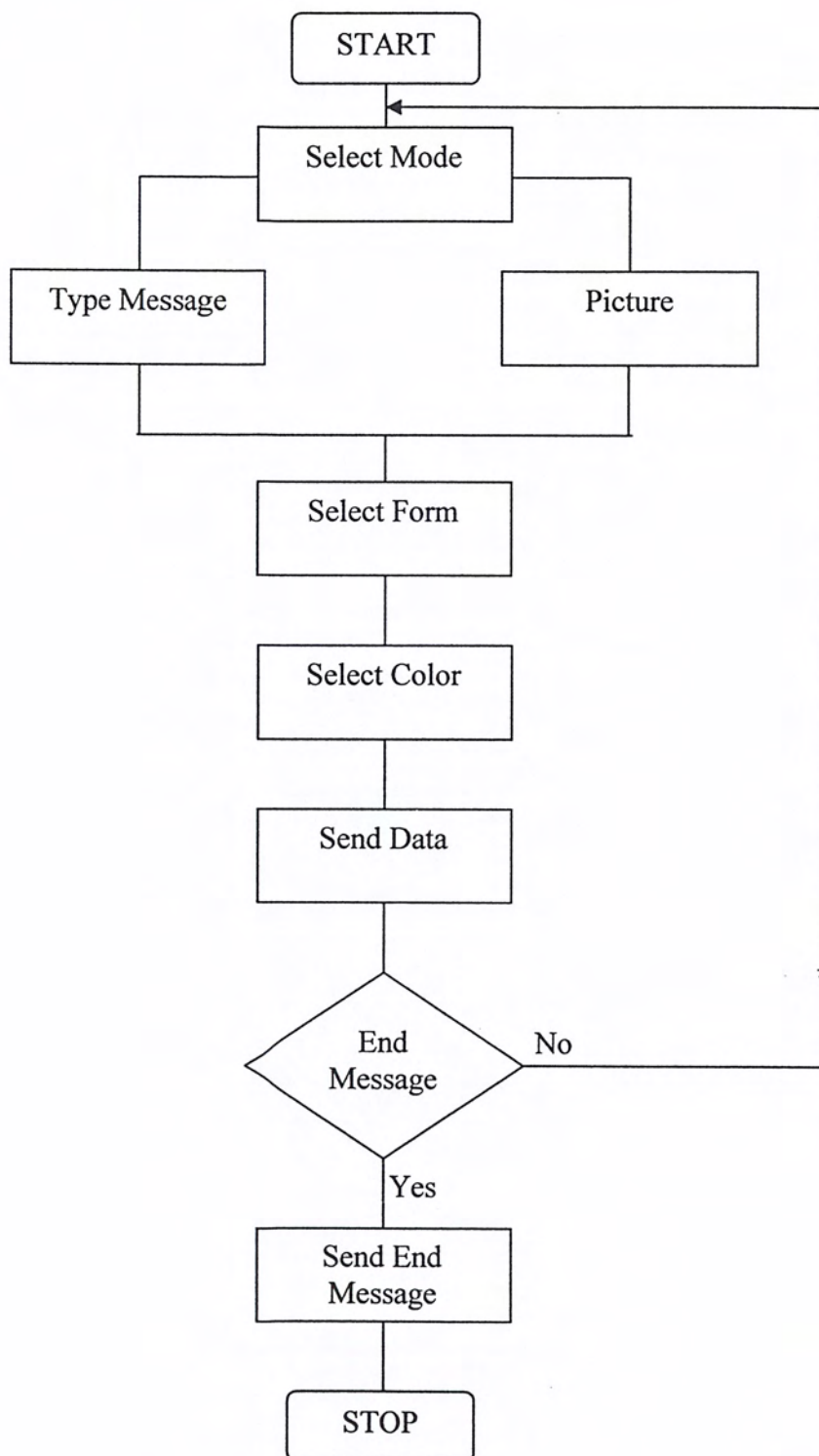
รูปที่ 4.14 โฟลว์ชาร์ตแสดงการรับค่าและควบคุมการแสดงผลของ Display

การทำงานของโปรแกรม

จากไฟล์ชาร์ตแสดงการทำงานของโปรแกรมการรับค่าจากคอมพิวเตอร์ และควบคุมการแสดงผลของ Display โดยควบคุมจากไมโครคอนโทรลเลอร์ ซึ่งจะคอยตรวจเช็คสัญญาณอินเทอร์รัพท์จากพอร์ตอนุกรมที่ถูกส่งมาจากคอมพิวเตอร์ วิธีการโดยตรวจสอบที่บิต RI ถ้ามีการส่งข้อมูลมาที่พอร์ตอนุกรมบิต RI จะทำการเซต แต่ถ้ายังไม่มีการส่งข้อมูลมาจากคอมพิวเตอร์หรือบิต RI ยังไม่ได้ถูกเซต ไมโครคอนโทรลเลอร์ก็จะทำการส่งข้อมูลที่ได้ออกแบบไว้ไปแสดงผลที่ Display หลังจากรับรู้ว่าการส่งข้อมูลมาแล้วไมโครคอนโทรลเลอร์ก็จะทำการอ่านข้อมูลนั้นเข้าเก็บไว้ในหน่วยความจำภายในพร้อมกับทำการเคลียร์บิต RI ด้วยเพื่อที่จะได้ตรวจเช็คการส่งข้อมูลชุดใหม่ได้เพราะบิตนี้จะต้องเคลียร์ด้วยโปรแกรม หลังจากนั้นไมโครคอนโทรลเลอร์ก็จะนำข้อมูลที่ได้รับมาซึ่งเป็นรหัสแอสกีมาประมวลผลโดยทำการเปรียบเทียบกับฐานข้อมูลที่ได้จัดทำไว้แล้ว และนำข้อมูลที่ได้ออกมาแล้วไปเก็บไว้ในหน่วยความจำอีกที่หนึ่ง เพื่อเตรียมที่จะนำไปแสดงผลที่ Display ซึ่งข้อมูลเหล่านี้จะประกอบไปด้วยตัวอักษรที่เป็นข้อความหรือภาพ รวมทั้งสีของข้อความนั้น ซึ่งมีให้เลือกสามสีคือ สีแดง สีเขียวและสีส้ม และรูปแบบของการแสดงผลได้แก่ข้อความอยู่นิ่งหรือว่าเลื่อนซ้ายขวา

จากนั้นไมโครคอนโทรลเลอร์จะทำการตรวจสอบว่าข้อมูลที่รับมานั้นเป็นข้อมูลชุดสุดท้ายหรือไม่หากยังไม่ใช่ก็จะรอรับข้อมูลชุดใหม่ แต่หากตรวจสอบแล้วพบว่าเป็นข้อมูลชุดสุดท้ายก็จะนำข้อมูลแต่ละชุดไปแสดงผลทีละชุดวนซ้ำกันไปตามข้อมูลที่ได้รับเข้ามาจากคอมพิวเตอร์ จนกว่าจะมีการปิดเครื่อง

4.2.2 การออกแบบโปรแกรมการส่งค่าจากคอมพิวเตอร์



รูปที่ 4.15 โฟลว์ชาร์ตการส่งค่ารูปแบบตัวอักษรจากคอมพิวเตอร์

การทำงานของโปรแกรม

จากโฟลว์ชาร์ตแสดงการการส่งค่ารูปแบบตัวอักษรจากคอมพิวเตอร์ มีการทำงาน คือ โปรแกรมจะให้ผู้ใช้งานเลือกก่อนว่าต้องการส่งข้อมูลในรูปแบบใด ซึ่งมีให้เลือก 2 รูปแบบด้วยกัน คือ โหมดการพิมพ์ ซึ่งเป็นการแสดงข้อความ และโหมดการเลือกรูปภาพ ซึ่งรูปภาพที่มีให้เลือกจะเป็นรูปภาพที่เก็บข้อมูลอยู่ในตัวไมโครคอนโทรลเลอร์ หลังจากทำการเลือกโหมดแล้วก็จะเข้าสู่การเลือกรูปแบบการแสดงผลซึ่งมีให้เลือก 3 รูปแบบ คือ ข้อความอยู่นิ่ง เลื่อนซ้าย และเลื่อนขวา เมื่อเลือกเสร็จแล้วก็จะทำการเลือกสีของข้อความซึ่งมีให้เลือก 3 สีด้วยกัน คือ สีแดง สีเขียว และ สีส้ม จากนั้นก็จะทำการส่งข้อมูลที่ได้เลือกให้ไมโครคอนโทรลเลอร์ได้รับรู้ และโปรแกรมจะถามว่าเป็นข้อมูลชุดสุดท้ายหรือไม่ ถ้ายังมีข้อมูลอีกก็จะวนกลับมาสู่การเลือกโหมดเหมือนเช่นในตอนแรก แต่ถ้าเป็นข้อมูลชุดสุดท้าย โปรแกรมก็จะทำการส่งค่าให้ไมโครคอนโทรลเลอร์ได้รับรู้ว่าจบการทำงานแล้ว และให้นำข้อมูลที่เลือกไปแสดงผลที่ Display ต่อไป

บทที่ 5

การใช้งานและผลการทดลอง

5.1 กล่าวนำ

เนื่องจากโครงงานชิ้นนี้เป็นบอร์ดการแสดงผลการทดลองที่ได้ส่วนมากจะไม่ได้ออกมาในปริมาณที่มีหน่วยทางฟิสิกส์หรือสัญญาณทางไฟฟ้า แต่โดยมากจะใช้วัดผลการทดลองจากการดูผลที่ได้ด้วยตามากกว่า เช่นดูความสว่างของ LED ดูการกระพริบของข้อความว่าอยู่ในอัตราที่ขอมรับได้หรือไม่ เป็นต้น ส่วนผลที่ได้จากการทดสอบโปรแกรมก็จะมีแค่เพียงใช้งานได้ตรงตามที่กำหนดหรือไม่เท่านั้น

5.2 วิธีการใช้งาน

5.2.1 ต่อสาย RS 232 จากคอมพิวเตอร์เข้ากับบอร์ด

5.2.2 เปิดโปรแกรมที่จะป้อนข้อมูลลงในไมโครคอนโทรลเลอร์

5.2.3 ป้อนข้อมูลที่ต้องการลงในไมโครคอนโทรลเลอร์

5.2.4 เมื่อข้อมูลถูกเก็บในไมโครคอนโทรลเลอร์แล้ว ถอดสาย RS 232 ออกจากบอร์ด

5.2.5 จ่ายไฟให้กับมอเตอร์แล้วทำการ ปรับความเร็วของมอเตอร์จนกว่าตัวอักษรที่ปรากฏจะนิ่ง ตอนนี้เราจะมองเห็นข้อความที่เราต้องการแล้ว

5.2.6 ถ้าต้องการเปลี่ยนข้อความให้หยุดจ่ายไฟให้กับมอเตอร์แล้วทำตามขั้นตอนที่กล่าวมาอีกครั้ง

5.3 ผลการทดลองทางฮาร์ดแวร์

5.3.1 การทดสอบการความนำไฟฟ้าของแปรงถ่าน ซึ่งหลังจากได้ทำการติดตั้งแปรงถ่านเป็นที่เรียบร้อยแล้วได้ทำการทดลองจ่ายกระแสไฟฟ้าขึ้นไปยังบอร์ด ในขณะที่ยังมีได้จ่ายไฟฟ้าให้มอเตอร์(มอเตอร์ยังไม่หมุน) ใช้โวลต์มิเตอร์วัดค่าความต้านทานได้เท่ากับศูนย์โอห์ม หลังจากนั้นทดลองใช้งานจริงโดยจ่ายไฟฟ้าให้มอเตอร์ ผลปรากฏว่าสามารถส่งสัญญาณไฟฟ้าได้เป็นอย่างดี

5.3.2 การทดสอบการสั่นสะเทือนของโครงส่วนประกอบ ในขณะที่ใช้งาน ผลปรากฏว่ายังมีการสั่นเล็กน้อยในระดับความเร็วที่พอใช้งานได้ แต่ถ้าเพิ่มพิกัดความเร็วของมอเตอร์ขึ้นไปอีกก็จะยังมีการสั่นมากขึ้น และมีเสียงดังจากการหนีสัมผัสของแปรงถ่านบ้าง

5.3.3 การทดสอบทางค่านวจร พบว่าวงจรแต่ละวงจรสามารถใช้งานได้ ตรงตามวัตถุประสงค์และข้อกำหนดที่ตั้งไว้ได้เป็นอย่างดี

5.3.4 การทดสอบการทำงานของอินฟราเรดเซนเซอร์ พบว่ามีความยุ่งยากพอสมควรในเรื่อง การตรวจสอบตำแหน่งของมอเตอร์เพราะช่วงในการรับแสงของอินฟราเรดเซนเซอร์ จะกว้างทำให้ พื้นที่ในการแสดงผลคือน้อยลงเพราะ โปรแกรมต้องหน่วงเวลารอด้วย

5.3.5 การทดสอบความสว่างของLED พบว่ามีความสว่างอยู่ในระดับที่พอมองเห็นได้ชัด แม้อยู่กลางแจ้งก็ตาม

5.4 ผลการทดลองทางด้านซอฟต์แวร์

5.4.1 การทดสอบการทำงานของโปรแกรมการรับค่า และการควบคุมการแสดงผลโดย ไมโครคอนโทรลเลอร์ พบว่ามีปัญหาเกี่ยวกับการเรื่องความแน่นอนพอสมควรจึงมีความจำเป็นต้อง คัดรายละเอียดที่ไม่จำเป็นออกจึงพอทำงานได้ตรงตามที่ได้ออกแบบพอสมควรจะมีผิดพลาดอยู่บ้าง แต่ก็ได้ทำการแก้ไขจนพอที่ใช้งานได้ในระดับหนึ่ง

5.4.2 การทดสอบการทำงานของโปรแกรม การส่งค่าตัวอักษรสู่ไมโครคอนโทรลเลอร์ พบว่าสามารถใช้งานได้ดีพอสมควร แต่ทั้งนี้ทั้งนั้น ก็ขึ้นอยู่กับการทำงานของโปรแกรมการรับค่า ด้วยซึ่งก็คือไมโครคอนโทรลเลอร์นั่นเอง ซึ่งบางครั้งก็บอกไม่ได้แน่ชัด ว่าเกิดความผิดพลาดขึ้นที่ ส่วนไหนกันแน่

บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

6.1 บทสรุป

จากการที่ได้ทำการออกแบบ และทดสอบการทำงานของบอร์ดแสดงผลแบบโปรแกรมได้ ทั้งในส่วนของฮาร์ดแวร์และซอฟต์แวร์ พบว่าในส่วนของฮาร์ดแวร์จะมีส่วนของความซับซ้อนน้อยกว่าส่วนของซอฟต์แวร์อยู่มาก ดังนั้นผลการทดลองที่ได้จากฮาร์ดแวร์ส่วนใหญ่จะได้ตรงตามที่ต้องการ ออกแบบไว้ เช่น การส่งกระแสไฟฟ้าไปสู่บอร์ดควบคุมการแสดงผล ความสว่างของ LED ขณะแสดงผล รวมทั้งวงจรอิเล็กทรอนิกส์ต่าง ๆ ได้แก่ วงจรบอร์ดควบคุมการแสดงผล วงจร RS232 Interface วงจร Infrared Sensor วงจร Power Supply และวงจร Motor Driver ก็ให้ผลการทดลองตรงตามที่ออกแบบ

จะมีปัญหาอยู่บ้างในส่วนของการสั่นสะเทือนของมอเตอร์เนื่องจากโหลดของมอเตอร์ที่ไม่สมดุลและความเร็วรอบของมอเตอร์ แต่ก็แก้ไขโดยการออกแบบบอร์ดควบคุมให้มีความสมดุลกันมากขึ้น พบว่าลดความสั่นสะเทือนลงไปได้

และปัญหาอีกอย่างหนึ่ง คือ ตัว Infrared Sensor ซึ่งต้องพยายามทำให้ช่วงระยะในการตรวจจับสัญญาณแคบที่สุดเท่าที่จะเป็นไปได้ เพราะจะส่งผลถึงพื้นที่ในการแสดงผลของ Display การแก้ปัญหาทำได้โดย หาวัดมุมกันแสงให้เหลือช่องน้อยที่สุด และต้องติดตั้งในแนวตั้งเท่านั้น

ส่วนผลการทดลองทางด้านซอฟต์แวร์ทางด้านบอร์ดควบคุมการแสดงผล เนื่องจากมีความซับซ้อนมาก ทำให้ผลที่ได้จากการทดสอบใช้งานจริงยังไม่ค่อยตรงกับที่ออกแบบไว้ ต้องทำการปรับปรุง เปลี่ยนแปลงโครงสร้างทางด้านซอฟต์แวร์ให้มีความซับซ้อนน้อยลง จึงทำให้ผลที่ได้จากการใช้งานจริงอยู่ในเกณฑ์ที่ใช้งานได้ในระดับหนึ่ง

ส่วนผลการทดลองทางการส่งข้อมูลจากคอมพิวเตอร์ เนื่องจากโครงสร้างโปรแกรมไม่ซับซ้อนมาก จึงไม่มีปัญหาด้านการใช้งานแต่อย่างใด

6.2 ข้อเสนอแนะ

แนวทางในการพัฒนาโครงการชิ้นนี้มีอยู่หลายแนวทางด้วยกัน คือ

1. รูปร่างและน้ำหนักของโครงสร้างควรทำให้เบาและเล็กที่สุดเท่าที่จะทำได้ซึ่งขึ้นอยู่กับมอเตอร์ที่ใช้เป็นหลัก
2. การส่งข้อมูลจากคอมพิวเตอร์ไปยังบอร์ด อาจพัฒนามาใช้สัญญาณอินฟราเรดในรูปของรีโมท เป็นตัวส่งสัญญาณ ทำให้ไม่ต้องผ่านหน้าสัมผัสจากแปรงถ่านซึ่งน่าจะมีความแน่นอนในการส่งข้อมูลได้มากกว่า

3. การเลือกใช้ LED มีให้เลือกอยู่หลายชนิดด้วยกันในท้องตลาด อาจเลือกใช้ LED ชนิดอื่น ๆ จะทำให้เกิดความแปลกใหม่มากขึ้น
4. การเพิ่มรัศมีของการแสดงผลจะทำให้ได้จำนวนตัวอักษรที่เพิ่มขึ้น

บรรณานุกรม

บริษัท ซีเอ็ดยูเคชั่น จำกัด. คู่มือเปรียบเทียบเบอร์ไอซี TTL. กทม : บริษัท ซีเอ็ดยูเคชั่น จำกัด 2543 หน้า 286-287

ศรัณย์ รัตนานันท์ คณิตศาสตร์มอคูเลเตอร์ ปรับความเร็วดีซีมอเตอร์ กทม : เขมิกอน-คักเตอร์อิเล็กทรอนิกส์ 2540 หน้า 212-215

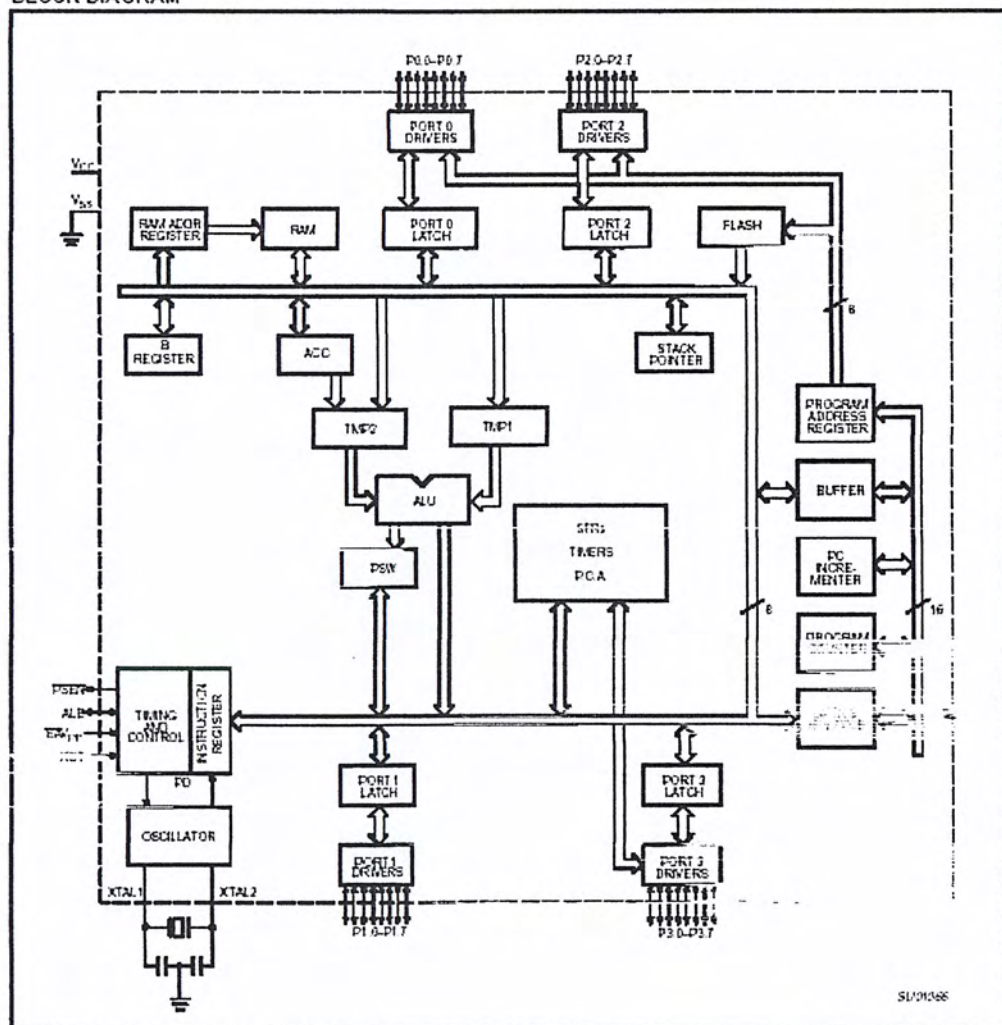
ปรเมษฐ์ ประณายนันท์ และ ปิยพงศ์ เห่าวิช ไมโครคอนโทรลเลอร์ MCS – 51 ซีเอ็ดยูเคชั่น, พ.ศ. 2536

ภาคผนวก

80C51 8-bit Flash microcontroller family
16KB/32KB/64KB ISP/IAP Flash with 512B/512B/1KB RAM

P89C51RB2/P89C51RC2/
P89C51RD2Hxx

BLOCK DIAGRAM



80C51 8-bit Flash microcontroller family
16KB/32KB/64KB ISP/IAP Flash with 512B/512B/1KB RAM

P89C51RB2/P89C51RC2/
P89C51RD2Hxx

When Timer 2 is in the baud rate generator mode, one should not try to read or write TH2 and TL2. As a baud rate generator, Timer 2 is incremented every state time ($t_{osc}/2$) or asynchronously from pin T2; under these conditions, a read or write of TH2 or TL2 may not be accurate. The RCAP2 registers may be read, but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

Table 3 shows commonly used baud rates and how they can be obtained from Timer 2.

Summary of Baud Rate Equations

Timer 2 is in baud rate generating mode. If Timer 2 is being clocked through pin T2(P1.0) the baud rate is:

$$\text{Baud Rate} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

If Timer 2 is being clocked internally the baud rate is:

$$\text{Baud Rate} = \frac{f_{osc}}{[n \times 65536 \pm (RCAP2H, RCAP2L)]}$$

$n = \begin{matrix} 16 & \text{in 8 clock mode} \\ 32 & \text{in 12 clock mode} \end{matrix}$

Where f_{osc} = Oscillator Frequency

To obtain the reload value for RCAP2H and RCAP2L, the above equation can be rewritten as:

$$RCAP2H, RCAP2L = 65536 \pm \left(\frac{f_{osc}}{n \times \text{Baud Rate}} \right)$$

Timer/Counter 2 Set-up

Except for the baud rate generator mode, the values given for T2CON do not include the setting of the TR2 bit. Therefore, bit TR2 must be set separately to turn the timer on; see Table 4 for set-up of Timer 2 as a timer. Also see Table 5 for set-up of Timer 2 as a counter.

Table 4. Timer 2 as a Timer

MODE	T2CON	
	INTERNAL CONTROL (Note 1)	EXTERNAL CONTROL (Note 2)
16-bit Auto-Reload	00H	00H
16-bit Capture	01H	09H
Baud rate generator receive and transmit same baud rate	30H	36H
Receive only	20H	24H
Transmit only	40H	46H

Table 5. Timer 2 as a Counter

MODE	T2CON	
	INTERNAL CONTROL (Note 1)	EXTERNAL CONTROL (Note 2)
16-bit	02H	0AH
Auto-Reload	03H	0BH

NOTES:

1. Capture/reload occurs only on timer/counter overflow.
2. Capture/reload occurs on timer/counter overflow and a 1-to-0 transition on T2EX (P1.1) pin except when Timer 2 is used in the baud rate generator mode.

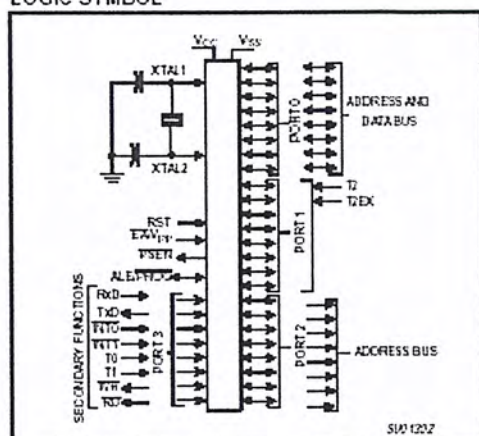
80C51 8-bit Flash microcontroller family

16KB/32KB/64KB ISP/IAP Flash with 512B/512B/1KB RAM

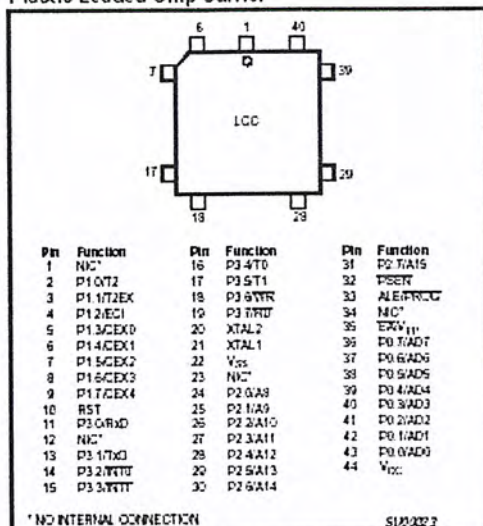
P89C51RB2/P89C51RC2/

P89C51RD2/HXX

LOGIC SYMBOL

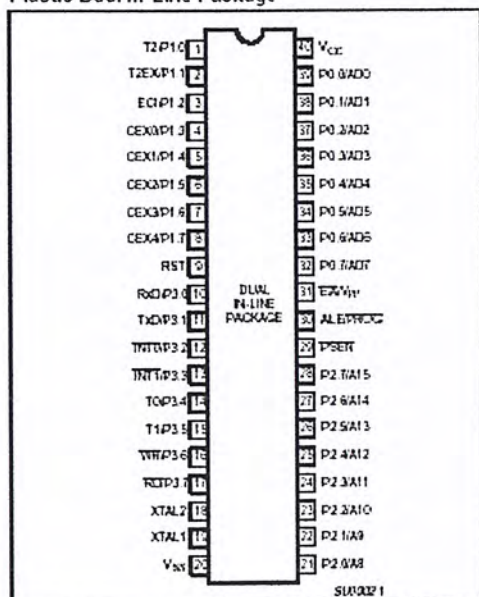


Plastic Leaded Chip Carrier

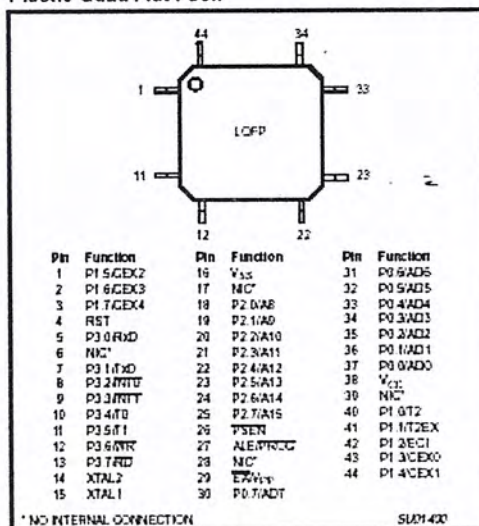


PINNING

Plastic Dual In-Line Package



Plastic Quad Flat Pack



80C51 8-bit Flash microcontroller family
16KB/32KB/64KB ISP/IAP Flash with 512B/512B/1KB RAM

P89C51RB2/P89C51RC2/
P89C51RD2Hxx

PIN DESCRIPTIONS

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
	PDIP	PLCC	LQFP		
V _{SS}	20	22	16	I	Ground; 0 V reference.
V _{CC}	40	44	38	I	Power Supply: This is the power supply voltage for normal, idle, and power-down operation.
P0.0-P0.7	39-32	43-36	37-30	I/O	Port 0: Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s.
P1.0-P1.7	1-8	2-9	40-44, 1-3	I/O	Port 1: Port 1 is an 8-bit bidirectional I/O port with internal pull-ups on all pins. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I _{IL}).
	1	2	40	I/O	Alternate functions for P89C51RB2/RC2/RD2Hxx Port 1 include: T2 (P1.0): Timer/Counter 2 external count input/clockout (see Programmable Clock-Out) T2EX (P1.1): Timer/Counter 2 Reload/Capture/Direction Control EC1 (P1.2): External Clock Input to the PCA CEX0 (P1.3): Capture/Compare External I/O for PCA module 0 CEX1 (P1.4): Capture/Compare External I/O for PCA module 1 CEX2 (P1.5): Capture/Compare External I/O for PCA module 2 CEX3 (P1.6): Capture/Compare External I/O for PCA module 3 CEX4 (P1.7): Capture/Compare External I/O for PCA module 4
P2.0-P2.7	21-28	24-31	18-25	I/O	Port 2: Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I _{IL}). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register. P2.7 must be a "1" to program and erase the device.
P3.0-P3.7	10-17	11, 13-19	5, 7-13	I/O	Port 3: Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I _{IL}). Port 3 also serves the special features of the P89C51RB2/RC2/RD2Hxx, as listed below: Rx/D (P3.0): Serial input port Tx/D (P3.1): Serial output port INT0 (P3.2): External interrupt INT1 (P3.3): External interrupt T0 (P3.4): Timer 0 external input T1 (P3.5): Timer 1 external input WR (P3.6): External data memory write strobe RD (P3.7): External data memory read strobe
RST	9	10	4	I	Reset: A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal resistor to V _{SS} permits a power-on reset using only an external capacitor to V _{CC} .
ALE	30	33	27	O	Address Latch Enable: Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted twice every machine cycle, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. ALE can be disabled by setting SFR auxiliary.0. With this bit set, ALE will be active only during a MOVX instruction.

80C51 8-bit Flash microcontroller family
16KB/32KB/64KB ISP/IAP Flash with 512B/512B/1KB RAM

P89C51RB2/P89C51RC2/
P89C51RD2Hxx

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
	PDIP	PLCC	LOFP		
PSEN	29	32	26	0	Program Store Enable: The read strobe to external program memory. When executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.
EA/V _{PP}	31	35	29	1	External Access Enable/Programming Supply Voltage; EA must be externally held low to enable the device to fetch code from external program memory locations. If EA is held high, the device executes from internal program memory. The value on the EA pin is latched when RST is released and any subsequent changes have no effect. This pin also receives the programming supply voltage (V _{PP}) during Flash programming.
XTAL1	19	21	15	1	Crystal 1: Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	14	0	Crystal 2: Output from the inverting oscillator amplifier.

NOTE:

To avoid "latch-up" effect at power-on, the voltage on any pin (other than V_{PP}) must not be higher than V_{CC} + 0.5 V or less than V_{SS} - 0.5 V.

80C51 8-bit Flash microcontroller family
16KB/32KB/64KB ISP/IAP Flash with 512B/512B/1KB RAM

P89C51RB2/P89C51RC2/
P89C51RD2Hxx

SPECIAL FUNCTION REGISTERS

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB				LSB				
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
AUXR*	Auxiliary	BEH	-	-	-	-	-	-	EXTRAM	A0	xxxxxx00B
AUXR1*	Auxiliary 1	A2H	-	-	EXBOOT	-	GF2	0	-	DPS	xxxxxx00B
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
CCAP0H*	Module 0 Capture High	FAH									xxxxxxxxB
CCAP1H*	Module 1 Capture High	FBH									xxxxxxxxB
CCAP2H*	Module 2 Capture High	FCH									xxxxxxxxB
CCAP3H*	Module 3 Capture High	FDH									xxxxxxxxB
CCAP4H*	Module 4 Capture High	FEH									xxxxxxxxB
CCAP0L*	Module 0 Capture Low	EAH									xxxxxxxxB
CCAP1L*	Module 1 Capture Low	EBH									xxxxxxxxB
CCAP2L*	Module 2 Capture Low	ECH									xxxxxxxxB
CCAP3L*	Module 3 Capture Low	EDH									xxxxxxxxB
CCAP4L*	Module 4 Capture Low	EEH									xxxxxxxxB
CCAPM0*	Module 0 Mode	DAH	-	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	xxxx0000B
CCAPM1*	Module 1 Mode	DBH	-	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	xxxx0000B
CCAPM2*	Module 2 Mode	DCH	-	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	xxxx0000B
CCAPM3*	Module 3 Mode	DDH	-	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	xxxx0000B
CCAPM4*	Module 4 Mode	DEH	-	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	xxxx0000B
CCON*	PCA Counter Control	D8H	DF	DE	DD	DC	DB	DA	D9	D8	00x00000B
	CH* PCA Counter High	F9H	CF	CR	-	CCF4	CCF3	CCF2	CCF1	CCF0	00H
	CL* PCA Counter Low	E9H									00H
CMOD*	PCA Counter Mode	D9H	CIDL	WDTE	-	-	-	CPS1	CPS0	ECF	00xxxx00B
DPTR:	Data Pointer (2 bytes)										
DPH	Data Pointer High	83H									00H
DPL	Data Pointer Low	82H									00H
IE*	Interrupt Enable 0	A8H	AF	AE	AD	AC	AB	AA	A9	A8	00H
			EA	EC	ET2	ES	ET1	EX1	ET0	EX0	
			BF	BE	BD	BC	BB	BA	B9	B8	
IP*	Interrupt Priority	B8H	-	PPC	PT2	PS	PT1	PX1	PT0	PX0	xxxx0000B
			B7	B6	B5	B4	B3	B2	B1	B0	
IPH*	Interrupt Priority High	B7H	-	PPCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	xxxx0000B
			87	86	85	84	83	82	81	80	
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
			97	96	95	94	93	92	91	90	
P1*	Port 1	90H	CEX4	CEX3	CEX2	CEX1	CEX0	ECI	T2EX	T2	FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port 2	A0H	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port 3	B0H	RD	WR	T1	T0	INTT	INT0	TxD	RxD	FFH
PCON*	Power Control	87H	SMOD1	SMOD0	-	POF	GF1	GF0	PD	IDL	00xxxx00B

* SFRs are bit addressable.

SFRs are modified from or added to the 80C51 SFRs.

- Reserved bits.

1. Reset value depends on reset source.

80C51 8-bit Flash microcontroller family
16KB/32KB/64KB ISP/IAP Flash with 512B/512B/1KB RAM

P89C51RB2/P89C51RC2/
P89C51RD2Hxx

SPECIAL FUNCTION REGISTERS (CONTINUED)

SPECIAL FUNCTION REGISTERS (CONTINUED)												
SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE	
			MSB				LSB					
PSW*	Program Status Word	D0H	D7	D6	D5	D4	D3	D2	D1	D0	00000000B	
	RCAP2H*	CBH	CY	AC	F0	RS1	RS0	OV	F1	P		00H
	RCAP2L*	CAH										00H
SADDR*	Slave Address	A9H									00H	
SADEN*	Slave Address Mask	B9H									00H	
SBUF	Serial Data Buffer	99H									xxxxxxxxB	
SCON*	Serial Control	98H	SM0FE	SM1	SM2	REN	TB8	RB8	TI	RI	00H	
	Stack Pointer	81H									07H	
TCON*	Timer Control	88H	8F	8E	8D	8C	8B	8A	89	88	00H	
			TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0		
			CF	CE	CD	CC	CB	CA	C9	C8		
T2CON*	Timer 2 Control	C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00H	
T2MOD*	Timer 2 Mode Control	C9H	-	-	-	-	-	-	T2OE	DCEN	xxxxxxxxB	
TH0	Timer High 0	8CH									00H	
TH1	Timer High 1	8DH									00H	
TH2*	Timer High 2	CDH									00H	
TL0	Timer Low 0	8AH									00H	
TL1	Timer Low 1	8BH									00H	
TL2*	Timer Low 2	CCH									00H	
TMOD	Timer Mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H	
WDTRST	Watchdog Timer Reset	A6H										

* SFRs are bit addressable.

SFRs are modified from or added to the 80C51 SFRs.

- Reserved bits.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. Minimum and maximum high and low times specified in the data sheet must be observed.

This device is configured at the factory to operate using 6 clock periods per machine cycle, referred to in this datasheet as "6 clock mode". (This yields performance equivalent to twice that of standard 80C51 family devices). It may be optionally configured on commercially-available EPROM programming equipment to operate at 12 clocks per machine cycle, referred to in this datasheet as "12 clock mode". Once 12 clock mode has been configured, it cannot be changed back to 6 clock mode.

RESET

A reset is accomplished by holding the RST pin high for at least two machine cycles (12 oscillator periods in 6 clock mode, or 24 oscillator periods in 12 clock mode), while the oscillator is running. To ensure a good power-on reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-on, the voltage on V_{CC} and RST must come up at the same time for a proper start-up. Ports 1, 2, and 3 will asynchronously be driven to their reset condition when a voltage above V_{RH1} (min.) is applied to RESET.

The value on the EA pin is latched when RST is deasserted and has no further effect.