



การประยุกต์ใช้งานไมโครคอนโทรลเลอร์สำหรับควบคุมลิฟต์
APPLICATION OF MICROCONTROLLER FOR CONTROL LIFT



โดย
นาย ชลากรณ์ ปัญญาโฉม
นาย นรา จิรายุวัฒนา
นางสาว อพินญา จินตคามณี

วัน เดือน ปี 1 ต.ค. 25๕๐
เลขทะเบียน ๐๓๗๑๕๐
เลขเรียกหนังสือ T ๐๓๘๑๔๓ ๖๓๗๖

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมบัณฑิต
สาขาวิศวกรรมการวัดคุมทางอุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2538

ปริญญาโทปีการศึกษา 2538

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

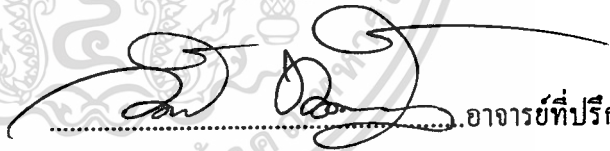
เรื่อง การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ สำหรับควบคุมลิฟต์

ผู้จัดทำ

นาย ชลากรณ์ ปัญญา โฉม 35104096

นาย นรา จิรายุวัฒนา 35104206

นางสาว อพิญญา จินคามณี 35104528


.....อาจารย์ที่ปรึกษา
(ผศ.วิทยา ทิพย์สุวรรณพร)

การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ สำหรับควบคุมลิฟท์

ชลากรณ์ ปัญญาโฉม

นรา จิรายุวัฒนา

อพิญญา จินคามณี

บทคัดย่อ

ในวิทยานิพนธ์ฉบับนี้ ได้มีการศึกษาถึงการสร้างแบบจำลองลิฟต์โดยสารและศึกษาถึงวิธีการเขียนโปรแกรมบนไมโครคอนโทรลเลอร์เพื่อใช้ในการควบคุมลิฟต์ โดยไมโครคอนโทรลเลอร์ จะทำการรับสัญญาณข้อมูลที่ได้จากการกดสวิชต์แต่ละครั้งรวมทั้งรับสัญญาณจากเซนเซอร์ด้วย แล้วใช้โปรแกรมนี้อในการวิเคราะห์เพื่อจัดลำดับ (Priority) ในการจัดการเคลื่อนที่ ขึ้น-ลง และหยุดของลิฟต์ เพื่อที่จะเป็นพื้นฐานในการจัดการกับระบบลิฟต์ได้อย่างมีประสิทธิภาพ

Abstract

This thesis present built model of lift and investigation a programing for system control elevator on microcontorller. Microcontroller get data which receive from push switch and sensor. We apply this program to analysis for piorty to move up-down and stop elevator. This is basic in an effectian management system control elevator.

สารบัญ

หัวข้อเรื่อง	หน้า
บทนำ	1-7
บทที่ 1 ทฤษฎีทั่วไป	8-18
- ไมโครคอนโทรลเลอร์ MCS -51	
บทที่ 2 ทฤษฎีเกี่ยวกับการสร้าง	19 - 34
- การสร้างแบบจำลองลิฟต์	
- การประยุกต์ใช้ ANT- 32	
- วงจรขับเคลื่อนมอเตอร์	
- การเชื่อมต่อเซนเซอร์	
บทที่ 3 การทดลองและผลการทดลอง	37 - 45
บทที่ 4 บทวิจารณ์และบทสรุป	46
ภาคผนวก	

สารบัญรูป

รูปที่ A แสดงวิธีแขวนแบบต่าง ๆ	หน้าที่ 3
รูปที่ B แสดงโครงสร้างและส่วนประกอบของลิฟต์	" 5
รูปที่ 1.2 แผนผัง โครงสร้างของไอซี 8255	" 16
รูปที่ 1.3 แผนผังวงจรภายในและการจัดขาของ ไอซี 8255	" 16
รูปที่ 1.4 ความหมายของบิตต่าง ๆ ในรหัสควบคุม	" 18
รูปที่ 2.1 แสดง Layout การติดรางอลูมิเนียมในบ่อลิฟต์และการติดตัวถ่วงน้ำหนัก	" 22
รูปที่ 2.1.1 แสดงแบบจำลองที่ สร้างเสร็จแล้ว	" 22
รูปที่ 2.2 แสดงการติดตั้งเซนเซอร์	" 23
รูปที่ 1 แสดงลักษณะ โครงสร้างแบบจำลองทั้งหมด	" 24
รูปที่ 2 แสดงลักษณะของลิฟต์ที่ประกอบเสร็จแล้ว	" 25
รูปที่ 3 แสดงส่วนของคอนโทรลลิฟต์ทั้งหมด	" 26
รูปที่ 4 แสดงกล่องเล็กที่ภายในบรรจุโปรแกรมควบคุมทั้งหมด	" 27
รูปที่ 2.3 แสดงบอร์ด และตำแหน่งจัมป์เปอร์	" 29
รูปที่ 2.4 แสดง Relay 2 Contact	" 32
รูปที่ 2.5 แสดงวงจร Drive Motor	" 33
รูปที่ 2.6 แสดงวงจร Senser Circuit	" 34
รูปที่ 2.7 แสดง Flow Chart การจัด Priority	" 35
รูปที่ 3.1 แสดง โครงสร้างลิฟต์จำลองและวงจรควบคุม	" 38
รูปที่ 3.2 แสดงการเรียกใช้ลิฟต์ไปยังชั้นที่ต้องการ	" 39
รูปที่ 3.3 แสดงการเคลื่อนที่ของลิฟต์ไปยังชั้นที่ต้องการ	" 40
รูปที่ 3.4 แสดงการเคลื่อนที่ขึ้น - ลง ของลิฟต์พร้อมกันทั้ง 2 ตัว	" 41

รูปที่ 3.5 แสดงการเรียกลิปต์ที่ชั้นบนสุดเพื่อต้องการลง	" 42
รูปที่ 3.6 แสดงการเคลื่อนที่ของลิปต์มายังชั้นที่เรียก	" 43
รูปที่ 3.7 แสดงการเรียกใช้ลิปต์ขณะอีกตัวกำลังเคลื่อนที่	" 44
รูปที่ 3.8 แสดงการเคลื่อนที่ของลิปต์ตามคำสั่งเรียกในรูปที่ 3.7	" 45



บทนำ

ลิฟต์

ลิฟต์และคำนิยาม

ความหมายของลิฟต์ (Lift) หรือตัวยก คืออุปกรณ์ที่ใช้ในการยกขึ้น-ลงแบบถาวรระหว่างพื้นที่ต่างระดับ หรือมากกว่านั้น และรวมหมายถึงเป็นพาหนะสำหรับขนส่งผู้โดยสาร และหรือสิ่งของอื่น ๆ โดยจะเคลื่อนที่อยู่ระหว่างรางที่ถูกยึดแน่นอยู่กับที่และมีรถเป็นตัวชักขึ้น-ลงในลักษณะเป็นแนวตั้งฉากกับพื้น หรือในแนวลาดเอียงทำมุมจากแนวตั้งฉากน้อยกว่า 15 องศา ชนิดของลิฟต์ที่แบ่งออกได้ตามคุณลักษณะที่ใช้ในการขับเคลื่อนลิฟต์มีดังนี้ คือ

1. ลิฟต์เคลื่อนที่ได้โดยไฟฟ้า (Electric Elevators)
2. ลิฟต์ที่เคลื่อนที่ได้โดยใช้กำลังขับของเหลว (Hydraulic Elevator) เช่น น้ำมัน
3. ลิฟต์ที่เคลื่อนที่ได้โดยใช้กำลังขับของลม (Pneumatic Elevator)

เนื่องจากลิฟต์ที่มีใช้อยู่ในปัจจุบันนี้ ส่วนใหญ่เป็นลิฟต์ที่เคลื่อนที่ได้โดยใช้ไฟฟ้า ดังนั้นต่อไปนี้จะขอกล่าวถึงเฉพาะเรื่องของลิฟต์ที่เคลื่อนที่ได้โดยใช้ไฟฟ้าเท่านั้น

ลิฟต์ที่เคลื่อนที่ได้โดยใช้ไฟฟ้า อาจแบ่งออกเป็น

ก.) การขับโดยใช้กำลังลากของเครื่องจักร (Traction Drive) โดยใช้เชือกแขวนไว้ ซึ่งเป็นการขับเคลื่อนที่อาศัยแรงเสียดทานในร่องของการขับเคลื่อนล้อในลูกกรงของเครื่องจักร

ข.) การขับเคลื่อนในทางบวก (Position Drive) ซึ่งเป็นการแขวนด้วยโซ่หรือเชือก และอาศัยการขับเคลื่อนด้วยวิธีอื่นมากกว่าอาศัยแรงเสียดทาน (การขับแบบดรัม, Drum Drive)

มาตรฐานของอังกฤษ BS 5655-PART 5:1987 ได้แบ่งลิฟต์ที่เคลื่อนที่ได้โดยใช้ไฟฟ้าแบบการขับ โดยใช้กำลังลากของเครื่องจักรออกเป็น 7 ชนิด โดยแบ่งตามคุณสมบัติความสัมพันธ์ภายนอกระหว่างขนาดกับภาระ (กิโลกรัม), ความเร็วในการเคลื่อนที่ (เมตร/วินาที) และการติดตั้ง

ตัวแปรหลักที่สำคัญทางเทคนิคของลิฟต์คือ พิกัด ภาระ (Rate Load) มีหน่วยเป็นกิโลกรัม และพิกัดความเร็ว (Rate Speed) ในการเคลื่อนที่ มีหน่วยเป็น เมตร/วินาที ซึ่งส่วนประกอบต่าง ๆ

จะถูกสร้างขึ้นจากความสัมพันธ์ของตัวแปรเหล่านี้ที่ภาวะการใช้งานปกติ และจะต้องได้รับการรับรองจากโรงงานผู้ผลิต

น้ำหนักของผู้โดยสารแต่ละคนที่ใช้ในการคำนวณ จะใช้ค่าเฉลี่ยประมาณ 75 กิโลกรัม /คน นอกจากนี้ยังมีคุณลักษณะของตัวแปรอื่น ๆ ที่จะต้องพิจารณาร่วมด้วยคือ

ก. ความสูงในการเคลื่อนที่ (ระยะทางที่ห้องผู้โดยสารเคลื่อนที่ขึ้น), จำนวนการหยุด และ ตำแหน่งที่หยุด

ข. ขนาดของช่องทาง (Elevator Well), ขนาดของห้องผู้โดยสาร (Car) และขนาดของห้อง เครื่องจักร

ค. แรงดันไฟฟ้าของแหล่งจ่าย, จำนวนของการเริ่มต้นสตาร์ทต่อชั่วโมง และค่าอัตราส่วน จำนวนการใช้ลิฟต์สูงสุด ต่อจำนวนการใช้ทั้งหมด ในช่วงระยะเวลาหนึ่ง (Load Factor)

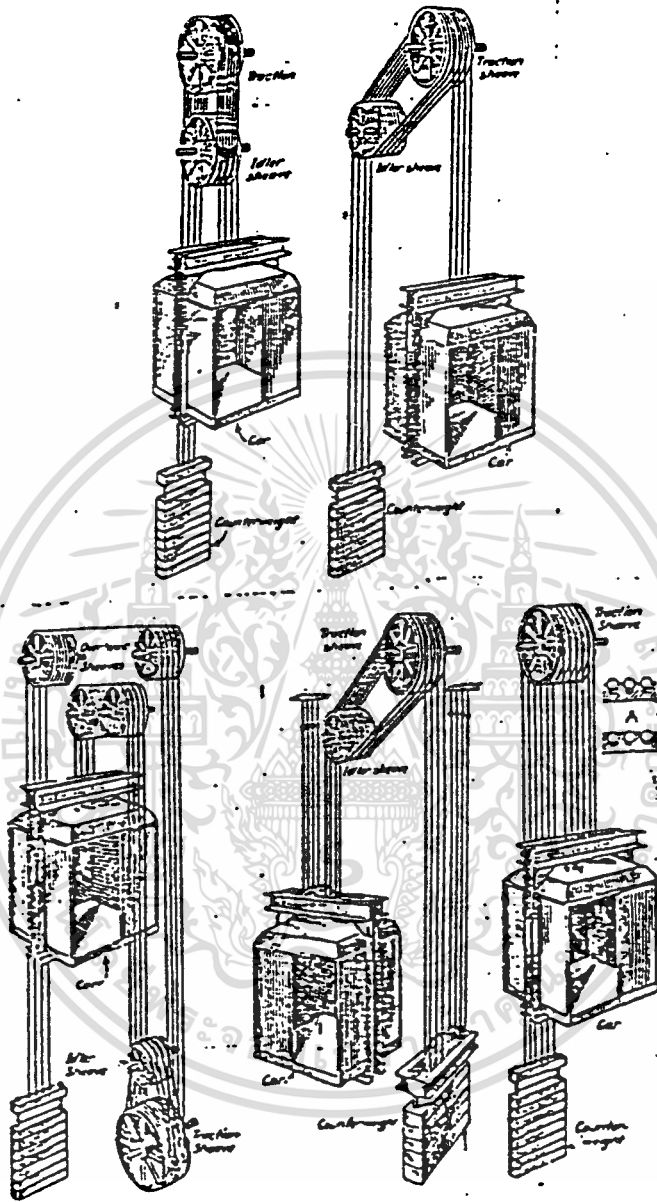
ง. ระบบควบคุม

จ. บริเวณพื้นที่ และการจัดวางประตูห้องผู้โดยสาร และชนิดของตัวทำงาน (Type of Operator)

ฉ. จำนวนของลิฟต์และตำแหน่งในการสร้าง

ช. สภาพแวดล้อม





รูปที่ A แสดงวิธีแขวนแบบต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบและลักษณะโครงสร้างของลิฟต์

ลิฟต์ที่ติดตั้งสมบูรณ์สามารถใช้งานได้ ในภาวะปกติจะต้องประกอบด้วยอุปกรณ์หลาย ๆ ส่วนซึ่งรวมกัน เรียกว่า ระบบลิฟต์ ซึ่งจะกล่าวถึงส่วนประกอบหลัก ๆ ที่มีความสำคัญดังนี้

- ก. วิธีของการแขวน (Suspension Means) ห้องผู้โดยสาร และตัวถ่วงน้ำหนัก (Counterweight) ซึ่งมีทั้งการใช้เชือกเหล็ก และใช้โซ่แขวน
- ข. เครื่องจักรที่ใช้ขับเคลื่อนลิฟต์ (Driving Machine)
1. มอเตอร์ไฟฟ้า (Electric Motor)
 2. เฟืองจักรกล (Mechanical Gearing)
 3. เบรค (Brake)
 4. ลูกกรอก, กว้าน หรือ ล้อเฟืองที่ใช้กับโซ่ (Sheav, Drum or Chain Sprockets)
 5. งานต่อเหลา, เพลา (Shafts), เพลาตอนที่วางอยู่บนฝาประทับเพลา (Joumalls) และลูกปืน (Bearing)

6. โครงของเครื่องจักร (Machine Frame)

ค. ห้องผู้โดยสาร ซึ่งเป็นพาหนะสำหรับผู้โดยสารและ/หรือภาระอื่น ๆ ซึ่งประกอบด้วย เชือก (Sling), โครงเหล็กที่ต่อกับเชือกแขวน, แท่นรองรับผู้โดยสาร

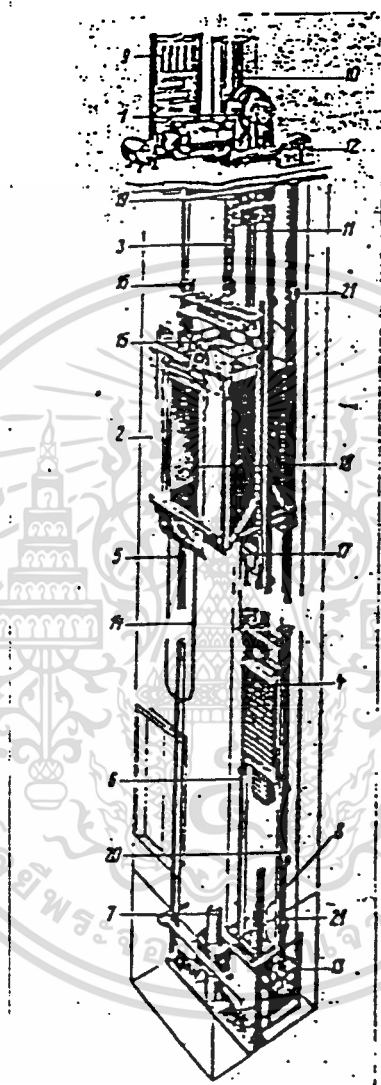
ซึ่งมีอุปกรณ์อื่น ๆ ประกอบอยู่ด้วยคือ

- ฟันเฟืองแบบแขวน
- รางนำทางสำหรับการเคลื่อนที่ของห้องผู้โดยสาร
- เฟืองนิรภัย
- ประตูห้องผู้โดยสารและตัวเปิด-ปิดประตู

ง. ตัวถ่วงน้ำหนักเพื่อให้ได้ความสมดุลของน้ำหนักที่สภาวะเต็มพิกัด และเป็นการลดกำลังขับของมอเตอร์จึงได้มีการเพิ่มอุปกรณ์ช่วงถ่วงน้ำหนักให้กับลิฟต์

จ. ช่องทางการเคลื่อนที่ของลิฟต์ คือช่องว่างที่มีระยะตั้งแต่พื้นล่างจนกระทั่งถึงหลังคา ที่มีขนาดพื้นที่เพียงพอสำหรับการเคลื่อนที่ของห้องผู้โดยสาร และตัวถ่วงน้ำหนัก รวมทั้งพื้นที่ในการจัดวางอุปกรณ์ที่ช่วยในการเคลื่อนที่ต่าง ๆ เช่น รางนำทาง สายเคเบิล ที่อยู่ในช่องว่างข้างล่าง

นอกเหนือจากส่วนประกอบที่กล่าวมาแล้วนั้น ระบบลิฟต์ยังประกอบด้วยส่วนอื่น ๆ อีกหลายส่วน
 ดังแสดงในรูปที่ B



รูปที่ B แสดงโครงสร้างและส่วนประกอบของลิฟต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เครื่องจักรที่ใช้ขับเคลื่อนลิฟต์
2. ห้องผู้โดยสาร
3. เส้นเชือกแขวน
4. ตัวถ่วงน้ำหนัก
5. รางนำทางของการเคลื่อนที่ของห้องผู้โดยสาร
6. รางนำทางของการเคลื่อนที่ของตัวถ่วงน้ำหนัก
7. เครื่องรับน้ำหนักห้องผู้โดยสาร
8. เครื่องรับน้ำหนักของตัวถ่วงน้ำหนัก
9. ตัวควบคุมการทำงานของระบบลิฟต์
10. ตัวเลือก
11. แถบการจับตัวเลือก
12. เครื่องควบคุมความเร็วเกิน
13. ตัวควบคุมความคิงของมู่เล่
14. สายเคเบิลที่เคลื่อนที่ขึ้น-ลงตามห้องผู้โดยสาร
15. ตัวเปิด-ปิดประตู
16. ลูกกลิ้งนำทาง
17. เฟืองปลอดภัยสำหรับห้องผู้โดยสาร
18. ระบบความนิรภัยของประตูสำหรับห้องผู้โดยสาร
19. มู่เล่เบียง
20. ลิมิตสวิทช์นิรภัย (Safety Limit Switch)
21. ลิมิตสวิทช์สุดท้าย (Final Limit Switch)

การทำงานของระบบลิฟต์ในภาวะปกติ

การเคลื่อนที่ของลิฟต์จะเคลื่อนที่ตามคำสั่งของวงจรควบคุม โดยเมื่อมีการกดปุ่มเรียกลิฟต์ ระบบวงจรควบคุมจะสั่งให้มีการปลดเบรกที่มอเตอร์ออก โดยป้อนไฟกระแสตรงเข้าที่กัมเบรก และมอเตอร์เหนี่ยวนำ (Induction Motor) ก็จะทำการขับเคลื่อนลิฟต์ให้เคลื่อนที่ขึ้น-ลง ตามคำสั่งของวงจรควบคุม เมื่อถึงชั้นที่ถูกระบุ ลิฟต์จะหยุด และเปิดประตูโดยการทำงานของตัวเปิด-ปิดของประตู

โดยหลักการทำงานทางกล (Mechanical) ของลิฟต์แม้จะอาศัยตัวถ่วงน้ำหนักถ่วงไว้ข้างหนึ่ง และอีกข้างหนึ่งเป็นห้องผู้โดยสาร เพื่อช่วยลดกำลังขับของมอเตอร์โดยที่รอกเดี่ยว และเส้นเชือก (Rope) เป็นตัวพ่วงห้องผู้โดยสารกับตัวถ่วงน้ำหนักได้โดยน้ำหนักของตัวห้องผู้โดยสาร



บทที่ 1

ทฤษฎีทั่วไป

1.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51

1.1.1 โครงสร้างและพื้นฐาน

ชิพไมโครคอนโทรลเลอร์ อาจแบ่งได้ออกเป็น 2 ประเภท คือ ไมโครโปรเซสเซอร์ และไมโครคอนโทรลเลอร์ ซึ่งไมโครโปรเซสเซอร์จะประกอบด้วยหน่วยควบคุม (Control Unit) และหน่วยคำนวณทางคณิตศาสตร์ และลอจิก (ALU) ส่วนไมโครคอนโทรลเลอร์จะประกอบด้วยหน่วยควบคุม, หน่วยควบคุมทางคณิตศาสตร์และลอจิก, หน่วยความจำ (memory), อินพุต/เอาต์พุตพอร์ต (I/O Port) ซึ่งอาจกล่าวได้ว่า ไมโครโปรเซสเซอร์ก็คือส่วนหนึ่งของไมโครคอนโทรลเลอร์นั่นเอง แต่เมื่อเปรียบเทียบกันแล้ว หน่วยการทำงานภายใน ไมโครคอนโทรลเลอร์นั้นสามารถใช้งานได้อย่างค่อนข้างจำกัดว่า

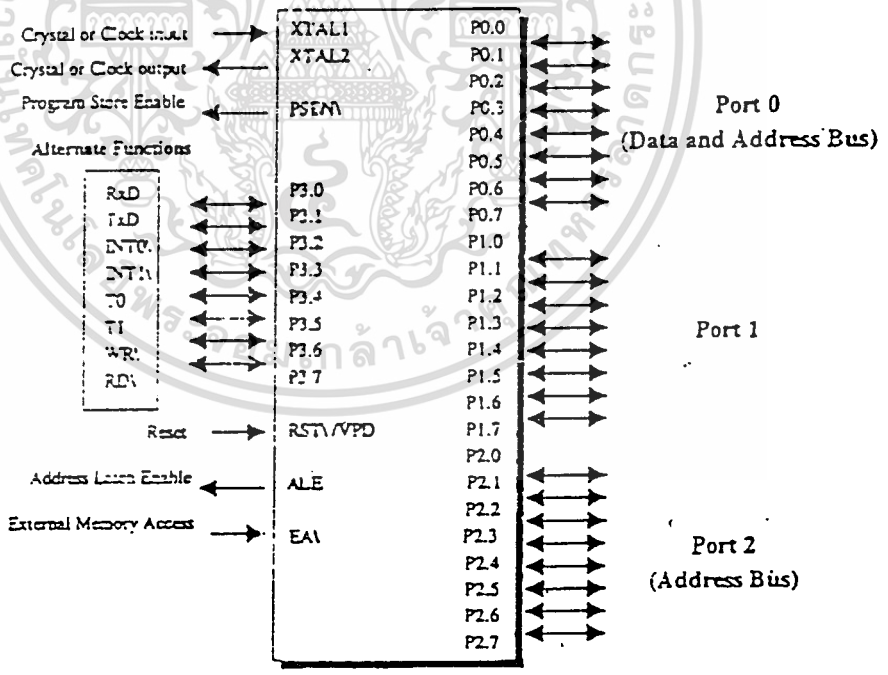
ลักษณะงานที่เหมาะสมกับการนำไมโครคอนโทรลเลอร์ไปใช้งาน มักเป็นงานประยุกต์ที่เกี่ยวข้องกับการควบคุม หรือการจัดการสัญญาณอินพุต/เอาต์พุตของวงจรดิจิทัล และวงจรอิเล็กทรอนิกส์ต่าง ๆ

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ประกอบด้วย ไมโครคอนโทรลเลอร์หลายรุ่น ซึ่งมีสถาปัตยกรรมพื้นฐานที่เหมือนกัน เพียงแต่มีขนาด หรือจำนวน หน่วยทำงาน-ภายในที่แตกต่างกันออกไปดังแสดงให้เห็นในตารางที่ 1.1

ตารางที่ 1.1 ตารางแสดงไมโครคอนโทรลเลอร์ตระกูล MCS -51 ของบริษัทอินเทล

Features	8051AH	8051BH	8751H	80C51BH	80C51BH	87C51	8052AH	8032AH	8752	8044H
Program Memory (Bytes)	4K	-	4K	4K	-	4K	8K	-	8K	4K
RAM Memory (Bytes)	128	128	128	128	128	128	256	256	256	192
Program Memory Expansion (On Chip) (Bytes)	64K	64K	64K	64K	64K	64K	64K	64K	64K	64K
Data Memory Expansion (On Chip) (Bytes)	64K	64K	64K	64K	64K	64K	64K	64K	64K	64K
Max Clock Frequency (MHz)	12	12	12	16	16	16	16	12	12	12
Typical Instruction Time (ns)	1	1	1	0.75	0.75	0.75	1	1	1	1
16-Bit Timer / Counters	2	2	2	2	2	2	3	3	3	2
Serial Communications	Synchronous Mode, Asynchronous Mode, 9 or 10-Bit Programmable									HDLC/SOLC
No. of I/O Lines	32	16	32	32	16	32	32	16	32	32
Interrupt Sources (Time Priority Levels)	5	5	5	5	5	5	6	6	6	5
Power Requirements (Vcc)	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
ICC Max. (mA)	-	-	-	10 mA	10 mA	10 mA	-	-	-	-
Programmable Power Modes (Idle Power Down)	-	-	-	20 uA	31 uA	52 uA	-	-	-	50 uA

ปกติไมโครคอนโทรลเลอร์ตระกูลนี้มักมีรูปร่างของไอซีเป็นแบบ DIP ขนาด 40 ขา ดังรูปที่ 1.1



รูปที่ 1.1 การกำหนดหน้าที่ขาสัญญาณของไอซี 8051

1.1.2 หน่วยความจำโปรแกรม MCS-51

หน่วยความจำโปรแกรมเป็นบริเวณหน่วยความจำสำหรับเก็บข้อมูล และคำสั่งใช้งานต่าง ๆ ซึ่งแม้ว่าจะไม่มีการจ่ายกระแสไฟฟ้าให้กับระบบ ข้อมูลเหล่านี้ก็ยังคงอยู่ไม่สูญหาย โครงสร้างของหน่วยความจำโปรแกรมมีลักษณะเช่นเดียวกับหน่วยความจำที่บรรจุอยู่ในไอซีหน่วยความจำประเภทต่าง ๆ เช่น หน่วยความจำแบบ ROM (Read Only Memory) หรือ EPROM (Erasable Programmable Read Only Memory) เป็นต้น

MCS-51 สามารถอ่านข้อมูลหน่วยความจำโปรแกรมนี้ได้สูงสุดไม่เกิน 64 กิโลไบต์ และแยกประเภทของหน่วยความจำโปรแกรมเป็น 2 ลักษณะ ตามตำแหน่งของหน่วยความจำนั้น คือ หน่วยความจำโปรแกรมภายใน (ซึ่งเป็นหน่วยความจำ ROM หรือ EPROM) ที่อยู่ภายในตัวไอซีไมโครคอนโทรลเลอร์เอง และหน่วยความจำโปรแกรมภายนอก ซึ่งเป็นการใช้ไอซีหน่วยความจำมาทำหน้าที่เป็นหน่วยความจำโปรแกรมของระบบ

1. หน่วยความจำโปรแกรมภายใน

ไมโครคอนโทรลเลอร์เบอร์ต่าง ๆ ที่อยู่ตระกูล MCS-51 นี้มีขนาดหน่วยความจำโปรแกรมภายในแตกต่างกันออกไป เพื่อความเหมาะสมกับการนำไปใช้งานในลักษณะต่าง ๆ กัน

2. หน่วยความจำโปรแกรมภายนอก

เป็นการใช้หน่วยความจำ EPROM (หรือ ROM) เชื่อมต่อเข้ากับระบบของ MCS-51 ซึ่งอาจเกิดจากการที่หน่วยความจำภายในมีขนาดความจุไม่เพียงพอกับการเก็บโปรแกรม หรือเกิดจากการที่ตัวไมโครคอนโทรลเลอร์เองไม่มีหน่วยความจำภายใน (เช่น 8031 และ 8032) ไมโครคอนโทรลเลอร์เบอร์ต่าง ๆ ของตระกูล MCS-51 นี้ สามารถขยายให้ใช้งานหน่วยความจำภายนอกได้ทั้งสิ้น ในกรณีที่มิมีหน่วยความจำภายในอยู่แล้ว การอ้างตำแหน่งแอดเดรสที่มีทั้งหน่วยความจำภายใน และภายนอกนั้น จะต้องทำการพิจารณาระดับลอจิกของสัญญาณ EA ในขณะที่นั้นด้วย

การเชื่อมต่อกับหน่วยความจำ EPROM จะเกี่ยวข้องกับสัญญาณ 4 กลุ่ม ด้วยกันคือ

-ขาสัญญาณสำหรับจ่ายไฟฟ้าให้กับไอซี (Power)

-บัสแอดเดรส

-บัสข้อมูล

บัสสัญญาณควบคุมการส่งออกข้อมูล

สำหรับการพิจารณาจำนวนเส้นของสัญญาณแอดเดรสที่ใช้กับ EPROM ตัวหนึ่ง ๆ นั้น ตัวอย่างเช่น EPROM ขนาด 4096*8 จะนำตัวเลขค่าแรกซึ่งบอกถึงความจุในการเก็บข้อมูล มาคำนวณ ดังนี้

$$2^x = 4096$$

$$x = \log(4096)/\log(2)$$

$$x = 12$$

ดังนั้นจำนวนเส้นสัญญาณแอดเดรสที่ใช้กับ EPROM จึงมีทั้งหมด 12 เส้น

สัญญาณที่เชื่อมต่อเข้ากับกลุ่มสัญญาณแอดเดรสของ EPROM มักเป็นกลุ่มของสัญญาณเอาต์พุตที่มาจากพอร์ตของไมโครคอนโทรลเลอร์ (โดยปกติจะเป็นบัสแอดเดรสของระบบ) เพื่อทำการกำหนดตำแหน่งข้อมูลที่ต้องการ ข้อมูลภายในบัสแอดเดรสนี้จะถูกนำมาทำการถอดรหัสเพื่อหาตำแหน่งที่ต้องการภายใน EPROM ส่วนบัสข้อมูลมีหน้าที่สำหรับการเอาต์พุตข้อมูลที่เกิดขึ้นในตำแหน่ง ภายในหน่วยความจำ ซึ่งถูกระบุมาจากบัสแอดเดรส

สำหรับสัญญาณควบคุมการส่งออก ได้แก่

-สัญญาณ CS (Chip Select): มีหน้าที่เลือกให้ EPROM ท

-สัญญาณ OE (Output Enable): จะทำให้มีการไปนำข้อมูลภายในตำแหน่งที่ถูกระบุตามข้อมูลของบัสแอดเดรส และทำการส่งออกมายังบัสข้อมูลของ EPROM สัญญาณควบคุมนี้มักต่อเข้ากับขาสัญญาณ RD ของไมโครคอนโทรลเลอร์

สัญญาณต่าง ๆ ของ MCS-51 ที่นำมาใช้ในการติดต่อกับหน่วยความจำภายนอก

-สัญญาณ EA (External Access): ใช้ในการเลือกว่าจะอ่านข้อมูลจากหน่วยความจำภายนอก หรือ ภายใน

-พอร์ต (P0.0-P0.7): ในช่วงเวลาเริ่มต้นในการติดต่อกับหน่วยความจำภายนอกพอร์ต 0 จะเป็นค่าของแอดเดรสไบต์ต่ำ และเวลาต่อมาจึงจะเป็นบัสข้อมูล การส่งค่าแอดเดรสไบต์ต่ำจะอยู่ในช่วงเวลาของขาลงสัญญาณ ALE และจะคงอยู่จนสัญญาณ PSEN เปลี่ยนเป็นลอจิกต่ำ

-สัญญาณ ALE: ใช้ทำให้ไอรีเลตซ์ ก้างค่าสัญญาณแอดเดรสไว้

-สัญญาณ PSEN: ใช้ในการเลือกให้ EPROM ทำงาน และอ่านค่าข้อมูลกลับมา

-พอร์ต 2 (P2.0-P2.3): เป็นข้อมูลแอดเดรสไบต์สูงของหน่วยความจำ ซึ่งมีค่าแอดเดรสที่พอร์ตนี้จะค้างอยู่ตลอดช่วงรอบเวลาของการติดต่อกับหน่วยความจำโปรแกรม

1.1.8 หน่วยความจำข้อมูลทาง MCS-51

หน่วยความจำข้อมูลมีหน้าที่สำหรับเก็บข้อมูล หรือตัวแปรที่เกิดขึ้น ในขณะที่กำลัง

ประมวลผล

โปรแกรมไว้เป็นการชั่วคราว ซึ่งหน่วยความจำข้อมูลจัดเป็นหน่วยความจำ RAM แบบสแตติก ดังนั้นเมื่อไม่มีการจ่ายไฟให้กับระบบ ก็จะทำให้ข้อมูลที่จัดเก็บไว้ภายในหน่วยความจำสูญหายไปทันที

หน่วยความจำข้อมูลของ MCS-51 มีได้สูงสุดไม่เกิน 64 กิโลไบต์ และแยกได้เป็น 2 ลักษณะ ตามตำแหน่งที่ตั้งของหน่วยความจำ คือ หน่วยความจำภายใน ซึ่งเป็น RAM ที่อยู่ภายในตัวไอซีไมโครคอนโทรลเลอร์ และหน่วยความจำข้อมูลภายนอก ซึ่งเป็นการใช้หน่วยความจำ RAM มาเพิ่มเข้าไปในวงจร

1. หน่วยความจำข้อมูลภายใน

มีจำนวนทั้งหมด 256 ไบต์ โดยจำแนกได้เป็น 2 ลักษณะ คือ พื้นที่เฉพาะสำหรับซีพียูใช้งาน (เรียกรีจิสเตอร์) และพื้นที่สำหรับโปรแกรมใช้งานที่ผู้ใช้สร้างขึ้นมา

1.1) หน่วยความจำ 128 ไบต์แรก

มีตำแหน่งแอดเดรสอยู่ในช่วง 00H-7FH ซึ่งแยกเป็น 3 ส่วนการใช้งาน

-แอดเดรส 00H-1FH : จำนวน 32 ไบต์ แบ่งเป็น 4 กลุ่ม (เรียก แบนจ์) แต่ละกลุ่มใช้งานในฐานะของรีจิสเตอร์ใช้งานทั่วไป เรียก รีจิสเตอร์ R0-R7 การสวิตซ์เลือกแต่ละกลุ่มกำหนดจากค่าบิตภายในรีจิสเตอร์ PSW แต่ปกติมักใช้งานในแบนจ์ 0 เท่านั้น แบนจ์อื่น ๆ ที่เหลือสามารถใช้งานในลักษณะหน่วยความจำข้อมูลภายในปกติ ด้วยการอ้างถึงแอดเดรสนั้น ๆ โดยตรง

-แอดเดรส 20H-2FH : จำนวน 16 ไบต์ เป็นพื้นที่ที่ผู้ใช้สามารถอ้างถึงหน่วยความจำได้ทั้งลักษณะของไบต์ข้อมูลตามปกติเท่านั้น

1.2) หน่วยความจำ 128 ไบต์ถัดมา

ตั้งแต่แอดเดรส 80H-FFH เป็นหน่วยความจำที่มีการใช้งานเฉพาะจาก MCS-51 เท่านั้น โดยจะนำมาใช้เป็นตำแหน่งของรีจิสเตอร์หน้าที่พิเศษ (Special Function Register : SFR) จำนวน 20 ตำแหน่ง SFR เป็นรีจิสเตอร์สำหรับการควบคุมหน้าที่ และการทำงานของอุปกรณ์ หรือพอร์ตของ MCS-51

2. หน่วยความจำข้อมูลภายนอก

การนำหน่วยความจำข้อมูลภายนอกมาใช้กรณีที่มีความต้องการหน่วยความจำสำหรับเก็บข้อมูลชั่วคราว หรือ ตัวแปรของโปรแกรมมีมากกว่าขนาดหน่วยความจำข้อมูลภายใน (ซึ่งมีขนาดเพียง 128 หรือ 256 ไบต์) จะเก็บได้ ในการนำหน่วยความจำข้อมูลภายนอกมาใช้จะมีผลทำ

ให้พอร์ตอินพุท/เอาต์พุทข้อมูลของ MCS-51 ถูกนำไปใช้เพื่อติดต่อกับหน่วยความจำเหล่านี้แทน ดังนั้นต้องมีการใช้วงจรประกอบอื่น ๆ เข้ามาชดเชย

ลักษณะของหน่วยความจำ ROM เหมือน EPROM มาก และยังใช้การติดต่อกันเหมือนกัน ด้วย คือ มีการใช้ บัสแอดเดรส, บัสข้อมูล, กลุ่มสัญญาณควบคุม แต่มีการเพิ่มสัญญาณ WR ขึ้นมา เพื่อใช้ในการนำข้อมูลจากบัสข้อมูลเก็บลงไปยังหน่วยความจำตามตำแหน่งที่ระบุแอดเดรสมา

ในการเชื่อมต่อ MCS-51 เข้ากับ RAM จะใช้วิธีการเดียวกับการเชื่อมต่อกับ EPROM โดยมีการแยกชุดคำสั่งที่ติดต่อกับหน่วยความจำทั้งสอง ดังแสดงในตารางที่ 1.2 และตารางที่ 1.3

ตารางที่ 1.2 ชุดคำสั่งในการโอนย้ายข้อมูลระหว่างหน่วยความจำข้อมูลภายใน

รูปแบบคำสั่ง	การทำงาน	เวลาการประมวลคำสั่ง (machine cycle)
MOV A,<src>	A = ,<src>	1
MOV <dest>, A	<dest> = A	1
MOV <dest>, <src>	<dest> = <src>	2
MOV DPTR, #data 16	DPTR = ค่าคงที่ 16 บิต	2
PUSH <src>	INC SP; MOV "@SP", <src>	2
POP <dest>	MOV <dest>, "@SP", DEC SP	2
XCH A,<byte>	ACC and <byte> exchange data	1
XCHD A, @Ri	ACC and @Ri exchange low nibbles	1

ตารางที่ 1.3 ชุดคำสั่งในการโอนย้ายข้อมูลระหว่างหน่วยความจำข้อมูลภายนอก

จำนวนบิต แอดเดรส	รูปแบบคำสั่ง	การทำงาน	เวลาการประมวลผลคำสั่ง (machine cycle)
8 บิต	MOVX A, @Ri	Read External RAM @Ri	2
8 บิต	MOVX @Ri, A	Write External RAM @Ri	2
16 บิต	MOVX A, @DPTR	Read External RAM @DPTR	2
16 บิต	MOVX @DPTR, A	Write External RAM @DPTR	2

กรณีคำสั่งติดต่อกับหน่วยความจำข้อมูลภายนอกนั้น ต้องทำการอ้างแอดเดรสโดยอ้อมผ่านรีจิสเตอร์ R0, R1, DPTR ซึ่งทำหน้าที่เก็บแอดเดรสหน่วยความจำที่ต้องการไว้แทน โดย R0, R1 จะทำได้ในช่วงแอดเดรสไม่เกิน 256 ไบต์ เนื่องจากเป็นรีจิสเตอร์ 8 บิต หากใช้ผ่านรีจิสเตอร์ DPTR จะทำได้ในขอบเขตที่กว้างมากเต็มพื้นที่ที่ MCS-51 สามารถอ้างได้ สำหรับตัวอักษร X ที่เพิ่มเติมท้ายคำสั่งแสดงให้ทราบว่าต้องการจะติดต่อกับหน่วยความจำข้อมูลภายนอกเท่านั้น

1.1.4 พอร์ตอินพุต/เอาต์พุตของ MCS-51

พอร์ต หมายถึง แอดเดรสหนึ่งที่ได้รับการกำหนดไว้เพื่อการ โอนย้ายข้อมูลระหว่างไมโครคอนโทรลเลอร์ กับอุปกรณ์ภายนอก การนำเข้าข้อมูลจากอุปกรณ์ภายนอก เรียก พอร์ตอินพุต ส่วนการส่งออกข้อมูล เรียก พอร์ตเอาต์พุต ซึ่งสามารถแยกประเภทพอร์ตได้เป็น 2 ลักษณะ คือ พอร์ตแบบขนาน ซึ่งทำการส่งจำนวนบิตข้อมูลทีละบิต ๆ จนครบจำนวน

MCS-51 จะหลักการ Memory mapped system คือ การอ้างถึงหน่วยความจำ ดังนั้นการนำเข้าหรือส่งออกข้อมูลกับพอร์ต จึงใช้คำสั่งการอ่านจากหน่วยความจำ หรือคำสั่งการเขียนค่าข้อมูลไปยังตำแหน่งหน่วยความจำแทน ดังนั้นชุดคำสั่งของ MCS-51 จะไม่มีคำสั่งที่เกี่ยวข้องกับการทำงานพอร์ต เช่น คำสั่ง IN (นำเข้าข้อมูลจากพอร์ต) หรือ OUT (ส่งข้อมูลออกจากพอร์ต)

MCS-51 ยังมีชุดคำสั่งที่จัดการข้อมูลแบบบิตได้โดยตรง ดังนั้นสามารถใช้คำสั่งเหล่านี้จัดการพอร์ตอินพุต/เอาต์พุตแบบเส้นสัญญาณเดี่ยวได้ โดยใช้คำสั่ง SETB เพื่อกำหนดค่าเป็น 1 หรือ CLR เพื่อทำให้บิตมีค่าเป็น 0 ส่วนการตรวจสอบบิตของแต่ละพอร์ตอินพุต/เอาต์พุตสามารถใช้ชุดคำสั่งในการทดสอบได้โดยตรง ไม่ต้องมีการอ้างค่าของพอร์ตไปยังรีจิสเตอร์ก่อน

ตารางที่ 1.4 คำสั่งการส่งข้อมูลออกทางพอร์ต

คำสั่ง	ความหมาย
MOV P0, A MOV P1, #00000101B MOV P1, #05H	นำค่าภายในรีจิสเตอร์ A ส่งออกไปยังพอร์ต 0 ส่งค่าข้อมูลออกไปยังพอร์ต 1 โดยตรง เพียงแต่กำหนดค่าข้อมูลเป็นเลขฐานสิบหก

ตารางที่ 1.5 คำสั่งการรับข้อมูลเข้ามาทางพอร์ต

คำสั่ง	ความหมาย
MOV A, P1	อ่านข้อมูลเข้ามาพอร์ต 1 และนำไปเก็บไว้ในรีจิสเตอร์ A
MOV R0, P2	อ่านข้อมูลเข้ามาจากพอร์ต 2 และนำไปเก็บไว้ในรีจิสเตอร์ R0

1.2 การใช้งาน 8255 กับ MCS-51

การขยายเพิ่มเติมพอร์ตอินพุต/เอาต์พุตของ MCS-51 นอกจากใช้ไอซีประเภทแลตซ์ และบัฟเฟอร์ประกอบเข้ากับบัสของระบบแล้ว ยังสามารถใช้วงจรถวาย LSI เบอร์ 8255 มาทำหน้าที่เป็นพอร์ตอินพุต/เอาต์พุต ตามการโปรแกรมด้วยซอฟต์แวร์

ไอซีเบอร์ 8255 ได้รับการออกแบบมาเพื่อทำหน้าที่เป็นพอร์ตสำหรับการรับส่งข้อมูลแบบขนานระหว่างอุปกรณ์ภายนอกกับไมโครคอนโทรลเลอร์ ซึ่ง 8255 นี้ สามารถเปลี่ยนแปลงลักษณะการทำงานของพอร์ตให้เป็นการอินพุต หรือเอาต์พุตได้โดยการส่งข้อมูลควบคุมจากไมโครคอนโทรลเลอร์ก่อนที่จะเริ่มต้นใช้งาน คือ สามารถโปรแกรมการทำงานได้

8255 เป็นไอซีที่มี 40 ขา สามารถที่จะต่อเป็นพอร์ตให้ไมโครโปรเซสเซอร์ได้ 3 พอร์ต โดยมีสร้างพื้นฐานแสดงได้ ดังรูป 1.2

การเรียกพอร์ตของ 8255 จะเรียกพอร์ตต่าง ๆ ว่า พอร์ต A พอร์ต B และพอร์ต C โดยพอร์ต C แยกเป็น 2 ส่วน คือ พอร์ต C ล่างหรือตั้งแต่ PC0-PC3 มีจำนวน 4 บิตและพอร์ต C บนหรือตั้งแต่ PC4-PC7 ที่พิเศษคือ พอร์ตทุกพอร์ตเป็นได้ทั้งพอร์ตอินพุตและพอร์ตเอาต์พุต

แผนผังภายในของไอซีและการจัดวางของไอซี 8255 แสดงในรูปที่ 1.2 การทำงานของวงจรถวายจะใช้สัญญาณควบคุมจากไมโครโปรเซสเซอร์มาควบคุมการทำงานโดยไมโครโปรเซสเซอร์จะส่งคำสั่งมาโปรแกรมการทำงาน หรือ กำหนดรูปแบบของพอร์ตให้เป็นการอินพุตหรือเอาต์พุตได้

การกำหนดโหมคการทำงาน จะต้องส่งข้อมูลคำสั่งเข้าโปรแกรมในพอร์ตควบคุมของ 8255 แต่ละบิตของข้อมูลที่ส่งไปจะมีความหมายในตัวเอง ลักษณะความหมายของแต่ละบิตในรหัสควบคุมแสดงได้ดังรูปที่ 1.4

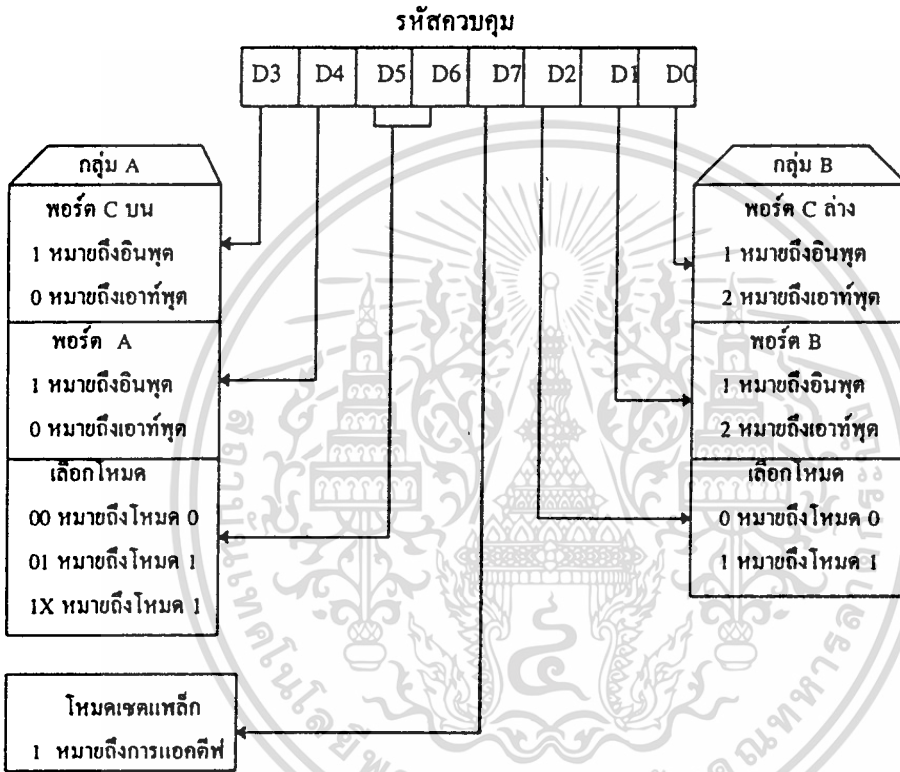


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การโปรแกรม 8255 คือ การใช้ค่ารหัสบิตต่าง ๆ เข้าไปในรหัสควบคุมแล้วส่งไปยังรีจิสเตอร์ของพอร์ตควบคุม ความหมายของบิตต่าง ๆ มีดังนี้

บิต D7 เป็นบิตที่แสดงรหัสคำสั่งควบคุม ถ้าบิตนี้เป็น “1” หมายถึงรหัสควบคุมนี้จะมีผลต่อการเปลี่ยนแปลงการเซตโหมดต่าง ๆ ของ 8255

บิต D6 และ D5 เป็นการเลือกโหมดของพอร์ต A ซึ่งมีโหมด 3 โหมดคือ โหมด 0 โหมด 1 และ โหมด 2 ดังแสดงในรูปที่ 3.4



รูปที่ 1.4 ความหมายของบิตต่าง ๆ ในรหัสควบคุม

บิต D4 ถ้ามีค่าเป็น “0” หมายถึงการกำหนดพอร์ต A เป็นเอาต์พุต ถ้ามีค่าเป็น “1” จะหมายถึงการกำหนดให้พอร์ต A เป็นอินพุต

บิต D3 เป็นบิตที่บอกถึงการเซตของพอร์ต C ถ้าเป็น “0” จะทำให้พอร์ต C บนเป็นเอาต์พุต

บิต D2 เป็นบิตที่บอกถึงการเซตโหมดของพอร์ต B ถ้าเป็น “0” หมายถึงการเลือกพอร์ต B เป็นโหมด 0 และถ้าเป็น “1” หมายถึงการเลือกโหมด 1

บิต D0 เป็นการกำหนดอินพุต/เอาต์พุตของพอร์ต C ถ้าเป็น “0” หมายถึง เอาต์พุต ถ้าเป็น 1 หมายถึง อินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



บทที่ 2 ทฤษฎีเกี่ยวกับการสร้าง

2.1 การสร้างแบบจำลองลิฟต์ในโครงการงาน

2.1.1 ส่วนประกอบต่าง ๆ ในการสร้าง

1. พลาสติกใสขนาด 3 มิลลิเมตร
2. โครงอลูมิเนียม ตัว U และตัว L
3. เชือก
4. มอเตอร์กระแสไฟตรงขนาด 12 โวลต์ จำนวน 2 ตัว
5. อินฟราเรด MRD 750 จำนวน 2 ตัว และ MLED 81 จำนวน 10 ตัว
6. มูเล่ห์ 4 ตัว
7. ตัวถ่วงน้ำหนัก

2.1.2 การสร้างแบบจำลองลิฟต์

1. นำพลาสติกมาตัดเพื่อทำเป็นบ่อลิฟต์โดยมีชั้นส่วนดังนี้
 - 1.1 ด้านข้างของบ่อลิฟต์ มีขนาด 20.4 x 80 cm. จำนวน 2 แผ่น
 - 1.2 ด้านหลังของบ่อลิฟต์ มีขนาด 30.2 x 80 cm. จำนวน 1 แผ่น
 - 1.3 แผ่นหน้าของบ่อลิฟต์มีขนาด 30.2 x 75 cm. จำนวน 1 แผ่นโดยแผ่นด้านหน้านี้จะทำการเจาะแผ่นพลาสติกเป็นช่อง ๆ โดยจะมีจำนวนช่องทั้งหมด 10 ช่อง เพราะบ่อลิฟต์เป็นลิฟต์ 5 ชั้นขนาดของช่องมีขนาด 5 x 10 cm. แต่ละชั้นมีระยะห่าง 10 cm.
- 1.4 แผ่นปิดด้านบนของบ่อลิฟต์จะนำมอเตอร์กระแสตรง 12 โวลต์ มายึดติดเพื่อเป็นแรงขับให้ห้องโดยสารเคลื่อนที่ มีการเจาะรูที่แผ่นเพื่อเป็นทางรอดผ่านของเชือกที่จะไปผูกคล้องระหว่างตัวถ่วงน้ำหนักและห้องผู้โดยสาร

1.5 แผ่นฐานของบอลลีฟต์ ขนาด 40 x 40 cm. จำนวน 1 แผ่น

1.6 แผ่นพลาสติกที่จะทำเป็นห้องควบคุมที่จะวางที่ด้านหน้าของบอลลีฟต์ โดยประกอบด้วย

1. ด้านข้าง ขนาด 10 x 5 cm. จำนวน 2 แผ่น
2. แผ่นด้านบน ขนาด 10 x 30.2 cm. จำนวน 1 แผ่น
3. แผ่นด้านหน้า ขนาด 5 x 30.2 cm. จำนวน 1 แผ่น

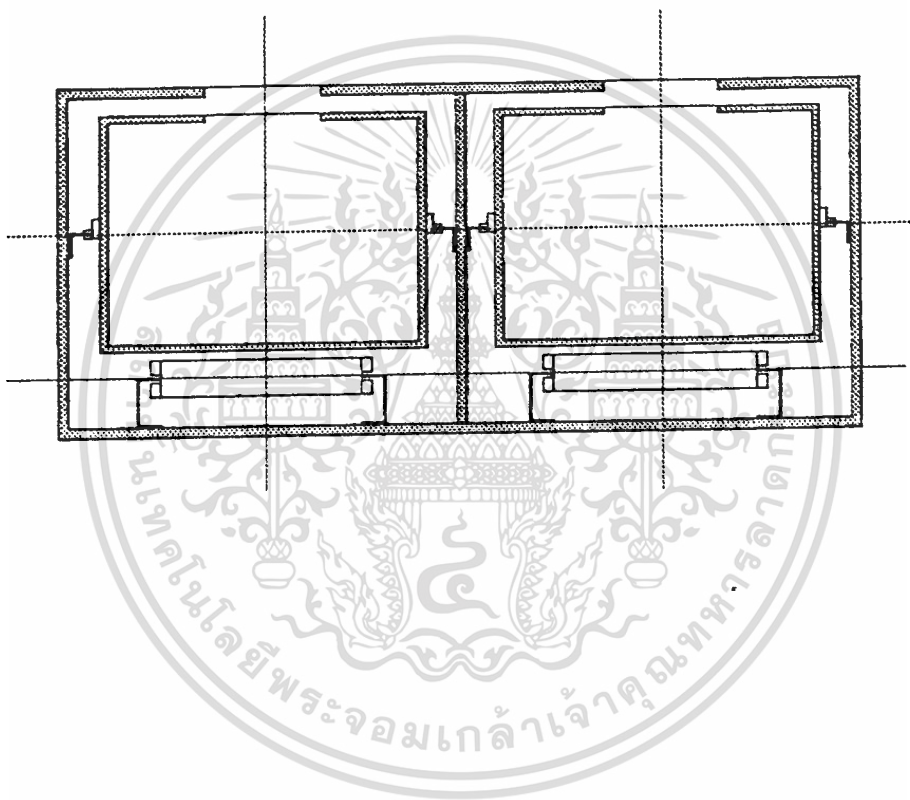
2. เมื่อได้แผ่นพลาสติกที่จะนำมาประกอบเป็นบอลลีฟต์แล้วให้นำรางอลูมิเนียมตัว L และตัว U มาตัดได้ขนาดความยาว 80 cm. แบบละ 4 เส้น

- รางอลูมิเนียมตัว L ยึดติดกับด้านข้างของบอลลีฟต์ทุกด้าน คือ ด้านข้างบอลลีฟต์ทั้ง 2 ด้าน และ ครึ่งแผ่นกั้นกลางของบอลลีฟต์ 2 เส้น ในการติดตั้งอลูมิเนียมตัว L นี้เพื่อเป็นร่องนำทางการวิ่งของห้องโดยสารของลิฟต์
- รางอลูมิเนียมตัว U ยึดติดด้านหลังของบอลลีฟต์ทั้ง 2 ช่อง โดยการหันหน้าของรางทั้งสองเข้าหากัน ในแต่ละบ่อจะมีรางอลูมิเนียมอยู่ 2 เส้น ดังนั้นจะใช้รางแบบนี้จำนวน 4 เส้น ในการติดตั้งนี้ด้านหลังเพื่อเป็นร่องนำทางในการเคลื่อนที่ของตัวถ่วงน้ำหนัก

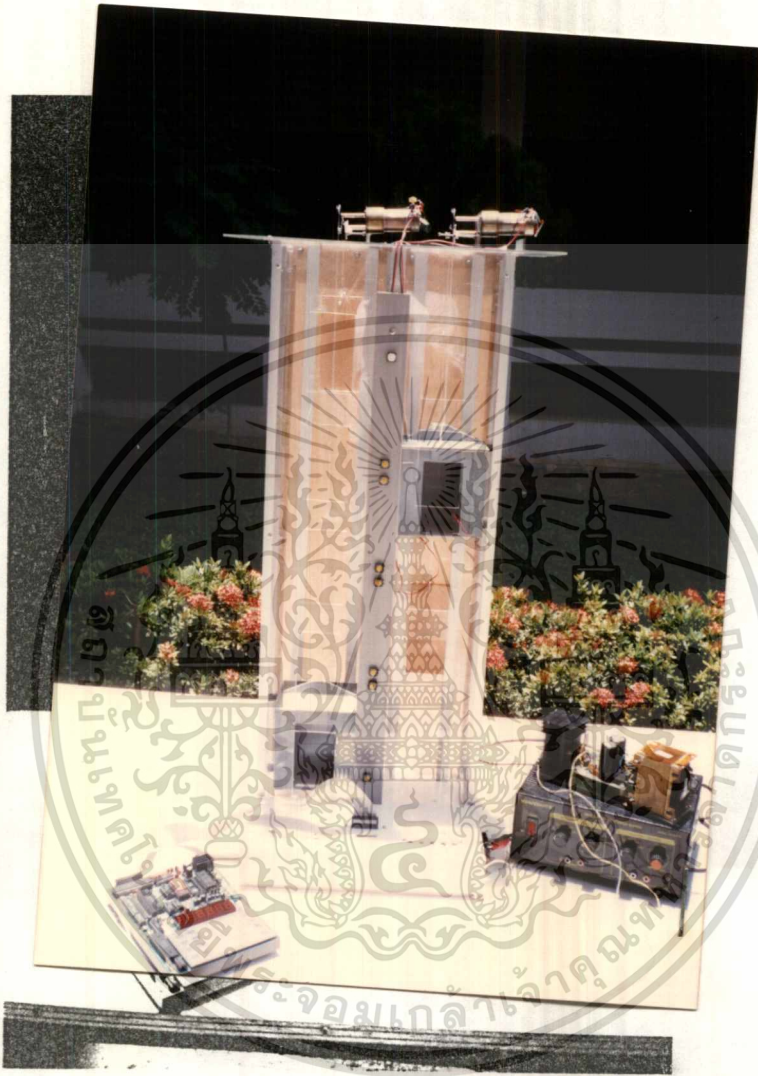
3. นำชิ้นส่วนต่าง ๆ ของบอลลีฟต์มาประกอบกัน จะได้เป็นบอลลีฟต์ที่มี 2 ช่อง คือจะมีห้องโดยสาร 2 ห้อง หรือ เรียกว่า Duplex Lift

4. ทำการเดินสายไปเพื่อจะควบคุมการทำงานของลิฟต์ โดยจะมีสวิทช์อยู่ด้านหน้าเป็นสวิทช์ของการบอกสถานะการขึ้นและลง ตามความต้องการ

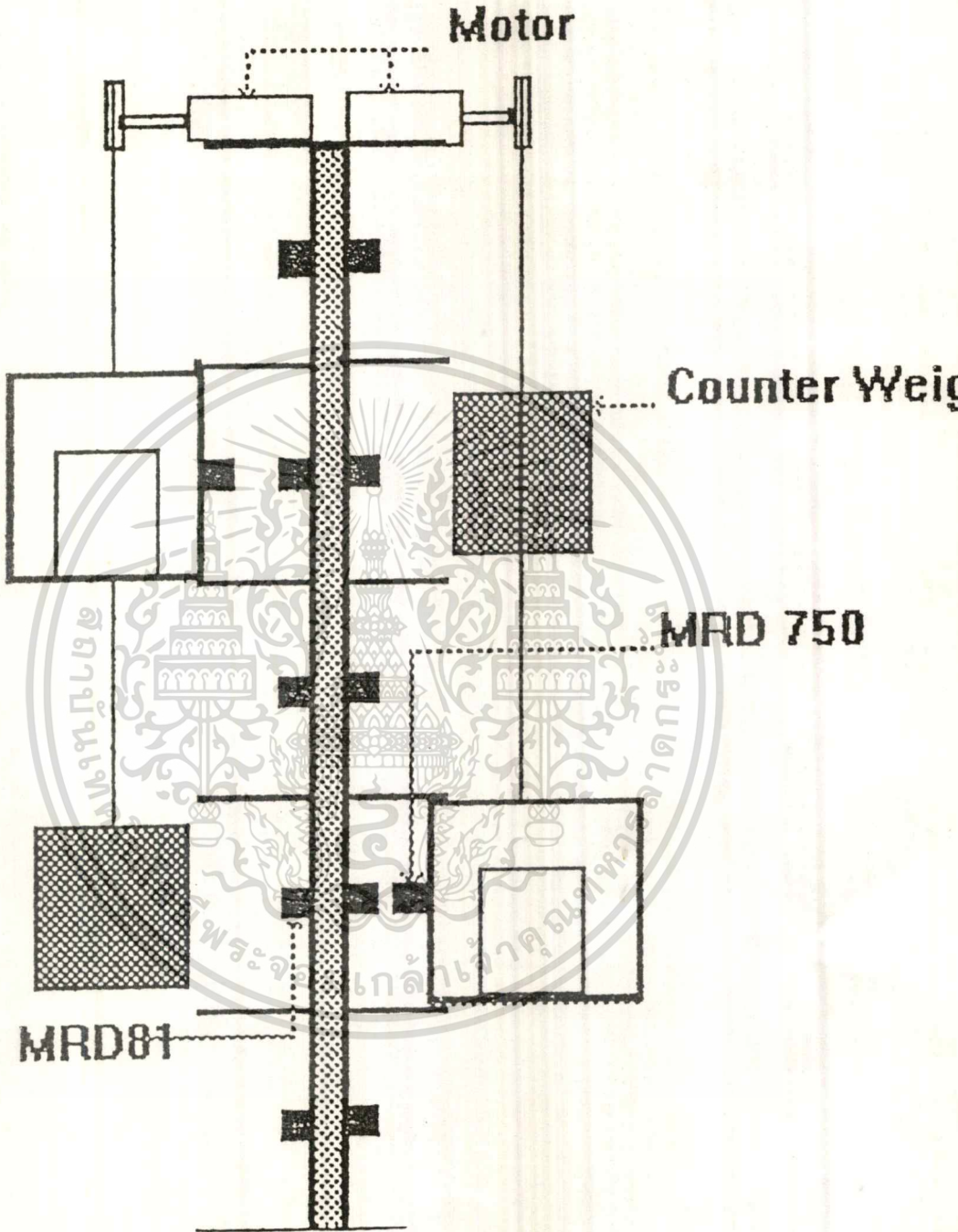
5. ถ่วงคอนโทรลที่เป็นถ่วงเล็ก ๆ อยู่ติดด้านหน้าของบอลลีฟต์ จะมีวงจรควบคุมการทำงานต่าง ๆ ของระบบลิฟต์ และด้านนอกของถ่วงจะมีปุ่มรับคำสั่งการใช้ลิฟต์ของผู้ใช้จะแยกกันคนละชุดในลิฟต์แต่ละข้าง



รูปที่ 2.1 แสดง Layout การติดตั้งอลูมิเนียมในบ่อลิฟต์ และการติดตั้งถ่วงน้ำหนัก

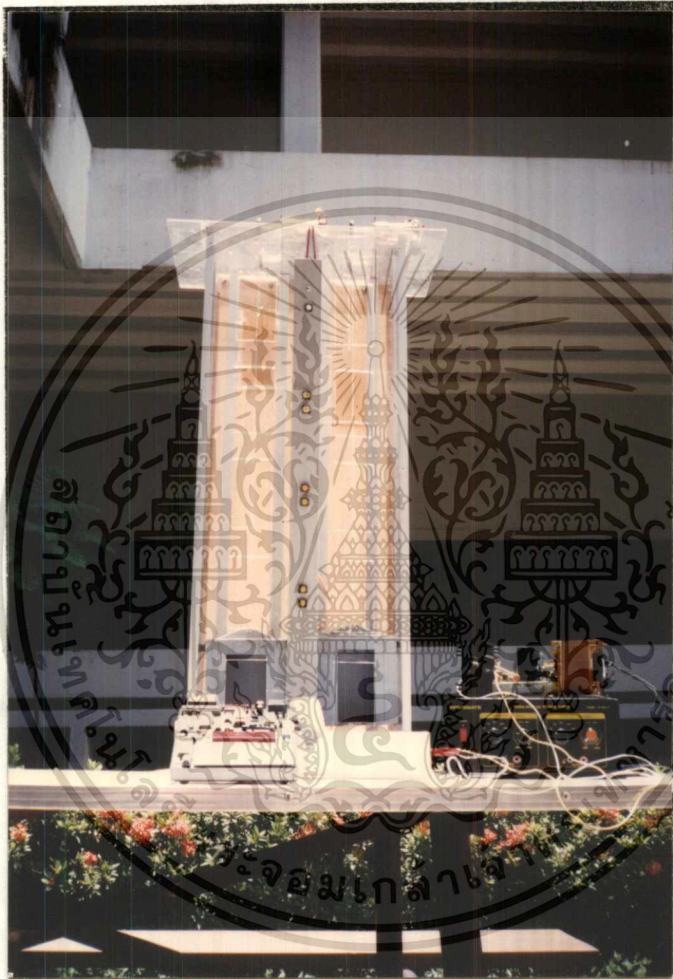


รูปที่ 2.1.1 แสดงแบบจำลองที่ สร้างเสร็จแล้ว



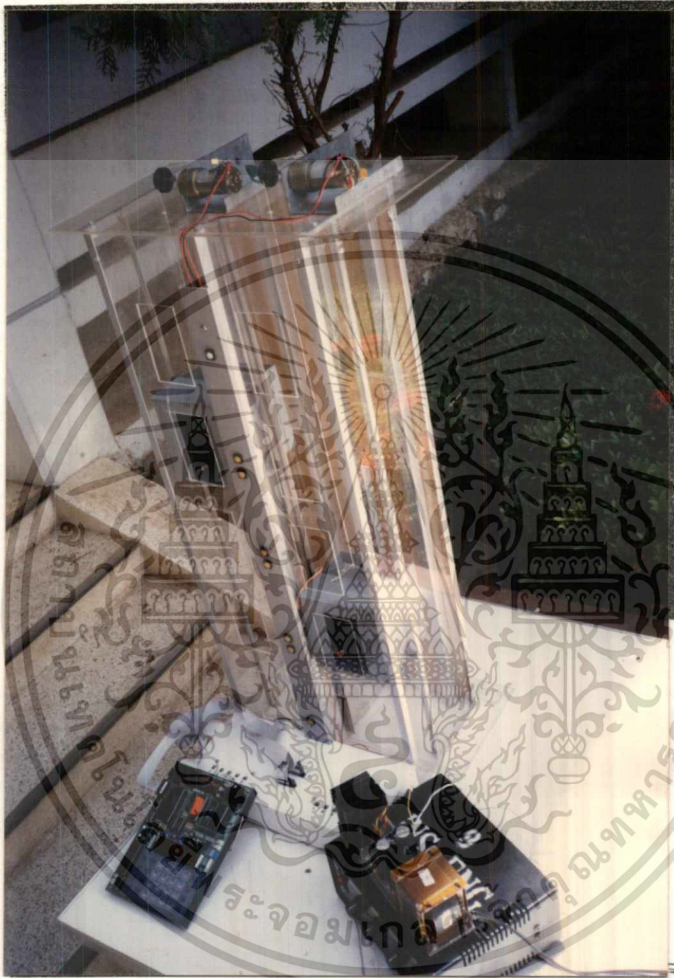
รูปที่ 2.2 แสดงการติดตั้งเซนเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



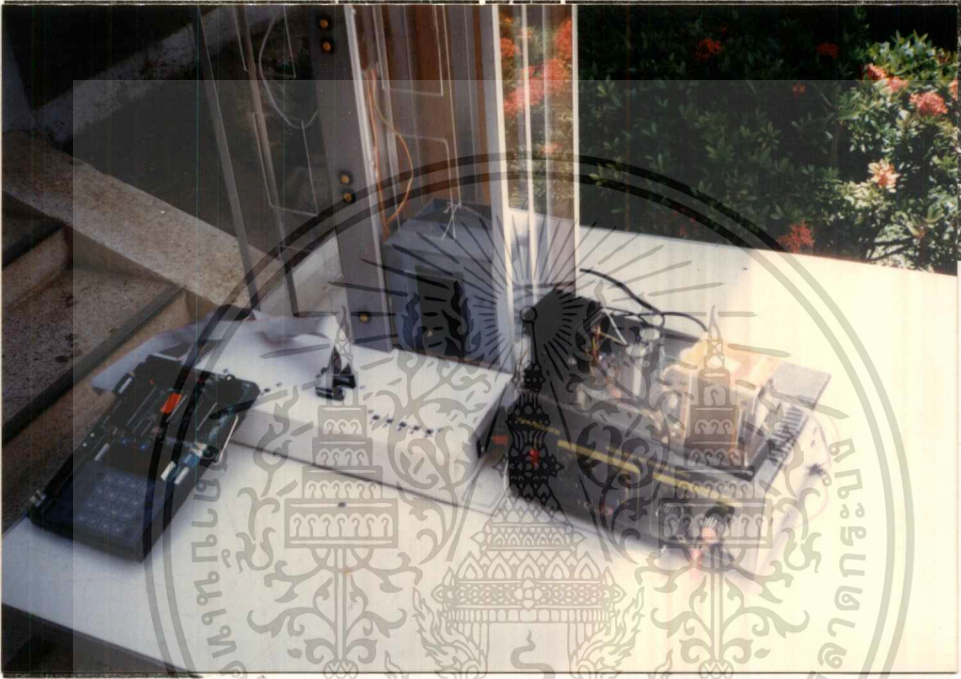
รูปที่ 1 แสดงลักษณะ โครงสร้างแบบจำลองทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



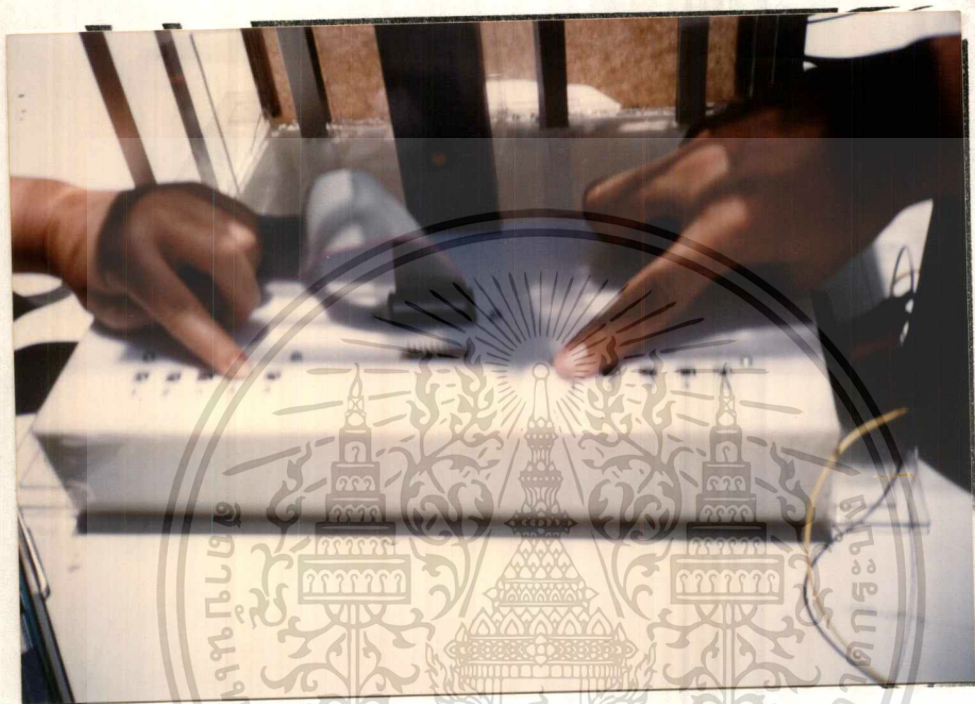
รูปที่ 2 แสดงลักษณะของลิฟต์ที่ประกอบเสร็จแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3 แสดงส่วนของคอนโทรลลิฟต์ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4 แสดงกล่องเล็กที่ภายในบรรจุโปรแกรมควบคุมทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การประยุกต์ใช้งาน ANT-32 กับโครงงาน

ในโครงงานนี้มีการนำ ANT-32 มาประยุกต์ใช้งาน เพื่อให้โครงงานมีประสิทธิภาพยิ่งขึ้น จึงต้องทำความเข้าใจกับบอร์ด ANT-32 ก่อน

2.2.1 คุณสมบัติของบอร์ด ANT-32

ANT-32 มีส่วนประกอบต่าง ๆ ดังภาพแสดงบอร์ด, ตำแหน่งจัมป์เปอร์ และพอร์ตต่าง ๆ

ดังรูปที่ 2.3

ทั้งนี้ ชิปในบางตำแหน่งที่ไม่ได้ถูกมาใช้ในโครงงาน

-เป็นบอร์ดคอนโทรลเลอร์ในตระกูล MCS-51 (8031/032) ซึ่งใช้ CPU เบอร์ 80C32 ทำหน้าที่ความถี่สัญญาณนาฬิกา 11.0592 Mhz

-ใช้งานหน่วยความจำบนบอร์ดได้ 3 ตำแหน่งด้วยกัน คือ

U2 เป็นหน่วยความจำโปรแกรม (Program Memory) ใช้กับอีพรอมขนาด 8-32 กิโลไบต์ เบอร์ 2764, 27128 หรือ 27256

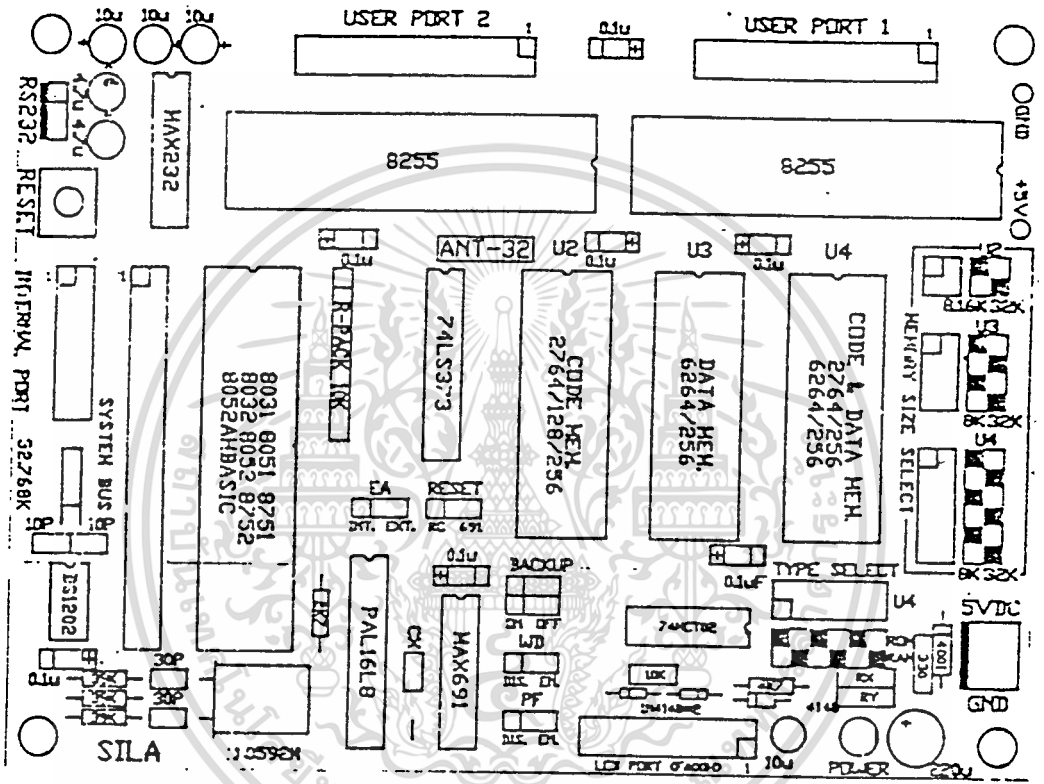
U3 เป็นหน่วยความจำข้อมูล (Data Memory) ใช้กับแรมขนาด 8 กิโลไบต์ เบอร์ 6264 หรือ 32 กิโลไบต์ เบอร์ 62256

U4 เป็นหน่วยความจำโปรแกรมและข้อมูล ใช้กับอีพรอม หรือแรม เบอร์ 2764, 27256, 6264 หรือ 62256

-พอร์ต I/O เบอร์ 8255 จำนวน 2 ตัว (48 บิต) สำหรับต่อเนื่องเพื่อใช้งานกับอุปกรณ์ภายนอกพอร์ต LCD สำหรับการต่อใช้งานกับ LCD แบบ DOT MATRIX

-วงจร SERIAL INTERFACE DRIVER-RS232 เบอร์ MAX232 ต่อเข้ากับเครื่องไมโครคอมพิวเตอร์

-วงจร RTC (Real Time Clock) เซอร์ DS1202



รูปที่ 2.3 แสดงบอร์ด และตำแหน่งจิมป์เปอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 การประยุกต์ใช้งาน ANT-32 กับโครงการงาน

จากคุณสมบัติของ ANT-32 ทำให้ทราบว่าใช้ไมโครคอนโทรลเลอร์เบอร์ 8032 เป็นซีพียู ซึ่ง 8032 ไม่มีหน่วยความจำโปรแกรม (program memory) อยู่ภายใน ดังนั้นในการใช้งาน เราต้องใช้หน่วยความจำโปรแกรมภายนอกเสมอ ในโครงการเราเลือก EPROM เบอร์ 2764 ซึ่งมีขนาด 8K*8 มาใช้เก็บโปรแกรมควบคุมการทำงานของอุปกรณ์ภายนอกที่นำมาเชื่อมต่อกับ ANT-32

เนื่องจากระบบบัสแอดเดรส และบัสข้อมูลของ 8032 เป็นลักษณะการมัลติเพล็กซ์จากพอร์ตเดียวกัน คั้นในช่วงเริ่มต้น เส้นสัญญาณเหล่านี้ของพอร์ตจะใช้ในการส่งค่าแอดเดรสของตำแหน่งที่ต้องการติดต่อด้วย ในช่วงเวลาต่อมาจึงเปลี่ยนไปเป็นสถานะอิมพีแดนซ์สูง เพื่อใช้งานในฐานะของบัสข้อมูล แต่ EPROM นั้นไม่ใช้การมัลติเพล็กซ์ มันจะมีขาสัญญาณบัสแอดเดรส และบัสข้อมูลแยกจากกัน ดังนั้นในการเชื่อมต่อ EPROM เพื่อทำหน้าที่เป็นหน่วยความจำโปรแกรมภายนอกกับ 8032 นั้น ต้อง

วงจรแลตซ์เพิ่มเติม เพื่อทำการค้างค่าแอดเดรสที่ส่งมาจาก 8032 ในช่วงเริ่มต้น ให้กับขาสัญญาณของ EPROM โดยในบอร์ด ANT-32 ใช้ไอซีเบอร์ 74HC7373 เป็นวงจรแลตซ์

แต่ 8032 มีหน่วยความจำข้อมูลภายในตัวมันเองจำนวน 256 ไบต์ ซึ่งก็ยังไม่เพียงพอในการเก็บข้อมูลที่ใช้ในการคำนวณ เราจึงใช้ RAM เบอร์ 6264 ซึ่งมีขนาด 8K*8 มาช่วยในการเก็บข้อมูล

การใช้งาน ANT-32 จะมีการกำหนดหน่วยความจำโปรแกรม และหน่วยความจำข้อมูลภายนอกให้เริ่มต้นที่แอดเดรส 0000H เสมอ เพราะเมื่อมีการรีเซ็ต หรือเริ่มต้นการจ่ายไฟฟ้าให้กับระบบ 8032

ได้เริ่มต้นทำงานตามคำสั่งที่แอดเดรสนี้ทันที

ในการที่เราเอา RAM 6264 มาใช้เป็นหน่วยความจำข้อมูลภายนอกจะทำให้มีการนำพอร์ตอินพุต/เอาต์พุตข้อมูลของ 8032 ไปใช้ติดต่อกับหน่วยความจำข้อมูลภายนอก ดังนั้นเพื่อเป็นการเพิ่มเติมพอร์ตอินพุต/เอาต์พุต ในบอร์ด ANT-32 จึงมีการนำไอซีเบอร์ 8255 มาใช้ ซึ่ง 8255 นี้สามารถทำหน้าที่เป็นได้ทั้งอินพุต/เอาต์พุต ตามแต่เราจะโปรแกรม (Program Peripheral Interface) และใช้เป็นส่วนที่เชื่อมต่อกับอุปกรณ์ภายนอกโดยตรง

สำหรับบอร์ด ANT-32 เราใช้ 8255 จำนวน 1 ตัว เพราะฉะนั้นจะมีพอร์ตอินพุต/เอาต์พุต

จำนวน $24 \times 2 = 48$ บิต โดยแบ่งเป็น USER PROT1 และ 2 ซึ่งมีตำแหน่งแอดเดรส และการทำงาน ดังนี้

USER.PORT1	แอดเดรส F800H+8255	offset addr = actual addr
พอร์ท A	แอดเดรส F800H+00H	= F800H
พอร์ท B	แอดเดรส F800H+01H	= F801H
พอร์ท C	แอดเดรส F800H+02H	= F802H
โหมคพอร์ท	แอดเดรส F800H+0311	= F803H

2.3 การเชื่อมต่อของไมโครโปรเซสเซอร์กับอุปกรณ์ภายนอก

แบ่งได้เป็น 2 ชนิด

1. การเชื่อมต่อกับมอเตอร์
2. การเชื่อมต่อกับเซนเซอร์

2.3.1 การเชื่อมต่อกับมอเตอร์

จากการทดลองป้อน Voltage ค่าต่าง ๆ ให้กับมอเตอร์ พบว่าการใช้ไฟ DC 12 V จะให้ผลตอบสนองที่รวดเร็วตามต้องการ จึงใช้ไฟ DC 12 V ในการจ่ายให้กับมอเตอร์ แต่เนื่องจากสัญญาณที่ออกจาก ไมโครโปรเซสเซอร์ เป็นสัญญาณแบบดิจิทัล เราจึงต้องทำการออกแบบวงจร INTERFACE ดังนี้

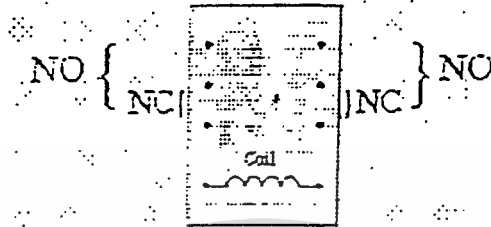
สิ่งที่เราต้องการควบคุม

คือทิศทาง (ขึ้น-ลง) และ ON-OFF MOTOR

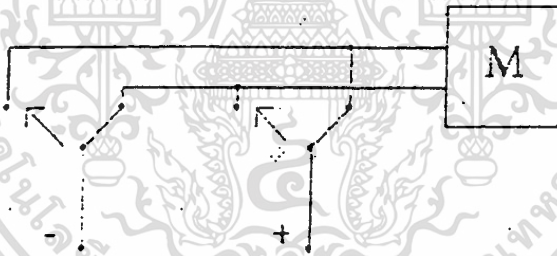
ทำการออกแบบวงจรเพื่อ Interface Motor กับ Microprocessor ดังนี้

เนื่องจากในการควบคุมมอเตอร์นั้น เราจะให้มอเตอร์สามารถเคลื่อนที่ได้สองทิศทาง คือ ขึ้นและลง ดังนั้นในการกลับทิศทางของมอเตอร์ เราจะต้องการกลับขั้ว คือ จากขั้วบวกเป็นขั้วลบ และจากขั้วลบเป็นขั้วบวก

Note : ข้อมูลที่ได้จาก Relay

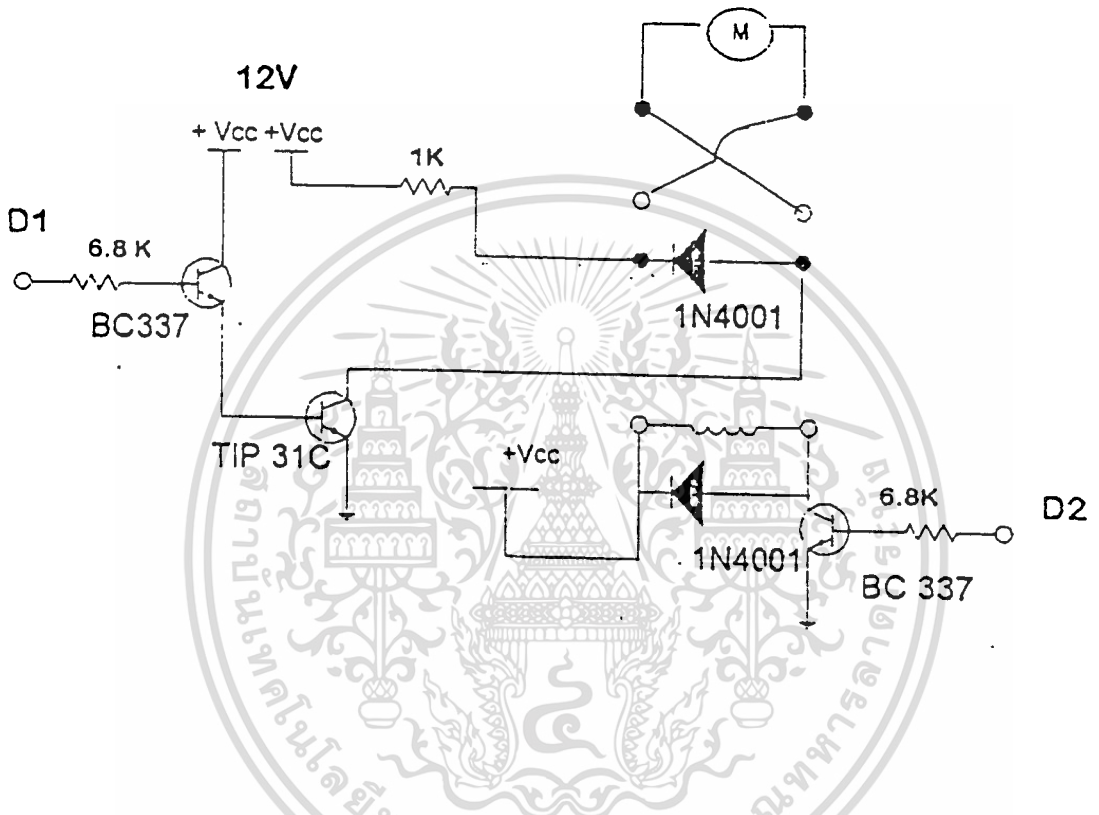


เนื่องจากลิตซ์ต้องสามารถขึ้นและลงได้ เราจึงต้องควบคุม Motor โดยใช้ Relay 2 Contact ตามวงจรอย่างง่ายได้ดังรูปที่ 2.4



รูปที่ 2.4 Relay 2 Contact

ส่วนวงจรในการไคร์ฟมอเตอร์เราได้ออกแบบดังนี้



รูปที่ 2.5 แสดงวงจร DRIVE MOTOR

ข้อกำหนดในการเชื่อมต่อวงจร
การขึ้นและลงของตัวลิฟต์

	Director Control Signal		Result
Logic	0	1	Up
Logic	1	1	down

หมายเหตุ การขึ้น-ลงของลิฟต์ จะใช้ไฟ DC 12 V

สำหรับ Relay จะใช้ไฟ DC 12 V. ระหว่าง Coil

ไม่ว่าการณ์ใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 การเชื่อมต่อกับระบบเซนเซอร์

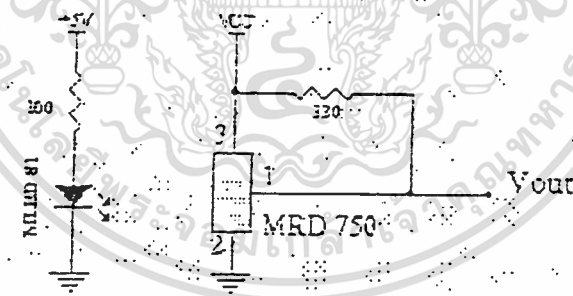
หลักการทำงานของ เซนเซอร์

ตัวเซนเซอร์ จะใช้หลักการสะท้อนของรังสี อินฟราเรด

การควบคุมการทำงานของลิฟต์จะประกอบด้วย 2 ส่วน

- ส่วนที่หนึ่งเป็น ตัวส่งสัญญาณ อินฟราเรด ที่ติดอยู่กับชั้นลิฟต์ เพื่อใช้ในการตัดสินใจในการขึ้น-ลงของลิฟต์
- ส่วนที่สองเป็น เซนเซอร์ ติดอยู่กับด้านข้างของกล่องลิฟต์เพื่อควบคุมให้ลิฟต์หยุด ณ ตำแหน่งที่ต้องการให้ลิฟต์หยุด

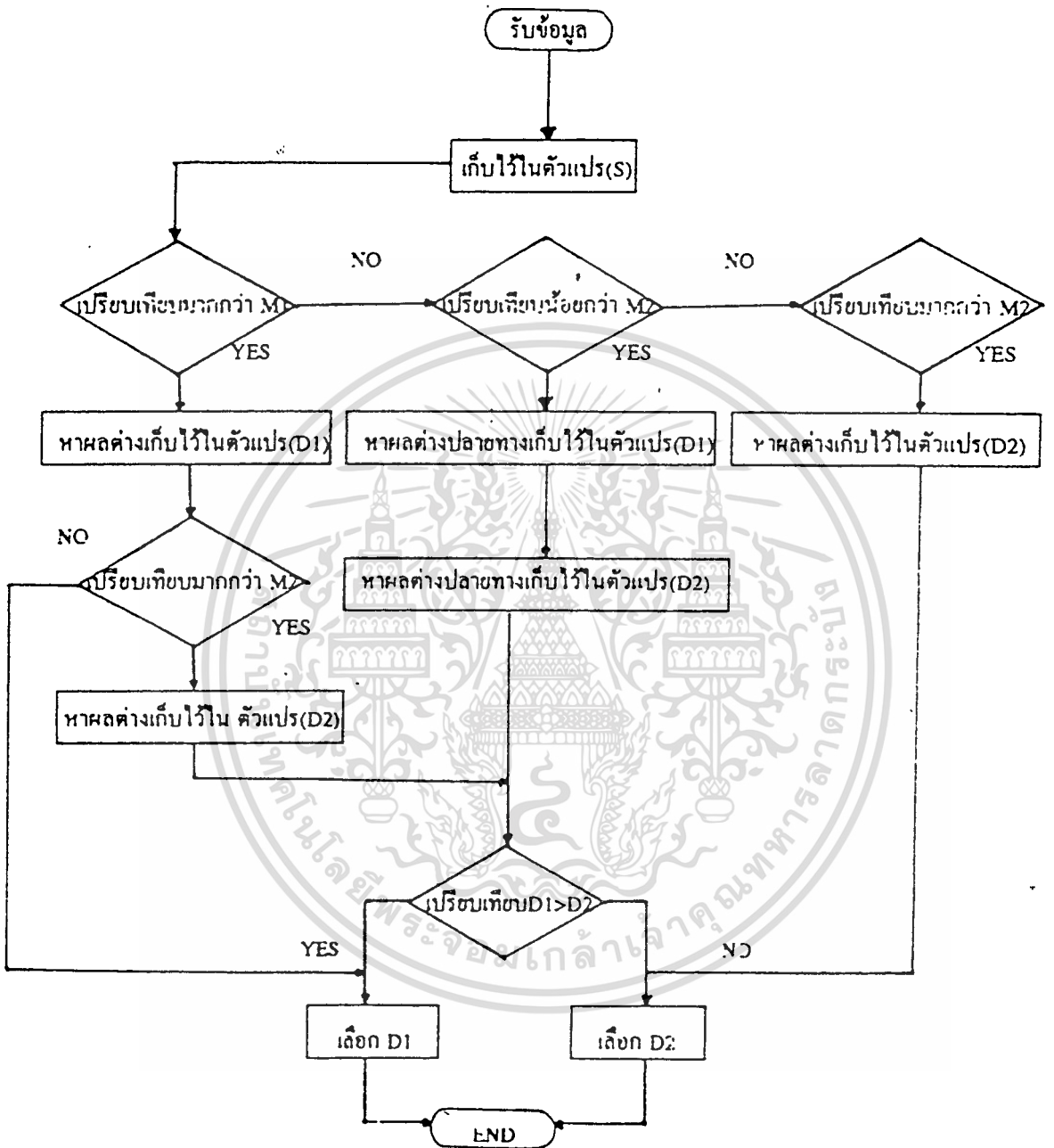
วงจรเซนเซอร์



รูป 2.6 SENSOR CIRCUIT

จากการทดสอบการ Bias ค่า R ที่ต่อกับ MLED81 หลาย ๆ ค่าพบว่า R ที่เหมาะสมคือ 100 โอห์ม จะได้ระยะทางประมาณ 1 นิ้ว ซึ่งเพียงพอต่อความต้องการของเรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 Note: ถ้าเราใช้ค่า R น้อยกว่านี้จะทำให้ MLED81 ร้อนได้ ซึ่งไม่ปลอดภัยกับตัวอุปกรณ์
 แม้ว่าการันเต้ทั้งสองอัน อีกทั้งห้ามมิให้ตัดแบบลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 ถึงแม้ว่าจะได้ระยะทางมากขึ้นก็ตาม



รูปที่ 2.7 แสดง FLOW CHART การจัด PRIORITY

การกำหนดค่า INPUT OUTPUT

ส่วน 8255 (PIO) เลือกโหมด 0

PORT A รับ INPUT จาก SWITCH UP-DOWN

- PA 0 ต่อกับ สัญญาณ SWITCH 1 UP
- PA 1 ต่อกับ สัญญาณ SWITCH 2 UP
- PA 2 ต่อกับ สัญญาณ SWITCH 2 DOWN
- PA 3 ต่อกับ สัญญาณ SWITCH 3 UP
- PA 4 ต่อกับ สัญญาณ SWITCH 3 DOWN
- PA 5 ต่อกับ สัญญาณ SWITCH 4 UP
- PA 6 ต่อกับ สัญญาณ SWITCH 4 DOWN
- PA 7 ต่อกับ สัญญาณ SWITCH 5 DOWN

PORT B รับ INPUT จาก NO.SWITCH LIFT 1/2

- PB 0 ต่อกับ สัญญาณ NO.1 SWITCH LIFT 1
- PB 1 ต่อกับ สัญญาณ NO.2 SWITCH LIFT 1
- PB 2 ต่อกับ สัญญาณ NO.3 SWITCH LIFT 1
- PB 3 ต่อกับ สัญญาณ NO.4 SWITCH LIFT 1
- PB 4 ต่อกับ สัญญาณ NO.5 SWITCH LIFT 1
- PB 5 ต่อกับ สัญญาณ NO.1 SWITCH LIFT 2
- PB 6 ต่อกับ สัญญาณ NO.2 SWITCH LIFT 2
- PB 7 ต่อกับ สัญญาณ NO.3 SWITCH LIFT 2

PORT C รับ INPUT จาก NO.SWITCH LIFT 2 และ SENSOR 1/2

OUTPUT สัญญาณ CONTROL MOTOR 1/2

- PC 0 ต่อกับ สัญญาณ CONTROL MOTOR 1.1
- PC 1 ต่อกับ สัญญาณ CONTROL MOTOR 1.2
- PC 2 ต่อกับ สัญญาณ CONTROL MOTOR 2.1
- PC 3 ต่อกับ สัญญาณ CONTROL MOTOR 2.2
- PC 4 ต่อกับ สัญญาณ NO.4 SWITCH LIFT 2
- PC 5 ต่อกับ สัญญาณ NO.5 SWITCH LIFT 2

PC 6 ต่อกับ สัญญาณ SENSOR 1

PC 7 ต่อกับ สัญญาณ SENSOR 2

บทที่ 8

การทดลองและผลการทดลอง

8.1 การทดลองที่ 1

จุดประสงค์

เพื่อทดสอบการไคร์ฟมอเตอร์ให้หมุน ขึ้น ลง และหยุด

อุปกรณ์

1. เครื่องกำเนิดสัญญาณกระแสตรง 12 โวลต์
2. ชุดวงจรไคร์ฟมอเตอร์
3. มอเตอร์กระแสตรง 12 โวลต์ และ 5 โวลต์

วิธีทดลอง

1. ต่ออุปกรณ์ต่าง ๆ เข้าด้วยกัน
2. ทดลองจ่ายค่าอินพุตด้วยสัญญาณไฟ 5 โวลต์

ผลการทดลอง

เมื่อลองจ่ายค่า Director Control Signal ด้วยสัญญาณ 5 โวลต์กับ 0 โวลต์ ผลปรากฏว่า มอเตอร์หมุนขึ้นและเมื่อจ่ายด้วยสัญญาณ 5 โวลต์ กับ 5 โวลต์ ผลปรากฏว่ามอเตอร์หมุนลง

8.2 การทดลองที่ 2

จุดประสงค์

เพื่อทดสอบโปรแกรมการใช้งาน

อุปกรณ์

1. ชุดรับ-ส่งสัญญาณอินฟราเรด
2. ชุดคอนโทรลมอเตอร์
3. เครื่องกำเนิดสัญญาณกระแสตรง 12 โวลต์ และ 5 โวลต์
4. ลิฟต์จำลอง
5. ไมโครคอนโทรลเลอร์ ชุด ANT-32

วิธีทดลอง

1. ต่ออุปกรณ์ต่าง ๆ เข้าด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น ยกเว้นให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ทดลอง run program
3. ทดลองกดสวิทช์เรียกลิฟต์ไปยังที่ต่าง ๆ

ผลการทดลอง



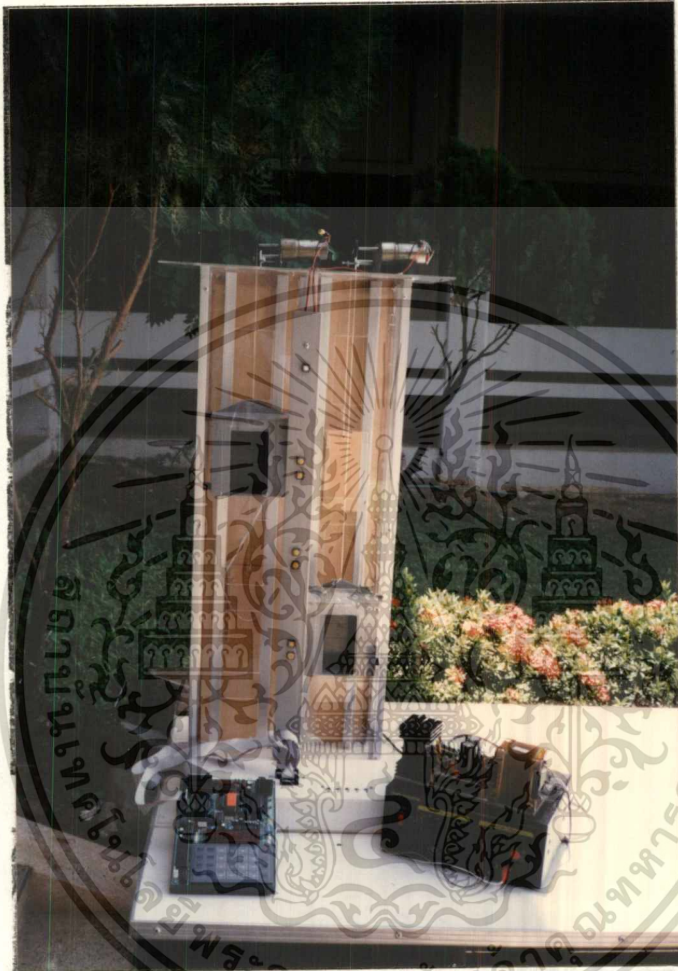
รูปที่ 3.1 แสดงโครงสร้างลิฟต์จำลองและวงจรควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงการเรียกใช้ลิตไฟไปยังชั้นที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงการเคลื่อนที่ของลิฟต์มายังชั้นที่เรียก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงการเคลื่อนที่ขึ้น-ลง ของลิฟต์พร้อมกันทั้ง 2 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



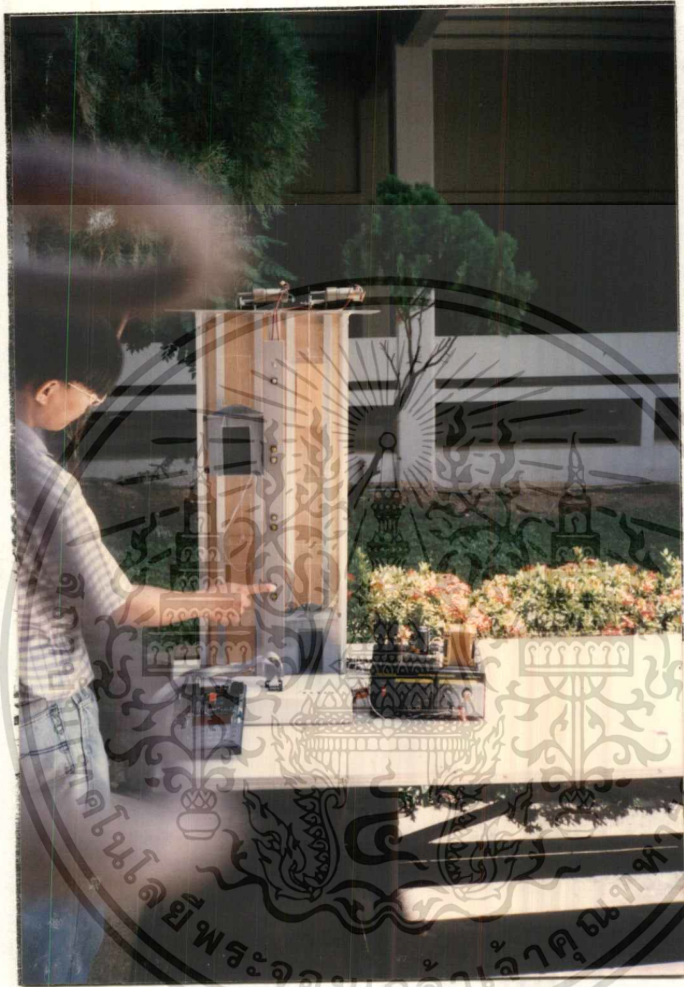
รูปที่ 3.5 แสดงการเรียกลิฟต์ที่ชั้นบนสุดเพื่อต้องการลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดงการเคลื่อนที่ของลิฟต์มายังชั้นที่เรียก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แสดงการเรียกใช้ลิฟต์ขณะที่ลิฟต์อีกตัวกำลังเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงการเคลื่อนที่ของลิฟต์ตามคำสั่งเรียกในรูปที่ 3.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

บทวิจารณ์และบทสรุป

4.1 บทวิจารณ์

ในโครงการนี้จะเน้นถึงการเคลื่อนที่ของลิฟต์ไปยังจุดหมายปลายทางด้วยวิธีที่ประหยัดเวลาที่สุดโดยอาศัยการประมวลผลจากโปรแกรม ที่เขียนขึ้นมาจากไมโครคอนโทรลเลอร์ 8051 ดังนั้นรายละเอียดของระบบลิฟต์ในส่วนของประตูและระบบเซฟต์เบรกจึงขาดหายไปเนื่องมาจากเป็นระบบที่ยุ่งยากและซับซ้อนมากเป็นข้อจำกัดของผู้จัดทำจึงไม่สามารถสร้างระบบจำลองขึ้นมาได้

4.2 สรุป

ในโครงการนี้เป็นเพียงแนวทางในการประยุกต์ใช้ไมโครคอนโทรลเลอร์ร่วมกับระบบลิฟต์จำลอง โดยการโปรแกรมภาษาแอสเซมบลีเพื่อที่จะจัดการการทำงานในส่วนต่าง ๆ ให้ได้รับความสะดวกและรวดเร็ว โดยเน้นในส่วนของการเคลื่อนที่ ซึ่งยังต้องได้รับการพัฒนาให้สามารถทำงานได้อย่างมีประสิทธิภาพและใกล้เคียงกับระบบลิฟต์ที่ใช้งานตามมาตรฐาน

กิติกรรมประกาศ

ในวิทยานิพนธ์ฉบับนี้ได้สำเร็จลุล่วงไปได้ด้วยดีตามวัตถุประสงค์เพราะได้รับความช่วยเหลือเป็นอย่างดีจากอาจารย์ที่ปรึกษา อาจารย์วิทยา ทิพย์สุวรรณพร ที่ช่วยอำนวยความสะดวก ทั้งในด้านอุปกรณ์และคำแนะนำที่มีประโยชน์มากมายจนกระทั่งผลงานชิ้นนี้สำเร็จและขอขอบคุณ คุณเชิดชาย บุญประเสริฐ ที่ให้ความรู้และคำแนะนำตลอดจนเพื่อน ๆ ที่มีส่วนช่วยเหลือทำให้วิทยานิพนธ์ชิ้นนี้สำเร็จลุล่วงไว้ ณ โอกาสนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

portcc equ 0Fc03h
porta equ 0Fc00h
portb equ 0Fc01h
portc equ 0Fc02h
command equ 10011010b
;***** ram define byte *****
direct1 equ 30h ; direction of lift1
direct2 equ 31h

pos1 equ 32h ; postion of lift1
pos2 equ 33h
;***** ram define bit *****
up1 equ 0h ; define 1 stair
up2 equ 1h ; define 2 stair
up3 equ 2h ; define 3 stair
up4 equ 3h ; define 4 stair
down2 equ 4h ; define 2 stair
down3 equ 5h ; define 3 stair
down4 equ 6h ; define 4 stair
down5 equ 7h ; define 5 stair

flag equ 8h

f1 equ 10h
f2 equ 11h
f3 equ 12h
f4 equ 13h
f5 equ 14h

g1 equ 15h
g2 equ 16h
g3 equ 17h
g4 equ 18h
g5 equ 19h

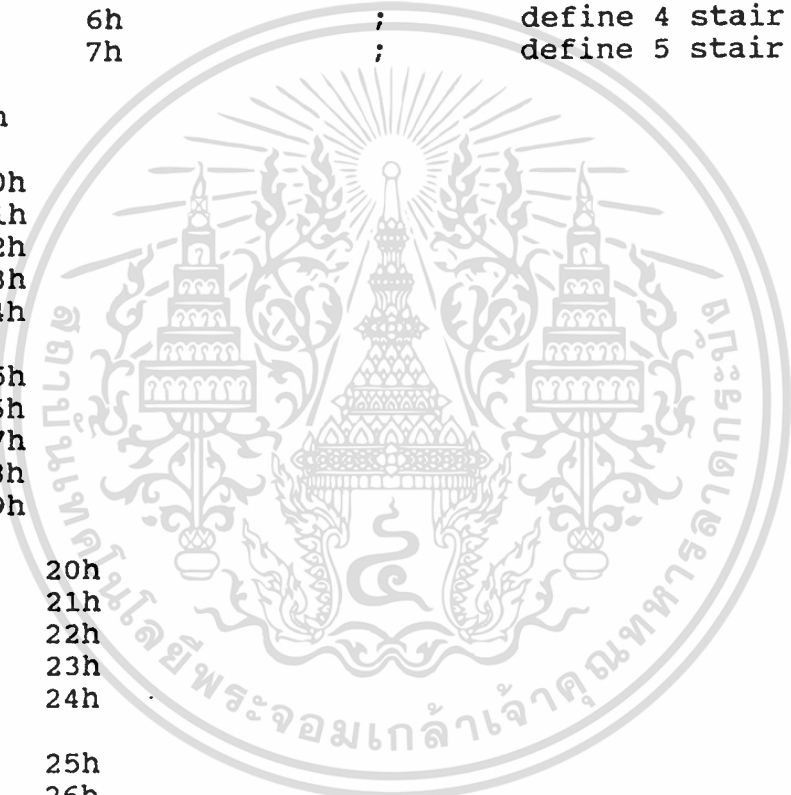
map1 equ 20h
map2 equ 21h
map3 equ 22h
map4 equ 23h
map5 equ 24h

map6 equ 25h
map7 equ 26h
map8 equ 27h
map9 equ 28h
map10 equ 29h

switch equ 30h ;special bit
upbit equ 31h
downbit equ 32h
stopbit equ 33h
movedown equ 34h
moveup equ 35h
up equ 11b
down equ 01b
stop equ 00b

;***** verify all define

```



```

mov p1,#00h
lcall set
lcall delay

lcall init_8255

lcall start
lcall delay
;*****8
    mov pos1,#1
    setb stopbit
    clr upbit
    clr downbit
    setb flag

```

```

loop:  lcall scan_key
;      lcall map           ; this routine can set ... can't reset
lcall check_flg ; verify      lcall check_flg
lcall walk
lcall walk2
;      sjmp loop

```

```

scan_key:  mov dptr,#porta
           movx a,@dptr
           mov p1,a
           jb acc.7,up_1           ; if acc.0 = 1 , push 1 up
           setb map1
up_1:     jb acc.6,down_2
           setb map7
down_2:   jb acc.5,up_2
           setb map2
up_2:     jb acc.4,down_3
           setb map8
down_3:   jb acc.3,up_3
           setb map3
up_3:     jb acc.2,down_4
           setb map9
down_4:   jb acc.1,up_4
           setb map4
up_4:     jb acc.0,down_5
           setb map10

```

```

down_5:   mov dptr,#portb
           movx a,@dptr
lt1_1:    jb acc.0,lt1_2
           setb map1
lt1_2:    jb acc.1,lt1_3
           setb map2
lt1_3:    jb acc.2,lt1_4
           setb map3
lt1_4:    jb acc.3,lt1_5
           setb map4
lt1_5:    jb acc.4,lt2_1
           setb map5
lt2_1:    jb acc.5,lt2_2
           setb g1
lt2_2:    jb acc.6,lt2_3
           setb g2
lt2_3:    jb acc.7,lt2_4
           setb g3

```

นี่เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lt2_4:      ret
:
      lcall delay
      mov dptr,#portc
      movx a,@dptr
:
      jb acc.4,lt2_5
          setb g4
lt2_5:      jb acc.5,lt2_6
          setb g5
lt2_6:      ; jb acc.6,sense_1
lt2_7:      ; jb acc.7,sense_2
lt2_8:
end_scan_key:      ret
;*****----- check_flr -----
check_flr:      mov dptr,#portc      ; check sensor if sensore det
                movx a,@dptr      ; so increase pos=pos+1
                jb acc.6,outsensor
                    jb flag,end_check_flr
                        setb flag
                        jb upbit,increase
                        dec pos1
                        sjmp end_check_flr
increase:      inc pos1
dada:          sjmp end_check_flr
outsensor:    clr flag
end_check_flr:      ret
;*****-----end check_flr -----
;-----
;checkend_temp:      ljmp checkend      ; make tempolary jump to che
walk:          jb stopbit,checkend_temp
                jnb upbit,check_movedown
:
                jnb map5,asdf1
                mov a,#5
                cjne a,pos1,endmap
                setb stopbit
                clr upbit
asdf1:        jnb map4,asdf2
                mov a,#4
                cjne a,pos1,endmap
                setb stopbit
                clr upbit
asdf2:        jnb map3,asdf3
                mov a,#3
                cjne a,pos1,endmap
                setb stopbit
                clr upbit
asdf3:        jnb map2,asdf4
                mov a,#2
                cjne a,pos1,endmap
                setb stopbit
                clr upbit
asdf4:        ;jnb map1,asdf5
                mov a,#1
                ; cjne a,pos1,endmap

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และใช้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; setb stopbit
; clr upbit
asdf5:      ljmp endmap

check_movedown:  jnb movedown,checkend

                jnb map1,sdf1
                mov a,#1
                cjne a,pos1,endmap
                setb stopbit
                clr upbit
                clr downbit
sdf1:        jnb map2,sdf2
                mov a,#2
                cjne a,pos1,endmap
                setb stopbit
                clr upbit
                clr downbit
sdf2:        jnb map3,sdf3
                mov a,#3
                cjne a,pos1,endmap
                setb stopbit
                clr upbit
                clr downbit
sdf3:        jnb map4,sdf4
                mov a,#4
                cjne a,pos1,endmap
                setb stopbit
                clr downbit
                clr upbit
sdf4:        ; jnb map5,sdf5
                ; mov a,#5
                ; cjne a,pos1,endmap
                ; setb stopbit
                ; clr downbit

sdf5:        nop
endmap:      nop
checkend:    nop
;***** Start sample space Table *****

; sjmp now_pos1 ; temporary jump make blank
now_pos2_temp:  ljmp now_pos2
now_pos1:      mov a,#01h ; compare pos1 with A = 1

                cjne a,pos1,now_pos2_temp
                jnb map1,aaa1
                clr map1
                clr upbit
                setb stopbit
                ljmp endwalk
aaa1:          jnb map2,aaa2
                clr stopbit
                setb upbit
                ljmp endwalk
aaa2:          jnb map3,aaa3
                clr stopbit
                setb upbit
                ljmp endwalk
aaa3:          jnb map4,aaa4
                clr stopbit

```

นี่เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setb upbit
      ljmp endwalk
aaa4: jnb map5,aaa5
      clr stopbit
      setb upbit
      ljmp endwalk
aaa5: jnb map6,aaa6 ; not occur
      clr map6
      ljmp endwalk
aaa6: jnb map7,aaa7
      clr stopbit
      setb upbit
      ljmp endwalk
aaa7: jnb map8,aaa8
      clr stopbit
      setb upbit
      ljmp endwalk
aaa8: jnb map9,aaa9
      clr stopbit
      setb upbit
      ljmp endwalk
aaa9: jnb map10,aaa10
      clr stopbit
      setb upbit
      ljmp endwalk
daa10: nop
;***** end lift 1 stop *****
;***** Start sample pos 2 *****
now_pos3_temp: sjmp now_pos2 ; tempolary jump make blank
now_pos2:      ljmp now_pos3
              mov a,#02h ; compare pos1 with A = 1
              cjne a,pos1,now_pos3_temp
              jnb map1,ddd1
              clr upbit
              clr stopbit
              setb downbit
              jmp endwalk
ddd1: jnb map2,ddd2
      clr map2
      clr upbit
      clr downbit
      setb stopbit
      jmp endwalk
ddd2: jnb map3,ddd3
      clr stopbit
      setb upbit
      jmp endwalk
ddd3: jnb map4,ddd4
      clr stopbit
      setb upbit
      jmp endwalk
ddd4: jnb map5,ddd5
      clr stopbit
      setb upbit
      jmp endwalk
ddd5: jnb map6,ddd6 ; not occur
      clr map6
      setb stopbit

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        clr downbit
        clr stopbit
        jmp endwalk
ddd6:   jnb map7,ddd7
        clr map7
        setb stopbit
        clr upbit
        clr downbit
        jmp endwalk
ddd7:   jnb map8,ddd8
        clr stopbit
        setb upbit
        jmp endwalk
ddd8:   jnb map9,ddd9
        clr stopbit
        setb upbit
        jmp endwalk
ddd9:   jnb map10,ddd10
        clr stopbit
        setb upbit
        jmp endwalk
ddd10:  jmp endwalk

;***** end lift 2 stop *****

;***** now pos 3 start *****
;***** Start sample pos 3 *****
;
sjmp    now_pos3      ; temporary jump make blank
now_pos4_temp:  ljmp    now_pos4
now_pos3:  mov a,#03h   ; compare pos1 with A = 1
           cjne a,pos1,now_pos4_temp

           jnb map1,ggg1
           clr upbit
           clr stopbit
           setb downbit
           jmp endwalk
ggg1:   jnb map2,ggg2
           clr upbit
           clr stopbit
           setb downbit
           jmp endwalk
ggg2:   jnb map3,ggg3
           clr map3
           setb stopbit
           clr upbit
           clr downbit
           jmp endwalk
ggg3:   jnb map4,ggg4
           clr stopbit
           setb upbit
           jmp endwalk
ggg4:   jnb map5,ggg5
           clr stopbit
           setb upbit
           jmp endwalk
ggg5:   jnb map6,ggg6      ; not occur
           clr stopbit
           clr map6
           jmp endwalk

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ggg6:      jnb map7,ggg7
           clr upbit
           clr stopbit
           setb downbit
           jmp endwalk
ggg7:      jnb map8,ggg8
           clr map8
           setb stopbit
           clr downbit
           clr upbit
           jmp endwalk
ggg8:      jnb map9,ggg9
           clr stopbit
           setb upbit
           jmp endwalk
ggg9:      jnb map10,ggg10
           clr stopbit
           setb upbit
           jmp endwalk
ggg10:     jmp endwalk

;*****      end lift 3 stop *****

;*****      now pos 4 start *****
;*****      Start sample pos 4 *****

now_pos5_temp:  sjmp  now_pos4      ; tempolary jump make blank
now_pos4:      ljmp  now_pos5
now_pos4:      mov  a,#04h      ; compare pos1 with A = 1
               cjne a,pos1,now_pos5_temp
               ;
               jnb  map1,hhh1
               clr  upbit
               clr  stopbit
               setb downbit
               jmp  endwalk
hhh1:         jnb  map2,hhh2
               clr  upbit
               clr  stopbit
               setb downbit
               jmp  endwalk
hhh2:         jnb  map3,hhh3
               clr  upbit
               clr  stopbit
               setb downbit
               jmp  endwalk
hhh3:         jnb  map4,hhh4
               clr  upbit
               setb stopbit
               clr  downbit
               clr  map4
               jmp  endwalk
hhh4:         jnb  map5,hhh5
               clr  stopbit
               setb upbit
               jmp  endwalk
hhh5:         jnb  map6,hhh6      ; not occur
               setb stopbit
               setb downbit
               clr  map6
               jmp  endwalk

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

hhh6:      jnb map7, hhh7
           clr upbit
           clr stopbit
           setb downbit
           jmp endwalk
hhh7:      jnb map8, hhh8
           clr upbit
           clr stopbit
           setb downbit
           jmp endwalk
hhh8:      jnb map9, hhh9
           setb stopbit
           clr upbit
           clr downbit
           clr map9
           jmp endwalk
hhh9:      jnb map10, hhh10
           clr stopbit
           setb upbit
           jmp endwalk
hhh10:     jmp endwalk

;***** end lift 4 stop *****

;***** now pos 5 start *****
;***** Start sample pos 5 *****

endwalk_temp:  sjmp now_pos5 ; temporary jump make blank
               ljmp endwalk
now_pos5:     mov a, #05h ; compare pos1 with A = 1
               cjne a, pos1, endwalk_temp
               clr upbit
               jnb map1, kkk1
               clr stopbit
               setb downbit
               jmp endwalk
kkk1:        jnb map2, kkk2
               clr stopbit
               setb downbit
               jmp endwalk
kkk2:        jnb map3, kkk3
               clr stopbit
               setb downbit
               jmp endwalk
kkk3:        jnb map4, kkk4
               clr stopbit
               setb downbit
               jmp endwalk
kkk4:        jnb map5, kkk5
               clr map5
               setb stopbit
               clr upbit
               clr downbit
               jmp endwalk
kkk5:        jnb map6, kkk6 ; not occur
               clr stopbit
               setb downbit
               clr upbit

```

```

                                jmp endwalk
kkk6:    jnb map7, kkk7
                                clr stopbit
                                setb downbit
                                clr upbit
                                jmp endwalk
kkk7:    jnb map8, kkk8
                                clr stopbit
                                clr upbit
                                setb downbit
                                jmp endwalk
kkk8:    jnb map9, kkk9
                                clr stopbit
                                clr upbit
                                setb downbit
                                jmp endwalk
kkk9:    jnb map10, kkk10
                                clr map10
                                setb stopbit
                                clr upbit
                                clr downbit
kkk10:   jmp endwalk

endwalk: ret
;*****
walk2:   jnb stopbit, check_move ; check stop bit
                                mov a, #stop ; stop motor 1 code
                                sjmp end_walk2
check_move: jnb upbit, motor_down ; check move up bit
motor_up:  mov a, #up ; move motor 1 up
                                sjmp end_walk2
motor_down: mov a, #down ; move motor 1 down

end_walk2: mov dptr, #portc ; now a store code to move
                                movx @dptr, a
                                ret
;-----

init_8255: mov dptr, #portcc
                                mov a, #command
                                movx @dptr, a
                                ret

set:     mov pos1, #01 ; start position lift
                                setb stopbit ; start direction stop
                                clr upbit
                                clr downbit
                                mov r1, #15
                                mov r0, #20h
set_zero: mov @r0, #0
                                inc r0
                                djnz r1, set_zero
                                ret

delay:   mov r0, #0ffh
qqqq:   mov r1, #0ffh
pppp:   nop
                                djnz r1, pppp
                                djnz r0, qqqq

```

ret

***** main map *****

```
map:      jnb up1,ch1
          setb map1
ch1:     jnb up2,ch2
          setb map2
ch2:     jnb up3,ch3
          setb map3
ch3:     jnb up4,ch4
          setb map4
ch4:     jnb down2,ch5
          setb map7
ch5:     jnb down3,ch6
          setb map8
ch6:     jnb down4,ch7
          setb map9
ch7:     jnb down5,ch8
          setb map10
ch8:     jnb f1,ch9
          setb map1
ch9:     jnb f2,ch10
          setb map2
ch10:    jnb f3,ch11
          setb map3
ch11:    jnb f4,ch12
          setb map4
ch12:    jnb f5,ch13
          setb map5
ch13:    jnb g1,ch14
ch14:    jnb g2,ch15
ch15:    jnb g3,ch16
ch16:    jnb g4,ch17
ch17:    jnb g5,ch18
ch18:    ret
```

; for lift 2 not done

***** END MAP *****

```
start:   mov dptr,#portc
          mov a,#down
          movx @dptr,a
          movx a,@dptr
          jb acc.6,start
          mov a,#stop
          movx @dptr,a
          ret
```

```
dscan_key: mov a,#stop
            mov dptr,#portc
            movx @dptr,a

            mov r0,#03ah
```

```
temp:    mov dptr,#porta
            movx a,@dptr
            mov pl,a
            jnb acc.7,dup_1
            setb map1
```

; if acc.0 = 1 , push 1 up

```
dup_1:   jnb acc.6,ddown_2
            setb map2
```

คู่มือนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ddown_2:    jb acc.5,dup_2
            setb map2
dup_2:      jb acc.4,ddown_3
            setb map8
ddown_3:    jb acc.3,dup_3
            setb map3
dup_3:      jb acc.2,ddown_4
            setb map9
ddown_4:    jb acc.1,dup_4
            setb map4
dup_4:      jb acc.0,ddown_5
            setb map10

ddown_5:    mov dptr,#portb
            movx a,@dptr
dlt1_1:     jb acc.0,dlt1_2
            setb map1
dlt1_2:     jb acc.1,dlt1_3
            setb map2
dlt1_3:     jb acc.2,dlt1_4
            setb map3
dlt1_4:     jb acc.3,dlt1_5
            setb map4
dlt1_5:     jb acc.4,dlt2_1
            setb map5
dlt2_1:     jb acc.5,dlt2_2
            setb g1
dlt2_2:     jb acc.6,dlt2_3
            setb g2
dlt2_3:     jb acc.7,dlt2_4
            setb g3
dlt2_4:     djnz r0,temp
            jb upbit,upup
down:       mov a,#up
            mov dptr,#portc
            movx @dptr,a
            sjmp end_delay
upup:       mov a,#down
            mov dptr,#portc
            movx @dptr,a

end_delay:  ret

end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

Advance Information
Digital Output Detector

Features:

- Popular Low Cost Plastic Package
- High Coupling Efficiency
- Wide V_{CC} Range
- Ideally Suited for MLED91 Emitter
- Usable to 125 KHz
- Open Collector Output
- Compatible with 3 Volt Systems
- New Mold Technology Improves Performance Under Variable Environmental Conditions
- New Lens Design Offers Improved Optical Performance
- EIA 468-A Compliant Tape and Reel Option Available (MRD950RLRE)

Applications:

- IR Remote Control Receiver
- Shaft Encoders
- Position Sensors
- Interruptive Sensors

MRD950

Motorola Preferred Device

PHOTO DETECTOR
SCHMITT TRIGGER
OUTPUT



CASE 422-01
Style 3

MAXIMUM RATINGS (T_A = 25°C unless otherwise noted)

Rating	Symbol	Value	Unit
Output Voltage Range	V _O	0 - 16	Volts
Supply Voltage Range	V _{CC}	3 - 16	Volts
Output Current	I _O	50	mA
Device Dissipation Derate above 25°C (1)	P _D	150 2.0	mW mW/°C
Maximum Operating Temperature	T _A	-40 to +100	°C
Storage Temperature Range	T _{stg}	-40 to +100	°C
Lead Soldering Temperature (5 Seconds Maximum) (2)	T _L	260	°C

(1) Measured with device soldered into a typical PC board.

DEVICE CHARACTERISTICS (T_A = 25°C)

Characteristic	Symbol	Min	Typ	Max	Unit
Operating Voltage	V _{CC}	3.0	—	16	Volts
Supply Current with Output High, Figure 4 (I _F = 0, V _{CC} = 5.0 V)	I _{CC(oh)}	—	1.0	5.0	mA
Output Current, High (I _F = 0, V _{CC} = V _O = 1.5 V, R _L = 270 Ω)	I _{OH}	—	—	100	μA

(1) Measured with device soldered into a typical printed circuit board.

(2) Maximum exposure time: five seconds. Minimum of 1/16 inch from the case. A heat sink should be applied in order to prevent the case temperature from exceeding 100°C.

This document contains information on a new product. Specifications and information herein are subject to change without notice. Preferred devices are Motorola recommended choices for future use and best overall value.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MRD950

COUPLED CHARACTERISTICS ($T_A = 0 - 70^\circ\text{C}$)

Characteristic	Symbol	Min	Typ	Max	Unit
Light Required to Trigger (Tungsten Source, 2870 K)	$H_{(on)}$	—	0.50	—	mW/cm^2

The following characteristics are measured with an MLED91 emitter at a separation distance of 4.0 mm (0.155 inches) with the lenses of the emitter and detector on a common axis within 0.1 mm and parallel within 5 degrees.

Supply Current with Output Low, Figure 5 ($I_F = I_{F(on)}$, $V_{CC} = 5.0\text{ V}$)	$I_{CC(on)}$	—	1.6	5.0	mA
Output Voltage, Low ($R_L = 270\ \Omega$, $V_{CC} = 5.0\text{ V}$, $I_F = I_{F(on)}$)	V_{OL}	—	0.2	0.4	Volts
Threshold Current, ON ($R_L = 270\ \Omega$, $V_{CC} = 5.0\text{ V}$)	$I_{F(on)}$	—	10	20	mA
Threshold Current, OFF ($R_L = 270\ \Omega$, $V_{CC} = 5.0\text{ V}$)	$I_{F(off)}$	1.0	7.5	—	mA
Hysteresis Ratio, Figure 1 ($R_L = 270\ \Omega$, $V_{CC} = 5.0\text{ V}$)	$\frac{I_{F(off)}}{I_{F(on)}}$	—	0.75	—	—

SWITCHING CHARACTERISTICS ($T_A = 25^\circ\text{C}$)

Turn-On Time	$R_L = 270\ \Omega$ $V_{CC} = 5.0\text{ V}$ $I_F = I_{F(on)}$	t_{on}	—	0.75	5.0	μs
Fall Time		t_f	—	0.1	—	
Turn-Off Time		t_{off}	—	2.0	5.0	
Rise Time		t_r	—	0.1	—	

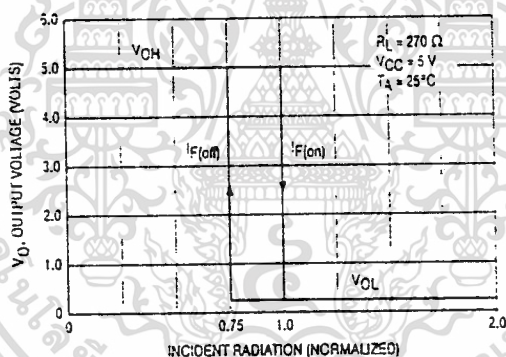


Figure 1. Transfer Characteristics

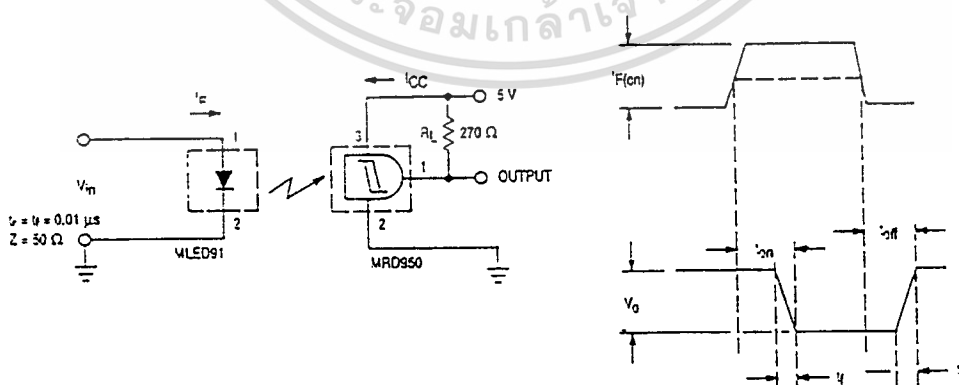


Figure 2. Switching Test Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MRD950

TYPICAL CHARACTERISTICS

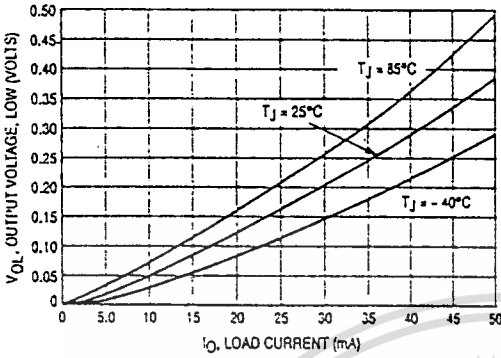


Figure 3. Output Voltage, Low versus Load Current

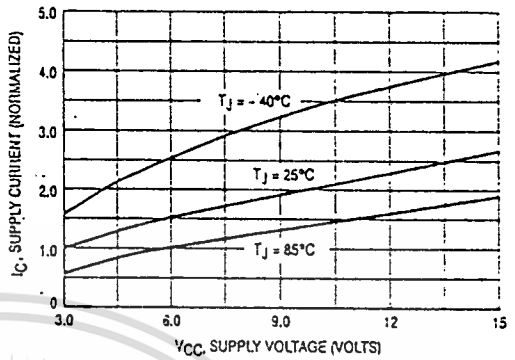


Figure 4. Supply Current versus Supply Voltage — Output High

TYPICAL COUPLED CHARACTERISTICS USING MLED91 EMITTER AND MRD950 DIGITAL OUTPUT DETECTOR

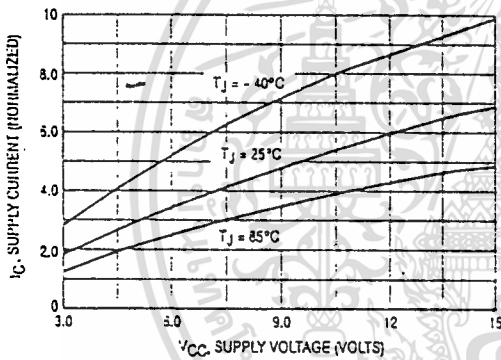


Figure 5. Supply Current versus Supply Voltage — Output Low

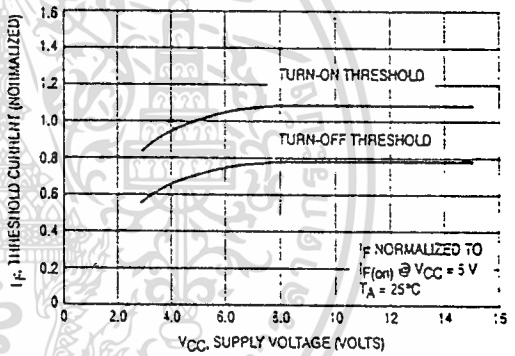


Figure 6. Threshold Current versus Supply Voltage

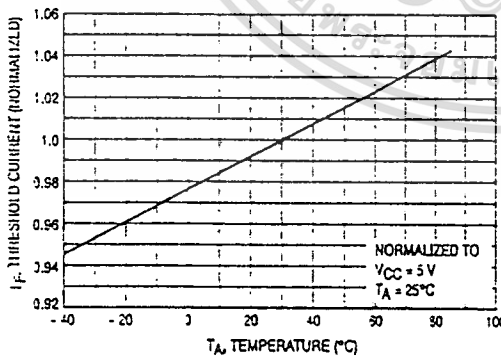


Figure 7. Threshold Current versus Temperature

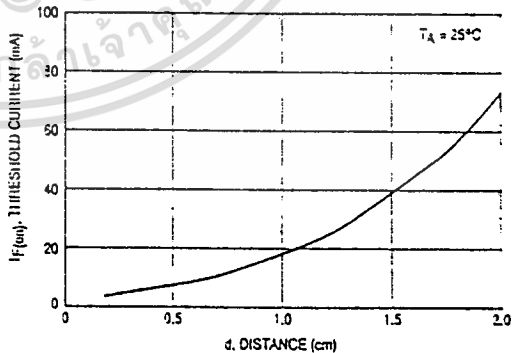


Figure 8. Threshold Current versus Lens to Lens Separation Distance at 25°C

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

Infrared LED

Features:

- Low Cost
- Popular T-1 $\frac{1}{4}$ Package
- Ideal Beam Angle for Most Remote Control Applications in Conjunction with MRD821
- Uses Stable Long-Life LED Technology
- Clear Epoxy Package

Applications:

Remote Controls and Long Distance Interruptive Sensing

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Reverse Voltage	V_R	5	Volts
Forward Current — Continuous	I_F	100	mA
Forward Current — Peak Pulse	I_F	1	A
Total Power Dissipation ($T_A = 25^\circ\text{C}$ Derate above 25°C)	P_D	100 2.2	mW mW/°C
Ambient Operating Temperature Range	T_A	-30 to -70	°C
Storage Temperature	T_{stg}	-30 to -80	°C
Lead Soldering Temperature, 5 seconds max, 1/16 inch from case	—	260	°C

ELECTRICAL CHARACTERISTICS ($T_A = 25^\circ\text{C}$ unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Reverse Leakage Current ($V_R = 3\text{ V}$)	I_R	—	10	—	nA
Reverse Leakage Current ($V_R = 5\text{ V}$)	I_R	—	1	10	μA
Forward Voltage ($I_F = 100\text{ mA}$)	V_F	—	1.35	1.7	V
Temperature Coefficient of Forward Voltage	ΔV_F	—	-1.6	—	mV/K
Capacitance ($f = 1\text{ MHz}$)	C	—	25	—	pF

OPTICAL CHARACTERISTICS ($T_A = 25^\circ\text{C}$ unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Peak Wavelength ($I_F = 100\text{ mA}$)	λ_p	—	940	—	nm
Spectral Half-Power Bandwidth	$\Delta\lambda$	—	50	—	nm
Total Power Output ($I_F = 100\text{ mA}$)	P_e	—	16	—	mW
Temperature Coefficient of Total Power Output	ΔP_e	—	-0.25	—	%/K
Axial Radiant Intensity ($I_c = 100\text{ mA}$)	I_a	10	15	—	mW/sr
Temperature Coefficient of Axial Radiant Intensity	ΔI_a	—	-0.25	—	%/K
Power Half-Angle	θ	—	≈ 30	—	°

MLED81

Motorola Preferred Device

INFRARED
LED
940 nm



CASE 279B-01
STYLE 1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MLED81

TYPICAL CHARACTERISTICS

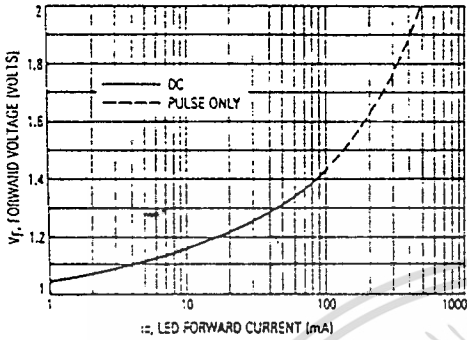


Figure 1. LED Forward Voltage versus Forward Current

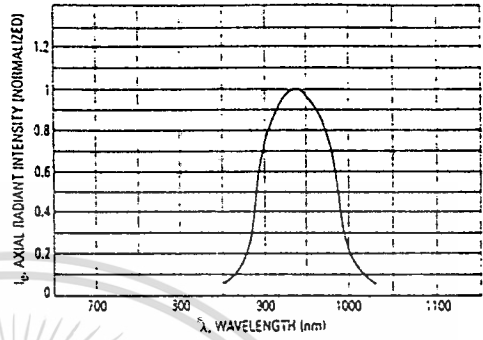


Figure 2. Relative Spectral Emission

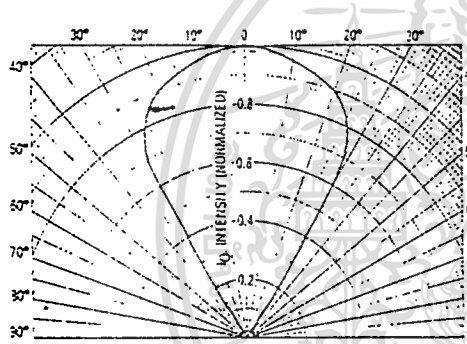


Figure 3. Spatial Radiation Pattern

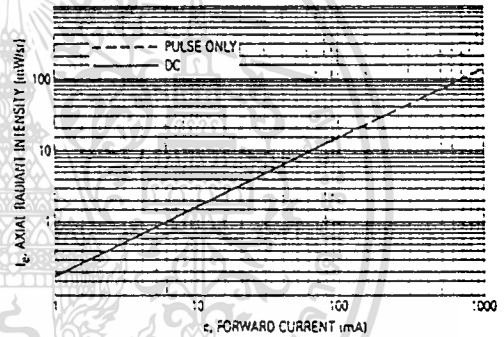


Figure 4. Intensity versus Forward Current

บรรณานุกรม

1. Fred A. Annet , “Elevator “ , McGraw - Hill Copany , 388 p . , 1960 .
2. G.C. Barney , “ Elavator Technology “ , International Association of Elevator Engineers / Ellis Horwood Limited , 354 p . , 1986 .
3. G. C. Barney and S . M . dos Santos , “ Elevator Traffic Analysis Design and Contorl “ , Peter Pergrinus Ltd . , 386 p . , 1985 .
4. คู่มือและการประยุกต์ใช้งานไมโครคอนโทรลเลอร์ MCS - 51 บทที่ 20 หน้าที่ 292
5. การโปรแกรมภาษาแอสเซมบลีของไมโครคอนโทรลเลอร์ตระกูล 8051 บทที่ 9 หน้าที่ 147 - 148
6. MOTOROLA Optoelectronics Device Data 7 - 3 , 7 - 35

