



การควบคุมระบบลิฟต์ด้วยไมโครคอนโทรลเลอร์  
Controlling Elevator System With Microcontroller

นายอดิเรก ปั่นมี รหัส 45380157  
นายกฤษฏา เล็งเรียบ รหัส 45380259

ห้องสมุดคณะวิศวกรรมศาสตร์  
วันที่รับ..... 15 พ.ย. 2549 .....

เลขทะเบียน..... Δ900170 .....

เลขเรียกหนังสือ.....

มหาวิทยาลัยนเรศวร

i507566x  
ร.ร.  
01270  
2548  
e.2

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2548



## ใบรับรองโครงการวิศวกรรม

|                      |   |                  |               |
|----------------------|---|------------------|---------------|
| หัวข้อโครงการ        | การควบคุมระบบลิฟต์ด้วยไมโครคอนโทรลเลอร์ |                  |               |
| ผู้ดำเนินโครงการ     | นายอดิเรก                               | ปิ่นมี           | รหัส 45380157 |
|                      | นายกฤษฎา                                | เส็งเรียบ        | รหัส 45380259 |
| อาจารย์ที่ปรึกษา     | ดร.สมพร                                 | เรืองสินชัยวานิช |               |
| อาจารย์ที่ปรึกษาร่วม | ดร.สมยศ                                 | เกียรติวนิชวิไล  |               |
|                      | ดร.สุรเชษฐ์                             | กานต์ประชา       |               |
| สาขาวิชา             | วิศวกรรมไฟฟ้า                           |                  |               |
| ภาควิชา              | วิศวกรรมไฟฟ้าและคอมพิวเตอร์             |                  |               |
| ปีการศึกษา           | 2548                                    |                  |               |

.....  
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยราชภัฏบรจรม อนุมัติให้โครงการฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า  
คณะกรรมการสอบโครงการวิศวกรรม

.....  
.....ประธานกรรมการ  
(ดร.สมพร เรืองสินชัยวานิช)

.....  
.....กรรมการ  
(ดร.สมยศ เกียรติวนิชวิไล)

.....  
.....กรรมการ  
(ดร.สุรเชษฐ์ กานต์ประชา)

|                      |   |                  |               |
|----------------------|---|------------------|---------------|
| หัวข้อโครงการ        | การควบคุมระบบลิฟต์ด้วยไมโครคอนโทรลเลอร์ |                  |               |
| ผู้ดำเนินโครงการ     | นายอดิเรก                               | ปิ่นมี           | รหัส 45380157 |
|                      | นายกฤษฎา                                | เส็งเรียบ        | รหัส 45380259 |
| อาจารย์ที่ปรึกษา     | ดร.สมพร                                 | เรืองสินชัยวานิช |               |
| อาจารย์ที่ปรึกษาร่วม | ดร.สมยศ                                 | เกียรติวินชวิไล  |               |
|                      | ดร.สุรเชษฐ์                             | กานต์ประชา       |               |
| สาขาวิชา             | วิศวกรรมไฟฟ้า                           |                  |               |
| ภาควิชา              | วิศวกรรมไฟฟ้าและคอมพิวเตอร์             |                  |               |
| ปีการศึกษา           | 2548                                    |                  |               |

บทคัดย่อ

โครงการนี้เป็นการศึกษาหลักการการทำงานของไมโครคอนโทรลเลอร์ (MCS-51) และการพัฒนาโปรแกรมด้วยภาษาซีในการควบคุมการทำงานของลิฟต์จำลอง เพื่อช่วยอำนวยความสะดวกในการเคลื่อนย้ายสิ่งของภายในอาคารสูง

ผลที่ได้จากการทำโครงการนี้ คือ รู้หลักการการทำงานของไมโครคอนโทรลเลอร์ (MCS-51) การนำไปใช้งานกับลิฟต์จำลองได้ดีและยังสามารถประยุกต์ใช้ในการควบคุมแบบต่างๆ ได้ อาทิ เช่น ระบบไฟจราจรและป้ายโฆษณา

**Project Title** Controlling Elevator System With Microcontroller  
**Name** Mr. Adirek Panmee ID. 45380157  
Mr. Kritsada Sengriab ID. 45380259  
**Project Advisor** Dr. Somporn Ruangsinchaiwanich  
**Co - Project Advisor** Dr. Somyot Kaitwanidvilai  
Dr. Surachet Kanprachar  
**Major** Electrical Engineering  
**Department** Electrical and Computer Engineering  
**Academic Year** 2005

---

### ABSTRACT

This project is to studied the operation of microcontroller (MCS-51) and to develop program with C language to control the elevator model's operation for convenient to move objects in the high building.

The finally of this project is understanding the operation of microcontroller (MCS-51) and can be apply others control such as traffic lights system and advertise board.

## กิตติกรรมประกาศ

โครงการวิศวกรรมเรื่องการควบคุมระบบลิฟต์ด้วยไมโครคอนโทรลเลอร์ (MCS-51) สำเร็จได้ด้วยดีเนื่องจากได้รับคำแนะนำและความช่วยเหลือรวมทั้งข้อคิดเห็นต่างๆ อันเป็นประโยชน์ในการทำโครงการนี้จาก ดร.สมพร เรืองสินชัยวานิช อาจารย์ที่ปรึกษาโครงการและอาจารย์พนัส นัดฤทธิ

ขอขอบคุณอาจารย์ประจำภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ทุกๆ ท่านที่ให้ความรู้ตลอดการเรียนที่ผ่านมาและเพื่อนๆ ที่คอยให้ความช่วยเหลือในทุกๆ ด้าน รวมทั้งคณะวิศวกรรมศาสตร์ที่ให้ความเอื้อเฟื้อเครื่องมือและอุปกรณ์ในการทำโครงการครั้งนี้

คณะผู้จัดทำโครงการ

นายอดิเรก ปั่นมี

นายกฤษฎา เส็งเรียบ



# สารบัญ

|                             | หน้า |
|-----------------------------|------|
| บทคัดย่อ (ภาษาไทย) .....    | ก    |
| บทคัดย่อ (ภาษาอังกฤษ) ..... | ข    |
| กิตติกรรมประกาศ .....       | ค    |
| สารบัญ .....                | ง    |
| สารบัญตาราง .....           | ช    |
| สารบัญรูป .....             | ซ    |

## บทที่ 1 บทนำ

|                               |   |
|-------------------------------|---|
| 1.1 ความสำคัญและที่มา .....   | 1 |
| 1.2 วัตถุประสงค์ .....        | 1 |
| 1.3 ขอบเขตของโครงการ .....    | 2 |
| 1.4 แผนการดำเนินงาน .....     | 2 |
| 1.5 ผลที่คาดว่าจะได้รับ ..... | 2 |

## บทที่ 2 หลักการและทฤษฎี

|  |    |
|--|----|
| 2.1 ไมโครคอนโทรลเลอร์ .....  | 3  |
| 2.2 ไมโครคอนโทรลเลอร์ตระกูล MCS-51 .....                             | 3  |
| 2.3 คุณสมบัติสำคัญของ MCS-51 .....                                   | 3  |
| 2.4 โครงสร้างภายในของไมโครคอนโทรลเลอร์ตระกูล MCS-51 .....            | 5  |
| 2.5 หน่วยความจำของ MCS-51 .....                                      | 5  |
| 2.5.1 แรม (RAM) .....  | 6  |
| 2.5.2 รอม (ROM) .....  | 6  |
| 2.6 รายละเอียดการทำงานของวงจรภายในพอร์ตต่างๆ .....                   | 6  |
| 2.7 ตำแหน่งของไมโครคอนโทรลเลอร์ตระกูล MCS-51 และหน้าที่การทำงาน .... | 7  |
| 2.8 การติดต่อทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ .....                | 10 |
| 2.9 โหมดการการติดต่อทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ .....         | 10 |
| 2.10 รีจิสเตอร์ที่ใช้ในการควบคุมและรับส่งข้อมูลของพอร์ตอนุกรม .....  | 12 |

## สารบัญ (ต่อ)

|  | หน้า |
|--|------|
| 2.11 การอินเทอร์รัพท์ (Interrupt) .....                            | 15   |
| 2.12 รีจิสเตอร์ที่เกี่ยวข้องกับกระบวนการอินเทอร์รัพท์ .....        | 15   |
| 2.12.1 รีจิสเตอร์ IE .....   | 15   |
| 2.12.2 รีจิสเตอร์ IP .....   | 16   |
| 2.12.3 รีจิสเตอร์ TCON .....                                       | 17   |
| 2.13 การควบคุมความเร็วของมอเตอร์กระแสตรง .....                     | 20   |
| 2.14 วิธีการมอดูเลชันทางความกว้างของพัลส์ (PWM) .....              | 20   |
| 2.15 การทำงานตรวจจับของตัวรับสัญญาณ .....                          | 21   |
| 2.16 ส่วนประกอบและการทำงานของรีเลย์ .....                          | 22   |
| 2.17 ไดโอดเปล่งแสง (LED) .....                                     | 23   |
| 2.18 การเลือกใช้ไดโอดเปล่งแสง (LED) .....                          | 24   |
| 2.19 การประยุกต์ใช้งานไดโอดเปล่งแสง (LED) .....                    | 26   |
| 2.20 การวัดค่าแรงดันฟอร์เวิร์ดของ LED .....                        | 28   |
| 2.21 การคำนวณหาค่าความต้านทานอนุกรม .....                          | 28   |
| <b>บทที่ 3 การควบคุมระบบลิฟต์ด้วยไมโครคอนโทรลเลอร์</b>             |      |
| 3.1 แผนผังการทำงานของการควบคุมระบบลิฟต์ด้วยไมโครคอนโทรลเลอร์ ..... | 31   |
| 3.2 ออกแบบและติดตั้งอุปกรณ์ให้กับลิฟต์จำลอง .....                  | 31   |
| 3.3 การออกแบบสำหรับตัวรับสัญญาณ (INPUT) .....                      | 33   |
| 3.4 การออกแบบ OUTPUT .....   | 33   |
| 3.4.1 ไดโอดเปล่งแสง (LED) .....                                    | 33   |
| 3.4.2 มอเตอร์ .....  | 33   |
| 3.5 การออกแบบ HARD WARE .....                                      | 34   |
| 3.6 การออกแบบ PROGRAM .....  | 35   |
| 3.7 การกำหนดอินพุตและเอาต์พุตของลิฟต์จำลอง .....                   | 36   |
| 3.8 การออกแบบวงจรควบคุมระบบลิฟต์ด้วยไมโครคอนโทรลเลอร์ .....        | 37   |
| 3.9 บอร์ดควบคุมระบบลิฟต์ด้วยไมโครคอนโทรลเลอร์ .....                | 38   |
| 3.10 ลิฟต์จำลองที่ใช้งานจริง .....                                 | 39   |

## สารบัญ (ต่อ)

|  | หน้า |
|--|------|
| บทที่ 4 ผลการทดลองและผลการวิเคราะห์                  |      |
| 4.1 การทดลองควบคุมระบบลิฟต์ .....                    | 40   |
| 4.1.1 จุดประสงค์ .....                               | 40   |
| 4.1.2 ขั้นตอนการเชื่อมต่อบอร์ดควบคุม .....           | 40   |
| 4.1.3 ทดสอบการควบคุมการทำงานของลิฟต์จำลอง .....      | 41   |
| 4.2 ผลการทดลอง .....                                 | 41   |
| 4.2.1 ผลการทดสอบการควบคุมการทำงานของลิฟต์จำลอง ..... | 41   |
| บทที่ 5 วิเคราะห์และสรุปผลของโปรแกรม                 |      |
| 5.1 สรุปผลของโครงการ .....                           | 42   |
| 5.2 ปัญหาที่เกิดขึ้นในระหว่างทำโครงการ .....         | 42   |
| 5.3 แนวทางในการพัฒนา .....                           | 42   |
| เอกสารอ้างอิง  | 43   |
| ภาคผนวก  |      |
| ก. ภาษาซีที่ใช้ควบคุมไมโครคอนโทรลเลอร์ .....         | 44   |
| ข. รายละเอียดของอุปกรณ์อิเล็กทรอนิกส์ .....          | 62   |
| ประวัติผู้เขียนโครงการ                               | 73   |



# สารบัญตาราง

| ตารางที่   | หน้า |
|--|------|
| 2.1 ตารางแสดงรายละเอียดของไมโครคอนโทรลเลอร์ตระกูล MCS-51 .....                   | 4    |
| 2.2 หน้าที่พิเศษของพอร์ต 3 .....   | 8    |
| 2.3 แสดงบิตต่างๆ ของรีจิสเตอร์ SCON (Serial Port Control Register) .....         | 12   |
| 2.4 แสดงการกำหนดบิต SM0 และ SM1 เพื่อกำหนดโหมดการทำงาน .....                     | 13   |
| 2.5 แสดงรีจิสเตอร์ PCON เพื่อกำหนดอัตราการรับส่งข้อมูล .....                     | 14   |
| 2.6 แสดงบิตต่างๆ ของรีจิสเตอร์ IE.....   | 15   |
| 2.7 แสดงลำดับความสำคัญของการอินเทอร์รัพท์ .....                                  | 16   |
| 2.8 แสดงบิตต่างๆ ของรีจิสเตอร์ IP .....  | 17   |
| 2.9 แสดงบิตต่างๆ ของรีจิสเตอร์ TCON .....  | 18   |
| 2.10 ค่าแรงดันฟอร์เวิร์คไบแอสของ LED ที่กระแส I <sub>f</sub> 20 มิลลิแอมป์ ..... | 26   |
| 3.1 การกำหนดอินพุตและเอาต์พุตของลิฟต์จำลอง .....                                 | 36   |



# สารบัญรูป

| รูปที่  | หน้า |
|---|------|
| 2.1 แสดงโครงสร้างพื้นฐานของ MCS-51 เบอร์ AT89C52 .....                                    | 4    |
| 2.2 โครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51 .....                                       | 5    |
| 2.3 การจัดขาของ MCS-51 เบอร์ AT89C52 .....  | 9    |
| 2.4 แสดงสัญญาณการรับและส่งข้อมูลในโหมด 0 .....  | 10   |
| 2.5 แสดงสัญญาณการส่งข้อมูลในโหมด 1 .....  | 11   |
| 2.6 แสดงสัญญาณการส่งข้อมูลในโหมด 2 .....  | 11   |
| 2.7 แสดงการเลือกใช้สัญญาณแอกทีฟที่ระดับลอจิก “0” หรือที่ขอบขาของสัญญาณ .....              | 19   |
| 2.8 แสดงความกว้างของพัลส์ขนาดต่างๆ และค่าควิตี้ไซเคิล ของช่วงพัลส์ที่มีความถี่ถึงที่ .... | 21   |
| 2.9 ลิมิตสวิทช์ .....   | 21   |
| 2.10 การทำงานของรีเลย์ก่อนจ่ายแรงดัน .....  | 22   |
| 2.11 การทำงานของรีเลย์ขณะมีกระแสผ่านคอยล์ .....   | 22   |
| 2.12 การทำงานของรีเลย์ 8 ขาในการกลับทางหมุนมอเตอร์ .....                                  | 23   |
| 2.13 ไดโอดเปล่งแสงและโครงสร้างภายใน .....   | 23   |
| 2.14 สัญลักษณ์ของ LED .....   | 24   |
| 2.15 ตำแหน่งขาของ LED .....   | 24   |
| 2.16 LED ชนิดสองสี ประกอบด้วย LED 2 ตัวต่อกลับขั้วและขนานกัน .....                        | 25   |
| 2.17 LED ซึ่งสามารถเปล่งแสงได้ถึง 4 สีโดยใช้ LED เพียง 2 ตัว .....                        | 26   |
| 2.18 เมื่อ LED ได้รับไบแอสกลับกราฟที่ได้จะมีลักษณะคล้ายกับของซีเนอร์ไดโอด .....           | 27   |
| 2.19 แสดงการหาค่าความต้านทานเมื่อทราบค่าแรงดัน $V_F$ และกระแส $I_F$ .....                 | 27   |
| 2.20 การวัดค่าแรงดันฟอร์เวิร์ดของ LED .....   | 28   |
| 2.21 LED ที่นำมาวัดค่าแรงดันฟอร์เวิร์ด .....  | 28   |
| 2.22 ตัวอย่างการเทียบสีตัวต้านทาน .....   | 29   |
| 3.1 แสดงด้านหน้าของลิฟต์จำลอง .....   | 31   |
| 3.2 แสดงด้านข้างของลิฟต์จำลอง .....   | 32   |
| 3.3 การออกแบบ INPUT .....   | 33   |
| 3.4 การออกแบบ OUTPUT .....  | 34   |
| 3.5 การออกแบบ HARD WARE .....   | 34   |
| 3.6 แผนผังการควบคุมระบบลิฟต์ด้วยไมโครคอนโทรลเลอร์ .....                                   | 35   |

## สารบัญรูป (ต่อ)

| รูปที่  | หน้า |
|---|------|
| 3.7 การออกแบบวงจรควบคุมระบบลิฟต์ด้วยไมโครคอนโทรลเลอร์ ..... | 37   |
| 3.8.1 บอร์ดไมโครคอนโทรลเลอร์ MCS-51 .....                   | 38   |
| 3.8.2 บอร์ดสั่งงานลิฟต์ .....                               | 38   |
| 3.8.3 ลิฟต์จำลอง .....                                      | 39   |
| 4.1 การต่อลิฟต์จำลองเพื่อทดสอบโปรแกรม .....                 | 40   |



# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มา

ปัจจุบันเทคโนโลยีสมัยใหม่เข้ามามีบทบาทอย่างมากต่อการดำรงชีวิต โดยเฉพาะเทคโนโลยีระบบฝังตัว (Embedded System) ซึ่งเป็นส่วนประกอบของอุปกรณ์ไฟฟ้าและอิเล็กทรอนิกส์ต่างๆ ได้มีการเพิ่มหน่วยประมวลผลเข้าไปในตัวอุปกรณ์นั้นๆ โดยทั่วไปแล้ววงจรควบคุมการทำงานของอุปกรณ์อิเล็กทรอนิกส์อาจอยู่ในรูปวงจรถติคอลลก็ได้ แต่ในปัจจุบันระบบฝังตัวจะใช้ไมโครคอนโทรลเลอร์มากขึ้นเนื่องจากมีความยืดหยุ่นสูง นอกจากนี้ไมโครคอนโทรลเลอร์ยังสามารถประยุกต์ใช้ในการควบคุมแบบต่างๆ อาทิ เช่น การควบคุมระบบไฟจราจรและการควบคุมระบบลิฟต์ในอาคาร

ดังนั้นทางนิสิตจึงได้จัดทำโครงการการควบคุมระบบลิฟต์ด้วยไมโครคอนโทรลเลอร์ขึ้น เพื่อเป็นประโยชน์ต่อการศึกษาและค้นคว้าเกี่ยวกับระบบควบคุม ตลอดจนสามารถนำไปประยุกต์ใช้ในระบบควบคุมแบบต่างๆ ได้

### 1.2 วัตถุประสงค์

- 1.2.1 เพื่อศึกษาหลักการการทำงานของไมโครคอนโทรลเลอร์ MCS-51
- 1.2.2 เพื่อศึกษาระบบการทำงานของลิฟต์จำลอง
- 1.2.3 เพื่อสามารถประยุกต์ใช้ไมโครคอนโทรลเลอร์ MCS-51 ในระบบควบคุมต่างๆ

### 1.3 ขอบเขตโครงการ

- 1.3.1 ศึกษาข้อมูลและคุณลักษณะของไมโครคอนโทรลเลอร์ MCS-51
- 1.3.2 ศึกษาข้อมูลระบบการทำงานของลิฟต์แบบควบคุมอิสระ
- 1.3.3 ศึกษาการทำงานของลิฟต์จำลอง
- 1.3.4 เขียนคำสั่งควบคุมไมโครคอนโทรลเลอร์ MCS-51 ด้วยภาษาซี
- 1.3.5 ทดลองการทำงานของไมโครคอนโทรลเลอร์กับลิฟต์จำลอง
- 1.3.6 วิเคราะห์และสรุปการทำงานของลิฟต์จำลอง

## 1.4 แผนการดำเนินงาน

| ลำดับ | การดำเนินงาน                      | ก.ค. | ส.ค. | ก.ย. | ต.ค. | พ.ย. | ธ.ค. | ม.ค. |
|-------|-----------------------------------|------|------|------|------|------|------|------|
| 1     | เขียนโครงการ Proposal             | ■    |      |      |      |      |      |      |
| 2     | รวบรวมข้อมูลและเอกสาร             |      | ■    |      |      |      |      |      |
| 3     | วิเคราะห์และออกแบบชิ้นงาน         |      |      | ■    |      |      |      |      |
| 4     | ติดตั้งและทดสอบชิ้นงาน            |      |      |      | ■    |      |      |      |
| 5     | ปรับปรุงแก้ไขชิ้นงาน              |      |      |      |      |      | ■    |      |
| 6     | จัดทำเอกสารและคู่มือ<br>การใช้งาน |      |      |      |      |      |      | ■    |

## 1.5 ผลที่คาดว่าจะได้รับ

- 1.5.1 สามารถเข้าใจหลักการทำงานของไมโครคอนโทรลเลอร์ MCS-51
- 1.5.2 สามารถเข้าใจระบบการทำงานของลิฟต์จำลองได้
- 1.5.3 สามารถนำไมโครคอนโทรลเลอร์ไปควบคุมระบบลิฟต์แบบควบคุมอิสระได้

## 1.6 งบประมาณ

- |   |          |       |     |
|---|----------|-------|-----|
| 1.6.1 ค่าอุปกรณ์ทางไฟฟ้าและอิเล็กทรอนิกส์ | เป็นเงิน | 1,300 | บาท |
| 1.6.2 ค่าเอกสารและข้อมูลสำหรับโครงการ     | เป็นเงิน | 500   | บาท |
| 1.6.3 ค่าวัสดุอื่นๆ                       | เป็นเงิน | 200   | บาท |

รวมเป็นเงินทั้งสิ้น 2,000 บาท (สองพันบาทถ้วน)

## บทที่ 2

# หลักการและทฤษฎี

### 2.1 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Micro Controller) คือ คอมพิวเตอร์ที่นำส่วนโปรเซสเซอร์หรือ ซีพียูและส่วนติดต่อกับภายนอกมาสร้างอยู่ในชิปเดียว โดยส่วนติดต่อกับภายนอกต่างๆ ได้แก่

- หน่วยความจำ (Memory)
- ตัวสร้างฐานเวลา (Timer)
- วงจรแปลงสัญญาณอะนาล็อกเป็นดิจิทัล (A/D Convert)
- วงจรแปลงสัญญาณดิจิทัลเป็นอะนาล็อก (D/A Convert)
- วงจรควบคุมการเข้าถึงหน่วยความจำโดยตรง (DMA Controller)
- ส่วนติดต่อพอร์ทขนาน (Parallel Interface)
- ส่วนติดต่อพอร์ทอนุกรม (Serial Interface)

ในการใช้งานสามารถนำไปควบคุมงานต่างๆ ได้ด้วยชิปเดียว เพียงป้อนสัญญาณไฟเลี้ยง และตัวกำเนิดความถี่ก็สามารถทำงานได้ซึ่งแตกต่างกับไมโครโปรเซสเซอร์ที่ต้องมีการสร้างส่วนต่ออินพุตและเอาต์พุตเพิ่มเติมทำให้งานมีความยุ่งยากซับซ้อน ซึ่งไม่เหมาะกับงานควบคุมขนาดเล็ก ตัวอย่างไมโครคอนโทรลเลอร์ ได้แก่ ซีพียูตระกูล ARM และตระกูล MCS-51

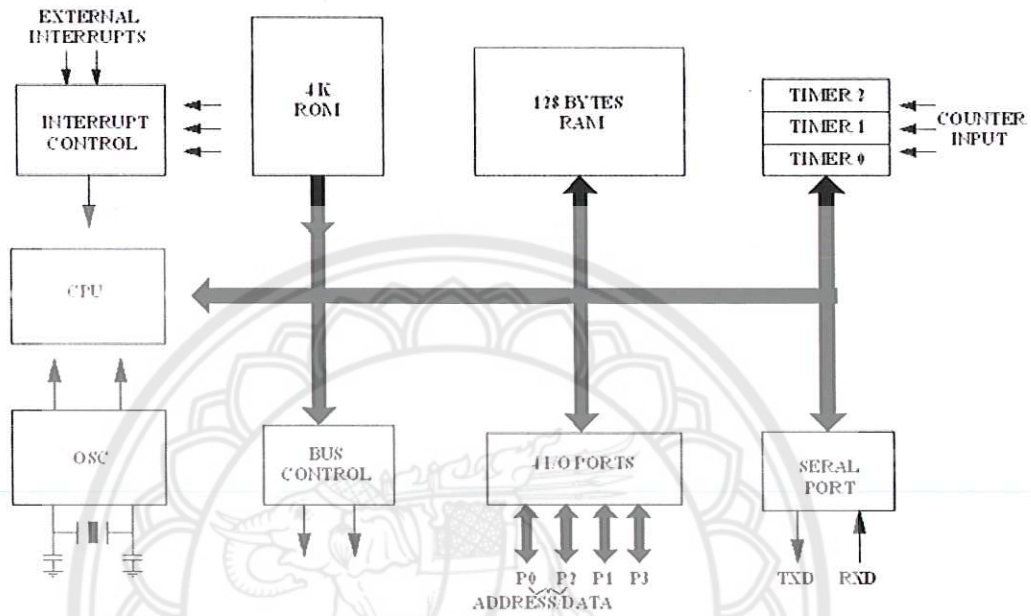
### 2.2 ไมโครคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ถูกผลิตขึ้นในปี 1980 โดยเบอร์ 8051 เป็นเบอร์แรกที่ผลิตออกมา ไมโครคอนโทรลเลอร์ตระกูลนี้มีด้วยกันหลายเบอร์ขึ้นกับโครงสร้างภายใน บางเบอร์มีหน่วยความจำภายในเป็นแบบ ROM บางเบอร์เป็นแบบ EPROM บางเบอร์มี RAM ภายใน 128 ไบต์ เป็นต้น

### 2.3 คุณสมบัติสำคัญของ MCS-51

- ซีพียู 8 บิต
- สามารถประมวลผลแบบบิตได้
- อ่างหน่วยความจำภายนอกสำหรับเก็บโปรแกรมได้ 64 กิโลไบต์
- อ่างหน่วยความจำภายนอกสำหรับเก็บข้อมูลได้ 64 กิโลไบต์
- หน่วยความจำภายใน (ROM) สำหรับเก็บโปรแกรม 4 กิโลไบต์
- หน่วยความจำภายใน (RAM) สำหรับเก็บข้อมูล 128 กิโลไบต์

- สามารถอินเทอร์รัพท์ได้ 5 แหล่ง
- มี Timer 16 บิต 2 ตัว
- พอร์ตควบคุมการสื่อสารอนุกรมที่สามารถรับส่งข้อมูลแบบ Full Duplex ความเร็วสูง
- มีพอร์ตขนาน (พอร์ต I/O) ขนาด 8 บิต 4 พอร์ต
- มีวงจรรอสซิงเกิลเตอร์และวงจรรนาฬิกาบนชิป



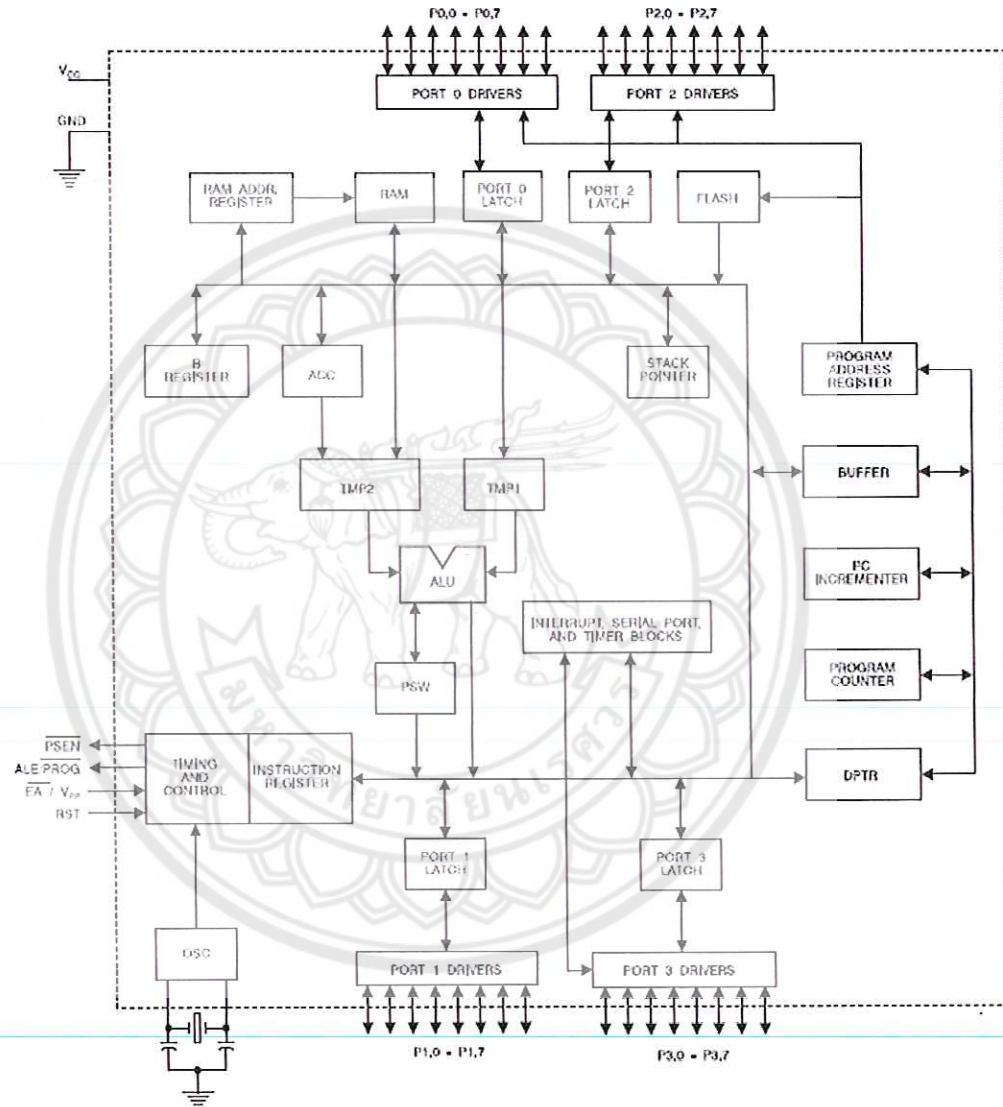
รูปที่ 2.1 แสดงโครงสร้างพื้นฐานของ MCS-51 เบอร์ AT89C52

ตารางที่ 2.1 ตารางแสดงรายละเอียดของไมโครคอนโทรลเลอร์ตระกูล MCS-51

| ไมโครคอนโทรลเลอร์ | หน่วยความจำภายใน (internal memory)   |             | ตั้งเวลา/นับ<br>เวลา<br>(time/counter) | สัญญาณ<br>อินเทอร์รัพท์<br>จากภายนอก |
|-------------------|--------------------------------------|-------------|--|--------------------------------------|
|                   | หน่วยความจำภายในแบบ<br>EPROM, EEPROM | ข้อมูล RAM  |  |                                      |
| AT89C1051         | 1 kb × 8                             | 64 × 8 bit  | 2 × 16 bit                             | 6                                    |
| AT89C2051         | 2 kb × 8                             | 128 × 8 bit | 2 × 16 bit                             | 6                                    |
| AT89C4051         | 4 kb × 8                             | 128 × 8 bit | 2 × 16 bit                             | 6                                    |
| AT89C51           | 4 kb × 8                             | 128 × 8 bit | 2 × 16 bit                             | 6                                    |
| AT89C52           | 8 kb × 8                             | 256 × 8 bit | 3 × 16 bit                             | 8                                    |
| AT89S52           | 8 kb × 8                             | 256 × 8 bit | 3 × 16 bit                             | 8                                    |
| AT89C55           | 20 kb × 8                            | 256 × 8 bit | 3 × 16 bit                             | 8                                    |
| AT89S8252         | 8 KB × 8 (2 kb EEPROM)               | 256 × 8 bit | 3 × 16 bit                             | 9                                    |
| AT89S53           | 12 kb × 8                            | 256 × 8 bit | 3 × 16 bit                             | 9                                    |

## 2.4 โครงสร้างภายในของไมโครคอนโทรลเลอร์ตระกูล MCS-51

วงจรรภายในของไมโครคอนโทรลเลอร์ตระกูล MCS-51 ประกอบด้วยวงจรรินพุตและเอาต์พุตทั้งหมด 4 พอร์ต แต่ละพอร์ตจะเป็น 8 บิต หน่วยความจำโปรแกรมภายใน (EPROM , EEPROM และ Flash) หน่วยความจำที่เป็นข้อมูล (RAM) นั้น ซึ่งจะอยู่ในวงจรรหลักของไมโครคอนโทรลเลอร์ตระกูล MCS-51 ตลอดจนวงจรรคำนวณทางคณิตศาสตร์และลอจิก (ALU) วงจรรีจิสเตอร์ทั่วไปและรีจิสเตอร์ฟังก์ชันการใช้งานเฉพาะ แสดงดังรูปที่ 2.2



รูปที่ 2.2 โครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51

## 2.5 หน่วยความจำของ MCS-51

หน่วยความจำ (Memory) คืออุปกรณ์อิเล็กทรอนิกส์ประเภทสารกึ่งตัวนำทำหน้าที่เก็บโปรแกรมและข้อมูลระหว่างรอให้ซีพียูประมวลผล หรือเก็บผลลัพธ์ที่ได้หลังจากซีพียูประมวลผลเสร็จแล้ว ประเภทของหน่วยความจำแบ่งออกเป็น 2 ชนิด คือ



### 2.5.1 แรม (RAM)

แรมเป็นหน่วยความจำที่สามารถอ่านและเขียนได้ คือ ซีพียูจะอ่านข้อมูลจากแรมมาประมวลผล และเมื่อประมวลผลเสร็จแล้วก็สามารถนำไปเก็บไว้ในแรมได้ แรมถูกแบ่งออกเป็น 2 ประเภทคือ

- ไดนามิกแรม (DRAM) เป็นหน่วยความจำที่ต้องรีเฟรชข้อมูลที่เก็บภายในตลอดเวลา เนื่องจากมีโครงสร้างมาจากคาปาซิเตอร์ ซึ่งมีคุณลักษณะการคายประจุ
- สแตติกแรม (SRAM) เป็นหน่วยความจำที่ไม่ต้องมีรีเฟรชข้อมูล เพราะทำมาจากทรานซิสเตอร์

### 2.5.2 รอม (ROM)

รอมเป็นหน่วยความจำที่ไม่ต้องมีไฟเลี้ยง สามารถอ่านข้อมูลได้อย่างเดียว แต่กระบวนการเขียนนั้นมีหลายวิธี สามารถแบ่งประเภทตามกระบวนการเขียนดังนี้

- Mask-programmed ROM หรือ MROM เป็นหน่วยความจำรอมที่ถูกเขียนข้อมูลมาจากโรงงานผู้ผลิต
- Programmable ROM หรือ PROM เป็นหน่วยความจำรอมที่ผู้ใช้สามารถเขียนข้อมูลได้เพียงครั้งเดียว
- Erasable Programmable ROM หรือ EPROM เป็นหน่วยความจำรอมที่สามารถลบได้ด้วยแสงอัลตราไวโอเล็ต ทำให้สามารถเขียนข้อมูลได้หลายครั้ง
- Electrically Erasable Programmable ROM หรือ EEPROM เป็นหน่วยความจำรอมที่เขียนได้หลายครั้งและลบด้วยไฟฟ้า โดยสามารถเลือกตำแหน่งที่ต้องการลบได้

## 2.6 รายละเอียดการทำงานของวงจรรภายในพอร์ตต่างๆ

วงจรมัลติเพล็กซ์ (Multiplex) (มีเฉพาะพอร์ต 0 และ พอร์ต 2) จะเป็นส่วนควบคุมการทำงานของพอร์ตว่าจะให้พอร์ตทำงานในลักษณะเป็น อินพุตเอาต์พุต/เอาต์พุตทั่วไป หรือเป็นพอร์ตเพื่อใช้ติดต่อกับหน่วยความจำภายนอก

วงจรถัด (Latch) จะเป็นส่วนของการคงสถานะ (Latch) ข้อมูลของพอร์ตบิตนั้นๆ คือ หลังจาก CPU ส่งข้อมูล ("0" หรือ "1") มาที่ขา D ของดี-ฟลิปฟล็อปแล้ว จะส่งสัญญาณ CLK ตามมาเพื่อให้ดี-ฟลิปฟล็อป รับค่าข้อมูลและทำการแลตช์เอาไว้ ซึ่งเอาต์พุตของดี-ฟลิปฟล็อปจะทำหน้าที่ควบคุมการทำงานของแฟตเพื่อควบคุมสถานะของพอร์ต

วงจรรเอาต์พุต (Out put) จะใช้แฟต (FET) เป็นตัวไดรฟ์เวอร์ (Driver) จะทำหน้าที่รับสัญญาณจากวงจรถัดเพื่อมาปรับเปลี่ยนระดับแรงดันและกระแสก่อนส่งออกไปยังขาพอร์ต

วงจรถูลอัพ (Pull-Up) โดยทั่วไปจะเป็นสารกึ่งตัวนำที่ต่ออยู่ระหว่างไฟบวก 5 โวลต์กับขาของพอร์ตทำหน้าที่รักษาระดับแรงดันที่ขาพอร์ตให้เป็นสถานะ High (Logic “1”) เมื่อสั่งให้ขาพอร์ตเป็น “1” กระแสจะไหลผ่านอุปกรณ์พูลอัพดังกล่าวเพื่อ ไปจ่ายให้กับอุปกรณ์ภายนอกที่ต่ออยู่ (พอร์ต 0 ไม่มีอุปกรณ์พูลอัพ)

การอ่านสถานะของฟลิปฟลอป จะทำการส่งสัญญาณ มาที่ขา Read Latch หลังจากนั้นจะทำการอ่านข้อมูล (“0” หรือ “1”) ที่ขา D ของดี-ฟลิปฟลอปกลับไปทางขา Internal Bus การทำงานจะเป็นเช่นนี้ทุกพอร์ตและบิต

การเขียนสถานะของฟลิปฟลอป จะทำโดยการส่งข้อมูล (“0” หรือ “1”) มาที่ขา D ของดี-ฟลิปฟลอป ทาง Internal Bus หลังจากนั้นจะส่งสัญญาณมาที่ขา CLK ของดี-ฟลิปฟลอป (Write to Latch) เพื่อทำการเขียนข้อมูล การทำงานจะเป็นลักษณะเช่นนี้ทุกพอร์ต

การอ่านสถานะของพอร์ต (PO.X) การทำงานในส่วนนี้จะไม่เกี่ยวข้องกับดี-ฟลิปฟลอป การอ่านสถานะจากพอร์ตสามารถทำได้โดยการส่งสัญญาณมาที่ขา Read Pin หลังจากนั้นจะอ่านข้อมูล (“0” หรือ “1”) กลับไปทางขา Internal Bus การทำงานจะเป็นลักษณะเช่นนี้ทุกพอร์ต ในการใช้งานเราไม่ได้ใส่ใจถึงลักษณะการทำงานของวงจรมานัก แต่สิ่งที่ควรให้ความสำคัญคือ ลักษณะหน้าที่การทำงานของพอร์ตต่างๆ โดยเฉพาะหน้าที่พิเศษของแต่ละพอร์ต ซึ่งถือได้ว่ามีความสำคัญในการใช้งานไมโครคอนโทรลเลอร์มากพอสมควร

## 2.7 ตำแหน่งของไมโครคอนโทรลเลอร์ตระกูล MCS-51 และหน้าที่การทำงาน

Port0 (P0.0-P0.7 ขาที่ 32-39) ทำหน้าที่เป็นสัญญาณควบคุมอุปกรณ์ภายนอกได้ 2 ทิศทาง สามารถรับข้อมูลอินพุตและส่งข้อมูลเอาต์พุตได้ มีขนาด 8 บิตและใช้งานเป็นพอร์ตสำหรับติดต่อกับหน่วยความจำภายนอก คือ รับ/ส่งข้อมูลและกำหนดแอดเดรสไบต์ต่ำ

Port1 (P1.0-P1.7 ขาที่ 1-8) ทำหน้าที่เป็นสัญญาณควบคุมอุปกรณ์ภายนอกได้ 2 ทิศทาง สามารถปรับได้ทั้งอินพุตและเอาต์พุต มีขนาด 8 บิต สามารถอ้างอิงถึงการทำงานได้ที่ละบิตและวงจรมานักมีตัวต้านทานเพิ่มกระแสไฟฟ้า (Pull up) ในกรณีที่ต้องการให้รับข้อมูลอินพุตก็สามารถทำได้เหมือนพอร์ต 0

Port2 (P2.0-P2.7 ขาที่ 21-28) ทำหน้าที่เป็นสัญญาณควบคุมอุปกรณ์ภายนอกได้ทั้ง 2 ทิศทาง คือ เป็นได้เป็นทั้งอินพุตและเอาต์พุต มีขนาด 8 บิต สามารถใช้เป็นขาสัญญาณที่กำหนดตำแหน่งหน่วยความจำ (A8-A15) และมีวงจรมีเพิ่มกระแสไฟภายใน การกำหนดให้เป็นขาอินพุตทำได้โดยการส่งข้อมูลสถานะ “1” ไปยังบิตที่ต้องการให้เป็นอินพุตก็จะสามารถทำการรับค่าข้อมูลอินพุตได้

Port3 (P3.0-P3.7 ขาที่ 10-17) ทำหน้าที่เป็นสัญญาณควบคุมอุปกรณ์ภายนอกอินพุตและเอาต์พุต 2 ทิศทาง มีขนาด 8 บิต คุณสมบัติทั่วไปจะเหมือนกับพอร์ตอื่นๆ แต่จะมีคุณสมบัติที่ต่างออกไป คือ ใช้ทำหน้าที่พิเศษเป็นสัญญาณควบคุมการทำงานต่างๆ ของไมโครคอนโทรลเลอร์ ดังตารางที่ 2.2

ตารางที่ 2.2 หน้าที่พิเศษของพอร์ต 3

| บิตของพอร์ต | สัญญาณ            | หน้าที่การทำงาน   |
|-------------|-------------------|---|
| P3.0        | RXD               | รับข้อมูลจากพอร์ตอนุกรม (serial input port)   |
| P3.1        | TXD               | ส่งข้อมูลจากพอร์ตอนุกรม (serial output port)  |
| P3.2        | $\overline{INT0}$ | รับสัญญาณอินเทอร์รัปต์หมายเลข 0 (external interrupt 0)                                |
| P3.3        | $\overline{INT1}$ | รับสัญญาณอินเทอร์รัปต์หมายเลข 1 (external interrupt 1)                                |
| P3.4        | T0                | ใช้ตั้งเวลา/นับเวลาที่ 0 (Timer 0 external input)                                     |
| P3.5        | T1                | ใช้ตั้งเวลา/นับเวลาที่ 1 (Timer 1 external input)                                     |
| P3.6        | $\overline{WR}$   | เป็นสัญญาณเขียนข้อมูลหน่วยความจำหรืออุปกรณ์ภายนอก (external data memory write strobe) |
| P3.7        | $\overline{RD}$   | เป็นสัญญาณอ่านข้อมูลหน่วยความจำหรืออุปกรณ์ภายนอก (external data memory read strobe)   |

**PSEN** (Program Store Enable ขาที่ 29) เป็นขาที่ทำงานที่สถานะลอจิกเป็น “0” ไมโครคอนโทรลเลอร์ต้องอ่านค่าจากหน่วยความจำภายนอกที่เป็นข้อมูล โดยโปรแกรมจะเก็บในหน่วยความจำถาวร (ROM, EPROM, EEPROM) ส่วนมากใช้ต่อเป็นขาเลือกทำงาน (Enable) แต่ถ้าไมโครคอนโทรลเลอร์ใช้หน่วยความจำภายในขานี้ก็จะไม่ได้ใช้งานและมีค่าลอจิกเป็น “1”

**ALE/PROG** (Address Latch Enable ขาที่ 30) ทำหน้าที่ควบคุมการทำงานของสัญญาณกำหนดตำแหน่งกับสัญญาณข้อมูลโดยใช้การเลือกเส้นทาง (data select หรือ multiplex) โดยปกติเมื่อไมโครคอนโทรลเลอร์ทำงานจะส่งสัญญาณกำหนดตำแหน่งออกมาพร้อมกับส่งสัญญาณให้ขา ALE ทำงาน

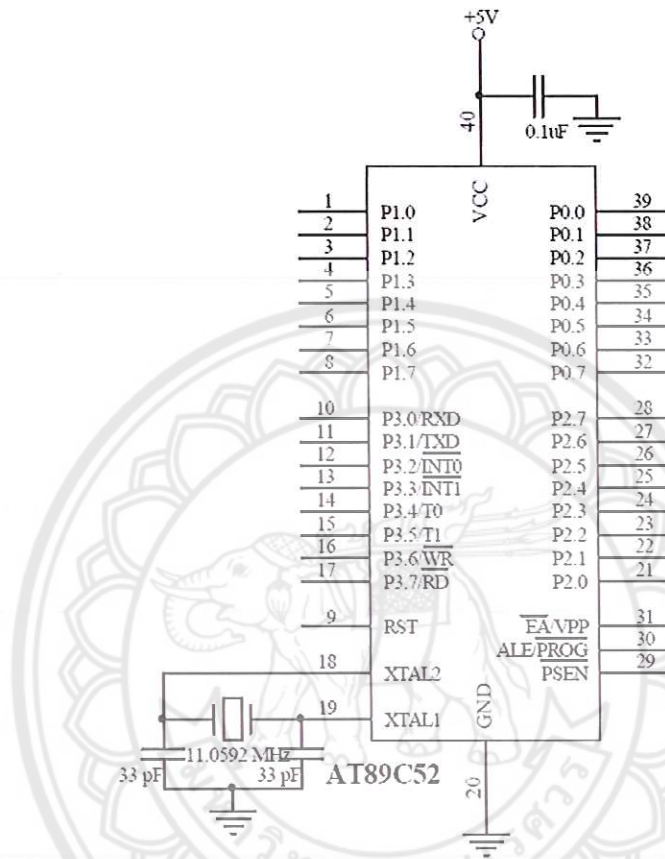
**EA/VPP** (External Access ขาที่ 31) ทำหน้าที่เลือกการทำงานของหน่วยความจำ ถ้ามีค่าลอจิกเป็น “1” หมายถึง ใช้ข้อมูลจากหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ แต่ถ้ามีค่าลอจิกเป็น “0” หมายถึง ใช้ข้อมูลจากหน่วยความจำภายนอก

**RST** (Reset ขาที่ 9) ทำหน้าที่เริ่มต้นการทำงานใหม่ของไมโครคอนโทรลเลอร์ (รีเซ็ตซีพียู) การทำงานที่ค่าลอจิก “1” นี้จะทำให้ไมโครคอนโทรลเลอร์เริ่มต้นทำงานที่ตำแหน่ง 0000 เพื่ออ่านข้อมูลโปรแกรมและจัดระบบการทำงาน

**XTAL1** และ **XTAL2** (ขาสัญญาณนาฬิกา ขาที่ 18-19) ซึ่งจะทำหน้าที่เป็นตัวกำหนดสัญญาณนาฬิกาให้กับไมโครคอนโทรลเลอร์ใช้เป็นฐานเวลาในการทำงาน โดยจะใช้แผ่นผลึก (Crystal) ที่มีความถี่ตั้งแต่ 0-24 เมกกะเฮิร์ตซ์ (MHz) ร่วมกับตัวเก็บประจุขนาด 20-33 pF

VCC (Power Supply ขาที่ 40) เป็นขาที่ต่อแหล่งจ่ายไฟบวกให้กับไมโครคอนโทรลเลอร์  
ซึ่งใช้แหล่งจ่ายไฟขนาดไม่เกิน 5 โวลต์

GND (Ground ขาที่ 20) จะเป็นขากราวด์



รูปที่ 2.3 การจัดขาของ MCS-51 เบอร์ AT89C52

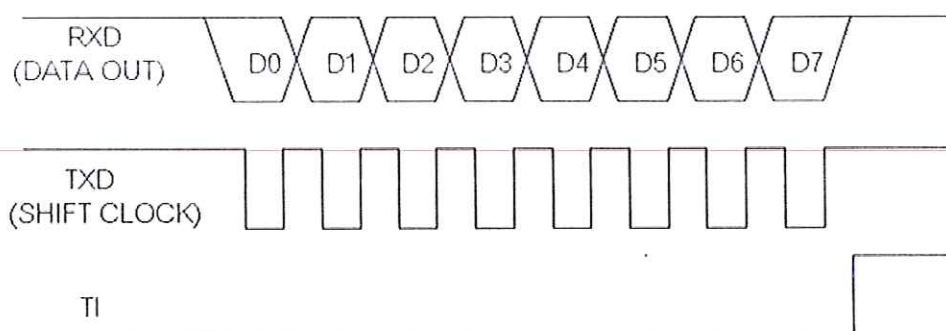
## 2.8 การติดต่อทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์

พอร์ตสื่อสารอนุกรมของไมโครคอนโทรลเลอร์จะมีโครงสร้างเป็นแบบฟูลดูเพล็กซ์ ซึ่งรับและส่งข้อมูลในเวลาเดียวกันได้โดยจะมีรีจิสเตอร์ SBUF (Serial Data Buffer) เป็นบัฟเฟอร์สำหรับการรับส่งข้อมูลอนุกรม โดยเริ่มต้นเมื่อมีการเขียนข้อมูลเก็บไว้ในรีจิสเตอร์ SBUF หลังจากนั้นข้อมูลจะถูกจัดการโดยวิธีทางฮาร์ดแวร์ในการเลื่อนบิตเพื่อส่งสัญญาณออกไปภายนอก หลังจากมีการส่งข้อมูลออกไปจนครบแล้วจึงจะทำการเซตบิตโดยกำหนดค่าของแฟล็ก TI ในรีจิสเตอร์ SCON ให้เป็นสถานะ “1” เพื่อแจ้งว่ารีจิสเตอร์ SBUF ว่างแล้ว และพร้อมที่จะส่งข้อมูลไปต่อการรับข้อมูลจากพอร์ตอนุกรมจะต้องเริ่มต้นโดยการกำหนดค่าของบิต REN ที่อยู่ในรีจิสเตอร์ SCON ให้มีค่าเป็นสถานะ “1” หลังจากนั้นเมื่อมีการรับข้อมูลเข้ามาจากภายนอกก็จะทำการเลื่อนข้อมูลไปโดยอัตโนมัติและเมื่อบิตสุดท้ายถูกเลื่อนบิตเข้ามาเรียบร้อยแล้ว ข้อมูลจะถูกย้ายมาเก็บไว้ในรีจิสเตอร์ SBUF และจะทำการเซตที่บิต RI ให้เป็นสถานะ “1” ซึ่งส่งผลให้เกิดการอินเตอร์รัปต์โปรแกรมขึ้น

## 2.9 โหมดการการติดต่อทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์

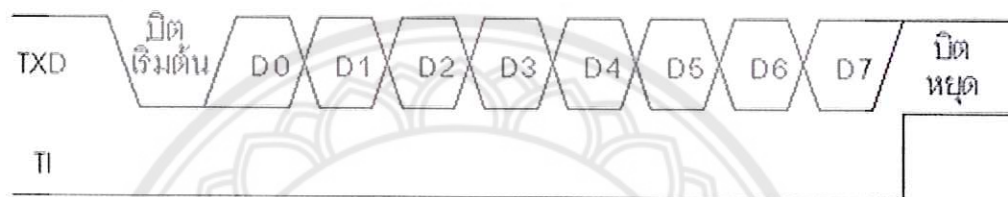
การสื่อสารอนุกรมของไมโครคอนโทรลเลอร์จะแบ่งออกได้เป็น 4 โหมดด้วยกันและในแต่ละโหมดจะสามารถสรุปหน้าที่ได้ดังนี้

โหมด 0 จะเป็นการรับส่งข้อมูลขนาด 8 บิตแบบอนุกรม การส่งข้อมูลจะเลื่อนออกไปทีละบิตโดยจะใช้งานของขา RxD เพียงขาเดียวและจะไม่มีบิตเริ่มต้น (Start Bit) ส่วนขา TxD จะใช้เป็นขาของสัญญาณนาฬิกาในการให้จังหวะการเลื่อนข้อมูลกับวงจรภายนอก (Shift Clock) อัตราการรับส่งข้อมูล (Baud Rate) จะเป็น 1/12 เท่าของสัญญาณนาฬิกา การรับและส่งข้อมูลจะเริ่มจากบิตต่ำ (LSB) ก่อนใช้สำหรับเป็นชิพรีจิสเตอร์ (Shift Register) จุดประสงค์เพื่อใช้ในการขยายพอร์ตอินพุตหรือพอร์ตเอาต์พุตให้มีจำนวนมาก แต่ในโหมด 0 เรามักจะไม่ค่อยนิยมนำมาใช้งาน เพราะไอซี ไมโครคอนโทรลเลอร์ในปัจจุบันที่เราใช้อยู่มีจำนวนพอร์ตที่มากพอและมีไอซีเบอร์อื่นๆ ในตระกูลเดียวกันให้เลือกจำนวนพอร์ตใช้งานมากมายอยู่แล้ว



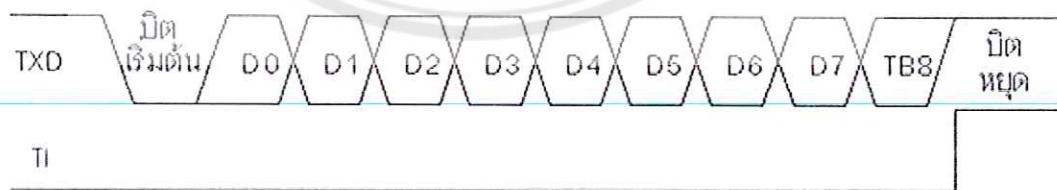
รูปที่ 2.4 แสดงสัญญาณการรับและส่งข้อมูลในโหมด 0

โหมด 1 จะเป็นการรับและส่งข้อมูลขนาด 10 บิตแบบ UART (Universal Asynchronous Receiver Transmitters) สามารถใช้ในการติดต่อสื่อสารอนุกรมกับมาตรฐานของ RS-232C ของไมโครคอมพิวเตอร์ได้ ซึ่งข้อมูลอนุกรม 10 บิตจะเข้ามาทางขา RXD และส่งข้อมูลออกแบบอนุกรมทางขา TXD โดยจะประกอบด้วย 1 บิตแรกเป็นบิตเริ่มต้น (Start Bit ค่า 0), 8 บิตต่อมาจะเป็นบิตของข้อมูล (การรับ/ส่งจะเริ่มจากบิตต่ำก่อน) และบิตหยุดอีก 1 บิต (Stop Bit ค่า 1) ส่วนทางด้านรับข้อมูลจะนำค่าบิตหยุด (Stop Bit) ที่รับเข้ามาได้นำไปเก็บไว้ในบิต RB8 ที่อยู่ในรีจิสเตอร์ SCON และความเร็วของการส่งข้อมูลในโหมด 1 จะขึ้นอยู่กับบิต SMOD ที่อยู่ในรีจิสเตอร์ PCON และอัตราโอเวอร์โพล์ของไทมเมอร์ 1 ซึ่งอัตราการรับส่งข้อมูลในโหมดนี้สามารถกำหนดได้ตามต้องการ



รูปที่ 2.5 แสดงสัญญาณการส่งข้อมูลในโหมด 1

โหมด 2 จะเป็นการรับและส่งข้อมูลขนาด 11 บิตแบบ UART (Universal Asynchronous Receiver Transmitters) ข้อมูลแบบอนุกรมจะถูกรับเข้ามาทางขา RXD และส่งข้อมูลออกไปทางขา TXD ซึ่งข้อมูล 11 บิตประกอบด้วย บิตแรกจะเป็นบิตเริ่มต้น (Start Bit ค่า 0), 9 บิตต่อมาจะเป็นบิตของข้อมูลและบิตสุดท้ายจะเป็นบิตหยุด 1 บิต (Stop Bit ค่า 1) สำหรับข้อมูลในบิตที่ 9 จะกำหนดไว้ใน TB8 ที่อยู่ในรีจิสเตอร์ SCON ซึ่งสามารถกำหนดเป็น 1 หรือ 0 ก็ได้ นิยมนำมาใช้ในการส่งบิตเพื่อตรวจสอบการส่งข้อมูล (Parity Bit)



รูปที่ 2.6 แสดงสัญญาณการส่งข้อมูลในโหมด 2

โหมด 3 เป็นการรับส่งข้อมูลแบบ 11 บิตแบบ UART (Universal Asynchronous Receiver Transmitters) เหมือนกับโหมด 2 แต่ในโหมด 3 สามารถกำหนดอัตราความเร็วในการรับและส่งข้อมูลได้ตามต้องการ

UART (Universal Asynchronous Receiver Transmitters) เป็นการส่งข้อมูลแบบอนุกรม โดยขึ้นอยู่กับความพร้อมของทางด้านส่งและด้านรับ เป็นการส่งข้อมูลโดยทำการเพิ่มเติมข้อมูลบางอย่างเข้าไป ( Start Bit ,Stop Bit ,Parity Bit ) เพื่อให้การรับและการส่งข้อมูลสามารถจะทำงานให้มีความถูกต้องของข้อมูลมากยิ่งขึ้น

## 2.10 รีจิสเตอร์ที่ใช้ในการควบคุมและรับส่งข้อมูลของพอร์ตอนุกรม

รีจิสเตอร์ที่ใช้งานในการติดต่อสื่อสารทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์จะประกอบด้วยรีจิสเตอร์ SCON ที่ทำหน้าที่ควบคุมการทำงาน รีจิสเตอร์ SBUF จะใช้ในการเก็บข้อมูลที่จะรับหรือส่ง และรีจิสเตอร์ PCON ซึ่งจะใช้ ในการกำหนดอัตรารับส่ง โดยรีจิสเตอร์แต่ละตัวจะมีหน้าที่และการทำงานในแต่ละบิตดังต่อไปนี้

รีจิสเตอร์ SCON (Serial Port Control Register) เป็นรีจิสเตอร์ขนาด 8 บิต อยู่ในส่วนของรีจิสเตอร์พิเศษ (Special Function Register) ในตำแหน่งแอดเดรสที่ 98H และสามารถเข้าถึงข้อมูลแบบไบนารีและแบบบิตได้ โดยจะทำหน้าที่ควบคุมการทำงานของพอร์ตอนุกรม การเลือกโหมดการทำงานและเก็บข้อมูลในบิตที่ 9 (ซึ่งโดยปกติข้อมูลจะมี 8 บิต อยู่ในรีจิสเตอร์ SBUF) ของการรับข้อมูล (RB8) และส่งข้อมูล (TB8) รายละเอียดของแต่ละบิตมีดังต่อไปนี้

ตารางที่ 2.3 แสดงบิตต่างๆของรีจิสเตอร์ SCON (Serial Port Control Register)

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

**SM0, SM1: (Serial port mode bit 0-1)** เป็นบิตที่ใช้ในการกำหนดโหมดการทำงานของพอร์ตอนุกรมจำนวน 4 โหมด

ตารางที่ 2.4 แสดงการกำหนดบิต SM0 และ SM1 เพื่อกำหนดโหมดการทำงาน

| SM0 | SM1 | โหมด การทำงาน อัตรารับ-ส่ง                  |
|-----|-----|---|
| 0   | 0   | 0 Shift Register $f_{osc}/12$               |
| 0   | 1   | 1 8 bit UART Variable                       |
| 1   | 0   | 2 9 bit UART $f_{osc}/32$ หรือ $f_{osc}/64$ |
| 1   | 1   | 3 9 bit UART Variable                       |

SM2 เป็นบิตที่ทำหน้าที่ควบคุมการทำงานและเลือกลักษณะการเชื่อมต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์แบบ Single Processor System หรือ Multi Processors System โดยกำหนดให้ SM2 = 1 เป็นการเลือกแบบ Multi Processors System คือ ระบบการสื่อสารแบบใช้ซีพียูหลายๆตัวร่วมกันทำงานจะใช้งานในโหมด 2 หรือโหมด 3 SM2 = 0 เป็นการเลือกแบบ Single Processor System โดยสามารถใช้ได้กับทุกโหมด (การใช้งานในโหมด 0 ต้องกำหนดให้ SM2 = 0) ในกรณีที่เลือกให้ SM2 = 1 แบบ Multi Processors System ถ้าข้อมูลที่รับเข้ามาบิตที่ 9 (อยู่ในบิต RB8) มีค่าเป็น "1" ทำให้แฟลคอินเตอร์รัปต์ทางด้านรับจะถูกเซตให้เป็น 1 (RI = 1) แต่ถ้าข้อมูลในบิตที่ 9 รับเข้ามามีค่าเป็น "0" จะทำให้แฟลคอินเตอร์รัปต์ทางด้านรับเป็น 0 (RI = 0) การทำงานในโหมด 1 ถ้าให้ SM2 = 1 แฟลคอินเตอร์รัปต์ทางด้านรับ (แฟลค RI) จะไม่ถูกเซตหากข้อมูลที่รับเข้ามาไม่มีบิตหยุด (Stop Bit)

REN (Enable Serial Reception) เป็นบิตที่ควบคุมการรับข้อมูลของพอร์ตอนุกรม กำหนดสถานะของบิตได้โดยซอฟต์แวร์ 1 = ให้มีการรับข้อมูลและ 0 = ไม่ให้มีการรับข้อมูล

TB8 (Transmit Bit D8) เป็นบิตของข้อมูลบิตที่ 9 ในการส่งข้อมูลใช้งานโหมด 2 และโหมด 3 กำหนดสถานะของบิตได้โดยซอฟต์แวร์

RB8 (Receive Bit D8) เป็นบิตของข้อมูลบิตที่ 9 ในการรับข้อมูลโดยใช้งานโหมด 2 และโหมด 3 หากใช้งานในโหมด 1 ถ้ากำหนดให้ SM2 = 0 บิตนี้จะเป็นค่าของ Stop Bit ที่รับเข้ามาสำหรับโหมด 0 จะไม่ใช้งานบิตนี้

TI (Transmit Interrupt Flag) เป็นบิตที่ใช้งานในการอินเตอร์รัปต์ด้านส่งข้อมูลและจะถูกเซตทางฮาร์ดแวร์เมื่อมีการส่งข้อมูลเสร็จสิ้นลงในบิตที่ 8 ของโหมด 0 (Shift Register) หรือเมื่อเริ่มต้นส่ง Stop Bit ในโหมด 1, 2 หรือ 3 และจะต้องเคลียร์บิตนี้ด้วยซอฟต์แวร์ทุกครั้งเมื่อ



โปรแกรมตอบสนองการอินเทอร์รัปต์ของการส่งข้อมูลเรียบร้อยแล้ว

RI : (Receive Interrupt Flag) เป็นบิตที่ใช้งานในการอินเทอร์รัปต์ทางด้านรับข้อมูล จะถูกเซตทางฮาร์ดแวร์ เมื่อมีการรับข้อมูลเสร็จสิ้นลงในบิตที่ 8 ในโหมด 0 (Shift Register) และจะต้องเคลียร์บิตนี้ด้วยซอฟต์แวร์ทุกครั้ง เมื่อโปรแกรมตอบสนองการอินเทอร์รัปต์ของการรับข้อมูลเรียบร้อยแล้วหรืออาจกล่าวได้ว่าถ้าบิต RI ถูกเซตเมื่อใด หมายถึง ข้อมูลได้เข้ามาเก็บไว้ที่รีจิสเตอร์ SBUF จนครบทั้ง 8 บิตแล้วสามารถที่จะอ่านข้อมูลจากรีจิสเตอร์ SBUF ได้

รีจิสเตอร์ SBUF (Serial Data Buffer Register) เป็นรีจิสเตอร์ขนาด 8 บิตหรือ 1 ไบต์มีแอดเดรสอยู่ตำแหน่งที่ 99H และเข้าถึงข้อมูลแบบไบต์ได้โดยตรง จะทำหน้าที่รับและส่งข้อมูลออกไปยังพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ ในการอ่านค่าข้อมูลจากภายนอกที่รับเข้ามาทางพอร์ตอนุกรมจะต้องอ่านค่าจากรีจิสเตอร์ SBUF ซึ่งเป็นบัฟเฟอร์ (Buffer) ในการเก็บข้อมูลที่รับเข้ามาได้จากภายนอกและในทำนองเดียวกันขณะที่ต้องการส่งข้อมูล ก็จะนำเอาค่าข้อมูลที่ส่งออกไปไว้ที่ในรีจิสเตอร์ SBUF ก่อนและหลังจากนั้นจึงจะส่งออกไป โดยจะใช้คำสั่งการโอนย้ายข้อมูลแบบไบต์ การรับข้อมูลในโหมด 0 จะเริ่มต้นรับเมื่อค่าของบิต RI = 0 และ REN = 1 ส่วนในโหมดอื่นๆ การรับข้อมูลจะเริ่มต้นเมื่อกำหนดบิต REN = 1 และมี Start Bit เข้ามาที่ขา RXD

รีจิสเตอร์ PCON (Power Control) เป็นรีจิสเตอร์ขนาด 1 ไบต์มีแอดเดรสอยู่ตำแหน่งที่ 87H เข้าถึงข้อมูลได้แบบไบต์โดยตรงเท่านั้น โดยจะประกอบด้วยบิตดังต่อไปนี้

ตารางที่ 2.5 แสดงรีจิสเตอร์ PCON เพื่อกำหนดอัตราการรับส่งข้อมูล

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| SMOD  | -     | -     | -     | GF1   | GF0   | PD    | IDL   |

PCON.7 SMOD : ในกรณีที่ใช้โหมด 1 เป็นตัวกำหนดอัตรารับส่ง (Baud Rate) และหากกำหนดให้บิตนี้มีค่าเป็น "0" ในการใช้งานกับพอร์ตสื่อสารอนุกรมโหมด 1, 2 และโหมด 3 ค่าอัตรารับส่ง (Baud Rate) จะเพิ่มขึ้นเป็นสองเท่า รีจิสเตอร์ PCON ไม่สามารถอ้างตำแหน่งแบบบิตได้ แต่จะใช้คำสั่งทางลอจิกของการ OR และการกำหนดให้บิตมีสถานะเป็น "0" หรือเคลียร์บิตจะใช้การ AND แทน

## 2.11 อินเทอร์รัปต์ (Interrupt)

การอินเทอร์รัปต์ คือ การขัดจังหวะการทำงานของซีพียูเพื่อให้ซีพียูหยุดการประมวลผลในปัจจุบันเอาไว้ชั่วครู่ก่อน จากนั้นจะกระโดดไปทำงานในฟังก์ชันหรือชุดคำสั่งของการอินเทอร์รัปต์ฟังก์ชันหรือชุดคำสั่งที่ซีพียูกระโดดไปประมวลผลเมื่อเกิดการอินเทอร์รัปต์ จะเรียกว่าโปรแกรมสำหรับบริการการอินเทอร์รัปต์ (Interrupt Service Routine : ISR) เมื่อซีพียูประมวลผลชุดคำสั่งในโปรแกรมสำหรับบริการการอินเทอร์รัปต์เสร็จสิ้นซีพียูจะกลับไปประมวลผลโปรแกรมที่ได้หยุดไว้ก่อนหน้าที่จะมีการอินเทอร์รัปต์ การเกิดอินเทอร์รัปต์ของไมโครคอนโทรลเลอร์ตระกูล MCS-51 ที่นิยมใช้กันทั่วไปจะเกิดขึ้นได้จาก 6 แหล่งกำเนิดสัญญาณอินเทอร์รัปต์ แต่มีเวกเตอร์ของการอินเทอร์รัปต์เพียง 5 เวกเตอร์ เนื่องจากการอินเทอร์รัปต์ที่เกิดจากพอร์ตอนุกรมทั้งการรับข้อมูลและการส่งข้อมูลจะใช้เวกเตอร์ร่วมกัน การอินเทอร์รัปต์ที่เกิดขึ้นได้ 2 ลักษณะ คือ การอินเทอร์รัปต์จากภายนอกและการอินเทอร์รัปต์จากภายใน

## 2.12 รีจิสเตอร์ที่เกี่ยวข้องกับกระบวนการอินเทอร์รัปต์

รีจิสเตอร์ที่คอยควบคุมการทำงาน ได้แก่ รีจิสเตอร์ IE (Interrupt Enable Register), รีจิสเตอร์ IP (Interrupt Priority Register) และรีจิสเตอร์ TCON (Timer Control Register) โดยจะอธิบายเป็นลำดับดังนี้

2.12.1 รีจิสเตอร์ IE Interrupt Enable Register (Bit Addressable) เป็นรีจิสเตอร์ฟังก์ชันพิเศษ (SFR) ที่อยู่ในตำแหน่งแอดเดรส A8H มีขนาด 8 บิตและสามารถเข้าถึงในระดับบิตได้ ในสถานะเริ่มต้นของไมโครคอนโทรลเลอร์หลังการรีเซตจะกำหนดให้ค่าของรีจิสเตอร์ IE มีค่า 0XX0000 เป็นค่าเริ่มต้น (AT89CX051) แต่ในส่วนของการเรียนรู้ตอนนี้จะเรียนรู้เพื่อใช้งานเพียงแค่ 3 บิต คือ EA, EX1 และ EX0

ตารางที่ 2.6 แสดงบิตต่างๆ ของรีจิสเตอร์ IE

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

IE : Interrupt Enable Register (Bit Addressable)

EA หรือ IE.7 (Enable/Disable All Interrupt) หากกำหนดให้บิตนี้มีค่าเป็นสถานะลอจิก "0" จะไม่มีการตอบรับอินเทอร์รัปต์ทั้งหมดและหากต้องการให้มีการตอบรับอินเทอร์รัปต์จะต้องเซตให้บิตที่ EA เป็นสถานะลอจิก "1" และเซตบิตควบคุมแต่ละอินเทอร์รัปต์ที่ต้องการใช้อินเทอร์รัปต์ให้เป็น "1" ด้วย โดยใช้คำสั่งเซตบิตหรือเคลียร์บิต

EX1 หรือ IE.2 เป็นบิตที่ทำหน้าที่ควบคุมให้ทำการตอบรับการอินเทอร์รัปต์จากภายนอก INT1 หรือไม่ (ถ้ากำหนดให้บิต EX1 เป็นสถานะลอจิก "1" จะเป็นการตอบรับการอินเทอร์รัปต์ที่ INT1 แต่ถ้ากำหนดให้ เป็นสถานะลอจิก "1" จะไม่ตอบรับการอินเทอร์รัปต์ที่ INT1)

EX0 หรือ IE.0 เป็นบิตที่ทำหน้าที่ควบคุมให้ทำการตอบรับการอินเทอร์รัปต์จากภายนอก INTO หรือไม่ (ถ้ากำหนดให้บิต EX0 เป็นสถานะลอจิก "1" จะเป็นการตอบรับการอินเทอร์รัปต์ที่ INTO แต่ถ้ากำหนดให้ เป็นสถานะลอจิก "0" จะไม่ตอบรับการอินเทอร์รัปต์ที่ INTO)

### 2.12.2 รีจิสเตอร์ IP : Interrupt Priority Register

เป็นรีจิสเตอร์ฟังก์ชันพิเศษ (SFR) ที่อยู่ในตำแหน่งแอดเดรสที่ B8H และมีขนาด 8 บิต สามารถเข้าถึงในระดับบิตได้ ในสถานะเริ่มต้นของไมโครคอนโทรลเลอร์หลังการรีเซตจะกำหนดให้ค่าของรีจิสเตอร์ IP มีค่า XXX0000 เป็นค่าเริ่มต้น (AT89CXX051) การจัดลำดับความสำคัญการอินเทอร์รัปต์ของไมโครคอนโทรลเลอร์จะจัดได้ 2 ระดับแตกต่างกัน โดยทำการเซตค่าในรีจิสเตอร์ควบคุมการจัดลำดับความสำคัญ (IP : Interrupt Priority Register) และในกรณีที่ 'ไม่มีการจัดลำดับความสำคัญของการอินเทอร์รัปต์หรือจัดให้มีความสำคัญในระดับเดียวกัน ไมโครคอนโทรลเลอร์ก็จะจัดให้มีลำดับความสำคัญของสัญญาณอินเทอร์รัปต์ตามลำดับความสำคัญจากสูงไปต่ำดังแสดงในตารางที่ 2.7 เพื่อแก้ปัญหาการขออินเทอร์รัปต์ในระดับเดียวกัน และเกิดขึ้นพร้อมกัน

ตารางที่ 2.7 แสดงลำดับความสำคัญของการอินเทอร์รัปต์

|               |                      |
|---------------|----------------------|
| IE0           | External Interrupt 0 |
| TF0           | Timer 0              |
| IE1           | External Interrupt 1 |
| TF1           | Timer 1              |
| RI หรือ TI    | Serial Port          |
| TF2 หรือ EXF2 | Timer 2              |

## ตารางที่ 2.8 แสดงบิตต่างๆ ของรีจิสเตอร์ IP

| IP.7 | IP.6 | IP.5 | IP.4 | IP.3 | IP.2 | IP.1 | IP.0 |
|------|------|------|------|------|------|------|------|
| -    | -    | PT2  | PS   | PT1  | PX1  | PT0  | PX0  |

การกำหนดลำดับความสำคัญของการอินเทอร์รัพท์ที่กำหนดได้จากบิตต่างๆ ของรีจิสเตอร์ IP โดยบิตที่มีค่าเป็น 1 จะมีลำดับความสำคัญสูงกว่า

PX1 IP.2 การกำหนดลำดับความสำคัญของสัญญาณ INT1 (External Interrupt 1)

PX0 IP.0 การกำหนดลำดับความสำคัญของสัญญาณ INT0 (External Interrupt 0)

หากบิตใดมีค่าสถานะลอจิกเป็น “1” จะมีลำดับความสำคัญสูงแต่ถ้าหากไม่มีการจัดลำดับความสำคัญแล้ว การอินเทอร์รัพท์จาก INTO จะมีลำดับสูงกว่าการอินเทอร์รัพท์แบบ INT1 ซึ่งเป็นค่าเริ่มต้นและเมื่อมีการกำหนดลำดับความสำคัญของการอินเทอร์รัพท์แตกต่างกันแล้ว หากมีการร้องขออินเทอร์รัพท์เกิดขึ้นพร้อมกันจากแหล่งกำเนิด 2 แหล่งกำเนิดที่มีระดับความสำคัญแตกต่างกัน ไมโครคอนโทรลเลอร์จะตอบรับการร้องขอการอินเทอร์รัพท์จากแหล่งกำเนิดสัญญาณอินเทอร์รัพท์ที่มีระดับความสำคัญสูงกว่าและสัญญาณการร้องขออินเทอร์รัพท์ที่มีระดับความสำคัญสูงกว่าจะสามารถร้องขอการอินเทอร์รัพท์ซ้อน ในขณะที่ไมโครคอนโทรลเลอร์กำลังทำโปรแกรมตอบสนองการอินเทอร์รัพท์ของสัญญาณที่มีความสำคัญต่ำกว่าได้

ในขณะที่ไมโครคอนโทรลเลอร์กำลังทำโปรแกรมตอบสนองการอินเทอร์รัพท์อยู่ จะไม่สามารถตอบรับการร้องขออินเทอร์รัพท์จากแหล่งกำเนิดระดับที่ต่ำกว่าหรือระดับเดียวกันได้อีกจนกว่าจะจบโปรแกรมตอบสนองการอินเทอร์รัพท์ก่อน

### 2.12.3 รีจิสเตอร์ TCON (Timer Control Register)

เป็นรีจิสเตอร์ที่ทำหน้าที่ควบคุมการเลือกลักษณะสัญญาณอินเทอร์รัพท์จากภายนอก และเป็นส่วนของรีจิสเตอร์ฟังก์ชันพิเศษ(SFR) ที่อยู่ในตำแหน่งแอดเดรสที่ 88H และมีขนาด 8 บิตสามารถเข้าถึงในระดับบิตได้ ในสภาวะเริ่มต้นของไมโครคอนโทรลเลอร์หลังการรีเซต จะกำหนดให้ค่าของรีจิสเตอร์ TCON มีค่า 0000000 เป็นค่าเริ่มต้น สัญญาณการอินเทอร์รัพท์ของ Timer โดยรีจิสเตอร์นี้จะทำหน้าที่เป็นแฟล็กแสดงสถานะการทำงาน 4 บิตที่เซต และเคลียร์ด้วยการทำงานของฮาร์ดแวร์ ส่วนที่เหลืออีก 4 บิตเป็นบิตควบคุมการทำงานของ Timer ซึ่งเราเซตและเคลียร์ด้วยคำสั่งทางซอฟต์แวร์จากการใช้คำสั่งการเซตบิต หรือการโอนย้ายข้อมูลก็ได้ บิตต่างๆ ของรีจิสเตอร์ TCON ส่วนของที่เป็น Timer มีลักษณะการทำงานดังนี้

## ตารางที่ 2.9 แสดงบิตต่างๆ ของรีจิสเตอร์ TCON

| TCON.7 | TCON.6 | TCON.5 | TCON.4 | TCON.3 | TCON.2 | TCON.1 | TCON.0 |
|--------|--------|--------|--------|--------|--------|--------|--------|
| TF1    | TR1    | TF0    | TR0    | IE1    | IT1    | IE0    | IT0    |

Timer/counter control register : TCON (Bit addressable)

**IE1:TCON.3** (External Interrupt 1 Edge Flag) เป็นแฟล็กการร้องขออินเทอร์รัปต์ภายนอกของสัญญาณ INT1 บิตนี้จะถูกเซตด้วยฮาร์ดแวร์เมื่อมีสัญญาณอินเทอร์รัปต์เข้ามาที่ขา INT1 และบิตนี้จะถูกเคลียร์อัตโนมัติเมื่อไมโครคอนโทรลเลอร์กระโดดไปทำโปรแกรมตอบสนองการอินเทอร์รัปต์ของสัญญาณ INT1 ซึ่งอยู่ในตำแหน่งเริ่มต้นที่แอดเดรส 0013H ของหน่วยความจำโปรแกรม

**IT1:TCON.2** (Interrupt 1 Type Control Bit) เป็นบิตควบคุมการเลือกรูปแบบการแอกทีฟของสัญญาณอินเทอร์รัปต์ จากภายนอกที่ขา INT1 ซึ่งเราสามารถเซตหรือเคลียร์ด้วยซอฟต์แวร์ดังนี้  
 1 = เลือกใช้สัญญาณอินเทอร์รัปต์ INT1 แอกทีฟที่เปลี่ยนจาก “1” ไป “0” (ขอบขาลง)  
 0 = เลือกใช้สัญญาณอินเทอร์รัปต์ INT1 แอกทีฟที่ลอคจิก “0” (Level)

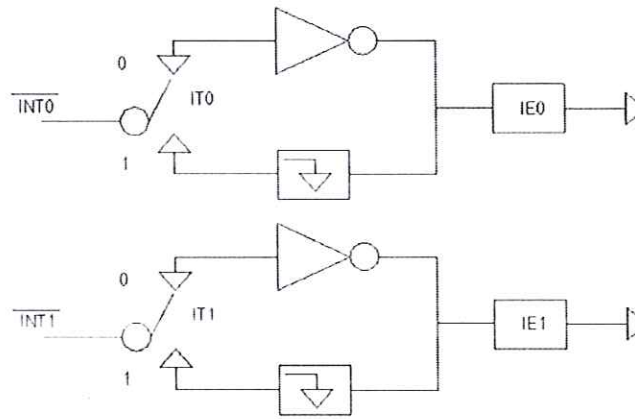
**IE0:TCON.1** (External Interrupt 0 Edge Flag) เป็นแฟล็กการร้องขออินเทอร์รัปต์จากสัญญาณ INT0 บิตนี้จะถูกเซตด้วยฮาร์ดแวร์เมื่อมีสัญญาณอินเทอร์รัปต์เข้ามาที่ขา INTO และบิตนี้จะถูกเคลียร์อัตโนมัติเมื่อไมโครคอนโทรลเลอร์กระโดดไปทำโปรแกรมตอบสนองการอินเทอร์รัปต์ของสัญญาณ INTO ซึ่งอยู่ในตำแหน่งแอดเดรส 0003H ของหน่วยความจำโปรแกรม  
**IT0:TCON.0** (Interrupt 0 Type Control Bit) เป็นบิตควบคุมการเลือกรูปแบบการแอกทีฟของสัญญาณอินเทอร์รัปต์จากภายนอกที่ขา INTO ซึ่งสามารถเซตหรือเคลียร์ด้วยซอฟต์แวร์

1 = เลือกใช้สัญญาณอินเทอร์รัปต์ INTO แอกทีฟที่เปลี่ยนจาก “1” ไป “0” (ขอบขาลง)  
 0 = เลือกใช้สัญญาณอินเทอร์รัปต์ INTO แอกทีฟที่ลอคจิก “0” (Level)

สามารถเลือกลักษณะสัญญาณร้องขอการอินเทอร์รัปต์ได้ 2 ลักษณะ โดยเลือกรีจิสเตอร์ TCON

1. เลือกใช้สัญญาณแอกทีฟที่ระดับลอคจิก “0” หรือ Low Level Triggered
2. เลือกใช้ที่ขอบขาของสัญญาณที่เปลี่ยนแปลงของสถานะจากลอคจิก “1” ไป “0”

(Falling Edge)



รูปที่ 2.7 แสดงการเลือกใช้สัญญาณแอดที่พีที่ระดับลอจิก “0” หรือที่ขอบขาลงของสัญญาณ

สัญญาณต่างๆ ของการร้องขออินเทอร์รัปต์ทั้งหมด สามารถเซตหรือเคลียร์ได้ด้วยซอฟต์แวร์เช่นเดียวกับฮาร์ดแวร์ ดังนั้นการร้องขออินเทอร์รัปต์จึงสามารถสร้างหรือยกเลิกได้ด้วยซอฟต์แวร์ แหล่งกำเนิดสัญญาณอินเทอร์รัปต์ทั้งหมดสามารถควบคุมให้ทำการร้องขออินเทอร์รัปต์หรือไม่ก็ได้ โดยการเซตหรือเคลียร์บิตต่างๆที่อยู่ใน รีจิสเตอร์ IE (Interrupt Enable) ภายในรีจิสเตอร์ IE มีบิต EA ที่ทำหน้าที่ควบคุมให้ไมโครคอนโทรลเลอร์ตอบรับการร้องขออินเทอร์รัปต์หรือไม่ตอบรับทั้งหมด หากกำหนดให้บิต EA เป็น “0” ไมโครคอนโทรลเลอร์จะไม่ตอบรับการร้องขออินเทอร์รัปต์ทั้งหมด โปรแกรมการตอบสนองของอินเทอร์รัปต์จะต้องทำคำสั่งเริ่มต้นของโปรแกรม ถ้ามีการใช้อินเทอร์รัปต์หากความยาวของโปรแกรมตอบสนองอินเทอร์รัปต์มีความยาวมากกว่า 8 ไบต์ ต้องนำไปเขียนเป็นโปรแกรมย่อยไว้ภายนอกแล้วใช้วิธีการเรียกโปรแกรมย่อยมาทำงาน เพราะระยะห่างตำแหน่งแอดเดรสของโปรแกรมตอบสนองอินเทอร์รัปต์แต่ละตัวจะมีระยะห่างแค่ 8 ไบต์

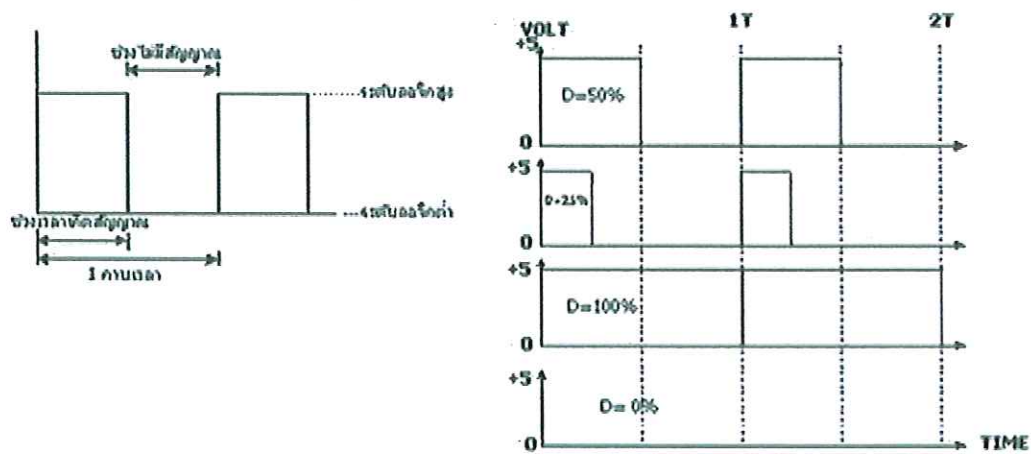
### 2.13 การควบคุมความเร็วของมอเตอร์กระแสตรง

การควบคุมความเร็วของมอเตอร์กระแสตรงมีหลายวิธีด้วยกัน ซึ่งอาจจะใช้วิธีการควบคุมแบบพื้นฐานทั่วไปเช่นการควบคุมด้วยวิธีการใช้ตัวต้านทานปรับค่าโดยต่ออนุกรมกับมอเตอร์หรือใช้วิธีการการควบคุมโดยการเปลี่ยนค่าของระดับแรงดันที่ป้อนให้กับมอเตอร์ การควบคุมในวิธีดังกล่าวถึงแม้ว่าจะควบคุมความเร็วมอเตอร์ให้คงที่ได้ แต่ที่ความเร็วต่ำจะส่งผลให้แรงบิดต่ำไปด้วย ดังนั้นเราจึงเลือกใช้วิธีการควบคุมโดยการจ่ายกระแสไฟให้กับมอเตอร์เป็นช่วงๆ โดยอาศัยกระแสไฟที่ป้อนให้กับมอเตอร์ให้เป็นค่าเฉลี่ยที่เกิดขึ้นในแต่ละช่วง ซึ่งเราเรียกว่าวิธีการของการมอดูเลชันทางความกว้างของพัลส์ PWM (Pulse Width Modulation)

### 2.14 วิธีการมอดูเลชันทางความกว้างของพัลส์ (PWM)

การมอดูเลชันทางความกว้างของพัลส์ จะเป็นการปรับเปลี่ยนที่สัดส่วนและความกว้างของสัญญาณพัลส์ โดยความถี่ของสัญญาณพัลส์จะไม่มีการเปลี่ยนแปลงหรือเป็นการเปลี่ยนแปลงที่ค่าของดิวตี้ไซเคิล (Duty Cycle) นั้นเอง ซึ่งค่าของดิวตี้ไซเคิล คือ ช่วงความกว้างของพัลส์ที่มีสถานะลอจิกสูง โดยคิดสัดส่วนเป็นเปอร์เซ็นต์จากความกว้างของพัลส์ทั้งหมด ยกตัวอย่าง เช่น ถ้าหากค่าดิวตี้ไซเคิลมีค่าเท่ากับเท่ากับ 50% ก็ หมายถึง ใน 1 วัฏสัญญาณพัลส์จะมีช่วงของสัญญาณที่เป็นสถานะลอจิกสูงอยู่ครึ่งหนึ่งและสถานะลอจิกต่ำอยู่อีกครึ่งหนึ่ง ดังรูป 2.8 และในทำนองเดียวกันถ้าหากค่าดิวตี้ไซเคิลมีค่ามาก หมายความว่าความกว้างของพัลส์ที่เป็นสถานะลอจิกสูงจะมีความกว้างมากขึ้น หากค่าดิวตี้ไซเคิลมีค่าเท่ากับ 100% ก็หมายความว่า จะไม่มีสถานะลอจิกต่ำเลย ซึ่งค่าดิวตี้ไซเคิลสามารถ จะหาได้จากค่าความสัมพันธ์ดังนี้

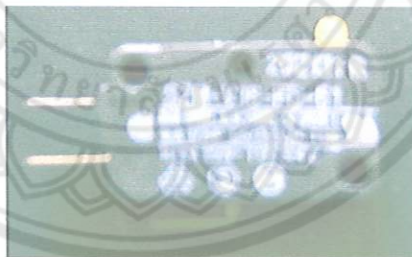
$$\text{ค่าดิวตี้ไซเคิล} = \frac{\text{ช่วงของสัญญาณพัลส์}}{\text{คาบเวลาทั้งหมดของสัญญาณ}} \times 100\% \quad (2.1)$$



รูปที่ 2.8 แสดงความกว้างของพัลส์ที่ขนาดต่างๆ และค่าความถี่ไซเคิลของช่วงพัลส์ที่มีความถี่คงที่

### 2.15 การทำงานตรวจจับของตัวรับสัญญาณ

การทำงานของตัวรับสัญญาณ คือ การตรวจเฝ้าดูจากนั้นก็ส่งสัญญาณเพื่อให้กระบวนการ (Process) ของลิฟต์ทำงานเป็นไปตามที่โปรแกรมกำหนดขั้นตอนการทำงานไว้ การตรวจจับวัตถุหรือการตรวจจับตำแหน่ง ซึ่งมีอยู่หลายชนิดในที่นี้จะกล่าวถึง ลิมิตสวิตช์จะทำงานโดยวัตถุหรือการตรวจจับจะต้องเคลื่อนที่เข้าไปสัมผัส กดก้านหรือปุ่มของลิมิตสวิตช์จนทำให้กลไกของลิมิตสวิตช์ทำงานทำให้น้ำสัมผัสทางไฟฟ้าต่อติดกันหรือจากออกแล้วนำสัญญาณไฟฟ้าที่ได้ไปควบคุมขั้นตอนอีกทีหนึ่ง



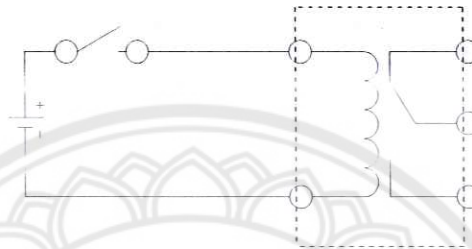
รูปที่ 2.9 ลิมิตสวิตช์



## 2.16 ส่วนประกอบและการทำงานของรีเลย์

ตัวรีเลย์มีส่วนประกอบที่สำคัญ คือ คอยล์สนามแม่เหล็กและหน้าสัมผัส การทำงานของรีเลย์ แสดงได้ดังรูปที่ 2.10 และรูปที่ 2.11 เมื่อทำการจ่ายกระแสไฟให้รีเลย์จะทำให้เกิดสนามแม่เหล็กดูดหน้าสัมผัส ทำให้หน้าสัมผัสเปลี่ยนสถานะไป หน้าสัมผัสของรีเลย์สามารถติดตั้งได้หลายชุด

รีเลย์ (Relay)



รูปที่ 2.10 การทำงานของรีเลย์ก่อนจ่ายแรงดัน

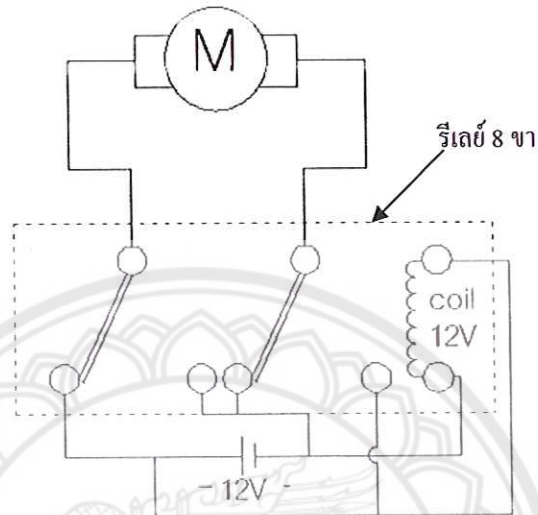
รีเลย์ (Relay)



รูปที่ 2.11 การทำงานของรีเลย์ขณะมีกระแสผ่านคอยล์

จากการทำงานของรีเลย์ทำให้สามารถนำรีเลย์มาต่อร่วมกับอุปกรณ์การวัดแบบหน้าสัมผัส (Switches) เช่น สวิตช์ความดัน (Pressure Switch), สวิตช์ระดับ (Level Switch), สวิตช์อุณหภูมิ (Temperature Switch) เป็นต้น เพื่อนำไปใช้ในการปิด/เปิด การทำงานของอุปกรณ์ต่างๆ ได้

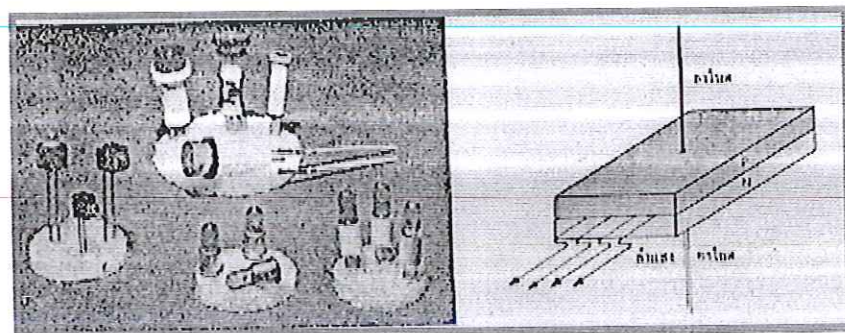
รีเลย์ชนิด 8 ขา หรือรีเลย์แบบ Double Poles Double Throws (DPDT) ขดลวด 12 V DC ทำหน้าที่เป็นสวิตช์ในการกลับทางหมุนของมอเตอร์ มีหน้าสัมผัสสองคู่ คือ ปกติปิด (NC) 1 คู่และ ปกติเปิด (NO) 1 คู่



รูปที่ 2.12 การทำงานของรีเลย์ 8 ขาในการกลับทางหมุนมอเตอร์

## 2.17 ไดโอดเปล่งแสง (LED)

ไดโอดเปล่งแสง (Light-Emitting Diode) เรียกย่อๆ ว่า LED คือ ไดโอดซึ่งสามารถเปล่งแสงออกมาได้แสงที่เปล่งออกมาประกอบด้วยคลื่นความถี่เดียวและเฟสต่อเนื่องกัน ซึ่งต่างกับแสงธรรมชาติที่ตาคนมองเห็น ประกอบด้วยคลื่นซึ่งมีเฟสและความถี่ต่างๆ กันมารวมกัน ไดโอดซึ่งสามารถให้แสงออกมาได้ทั้งชนิดที่เป็นสารกึ่งตัวนำ, ของเหลวและก๊าซ ในที่นี้จะกล่าวถึงชนิดที่เป็นสารกึ่งตัวนำเท่านั้น



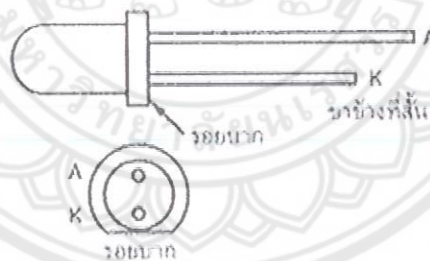
รูปที่ 2.13 ไดโอดเปล่งแสงและ โครงสร้างภายใน

ไดโอดชนิดนี้เหมือนไดโอดทั่ว ๆ ไปที่ประกอบด้วยสารกึ่งตัวนำชนิด P และ N ประกอบกันมีผิวข้างหนึ่งเรียบเป็นมันคล้ายกระจก เมื่อไดโอดถูกไบแอสตรง (Forward Bias) จะทำให้อิเล็กตรอนที่สารกึ่งตัวนำชนิด N มีพลังงานสูงขึ้นจนสามารถวิ่งข้ามรอยต่อไปรวมกับโฮลในสารกึ่งตัวนำชนิด P ต่อให้เกิดพลังงานในรูปของประจุโฟตอน ซึ่งจะเปล่งแสงออกมา การประยุกต์ LED ไปใช้งานอย่างกว้างขวางส่วนมากใช้ในภาคแสดงผล (Display Unit) LED โดยทั่วไปมี 2 ชนิดใหญ่ ๆ คือ LED ชนิดที่ตาคนเห็นได้กับชนิดที่ตาคนมองไม่เห็นต้องใช้ทรานซิสเตอร์มาเป็นตัวรับแสงแทนตาคน



รูปที่ 2.14 สัญลักษณ์ของ LED

## 2.18 การเลือกใช้ไดโอดเปล่งแสง (LED)

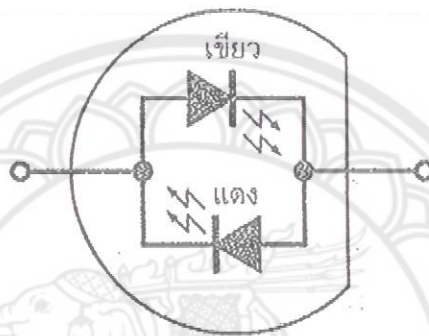


รูปที่ 2.15 ตำแหน่งขาของ LED

ปัญหาแรกที่จะพบในการใช้งาน LED คือ การหาขั้วของ LED ซึ่งโดยทั่ว ๆ ไป LED จะมีขั้วแคโทดอยู่ตรงด้านรอยบากหรือเป็นขาที่สั้นกว่าอีกข้างหนึ่ง ดังแสดงในรูปที่ 2.15 หรืออาจจะทดสอบหาขั้วโดยต่อกับแบตเตอรี่ก็ได้ โดยการทดลองกลับขั้วของ LED และถ้า LED ติดสว่าง ขั้วแคโทดจะเป็นขาที่ต่ออยู่กับขั้วลบของแบตเตอรี่และขั้วแอนโนดจะเป็นขาที่ต่ออยู่กับขั้วบวกของแบตเตอรี่

ในการใช้งาน LED อาจใช้ร่วมกับชุดยึัดจับ ซึ่งประกอบด้วยตัวยึดและวงแหวนพลาสติก เพื่อช่วยให้ LED สามารถยึดติดกับแผ่นวงจรพิมพ์หรือหน้าปัดของกล่องได้

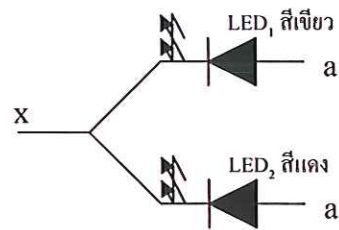
โดยทั่วไป LED จะผลิตเป็นแบบที่มี LED ตัวเดียวในตัวถังเดียว สำหรับ LED ชนิดที่มีหลายๆ ตัวอยู่ในตัวถังเดียวกันรู้จักกัน คือ LED 7 ส่วน (7-Segment Display) ซึ่งภายในประกอบด้วย LED เจ็ดตัว (หรือแปดตัว) อยู่ในตัวถังเดียวกันสามารถแสดงผลได้ทั้งตัวอักษรและตัวเลขตัวแสดงผลอีกอย่างเป็นตัวแสดงผลแบบบาร์กราฟ (Bargraph-Display) ซึ่งประกอบด้วย LED 10 ถึง 30 ตัวอยู่ในตัวถังเดียวกัน



4900170  
 ๑๕.  
 ๐/๒๗๐  
 ๒๕๔๘  
 C.2

รูปที่ 2.16 LED ชนิดสองสี ประกอบด้วย LED 2 ตัวต่อกลับขั้วและขนานกัน

LED ส่วนใหญ่สามารถเปล่งแสงออกมาได้เพียงสีเดียวต่อตัวเท่านั้น แต่สำหรับ LED ที่สามารถเปล่งแสง ได้หลายสีก็มีการผลิตขึ้นมาเช่นกัน ดังแสดงในรูปที่ 2.16 ซึ่งเป็น LED ชนิดสองสีจะประกอบด้วย LED 2 ตัวต่อกลับขั้วและขนานกันจะเปล่งแสงสีเขียวเมื่อได้รับการไบแอสในทิศทางหนึ่งและเปล่งแสงสีแดง หรือเหลืองเมื่อได้รับการไบแอส ในทิศทางตรงกันข้าม LED ชนิดนี้มีประโยชน์อย่างมากในการนำมาแสดงผลเกี่ยวกับทิศทางของกระแสหรือการแสดงขั้ว LED ชนิดหลายสีอีกชนิดหนึ่งแสดงดังรูปที่ 2.17 ซึ่งประกอบด้วย LED สีเขียวและสีแดงต่อกัน โดยขาแคโทดจะต่อรวมกัน LED ชนิดนี้จะเปล่งแสงสีเขียวหรือสีแดง เมื่อได้รับการไบแอสที่ขั้วแอนโอด ขั้วใดขั้วหนึ่ง และจะเปล่งแสงสีส้มและสีเหลือง เมื่อได้รับการไบแอสทั้ง 2 ขั้วของแอนโอดในอัตราส่วนของปริมาณกระแสที่แตกต่างกัน ดังแสดงในตารางของรูปที่ 2.17



| สีของ LED      | สีแดง | สีส้ม | สีเหลือง | สีเขียว |
|----------------|-------|-------|----------|---------|
| กระแสของ LED 1 | 0 mA  | 5 mA  | 10 mA    | 15 mA   |
| กระแสของ LED 2 | 5 mA  | 3 mA  | 2 mA     | 0 mA    |

รูปที่ 2.17 LED ซึ่งสามารถเปล่งแสงได้ถึง 4 สีโดยใช้ LED เพียง 2 ตัว

## 2.19 การประยุกต์ใช้งานไดโอดเปล่งแสง (LED)

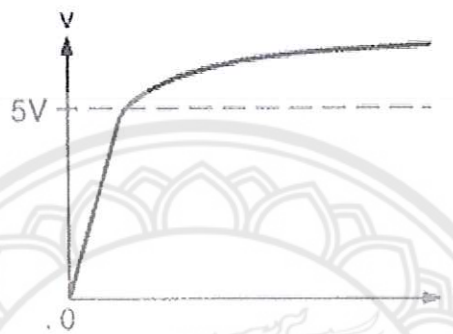
LED ไดโอดเปล่งแสงเป็นไดโอดชนิดพิเศษที่สามารถเปล่งแสงออกมาได้ ไม่ว่าจะเป็นแสงในช่วงที่สายตามองเห็นและมองไม่เห็น

เมื่อเปรียบเทียบกับหลอดไฟชนิดทั้งสแตนด์แล้ว LED จะมีประสิทธิภาพสูงกว่าประมาณ 10-15 เท่าและยังมีการตอบสนองต่อแสงที่เร็วกว่าด้วย คือ ประมาณ 0.1 ไมโครวินาที เมื่อเทียบกับหลอดไฟชนิดทั้งสแตนด์ ซึ่งใช้เวลาในระดับมิลลิวินาที ดังนั้นโดยทั่วไป LED จึงเหมาะที่จะใช้เป็นตัวแสดงผลหรือใช้เป็นไฟกะพริบ

ตารางที่ 2.10 ค่าแรงดันฟอร์เวิร์ด ไบแอสของ LED ที่กระแส  $I_f$  20 มิลลิแอมป์

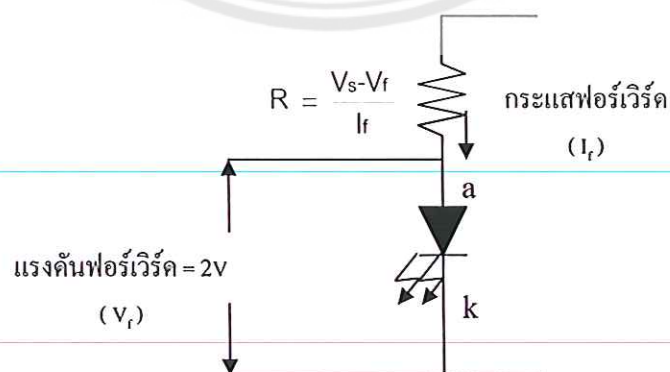
| LED                 | สีแดง | สีส้ม | สีเหลือง | สีเขียว |
|---------------------|-------|-------|----------|---------|
| ค่าแรงดันฟอร์เวิร์ด |       |       |          |         |
| $V_f$ (โวลต์)       | 1.8   | 2.0   | 2.1      | 2.2     |

เมื่อ LED ได้รับกระแสไบแอสตรงจะมีแรงดันตกคร่อม LED ประมาณ 2 โวลต์ ดังแสดงในตารางที่ 2.10 ซึ่งเป็นค่าแรงดันไบแอสตรงของ LED สีต่างๆ ที่กระแสไบแอสตรง 20 มิลลิแอมป์ และถ้า LED ได้รับกระแสไบแอสกลับ ก็จะเกิดปรากฏการณ์อะวาลานซ์หรือซีเนอร์เช่นเดียวกับไดโอดชนิดอื่นๆ แต่ค่าแรงดันนั้นจะมีค่าน้อยกว่า ดังแสดงในรูปที่ 2.18 โดยทั่วไปแล้ว LED จะทนแรงดันย้อนกลับได้สูงสุดประมาณ 3-5 โวลต์



รูปที่ 2.18 เมื่อ LED ได้รับไบแอสกลับกราฟที่ได้จะมีลักษณะคล้ายกับของซีเนอร์ไดโอด

ในการนำ LED ไปใช้งานจะต้องต่อตัวต้านทานจำกัดกระแสอนุกรมเข้ากับ LED ด้วย เพื่อป้องกันไม่ให้ LED เสียหาย รูปที่ 2.13 แสดงการหาค่าตัวต้านทาน R เพื่อให้ได้กระแสที่เหมาะสมแก่ LED โดย R จะต่อที่ขั้วแคโทดหรือแอโนดของ LED ก็ได้ ความสว่างของ LED จะเป็นสัดส่วนโดยตรงกับปริมาณกระแสที่ไหลผ่านตัวมัน โดยทั่วไป LED จะสามารถทนกระแสได้สูงสุดประมาณไม่เกิน 30-40 มิลลิแอมป์



รูปที่ 2.19 แสดงการหาค่าความต้านทานเมื่อทราบค่าแรงดัน  $V_f$  และกระแส  $I_f$

## 2.20 การวัดค่าแรงดันฟอร์เวิร์ดของ LED

เตรียมมัลติมิเตอร์ให้พร้อม โดยปรับย่านการวัดมาที่สัญลักษณ์รูปไดโอด เมื่อเราทราบว่ายาวเป็นขั้วแอโนดและขาสั้นเป็นขั้วแคโทด ทำการวัดโดยเอาสายสีแดงและขั้วแอโนดและสายสีดำและขั้วแคโทด ไฟจะสว่างขึ้นพร้อมแสดงค่าแรงดันฟอร์เวิร์ดที่จอแสดงผลดังรูปที่ 2.20



รูปที่ 2.20 การวัดค่าแรงดันฟอร์เวิร์ดของ LED



รูปที่ 2.21 LED ที่นำมาวัดค่าแรงดันฟอร์เวิร์ด

## 2.21 การคำนวณหาค่าความต้านทานอนุกรม

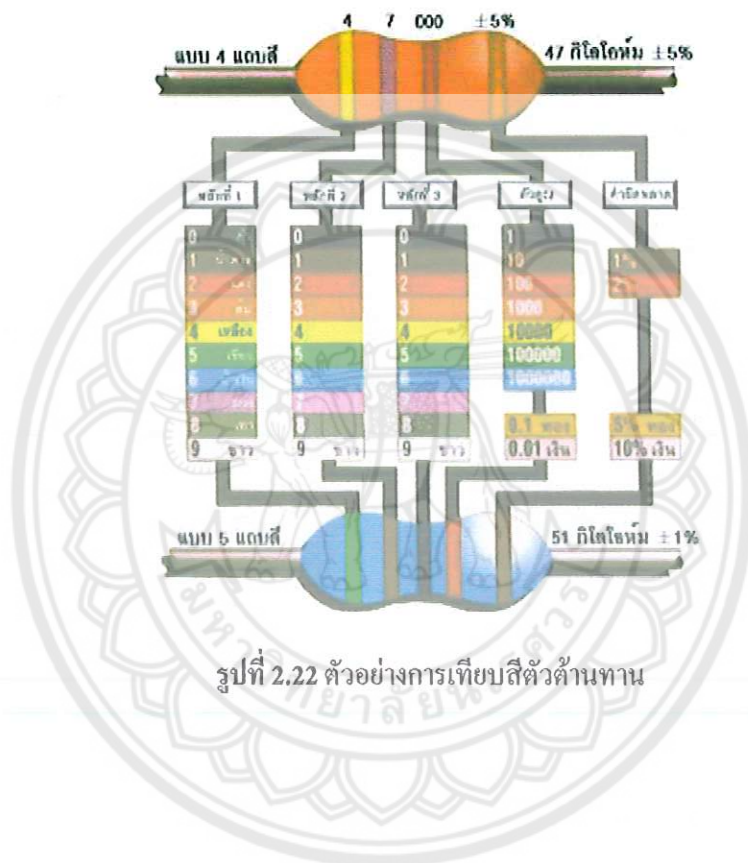
เนื่องจาก LED มีขีดจำกัดในการใช้งานซึ่งโดยทั่วไป LED สามารถทนกระแสได้สูงสุดไม่เกิน 30 – 40 มิลลิแอมป์ จึงต้องนำตัวต้านทานมาต่ออนุกรมเข้าไปเพื่อทำการลดกระแสสามารถคำนวณค่าความต้านทานจากสมการดังนี้

$$R = \frac{V_S - V_F}{I_F} \quad (2.2)$$

โดยที่

|       |     |                     |           |
|-------|-----|---------------------|-----------|
| R     | คือ | ค่าความต้านทาน      | (โอห์ม)   |
| $V_s$ | คือ | แรงดันของแหล่งจ่าย  | (โวลต์)   |
| $V_F$ | คือ | ค่าแรงดันฟอร์เวิร์ด | (โวลต์)   |
| $I_F$ | คือ | ค่ากระแสไบแอส       | (แอมแปร์) |

นำค่าความต้านทานที่ได้ไปเทียบแถบสีเพื่อหาค่าความต้านทานที่ต้องการต่อไป



รูปที่ 2.22 ตัวอย่างการเทียบสีตัวต้านทาน



### บทที่ 3

## การควบคุมระบบลิฟต์ด้วยไมโครคอนโทรลเลอร์

### 3.1 แผนผังการทำงานของการควบคุมระบบลิฟต์ด้วยไมโครคอนโทรลเลอร์

ลิฟต์แบบควบคุมอิสระด้วยไมโครคอนโทรลเลอร์ เป็นการควบคุมลิฟต์เพียงเครื่องเดียว

3.1.1 เริ่มต้นโดยการตรวจสอบว่ามีผู้กดเรียกลิฟต์หรือไม่

3.1.2 ทำการตรวจสอบว่าจะไปชั้นไหน

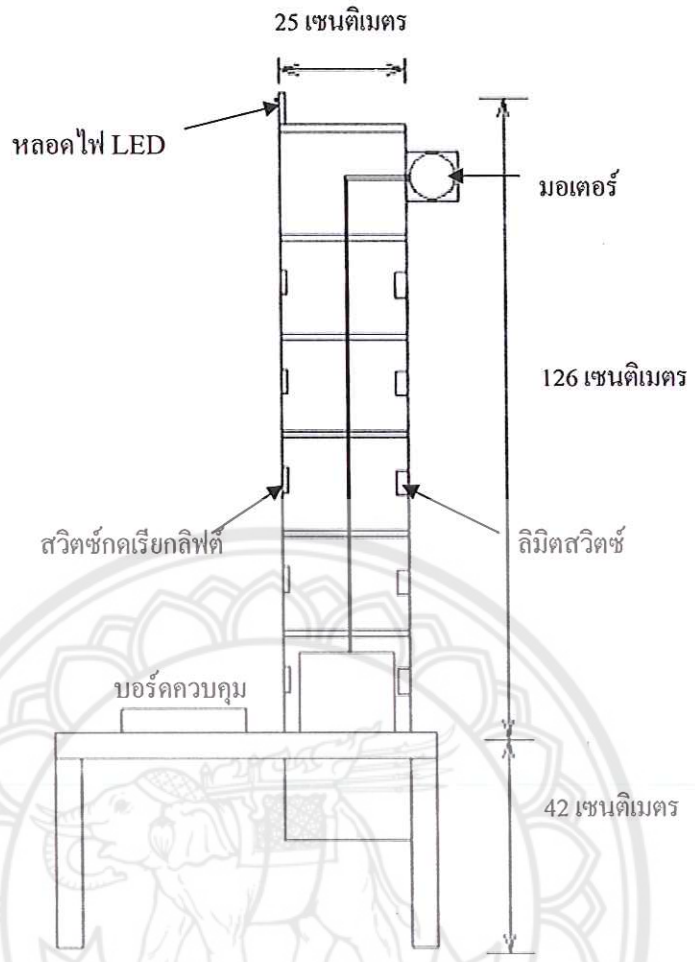
3.1.3 ทำการตรวจสอบว่าลิฟต์อยู่ชั้นไหนควรจะขึ้นหรือลง ซึ่งเป็นการควบคุมมอเตอร์ว่าให้มอเตอร์หมุนตามเข็มนาฬิกาหรือหมุนทวนเข็มนาฬิกา

3.1.4 หน่วงเวลาเพื่อรอลิฟต์ปิดประตู

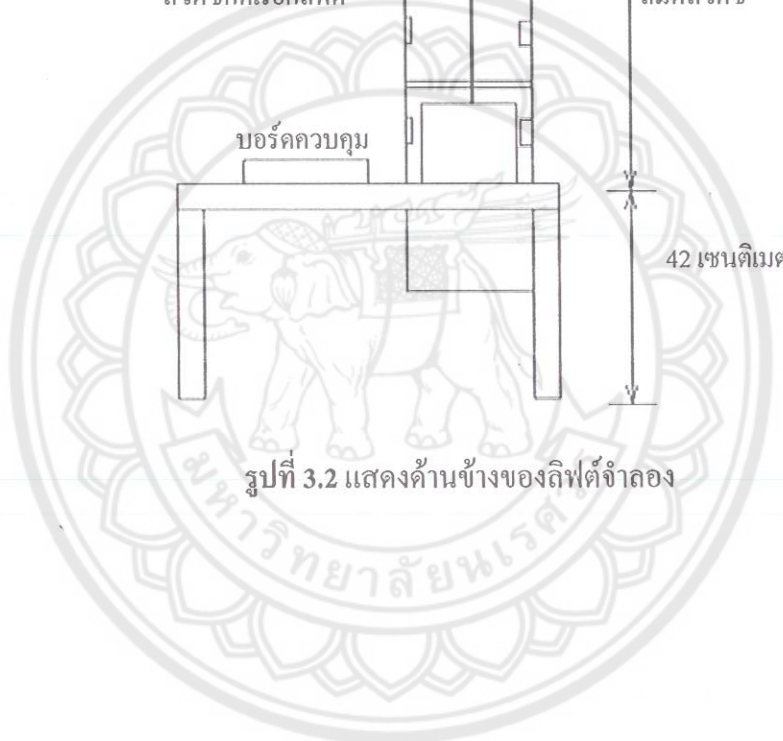
3.1.5 ลิฟต์เคลื่อนที่หยุดตามชั้นที่ต้องการ โดยใช้ลิมิตสวิตช์เป็นตัวส่งสัญญาณบอกว่าลิฟต์ไปถึงชั้นที่ต้องการหรือยัง

จากการที่ได้ศึกษาทฤษฎีและหลักการการทำงานของไมโครคอนโทรลเลอร์ในบทที่ 2 ซึ่งอาศัยหลักการและการคำนวณต่างๆ เพื่อนำมาเขียนโปรแกรมควบคุม โดยใช้แผนผังขั้นตอนการดำเนินงาน ดังนี้





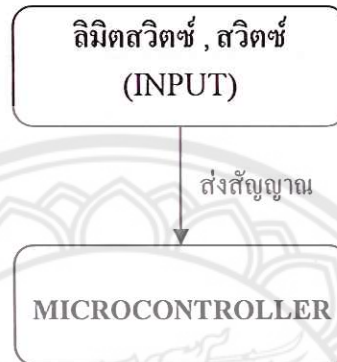
รูปที่ 3.2 แสดงด้านข้างของลิฟต์จำลอง



### 3.3 การออกแบบสำหรับตัวรับสัญญาณ (INPUT)

3.2.1 ลิ้มิตสวิตช์ ซึ่งทำหน้าที่ตรวจจับว่าลิฟต์อยู่ที่ชั้นไหน มีหลักการทำงาน คือ เมื่อลิฟต์เคลื่อนที่ผ่านลิ้มิตสวิตช์จะทำให้ลิ้มิตสวิตช์ถูกกดลงทำให้สถานะลอจิกเปลี่ยนจากลอจิก “1” เป็นลอจิก “0” และส่งสัญญาณไปยังไมโครคอนโทรลเลอร์

3.2.2 สวิตช์ มีไว้สำหรับกดเรียกลิฟต์ มีหลักการทำงาน คือ เมื่อมีการกดเรียกลิฟต์จะทำให้สถานะลอจิกเปลี่ยนจากลอจิก “1” เป็นลอจิก “0” และส่งสัญญาณไปยังไมโครคอนโทรลเลอร์

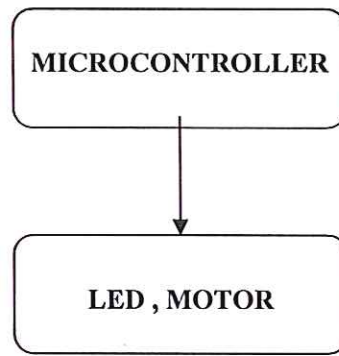


รูปที่ 3.3 การออกแบบ INPUT

### 3.4 การออกแบบ OUTPUT

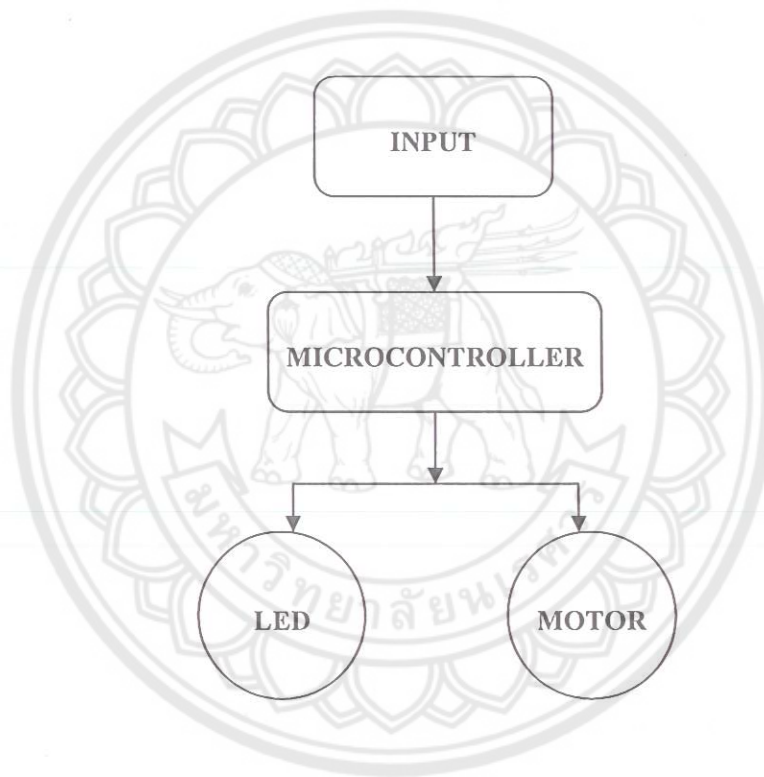
3.4.1 ไดโอดเปล่งแสง (LED) มีไว้สำหรับแสดงผล LED ในระบบลิฟต์จำลองมีอยู่ 2 หน้าที คือ LED แสดงผลว่าลิฟต์อยู่ที่ชั้นไหนและ LED แสดงผลเมื่อมีการกดเรียกลิฟต์ที่หน้าลิฟต์และที่บอร์ดควบคุม ซึ่งมีหลักการทำงานเหมือนกัน คือ เมื่อจ่ายแรงดันไฟที่ขั้วแอนโอด +5V และต่อขั้วแคโทดเข้ากับขาสวิตช์ข้างที่มีแรงดัน +5V เข้ามา ทำให้แรงดันตกคร่อมเป็นศูนย์และเมื่อสวิตช์ถูกกดจะทำให้แรงดันที่ขา +5V ถูกช้อดลงกราวด์เป็นผลทำให้มีกระแสไหลผ่าน LED จึงทำให้ LED สว่างขึ้น

3.4.2 มอเตอร์ ทำหน้าที่หมุนคังลิฟต์ขึ้นหรือลง มีหลักการทำงาน คือ เนื่องจากมอเตอร์เป็นแบบกระแสตรงการคังลิฟต์ขึ้นหรือลงสามารถทำได้ง่ายโดยการสลับขั้วของแหล่งจ่ายไฟจากบวกเป็นลบและจากลบเป็นบวก ในที่นี้การสลับขั้วแหล่งจ่ายไฟถูกควบคุมอัตโนมัติโดยรีเลย์ 8 ขา



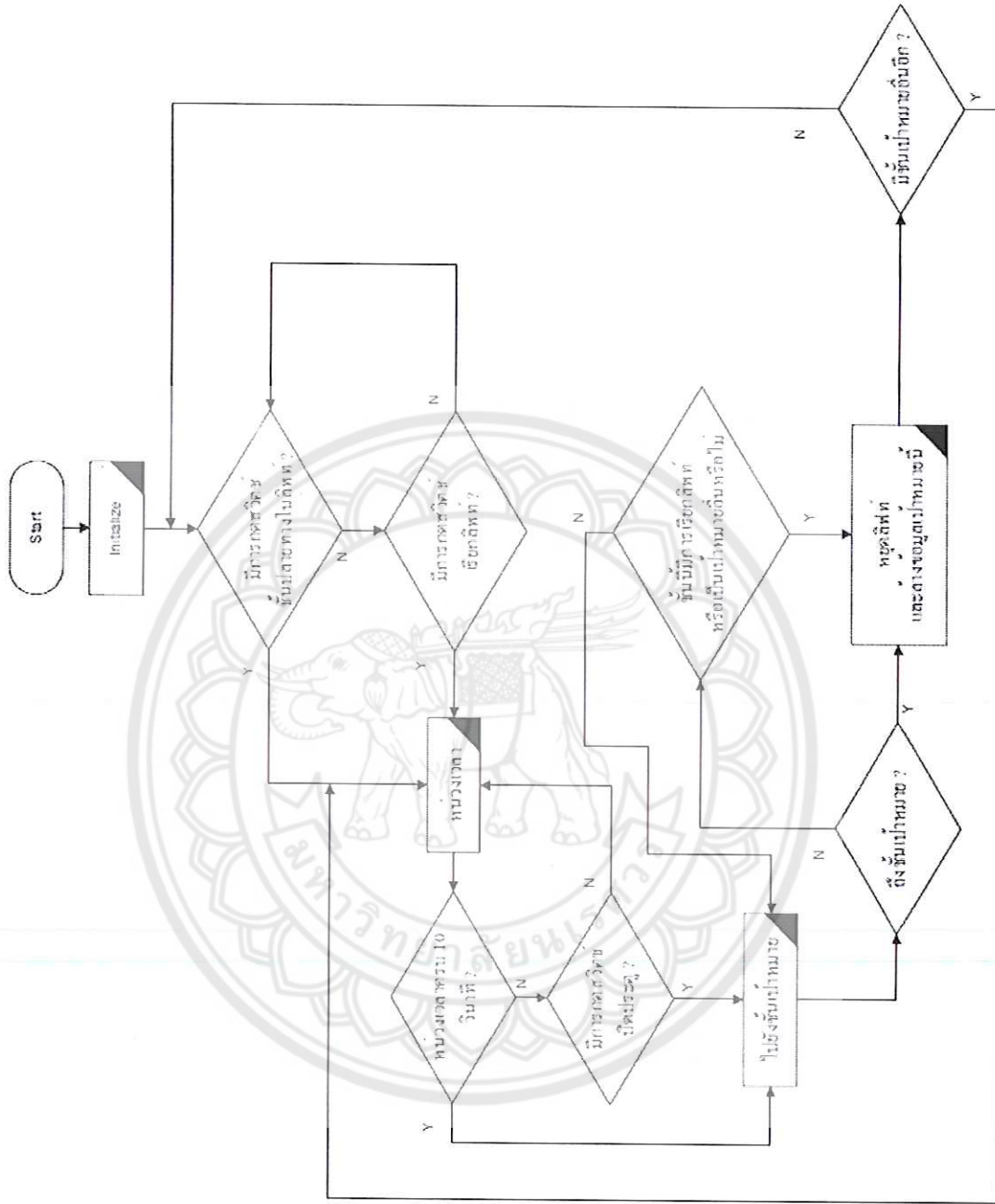
รูปที่ 3.4 การออกแบบ OUTPUT

### 3.5 การออกแบบ HARD WARE



รูปที่ 3.5 การออกแบบ HARD WARE

### 3.6 การออกแบบ PROGRAM



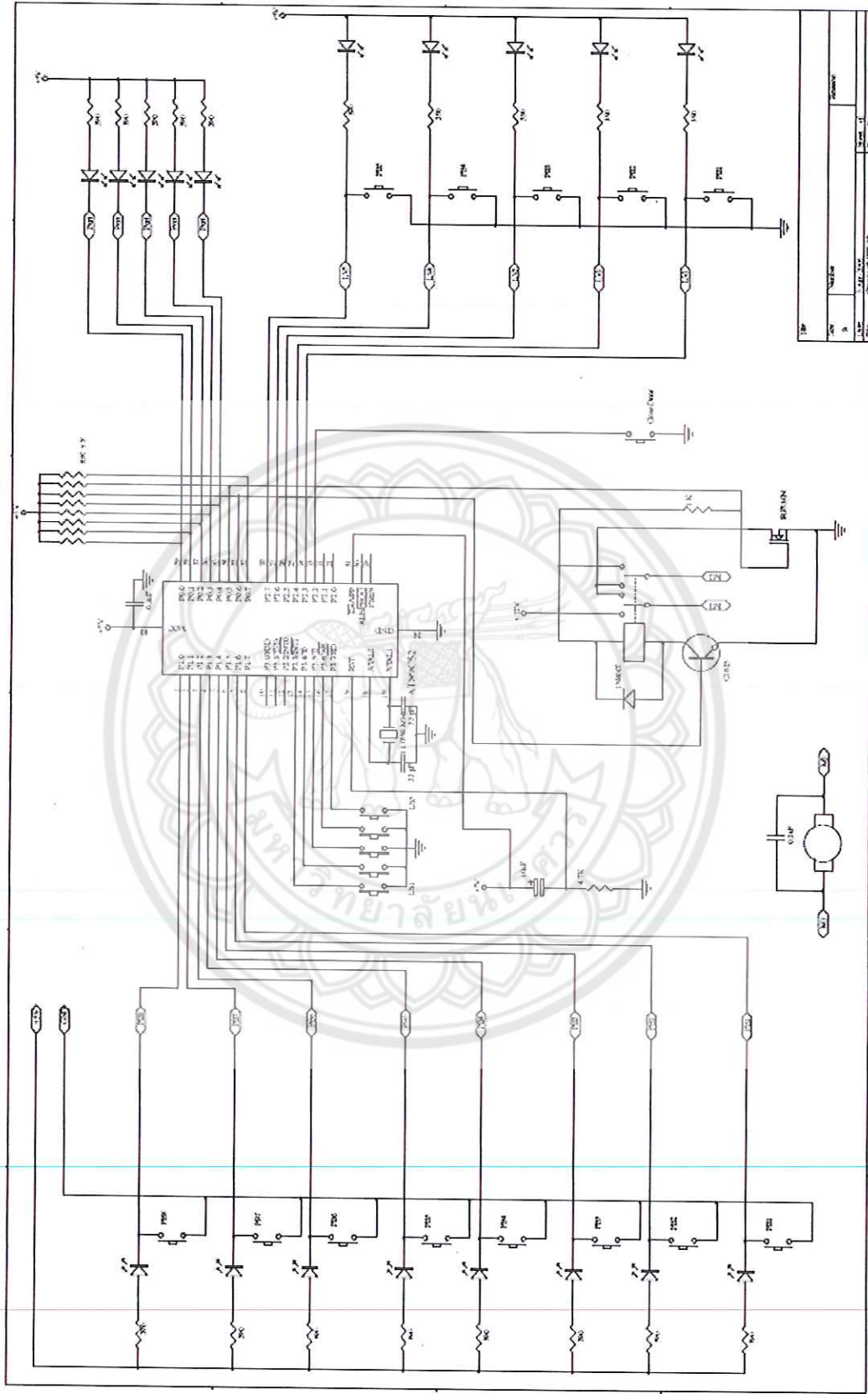
รูปที่ 3.6 แผนผังการควบคุมระบบลิฟต์ด้วยไมโครคอนโทรลเลอร์

### 3.7 การกำหนดอินพุตและเอาต์พุตของลิฟต์จำลอง

ตารางที่ 3.1 การกำหนดอินพุตและเอาต์พุตของลิฟต์จำลอง

| แอดเดรส | ความหมาย               |
|---------|------------------------|
| P 1.0   | สวิทช์เรียกลงชั้น 5    |
| P 1.1   | สวิทช์เรียกขึ้นชั้น 4  |
| P 1.2   | สวิทช์เรียกลงชั้น 4    |
| P 1.3   | สวิทช์เรียกขึ้นชั้น 3  |
| P 1.4   | สวิทช์เรียกลงชั้น 3    |
| P 1.5   | สวิทช์เรียกขึ้นชั้น 2  |
| P 1.6   | สวิทช์เรียกลงชั้น 2    |
| P 1.7   | สวิทช์เรียกขึ้นชั้น 1  |
| P 2.4   | สวิทช์เลือกชั้น 1      |
| P 2.5   | สวิทช์เลือกชั้น 2      |
| P 2.6   | สวิทช์เลือกชั้น 3      |
| P 2.7   | สวิทช์เลือกชั้น 4      |
| P 2.8   | สวิทช์เลือกชั้น 5      |
| P 2.2   | สวิทช์ปิดประตูลิฟต์    |
| P 0.6   | มอเตอร์หมุนซ้าย - ขวา  |
| P 0.5   | มอเตอร์หมุนซ้าย - เร็ว |
| P 0.0   | หลอดไฟแสดงชั้น 1       |
| P 0.1   | หลอดไฟแสดงชั้น 2       |
| P 0.2   | หลอดไฟแสดงชั้น 3       |
| P 0.3   | หลอดไฟแสดงชั้น 4       |
| P 0.4   | หลอดไฟแสดงชั้น 5       |
| P 3.3   | ลิมิตสวิทช์ชั้นที่ 1   |
| P 3.4   | ลิมิตสวิทช์ชั้นที่ 2   |
| P 3.5   | ลิมิตสวิทช์ชั้นที่ 3   |
| P 3.6   | ลิมิตสวิทช์ชั้นที่ 4   |
| P 3.7   | ลิมิตสวิทช์ชั้นที่ 5   |

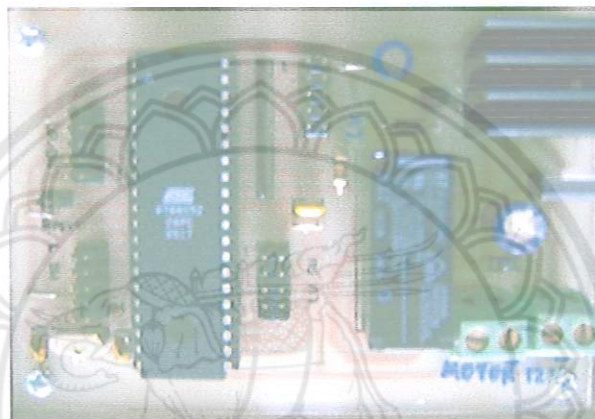
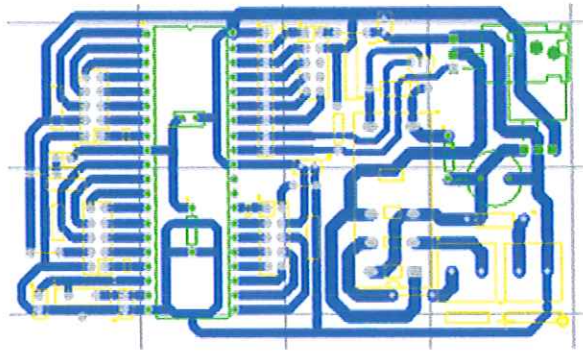
### 3.8 การออกแบบวงจรควบคุมระบบลิฟต์ด้วยไมโครคอนโทรลเลอร์



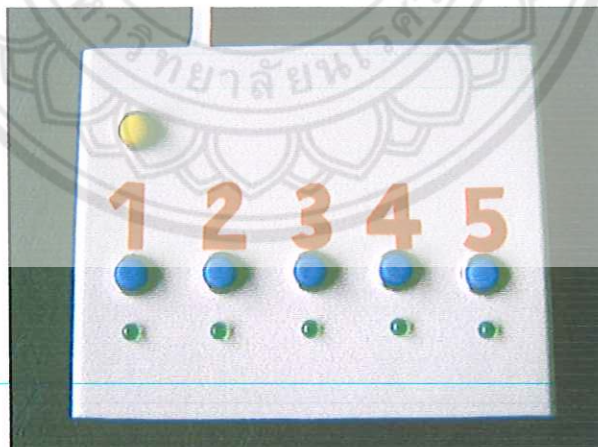
รูปที่ 3.7 การออกแบบวงจรควบคุมระบบลิฟต์ด้วยไมโครคอนโทรลเลอร์



### 3.9 บอร์ดควบคุมระบบลิฟต์ด้วยไมโครคอนโทรลเลอร์

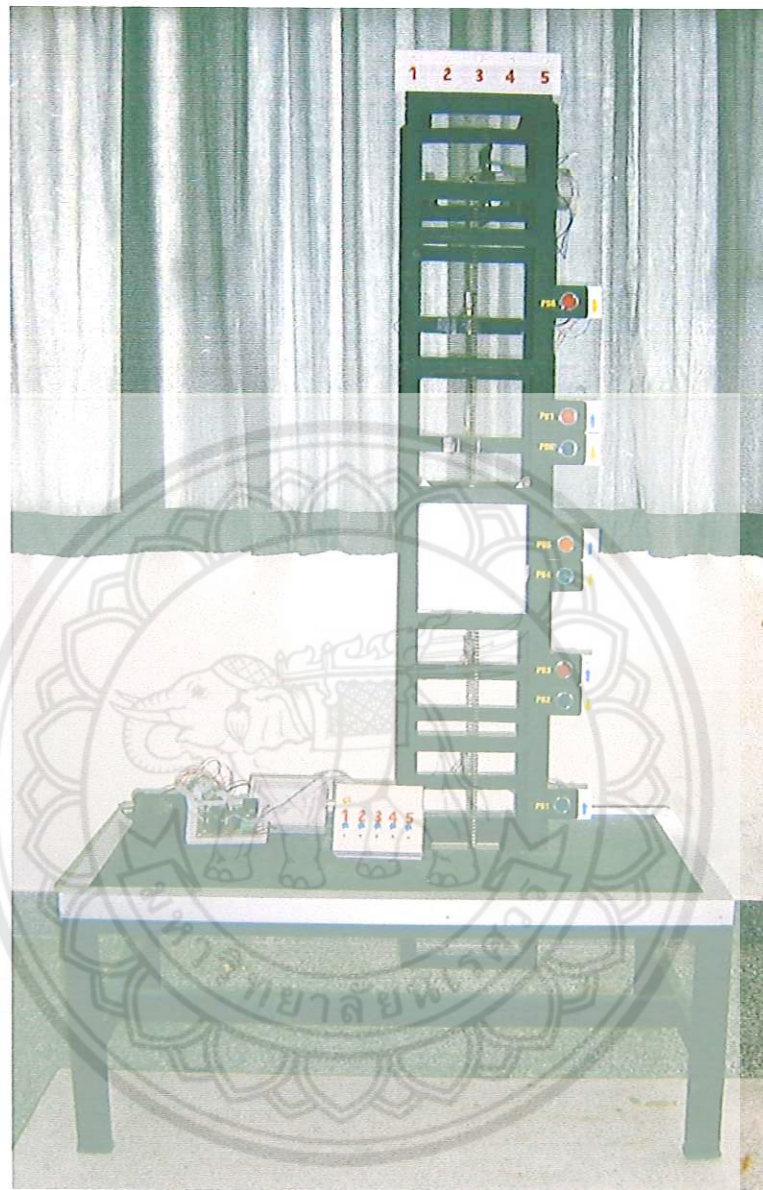


รูปที่ 3.8.1 บอร์ดไมโครคอนโทรลเลอร์ MCS-51



รูปที่ 3.8.2 บอร์ดสั่งงานลิฟต์

### 3.10 ลิฟต์จำลองที่ใช้งานจริง



รูปที่ 3.8.3 ลิฟต์จำลอง

## บทที่ 4

### ผลการทดลองและผลการวิเคราะห์

#### 4.1 การทดลองควบคุมระบบลิฟต์

การทดสอบ โปรแกรมเป็นการตรวจสอบความถูกต้องเพื่อให้สอดคล้องกับการทำงานของลิฟต์จำลอง

##### 4.1.1 จุดประสงค์

4.1.1.1 เพื่อศึกษาการทำงานสัมพันธ์กันระหว่างไมโครคอนโทรเลอร์กับลิฟต์จำลอง

4.1.1.2 เพื่อศึกษาวิเคราะห์ข้อมูล อินพุต/เอาต์พุต ของอุปกรณ์ที่ใช้ในวงจร อุปกรณ์ที่ใช้ในการเรียกลิฟต์และบอกตำแหน่งของลิฟต์

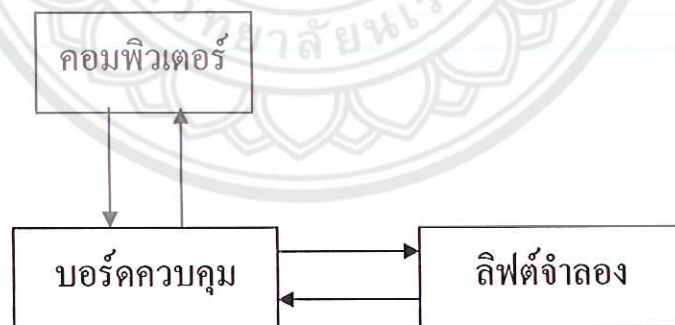
##### 4.1.2 ขั้นตอนการเชื่อมต่อบอร์ดควบคุม

4.1.2.1 ทำการบัดกรีสายสัญญาณต่างๆ เข้ากับบอร์ดควบคุม

4.1.2.2 ต่อบอร์ดควบคุมเข้ากับลิฟต์จำลอง

4.1.2.3 ตรวจสอบการเชื่อมต่อสายทั้งหมด

4.1.2.4 จ่ายไฟ 12V DC ให้กับบอร์ดควบคุม



รูปที่ 4.1 การต่อลิฟต์จำลองเพื่อทดสอบโปรแกรม

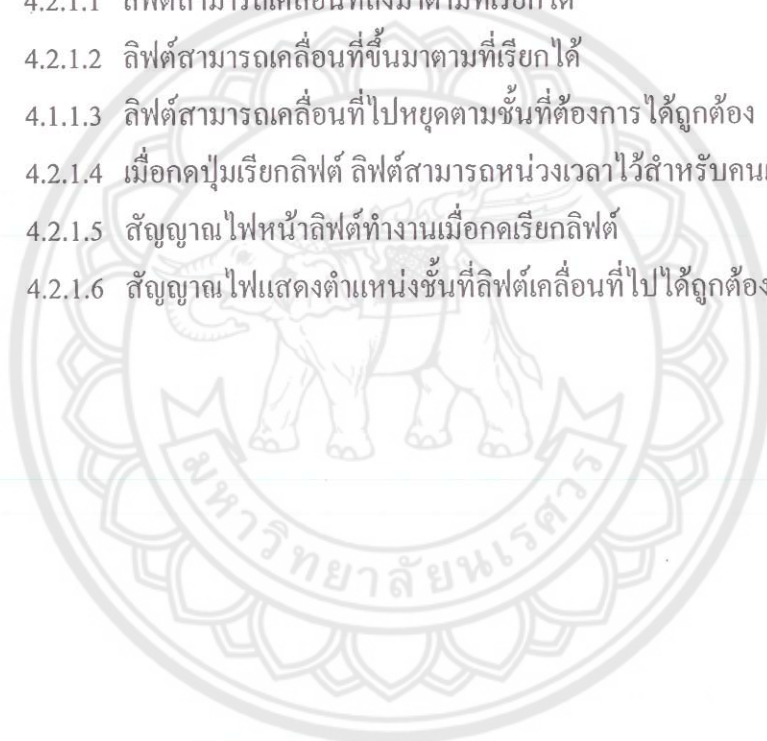
### 4.1.3 ทดสอบการควบคุมการทำงานของลิฟต์จำลอง

- 4.1.3.1 ทดสอบการเรียกลิฟต์ลง
- 4.1.3.2 ทดสอบการเรียกลิฟต์ขึ้น
- 4.1.3.3 ทดสอบการเลือกชั้นที่ต้องการ
- 4.1.3.4 ทดสอบการหน่วงเวลาการรอรับ
- 4.1.3.5 ทดสอบสัญญาณไฟเรียกชั้นของลิฟต์
- 4.1.3.6 ทดสอบสัญญาณไฟตำแหน่งของลิฟต์

## 4.2 ผลการทดลอง

### 4.2.1 ผลการทดสอบการควบคุมการทำงานของลิฟต์จำลอง

- 4.2.1.1 ลิฟต์สามารถเคลื่อนที่ลงมาตามที่เรียกได้
- 4.2.1.2 ลิฟต์สามารถเคลื่อนที่ขึ้นมาตามที่เรียกได้
- 4.1.1.3 ลิฟต์สามารถเคลื่อนที่ไปหยุดตามชั้นที่ต้องการได้ถูกต้อง
- 4.2.1.4 เมื่อกดปุ่มเรียกลิฟต์ ลิฟต์สามารถหน่วงเวลาไว้สำหรับคนเข้าออกลิฟต์ได้
- 4.2.1.5 สัญญาณไฟหน้าลิฟต์ทำงานเมื่อกดเรียกลิฟต์
- 4.2.1.6 สัญญาณไฟแสดงตำแหน่งชั้นที่ลิฟต์เคลื่อนที่ไปได้ถูกต้อง



## บทที่ 5

# สรุปผลของโครงการ

### 5.1 สรุปผลของโครงการ

จากการทดลองควบคุมระบบลิฟต์จำลองด้วยไมโครคอนโทรลเลอร์ ลิฟต์จำลองสามารถเคลื่อนที่ไปยังตำแหน่งที่ผู้ใช้เรียกได้อย่างถูกต้อง ไม่ว่าจะเป็นการกดเรียกที่หน้าลิฟต์หรือกดเลือกชั้นภายในลิฟต์ ลิฟต์จำลองนี้สามารถหน่วงเวลาเพื่อให้คนเข้าออกลิฟต์ได้ สัญญาณไฟต่างๆ ที่หน้าลิฟต์ บอร์ดควบคุมและไฟแสดงชั้นทำงานได้ถูกต้อง

จะเห็นว่าลิฟต์จำลองสามารถทำงานได้ตามความต้องการของผู้ใช้ใกล้เคียงกับลิฟต์ที่ใช้งานจริงๆ ไปในสถานที่ต่างๆ

### 5.2 ปัญหาที่เกิดขึ้นในระหว่างทำโครงการ

- 5.2.1 ปัญหาในการปรับปรุงแก้ไขโครงสร้างของลิฟต์จำลอง เนื่องจากทำด้วยเหล็ก
- 5.2.2 อุปกรณ์ทำงานผิดพลาดบ้างบางครั้ง เนื่องจากการติดตั้งไม่ตรงกับตำแหน่งที่ต้องการ
- 5.2.3 การเคลื่อนที่ของลิฟต์ไม่สม่ำเสมอ เนื่องจากรางเลื่อนของลิฟต์จำลองทำจากไม้จึงทำให้เกิดความฝืด
- 5.2.4 การตรวจสอบการทำงานของอุปกรณ์แต่ละชิ้นทำได้ยาก เนื่องจากมีการเดินสายไฟในท่อเหล็ก

### 5.3 แนวทางในการพัฒนา

- 5.3.1 พัฒนาโครงสร้างของลิฟต์จำลองเพื่อง่ายต่อการแก้ไข
- 5.3.2 พัฒนาโปรแกรมในด้านการควบคุมระบบลิฟต์จำลองด้วยไมโครคอนโทรลเลอร์ให้มีประสิทธิภาพมากขึ้น เพื่อเพิ่มความทันสมัย เช่น การสั่งงานปิดเปิดประตูลิฟต์
- 5.3.3 พัฒนาอุปกรณ์อินพุตเพื่อลดความผิดพลาดให้น้อยลง เช่น ใช้เซนเซอร์เป็นอินพุตบอกตำแหน่งของลิฟต์แทนลิมิตสวิตช์
- 5.3.4 พัฒนาอุปกรณ์แสดงผล เช่น การใช้ 7-Segment แสดงชั้นที่ลิฟต์เคลื่อนที่ไป

## เอกสารอ้างอิง

- [1] รองศาสตราจารย์ ชีรวัดณี ประกอบผล. ไมโครคอนโทรลเลอร์ด้วยภาษาC. กรุงเทพฯ : สำนักพิมพ์ ศ.ส.ท., 2545.
- [2] รองศาสตราจารย์ ชีรวัดณี ประกอบผล. การประยุกต์ใช้งานไมโครคอนโทรลเลอร์. กรุงเทพฯ : สำนักพิมพ์ ศ.ส.ท., 2547.
- [3] อุดม รานอก. ภาษาC สำหรับงานควบคุมไมโครคอนโทรลเลอร์ MCS-51. นนทบุรี : ไอดีซีฯ, 2548.
- [4] ประภาพร ช่างไม้. คู่มือเขียนโปรแกรมภาษาC, พิมพ์ครั้งที่1, 2545.
- [5] แผนกหนังสือพิเศษด้านอิเล็กทรอนิกส์. ทฤษฎีและการประยุกต์ใช้งานอุปกรณ์อิเล็กทรอนิกส์. กรุงเทพฯ : ซีเอ็ดยูเคชั่น, 2538.



## ภาคผนวก

### ก. ภาษาซีที่ใช้ควบคุมไมโครคอนโทรลเลอร์

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

sbit      PB1 = P1^7;
sbit      PB2 = P1^6;
sbit      PB3 = P1^5;
sbit      PB4 = P1^4;
sbit      PB5 = P1^3;
sbit      PB6 = P1^2;
sbit      PB7 = P1^1;
sbit      PB8 = P1^0;

sbit      LS1 = P3^7;
sbit      LS2 = P3^6;
sbit      LS3 = P3^5;
sbit      LS4 = P3^4;
sbit      LS5 = P3^3;

//sbit      LED1 = P2^0;
sbit      BTNclose = P2^2;
//sbit      LED2 = P2^1;

sbit      LVS1 = P2^7;
sbit      LVS2 = P2^6;
sbit      LVS3 = P2^5;
sbit      LVS4 = P2^4;
sbit      LVS5 = P2^3;
```

```

sbit      Busy   =P3^2;
sbit      LiftUP=P3^1;
sbit      LiftDN=P3^0;

```

```

sbit      L1 = P0^0;
sbit      L2 = P0^1;
sbit      L3 = P0^2;
sbit      L4 = P0^3;
sbit      L5 = P0^4;
sbit      MOTOR = P0^7;
sbit      POLAR = P0^6;

```

```

unsigned char temp1,temp2;
unsigned char LiftPos,OldLiftPos,CallPos;
unsigned char IsDirUP;
//-----
unsigned int h,i,j,k;
unsigned int a1,a2;
unsigned char LV[6][2]; // LV[Floor][Up=0,Down=1]
unsigned char LVS[6];
unsigned char LVmin,LVmax,CLVmax,CLVmin;

```

```

void DirUP(void)

```

```

{
    LiftUP=0;
    LiftDN=1;
    POLAR=0;
    IsDirUP=1;
}

```

```

void DirDN(void)

```

```

{

```



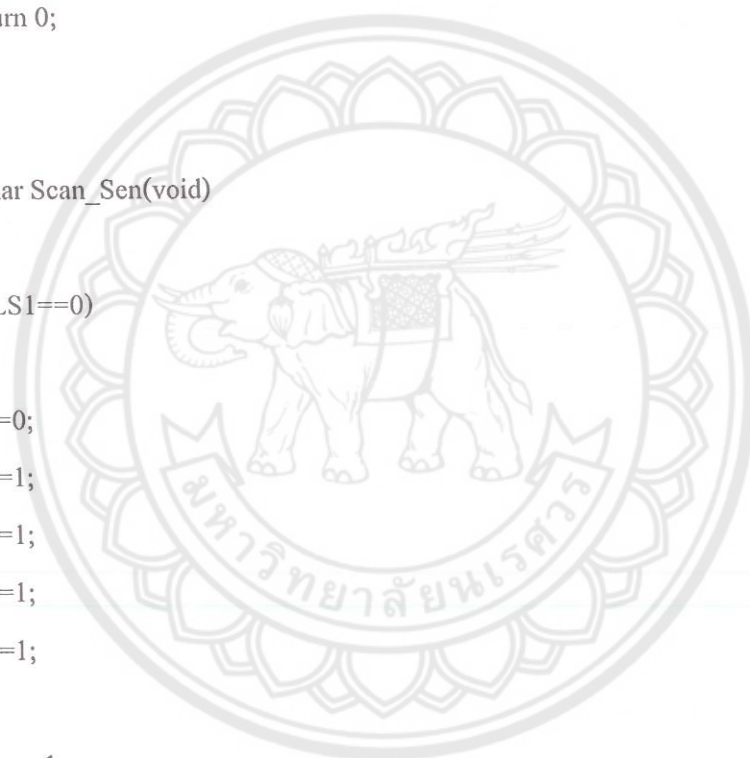
```
LiftUP=1;
LiftDN=0;
POLAR=1;
IsDirUP=0;
}
```

```
unsigned char IsDirDN(void)
{
    if(IsDirUP==0)return 1;
    return 0;
}
```

```
unsigned char Scan_Sen(void)
```

```
{
    if(LS1==0)
    {
        L1=0;
        L2=1;
        L3=1;
        L4=1;
        L5=1;

        return 1;
    }
    if(LS2==0)
    {
        L1=1;
        L2=0;
        L3=1;
        L4=1;
        L5=1;
    }
}
```



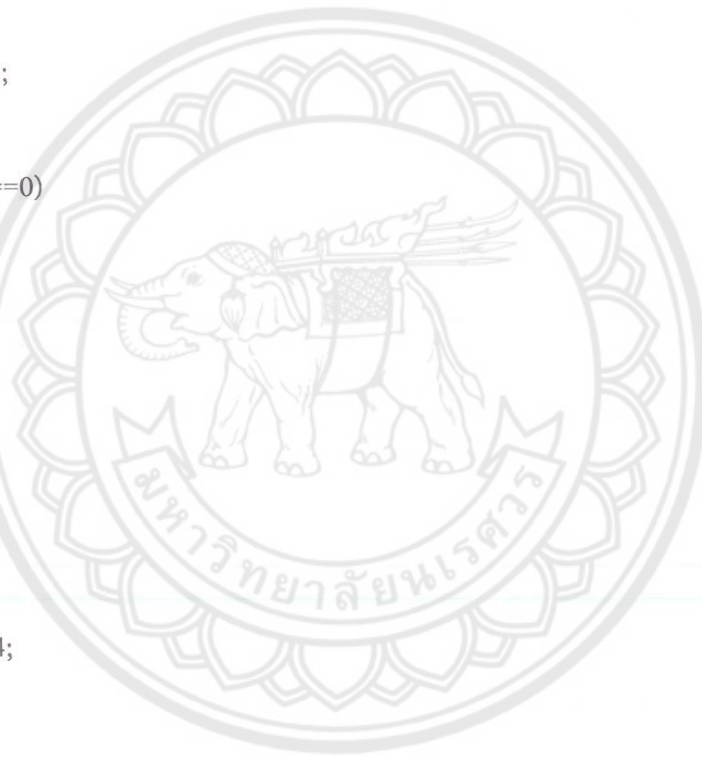
```
return 2;  
}  
if(LS3==0)  
{  
L1=1;  
L2=1;  
L3=0;  
L4=1;  
L5=1;
```

```
return 3;  
}  
if(LS4==0)  
{  
L1=1;  
L2=1;  
L3=1;  
L4=0;  
L5=1;
```

```
return 4;  
}
```

```
if(LS5==0)  
{  
L1=1;  
L2=1;  
L3=1;  
L4=1;  
L5=0;
```

```
return 5;
```



```
    }  
    return 0;  
}
```

```
unsigned char ReadPosition(void)  
{  
    if(L1==0)return 1;  
    if(L2==0)return 2;  
    if(L3==0)return 3;  
    if(L4==0)return 4;  
    if(L5==0)return 5;  
    return 0;  
}
```

```
unsigned char ReadLvSelect(void)  
{  
    if(LVS1==0)return 1;  
    if(LVS2==0)return 2;  
    if(LVS3==0)return 3;  
    if(LVS4==0)return 4;  
    if(LVS5==0)return 5;  
    return 0;  
}
```

```
void Latch_Switch(void) // Latch switch and save data to array  
{
```

```
    PB1=PB1;
```

```
    PB2=PB2;
```

```
    PB3=PB3;
```

```
    PB4=PB4;
```

```
PB5=PB5;
```

```
PB6=PB6;
```

```
PB7=PB7;
```

```
PB8=PB8;
```

```
LVS1=LVS1;
```

```
LVS2=LVS2;
```

```
LVS3=LVS3;
```

```
LVS4=LVS4;
```

```
LVS5=LVS5;
```

```
/*
```

```
if(LiftPos==1)LVS1=1;
```

```
if(LiftPos==2)LVS2=1;
```

```
if(LiftPos==3)LVS3=1;
```

```
if(LiftPos==4)LVS4=1;
```

```
if(LiftPos==5)LVS5=1;
```

```
*/
```

```
}
```

```
unsigned char Call_Switch(void)
```

```
{
```

```
    Latch_Switch();
```

```
    if(PB1==0)return 5;
```

```
    if(PB2==0)return 4;
```

```
    if(PB3==0)return 4;
```

```
    if(PB4==0)return 3;
```

```
    if(PB5==0)return 3;
```

```
    if(PB6==0)return 2;
```

```
    if(PB7==0)return 2;
```

```
    if(PB8==0)return 1;
```

```
    return 0;
```

}

```
void LoadSwToArr(void)
```

```
{
```

```
    LV[1][0]=PB8; //UP
```

```
    LV[1][1]=1;
```

```
    LV[2][0]=PB6; //UP
```

```
    LV[2][1]=PB7; //DN
```

```
    LV[3][0]=PB4; //UP
```

```
    LV[3][1]=PB5; //DN
```

```
    LV[4][0]=PB2; //UP
```

```
    LV[4][1]=PB3; //DN
```

```
    LV[5][0]=1;
```

```
    LV[5][1]=PB1; //DN
```

```
    LVS[1]=LVS1;
```

```
    LVS[2]=LVS2;
```

```
    LVS[3]=LVS3;
```

```
    LVS[4]=LVS4;
```

```
    LVS[5]=LVS5;
```

```
}
```

```
void LoadArrToSw(void)
```

```
{
```

```
    PB8=LV[1][0];
```

```
    PB6=LV[2][0];
```

```
PB7=LV[2][1];
```

```
PB4=LV[3][0];
```

```
PB5=LV[3][1];
```

```
PB2=LV[4][0];
```

```
PB3=LV[4][1];
```

```
PB1=LV[5][1];
```

```
LVS1=LVS[1];
```

```
LVS2=LVS[2];
```

```
LVS3=LVS[3];
```

```
LVS4=LVS[4];
```

```
LVS5=LVS[5];
```

```
}
```

```
void ScanSwitch(void)
```

```
{
```

```
    Scan_Sen();
```

```
    Latch_Switch();
```

```
    LoadSwToArr();
```

```
}
```

```
void delay_10ms(unsigned char d10ms)
```

```
{
```

```
    while(d10ms--)
```

```
        for(h=0;h<=700;h++); //time delay 10ms
```

```
}
```

```
void delay_50ms(unsigned char d50ms)
```

```
{
    while(d50ms--)
    {
        ScanSwitch();/*****
        for(i=0;i<=3550;i++); //time delay 50ms
    }
}

void delay_100ms(unsigned char d100ms)
{
    while(d100ms--)
        delay_50ms(2);
}

void delay_1s(unsigned char d1s)
{
    while(d1s--)
        delay_100ms(10);
}

unsigned char hh,cc;
unsigned int a0,a1;
unsigned char ChkForStop(void)
unsigned char AI(void)
{
    if(IsDirDN()==1)
    {
        if(LVS1==0)return 1;
        if(LVS2==0)return 2;
        if(LVS3==0)return 3;
        if(LVS4==0)return 4;
        if(LVS5==0)return 5;
```

```
return 0;
}

    if(LVS5==0)return 5;
    if(LVS4==0)return 4;
    if(LVS3==0)return 3;
    if(LVS2==0)return 2;
    if(LVS1==0)return 1;
return 0;
}

unsigned char CallAI(void)
{
    if(IsDirDN()==0)
    {
        if(PB1==0)return 5;
        if(PB2==0)return 4;
        if(PB4==0)return 3;
        if(PB6==0)return 2;
        if(PB8==0)return 1;
    }
    if(IsDirDN()==1)
    {
        if(PB8==0)return 1;
        if(PB7==0)return 2;
        if(PB5==0)return 3;
        if(PB3==0)return 4;
        if(PB1==0)return 5;
    }

    if(IsDirDN()==1)
    {
```



```
        if(PB1==0)return 5;
        if(PB2==0)return 4;
        if(PB4==0)return 3;
        if(PB6==0)return 2;
        if(PB8==0)return 1;
    }
    if(IsDirDN()==0)
    {
        if(PB8==0)return 1;
        if(PB7==0)return 2;
        if(PB5==0)return 3;
        if(PB3==0)return 4;
        if(PB1==0)return 5;
    }
    return 0;
}

void MotorStop(void)
{
    MOTOR=0;
    delay_1s(1);
}

void MotorStart(void)
{
    unsigned int MScount;

    ScanSwitch();
    temp1=Scan_Sen();
    if(IsDirDN()==1)temp2=1;
    if(IsDirDN()==0)temp2=0;
```

```
if(LV[LiftPos][temp2]==0||LVS[LiftPos]==0)
{
//      LED1=0;
//      LV[LiftPos][temp2]=1;
//      LVS[LiftPos]=1;
//      LoadArrToSw();
      return;
}
```

```
for(MScount=0;MScount<=200;MScount++)
```

```
{
if(BTNclose==0)MScount=200;
delay_50ms(1);
}
```

```
for(a1=5;a1<=500;a1+=2)
```

```
{
MOTOR=0;
for (a0=0;a0<=500-a1;a0++);
if(ChkForStop()==1)
```

```
{
MotorStop();
return;
}
```

```
MOTOR=1;
```

```
for (a0=0;a0<=a1;a0+=2);
```

```
}
```

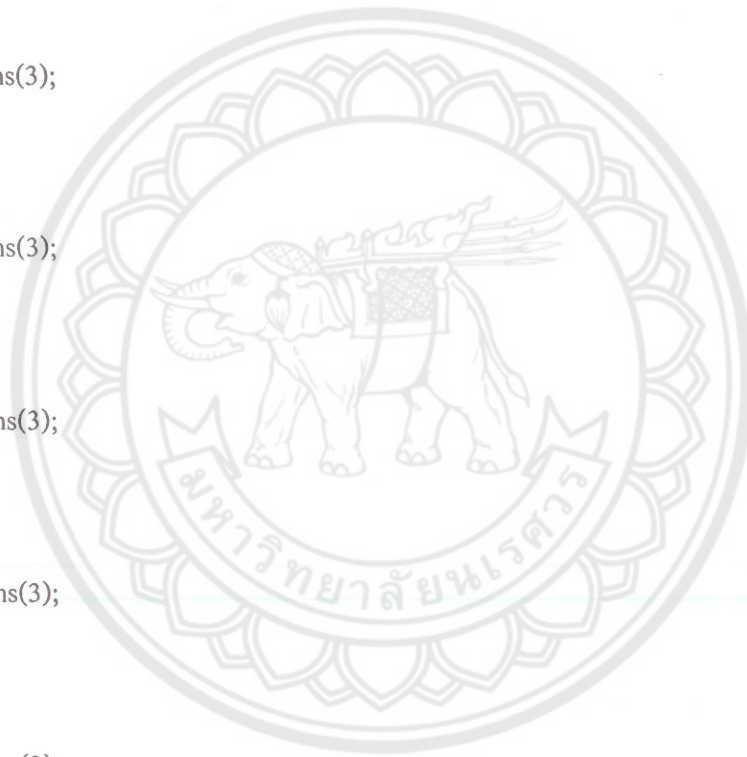
```
if(ChkForStop()==1)
```

```
{
MotorStop();
return;
```

```
    }  
    MOTOR=1;  
}  
void init(void)  
{  
    P1=0xff;  
    P2=0xff;  
    P3=0xff;  
    MOTOR=0;  
    POLAR=0;  
  
    LV[0][0]=1;  
    LV[0][1]=1;  
    LVS[0]=1;  
  
    delay_1s(1);  
    DirUP();  
    delay_1s(1);  
    DirDN();  
    delay_1s(1);  
    DirUP();  
    delay_1s(1);  
    DirDN();  
    delay_1s(1);  
    DirUP();  
  
    delay_1s(2);  
  
    P1=0x00;  
    P2=0xff;  
    delay_1s(2);  
    P1=0xff;
```

```
P2=0xff;
delay_1s(2);
P1=0xff;
P2=0x00;
delay_1s(2);
P1=0xff;
P2=0xff;
```

```
delay_1s(2);
L1=0;
delay_100ms(3);
L1=1;
L2=0;
delay_100ms(3);
L2=1;
L3=0;
delay_100ms(3);
L3=1;
L4=0;
delay_100ms(3);
L4=1;
L5=0;
delay_100ms(3);
L5=1;
L4=0;
delay_100ms(3);
L4=1;
L3=0;
delay_100ms(3);
L3=1;
L2=0;
delay_100ms(3);
```



```
L2=1;
L1=0;
delay_100ms(3);
L1=1;
}
```

```
void GotoLevel(unsigned char GTL)
```

```
{
    ScanSwitch();
    LiftPos=ReadPosition();
    OldLiftPos=LiftPos;
    if(GTL==0)return;

    if(GTL==LiftPos)
    {
        LV[LiftPos][0]=1;
        LV[LiftPos][1]=1;
        LVS[LiftPos]=1;
        if(LiftPos==5)PB1=1;
        if(LiftPos==1)PB8=1;
        LoadArrToSw();
        return;
    }
}
```

```
ScanSwitch();
LiftPos=ReadPosition();
OldLiftPos=LiftPos;
if(LiftPos==GTL)return;
```

```
delay_1s(1);
GTL=AI();
if(GTL==0)GTL=CallAI();
```

```

if(GTL>OldLiftPos)DirUP();
if(GTL<OldLiftPos)DirDN();

```

```

    delay_1s(1);

```

```

        ScanSwitch();

```

```

temp1=Scan_Sen();

```

```

if(IsDirDN()==1)temp2=1;

```

```

if(IsDirDN()==0)temp2=0;

```

```

if(LV[LiftPos][temp2]==1&&LVS[LiftPos]==1)MotorStart();

```

```

//*****

```

```

GTL=AI();

```

```

if(GTL==0)GTL=CallAI();

```

```

while(LiftPos!=GTL)

```

```

{

```

```

    Scan_Sen();

```

```

    ScanSwitch();

```

```

    LiftPos=ReadPosition();

```

```

    ScanSwitch();

```

```

temp1=Scan_Sen();

```

```

if(IsDirDN()==1)temp2=1;

```

```

if(IsDirDN()==0)temp2=0;

```

```

    ScanSwitch();

```

```

if(LV[LiftPos][temp2]==0||LVS[LiftPos]==0)

```

```

{

```

```

    MotorStop();

```

```

    break;

```

```
    }
    LoadSwToArr();
    if(LVS[LiftPos]==0)
    {
    MotorStop();
    break;
    }

    if(LV[LiftPos][temp2]==0)
    {
    MotorStop();
    break;
    }
    //-----
    }
MotorStop();

if(LV[LiftPos][0]==1&&LV[LiftPos][1]==0)LV[LiftPos][1]=1;
if(LV[LiftPos][1]==1&&LV[LiftPos][0]==0)LV[LiftPos][0]=1;
if(LV[LiftPos][0]==0&&LV[LiftPos][1]==0)LV[LiftPos][temp2]=1;

if(LiftPos==5)PB1=1;
if(LiftPos==1)PB8=1;

LV[LiftPos][temp2]=1;
LVS[LiftPos]=1;

LoadArrToSw();
if(LiftPos==GTL)
{

Latch_Switch();
```

```
    }  
}  
//***** Visual Session *****  
//***** End Visual Session *****  
/*  
void InitPosition(void)  
{  
    LiftPos=ReadPosition();  
}  
*/  
  
void main()  
{  
    init();  
  
    while(1)  
    {  
        ScanSwitch();  
        temp1=ReadLvSelect();  
        // temp2=Call_Switch();  
  
        if(AI()!=0)  
        {  
            GotoLevel(AI());  
        }  
  
        if(AI()==0)  
        {  
            GotoLevel(CallAI());  
        }  
    }  
}
```



ข. รายละเอียดของอุปกรณ์อิเล็กทรอนิกส์

<http://www.datasheetcatalog.com>

**Features**

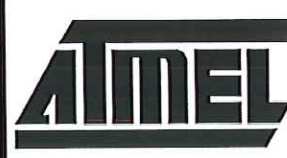
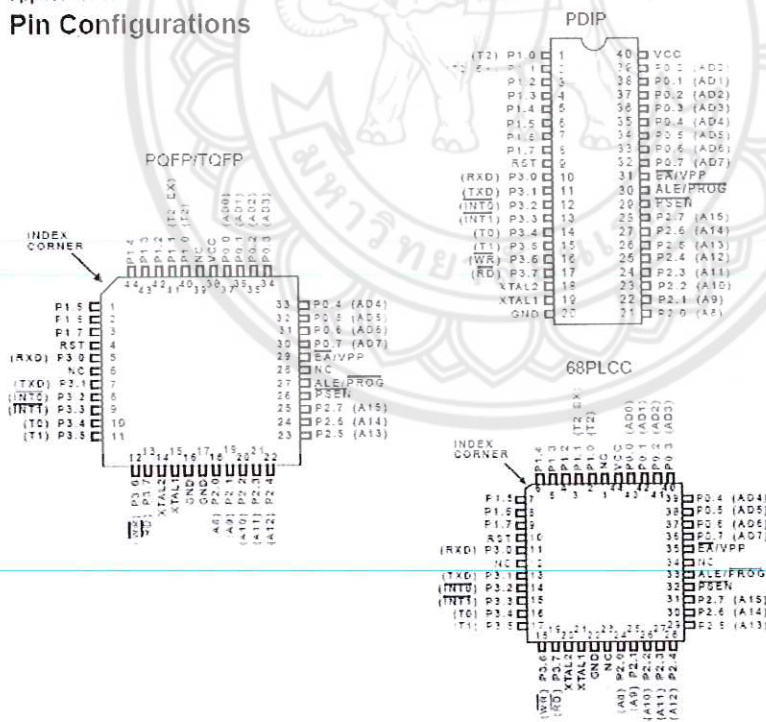
- Compatible with MCS-51™ Products
- 8K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 256 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-Bit Timer/Counters
- Eight Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

**Description**

The AT89C52 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard 80C51 and 80C52 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C52 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

(continued)

**Pin Configurations**



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

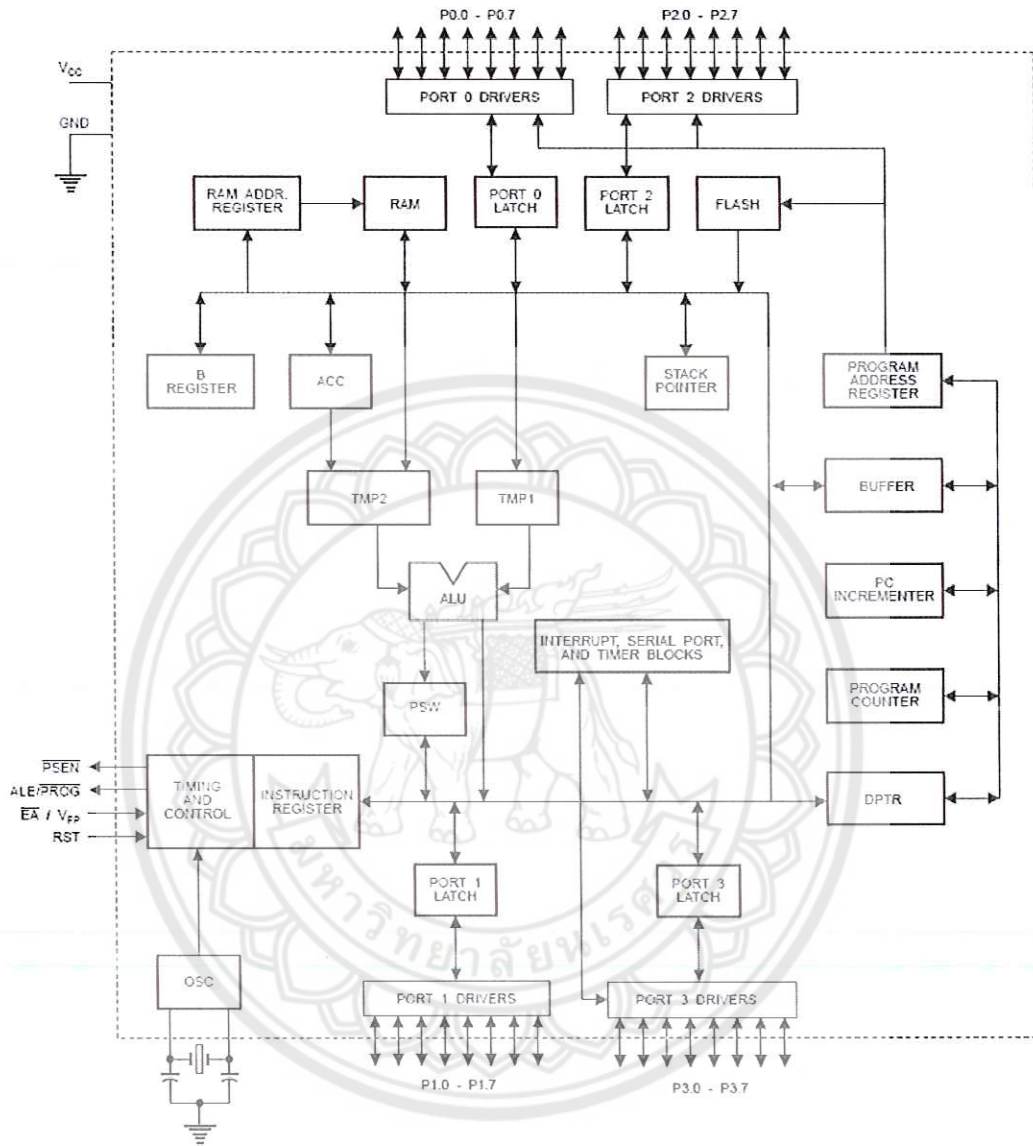
**AT89C52**

0313F-A-12/97

รูปที่ ๓1 แสดงตำแหน่งพอร์ทของ AT89C52

<http://www.datasheetcatalog.com>

### Block Diagram



รูปที่ ๗2 Block Diagram แสดงหน้าที่ของพอร์ท AT89C52

The AT89C52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89C52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next hardware reset.

## Pin Description

**V<sub>CC</sub>**  
Supply voltage.

**GND**  
Ground.

### Port 0

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

### Port 1

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

In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table.

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

| Port Pin | Alternate Functions   |
|----------|---|
| P1.0     | T2 (external count input to Timer/Counter 2), clock-out             |
| P1.1     | T2EX (Timer/Counter 2 capture/reload trigger and direction control) |

### $\overline{EA}/V_{PP}$

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

$\overline{EA}$  should be strapped to V<sub>CC</sub> for internal program executions.

This pin also receives the 12-volt programming enable voltage (V<sub>PP</sub>) during Flash programming when 12-volt programming is selected.

### XTAL1

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

### XTAL2

Output from the inverting oscillator amplifier.

### Port 2

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

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

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

### Port 3

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

Port 3 also serves the functions of various special features of the AT89C51, as shown in the following table.

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

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

### RST

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

### ALE/PROG

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

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

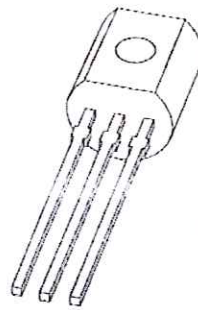
pulse is skipped during each access to external data memory.

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

### PSEN

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

When the AT89C52 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.



**2PC1815**  
NPN general purpose transistor

## NPN general purpose transistor

**2PC1815**

### FEATURES

- Low current (max. 150 mA)
- Low voltage (max. 50 V).

### APPLICATIONS

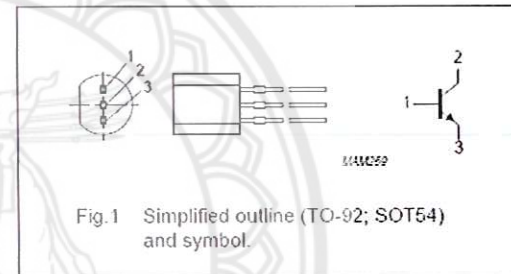
- General purpose switching and amplification, e.g. audio amplifier driver stages.

### DESCRIPTION

NPN transistor in a TO-92 (SOT54) plastic package.  
PNP complement: 2PA1015.

### PINNING

| PIN | DESCRIPTION |
|-----|-------------|
| 1   | base        |
| 2   | collector   |
| 3   | emitter     |



### LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

| SYMBOL    | PARAMETER                     | CONDITIONS                               | MIN. | MAX. | UNIT             |
|-----------|-------------------------------|--|------|------|------------------|
| $V_{CB0}$ | collector-base voltage        | open emitter                             | -    | 60   | V                |
| $V_{CE0}$ | collector-emitter voltage     | open base                                | -    | 50   | V                |
| $V_{EB0}$ | emitter-base voltage          | open collector                           | -    | 5    | V                |
| $I_C$     | collector current (DC)        |  | -    | 150  | mA               |
| $I_{CV}$  | peak collector current        |  | -    | 200  | mA               |
| $I_{BM}$  | peak base current             |  | -    | 200  | mA               |
| $P_{tot}$ | total power dissipation       | $T_{amb} \leq 25^\circ\text{C}$ ; note 1 | -    | 500  | mW               |
| $T_{stg}$ | storage temperature           |  | -65  | +150 | $^\circ\text{C}$ |
| $T_J$     | junction temperature          |  | -    | 150  | $^\circ\text{C}$ |
| $T_{amb}$ | operating ambient temperature |  | -65  | +150 | $^\circ\text{C}$ |

### Note

- 1 Transistor mounted on an FR4 printed-circuit board.

รูปที่ ๗4 แสดงตำแหน่งของขา 2PC1815

## ตารางที่ ๗1 แสดงคุณสมบัติเฉพาะของทรานซิสเตอร์

NPN general purpose transistor

2PC1815

## THERMAL CHARACTERISTICS

| SYMBOL        | PARAMETER                                   | CONDITIONS | VALUE | UNIT |
|---------------|---|------------|-------|------|
| $R_{th(j-a)}$ | thermal resistance from junction to ambient | note 1     | 250   | K/W  |

## Note

1. Transistor mounted on an FR4 printed-circuit board.

## CHARACTERISTICS

 $T_j = 25\text{ }^{\circ}\text{C}$  unless otherwise specified.

| SYMBOL      | PARAMETER                            | CONDITIONS   | MIN. | TYP. | MAX. | UNIT |
|-------------|--------------------------------------|--|------|------|------|------|
| $I_{CBO}$   | collector cut-off current            | $I_E = 0; V_{CB} = 60\text{ V}$  | –    | –    | 100  | nA   |
| $I_{EBO}$   | emitter cut-off current              | $I_C = 0; V_{EB} = 5\text{ V}$   | –    | –    | 100  | nA   |
| $h_{FE}$    | DC current gain                      | $I_C = 150\text{ mA}; V_{CE} = 6\text{ V}$   | 25   | –    | –    |      |
| $h_{FE}$    | DC current gain                      | $I_C = 2\text{ mA}; V_{CE} = 6\text{ V}$   |      |      |      |      |
|             | 2PC1815                              |  | 120  | –    | 700  |      |
|             | 2PC1815Y                             |  | 120  | –    | 240  |      |
|             | 2PC1815GR                            |  | 200  | –    | 400  |      |
|             | 2PC1815BL                            |  | 350  | –    | 700  |      |
| $V_{CEsat}$ | collector-emitter saturation voltage | $I_C = 100\text{ mA}; I_B = 10\text{ mA}$  | –    | –    | 300  | mV   |
| $V_{BEsat}$ | base-emitter saturation voltage      | $I_C = 100\text{ mA}; I_B = 10\text{ mA}$  | –    | –    | 1.1  | V    |
| $C_c$       | collector capacitance                | $I_E = I_C = 0; V_{CB} = 10\text{ V}; f = 1\text{ MHz}$  | –    | 2.5  | 3.5  | pF   |
| $f_T$       | transition frequency                 | $I_C = 1\text{ mA}; V_{CE} = 6\text{ V}; f = 100\text{ MHz}$                                   | 80   | –    | –    | MHz  |
| F           | noise figure                         | $I_C = 200\text{ }\mu\text{A}; V_{CE} = 6\text{ V}; R_S = 2\text{ k }\Omega; f = 1\text{ kHz}$ | –    | –    | 10   | dB   |

<http://www.datasheetcatalog.com>

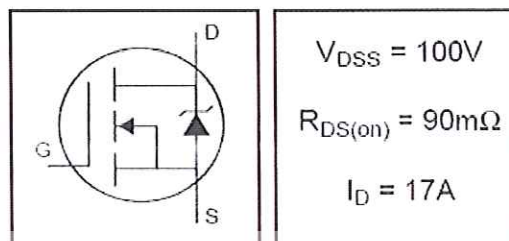
International  
**IR** Rectifier

PD - 91351

**IRF530N**

HEXFET® Power MOSFET

- Advanced Process Technology
- Ultra Low On-Resistance
- Dynamic dv/dt Rating
- 175°C Operating Temperature
- Fast Switching
- Fully Avalanche Rated



### Description

Advanced HEXFET® Power MOSFETs from International Rectifier utilize advanced processing techniques to achieve extremely low on-resistance per silicon area. This benefit, combined with the fast switching speed and ruggedized device design that HEXFET power MOSFETs are well known for, provides the designer with an extremely efficient and reliable device for use in a wide variety of applications.

The TO-220 package is universally preferred for all commercial-industrial applications at power dissipation levels to approximately 50 watts. The low thermal resistance and low package cost of the TO-220 contribute to its wide acceptance throughout the industry.



### Absolute Maximum Ratings

|                           | Parameter                                | Max.                  | Units |
|---------------------------|--|-----------------------|-------|
| $I_D @ T_C = 25^\circ C$  | Continuous Drain Current, $V_{GS} @ 10V$ | 17                    | A     |
| $I_D @ T_C = 100^\circ C$ | Continuous Drain Current, $V_{GS} @ 10V$ | 12                    |       |
| $I_{DM}$                  | Pulsed Drain Current ①                   | 60                    |       |
| $P_D @ T_C = 25^\circ C$  | Power Dissipation                        | 70                    | W     |
|                           | Linear Derating Factor                   | 0.47                  | W/°C  |
| $V_{GS}$                  | Gate-to-Source Voltage                   | $\pm 20$              | V     |
| $I_{AR}$                  | Avalanche Current ②                      | 9.0                   | A     |
| $E_{AR}$                  | Repetitive Avalanche Energy ②            | 7.0                   | mJ    |
| dv/dt                     | Peak Diode Recovery dv/dt ③              | 7.4                   | V/ns  |
| $T_J$                     | Operating Junction and                   | -55 to +175           | °C    |
| $T_{STG}$                 | Storage Temperature Range                |                       |       |
|                           | Soldering Temperature, for 10 seconds    | 300 (1.6mm from case) |       |
|                           | Mounting torque, 6-32 or M3 screw        | 10 lbf-in (1.1N-m)    |       |

### Thermal Resistance

|                 | Parameter                           | Typ. | Max. | Units |
|-----------------|-------------------------------------|------|------|-------|
| $R_{\theta JC}$ | Junction-to-Case                    | —    | 2.15 | °C/W  |
| $R_{\theta CS}$ | Case-to-Sink, Flat, Greased Surface | 0.50 | —    |       |
| $R_{\theta JA}$ | Junction-to-Ambient                 | —    | 62   |       |

รูปที่ ๗5 แสดง โครงสร้างภายในของ IRF530N

http://www.datasheetcatalog.com

ตารางที่ ๒ แสดงคุณสมบัติทางไฟฟ้าของ IRF530N

# IRF530N

International  
IR Rectifier

## Electrical Characteristics @ $T_J = 25^\circ\text{C}$ (unless otherwise specified)

|                                 | Parameter                            | Min. | Typ.  | Max. | Units      | Conditions   |
|---------------------------------|--------------------------------------|------|-------|------|------------|--|
| $V_{(BR)DSS}$                   | Drain-to-Source Breakdown Voltage    | 100  | ---   | ---  | V          | $V_{GS} = 0V, I_D = 250\mu A$                                      |
| $\Delta V_{(BR)DSS}/\Delta T_J$ | Breakdown Voltage Temp. Coefficient  | ---  | 0.11  | ---  | V/°C       | Reference to $25^\circ\text{C}, I_D = 1\text{mA}$                  |
| $R_{DS(on)}$                    | Static Drain-to-Source On-Resistance | ---  | ---   | 90   | m $\Omega$ | $V_{GS} = 10V, I_D = 9.0A$ ①                                       |
| $V_{GS(th)}$                    | Gate Threshold Voltage               | 2.0  | ---   | 4.0  | V          | $V_{DS} = V_{GS}, I_D = 250\mu A$                                  |
| $g_{fs}$                        | Forward Transconductance             | 12   | ---   | ---  | S          | $V_{DS} = 50V, I_D = 9.0A$ ②                                       |
| $I_{DSS}$                       | Drain-to-Source Leakage Current      | ---  | ---   | 25   | $\mu A$    | $V_{DS} = 100V, V_{GS} = 0V$                                       |
|                                 |                                      | ---  | ---   | 250  | $\mu A$    | $V_{DS} = 80V, V_{GS} = 0V, T_J = 150^\circ\text{C}$               |
| $I_{GSS}$                       | Gate-to-Source Forward Leakage       | ---  | ---   | 100  | nA         | $V_{DS} = 20V$   |
|                                 | Gate-to-Source Reverse Leakage       | ---  | ---   | -100 | nA         | $V_{GS} = -20V$  |
| $Q_g$                           | Total Gate Charge                    | ---  | ---   | 37   | nC         | $I_D = 9.0A$   |
| $Q_{gs}$                        | Gate-to-Source Charge                | ---  | ---   | 7.2  | nC         | $V_{DS} = 80V$   |
| $Q_{gd}$                        | Gate-to-Drain ("Miller") Charge      | ---  | ---   | 11   | nC         | $V_{GS} = 10V$ , See Fig. 6 and 13                                 |
| $t_{d(on)}$                     | Turn-On Delay Time                   | ---  | 9.2   | ---  | ns         | $V_{DD} = 50V$   |
| $t_r$                           | Rise Time                            | ---  | 22    | ---  | ns         | $I_D = 9.0A$   |
| $t_{d(off)}$                    | Turn-Off Delay Time                  | ---  | 35    | ---  | ns         | $R_G = 12\Omega$   |
| $t_f$                           | Fall Time                            | ---  | 25    | ---  | ns         | $V_{GS} = 10V$ , See Fig. 10 ③                                     |
| $L_D$                           | Internal Drain Inductance            | ---  | 4.5   | ---  | nH         | Between lead, 6mm (0.25in.) from package and center of die contact |
| $L_S$                           | Internal Source Inductance           | ---  | 7.5   | ---  | nH         |  |
| $C_{iss}$                       | Input Capacitance                    | ---  | 920   | ---  | pF         | $V_{DS} = 0V$  |
| $C_{oss}$                       | Output Capacitance                   | ---  | 130   | ---  | pF         | $V_{DS} = 25V$   |
| $C_{rss}$                       | Reverse Transfer Capacitance         | ---  | 19    | ---  | pF         | $f = 1.0\text{MHz}$ , See Fig. 5                                   |
| $E_{AS}$                        | Single Pulse Avalanche Energy ④      | ---  | 340 ⑤ | 93 ⑥ | mJ         | $I_{AS} = 9.0A, L = 2.3\text{mH}$                                  |

## Source-Drain Ratings and Characteristics

|          | Parameter                              | Min. | Typ. | Max. | Units | Conditions  |
|----------|--|------|------|------|-------|---|
| $I_S$    | Continuous Source Current (Body Diode) | ---  | ---  | 17   | A     | MOSFET symbol showing the integral reverse p-n junction diode.            |
| $I_{SV}$ | Pulsed Source Current (Body Diode) ①   | ---  | ---  | 60   | A     |   |
| $V_{SD}$ | Diode Forward Voltage                  | ---  | ---  | 1.3  | V     | $T_J = 25^\circ\text{C}, I_S = 9.0A, V_{GS} = 0V$ ②                       |
| $t_{rr}$ | Reverse Recovery Time                  | ---  | 93   | 140  | ns    | $T_J = 25^\circ\text{C}, I_F = 9.0A$                                      |
| $Q_{rr}$ | Reverse Recovery Charge                | ---  | 320  | 480  | nC    | $di/dt = 100A/\mu s$ ③  |
| $t_{on}$ | Forward Turn-On Time                   | ---  | ---  | ---  | ns    | Intrinsic turn-on time is negligible (turn-on is dominated by $L_S+L_D$ ) |

### Notes:

- ① Repetitive rating; pulse width limited by max. junction temperature. (See fig. 11)
- ② Starting  $T_J = 25^\circ\text{C}, L = 2.3\text{mH}$   
 $R_G = 25\Omega, I_{AS} = 9.0A, V_{GS} = 10V$  (See Figure 12)
- ③  $I_{SD} \leq 9.0A, di/dt \leq 410A/\mu s, V_{DD} \leq V_{(BR)DSS}, T_J \leq 175^\circ\text{C}$

- ④ Pulse width  $\leq 400\mu s$ ; duty cycle  $\leq 2\%$ .
- ⑤ This is a typical value at device destruction and represents operation outside rated limits.
- ⑥ This is a calculated value limited to  $T_J = 175^\circ\text{C}$ .

<http://www.datasheetcatalog.com>



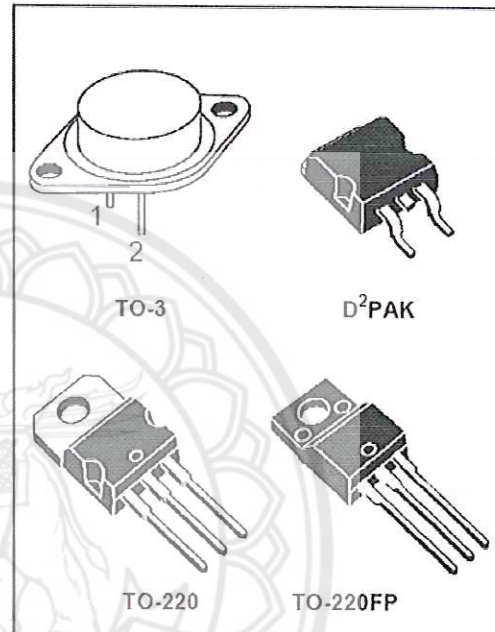
## L7800 SERIES

### POSITIVE VOLTAGE REGULATORS

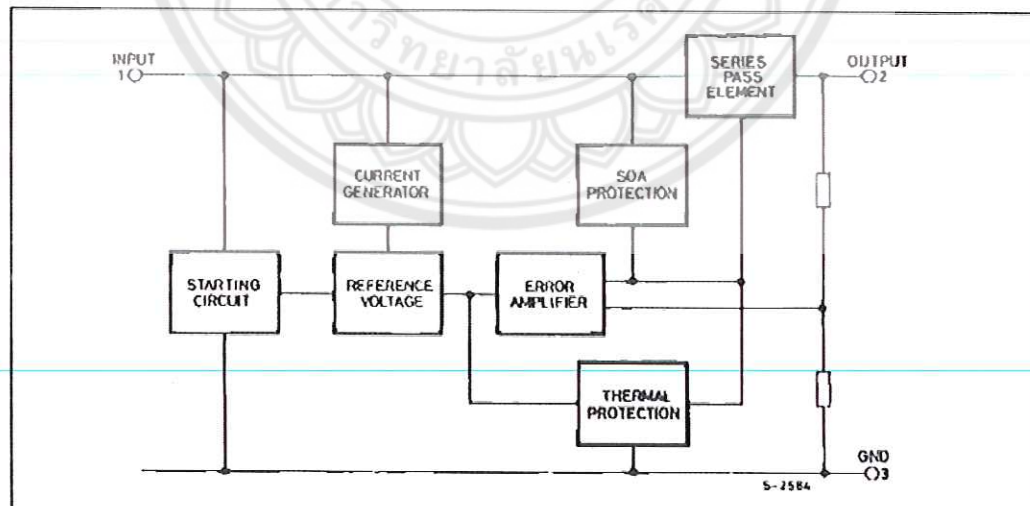
- OUTPUT CURRENT UP TO 1.5 A
- OUTPUT VOLTAGES OF 5; 5.2; 6; 8; 8.5; 9; 12; 15; 18; 24V
- THERMAL OVERLOAD PROTECTION
- SHORT CIRCUIT PROTECTION
- OUTPUT TRANSITION SOA PROTECTION

#### DESCRIPTION

The L7800 series of three-terminal positive regulators is available in TO-220 TO-220FP TO-3 and D<sup>2</sup>PAK packages and several fixed output voltages, making it useful in a wide range of applications. These regulators can provide local on-card regulation, eliminating the distribution problems associated with single point regulation. Each type employs internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible. If adequate heat sinking is provided they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.



#### BLOCK DIAGRAM



รูปที่ ๗6 Block Diagram แสดงการทำงานภายในของ L7800



### ตารางที่ ๗3 แสดง Output Voltage Regulator แต่ละเบอร์

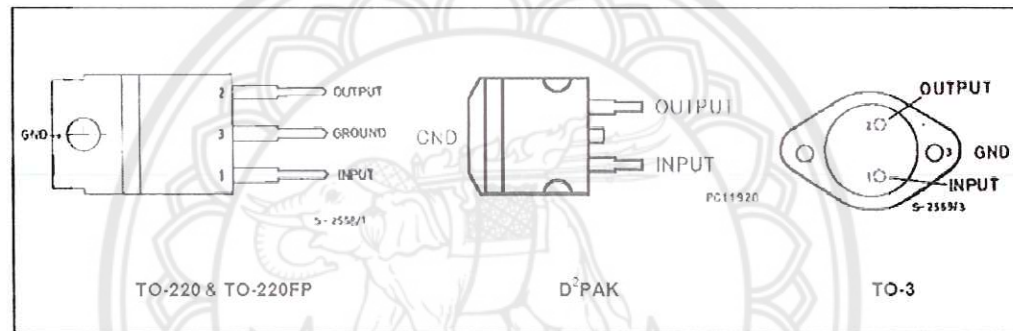
#### ABSOLUTE MAXIMUM RATINGS

| Symbol           | Parameter   | Value              | Unit |
|------------------|---|--------------------|------|
| V <sub>i</sub>   | DC Input Voltage (for V <sub>o</sub> = 5 to 18V)<br>(for V <sub>o</sub> = 20-24V) | 35                 | V    |
|                  |   | 40                 | V    |
| I <sub>o</sub>   | Output Current  | Internally limited |      |
| P <sub>tot</sub> | Power Dissipation   | Internally limited |      |
| T <sub>op</sub>  | Operating Junction Temperature Range (for L7800)<br>(for L7800C)                  | -55 to 150         | °C   |
|                  |   | 0 to 150           | °C   |
| T <sub>stg</sub> | Storage Temperature Range   | -65 to 150         | °C   |

#### THERMAL DATA

| Symbol                | Parameter                               | D <sup>2</sup> PAK | TO-220 | TO-220FP | TO-3 | Unit |
|-----------------------|---|--------------------|--------|----------|------|------|
| R <sub>thj-case</sub> | Thermal Resistance Junction-case Max    | 3                  | 3      | 5        | 4    | °C/W |
| R <sub>thj-amb</sub>  | Thermal Resistance Junction-ambient Max | 62.5               | 50     | 60       | 35   | °C/W |

#### CONNECTION DIAGRAM AND ORDERING NUMBERS (top view)

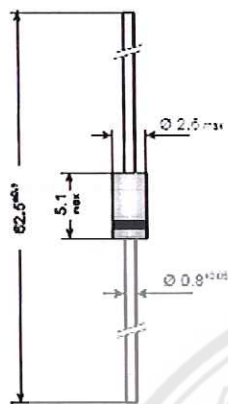


| Type   | TO-220  | D <sup>2</sup> PAK (*) | TO-220FP | TO-3    | Output Voltage |
|--------|---------|------------------------|----------|---------|----------------|
| L7805  |         |                        |          | L7805T  | 5V             |
| L7805C | L7805CV | L7805CD2T              | L7805CP  | L7805CT | 5V             |
| L7852C | L7852CV | L7852CD2T              | L7852CP  | L7852CT | 5.2V           |
| L7806  |         |                        |          | L7806T  | 6V             |
| L7806C | L7806CV | L7806CD2T              | L7806CP  | L7806CT | 6V             |
| L7808  |         |                        |          | L7808T  | 8V             |
| L7808C | L7808CV | L7808CD2T              | L7808CP  | L7808CT | 8V             |
| L7885C | L7885CV | L7885CD2T              | L7885CP  | L7885CT | 8.5V           |
| L7809C | L7809CV | L7809CD2T              | L7809CP  | L7809CT | 9V             |
| L7812  |         |                        |          | L7812T  | 12V            |
| L7812C | L7812CV | L7812CD2T              | L7812CP  | L7812CT | 12V            |
| L7815  |         |                        |          | L7815T  | 15V            |
| L7815C | L7815CV | L7815CD2T              | L7815CP  | L7815CT | 15V            |
| L7818  |         |                        |          | L7818T  | 18V            |
| L7818C | L7818CV | L7818CD2T              | L7818CP  | L7818CT | 18V            |
| L7820  |         |                        |          | L7820T  | 20V            |
| L7820C | L7820CV | L7820CD2T              | L7820CP  | L7820CT | 20V            |
| L7824  |         |                        |          | L7824T  | 24V            |
| L7824C | L7824CV | L7824CD2T              | L7824CP  | L7824CT | 24V            |

(\*) AVAILABLE IN TAPE AND REEL WITH "TR" SUFFIX

ตารางที่ ๗4 แสดง peak reverse voltage ของไดโอดแต่ละเบอร์

**Silicon Rectifier**



Dimensions / Maße in mm

**Silizium Gleichrichter**

|   |                               |
|---|-------------------------------|
| Nominal current – Nennstrom   | 1 A                           |
| Repetitive peak reverse voltage<br>Periodische Spitzenspannung                        | 50...2000 V                   |
| Plastic case<br>Kunststoffgehäuse   | DO-41                         |
| Weight approx. – Gewicht ca.  | 0.4 g                         |
| Plastic material has UL classification 94V-0<br>Gehäusematerial UL94V-0 klassifiziert |                               |
| Standard packaging taped in ammo pack<br>Standard Lieferform gegurtet in Ammo-Pack    | see page 17<br>siehe Seite 17 |

**Maximum ratings**

**Grenzwerte**

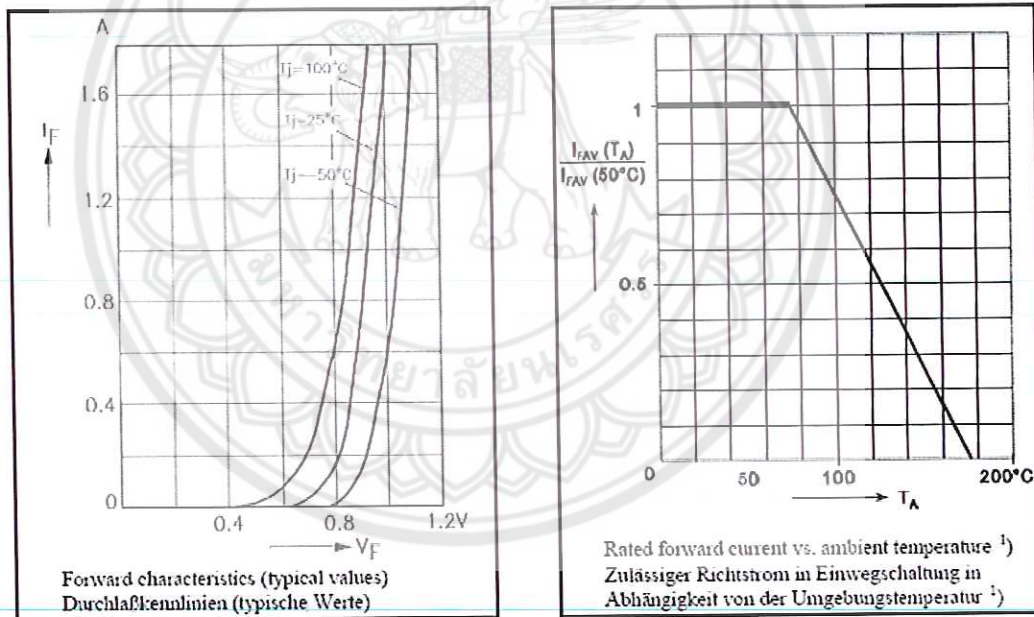
| Type<br>Typ  | Repetitive peak reverse voltage<br>Periodische Spitzenspannung<br>$V_{FRM}$ [V] | Surge peak reverse voltage<br>Stoßspitzenspannung<br>$V_{FSM}$ [V] |
|--------------|---|--|
| 1N 4001      | 50  | 50   |
| 1N 4002      | 100   | 100  |
| 1N 4003      | 200   | 200  |
| 1N 4004      | 400   | 400  |
| 1N 4005      | 600   | 600  |
| 1N 4006      | 800   | 800  |
| 1N 4007      | 1000  | 1000   |
| 1N 4007-1300 | 1300  | 1300   |
| EM 513       | 1600  | 1600   |
| EM 516       | 1800  | 1800   |
| EM 518       | 2000  | 2000   |

|   |   |                        |   |
|---|---|------------------------|---|
| Max. average forward rectified current, R-load<br>Dauergrenzstrom in Einwegschaltung mit R-Last | $T_A = 75^\circ\text{C}$<br>$T_A = 100^\circ\text{C}$ | $I_{FAV}$<br>$I_{FAV}$ | 1 A <sup>1)</sup><br>0.75 A <sup>1)</sup> |
| Repetitive peak forward current<br>Periodischer Spitzenstrom                                    | $f > 15$ Hz   | $I_{FRM}$              | 10 A <sup>1)</sup>                        |

<http://www.datasheetcatalog.com>

|  |                          |                |  |
|--|--------------------------|----------------|--|
| Rating for fusing, $t < 10$ ms<br>Grenzlastintegral, $t < 10$ ms                                     | $T_A = 25^\circ\text{C}$ | $i^2t$         | $12.5 \text{ A}^2\text{s}$                                       |
| Peak fwd. surge current, 50 Hz half sine-wave<br>Stoßstrom für eine 50 Hz Sinus-Halbwellen,          | $T_A = 25^\circ\text{C}$ | $I_{FSM}$      | 50 A   |
| Operating junction temperature – Sperrschichttemperatur<br>Storage temperature – Lagerungstemperatur |                          | $T_J$<br>$T_S$ | $-50 \dots +175^\circ\text{C}$<br>$-50 \dots +175^\circ\text{C}$ |

| Characteristics   |                           |                     | Kennwerte       |                       |
|---|---------------------------|---------------------|-----------------|-----------------------|
| Forward voltage – Durchlaßspannung  | $T_J = 25^\circ\text{C}$  | $I_F = 1 \text{ A}$ | $V_F$           | $< 1.1 \text{ V}$     |
| Leakage current – Sperrstrom  | $T_J = 25^\circ\text{C}$  | $V_R = V_{RRM}$     | $I_R$           | $< 5 \mu\text{A}$     |
|   | $T_J = 100^\circ\text{C}$ | $V_R = V_{RRM}$     | $I_R$           | $< 50 \mu\text{A}$    |
| Thermal resistance junction to ambient air<br>Wärmewiderstand Sperrschicht – umgebende Luft |                           |                     | $R_{\theta JA}$ | $< 45 \text{ K/W}^1)$ |



รูปที่ ๗7 แสดงคุณสมบัติสำคัญของ 1N4007

# ประวัติผู้เขียนโครงการ



ชื่อ นายอดิเรก ปั่นมี  
 เกิดวันที่ 22 พฤษภาคม 2526  
 สถานที่อยู่ปัจจุบัน 21 หมู่ 4 ตำบลจี้วราย อำเภอตะพานหิน  
 จังหวัดพิจิตร 66110

## ประวัติการศึกษา

- จบระดับมัธยมศึกษาตอนปลายจากโรงเรียนตะพานหิน
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4  
 คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมไฟฟ้า  
 มหาวิทยาลัยนเรศวร  
 e-mail: ton\_ton\_ice@hotmail.com



ชื่อ นายกฤษฏา เต็งเรียบ  
 เกิดวันที่ 14 ตุลาคม 2526  
 สถานที่อยู่ปัจจุบัน 69 หมู่ 8 ตำบลคลองมะเดื่อ  
 อำเภอกะทู้มบ้าน จังหวัดสมุทรสาคร  
 74110

## ประวัติการศึกษา

- จบระดับมัธยมศึกษาตอนปลายจาก โรงเรียนนวมินทราชินูทิศ  
 สตรีวิทยา พุทธมณฑล
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4  
 คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมไฟฟ้า  
 มหาวิทยาลัยนเรศวร  
 e-mail: toei08\_@hotmail.com