

# เครื่องทำอาหารอัตโนมัติ

## AUTOMATIC COOKING MACHINE



เลขหม.....  
เลขทะเบียน..... 42258  
วัน, เดือน, ปี..... 16 พ.ค. 2545

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

ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต  
สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**AUTOMATIC COOKING MACHINE**



**MR. NATEE OUPORN**

**MR. NORARAT SIMAPICHATE**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE  
REQUIREMENT FOR THE DEGREE OF MACHELOR OF TECHNOLOGY  
ELECTRONICS**

**FACULTY OF ENGINEERING**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG 2000**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์

เครื่องทำอาหารอัตโนมัติ

AUTOMATIC COOKING MACHING

นักศึกษา

นาย นที อวยพร

เลขประจำตัว 41013293

นาย นรรัตน์ สิมะพิเชฐ

เลขประจำตัว 41013294

อาจารย์ที่ปรึกษา

อาจารย์ มยุรี เลิศเวชกุล

ผศ.ดร. ปิติเขต สุรักษา

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้  
นับปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมบัณฑิต

อาจารย์ที่ปรึกษา

(อาจารย์ มยุรี เลิศเวชกุล)

อาจารย์ที่ปรึกษา

(ผศ.ดร. ปิติเขต สุรักษา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์

เครื่องทำอาหารอัตโนมัติ

AUTOMATIC COOKING MACHING

นักศึกษา

นาย นที อวยพร เลขประจำตัว 41013293

นาย นรรัตน์ สิมะพิเชฐ เลขประจำตัว 41013294

อาจารย์ที่ปรึกษา

ผศ.ดร. ปิติเขต ผู้รักษา

อาจารย์ มยุรี เลิศเวชกุล

ภาควิชา

เทคนิคอุตสาหกรรม

ปีการศึกษา

2543

บทคัดย่อ

โครงการระดับปริญญาตรีในหัวข้อเรื่อง “ เครื่องทำอาหารอัตโนมัติ ” ซึ่งในโครงการนี้ ได้มีการศึกษาการออกแบบ โครงสร้างการทำงานในส่วนต่าง ๆ ให้ทำงานมีความสามารถเท่า มนุษย์ ที่สำคัญจะประหยัดเวลาในการทำอาหารและได้อาหารที่มีรสชาติมาตรฐานทุกครั้ง ดังนั้น จึงได้มีการออกแบบโครงสร้าง , ภาควควบคุม , การใช้งานจากการหมุนของมอเตอร์ ซึ่งจะออกแบบเป็นการทำงานในส่วน Hardware , Software และวงจร

**Thesis Title** AUTOMATIC COOKING MACHING

**Student** Mr. Natee Ouporn ID 41013293  
Mr. Norarat Simapichate ID 41013294

**Advisor** Mr. Mayuree Lertwatechakul  
Asst. Prof. Dr. Pitikhate Sooraksa

**Academic Year** 2000



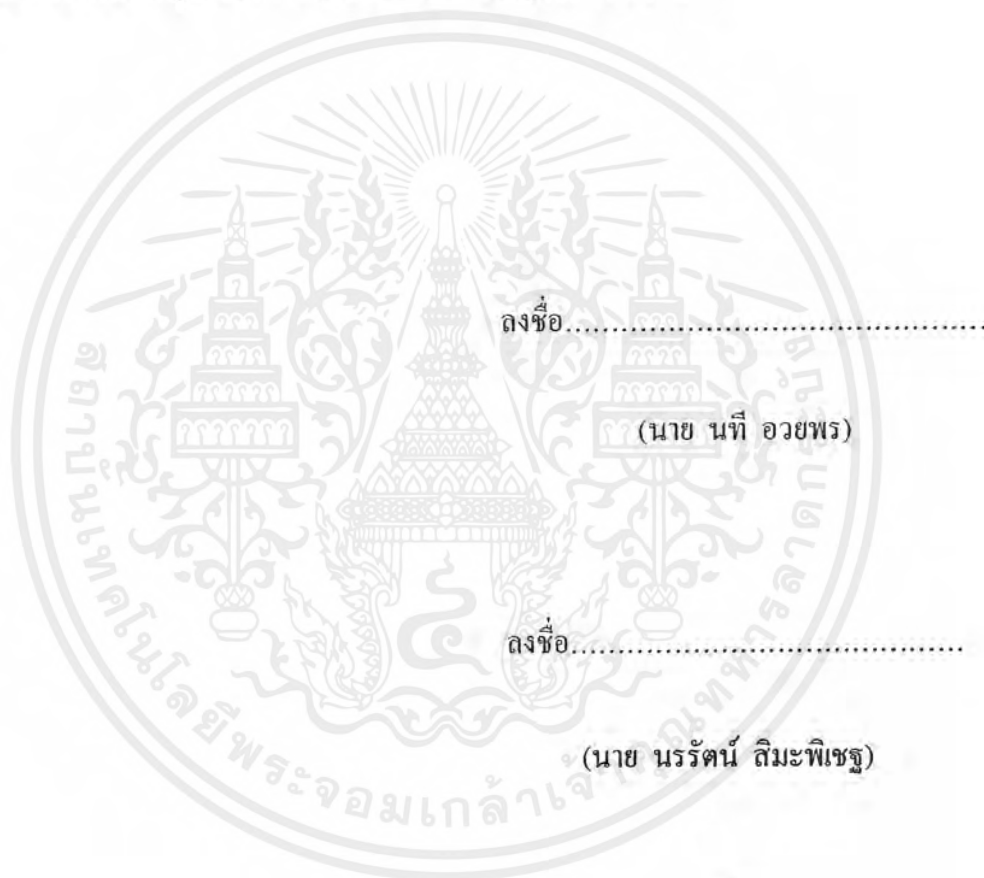
### ABSTRACT

Undergraduated student project topic "Automatic Cooking Machine" Automatic Cooking Machine is the well designed machine for cooking. Not only it can give you great dishes like cooking by a chef, but it also spend less time while still maintaining the standard tastes. The motor which is the control part that make the machine run, consists of the programmed hardware, and circuits.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

จากความสำเร็จในการสร้าง เครื่องทำอาหารอัตโนมัติ คณะผู้จัดทำขอขอบพระคุณท่าน อาจารย์ ผศ.ดร. ปิติเขต สุรักษา (อาจารย์ที่ปรึกษา) และ อาจารย์ มยุรี เลิศเวชกุล (อาจารย์ที่ปรึกษา) ที่ได้ให้คำชี้แนะให้การสนับสนุนให้ความช่วยเหลือในทุก ๆ ด้าน และขอขอบพระคุณ ภาควิชาเทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ได้ให้การสนับสนุน จนกระทั่งโครงการนี้สำเร็จลุล่วงไปได้ด้วยดี



คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูปภาพ	จ
<b>บทที่ 1 บทนำ</b>	
1.1 ความเป็นมาและความสำคัญ	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	1
1.3 เนื้อหาปริญญานิพนธ์	1
<b>บทที่ 2 ทฤษฎีและหลักการ</b>	
2.1 ไมโครคอนโทรลเลอร์ตระกูล MCS – 51	3
2.2 สัญญาณต่าง ๆ ของไมโครคอนโทรลเลอร์	5
2.3 หน้าที่และการทำงานของสัญญาณต่าง ๆ	6
2.4 มอเตอร์	7
2.5 คุณลักษณะประจำของมอเตอร์	15
2.6 การควบคุมทิศทางการหมุนของมอเตอร์	16
2.7 การควบคุมความเร็วของมอเตอร์	19
2.8 การเข้ารหัสสำหรับเพลาหมุน	21
2.9 อุปกรณ์ตรวจจับ	22
2.10 ระบบส่งกำลังของโครงการ	27
<b>บทที่ 3 การออกแบบและวงจร</b>	
3.1 หลักการทำงานต่าง ๆ	35
3.2 หน้าที่ต่าง ๆ ของมอเตอร์ไฟฟ้ากระแสตรง	46
3.3 ขั้นตอนในการทำงานของมอเตอร์กระแสตรง	46
<b>บทที่ 4 สรุปผลการทำงาน</b>	
ภาคผนวก ก. ลายวงจร	
ภาคผนวก ข. โปรแกรม	
ภาคผนวก ค. คู่มือการใช้งานต่าง ๆ	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

เรื่อง	หน้า
รูปที่ 2.1 ตำแหน่งขา MCS-51	5
รูปที่ 2.2 หลักการของมอเตอร์	8
รูปที่ 2.3 อากัปกิริยาของเครื่องกำเนิดไฟฟ้าเมื่อได้รับ Load	10
รูปที่ 2.4 อากัปกิริยาของมอเตอร์	11
รูปที่ 2.5 แสดงทิศทางแรงเคลื่อนไฟ และกระแสที่เกิดขึ้นในมอเตอร์ซึ่งไหลไปในทางเดียวกันแต่ตรงข้ามกับแรงเคลื่อนไฟสวน	12
รูปที่ 2.6 วงจรของมอเตอร์ไฟตรง	12
รูปที่ 2.7 แสดงวงจรควบคุมมอเตอร์ที่ใช้รีเลย์	17
รูปที่ 2.8 แสดงการต่อทรานซิสเตอร์ 4 ตัว เพื่อควบคุมทิศทางการหมุนของมอเตอร์	18
รูปที่ 2.9 แสดงวงจรควบคุมมอเตอร์ควบคุมด้วยเพาเวอร์มอสเฟต	20
รูปที่ 2.10 แสดงวงจรตัวอย่างที่ใช้ในการควบคุมทิศทางการหมุน, การหยุดหมุนและความเร็วของมอเตอร์	20
รูปที่ 2.11 ตัวเข้ารหัสสำหรับเพลาหมุนอย่างง่ายโดยใช้ไมโครสวิทช์	23
รูปที่ 2.12 ตัวเข้ารหัสแบบใช้หลักการฮอลล์เอฟเฟกต์	24
รูปที่ 2.13 ตัวเข้ารหัสแบบใช้หลักการการสะท้อนแสง	25
รูปที่ 2.14 ตัวเข้ารหัสแบบใช้หลักการตัดแสง	26
รูปที่ 2.15 เฟืองตรง	30
รูปที่ 2.16 เฟืองเฉียงหรือเฟืองฟันเฉียง	31
รูปที่ 2.17 เฟืองคอกจอก	32
รูปที่ 2.18 เฟืองหนอน	33
รูปที่ 3.1 ชุดดำเตียงหรือชุดส่งอาหาร	37
รูปที่ 3.2 ชุดผัดอาหาร	38
รูปที่ 3.3 ชุดเทอาหาร	39
รูปที่ 3.4 โครงสร้างทั้งหมดของเครื่องทำอาหารอัตโนมัติ	40
รูปที่ 3.5 วงจรการทำงาน	42
รูปที่ 3.6 Flowchart แสดงการทำงานของเครื่องทำอาหารอัตโนมัติ	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาและความสำคัญ

ในโลกยุคของ IT ในปัจจุบันนี้ ได้มีการคิดค้นเครื่องอำนวยความสะดวกต่าง ๆ ขึ้นอย่างมากมายทั้งเครื่องใช้ หรือ รวมถึงอุปกรณ์การติดต่อสื่อสาร ที่มีประสิทธิภาพ ซึ่งรวมไปถึงการพัฒนาทางด้านคอมพิวเตอร์ ให้ใช้ประโยชน์ได้อย่างอเนกประสงค์ ซึ่งใช้ในการควบคุมสายการผลิตในโรงงานอุตสาหกรรม ส่วนอุปกรณ์ที่มีประสิทธิภาพต่ำกว่าคอมพิวเตอร์ลงมาก็จะเป็นการใช้ไมโครคอนโทรลเลอร์ในการควบคุมสายการผลิต ที่มีความซับซ้อนไม่มากนัก จุดนี้ได้นำมาประยุกต์ใช้งานในด้านควบคุมสายการผลิตเช่นเดียวกัน ซึ่งในปัจจุบันเวลาเป็นสิ่งที่สำคัญมากต่อชีวิตประจำวันของโลกยุค IT และกิจกรรมที่สำคัญ คือ การรับประทานอาหาร ซึ่งจะต้องใช้เวลาในช่วงนี้เป็นเวลาพอสมควร จากจุดนี้เอง จึงได้มีการประดิษฐ์โครงงานที่มีชื่อเรียกว่า “ Automatic Cooking Machine ” ก็คือเครื่องทำอาหารอัตโนมัติ ที่สามารถควบคุมเวลา (Fixed time) ที่แน่นอนในการทำอาหารจะได้รสชาติของอาหารที่คงรสชาติที่แน่นอนอีกด้วย

#### 1.2 วัตถุประสงค์

การทำโครงงานนี้ (Automatic Cooking Machine) ขึ้นมานั้น จุดประสงค์หลัก ๆ คือ สามารถทำอาหารได้อย่างมีรสชาติมาตรฐาน โดยที่ไม่ต้องใช้นุ้ย ซึ่งการทำงานหรือทำอาหารในแต่ละครั้งนั้นต้องใช้เวลาที่แน่นอน การทำงานในลักษณะนี้ต้องใช้อุปกรณ์ที่เรียกว่า ไมโครคอนโทรลเลอร์ มาช่วยในการควบคุมการทำงานด้วยโปรแกรมสั่งการให้โครงงานทำงานได้อย่างอัตโนมัติ และกำหนดเวลาที่แน่นอน

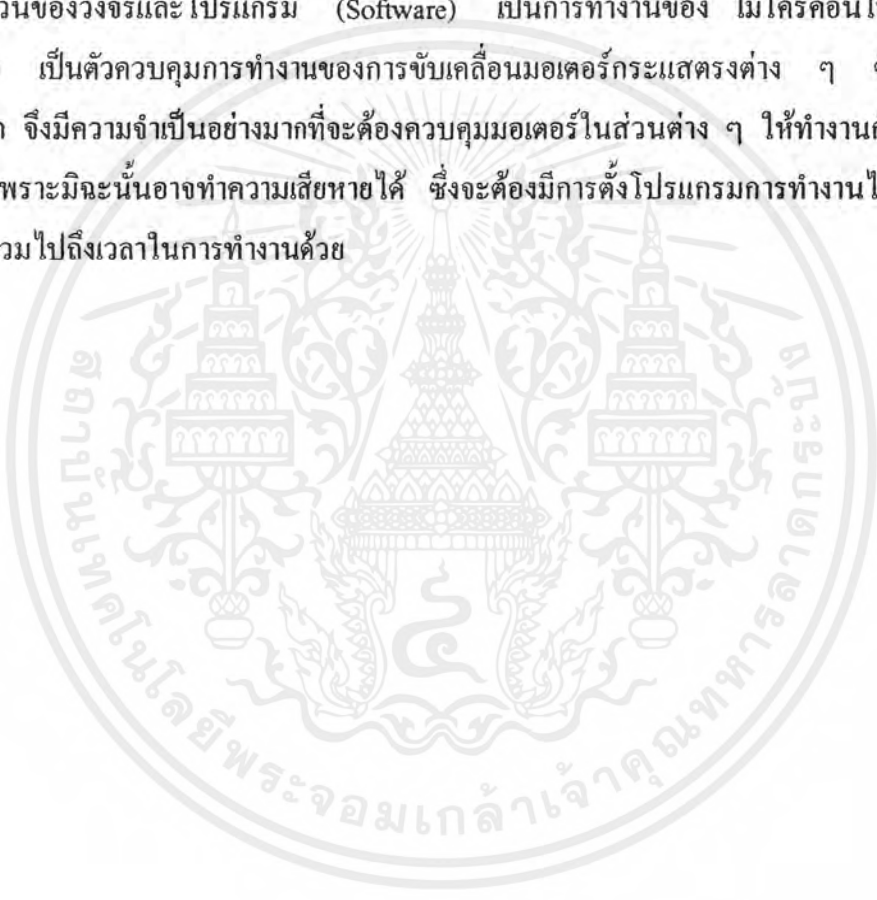
#### 1.3 เนื้อหา

โครงงานเครื่องทำอาหารอัตโนมัติ (Automatic Cooking Machine) นี้ เสมือนเป็นการรวมการทำงานของหลาย ๆ อย่างเอาไว้ในเครื่องเดียวกัน ไม่ว่าจะเป็นในส่วนของโครงสร้างทาง Hardware ทางโปรแกรม (Software) , ทางวงจร และไม่ว่าจะเป็นการทำอาหารให้ได้เวลาที่ดี ซึ่งจะกล่าวเป็นส่วน ๆ ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของโครงสร้างทาง Hardware จะประกอบไปด้วย ชุดฝึก , ชุดลำเลียง , ชุดกระทะ โดยแต่ละส่วนจะทำงานอย่างสัมพันธ์กันซึ่งได้โปรแกรมไว้ในส่วนของโปรแกรม โดยที่ชุดฝึกจะทำได้โดยมีมอเตอร์กระแสตรงเป็นตัวขับเคลื่อนเพื่อที่จะขับกันให้หมุนไปผัดอาหารได้ ในส่วนของชุดลำเลียงจะอาศัยหลักการของการหมุนของสายพานเป็นตัวลำเลียงอาหาร ซึ่งเป็นส่วนประกอบในการปรุงอาหารให้มาเทลงกระทะ โดยสามารถขับเคลื่อนด้วยมอเตอร์กระแสตรง ส่วนชุดกระทะจะเป็นฐานโครง โดยจะติดกระทะไว้กับแท่งเกลียวโดยจะติดกับสายพานซึ่งมีมอเตอร์กระแสตรงเป็นตัวหมุนให้กระทะสามารถไปเทลงจานได้

ส่วนของวงจรและโปรแกรม (Software) เป็นการทำงานของ ไมโครคอนโทรลเลอร์ MCS -51 เป็นตัวควบคุมการทำงานของการทำงานขับเคลื่อนมอเตอร์กระแสตรงต่าง ๆ ซึ่งใช้เป็นจำนวนมาก จึงมีความจำเป็นอย่างมากที่จะต้องควบคุมมอเตอร์ในส่วนต่าง ๆ ให้ทำงานกันอย่างมีขั้นตอน เพราะมิฉะนั้นอาจทำความเสียหายได้ ซึ่งจะต้องมีการตั้งโปรแกรมการทำงานไว้ ซึ่งในส่วนนี้จะรวมไปถึงเวลาในการทำงานด้วย



## บทที่ 2

### ทฤษฎีและหลักการ

#### 2.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51

เป็นไมโครโพรเซสเซอร์ขนาด 8 บิตที่มีอุปกรณ์สนับสนุนอยู่ภายในหลายอย่างได้แก่ หน่วยความจำสำหรับเก็บข้อมูล หน่วยความจำสำหรับเก็บโปรแกรม ตัวตั้งเวลา/ตัวนับ อุปกรณ์รับส่งข้อมูลแบบอนุกรม เนื่องจากโครงสร้างของไมโครคอนโทรลเลอร์มีอุปกรณ์สนับสนุนอยู่ภายในนี้เอง ทำให้การใช้งานง่ายขึ้นและมีประสิทธิภาพมากขึ้นโดยไม่ต้องมีการเชื่อมต่ออุปกรณ์ภายนอกเพิ่มเติมมากเหมือนตัวไมโครโปรเซสเซอร์ทั่วไป นอกจากนี้หากเราต้องการใช้งานร่วมกับอุปกรณ์อื่นเพิ่มเติมเช่น ไอซี 8255 หรือหน่วยความจำภายนอก เรายังสามารถนำมาเชื่อมต่อเพิ่มเติมเข้ากับไมโครคอนโทรลเลอร์ได้อีกด้วย

#### 2.1.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51 ประกอบด้วยอุปกรณ์ต่างๆ

ดังนี้

- หน่วยความจำภายในสำหรับเก็บข้อมูลขนาด 128 ไบต์
- หน่วยความจำภายในสำหรับเก็บโปรแกรมขนาด 4 กิโลไบต์
- อุปกรณ์ควบคุมการอินเตอร์รัพต์
- ตัวตั้งเวลาและตัวนับขนาด 16 บิต 2 ชุด
- พอร์ตควบคุมการสื่อสารอนุกรมแบบ Full Duplex ซึ่งสามารถรับส่งข้อมูลพร้อมกันได้
- พอร์ตขานสำหรับติดต่อกับอุปกรณ์ภายนอกจำนวน 4 พอร์ตๆละ 8 บิต
- วงจรผลิตสัญญาณภายใน

#### 2.1.2 หน่วยความจำสำหรับเก็บโปรแกรม (Internal Program Memory)

หน่วยความจำสำหรับเก็บโปรแกรม ทำหน้าที่เก็บโปรแกรมที่ผู้ใช้เขียนขึ้นเพื่อควบคุมการทำงานของไมโครคอนโทรลเลอร์โดยหน่วยความจำจะเป็นแบบ ROM มีความจุ 4 Kbytes (ตำแหน่ง 0000H - 0FFFH) ในการใช้เราสามารถกำหนดให้ไมโครคอนโทรลเลอร์เลือกใช้โปรแกรมที่เก็บอยู่ภายในตัวไมโครคอนโทรลเลอร์หรือโปรแกรมที่เก็บอยู่ในหน่วยความจำที่อยู่ภายนอกได้ การเลือกติดต่อกันได้โดยการป้อนสัญญาณควบคุมให้ที่ขา EA ถ้าต้องการให้ไมโครคอนโทรลเลอร์เลือกโปรแกรมที่เก็บอยู่ภายนอกให้ใช้ขา EA นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนโทรลเลอร์ติดต่อกับโปรแกรมที่อยู่ในหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ จะต่อขานี้กับลอจิก 1 หากต้องการให้ไมโครคอนโทรลเลอร์ติดต่อกับโปรแกรมที่อยู่ในหน่วยความจำภายนอกจะต่อขานี้กับลอจิก 0 การติดต่อกับหน่วยความจำโปรแกรมภายนอกได้ทั้งหมด 64 Kbyte (ตำแหน่ง 0000H – FFFFH)

### 2.1.3 หน่วยความจำข้อมูลภายใน (Internal Data Memory)

หน่วยความจำข้อมูลภายในทำหน้าที่เก็บข้อมูลทั่วไป และทำหน้าที่เป็นสแตค (Stack) บางส่วน หน่วยความจำข้อมูลภายในเบอร์ 8051 มีอยู่ 128 ไบต์ โดยอยู่ที่ตำแหน่ง 00H – 7FH

### 2.1.4 อุปกรณ์ควบคุมการอินเทอร์รัพต์ (Interrupt Control Unit)

เป็นส่วนที่ทำหน้าที่ควบคุมการอินเทอร์รัพต์ของไมโครคอนโทรลเลอร์ MCS – 51 ซึ่งมีการร้องขอการอินเทอร์รัพต์ได้จาก 6 แหล่งกำเนิด คือ สัญญาณจากภายนอก 2 สัญญาณ จากตัว Timer 0, Timer 1 และ Timer 2 และจากอุปกรณ์รับส่งข้อมูลแบบอนุกรม 1 สัญญาณ สัญญาณอินเทอร์รัพต์ที่เกิดขึ้นเราสามารถควบคุมให้ไมโครคอนโทรลเลอร์ตอบรับหรือไม่ตอบรับก็ได้ นอกจากนี้เราสามารถจัดลำดับความสำคัญของการอินเทอร์รัพต์จากสัญญาณได้เป็น 2 ระดับแตกต่างกัน

### 2.1.5 ตัวตั้งเวลาและตัวนับ (Timer/Counter)

ในไมโครคอนโทรลเลอร์เบอร์ 8051 ประกอบด้วยรีจิสเตอร์ตัวตั้งเวลาและตัวนับ ขนาด 16 บิต จำนวน 2 ชุด คือ Timer 0 และ Timer 1 โดย Timer ทั้งหมดสามารถกำหนดให้ทำงานในลักษณะตัวนับและตัวตั้งเวลาได้

การทำงานในโหมดของตัวตั้งเวลา ค่าในรีจิสเตอร์จะเพิ่มขึ้นทุกๆแมกซ์ิมัซเกิด โดย 1 แมกซ์ิมัซเกิดประกอบด้วยสัญญาณนาฬิกา 12 ลูก ดังนั้นอัตราจับเวลาจะเป็น 1/12 เท่าของความถี่สัญญาณนาฬิกาของระบบ ค่าสูงสุดที่ตั้งได้คือ  $2^{16}$

การทำงานในโหมดการนับ ค่าของการนับจะเพิ่มขึ้นเมื่อมีสัญญาณนาฬิกาเข้ามาที่ขา T0 หรือ T1 เปลี่ยนจาก 1 เป็น 0 ความเร็วในการนับสูงสุดคือ 1/24 เท่าของสัญญาณนาฬิกา โดยสัญญาณนาฬิกาที่เข้ามาที่ขา T0 หรือ T1 จะมี Duty Cycle เท่าใดก็ได้

### 2.1.6 พอร์ตอินพุตเอาต์พุต

ในไมโครคอนโทรลเลอร์ MCS-51 ประกอบด้วยพอร์ตอินพุตเอาต์พุต 2 ทิศทาง จำนวน 4 พอร์ต แต่ละพอร์ตมีอุปกรณ์แลตช์ข้อมูลและเอาต์พุตไครเวอ์ประกอบอยู่ทางด้านเอาต์พุต และ

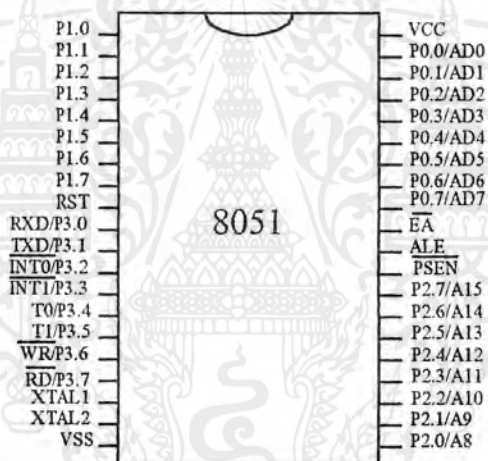
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางด้านอินพุตจะมีบัฟเฟอร์ ( พอร์ตทั้ง 4 เป็นริจิสเตอร์พิเศษชื่อ P0,P1,P2,P3 ) เราสามารถใช้งานแต่ละพอร์ตเป็นอินพุตหรือเอาต์พุตได้ตามต้องการ แต่ละพอร์ตสามารถเชื่อมต่อกับสัญญาณ TTL ได้โดยตรง

ในการติดต่อกับหน่วยความจำภายนอกพอร์ต P0,P2 จะใช้สำหรับกำหนดตำแหน่งของหน่วยความจำภายนอก โดยพอร์ต P0 จะทำงานในลักษณะมัลติเพล็กซ์คือเป็นทั้งพอร์ตตำแหน่งและพอร์ตข้อมูล โดย P0 จะเป็นตำแหน่งของหน่วยความจำด้านต่ำ และ P2 จะเป็นตำแหน่งของหน่วยความจำด้านสูง

## 2.2 สัญญาณต่างๆของไมโครคอนโทรลเลอร์MCS-51

ไมโครคอนโทรลเลอร์ในตระกูล MCS-51 เป็นไอซีขนาด 40 ขาซึ่งมีสัญญาณต่าง ๆ ดังรูป



รูปที่ 2.1

สัญญาณต่างๆของไมโครคอนโทรลเลอร์ สามารถจำแนกการทำงานเป็น 3 กลุ่มคือ

- กลุ่มสัญญาณตำแหน่งเป็นตัวกำหนดตำแหน่งของหน่วยความจำ
- กลุ่มสัญญาณควบคุม เป็นสัญญาณควบคุมการทำงานของไมโครคอนโทรลเลอร์
- กลุ่มสัญญาณข้อมูลเป็นทางผ่านของข้อมูลระหว่างไมโครคอนโทรลเลอร์กับหน่วยความจำ

## 2.3 หน้าที่ใช้และการใช้งานของสัญญาณต่างๆเป็นดังนี้

VCC สำหรับต่อกับไฟเลี้ยง 5V

VSS สำหรับต่อกราวด์

XTAL1 เป็นอินพุตของภาคขยายสัญญาณแบบอินเวอร์สของวงจรผลิตสัญญาณนาฬิกา

XTAL2 เป็นเอาต์พุตของภาคขยายสัญญาณแบบอินเวอร์สของวงจรผลิตสัญญาณนาฬิกา

RST สัญญาณรีเซ็ต ไมโครคอนโทรลเลอร์จะถูกรีเซ็ตเมื่อสัญญาณที่ขานี้มีค่าเป็นลอจิก 1 นานไม่ต่ำกว่า 2 แมกซ์ซีไนซ์เกิด

ALE/PROG (Address Latch Enable) เป็นสัญญาณเอาต์พุต ซึ่งไมโครคอนโทรลเลอร์จะส่งออกไปเป็นพัลส์เพื่อแลตช์ค่าตำแหน่งไบต์ค่าที่อยู่ที่พอร์ต P0 ในขณะที่ติดต่อกับหน่วยความจำภายนอกสัญญาณนี้จะถูกส่งด้วยอัตราคงที่คือ 1/6 เท่าของความถี่สัญญาณนาฬิกาซึ่งสามารถนำไปใช้เป็นสัญญาณนาฬิกาให้กับอุปกรณ์ภายนอกได้ สัญญาณพัลส์นี้จะถูกข้ามไป 1 พัลส์เมื่อมีการติดต่อกับหน่วยความจำข้อมูลภายนอกและสัญญาณนี้จะใช้เป็นอินพุตเพื่อควบคุมโปรแกรม PROM ภายในตัวไมโครคอนโทรลเลอร์ด้วย

PSEN (Program Store Enable) เป็นเอาต์พุต สำหรับส่งสัญญาณสไตรป (พัลส์ต่ำ) เพื่ออ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก (External Program Memory) เมื่อซีพียูอ่านรหัสคำสั่งจากหน่วยความจำภายนอก จะส่งสัญญาณสไตรปออกมา 2 ครั้งใน 1 แมกซ์ซีไนซ์เกิดแต่สัญญาณสไตรปทั้ง 2 ครั้งจะถูกข้ามไปหากเป็นช่วงที่ซีพียูติดต่อกับ External Program Memory

EA (External Access) เป็นสัญญาณอินพุต ใช้สำหรับควบคุมให้ไมโครคอนโทรลเลอร์เลือกติดต่อกับโปรแกรมที่เก็บอยู่ในหน่วยความจำโปรแกรมภายใน หรือโปรแกรมที่เก็บอยู่ในหน่วยความจำภายนอกตัวไมโครคอนโทรลเลอร์ หากให้ค่าลอจิก 1 ที่ขาเป็นการเลือกใช้หน่วยความจำโปรแกรมภายในตัวไมโครคอนโทรลเลอร์ หากต้องการให้ซีพียูติดต่อกับหน่วยความจำโปรแกรมภายนอกต่อสัญญาณเข้ากับลอจิก 0 หรือ VSS ถึงแม้ว่าเบอร์ 8031 ไม่มี EPROM ภายในก็ต้องต่อขานี้ลงกราวด์ด้วย ในกรณีของการโปรแกรม ROM ภายในไมโครคอนโทรลเลอร์จะต่อขานี้เข้ากับไฟ 21 V ถ้าเป็น 8751 AH แต่หากเป็น 8751 BH ต้องต่อกับ 12.75 V

Port 0 ได้แก่ขาที่ 32 – 39 ของ MCS-51 สามารถใช้เป็นอินพุตเอาต์พุตได้นอกจากนี้ในการติดต่อกับหน่วยความจำภายนอกยังใช้เป็นขา Address Bus และ Data Bus อีกด้วย

Port 1 ได้แก่ขาที่ 1 – 8 เป็นพอร์ต 8 บิต สามารถอ้างทีละบิตได้ คือ P1.0,P1.1,...etc

Port 2 ได้แก่ขาที่ 21 – 28 จะใช้งาน 2 หน้าที คือใช้เป็นพอร์ต 8 บิตกับใช้เป็นขาแอดเดรส 8 บิต ในการอ้างหน่วยความจำภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Port 3 ได้แก่ขาที่ 10 – 17 จะใช้งานสองหน้าที่คือ เป็นพอร์ตอินพุตและเอาต์พุต และใช้เป็นขาควบคุมต่างๆ

## 2.4 มอเตอร์ (Motor)

มอเตอร์ที่ใช้ในโครงงานนี้จะเป็นมอเตอร์ไฟฟ้ากระแสตรง (Direct Current Motors) ซึ่งจะกล่าวถึงหลักการต่างๆ ดังนี้

### 2.4.1 หลักการของ Motor (Motor's Principle)

มอเตอร์ก็คือ ตัวเปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานกล เพื่อนำพลังงานกลที่ได้ไปขับเคลื่อนสิ่งต่าง ๆ ตามที่ต้องการ อาการทางกลที่เกิดขึ้น ก็อาศัยหลักการที่ว่า เมื่อมีกระแสไหลในตัวนำ ซึ่งอยู่ในสนามแม่เหล็กนั้น ย่อมทำให้เกิดแรงขึ้นในทิศทางที่หาได้จากกฎมือซ้ายของเฟลมมิ่ง ขนาดของแรงที่เกิดขึ้น หาได้จาก

$$F = BIl \quad (2.1)$$

เมื่อ  $F$  = แรงที่เกิดขึ้นบนตัวนำหนึ่งตัว (นิวตัน)

$B$  = ความหนาแน่นของสนามไฟฟ้า (เวเบอร์)

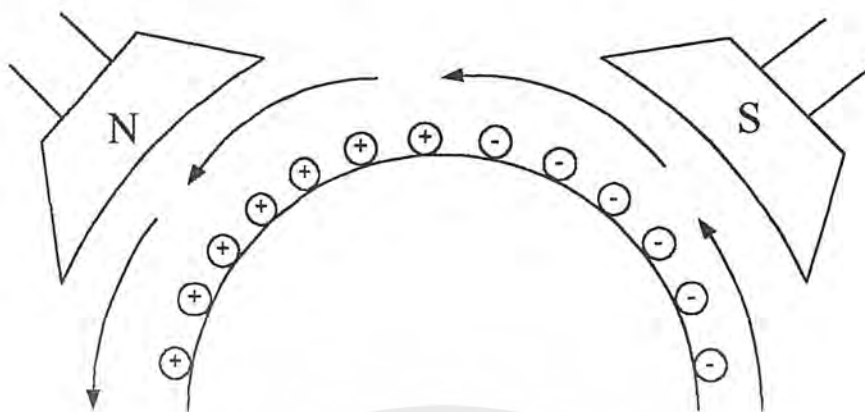
$I$  = กระแสที่ไหลในตัวนำ (แอมป์)

$L$  = ความยาวของตัวนำ (เมตร)

แรง  $F$  ที่เกิดขึ้น จะอยู่ในแนวที่ตั้งฉากกับสนามแม่เหล็ก และกระแสที่ไหลผ่านในตัวนำนั้น ๆ

เครื่องกลไฟฟ้ากระแสตรงนี้ โดยหลักการพื้นฐานของโครงสร้างของเครื่องกำเนิดไฟและมอเตอร์นั้น ไม่มีความแตกต่างในโครงสร้างเลยนั่นก็คือ เครื่องกลกระแสตรงเครื่องเดียวกันนี้สามารถนำไปใช้งานสลับกันได้

มอเตอร์ไฟฟ้ากระแสตรงก็คล้ายกับเครื่องกำเนิดไฟกระแสตรง นั่นก็คือ มีเครื่องแบบ Shunt – Wound หรือ Series – Wound หรือ Compound – Wound เช่นกัน



รูปที่ 2.2 หลักการของมอเตอร์

จากรูปแสดงส่วนหนึ่งของมอเตอร์ไฟฟ้ากระแสตรงที่มีหลายขั้วแม่เหล็ก เมื่อใส่ไฟเข้าไปในสนามกระตุ้น ขั้วทำให้เกิดอำนาจแม่เหล็ก และเมื่อป้อนกระแสให้ไหลผ่านตัวนำที่อาร์เมเจอร์ ก็จะทำให้เกิดแรงขึ้นบนตัวนำที่อยู่ในสนามแม่เหล็กนั้นคือ ทำให้อาร์เมเจอร์หมุน ในที่นี้ได้กำหนดให้กระแสที่ไหลในตัวนำของอาร์มาเจอร์ที่อยู่ภายใต้ขั้วเหนือ N มีทิศทางของกระแสพุ่งเข้าไปข้างในค้ำทางลูกศรที่เป็นกากะบาด ส่วนตัวนำที่อยู่ภายใต้ขั้วใต้นั้น ให้กระแสพุ่งออกมาข้างนอกค้ำหัวลูกศรที่เป็นจุด เมื่อเป็นเช่นนี้จึงหาทิศทางเคลื่อนที่ของตัวนำทุก ๆ ตัวที่อยู่ภายใต้ขั้วแม่เหล็กทั้ง N และ S ได้ โดยใช้กฎมือซ้ายของเฟลมมิ่ง วิธีหาก็คือ กางมือซ้ายออกโดยให้นิ้วหัวแม่มือ นิ้วชี้และนิ้วกลาง ตั้งฉากซึ่งกันและกัน จากนั้น ให้นิ้วชี้ ชี้ไปตามทิศทางของสนามแม่เหล็ก คือชี้จากขั้ว N ไป S ให้นิ้วกลางชี้ไปตามทิศทางกระแสไหลของกระแส ณ ที่ตัวนำที่ต้องการหาทิศทางที่เคลื่อนที่นั้น ๆ นั่นคือ นิ้วหัวแม่มือจะชี้ทิศทางเคลื่อนที่ของตัวนำนั้น ๆ จะพบว่าแรงที่เกิดขึ้นบนตัวนำทุก ๆ ตัว ภายใต้ขั้วแม่เหล็กเดียวกัน จะมีทิศไปในทางเดียวกัน และจะอยู่ในแนวสัมผัสกับเส้นรอบวงของอาร์เมเจอร์นั้น ๆ และพบว่าแรงที่เกิดขึ้นภายใต้ขั้วแม่เหล็กทุก ๆ ขั้ว ที่สลับกันไปนั้น จะมีทิศทางไปในทางเดียวกันทั้งสิ้น โดยแต่ละแรง จะอยู่ในแนวสัมผัสกับเส้นรอบวงของอาร์เมเจอร์นั้นคือ ภายใต้ขั้วแม่เหล็กแต่ละขั้วก็เกิดแรงลัพธ์ขึ้นแรงหนึ่ง ฉะนั้นเมื่อมีหลายขั้วก็มีหลายแรงและทุก ๆ แรงต่างก็รวมกันเป็นแรงบิดขึ้นมาแรงหนึ่ง ฉะนั้นจึงทำให้มอเตอร์หมุนไปได้จากแรงที่เกิดขึ้น

## 2.4.2 การเปรียบเทียบอากัปกริยาของเครื่องกำเนิดและมอเตอร์

### (Comparison of Generator and Motor Action)

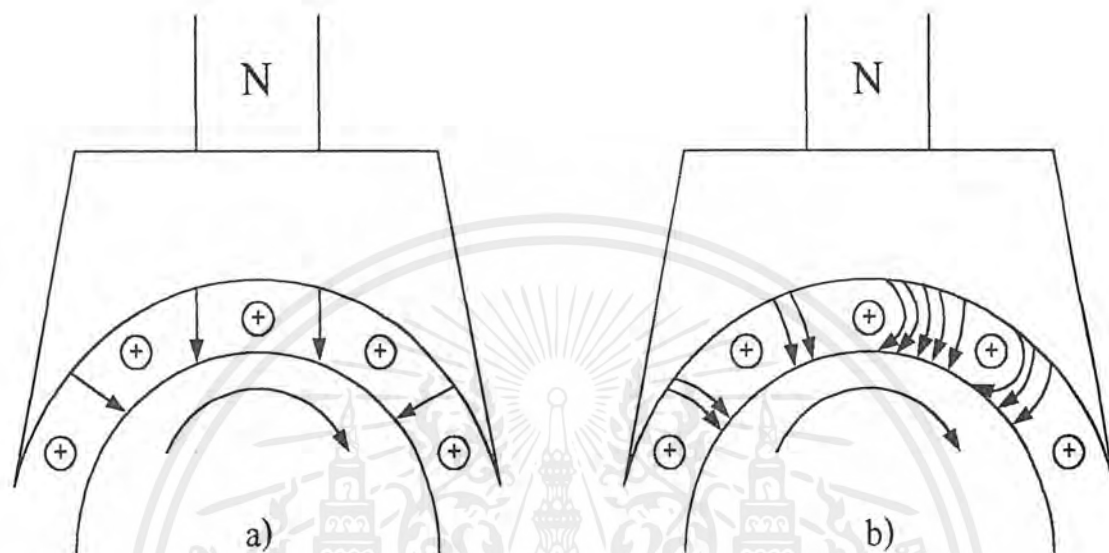
ดังได้กล่าวมาแล้วในตอนต้นว่า เครื่องกลไฟฟ้ากระแสตรงเดียวกันนี้ สามารถใช้เป็นมอเตอร์ไฟฟ้าหรือเครื่องกำเนิดไฟฟ้าก็ได้ นั่นคือ เมื่อทำงานเป็นมอเตอร์นั้น ก็จำเป็นต้องใส่พลังงานไฟฟ้าเข้าไปเพื่อให้ได้พลังงานกลออกมา และเมื่อนำมาใช้เพื่อผลิตกระแสไฟฟ้านั้น ก็จำเป็นต้องใส่พลังงานกลเข้าไปขับให้ตัวหมุนเคลื่อนที่ เพื่อให้ได้พลังงานไฟฟ้าออกมามาตามที่ต้องการ

จากที่กล่าวมาแล้วข้างบนนี้ เราลองมาทำการพิจารณาการทำงานของเครื่องผลิตไฟฟ้าว่า ทำอย่างไรให้เปลี่ยนพลังงานกลมาเป็นพลังงานไฟฟ้าได้ ทั้งนี้เพื่อเปรียบเทียบกับอากัปกริยาของภายในอาร์มาเจอร์ของมอเตอร์ที่เกิดขึ้น

ตามรูป 2.3 (a) แสดงให้เห็นเพียงส่วนหนึ่งของเครื่องกำเนิดไฟฟ้า โดยการหมุนอาร์มาเจอร์ซึ่งมีตัวนำวิ่งผ่านสนามแม่เหล็กหลักที่ขั้ว N หรือขั้ว S ก็ตาม แต่ในที่นี้ได้ทำการพิจารณาเฉพาะตัวนำ A ที่อยู่บนอาร์มาเจอร์วิ่งผ่านขั้ว N ในทิศทางตามเข็มนาฬิกา จากกฎมือขวาของเฟลมมิง ก็พบว่ามีความแรงเคลื่อนไฟฟ้าพุ่งเข้าไป ดังแสดงในรูป 2.3 (a) เมื่อเครื่องผลิตไฟจ่ายกระแสไฟออกไปให้กับภาระไฟฟ้า จะทำให้เกิดสนามแม่เหล็กขึ้นรอบ ๆ ตัวนำทุกตัว แต่ในที่นี้จะพิจารณาเฉพาะตัวนำ A เท่านั้น ดังนั้นเมื่ออาร์มาเจอร์ยังถูกขับให้หมุนไปทางเดิม (ตามเข็มนาฬิกา) โดยพลังงานกลภายนอกอยู่ด้วยกำลังแรงเท่าเดิมอยู่ก็จะทำให้มีความรู้สึกว่าการกำเนิดไฟฟ้าเริ่มหมุนช้าลงไป ที่เป็นเช่นนี้เพราะมีแรงลากแม่เหล็กไฟฟ้า (Magnetic drag) คอยต้านการเคลื่อนที่ของตัวขั้วนั่นเอง อากัปกริยา หรืออาการที่เกิดขึ้นเช่นนี้นั้น สามารถมองให้เห็นชัดหรือเข้าใจได้ง่ายดังรูป 2.3 (b) ซึ่งแสดงให้เห็นว่า เส้นแรงทางด้านขวาของตัวนำ A ที่อยู่ในทิศทางการเคลื่อนที่ของอาร์มาเจอร์นั้น มีจำนวนแม่เหล็กมากกว่าทางด้านซ้าย ทั้งนี้เพราะเส้นแรงแม่เหล็กที่เกิดขึ้นบนตัวนำ A ของทางขวามือนั้น ไปอัดหรือรวมตัวกันกับเส้นแรงแม่เหล็กสำคัญ N ส่วนข้างซ้ายจะไปแยกตัวกับแม่เหล็กหลัก ฉะนั้นทางด้านซ้ายจะมีความหนาแน่นของเส้นแรงแม่เหล็กน้อยกว่า การเกิดเส้นแรงแม่เหล็กบนตัวนำ A นี้ หาได้โดยใช้ Cork's Screw Rule เพื่อให้ง่ายแก่การเข้าใจ จึงอาจจะเปรียบเทียบได้ว่าเส้นแรงแม่เหล็กที่เกิดขึ้นในกริยาอาการต่าง ๆ เหล่านี้ต่างก็เปรียบเสมือนเส้นยาง หรือหนังสติ๊กที่ใช้ขึงนก โดยมี A ซึ่งเป็นตัวนั้นเปรียบเสมือนว่าเป็นลูกหิน ฉะนั้นถ้าที่เส้นแรงเกิดขึ้นทางขวามากเท่าไร ก็จะทำให้เกิดแรงต้านที่อาร์มาเจอร์มากขึ้น หรืออาจพูดว่าเมื่อยิ่งดึงหนังสติ๊กให้ยืดออกไปไกลเท่าไรก็ย่อมจะมีแรงต้านในทิศทางตรงกันข้ามกับที่ยืดหนังสติ๊ก ออกไปมากเท่านั้น ดังนั้นจึงสรุปได้ว่า เมื่อเครื่องผลิตกระแสออกไปมากเท่าไร ก็จะทำให้เครื่องหมุนช้าลงไปทุกที ถ้าหากว่าตัวที่หมุนอาร์มาเจอร์ยังมีแรงจูงเท่าเดิม ฉะนั้นถ้าจะให้เครื่องผลิตไฟฟ้าวิ่งด้วยความเร็วเท่าเดิมได้ จึงจำเป็นต้องเพิ่มพลังงานกลขึ้นเช่นเดียวกันสำหรับตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำอินที่อาร์มาเจอร์ ก็จะมีอากัปกริยาเช่นเดียวกับตัวนำ A ไม่ว่าตัวนำนั้นจะอยู่ภายใต้ขั้ว N และ S นั่นคือแรงต้านจะเกิดขึ้นทุก ๆ ตัวนำรวมกันตาม Vector ก็จะเป็นแรงลัพธ์แรงหนึ่งที่คอยต้านการหมุนของอาร์มาเจอร์ในทิศทางสวนเข็มนาฬิกาตลอดเวลาที่เครื่องกำลังจ่ายโหลด



รูป 2.3 อากัปกริยาของเครื่องกำเนิดไฟฟ้าเมื่อได้รับ Load

สมมุติว่าเครื่องเดียวกันนี้ ถอดเอาตัวต้นกำลัง (Primover) ที่ใช้หมุนเครื่องกำเนิดไฟออกจากนั้นก็ใส่ไฟเข้าไป โดยให้กระแสไหลผ่านเข้าไปภายใต้ขั้ว N ดังรูปที่ 2.4

เมื่อมีกระแสไหลผ่านตัวนำในอาร์เมเจอร์ ซึ่งอยู่ภายใต้สนามแม่เหล็กขั้ว N ก็ย่อมจะทำให้เกิดเส้นแรงแม่เหล็กขึ้นในตัวนำต่าง ๆ บนอาร์เมเจอร์นั้น ๆ นั่นก็คือ เกิดแรงบิด (Torque) ขึ้นบนอาร์เมเจอร์ในทิศทางสวนเข็มนาฬิกา อากัปกริยานี้ เกิดขึ้นเช่นเดียวกับอากัปกริยาที่เกิดขึ้นในเครื่องผลิตไฟเช่นกัน แรงบิดที่เกิดขึ้นนี้ หากทิศทางการเคลื่อนที่ได้โดยใช้กฎมือซ้ายของเฟลมมิ่งนั่นคือ ขณะที่เครื่องได้ทำหน้าที่เป็นมอเตอร์ไปแล้วจะสังเกตเห็นว่า เมื่อกระแสที่ไหลในตัวนำยังอยู่ที่ทิศทางเดิม จะทำให้การเคลื่อนที่ไปในทิศทางสวนเข็มนาฬิกาเช่นเดิมซึ่งตรงกันข้ามกับทิศทางของต้นกำลัง (Primover) ที่ใช้ขับเครื่องกำเนิดไฟ เมื่อเครื่องทำหน้าที่เป็นตัวผลิตไฟ (Generator) จากที่กล่าวมาแล้วข้างต้นพบว่า ในการเปลี่ยนรูปพลังงานนั้น ย่อมมีแรงต้านเกิดขึ้น นั่นคือสำหรับเครื่องผลิตไฟนั้นจะเกิดแรงต้านทางกล หรือแรงแม่เหล็กลาก (Magnetic drag) ขึ้นในทิศทางตรงกันข้ามกับแรงที่ใส่เข้าไป ฉะนั้นจึงต้องใส่กำลังเข้าไปให้มากกว่าแรงลากที่เกิดขึ้นนี้ จึงจะได้กำลังไฟออกมาจ่ายให้กับภาระไฟฟ้าตามที่ต้องการได้นั่นคือ เครื่องผลิตไฟยังคง

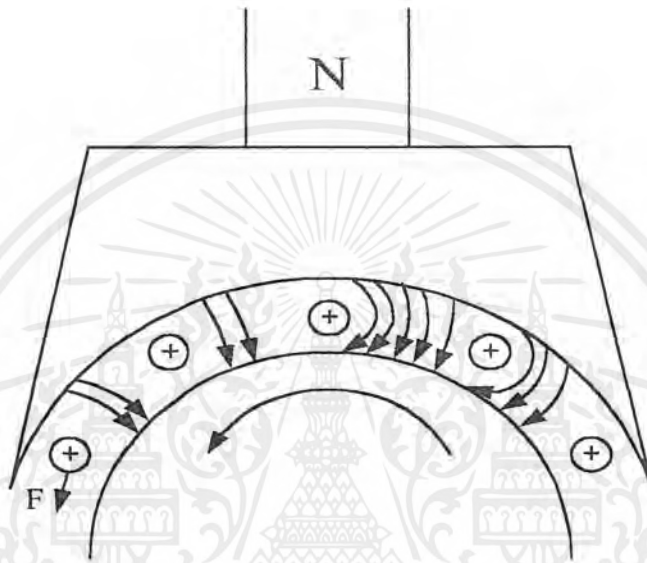
ความเร็วที่ค่า ๆ หนึ่ง บางทีอาจจะให้หมุนที่ความเร็วเดิมหรือน้อยกว่าก็ได้ ทั้งนี้ขึ้นอยู่กับว่าจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญญาติให้ไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้ Voltage ที่ขั้วที่จ่ายภาระไฟฟ้า (Load) นั้นมีค่าคงที่หรือใกล้เคียงตอนไม่มีภาระไฟฟ้า ส่วนมอเตอร์นั้นก็ย่อมมีตัวต้านเกิดขึ้นเช่นกัน คือ แรงเคลื่อนไฟสวน (Back e.m.f.)

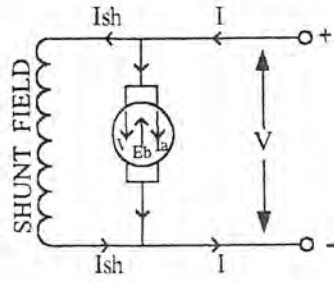
ทั้งมอเตอร์และเครื่องกำเนิดไฟฟ้า ต่างก็เกิดอาการปฏิกิริยาของมอเตอร์ และการผลิตไฟที่เกิดขึ้นในเครื่องเดียวกัน นั่นคือ เมื่อเป็นมอเตอร์ก็เกิดอาการปฏิกิริยาของมอเตอร์ (Motor action) ขึ้นก่อน แล้วจึงค่อยเกิดอาการของเครื่องกำเนิดไฟ (Generator action)



รูป 2.4 อาการปฏิกิริยาของมอเตอร์

จากที่กล่าวมาข้างบนนี้จะพบว่า เมื่อใส่กระแสเข้าไปในตัวนำย่อมทำให้เกิดการเคลื่อนที่ นั่นคือ อาร์เมเจอร์ของมอเตอร์หมุน เมื่อตัวนำวิ่งตัดผ่านสนามแม่เหล็กย่อมทำให้เกิดแรงเคลื่อนไฟฟ้าขึ้นในตัวนำตามหลักการของเครื่องกำเนิดไฟฟ้า ซึ่งหาได้โดยการใช้มือขวาของเฟลมมิ่ง แรงเคลื่อนไฟที่เกิดขึ้นนี้ จะมีทิศทางสวนหรือตรงกันข้ามกับแรงเคลื่อนที่ใส่เข้าไปให้กับมอเตอร์ ดังรูปที่ 2.5 ฉะนั้นจะเรียกแรงเคลื่อนไฟฟ้าที่เกิดขึ้นนี้ว่า “แรงเคลื่อนสวน = Back e.m.f.” แรงเคลื่อนสวน  $E_b$  นี้ขึ้นอยู่กับจำนวนเส้นแรงแม่เหล็กต่อขั้ว จำนวนตัวนำทั้งหมดที่ต่ออนุกรมกัน ความเร็วที่มอเตอร์หมุนไป จำนวนขั้วแม่เหล็กที่อยู่ภายใน และจำนวนแฉก (Path) ที่ขนานกันแรงเคลื่อนไฟนี้มีแรงเคลื่อนเช่นเดียวกับแรงเคลื่อนที่เกิดจากการหมุน (Motional e.m.f.) นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

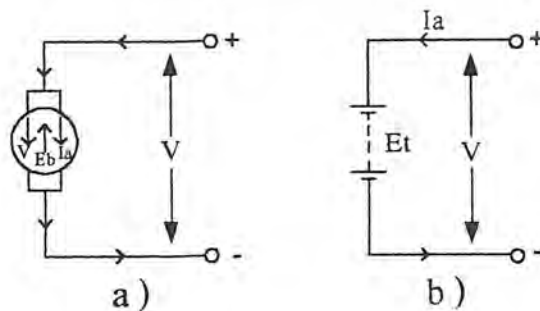


รูป 2.5 แสดงทิศทางแรงเคลื่อนไฟ และกระแสที่เกิดขึ้นในมอเตอร์ซึ่งไหลไปในทางเดียวกันแต่ตรงข้ามกับแรงเคลื่อนไฟสวน

สรุป จะเห็นได้ว่าแรงต้านทานที่เกิดขึ้นนั้น ๆ จะเป็นแรงชนิดเดียวกันกับแรงที่ใส่เข้าไป นั่นคือ ถ้าเป็นเครื่องกำเนิดก็เกิดแรงต้านทานกลเกิดขึ้น เพราะใส่แรงกลเข้าไป และถ้าเป็นมอเตอร์ก็เกิดแรงต้านทานไฟฟ้าขึ้นเพราะใส่แรงเคลื่อนเข้าไป

2.4.3 ความสำคัญของแรงเคลื่อนไฟสวน (Significance of the Back e.m.f.)

เมื่ออาร์เมเจอร์เริ่มหมุน จะทำให้ตัวนำที่อยู่ในอาร์มาเจอร์ตัดเส้นแรงแม่เหล็กที่มาจากสนามแม่เหล็กหลัก หรือสนามแม่เหล็กพัทธ์ที่อยู่ในมอเตอร์นั้น นั่นคือ เกิดการเหนี่ยวนำแรงเคลื่อนไฟบนตัวนำภายในอาร์เมเจอร์ ซึ่งเป็นไปตามกฎการเหนี่ยวนำแรงเคลื่อนแม่เหล็กไฟฟ้า แรงเคลื่อนไฟฟ้าที่เกิดขึ้นนี้หาทิศทางของแรงเคลื่อนไฟฟ้าได้ โดยใช้กฎมือขวาของเฟมมิงและพบว่า มีทิศทางสวนกับแรงเคลื่อนไฟที่ใส่เข้าไปให้กับมอเตอร์ คั้งนั้น จึงเรียกแรงเคลื่อนที่เกิดขึ้นนี้ว่าแรงเคลื่อนสวน (Back e.m.f. =  $E_b$ ) ดังรูปที่ 2.6 (a) จากที่เกิด  $E_b$  ขึ้นนี้ จึงเขียนวงจรสมมูลย์ของมอเตอร์ได้ดังรูปที่ 2.6 (b) ซึ่งเหมือนกับว่า มีแบตเตอรี่  $E_b$  ต่อคร่อมอยู่ระหว่างแรงเคลื่อนไฟ  $V$  ที่จ่ายให้กับมอเตอร์ นั่นคือ ในการที่จะให้  $I_a$  ไหลได้ในอาร์มาเจอรันั้น จะต้องมีกำลังไฟจำนวนหนึ่งที่เราจะหาการสวนนี้ กำลังไฟที่เราจะหาการสวนนี้ได้ คือ  $E_b I_a$



รูป 2.6 (a) วงจรของมอเตอร์ไฟตรง (b) วงจรของมอเตอร์ที่เขียนแทนแรงเคลื่อนไฟสวนได้ด้วย

แบตเตอรี่  $E_b$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่เซลหรือแบตเตอรี่ กำลังที่เอาชนะการสวนนี้ก็คือ กำลังที่เปลี่ยนไปเป็นกำลังทางเคมี แต่ในมอเตอร์นั้นกำลังที่เอาชนะการสวนนี้ ก็คือ กำลังที่เปลี่ยนไปเป็นกำลังกลนั่นเอง

$$\begin{aligned} \text{นั่นคือ กระแส } I_a \text{ จะมีค่า} &= \frac{\text{แรงเคลื่อนไฟที่ได้ในวงจร}}{\text{ค.ต.ท.ในวงจรของอาร์มาเจอร์}} \\ &= \frac{\text{net voltage}}{\text{armature circuit resistance}} \end{aligned}$$

$$= \frac{V - E_b}{R_a}$$

จากแรงเคลื่อนที่หมุนของตัวนำในสนามแม่เหล็กมีค่า  $E = \frac{\phi Z P S}{60 a}$

ดังนั้นจะได้

$$E_b = \frac{\phi Z P S}{60 a} \quad (2.2)$$

จะพบว่า  $E_b$  ขึ้นอยู่กับตัวประกอบต่างๆ เช่น  $\phi, Z, P$  และ  $a$  แต่เนื่องจากว่า  $Z, P/a$  เป็นค่าคงที่ในการพันอาร์มาเจอร์ของแต่ละเครื่องที่แตกต่างกันไปดังนั้นจึงได้

$$E_b = K_e \phi S \quad (2.3)$$

$K_e$  = ค่าคงที่ทางไฟฟ้า =  $PZ / 60a$

$P$  = Pole

$Z$  = จำนวนตัวนำทั้งหมดของอาร์มาเจอร์

$a$  = จำนวนแฉกที่ขนานกันในอาร์มาเจอร์

$S$  = จำนวนรอบต่ออนาที

$\phi$  = เส้นแรงแม่เหล็กต่อหนึ่งขั้ว (Weber)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 2.4.4 TORQUE (แรงบิด)

แรงบิดก็คือ การหมุนหรือเกิดการบิดของโมเมนต์ของแรง ๆ หนึ่งที่กระทำรอบแกนอันหนึ่ง วัดได้จากผลคูณของแรงกับรัศมี ณ ที่แรงไปกระทำ

พิจารณาที่รัศมี  $r$  มีแรง  $F$  นิวตันมากระทำบนวงดัดนี้ จึงทำให้เกิดการหมุนด้วยความเร็ว  $S$  รอบ/วินาที

$$\text{แรงบิด} = F \times r \text{ นิวตัน - เมตร}$$

$$\text{งานที่ได้ใน 1 รอบจากแรงนี้} = \text{แรง} \times \text{ระยะทาง}$$

$$= F \times 2 \pi r \text{ จูลล์}$$

$$\text{งานที่ทำต่อวินาที} \quad W = F \times 2 \pi r \times S$$

$$= F \times r \times 2 \pi S$$

$$\text{แต่} \quad 2 \pi S = \text{มุมเป็นเรเดียนต่อวินาที}$$

$$F \times r = \text{แรงบิด } T$$

$$\text{งานที่ทำต่อวินาที} \quad = T \times \theta \text{ จูลล์}$$

$$\text{กำลังที่เกิดขึ้น} \quad = T \times \theta \text{ วัตต์}$$

#### 2.4.5 แรงบิดที่เกิดขึ้นในอาร์มาเจอร์ (Armature Torque of a Motor)

ให้  $T_m$  เป็นแรงบิดที่เกิดขึ้นจากอาร์มาเจอร์ เมื่อมอเตอร์วิ่งด้วยความเร็ว  $S$  รอบต่อวินาที

$$T_m = \frac{0.159 E b l a}{S} \quad \text{N-m} \quad (2.4)$$

$$= \frac{0.162 E b l a}{S} \quad \text{Kg-m} \quad (2.5)$$

เมื่อ  $S$  เป็นความเร็ว (r.p.s)

#### 2.4.6 แรงบิดที่แกน (SHAFT TORQUE = $T_{sh}$ )

แรงบิดที่เกิดขึ้นในอาร์มาเจอรันั้น ไม่ได้นำไปใช้งานทีเดียว ทั้งนี้ต้องจ่ายให้กับการสูญเสียอันเนื่องมาจาก Iron และ Friction ใน Motor

$$T_{sh} = \frac{735.5 B.H.P.}{2 \pi S} \quad \text{Tsh metric} \quad (2.6)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

B.H.P = กำลังมาที่่ได้จากแกนหมุนของมอเตอร์ซึ่งเป็นกำลังมาที่่ได้จากการ Brake Motor

## 2.4.7 SPEED REGULATION

Speed Regulation เป็นค่าผลต่างของความเร็วระหว่างไม่มีโหลด (Load) กับเมื่อได้รับ โหลด (Load) เต็มที่

$$\% = \frac{(N.L.Speed - F.L.Speed) \times 100}{F.L.Speed} \quad (2.7)$$

## 2.5 คุณลักษณะประจำของมอเตอร์ (Motor Characteristics)

### 2.5.1 คุณลักษณะประจำของมอเตอร์แบบขนาน (Characteristics of Shunt Motor)

1. การ Start ของ Shunt Motor ไม่สามารถ Start ตอนมีโหลด (Load) ที่หนัก ได้ เพราะการกระทำเช่นนี้ยอมทำให้มอเตอร์ต้องดึงกระแสจำนวนมาก ไปใช้ด้วย
2. เนื่องจากมอเตอร์แบบขนาน (Shunt Motor) นี้ไม่มีการเปลี่ยนหรือลดความเร็ว ไปมากนัก จากตอนมีภาระถึงตอนที่มีภาระเต็มที่ ดังนั้นจึงต่อเข้าหรือตัดออกได้จากภาระโดยทันที ทันใดได้ โดยไม่ต้องเกรงว่าจะมีความเร็วสูงมากเกินไปจนเป็นอันตราย อย่างไรก็ตาม มอเตอร์แบบนี้ ให้แรงบิดตอนเริ่มแรกต่ำ (Low Starting Torque) ฉะนั้นจึงไม่นิยมใช้กับโหลด (Load) ที่ ต้องการแรงบิดเริ่มแรกสูง แต่เนื่องจากว่าเป็นมอเตอร์ที่มีความเร็วค่อนข้างคงที่ ดังนั้นจึงเหมาะที่จะนำไปใช้ขับแกนของเครื่องกลึงและเครื่องจักรชนิดต่าง ๆ ที่ต้องการความเร็วคงที่โดยประมาณ

### 2.5.2 คุณลักษณะประจำของมอเตอร์ชนิดอนุกรม (Characteristics of Series Motor)

1. มอเตอร์แบบอนุกรมนี้เหมาะสมสำหรับที่จะจุดโหลด (Load) หนัก ๆ ได้ ฉะนั้นจึงจะเหมาะสมกับโหลด (Load) หนักขณะเริ่มแรกตัวอย่างเช่น ปั่นจัน (Hoist) รถไฟฟ้า (Electric Trains)
2. เมื่อโหลดมาก ๆ จะทำให้มอเตอร์ใช้กระแส  $I_a$  มาก ๆ ด้วย ดังนั้นความเร็วของมอเตอร์จะลดลงอย่างรวดเร็ว แต่ถ้าหากว่ามอเตอร์มีโหลด (Load) น้อย ๆ จะทำให้ความเร็วมีค่าสูงขึ้นจะเป็นอันตรายต่อมอเตอร์ได้ นั่นคือ มอเตอร์อนุกรมต้องไม่ Start หรือหมุนในขณะที่ไม่มีโหลด (Load) มิฉะนั้นจะทำให้มีความเร็วสูงมากเกินไปจนเป็นอันตรายกับมอเตอร์ได้ อันเนื่องมาจากแรงหนีศูนย์กลางที่เกิดขึ้น

### 2.5.3 คุณลักษณะประจำตัวของมอเตอร์แบบผสม

#### (Characteristics of Compound Motor)

ลักษณะประจำตัวของมอเตอร์ชนิดนี้ก็คือ การนำเอาคุณลักษณะของมอเตอร์ทั้ง 2 แบบมารวมกันทั้งนี้เพื่อให้เหมาะสมกับการใช้งาน มอเตอร์ผสมผสานสามารถแบ่งออกได้เป็น 2 แบบ โดยแบ่งตามลักษณะการต่อ Series Field เพิ่มเข้าไปให้กับ Shunt Motor

### 2.5.4 มอเตอร์แบบผสมชนิดสะสมหรือเสริมสนามแม่เหล็กให้มากขึ้น

#### (Cumulative Compound Motor)

มอเตอร์แบบนี้จะเกิดแรงบิดที่อาร์มาเจอร์สูงในขณะที่ได้รับโหลดทันทีทันใด และยังมีความเร็วที่ต่ำ ๆ หนึ่ง ในขณะที่มีภาระหรือโหลดนั้นก็คือแม้ว่าจะไม่มีภาระ ก็ไม่ทำให้มอเตอร์มีความเร็วสูงเกินไปจนเกิดอันตรายได้ ในการนำมอเตอร์นี้ไปใช้งานกับการขับโหลดที่หนัก ๆ เครื่องโม่หิน เครื่องตัดเหล็ก หรือเครื่องอัดยา ลิฟท์ เป็นต้น

### 2.5.5 มอเตอร์แบบชนิดที่มีสนามแตกต่าง (Differential – Compound)

เมื่อมอเตอร์ได้รับโหลดมากขึ้น แทนที่จะมีความเร็วลดลงไปก็กลายเป็นว่ามีความเร็วเพิ่มขึ้นมาเท่าเดิมตามที่โหลดเพิ่มขึ้น ดังนั้นมอเตอร์ชนิดนี้จึงต้องการนำไปใช้งานกับโหลดที่ต้องการให้มีค่าความเร็วคงที่มาก ๆ ในขณะที่โหลดเพิ่มมากขึ้น แต่มอเตอร์ชนิดนี้มีข้อเสียตรงที่ว่า เมื่อได้รับโหลดมากเกินไป ย่อมทำให้สนามแม่เหล็กอ่อนลงไปมาก นั่นคือความเร็วของมอเตอร์ ก็จะมีแนวโน้มเพิ่มสูงขึ้น ๆ อย่างไม่เสถียรภาพ (Instability) และทำให้มอเตอร์วิ่งเกินเลยไป ซึ่งทำความเสียหายให้กับมอเตอร์ได้

## 2.6 การควบคุมทิศทางการหมุนของมอเตอร์

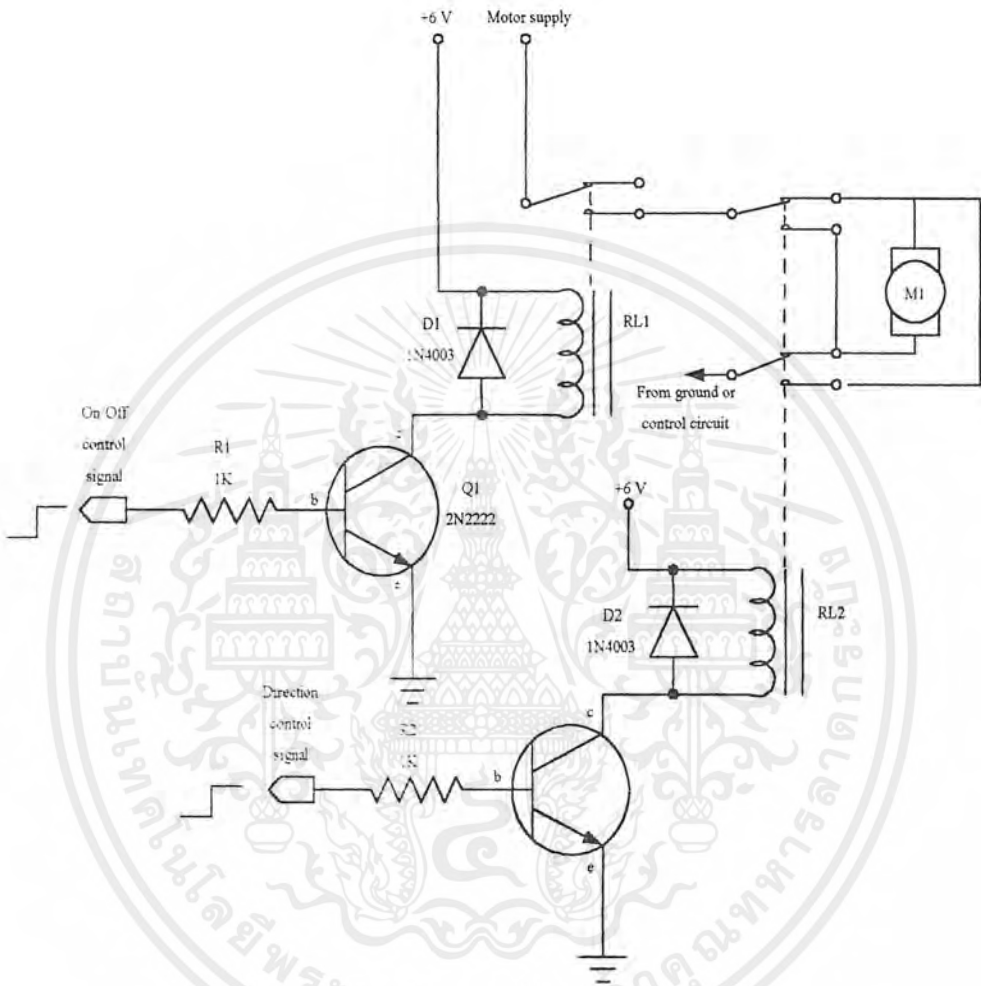
เรามีวิธีการที่จะบังคับทิศทางการหมุนของตัวมอเตอร์ให้หมุนในทิศทางทวนเข็มนาฬิกาหรือตามเข็มนาฬิกาได้หลายวิธี เช่น การใช้สวิทช์แบบดับเบิ้ลโพล - ดับเบิ้ลโธรล (Double – Pole Double – Throw : DPDT) ซึ่งเราสามารถที่จะควบคุมให้มอเตอร์หมุนหรือหยุดหมุนหรือกลับทิศทางการหมุนของตัวมอเตอร์ได้ ซึ่งเราจะไม่กล่าวในที่นี้เพราะไม่สามารถควบคุมด้วยไมโครโปรเซสเซอร์ได้

สำหรับวิธีการอื่น ๆ ก็เช่นอาศัยการใช้รีเลย์เป็นสวิทช์เปิดปิดไฟที่จะป้อนให้กับตัวมอเตอร์หรือการใช้งานทรานซิสเตอร์ในลักษณะของสวิทช์ซึ่งอาจใช้แบบชนิดไปโพลาร์หรือแบบเฟทก็ได้ดังมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 การใช้รีเลย์ควบคุมมอเตอร์

รูปที่ 2.7 เป็นวงจรควบคุมมอเตอร์ที่ใช้รีเลย์เป็นสวิทช์ ซึ่งมีหลักการดังนี้



รูปที่ 2.7 แสดงวงจรควบคุมมอเตอร์ที่ใช้รีเลย์

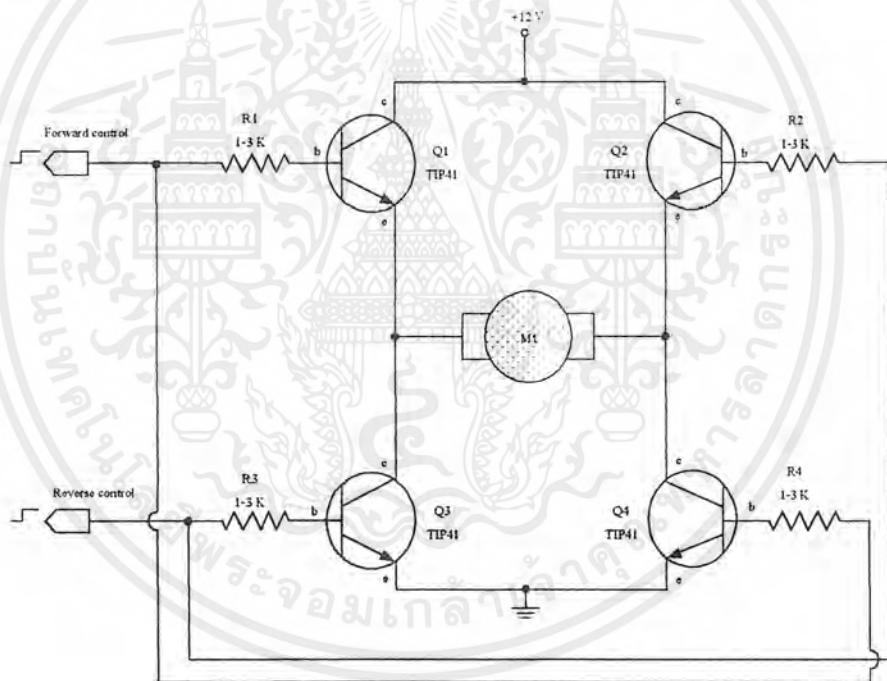
โดยรีเลย์ RL1 เป็นรีเลย์แบบที่เรียกว่าซิงเกิ้ลโพล - ดับเบิลโธรว (Single - Pole Double - Throw : SPDT) คือมีขั้วเดียวใช้เป็นรีเลย์สำหรับควบคุมการเปิด - ปิดของตัวมอเตอร์ โดยการป้อนสัญญาณควบคุมที่ขาเบสของทรานซิสเตอร์ Q1 ที่ทำงานเป็นสวิทช์ ส่วน RL2 เป็นรีเลย์แบบดับเบิลโพล - ดับเบิลโธรวที่มี 2 ขั้ว ทำให้สามารถกลับทิศทางหมุนของกระแสที่ป้อนให้กับตัวมอเตอร์ได้ สำหรับสัญญาณที่ป้อนให้ขาเบสของทรานซิสเตอร์ทั้ง 2 ตัว นั้นสามารถใช้เอาท์พุทจากวงจรดิจิทัลได้โดยตรงเลยและยังสามารถแยกไฟเลี้ยงของวงจรอิเล็กทรอนิกส์และไฟเลี้ยงที่ป้อนให้ตัวมอเตอร์ออกจากกันได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.6.2 การใช้ทรานซิสเตอร์ในการควบคุมมอเตอร์

การใช้ทรานซิสเตอร์ในงานควบคุมมอเตอร์นั้นเป็นการนำทรานซิสเตอร์มาใช้งานในลักษณะของสวิตช์ ถ้าเราต้องการใช้ทรานซิสเตอร์แค่ 2 ตัว ในการควบคุมเราจะต้องต่อทรานซิสเตอร์ในลักษณะคอมพลิเมนต์ารี (Complementary) และต้องใช้ไฟเลี้ยงทั้งซีกบวกและซีกลบ ซึ่งไม่สะดวกในการใช้งานดังนั้นเราจึงต้องวงจรในลักษณะดังรูปที่ 2.8

จากรูปที่ 2.8 ในขณะใดขณะหนึ่งเราจะให้มีทรานซิสเตอร์เพียง 2 ตัวเท่านั้นที่นำกระแสคือ Q1 กับ Q4 หรือ Q2 กับ Q3 (คือการควบคุมเดินหน้าหรือถอยหลัง จะต้องไม่เป็น '1' ในเวลาเดียว) ซึ่งการที่เราให้ทรานซิสเตอร์นำกระแสเป็นคู่นี้จะทำให้เราสามารถบังคับทิศทางกระแสที่ป้อนให้กับตัวมอเตอร์ได้ ซึ่งก็คือการบังคับทิศทางการหมุนของตัวมอเตอร์นั่นเอง



รูปที่ 2.8 แสดงการต่อทรานซิสเตอร์ 4 ตัว เพื่อควบคุมทิศทางการหมุนของมอเตอร์

ซึ่งวงจรในลักษณะดังรูปที่ 2.8 นั้นเราสามารถที่จะใช้แหล่งจ่ายไฟขั้วเดียวและสามารถแยกแหล่งจ่ายไฟจากวงจรอิเล็กทรอนิกส์ได้ ทำให้เราสามารถใช้อาชีพจากวงจรดิจิทัลมาใช้ในการเปิด (Turn On) หรือปิด (Turn Off) ทรานซิสเตอร์ได้โดยตรง

### 2.6.3 การใช้พาเวอริมอสเฟต (Power MOSFET) ในการคุมมอเตอร์

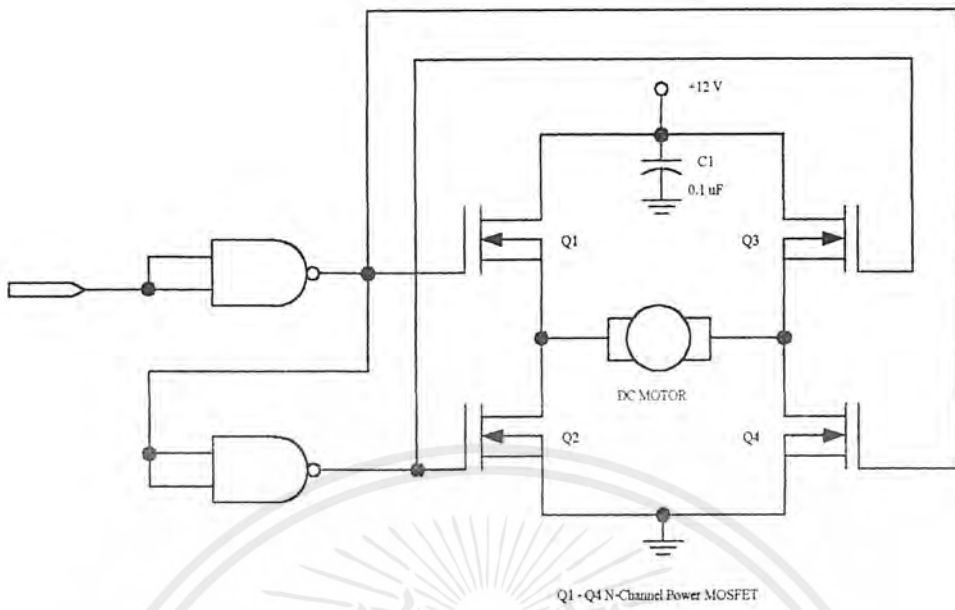
สำหรับวงจรใช้งานโดยทั่วไปของวงจรควบคุมมอเตอร์ที่ใช้มอสเฟตนั้น แสดงในรูปที่ 2.9 ซึ่งต่อมอสเฟตเหมือนกับทรานซิสเตอร์ในรูปที่ 2.8 แต่ได้มีการเพิ่มอินเวอร์เตอร์ซึ่งทำให้สามารถลดอินพุตที่ต้องใช้ในการควบคุมทิศทางการหมุนของมอเตอร์ได้ แต่เราต้องเพิ่มเติมมอสเฟตที่จะใช้ในการควบคุมให้มอเตอร์หมุนหรือหยุดหมุนซึ่งทำให้เราสามารถที่จะควบคุมความเร็วของมอเตอร์ได้

## 2.7 การควบคุมความเร็วของมอเตอร์

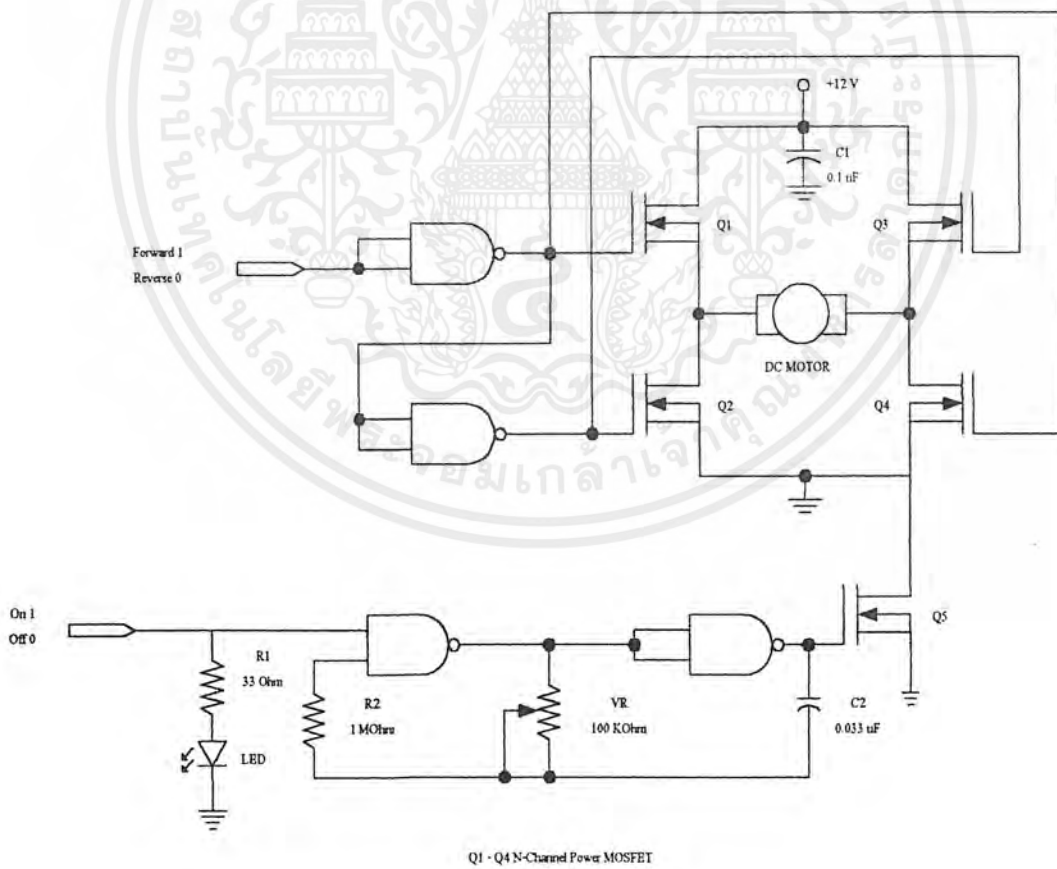
ได้มีความพยายามที่จะปรับปรุงความเร็วของมอเตอร์ เพราะว่าการที่เราจะปรับความเร็วในการเคลื่อนที่ของหุ่นยนต์โดยการปรับที่ขนาดของล้อหรืออัตราทดของเฟืองทดนั้นเป็นไปได้ยาก ดังนั้นเราจึงใช้การปรับความเร็วในการเคลื่อนที่ของหุ่นยนต์โดยการปรับความเร็วรอบของมอเตอร์แทน ซึ่งจากการที่มอเตอร์ไฟฟ้ากระแสตรงถ้าเราป้อนแรงดันน้อยลงก็จะทำให้ความเร็วในการหมุนลดลงด้วย ดังนั้นเราก็จะสามารถปรับความเร็วในการหมุนของมอเตอร์ได้โดยการเปลี่ยนค่าแรงดันที่ป้อนให้กับมอเตอร์นั่นเอง

วิธีการหนึ่งที่เราใช้ในการปรับค่าแรงดันที่จ่ายให้กับมอเตอร์ คือเราใช้วงจรดังรูปที่ 2.10 จากรูปมอสเฟต Q5 จะทำหน้าที่เป็นสวิตช์ควบคุมการจ่ายกระแสให้กับมอเตอร์ ถ้าเราป้อนแรงดันที่ขาเกตของ Q5 เป็นความถี่ค่าหนึ่ง (อาจจะให้เป็นพัลส์ก็ได้) ก็จะทำให้ Q5 ทำงานเป็นช่วง ๆ ซึ่งจะทำให้มีแรงดันตกคร่อมมอเตอร์เป็นช่วง ๆ มีค่าความถี่เดียวกันกับ  $V_g$  ซึ่งในตอนนี้จะทำให้มีค่าแรงดันเฉลี่ยที่ตกคร่อมมอเตอร์น้อยกว่าการที่เราป้อนแรงดันเป็นไฟกระแสตรง ค่าเฉลี่ยของแรงดันนี้สามารถหาได้จากสูตร แรงดันสูงสุด\*ค่าคิวตี้ไซเคิล (Duty Cycle) ดังนั้นถ้าเราสามารถปรับค่าคิวตี้ไซเคิลได้ก็เท่ากับว่าเราสามารถปรับความเร็วของมอเตอร์ได้

จากรูปที่ 2.10 เราใช้สัญญาณควบคุมการหมุนจากส่วนควบคุมมาป้อนให้วงจรกำเนิดความถี่เพื่อป้อนให้กับขาเกตของ Q5 ต่อไป เราอาจจะคิดแปลงวงจรได้โดยใช้วงจรกำเนิดสัญญาณที่สามารถปรับค่าคิวตี้ไซเคิลได้ หรือไม่ก็เอาสัญญาณจากไมโครโปรเซสเซอร์มาป้อนให้กับขาเกตโดยตรงก็ได้ แต่สัญญาณที่นำมาป้อนนี้ต้องมีลักษณะเป็นพัลส์ซึ่งสามารถทำได้โดยใช้การเขียนโปรแกรมให้ส่งสัญญาณเป็นพัลส์ออกมาตามความถี่และคิวตี้ไซเคิลตามต้องการ



รูปที่ 2.9 แสดงวงจรควบคุมมอเตอร์ควบคุมด้วยเฟาเวอร์มอสเฟต



รูปที่ 2.10 แสดงวงจรตัวอย่างที่ใช้ในการควบคุมทิศทาง การหยุดหมุนและความเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.8 การเข้ารหัสสำหรับเพลาหมุน

การเข้ารหัสสำหรับเพลาหมุน (Rotary Encoders) คือการเปลี่ยนลักษณะการหมุนเชิงกล เช่น การหมุนของแกนมอเตอร์หรือการหมุนเหวี่ยงของเครื่องจักรส่วนใดส่วนหนึ่ง มาเข้ารหัสเป็นสัญญาณดิจิทัล เพื่อนำไปเข้าสู่ระบบควบคุม สำหรับตัวอย่างการใช้งานที่เห็นได้ชัดได้แก่การนำมาร่างเป็นแทรีกบอลและเมาส์ที่ใช้ในคอมพิวเตอร์, ระบบจุดระเบิดอิเล็กทรอนิกส์, หุ่นยนต์ในงานอุตสาหกรรม, อุปกรณ์ที่ใช้ในการออกกำลังกาย เป็นต้น ดังนั้นในการออกแบบควบคุมโดยใช้วงจรอิเล็กทรอนิกส์หรือคอมพิวเตอร์เทคนิคการเข้ารหัสจึงมีความสำคัญมากส่วนหนึ่ง

### 2.8.1 พื้นฐานของการเข้ารหัส

ตัวเข้ารหัสใช้หลักการของการเปิดปิดสวิตช์ซึ่งเป็นจังหวะที่สอดคล้องและสัมพันธ์กับการหมุนของเพลาหรือแกนหมุน จะเห็นว่ามีการเข้ารหัส (Encoder Disc) ซึ่งแบ่งครึ่งระหว่างสี่ขาและสี่ขา ต่ออยู่กับเพลาและมีอุปกรณ์ตรวจจับ (Sensor) ติดตั้งอยู่ด้านบนทำหน้าที่ตรวจจับเฉพาะสี่ขาเท่านั้น ดังนั้นเมื่อแถบสี่ขาเคลื่อนที่ผ่านอุปกรณ์ตรวจจับเมื่อใด มันก็จะส่งสัญญาณเอาต์พุตออกมาสัญญาณเอาต์พุตที่ได้จึงมีลักษณะเป็นขบวนพัลส์ ซึ่งความถี่ที่ผลิตจะขึ้นอยู่กับความเร็วในการหมุนของแกนหรือเป็นส่วนกลับของคาบเวลาดังสมการที่ 1 และสมการที่ 2 ตามลำดับ

$$\text{ความถี่ (เฮิรตซ์)} = \text{จำนวนพัลส์ (ไซเคิล)} / \text{ช่วงระยะเวลาหนึ่ง (วินาที)} \quad \dots\dots(1)$$

$$\text{ความถี่ (เฮิรตซ์)} = 1 / \text{คาบเวลา (วินาที)} \quad \dots\dots(2)$$

การหาความเร็วของเพลาหรือความถี่สามารถใช้คอมพิวเตอร์ช่วยในการคำนวณได้ ซึ่งสามารถกำหนดได้โดยคอมพิวเตอร์นับจำนวนพัลส์ที่เกิดขึ้น ในช่วงระยะเวลาคงที่ที่กำหนดคขึ้นแล้ว แทนค่าลงในสมการที่ 1 ไซเคิล แล้วแทนค่าลงในสมการที่ 2 ซึ่งทั้งสองวิธีให้ผลลัพธ์ที่เท่ากัน

เพื่อให้เกิดความเข้าใจมากขึ้นจึงขอยกตัวอย่างสัก 1 ข้อ เช่น กำหนดให้ตัวเข้ารหัสออกแบบให้เพลาหมุน 1 รอบจะได้เอาต์พุตออกมา 1 พัลส์เมื่อนำมาต่อใช้งานปรากฏว่าใน 1 วินาที นับพัลส์ได้ทั้งหมด 10 พัลส์ หรือกล่าวได้ว่าเพลากำลังหมุนด้วยความเร็ว 10 รอบต่อวินาที (Rps) หรือ 600 รอบต่อนาที (Rpm) และรูปคลื่นพัลส์ที่ได้จะมีความถี่เป็น 10 เฮิรตซ์ มีคาบเวลาของพัลส์เป็น 0.1 วินาที

จากตัวอย่างเห็นได้ว่าความเร็วรอบ (Rps) และความถี่ของพัลส์ที่ได้มีค่าเท่ากัน แต่ในทางปฏิบัติตัวเข้ารหัสอาจถูกออกแบบให้ผลิตพัลส์จำนวนมากต่อการหมุนทางกล 1 รอบ ซึ่งให้ผลดีในการควบคุมและป้องกันสัญญาณรบกวน ดังนั้นการคำนวณหาความถี่และความเร็วรอบจึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แตกต่างกัน ซึ่งค่าความถี่ของพัลส์ยังสามารถคำนวณได้ตามสมการที่ 1 และ สมการที่ 2 เช่นเดิม แต่ความเร็วรอบต้องคำนวณ โดยใช้สมการที่ 3 แทน

$$\text{ความเร็วรอบ (รอบต่อวินาที)} = \text{ความถี่ (เฮิรตซ์)} / N \quad \dots\dots(3)$$

N คือ จำนวนพัลส์ที่เกิดขึ้นต่อการหมุนทางกลครบ 1 รอบ

นอกจากการหาความเร็วของเพลลาได้แล้ว การหาทิศทางการหมุนของเพลลาก็ทำได้เช่นกัน โดยใช้เทคนิคการติดตั้งตัวตรวจจับ และการวิเคราะห์รูปคลื่นที่ได้มาเป็นข้อกำหนด การออกแบบทำได้โดยการเพิ่มตัวตรวจจับอีกตัวหนึ่งติดตั้งให้ทำมุม 90 องศา กับตัวตรวจจับตัวแรก ซึ่งทำให้ได้สัญญาณพัลส์อีกชุดหนึ่งที่มีความต่างเฟสกัน 90 องศา

ดังนั้นการติดตั้งเข้ารหัสแบบมีตัวตรวจจับ 2 ชุดนี้ จึงสามารถวัดได้ทั้งค่าความเร็วรอบ และทิศทางการหมุน โดยนำสัญญาณจากตัวตรวจจับตัวใดตัวหนึ่งไปคำนวณหาค่าความเร็วรอบ และนำสัญญาณจากตัวตรวจจับทั้ง 2 ตัว มาวิเคราะห์หาทิศทางการหมุนได้ เมื่อทราบทฤษฎีเบื้องต้นกันแล้ว ต่อไปเราจะมาศึกษาถึงระบบจริงกันบ้างโดยเริ่มจากอุปกรณ์ตรวจจับชนิดต่าง ๆ ที่มีใช้งาน

## 2.9 อุปกรณ์ตรวจจับ

อุปกรณ์ตรวจจับ (Sensor) ที่ใช้ในการเข้ารหัสของเพลลาหมุนนั้นสามารถแบ่งตามหลักการทำงานพื้นฐานได้ 4 ประเภท คือ

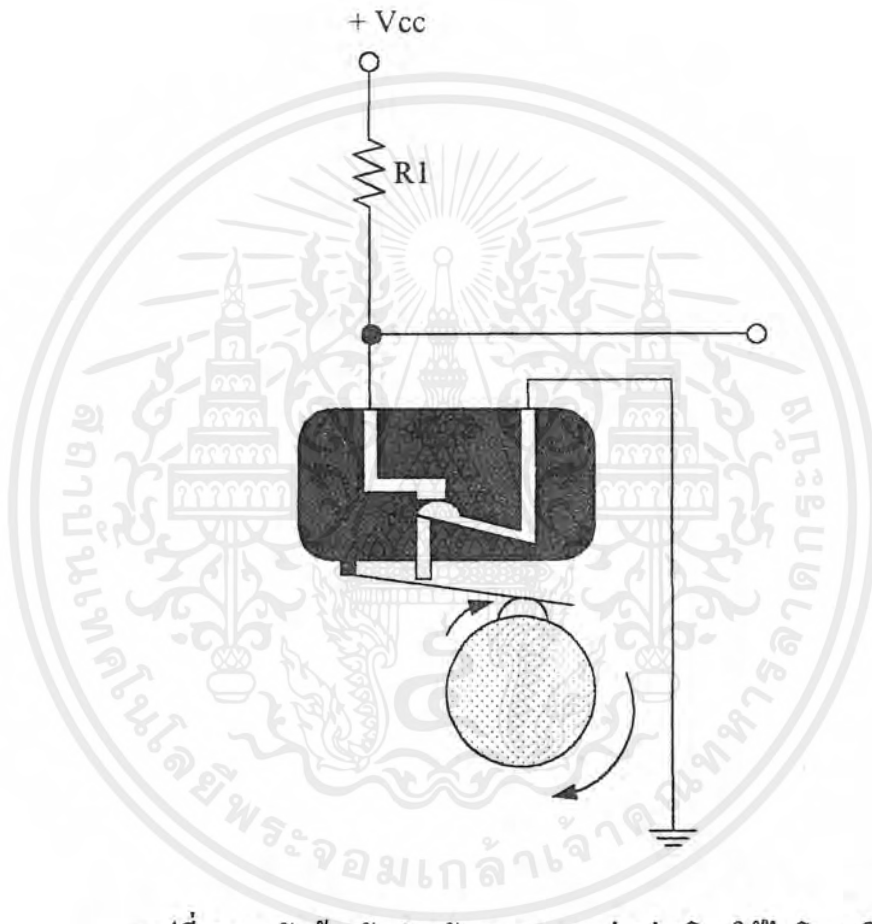
1. ใช้หลักการสวิตซ์ทางกล
2. สวิตซ์แบบฮอลล์เอฟเฟกต์
3. ใช้หลักการสะท้อนของแสง
4. หลักการของการตัดแสง

ซึ่งในปัจจุบันอุปกรณ์เหล่านี้ได้มีผู้ผลิตออกมาจำหน่ายมากมาย โดยจัดทำเป็นโมดูลที่ติดตั้งและใช้งานได้สะดวก จึงเริ่มเป็นที่นิยมใช้กันมากขึ้น แต่การเลือกใช้ก็จะต้องคำนึงถึงลักษณะของงานและความสะดวกด้วย ดังนั้นมาทำความรู้จักตัวตรวจจับที่ใช้หลักการทำงานต่าง ๆ กันก่อน

### 2.9.1 แบบใช้หลักการทางกล (Switch Sensors)

เป็นตัวตรวจจับที่ใช้หลักการเหมือนสวิตซ์เปิดปิดวงจรไฟฟ้าทั่วไป ซึ่งเป็นชนิดไมโครสวิตซ์ แต่แตกต่างกันที่ตัวตรวจจับแบบนี้จะมีตัวกระตุ้นหน้าสัมผัส (Actuator) ที่ติดตั้งให้หมุนไป เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พร้อมกับเพลาดังรูปที่ 2.11 ตัวกระตุ้นหน้าสัมผัสอาจทำขึ้นจากหัวสกรูก็ได้ ซึ่งจากรูปเมื่อเพลามหมุนไปในตำแหน่งที่หัวสกรูอยู่ด้านบนจะทำให้หน้าสัมผัสถูกดันขึ้น และเคลื่อนที่ไปกดสวิตช์ให้ปิดวงจรเอาต์พุตที่ได้จึงเป็นลอจิก “0” แต่เมื่อเพลามหมุนต่อไปอีกตัวกระตุ้นก็จะเปลี่ยนตำแหน่งไปค้ำย ทำให้สวิตช์เปิดวงจรอีกครั้ง เอาต์พุตที่ได้เป็นลอจิก “1” เมื่อหมุนครบ 1 รอบ จึงทำให้เกิดสัญญาณขึ้น 1 พัลส์



รูปที่ 2.11 ตัวเข้ารหัสสำหรับเพลามหมุนอย่างง่ายโดยใช้ไมโครสวิตช์

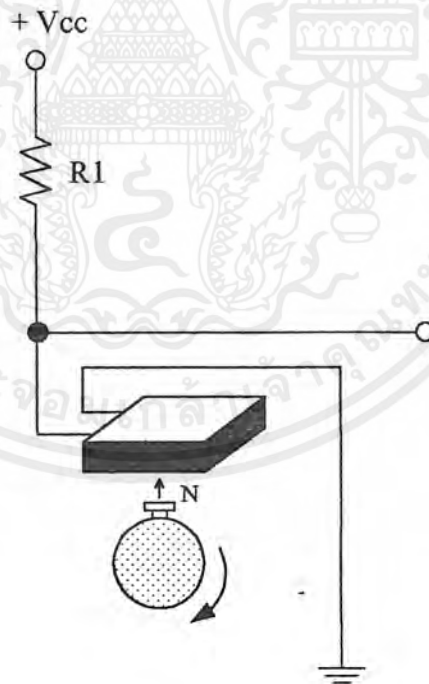
ตัวตรวจจับแบบนี้ง่ายต่อการนำไปใช้งานและออกแบบ แต่มีข้อเสียอันเกิดจากกลไกของหน้าสัมผัสสวิตช์ซึ่งอาจทำให้เกิดเขม่าขึ้น ทำให้การวัดผิดพลาดไป และความเร็วในการสวิตช์ก็มีค่าทำให้ไม่สามารถนำไปใช้งานกับเพลามีความเร็วรอบสูง ๆ ได้ อย่างไรก็ตามเมื่อใช้กับเพลามีความเร็วต่ำ ๆ ตัวตรวจจับแบบนี้จะทำให้ง่ายต่อการออกแบบและติดตั้ง ข้อเสียอีกประการหนึ่งก็คืออายุการใช้งานค่อนข้างสั้น เพราะเกิดจากการสวิตช์อย่างรวดเร็วทำให้เกิดการสึกหรอและเสียหายไปในที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.9.2 แบบใช้หลักการฮอลล์เอฟเฟกต์ (Hall Effect Seitch)

ตัวตรวจจับแบบฮอลล์เอฟเฟกต์สวิทช์มีข้อดีกว่าแบบกลไกคือ ไม่มีการสัมผัสระหว่างตัวตรวจจับและเพลทหมุน หลักการฮอลล์เอฟเฟกต์เป็นการนำปรากฏการณ์หนึ่งที่เกิดขึ้นกับสารกึ่งตัวนำชนิดพิเศษ ซึ่งเมื่อกระทบกับสนามแม่เหล็กจะเกิดการเปลี่ยนแปลงขึ้น ทำให้แรงดันเอาต์พุตที่ได้มีการเปลี่ยนแปลงตามไปด้วยและเป็นสัดส่วนกับความเข้มของสนามแม่เหล็กฮอลล์เอฟเฟกต์สวิทช์ที่สร้างขึ้นจะอยู่ในรูปของวงจรรวมซึ่งประกอบด้วยตัวตรวจจับความเข้มสนามแม่เหล็ก (Hall Effect Sensors) และวงจรรายสัญญาณเพื่อทำหน้าที่ขับสัญญาณเอาต์พุตที่ได้

จากรูปที่ 2.11 เป็นวงจรถ่ายสัญญาณฮอลล์เอฟเฟกต์สวิทช์ ซึ่งทำงานเมื่อขั้วเหนือของแม่เหล็กถาวรขนาดเล็กที่ติดบนเพลทเคลื่อนที่ผ่านผิวหน้าของตัวตรวจจับ โดยมีระยะห่างที่ตัวตรวจจับสามารถรับความเข้มของสนามแม่เหล็กได้พอเพียงถึงระดับเทรชโฮลด์ คือประมาณ 1/8 นิ้ว และไม่ควรเกิน 3/16 นิ้ว สำหรับการต่อสายมีลักษณะคล้ายกับแบบตัวตรวจจับชนิดกลไกซึ่งไม่ยุ่งยากและติดตั้งง่าย



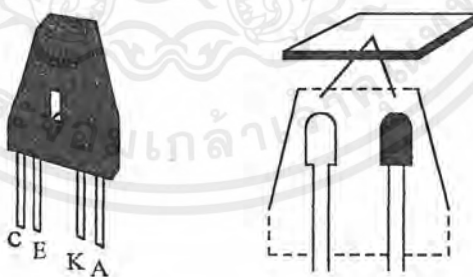
รูปที่ 2.12 ตัวเข้ารหัสแบบใช้หลักการฮอลล์เอฟเฟกต์

ข้อเสียของตัวตรวจจับแบบฮอลล์เอฟเฟกต์ก็คือ การเกิดฮิสเทอรีซิส (Hysteresis) ทำให้การเปลี่ยนแปลงสถานะการทำงานทั้งขณะเปิดและปิดไม่สามารถเปลี่ยนในทันทีทันใดได้ ดังนั้นเอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระยะห่างของตัวตรวจจับและแม่เหล็กถาวรที่ติดตั้งอยู่ที่เพลาคือพอเหมาะ รวมทั้งความเข้มของสนามแม่เหล็กจะต้องมีค่าพอเหมาะทำให้ตัวตรวจจับทำงานได้ อย่างไรก็ตามปัญหานี้แก้ไขได้โดยติดตั้งแม่เหล็กถาวรอีกตัวหนึ่งเพิ่มเข้าไปทางด้านหลังของฮอลล์เอฟเฟกต์สวิทช์ โดยหันขั้วได้เข้าหาสวิทช์เพื่อเสริมกับสนามแม่เหล็กที่เกิดขึ้นจากแม่เหล็กถาวรที่เพลามุมุน นั่นคือแม่เหล็กถาวรที่เพิ่มเติมขึ้นจะสร้างสนามแม่เหล็กไบแอส (Biasing Field) ทำให้ระยะห่างระหว่างตัวตรวจจับและเพลามุมุนขยายได้มากขึ้น นอกจากนี้การต่อใช้งานตัวตรวจจับแบบฮอลล์เอฟเฟกต์ในส่วนของเขาต์พุตที่จะนำไปใช้งานต้องต่อผ่านวงจรฟลิปฟลอปเพื่อปรับปรุงคลื่นให้เป็นคลื่นสี่เหลี่ยมสมมาตร (Symmetrical Square Waves) ช่วยแก้ไขผลอันเกิดจากฮิสเตอร์ซิซิสอีกครั้งหนึ่งก่อนนำไปใช้งาน

### 2.9.3 แบบใช้หลักการสะท้อนของแสง (Reflective Optics : Sensors)

อุปกรณ์ตรวจจับแบบใช้หลักการสะท้อนมีข้อดีที่เหนือกว่าแบบฮอลล์เอฟเฟกต์ตรงที่ การตรวจจับสามารถกำหนดพื้นที่การทำงานได้แน่นอนทำให้ออกแบบระบบได้ง่าย รูปที่ 3.5 แสดงตัวตรวจจับแบบใช้หลักการสะท้อนแสง จากรูปจะเห็นได้ว่าประกอบด้วยแอลอีดีอินฟราเรด (Infrared LED) และโฟโตทรานซิสเตอร์ ซึ่งจัดวางในตำแหน่งที่หันทิศทางไปในทางเดียวกัน ดังนั้นโฟโตทรานซิสเตอร์จึงไม่ได้กระทบกับแสงอินฟราเรด ทำให้โฟโตทรานซิสเตอร์อยู่ในสภาวะหยุดทำงานตลอดเวลา



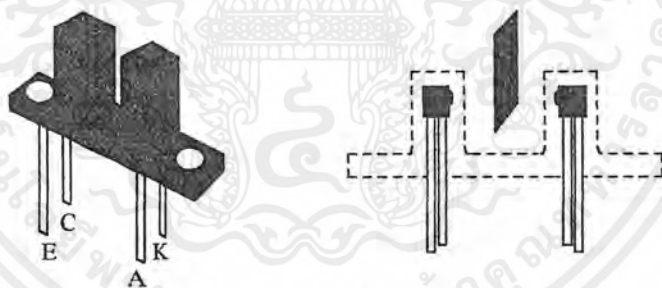
รูปที่ 2.13 ตัวเข้ารหัสแบบใช้หลักการการสะท้อนแสง

การตรวจจับจะอาศัยวัสดุที่สะท้อนแสงได้ เช่น โลหะสะท้อนแสง หรือ แผ่นฟอยล์ (Foil) ซึ่งถูกเรียกว่าตัวสะท้อนแสง (Reflector) มาวางไว้บริเวณด้านหน้าของตัวตรวจจับในระยะที่สามารถทำงานได้ ถ้าแสงอินฟราเรดจากแอลอีดีสะท้อนกลับมาถึงโฟโตทรานซิสเตอร์ทำให้โฟโตทรานซิสเตอร์ทำงาน ได้สัญญาณที่ได้จากเอาต์พุตโฟโตทรานซิสเตอร์จะมีระดับลอจิกที่แน่นอน เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเปลี่ยนแปลงระดับลอจิกได้อย่างรวดเร็วหรือเก็ทสเคอร์ชิสน้อยมาก ทำให้รูปคลื่นของสัญญาณที่ได้ไม่เกิดการบิดเบี้ยวเนื่องจากการทำงานอาศัยหลักการสะท้อนแสงทำให้มีประสิทธิภาพสูง ตัวตรวจจับประเภทนี้จึงนิยมถูกนำมาออกแบบใช้งานกับระบบตัวตรวจจับ 2 ชุด เพราะสามารถนำมาติดตั้งกับงานเข้ารหัสที่ติดตั้งตัวสะท้อนแสงได้หลายรูปแบบ เช่น ติดตั้งในด้านเดียวกันแต่ละคนละตำแหน่ง หรือติดตั้งด้านหน้าหรือด้านหลังของงานเข้ารหัสก็ได้ทำให้ประยุกต์ใช้งานได้สูงขึ้น

#### 2.9.4 แบบใช้หลักการตัดลำแสง (Interruption Sensors)

ส่วนประกอบที่สำคัญของตัวตรวจจับแบบนี้ยังคงใช้แอลอีดีอินฟราเรดและโฟโตทรานซิสเตอร์เช่นเดิม แต่แตกต่างกันตรงที่การจัดตำแหน่งของแอลอีดีอินฟราเรดและโฟโตทรานซิสเตอร์ ซึ่งจะวางหันหน้าเข้าหากันโดยมีช่องว่างแคบ ๆ ดังรูปที่ 2.14 ลำแสงอินฟราเรดจากแอลอีดีจะส่งผ่านช่องว่างมากระทบกับโฟโตทรานซิสเตอร์ให้ทำงานตลอดเวลา แต่เมื่อใดที่มีวัตถุทึบแสงมาตัดผ่านในช่องว่าง จะทำให้แสงที่กระทบกับโฟโตทรานซิสเตอร์ขาดหายไปและทำให้หยุดทำงานทันที



รูปที่ 2.14 ตัวเข้ารหัสแบบใช้หลักการตัดแสง

การใช้งานตัวตรวจจับแบบนี้ต้องใช้ร่วมกับงานเข้ารหัสที่มีขนาดพอเหมาะและไม่บิดเบี้ยว เพื่อป้องกันการเสียดสีระหว่างงานเข้ารหัสและตัวตรวจจับในช่องแคบ งานเข้ารหัสที่ใช้จะเป็นวัสดุโปร่งแสงเช่นพลาสติกใส และพิมพ์ลายช่องด้วยวัสดุทึบแสง การทำงานของตัวตรวจจับแบบใช้หลักการตัดแสงจะให้ความแม่นยำสูง เพราะไม่ต้องอาศัยการสะท้อนแสง แต่อย่างไรก็ตามการประยุกต์ใช้งานทำได้ค้อยกว่าแบบใช้หลักสะท้อนแสงเพราะสามารถติดตั้งได้เพียงลักษณะเดียวคือ คล่อมระหว่างด้านหน้าและด้านหลังของงานเข้ารหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ตรวจจับที่ใช้หลักการของแสงทั้งแบบสะท้อนและการตัดลำแสงมีข้อเสียที่ไม่สามารถทำงานในพื้นที่ที่มีความสว่างมาก ๆ ได้ เพราะแสงจากแหล่งอื่นจะตกกระทบโฟโตรีสซิสเตอร์จนไม่สามารถตรวจจับแสงได้ ทั้งประเภทส่งลำแสงแบบตรง และสะท้อนจาก LED เพื่อแก้ปัญหาที่อุปกรณ์ตรวจจับที่ใช้หลักการของแสงจึงต้องมีชิลด์ (Shield) เพื่อป้องกันแสงจากแหล่งภายนอกด้วย ซึ่งอุปกรณ์เข้ารหัสสำหรับเพลลาหมุนที่ใช้งานจริงจะถูกออกแบบให้มีการชิลด์ในตัวพร้อมใช้งานได้เลย ตัวเข้ารหัสที่ใช้งานจริงในอุตสาหกรรม ซึ่งมีทั้งแบบตรวจจับอย่างหยาบและแบบละเอียดให้เลือกใช้งานสำหรับระบบควบคุมโดยคู่ร่วมกับเพลลาได้ทันที วงจรต่าง ๆ จะถูกบรรจุอยู่ภายในทั้งหมด แต่สำหรับในรุ่นเกรฮิลล์ (Grayhill) ต้องมีการต่อตัวต้านทานพูลอัพไว้ภายนอกด้วย เพราะเอาต์พุตจะเป็นแบบคอลเล็กเตอร์เปิด ลักษณะการหมุนของแกนจะเป็นแบบคลิก (Click Stop) ซึ่งเป็นผลมาจากงานเข้ารหัสภายในเอาต์พุตที่ได้ต่อการหมุนครบ 1 รอบ เป็น 32 พัลส์ ส่วนในรุ่นของเคลโรสแตต (ClaroStat) จะสามารถผลิตพัลส์ได้มากกว่าเป็น 128 พัลส์ต่อรอบ และมีลักษณะการทำงานที่เป็นแบบคลิกทำให้การทำงานเรียบไม่เกิดการสั่นขึ้นภายใน

## 2.10 ระบบส่งกำลังของโครงการ

ในการพิจารณาวัสดุและอุปกรณ์ที่จะนำมาใช้ในการส่งกำลังให้เป็นตามจุดประสงค์นั้นเราจะพิจารณาเป็นอย่าง ๆ ไป คือ

- 2.10.1 เพลลา (SHAFT)
- 2.10.2 แบริ่ง (BEARING)
- 2.10.3 โซ่, เฟือง และ สายพาน

### หลักการออกแบบส่วนประกอบและโครงสร้างต่าง ๆ

#### 2.10.1 เพลลา (Shaft)

เพลลาอาจมีชื่อเรียกแตกต่างกันออกไป ตามลักษณะการใช้งานดังนี้

- เพลลา (Shaft) เป็นชิ้นส่วนที่หมุนและใช้ในการส่งกำลัง
- แกน (Axle) เป็นชิ้นส่วนลักษณะเดียวกับเพลลาแต่ไม่หมุน ส่วนมากเป็นตัว

รองรับชิ้นส่วนที่หมุน เช่น ล้อ ล้อสายพาน เป็นต้น อย่างไรก็ตามทั้งเพลลาและแกนก็นิยมเรียกรวมกันว่า เพลลา ไม่ว่าจะชิ้นส่วนจะหมุนหรือไม่หมุนก็ตาม

- สปินเดิล (Spindle) เป็นเพลาขนาดสั้นที่ไม่หมุน เช่น เพลาที่แทนกลิ้ง (Head Shaft) เป็นเพลาที่ติดเป็นชิ้นส่วนต่อเนื่องกับเครื่องยนต์ มอเตอร์ หรือเครื่องต้นกำลังอื่น ๆ มีขนาดรูปร่างและส่วนยื่นออกมาต่อกับเพลอื่น ๆ

- เพลาแนว (Line Shaft) หรือเพลาที่ส่งกำลัง (Power Transmission Shaft) หรือเพลामน (Main Shaft) เป็นเพลาที่ต่อตรงจากเครื่องต้นกำลังและใช้ในการส่งกำลังไปยังเครื่องจักรอื่น ๆ โดยเฉพาะ

- แจ็กชาฟ (Jick Shaft) หรือเค้าเคอร์ชาฟ (Counter Shaft) เป็นเพลาขนาดสั้นที่ต่อระหว่างเครื่องต้นกำลังกับเพลामนหรือเครื่องจักรกล

- เพลาอ่อน (Flexible Shaft) เป็นเพลาที่สามารถอ่อนตัวหรือโค้งงอได้ เพลาประเภทนี้ทำด้วยสายลวดใหญ่ (Cable) ลวดสปริงหรือลวดเกลียว (Wire Rope) ใช้ในการส่งกำลังในลักษณะที่แกนหมุนทำมุมกัน ได้แก่ส่งกำลังได้น้อย

เพลาอาจรับแรงดึง แรงกด แรงบิด หรือแรงดัน หรือหลายแรงร่วมกันก็ได้ ดังนั้นการคำนวณก็ต้องใช้ความเค้นผสมเข้าช่วย แรงเหล่านี้ยังอาจจะมีการเปลี่ยนแปลงขนาดตลอดเวลาทำให้เพลาสีหายเพราะความล้าได้ ฉะนั้นจะต้องออกแบบเพลาให้มีความแข็งแรงเพียงพอสำหรับการใช้งาน นอกจากนี้เพลายังมีความแข็งแรง (Rigidity) เพียงพอเพื่อลดมุมบิดภายในเพลาให้อยู่ในขีดจำกัดที่เหมาะสม ระยะโก่ง (Deflection) ของเพลาก็เป็นสิ่งสำคัญในการกำหนดขนาดเพลา เช่นเดียวกัน เพราะถ้าเพลามีระยะโก่งมากก็จะเกิดการแคงขณะหมุนทำให้เกิดความเร็ววิกฤต (Critical Speed) ของเพลาลดลง ซึ่งอาจทำให้เพลามีการสั่นอย่างรุนแรง ในขณะที่ความเร็วของเพลาเข้าใกล้ความเร็ววิกฤตนี้ได้ ระยะโก่งนี้ยังมีผลต่อการเลือกชนิดของเพลาและที่รองรับเพลา เช่น บอลเบริง (Ball Bearing) ก็ต้องมีการเอียงแนว (Misalignment) ในการใช้งานที่พอเหมาะกับเพลาค้วย

#### การพิจารณาในการออกแบบเพลา

การคำนวณหาขนาดของเพลาที่เหมาะสม ขึ้นอยู่กับลักษณะการใช้งานในบางครั้ง การหาขนาดของเพลา เพื่อให้เพลาทนต่อแรงที่มากระทำอย่างเฉียวเท่านั้นไม่เพียงพอ เช่น ในกรณีของเพลากลูกเบี้ยว (Cam Shaft) ในเครื่องยนต์สันดาปภายในต้องการให้มีตำแหน่งที่เที่ยงตรง ดังนั้นมุมบิดของเพลาที่เกิดขึ้นในขณะที่ใช้งานจะต้องมีค่าไม่มากกว่าที่กำหนดไว้ เป็นต้น นั่นคือ เพลาจะต้องมีความแข็งแรงอยู่ในพิสัยที่ต้องการ ถ้ามุมบิดมากไปนอกจากจะเสียความเที่ยงตรงทางด้านตำแหน่งแล้ว ยังอาจจะก่อให้เกิดความสั่นสะเทือนซึ่งมีผลทำให้เฟืองและเบริงที่รองรับเพลาอยู่เกิดความเสียหายได้ง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถึงแม้ว่าจะไม่มีมาตรฐาน สำหรับพิกัดมุมของเพลลาไว้ก็ตาม ในทางปฏิบัติแล้วมัก จะใช้ มุมบิดของเพลลาในเครื่องจักรทั่วไป ไม่เกิน 0.3 องศาต่อความยาวเพลลา 1 เมตร สำหรับเพลลาส่ง กำลังทั่วไป อาจจะทำให้มีมุมบิดถึง 1 องศาต่อความยาวเพลลา 20 เท่าของขนาดเส้นผ่าศูนย์กลางของ เพลลา ในกรณีของเพลลาสำหรับลูกเบี้ยวของเครื่องยนต์สันดาปภายในแล้วจะทำให้มุมบิดไม่เกิน 0.5 องศาต่อความยาวของเพลลา

ความแข็งแรงที่สำคัญอีกอย่างหนึ่งก็คือ ความแข็งแรงทางด้านระยะโก่ง เพราะจะต้อง ใช้ระยะโก่งของเพลลาที่อยู่ภายใต้แรงภายนอกเป็นตัวสำคัญในการกำหนดระยะเบียดด้วย (Clearanc) ระหว่างล้อสายพาน เพื่อ โครงสร้างของเครื่องจักรตลอดจนการเลือกชนิดของแบร็ง สำหรับรองรับเพลลาที่หมุนให้เหมาะสม ถ้าเพลลามีระยะโก่งมากเกินไป จะทำให้ความยาวของฟัน เฟืองส่วนที่สัมผัสหรือขบกันลดลง เป็นผลให้อัตราส่วนการขบ (Contact Ratio) ของเฟืองลดลง ด้วย ทำให้ส่งกำลังของเฟืองไม่ราบเรียบเท่าที่ควร การเลือกแบร็งชนิดที่อนุญาตให้มีการเอียงแนว ใต้พอเหมาะกับระยะโก่งของเพลลาที่จะเกิดขึ้น

ระยะโก่งดังกล่าวนี้ไม่มีมาตรฐานกำหนดเป็นแนวทางไว้ โดยทั่ว ๆ ไปแล้วผู้ออกแบบ อาจถือค่าต่อไปนี้เป็นแนวทางในการคำนวณความแข็งแรงทางด้านระยะโก่งได้ดังนี้คือ

- สำหรับเพลลาเครื่องจักร ๆ ทั่วไป ค่าระยะโก่งระหว่างจุดที่รองรับด้วยแบร็งควรจะไม่ เกิน 0.08 มม/ม
- สำหรับเพลลาที่มีเฟืองตรงคุณภาพดีอยู่แล้ว ระยะโก่ง ณ. ตำแหน่งที่มีเฟืองขบกันไม่ควร เกิน 0.125 มม. และความลาดเอียงของเพลลา ณ. ตำแหน่งนี้ควรจะน้อยกว่า 0.0286 องศา
- สำหรับเพลลาที่มีเฟืองคอกงอกคุณภาพดีอยู่แล้ว ระยะโก่ง ณ. ตำแหน่งที่มีเฟืองขบกัน ไม่ควรเกิน 0.075 มม.

จากเหตุผลดังกล่าวมาแล้วจะเห็นว่าขนาดของเพลลา อาจจะสามารถหาได้โดยใช้ความแข็งแรง ที่ต้องการ แทนที่จะเป็นความแข็งแรงในด้านการรับแรงภายนอกก็ได้ การหาระยะโก่งของเพลลา อาจทำได้โดยใช้วิธีกลศาสตร์วัสดุ

### 2.10.2 แบร็ง (Bearing)

เป็นตัวช่วยลดความเสียดทาน ความสึกหรอและความร้อน ที่เกิดขึ้นในชิ้นส่วนของเครื่อง จักรที่มีการเสียดสีกัน การหล่อลื่นทำได้โดยการใส่สารหล่อลื่น (Lubrication oil) หรือจารบี (Grease) หรืออากาศและอื่น ๆ ซึ่งขึ้นอยู่กับการใช้งาน ในการเลือกใช้แบร็งนั้นจะต้องเลือกใช้ แบบให้เหมาะสมกับเพลลา

### 2.10.3 โข้ เฟือง และสายพาน

ในการส่งกำลังทางกลส่วนใหญ่แล้วจะมีการส่งกำลังไปยังเพลาอาจจะทำได้ 3 วิธีคือ

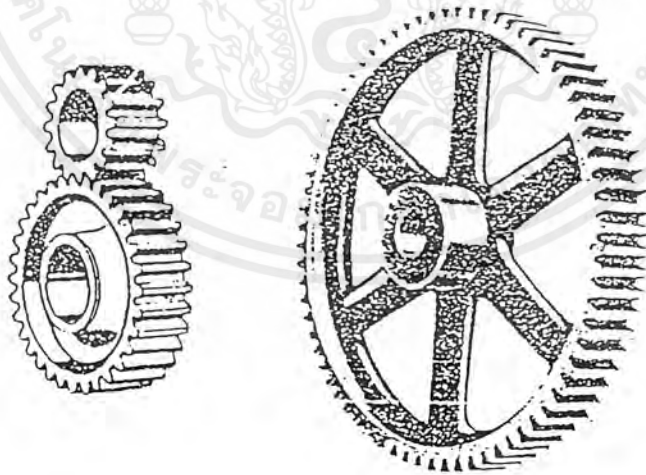
#### 2.10.3.1 เฟือง (Gears)

เฟืองเป็นชิ้นส่วนของเครื่องจักรกลอีกอย่างที่มีความสำคัญมากมีใช้ในงานวิศวกรรมอย่างกว้างขวาง เพราะสามารถส่งกำลังได้แม่นยำเที่ยงตรง ไม่มีการสั่นไถล ให้อัตราทดที่แน่นอนเช่นเดียวกันกับการส่งกำลังด้วยสายพาน

เฟืองแบ่งตามลักษณะรูปร่างจะมีอยู่ 4 แบบคือ

##### - เฟืองตรง (Spur gear)

เฟืองตรงจะมีลักษณะเป็นล้อทรงกระบอกกลม มีฟันขนานกับแกนของตัวเฟืองหรือแกนของเพลา หน้าตัดของฟันจะมีขนาดเท่ากันและเหมือนกันตลอดทั้งฟันเฟือง ฟันเฟืองตรงจะมีการใช้งานกันอย่างกว้างขวาง เช่น ในเครื่องจักรนาฬิกาข้อมือ ในเครื่องวัดที่มีเข็ม (Meters) ชุดเฟืองทดในเครื่องจักรกลทุกชนิด ในกระปุกเกียร์รถยนต์ สำหรับเฟืองขนาดใหญ่จะพบได้ตามโรงเจารีดเหล็ก เฟืองตรงมีลักษณะดังรูปที่ 2.15

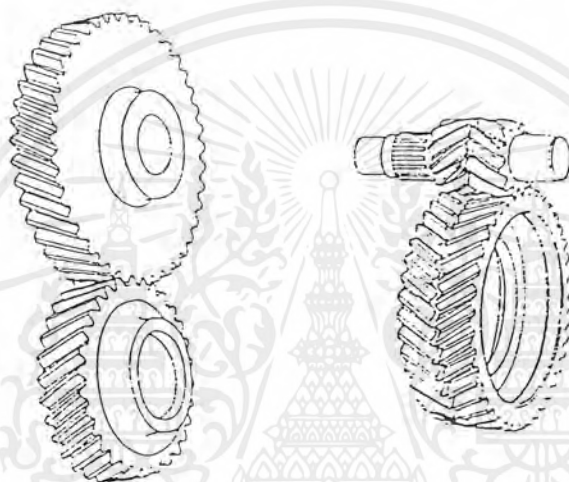


รูป 2.15 เฟืองตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### - เฟืองเฉียงหรือเฟืองพินเลี้ยว (Helical and Spiral Gears)

เฟืองเฉียงมีลักษณะคล้ายกับเฟืองตรง จะต่างกันตรงที่ฟันของเฟืองเฉียงจะเอียงทำมุมกับแกนของตัวเฟืองหรือแกนของเพลา เฟืองเฉียงจะใช้ส่งกำลังระหว่างเพลาที่ขนานกัน เพลาที่ทำมุมตั้งฉากกันและเพลาทำมุมน้อยกว่ามุมฉาก เฟืองเฉียงอาจจะก่อร่องเฉียงเป็นมุมสองอยู่คนละด้านกัน ซึ่งจะเรียกว่า เฟืองก้างปลา (Gearing-Bone gear) ซึ่งจะใช้ส่งกำลังได้ระหว่างเพลาที่ขนานกันเท่านั้น เฟืองเฉียงมีลักษณะดังรูป 2.16



รูป 2.16 เฟืองเฉียงหรือเฟืองพินเลี้ยว

### - เฟืองคอกจอก (Bevel Gear)

เฟืองคอกจอกหรือบางครั้งอาจเรียกว่า เฟืองบายศรี มีลักษณะรูปร่างเป็นรูปทรงกรวย (Cone) มีฟันเฟืองอยู่โดยรอบผิวของทรงกรวย ฟันจะถูกตัดขนานกับแกนของเฟืองหรือเพลา

แกนหรือตัดด้านหน้ากับด้านหลังไม่เท่ากันเฟืองคอกจอกจะใช้สำหรับส่งกำลังสำหรับเพลาที่ตั้งฉากกัน ส่งกำลังระหว่างเพลาที่ทำมุมกันน้อยกว่า 90 องศา และมากกว่า 90 องศา

ฟันของเฟืองคอกจอกจะมีหลายแบบได้แก่ ฟันตรง ฟันเลี้ยว ฟันเอียง เป็นต้น เฟืองคอกจอกจะใช้งานในระบบการส่งกำลังของเฟืองท้ายรถยนต์ ในระบบการเคลื่อนที่ของชิ้นส่วนในเครื่องจักรกลทั่ว ๆ ไป เฟืองคอกจอกมีลักษณะดังรูปที่ 2.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

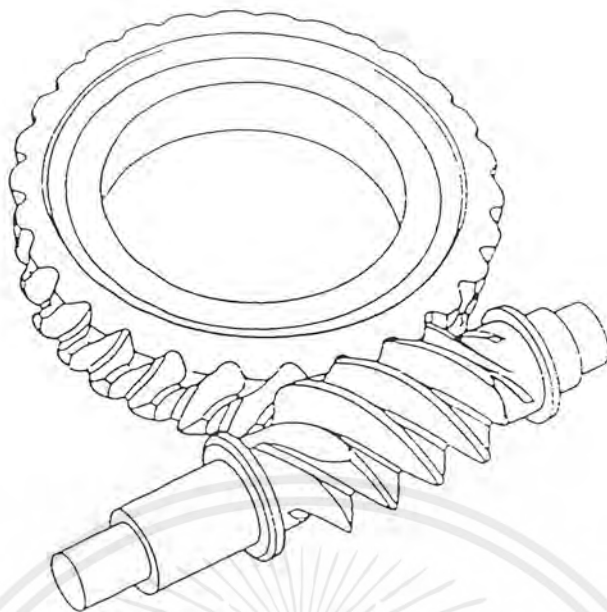


รูปที่ 2.17 เฟืองคอกจอก

- เฟืองหนอน (Worm Gearing)

เฟืองหนอนจะประกอบด้วยตัวเกลิยหนอน และเฟืองหนอน ซึ่งจะใช้งานร่วมกันโดยเกลิยหนอนจะส่งกำลังหมุน ไปขับให้เฟืองหนอนหมุนตาม ระยะพิตช์ของเกลิยกับระยะพิตช์ของเฟืองหนอนจะต้องมีค่าเท่ากัน เกลิยหนอนและเฟืองหนอนจะส่งกำลังระหว่างเพลลาที่ตั้งฉากและเพลลาที่วางข้ามกัน ชุดเฟืองหนอนจะใช้ในชุดหัวแบ่ง (Indexing Head) ในเครื่องมือวัดบางอย่างและในเครื่องจักรกลบางชนิด ที่ต้องการทกรอบจากความเร็วสูงให้เป็นความเร็วต่ำมาก ๆ เฟืองหนอนมีลักษณะดังรูปที่ 2.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 เฟืองหนอน

### 2.10.3.2 สายพาน

ในการส่งกำลังแบบสายพานนั้น เป็นการส่งกำลังแบบอ่อนตัวได้ ซึ่งมีทั้งข้อดี และข้อเสียหลายประการเมื่อเปรียบเทียบการส่งกำลังแบบใช้เฟือง

ข้อดี คือ มีราคาถูกและใช้งานได้ง่ายและสะดวกในการติดตั้ง รับแรงกระตุ้นและ แรงสั่นสะเทือนได้ดี ขณะใช้งานไม่เกิดเสียงดัง เหมาะสำหรับส่งกำลังระหว่างเพลาที่อยู่ห่างกัน มาก ๆ และค่าใช้จ่ายในการบำรุงรักษาต่ำ

ข้อเสีย คือ อัตราทดไม่แน่นอนต่อเนื่องจากสลิปหรือจากการเลื่อนไถลและการค รีพของสายพาน จะต้องมีการปรับระยะห่างระหว่างเพลาหรือปรับแรงดึงในสายพานระหว่างการ ใช้งาน นอกจากนี้ยังไม่อาจใช้งานที่มีอัตราทดสูงมากได้ซึ่งมักจะใช้กับอัตราทดไม่เกิน 6

### 2.10.3.3 โซ่

การขับเคลื่อนโซ่มีใช้กันมากในงานเครื่องกล เนื่องจากมีลักษณะการขับเคลื่อนด้วยสาย พาน โดยโซ่จะคล้อยอยู่กับล้อโซ่หรือเฟืองโซ่ซึ่งติดอยู่บนเพลาขับและเพลาตาม อัตราทดของการ ขับจะขึ้นอยู่กับขนาดของเฟืองโซ่ทั้งสองข้าง และการขับโซ่นี้จะไม่มีการสลิปขึ้นระหว่างโซ่กับ เฟืองโซ่

การขับเคลื่อนโซ่ มีข้อดีอยู่ระหว่างการขับเคลื่อนด้วยสายพานและ การขับเคลื่อนเฟืองทาง ด้านราคาสมรรถนะในการส่งกำลังและการบำรุงรักษา โซ่สามารถขับได้ในระยะทางไกลกว่าการ ขับด้วยสายพานและการขับได้พร้อมกันหลาย ๆ เพลา ซึ่งมีทิศทางการหมุนตามกันหรือสวนกันก็ได้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่หรือใช้เพื่อการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ข้อดี - ในการติดตั้งไม่ต้องการความเที่ยงตรงเท่ากับเฟือง
- ไม่จำเป็นต้องมีแรงดึงขึ้นต้นในโซ่ด้านดึงเหมือนกับสายพาน ทำให้อายุการใช้งานของแบร็งที่รองรับเพลามากขึ้น
  - ไม่มีการสลิปในขณะที่ส่งกำลังเหมือนสายพาน ทำให้ได้อัตราทดที่แน่นอน
  - มีขนาดกระทัดรัดกว่าสายพาน เมื่อใช้งานที่อัตราทดที่เท่ากัน เฟืองโซ่จะขนาดเล็กกว่าสายพาน และถ้าต้องการส่งกำลังเท่ากันความกว้างของโซ่จะน้อยกว่าความกว้างของสายพาน
  - ติดตั้งง่ายกว่าสายพาน เพราะเพียงแต่คล้องเข้ากับเฟืองโซ่แล้วถอดสลักเท่านั้น
  - ใช้งานได้อุณหภูมิสูง บริเวณที่มีความชื้นและฝุ่นละออง
- ข้อเสีย - มีเสียงดัง

- หากความเร็วรอบสูง จะเป็นอันตรายเมื่อโซ่ขาด
- เพลางจะต้องขนานกัน และไม่มีความปลอดภัยในการส่งกำลัง
- ราคาแพงกว่าสายพาน และต้องมีการหล่อลื่น

สิ่งจำกัดในการส่งกำลังด้วยโซ่เมื่อใช้งานด้วยความเร็วปานกลางจนถึงความเร็วสูงสุด คือการสึกหรอของข้อต่อ และความต้านทานแรงดึงของวัสดุชิ้นส่วนโซ่ นอกจากนี้แล้วถ้าใช้งานด้วยความเร็วสูงจะต้องคำนึงถึงแรงที่เกิดขึ้นในโซ่เนื่องจากแรงหนีศูนย์กลางในการขับโดยใช้โซ่หลายเส้นขนานกัน อาจทำให้แรงที่กระทำกับโซ่ไม่กระจายไปเท่ากันตลอดความกว้างของโซ่ และถ้าหล่อลื่นไม่ดีพอก็จะเป็นสาเหตุประการสำคัญที่ทำให้โซ่สึกหรอได้มาก

โซ่คล้องอยู่บนเฟืองโซ่จะมีลักษณะคล้ายรูปหลายเหลี่ยมบนเฟืองโซ่ ทำให้แกนของแรงในแนวสัมผัสกับเฟืองโซ่เปลี่ยนแปลงตลอดเวลาที่ส่งกำลัง นอกจากนี้ในระหว่างเคลื่อนที่เข้าหาและเคลื่อนที่ออกจากเฟืองโซ่ ข้อต่อโซ่แต่ละข้อจะเกิดการหมุนซึ่งการหมุนจะทำให้เกิดแรงเสียดทานขึ้น เป็นผลทำให้สูญเสียกำลังและสึกหรอ ผลต่อมาก็คือระยะพิทของโซ่เพิ่มขึ้นจนอาจจะหลุดออกจากเฟืองโซ่ได้

## บทที่ 3

### การออกแบบและวงจร

ในปัจจุบันนี้เราได้เห็นเครื่องอำนวยความสะดวกและเครื่องอัตโนมัติต่าง ๆ มากมาย อาทิ เช่น เครื่องเล่น CD อัตโนมัติ ตู้ชื้อน้ำอัตโนมัติ หรือการใช้เตาไมโครเวฟในการอุ่นอาหาร หรือทำอาหารแทนการใช้เตาแก๊ส ซึ่งที่กล่าวมาทั้งหมดนี้เราสามารถจะใช้เครื่องใช้ต่าง ๆ ได้โดยการ กดปุ่มเพียงไม่กี่ปุ่ม ฉะนั้นเราจึงอาศัยแนวความคิดต่าง ๆ ที่ได้เห็นเหล่านี้นำมาประยุกต์ใช้ในการ ออกแบบการทำงานของเครื่องอัตโนมัติอย่างหนึ่งเพื่ออำนวยความสะดวกในการทำอาหาร ซึ่งเรียกว่า “เครื่องทำอาหารอัตโนมัติ” โดยที่เครื่องดังกล่าวนี้สามารถจะประกอบอาหารซึ่งได้รสชาติที่เป็นมาตรฐานแทนการทำอาหารของมนุษย์ ทั้งนี้เนื่องจากบางคนทำอาหารไม่อร่อยหรือทำไม่เป็น ก็สามารถใช้เครื่องทำอาหารอัตโนมัตินี้แทนได้

#### 3.1 หลักการทำงานซึ่งประกอบไปด้วย 3 ส่วนใหญ่ ๆ ดังนี้ คือ

3.1.1 ส่วนทางด้าน Hardware

3.1.2 ส่วนทางด้านวงจร

3.1.3 ส่วนทางด้านโปรแกรม

##### 3.1.1 ส่วนทางด้าน Hardware

เป็นโครงสร้างทั้งหมดของเครื่องทำอาหารอัตโนมัติ โดยที่จะมีส่วนต่าง ๆ แยกกัน ทำงานตามหน้าที่ของแต่ละชุด ซึ่งจะแยกการทำงานออกเป็น 3 ชุด ได้แก่ ชุดลำเลียงหรือชุดส่งอาหาร ชุดผัดอาหารและชุดทอดอาหาร ซึ่งจะกล่าวในรายละเอียดต่อไป

##### 3.1.1.1 ชุดลำเลียงหรือชุดส่งอาหาร

ชุดลำเลียงหรือชุดส่งอาหารนี้เป็นชุดที่ใช้สำหรับลำเลียงเครื่องประกอบอาหาร และเครื่องประกอบอาหารในที่นี้จะใช้อะไรก็ได้แล้วแต่อาหารที่จะทำ ซึ่งในโครงงานนี้จะนำเสนอในการทำผัดซีอิ๊ว โดยในชุดลำเลียงจะมีเครื่องประกอบอาหารของผัดซีอิ๊วอยู่ โดยโครงสร้างของชุดลำเลียงหรือชุดส่งอาหารนี้จะมียู 2 ชุดด้วยกัน มีการทำงานคล้ายสายพานลำเลียงหมุนได้ทางเดียว โดยสายพานจะติดกับภาชนะที่ใช้ใส่เครื่องประกอบอาหาร การหมุนของชุดลำเลียงหรือชุดส่งอาหารจะอาศัยการทำงานของมอเตอร์ไฟฟ้ากระแสตรงจะติดชุดละ 1 ตัว การทำงานของมอเตอร์กระแสตรงในชุดนี้นั้นเมื่อถึงขั้นตอนที่ลำเลียงอาหารมาที่นั้น มอเตอร์ก็จะหมุนภาชนะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ติดอยู่มาที่เครื่องประกอบอาหารนั้นลงกะทะ โดยจะมีสวิทช์เซ็นเซอร์หยุดการทำงานของมอเตอร์กระแสตรงตัวนั้น แล้วมอเตอร์ในชุดอื่นก็จะทำงานต่อไปแค่เมื่อถึงเวลาที่จะใช้ชุดลำเลียงหรือชุดส่งอาหารอีกครั้งก็จะทำงานตามขั้นตอนดังกล่าวอีกครั้ง แต่จะสลับไปทำงานในชุดลำเลียงอีกชุดหนึ่ง เนื่องจากชุดลำเลียงมี 2 ชุดด้วยกัน สามารถดูได้ดังรูปที่ 3.1

### 3.1.1.2 ชุดผัดอาหาร

ชุดผัดอาหารเป็นชุดที่ใช้สำหรับการผัดอาหาร ที่ชุดลำเลียงส่งมาให้ในกะทะ การผัดอาหารนั้นข้อสำคัญจะต้องทำการผัดอาหารให้ทั่วกะทะ ดังนั้นจึงอาศัยระบบการทำงานของเฟืองเข้ามาช่วย โดยเฟืองที่นำมาใช้นั้นเป็นเฟืองตรง (Spur gear) ด้านบนของชุดเฟืองนี้จะติดกับมอเตอร์ไฟฟ้ากระแสตรง 1 ตัว ส่วนด้านล่างจะทำแกนออกมายึดติดกับพายซึ่งใช้สำหรับผัด เมื่อมอเตอร์ด้านบนหมุนพาด้านล่างก็จะหมุนไปด้วยเพื่อผัดอาหารที่อยู่ในกะทะให้ทั่วกะทะ โดยที่มอเตอร์ตัวนี้จะต้องทำงานตลอดเวลานกว่าอาหารนั้นจะสุก แต่ชุดผัดอาหารนั้นประกอบไปด้วยมอเตอร์กระแสตรง 2 ตัว ซึ่งตัวแรกได้กล่าวไปแล้ว ส่วนอีกตัวหนึ่งทำหน้าที่สำหรับยกชุดเฟืองผัดอาหารขึ้นและลง เนื่องจากเมื่อทำการผัดอาหารเสร็จแล้วจะนำอาหารไปเทต้องนำชุดเฟืองผัดอาหารขึ้นเสียก่อนถึงจะนำอาหารไปเทได้ และการจะทำอาหารก็ต้องนำชุดเฟืองผัดอาหารลงมาที่กะทะเพื่อเตรียมการผัดอาหาร โดยที่การยกขึ้นและลงของชุดเฟืองผัดอาหารนี้จะอาศัยมอเตอร์ตัวที่ 2 นี้ ทำงานซึ่งแกนของมอเตอร์จะติดกับเหล็กเกี่ยวกับการหมุนให้ชุดเฟืองผัดอาหารยกขึ้นจนชนสวิทช์จึงหยุดการทำงาน และเมื่อยกลงก็ทำการกลับทางหมุนมอเตอร์ตัวเดิมหยุดการทำงานเมื่อชนสวิทช์ สามารถดูได้ดังรูปที่ 3.2

### 3.1.1.3 ชุดเทอาหาร

ชุดเทอาหารเป็นชุดสุดท้ายในการทำงานของเครื่องอัตโนมัติ ซึ่งจะทำงานต่อจากชุดผัดอาหาร คือเมื่อชุดผัดอาหารยกชุดเฟืองผัดอาหารขึ้นจนชนสวิทช์ก็จะหยุดการทำงานแล้วชุดเทอาหารจะทำงานต่อ โดยชุดเทอาหารจะมีลักษณะเป็นโครงสี่เหลี่ยม มีเหล็กเกลียวทั้ง 2 ข้าง นั้นจะต้องหมุนพร้อมกันทั้ง 2 ข้าง ทั้งนี้ก็เพื่อนำกะทะไปเทอาหารให้ได้แนวระดับไม่เอียงจากจุดนี้จึงสามารถใช้หลักการโซ่เป็นตัวขับเคลื่อนเหล็กเกลียวทั้ง 2 ข้าง โดยที่เหล็กเกลียวด้านหนึ่งจะขบเฟืองอยู่กับมอเตอร์ไฟฟ้ากระแสตรง เมื่อมอเตอร์ทำงานเหล็กเกลียวด้านหนึ่งหมุน โซ่ก็จะทำให้เหล็กเกลียวอีกด้านหนึ่งหมุนตามไปพร้อมกัน มอเตอร์จะทำการหมุนไปข้างหน้าเรื่อย ๆ จนชนสวิทช์หยุดการทำงาน แล้วที่กะทะจะมีมอเตอร์อีกตัวหนึ่งที่ใช้สำหรับพลิกกะทะเทอาหารที่อยู่ข้างในลงจานอาหาร เมื่อพลิกชนสวิทช์มอเตอร์ก็จะกลับทางหมุนพลิกกะทะกลับมายังตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เดิม หลังจากนั้นมอเตอร์ตัวแรกก็จะทำการหมุนอีกครั้งหนึ่ง แต่กลับทางหมุนจากเดิมเพื่อนำกะทะ กลับมายังตำแหน่งเดิมซึ่งจะตรงกับชุดผัดอาหารพอดีเป็นการเสร็จสิ้นการทำงาน สามารถดูได้ดัง รูปที่ 3.3



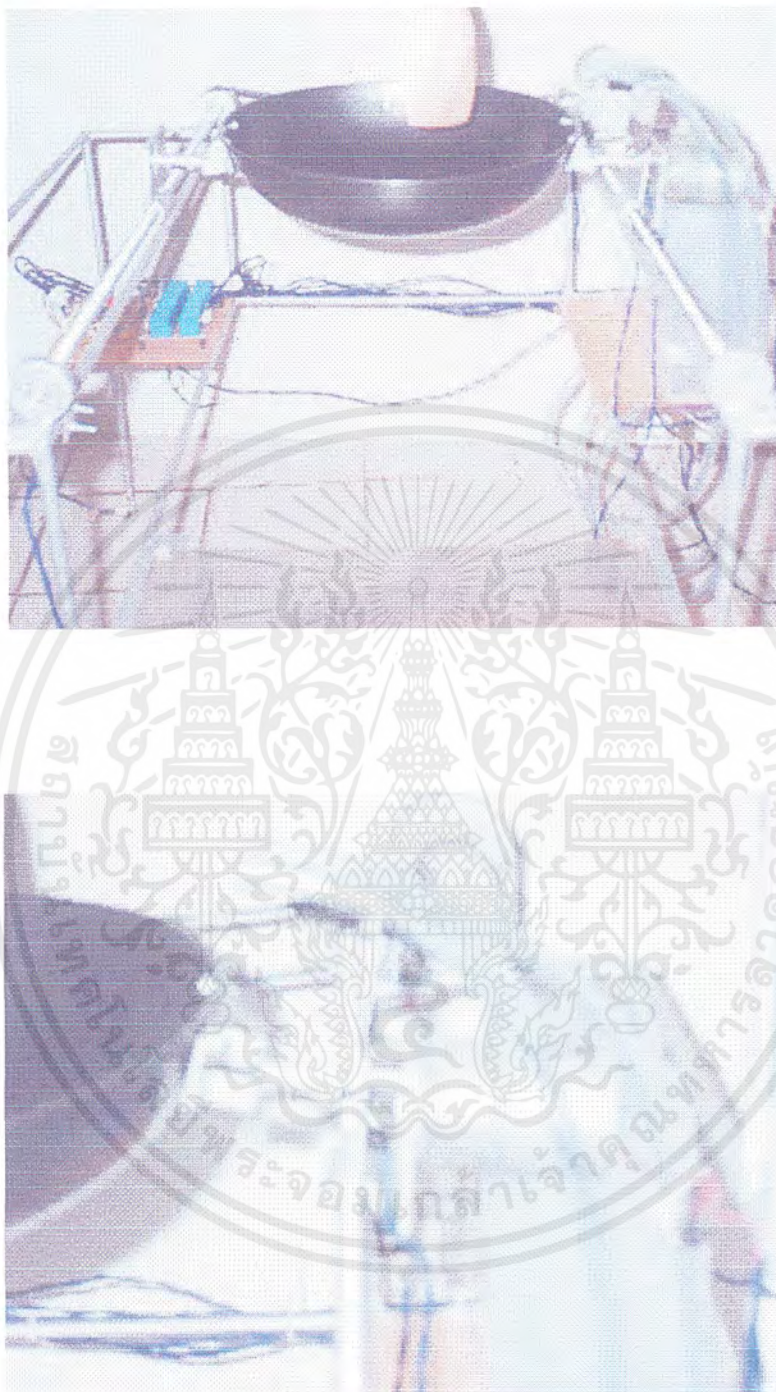
รูปที่ 3.1 ชุดถาดเลี้ยงหรือชุดส่งอาหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



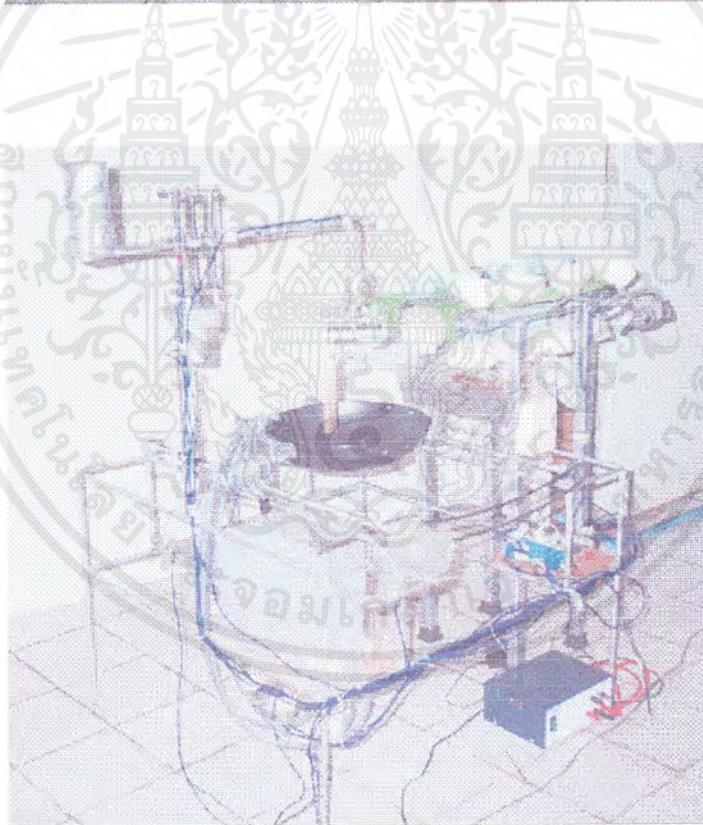
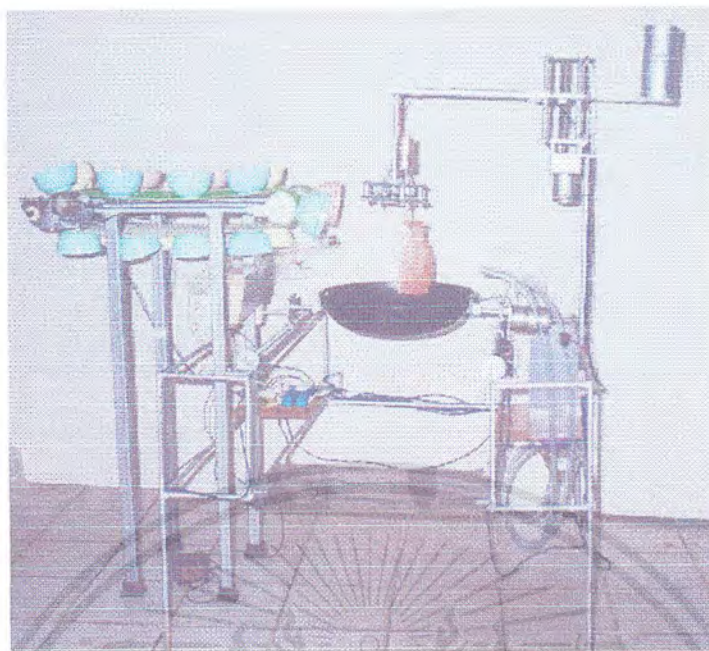
รูปที่ 3.2 ชุดผลิตอาหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 ชุดเตาอาหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 โครงสร้างทั้งหมดของเครื่องทำอาหารอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

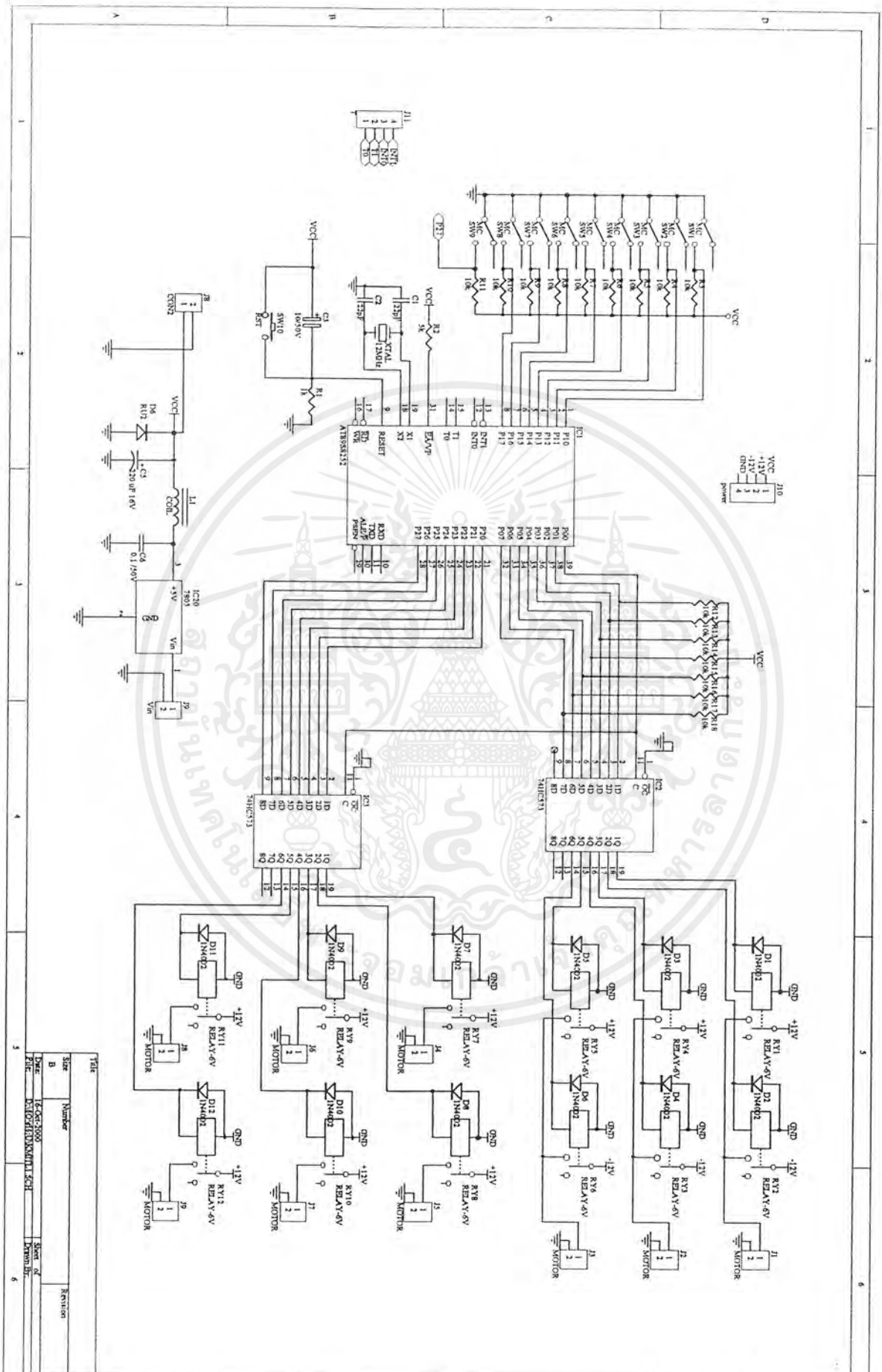
### 3.1.2 ส่วนด้านวงจร

เราใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ AT89C8252 ในการทำงาน โดยใช้ PORT 0 ทำการขับ dc motor ในการทำงาน ซึ่งการทำงานโดยทั่วไปของ PORT 0 คือ เป็นพอร์ตอินพุตเอาต์พุต 2 ทาง แบบ Open drain ขนาด 8 บิต ( P0.1 – P0.7 ) เมื่อใช้เป็นเอาต์พุตสามารถต่อกับ ไอซี TTL ตระกูลLS ได้ 8 ตัวและ PORT 0 จะทำงานอีกหน้าที่หนึ่งคือ เป็นมัลติเพล็กซ์ของสัญญาณ ค้านค่า และสัญญาณข้อมูลในการติดต่อกับหน่วยความจำภายนอก การทำงานในลักษณะนี้จะใช้ในการพูลอัพ ( Pull up ) จากภายนอกที่สามารถจ่ายกระแสให้กับอินพุตของ TTL ได้ 8 ตัว นอกจาก 2 หน้าที่ดังกล่าวแล้ว PORT 0 ยังเป็นตัวรับข้อมูลในช่วงการโปรแกรม EPROM และเป็นตัวส่งข้อมูลออกมาในช่วงการตรวจสอบโปรแกรมภายใน ROM หรือ EPROM ซึ่งจะต้องใช้พูลอัพจากภายนอกในขณะที่ทำการตรวจสอบโปรแกรม แต่ในที่นี้ใช้ IC เบอร์ 74HC573 เพื่อแลทช์ ( Latch ) สัญญาณในสถานะที่เราต้องการ

ซึ่งมีหลักการดังนี้ ในส่วนของวงจรประกอบไปด้วยอุปกรณ์ ดังนี้

1. MCS-51 เบอร์ AT89C8252	1	ตัว
2. IC 74HC573	2	ตัว
3. R. pull up 10k	2	ตัว
4. R 5k,10k	1	ตัว
5. C 20pF	2	ตัว
6. C10/50V	1	ตัว
7. XTAL 12MHz	1	ตัว
8. D 1N4002	12	ตัว
9. RELAY 6V	12	ตัว

ในวงจรนี้จะใช้ MCS-51 เบอร์ AT89C8252 ในการ control การทำงานของการทำงานทั้งหมด ซึ่งมีการทำงานใน Port 1 เป็น อินพุต Port 0 , Port 2 เป็น เอาต์พุตตามลำดับ โดยมีหลักการคือทำให้พอร์ตบิตตามที่กล่าวมาขึ้นต้น ในพอร์ต1 ให้เป็นอินพุตโดยต่อสวิทช์ผ่าน R pull-up ไปยังพอร์ต1 เมื่อมีสัญญาณบิตเข้ามาที่ PI.0-PI.7 เป็น “1” จะมีการตอบสนองทำให้มีการทำงานที่พอร์ต0 กับ พอร์ต 2 ให้ทำงานเป็นเอาต์พุตที่ขาใดขาหนึ่งตามกำหนดในโปรแกรม จากนั้นมีบิตออกไปยัง RELAY โดยที่มีแรงดันไฟ 5Vไปทำการแลทช์ให้ RELAY ทำงานจ่ายไฟ 12V ให้กับมอเตอร์ทำงาน ดังแสดงให้เห็นดังรูปที่ 3.5

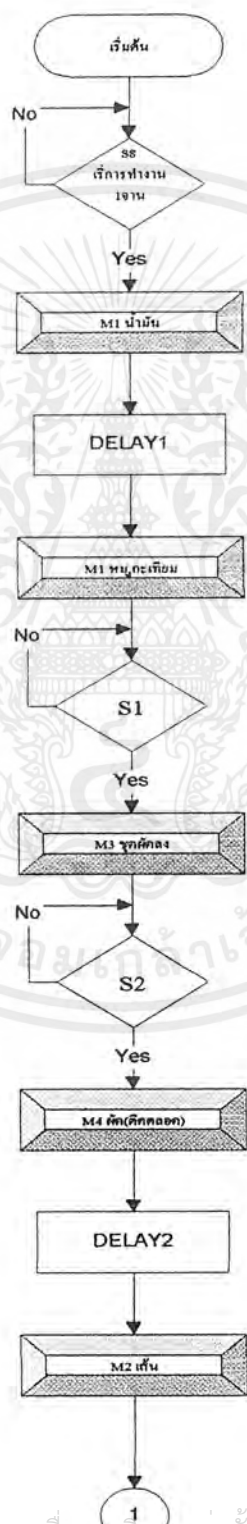


Size	Number	Revision
B	2-Oct-2000	
FILE: C:\PROG\PLC\HW\HWK\HWK1.SCH Sheet 6 of 6 Drawn By:		

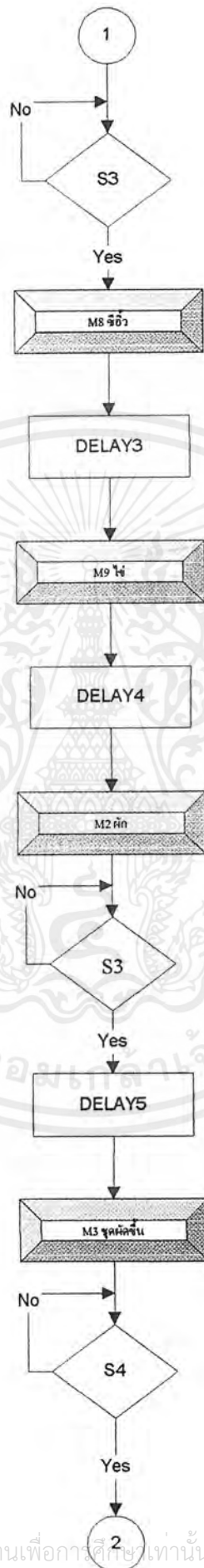
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.3 ส่วนทางด้านโปรแกรม

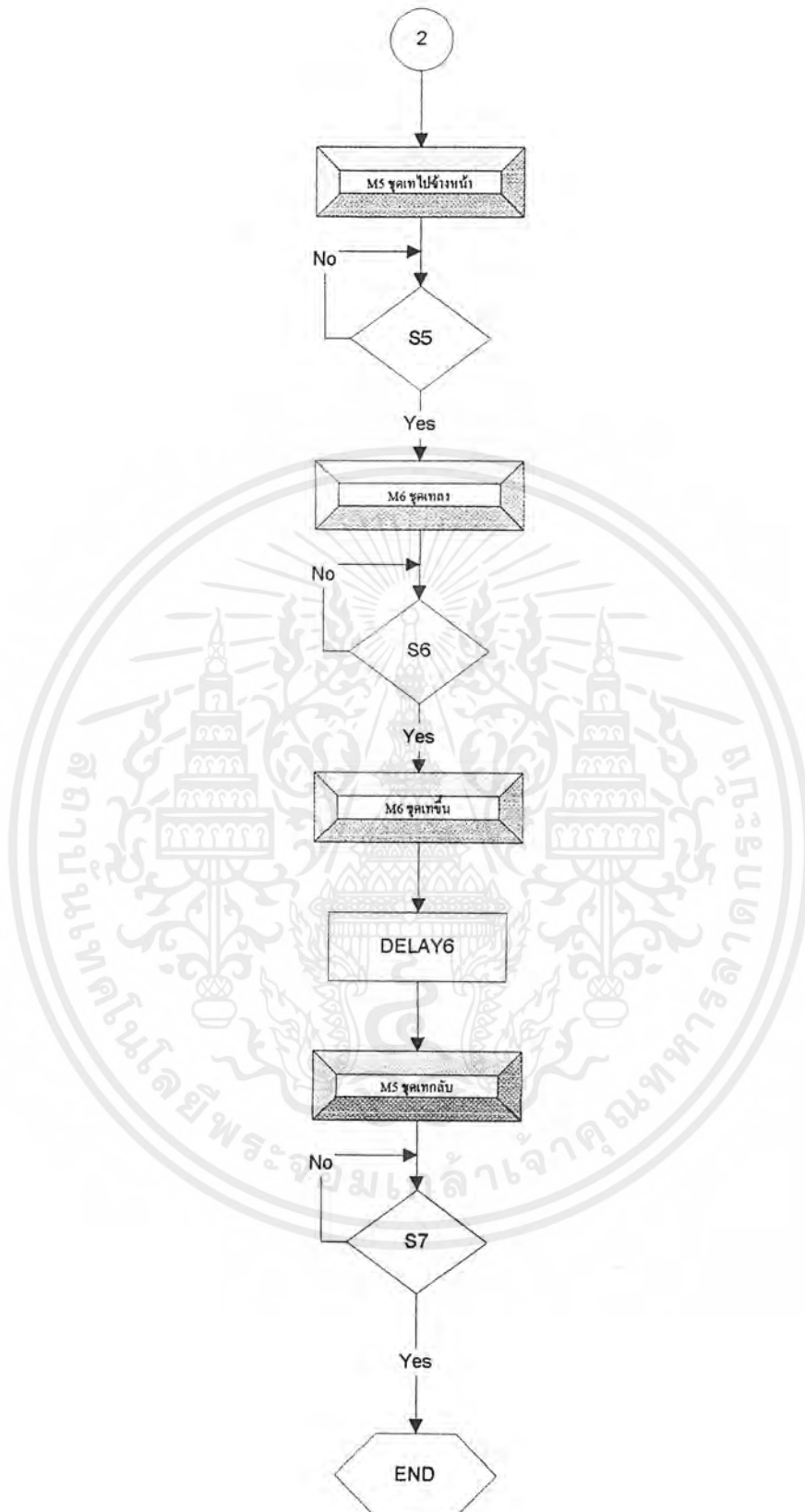
ส่วนนี้จะ เป็น Flowchart แสดงให้เห็นถึงขั้นตอนการทำงานทั้งหมดของโปรแกรม ซึ่งเป็นส่วนที่ควบคุมการทำงานในส่วนต่างๆของ Hardware โดยขั้นแรกของการเขียน โปรแกรมนั้นจำเป็นจะต้องเขียนเป็น Flowchart เสียก่อนก็เพื่อจะเป็นแนวทางในการเขียนโปรแกรมและง่ายต่อการแก้ไข จึงแสดงมาให้เห็นดังรูปต่าง ๆ ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 Flowchart แสดงการทำงานของเครื่องทำอาหารอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 หน้าที่ต่าง ๆ ของมอเตอร์ไฟฟ้ากระแสตรงในแต่ละตัว มีดังนี้

1. M1 ชุดลำเลียงตัวที่ 1
2. M2 ชุดลำเลียงตัวที่ 2
3. M3 ชุดผัดอาหารชั้น - ลง
4. M4 ชุดเฟืองผัดอาหาร
5. M5 ชุดเคลื่อนที่ไปข้างหน้า - หลัง
6. M6 ชุดพลิกเทอาหารลงจาน
7. M7 ฉีดน้ำมัน
8. M8 ฉีดชีอิ้ว
9. M9 ฉีดไข่

### 3.3 ขั้นตอนในการทำงานของมอเตอร์กระแสตรง

ในโครงการเครื่องทำอาหารอัตโนมัติ ได้นำเสนอการทำอาหารคือผัดชีอิ้ว ดังนั้นขั้นตอนต่าง ๆ เหล่านี้และเครื่องประกอบอาหารที่ใช้จะเป็นของอาหารประเภทผัดชีอิ้วทั้งสิ้น โดยมีขั้นตอนการทำงานดังนี้

1. ให้ M7 ฉีดน้ำมันลงกะทะ และหน่วยเวลาการทำงาน
2. ให้ M1 ทำงานเทหมู, กระทบลงกะทะ คัดสวิทซ์เซนเซอร์หยุดการทำงานของ M1
3. หน่วยเวลาการทำงานเพื่อให้หมู, กระทบสุก
4. ต่อจากนั้น M3 เคลื่อนที่ลงจนแตะสวิทซ์จะหยุดการทำงานเพื่อนำชุดเฟืองผัดอาหารให้อยู่ในกะทะ
5. ให้ M4 ทำงานชุดเฟืองผัดอาหารทำการผัดอาหารที่อยู่ในกะทะ ซึ่งจะทำงานตลอดเวลา
6. ให้ M1 ทำงานอีกครั้งเพื่อเทเส้นลงกะทะคัดสวิทซ์เซนเซอร์อีกครั้งหนึ่ง
7. ให้ M8 ฉีดชีอิ้วและหน่วยเวลาการทำงาน
8. ให้ M9 ฉีดไข่ลงกะทะและหน่วยเวลาการทำงาน
9. ให้ M2 ซึ่งเป็นชุดลำเลียงอีกชุดหนึ่งทำงานเทผัดลงกะทะ คัดสวิทซ์เซนเซอร์หยุดการทำงานของ M2
10. หน่วยเวลาการทำงานเพื่อให้อาหารสุกหมดโดยมี M4 ทำงานอยู่ตลอดเวลา
11. ให้ M4 หยุดการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. ต่อจากนั้นให้ M3 เคลื่อนที่ขึ้นจนแตะสวิทช์จะหยุดการทำงาน
13. ให้ M5 ทำงานจะเคลื่อนที่นำกะทะไปข้างหน้าจนแตะสวิทช์หยุดการทำงาน
14. หลังจากนั้น M6 เทอาหารที่อยู่ในกะทะลงจาน หน่วยเวลาเพื่อให้เทอาหารลงจาน  
หมด
15. ให้ M6 กลับทางหมุนมอเตอร์อยู่ในสภาวะปกติ
16. ให้ M5 ทำงานเคลื่อนที่นำกะทะกลับสู่ตำแหน่งเดิม จนชนสวิทช์หยุดการทำงาน
17. จบการทำงาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### สรุปผลการทำงาน

ในโครงการเครื่องทำอาหารอัตโนมัตินี้ ได้ประยุกต์แนวความคิดที่เห็นเครื่องทำอาหารอัตโนมัติต่าง ๆ ในปัจจุบันมาประดิษฐ์เป็นเครื่องทำอาหารอัตโนมัติ ซึ่งสามารถทำอาหารแทนมนุษย์ได้ โดยที่มีรสชาติมาตรฐาน ทั้งนี้เพื่อให้เกิดแนวคิดใหม่ในการทำงานอัตโนมัติหรือในอุตสาหกรรมของประเทศ แต่เนื่องจากเครื่องทำอาหารอัตโนมัติใช้การทำงานของมอเตอร์ไฟฟ้า กระแสตรงเป็นหลัก ในการขับเคลื่อนชุดต่าง ๆ ในการทำงานไม่ว่าจะเป็นชุดลำเลียงหรือชุดส่งอาหาร ชุดคั่วอาหาร และชุดเทอาหาร โดยการควบคุมการทำงานต่าง ๆ ให้ทำงานอัตโนมัติโดยใช้ไมโครคอนโทรลเลอร์ Mcs-51 เบอร์ AT89S8252 ควบคุม คือเมื่อมีการทำงานของมอเตอร์ไฟฟ้ากระแสตรงก็จะใช้สวิทช์เป็นตัวหยุดการทำงานของมอเตอร์ตัวนั้น แล้วให้มอเตอร์ตัวอื่นทำงานต่อไป ซึ่งการใช้มอเตอร์ไฟฟ้ากระแสตรงในการทำงานนี้ อาจเกิดปัญหาอยู่บ้างในเรื่องของการควบคุมตำแหน่งให้กลับสู่ตำแหน่งเดิมอาจจะมีการคลาดเคลื่อนอยู่บ้าง แต่ก็สามารถทำงานได้อย่างไม่มีปัญหา

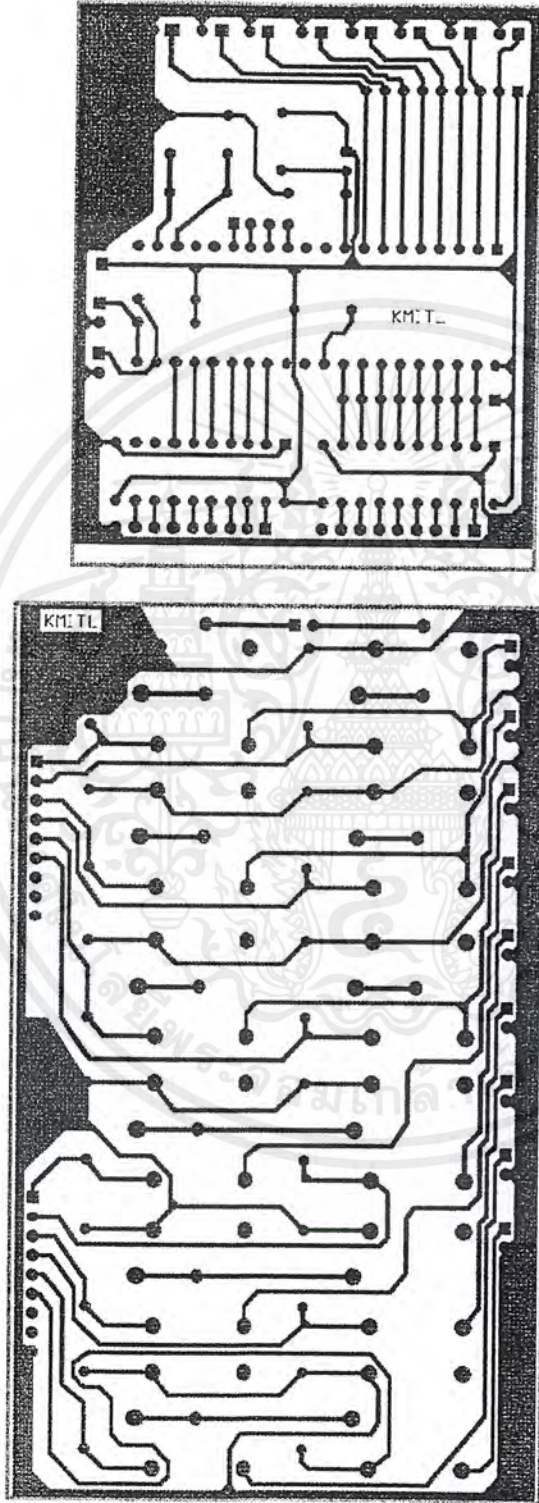
เครื่องทำอาหารอัตโนมัติในโครงการนี้อาจจะมีข้อจำกัดของการทำงานอยู่บ้าง คือ สามารถทำอาหารได้เพียงอย่างเดียว ซึ่งโครงการนี้ได้นำเสนอในการทำคั่วซีอิ๊ว โดยที่ชุดลำเลียงหรือชุดส่งอาหารจะมีเครื่องประกอบอาหารของคั่วซีอิ๊วอยู่ และเนื่องจากได้โปรแกรมการทำงานของไมโครคอนโทรลเลอร์ ให้ทำงานไปตามลำดับตลอดเสร็จสิ้นกระบวนการทำอาหาร และไม่สามารถเปลี่ยนแปลงการทำอาหารเป็นอย่างอื่นได้ ซึ่งเป็นขอบเขตของโครงการนี้ ปัญหาต่าง ๆ เหล่านี้สามารถแก้ไขได้โดยการ โปรแกรมให้เครื่องอัตโนมัติมีความสามารถมากขึ้น โดยมีฟังก์ชันปุ่มเลือกการทำอาหารประเภทใด ๆ ก็ได้แล้วแต่ผู้บริโภคว่าต้องการให้ทำอาหารอะไร ซึ่งทั้งนี้ อาจจะเป็นโครงการต่อไปในอนาคต



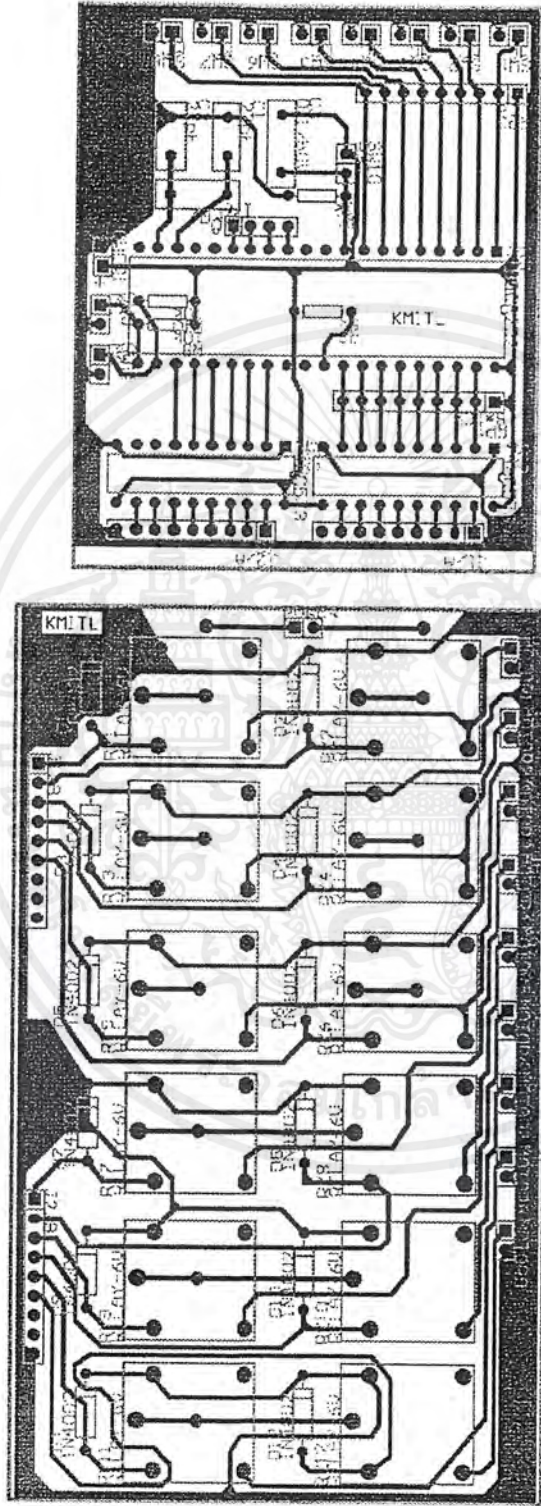
ภาคผนวก ก.

ลายวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

;  
; PROGRAM: auto matic cooking machine  
;

ORG 0000H

M3R EQU P0.0  
M3L EQU P0.1  
M5R EQU P0.2  
M5L EQU P0.3  
M6R EQU P0.4  
M6L EQU P0.5  
M7 EQU P0.6  
M4 EQU P2.7  
M1 EQU P2.6  
M2 EQU P2.5  
M8 EQU P2.4  
M9 EQU P2.3  
S1 EQU P1.0  
S2 EQU P1.1  
S3 EQU P1.2  
S4 EQU P1.3  
S5 EQU P1.4  
S6 EQU P1.5  
S7 EQU P1.6  
S8 EQU P1.7

;-----INITIAL-----

MOV P0,#00H  
MOV P1,#0FFH  
MOV P2,#00H

;

MAIN: MOV A,#00000000B  
MOV P0,A  
CLR C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

START: MOV C,P1.7

JC START

SETB M7

LCALL DE1

CLR M7

SE1: SETB M1

MOV C,S1

JC SE1

CLR M1

SE2: SETB M3R

MOV C,S2

JC SE2

CLR M3R

SETB M4

LCALL DE2

SETB M2

SE3: MOV C,S1

JC SE3

CLR M2

SETB M8

LCALL DE3

CLR M8

SETB M9

LCALL DE4

CLR M9

SETB M2

CLR C

SE4: MOV C,S1

JC SE4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CLR    M2
LCALL DE5
CLR    M4
SETB   M3L

SE5:  MOV    C,S4
      JC     SE5
      CLR    M3L
      SETB   M5R

SE6:  MOV    C,S5
      JC     SE6
      CLR    M5R
      SETB   M6R

SE7:  MOV    C,S6
      JC     SE7
      CLR    M6R
      LCALL  DE6
      SETB   M6L
      LCALL  DE7
      CLR    M6L
      SETB   M5L

SE8:  MOV    C,S7
      JC     SE8
      CLR    M5L
      MOV    A,#00H
      MOV    P2,A
      MOV    A,#0FFH
      MOV    P1,A
      LJMP   MAIN

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DE1:  MOV  R2,#0FH
DE13: MOV  R0,#0FFH
DE12: MOV  R1,#0FFH
DE11: DJNZ  R1,DE11
      DJNZ  R0,DE12
      DJNZ  R2,DE13
      RETI
```

```
DE2:  MOV  R2,#0FH
DE23: MOV  R0,#0FFH
DE22: MOV  R1,#0FFH
DE21: DJNZ  R1,DE21
      DJNZ  R0,DE22
      DJNZ  R2,DE23
      RETI
```

```
DE3:  MOV  R2,#0FH
DE33: MOV  R0,#0FFH
DE32: MOV  R1,#0FFH
DE31: DJNZ  R1,DE31
      DJNZ  R0,DE32
      DJNZ  R2,DE33
      RETI
```

```
DE4:  MOV  R2,#0FH
DE43: MOV  R0,#0FFH
DE42: MOV  R1,#0FFH
DE41: DJNZ  R1,DE41
      DJNZ  R0,DE42
      DJNZ  R2,DE43
      RETI
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DE5:  MOV  R2,#0FH
DE53: MOV  R0,#0FFH
DE52: MOV  R1,#0FFH
DE51: DJNZ R1,DE51
      DJNZ R0,DE52
      DJNZ R2,DE53
      RETI
```

```
DE6:  MOV  R2,#0FH
DE63: MOV  R0,#0FFH
DE62: MOV  R1,#0FFH
DE61: DJNZ R1,DE61
      DJNZ R0,DE62
      DJNZ R2,DE63
      RETI
```

```
DE7:  MOV  R2,#0FH
DE73: MOV  R0,#0EFH
DE72: MOV  R1,#0EFH
DE71: DJNZ R1,DE71
      DJNZ R0,DE72
      DJNZ R2,DE73
      RETI
```

```
END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Features

- Compatible with MCS-51™ Products
- 8K Bytes of In-System Reprogrammable Downloadable Flash Memory
  - SPI Serial Interface for Program Downloading
  - Endurance: 1,000 Write/Erase Cycles
- 2K Bytes EEPROM
  - Endurance: 100,000 Write/Erase Cycles
- 4.0V to 6V Operating Range
- 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
- Nine Interrupt Sources
- Programmable UART Serial Channel
- SPI Serial Interface
- Low Power Idle and Power Down Modes
- Interrupt Recovery From Power Down
- Programmable Watchdog Timer
- Dual Data Pointer
- Power Off Flag

## Description

The AT89S8252 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of Downloadable Flash programmable and erasable read only memory and 2K bytes of EEPROM. The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard 80C51 instruction set and pinout. The on-chip Downloadable Flash allows the program memory to be reprogrammed in-system through an SPI serial interface or by a conventional non-volatile memory programmer. By combining a versatile 8-bit CPU with Downloadable Flash on a monolithic chip, the Atmel AT89S8252 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

The AT89S8252 provides the following standard features: 8K bytes of Downloadable Flash, 2K bytes of EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog timer, two Data Pointers, 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 AT89S8252 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 interrupt or hardware reset.

The Downloadable Flash can be changed a single byte at a time and is accessible through the SPI serial interface. Holding RESET active forces the SPI bus into a serial programming interface and allows the program memory to be written to or read from unless Lock Bit 2 has been activated.



## 8-Bit Microcontroller with 8K Bytes Flash

AT89S8252

0401D-A-12/97

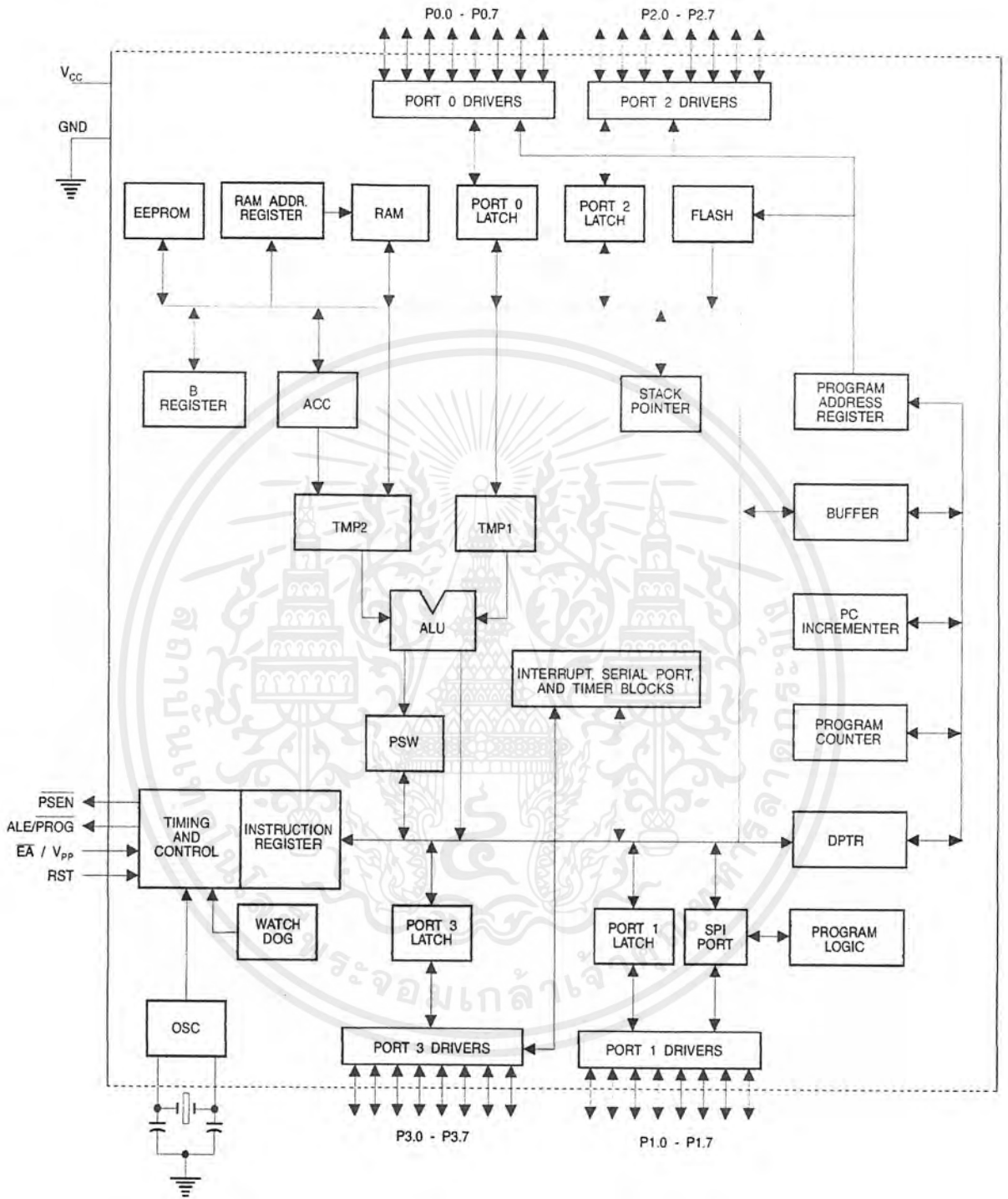


4-105

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Furthermore, P1.4, P1.5, P1.6, and P1.7 can be configured as the SPI slave port select, data input/output and shift clock input/output pins as shown in the following table.

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)
P1.4	$\overline{SS}$ (Slave port select input)
P1.5	MOSI (Master data output, slave data input pin for SPI channel)
P1.6	MISO (Master data input, slave data output pin for SPI channel)
P1.7	SCK (Master clock output, slave clock input pin for SPI channel)

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

#### 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 @ R1), 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 AT89S8252, 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 AT89S8252 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.

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

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  $V_{CC}$  for internal program executions. This pin also receives the 12-volt programming enable voltage ( $V_{PP}$ ) 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.

**Special Function Registers**

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

**Table 1. AT89S8252 SFR Map and Reset Values**

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000					SPCR 000001XX			0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			0CFH
0C0H									0C7H
0B8H	IP XX000000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0X000000		SPSR 00XXXXXX						0AFH
0A0H	P2 11111111								0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111						WMCON 00000010		97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000			8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	SPDR XXXXXXXX	PCON 0XXXX000	87H



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

**Timer 2 Registers** Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 9) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16 bit capture mode or 16-bit auto-reload mode.

**Watchdog and Memory Control Register** The WMCON register contains control bits for the Watchdog Timer (shown in Table 3). The EEMEN and EEMWE bits are used to select the 2K bytes on-chip EEPROM, and to enable byte-write. The DPS bit selects one of two DPTR registers available.

**SPI Registers** Control and status bits for the Serial Peripheral Interface are contained in registers SPCR (shown in Table 4) and SPSR (shown in Table 5). The SPI data bits are contained in the SPDR register. Writing the SPI data register during serial data transfer sets the Write Collision bit, WCOL, in the SPSR register. The SPDR is double buffered for writing and the values in SPDR are not changed by Reset.

**Interrupt Registers** The global interrupt enable bit and the individual interrupt enable bits are in the IE register. In addition, the individual interrupt enable bit for the SPI is in the SPCR register. Two priorities can be set for each of the six interrupt sources in the IP register.

**Table 2. T2CON—Timer/Counter 2 Control Register**

T2CON Address = 0C8H		Reset Value = 0000 0000B						
Bit Addressable								
Bit	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/ $\overline{T2}$	CP/RL2
7								
6								
5								
4								
3								
2								
1								
0								

Symbol	Function
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
RCLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflows to be used for the receive clock.
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
C/ $\overline{T2}$	Timer or counter select for Timer 2. C/ $\overline{T2}$ = 0 for timer function. C/ $\overline{T2}$ = 1 for external event counter (falling edge triggered).
CP/RL2	Capture/Reload select. CP/RL2 = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/RL2 = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

**Dual Data Pointer Registers** To facilitate accessing both internal EEPROM and external data memory, two banks of 16 bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H-85H. Bit DPS = 0 in SFR WMCON selects DP0 and DPS = 1 selects DP1. The user should always initialize the DPS bit to the

appropriate value before accessing the respective Data Pointer Register.

**Power Off Flag** The Power Off Flag (POF) is located at bit\_4 (PCON.4) in the PCON SFR. POF is set to "1" during power up. It can be set and reset under software control and is not affected by RESET.

**Table 3. WMCON—Watchdog and Memory Control Register**

WMCON Address = 96H						Reset Value = 0000 0010B		
	PS2	PS1	PS0	EEMWE	EEMEN	DPS	WDTRST	WDTEN
Bit	7	6	5	4	3	2	1	0

Symbol	Function
PS2 PS1 PS0	Prescaler Bits for the Watchdog Timer. When all three bits are set to "0", the watchdog timer has a nominal period of 16 ms. When all three bits are set to "1", the nominal period is 2048 ms.
EEMWE	EEPROM Data Memory Write Enable Bit. Set this bit to "1" before initiating byte write to on-chip EEPROM with the MOVX instruction. User software should set this bit to "0" after EEPROM write is completed.
EEMEN	Internal EEPROM Access Enable. When EEMEN = 1, the MOVX instruction with DPTR will access on-chip EEPROM instead of external data memory. When EEMEN = 0, MOVX with DPTR accesses external data memory.
DPS	Data Pointer Register Select. DPS = 0 selects the first bank of Data Pointer Register, DP0, and DPS = 1 selects the second bank, DP1
WDTRST RDY/BSY	Watchdog Timer Reset and EEPROM Ready/Busy Flag. Each time this bit is set to "1" by user software, a pulse is generated to reset the watchdog timer. The WDTRST bit is then automatically reset to "0" in the next instruction cycle. The WDTRST bit is Write-Only. This bit also serves as the RDY/BSY flag in a Read-Only mode during EEPROM write. RDY/BSY = 1 means that the EEPROM is ready to be programmed. While programming operations are being executed, the RDY/BSY bit equals "0" and is automatically reset to "1" when programming is completed.
WDTEN	Watchdog Timer Enable Bit. WDTEN = 1 enables the watchdog timer and WDTEN = 0 disables the watchdog timer.



**Table 4. SPCR—SPI Control Register**

SPCR Address = D5H		Reset Value = 0000 01XXB						
Bit	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
	7	6	5	4	3	2	1	0

Symbol	Function															
SPIE	SPI Interrupt Enable. This bit, in conjunction with the ES bit in the IE register, enables SPI interrupts: SPIE = 1 and ES = 1 enable SPI interrupts. SPIE = 0 disables SPI interrupts.															
SPE	SPI Enable. SPI = 1 enables the SPI channel and connects $\overline{SS}$ , MOSI, MISO and SCK to pins P1.4, P1.5, P1.6, and P1.7. SPI = 0 disables the SPI channel.															
DORD	Data Order. DORD = 1 selects LSB first data transmission. DORD = 0 selects MSB first data transmission.															
MSTR	Master/Slave Select. MSTR = 1 selects Master SPI mode. MSTR = 0 selects Slave SPI mode.															
CPOL	Clock Polarity. When CPOL = 1, SCK is high when idle. When CPOL = 0, SCK of the master device is low when not transmitting. Please refer to figure on SPI Clock Phase and Polarity Control.															
CPHA	Clock Phase. The CPHA bit together with the CPOL bit controls the clock and data relationship between master and slave. Please refer to figure on SPI Clock Phase and Polarity Control.															
SPR0 SPR1	SPI Clock Rate Select. These two bits control the SCK rate of the device configured as master. SPR1 and SPR0 have no effect on the slave. The relationship between SCK and the oscillator frequency, $F_{osc}$ , is as follows: <table border="1" style="margin-left: 40px;"> <thead> <tr> <th>SPR1</th> <th>SPR0</th> <th>SCK = <math>F_{osc}</math> divided by</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>4</td> </tr> <tr> <td>0</td> <td>1</td> <td>16</td> </tr> <tr> <td>1</td> <td>0</td> <td>64</td> </tr> <tr> <td>1</td> <td>1</td> <td>128</td> </tr> </tbody> </table>	SPR1	SPR0	SCK = $F_{osc}$ divided by	0	0	4	0	1	16	1	0	64	1	1	128
SPR1	SPR0	SCK = $F_{osc}$ divided by														
0	0	4														
0	1	16														
1	0	64														
1	1	128														

**Table 5. SPSR—SPI Status Register**

SPSR Address = AAH		Reset Value = 00XX XXXXB						
Bit	SPIF	WCOL	—	—	—	—	—	—
	7	6	5	4	3	2	1	0

Symbol	Function
SPIF	SPI Interrupt Flag. When a serial transfer is complete, the SPIF bit is set and an interrupt is generated if SPIE = 1 and ES = 1. The SPIF bit is cleared by reading the SPI status register with SPIF and WCOL bits set, and then accessing the SPI data register.
WCOL	Write Collision Flag. The WCOL bit is set if the SPI data register is written during a data transfer. During data transfer, the result of reading the SPDR register may be incorrect, and writing to it has no effect. The WCOL bit (and the SPIF bit) are cleared by reading the SPI status register with SPIF and WCOL set, and then accessing the SPI data register.

Table 6. SPDR—SPI Data Register

SPDR Address = 86H					Reset Value = unchanged			
	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
Bit	7	6	5	4	3	2	1	0

**Data Memory—EEPROM and RAM**

The AT89S8252 implements 2K bytes of on-chip EEPROM for data storage and 256 bytes of RAM. The upper 128 bytes of RAM occupy a parallel space to the Special Function Registers. That means the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the address mode used in the instruction specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions that use direct addressing access SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

```
MOV 0A0H, #data
```

Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

```
MOV @R0, #data
```

Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

The on-chip EEPROM data memory is selected by setting the EEMEN bit in the WMCON register at SFR address location 96H. The EEPROM address range is from 000H to 7FFH. The MOVX instructions are used to access the EEPROM. To access off-chip data memory with the MOVX instructions, the EEMEN bit needs to be set to "0".

The EEMWE bit in the WMCON register needs to be set to "1" before any byte location in the EEPROM can be written. User software should reset EEMWE bit to "0" if no further EEPROM write is required. EEPROM write cycles in the serial programming mode are self-timed and typically take 2.5 ms. The progress of EEPROM write can be monitored by reading the RDY/BSY bit (read-only) in SFR WMCON. RDY/BSY = 0 means programming is still in progress and RDY/BSY = 1 means EEPROM write cycle is completed and another write cycle can be initiated.

In addition, during EEPROM programming, an attempted read from the EEPROM will fetch the byte being written with the MSB complemented. Once the write cycle is completed, true data are valid at all bit locations.

**Programmable Watchdog Timer**

The programmable Watchdog Timer (WDT) operates from an independent oscillator. The prescaler bits, PS0, PS1 and PS2 in SFR WMCON are used to set the period of the Watchdog Timer from 16 ms to 2048 ms. The available timer periods are shown in the following table and the actual timer periods (at V<sub>CC</sub> = 5V) are within ±30% of the nominal.

The WDT is disabled by Power-on Reset and during Power Down. It is enabled by setting the WDTEN bit in SFR WMCON (address = 96H). The WDT is reset by setting the WDTRST bit in WMCON. When the WDT times out without being reset or disabled, an internal RST pulse is generated to reset the CPU.

Table 7. Watchdog Timer Period Selection

WDT Prescaler Bits			Period (nominal)
PS2	PS1	PS0	
0	0	0	16 ms
0	0	1	32 ms
0	1	0	64 ms
0	1	1	128 ms
1	0	0	256 ms
1	0	1	512 ms
1	1	0	1024 ms
1	1	1	2048 ms



## Timer 0 and 1

Timer 0 and Timer 1 in the AT89S8252 operate the same way as Timer 0 and Timer 1 in the AT89C51, AT89C52 and AT89C55. For further information, see the October 1995 Microcontroller Data Book, page 2-45, section titled, "Timer/Counters."

## Timer 2

Timer 2 is a 16 bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit  $C/\overline{T}2$  in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 8.

Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

Table 8. Timer 2 Operating Modes

RCLK + TCLK	CP/RL2	TR2	MODE
0	0	1	16-bit Auto-Reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

### Capture Mode

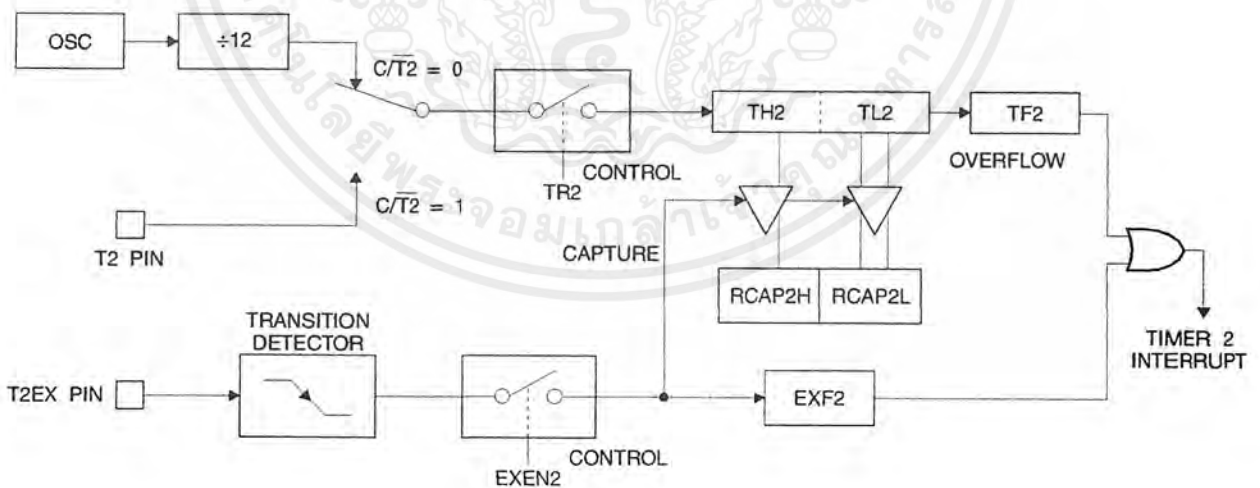
In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16 bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 1.

### Auto-Reload (Up or Down Counter)

Timer 2 can be programmed to count up or down when configured in its 16 bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 9). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to

Figure 1. Timer 2 in Capture Mode



0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16 bit value in RCAP2H and RCAP2L. The values in RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16 bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 3. In this mode, the T2EX pin controls the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16 bit value in

RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.

Figure 2. Timer 2 in Auto Reload Mode (DCEN = 0)

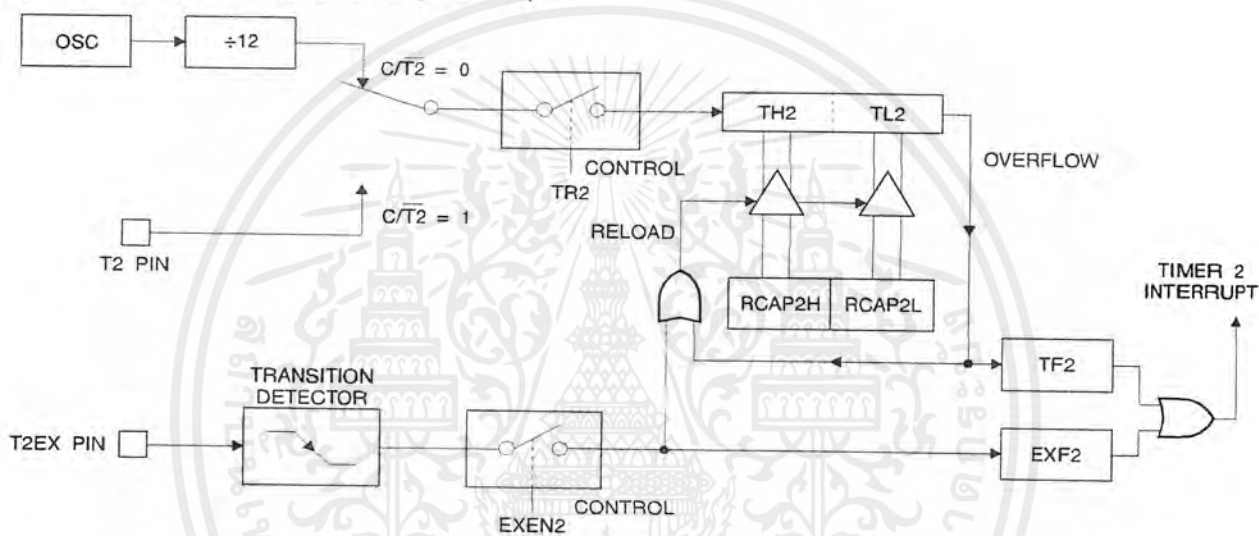


Table 9. T2MOD—Timer 2 Mode Control Register

T2MOD Address = 0C9H							Reset Value = XXXX XX00B	
Not Bit Addressable								
Bit	7	6	5	4	3	2	T2OE	DCEN

Symbol	Function
—	Not implemented, reserved for future use.
T2OE	Timer 2 Output Enable bit.
DCEN	When set, this bit allows Timer 2 to be configured as an up/down counter.



Figure 3. Timer 2 Auto Reload Mode (DCEN = 1)

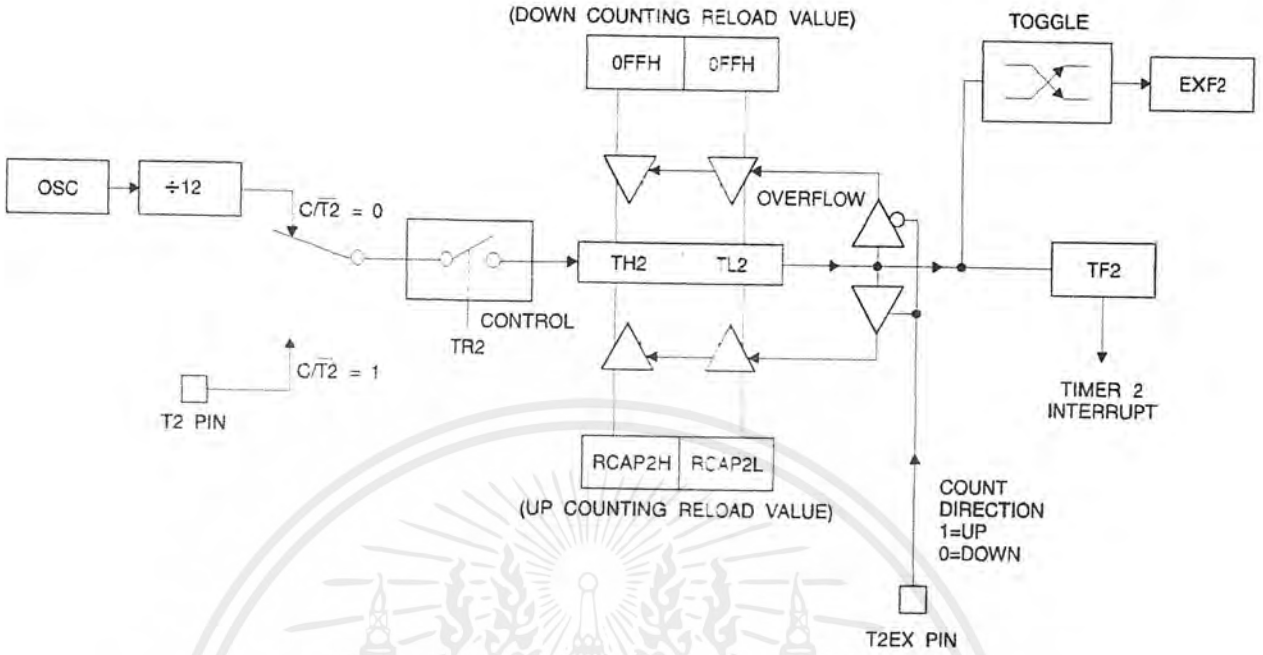
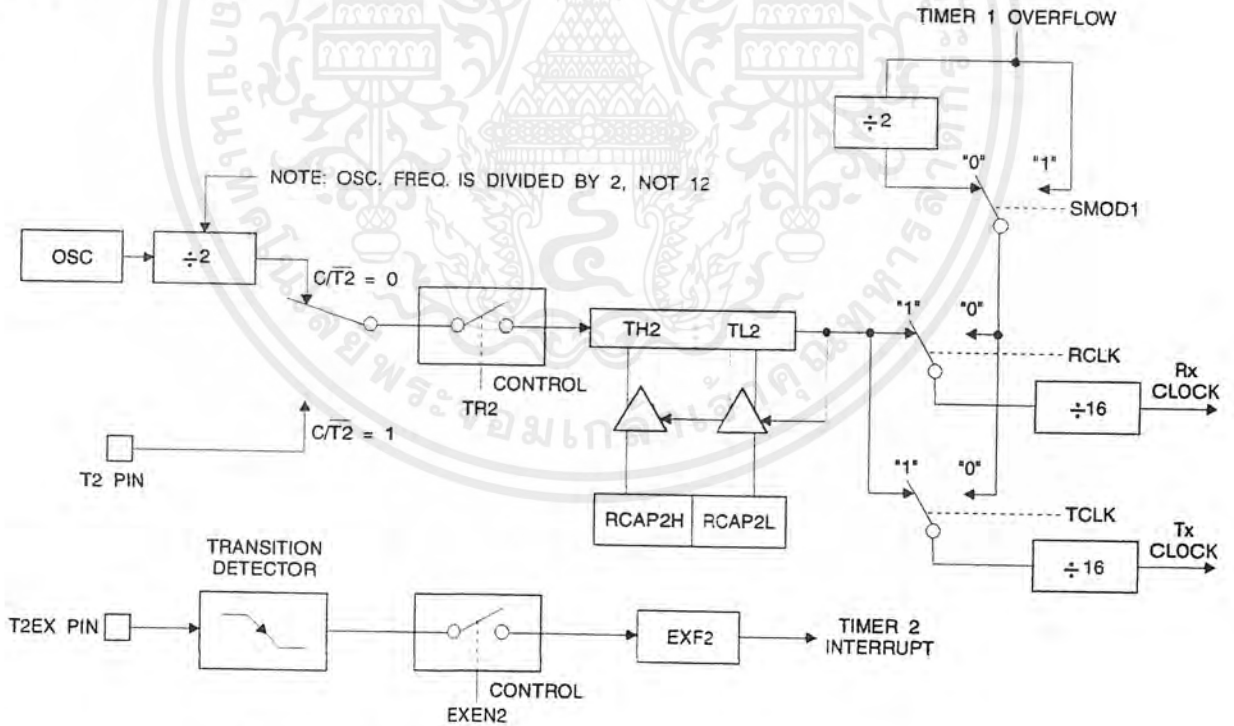


Figure 4. Timer 2 in Baud Rate Generator Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Baud Rate Generator**

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 4.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16 bit value in registers RCAP2H and RCAP2L, which are preset by software.

The baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate according to the following equation.

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation ( $CP/\overline{T2} = 0$ ). The timer operation is different for Timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

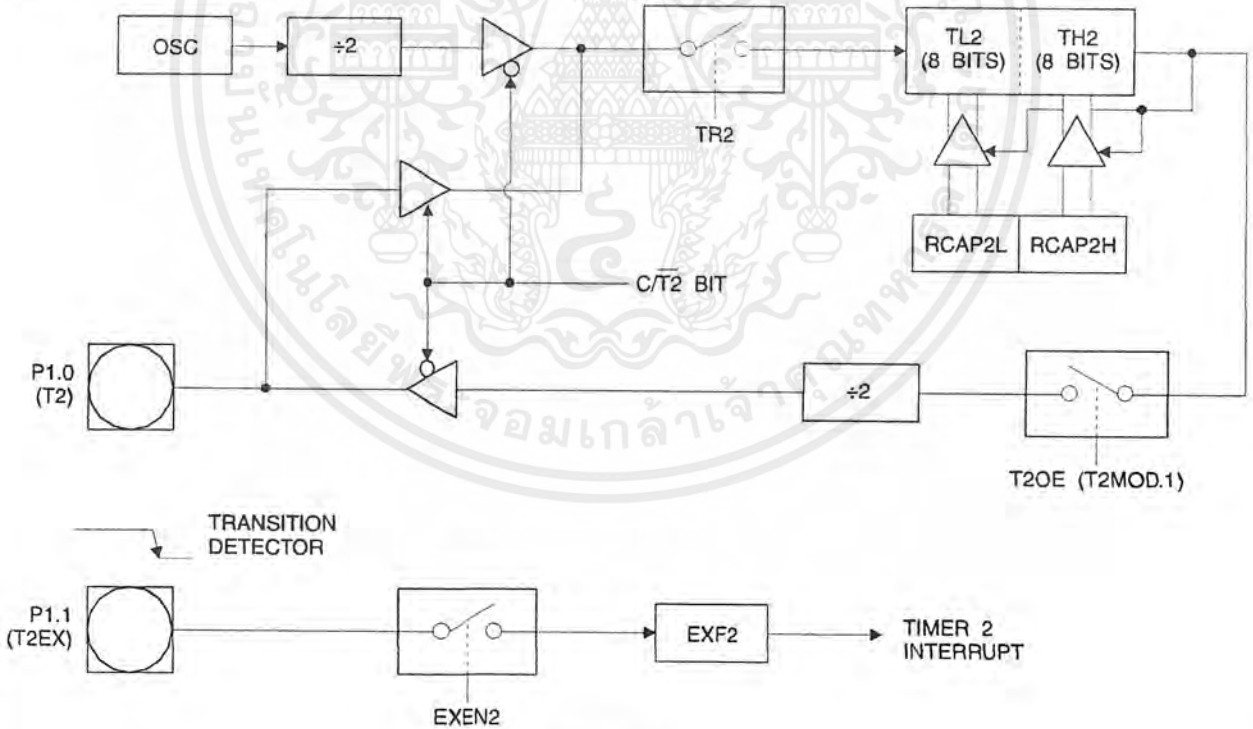
$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16 bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 4. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

Note that when Timer 2 is running ( $TR2 = 1$ ) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

Figure 5. Timer 2 in Clock-Out Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 5. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency.

To configure the Timer/Counter 2 as a clock generator, bit C/T2 (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

The clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L), as shown in the following equation.

$$\text{Clock Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]}$$

In the clock-out mode, Timer 2 rollovers will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and clock-out frequencies cannot be determined independently from one another since they both use RCAP2H and RCAP2L.

### UART

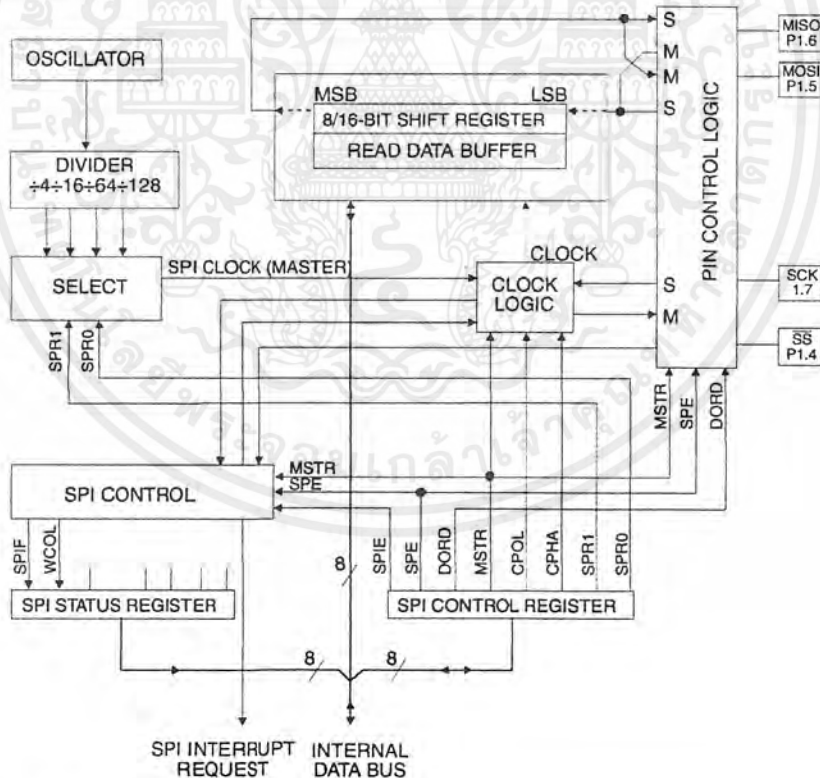
The UART in the AT89S8252 operates the same way as the UART in the AT89C51, AT89C52 and AT89C55. For further information, see the October 1995 Microcontroller Data Book, page 2-49, section titled, "Serial Interface."

### Serial Peripheral Interface

The serial peripheral interface (SPI) allows high-speed synchronous data transfer between the AT89S8252 and peripheral devices or between several AT89S8252 devices. The AT89S8252 SPI features include the following:

- Full-Duplex, 3-Wire Synchronous Data Transfer
- Master or Slave Operation
- 1.5-MHz Bit Frequency (max.)
- LSB First or MSB First Data Transfer
- Four Programmable Bit Rates
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Wakeup from Idle Mode (Slave Mode Only)

Figure 6. SPI Block Diagram



The interconnection between master and slave CPUs with SPI is shown in the following figure. The SCK pin is the clock output in the master mode but is the clock input in the slave mode. Writing to the SPI data register of the master CPU starts the SPI clock generator, and the data written shifts out of the MOSI pin and into the MOSI pin of the slave CPU. After shifting one byte, the SPI clock generator stops, setting the end of transmission flag (SPIF). If both the SPI interrupt enable bit (SPIE) and the serial port interrupt enable bit (ES) are set, an interrupt is requested.

The Slave Select input,  $\overline{SS}/P1.4$ , is set low to select an individual SPI device as a slave. When  $\overline{SS}/P1.4$  is set high, the SPI port is deactivated and the MOSI/P1.5 pin can be used as an input.

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figures 8 and 9.

Figure 7. SPI Master-Slave Interconnection

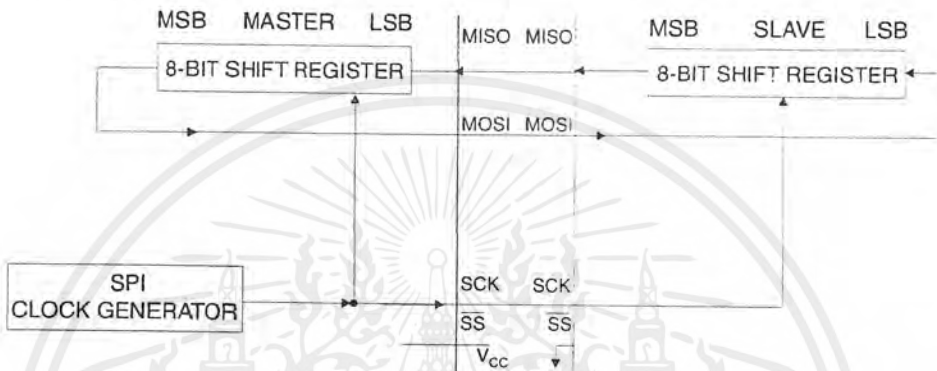
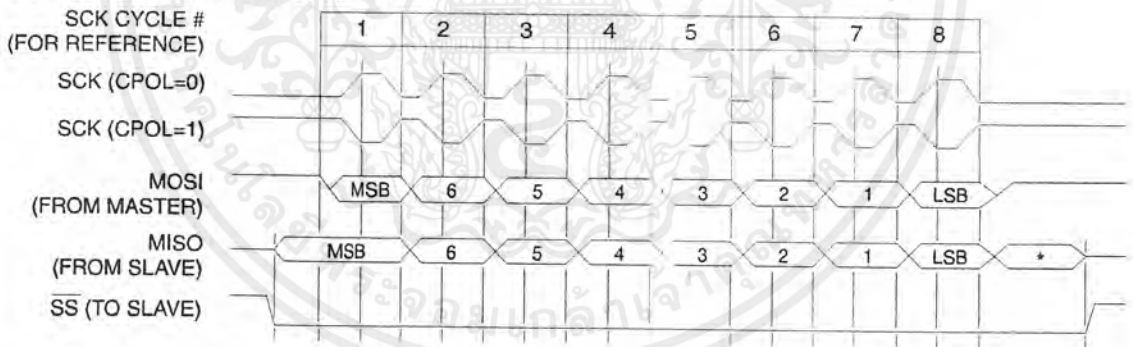


Figure 8. SPI transfer Format with CPHA = 0

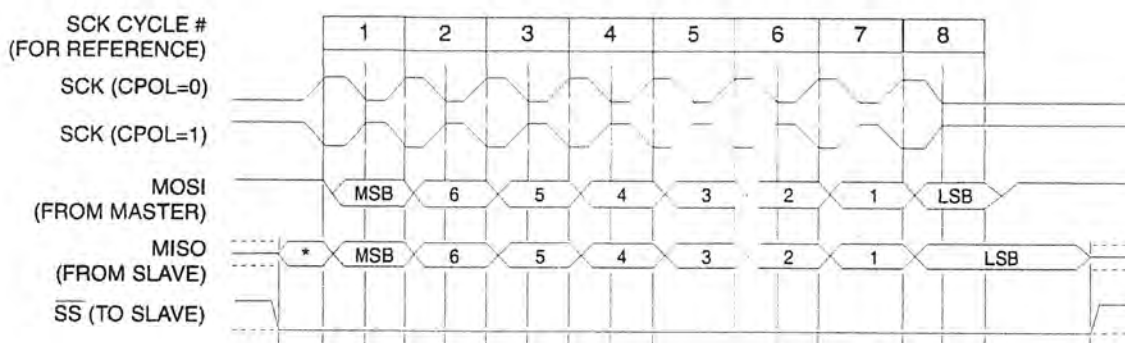


\*Not defined but normally MSB of character just received



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Figure 9.** SPI Transfer Format with CPHA = 1



\*Not defined but normally LSB of previously transmitted character

### Interrupts

The AT89S8252 has a total of six interrupt vectors: two external interrupts ( $\overline{INT0}$  and  $\overline{INT1}$ ), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 10.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once.

Note that Table 10 shows that bit position IE.6 is unimplemented. In the AT89C51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

Timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and that bit will have to be cleared in software.

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

**Table 10.** Interrupt Enable (IE) Register

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	SPI and UART interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.

User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.

**Figure 10.** Interrupt Sources

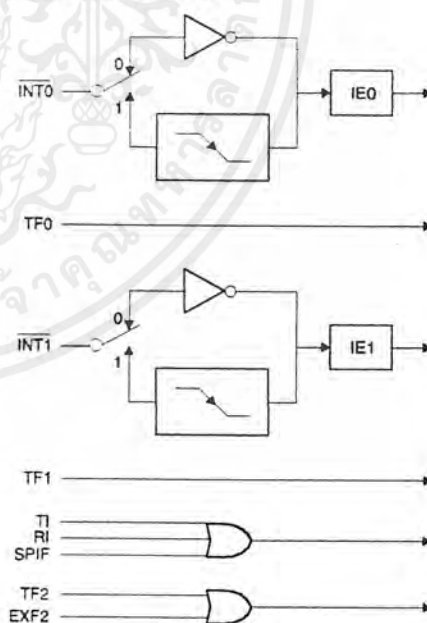


Figure 11. Oscillator Connections

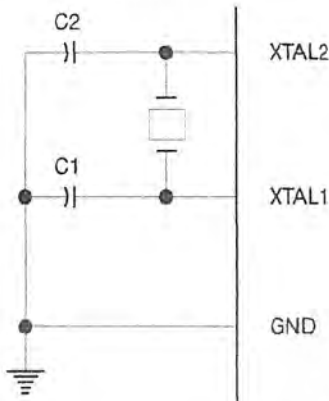
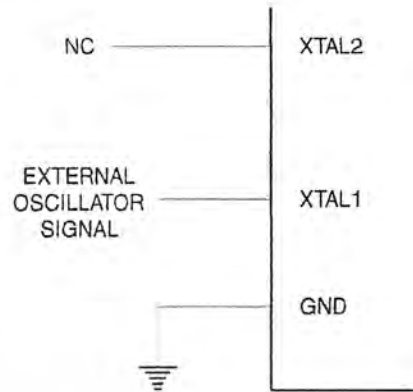


Figure 12. External Clock Drive Configuration



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

**Oscillator Characteristics**

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 11. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 12. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

**Idle Mode**

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

Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to two machine cycles before the

internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

**Power Down Mode**

In the power down mode, the oscillator is stopped and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. Exit from power down can be initiated either by a hardware reset or by an enabled external interrupt. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V<sub>CC</sub> is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

To exit power down via an interrupt, the external interrupt must be enabled as level sensitive before entering power down. The interrupt service routine starts at 16 ms (nominal) after the enabled interrupt pin is activated.

**Status of External Pins During Idle and Power Down Modes**

Mode	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data



## Program Memory Lock Bits

The AT89S8252 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

When lock bit 1 is programmed, the logic level at the  $\overline{EA}$  pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random

value and holds that value until reset is activated. The latched value of  $\overline{EA}$  must agree with the current logic level at that pin in order for the device to function properly.

Once programmed, the lock bits can only be unprogrammed with the Chip Erase operations in either the parallel or serial modes.

## Lock Bit Protection Modes<sup>(1)(2)</sup>

	Program Lock Bits			Protection Type
	LB1	LB2	LB3	
1	U	U	U	No internal memory lock feature.
2	P	U	U	MOV <sub>C</sub> instructions executed from external program memory are disabled from fetching code bytes from internal memory. $\overline{EA}$ is sampled and latched on reset and further programming of the Flash memory (parallel or serial mode) is disabled.
3	P	P	U	Same as Mode 2, but parallel or serial verify are also disabled.
4	P	P	P	Same as Mode 3, but external execution is also disabled.

Notes: 1. U = Unprogrammed  
2. P = Programmed

## Programming the Flash and EEPROM

Atmel's AT89S8252 Flash Microcontroller offers 8K bytes of in-system reprogrammable Flash Code memory and 2K bytes of EEPROM Data memory.

The AT89S8252 is normally shipped with the on-chip Flash Code and EEPROM Data memory arrays in the erased state (i.e. contents = FFH) and ready to be programmed. This device supports a High-Voltage (12V) Parallel programming mode and a Low-Voltage (5V) Serial programming mode. The serial programming mode provides a convenient way to download the AT89S8252 inside the user's system. The parallel programming mode is compatible with conventional third party Flash or EPROM programmers.

The Code and Data memory arrays are mapped via separate address spaces in the serial programming mode. In the parallel programming mode, the two arrays occupy one contiguous address space: 0000H to 1FFFH for the Code array and 2000H to 27FFH for the Data array.

The Code and Data memory arrays on the AT89S8252 are programmed byte-by-byte in either programming mode. An auto-erase cycle is provided with the self-timed programming operation in the serial programming mode. There is no need to perform the Chip Erase operation to reprogram any memory location in the serial programming mode unless any of the lock bits have been programmed.

In the parallel programming mode, there is no auto-erase cycle. To reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

## Parallel Programming Algorithm

To program and verify the AT89S8252 in the parallel programming mode, the following sequence is recommended:

- Power-up sequence:  
Apply power between  $V_{CC}$  and GND pins.  
Set RST pin to "H".  
Apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
- Set  $\overline{PSEN}$  pin to "L"  
ALE pin to "H"  
 $\overline{EA}$  pin to "H" and all other pins to "H".
- Apply the appropriate combination of "H" or "L" logic levels to pins P2.6, P2.7, P3.6, P3.7 to select one of the programming operations shown in the Flash Programming Modes table.
- Apply the desired byte address to pins P1.0 to P1.7 and P2.0 to P2.5.  
Apply data to pins P0.0 to P0.7 for Write Code operation.
- Raise  $\overline{EA}/V_{PP}$  to 12V to enable Flash programming, erase or verification.
- Pulse ALE/ $\overline{PROG}$  once to program a byte in the Code memory array, the Data memory array or the lock bits. The byte-write cycle is self-timed and typically takes 1.5 ms.
- To verify the byte just programmed, bring pin P2.7 to "L" and read the programmed data at pins P0.0 to P0.7.

8. Repeat steps 3 through 7 changing the address and data for the entire 2K or 8K bytes array or until the end of the object file is reached.
9. Power-off sequence:
  - Set XTAL1 to "L".
  - Set RST and  $\bar{E}A$  pins to "L".
  - Turn  $V_{CC}$  power off.

In the parallel programming mode, there is no auto-erase cycle and to reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

### DATA Polling

The AT89S8252 features  $\overline{DATA}$  Polling to indicate the end of a write cycle. During a write cycle in the parallel or serial programming mode, an attempted read of the last byte written will result in the complement of the written datum on P0.7 (parallel mode), and on the MSB of the serial output byte on MISO (serial mode). Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin.  $\overline{DATA}$  Polling may begin any time after a write cycle has been initiated.

### Ready/Busy

The progress of byte programming in the parallel programming mode can also be monitored by the RDY/BSY output signal. Pin P3.4 is pulled Low after ALE goes High during programming to indicate BUSY. P3.4 is pulled High again when programming is done to indicate READY.

### Program Verify

If lock bits LB1 and LB2 have not been programmed, the programmed Code or Data byte can be read back via the address and data lines for verification. The state of the lock bits can also be verified directly in the parallel programming mode. In the serial programming mode, the state of the lock bits can only be verified indirectly by observing that the lock bit features are enabled.

### Chip Erase

Both Flash and EEPROM arrays are erased electrically at the same time. In the parallel programming mode, chip erase is initiated by using the proper combination of control signals and by holding ALE/PROG low for 10 ms. The Code and Data arrays are written with all "1"s in the Chip Erase operation.

In the serial programming mode, a chip erase operation is initiated by issuing the Chip Erase instruction. In this mode, chip erase is self-timed and takes about 16 ms.

During chip erase, a serial read from any address location will return 00H at the data outputs.

### Serial Programming Fuse

A programmable fuse is available to disable Serial Programming if the user needs maximum system security. The Serial Programming Fuse can only be programmed or erased in the Parallel Programming Mode.

*The AT89S8252 is shipped with the Serial Programming Mode enabled.*

**Reading the Signature Bytes:** The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows:

- (030H) = 1EH indicates manufactured by Atmel
- (031H) = 72H indicates 89S8252

### Programming Interface

Every code byte in the Flash and EEPROM arrays can be written, and the entire array can be erased, by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

### Serial Downloading

Both the Code and Data memory arrays can be programmed using the serial SPI bus while RST is pulled to  $V_{CC}$ . The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming Enable instruction needs to be executed first before program/erase operations can be executed.

An auto-erase cycle is built into the self-timed programming operation (in the serial mode ONLY) and there is no need to first execute the Chip Erase instruction unless any of the lock bits have been programmed. The Chip Erase operation turns the content of every memory location in both the Code and Data arrays into FFH.

The Code and Data memory arrays have separate address spaces:

0000H to 1FFFFH for Code memory and 000H to 7FFFH for Data memory.

Either an external system clock is supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/40 of the crystal frequency. With a 24 MHz oscillator clock, the maximum SCK frequency is 600 kHz.



### Serial Programming Algorithm

To program and verify the AT89S8252 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence:  
Apply power between V<sub>CC</sub> and GND pins.  
Set RST pin to "H".  
If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 40.
3. The Code or Data array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. The selected memory location is first automatically erased before new data is

written. The write cycle is self-timed and typically takes less than 2.5 ms at 5V.

4. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/P1.6.
5. At the end of a programming session, RST can be set low to commence normal operation.


Power-off sequence (if needed):

- Set XTAL1 to "L" (if a crystal is not used).
- Set RST to "L".
- Turn V<sub>CC</sub> power off.

### Serial Programming Instruction

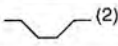
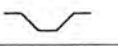
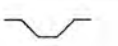


The Instruction Set for Serial Programming follows a 3-byte protocol and is shown in the following table:

### Instruction Set

Instruction	Input Format			Operation
	Byte 1	Byte 2	Byte 3	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	Enable serial programming interface after RST goes high.
Chip Erase	1010 1100	xxxx x100	xxxx xxxx	Chip erase both 8K & 2K memory arrays.
Read Code Memory	aaaa a001	low addr	xxxx xxxx	Read data from Code memory array at the selected address. The 5 MSBs of the first byte are the high order address bits. The low order address bits are in the second byte. Data are available at pin MISO during the third byte.
Write Code Memory	aaaa a010	low addr	data in	Write data to Code memory location at selected address. The address bits are the 5 MSBs of the first byte together with the second byte.
Read Data Memory	00aa a101	low addr	xxxx xxxx	Read data from Data memory array at selected address. Data are available at pin MISO during the third byte.
Write Data Memory	00aa a110	low addr	data in	Write data to Data memory location at selected address.
Write Lock Bits	1010 1100	 x x111	xxxx xxxx	Write lock bits. Set LB1, LB2 or LB3 = "0" to program lock bits.

- Notes:
1.  $\overline{DATA}$  polling is used to indicate the end of a write cycle which typically takes less than 2.5 ms at 5V.
  2. "aaaaa" = high order address.
  3. "x" = don't care.

Flash and EEPROM Parallel Programming Modes

Mode	RST	PSEN	ALE/PROG	EA/V <sub>PP</sub>	P2.6	P2.7	P3.6	P3.7	Data I/O P0.7:0	Address P2.5:0 P1.7:0
Serial Prog. Modes	H	h <sup>(1)</sup>	h <sup>(1)</sup>	x						
Chip Erase	H	L	 <sup>(2)</sup>	12V	H	L	L	L	X	X
Write (10K bytes) Memory	H	L		12V	L	H	H	H	DIN	ADDR
Read (10K bytes) Memory	H	L	H	12V	L	L	H	H	DOU <sub>T</sub>	ADDR
Write Lock Bits:	H	L		12V	H	L	H	L	DIN	X
Bit - 1									P0.7 = 0	X
Bit - 2									P0.6 = 0	X
Bit - 3									P0.5 = 0	X
Read Lock Bits:	H	L	H	12V	H	H	L	L	DOU <sub>T</sub>	X
Bit - 1									@P0.2	X
Bit - 2									@P0.1	X
Bit - 3									@P0.0	X
Read Atmel Code	H	L	H	12V	L	L	L	L	DOU <sub>T</sub>	30H
Read Device Code	H	L	H	12V	L	L	L	L	DOU <sub>T</sub>	31H
Serial Prog. Enable	H	L	 <sup>(2)</sup>	12V	L	H	L	H	P0.0 = 0	X
Serial Prog. Disable	H	L	 <sup>(2)</sup>	12V	L	H	L	H	P0.0 = 1	X
Read Serial Prog. Fuse	H	L	H	12V	H	H	L	H	@P0.0	X

- Notes:
1. "h" = weakly pulled "High" internally.
  2. Chip Erase and Serial Programming Fuse require a 10-ms PROG pulse. Chip Erase needs to be performed first before reprogramming any byte with a content other than FFH.
  3. P3.4 is pulled Low during programming to indicate RDY/BSY.
  4. "X" = don't care



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 14. Programming the Flash/EEPROM Memory

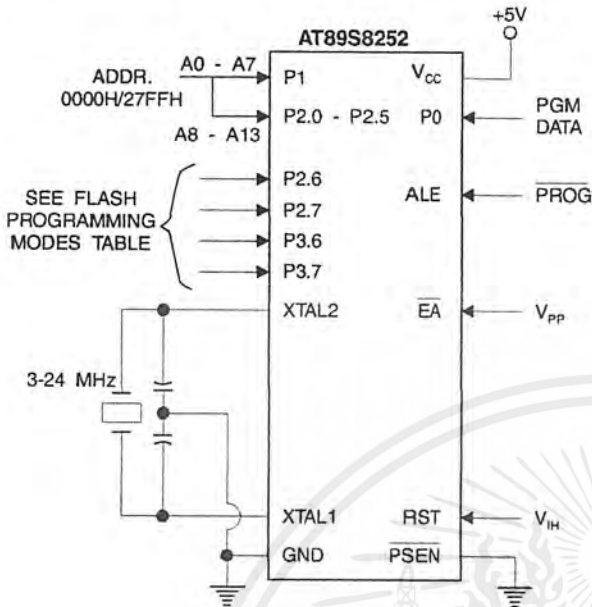


Figure 15. Flash/EEPROM Serial Downloading

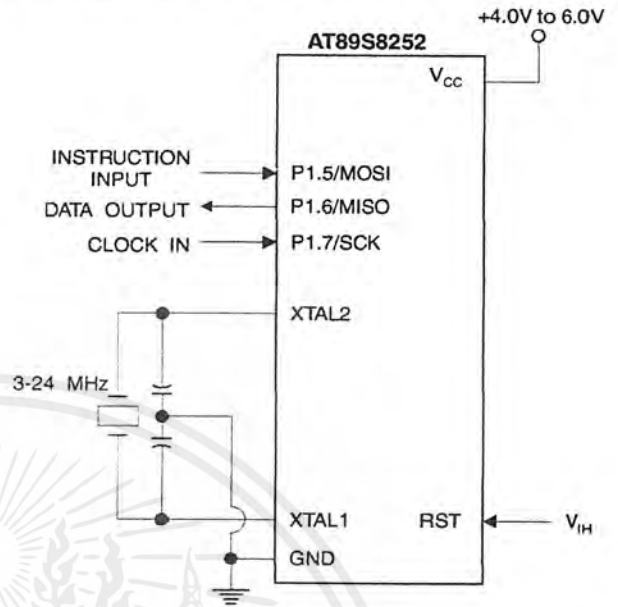
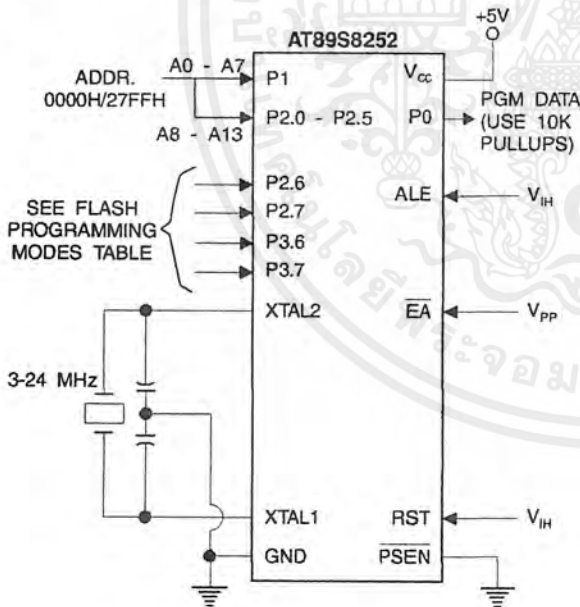


Figure 16. Verifying the Flash/EEPROM Memory



**Flash Programming and Verification Characteristics-Parallel Mode**

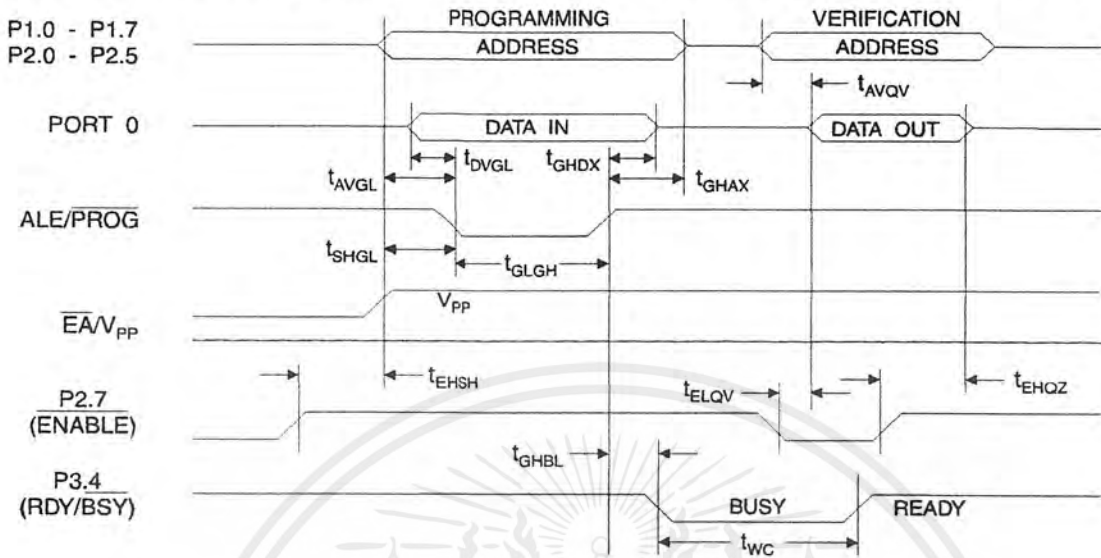
$T_A = 0^\circ\text{C}$  to  $70^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 10\%$

Symbol	Parameter	Min	Max	Units
$V_{PP}$	Programming Enable Voltage	11.5	12.5	V
$I_{PP}$	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	3	24	MHz
$t_{AVGL}$	Address Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
$t_{GHAX}$	Address Hold After $\overline{\text{PROG}}$	$48t_{CLCL}$		
$t_{DVGL}$	Data Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
$t_{GHDX}$	Data Hold After $\overline{\text{PROG}}$	$48t_{CLCL}$		
$t_{EHS}$	P2.7 ( $\overline{\text{ENABLE}}$ ) High to $V_{PP}$	$48t_{CLCL}$		
$t_{SHGL}$	$V_{PP}$ Setup to $\overline{\text{PROG}}$ Low	10		$\mu\text{s}$
$t_{GLGH}$	$\overline{\text{PROG}}$ Width	1	110	$\mu\text{s}$
$t_{AVQV}$	Address to Data Valid		$48t_{CLCL}$	
$t_{ELQV}$	$\overline{\text{ENABLE}}$ Low to Data Valid		$48t_{CLCL}$	
$t_{EHQZ}$	Data Float After $\overline{\text{ENABLE}}$	0	$48t_{CLCL}$	
$t_{GHBL}$	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		1.0	$\mu\text{s}$
$t_{WC}$	Byte Write Cycle Time		2.0	ms

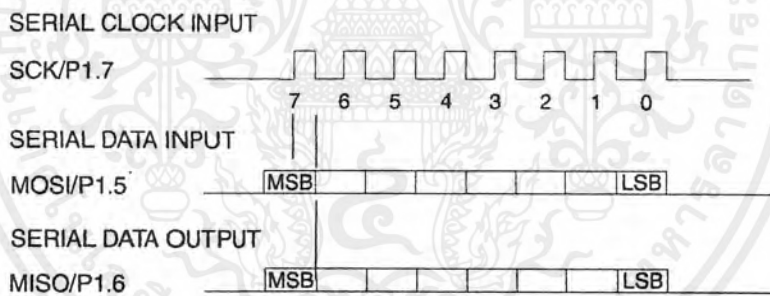


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Flash/EEPROM Programming and Verification Waveforms - Parallel Mode



## Serial Downloading Waveforms



**Absolute Maximum Ratings\***

Operating Temperature .....	-55°C to +125°C
Storage Temperature .....	-65°C to +150°C
Voltage on Any Pin with Respect to Ground .....	-1.0V to +7.0V
Maximum Operating Voltage.....	6.6V
DC Output Current.....	15.0 mA

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

**DC Characteristics**

The values shown in this table are valid for  $T_A = -40^\circ\text{C}$  to  $85^\circ\text{C}$  and  $V_{CC} = 5.0\text{V} \pm 20\%$ , unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
$V_{IL}$	Input Low Voltage	(Except $\overline{EA}$ )	-0.5	$0.2 V_{CC} - 0.1$	V
$V_{IL1}$	Input Low Voltage ( $\overline{EA}$ )		-0.5	$0.2 V_{CC} - 0.3$	V
$V_{IH}$	Input High Voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
$V_{IH1}$	Input High Voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
$V_{OL}$	Output Low Voltage <sup>(1)</sup> (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.5	V
$V_{OL1}$	Output Low Voltage <sup>(1)</sup> (Port 0, ALE, PSEN)	$I_{OL} = 3.2 \text{ mA}$		0.5	V
$V_{OH}$	Output High Voltage (Ports 1,2,3, ALE, PSEN)	$I_{OH} = -60 \mu\text{A}, V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -25 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -10 \mu\text{A}$	$0.9 V_{CC}$		V
$V_{OH1}$	Output High Voltage (Port 0 in External Bus Mode)	$I_{OH} = -800 \mu\text{A}, V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -300 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -80 \mu\text{A}$	$0.9 V_{CC}$		V
$I_{IL}$	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45\text{V}$		-50	$\mu\text{A}$
$I_{TL}$	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{V}, V_{CC} = 5\text{V} \pm 10\%$		-650	$\mu\text{A}$
$I_{LI}$	Input Leakage Current (Port 0, EA)	$0.45 < V_{IN} < V_{CC}$		$\pm 10$	$\mu\text{A}$
RRST	Reset Pulldown Resistor		50	300	$\text{K}\Omega$
$C_{IO}$	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
$I_{CC}$	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power Down Mode <sup>(2)</sup>	$V_{CC} = 6\text{V}$		100	$\mu\text{A}$
		$V_{CC} = 3\text{V}$		40	$\mu\text{A}$

Notes: 1. Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:  
 Maximum  $I_{OL}$  per port pin: 10 mA  
 Maximum  $I_{OL}$  per 8-bit port:  
 Port 0: 26 mA  
 Ports 1, 2, 3: 15 mA

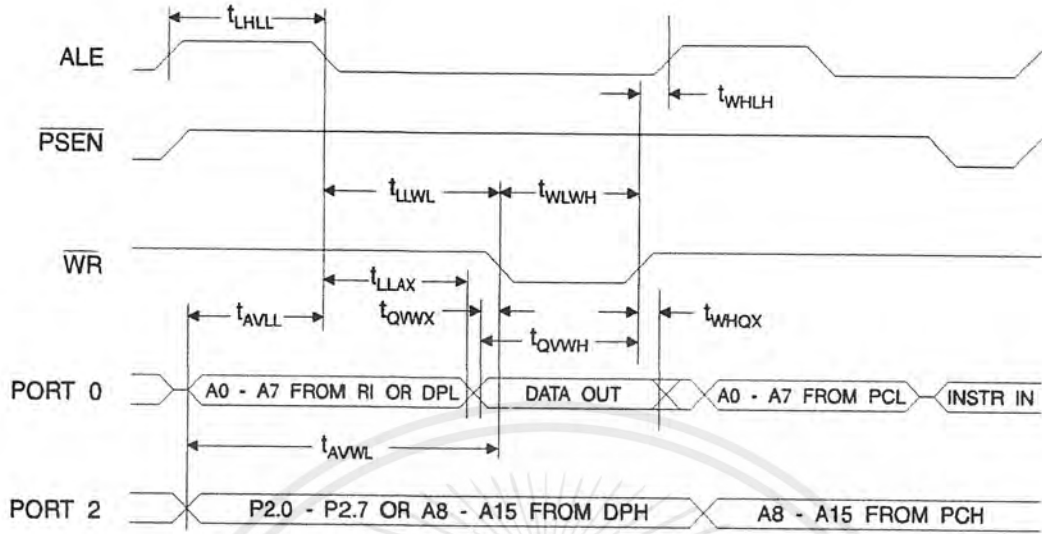
Maximum total  $I_{OL}$  for all output pins: 71 mA  
 If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum  $V_{CC}$  for Power Down is 2V

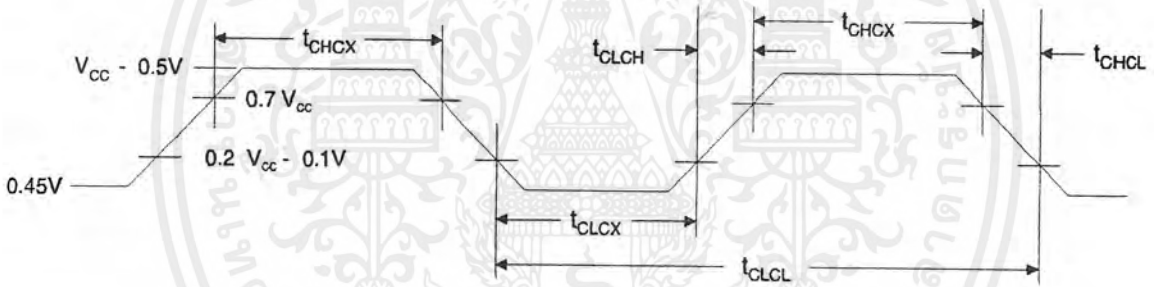


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### External Data Memory Write Cycle



### External Clock Drive Waveforms



### External Clock Drive

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

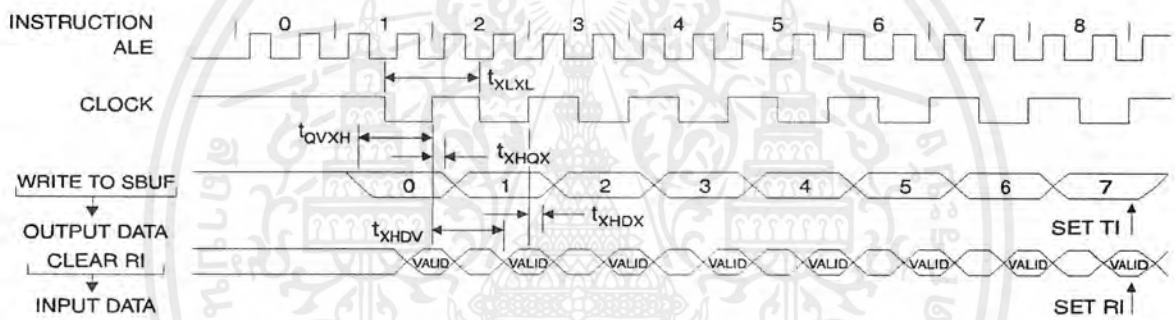
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Serial Port Timing: Shift Register Mode Test Conditions**

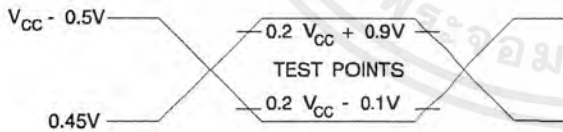
The values in this table are valid for  $V_{CC} = 4.0V$  to  $6V$  and Load Capacitance =  $80\text{ pF}$ .

Symbol	Parameter	Variable Oscillator		Units
		Min	Max	
$t_{XLXL}$	Serial Port Clock Cycle Time	$12t_{CLCL}$		$\mu s$
$t_{QVXH}$	Output Data Setup to Clock Rising Edge	$10t_{CLCL} - 133$		ns
$t_{XHQX}$	Output Data Hold After Clock Rising Edge	$2t_{CLCL} - 117$		ns
$t_{XHDX}$	Input Data Hold After Clock Rising Edge	0		ns
$t_{XHDV}$	Clock Rising Edge to Input Data Valid		$10t_{CLCL} - 133$	ns

**Shift Register Mode Timing Waveforms**

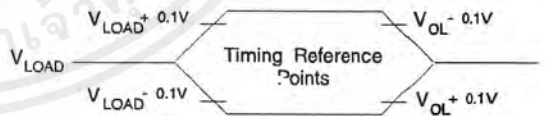


**AC Testing Input/Output Waveforms<sup>(1)</sup>**



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

**Float Waveforms<sup>(1)</sup>**



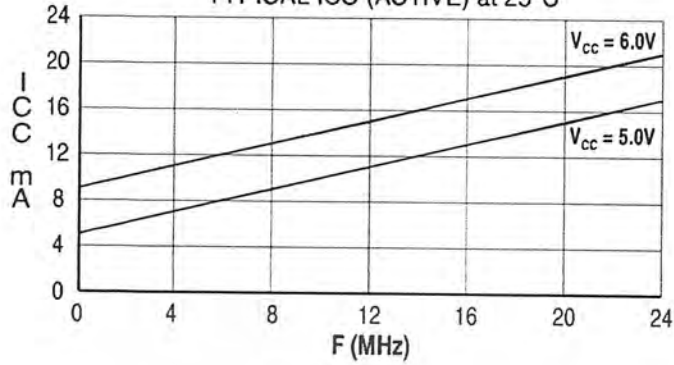
Notes: 1. For timing purposes, a port pin is no longer floating when a  $100\text{ mV}$  change from load voltage occurs. A port pin begins to float when a  $100\text{ mV}$  change from the loaded  $V_{OH}/V_{OL}$  level occurs.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

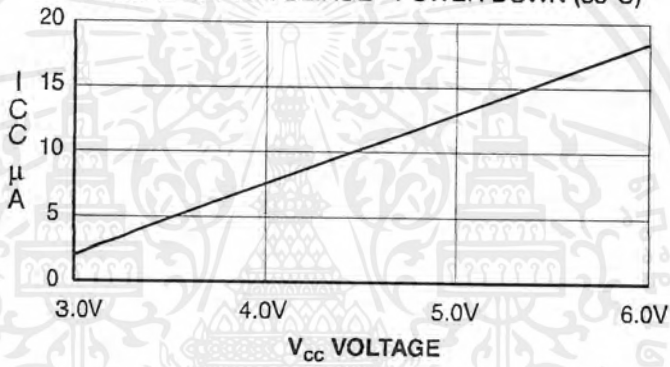
### AT89S8252

TYPICAL ICC (ACTIVE) at 25°C



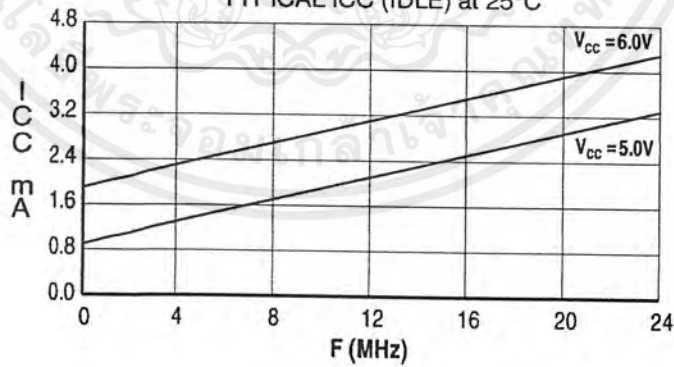
### AT89S8252

TYPICAL ICC vs. VOLTAGE - POWER DOWN (85°C)



### AT89S8252


TYPICAL ICC (IDLE) at 25°C

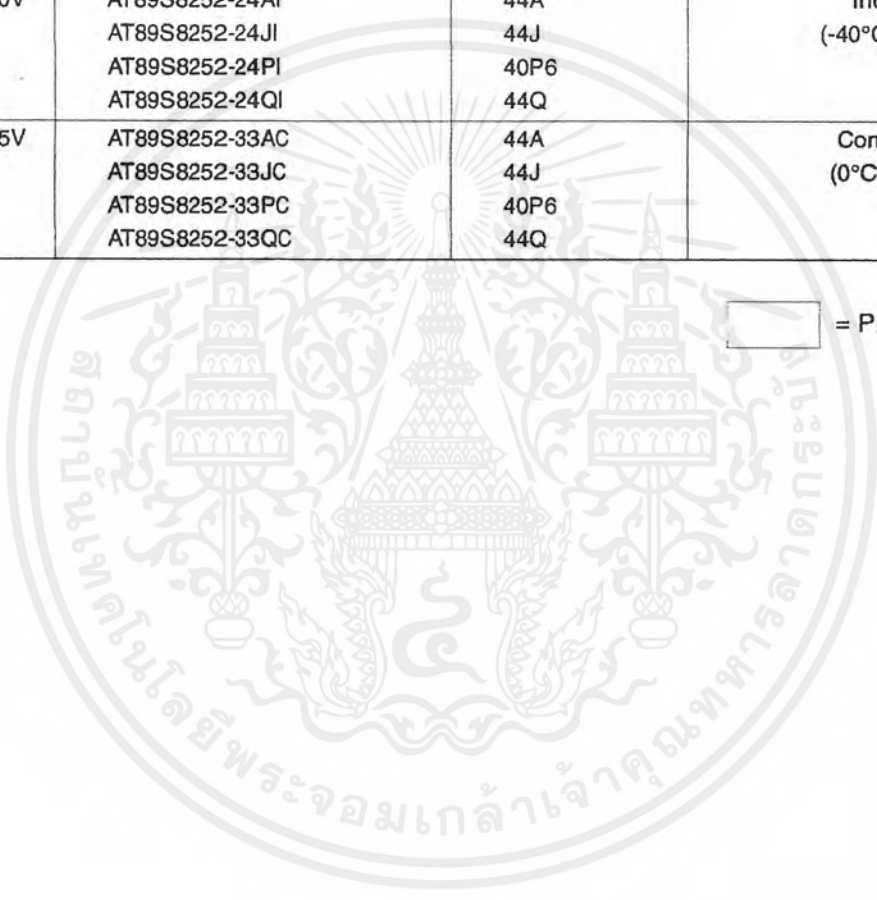


- Notes:
1. XTAL1 tied to GND for I<sub>cc</sub> (power down)
  2. Lock bits programmed

## Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
16	4.0V to 6.0V	AT89S8252-16AA	44A	Automotive (-40°C to 105°C)
		AT89S8252-16JA	44J	
		AT89S8252-16PA	40P6	
		AT89S8252-16QA	44Q	
24	4.0V to 6.0V	AT89S8252-24AC	44A	Commercial (0°C to 70°C)
		AT89S8252-24JC	44J	
		AT89S8252-24PC	40P6	
		AT89S8252-24QC	44Q	
	4.0V to 6.0V	AT89S8252-24AI	44A	Industrial (-40°C to 85°C)
		AT89S8252-24JI	44J	
		AT89S8252-24PI	40P6	
		AT89S8252-24QI	44Q	
33	4.5V to 5.5V	AT89S8252-33AC	44A	Commercial (0°C to 70°C)
		AT89S8252-33JC	44J	
		AT89S8252-33PC	40P6	
		AT89S8252-33QC	44Q	

 = Preliminary Information



Package Type	
<b>44A</b>	44 Lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
<b>44J</b>	44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
<b>40P6</b>	40 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
<b>44Q</b>	44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## MM54HC573/MM74HC573 TRI-STATE® Octal D-Type Latch

### General Description

These high speed octal D-type latches utilize advanced silicon-gate P-well CMOS technology. They possess the high noise immunity and low power consumption of standard CMOS integrated circuits, as well as the ability to drive 15 LS-TTL loads. Due to the large output drive capability and the TRI-STATE feature, these devices are ideally suited for interfacing with bus lines in a bus organized system.

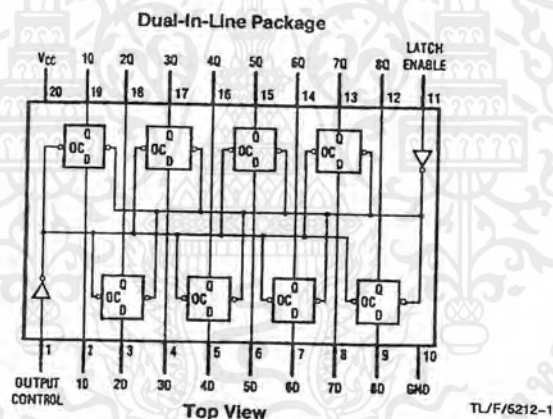
When the LATCH ENABLE(LE) input is high, the Q outputs will follow the D inputs. When the LATCH ENABLE goes low, data at the D inputs will be retained at the outputs until LATCH ENABLE returns high again. When a high logic level is applied to the OUTPUT CONTROL OC input, all outputs go to a high impedance state, regardless of what signals are present at the other inputs and the state of the storage elements.

The 54HC/74HC logic family is speed, function and pinout compatible with the standard 54LS/74LS logic family. All inputs are protected from damage due to static discharge by internal diode clamps to  $V_{CC}$  and ground.

### Features

- Typical propagation delay: 18 ns
- Wide operating voltage range: 2 to 6 volts
- Low input current: 1  $\mu$ A maximum
- Low quiescent current: 80  $\mu$ A maximum (74HC Series)
- Compatible with bus-oriented systems
- Output drive capability: 15 LS-TTL loads

### Connection Diagram



Order Number MM54HC573 or MM74HC573  
54HCT573 (J) 74HCT573 (N, WM)

### Truth Table

Output Control	Latch Enable	Data	Output
L	H	H	H
L	H	L	L
L	L	X	$Q_0$
H	X	X	Z

H = high level, L = low level  
 $Q_0$  = level of output before steady-state input conditions were established.  
 Z = high impedance  
 X = Don't care

TRI-STATE® is a registered trademark of National Semiconductor Corporation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Absolute Maximum Ratings (Notes 1 & 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage ( $V_{CC}$ )	-0.5 to +7.0V
DC Input Voltage ( $V_{IN}$ )	-1.5 to $V_{CC} + 1.5V$
DC Output Voltage ( $V_{OUT}$ )	-0.5 to $V_{CC} + 0.5V$
Clamp Diode Current ( $I_{IK}, I_{OK}$ )	$\pm 20$ mA
DC Output Current, per pin ( $I_{OUT}$ )	$\pm 35$ mA
DC $V_{CC}$ or GND Current, per pin ( $I_{CC}$ )	$\pm 70$ mA
Storage Temperature ( $T_{STG}$ )	-65°C to +150°C
Power Dissipation ( $P_D$ ) (Note 3)	600 mW
S.O. Package only	500 mW
Lead Temperature ( $T_L$ ) (Soldering 10 seconds)	260°C

### Operating Conditions

	Min	Max	Units
Supply Voltage ( $V_{CC}$ )	2	6	V
DC Input or Output Voltage ( $V_{IN}, V_{OUT}$ )	0	$V_{CC}$	V
Operating Temp. Range ( $T_A$ )			
MM74HC	-40	+85	°C
MM54HC	-55	+125	°C
Input Rise or Fall Times ( $t_r, t_f$ )			
$V_{CC} = 2.0V$		1000	ns
$V_{CC} = 4.5V$		500	ns
$V_{CC} = 6.0V$		400	ns

### DC Electrical Characteristics (Note 4)

Symbol	Parameter	Conditions	$V_{CC}$	$T_A = 25^\circ C$			74HC	54HC	Units
				Typ		$T_A = -40$ to $85^\circ C$	$T_A = -55$ to $125^\circ C$		
$V_{IH}$	Minimum High Level Input Voltage		2.0V	1.5		1.5	1.5	V	
			4.5V	3.15		3.15	3.15	V	
			6.0V	4.2		4.2	4.2	V	
$V_{IL}$	Maximum Low Level Input Voltage**		2.0V	0.5		0.5	0.5	V	
			4.5V	1.35		1.35	1.35	V	
			6.0V	1.8		1.8	1.8	V	
$V_{OH}$	Minimum High Level Output Voltage	$V_{IN} = V_{IH}$ or $V_{IL}$ $ I_{OUT}  \leq 20 \mu A$	2.0V	2.0	1.9	1.9	1.9	V	
			4.5V	4.5	4.4	4.4	4.4	V	
			6.0V	6.0	5.9	5.9	5.9	V	
		$V_{IN} = V_{IH}$ or $V_{IL}$ $ I_{OUT}  \leq 6.0$ mA $ I_{OUT}  \leq 7.8$ mA	4.5V	4.2	3.98	3.84	3.7	V	
			6.0V	5.7	5.48	5.34	5.2	V	
$V_{OL}$	Maximum Low Level Output Voltage	$V_{IN} = V_{IH}$ or $V_{IL}$ $ I_{OUT}  \leq 20 \mu A$	2.0V	0	0.1	0.1	0.1	V	
			4.5V	0	0.1	0.1	0.1	V	
			6.0V	0	0.1	0.1	0.1	V	
		$V_{IN} = V_{IH}$ or $V_{IL}$ $ I_{OUT}  \leq 6.0$ mA $ I_{OUT}  \leq 7.8$ mA	4.5V	0.2	0.26	0.33	0.4	V	
			6.0V	0.2	0.26	0.33	0.4	V	
$I_{IN}$	Maximum Input Current	$V_{IN} = V_{CC}$ or GND	6.0V	$\pm 0.1$		$\pm 1.0$	$\pm 1.0$	$\mu A$	
$I_{OZ}$	Maximum TRI-STATE Output Leakage Current	$V_{OUT} = V_{CC}$ or GND $O_C = V_{IH}$	6.0V	$\pm 0.5$		$\pm 5.0$	$\pm 10$	$\mu A$	
$I_{CC}$	Maximum Quiescent Supply Current	$V_{IN} = V_{CC}$ or GND $I_{OUT} = 0 \mu A$	6.0V	8.0		80	160	$\mu A$	
$\Delta I_{CC}$	Quiescent Supply Current per Input Pin	$V_{CC} = 5.5V$ $V_{IN} = 2.4V$ or 0.4V (Note 4)	OE	1.0	1.5	1.8	2.0	mA	
			LE	0.6	0.8	1.0	1.1	mA	
			DATA	0.4	0.5	0.6	0.7	mA	

Note 1: Absolute Maximum Ratings are those values beyond which damage to the device may occur.

Note 2: Unless otherwise specified all voltages are referenced to ground.

Note 3: Power Dissipation temperature derating — plastic "N" package: -12 mW/°C from 65°C to 85°C; ceramic "J" package: -12 mW/°C from 100°C to 125°C.

Note 4: For a power supply of  $5V \pm 10\%$  the worst-case output voltages ( $V_{OH}$  and  $V_{OL}$ ) occur for HC at 4.5V. Thus the 4.5V values should be used when designing with this supply. Worst-case  $V_{IH}$  and  $V_{IL}$  occur at  $V_{CC} = 5.5V$  and 4.5V respectively. (The  $V_{IH}$  value at 5.5V is 3.85V.) The worst-case leakage current ( $I_{IN}$ ,  $I_{CC}$ , and  $I_{OZ}$ ) occur for CMOS at the higher voltage and so the 6.0V values should be used.

\*\* $V_{IL}$  limits are currently tested at 20% of  $V_{CC}$ . The above  $V_{IL}$  specification (30% of  $V_{CC}$ ) will be implemented no later than Q1, CY'89.

### AC Electrical Characteristics $V_{CC}=5V, T_A=25^\circ C, t_r=t_f=6\text{ ns}$

Symbol	Parameter	Conditions	Typ	Guaranteed Limit	Units
$t_{PHL}, t_{PLH}$	Maximum Propagation Delay, Data to Q	$C_L = 45\text{ pF}$	17	27	ns
$t_{PHL}, t_{PLH}$	Maximum Propagation Delay, LE to Q	$C_L = 45\text{ pF}$	16	27	ns
$t_{PZH}, t_{PZL}$	Maximum Output Enable Time	$R_L = 1\text{ k}\Omega$ $C_L = 45\text{ pF}$	21	30	ns
$t_{PHZ}, t_{PLZ}$	Maximum Output Disable Time	$R_L = 1\text{ k}\Omega$ $C_L = 5\text{ pF}$	14	23	ns
$t_S$	Minimum Set Up Time, Data to LE		25	5	ns
$t_H$	Minimum Hold Time, LE to Data		2	12	ns
$t_W$	Minimum Pulse Width, LE or Data		10	15	ns

### AC Electrical Characteristics

Symbol	Parameter	Conditions	$V_{CC}$	$T_A = 25^\circ C$		74HC $T_A = -40\text{ to }85^\circ C$		54HC $T_A = -55\text{ to }125^\circ C$		Units
				Typ	Guaranteed Limits	Typ	Guaranteed Limits	Typ	Guaranteed Limits	
$t_{PHL}, t_{PLH}$	Maximum Propagation Delay Data to Q	$C_L = 50\text{ pF}$ $C_L = 150\text{ pF}$	2.0V	18	30	38	45	225	ns	
			2.0V	58	150	188	225	ns		
		$C_L = 50\text{ pF}$ $C_L = 150\text{ pF}$	4.5V	14	22	28	33	40	ns	
			4.5V	21	30	38	40	ns		
$t_{PHL}, t_{PLH}$	Maximum Propagation Delay, LE to Q	$C_L = 50\text{ pF}$ $C_L = 150\text{ pF}$	6.0V	12	19	24	29	39	ns	
			6.0V	19	26	33	39	ns		
		$C_L = 50\text{ pF}$ $C_L = 150\text{ pF}$	2.0V	17	30	38	45	233	ns	
			2.0V	60	155	194	233	ns		
$t_{PZH}, t_{PZL}$	Maximum Output Enable Time	$R_L = 1\text{ k}\Omega$ $C_L = 50\text{ pF}$ $C_L = 150\text{ pF}$	4.5V	14	23	29	35	47	ns	
			4.5V	21	31	47	47	ns		
		$C_L = 50\text{ pF}$ $C_L = 150\text{ pF}$	6.0V	12	20	25	30	41	ns	
			6.0V	19	27	34	41	ns		
$t_{PHZ}, t_{PLZ}$	Maximum Output Disable Time	$R_L = 1\text{ k}\Omega$ $C_L = 50\text{ pF}$	2.0V	15	30	38	45	32	ns	
			4.5V	13	25	31	38	32	ns	
		$C_L = 50\text{ pF}$ $C_L = 150\text{ pF}$	6.0V	12	21	27	32	32	ns	
			6.0V	13	21	27	32	32	ns	
$t_S$	Minimum Set Up Time Data to LE		2.0V	-3	5	6	8	ns		
			4.5V	10	15	19	22	ns		
			6.0V	9	13	16	19	ns		
$t_H$	Minimum Hold Time LE to Data		2.0V	4	12	15	18	ns		
			4.5V		5	6	7	ns		
			6.0V		4	5	6	ns		
$t_W$	Minimum Pulse Width LE, or Data		2.0V	30	15	20	24	ns		
			4.5V	9	16	20	24	ns		
			6.0V	8	14	18	20	ns		
$t_{TLH}, t_{THL}$	Maximum Output Rise and Fall Time, Clock	$C_L = 50\text{ pF}$	2.0V	6	12	15	18	ns		
			4.5V	7	12	15	18	ns		
			6.0V	6	10	13	15	ns		
$C_{PD}$	Power Dissipation Capacitance (Note 5) (per latch)	$OC = V_{CC}$ $OC = GND$		5				pF		
$C_{IN}$	Maximum Input Capacitance			10	10	10		pF		
$C_{OUT}$	Maximum Output Capacitance			20	20	20		pF		

Note 5:  $C_{PD}$  determines the no load dynamic power consumption,  $P_D = C_{PD} V_{CC}^2 f + I_{CC} V_{CC}$ , and the no load dynamic current consumption,  $I_S = C_{PD} V_{CC} f + I_{CC}$ .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## บรรณานุกรม

- ธีรวัฒน์ ประกอบผล. 2542. การประยุกต์ใช้งานไมโครคอนโทรลเลอร์. กรุงเทพฯ : สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น)
- อุดม จินประดับ. 2541. ไมโครคอนโทรลเลอร์ MCS-51. กรุงเทพฯ : สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ
- มนตรี สุวรรณกิจการ. 2537. เทคโนโลยีไฟฟ้า. กรุงเทพฯ : สถาบันเทคโนโลยีพระจอมเกล้าธนบุรี
- มนตรี สุวรรณกิจการ. 2530. เครื่องกลไฟฟ้า. กรุงเทพฯ : สถาบันเทคโนโลยีพระจอมเกล้าธนบุรี
- สัมพันธ์ หาญชล. 2538. เครื่องกลไฟฟ้า. กรุงเทพฯ : สถาบันเทคโนโลยีพระจอมเกล้าธนบุรี
- มานพ ดันตระบัณฑิตย์. 2542. งานซ่อมบำรุงชิ้นส่วนเครื่องจักรกล. กรุงเทพฯ : สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น)
- มานพ ดันตระบัณฑิตย์ , สำลี แสงห้าว และสุทิน จิตร์เจริญ. 2542. ชิ้นส่วนเครื่องจักรกล. กรุงเทพฯ : สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น)
- [www.hth.com/filelibrary/pdffiles/AT89s8252.pdf](http://www.hth.com/filelibrary/pdffiles/AT89s8252.pdf)