

# สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ปีการศึกษา 2541

การควบคุมตำแหน่งของเซอร์โวมอเตอร์กระแสตรงโดยไมโครโปรเซสเซอร์  
MICROPROCESSOR CONTROL OF DC SERVO MOTOR FOR POSITION CONTROL



โดย  
นายสมหมาย ลักศิธรรม  
นายสุทธิชัย อินนาวา  
นายอารีย์ หวังสุภผล

อาจารย์ที่ปรึกษา

อาจารย์ ดร.วิจิตร กิณรศ

อาจารย์อนุวัฒน์ จางวนิชเลิศ

เลขหน้.....  
เลขทะเบียน..... 34167  
วัน, เดือน, ปี..... 6 ต.ค. 2542

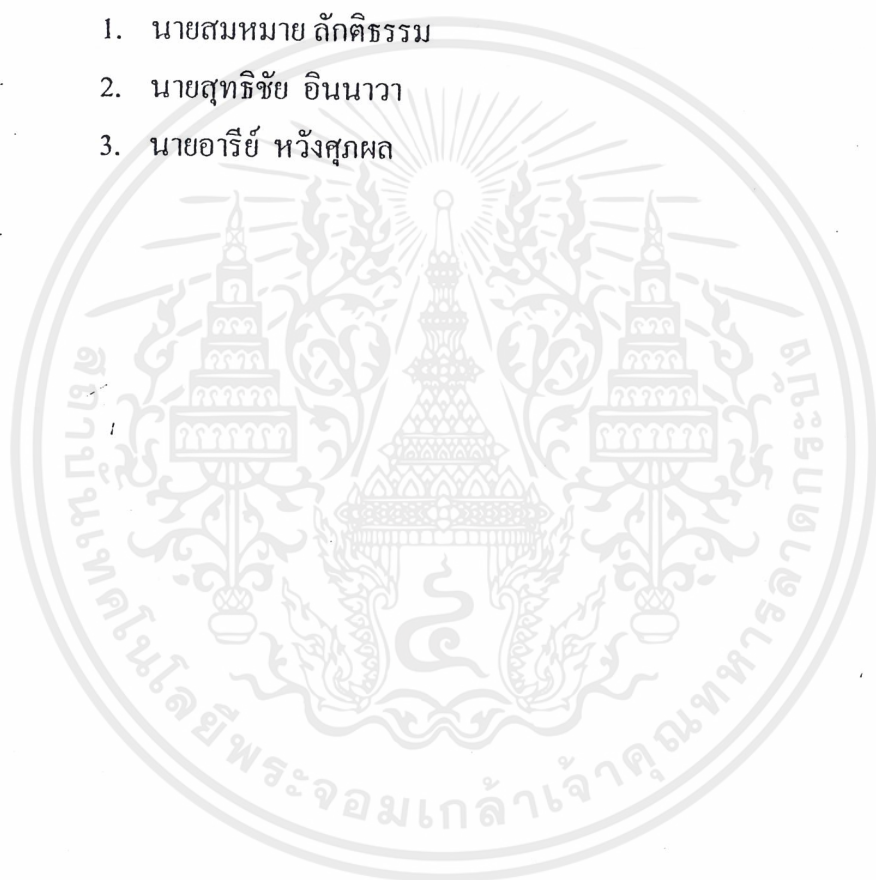
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2541

ภาควิชา วิศวกรรมไฟฟ้า

คณะ วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
เรื่อง การควบคุมตำแหน่งของเซอร์โวมอเตอร์กระแสตรงโดยไมโครโปรเซสเซอร์  
ผู้จัดทำ

1. นายสมหมาย ลักศิธรรม
2. นายสุทธิชัย อินนาวา
3. นายอารีย์ หวังสุภผล



*[Handwritten signature]*

.....อาจารย์ที่ปรึกษา

(ดร. วิจิตร กิณเรศ)

*[Handwritten signature]*

.....อาจารย์ที่ปรึกษา

(อ.อนุวัฒน์ จางวนิชเลิศ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมตำแหน่งของเซอร์โวมอเตอร์กระแสตรงโดยไมโครโปรเซสเซอร์

นายสมหมาย ลักติธรรม

นายสุทธิชัย อินนาวา

นายอารีย์ หวังสุภผล

ดร.วิจิตร กิณเรศ อาจารย์ที่ปรึกษา

อ.อนุวัฒน์ จางวนิชเลิศ อาจารย์ที่ปรึกษา

ปีการศึกษา 2541

บทคัดย่อ

โครงการนี้เป็นการศึกษาวิธีการควบคุมตำแหน่งของดีซีเซอร์โวมอเตอร์ โดยใช้การควบคุมแบบป้อนกลับและใช้ไมโครคอนโทรลเลอร์ควบคุมการทำงานทั้งหมด การขับมอเตอร์ใช้เทคนิคพีดับบลิวเอ็ม-เซอร์โวแอมพลิฟายเออร์ โดยสร้างสัญญาณพีดับบลิวเอ็มจากชิพสนับสนุน 8254 (PIT) วงจรกำลังวางตัวในลักษณะ H-Bridge ข้อดีของเทคนิคการขับและวงจรถูกตั้งแบบนี้เป็นคือ การใช้พีดับบลิวเอ็มจะลดกำลังสูญเสียบนอุปกรณ์สวิทซ์ เป็นการป้องกันการเสียหายจากความร้อนที่เกิดขึ้นบนอุปกรณ์ เมื่อเปรียบเทียบกับการขับแบบลิเนียร์แอมพลิฟายเออร์ ทำให้ประสิทธิภาพของวงจรสูงขึ้น สำหรับวงจรกำลังสามารถทำงานได้ 4-ควอดแรนต์ และใช้เพียงแหล่งจ่ายเดียวในการทำงานในส่วนป้อนกลับสัญญาณพัลส์จากเอ็นโค้ดเดอร์จะถูกนำมาประมวลผลเพื่อควบคุมตำแหน่ง และจากการทดลองได้ผลลัพธ์เป็นที่น่าพอใจ

## MICROPROCESSOR CONTROL OF DC SERVO MOTOR FOR POSITION CONTROL

Somma Lucktithum

Suttichai Innawa

Aree Wangsupphaphol

Dr.Vijit Kinnarase Advisor

Anuwat Jangwanitlert Advisor

1998

### ABSTRACT

This thesis presents a manner to control the position of DC-servo motor on the principle of feedback control. The microcontroller is employed to deal with this scheme. The PWM servo-amplifiers are used for driving motor. The PWM waveforms are generated by IC 8254 (PIT). The presence of power circuit is the H-bridge scheme. The merits of PWM technique and this power circuit are strategies to reduce the power loss in switching devices and protect them from thermal damage, when comparing to a conventional linear amplifier. Therefore it improves the efficiency, the power circuit can operate as four-quadrants and employ a power supply. The pulse from encoder is processed to control the position. The tests have been performed. The results are satisfactory.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
สารบัญ	I
สารบัญภาพ	III
สารบัญตาราง	V
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีที่ใช้ในการออกแบบ	3
2.1 เซอร์โวมอเตอร์	3
2.1.1 เซอร์โวมอเตอร์ไฟฟ้ากระแสสลับ	3
2.1.2 เซอร์โวมอเตอร์ไฟฟ้ากระแสตรง	5
2.2 เซ็นเซอร์และเอ็นโค้ดเดอร์	10
2.2.1 ทาโคยอนเนอร์เรเตอร์	10
2.2.2 เอ็นโค้ดเดอร์	10
2.3 คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์	12
2.3.1 โครงสร้างภายนอกของ MCS-51	13
2.3.2 โครงสร้างภายในของ MCS-51	16
2.4 แนวทางการใช้งาน LCD MODULE	17
2.4.1 การต่อเข้ากับระบบไมโครโปรเซสเซอร์	18
2.4.2 ชุดควบคุมคำสั่งและแสดงข้อความ	19
2.4.3 ความเข้าใจพื้นฐาน	20
2.5 การใช้งานชิพสนับสนุน	21
บทที่ 3 การคำนวณและการสร้าง	26
3.1 บทนำ	26
3.2 รูปแบบโครงการ	26
3.3 เทคนิคการสร้างสัญญาณพัลส์วิดท์มอดูเลชัน	27

สารบัญ (ต่อ)

	หน้า
3.4 วงจรกำลัง	28
3.4.1 วงจรปริคัจเรคติฟายต์	28
3.4.2 วงจรขับเคลื่อน	28
3.4.3 วงจรขับเคลื่อนมอเตอร์	31
3.5 วงจรควบคุมการหมุนมอเตอร์	42
3.6 คีย์บอร์ด	43
3.7 การตรวจจับตำแหน่ง	44
บทที่ 4 การทดสอบโครงการ	46
4.1 การทดสอบหาค่าพารามิเตอร์ต่างๆ ของมอเตอร์	46
4.2 การทดสอบการสร้างสัญญาณพัลส์วิดท์มอดูเลชัน	49
4.3 การทดสอบภาคขับเคลื่อนมอเตอร์	51
4.4 การทดสอบสัญญาณพัลส์จากเอ็นโค้ดเดอร์	54
4.5 การทดสอบโครงการทั้งหมด	55
บทที่ 5 สรุปและแนวทางการพัฒนา	63
ภาคผนวก	
กิตติกรรมประกาศ	
เอกสารอ้างอิง	

## สารบัญภาพ

ภาพที่		หน้า
2.1	แสดงเอซีเซอร์ไวมอเตอร์แบบ 2 เฟส	4
2.2	แสดงคุณลักษณะการทำงานที่เป็นลิเนียร์ของเอซีมอเตอร์	4
2.3	แสดงคุณลักษณะระหว่างความเร็วและแรงบิดของมอเตอร์	5
2.4	แสดงเซอร์ไวมอเตอร์แบบ โรเตอร์เป็นแกนเหล็ก	6
2.5	แสดงเซอร์ไวมอเตอร์แบบ Drag-Cup	6
2.6	แสดงดีซีมอเตอร์แบบอาร์เมเจอร์ต่ออนุกรมกับขดลวดสนามแม่เหล็ก	7
2.7	แสดงคุณลักษณะระหว่างความเร็วและแรงบิดของดีซีซีรี่ส์มอเตอร์ภายใต้ สภาวะโวลต์เตจคงที่	7
2.8	แสดงดีซีมอเตอร์แบบแยกสนามสนามแม่เหล็ก ได้	8
2.9	แสดงคุณลักษณะระหว่างความเร็วและแรงบิดของดีซีซีรี่ส์มอเตอร์ ภายใต้ สภาวะสนามแม่เหล็ก และโวลต์เตจคงที่	9
2.10	แสดงซีรี่ส์มอเตอร์แบบแม่เหล็กถาวร	9
2.11	แสดงดีซีมอเตอร์แบบฟิลด์เป็นแม่เหล็กถาวร	9
2.12	แสดงตัวอย่างกลไกของเอ็นโค้ดเดอร์	11
2.13	แสดงสัญญาณเอ็นโค้ดเดอร์ 2 ช่อง มีมุมต่างเฟส 90 องศา	12
2.14	แสดงบล็อกไดอะแกรมของ MCS-51	16
2.15	แสดงบล็อกไดอะแกรมของ 8254	21
2.16	แสดงรูปแบบของคำสั่งควบคุม	22
2.17	แสดงบล็อกไดอะแกรมที่ช่วยในการอ่านเพื่อป้องกันความผิดพลาด	24
3.1	แสดงวงจรการควบคุมตำแหน่งดีซีเซอร์ไวมอเตอร์โดยไมโคร คอนโทรลเลอร์	26
3.2	แสดงการต่อวงจรสร้างสัญญาณพัลส์วิดท์มอดูเลชัน	27
3.3	แสดงรูปสัญญาณพัลส์วิดท์มอดูเลชัน	28
3.4	แสดงวงจรบริดจ์เรกติไฟายต์ 1 เฟส	28
3.5	แสดงรูปร่างของกระแสเบสในอุคมคติ	29
3.6	แสดงวงจรขับเคลื่อน	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ (ต่อ)

ภาพที่		หน้า
3.7	แสดงวงจรขับเกททั้ง 4 ชุด	31
3.8a	แสดงวงจรการต่อเพาเวอร์ทรานซิสเตอร์เป็นวงจรสวิทซ์	32
3.8b	แสดงรูปโฟโตคัปเบิล และวงจรขยายสัญญาณ (Amplifier)	32
3.8c	แสดงการแยกระบบกำลัง	33
3.9	แสดงลักษณะของกระแสและแรงดันตกคร่อมเพาเวอร์มอสเฟต	34
3.10	แสดงตัวเก็บประจุแผ่นที่ขาต่างๆ ภายในตัวเพาเวอร์มอสเฟต	36
3.11	แสดงลักษณะของแรงดันและกระแสที่ขาเกตขณะเพาเวอร์มอสเฟตถูกไปอัส	36
3.12	แสดงตัวอย่างของ Gate Charge Chart	37
3.13	แสดงการชาร์จประจุที่ขาเกตตามเวลาที่มีผลต่อกระแส	38
3.14	แสดงตัวอย่างกราฟฟิสิกัลพลาตท็อกซ์ FBSOA ของเพาเวอร์มอสเฟต	40
3.15	แสดงตัวอย่างกราฟฟิสิกัลพลาตท็อกซ์ SSOA ของเพาเวอร์มอสเฟต	40
3.16a	แสดงวงจรการต่อเพาเวอร์มอสเฟตเป็นวงจรสวิทซ์	41
3.16b	แสดงรูปโฟโตคัปเบิลและวงจรขยายสัญญาณ (Amplifier)	41
3.16c	แสดงการแยกระบบกำลัง	42
3.17	แสดงวงจรควบคุมทิศทางการหมุนของมอเตอร์	42
3.18	แสดงการต่อคีย์บอร์ดขนาด 4X3 กับพอร์ต P1 ของไมโครคอนโทรลเลอร์	43
3.19	แสดงโฟล์ทรีจการทำงานของโปรแกรมสแกนคีย์บอร์ด	44
3.20	แสดงบล็อกไดอะแกรมการทำงานคาน์เตอร์ (Model)	45
4.1	แสดงกราฟความสัมพันธ์ระหว่างแรงดันและความเร็วรอบของมอเตอร์	47
4.2	แสดงวงจรการทดสอบค่า Inductance ของมอเตอร์	48
4.3	แสดงสัญญาณพัลส์วิดท์มอดูเลชั่น	50
4.4	แสดงสัญญาณพัลส์วิดท์มอดูเลชั่น	51
4.5	แสดงสัญญาณแรงดันและกระแสที่ความเร็วรอบ 1500 rpm ที่ความถี่ 30 kHz	52
4.6	แสดงสัญญาณแรงดันและกระแสที่ความเร็วรอบ 3000 rpm ที่ความถี่ 30 kHz	52
4.7	แสดงสัญญาณแรงดันและกระแสที่ความเร็วรอบ 1500 rpm ที่ความถี่ 20 kHz	53
4.8	แสดงสัญญาณพัลส์ทั้ง 2 Channel จากเอ็นโค้ดเดอร์	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ตารางที่		หน้า
2.1	แสดงคุณสมบัติของไมโครคอนโทรลเลอร์แต่ละเบอร์ตระกูล MCS-51	13
2.2	แสดงหน้าที่พิเศษของแต่ละขาของพอร์ต P3	15
2.3	แสดงขาสัญญาณของ LCD module	18
2.4	แสดงรายละเอียดของชุดคำสั่ง	20
2.5	แสดงค่าการควบคุมการเขียนหรือการอ่าน 8254	22
2.6	แสดงค่าการเลือกแกนเตอร์	23
2.7	แสดงการอ่าน/โหลด	23
2.8	แสดงค่าการเลือกโหมด	25
4.1	แสดงค่าพารามิเตอร์ที่ได้จากการทดลอง	47
4.2	แสดงผลการทดลองขณะ NO LOAD ที่ความเร็วสายพาน 100 RPM	55
4.3	แสดงผลการทดลองขณะ NO LOAD ที่ความเร็วสายพาน 160 RPM	56
4.4	แสดงผลการทดลองขณะขับ LOAD ขนาด 250 กรัม ที่ความเร็วสายพาน 160 RPM	57
4.5	แสดงผลการทดลองขณะขับ LOAD ขนาด 250 กรัม ที่ความเร็วสายพาน 100 RPM	58
4.6	แสดงผลการทดลองขณะขับ LOAD ขนาด 500 กรัม ที่ความเร็วสายพาน 100 RPM	59
4.7	แสดงผลการทดลองขณะขับ LOAD ขนาด 500 กรัม ที่ความเร็วสายพาน 160 RPM	60
4.8	แสดงผลการทดลองขณะขับ LOAD ขนาด 750 กรัม ที่ความเร็วสายพาน 100 RPM	61
4.9	แสดงผลการทดลองขณะขับ LOAD ขนาด 750 กรัม ที่ความเร็วสายพาน 160 RPM	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาและความสำคัญของโครงการ

ในงานที่เกี่ยวข้องกับการเคลื่อนที่ทางกลต่างๆ นั้นโดยทั่วไปจะให้มอเตอร์เป็นตัวขับเคลื่อนกำลัง ซึ่งแต่ละงานก็จะมีการควบคุมค่าต่างๆ ให้มอเตอร์ทำงานให้เราได้ตามที่ต้องการ ค่าต่างๆ ที่ถูกควบคุมได้แก่ ความเร็วรอบ แรงบิด และตำแหน่งในการหมุน มอเตอร์ส่วนใหญ่ที่ใช้เป็นมอเตอร์กระแสตรง เพราะเนื่องจากควบคุมความเร็วได้ง่าย และให้แรงบิดที่ดี

ในการควบคุมให้มอเตอร์ทำงานได้ตามที่ต้องการนั้นที่ตัวมอเตอร์จะต้องมีตัวตรวจจับ (Sensor) ซึ่งเป็นตัวตรวจจับค่าที่เราควบคุมออกมาในรูปของสัญญาณไฟฟ้า ตัวตรวจจับดังกล่าวได้แก่ ทาโคเจนเนอร์เรเตอร์ อินคริเมนต์เอ็นโค้ดเดอร์ โปรเทนซิโอมิเตอร์ เป็นต้น มอเตอร์ที่มีอุปกรณ์ตรวจจับดังกล่าว แล้วนำสัญญาณนั้นป้อนกลับเข้ามาควบคุมเรียกว่าเซอร์โวมอเตอร์

โครงการนี้จะศึกษาการควบคุม ดีซีเซอร์โวมอเตอร์ (dc servo motor) ด้วยสัญญาณดิจิทัล โดยการสร้างสัญญาณ พัลส์วidthมอดูเลชัน (Pulse Width Modulation) เพื่อนำสัญญาณไปจ่ายให้แก่ภาควงจรถับ และจะใช้ไมโครคอนโทรลเลอร์มาประยุกต์ในงานการควบคุม การตรวจจับตำแหน่งจะใช้ อินคริเมนต์เอ็นโค้ดเดอร์ที่ติดตั้งอยู่ที่เพลาส่งสัญญาณที่เป็นพัลส์ เข้าไปยังวงจรถาน์เตอร์ (Counter) แล้วทำการประมวลผลโดยไมโครคอนโทรลเลอร์

#### 1.2 วัตถุประสงค์

- 1.2.1 ศึกษาการเทคนิคการสร้างพัลส์วidthมอดูเลชัน
- 1.2.2 ศึกษาการควบคุม ดีซีเซอร์โวมอเตอร์แบบแกนเดี่ยวโดยเทคนิคพัลส์วidthมอดูเลชัน
- 1.2.3 ศึกษาการควบคุมดีซีเซอร์โวมอเตอร์โดยใช้เฟาเวอร์มอสเฟต
- 1.2.4 ศึกษาการควบคุมดีซีเซอร์โวมอเตอร์แบบป้อนกลับเพื่อควบคุมการหยุดตามตำแหน่งโดย ไมโครคอนโทรลเลอร์
- 1.2.5 ศึกษาการใช้งานไมโครคอนโทรลเลอร์ในงานควบคุม

### 1.3 วิธีการดำเนินงาน

1.3.1 ศึกษารายละเอียด ส่วนประกอบ โครงสร้าง การทำงานของ คีชีเซอร์ไวมอเตอร์ และ การประยุกต์ ใช้ ไมโครคอนโทรลเลอร์ ในงานควบคุม

1.3.2 ออกแบบชุดขับ คีชีมอเตอร์โดยใช้ เพาเวอร์มอสเฟต (Power Mosfet)

1.3.3 เขียนคำสั่งมาควบคุม คีชีเซอร์ไวมอเตอร์

### 1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 สามารถสร้างสัญญาณพัลส์วืดท์มอดูเลชั่น จาก IC8254 (Programmable Interval) โดย การควบคุมจากไมโครคอนโทรลเลอร์

1.4.2 รู้จักการขับเคลื่อนมอเตอร์โดยใช้สัญญาณพัลส์วืดท์มอดูเลชั่น

1.4.3 สามารถควบคุมตำแหน่งการหยุดโดยการป้อนกลับของคีชีเซอร์ไวมอเตอร์ ได้

1.4.4 รู้จักการประยุกต์ใช้ไมโครคอนโทรลเลอร์ในงานต่างๆ ได้

## บทที่ 2

### ทฤษฎีที่ใช้ในการออกแบบ

#### คำนำ

ในการควบคุมระบบการทำงานของเซอร์โวมอเตอร์ สามารถควบคุมได้หลายแบบ ตามแต่การใช้ประโยชน์และวัตถุประสงค์ในการใช้งาน เช่น ควบคุมความเร็วรอบ ควบคุมแรงบิด ควบคุมตำแหน่งที่เพลามอเตอร์ซึ่งประกอบด้วยส่วนต่างๆ เช่น ตัวเซอร์โวมอเตอร์ ชุดเซ็นเซอร์ต่างๆ ชุดขับเคลื่อนเซอร์โวมอเตอร์ เป็นต้น ซึ่งในบทนี้จะพูดถึงทฤษฎีต่างๆที่นำมาใช้ในการออกแบบ

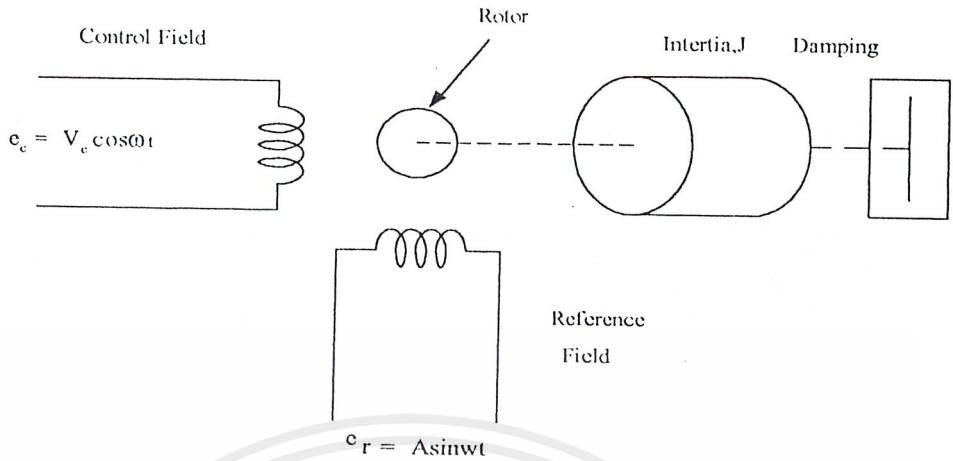
#### 2.1 เซอร์โวมอเตอร์ (SERVO MOTER)

เซอร์โวมอเตอร์ส่วนใหญ่ใช้ในงานที่มีการควบคุมแบบลูปปิด (Close Loop) หรือแบบมีการป้อนกลับ ซึ่งสามารถควบคุมความเร็วและตำแหน่งการทำงานได้อย่างแม่นยำ จะพบได้ในงานเครื่องจักรกลอัตโนมัติ เช่น หุ่นยนต์คอมพิวเตอร์ เครื่องมือวัดคุม เป็นต้น

เซอร์โวมอเตอร์สามารถแบ่งตามโครงสร้างได้เป็น 2 ชนิดใหญ่ๆ คือ เซอร์โวมอเตอร์ไฟฟ้ากระแสสลับ (AC Servo Motor) และเซอร์โวมอเตอร์ไฟฟ้ากระแสตรง (DC Servo Motor)

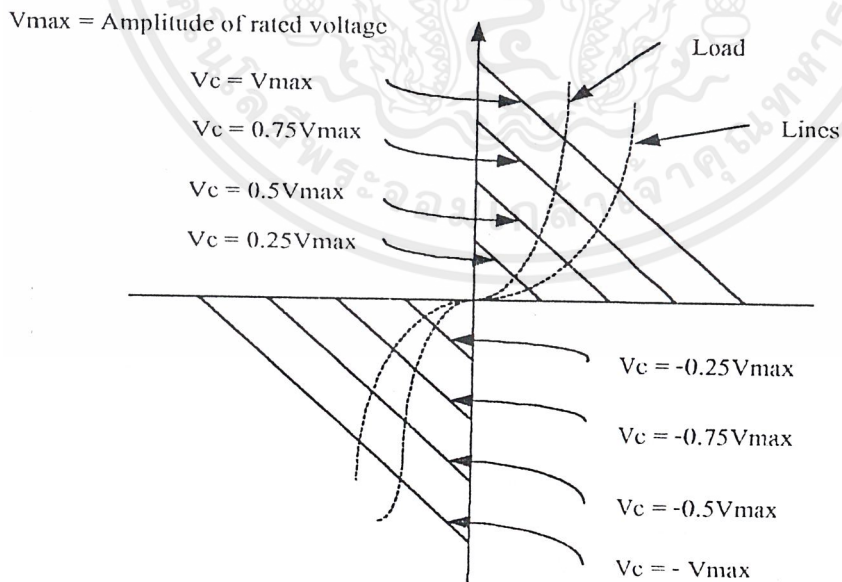
##### 2.1.1 เซอร์โวมอเตอร์ไฟฟ้ากระแสสลับ (AC Servo Motor)

เซอร์โวมอเตอร์กระแสสลับจะใช้กันทั่วไปในระบบคอนโทรลที่ต้องการเพียงกำลังต่ำ สามารถปรับความเร็วได้ ข้อได้เปรียบเบื้องต้นของเซอร์โวมอเตอร์กระแสสลับที่เหนือกว่ามอเตอร์กระแสตรง คือสามารถใช้ร่วมกับซิงโคร (Synchro) ได้ดีโดยไม่ต้องมีการตีโมดูละชั้นสัญญาณความคลาดเคลื่อนไดอะแกรมทางไฟฟ้าของเซอร์โวมอเตอร์กระแสสลับ แสดงในรูปที่ 2.1 มอเตอร์ประกอบด้วยอินดักชันโรเตอร์และขดลวดฟิลด์จำนวน 2 ชุด วางอยู่ในตำแหน่งทำมุมกัน 90 องศา ฟิลด์ขดหนึ่งทำหน้าที่เป็นฟิลด์อ้างอิงที่มีค่าคงที่ ส่วนอีกฟิลด์หนึ่งเป็นฟิลด์คอนโทรล สัญญาณ ERROR AC ที่ได้รับการขยายแล้วจะป้อนให้กับฟิลด์คอนโทรล สัญญาณนี้จะมีขนาดของแอมพลิจูดที่แปรค่าได้ และเฟสก็จะแปรค่าอยู่ที่ศูนย์หรือ 180 องศาฟิลด์อ้างอิงจะได้รับการป้อนกลับด้วยโวลต์เตจที่มีค่าคงที่ผ่านวงจรเลื่อนเฟสไป 90 องศา สัญญาณนี้จะมีขนาดของแอมพลิจูดคงที่และมุมของเฟสเป็น  $-90$  องศา



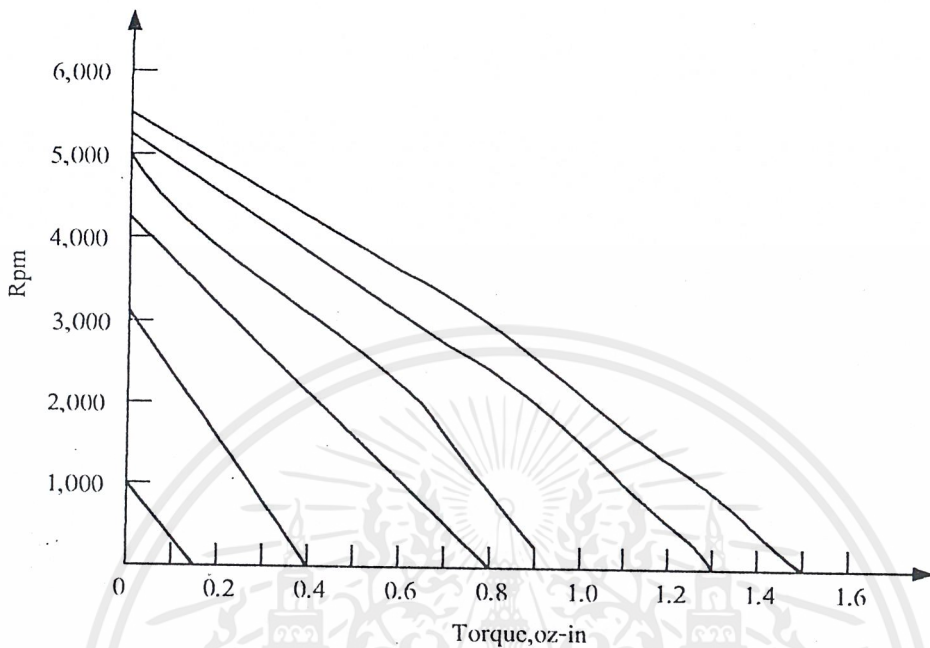
รูปที่ 2.1 เอซิงเซอร์ไวมอเตอร์แบบ 2 เฟส

คุณลักษณะการทำงานที่เป็นลิเนียร์ของเซอร์ไวมอเตอร์แสดงในรูปที่ 2.2 และ 2.3 เส้นทางที่แสดงถึงการทำงานจริงๆจะขึ้นอยู่กับวิถีทางของคุณลักษณะระหว่างความเร็วและแรงบิดของมอเตอร์ ตัวอย่างของวิถีทางของเส้นโหลด 2 เส้น แสดงด้วยเส้นประ การเปลี่ยนแปลงจากเส้นโหลดหนึ่งไปยังอีกเส้นโหลดหนึ่งเป็นตัวอย่างของวิถีทางการเปลี่ยนแปลงของโหลดมอเตอร์  $V_c$  ที่เป็นลบในควอดแรนต์ที่ 3 แสดงให้เห็นว่ามอเตอร์กลับทิศทางเมื่อ  $V_c$  เป็นลบ



รูปที่ 2.2 คุณลักษณะการทำงานที่เป็นลิเนียร์ของเอซิมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 คุณลักษณะระหว่างความเร็วและแรงบิดของมอเตอร์

### 2.1.2 เซอร์โวมอเตอร์กระแสตรง (DC Servo Motor)

คิซีเซอร์โวมอเตอร์ เราสามารถแบ่งออกเป็นหลายประเภทขึ้นอยู่กับวิธีการสร้างสนามแม่เหล็กของตัวมอเตอร์และขึ้นอยู่กับพื้นฐานการออกแบบ โครงสร้างของอาร์เมเจอร์ การแบ่งประเภทตามลักษณะการจ่ายสนามแม่เหล็กแยกได้เป็น 2 แบบคือ

1. คิซีมอเตอร์แบบปรับเส้นแรงแม่เหล็กได้
2. คิซีมอเตอร์แบบเส้นแรงแม่เหล็กคงที่

ถ้าพิจารณาแยกประเภทตามลักษณะการออกแบบ โครงสร้างอาร์เมเจอร์สามารถแยกออกได้เป็น 3 แบบคือ

1. คิซีมอเตอร์แบบอาร์เมเจอร์เป็นแกนเหล็ก
2. คิซีมอเตอร์แบบอาร์เมเจอร์ที่มีขดลวดพันอยู่บนพื้นผิว
3. คิซีมอเตอร์แบบอาร์เมเจอร์เป็นขดลวดหมุน

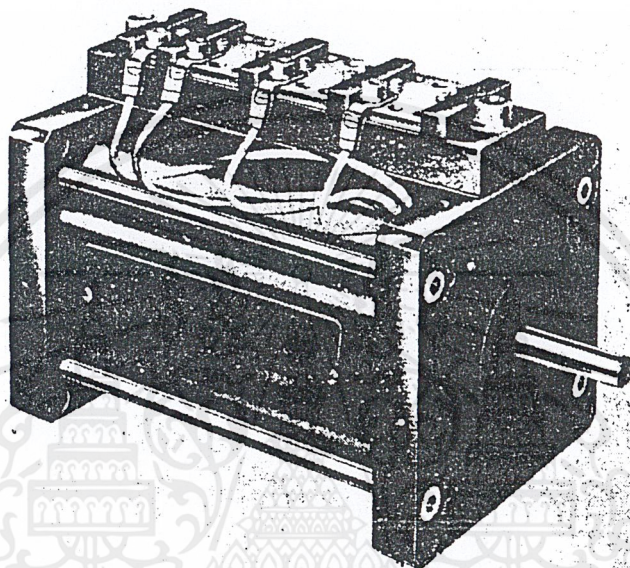
นอกจากนี้ยังมี คิซีมอเตอร์ชนิดพิเศษอีกแบบหนึ่งคือ แบบไม่มีแปรงถ่าน (Brushless DC Motor) ซึ่งมีหลักการทางเทคโนโลยีเหมือนกับ คิซีมอเตอร์ชนิดแปรงถ่าน ยกเว้นการคอมมิวเตชันกระทำโดยเทคนิคทางอิเล็กทรอนิกส์แทนที่จะกระทำโดยวิธีการทางกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

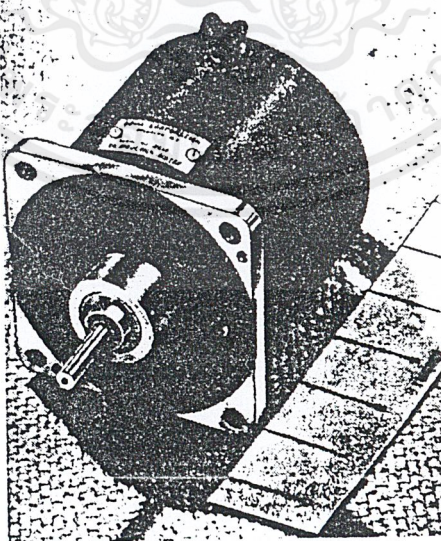
### 2.1.2.1 ดีซีมอเตอร์แบบปรับเส้นแรงแม่เหล็กได้

ดีซีมอเตอร์แบบปรับเส้นแรงแม่เหล็กได้แบ่งออกเป็น 2 แบบคือ

- ก) แบบขดลวดสนามแม่เหล็กต่ออนุกรมกับขดลวดอาร์เมเจอร์
- ข) แบบขดลวดสนามแม่เหล็กแยกกระตุ้น



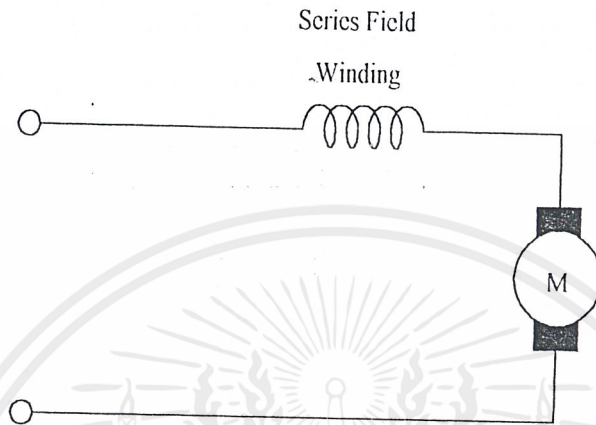
รูปที่ 2.4 เซอร์โวมอเตอร์แบบ โรเตอร์เป็นแกนเหล็ก



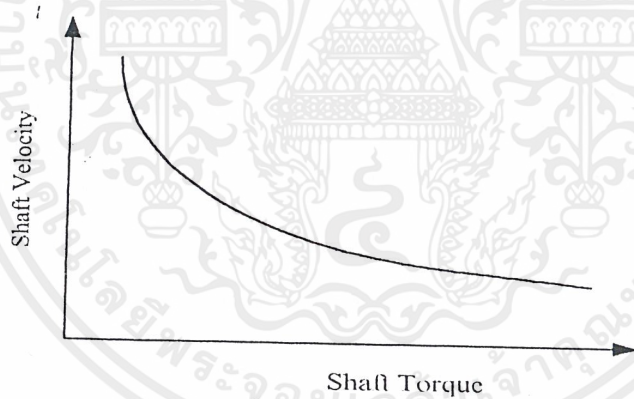
รูปที่ 2.5 เซอร์โวมอเตอร์แบบ Drag-Cup

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มอเตอร์แบบขดลวดสนามแม่เหล็กต่ออนุกรมแสดงในรูปที่ 2.6 มอเตอร์แบบนี้จะมีเส้นแรงแม่เหล็กเป็นสัดส่วนกับกระแส ดังนั้นเส้นแรงของสนามแม่เหล็กจึงสามารถปรับค่าได้ และเราจะได้ความสัมพันธ์ระหว่างความเร็วและแรงบิดเป็นนอนลิเนียร์ดังแสดงในรูปที่ 2.7



รูปที่ 2.6 คีชีมอเตอร์แบบอาร์เมเจอร์ต่ออนุกรมกับขดลวดสนามแม่เหล็ก

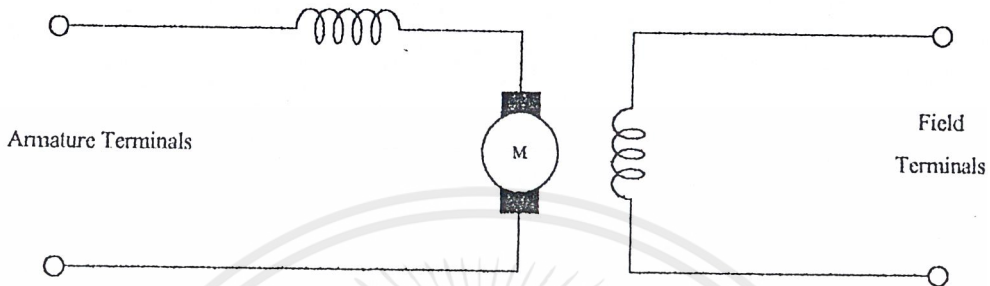


รูปที่ 2.7 คุณลักษณะระหว่างความเร็วและแรงบิดของคีชีซีรี่ส์มอเตอร์ภายใต้ภาวะ โวลต์เตจคงที่

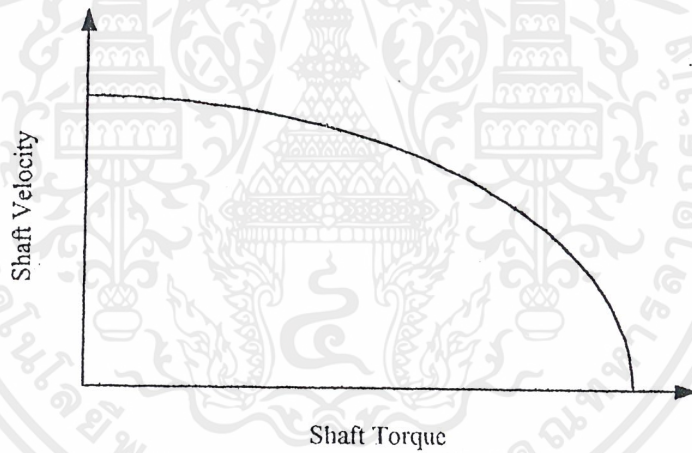
มอเตอร์ดังกล่าวจะใช้งานในสภาวะเฉพาะเมื่อต้องการแรงบิดสูงที่ความเร็วต่ำ และแรงบิดต่ำที่ความเร็วสูงเช่น ระบบขับเคลื่อนรถลาก

มอเตอร์แบบขดลวดสนามแม่เหล็กแยกกระตุ้น แสดงในรูปที่ 2.8 คีชีมอเตอร์แบบนี้มักนิยมเรียกว่ามอเตอร์ชาน (Shunt Motor) มอเตอร์แบบนี้สามารถปรับเส้นแรงแม่เหล็กได้อย่างอิสระต่อกระแสของอาร์เมเจอร์ยังผลให้สามารถควบคุมพารามิเตอร์ของมอเตอร์ให้มีค่าคงที่ได้ตลอดช่วงเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิสัยที่กว้าง มอเตอร์นี้มักจะใช้งานในกรณีระบบการบังคับการเคลื่อนที่ ที่ต้องการแรงบิดสูง ในรูปที่ 2.9 แสดงถึงระหว่างแรงบิดกับความเร็วของ Shunt Motor ภายใต้ภาวะการกระตุ้นสนามแม่เหล็กคงที่และอาร์เมเจอร์ โวลต์เตจคงที่



รูปที่ 2.8 คีซีมอเตอร์แบบแยกสนามแม่เหล็กได้

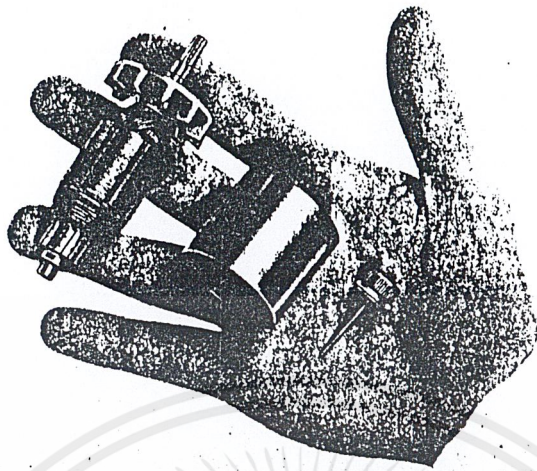


รูปที่ 2.9 แสดงคุณสมบัติระหว่างความเร็วและแรงบิดของคีซีซันท์มอเตอร์ภายใต้สนามแม่เหล็ก และ โวลต์เตจคงที่

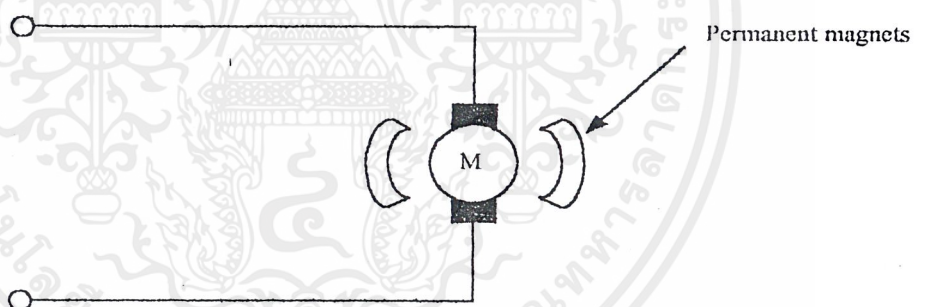
### 2.1.2.2 คีซีมอเตอร์แบบเส้นแรงแม่เหล็กคงที่

ระบบการกระตุ้นฟิลด์ของมอเตอร์โดยทั่วไปในปัจจุบันมักจะใช้เป็นแบบแม่เหล็กถาวร ดังแสดงในรูปที่ 2.10 ในระบบนี้เส้นแรงแม่เหล็กของฟิลด์มีค่าคงที่ ดังนั้นอัตราส่วนระหว่างกระแสอาร์เมเจอร์และแรงบิดจะมีค่าคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 ชั้นท์มอเตอร์แบบแม่เหล็กถาวร



รูปที่ 2.11 ดีไซน์มอเตอร์แบบฟิวด์เป็นแม่เหล็กถาวร

อย่างไรก็ตามข้อดีของมอเตอร์แบบฟิวด์แม่เหล็กถาวรซึ่งเหนือกว่ามอเตอร์แบบมีโครงสร้างของฟิวด์ด้วยการพันขดลวดคือ ไม่มีกำลังสูญเสียในฟิวด์ มีประสิทธิภาพสูงกว่า และมีขนาดเล็กกว่าเมื่อเปรียบเทียบกับมอเตอร์ที่ขนาดของกำลังเท่ากัน นอกจากนี้ความสัมพัทธ์เชิงเส้นยังให้ค่าของกระแสอาร์เมเจอร์ที่ดีกว่าดีไซน์มอเตอร์แบบฟิวด์เป็นขดลวด การประยุกต์ใช้งานเหมาะกับระบบที่ต้องการแรงบิดของไหลสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 เซ็นเซอร์และเอ็นโค้ดเดอร์ (Sensor and Encoder)

เซ็นเซอร์และเอ็นโค้ดเดอร์ เป็นส่วนประกอบที่สำคัญในระบบควบคุมที่มีการป้อนกลับในระบบควบคุมแบบป้อนกลับ ใช้สำหรับการตรวจสอบคุณสมบัติการทำงานของระบบ เซ็นเซอร์และเอ็นโค้ดเดอร์ใช้เป็นตัวป้อนสัญญาณกลับเพื่อการควบคุม เซ็นเซอร์แบ่งออกเป็น 2 ลักษณะคือ แบบอนาล็อกเซ็นเซอร์และแบบดิจิทัลเซ็นเซอร์ แบบอนาล็อกได้แก่ ทาโคเจนเนอเรเตอร์ และซิงโครส ส่วนเซ็นเซอร์แบบดิจิทัลได้แก่ เอ็นโค้ดเดอร์

### 2.2.1 ทาโคเจนเนอเรเตอร์ (Tacho Generator)

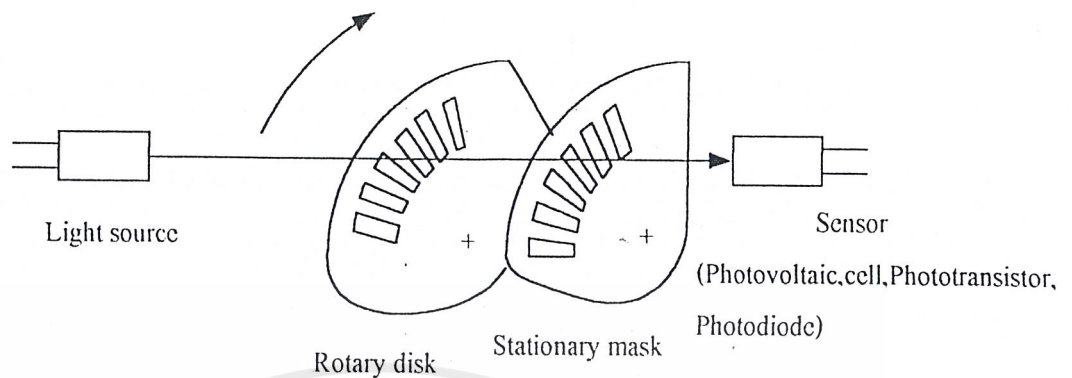
ทาโคเจนเนอเรเตอร์ เป็นเครื่องมือที่สามารถเปลี่ยนพลังงานกลไปเป็นพลังงานไฟฟ้าและให้กำเนิดเอาต์พุตโวลต์เตจ เป็นสัดส่วนกับแมกนิจูดของความเร็วเชิงมุม ในระบบการบังคับมอเตอร์ โดยทั่วไปจะใช้ทาโคมิเตอร์เป็นตัวแสดงความเร็วของเพลลาของมอเตอร์หรือป้อนความเร็วกลับสำหรับการบังคับความเร็ว หรือสำหรับทำให้เสถียรภาพของระบบดีขึ้น

### 2.2.2 เอ็นโค้ดเดอร์ (Encoder)

ระบบการบังคับตำแหน่งของเพลลามอเตอร์ ต้องใช้เอ็นโค้ดเดอร์สำหรับรักษาค่าตำแหน่ง และสร้างสัญญาณป้อนกลับโดยที่ตัวเอ็นโค้ดเดอร์จะสร้างสัญญาณพัลส์ที่แปรผันตรงกับการหมุนของเพลลา ซึ่งสามารถนำไปใช้ในการรับรู้ความเร็วและตำแหน่งเพลลาของมอเตอร์ในรูปของอัตราจำนวนพัลส์ได้

เอ็นโค้ดเดอร์ประกอบด้วยส่วนที่สำคัญคือตัวกำเนิดแสง จานหมุน (rotary disk) บนแผ่นจานหมุนทำเป็นช่องโคจรอบคังแสดงในรูป ที่ใช้กันอยู่ทั่วไปมีด้วยกัน 2 ชนิดคือ แบบสมบูรณ์ในตัวเอง และแบบไม่สมบูรณ์ในตัวเอง

เอ็นโค้ดเดอร์แบบสมบูรณ์ในตัวเองรูปที่ 1.12 และบนแผ่นจานหมุนจะมีช่องสำหรับให้แสงผ่านข้ามไปตัวเซ็นเซอร์ ส่วนตัวกำเนิดแสงจะใช้เป็นหลอดไฟหรือแอลซีดี



รูปที่ 2.12 ตัวอย่างกลไกของเอ็นโคดเดอร์

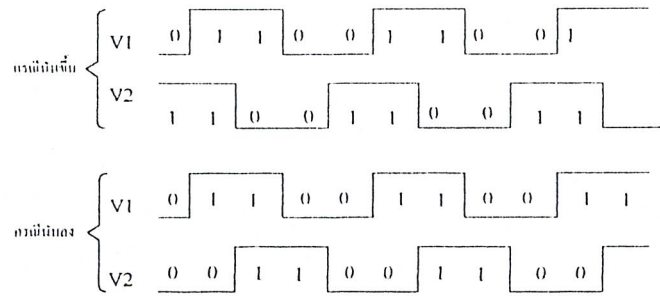
#### ความละเอียดของเอ็นโคดเดอร์

ความละเอียดของเอ็นโคดเดอร์ คือจำนวนคาบเวลาของสัญญาณเอาต์พุตต่อการหมุนของเพลา 1 รอบ ซึ่งบอกเป็นจำนวนพัลส์ต่อรอบ หรือจำนวนไซเคิลต่อองศา เอ็นโคดเดอร์ที่ใช้กันโดยทั่วไปมีค่าความละเอียดตั้งแต่ 15 ถึง 10,000 พัลส์ต่อรอบ จำนวนพัลส์ต่อ 1 รอบ ของสัญญาณที่เอ็นโคดเดอร์สร้างออกมาจะเท่ากับจำนวนช่องว่างบนแผ่นจานหมุน

#### เอาต์พุตของเอ็นโคดเดอร์

โดยทั่วไปแล้วสัญญาณเอาต์พุตที่ออกจากเอ็นโคดเดอร์โดยตรงจะมีระดับไม่เพียงพอในการควบคุม หรือสำหรับการประมวลสัญญาณ ดังนั้นจึงต้องมีวงจรมีขยายและแปลงรูปร่างลูกคลื่นสัญญาณต่อไว้ในตัวเอ็นโคดเดอร์ด้วยเสมอ สัญญาณลูกคลื่นที่ได้จากตัวเซ็นเซอร์ปกติแล้วจะเป็นรูปสัญญาณสามเหลี่ยม หรือรูปสัญญาณซายน์ขึ้นอยู่กับความละเอียดที่ต้องการ รูปสัญญาณเหล่านี้สามารถทำให้เป็นรูปสี่เหลี่ยมได้โดยการต่อคอมพาราเตอร์เข้ากับทรีนารีแอมพลิไฟของเอ็นโคดเดอร์ก็จะได้อาต์พุตเป็นรูปสี่เหลี่ยมที่ต้องการ

สัญญาณ 2 ชุดที่ได้จากเอ็นโคดเดอร์ชนิด 2 ช่องเฟสของสัญญาณ 2 ช่องนี้จะต่างกัน 90 องศาทางไฟฟ้า เราเรียกสัญญาณ 2 ช่องนี้ว่าเป็นควอดคราเจอร์ (quadrature) กันซึ่งเหมาะที่จะใช้ในการรับรู้ทิศทางของการหมุนของเพลาหรือใช้ควบคุมระบบที่ซับซ้อนอื่นๆ



รูปที่ 2.13 สัญญาณเอ็นโคเดอร์ 2 ช่อง มีมุมต่างเฟส 90 องศา

### 2.3 คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์

คุณสมบัติทั่วไปที่สำคัญของไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีดังนี้

1. เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต
2. มีวงจรถอดสซิลเลเตอร์ และวงจรถอดสัญญาณนาฬิกาภายในไอซี
3. มีขาสัญญาณเชื่อมต่อหน่วยความจำข้อมูลภายนอก (external data memory) โดยอ้างตำแหน่งแอดเดรสได้ถึง 64 K
4. สามารถเชื่อมต่อหน่วยความจำโปรแกรมภายนอก (external program memory) โดยอ้างตำแหน่งแอดเดรสได้สูง 64 K
5. มีหน่วยความจำโปรแกรมภายในตัว (on-chip program memory) ขนาด 4 K โดยเฉพาะเบอร์ 8052 จะมีหน่วยความจำในส่วนนี้ถึง 8 K สำหรับเบอร์ 8031 และ M8032 จะไม่มีหน่วยความจำในส่วนนี้
6. มีหน่วยความจำข้อมูลภายในตัว (on-chip data memory) ขนาด 128 ไบต์ โดยเฉพาะเบอร์ 8032 และ 8052 จะมีหน่วยความจำในส่วนนี้ถึง 256 ไบต์
7. หน่วยความจำข้อมูลภายในบางส่วนสามารถเข้าถึงข้อมูลระดับบิตได้ด้วย ทำให้การควบคุมหรือการตรวจสอบสถานะบิตทำได้ง่าย ส่งผลให้การเขียนโปรแกรมทำได้ง่ายมากขึ้น

เบอร์	หน่วยความจำภายใน		ตัวจับเวลา/ ตัวนับจำนวน	อินเทอร์รัปต์
	โปรแกรม	ข้อมูล		
8052 AH	8K × 8 ROM	256 × 8 RAM	3 × 16 BIT	6
8051 AH	4K × 8 ROM	128 × 8 RAM	2 × 16 BIT	5
8051	4K × 8 ROM	128 × 8 RAM	2 × 16 BIT	5
8032 AH	NO ROM	256 × 8 RAM	3 × 16 BIT	6
8031 AH	NO ROM	128 × 8 RAM	2 × 16 BIT	5
8031	NO ROM	128 × 8 RAM	2 × 16 BIT	5
8751 H	4K × 8 EPROM	128 × 8 RAM	2 × 16 BIT	5
8752 H	8K × 8 EPROM	256 × 8 RAM	3 × 16 BIT	6

### ตารางที่ 2.1 แสดงคุณสมบัติของไมโครคอนโทรลเลอร์แต่ละเบอร์ตระกูล MCS-51

8. มีไทม์เมอร์/เคาน์เตอร์ (timer/counter) ขนาด 16 บิต จำนวน 2 ตัว โดยเฉพาะเบอร์ 8032 หรือ 8052 จะมีไทม์เมอร์/เคาน์เตอร์ จำนวน 3 ตัว
9. การอินเทอร์รัปต์สามารถทำได้จาก 5 แหล่งกำเนิด โดยเฉพาะเบอร์ 8032 และ 8052 จะทำอินเทอร์รัปต์ได้ 6 แหล่งกำเนิด โดยการอินเทอร์รัปต์ยังสามารถจัดระดับความสำคัญได้เป็น 2 ระดับ
10. มีพอร์ตการสื่อสารอนุกรมภายในตัวเอง ซึ่งการทำงานเป็นแบบฟูลดูเพล็กซ์ (full duplex)
11. มีคำสั่งคำนวณทางคณิตศาสตร์และทางตรรกศาสตร์
12. คำสั่งส่วนใหญ่ใช้เวลาการทำงานเพียง 1 ไมโครวินาที เมื่อใช้คริสตอลความถี่ 12 เมกะเฮิรตซ์
13. ต้องการแหล่งจ่ายไฟ 5 โวลต์ เพียงชุดเดียว

#### 2.3.1 โครงสร้างภายนอกของ MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทุกเบอร์จะมีตำแหน่งขาพื้นฐานที่เหมือนกัน ดังแสดงในรูปที่ 2.1 สำหรับหน้าที่การใช้งานของแต่ละขามีดังนี้

- ขา Vcc เป็นขาป้อนแรงดันไฟเลี้ยง +5 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขา Vss เป็นขากราวด์
- ขาพอร์ต 0 (Port 0) มี 8 ขา ได้แก่ขา  $P_{0,0}$ - $P_{0,7}$  เป็นขาพอร์ตอินพุทเอาต์พุทแบบ 2 ทิศทางสำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุทพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้ขาพอร์ตเหล่านั้นอยู่ในสถานะปล่อยลอย ซึ่งในสถานะนี้เองที่สามารถนำมาใช้เป็นพอร์ตอินพุทอิมพีแดนซ์สูงได้ นอกจากพอร์ตนี้จะใช้งานเป็นพอร์ตอินพุทเอาต์พุทแล้วมันยังถูกใช้งานในการติดต่อกับหน่วยความจำภายนอกด้วย โดยทำหน้าที่ในการกำหนดตำแหน่งแอดเดรสไบต์ต่ำ ( $A_0$ - $A_7$ ) ซึ่งจะใช้งานเป็นแบบมัลติเพล็กซ์กับการรับส่งข้อมูลขนาด 8 บิต ( $D_0$ - $D_7$ )
- ขาพอร์ต 1 (Port 1) มี 8 ขา ได้แก่ขา  $P_{1,0}$ - $P_{1,7}$  เป็นขาพอร์ตอินพุทเอาต์พุทแบบ 2 ทิศทาง สำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุทพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้เป็นพอร์ตอินพุท นอกจากนี้สำหรับเบอร์ 8032 และ 8052 ขาพอร์ต  $P_{1,0}$  และ  $P_{1,1}$  จะถูกนำมาใช้งานเป็นขา T2 และ T2EX ตามลำดับด้วย
- ขาพอร์ต 2 (Port 2) มี 8 ขา ได้แก่ขา  $P_{2,0}$ - $P_{2,7}$  เป็นขาพอร์ตอินพุทเอาต์พุทแบบ 2 ทิศทางสำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุทพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้เป็นพอร์ตอินพุท นอกจากพอร์ตนี้จะใช้งานเป็นพอร์ตอินพุทเอาต์พุทแล้วมันยังถูกใช้งานในการติดต่อกับหน่วยความจำภายนอกด้วย โดยทำหน้าที่ในการกำหนดตำแหน่งแอดเดรสไบต์สูง ( $A_8$ - $A_{15}$ )
- ขาพอร์ต 3 (Port 3) มี 8 ขา ได้แก่ขา  $P_{3,0}$ - $P_{3,7}$  เป็นขาพอร์ตอินพุทเอาต์พุทแบบ 2 ทิศทางสำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุทพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้เป็นพอร์ตอินพุท นอกจากพอร์ตนี้จะใช้งานเป็นพอร์ตอินพุทเอาต์พุทแล้วมันยังถูกใช้งานในหน้าที่พิเศษต่างๆ ดังแสดงในตารางที่
- ขารีเซต (RST) ใช้สำหรับการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยการรีเซตต้องคงสถานะเป็น 1 อย่างน้อยนาน 2 แมจิ้นไซเคิล ในขณะที่ออสซิลเลเตอร์ยังทำงานอยู่
- ขา ALE/PROG เป็นขาสัญญาณเพื่อทำหน้าที่ควบคุมการแลทซ์ ค่าตำแหน่งแอดเดรสไบต์ต่ำ (Address Latch Enable) เมื่อต้องการติดต่อกับหน่วยความจำภายนอก นอกจากนี้ขานี้ยังทำหน้าที่เป็นอินพุทรับพัลส์ในการโปรแกรม (program pulse input) ในส่วนของหน่วยความจำ EPROM สำหรับไมโครคอนโทรลเลอร์ในตระกูล MCS 51 ที่มีหน่วยความจำโปรแกรมภายในเป็น EPROM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

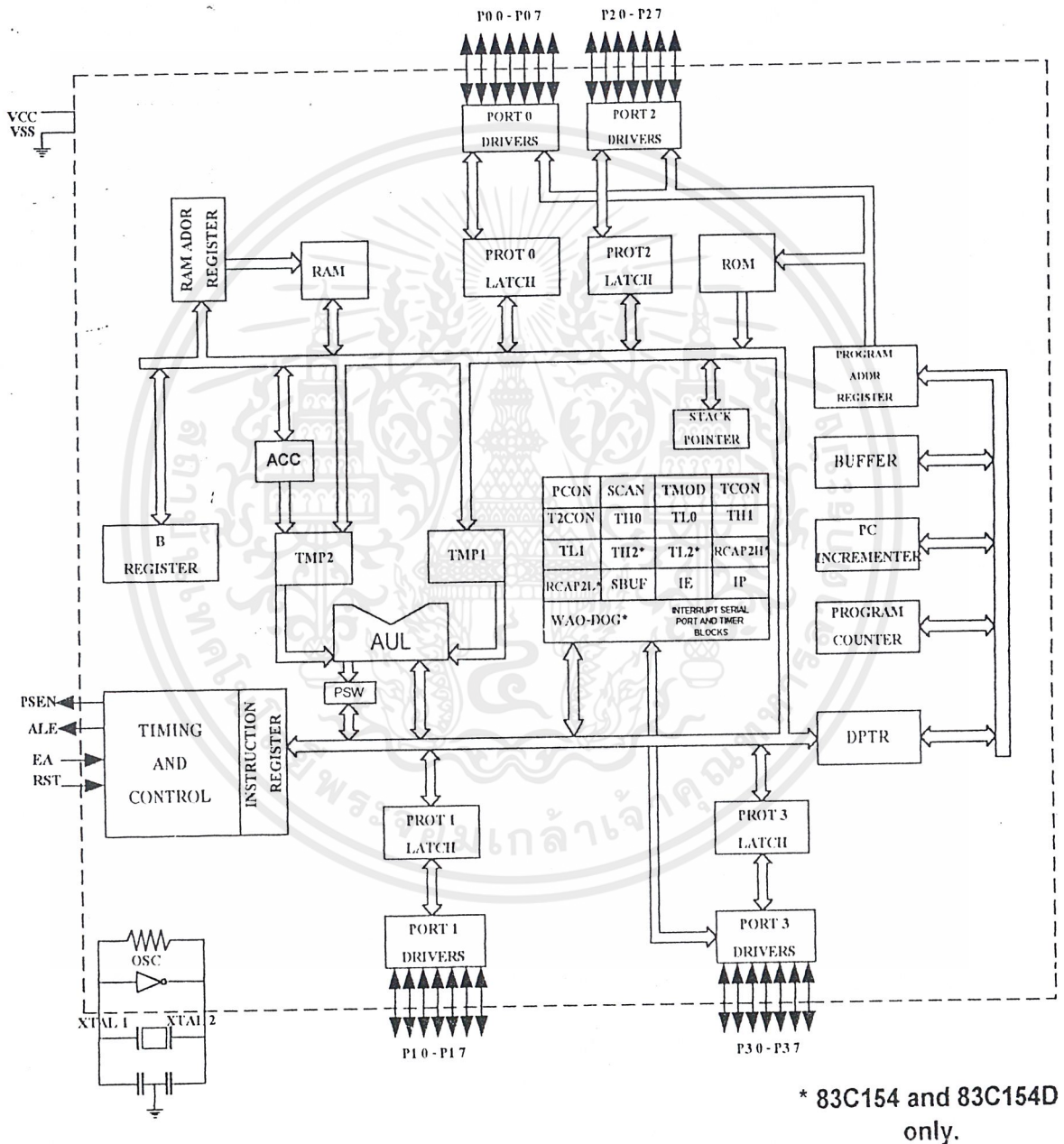
ขาพอร์ท	หน้าที่พิเศษ
P3.0	RXD (serial input port)
P3.1	TXD (serial out port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

ตารางที่ 2.2 แสดงหน้าที่พิเศษของแต่ละขาของพอร์ท P<sub>3</sub>

- ขา PSEN (Program Store Enable) ทำหน้าที่เป็นสัญญาณสโตรบเพื่ออ่านคำสั่งจากหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ประมวลผลคำสั่งจากหน่วยความจำภายนอก ขานี้จะส่งสัญญาณสโตรบจำนวน 2 ครั้งในแต่ละเมซินไซเคิล แต่ในขณะที่ติดต่อกับหน่วยความจำข้อมูลภายนอกจะไม่มีการส่งสัญญาณสโตรบแต่อย่างใด
- ขา EA/VPP (External Access enable/VPP) เป็นขาสำหรับการเลือกใช้หน่วยความจำโปรแกรมจากภายในหรือจากภายนอก โดยถ้ามีสถานะเป็น 0 จะหมายถึงให้ไมโครคอนโทรลเลอร์รับคำสั่งจากหน่วยความจำภายนอกที่ตำแหน่งแอดเดรส 0-0FFFH (0-1FFFH ถ้าเป็นเบอร์ 8052) อย่างไรก็ตามถ้าบิตป้องกัน (security bit) ในหน่วยความจำ EPROM ถูกโปรแกรมไว้ ไมโครคอนโทรลเลอร์จะไม่รับคำสั่งจากหน่วยความจำภายนอกเลย นอกจากนี้ขานี้ยังทำหน้าที่รับแรงดันไฟสำหรับการโปรแกรม (V<sub>pp</sub>) ขนาด 21 โวลต์ เพื่อใช้ในระหว่างการโปรแกรม EPROM
- ขา XTAL<sub>1</sub> และขา XTAL<sub>2</sub> เป็นขาอินพุตและเอาต์พุตของวงจรรีจิสเตอร์ออสซิลเลเตอร์แอมพลิไฟเออร์ (inverting oscillator amplifier) สำหรับใช้คู่ร่วมกับคริสตัลภายนอก

2.3.2 โครงสร้างภายในของ MCS-51

โครงสร้างภายในของไมโครคอนโทรลเลอร์ตระกูล MCS-51 แสดงดังในรูปที่ 2.14



รูปที่ 2.14 บล็อกไดอะแกรมของ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

### 2.4 แนวทางการใช้งาน LCD MODULE

ปัจจุบัน LCD เป็นที่นิยมกันอย่างมาก สำหรับการแสดงผลในเครื่องมือ เครื่องใช้ต่างๆ ทั้งนี้เนื่องจากความเหมาะสมด้วยประการทั้งปวง ทั้งในด้านการกินกระแสต่ำสามารถแสดงผลเป็นตัวอักษรตัวเลข หรือแสดงเป็นกราฟฟิก(Graphic)ได้ (เฉพาะรุ่น) จะคิดปัญหาที่ถือในด้านวงจร ซึ่งมีระบบการทำงานที่ซับซ้อนและหาอุปกรณ์ได้ค่อนข้างยาก แต่ขณะนี้ผู้ผลิต LCD จะทำรุ่นที่เป็น LCD module ออกมาก็ถือเป็น module ที่มีตัว LCD และวงจรควบคุมมาให้พร้อม เรียกว่า LCM ซึ่งทำให้ผู้ใช้สามารถต่อเข้ากับระบบไมโครโปรเซสเซอร์ ได้ง่ายและสำหรับการเขียนโปรแกรม รวมทั้งมีการจำหน่ายกันอย่างกว้างขวาง และมีราคาที่เหมาะสม ทำให้ผู้ใช้งานด้านไมโครโปรเซสเซอร์หันมาใช้แผงแสดงด้วย LCD module กันมากขึ้น

LCD module มีอยู่มากมายหลายรุ่น และมีคุณสมบัติแตกต่างกันไป ซึ่งแบ่งได้เป็น 2 กลุ่มคือ แบบค็อตเมตริกซ์ (dot matrix) และกราฟฟิก (graphic) โดยแบบค็อตเมตริกซ์จะแสดงผลเป็นตัวอักษรขนาด 5x8 dot และมีตัวอักษรและบรรทัดแตกต่างกันไปในแต่ละรุ่น ส่วนแบบกราฟฟิกจะสามารถแสดงผลในแบบบิตแมพ (bitmap) คือสร้างภาพใดๆก็ได้ตามต้องการ แนวทางในการใช้งานของทั้ง 2 แบบจะมีลักษณะใกล้เคียงกัน การใช้งานโดยทั่วไปมักจะใช้แบบค็อตเมตริกซ์มากกว่า เนื่องจากมีราคาถูก และเพียงพอต่องานส่วนใหญ่ ดังนั้นจะกล่าวถึงการใช้งานกับค็อตเมตริกซ์เท่านั้น คุณสมบัติของค็อตเมตริกซ์ LCD module สามารถสรุปเป็นข้อๆ ได้ดังนี้

1. มีให้เลือกหลายรุ่นตามการใช้งาน โดยมีจำนวนตัวอักษรและบรรทัดแตกต่างกันไป
2. ตัวอักษรแสดงด้วยค็อตเมตริกซ์ขนาด 5x8 dot
3. สามารถต่อเข้ากับระบบไมโครโปรเซสเซอร์ได้ 2 ลักษณะ คือแบบมมโมรีแมพ และแบบต่อผ่าน 8255 พอร์ต
4. การใช้งานง่าย และสะดวกระบบไมโครโปรเซสเซอร์เพียงส่งข้อมูลให้กับ LCD module เท่านั้น ข้อความก็จะปรากฏบนแผงแสดง และจะคงค้างไว้ตลอด ทำให้ไม่ต้องเสียเวลาหลักของระบบไมโครโปรเซสเซอร์
5. มีคำสั่งพิเศษสำหรับอำนวยความสะดวกมากมาย เช่น clear display, home cursor, on off cursor, blink character และอื่นๆอีก
6. สามารถแสดงผลเป็นตัวอักษรภาษาอังกฤษและตัวเลขได้ 160 ตัว และสัญลักษณ์พิเศษอีก 32 ตัว รวมทั้งกำหนดอักษรที่ออกแบบเองได้อีก 8 ตัว
7. กินกระแสต่ำ และมีน้ำหนักเบา รวมทั้งทำงานได้ด้วยไฟเลี้ยงระดับ 5 โวลต์เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาสัญญาณของ LCD module

ขา	สัญลักษณ์	ระดับ	หน้าที่
1	V <sub>ss</sub>		0 V. GND
2	V <sub>cc</sub>		+ 5 V. power supply
3	V <sub>cc</sub>		+ V for liquid crystal drive
4	RS	H/L	register select H: data input L: instruction input
5	R/W	H/L	H: data read L: data write
6	E	H	Ebable signal (LH)
7	DB 0	H/L	Data but bit 0
8	DB 1	H/L	Data but bit 1
9	DB 2	H/L	Data but bit 2
10	DB 3	H/L	Data but bit 3
11	DB 4	H/L	Data but bit 4
12	DB 5	H/L	Data but bit 5
13	DB 6	H/L	Data but bit 6
14	DB 7	H/L	Data but bit 7

ตารางที่ 2.3 แสดงขาสัญญาณของ LCD module

#### 2.4.1 การต่อเข้ากับระบบไมโครโปรเซสเซอร์

LCD module จะต่อเข้ากับระบบไมโครโปรเซสเซอร์ได้ 2 ลักษณะคือ แบบเมมโมรีแมพ โดยผ่าน LCD bus ขนาด 20 pin และแบบ I/O port โดยผ่าน 8255 bus ขนาด 26 pin โดยแต่ละแบบจะมีหลักการดังนี้

##### 2.4.1.1 การต่อแบบเมมโมรีแมพ

1. สามารถต่อเข้ากับชิพ (chip) เบอร์ ทั่วๆไปได้เช่น 8051 หรือ Z80 โดยจะทำให้ระบบไมโครโปรเซสเซอร์มองเห็น LCD module ในลักษณะของหน่วยความจำ(memory) ได้ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ผู้ใช้สามารถเขียนและอ่านข้อมูลจาก LCD module ได้ ทำให้มองเสมือนว่าเป็นเมมโมรีบัฟเฟอร์(memory buffer)ไปในตัว
3. เนื่องจากสามารถอ่านข้อมูลกลับได้ จึงทำให้สามารถตรวจสอบแฟล็ก(flag)ความพร้อม ขณะใน LCD module กำลังทำงานได้
4. ใช้ได้กับบอร์ด (board) ที่มี LCD bus มาให้พร้อมเท่านั้น
5. ทำให้กินพื้นที่ของหน่วยความจำไปส่วนหนึ่งและจะต้องมีรหัสการเข้า(decode)ละเอียดพอสมควร
6. การจัดหาสัญญาณจะต้องเป็นไปตามชีพของแต่ละเบอร์

#### 2.4.1.2 การต่อแบบ I/O port

1. สามารถต่อเข้ากับ I/O port ใดๆก็ได้ โดยใช้สัญญาณจำนวน 11 เส้น และใช้โปรแกรมเป็นตัวสร้างสัญญาณขึ้นมาให้ตรงกับข้อกำหนดของ LCD module
2. ผู้ใช้จะเขียนข้อมูลให้ LCD module ได้อย่างเดียว ซึ่งผู้ใช้ควรที่จะกำหนดหน่วยความจำส่วนหนึ่งให้เป็นเสมือนบัฟเฟอร์ให้กับ LCD module อีกที
3. เนื่องจากไม่สามารถอ่านข้อมูลกลับได้ จึงต้องใช้การหน่วงเวลาของระบบไมโครโปรเซสเซอร์เอง เพื่อรอให้ LCD module กระทำขบวนการต่างๆ
4. ใช้ได้กับบอร์ดทั่วไปที่มีพอร์ต
5. ไม่เปลืองส่วนของหน่วยความจำในการใช้งาน
6. การจัดหาสัญญาณกระทำได้อย่างอิสระ

#### 2.4.2 ชุดควบคุมคำสั่งและแสดงข้อความ

การเขียนหรืออ่านข้อมูลกับ LCD module ก็คือการกำหนดคุณสมบัติต่างๆในการใช้งาน LCD ตามชุดคำสั่งควบคุม และรวมถึงการเขียนข้อมูลที่เป็นข้อความ เพื่อให้ปรากฏบนแผงแสดงด้วย โดยมีรายละเอียดตามตารางที่ 2.4

INSTRUCTION	RS	R/ W	DATA BIT								EXE TIME ( $\mu$ S)	
			7	6	5	4	3	2	1	0		
CLEAR DISPLAY	0	0	0	0	0	0	0	0	0	0	1	1640
CURSOR AT HOME	0	0	0	0	0	0	0	0	0	1	*	1640
ENTRY MODE SET	0	0	0	0	0	0	0	0	1	I/D	S	40
DISPLAY ON/OFF	0	0	0	0	0	0	1	D	C	B		40
DISPLAT SHIFT	0	0	0	0	0	1	S/ C	R/ L	*	*		40
FUNCTION SET	0	0	0	0	1	DL	N	F	*	*		40
SET CGRAM ADD.	0	0	0	1	CGRAM ADDRESS						40	
SET DDRAM ADD.	0	0	1	DDRAM ADDRESS							40	
BUSY.ADD READ	0	1	BF	ADDRESS							40	
CGRAM.DDRAM WR	1	0	WRITE DATA								40	
CGRAM.DDRAM RD	1	1	READ DATA								40	

ตารางที่ 2.4 แสดงรายละเอียดของชุดคำสั่ง

### 2.4.3 ความเข้าใจพื้นฐาน

1. การเขียนข้อมูลให้กับ LCD module แบ่งเป็น 2 ลักษณะคือคำสั่งควบคุม(instruction) และข้อมูล(data) โดยจะกำหนดด้วยสัญญาณ RS คือถ้า RS จะหมายถึงส่งคำสั่งควบคุม หรืออ่านค่าเฟลคสภาพการทำงานของ LCD module และถ้า RS =1จะหมายถึงการเขียนหรืออ่านข้อมูลกับ LCD module
2. หลักการในการเขียนข้อมูลให้ LCD moduleนี้ คือ เมื่อมีการเขียนข้อมูลไปแล้วตัว LCD module จะต้องใช้เวลาในการทำงานชั่วขณะหนึ่ง(ตามค่า execute time ในตาราง)

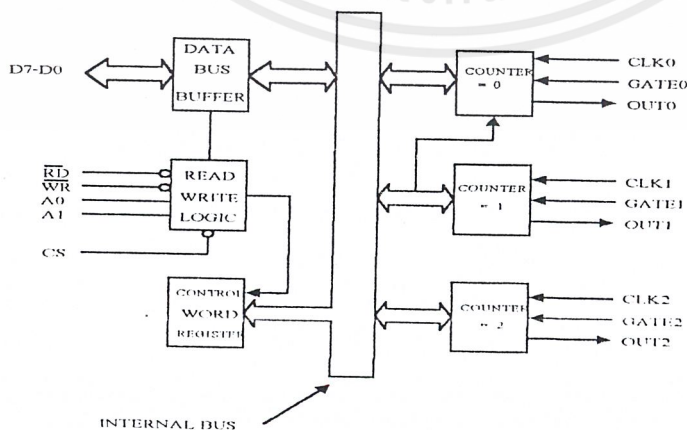
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งระบบไมโครโปรเซสเซอร์สามารถตรวจสอบได้จากบิตที่แฟล็ก(busy flag : BF) และถ้าเรียบร้อยแล้ว จึงจะสามารถเขียนข้อมูลอันต่อไปได้ ในการที่การต่อวงจรเป็นแบบอินพุต/เอาต์พุตพอร์ตคือไม่สามารถอ่านข้อมูลย้อนกลับได้ ระบบไมโครโปรเซสเซอร์ก็จะต้องหน่วงเวลาแทน

3. การเขียนข้อมูลให้กับ LCD module นี้ สามารถทำได้แบบ 8 บิตและ 4 บิต โดยกรณี 4 บิต จะใช้สายส่งสัญญาณเพียง 4 เส้นคือ DB4-DB7 (ใช้สำหรับระบบไมโครโปรเซสเซอร์แบบ 4 บิต หรือเพื่อการประหยัดสาย) การเขียนข้อมูลจะกระทำเหมือนกับ 8 บิต เพียงแต่ให้เขียน 2 ครั้งคือ DB4-DB7 ก่อน แล้วตามด้วย DB0-DB3 และจะต้องกำหนดคุณสมบัติตามค่า DL ในชุดคำสั่ง (function set) ด้วย
4. DDRAM (display data RAM) คือหน่วยความจำภายในตัว LCD module ที่เป็นบัฟเฟอร์ของข้อมูล โดยการเขียนรหัสแอสกี(ASCII)ใดๆลงไป ในหน่วยความจำนี้ ก็จะปรากฏเป็นตัวอักษรที่แสดงผลทันที
5. CGRAM (character generator RAM) คือหน่วยความจำภายในตัว LCD module สำหรับเก็บภาพตัวอักษรที่ผู้ใช้สามารถสร้างได้เอง(8ตัว) โดยจะอ้างแอดเดรส(address) ได้ทั้งหมด 64 ไบต์(byte)คือ 8 ตัวอักษร คูณกับ 8 row

## 2.5 การใช้งานชิพสนับสนุน 8254

8254 เป็นชิพสนับสนุนที่ทำหน้าที่เป็น ไทม์เมอร์ที่สามารถโปรแกรมได้ภายในชิพประกอบด้วยเคาน์เตอร์ที่เป็นอิสระแก่กัน 3 ชุด แต่ละชุดสามารถนับความถี่ได้สูงถึง 2 เมกกะเฮิร์ตซ์ (ถ้าเป็น CMOS จะใช้กับความถี่ได้สูงถึง 10 เมกกะเฮิร์ตซ์) การทำงานของเคาน์เตอร์แต่ละตัวสามารถเลือกได้ถึง 6 โหมด ฟังก์ชันในการนับจะเป็นแบบไบนารีหรือ BCD บล็อกไดอะแกรม จะเป็นดังรูปที่



รูปที่ 2.15 บล็อกไดอะแกรมของ 8254

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากบล็อกไคอะแกรมรูปที่ 2.15 ข้อมูลต่างๆ ที่จะติดต่อกับ 8254 จะกระทำผ่านทางบัสข้อมูลซึ่งมีบัฟเฟอร์อยู่ภายใน (Data bus buffer) บล็อกถัดมาจะเป็นลอจิกที่ใช้ควบคุมการเขียนหรืออ่าน รีจิสเตอร์ภายใน ของ 8254 ให้สอดคล้องกับสัญญาณควบคุมจากไมโครคอนโทรลเลอร์บล็อกถัดมาเป็นรีจิสเตอร์คำสั่งควบคุม (Control Word Register) ใช้สำหรับเก็บ ข้อมูลที่ควบคุมจากรีจิสเตอร์ควบคุมทั้งสิ้น

### การควบคุมการเขียนหรือการอ่าน

การควบคุมสั่งการให้ 8254 ทำงานตามที่ต้องการ เช่น การส่งผ่านข้อมูลระหว่างรีจิสเตอร์ภายในกับบัสข้อมูลของระบบจะเป็นดังนี้

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A1	A0	ฟังก์ชัน
0	1	0	0	0	การเขียนข้อมูลจากบัสข้อมูลไปยังแชนแนล 0
0	1	0	0	1	การเขียนข้อมูลจากบัสข้อมูลไปยังแชนแนล 1
0	1	0	1	0	การเขียนข้อมูลจากบัสข้อมูลไปยังแชนแนล 2
0	0	1	0	0	การอ่านข้อมูลจากแชนแนล 0 วางลงบัสข้อมูล
0	0	1	0	1	การอ่านข้อมูลจากแชนแนล 1 วางลงบัสข้อมูล
0	1	*	*	*	บัสข้อมูลเป็นอิมพีแดนซ์สูง

ตารางที่ 2.5 แสดงค่าการควบคุมการเขียนหรือการอ่าน 8254

### การกำหนดคำสั่งควบคุม

การจะใช้งานแชนแนลใต้นั้นจะต้องการกำหนดสิ่งต่างๆ ดังนี้คือ แชนแนลที่จะใช้ รูปแบบของการนับโหมดที่จะใช้งานซึ่งการกำหนดค่าเหล่านี้จะทำโดยโปรแกรมเข้าไปให้รีจิสเตอร์คำสั่งควบคุมจะมีรูปแบบดังนี้

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

รูปที่ 2.16 รูปแบบของคำสั่งควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากแคว้นเตอร์ของ 8254 มีจำนวน 3 ตัวด้วยกัน ฉะนั้นการโปรแกรมแคว้นเตอร์นั้น จำเป็นต้องกำหนดแคว้นเตอร์ ที่ต้องการจะโปรแกรมเสียก่อน การกำหนดทำได้โดยให้ลอจิกที่ถูกค้องกับบิต D7 และ D6 ซึ่งมีชื่อว่า SC1 และ SC0 ของรูปที่ 2.16 เมื่อได้แคว้นเตอร์ที่ต้องการแล้ว แคว้นเตอร์นั้นจะถูกเซ็ทและจะอยู่ในสภานั้นจนกว่าจะมีคำสั่งควบคุมอื่นมาทำให้เปลี่ยนแปลง ส่วนการกำหนดค่าลอจิกของบิต D7 และ D6 สำหรับใช้เลือกแคว้นเตอร์เป็นดังนี้

D7	D6	เลือกใช้แคว้นเตอร์
0	0	0
0	1	1
1	0	2
1	1	ไม่มีความหมาย

ตาราง 2.6 แสดงค่าการเลือกแคว้นเตอร์

เมื่อเลือกแคว้นเตอร์จากการใช้บิต D7 และ D6 ได้แล้วต่อไปบิต D5 และ D4 จะเป็นตัวกำหนดว่าแคว้นเตอร์นี้จะใช้ใน READ หรือ LOAD MODE ซึ่งโหมดการอ่าน (READ MODE) เป็นโหมดที่ไม่โคร โปรเซสเซอร์ อ่านข้อมูลจากแคว้นเตอร์ ส่วนโหมดการโหลด (LOAD MODE) เป็นโหมดที่ไม่โคร โปรเซสเซอร์เขียนข้อมูลเข้าไปให้แคว้นเตอร์บิต D5 และ D4 ถูกกำหนดดังนี้

D5	D4	รายละเอียด
0	0	ค่าในแคว้นเตอร์ถูกแลทช์
0	1	READ/LOAD เฉพาะไบต์ที่มีนัยสำคัญต่ำ (LEAST-SIGNIFICANT BYTE)
1	0	READ/LOAD เฉพาะไบต์ที่มีนัยสำคัญสูง (MOST-SIGNIFICANT BYTE)

ตารางที่ 2.7 แสดงการอ่าน/โหลด

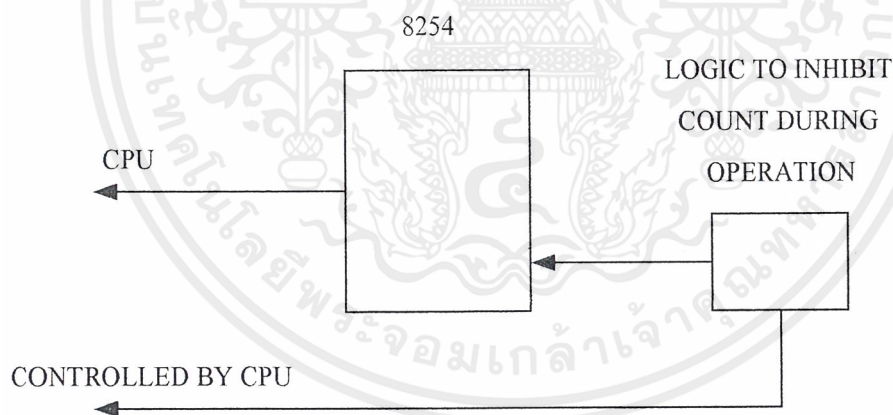
หมายเหตุ แคว้นเตอร์ใน 8254 จะมีขนาด 16 บิต โดยแบ่งเป็น Least Significant Byte และ Most Significant Byte

เมื่อ D5 และ D4 มีค่าเป็น 00H แคว้นเตอร์จะถูกทำให้อยู่ในโหมดการแลทช์ (Latch) ซึ่งเป็นโหมดที่ใช้สำหรับการอ่านค่าของแคว้นเตอร์ขณะที่แคว้นเตอร์ยังทำงานอยู่การเขียนโหมดนี้ให้

กับรีจิสเตอร์ควบคุม จะทำให้ค่าที่อยู่ในเคาน์เตอร์ถูกแลตซ์ให้กับรีจิสเตอร์ภายใน 8254 และเมื่อทำการอ่านเคาน์เตอร์ค่านี้จะถูกอ่านออกไป

การอ่านข้อมูลถ้าไม่อยู่ในโหมดการแลตซ์ อาจเกิดผิดพลาดขึ้นได้เพราะขณะที่ทำการอ่านข้อมูล นั้นขบวนการที่เกิดขึ้นในเคาน์เตอร์จะทำให้ข้อมูลที่อยู่เดิมเปลี่ยนไป ( ดังแสดงในแผนภาพของเวลา ดังรูป ) เป็นผลทำให้ข้อมูลที่ป้อนเข้า CPU เกิดผิดพลาดขึ้น

ยังมีอีกวิธีหนึ่งสำหรับการทำให้ค่าข้อมูลของเคาน์เตอร์ไม่เปลี่ยนแปลงขณะทำการอ่าน โดยการใช้วงจรรภายนอก ที่ทำให้การนับของเคาน์เตอร์หยุดชั่วขณะในระหว่างที่ทำการอ่านซึ่งเทคนิคนี้แสดงไว้ในรูปที่ 2.17 แต่ละวิธีที่กล่าวมานี้ก็มีข้อเสียในตัวเอง ในวิธีการแลตซ์อาจทำให้ไมโครโปรเซสเซอร์ทำการอ่านข้อมูลเก่าที่เกิดก่อนหน้าหลายๆ รอบ ซึ่งขึ้นกับความเร็วของการนับ และไบท์ของเคาน์เตอร์ที่กำลังถูกอ่าน ส่วนวิธีทำให้การนับของเคาน์เตอร์หยุดชั่วขณะโดยวงจรรภายนอก นั้นมีข้อเสียก็คือ จะต้องใช้อุปกรณ์ทางฮาร์ดแวร์มาเพิ่มเติมขึ้นอีกดังนั้นการจะอ่านข้อมูลจากเคาน์เตอร์ด้วยวิธีการใดขึ้นอยู่กับความเหมาะสม



รูปที่ 2.17 บล็อกโคเออร์แกรมที่ใช้ช่วยในการอ่านเพื่อป้องกันความผิดพลาด

บิต D3,D2,D1 เป็นบิตที่กำหนดโหมดการทำงานของเคาน์เตอร์ซึ่งมีรายละเอียดดังนี้

D3	D2	D1	โหมดการทำงาน
0	0	0	MODE 0 : INTERRUPT ON TERMINAL COUNT
0	0	1	MODE 1 : PROGRAMABLE ONE-SHOT
*	1	0	MODE 2 : RATE GENERATER
*	1	1	MODE 3 : SQUARE WAVE GENERATER
1	0	0	MODE 4 : SOFTWARE TRIGGERED STROBE
1	0	1	MODE 5 : SOFTWARE TRIGGER STORE

### ตารางที่ 2.8 แสดงค่าการเลือกโหมด

บิตสุดท้ายของค่าตั้งควบคุมคือ D0 ใช้กำหนดลักษณะการนับของเคาน์เตอร์ว่ามีลักษณะการนับเป็นอย่างไร นั่นคือ จะนับเป็น BCD หรือเป็นไบนารี ถ้า D0 มีลอจิกเป็น '1' เคาน์เตอร์จะนับแบบ BCD ค่าการนับจะวนจาก 9999-0000 ถ้า D0 มีลอจิกเป็น '0' จะนับแบบไบนารี ค่าการนับจะวนจาก FFFFH-0000H

เมื่อกำหนดวิธีการทำงานของเคาน์เตอร์แล้ว ต่อไปจะเป็นการกำหนดเป็นค่าที่เริ่มนับ (Count Value) ค่าที่เขียนลงไปนี้จะต้องสอดคล้องกับรูปแบบของการอ่านหรือการโหลดที่ได้เลือกไว้ด้วยเช่น เลือกการนับแบบ BCD ค่าการนับก็ต้องเป็นลักษณะ BCD เป็นต้น ควรจำไว้ว่า ค่าของเคาน์เตอร์ที่อยู่ภายในจะถูกรีเซ็ตเป็น 0000H เมื่อมีการเขียนค่าตั้งควบคุม

ในกรณีที่กำหนดวิธีการอ่านหรือเขียนเป็นแบบ 0 ไบท์ (RLO,RLI = 1,1) จะต้องส่งค่าไปให้เคาน์เตอร์ 2 ครั้ง โดยการเขียนค่าที่จะต้องส่ง LSB ก่อนและตามด้วย MSB เสมอต่อเนื่องกันไปทันที

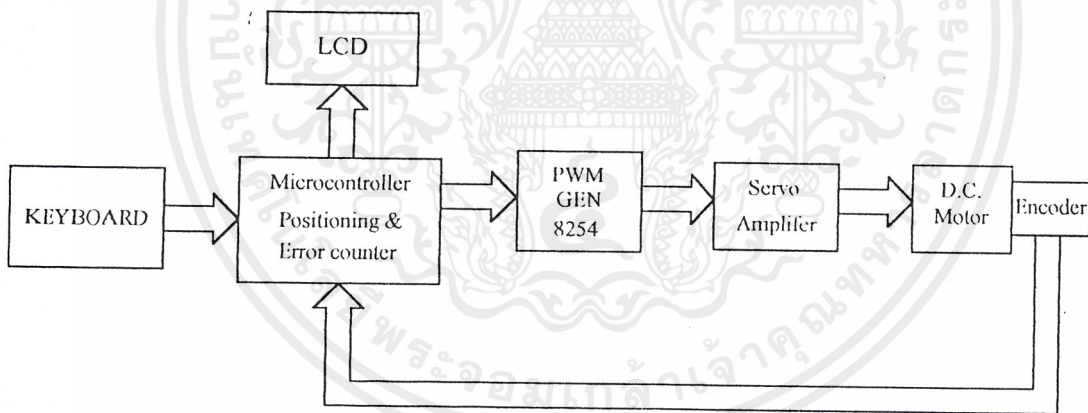
### บทที่ 3

#### การคำนวณ และการสร้าง

##### 3.1 บทนำ

ในบทนี้จะเป็นการศึกษารูปแบบและการสร้างวงจรที่ใช้ในการขับเคลื่อนมอเตอร์ (dc servomotor) โดยการควบคุมจากไมโครคอนโทรลเลอร์ ซึ่งในการขับเคลื่อนมอเตอร์จะใช้เทคนิคพัลส์วidthมอดูเลชั่น PWM (Pulse Width Modulation) และใช้เพาเวอร์มอสเฟต (Power Mosfet) ต่อแบบ H-bridge เป็นวงจรขับ ส่วนในการตรวจจับตำแหน่งจะใช้เอนคอดอร์ ภายในไมโครคอนโทรลเลอร์เป็นตัวจับสัญญาณพัลส์จากเอนคอดอร์ (Encoder) การเขียนโปรแกรมควบคุมการทำงานจะใช้ภาษาแอสเซมบลี (Assembly – Language)

##### 3.2 รูปแบบโครงการ

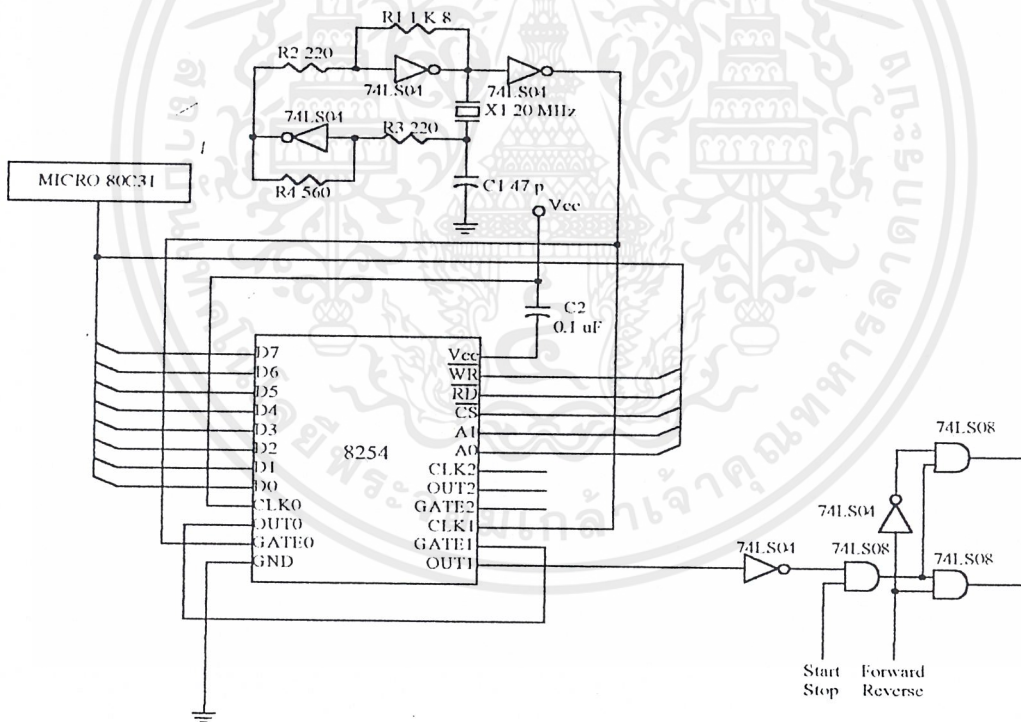


รูปที่ 3.1 แสดงวงจรการควบคุมตำแหน่ง ดิจิตอลเซอร์โวมอเตอร์โดยไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

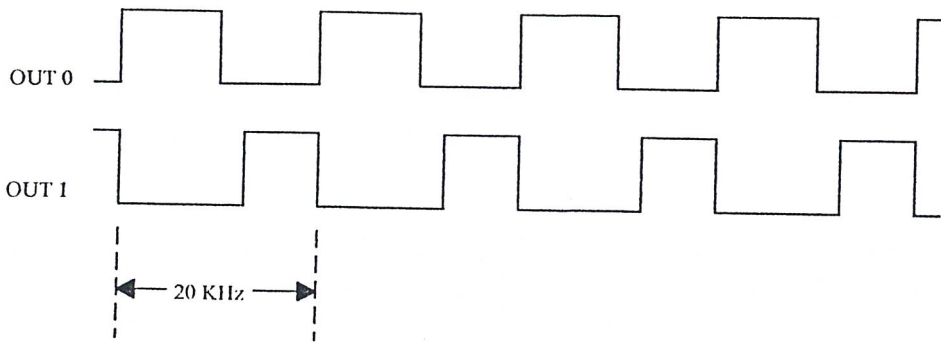
### 3.3 เทคนิคการสร้างสัญญาณพัลส์วิดท์มอดูเลชัน

ในการสร้างสัญญาณ PWM ของโครงงานนี้สามารถอธิบายการทำงานจากรูปที่ 3.2 โดยให้ เคาน์เตอร์ #0 ถูกโปรแกรมให้ทำงานที่โหมด 3 (Square Wave Generator) ก็คือเป็นตัวสร้าง Square Wave โดยที่ขาเกต (Gate) ถูกเชื่อมต่อให้เป็น "1" และขาคล็อก (CLK) ต่อกับส่วนสร้างสัญญาณนาฬิกา (Clock Device) ความถี่ 10 MHz เพื่อเป็นฐานเวลาในสร้างควมถี่ของสัญญาณ PWM ที่ 30 KHz ซึ่งเป็นความถี่ของโครงงานนี้ สัญญาณที่ได้จาก เคาน์เตอร์ #0 จะถูกต่อเข้ากับขาเกต ของ เคาน์เตอร์ #1 โดยที่ เคาน์เตอร์ตัวนี้จะถูกโปรแกรมให้ทำงานในโหมดที่ 1 ซึ่งเรียกว่า โมโนสเตเบิล มัลติไวเบเรเตอร์ (Monostable Multivibrator) จะเป็นการสร้างพัลส์ (Pulse) ให้มีความกว้างของคาบเวลา (Duty Cycle) ตามค่าที่โปรแกรม โดยที่ขาคล็อกของเคา์เตอร์จะต้องป้อนสัญญาณนาฬิกาให้ด้วย ซึ่งจากหลักการดังกล่าวเราสามารถที่จะสร้างสัญญาณ PWM ความถี่ 30MHz ได้อย่างง่ายดายโดยการต่อ 8254 เข้ากับไมโครคอนโทรลเลอร์แสดงได้ดังรูปที่ 3.3



รูปที่ 3.2 แสดงการต่อวงจรสร้างสัญญาณพัลส์วิดท์มอดูเลชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



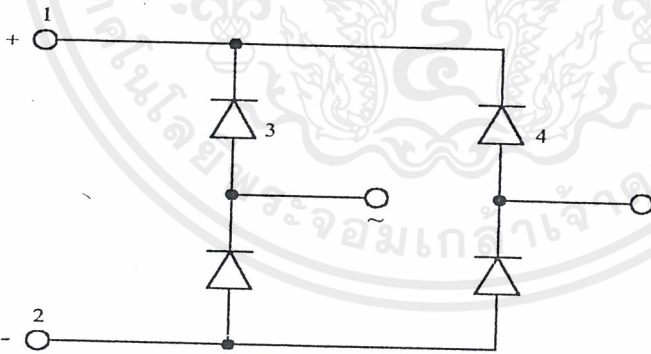
รูปที่ 3.3 แสดงรูปสัญญาณพัลส์วิคท์มอดูเลชั่น

### 3.4 วงจรกำลัง

วงจรกำลังประกอบด้วย

#### 3.4.1 วงจรบริดจ์เรกติไฟยต์

วงจรบริดจ์เรกติไฟยต์ เป็นวงจรที่ใช้สำหรับแปลงสัญญาณไฟกระแสสลับ 1 เฟส ให้เป็นสัญญาณไฟตรง ซึ่งสัญญาณนี้จะนำไปผ่านวงจรกรองกระแสเพื่อให้สัญญาณเรียบขึ้น และนำสัญญาณไฟตรงที่ได้นี้ไปเป็นไฟเลี้ยงให้ส่วนของวงจรกำลัง ซึ่งควบคุมโดยสัญญาณทริกที่ขาเกต ลักษณะของวงจรบริดจ์เรกติไฟยต์ สามารถแสดงได้ดังรูปที่ 3.4



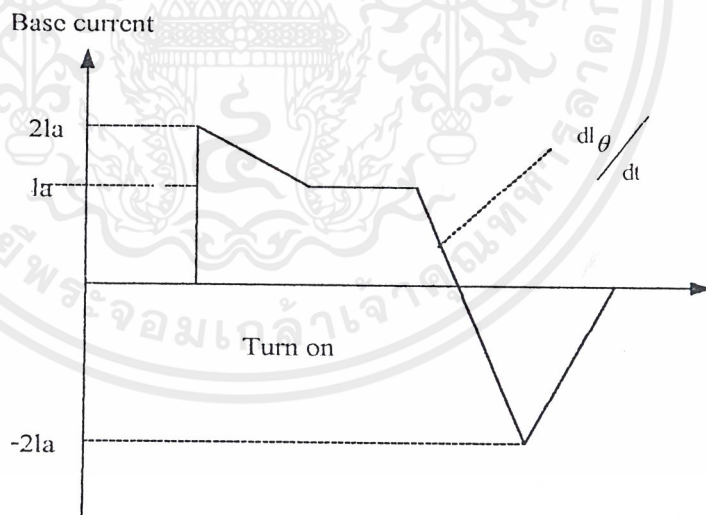
รูปที่ 3.4 แสดงวงจรบริดจ์เรกติไฟยต์ 1 เฟส

#### 3.4.2 วงจรขับเคลื่อน

วงจรขับเคลื่อนที่ใช้จะใช้ในการขับให้เพาเวอร์สวิตชิงในวงจรกำลัง ซึ่งทั้งหมดจะใช้ 4 ซุก โดยปกติแล้ววงจรขับเคลื่อนจะต้องจ่ายกระแสได้เพียงพอ กระแสเบสนี้จะต้องมากเพียงพอที่จะทำให้เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพาเวอร์สวิทช์ซึ่งทำงานอยู่ในย่านอิ่มตัว(Saturate) จากการศึกษาพบว่ากระแสนี้จะอยู่ในย่าน 500-700 mA ดังนั้นวงจรขับเบสโดยส่วนใหญ่จะมีวงจรคาร์ดิ่งตันเพื่อทำหน้าที่ขยายกระแสให้สูงขึ้นในทางทฤษฎีควรให้กระแสมีการไหลย้อนกลับและมีค่าเกือบเป็น 2 เท่าของกระแสเบสเฟอ์เวอร์สในขณะเทอร์น-ออฟ ส่วนในขณะเทอร์น-ออน กระแสเบสควรมีน้อยทั้งนี้เพราะจะทำให้สวิตช์ใหม่มีค่าสูงขึ้น ดังนั้นควรจะให้กระแสเบสมีการโอเวอร์ชูดของขอบขาขึ้นของกระแสเบสที่จะทำให้การทำงานเหมาะสมสำหรับความเร็วของการสวิทช์ที่ต้องการ ดังนั้นในการออกแบบวงจรขับเกทจึงต้องคำนึงถึงกระแสเบสอย่างเหมาะสม โดยวงจรขับเกทส่วนใหญ่ประกอบด้วย

1. ส่วนแยกกราวด์ระหว่างวงจรควบคุมและวงจรถูกัด เพื่อป้องกันการรบกวนและเป็นอันตรายจากไฟแรงดันสูงจากวงจรถูกัดโดยใช้ออปโตคัปเปิล (opto couple) ในที่นี้ใช้ไอซีเบอร์ P521 และมีการต่อวงจรช่วยในลักษณะที่เพิ่มขีดความสามารถในด้านความเร็ว และให้มีการเลื่อนเฟสระหว่างสัญญาณอินพุตและเอาต์พุตให้น้อยที่สุด
2. ส่วนของการให้กระแสไหลย้อนกลับ โดยใช้ไดโอด D2-D4 เร่งการหยุดนำกระแสให้เร็วขึ้น

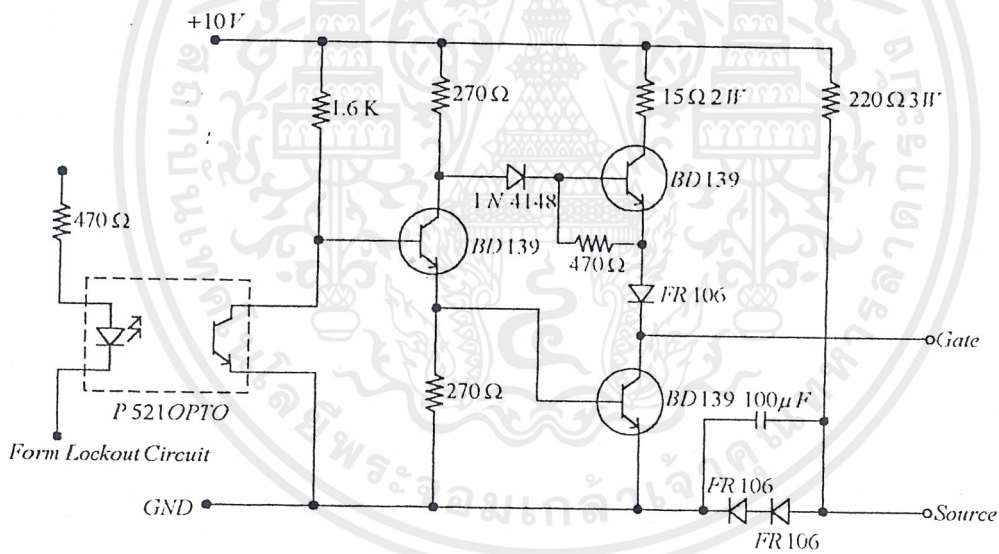


รูปที่ 3.5 แสดงรูปร่างของกระแสเบสในอุดมคติ

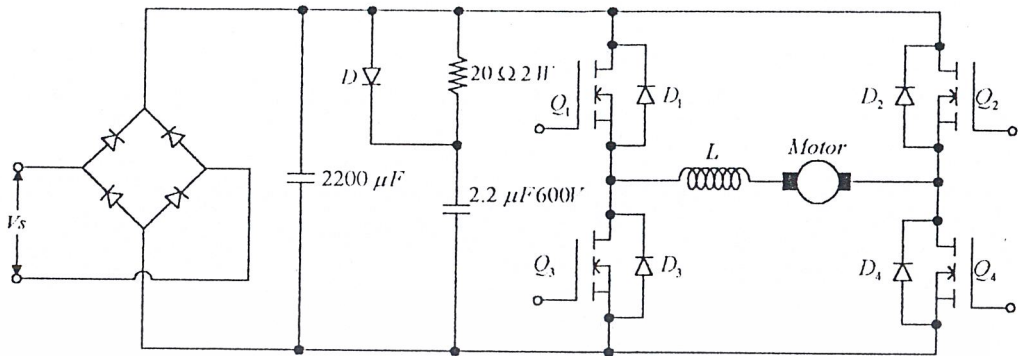
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการทำงาน

จากรูปที่ 3.6 เมื่อสัญญาณพีคบบทวิเอ็มเข้ามาที่ออปโต-คัปเปิล โดยมี R1 ทำหน้าที่จำกัดกระแสให้กับออปโต-คัปเปิล กระแสที่ไหลเข้าออปโต-คัปเปิลจะไปไบอัสให้แอลอีดี(LED) ที่อยู่ในออปโตคิกและดับตามสัญญาณพีคบบทวิเอ็มที่ป้อนเข้าไปเมื่อแอลอีดีคิกโฟโตทรานซิสเตอร์(Photo-transistor) ที่อยู่ภายในไอซีได้รับการไบอัสให้เกิดการนำกระแส และสัญญาณเอาท์พุทที่ได้จะถูกนำไปขยายสัญญาณโดยใช้ทรานซิสเตอร์ CPN2222A และนำสัญญาณที่ได้จากทรานซิสเตอร์ตัวนี้ผ่านวงจรคาร์ลิงตันเพื่อทำหน้าที่ขยายกระแสให้สูงขึ้นพอที่จะนำไปขับให้แก่เพาเวอร์สวิทช์ซึ่งได้ โดยสัญญาณที่ได้จะเป็นสัญญาณพีคบบทวิเอ็มแบบ 3 ระดับ (บวก กราวน์ และลบ) ทั้งนี้เพื่อให้แรงขับชีกลบทำให้เพาเวอร์สวิทช์ซึ่งหยุดทำงานได้เร็วขึ้นจากวงจรจะเป็นว่าสัญญาณชีกลบนี้จะไดโอด D2-D4 นั่นเอง



รูปที่ 3.6 แสดงวงจรขับเกท



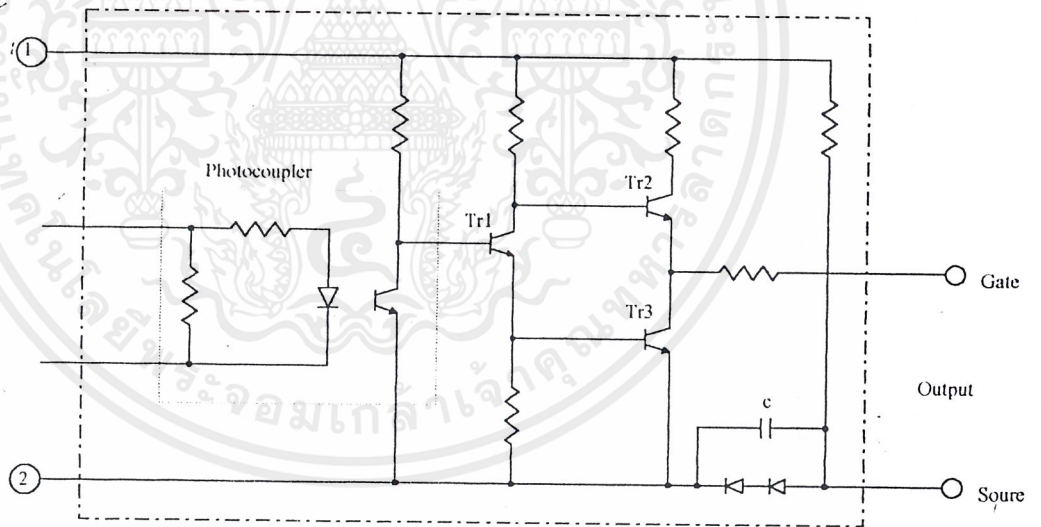
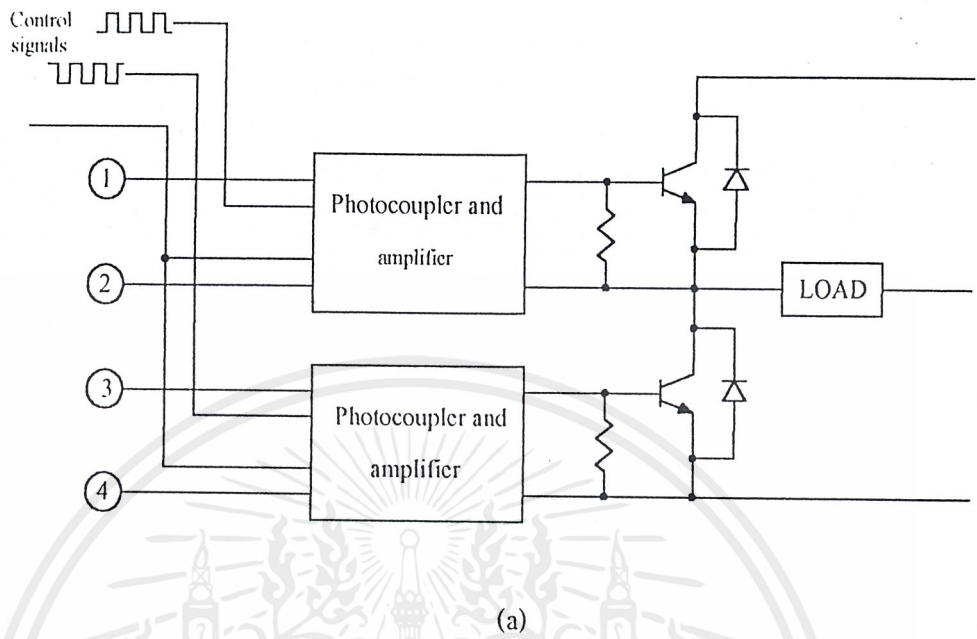
รูปที่ 3.7 แสดงวงจรขับเคลื่อนทั้ง 4 ชั่ว

### 3.4.3 วงจรขับเคลื่อนมอเตอร์

เนื่องจากวงจรขับเคลื่อนมอเตอร์ในวงจรนี้สามารถใช้ในการขับเคลื่อนได้หลายชนิด คือ เพาเวอร์ทรานซิสเตอร์ เพาเวอร์มอสเฟต เป็นต้น แต่ในวงจรนี้เลือกใช้เพาเวอร์มอสเฟตมาเป็นอุปกรณ์ในการสวิตช์ซึ่ง ซึ่งจะกล่าวถึง โครงสร้างและข้อดี-ข้อเสียของอุปกรณ์ทั้งสองชนิดดังต่อไปนี้

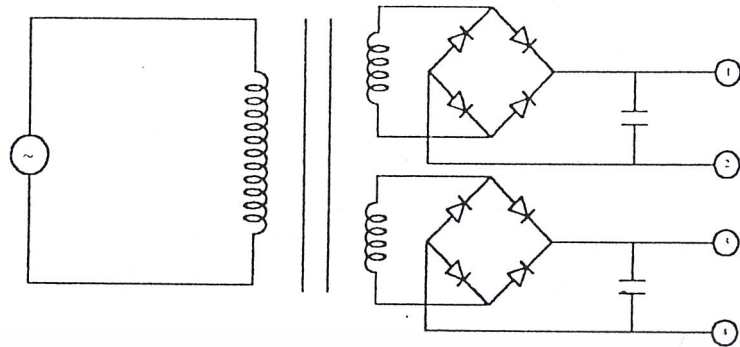
#### 3.4.3.1 เพาเวอร์ทรานซิสเตอร์ (Power Transistor)

เพาเวอร์ทรานซิสเตอร์สามารถนำมาใช้เป็นอุปกรณ์ในการสวิตช์ซึ่งได้โดยนำมาต่อให้ทำงานในย่านอิ่มตัว (Saturation Region) โดยทำหน้าที่เหมือนกับสวิตช์ปิด-เปิด แต่สัญญาณที่นำมาขับขาเบส(Base)จะต้องมีซีกลม เพื่อให้เพาเวอร์ทรานซิสเตอร์สามารถหยุดทำงานได้เร็วขึ้น และทำงานที่ความถี่สูงได้ และมีข้อดีอีกอย่างหนึ่งของเพาเวอร์ทรานซิสเตอร์คือ ไม่ต้องมีวงจรช่วยหยุดนำกระแสทำให้การควบคุมไม่ยุ่งยาก เพียงแค่ควบคุมกระแสที่จ่ายแก่ขาเบสก็สามารถควบคุมการนำและหยุดนำกระแสของเพาเวอร์ทรานซิสเตอร์ได้ และนอกจากนี้ตัวมันยังมีฟรีวิลลิงไดโอดต่อขนานกับขาคอลเลคเตอร์ (Collector) และ อิมิตเตอร์ (Emitter) เพื่อให้ทางเดินของกระแสไหลอย่างต่อเนื่องเมื่อโหลดเป็นชนิดอินดักทีฟโหลด



(b)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(c)

- รูปที่ 3.8 (a) แสดงวงจรการต่อเพาเวอร์ทรานซิสเตอร์เป็นวงจรสวิทชิง  
 (b) แสดงรูปโฟโตคัปเปิ้ล และวงจรขยายสัญญาณ (Amplifier)  
 (c) แสดงการแยกระบบกำลัง

### 3.4.3.2 เพาเวอร์มอสเฟต

เพาเวอร์มอสเฟต ( MOSFET Power Transistor ) สามารถทำงานได้ดีที่ความถี่สูง ตั้งแต่ 50 กิโลเฮิรตซ์ไปจนถึงประมาณ 200 หรือ 400 กิโลเฮิรตซ์ เนื่องจากมันใช้เวลาในการเปลี่ยนสถานะค่อนข้างสั้น และการพัฒนาในปัจจุบันมีแนวโน้มว่าเพาเวอร์มอสเฟต ทำงานได้ที่ความถี่สูงขึ้นไปอีก ซึ่งจะเป็นผลดีในการเลือกใช้กับงานที่ความถี่สูง ในส่วนวงจรขับเพาเวอร์มอสเฟตนั้นสามารถทำได้ง่าย โดยอาจขับเพาเวอร์มอสเฟตให้ทำงานได้จากไอซีควบคุม เช่น UC 3842 โดยตรงเพาเวอร์มอสเฟตจึงเข้ามาแทนที่ไบโพลาร์เพาเวอร์ทรานซิสเตอร์อย่างรวดเร็วในการผลิตสวิทชิงเพาเวอร์ซัพพลายในตลาดอุตสาหกรรมในปัจจุบัน

โครงสร้างของเพาเวอร์มอสเฟตจะเป็นได้ทั้งในลักษณะ N-channel และ P-channel อย่างไรก็ตามในบทนี้จะกล่าวถึงเฉพาะเพาเวอร์มอสเฟตชนิด N-channel เท่านั้น

#### กำลังงานสูญเสียในรูปความร้อนของเพาเวอร์มอสเฟต

กำลังงานสูญเสียที่เกิดขึ้นในเพาเวอร์มอสเฟตขณะทำงาน จะเป็นไปได้ 2 ลักษณะเช่นเดียวกับไบโพลาร์เพาเวอร์ทรานซิสเตอร์คือ กำลังงานสูญเสียขณะเปลี่ยนสถานะและกำลังสูญเสียขณะนำกระแส แต่สำหรับเพาเวอร์มอสเฟตจะมีช่วงเวลาดำเนินการนำกระแส และเวลาช่วงหยุดนำกระแสสั้นกว่าไบโพลาร์ทรานซิสเตอร์มาก เพราะตามโครงสร้างของเพาเวอร์มอสเฟตจะไม่มีประจุสะสมเกิดขึ้นจึงไม่มีปัญหาในลักษณะเดียวกับไบโพลาร์เพาเวอร์ทรานซิสเตอร์ อย่างไรก็ตาม เพาเวอร์มอสเฟตเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฟตจะมีค่าความต้านทานขณะมันนำกระแสค่อนข้างสูง ความสูญเสียของเพาเวอร์มอสเฟตขณะนำกระแสจึงสูงกว่าไบโพลาร์ทรานซิสเตอร์ รูปที่ 3.9 แสดงลักษณะการตอบสนองของเพาเวอร์มอสเฟตเมื่อทำงานเป็นสวิตช์

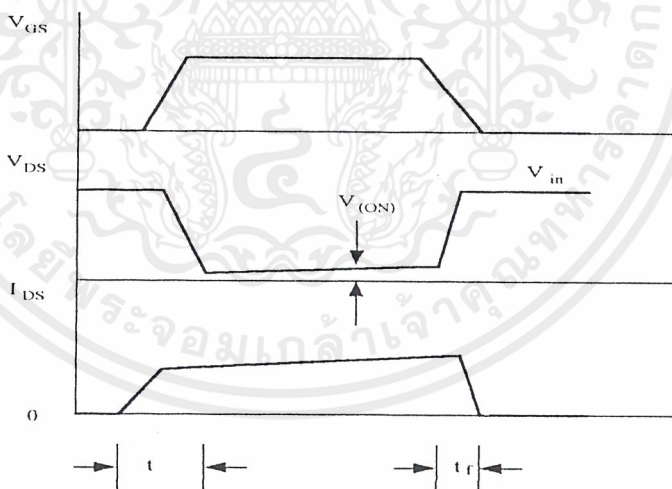
ถึงแม้ว่าช่วงเวลาเริ่มนำกระแสและเริ่มหยุดนำกระแสของเพาเวอร์มอสเฟตจะค่อนข้างสั้น แต่โดยทั่วไปเพาเวอร์มอสเฟตมักถูกใช้งานที่ความถี่สูง การใช้งานที่ความถี่สูงกว่า 50 กิโลเฮิร์ตซ์ การคิดค่ากำลังงานสูญเสียขณะทำงานจำเป็นต้องนำค่ากำลังงานขณะเปลี่ยนสถานะมาคิดด้วย และเนื่องจากช่วงเวลาเริ่มหยุดนำกระแส(turn off time) กับช่วงเวลาเริ่มนำกระแส(turn on time) ของเพาเวอร์มอสเฟตมีค่าใกล้เคียงกันจึงนำมาคิดทั้ง 2 ช่วงเวลาด้วย ดังนั้นกำลังงานสูญเสียของเพาเวอร์มอสเฟตขณะทำงาน  $P_D$  จะมีค่าเท่ากับ

$$P_D = P_{SW(on)} + P_{SW(off)} + P_C \quad \text{Watt} \quad \dots\dots\dots(3.1)$$

$$P_{SW(on)} = 0.5I_{PK(on)} V_{in} \cdot t_r / T \quad \text{Watt} \quad \dots\dots\dots(3.2)$$

$$P_{SW(off)} = 0.5I_{PK(off)} V_{in} \cdot t_f / T \quad \text{Watt} \quad \dots\dots\dots(3.3)$$

$$P_C = I_{Drms}^2 \cdot R_{DSon} (T_j) \quad \text{Watt} \quad \dots\dots\dots (3.4)$$



รูปที่ 3.9 แสดงลักษณะของกระแสและแรงดันตกคร่อมเพาเวอร์มอสเฟตเมื่อเริ่มนำกระแส และเริ่มหยุดนำกระแส(ในส่วนของแรงจะเป็นส่วนที่เกิดกำลังงานสูญเสียในรูปความร้อนได้สูง)

$P_{SW(on)}$  คือ กำลังงานสูญเสียขณะเริ่มนำกระแสของเพาเวอร์มอสเฟต

$P_{SW(off)}$  คือ กำลังงานสูญเสียขณะเริ่มหยุดนำกระแสของเพาเวอร์มอสเฟต

$P_C$  คือ กำลังงานสูญเสียขณะนำกระแสของเพาเวอร์มอสเฟต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$0.5 I_{PK(on)}$	คือ กระแสสูงสุดขณะเริ่มนำกระแส
$0.5 I_{PK(off)}$	คือ กระแสสูงสุดขณะเริ่มหยุดนำกระแส
$I_{Drms}$	คือ กระแสอาร์เอ็มเอสที่ไหลผ่านเพาเวอร์มอสเฟตขณะทำงาน
$R_{DSon}(T_j)$	คือ ค่าความต้านทานระหว่างเคนและซอร์สที่อุณหภูมิรอยต่อสูงสุดขณะเพาเวอร์มอสเฟตทำงาน
$t_r$	คือ ช่วงเวลาเริ่มนำกระแสของเพาเวอร์มอสเฟต
$t_f$	คือ ช่วงเวลาเริ่มหยุดนำกระแสของเพาเวอร์มอสเฟต

หมายเหตุ สำหรับเพาเวอร์มอสเฟตโดยทั่วไป ช่วงเวลาเริ่มนำกระแสและหยุดนำกระแสในค่าดัชนีของผู้ผลิตส่วนใหญ่มีกระบวนมาเป็นค่า  $t_r$  (current rise time) และ  $t_f$  (current fall time) ตามลำดับ ในที่นี้จึงใช้สัญลักษณ์เป็น  $t_r$  และ  $t_f$  แทน  $t_{on}$  และ  $t_{off}$  เมื่อเปรียบเทียบกับไบโพลาร์เพาเวอร์ทรานซิสเตอร์

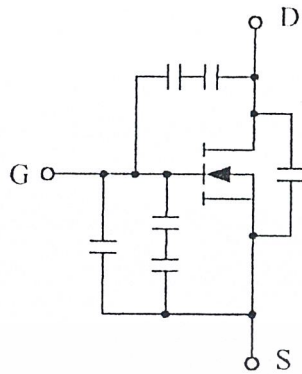
### เงื่อนไขของวงจรขับเพาเวอร์มอสเฟต

การขับเพาเวอร์มอสเฟตให้นำกระแสที่แตกต่างจากการขับกระแสไบอัสในไบโพลาร์เพาเวอร์ทรานซิสเตอร์ เนื่องจากการมีเงื่อนไขการไบอัสที่แตกต่างกัน สำหรับไบโพลาร์เพาเวอร์ทรานซิสเตอร์กระแสไหลผ่านคอลเลกเตอร์และอีมิเตอร์ได้ก็ต่อเมื่อ มีกระแสไบอัสไหลผ่านที่เบสและอีมิเตอร์ แต่เพาเวอร์มอสเฟตจะมีกระแสไหลผ่านเคนและซอร์สได้ก็ต่อเมื่อแรงดันตกคร่อมที่ขาเกตและซอร์สมีค่าอย่างต่ำเท่ากับแรงดันขีดเริ่ม (Threshold Voltage) ของมันแต่ใช้กระแสต่ำการขับเพาเวอร์มอสเฟตให้นำกระแสจึงทำได้ง่ายและยุ่งยากน้อยกว่าเพาเวอร์ทรานซิสเตอร์มาก

อย่างไรก็ตามเพื่อให้เพาเวอร์มอสเฟตทำงานได้อย่างมีประสิทธิภาพสูงสุด จึงจำเป็นต้องศึกษาเงื่อนไขต่างๆ สำหรับการบังคับเพาเวอร์มอสเฟตนำกระแสเป็นอันดับแรกเสียก่อน

### ค่าความจุไฟฟ้าด้านอินพุท

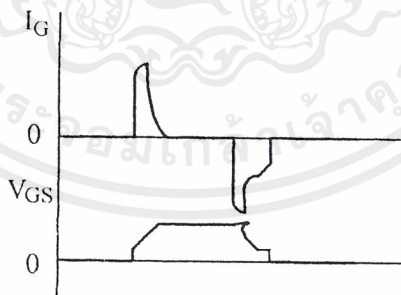
เนื่องจากลักษณะ โครงสร้างภายใน ตัวเพาเวอร์มอสเฟตจึงเหมือนกับมีตัวเก็บประจุต่ออยู่รอบๆ ขาต่างๆ ของมันดังรูปที่ 3.10



รูปที่ 3.10 แสดงตัวเก็บประจุแฝงอยู่ที่ขาต่างๆภายในตัวเพาเวอร์มอสเฟต

ตัวเก็บประจุเหล่านี้บังคับให้เพาเวอร์มอสเฟตต้องชาร์จประจุเข้าไปที่ตัวเก็บประจุเสียก่อน เพื่อให้แรงดันที่ตกคร่อมขาเกต  $V_{GS}$  มีค่าเพิ่มขึ้นจนถึงแรงดันขีดเริ่ม เพาเวอร์มอสเฟตจึงจะเริ่มนำกระแส ในทางกลับกันการหยุดนำกระแสของเพาเวอร์มอสเฟต จะต้องทำให้เพาเวอร์มอสเฟตจึงหยุดนำกระแส ลักษณะของกระแสและแรงดันที่ขาเกตจึงมีลักษณะดังรูปที่ 3.11

โดยทั่วไปแล้ว ค่าความจุของตัวเก็บประจุในตัวเพาเวอร์มอสเฟตนี้เอง จะเป็นตัวกำหนดความเร็วในการเปลี่ยนสถานะของมัน

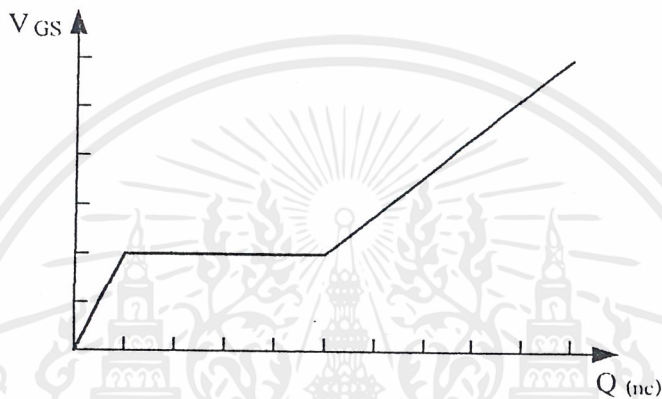


รูปที่ 3.11 แสดงลักษณะของแรงดันและกระแสที่ขาเกตขณะเพาเวอร์มอสเฟตถูกไบอัสให้นำกระแส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### การกำหนดเวลาในการเปลี่ยนสถานะ

ปกติแล้วผู้ผลิตจะให้กราฟค่าแรงสูง  $V_{GS}$  ที่เพิ่มขึ้นในขณะที่ค่าประจุสะสมที่ขาทเพิ่มเพิ่มขึ้น หรือที่เรียกว่า Gate Charge Chart มาในค่าชิตด้วย (รูปที่ 3.12) กราฟนี้มีประโยชน์มากในการคำนวณค่ากระแสไบแอสเกตและเวลาในการเปลี่ยนสถานะของเพาเวอร์มอสเฟต



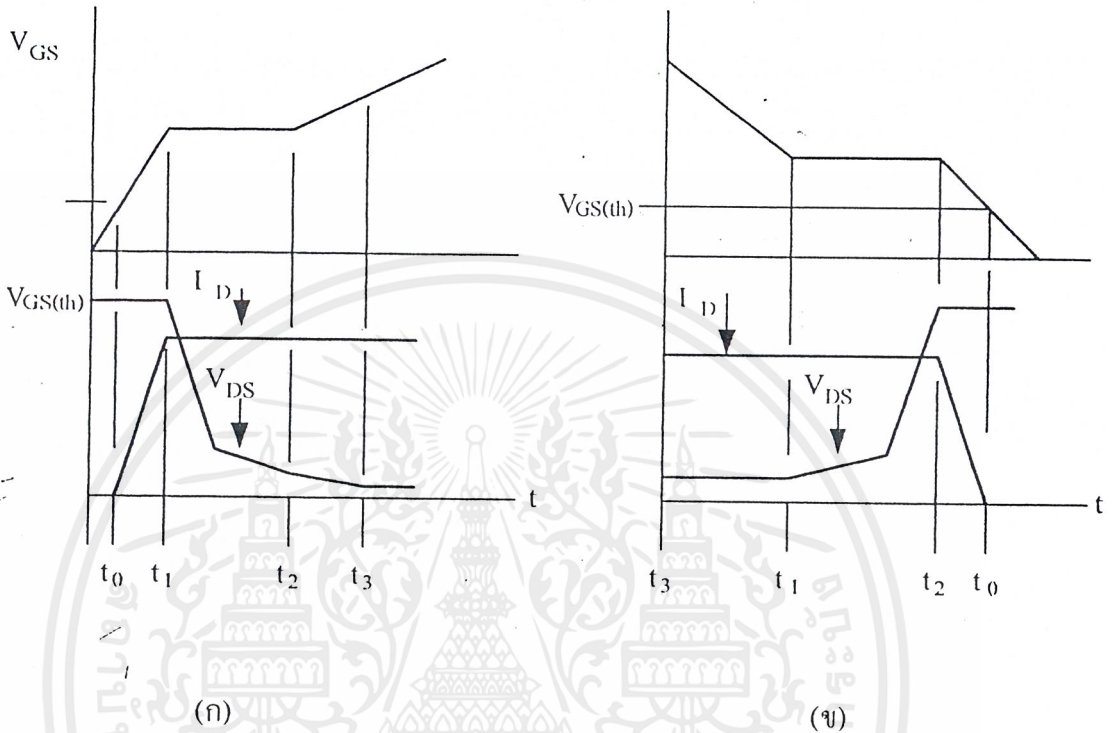
รูปที่ 3.12 แสดงตัวอย่างของ Gate Charge Chart

ลักษณะของกราฟแบ่งออกได้เป็น 3 ช่วง ตามผลของประจุที่เพิ่มขึ้นคือ

1. ช่วงเวลาหน่วงก่อนเริ่มนำกระแส ( Turn-on Delay Time )
2. ช่วงเวลาเริ่มนำกระแส ( Rise Time )
3. ช่วงเวลาสะสมประจุส่วนเกิน ( Excess Charge Time )

เมื่อเพาเวอร์มอสเฟตเริ่มซาร์จประจุที่ขาท จนกระทั่งช่วงเวลาน่วงก่อนเริ่มนำกระแส เมื่อแรงดันที่ขาทมากกว่าแรงดันขีดเริ่ม ( $V_{GS(th)}$ ) จึงจะเริ่มมีกระแสไหลผ่านเดรนและซอร์สที่เวลา  $t_0$  ค่าของแรงดันจะไม่ลดลงจนกว่าจะผ่านเวลาเท่ากับ  $t_1$  แรงดันตกคร่อมเดรนและซอร์ส  $V_{DS}$  สูงสุด เพาเวอร์มอสเฟตจะนำกระแสได้เต็มที่ในช่วงเวลาเท่ากับ  $t_1$  ถึง  $t_2$  นี้เอง แรงดันที่ขาท  $V_{GS}$  จะคงที่จนกว่าเพาเวอร์มอสเฟตจะนำกระแสได้อย่างเต็มที่ ดังแสดงในรูปที่ 4.6 ในเวลาสะสมประจุส่วนเกินค่าความต้านทานระหว่างคอนเดนเซอร์และซอร์สจะมีค่าลดลงได้เรื่อยๆ หากปล่อยให้มีการสะสมประจุต่อไปในช่วงเวลา  $t_1$  ถึง  $t_2$  แต่ประจุที่สะสมเพิ่มขึ้น จะทำให้เกิดการหน่วงขณะเริ่มหยุดนำกระแส เนื่องจากเพาเวอร์มอสเฟตจะต้องใช้เวลามากในการคายประจุส่วนเกินนี้ทิ้งไป ดังนั้น

การขับพาเวอร์ที่ขาเกตด้วยแรงดันสูงเกินความจำเป็นจะทำให้ช่วงเวลาเริ่มหยุดนำกระแสเพิ่มขึ้น  
ด้วย



รูปที่ 3.13 แสดงการชาร์จประจุที่ขาเกตตามเวลาที่มีผลต่อการนำกระแส (ก) และผลเมื่อเริ่มหยุดนำกระแสของพาเวอร์มอสเฟต (ข)

เนื่องจากพาเวอร์มอสเฟตไม่เกิดประจุสะสมขึ้นขณะเริ่มนำกระแส การหยุดนำกระแสของพาเวอร์มอสเฟตจึงทำได้ง่าย ๆ ด้วยการคายประจุที่ขาเกตทิ้งไปเช่นเดียวกับขณะเริ่มนำกระแส และถ้าขนาดกระแสเพื่อชาร์จประจุมีค่าเท่ากัน ยกเว้นหากมีประจุสะสมมากในช่วงเวลา  $t_1$  ถึง  $t_2$  การเริ่มหยุดนำกระแสจะมีเวลาหน่วงเพิ่มขึ้น เนื่องจากต้องใช้เวลาในการคายประจุส่วนเกินทิ้งไป

จากกราฟรูปที่ 3.13 ของพาเวอร์มอสเฟตแต่ละเบอร์ จะนำมาหาช่วงเวลาเริ่มนำกระแส  $t_r$  และช่วงเวลาเริ่มหยุดนำกระแส  $t_f$  ได้จาก

$$t = Q_G / I_G$$

เมื่อ  $Q_G$  คือกระแสที่ประจุจากกราฟ คูณอมป์

$I_G$  คือค่ากระแสที่ได้ชาร์จประจุที่ค่ากระแสคงที่ แอมป์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ กราฟ Gate Charge Chart ที่ได้จากค่าชิต ปกติในการวัดผู้ผลิตจะให้ กระแสเกต  $I_g$  สำหรับซาร์จประจามีค่าคงที่ ในการใช้งานจริงแล้ว วงจรขับเพาเวอร์มอสเฟตอาจ ให้ค่า  $I_g$  ไม่คงที่ จึงจำเป็นต้องใช้ข้อมูลทีนอกเหนือข้อมูลใน Gate Charge Chart ซึ่งจะไม่กล่าว ถึงในที่นี้ อย่างไรก็ตาม ถ้าสามารถออกแบบวงจรขับเพาเวอร์มอสเฟตให้จ่ายกระแสได้มากเพียง การคำนวณที่ได้จากสมการที่ (3.5) ก็นับว่าได้ใกล้เคียง

จะเห็นได้ว่าการขับเพาเวอร์มอสเฟตให้นำกระแสได้นั้น จะต้องมีการซาร์จและคายประจุ ทีขากเกต วงจรขับเพาเวอร์มอสเฟตจะต้องมีลักษณะของการจ่ายและนำกระแส (Source and Sink) ได้ทีประมาณ 200 ถึง 400 มิลลิแอมป์ด้วย และต้องให้แรงดันตกคร่อมทีขากเกต  $V_{GS}$  ค่ามากพอเพือที เพาเวอร์มอสเฟตจะทำงานได้ทีเต็มที่ (ประมาณ 10 โวลต์)

### การพิจารณาเลือกใช้งานเพาเวอร์มอสเฟต

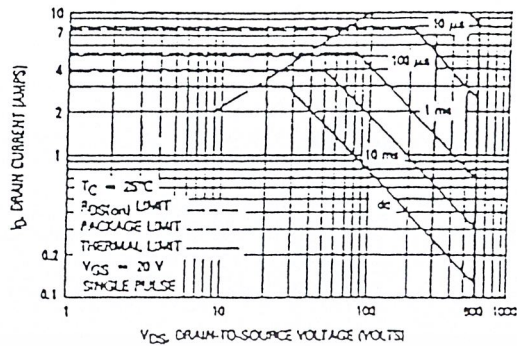
สำหรับเพาเวอร์มอสเฟตการเกิดเซกันคาร์รีเบรคคาวน์เช่นในไบโพลาร์เพาเวอร์ ทรานซิสเตอร์จะไม่เกิดขึ้น เพราะค่าความต้านทานระหว่างดรนและซอร์สขณะนำกระแส  $V_{DS(on)}$  ของมันจะมีค่าเพิ่มขึ้นเมื่ออุณหภูมิเพิ่มขึ้น ทำให้กระแสทีไหลผ่านมีค่าน้อยลงต่างจากกรณีของไบ โพลาร์เพาเวอร์ทรานซิสเตอร์ เมื่ออุณหภูมิเพิ่มขึ้นกระแสจะไหลมากขึ้น ซึ่งมักจะเกิดการพังเสียหายได้อย่างรวดเร็ว เพาเวอร์มอสเฟตจึงมีพิคคความปลอดภัยในกราฟ SOA กว้างกว่าเมื่อเทียบกับ ไบโพลาร์ทรานซิสเตอร์ และเนื่องจากเพาเวอร์มอสเฟตไม่เกิดเซกันคาร์รีเบรคคาวน์ อัตราการทน กำลึงสูญเสียดังสูงที่สุดของมันจะถูกจำกัดด้วยค่าความร้อนทีเกิดขึ้นทีรอยต่อภายในตัวมันเองเท่านั้น

### พิคคปลอดภัย FBSOA

ขีดจำกัดกำลึงทีปลอดภัย FBSOA ของเพาเวอร์มอสเฟตนั้น อาจแบ่งได้เป็น 4 ลักษณะดังนี้

1. ขีดจำกัดของแรงดันทีตกคร่อมดรนและซอร์ส  $V_{DS}$  สูงสุดทีสามารถทนได้ (Maximum Drain-Source Voltage)
2. ค่ากระแสดรน  $I_D$  สูงสุดทีสามารถไหลผ่านได้ และไม่ทำให้เกิดโครงสร้างภายในเสียหาย (Thermal Limit)
3. ค่าจำกัดเนื่องจากค่าความต้านทานขณะนำกระแสหาย ( $V_{DS(on)}$  Limit)
4. ค่าจำกัดการระบายความร้อนของตัวถัง (Package Limit)

เอกสารนี้เป็นเอกสารทีสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

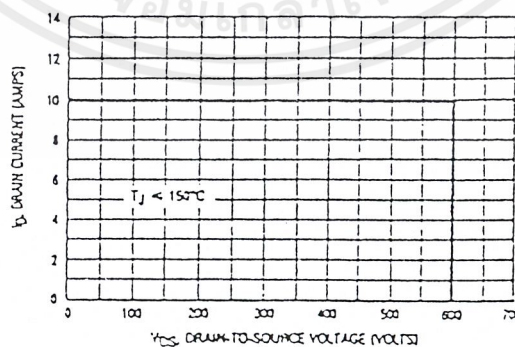


รูปที่ 3.14 แสดงตัวอย่างกราฟฟิวด์ปลอดภัย FBSOA ของเพาเวอร์มอสเฟต(ที่มา: Motorola)

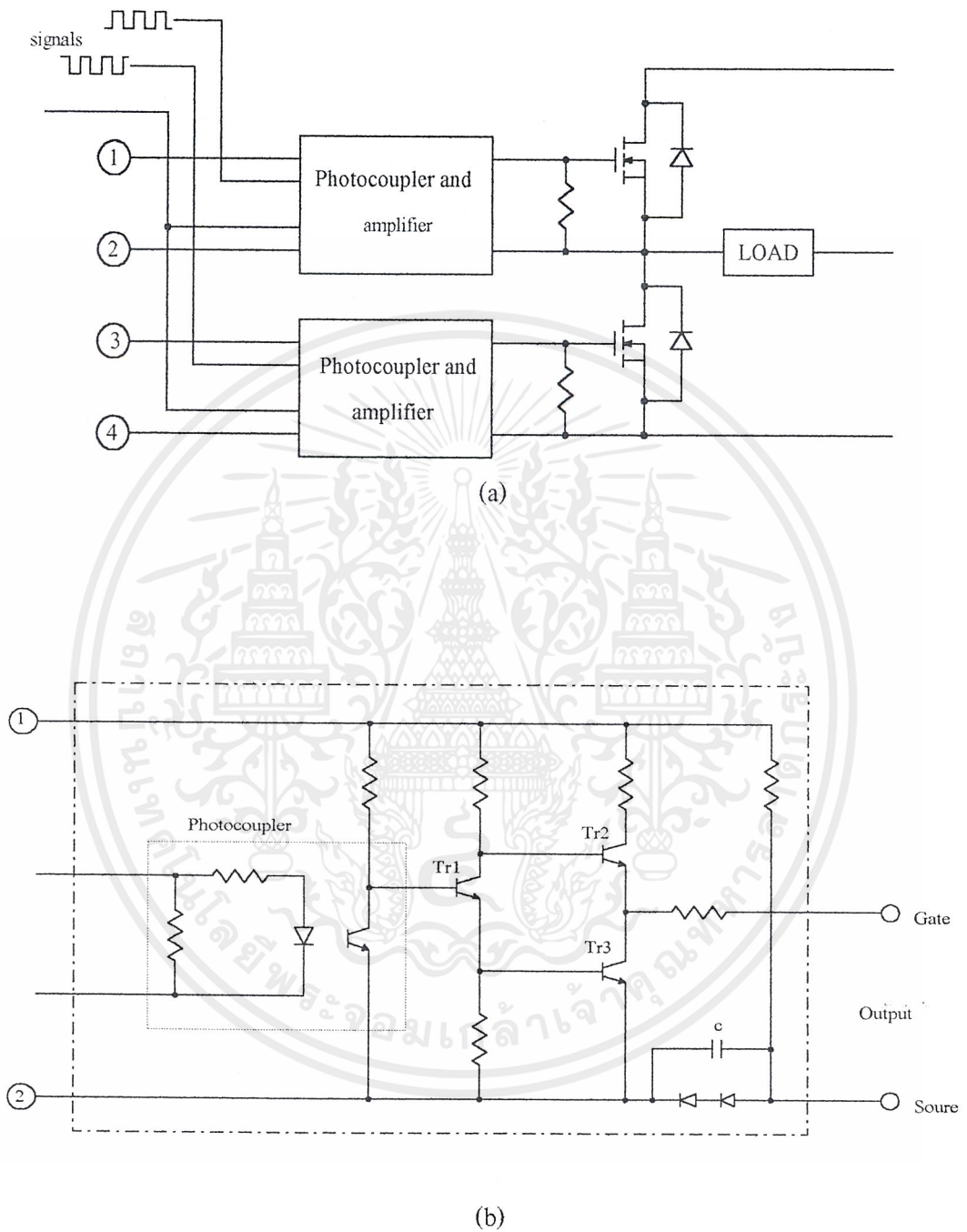
### ฟิวด์ปลอดภัย SSOA

SSOA (Switching Safe Operating Area) เป็นกราฟแสดงฟิวด์กระแสและแรงดันใช้งานที่ปลอดภัยของเพาเวอร์มอสเฟตขณะเปลี่ยนสถานะ การใช้ SSOA สามารถใช้ได้ทั้งในการพิจารณาฟิวด์ปลอดภัยทั้งในช่วงเวลาเริ่มจะนำกระแส และช่วงเวลาที่หยุดนำกระแสของเพาเวอร์มอสเฟต

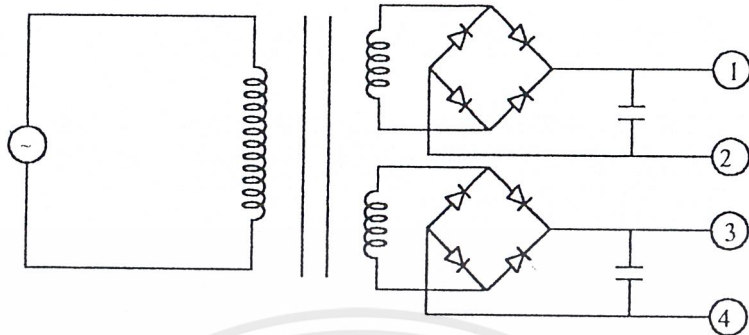
ปกติกราฟ SSOA จะใช้เพื่อพิจารณาเงื่อนไขปลอดภัยของมอสเฟตเช่นเดียวกับกราฟ RBSOA ของไบโพลาร์เพาเวอร์ทรานซิสเตอร์ และเมื่อเปรียบเทียบกับกราฟ RBSOA ของไบโพลาร์เพาเวอร์ทรานซิสเตอร์แล้วจะเห็นว่า SSOA จะมีพื้นที่ปลอดภัยมากกว่าและขีดจำกัดกำลังจะอยู่ที่แรงดันตกคร่อม  $V_{GS}$  สูงสุดและกระแสครอน  $I_D$  สูงสุดเท่านั้น



รูปที่ 3.15 แสดงตัวอย่างกราฟฟิวด์ปลอดภัย SSOA ของเพาเวอร์มอสเฟต(ที่มา: Motorola)  
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

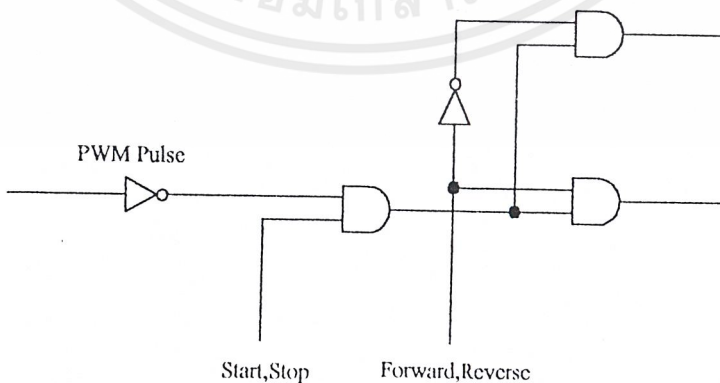


(c)

รูปที่ 3.16 (a) แสดงวงจรการต่อเพาเวอร์มอสเฟตเป็นวงจรสวิทช์  
 (b) แสดงรูปโฟโตคัปเปิล และวงจรขยายสัญญาณ (Amplifier)  
 (c) แสดงการแยกระบบกำลัง

3.5 วงจรควบคุมการหมุนของมอเตอร์

ในที่นี้จะใช้ IC TTL เบอร์ 74LS04 และ 74LS08 ค่อดังรูป โดยการควบคุมให้มอเตอร์หมุนซ้าย หรือขวาจากการควบคุมโดยไมโครคอนโทรลเลอร์โดยให้ขา INTO ควบคุมการหยุดและการเริ่มหมุนของมอเตอร์ และให้ขา INT1 ควบคุมทิศทางการหมุนของมอเตอร์



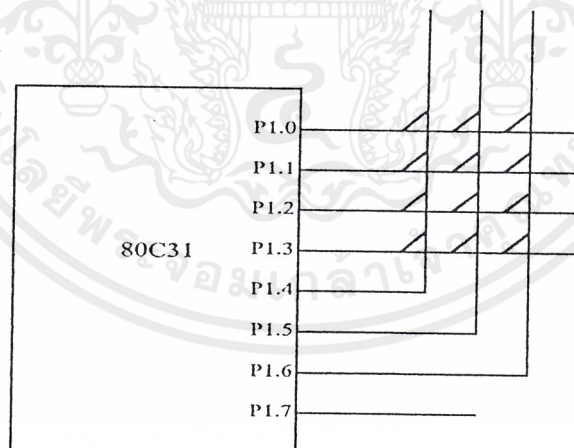
รูปที่ 3.17 แสดงวงจรควบคุมทิศทางการหมุนของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

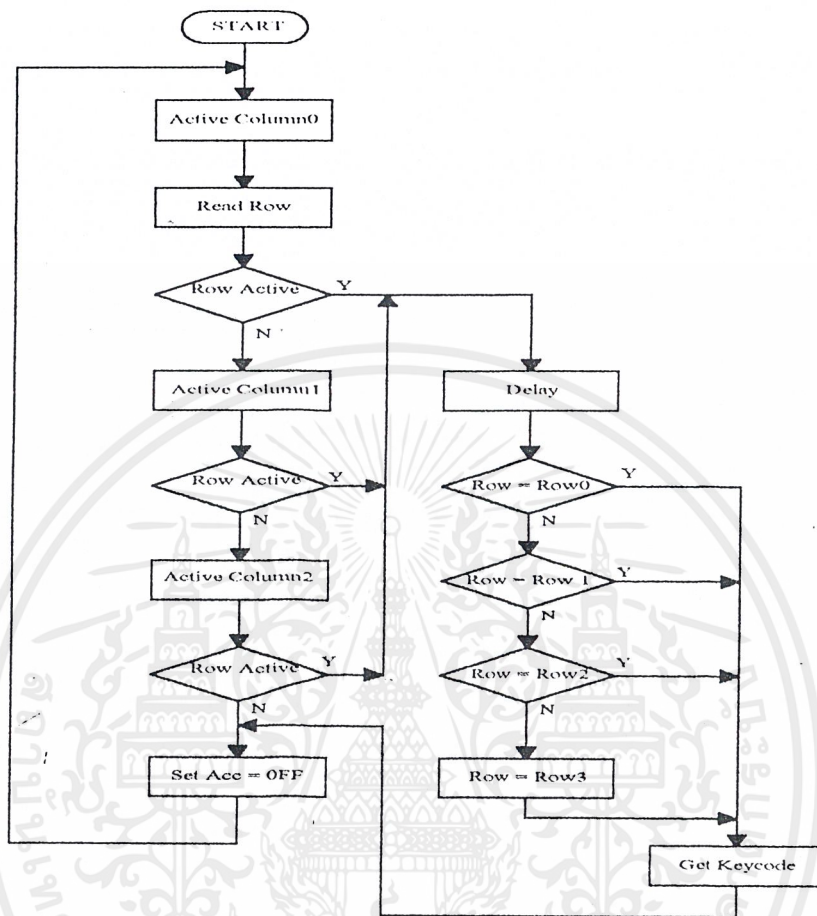
### 3.5 คีย์บอร์ด

คีย์บอร์ดเป็นอุปกรณ์ที่ใช้ในการส่งค่าข้อมูล โดยส่งข้อมูลจากหลักการของเมตริกซ์ (Matrix) กล่าวคือไมโครคอนโทรลเลอร์จะส่งค่าสัญญาณขนาด 8 บิต ออกมาที่พอร์ต (Port) ซึ่งมีการต่อคีย์บอร์ดอยู่โดยค่าสัญญาณขนาด 8 บิตนี้ จะเปลี่ยนแปลงเมื่อกดคีย์ใดคีย์หนึ่ง

หลักการการทำงานของโปรแกรมจะใช้หลักการในการกวาด (Scanning) เพื่อหาว่าสวิตช์ใดที่ถูกกด ซึ่งในการเขียนโปรแกรมสแกนคีย์บอร์ด (Scan Keyboard) นี้จะใช้พอร์ต P1 เป็นพอร์ตอินพุต (Input) เอาท์พุต (Out put) ในการใช้พอร์ต P1 เป็นพอร์ตอินพุตนั้นจะต้องส่งค่า 1 ไปยังแต่ละบิต (Bit) ของพอร์ตเสียก่อนเพื่อกำหนดให้เป็นพอร์ตอินพุตซึ่งในการรับรู้ของไมโครคอนโทรลเลอร์ว่าคีย์ใดถูกกดนั้นจำเป็นที่จะต้องมิตัวแก้อสัญญาณบราวซ์ (Bounce) โดยในโครงงานนี้จะใช้ซอฟต์แวร์เป็นตัวแก้อสัญญาณบราวซ์ซึ่งในการแก้อสัญญาณบราวซ์โดยใช้ซอฟต์แวร์โปรแกรมจะทำงาน โดยการตรวจสอบการกดคีย์ครั้งแรกแล้วรอติกระยะเวลาช่วงหนึ่ง เพื่อที่จะให้ผ่านช่วงเวลาที่เกิดการบราวซ์ผ่านไป แล้วก็ทำการอ่านข้อมูลจากสวิตช์อีกครั้งหนึ่งเพื่อจะนำไปเป็นข้อมูลต่อไป เวลาที่ใช้หน่วงในการแก้บราวซ์ประมาณ 10-40 ms ก็เพียงพอแล้ว



รูปที่ 3.18 แสดงการต่อคีย์บอร์ดขนาด 4x3 กับพอร์ต P1 ของไมโครคอนโทรลเลอร์



รูปที่ 3.19 โฟร์ซาร์ทแสดงการทำงานของโปรแกรมสแกนคีย์บอร์ด

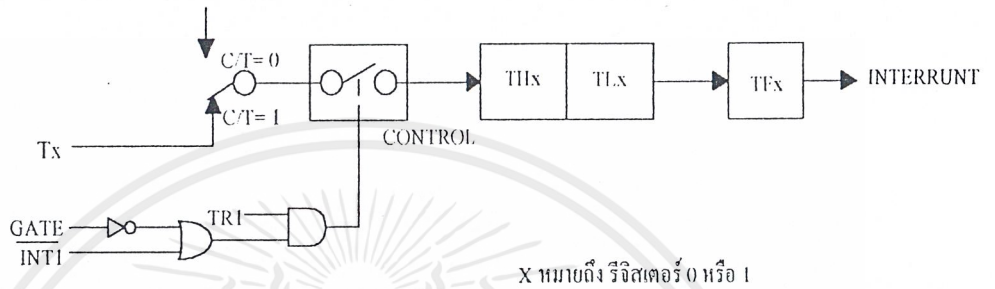
### 3.6 การตรวจจับตำแหน่ง

การตรวจจับตำแหน่งจะใช้ไทม์เมอร์/เคาน์เตอร์ (Timer/Counter) ในไมโครคอนโทรลเลอร์เป็นตัวตรวจจับ ซึ่งในไมโครคอนโทรลเลอร์จะมีไทม์เมอร์/เคาน์เตอร์ อยู่ 2 ชุด คือไทม์เมอร์/เคาน์เตอร์ 0 และ 1 ในแต่ละชุดจะมีรีจิสเตอร์ขนาด 8 บิต อยู่ 2 ตัว เพื่อเก็บค่าการนับของไทม์เมอร์/เคาน์เตอร์ ได้สูงสุด 16 บิต ในไทม์เมอร์/เคาน์เตอร์ 0 และ 1 จะมีรีจิสเตอร์ คือ TH0,TLO และ TH1,TLI ตามลำดับ โดย TH0,TH1 จะเก็บค่าของการนับ 8 บิตบน ส่วนTLO,TLI จะเก็บค่าของการนับ 8 บิตล่าง ซึ่งจะสามารถควบคุมการทำงานของ ไทม์เมอร์/เคาน์เตอร์ ในโหมดไทม์เบอ์หรือโหมดของเคาน์เตอร์ได้ โดยการกำหนดในรีจิสเตอร์ชื่อ TMOD (Timer/Counter Mode control Register) ซึ่งในโครงงานนี้จะใช้ โหมดเคาน์เตอร์ โดยที่โหมดเคาน์เตอร์จะถูกเพิ่มค่าทีละหนึ่ง เมื่อป้อนสัญญาณคล็อกจากภายนอกเข้ามา 1 ลูก (เข้ามาทางขา To หรือ T1 ของไมโครคอนโทรลเลอร์ เมื่อเคาน์เตอร์นับครบก็จะทำการเซ็ท TFX ภายในรีจิสเตอร์ TCON (Timer/Counter Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Register) เพื่อแสดงการเกิดโอเวอร์โฟลว์ (Overflow) ของเคาน์เตอร์ และจะถูกเคลียร์เองเมื่อ CPU ย้ายไปที่โปรแกรมบริการอินเทอร์รัปต์ หรือใช้คำสั่ง CLR TFX (x หมายถึง 0 หรือ 1)

โหมดการทำงานของเคาน์เตอร์ (Counter Mode) จะมีด้วยกัน 4 โหมดซึ่งในที่นี้จะเลือกในโหมดที่ 1 (เป็นโหมด 16 บิต)



รูปที่ 3.20 แสดงบล็อกไดอะแกรมการทำงานของเคาน์เตอร์ (Mode 1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การทดสอบโครงการ

#### 4.1 การทดสอบหาค่าพารามิเตอร์ต่างๆ ของมอเตอร์

##### อุปกรณ์ที่ใช้

1. DC. permanent magnet motor(Hitachi Model D05B601E 200) 31.3 V. 3 A.  
4600 rpm. Encoder 200 pulse per revolution. Torque 1.4 kg.cm. INS.CL. E
2. แหล่งจ่ายไฟ 0-30 V. 1 A.
3. DC. Ammeter 0-3 A.
4. Tachometer

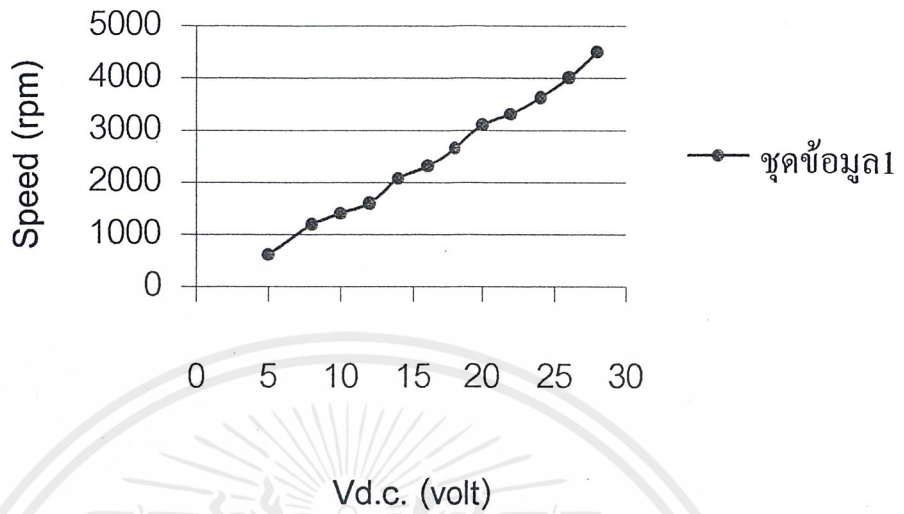
##### วิธีทดสอบ

1. ต่อแหล่งจ่ายและแอมมิเตอร์เข้ากับมอเตอร์
2. ค่อยๆป้อนแรงดันระดับต่างๆให้กับมอเตอร์ อ่านค่ากระแสและวัดความเร็วรอบ  
บันทึกค่า
3. วัดค่าความต้านทานโดยใช้มัลติมิเตอร์วัดที่ขั้วมอเตอร์

##### ผลการทดสอบ

เนื่องจากมอเตอร์ที่ใช้ในโครงการมีสภาพไม่สมบูรณ์ ทำให้ที่แรงดันต่ำๆ ไม่สามารถออก  
ตัวได้ แต่สามารถทำการทดสอบได้จนเสร็จได้ผลการทดสอบดังนี้

- วัดค่าความต้านทานของมอเตอร์ที่ขั้วของมอเตอร์โดยใช้มัลติมิเตอร์ได้ 2.2 โอห์ม
- ป้อนแรงดันให้มอเตอร์ บันทึกความเร็วและคำนวณค่าพารามิเตอร์ต่างๆ ของมอเตอร์



รูปที่ 4.1 กราฟแสดงความสัมพันธ์ระหว่างแรงดันและความเร็วรอบของมอเตอร์

ตารางที่ 4.1 แสดงค่าพารามิเตอร์ของมอเตอร์ที่ได้จากการทดลอง

$V_i$ n (Volt)	I (Amp)	$\omega$ (rps)	$R_a$ (ohm)	$K_m$ (v-s/rad)	$E_g$ (Volt)
5	0.22	62.8	2.2	0.0719	4.516
10	0.29	146.53	2.2	0.0639	9.363
15	0.39	230.27	2.2	0.0614	14.142
20	0.44	324.46	2.2	0.0586	19.032
25	0.48	413.43	2.2	0.0579	23.994

ตัวอย่างการคำนวณ

$$E_g = V_t - I_a R_a$$

$$= 15 - (0.39)(2.2) = 14.142 \text{ v}$$

$$\omega = 2200 * 2\pi/60 = 230.27 \text{ rps}$$

$$K_m = E_g/\omega$$

$$K_m = 14.142/230.27 \text{ v-s/rad}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น หาค่าเฉลี่ยได้

$$K_m (\text{mean}) = (0.0719 + 0.0639 + 0.0614 + 0.0586 + 0.0579) / 5$$

$$= 0.0627 \text{ v-s/rad}$$

-ทำการต่อวงจรดังรูป เพื่อทดสอบหาค่า Inductance ของมอเตอร์  
กำหนดให้

$$V_i = 30 \text{ V}$$

$$R_x = 0.47 \Omega, 5 \text{ W}$$

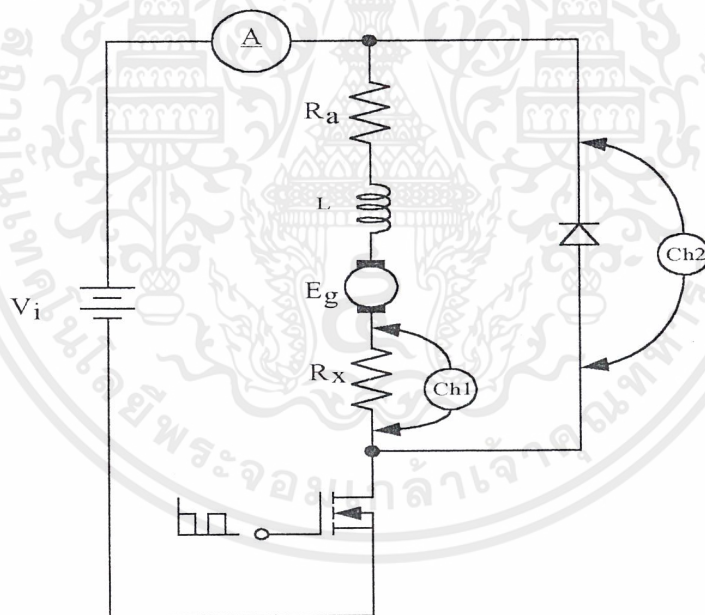
$R_a$  = ค่าความต้านทานของมอเตอร์

$L$  = ค่า Inductance ของมอเตอร์

$$\text{Ch1} = 2\text{v/div} * 10, \text{Ch2} = 0.5\text{v/div}, \text{Time/div} = 10 \mu\text{s/div}$$

$$\text{Pulse generator} = 0-5 \text{ v}, \text{duty cycle } 50\% \text{ } 20 \text{ kHz}$$

$V_{sw}$  คือ แรงดันที่ตกคร่อม Mosfet ขณะ ON  $\approx 0\text{V}$



รูปที่ 4.2 แสดงวงจรทดสอบค่า Inductance ของมอเตอร์

$V_i$ (volt)	$E_g$	Ch1		Ch2 (peak)	$V_{sw}$	$V_t$	$L$ (mH)
		$I_{max}$ (A)	$I_{min}$				
30	14.23	0.383	0.128	30	0	15	1.48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการคำนวณ

$$I_{\max} = I_{\min} (1 - e^{-(R_a + R_x)t_{on} / L_a}) + \left( \frac{V + E_g + V_{sw}}{R_a + R_x} \right) (1 - e^{-(R_a + R_x)t_{on} / L_a})$$

$$0.383 = 0.128 e^{-(2.67 \cdot 25 \cdot 10^{-6}) / L_a} + \left( \frac{30 - 14.23}{2.2 + 0.47} \right) (1 - e^{-2.67 \cdot 25 \cdot 10^{-6} / L_a})$$

$$L_a = 1.48 \text{mH}$$

### สรุปผลการทดลอง

จากการทดลองสามารถหาค่าพารามิเตอร์ต่างๆของมอเตอร์ได้ตามที่ต้องการ และจากการหาค่า  $L_a$  และ  $R_a$  ทำให้สามารถตัดค่ากระแสรีเบิ้ลทิ้งได้จากสมการ  $T < L_a/R_a$ ;  $50\mu\text{s} < 627\mu\text{s}$  ซึ่งเป็นค่าความถี่ที่ใช้ในการทดลองเท่านั้น เนื่องจากกระแสมีความต่อเนื่องที่ 50 เฮอร์เซ็นต์ แต่ความถี่ที่ใช้งานจริง คือ 30 kHz เพราะที่ความถี่ที่ความเร็วต่ำสุดของโครงงานนี้ กระแสยังต่อเนื่องอยู่ เป็นสภาพที่ทำให้มอเตอร์หมุนได้อย่างราบเรียบ

### 4.2 การทดสอบการสร้างสัญญาณพัลส์วิดท์มอดูเลชัน (PWM)

โปรแกรมทดสอบการสร้างสัญญาณ PWM

```

org      0000h
chn0    equ      0fc00h
chn1    equ      0fc01h
ctrl    equ      0fc03h

mov     a,#37h
mov     dptr,# ctrl
mov     @dptr,a      ;square wave mode
mov     dptr,chn0
mov     a,#33h

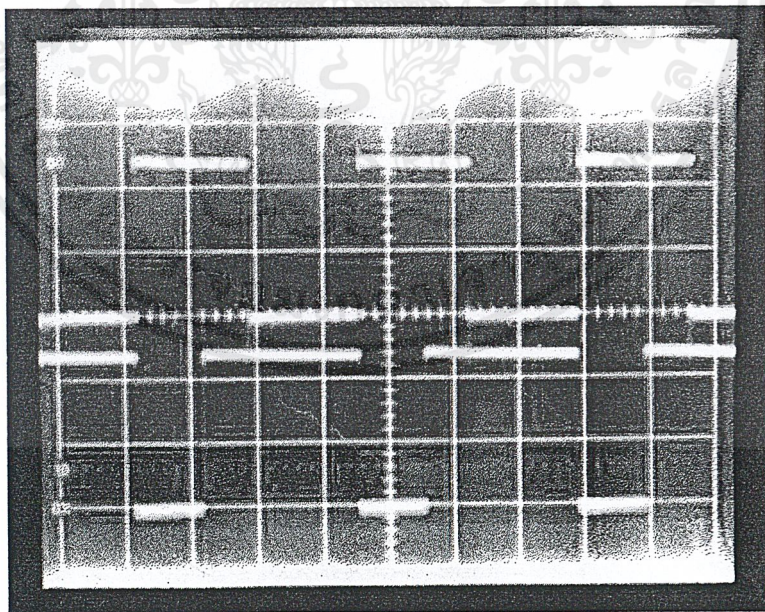
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

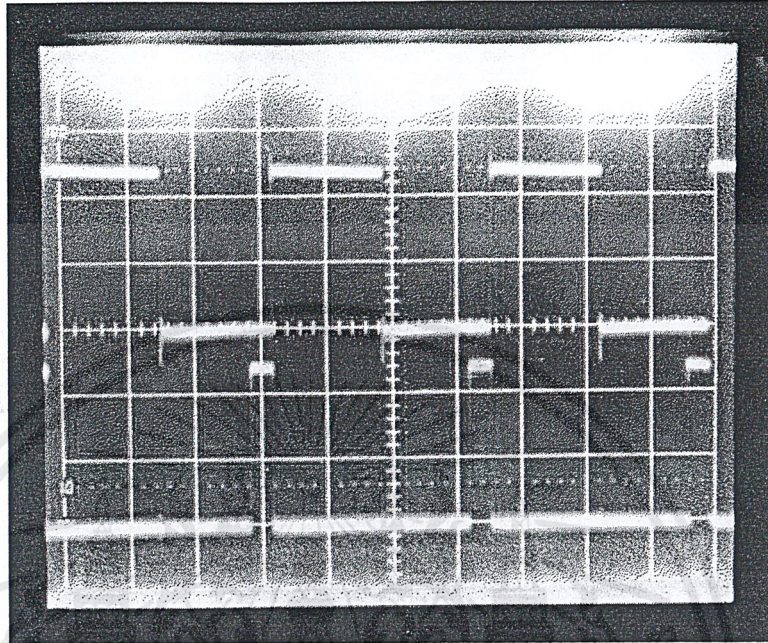
movx @dptr,a
mov a,#03h
movx @dptr,a ;20 kHz square wave
mov a,#73h
mov dptr,#ctrl
movx @dptr,a ;monosatable mode
mov dptr,#chn1
movx @dptr,a
mov a,#50h
movx @dptr,a
mov a,#02h
movx dptr,a
end

```



รูปที่ 4.3 แสดงสัญญาณพัลส์วิดท์มอดูเลชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงสัญญาณพัลส์วัดที่มอดูเลชั่น

หมายเหตุ : Ch1, Ch2 = 2 v/div, Time/div 10  $\mu$ s

#### 4.4 การทดสอบภาคขับเคลื่อนมอเตอร์

##### อุปกรณ์ที่ใช้

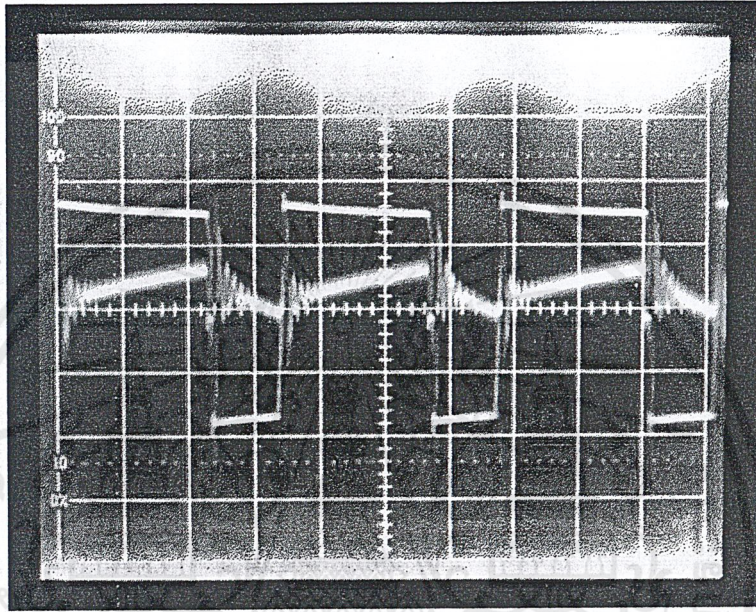
1. คอมพิวเตอร์
2. แผงวงจรของโครงการ
3. ชุดขับเคลื่อนสายพาน
4. Oscilloscope

##### วิธีทดสอบ

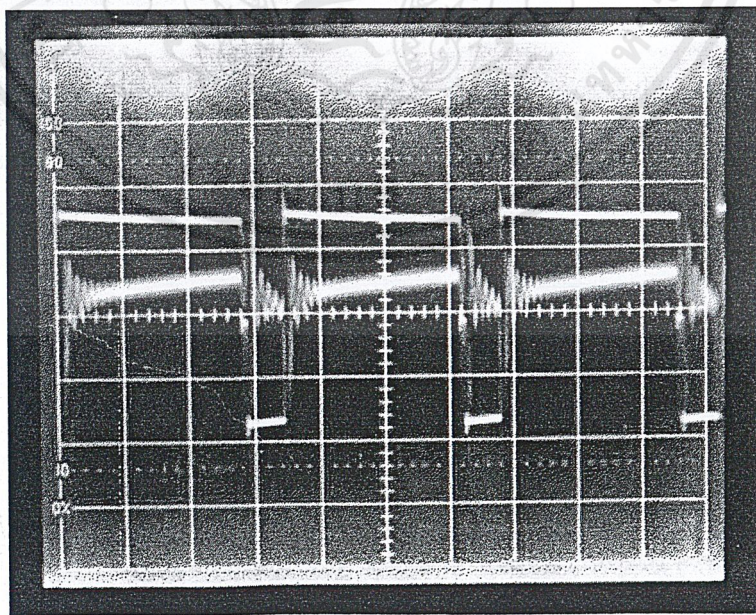
ต่อชุดไมโครคอนโทรลเลอร์ผ่านอีพรอมอีโมเลเตอร์ และต่อภาคขับเคลื่อนมอเตอร์เข้าจับชุดสร้างสัญญาณพัลส์วัดที่มอดูเลชั่น และต่อชุดขับเคลื่อนสายพานเข้ากับแผงวงจร แล้วทำการวัดรูปคลื่นสัญญาณของกระแสและแรงดัน โดยใช้ความต้านต่ออนุกรมในวงจรเพื่อวัดรูปคลื่นกระแส

### ผลการทดสอบ

เมื่อทำการจับเคลื่อนมอเตอร์ที่ความถี่ 30 KHz ที่ความถี่ที่ต่างกัน จะเห็นได้ว่าสัญญาณกระแสต่อเนื่อง มอเตอร์ไม่มีเสียงดังรบกวน และเมื่อทดลองที่ความถี่ 20 kHz จะเห็นว่าสัญญาณกระแสไม่ต่อเนื่อง สัญญาณรูปคลื่นต่างๆ ได้แสดงไว้ดังรูป

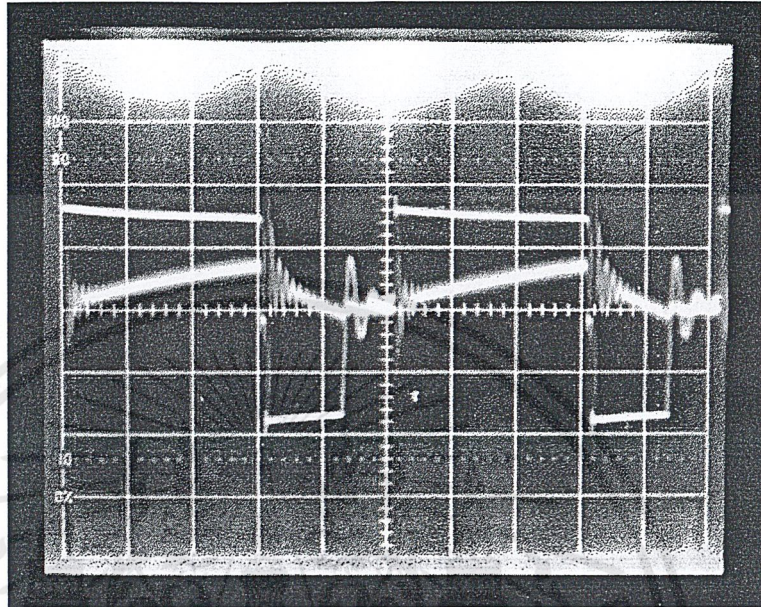


รูปที่ 4.5 แสดงสัญญาณแรงดันและกระแสที่ความเร็วรอบ 1500 rpm ที่ความถี่ 30 kHz



รูปที่ 4.6 แสดงสัญญาณแรงดันและกระแสที่ความเร็วรอบ 3000 rpm ที่ความถี่ 30 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 แสดงสัญญาณแรงดันและกระแสที่ความเร็วรอบ 1500 rpm ที่ความถี่ 20 kHz

หมายเหตุ : Ch1(แรงดัน) คือ สัญญาณพัลส์สี่เหลี่ยม 2 v/div\*10

Ch2 (กระแส) คือ สัญญาณ Sawtooth 0.5v/div

Time/div 10  $\mu$ s

#### สรุปผลการทดลอง

จากความถี่ที่ใช้ในการทำงาน 30 kHz ทำให้ที่ความเร็วต่ำสุดของ โครงาน กระแสมีความต่อเนื่อง เป็นสภาพที่จะทำให้มอเตอร์หมุนได้อย่างราบเรียบ เมื่อเปรียบเทียบกับที่ความถี่ 20 kHz จะเห็นได้ว่ากระแสไม่มีความต่อเนื่องมอเตอร์จะหมุนได้ไม่ราบเรียบเท่าที่ควร

จากวิธีการขับมอเตอร์ที่ใช้ในโครงานนี้ เป็นแบบให้Mosfet คู่ใดคู่หนึ่ง ON-OFF ตลอดเวลา (ในการหมุนไปทิศทางหนึ่ง) ซึ่งต่างกับการขับแบบเซอร์โว แต่รูปคลื่นที่ได้คล้ายกับการขับแบบเซอร์โว ทำให้การขับของโครงานนี้ไม่สามารถควบคุมตำแหน่งได้อย่างแม่นยำเหมือนการขับแบบเซอร์โว

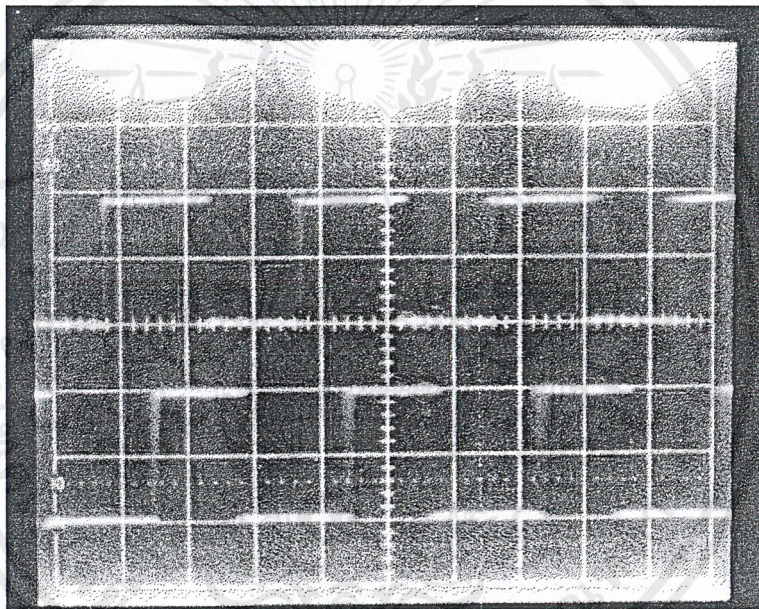
### 4.3 การทดสอบสัญญาณพัลส์จากเอ็นโค้ดเดอร์

#### อุปกรณ์

1. มอเตอร์ที่ใช้ในโครงการ
2. แหล่งจ่าย DC. 30 V. 1 A
3. Oscilloscope

#### วิธีทดสอบ

ต่อแหล่งจ่ายใช้แรงดัน 5 V เข้ากับเอ็นโค้ดเดอร์ที่ตัวมอเตอร์ และจ่ายแรงดันเล็กน้อยให้มอเตอร์ วัดสัญญาณพัลส์ทั้งสองของเอ็นโค้ดเดอร์



รูปที่ 4.8 แสดงสัญญาณพัลส์ทั้ง 2 Channel จากเอ็นโค้ดเดอร์

#### ผลการทดสอบ

สัญญาณพัลส์มีความถี่สูงมากถ้ามอเตอร์หมุนเร็ว ดังนั้นจึงให้แรงดันแก่มอเตอร์ที่พอเหมาะเพื่อให้มอเตอร์หมุนช้าพอที่จะสามารถวัดสัญญาณได้ ดังแสดงในรูปที่ 4.8

#### สรุปผลการทดลอง

สัญญาณพัลส์ที่ได้มีระดับแรงดันเพียงพอที่จะนำไปใช้งาน(0-5V) การนำไปใช้งานจะใช้เพียงเฟสเดียวในการนับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.5 การทดสอบโครงงานทั้งหมด

##### อุปกรณ์

1. แผงวงจรชุดขับและชุดสายพานขับเคลื่อน
2. คอมพิวเตอร์
3. โหลดน้ำหนักต่างๆ

##### วิธีทดสอบ

ตั้งการทำงานจากคอมพิวเตอร์โดยผ่านอีพรอมอีมูเลเตอร์ ให้มอเตอร์ทำงานที่ความเร็วและระยะทางต่างๆ และจดบันทึกค่า

##### ผลการทดสอบ

ผลการทดลองแสดงดังตาราง

ตารางที่ 4.2 แสดงการทดสอบขณะ NO LOAD ที่ความเร็วสายพาน 100 RPM.

ระยะทางมาตรฐาน (cm)	ผลการทดลองครั้งที่			ระยะทางเฉลี่ย (cm)	ค่าความคลาดเคลื่อน (%)
	1	2	3		
5	5.30	5.40	5.30	5.33	6.67
10	10.40	10.40	10.30	10.37	3.67
15	15.20	15.20	15.30	15.23	1.56
20	20.30	20.30	20.20	20.27	1.33
25	25.30	25.30	25.40	25.33	1.33
30	30.00	30.20	30.20	30.13	0.44
35	35.00	35.20	35.00	35.07	0.19
40	39.90	40.10	39.90	39.97	-0.08
45	45.10	44.90	45.20	45.07	0.15
50	50.20	50.20	50.10	50.17	0.33
55	55.20	55.20	55.10	55.17	0.30
60	60.00	59.80	60.20	60.00	0.00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 การทดสอบขณะ NO LOAD ที่ความเร็วสายพาน 160RPM.

ระยะทาง มาตรฐาน	ผลการทดลองครั้งที่			ระยะทางเฉลี่ย	ค่าความคลาดเคลื่อน
	1	2	3		
(cm)				(cm)	(%)
5	5.20	5.30	5.30	5.27	5.33
10	10.40	10.50	10.30	10.40	4.00
15	15.20	15.30	15.30	15.27	1.78
20	20.20	20.20	20.20	20.20	1.00
25	25.20	25.10	25.20	25.17	0.67
30	30.00	30.10	30.20	30.10	0.33
35	35.00	35.10	35.00	35.03	0.10
40	40.10	40.10	40.10	40.10	0.25
45	45.10	45.20	45.20	45.17	0.37
50	50.20	50.20	50.20	50.20	0.40
55	55.10	55.20	55.10	55.13	0.24
60	60.10	60.10	60.20	60.13	0.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 แสดงการทดสอบขณะขับ LOAD ขนาด 250กรัม ที่ความเร็วสายพาน 100 RPM.

ระยะทาง มาตรฐาน (cm)	ผลการทดลองครั้งที่			ระยะทางเฉลี่ย (cm)	ค่าความคลาดเคลื่อน (%)
	1	2	3		
5	5.20	5.20	5.30	5.23	4.67
10	10.30	10.30	10.30	10.30	3.00
15	15.20	15.20	15.20	15.20	1.33
20	20.20	19.90	20.10	20.07	0.33
25	24.90	24.80	24.60	24.77	-0.93
30	29.80	29.90	29.80	29.83	-0.56
35	34.90	34.90	35.00	34.93	-0.19
40	39.90	39.80	39.90	39.87	-0.33
45	44.90	44.90	45.00	44.93	-0.15
50	49.50	49.80	49.70	49.67	-0.67
55	54.90	54.80	54.70	54.80	-0.36
60	59.60	59.70	59.60	59.63	-0.61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.5 แสดงการทดสอบขณะขับ LOAD ขนาด 250 กรัม ที่ความเร็วสายพาน 160 RPM.

ระยะทาง มาตรฐาน	ผลการทดลองครั้งที่			ระยะทางเฉลี่ย	ค่าความคลาดเคลื่อน
	1	2	3		
(cm)					
5	5.20	5.30	5.30	5.27	5.33
10	10.30	10.30	10.20	10.27	2.67
15	15.20	15.00	15.10	15.10	0.67
20	20.10	20.10	20.00	20.07	0.33
25	25.20	25.30	25.10	25.20	0.80
30	30.10	30.00	29.90	30.00	0.00
35	34.90	34.90	34.80	34.87	-0.38
40	39.60	39.80	39.70	39.70	-0.75
45	44.80	44.90	44.80	44.83	-0.37
50	49.70	49.80	49.70	49.73	-0.53
55	54.70	54.70	54.50	54.63	-0.67
60	59.40	59.70	59.80	59.63	-0.61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.6 แสดงการทดสอบขณะรับ LOAD ขนาด 500 กรัม ที่ความเร็วสายพาน 100 RPM.

ระยะทาง มาตรฐาน	ผลการทดลองครั้งที่			ระยะทางเฉลี่ย	ค่าความคลาดเคลื่อน
	1	2	3		
(cm)				(cm)	(%)
5	4.8	5	4.9	4.97	-0.67
10	9.8	9.9	9.7	9.8	-2
15	15.1	14.8	15	14.97	0.22
20	20.2	19.7	19.9	19.9	-0.33
25	24.5	24.9	25	24.8	-0.8
30	29.8	29.9	29.5	29.7	-0.89
35	34.5	34.9	34.5	34.6	-1.04
40	39.5	39.7	39.6	39.6	-1
45	44.6	44.7	44.5	44.6	-0.81
50	49.5	49.7	49.4	49.5	-0.93
55	54.5	54.6	54.4	54.5	-0.90
60	59.5	59.7	59.4	59.5	-0.77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.7แสดงการทดสอบขณะจับ LOAD ขนาด 500 กรัม ที่ความเร็วสายพาน 160 RPM.

ระยะทาง มาตรฐาน	ผลการทดลองครั้งที่			ระยะทางเฉลี่ย	ค่าความคลาดเคลื่อน
	1	2	3		
(cm)				(cm)	(%)
5	5.3	5.5	5.2	5.33	6.7
10	10.5	10.4	10.4	10.433	4.33
15	15.2	15.3	15.2	15.23	1.55
20	20.4	20.1	20.2	20.23	1.16
25	25	25.1	25	25.03	0.13
30	29.9	30	30	29.97	-0.11
35	34.8	34.9	34.9	34.87	-0.38
40	39.5	39.6	39.8	39.63	-0.92
45	44.6	44.8	44.8	44.83	-0.37
50	49.6	49.7	49.6	49.6	-0.73
55	54.7	54.8	54.6	54.7	-0.24
60	59.4	59.5	59.5	59.47	-0.88

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.8 แสดงการทดสอบขณะขับ LOAD ขนาด 750 กรัม ที่ความเร็วสายพาน 100 RPM.

ระยะทาง มาตรฐาน	ผลการทดลองครั้งที่			ระยะทางเฉลี่ย	ค่าความคลาดเคลื่อน
	1	2	3		
(cm)				(cm)	(%)
5	5	4.9	5	4.96	-0.66
10	10.1	15	15	10.06	0.66
15	14.7	15	15	14.96	-0.22
20	19.9	19.8	19.9	19.87	-0.67
25	24.5	24.7	24.6	24.63	-1.46
30	29.6	29.5	29.6	29.57	-1.44
35	34.6	34.5	34.5	34.53	-1.33
40	39.5	39.6	39.5	39.53	-1.16
45	44.3	44.5	44.5	44.43	-1.26
50	49.5	49.4	49.4	49.43	-1.13
55	54.5	54.4	54.4	54.6	-0.96
60	59.5	59.4	59.4	59.43	-0.94

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.9แสดงการทดสอบขณะจับ LOAD ขนาด 750 กรัม ที่ความเร็วสายพาน 160 RPM.

ระยะทาง มาตรฐาน	ผลการทดลองครั้งที่			ระยะทางเฉลี่ย	ค่าความคลาดเคลื่อน
	1	2	3		
(cm)				(cm)	(%)
5	5.2	5.1	5.3	5.2	4
10	10.5	10.3	10.3	10.36	3.67
15	15.4	15.3	15.2	15.3	2
20	19.8	20.1	20.1	20	0
25	25.3	25.1	25.2	25.2	0.8
30	30.1	30.2	30.1	30.13	0.44
35	35.3	35.2	35.1	35.2	0.57
40	40.1	40.3	40.3	40.233	-0.58
45	44.8	44.7	44.9	44.8	-0.44
50	49.8	49.7	49.5	49.6	-0.66
55	54.7	54.4	54.5	54.53	-0.84
60	59.8	59.7	59.8	59.76	-0.38

### สรุปผลการทดลอง

การทดสอบที่การทำงานต่างๆ ทั้ง ไม่มีโหลดและมีโหลด ค่าผิดพลาดที่เกิดขึ้นที่ระยะทางและความเร็วเดียวกันไม่เท่ากัน เนื่องมาจากสาเหตุ เช่น ชุดขับเคลื่อนสายพานมีการติดตัว ความฝืดจากโหลดกับรางสายพาน ค่าผิดพลาดจากโปรแกรมคำนวณ การสั่งเบรคมอเตอร์ เป็นต้น ซึ่งวิธีที่ใช้ในการเบรคจะใช้วิธีการสร้างทอร์คในทิศทางตรงกันข้าม โดยใช้ระยะเวลาคงที่ๆความเร็วหนึ่งทำให้ค่าผิดพลาดในแต่ละระยะทางมีค่าไม่คงที่ และการจับมอเตอร์แบบนี้ค่าผิดพลาดที่เกิดขึ้นจะมากกว่าการจับแบบเซอร์โว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลและแนวทางในการพัฒนา

1. จากผลการทดลองในส่วนการสร้างสัญญาณพัลส์วิดท์มอดูเลชัน สามารถที่จะกำหนดคิวดีไซ์เกิดต่างๆ ได้ง่ายมาก เพียงแค่โปรแกรมค่าการนับให้กับ 8254 เข้าไปโดยที่ความถี่ในการมอดูเลชันคงที่ และขณะเดียวกันเราก็สามารถที่จะเปลี่ยนความถี่ในการขับเคลื่อนมอเตอร์ได้อย่างง่ายดายเช่นกัน

2. ในส่วนการตรวจจับตำแหน่งมีความแน่นอนพอสมควร ค่าที่คลาดเคลื่อนเกิดขึ้นเนื่องจากตัวฮาร์ดแวร์ หน่วยประมวลผลกลาง (CPU) และส่วนของ โปรแกรม ค่าความคลาดเคลื่อนที่เกิดขึ้นเนื่องจากตัวฮาร์ดแวร์ คือตัวสายพานเวลาหมุนจะมีความตึงหย่อนของสายพานอยู่ในตัวส่งผลทำให้เกิดค่าผิดพลาดขึ้น ในส่วนของหน่วยประมวลผลกลางถ้าจะให้ค่าความผิดพลาดลดน้อยลงควรใช้หน่วยประมวลผลกลางให้มีความเร็วมากกว่าในโครงการนี้ และค่าความผิดพลาดที่เกิดจากตัวของโปรแกรมเกิดจากส่วนของการคำนวณค่าระยะทางออกมาเป็นทั้งจำนวนเต็มและทศนิยม ดังนั้นค่าทศนิยมที่เกิดขึ้นนี้ไม่อาจเขียนเป็นคำสั่งให้หน่วยประมวลผลเข้าใจได้ทำให้เกิดค่าความคลาดเคลื่อนในส่วนนี้ขึ้น

3. ส่วนการติดต่อระหว่างผู้ใช้กับตัวหน่วยประมวลผลกลางเป็นไปได้อย่างดี

4. การขับเคลื่อนมอเตอร์ของโครงการถึงแม้ว่าจะไม่ใช่การขับเคลื่อนแบบเซอร์โวแต่ก็สามารถควบคุมตำแหน่ง

ได้ดีพอสมควรสามารถพัฒนาและนำไปประยุกต์ใช้กับงานที่ไม่ต้องการความละเอียดมากได้

#### ข้อเสนอแนะสำหรับการปรับปรุง

- ควรใช้ภาษาอื่นในการเขียนโปรแกรม เพราะจะเห็นว่าในโครงการนี้ซึ่งใช้ภาษาแอสเซมบลีจะมีข้อจำกัดในการเขียนอยู่มาก และควรจะมีส่วนควบคุมความเร็วของมอเตอร์ด้วยซึ่งส่วนนี้ผู้ทำโครงการนี้ได้ทดลองการควบคุมความเร็วโดยใช้เอ็นโค้ดเดอร์ส่งสัญญาณพัลส์ให้เคาน์เตอร์ 8254 จับจำนวนพัลส์ในช่วงเวลาหนึ่ง (เวลาในการ Sampling) ซึ่งช่วงเวลาในการจับสัญญาณพัลส์จากเอ็นโค้ดเดอร์ควรจะมีค่าน้อยเพื่อการแก้ไขความเร็วได้เร็วขึ้น ในส่วนนี้ผู้ทำโครงการนี้จะเสนอแนวความคิดการควบคุมความเร็วแบบดิจิตอล เพื่อเป็นแนวทางในการพัฒนาต่อไป

- ใช้ IC8254 ให้เคาน์เตอร์ตัวที่ 0 ทำงานโหมด 0 เพื่อตั้งค่าเวลาในการ sampling และให้เคาน์เตอร์ ตัวที่ 1 ทำงานในโหมด 0 เช่นกันเพียงแต่ต้องตั้งค่าโปรแกรมการแลทซ์ให้ ซึ่งทำให้ค่าที่อยู่ในเคาน์เตอร์ถูกแลทซ์ให้กับบริจิสเตอร์ภายใน 8254 และเมื่อทำการอ่านเคาน์เตอร์ ค่านี้จะถูกอ่านออกไป ถ้าการอ่านของข้อมูลไม่ได้อยู่ในโหมดแลทซ์อาจจะเกิดค่าผิดพลาดขึ้น ค่าที่อ่านได้จะเอาไปคำนวณหาความเร็วและเปรียบเทียบกับค่าความเร็วที่ตั้งค่าไว้ ถ้ามีค่าไม่เท่ากันก็ทำการแก้ไขความเร็ว การแก้ไขความเร็วช้าหรือเร็วขึ้นอยู่กับ ช่วงเวลาในการ sampling ความละเอียดของสัญญาณพัลส์จากเอ็น โค้ดเดอร์ และความเร็วของซีพียู

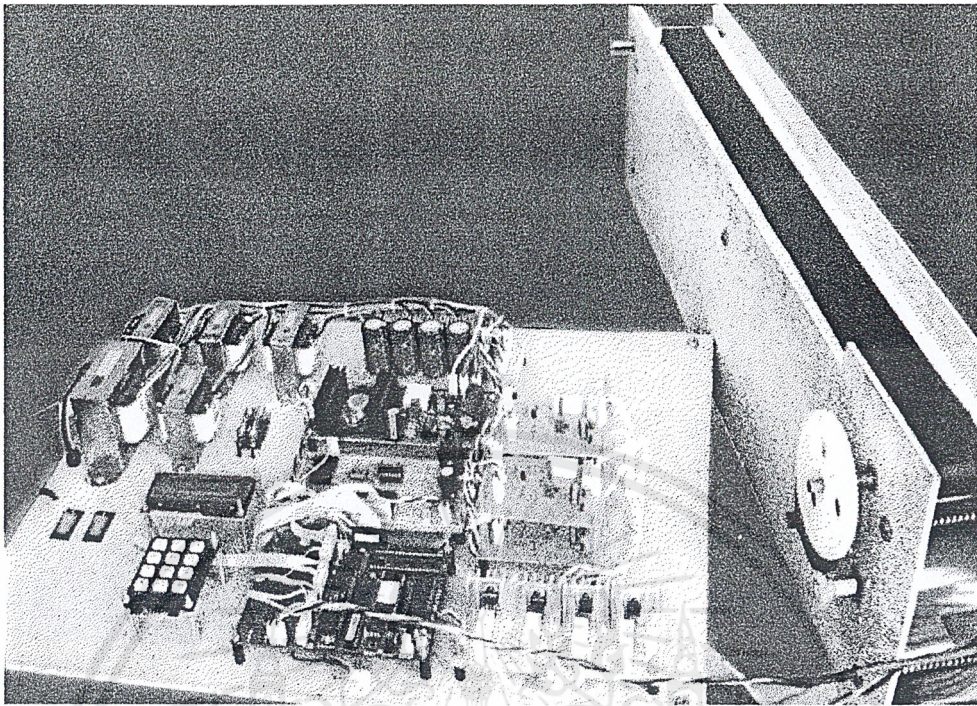
- ควรเปลี่ยนชุดขับเคลื่อนสายพานให้มีประสิทธิภาพการทำงานมากกว่าขึ้น ซึ่งจะส่งผลให้การทำงานดีขึ้น

- ควรเปลี่ยนรูปแบบการจับให้เป็นแบบเซอร์โว เช่น ยูนิโพลาร์,ไบโพลาร์ หรือ ลิมิตยูนิโพลาร์ ซึ่งชุดขับที่ใช้ในโครงงานนี้สามารถรองรับได้ และสามารถควบคุมตำแหน่งหรือควบคุมความเร็วและทอร์กได้ ซึ่งขึ้นอยู่กับ โปรแกรมการทำงาน

- ควรใช้ Hardware ให้ตอบรับกับการทำงานแบบเซอร์โวคือควรใช้ D.C. servo motor ที่โรเตอร์มี Inertia ต่ำๆ เช่น Brushless เป็นต้น ใช้เอ็น โคดเดอร์ที่มีความละเอียดสูงและใช้เฟืองทดที่อัตราทดสูงๆ จะทำให้ควบคุมตำแหน่งได้ละเอียดขึ้น

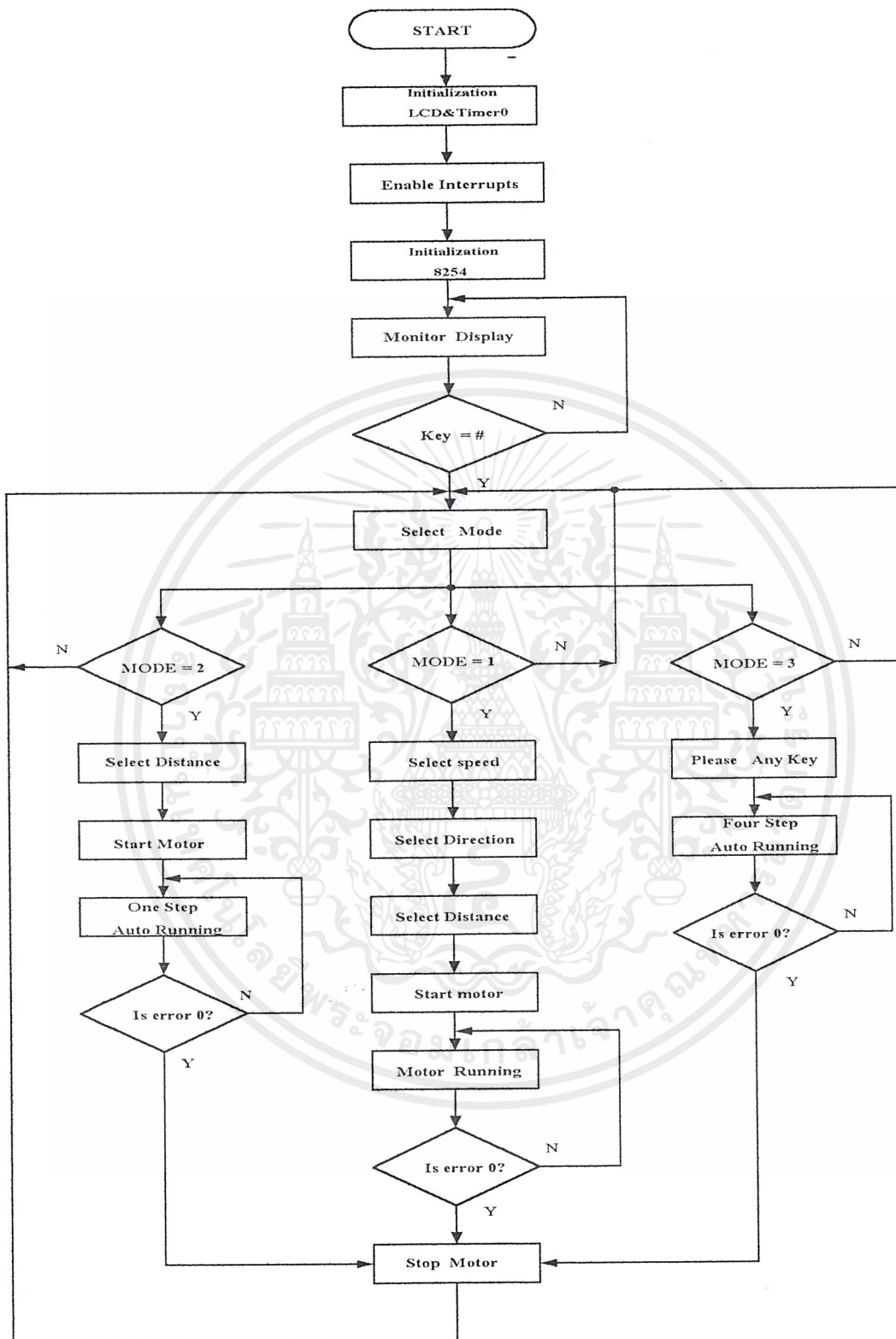


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป Hardware ที่ใช้ในโรงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



โฟวีชาร์ทแสดงการทำงานของโปรแกรมการควบคุมตำแหน่งของดีซีเซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โปรแกรมควบคุมตำแหน่งดีซีเซอร์มอเตอร์

```
;***** Channel Address of 8254*****
CHN0      EQU   0FC00H
CHN1      EQU   0FC01H
CTRL      EQU   0FC03H

;***** Internal ram variable *****

;* Keyboard Segment
last      equ   22h
keybuf    equ   last+1      ;4 byte
pos       equ   last+6

;* Use by Routine in This Program
countl    equ   50h
counth    equ   countl+1    ;distance to count in Th0,T10
direct    equ   countl+2    ;is 'f' or 'r'
speed1    equ   direct+1
speed2    equ   speed1+1
idx_speed equ   speed2+1    ;index to open table speed
flag_dis  equ   70h        ;flag bit at 20h to check interrupt timer0

;* Constant
key_up    equ   rxd
key_down  equ   t1
dly_display equ   2
way       equ   int1
en_motor  equ   int0
startcount equ   tr0
```

```
ORG 0000h
```

```
origin:  sjmp  main
```

```
;* Timer 0 Interrupt
```

```
org  origin+0Bh
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        ljmp  Timer0ISR

main:

;* Initial LCD segment

        clr  cn_motor
        call init_lcd
        call clrscr
        call cursor_on

;* Initial Timer 0 Segment

        mov  tmod,#05h    ;0000 0101b
        call delay
        call scankey
        cjne a,#Enter,main1
        call Diagno
main1:   setb  ca
        setb ct0
        call Init8254
main2:   call  main_menu
        sjmp main2

;*****
;Main Menu Mode1 or Mode2
;Pre:-
;Post: Operate selectioned by user
;*****

main_menu:  mov   b,#dly_display
            call  clrscr
            call  printlcd
            db   "DC SERVO DRIVE.",00
            mov  a,#40h
            call  gotoled
            call  printlcd
            db   " PLEASE #. ",00

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

dly_menu:    call  delay
             djnz  b,dly_menu
             call  scankey
             cjnc  a,#0bh,main_menu

main_menu1:  call  clrscr
             call  printlcd
             db   "SELECT MODE1/2/3",00
             mov  a,#40h
             call  gotoled
             call  printlcd
             db   ":",00
             clr  a
lop1_menu:   call  scankey
             cjnc  a,#0ffh,checkmode
             sjmp  lop1_menu
checkmode:   cjnc  a,#1,gotomode2
             call  printlcd
             db   "1",00
             call  delay
             call  mode1
             sjmp  main_menu1
gotomode2:   cjnc  a,#2,gotomode3
             call  printlcd
             db   "2",00
             call  delay
             call  mode2
             ajmp  main_menu1
gotomode3:   cjnc  a,#3,gotoerror
             call  printlcd
             db   "3",00
             call  delay
             call  mode3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                                ajmp  main_menu1
gotoerror:  call  clrscr
                                call  printlcd
                                db    "PLEASE PRESS '1'",00
                                mov   a,#40h
                                call  gotolcd
                                call  printlcd
                                db    "OR '2'",00
                                call  delay
                                call  delay
                                ajmp  main_menu1

```

```

;*****

```

```

;Mode2 Operation = Automatic Running

```

```

;*****

```

```

mode2:  call  clrscr
                                call  printlcd
                                db    "MODE 2 OPERATION",00
                                mov   b,#dly_display
dly_mode2:  call  delay
                                djnz  b,dly_mode2
                                call  clrscr
                                call  printlcd
                                db    "PRESS ANY KEY ",00
                                mov   a,#40h
                                call  gotolcd
                                call  printlcd
                                db    "TO RUNNING",00

```

```

any_key:  call  scankey
                                cjnc  a,#0ffh,press_key
                                sjmp  any_key

```

```

press_key:  mov   idx_spced,#1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

call read_stab    ;Speed 2006 rpm
clr way          ;forward
mov direct,#'f'
mov r3,#0
mov r2,#0ah      ;10 cm
call Step_run
call Wait_step
mov r3,#0
mov r2,#14h      ;30 cm
call Step_run
call Wait_step
mov r3,#0
mov r2,#0ah      ;40 cm
call Step_run
call Wait_step
mov r3,#0
mov r2,#14h      ;60 cm
call Step_run
call Wait_step
mov r3,#0
mov r2,#41h      ;60 cm
setb way         ;revese
mov direct,#'r'
call Step_run
call clrscr
call printlcd
db "FINISH.",00
mov b,#dly_display

dly_finish2:    call delay
                djnz b,dly_finish2
                ret

Wait_step:      call clrscr

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

call printlcd
db "WAIT...",00
mov b,#4
dly_wait: call delay
          djnz b,dly_wait
          ret

```

```

;*****
;

```

```

;Step running for mode2 Operation

```

```

;Pre: Speed1:Speed2 , R3:R2

```

```

;Post: Running in one Step

```

```

;*****
;

```

```

Step_run: call Arith_count
          mov r3,countl
          mov r4,counth
          call Two_comp
          mov th0,r4
          mov tl0,r3
          call load_speed
          call clrscr
          call printlcd
          db "One Step Running",00
          clr flag_dis
          call start_motor
          jnb flag_dis,$ ; Waiting for finish operation
          call Stop_motor
          ret

```

```

;*****
;

```

```

; **mode2 1 step automatic running

```

```

;*****
;

```

```

mode2:   call clrscr
          call printlcd

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

db      "MODE3 OPERATION",00
mov     b,#dly_display
dly_mode2: call  delay
        djnz  b,dly_mode2
mov     speed1,#0
mov     speed2,#0
mov     idx_speed,#1
call    read_stab
call    load_speed
clr     way
mov     direct,#'f'
call    distance
mov     r4,counth
mov     r3,countl
call    two_comp
mov     tl0,r4
mov     tl0,r3
mov     counth,r4
mov     countl,r3
call    clrscr
call    printlcd
db      "NOW Running....",00
clr     flag_dis
call    start_motor
jnb     flag_dis,$
call    stop_motor
call    delay
setb    way
mov     direct,#'r'
mov     tl0,counth
mov     tl0,countl
clr     flag_dis

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        call    start_motor

        jnb    flag_dis,$

        call    stop_motor

        ret

;*****
;Mode 1 Operation from user selection
;*****

model:    call    clrscr
          call    printlcd
          db     "MODE 1 Operation",00
          mov    b,#dly_display

dly_model: call    delay
          djnz   b,dly_model
          mov    speed1,#0
          mov    speed2,#0
          call   Read_rpm      ;input rpm speed
          call   load_speed    ;load speed to 8254
          call   Direction     ;forward or reverse
          call   Distance      ;distance for run
          mov    r4,counth
          mov    r3,countl
          call   Two_comp
          mov    th0,r4
          mov    tl0,r3        ;* load Distance
          call   clrscr
          call   printlcd
          db     "NOW Running...",00
          clr    flag_dis
          call   start_motor
          jnb    flag_dis,$    ;Phase1 not slow
          call   stop_motor

          ret

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
*****
```

```
;Load Speed for Speed1:Speed2 Variable to 8254 Channel #1
```

```
;Pre: Speed1:Speed2
```

```
;Post: Channel 1 of 8254
```

```
*****
```

```
load_speed:    mov    dptr,#chn1
               mov    a,speed2
               movx   @dptr,a
               mov    a,speed1    ; load Speed
               movx   @dptr,a    ; speed1:speed2
               rct
```

```
*****
```

```
;* Initial 8254 Segment *
```

```
*****
```

```
Init8254:     MOV    A,#37H
               MOV    DPTR,#CTRL
               MOVX   @DPTR,A
               MOV    DPTR,#CHNO
               MOV    A,#33h
               MOVX   @DPTR,A
               MOV    A,#03H
               MOVX   @DPTR,A    ; 20 kHz square wave

               MOV    A,#73H
               MOV    DPTR,#CTRL
               MOVX   @DPTR,A    ; Monostable mode
               RET
```

```
*****
```

```
;* Interrupt Routine Timer 0 *
```

```
*****
```

```
Timer0ISR:    setb   flag_dis
               reti
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
*****
```

```
;Post: Motor Start
```

```
*****
```

```
Start_motor:  setb  startcount  
              setb  cn_motor  
              ret
```

```
*****
```

```
;Post: Motor Stop
```

```
*****
```

```
Stop_motor:  clr  startcount  
            mov  a,direct  
            cjne a,#'f',st_rwd  
            setb way      ;reverse  
            call dlystop  
            clr  way      ;restored way [forward]  
            sjmp end_st  
st_rwd:     cjne a,#'r',end_st  
            clr  way      ;forward  
            call dlystop  
            setb way      ;restored way [reverse]  
end_st:     clr  cn_motor  
            ret
```

```
*****
```

```
;Pre: sdelyh:sdelyl
```

```
*****
```

```
dlystop:    push  acc  
            push  dpl  
            push  dph  
            mov  dptr,#Stop_Tab  
            mov  a,idx_speed  
            call lookupw
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

dlst1:      inc  dptr
            mov  a,dph
            orl  a,dpl
            jnz  dlst1
            pop  dph
            pop  dpl
            pop  acc
            ret

Stop_Tab:   dw  65535-3000
            dw  65535-5000
            dw  65535-7000
            dw  65535-10000

;*****
;input speed from up key & down key
;Pre: -
;Post: Speed value is BCD format in speed1:speed2 variable
;*****

Read_rpm:   push acc
            call clrscr
            call printlcd
            db  "SPEED MOTOR(rpm)",00
            mov  idx_speed,#0
            call wr_speed

get_speed:  call debounce
            jnb  key_up,up_key
            jnb  key_down,down_key
            call scankey
            cjne a,#Enter,get_speed
            sjmp end_speed

up_key:     call debounce
            inc  idx_speed
            mov  a,idx_speed

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                cjne a,#4,disply_up
                mov  idx_speed,#0
disply_up:     call  wr_speed
                sjmp key_speed
down_key:     call  debounce
                dec  idx_speed
                mov  a,idx_speed
                cjne a,#0ffh,disply_dn
                mov  idx_speed,#3
disply_dn:     call  wr_speed
key_speed:    jnb  key_down,key_speed
                jnb  key_up,key_speed
                sjmp get_speed
end_speed:    pop  acc
                ret
Speed_tab:    db  01h,00h
                db  01h,15h
                db  01h,30h
                db  01h,45h
                db  01h,60h
                db  01h,75h
                db  01h,90h
Pluse_tab:    db  01h,24h
                db  01h,28h
                db  01h,34h
                db  01h,38h
                db  01h,44h
                db  01h,49h
                db  01h,53h

speed_prompt EQU  40h
wr_speed:     push  acc
                mov  a,#speed_prompt

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

call gotoled
mov  dptr,#speed_tab
mov  a,idx_speed
call lookupw
mov  a,dph
call wr_2bcd
mov  a,dpl
call wr_2bcd
call read_stab
pop  acc
ret

```

```

;*****

```

```

;Pre: idx_speed variable

```

```

;Post: speed1:speed2

```

```

;*****

```

```

read_stab:  mov  dptr,#pluse_tab
            ;
            mov  a,idx_speed
            call lookupw
            mov  speed1,dph
            mov  speed2,dpl
            ret

```

```

;*****

```

```

;input direction of motor & Active Immediately

```

```

;Pre: -

```

```

;Post: Motor is run in user direction defined

```

```

; (Set/Clear P1.1)

```

```

; "Direct" Variable = 'f' or 'r'

```

```

;*****

```

```

dir_key_prompt      EQU  40h      ;Positon on direction input

```

```

Direction:          push  acc

```

```

begin_dir:          call  clrscr

```

```

                    call  printled

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        db "DIRECTION MOTOR",00
default_dir: mov direct,#'f'
            mov a,#dir_key_prompt
            call gotoled
            call printlcd
            db "FORWARD",00
get_dir:   call debounce
            jnb key_up,toggle_key
            jnb key_down,toggle_key
            call scankey
            cjne a,#Enter,get_dir
            mov r0,direct
            cjne r0,#'f',reverse_dir
            sjmp forward_dir
toggle_key: call debounce
            mov r0,direct
            cjne r0,#'r',toggle_key1
            mov direct,#'f'
            mov a,#dir_key_prompt
            call gotoled
            call printlcd
            db "FORWARD",00 ;Forwrđ Prompt
            sjmp key_dir
toggle_key1: mov direct,#'r'
            mov a,#dir_key_prompt
            call gotoled
            call printlcd
            db "REVERSE",00 ;Reverse Prompt
key_dir:   jnb key_up,key_dir
            jnb key_down,key_dir
            sjmp get_dir
forward_dir: clr way ;forward

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                sjmp  end_dir

reverse_dir:   setb  way  ;reverse

end_dir:      pop   acc

                ret

;*****
;input distance from user
;Pre: -
;Post: Counth:Countl Variable
;*****

Dis_prompt    EQU   40h

Distance:     push  acc

begin_dis:    call  clrscr
              call  printlcd
              db   "DISTANCE1-65[cm]",00
              mov  a,#Dis_prompt
              call  gotolcd
              mov  r2,#0
              mov  r3,#0
              call  Readln
              cjne r2,#0,comp0
              call  clrscr
              call  printlcd
              db   "NO DISTANCE !",00
              mov  b,#dly_display

dly_dis2:     call  delay
              djnz b,dly_dis2
              jmp  begin_dis

comp0:        cjne r2,#66,comp1  ;compare with line distance (65cm)

comp1:        jc   comp2        ;>=66
              call  clrscr
              call  printlcd
              db   "OVER RANGE !",00

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        mov     b,//dly_display
dly_dis:    call  delay
           djnz  b,dly_dis
           jmp   begin_dis
comp2:     call  Arith_count ;0-65
           pop   acc
           ret

;*****
;Arithmetics for counting and load to counter register
;Pre: R3:R2
;Post: counth:countl
;*****
Arith_count:  mov   r5,//0dh      ;input R3:R2
             mov   r4,//02ah    ;R5:R4 D26h = 3366 decimal
             call  MULTIPLY     ;3366 * input value o/p R7:R6:R5:R4
             mov   a,r6
             xch  a,r5
             xch  a,r4
             xch  a,r3      ;Prepare for Divide16 Routine
             mov  r2,#0
             mov  r1,#10
             call Divide24 ;Output R4:R3 = 3366*input value/10
             mov  counth,r4
             mov  countl,r3
             ret

;*****
;Input Pluse for distance running
;Pre:-
;Post:counth:countl
;*****
Pluse_run:   call  clrscr
             call  printlcd

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

db "Pluse to Run",00
mov a,#40h
call gotolcd
call Readln6
mov counth,r3
mov countl,r2
ret

;*****
;Percent for slow and Stop
;Pre: r4:r3 = counth:countl , percent
;Post: runh:runl , breakh:breakl
;*****
Arith_per:  mov  a,r4
            xch  a,r3
            xch  a,r2
            mov  r5,#0
            mov  r4,percent
            call MULTIPLY
            mov  a,r6
            xch  a,r5
            xch  a,r4
            xch  a,r3 ;Prepare for Divide16 Routine
            mov  r2,#0
            mov  r1,#100
            call Divide24 ;Output R4:R3 = percent/100*counth:countl
            mov  breakh,r4
            mov  breakl,r3
            clr  c
            mov  a,countl
            subb a,breakl
            mov  runl,a
            mov  a,counth

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

subb a,breakh
mov runh,a ;runh:runl = counth:countl-breakh;breakl
ret

```

```

;*****

```

```

;Two complement 16 bit

```

```

;Pre: R4:R3

```

```

;Post: R4:R3

```

```

;*****

```

```

Two_comp:  push  acc
           mov   a,r3
           cpl  a
           add  a,#1
           mov  r3,a
           mov  a,r4
           cpl  a
           addc a,#0
           mov  r4,a ;Two Complement R4:R3
           pop  acc
           ret

```

```

;*****

```

```

;* Keyboard Segment *

```

```

;*****

```

```

Enter      equ  0bh      ;enter key

```

```

Esc        equ  01bh     ;csc key

```

```

;***** bit addressable *****

```

```

keyprs     equ  8        ;int RAM addr.20h bit

```

```

;*****

```

```

;* Keyboard Section *

```

```

;*****

```

```

row0      equ  acc.0

```

```

row1      equ  acc.1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

row2          equ  acc.2
row3          equ  acc.3
col0          equ  p1.4
col1          equ  p1.5
col2          equ  p1.6
;*****
;* Scankey board 4 row * 3 column *
;* Output : ACC if equ 0ffh nokey*
;*****
scankey:      push  dpl
              push  dph
              mov   p1,#0ffh
actvcol0:     clr   col0          ;active column 0
              mov   b,#0
              setb  col1
              call  what_row
              cjne  a,#0ffh,keycode
actvcol1:     setb  col0
              clr   col1          ;active column 1
              mov   b,#1
              call  what_row
              cjne  a,#0ffh,keycode
actvcol2:     setb  col1
              clr   col2          ;active column 2
              mov   b,#2
              call  what_row
              cjne  a,#0ffh,keycode
              clr   keyprs        ;key already released
              sjmp  out_snk       ;no key pressed
keycode:      call  debounce
              jnb  keyprs,kcod1
              mov   a,#0ffh

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                sjmp out_snk
kcod1:         setb keyprs
out_snk:      pop  dph
                pop  dpl
                mov  p1,#0ffh
                ret
KeyTab:       db   01h,02h,03h  ;1 2 3
                db   04h,05h,06h  ;4 5 6
                db   07h,08h,09h  ;7 8 9
                db   1Bh,00h,0Bh  ;c 0 B

;*****
;* Search row pressed & get offset *
;* Input  : B = Column      *
;* Output : ACC= Key code   *
;*****
what_row:     mov  a,p1
                orl  a,#11110000b
                xrl  a,#11111111b
                jnz  sk3      ;keypressed
                mov  a,#0ffh  ;key wasn't pressed
                sjmp out_row
sk3:          call debounce
                jnb  row0,whrw1
                mov  dptr,#KeyTab+0 ;row 0
                sjmp koffset
whrw1:       jnb  row1,whrw2
                mov  dptr,#KeyTab+4 ;row 1
                sjmp koffset
whrw2:       jnb  row2,whrw3
                mov  dptr,#KeyTab+8 ;row 2
                sjmp koffset

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

whrw3:    mov  dptr,#KeyTab+12 ;row 3
koffset:  mov  a,b
          movc a,@a+dptr
out_row:  ret

debounce: push  acc
          push  dpl
          push  dph
          mov  dptr,#65535-3000
dbnc1:   inc  dptr
          mov  a,dph
          orl  a,dpl
          jnz  dbnc1
          pop  dph
          pop  dpl
          pop  acc
          ret

;*****
;*   Read key   *
;* Read 4 character from k/b *
;* Output : R4:R3:R2 = hex *
;*   Keybuf1:2=BCD *
;*****
;

ReadLn:   call  waitbf
          clr  acc.7
          mov  pos,a
r18:     mov  keybuf+2,#0
          mov  keybuf+1,#0
          mov  keybuf+0,#0
          call  gotolcd
r10:     call  scankey

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        cjne a,#0ffh,r11
        sjmp r10
r11:    cjne a,#Enter,r12
        sjmp r16
r12:    cjne a,#Esc,r13
        mov a,keybuf+2
        orl a,keybuf+1
        jnz r111
        mov a,#Esc
        sjmp r17 ;no key but exit
r111:   mov a,pos
        call gotoLcd
        call printLcd
        db " ",00
        mov a,pos
        sjmp r18
r13:    cjne a,#0ah,r14 ;get char
r14:    jc r15
        sjmp r10
r15:    mov r2,a
        mov a,keybuf+0
        xrl a,#4 ;limit 4 digit
        jz r10
        inc keybuf+0
        mov a,r2
        orl a,#30h
        call writeLcd
        mov a,r2
        swap a
        mov r2,a
        mov r3,#4 ;shift 4 bit
r19:    mov r0,#keybuf+2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov r1,#2 ;2 byte =4 digits
mov a,r2
clr c
rlc a
mov r2,a
r110: mov a,@r0
rlc a
mov @r0,a
dec r0
djnz r1,r110
djnz r3,r19
sjmp r10
r16: mov r4,#0
mov r3,keybuf+1
mov r2,keybuf+2
call bcd_hexs ;return in R4:R3:R2
r17: rct

```

```

;*****

```

```

;* Lcd management section *

```

```

;*****

```

```

COMMAND EQU 0FA00H ; Write instruction
READBUSY EQU 0FA01H ; Read BF(Busy Flag) and address
WRITEDATA EQU 0FA02H ; Write character
READDATA EQU 0FA03H ; Read Data from DD ram

```

```

;*****

```

```

;* Write ASCII to lcd 16X1 *

```

```

;* Input : ACC = ASCII *

```

```

;*****

```

```

WRITELCD: PUSH DPL
          PUSH DPH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV DPTR,#WRITEDATA
MOVX @DPTR,A
CALL WAITBF ;Wait LCD module ready
POP DPH
POP DPL
RET

```

; \*\*\* Wait for ready

```

WAITBF:  PUSH DPL
          PUSH DPH
          PUSH B
          MOV DPTR,#READBUSY
          MOV B,#0
RDY1:    MOVX A,@DPTR
          JNB ACC.7,RDY2 ;Busy Flag
          DJNZ B,RDY1
RDY2:    POP B
          POP DPH
          POP DPL
          clr acc.7
          RET

```

;\*\*\*\*\*

;\* Initial LCD \*

;\*\*\*\*\*

;

```

INIT_LCD:  PUSH DPL
            PUSH DPH
            MOV DPTR,#COMMAND
            MOV A,#38H ;8bit, 2 line, 5x7 dot
            MOVX @DPTR,A
            LCALL WAITBF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV A,#0CH ;0FH
MOVX @DPTR,A
LCALL WAITBF
MOV A,#6 ;increment cursor
MOVX @DPTR,A
LCALL WAITBF
MOV A,#1 ;clear and home
MOVX @DPTR,A
LCALL WAITBF
POP DPH
POP DPL
RET

```

```

;*****
;* Set cursor to ddram location *
;* Input : ACC = ddram address *
;* Output : LCD *
;*****

```

```

gotolcd:   push  dpl
           push  dph
           setb  acc.7
           mov   dptr,#command
           movx  @dptr,a
           call  waitbf
           pop   dph
           pop   dpl
           ret

```

```

;*****
;* Clear screen & cursor to home*
;*****

```

```

clrscr:   push  acc

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov a,#1
mov dptr,#command
movx @dptr,a
call waitbf
pop acc
ret

```

```

;*****
;

```

```

;* Write BCD to LCD *
```

```

;* Input : ACC= BCD *
```

```

;*****
;

```

```

Wr_2bcd:   call hex_2asc
          call writelcd
          mov a,b
          call writelcd
          ret

```

```

;*****
;

```

```

;* Set cursor on and blink *
```

```

;*****
;

```

```

Cursor_On: mov a,#0fh
          mov dptr,#Command
          movx @dptr,a
          ret

```

```

;*****
;

```

```

;* Write character string followed by the*
```

```

;* hcall printlcdh instruction to lcd. *
```

```

;* Input : - *
```

```

;* Register : Destroy DPTR,ACC *
```

```

;*****
;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;
;Note that the end of string table must be 00
;
PRINTLCD:   POP  DPH    ; Get Pointer(stack)
            POP  DPL

STSD0:      CLR  A
            MOVC A,@A+DPTR
            JZ   STSD2

STSD1:      call writelcd
            call waitbf

stdsd3:     INC  DPTR
            SJMP STSD0

STSD2:      INC  DPTR
            PUSH DPL
            PUSH DPH
            RET

;*****
;* Convert HEX to ASCII *
;* Input   : A *
;* Output  : A (high-byte) *
;*         : B (low-byte) *
;*****

HEX_2ASC:   PUSH ACC
            call HEX_TO_ASC
            MOV  B,A    ; B = Lo byte
            POP  ACC
            SWAP A
            call HEX_TO_ASC
            RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*****
;* Convert hex (0-F) to ASCII *
;* Input : ACC (HEX code) *
;* Output : ACC (ASCII code) *
;*****

```

```

HEX_TO_ASC:    PUSH PSW
               CLR CY
               ANL A,#00001111B
               CJNE A,#09H,HTOA1
               SETB CY

```

```

HTOA1:        JNB CY,HTOA2    ;> 0AH
               ORL A,#30H     ;< 0AH
               SJMP END_HTOA

```

```

HTOA2:        SUBB A,#09H     ;> 0AH
               ORL A,#40H

```

```

END_HTOA:     POP PSW
               RET

```

```

;*****
;* LOOK UP TABLE (DATA WORD) *
;* Input : ACC = The order of table*
;* DPTR = Address of table *
;* Output : DPTR = Value in table *
;*****

```

```

LOOKUPW:     PUSH PSW
             PUSH B
             MOV B,A          ;Save offset
             MOV A,#2
             MUL AB
             ADD A,DPL
             MOV DPL,A
             MOV A,DPH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ADDC A,B
MOV DPH,A
clr a
MOVC A,@A+DPTR
PUSH ACC ;Save hi byte
INC DPTR
clr a
MOVC A,@A+DPTR
MOV DPL,A
POP DPH ;Retreive hi byte
POP B
POP PSW
RET

```

```

;*****

```

```

;* General delay *
```

```

;*****

```

```

delay:  push dpl
        push dph
        push acc
        mov  dptr,#0 ;#65535-2000

```

```

dely1:  inc  dptr
        mov  a,dph
        orl  a,dpl
        jnz  dely1
        pop  acc
        pop  dph
        pop  dpl
        ret

```

```

;*****

```

```

;* Sub routine *
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;* Convert decimal to hex *
;* INPUT : BCD_BUF = BCD *
;* OUTPUT : BIN_BUF = HEX *
;*****
DEC_2HEX: LCALL CLR_BINBUF
          MOV R7,#24
DHX0:    MOV R5,#5
          MOV DPTR,#BCD_BUF
DHX1:    MOVX A,@DPTR
          JNZ DHX2 ;BCD not zero
          INC DPTR
          DJNZ R5,DHX1
          SJMP OUT_DHX ;All zero
          ;** Devide BCD by two
DHX2:    LCALL DV_BCD
          DEC R7
          SJMP DHX0
OUT_DHX: CLR CY
          MOV DPTR,#BIN_BUF
          MOVX A,@DPTR
          RRC A
          MOVX @DPTR,A
          INC DPTR
          MOVX A,@DPTR
          RRC A
          MOVX @DPTR,A
          INC DPTR
          MOVX A,@DPTR
          RRC A
          MOVX @DPTR,A
          DJNZ R7,OUT_DHX
          RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*****
;* Devide BCD keep remainder in BINBUF *
;*****
;
DV_BCD:   MOV  DPTR,#BCD_BUF+4
          MOV  R6,#5
          CLR  CY
DVB0:    MOVX A,@DPTR
          RRC  A
          MOVX @DPTR,A
          MOV  A,DPL
          JNZ  DVB1
          DEC  DPH
DVB1:    DEC  DPL
          DJNZ R6,DVB0
          ;** Carry is an output **
          MOV  DPTR,#BIN_BUF
          MOVX A,@DPTR
          RRC  A      ;Rotate right with carry
          MOVX @DPTR,A
          INC  DPTR
          MOVX A,@DPTR
          RRC  A
          MOVX @DPTR,A
          INC  DPTR
          MOVX A,@DPTR
          RRC  A
          MOVX @DPTR,A
          ;Decimal adjust
          MOV  R6,#5
          MOV  DPTR,#BCD_BUF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DBV5:    MOVX A,@DPTR
          JNB ACC.3,DBV2
          CLR CY
          MOV B,#3
          SUBB A,B
          MOVX @DPTR,A
DBV2:    JNB ACC.7,DBV4
          CLR CY
          MOV B,#30H
DBV3:    SUBB A,B
          MOVX @DPTR,A
DBV4:    INC DPTR
          DJNZ R6,DBV5
          RET

CLR_BINBUF: CLR A
           MOV DPTR,#BIN_BUF
           MOVX @DPTR,A
           INC DPTR
           MOVX @DPTR,A
           INC DPTR
           MOVX @DPTR,A
           INC DPTR
           MOVX @DPTR,A
           RET

CLR_BCDBUF: PUSH B
            CLR A
            MOV DPTR,#BCD_BUF
            MOV B,#5
CBCD1:    MOVX @DPTR,A
            INC DPTR
            DJNZ B,CBCD1
            POP B

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RET
;*****
;* System call,convert BCD to HEX *
;* INPUT   : R4:R3:R2 = BCD   *
;* OUTPUT  : R4:R3:R2 = HEX  *
;* Registered : ACC,DPTR      *
;*****
;

```

```
BCD_HEX5: LCALL CLR_BCDBUF
```

```

    LCALL CLR_BINBUF
    MOV DPTR,#BCD_BUF
    MOV A,R2
    MOVX @DPTR,A
    INC DPTR
    MOV A,R3
    MOVX @DPTR,A
    INC DPTR
    MOV A,R4
    MOVX @DPTR,A
    LCALL DEC_2HEX
    MOV DPTR,#BIN_BUF
    MOVX A,@DPTR
    MOV R4,A
    INC DPTR
    MOVX A,@DPTR
    MOV R3,A
    INC DPTR
    MOVX A,@DPTR
    MOV R2,A
    RET

```

```
;*****
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;* Multiply 16 bit *
;* Input R5:R4 *
;* multiplier = R3:R2 *
;* Output R7:R6:R5:R4 *
;*****
MULTIPLY:  mov r6,#0
           mov r7,#0
           mov dptr,#result
           mov r1,#4
           clr a
mul0:     movx @dptr,a
           inc dptr
           djnz r1,mul0
           mov r1,#16 ;16 bit multiply
mul1:     clr c
           mov a,r3
           rrc a
           mov r3,a
           mov a,r2
           rrc a
           mov r2,a
           jnc mul2
           call addressult
mul2:     call shiftleft
           djnz r1,mul1
           mov r1,#4
           mov dptr,#result
           movx a,@dptr
           mov r4,a
           inc dptr
           movx a,@dptr
           mov r5,a

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
inc dptr
movx a,@dptr
mov r6,a
inc dptr
movx a,@dptr
mov r7,a
End_Mul: ret
```

```
shiftright: clr c
mov a,r4
rlc a
mov r4,a
mov a,r5
rlc a
mov r5,a
mov a,r6
rlc a
mov r6,a
mov a,r7
rlc a
mov r7,a
ret
```

```
address: mov dptr,#result
movx a,@dptr
add a,r4
movx @dptr,a
inc dptr
movx a,@dptr
adde a,r5
movx @dptr,a
inc dptr
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

movx a,@dptr
addc a,r6
movx @dptr,a
inc dptr
movx a,@dptr
addc a,r7
movx @dptr,a
ret

```

```

;*****
;*   Devide 24 bit   *
;* Input : R5:R4:R3   *
;*   R2:R1 (Divider) *
;* Output: R5:R4:R3 (Result)*
;*   R7:R6 (Remainder)*
;*****

```

```

Divide24:  mov  r7,#0
           mov  r6,#0
           mov  b,#24      ;24 bit
           clr  c

div161:   call rotowork
           clr  c           ;check < or >
           mov  a,r6       ;(R7:R6 - R2:R1)
           push acc
           subb a,r1
           mov  r6,a
           mov  a,r7
           push acc
           subb a,r2
           mov  r7,a
           jnc  div162     ;R7:R6 >= R2:R1
           pop  acc       ;R7:R6 < R2:R1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov r7,a
pop acc
mov r6,a
setb c
sjmp div163
div162: pop acc ;balance stack
pop acc
div163: cpl c
djnz b,div161
mov a,r3 ;get last result bit
rlc a
mov r3,a
mov a,r4
rlc a
mov r4,a
mov a,r5
rlc a
mov r5,a
ret
otowork: mov a,r3
rlc a
mov r3,a
mov a,r4
rlc a
mov r4,a
mov a,r5
rlc a
mov r5,a
mov a,r6
rlc a
mov r6,a
mov a,r7

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
rlc a
mov r7,a
rct
```

```
;***** RAM Working arear *****
```

```
;*****
```

```
org 8000h
```

```
BIN_BUF: DS 3 ;Binary buffer (convert to BCD)
```

```
BCD_BUF: DS 5 ;BCD buffer(Convert to hex)
```

```
RESULT: DS 4
```

```
end
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

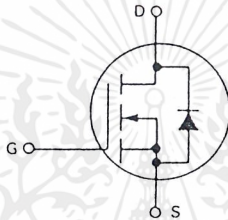
MOTOROLA  
SEMICONDUCTOR  
TECHNICAL DATA

IRF610

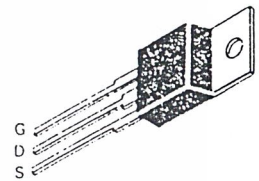
N-CHANNEL ENHANCEMENT-MODE SILICON GATE  
TMOS POWER FIELD EFFECT TRANSISTOR

This TMOS Power FET is designed for low voltage, high speed power switching applications such as switching regulators, converters, solenoid and relay drivers.

- Silicon Gate for Fast Switching Speeds
- Rugged — SOA is Power Dissipation Limited
- Source-to-Drain Diode Characterized for Use With Inductive Loads



Part Number	V <sub>DSS</sub>	R <sub>DS(on)</sub>	I <sub>D</sub>
IRF610	200 V	1.5 Ω	2.5 A



CASE 221A-06  
(TO-220AB)

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Drain-Source Voltage	V <sub>DSS</sub>	200	Vdc
Drain-Gate Voltage (R <sub>GS</sub> = 1.0 MΩ)	V <sub>DGR</sub>	200	Vdc
Gate-Source Voltage	V <sub>GS</sub>	± 20	Vdc
Continuous Drain Current T <sub>C</sub> = 25°C	I <sub>D</sub>	2.5	Adc
Continuous Drain Current T <sub>C</sub> = 100°C	I <sub>D</sub>	1.5	Adc
Drain Current — Pulsed	I <sub>DM</sub>	10	Adc
Total Power Dissipation @ T <sub>C</sub> = 25°C Derate above 25°C	P <sub>D</sub>	20 0.16	Watts W/°C
Operating and Storage Temperature Range	T <sub>J</sub> , T <sub>stg</sub>	- 55 to 150	°C

THERMAL CHARACTERISTICS

Thermal Resistance Junction to Case	R <sub>θJC</sub>	6.4	°C/W
Junction to Ambient	R <sub>θJA</sub>	62.5	
Maximum Lead Temp. for Soldering Purposes, 1.8" from case for 5 seconds	T <sub>L</sub>	260	°C

See the MTP5N20 Designers' Data Sheet for a complete set of design curves for this product.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# PC816 Series

High Collector-emitter Voltage, High Density Mounting Type Photocoupler

## ■ Features

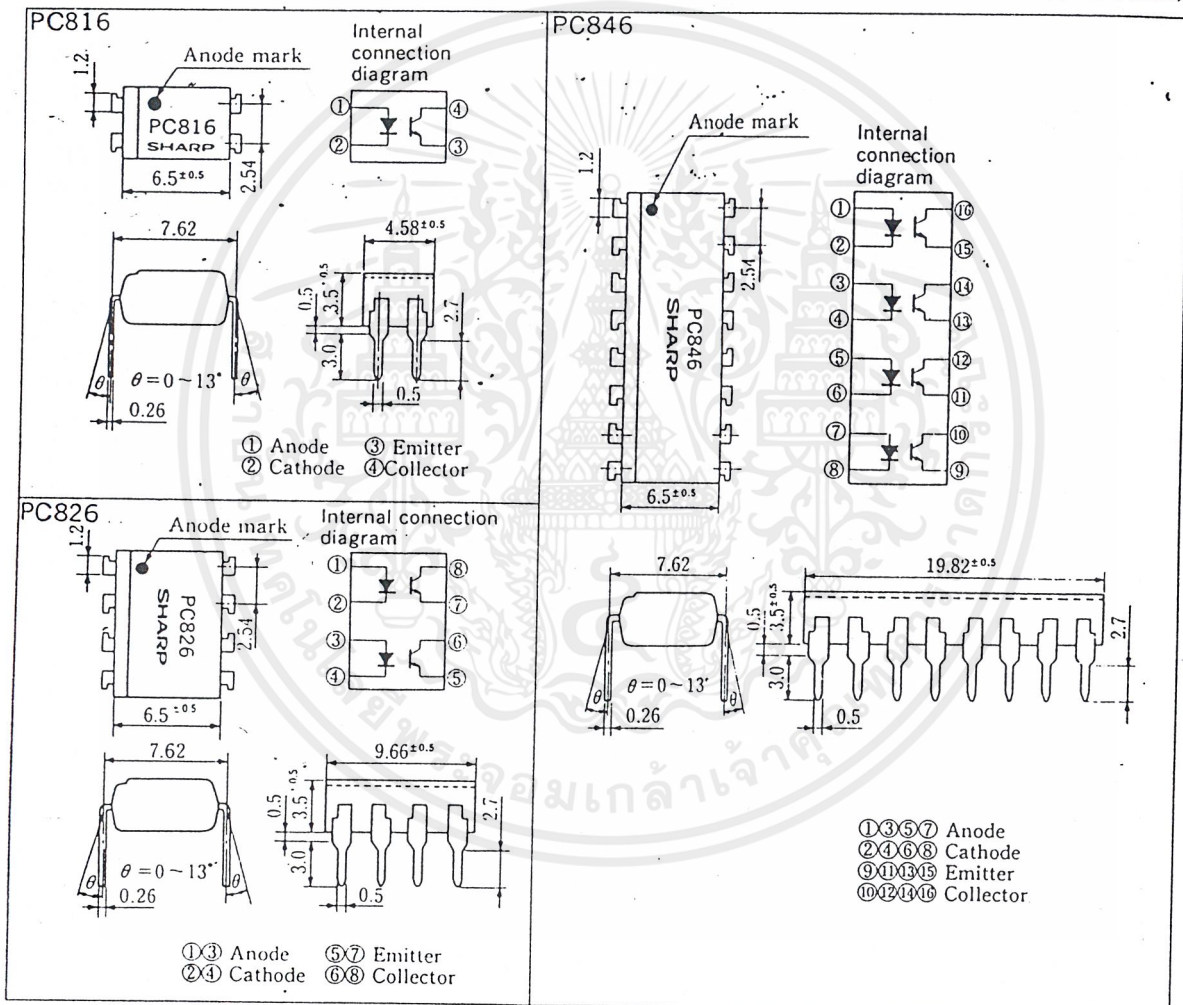
1. High collector-emitter voltage ( $V_{CE0}$ : 70V)
2. Compact dual-in-line package  
PC816: 1-channel type  
PC826: 2-channel type  
PC846: 4-channel type
3. High isolation voltage between input and output ( $V_{iso}$ : 5,000Vrms)
4. Current transfer ratio  
CTR: MIN. 50% at  $I_F=5mA$ ,  $V_{CE}=5V$
5. UL recognized, file No. E64380

## ■ Applications

1. Programmable controllers, Computers
2. System appliances, measuring instruments
3. Signal transmission between circuits of different potentials and impedances

## ■ Outline Dimensions

(Unit : mm)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# IRF610

## ELECTRICAL CHARACTERISTICS (T<sub>C</sub> = 25°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
<b>OFF CHARACTERISTICS</b>					
Drain-Source Breakdown Voltage (V <sub>GS</sub> = 0, I <sub>D</sub> = 250 μA)	V <sub>(BR)DSS</sub>	200	—	—	Vdc
Zero Gate Voltage Drain Current (V <sub>GS</sub> = 0 V, V <sub>DS</sub> = Rated V <sub>DSS</sub> ) (V <sub>GS</sub> = 0 V, V <sub>DS</sub> = 0.8 Rated V <sub>DSS</sub> , T <sub>C</sub> = 125°C)	I <sub>DSS</sub>	—	—	0.25 1.0	mAdc
Forward Gate-Body Leakage Current (V <sub>GS</sub> = 20 V, V <sub>DS</sub> = 0)	I <sub>GSSF</sub>	—	—	100	nAdc
Reverse Gate-Body Leakage Current (V <sub>GS</sub> = -20 V, V <sub>DS</sub> = 0)	I <sub>GSSR</sub>	—	—	-100	nAdc

### ON CHARACTERISTICS\*

Gate Threshold Voltage (V <sub>DS</sub> = V <sub>GS</sub> , I <sub>D</sub> = 250 μA)	V <sub>GS(th)</sub>	2.0	—	4.0	Vdc
On-State Drain Current (V <sub>DS</sub> = 25 Vdc, V <sub>GS</sub> = 10 V)	I <sub>D(on)</sub>	2.5	—	—	Adc
Static Drain-Source On-Resistance (V <sub>GS</sub> = 10 V, I <sub>D</sub> = 1.25 A)	R <sub>DS(on)</sub>	—	—	1.5	Ohms
Forward Transconductance (V <sub>DS</sub> = 15 V, I <sub>D</sub> = 1.25 A)	g <sub>FS</sub>	0.8	—	—	mhos

### DYNAMIC CHARACTERISTICS

Input Capacitance	(V <sub>DS</sub> = 25 V, V <sub>GS</sub> = 0, f = 1.0 MHz)	C <sub>iss</sub>	—	—	150	pF
Output Capacitance		C <sub>oss</sub>	—	—	80	
Reverse Transfer Capacitance		C <sub>rss</sub>	—	—	25	

### SWITCHING CHARACTERISTICS\*

Turn-On Delay Time	(V <sub>DD</sub> = 0.5 V <sub>DSS</sub> , I <sub>D</sub> = 1.25 A Z <sub>o</sub> = 50 Ω)	t <sub>d(on)</sub>	—	—	15	ns
Rise Time		t <sub>r</sub>	—	—	25	
Turn-Off Delay Time		t <sub>d(off)</sub>	—	—	15	
Fall Time		t <sub>f</sub>	—	—	15	

### SOURCE DRAIN DIODE CHARACTERISTICS\*

Characteristic	Symbol	Typ	Unit
Forward On-Voltage	V <sub>SD</sub>	1.8	Vdc
Forward Turn-On Time	t <sub>on</sub>	Limited by stray inductance	
Reverse Recovery Time	t <sub>rr</sub>	290	ns

### INTERNAL PACKAGE INDUCTANCE (TO-220)

Characteristic	Symbol	Min	Typ	Max	Unit
Internal Drain Inductance (Measured from the contact screw on tab to center of die) (Measured from the drain lead 0.25" from package to center of die)	L <sub>d</sub>	—	3.5 4.5	—	nH
Internal Source Inductance (Measured from the source lead 0.25" from package to source bond pad.)	L <sub>s</sub>	—	7.5	—	

\*Pulse Test: Pulse Width < 300 μs, Duty Cycle < 2.0%.

FIGURE 1 — SWITCHING TEST CIRCUIT

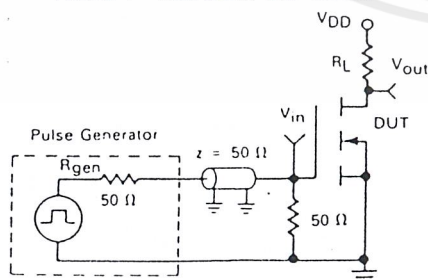
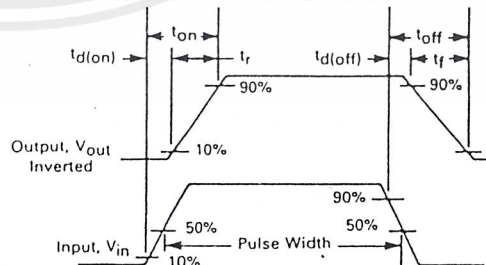


FIGURE 2 — SWITCHING WAVEFORMS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Absolute Maximum Ratings

(Ta = 25°C)

Parameter	Symbol	Rating	Unit
Forward current	$I_F$	50	mA
*1 Peak forward current	$I_{FM}$	1	A
Reverse voltage	$V_R$	6	V
Power dissipation	$P$	70	mW
Collector-emitter voltage	$V_{CEO}$	70	V
Emitter-collector voltage	$V_{CEO}$	6	V
Collector current	$I_C$	50	mA
Collector power dissipation	$P_C$	150	mW
Total power dissipation	$P_{tot}$	200	mW
*2 Isolation voltage	$V_{iso}$	5,000	Vrms
Operating temperature	$T_{opr}$	-30 to +100	°C
Storage temperature	$T_{stg}$	-55 to +125	°C
*3 Soldering temperature	$T_{sol}$	260	°C

\*1 Pulse width  $\leq 100\mu s$ , Duty ratio = 0.001

\*2 RH = 40 to 60%, AC for 1 minute

\*3 For 10 seconds

### Electro-optical Characteristics

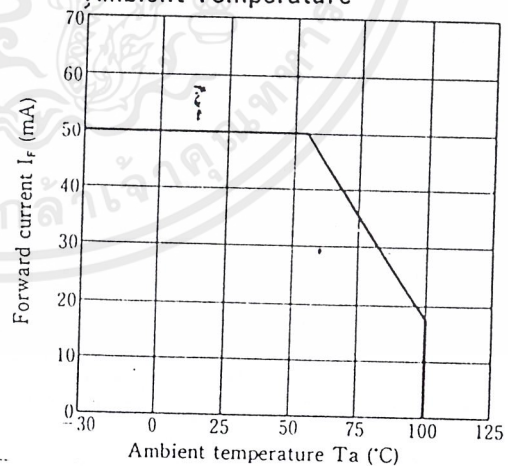
(Ta = 25°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Forward voltage	$V_F$	$I_F = 20\text{mA}$	—	1.2	1.4	V
Peak forward voltage	$V_{FM}$	$I_{FM} = 0.5\text{A}$	—	—	3.0	V
Reverse current	$I_R$	$V_R = 4\text{V}$	—	—	10	$\mu\text{A}$
Terminal capacitance	$C_t$	$V = 0, f = 1\text{kHz}$	—	30	250	pF
Collector dark current	$I_{CEO}$	$V_{CE} = 20\text{V}, I_F = 0$	—	—	$10^{-7}$	A
*Current transfer ratio	CTR	$I_F = 5\text{mA}, V_{CE} = 5\text{V}$	50	—	600	%
Collector emitter saturation voltage	$V_{CE(sat)}$	$I_F = 20\text{mA}, I_C = 1\text{mA}$	—	0.1	0.2	V
Isolation resistance	$R_{iso}$	DC500V, RH = 40 to 60%	$5 \times 10^{10}$	$1 \times 10^{11}$	—	$\Omega$
Floating capacitance	$C_f$	$V = 0, f = 1\text{MHz}$	—	0.6	1.0	pF
Cut-off frequency	$f_c$	$V_{CE} = 5\text{V}, I_C = 2\text{mA}, R_L = 100\Omega$	—	80	—	kHz
Response time (Rise)	$t_r$	$V_{CE} = 2\text{V}, I_C = 2\text{mA}$	—	4	18	$\mu\text{s}$
Response time (Fall)	$t_f$	$R_L = 100\Omega$	—	3	18	$\mu\text{s}$

\*4 Classification table of current transfer ratio is shown below. (11 models)

Model No.	CTR (%)
PC816A	80 to 160
PC816B	130 to 260
PC816C	200 to 400
PC816D	300 to 600
PC816AB	80 to 260
PC816BC	130 to 400
PC816CD	200 to 600
PC816AC	80 to 400
PC816BD	130 to 600
PC816AD	80 to 600
PC816	50 to 600

Fig. 1 Forward Current vs. Ambient Temperature



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# IRF610, IRF611, IRF612, IRF613

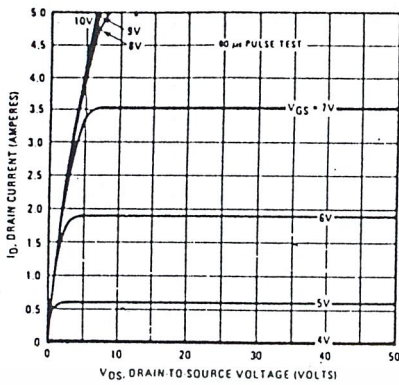


Fig. 1 - Typical Output Characteristics

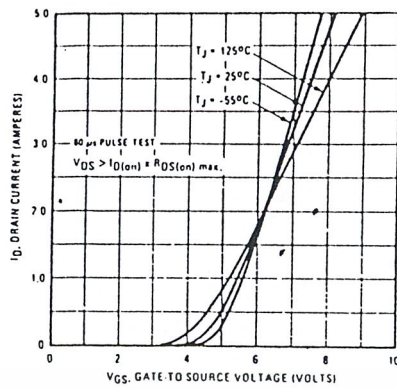


Fig. 2 - Typical Transfer Characteristics

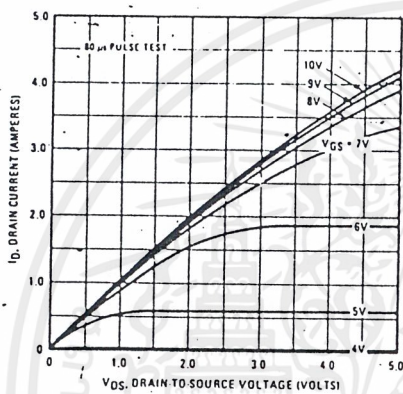


Fig. 3 - Typical Saturation Characteristics

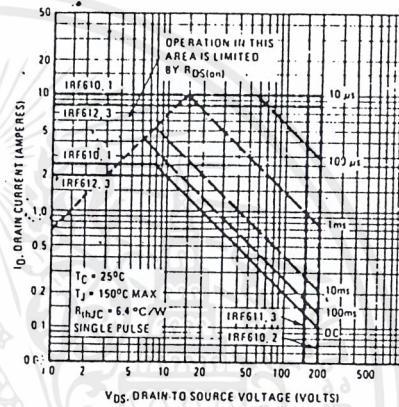


Fig. 4 - Maximum Safe Operating Area

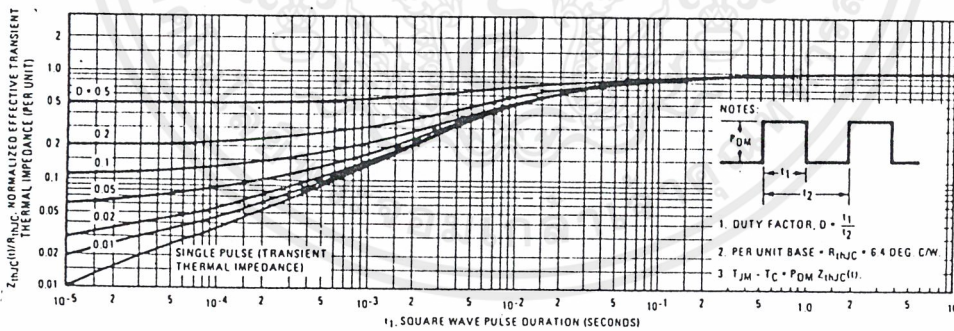


Fig. 5 - Maximum Effective Transient Thermal Impedance, Junction-to-Case Vs. Pulse Duration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# IRF610, IRF611, IRF612, IRF613

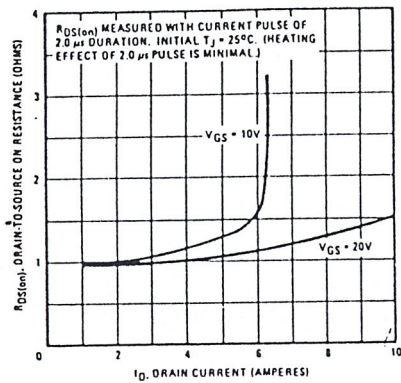


Fig. 12 — Typical On-Resistance Vs. Drain Current

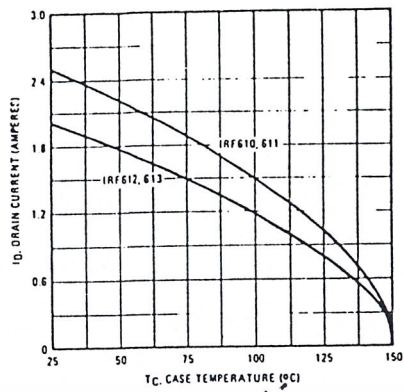


Fig. 13 — Maximum Drain Current Vs. Case Temperature

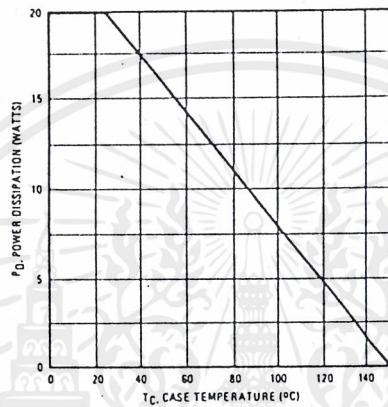


Fig. 14 — Power Vs. Temperature Derating Curve

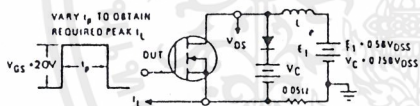


Fig. 15 — Clamped Inductive Test Circuit

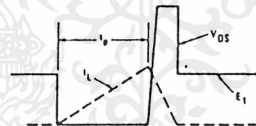


Fig. 16 — Clamped Inductive Waveforms

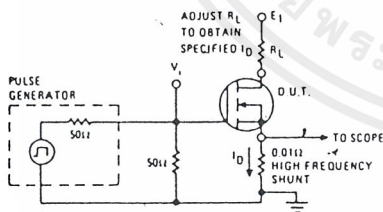


Fig. 17 — Switching Time Test Circuit

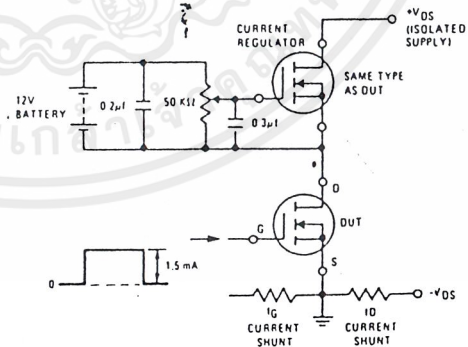
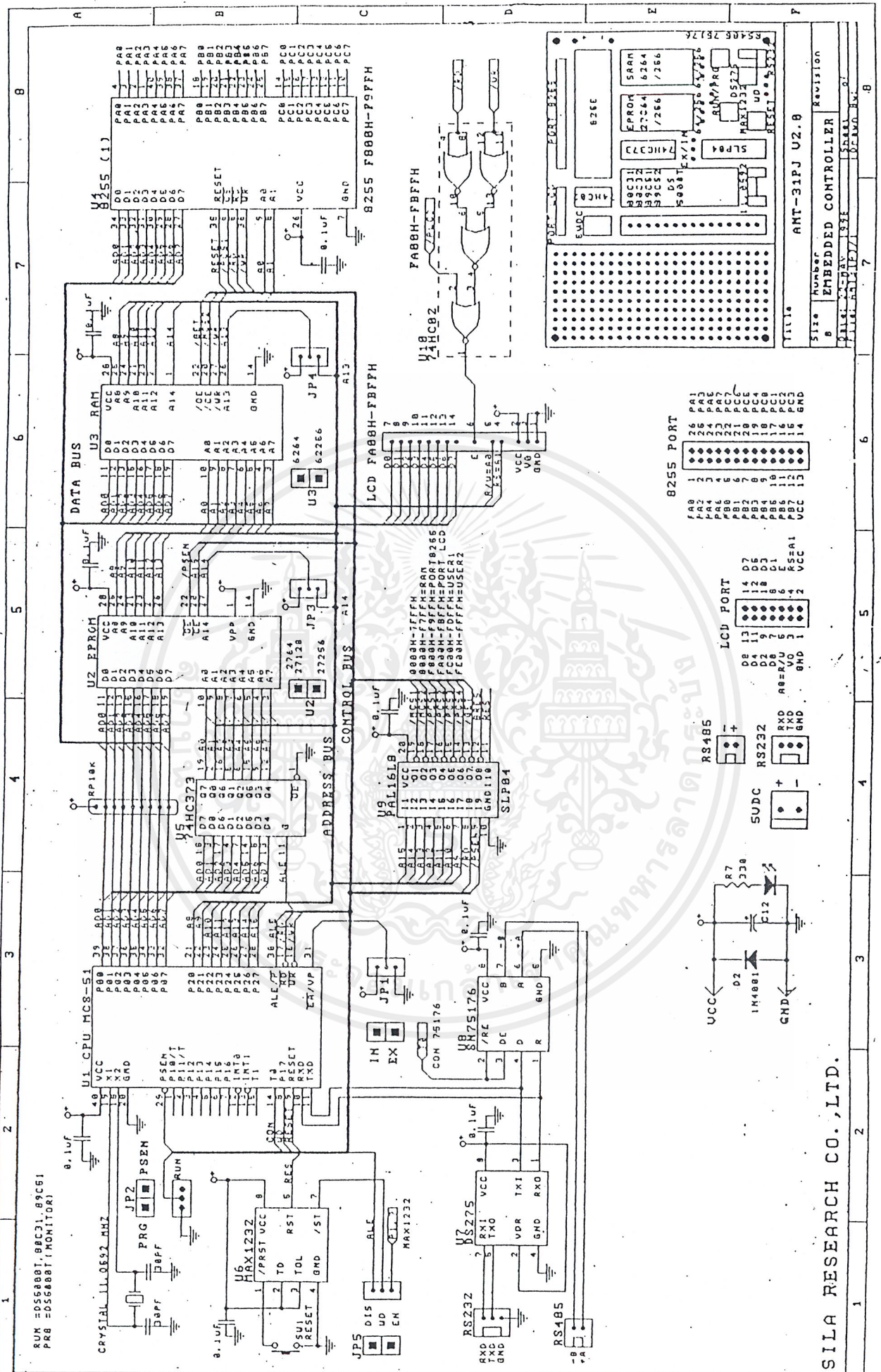


Fig. 18 — Gate Charge Test Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



SILA RESEARCH CO., LTD.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านธุรกิจ  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# DATA SHEET

# 80C51-L / 80C31-L

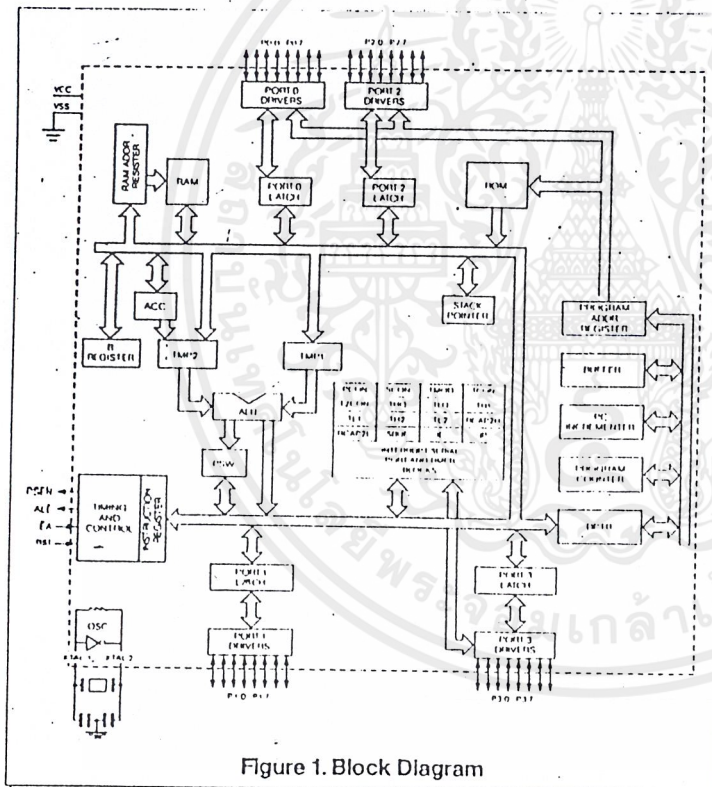
## CMOS SINGLE-CHIP 8 BIT 3V-MICROCONTROLLER

- 80C51-L- CMOS SINGLE-CHIP 8-BIT MICROCONTROLLER with factory mask-programmable ROM
- 80C31-L- CMOS SINGLE-CHIP 8-BIT CONTROL-ORIENTED CPU with RAM and I/O
- 80C51-L/C31-L: 0 TO 6 MHz, VCC = 2.7V TO 6V

### FEATURES

- POWER CONTROL MODES
- 128 x 8 BIT RAM
- 32 PROGRAMMABLE I/O LINES
- TWO 16-BIT TIMER/COUNTERS
- 64K PROGRAM MEMORY SPACE
- FULLY STATIC DESIGN
- HIGH PERFORMANCE SAJI VI CMOS PROCESS
- BOOLEAN PROCESSOR
- 5 INTERRUPT SOURCES
- PROGRAMMABLE SERIAL PORT
- 64K DATA MEMORY SPACE
- TEMPERATURE RANGE: 0 TO 70°C

### DESCRIPTION



MHS's 80C51 and 80C31 are high performance CMOS versions of the 8051/8031 NMOS single chip 8 bit  $\mu$ C and is manufactured using a self-aligned silicon gate CMOS process (SAJI VI).

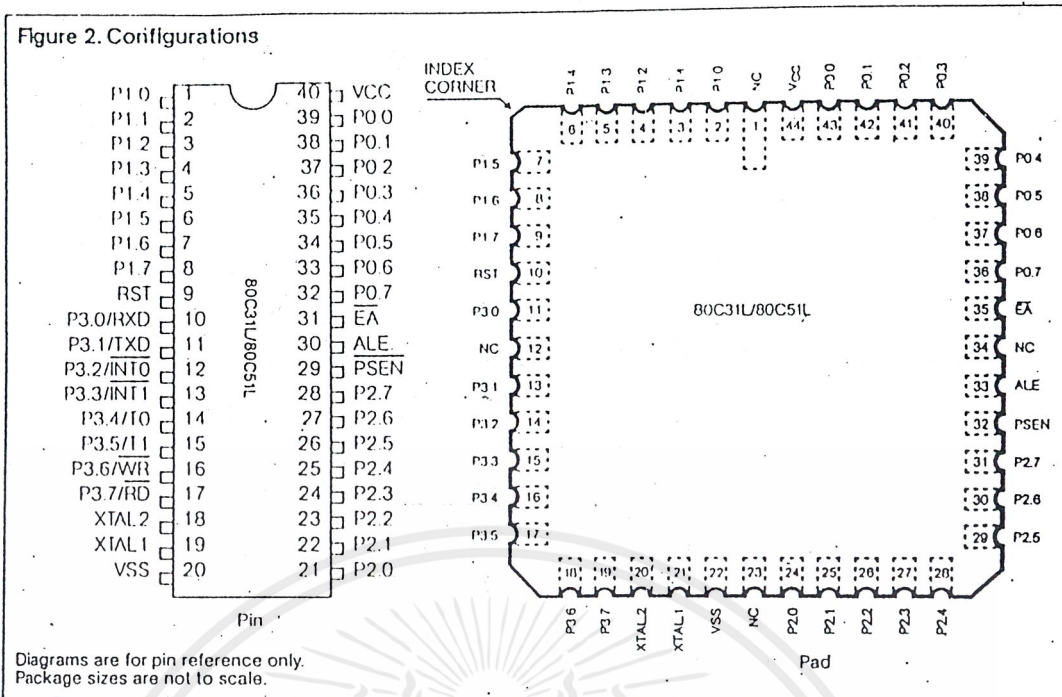
The fully static design of the MHS 80C51/80C31 allows to reduce system power consumption by bringing the clock frequency down to any value, even DC, without loss of data.

The 80C51 retains all the features of the 8051: 4K bytes of ROM; 128 bytes of RAM; 32 I/O lines; two 16 bit timers; a 5-source 2-level interrupt structure; a full duplex serial port; and on-chip oscillator and clock circuits.

In addition, the 80C51 has two software selectable modes of reduced activity for further reduction in power consumption. In the Idle Mode the CPU is frozen while the RAM, the timers, the serial port, and the interrupt system continue to function. In the Power Down Mode the RAM is saved and all other functions are inoperative.

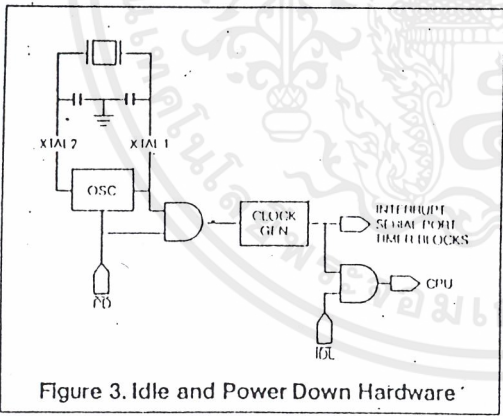
The 80C31 is identical to the 80C51 except that it has no on-chip ROM.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**IDLE AND POWER DOWN OPERATION**

Figure 3 shows the internal Idle and Power Down clock configuration. As illustrated, Power Down operation stops the oscillator. Idle mode operation allows the interrupt, serial port, and timer blocks to continue to function while the clock to the CPU is gated off. These special modes are activated by software via the Special Function Register. Its hardware address is 87H. PCON is not bit addressable.



PCON: Power Control Register

(MSB)							(LSB)
SMOD	-	-	-	GF1	GF0	PD	IDL

Symbol	Position	Name and Function
SMOD	PCON.7	Double Baud rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either modes 1, 2 or 3.
-	PCON.6	(Reserved)
-	PCON.5	(Reserved)
-	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit.
GF0	PCON.2	General-purpose flag bit.
PD	PCON.1	Power Down bit. Setting this bit activates power down operation.
IDL	PCON.0	Idle mode bit. Setting this bit activates idle mode operation.

If 1's are written to PD and IDL at the same time, PD takes precedence. The reset value of PCON is (OXXX0000).

Table 1. Status of the external pins during Idle and Power Down modes

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

**IDLE MODE**

The instruction that sets PCON.0 is the last instruction executed before the Idle mode is activated. Once in the Idle mode the CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM, and all other registers maintain their data during Idle. Table 1 describes the status of the external pins during Idle mode.

There are two ways to terminate the Idle mode. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating Idle mode. The interrupt is serviced, and following RETI, the next instruction to be executed will be the one following the instruction that wrote a 1 to PCON.0.

The flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during the Idle mode. For example, the instruction that writes to PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an enabled interrupt, the service routine can examine the status of the flag bits.

The second way of terminating the Idle mode is with a hardware reset. Since the oscillator is still running, the hardware reset needs to be active for only 2 machine cycles (24 oscillator periods) to complete the reset operation.

**POWER DOWN MODE**

The instruction that sets PCON.1 is the last executed prior to entering power down. Once in power down, the oscillator is stopped. The contents of the onchip RAM and the Special Function Register is saved during power down mode. A hardware reset is the only way of exiting the power down mode. The hardware reset initiates the Special Function Register (see Table 1).

In the Power Down mode, VCC may be lowered to minimize circuit power consumption. Care must be taken to ensure the voltage is not reduced until the power down mode is entered, and that the voltage is restored before the hardware reset is applied which frees the oscillator. Reset should not be released until the oscillator has restarted and stabilized.

Table 1 describes the status of the external pins while in the power down mode. It should be noted that if the power down mode is activated while in external program memory, the port data that is held in the Special Function Register P2 is restored to Port 2. If the data is a 1, the port pin is held high during the power down mode by the strong pullup, T1, shown in Figure 4.

**STOP CLOCK MODE**

Due to static design, the 80C31/C51 clock speed can be reduced until 0 MHz without any data loss in memory or registers. This mode allows step by step utilization, and permits to reduce system power consumption by bringing the clock frequency down to any value. At 0 MHz, the power consumption is the same as in the Power Down Mode.

**80C51 I/O PORTS**

The I/O port drive of the 80C51 is similar to the 8051. The I/O buffers for Ports 1, 2, and 3 are implemented as shown in figure 4.

When the port latch contains a 0, all pFETs in figure 4 are off while the nFET is turned on. When the port latch makes a 0-to-1 transition, the nFET turns off. The strong pullup pFET, T1, turns on for two oscillator periods, pulling the output high very rapidly. As the output line is drawn high, pFET T3 turns on through the inverter to supply the IOH source current. This inverter and T3 form a latch which holds the 1 and is supported by T2. When Port 2 is used as an address port, for access to external program of data memory, any address bit that contains a 1 will have his strong pullup turned on for the entire duration of the external memory access.

When an I/O pin on Ports 1, 2, or 3 is used as an input, the user should be aware that the external circuit must sink current during the logical 1-to-0 transition. The maximum sink current is specified as I<sub>TL</sub> under the D.C. Specifications. When the input goes below approximately 2V, T3 turns off to save ICC current. Note, when returning to a logical 1, T2 is the only internal pullup that is on. This will result in a slow rise time if the user's circuit does not force the input line high.

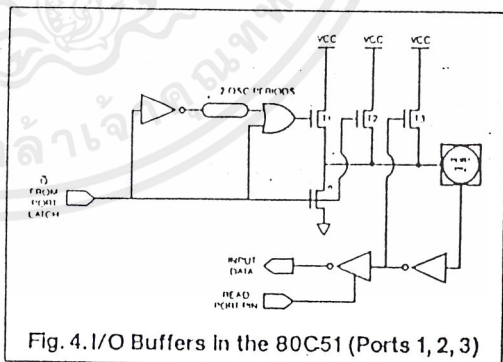


Fig. 4. I/O Buffers in the 80C51 (Ports 1, 2, 3)

## 80C51 PIN DESCRIPTIONS

### VSS

Circuit ground potential

### VCC

Supply voltage during normal, Idle, and Power Down operation.

### Port 0

Port 0 is an 8-bit open drain bi-directional I/O port. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1's. Port 0 also outputs the code bytes during program verification in the 80C51. External pullups are required during program verification. Port 0 can sink eight LS TTL inputs.

### Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. Port 1 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups.

Port 1 also receives the low-order address bytes during program verification. In the 80C51, Port 1 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

### Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. Port 2 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL, on the data sheet) 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, it uses strong internal pullups when emitting 1's. During accesses to external Data Memory that uses 8-bit addresses (MOVX @ Ri), Port 2 emits the contents of the P2 Special Function Register.

It also receives the high-order address bits and control signals during program verification in the 80C51. Port 2 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

### Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pullups. Port 3 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the pullups. It also serves the functions of various special features of the MCS-51 Family, as listed below.

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	WR (external Data Memory write strobe)
P3.7	RD (external Data Memory read strobe)

Port 3 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

### RST

A high level on this for two machine cycles while the oscillator is running resets the device. An internal pull-down resistor permits Power-On reset using only a capacitor connected to VCC.

### ALE

Address Latch Enable output for latching the low byte of the address during accesses to external memory. ALE is activated as though for this purpose at a constant rate of 1/6 the oscillator frequency except during an external data memory access at which time one ALE pulse is skipped. ALE can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pullup.

### PSEN

Program Store Enable output is the read strobe to external Program Memory. PSEN is activated twice each machine cycle during fetches from external Program Memory. (However, when executing out of external Program Memory, two activations of PSEN are skipped during each access to external Data Memory). PSEN is not activated during fetches from internal Program Memory. PSEN can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pullup.

### EA

When EA is held high, the CPU executes out of internal Program Memory (unless the Program Counter exceeds 0FFFh). When EA is held low, the CPU executes only out of external Program Memory. EA must not be floated.

### XTAL1

Input to the inverting amplifier that forms the oscillator. Receives the external oscillator signal when an external oscillator is used.

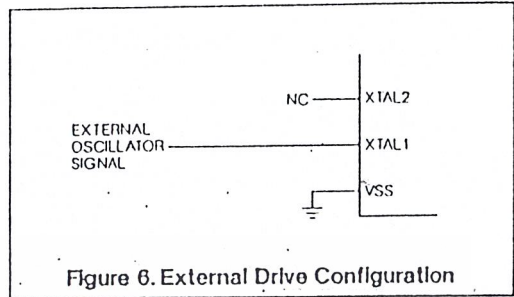
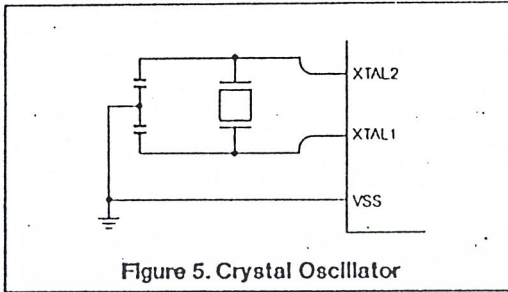
### XTAL2

Output of the inverting amplifier that forms the oscillator, and input to the internal clock generator. This pin should be floated when an external oscillator is used.

### OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output respectively, of an inverting amplifier which is configured for use as an on-chip oscillator, as shown in figure 5. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left

unconnected as shown in figure 6. 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 high and low times specified on the Data Sheet must be observed.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ABSOLUTE MAXIMUM RATINGS\*

Ambient Temperature Under Bias:	
Commercial .....	0°C to 70°C
Industrial .....	- 40°C to 85°C
Storage Temperature .....	- 65°C to + 150°C
Voltage on VCC to VSS .....	- 0.5V to + 7V
Voltage on Any Pin to VSS .....	- 0.5V to VCC + 0.5V
Power Dissipation .....	1W*

\* This value is based on the maximum allowable die temperature and the thermal resistance of the package.

### \*NOTICE:

Stresses at or above 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 above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.

### DC CHARACTERISTICS

TA = -40°C to 85°C; VCC = 2.7V to 6V; VSS = 0V; F = 0 to 6 MHz

Symbol	Parameter	Min	Max	Unit	Test Conditions
VIL	Input Low Voltage	0.5	0.2VCC - 0.1	V	
VIH	Input High Voltage (Except XIALs and RST)	0.2VCC + 0.9	VCC + 0.5	V	
VIH1	Input High Voltage to RST for Reset	0.7VCC	VCC + 0.5	V	
VIH2	Input High Voltage To XIAL 1	0.7VCC	VCC + 0.5	V	
V <sub>PD</sub>	Power Down Voltage To VCC in PD Mode	2.0	6.0	V	
VOL	Output Low Voltage (Ports 1, 2, 3)		0.45	V	IOL = 1.6mA (note 1)
VOL1	Output Low Voltage Port 0, ALE, PSEN		0.45	V	IOL = 3.2mA (note 1)
VOH	Output High Voltage Ports 1, 2, 3	0.9VCC		V	IOH = - 10µA
		2.4		V	IOH = - 60µA VCC = 5V ± 10%
VOH1	Output High Voltage (Port 0 in External in External Bus Mode), ALE, PSEN	0.9VCC		V	IOH = - 40µA
		2.4		V	IOH = - 400µA VCC = 5V ± 10%
IIL	Logical 0 Input Current Ports 1, 2, 3		- 50	µA	Vin = 0.45V
ILI	Input Leakage Current		± 10	µA	0.45 < Vin < VCC
ITL	Logical 1 to 0 Transition Current (Ports 1, 2, 3)		- 500	µA	Vin = 2.0V
ICCPD	Power Supply Current (Power Down Mode)	50	10	µA	VCC = 2.0V to 5.5V (note 2)
RRST	RST Pulldown Resistor	50	150	kΩ	
CIO	Capacitance of I/O Buffer		10	pF	fC = 1MHz, TA = 25°C

#### Note 1:

Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the VOLs of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0

transitions during bus operations. In the worst cases (capacitive loading 100 pF), the noise pulse on the ALE line may exceed 0.45V with maxi VOL peak 0.6V. A Schmitt Trigger use is not necessary.

Table 1. MCS<sup>®</sup>-51 Instruction Set Description

ARITHMETIC OPERATIONS				
Mnemonic		Description	Byte	Cyc
ADD	A,Rn	Add register to Accumulator	1	1
ADD	A,direct	Add direct byte to Accumulator	2	1
ADD	A,@Ri	Add indirect RAM to Accumulator	1	1
ADD	A,#data	Add immediate data to Accumulator	2	1
ADDC	A,Rn	Add register to Accumulator with Carry	1	1
ADDC	A,direct	Add direct byte to A with Carry flag	2	1
ADDC	A,@Ri	Add indirect RAM to A with Carry flag	1	1
ADDC	A,#data	Add immediate data to A with Carry flag	2	1
SUBB	A,Rn	Subtract register from A with Borrow	1	1
SUBB	A,direct	Subtract direct byte from A with Borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A with Borrow	1	1
SUBB	A,#data	Subtract immed. data from A with Borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
INC	DPTR	Increment Data Pointer	1	2
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
MUL	AB	Multiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal Adjust Accumulator	1	1
LOGICAL OPERATIONS				
Mnemonic		Destination	Byte	Cyc
ANL	A,Rn	AND register to Accumulator	1	1
ANL	A,direct	AND direct byte to Accumulator	2	1
ANL	A,@Ri	AND indirect RAM to Accumulator	1	1
ANL	A,#data	AND immediate data to Accumulator	2	1
ANL	direct,A	AND Accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to Accumulator	1	1
ORL	A,direct	OR direct byte to Accumulator	2	1
ORL	A,@Ri	OR indirect RAM to Accumulator	1	1
ORL	A,#data	OR immediate data to Accumulator	2	1
ORL	direct,A	OR Accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to Accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	1
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1
XRL	A,#data	Exclusive-OR immediate data to A	2	1
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear Accumulator	1	1
CPL	A	Complement Accumulator	1	1
RL	A	Rotate Accumulator Left	1	1
RLC	A	Rotate A Left through the Carry flag	1	1
RR	A	Rotate Accumulator Right	1	1
RRC	A	Rotate A Right through Carry flag	1	1
SWAP	A	Swap nibbles within the Accumulator	1	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1. (Cont.)

DATA TRANSFER				
Mnemonic		Description	Byte	Cyc
MOV	A,Rn	Move register to Accumulator	1	1
MOV	A,direct	Move direct byte to Accumulator	2	1
MOV	A,@Ri	Move indirect RAM to Accumulator	1	1
MOV	A,#data	Move immediate data to Accumulator	2	1
MOV	Rn,A	Move Accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move Accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move Accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data 16	Load Data Pointer with a 16-bit constant	3	2
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to A	1	2
MOVC	A,@A+PC	Move Code byte relative to PC to A	1	2
MOVX	A,@Ri	Move External RAM (8-bit addr) to A	1	2
MOVX	A,@DPTR	Move External RAM (16-bit addr) to A	1	2
MOVX	@Ri,A	Move A to External RAM (8-bit addr)	1	2
MOVX	@DPTR,A	Move A to External RAM (16-bit addr)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A,Rn	Exchange register with Accumulator	1	1
XCH	A,direct	Exchange direct byte with Accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with A	1	1
XCHD	A,@Ri	Exchange low-order nibble ind RAM with A	1	1
BOOLEAN VARIABLE MANIPULATION				
Mnemonic		Description	Byte	Cyc
CLR	C	Clear Carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set Carry flag	1	1
SETB	bit	Set direct Bit	2	1
CPL	C	Complement Carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to Carry flag	2	2
ANL	C,1 bit	AND complement of direct bit to Carry	2	2
ORL	C/bit	OR direct bit to Carry flag	2	2
ORL	C,1 bit	OR complement of direct bit to Carry	2	2
MOV	C/bit	Move direct bit to Carry flag	2	1
MOV	bit,C	Move Carry flag to direct bit	2	2
PROGRAM AND MACHINE CONTROL				
Mnemonic		Description	Byte	Cyc
ACALL	addr 11	Absolute Subroutine Call	2	2
LCALL	addr 16	Long Subroutine Call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr 11	Absolute Jump	2	2
LJMP	addr 16	Long Jump	3	2
SJMP	rel	Short Jump (relative addr)	2	2
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if Accumulator is Zero	2	2
JNZ	rel	Jump if Accumulator is Not Zero	2	2
JC	rel	Jump if Carry flag is set	2	2
JNC	rel	Jump if No Carry flag	2	2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1. (Cont.)

PROGRAM AND MACHINE CONTROL (cont.)				
Mnemonic		Description	Byte	Cyc
JB	bit,rel	Jump if direct Bit set	3	2
JNB	bit,rel	Jump if direct Bit Not set	3	2
JBC	bit,rel	Jump if direct Bit is set & Clear bit	3	2
CJNE	A,direct,rel	Compare direct to A & Jump if Not Equal	3	2
CJNE	A,#data,rel	Comp. immed. to A & Jump if Not Equal	3	2
CJNE	Rn,#data,rel	Comp. immed. to reg & Jump if Not Equal	3	2
CJNE	@Ri,#data,rel	Comp. immed. to ind. & Jump if Not Equal	3	2
DJNZ	Rn,rel	Decrement register & Jump if Not Zero	2	2
DJNZ	direct,rel	Decrement direct & Jump if Not Zero	3	2
NOP		No operation	1	1

## Notes on data addressing modes:

- Rn - Working register R0-R7  
 direct - 128 internal RAM locations, any I/O port, control or status register  
 @Ri - Indirect internal RAM location addressed by register R0 or R1  
 #data - 8-bit constant included in instruction  
 #data 16 - 16-bit constant included as bytes 2 & 3 of instruction  
 bit - 128 software flags, any I/O pin, control or status bit

## Notes on program addressing modes:

- addr 16 - Destination address for LCALL & LJMP may be anywhere within the 64-k program memory address space  
 Addr 11 - Destination address for ACALL & AJMP will be within the same 2-k page of program memory as the first byte of the following instruction  
 rel - SJMP and all conditional jumps include an 8-bit offset byte. Range is +127-128 bytes relative to first byte of the following instruction.

All mnemonics copyrighted © Intel Corporation 1979

Table 2. Instruction Opcodes in Hexadecimal Order

Hex Code	Number of Bytes	Mnemonic	Operands
00	1	NOP	
01	2	AJMP	code addr
02	3	LJMP	code addr
03	1	RR	A
04	1	INC	A
05	2	INC	data addr
06	1	INC	@R0
07	1	INC	@R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	bit addr,code addr
11	2	ACALL	code addr
12	3	LCALL	code addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	data addr
16	1	DEC	@R0
17	1	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7
20	3	JB	bit addr,code addr
21	2	AJMP	code addr
22	1	RET	
23	1	RL	A
24	2	ADD	A,data
25	2	ADD	A,data addr
26	1	ADD	A,@R0
27	1	ADD	A,@R1
28	1	ADD	A,R0
29	1	ADD	A,R1
2A	1	ADD	A,R2
2B	1	ADD	A,R3
2C	1	ADD	A,R4
2D	1	ADD	A,R5
2E	1	ADD	A,R6
2F	1	ADD	A,R7
30	3	JNB	bit addr,code addr
31	2	ACALL	code addr
32	1	RETI	
33	1	RLC	A
34	2	ADDC	A,#data
35	2	ADDC	A,data addr
36	1	ADDC	A,@R0
37	1	ADDC	A,@R1
38	1	ADDC	A,R0
39	1	ADDC	A,R1
3A	1	ADDC	A,R2
3B	1	ADDC	A,R3
3C	1	ADDC	A,R4
3D	1	ADDC	A,R5
3E	1	ADDC	A,R6
3F	1	ADDC	A,R7
40	2	JC	code addr
41	2	AJMP	code addr
42	2	ORL	data addr,A
43	3	ORL	data addr,#data
44	2	ORL	A,#data
45	2	ORL	A,data addr
46	1	ORL	A,@R0
47	1	ORL	A,@R1
48	1	ORL	A,R0
49	1	ORL	A,R1
4A	1	ORL	A,R2
4B	1	ORL	A,R3
4C	1	ORL	A,R4
4D	1	ORL	A,R5
4E	1	ORL	A,R6
4F	1	ORL	A,R7
50	2	JNC	code addr
51	2	ACALL	code addr
52	2	ANL	data addr,A
53	3	ANL	data addr,#data
54	2	ANL	A,#data
55	2	ANL	A,data addr
56	1	ANL	A,@R0
57	1	ANL	A,@R1
58	1	ANL	A,R0
59	1	ANL	A,R1
5A	1	ANL	A,R2
5B	1	ANL	A,R3
5C	1	ANL	A,R4
5D	1	ANL	A,R5
5E	1	ANL	A,R6
5F	1	ANL	A,R7
60	2	JZ	code addr
61	2	AJMP	code addr
62	2	XRL	data addr A
63	3	XRL	data addr,#data
64	2	XRL	A,#data
65	2	XRL	A,data addr

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 2. (Cont.)

Hex Code	Number of Bytes	Mnemonic	Operands
CC	1	XCH	A,R4
CD	1	XCH	A,R5
CE	1	XCH	A,R6
CF	1	XCH	A,R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	data addr,code addr
D6	1	XCHD	A,@R0
D7	1	XCHD	A,@R1
D8	2	DJNZ	R0,code addr
D9	2	DJNZ	R1,code addr
DA	2	DJNZ	R2,code addr
DB	2	DJNZ	R3,code addr
DC	2	DJNZ	R4,code addr
DD	2	DJNZ	R5,code addr
DE	2	DJNZ	R6,code addr
DF	2	DJNZ	R7,code addr
E0	1	MOVX	A,@DPTR
E1	2	AJMP	code addr
E2	1	MOVX	A,@R0
E3	1	MOVX	A,@R1
E4	1	CLR	A
E5	2	MOV	A,data addr
E6	1	MOV	A,@R0
E7	1	MOV	A,@R1
E8	1	MOV	A,R0
E9	1	MOV	A,R1
EA	1	MOV	A,R2
EB	1	MOV	A,R3
EC	1	MOV	A,R4
ED	1	MOV	A,R5
EE	1	MOV	A,R6
EF	1	MOV	A,R7
F0	1	MOVX	@DPTRA
F1	2	ACALL	code addr
F2	1	MOVX	@R0,A
F3	1	MOVX	@R1,A
F4	1	CPL	A
F5	2	MOV	data addr,A
F6	1	MOV	@R0,A
F7	1	MOV	@R1,A
F8	1	MOV	R0,A
F9	1	MOV	R1,A
FA	1	MOV	R2,A
FB	1	MOV	R3,A
FC	1	MOV	R4,A
FD	1	MOV	R5,A
FE	1	MOV	R6,A
FF	1	MOV	R7,A



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 2. (Cont.)

Hex Code	Number of Bytes	Mnemonic	Operands
66	1	XRL	A,@R0
67	1	XRL	A,@R1
68	1	XRL	A,R0
69	1	XRL	A,R1
6A	1	XRL	A,R2
6B	1	XRL	A,R3
6C	1	XRL	A,R4
6D	1	XRL	A,R5
6E	1	XRL	A,R6
6F	1	XRL	A,R7
70	2	JNZ	code addr
71	2	ACALL	code addr
72	2	ORL	C, bit addr
73	1	JMP	@A1, DPTR
74	2	MOV	A, #data
75	3	MOV	data addr, #data
76	2	MOV	@R0, #data
77	2	MOV	@R1, #data
78	2	MOV	R0, #data
79	2	MOV	R1, #data
7A	2	MOV	R2, #data
7B	2	MOV	R3, #data
7C	2	MOV	R4, #data
7D	2	MOV	R5, #data
7E	2	MOV	R6, #data
7F	2	MOV	R7, #data
80	2	SJMP	code addr
81	2	AJMP	code addr
82	2	ANL	C, bit addr
83	1	MOVC	A, @A1, PC
84	1	DIV	AB
85	3	MOV	data addr, data addr
86	2	MOV	data addr, @R0
87	2	MOV	data addr, @R1
88	2	MOV	data addr, R0
89	2	MOV	data addr, R1
8A	2	MOV	data addr, R2
8B	2	MOV	data addr, R3
8C	2	MOV	data addr, R4
8D	2	MOV	data addr, R5
8E	2	MOV	data addr, R6
8F	2	MOV	data addr, R7
90	3	MOV	DPTR, #data
91	2	ACALL	code addr
92	2	MOV	bit addr, C
93	1	MOVC	A, @A1, DPTR
94	2	SUBB	A, #data
95	2	SUBB	A, data addr
96	1	SUBB	A, @R0
97	1	SUBB	A, @R1
98	1	SUBB	A, R0
99	1	SUBB	A, R1
9A	1	SUBB	A, R2
9B	1	SUBB	A, R3
9C	1	SUBB	A, R4
9D	1	SUBB	A, R5
9E	1	SUBB	A, R6
9F	1	SUBB	A, R7
A0	2	ORL	C, bit addr
A1	2	AJMP	code addr
A2	2	MOV	C, bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5			reserved
A6	2	MOV	@R0, data addr
A7	2	MOV	@R1, data addr
A8	2	MOV	R0, data addr
A9	2	MOV	R1, data addr
AA	2	MOV	R2, data addr
AB	2	MOV	R3, data addr
AC	2	MOV	R4, data addr
AD	2	MOV	R5, data addr
AE	2	MOV	R6, data addr
AF	2	MOV	R7, data addr
B0	2	ANL	C, bit addr
B1	2	ACALL	code addr
B2	2	CPL	bit addr
B3	1	CPL	C
B4	3	CJNE	A, #data, code addr
B5	3	CJNE	A, data addr, code addr
B6	3	CJNE	@R0, #data, code addr
B7	3	CJNE	@R1, #data, code addr
B8	3	CJNE	R0, #data, code addr
B9	3	CJNE	R1, #data, code addr
BA	3	CJNE	R2, #data, code addr
BB	3	CJNE	R3, #data, code addr
BC	3	CJNE	R4, #data, code addr
BD	3	CJNE	R5, #data, code addr
BE	3	CJNE	R6, #data, code addr
BF	3	CJNE	R7, #data, code addr
C0	2	PUSH	data addr
C1	2	AJMP	code addr
C2	2	CLR	bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A, data addr
C6	1	XCH	A, @R0
C7	1	XCH	A, @R1
C8	1	XCH	A, R0
C9	1	XCH	A, R1
CA	1	XCH	A, R2
CB	1	XCH	A, R3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## CMOS Programmable Interval Timer

### Features

- 8MHz to 12MHz Clock Input Frequency
- Compatible with NMOS 8254
  - Enhanced Version of NMOS 8253
- Three Independent 16-Bit Counters
- Six Programmable Counter Modes
- Status Read Back Command
- Binary or BCD Counting
- Fully TTL Compatible
- Single 5V Power Supply
- Low Power
  - ICCSB ..... 10 $\mu$ A
  - ICCOP ..... 10mA at 8MHz
- Operating Temperature Ranges
  - C82C54 ..... 0 $^{\circ}$ C to +70 $^{\circ}$ C
  - I82C54 ..... -40 $^{\circ}$ C to +85 $^{\circ}$ C
  - M82C54 ..... -55 $^{\circ}$ C to +125 $^{\circ}$ C

### Description

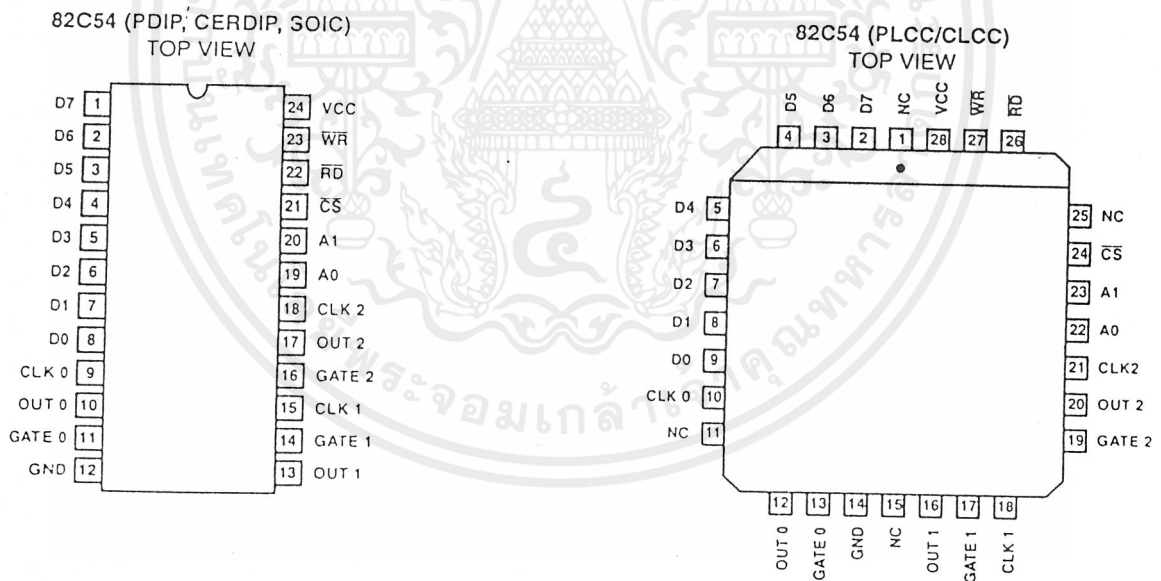
The Harris 82C54 is a high performance CMOS Programmable Interval Timer manufactured using an advanced 2 micron CMOS process.

The 82C54 has three independently programmable and functional 16-bit counters, each capable of handling clock input frequencies of up to 8MHz (82C54) or 10MHz (82C54-10) or 12MHz (82C54-12).

The high speed and industry standard configuration of the 82C54 make it compatible with the Harris 80C86, 80C88, and 80C286 CMOS microprocessors along with many other industry standard processors. Six programmable timer modes allow the 82C54 to be used as an event counter, elapsed time indicator, programmable one-shot, and many other applications. Static CMOS circuit design insures low power operation.

The Harris advanced CMOS process results in a significant reduction in power with performance equal to or greater than existing equivalent products.

### Pinouts



## กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ ต้องขอขอบคุณ

- ดร.วิจิตร กิมเรศ อ.อนุวัฒน์ จางวนิชเลิศ และ อ.สุรินทร์ คำฝอย ที่ได้ให้คำแนะนำต่างๆตลอด  
ทำโครงการ
- ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ของสถาบันฯ ที่ได้เอื้อเฟื้อสถานที่ เครื่องมือ  
และอุปกรณ์ต่างๆในการทำโครงการ
- เพื่อนร่วมห้อง 3 W ที่ให้อุปกรณ์ด้วยดีเสมอมา

ซึ่งบุคคลเหล่านี้มีส่วนทำให้โครงการนี้สามารถสำเร็จลุล่วงไปได้ด้วยดีและสามารถนำผลการ  
ทดลองที่ได้มาเขียนเป็นปริญญานิพนธ์ฉบับนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เอกสารอ้างอิง

1. โยธิน เปรมปราณีรัตน์, “วิเคราะห์และออกแบบระบบการควบคุมมอเตอร์”, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, กรุงเทพมหานคร, 2526
2. สุวัฒน์ คั่น, “เทคนิคและการออกแบบสวิทช์িংเพาเวอร์ซัพพลาย”, เอนเทคไทย จำกัด, กรุงเทพ, 2538
3. สมยศ จุณณะปิยะ, “การใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51”, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, กรุงเทพมหานคร, 2537
4. Rashid M.H., “Power Electronics”, Prentice-Hall, Inc., USA 1988, 585pp
5. Takashi Kenjo, “Power electronics for the Microprocessor Age”, Oxford University Press, 1990

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้