สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ปีการศึกษา 2543

การควบคุมตำแหน่งของเซอร์โวมอเตอร์กระแสตรงโดยไมโครโปรเซสเซอร์

MICROPROCESSOR CONTROL OF DC SERVO MOTOR FOR POSITION CONTROL

T042610

โดย

นายทศพล คีระกิตติสุวรรณ์

นายปรีชา คงถาวรธรรม

นายสุรัตน์ ตรีธนากร

เลขหมู่.....
เลขทะเบียน 42610
วัน, เดือน, ปี - 4 มิ.ย. 2545

อาจารย์ที่ปรึกษา

.b.....

.i.....

ผศ.ดร.วิจิตร กิณเรศ

อ.สุรินทร์ คำฝอย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2543

ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การควบคุมตำแหน่งของเซอร์โวมอเตอร์กระแสตรงโดยไมโครโปรเซสเซอร์

ผู้จัดทำ

1. นายทศพล คีระกิตติสุวรรณ์
2. นายปรีชา คงถาวรธรรม
3. นายสุรัตน์ ตรีธนากร

_____ อาจารย์ที่ปรึกษา

(ผศ.ดร.วิจิตร กิณเรศ)

_____ อาจารย์ที่ปรึกษา

(อ.สุรินทร์ คำฝอย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมตำแหน่งของเซอร์โวมอเตอร์กระแสตรงโดยไมโครโปรเซสเซอร์

นายทศพล คีระกิตติสุวรรณ์
นายปรีชา คงถาวรธรรม
นายสุรัตน์ ตรีธนากร
ผศ.ดร.วิจิตร กิณเรศ อาจารย์ที่ปรึกษา
อ.สุรินทร์ คำฝอย อาจารย์ที่ปรึกษา
ปีการศึกษา 2543

บทคัดย่อ

โครงงานนี้เป็นการศึกษาวิชาการควบคุมตำแหน่งของเซอร์โวมอเตอร์กระแสตรงขนาดพิกัดกำลัง80วัตต์ ซึ่งมีเกียร์-บล็อคอยู่ที่ตัวมอเตอร์และมีเอ็นโค้ดเดอร์โดยใช้การควบคุมแบบป้อนกลับและใช้ไมโครคอนโทรลเลอร์ควบคุมการทำงานทั้งหมด ให้สามารถขับเคลื่อนมอเตอร์ได้ใน 4 ควอดแดรนด์ การขับมอเตอร์ใช้เทคนิคพีดับบลิวเอ็ม – เซอร์โวแอมปลิฟายเออร์ โดยสร้างสัญญาณพีดับบลิวเอ็มจากชิพสนับสนุน 8253 เพาเวอร์มอสเฟตจะเป็นวงจรกำลังที่ใช้เป็นอุปกรณ์สวิตชิ่ง ซึ่งวงจรกำลังวางตัวในลักษณะ H – Bridgc ซึ่งเทคนิคพีดับบลิวเอ็มจะช่วยลดกำลังสูญเสียในอุปกรณ์สวิตชิ่งและความร้อน การควบคุมมีการป้อนกลับค่าความเร็วรอบโดยใช้เอ็นโค้ดเดอร์เป็นอุปกรณ์ในการวัดเพื่อนำค่าที่ได้กลับมาใช้ในการประมวลผลสัญญาณควบคุมตำแหน่ง การเปลี่ยนทิศทาง และการเบรก โดยการจำลองการทำงานของชุดสายพาน เพื่อให้ได้ค่าความแม่นยำมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MICROPROCESSOR CONTROL OF DC SERVO MOTOR FOR POSITIONCONTROL

Thossaphol Keerakittisuwan
Preecha Khongtawontam
Surat Thithanakorn
Ast.Dr.Vijit Kinnarase Advisor
Surin Comefoy Advisor
2000

ABSTRACT

This project presents a manner to control the position of 80 W DC servo motor with gear box and encoder 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 IC8253. The presence of power circuit is the mosfet power transistor for switching by 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 perform. The results are satisfactory.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
ABSTRACT	II
สารบัญ	III
สารบัญรูป	V
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีพื้นฐาน	3
2.1 เซอร์โวมอเตอร์	3
2.2 เซ็นเซอร์และเอ็นโค้ดเดอร์	9
2.2.1 ทาโคเยนเนอร์โรเตอร์	9
2.2.2 เอ็นโค้ดเดอร์	9
2.3 PWM Servo-Amplifier	14
2.3.1 หลักการของเซอร์โวแอมปลิไฟร์ชนิด PWM แบบควบคุมแรงดัน	15
2.3.2 เซอร์โวแอมปลิไฟร์ชนิด PWM แบบควบคุมกระแส	17
2.3.3 เซอร์โวแอมปลิไฟร์ชนิด PWM แบบบริดจ์	17
2.4 การเลือกใช้งานอุปกรณ์สวิตชิ่ง	20
2.4.1 เพาเวอร์ทรานซิสเตอร์	20
2.4.2 เพาเวอร์มอสเฟต	20
2.4.3 ข้อดีของมอสเฟตเมื่อเปรียบเทียบกับทรานซิสเตอร์	21
บทที่ 3 การใช้งานไมโครคอนโทรลเลอร์	24
3.1 คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ตระกูล MCS-51	25
3.1.1 โครงสร้างภายนอกของMCS-51	26
3.1.2 โครงสร้างภายในของMCS-51	28
3.2 การเชื่อมโยง 8253 กับ MCS-51	29
3.2.1 โครงสร้างของ 8253	29
3.2.2 รีจีสเตอร์ควบคุม	31
3.2.3 โหมดและการทำงานของ 8253	31
3.2.4 การโปรแกรม 8253	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 การต่อ 8253 เข้ากับ MCS-51 36
3.2.6 การทำงานโหมดต่างๆ 37
3.3 การใช้ LCD Module 40
3.3.1 คอนโทรลเลอร์และการควบคุม 41
3.3.2 การอ่านและการเขียนข้อมูล 45
3.3.3 การเชื่อมต่อ 47
3.4 คีย์บอร์ด 48
บทที่ 4 หลักการและขั้นตอนการทำงาน 51
4.1 รูปแบบโครงงาน 51
4.2 เทคนิคการสร้างสัญญาณ PWM 52
4.3 วงจรควบคุม และขับเกทมอสเฟต 53
4.4 วงจรขับมอเตอร์ 56
4.5 การตรวจจับตำแหน่ง 56
4.6 ส่วนของโปรแกรมการทำงาน 57
4.7 ชุดสายพานลำเลียง 59
บทที่ 5 การทดสอบโครงงาน 60
5.1 การทดสอบหาค่าพารามิเตอร์ต่างของมอเตอร์ 60
5.2 การทดสอบสัญญาณพัลซ์จากเอ็นโค้ดเดอร์ 64
5.3 การทดสอบการสร้างสัญญาณพัลซ์วิดท์มอดูเลชั่น 66
5.4 การทดสอบสัญญาณขับเกท และ การทดสอบภาคขับมอเตอร์ 68
5.4.1 การวัดสัญญาณที่ได้จากวงจรขับเกท 68
5.4.2 การวัดสัญญาณที่ได้จากมอเตอร์ 69
5.5 การทดสอบโครงงานทั้งหมด 70
บทที่ 6 สรุปผลและแนวทางการพัฒนา 80
ภาคผนวก
กิตติกรรมประกาศ
เอกสารอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
รูปที่ 2.1 แสดงบล็อกโมเดลอีเล็คโทรแมคคานิคอล	5
รูปที่ 2.2 แสดงโมเดลของดีซีมอเตอร์แบบฟิลด์แยกกระตุ้น	6
รูปที่ 2.3 แสดงถึงแรงบิดต่างๆที่เกิดขึ้นต่อโหลดของมอเตอร์	8
รูปที่ 2.4 ตัวอย่างกลไกของเอ็นโค้ดเดอร์	10
รูปที่ 2.5 (a) ตัวอย่างรูปคลื่นเอาท์พุทสี่เหลี่ยมของอุปกรณ์เอนโค้ดเดอร์ช่องเดียว (ไบไดเร็ก) (b) ตัวอย่างสัญญาณเอนโค้ดเดอร์ 2 ช่องมีมุมเฟสต่างกัน 90° (สองทิศทาง)	11
รูปที่ 2.6 เซอร์โวมอเตอร์ที่มีทาโคเยนเนอเรเตอร์ร่วมอยู่บนเพลาเดียวกัน	12
รูปที่ 2.7 ดีซีมอเตอร์ที่มีเอ็นโค้ดเดอร์และทาโคเยนเนอเรเตอร์	13
รูปที่ 2.8 ดีซีมอเตอร์ที่มีเกียร์บล็อค	13
รูปที่ 2.9 เซอร์โวแอมปลิไฟร์ชนิด PWMแบบควบคุมแรงดันพื้นฐาน (a) วงจร (b) รูปคลื่น	14
รูปที่ 2.10 (a) รูปคลื่นและกระแสพัลส์วิดธ์มอดูเลชั่น (b) ทางไหลของกระแสเมื่อแรงดันเฉลี่ยเป็นบวกกระแสไหลผ่าน Q1 เมื่อมันทำงาน และQ2 ไม่ทำงาน และจะไหลผ่าน D2 เมื่อ Q1 ไม่ทำงาน Q2 ทำงาน	15
รูปที่ 2.11 แรงดันเอาท์พุทของเซอร์โวแอมปลิไฟร์ชนิด PWM	16
รูปที่ 2.10 เป็นฟังก์ชั่นของอัตราส่วนเวลา	
รูปที่ 2.12 เซอร์โวแอมปลิไฟร์ชนิด PWM แบบความคุมกระแส	17
รูปที่ 2.13 วงจรพื้นฐานของเซอร์โวแอมปลิไฟร์ชนิด PWM แบบบริดจ์	17
รูปที่ 2.14 รูปคลื่น PWM ที่สวิงระหว่าง +E และ -E เมื่อใช้ทรานซิสเตอร์ทั้งสี่ตัวทำงานแบบสวิตช์	18
รูปที่ 2.15 (a) รูปคลื่น PWM ที่สวิงระหว่าง +E และศูนย์ หรือ -E และศูนย์ เมื่อใช้ทรานซิสเตอร์สองตัวทำงานแบบสวิตช์	19
(b) Q4 ทำงาน และ Q1 ถูกขับด้วย PWM (d) Q2 ทำงาน และ Q3 ถูกขับด้วย PWM	
(c) Q1 ทำงาน และ Q4 ถูกขับด้วย PWM (e) Q3 ทำงาน และ Q2 ถูกขับด้วย PWM	
รูปที่ 2.16 แสดงการรวมวิธีการในรูปที่ 2.14 และ 2.15 เข้าด้วยกัน	19
รูปที่ 2.17 แสดงลักษณะของกระแสและแรงดันตกคร่อมมอสเฟต เมื่อนำกระแสและหยุดนำกระแสซึ่งจะเกิดกำลังงานสูญเสียความร้อน	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.18 (a) ลักษณะสมบัติทางเอาท์พุทของมอสเฟต 22
(b) ลักษณะสมบัติทางเอาท์พุทของทรานซิสเตอร์
รูปที่ 2.19 แสดงพื้นที่ปลอดภัยของมอสเฟตและทรานซิสเตอร์ 23
รูปที่ 3.1 บล็อกไดอะแกรมของ MCS-51 28
รูปที่ 3.2 การวางขาของ 8253 29
รูปที่ 3.3 บล็อกไดอะแกรมแสดงผังภายในของ 8253 30
รูปที่ 3.4 แสดงการทำงานของ 8253 ทั้ง 6 โหมด 32
รูปที่ 3.5 แสดงการต่อ 8253 กับ MCS-51 36
รูปที่ 3.6 ผังเวลาการทำงาน และการป้อนสัญญาณของ 8253 (โหมด 0) 37
รูปที่ 3.7 การทำงานของโหมด 1 เมื่อทริกครั้งเดียว 37
รูปที่ 3.8 การทำงานของโหมด 1 เมื่อมีการทริกซ้ำ 38
รูปที่3.9 เอาท์พุทของโหมด 2 38
รูปที่ 3.10 รูปคลื่นเอาท์พุทเมื่อใช้ค่านับเลขคู่ และเลขคี่ (โหมด 3) 39
รูปที่ 3.11 การทำงานของโหมด 4 39
รูปที่ 3.12 ไทม์มิ่งไดอะแกรม 46
(ก) เป็นการเขียนข้อมูลลงบนโมดูล LCD
(ข) เป็นการอ่านข้อมูลจากโมดูล LCD
รูปที่ 3.13 แสดงการต่อโมดูล LCD ร่วมกับไมโครคอนโทรลเลอร์ 8051 47
รูปที่ 3.14 แสดงวงจรของสวิตซ์ 48
รูปที่ 3.15 แสดงการเกิดบราวซ์ขณะกดสวิตซ์ 48
รูปที่ 3.16 โฟลว์ชาร์ทแสดงการแก้สัญญาณบราวซ์ 49
รูปที่ 3.17 แสดงการต่อวงจร 8051 กับคีย์บอร์ด แบบเมตริกซ์ 50
รูปที่ 4.1 แสดงระบบโดยรวมของโครงงาน 51
รูปที่ 4.2 แสดงการต่อวงจรสร้างพัลซ์วิดท์มอดูเลชั่น 52
รูปที่ 4.3 แสดงวงจรการควบคุมการหมุนของมอเตอร์ 53
รูปที่ 4.4 แสดงวงจรที่ใช้ขับมอสเฟต 54
รูปที่ 4.5 แสดงวงจรที่ใช้ในโครงงานในการขับเกทมอสเฟต 55
รูปที่ 4.6 แสดงวงจรที่ใช้ในการขับมอเตอร์ 56
รูปที่ 4.7 แสดงบล็อกไดอะแกรมการทำงานเคาน์เตอร์ (Mode 1) 57
รูปที่ 4.8 แสดง โฟล์วชาร์ทการทำงานของโปรแกรมหลัก 58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.9 แสดงชุดสายพานลำเลียงที่ใช้ในโครงงาน 59
รูปที่ 4.10 แสดงภาพของโครงงานทั้งหมด 59
รูปที่ 5.1 แสดงกราฟความสัมพันธ์ระหว่างแรงดันและความเร็วรอบของมอเตอร์ 61
รูปที่ 5.2 แสดงวงจรการทดสอบค่า Inductance ของมอเตอร์ 63
รูปที่ 5.3 แสดงสัญญาณพัลซ์ที่ได้จากเอ็น โค้ดเดอร์ 65
รูปที่ 5.4 แสดงสัญญาณพัลซ์วิดท์มอดูเลชั่น 67
รูปที่ 5.5 แสดงสัญญาณที่ใช้ในการขับเพาเวอร์มอสเฟต 68
รูปที่ 5.6 แสดงสัญญาณที่จ่ายให้กับตัวมอเตอร์ 69
รูปที่ 5.7 แสดงวิธีการวัดระยะทางในขณะทดลอง 70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 3.1 แสดงคุณสมบัติของไมโครคอนโทรลเลอร์แต่ละเบอร์ตระกูล MCS-51	25
ตารางที่ 3.2 แสดงหน้าที่พิเศษของแต่ละขาของพอร์ต P3	27
ตารางที่ 3.3 แสดงค่าความจริงของ 8253	31
ตารางที่ 3.4 แสดงการทำงานของขาเกททั้ง 6 โหมด	33
ตารางที่ 3.5 แสดงวิธีการเลือก Counter	34
ตารางที่ 3.6 แสดงวิธีการเลือก RL	34
ตารางที่ 3.7 แสดงการเลือกโหมดการทำงาน	34
ตารางที่ 3.8 แสดงพอร์ตของ 8253 ตามรูปที่ 3.5	36
ตารางที่ 3.9 แสดงตำแหน่งขาต่างๆที่ใช้เชื่อมต่อกับ LCD	42
ตารางที่ 3.10 แสดงรายละเอียดของชุดคำสั่ง	44
ตารางที่ 4.1 แสดงค่าความจริงในการควบคุมมอเตอร์	54
ตารางที่ 5.1 แสดงการบันทึกค่าความเร็วรอบและกระแส	61
ตารางที่ 5.2 แสดงค่าพารามิเตอร์ของมอเตอร์ที่ได้	62
ตารางที่ 5.3 แสดงค่า Inductance ของมอเตอร์ที่ได้จากการทดลอง	63
ตารางที่ 5.4 แสดงการทดสอบขณะ NO-LOAD ที่ความเร็วสายพาน 3 เมตรต่อนาที	70
ตารางที่ 5.5 แสดงการทดสอบขณะ NO-LOAD ที่ความเร็วสายพาน 6 เมตรต่อนาที	71
ตารางที่ 5.6 แสดงการทดสอบขณะ NO-LOAD ที่ความเร็วสายพาน 9 เมตรต่อนาที	72
ตารางที่ 5.7 แสดงการทดสอบขณะขับ LOAD ขนาด 0.5 kg ที่ความเร็วสายพาน 3 เมตรต่อนาที	73
ตารางที่ 5.8 แสดงการทดสอบขณะขับ LOAD ขนาด 0.5 kg ที่ความเร็วสายพาน 6 เมตรต่อนาที	74
ตารางที่ 5.9 แสดงการทดสอบขณะขับ LOAD ขนาด 0.5 kg ที่ความเร็วสายพาน 9 เมตรต่อนาที	75
ตารางที่ 5.10 แสดงการทดสอบขณะขับ LOAD ขนาด 1 kg ที่ความเร็วสายพาน 3 เมตรต่อนาที	76
ตารางที่ 5.11 แสดงการทดสอบขณะขับ LOAD ขนาด 1 kg ที่ความเร็วสายพาน 6 เมตรต่อนาที	77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.12 แสดงการทดสอบขณะขับ LOAD ขนาด 1 kg ที่ความเร็วสายพาน 9 เมตรต่อนาที 78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของโครงงาน

ในปัจจุบันมอเตอร์ได้ถูกนำมาใช้ในงานการเคลื่อนที่ทางกลอย่างกว้างขวาง ซึ่งแต่ละงานก็จะมีการควบคุมค่าต่างๆเพื่อให้มอเตอร์ทำงานได้ตามที่ต้องการ ค่าต่างๆที่ถูกควบคุมก็คือ ความเร็วรอบ แรงบิด และตำแหน่งในการหมุน มอเตอร์ส่วนใหญ่ที่ใช้จึงเป็นมอเตอร์กระแสตรง เพราะเนื่องจากควบคุมได้ง่าย และให้แรงบิดที่ดี ในการควบคุมให้มอเตอร์ทำงานได้ตามที่ต้องการตามโครงงานนั้น ตัวมอเตอร์ต้องมีตัวตรวจจับค่าที่จะควบคุมที่อยู่ในรูปของสัญญาณไฟฟ้า ซึ่งก็คือเอ็นโค้ดเดอร์ เพื่อนำสัญญาณนั้นป้อนกลับเข้ามาควบคุม เราเรียกมอเตอร์ที่มีตัวตรวจจับสัญญาณนี้ว่า เซอร์โวมอเตอร์ ซึ่งเซอร์โวมอเตอร์มีความแม่นยำมาก จึงได้นำมาใช้ในโครงงานนี้เพื่อที่จะควบคุมตำแหน่ง และความเร็วรอบโดยการสร้างสัญญาณพัลซ์วิดท์มอดูเลชั่นเพื่อนำสัญญาณนี้ไปจ่ายให้แก่ภาควงจรขับ และใช้ไมโครคอนโทรโทรลเลอร์มาควบคุมการทำงานของมอเตอร์ต่อไป

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อศึกษาโครงสร้าง การทำงาน และการควบคุมดีซีเซอร์โวมอเตอร์แบบป้อนกลับ เพื่อควบคุมการหยุดตามตำแหน่ง

1.2.2 เพื่อศึกษาการสร้างสัญญาณพัลซ์วิดท์มอดูเลชั่น

1.2.3 เพื่อศึกษาการใช้งานของไมโครคอนโทรลเลอร์ในการควบคุม

1.2.4 เพื่อศึกษาการทำงานของเพาเวอร์มอสเฟต

1.2.5 เพื่อศึกษาการเขียนโปรแกรมด้วยภาษาแอสแซมบลี

1.3 ขั้นตอนการดำเนินงาน

1.3.1 ศึกษาถึงรายละเอียดการทำงานของดีซีเซอร์โวมอเตอร์ การใช้ไมโครคอนโทรลเลอร์ตลอดจนวงจรที่ใช้ในโครงงาน

1.3.2 ทำการออกแบบชุดสายพาน ชุดควบคุม และชุดวงจรกำลัง

1.3.3 เขียนโปรแกรมคำสั่งเพื่อมาควบคุมการทำงานของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 สามารถควบคุมตำแหน่งการหยุดโดยการป้อนกลับของดีซีเซอร์โวมอเตอร์ได้

1.4.2 สามารถสร้างสัญญาณพัลซ์วิดท์มอดูเลชั่นจากไอซี 8253 ที่สามารถปรับค่าดิวตี้ไซเคิลได้โดยการควบคุมจากไมโครคอนโทรลเลอร์

1.4.3 สามารถใช้คำสั่งต่างๆในภาษาแอสแซมบลีเพื่อนำมาควบคุมได้

1.4.4 สามารถประยุกต์การใช้งานไมโครคอนโทรลเลอร์ในงานอื่นๆได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2
ทฤษฎีพื้นฐาน

คำนำ

ในบทนี้จะกล่าวถึง ลักษณะและคุณสมบัติของมอเตอร์ และทฤษฎีต่างๆที่นำมาใช้ในการออกแบบการควบคุมการทำงานของมอเตอร์ ซึ่งสามารถควบคุมได้หลายแบบแล้วแต่วัตถุประสงค์การใช้งาน

2.1 เซอร์โวมอเตอร์ (SERVO MOTER)

เซอร์โวมอเตอร์ส่วนใหญ่ใช้ในงานที่มีการควบคุมแบบลูปปิด (Close Loop) หรือแบบมีการป้อนกลับ ซึ่งสามารถควบคุมความเร็วและตำแหน่งของการทำงานได้อย่างแม่นยำ จะพบได้ในงานเครื่องจักรกลอัตโนมัติทั่วไป เช่น หุ่นยนต์คอมพิวเตอร์ เครื่องมือวัดคุมเป็นต้น

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

เซอร์โวมอเตอร์กระแสตรง (DC Servo Motor)

ดีซีเซอร์โวมอเตอร์ เราสามารถแบ่งออกเป็นหลายประเภทขึ้นอยู่กับวิธีสร้างสนามเหล็กของตัวมอเตอร์และขึ้นอยู่กับพื้นฐานการออกแบบโครงสร้างของอาเมเจอร์ การแบ่งประเภทตามลักษณะการจ่ายสนามแม่เหล็กแยกได้เป็น 2 แบบ คือ

1. ดีซีมอเตอร์แบบปรับเส้นแรงแม่เหล็กได้
2. ดีซีมอเตอร์แบบเส้นแรงแม่เหล็กคงที่

ถ้าพิจารณาแยกประเภทตามลักษณะการออกแบบโครงสร้างอาร์เมเจอร์สามารถแยกออกได้เป็น 3 แบบ คือ

1. ดีซีมอเตอร์แบบอาร์เมเจอร์เป็นแกนเหล็ก
2. ดีซีมอเตอร์แบบอาร์เมเจอร์ที่มีขดลวดพันอยู่บนพื้นผิว
3. ดีซีมอเตอร์แบบอาร์เมเจอร์เป็นขดลวดหมุน

นอกจากนี้ยังมีดีซีมอเตอร์ชนิดพิเศษอีกแบบหนึ่งคือ แบบไม่มีแปรงถ่าน (Brushless DC Motor) ซึ่งมีหลักการทางเทคโนโลยีเหมือนกับ ดีซีมอเตอร์ชนิดแปรงถ่าน ยกเว้นการคอมมิวเตชั่นกระทำโดยทางอิเล็คทรอนิคส์แทนที่จะกระทำโดยวิธีการทางกลเหมือนดีซีมอเตอร์ทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบการกระตุ้นฟิลด์ของมอเตอร์โดยทั่วไป ในปัจจุบันมักใช้เป็นแบบแม่เหล็กถาวร ในระบบนี้เส้นแรงแม่เหล็กของฟิลด์มีค่าคงที่ ดังนั้นอัตราส่วนระหว่างกระแสอามาเจอร์และแรงบิดจะมีค่าคงที่

สามารถเขียนเป็นสมการได้ดังนี้

$$T = K_t I \qquad (1)$$

T คือ แรงบิดของเพลา มีหน่วยเป็น นิวตัน-เมตร

K_t คือ ค่าคงที่

I คือ กระแส มีหน่วยเป็น แอมแปร์

$$E = K_e \omega \qquad (2)$$

E คือ แรงดันย้อนกลับ emf มีหน่วยเป็น โวลท์

ω คือ ความเร็วของมอเตอร์ มีหน่วยเป็น เรเดียน/วินาที

K_e คือ ค่าคงที่

โมเดลคณิตศาสตร์ของดีซีมอเตอร์ ระบบนี้จะให้ความสัมพันธ์ระหว่างกระแสอาเมเจอร์ แรงบิด และความเร็วอยู่ในลักษณะลิเนียร์ สมการทางไฟฟ้าของดีซีมอเตอร์แบบนี้เขียนได้เป็น

$$V = K_e \omega + L(dI/dt) + RI \qquad (3)$$

เมื่อ V คือ แรงดันที่ป้อนให้กับมอเตอร์

K_e คือ คือค่าคงที่ของแรงดันย้อนกลับ

L คือ อินดัคแตนซ์ของอาเมเจอร์

R คือ ความต้านทานที่ขั้วของอาเมเจอร์

สมการไดนามิคของมอเตอร์ คือ

$$T_g = J(d\omega/dt) + B\omega + T_f + T_L \qquad (4)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ T_g คือ แรงบิดที่กำเนิดโดยมอเตอร์

J คือ ผลรวมของโมเมนต์ของแรงเฉื่อยของมอเตอร์

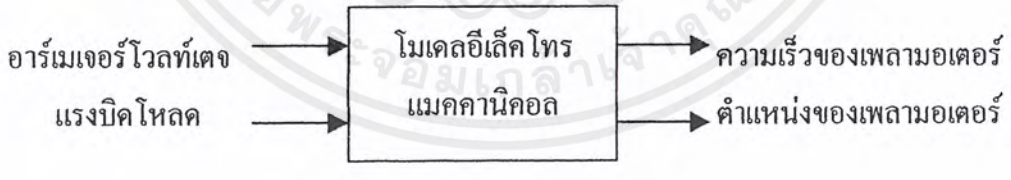
B คือ สัมประสิทธิ์ของวิสคอสแด้มพ์ปิ้ง

T_f คือ แรงบิดเสียดทานภายใน

T_L คือ แรงบิดโหลด

ข้อดีของมอเตอร์แบบฟิลด์แม่เหล็กถาวร ซึ่งเหนือกว่ามอเตอร์แบบมีโครงสร้างของฟิลด์ด้วยการพันของขดลวด คือไม่มีกำลังสูญเสียในฟิลด์มีประสิทธิภาพสูงกว่าและมีขนาดเล็กกว่าเมื่อเทียบกับมอเตอร์ที่มีขนาดเท่ากัน

ดีซีมอเตอร์ที่ใช้ร่วมกับดีซีแอมปริไฟทั้งในระบบการบังคับตำแหน่งและการบังคับความเร็วมักจะได้รับการประยุกต์ใช้ เป็นส่วนประกอบสร้างกำลังงานในระบบการนำร่องและระบบการบังคับต่างๆ และเนื่องจากวิทยาการเกี่ยวกับสารแม่เหล็กและการขยายตัวโซลิดสเตท ทำให้ดีซีมอเตอร์แบบแม่เหล็กถาวรได้รับความนิยมใช้ปีนส่วนประกอบการขับเคลื่อนในระบบการบังคับแบบลูปปิดต่างๆมากขึ้น การออกแบบและการชดเชยระบบดังกล่าวได้อย่างเหมาะสมจะต้องใช้โมเดลคณิตศาสตร์ของส่วนประกอบทั้งหมดในระบบ ในหัวนี้เราจะได้พัฒนาลิเนียร์โมเดลของดีซีมอเตอร์แบบแม่เหล็กถาวรและแบบฟิลด์แยกกระตุ้น



รูปที่ 2.1 แสดงบล็อกโมเดลอีเล็คโทรแมคคานิคอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมเดลอีเล็คโทรแมคคานิคอล

ส่วนสำคัญของดีซีมอเตอร์แบบฟิลด์แยกกระตุ้นมีโมเดล ดังแสดงในรูปที่ 2.2

R_a : ความต้านทานของอาเมเจอร์

L_a : อินดัคแตนซ์ของอาร์เมเจอร์

V_g : โวลท์เต็จกำหนดในอาร์เมเจอร์ (โวลท์เต็จย้อนกลับ)

R_f : ความต้านทานของฟิลด์

L_f : อินดัคแตนซ์ของฟิลด์

φ : ช่องว่างอากาศของเส้นแรงสนามแม่เหล็ก

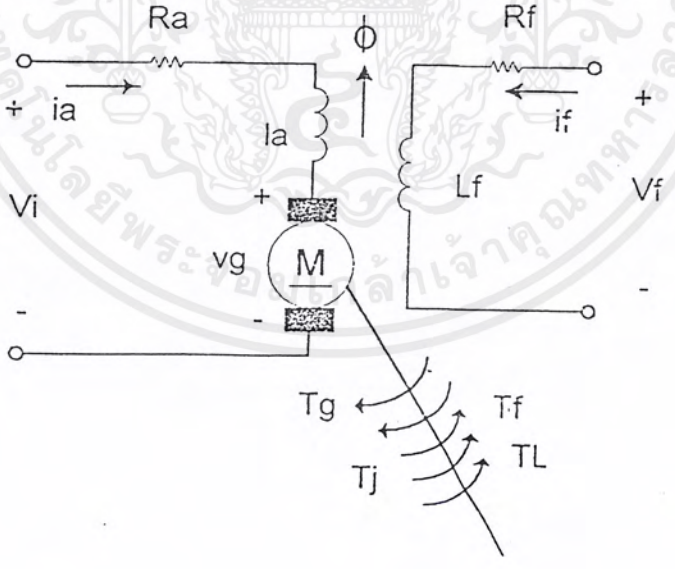
ω : ความเร็วของเพลาอาเมเจอร์

T_g : แรงบิดที่พัฒนาขึ้นในมอเตอร์

T_f : แรงบิดเสียดทานของมอเตอร์

T_j : แรงเฉื่อยของมอเตอร์

T_L : แรงบิดโหลดบนเพลาของมอเตอร์



รูปที่ 2.2 แสดงโมเดลของดีซีมอเตอร์แบบฟิลด์แยกกระตุ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นแรกเราจะหาสมการพื้นฐานโมเดลของดีซีมอเตอร์ได้จากลูปของอาร์เมเจอร์

$$V_I(t) = R_a I_a + L_a dI_a(t)/dt + V_g(t) \quad (5)$$

เทอมโวลท์เต็จ $V_g(t)$ ในสมการที่ 5 คือโวลท์เต็จย้อนกลับของมอเตอร์ซึ่งเกิดขึ้นเมื่อเส้นลวดตัวนำของอาร์เมเจอร์หมุนตัดเส้นแรงแม่เหล็ก ซึ่งเกิดขึ้นโดยกระแสของฟิลด์ (I_f) ตามกฎของฟาราเดย์ลูปของเส้นลวดตัวนำหมุนในฟิลด์แม่เหล็กคงที่จะมีการเหนี่ยวนำโวลท์เต็จขึ้นในขดลวดนั้น

$$V(t) = d\lambda(t) / dt \quad (6)$$

เมื่อ $\lambda(t)$ เส้นแรงแม่เหล็กที่รั่วไปยังขดลวดและ t คือ เวลาในการหมุนของคอมมิวเตเตอร์การควบคุมวงจรของแต่ละส่วนของตัวนำในโรเตอร์จะเกิดโวลท์เต็จขึ้นในอากาศและความเร็วเชิงมุม $\omega(t)$ หรือเราจะได้ว่า

$$V_g(t) = K \varphi(t)\omega(t) \quad (7)$$

สมมติให้กระแสของฟิลด์มีค่าคงที่ และไม่คิดถึงส่วนการเปลี่ยนแปลงในเส้นแรงฟิลด์เนื่องจากอาร์เมเจอร์รีแอคชั่นเส้นแรงฟิลด์ก็มีค่าคงที่ดังนั้นสมการที่ 7 ก็จะกลายเป็น

$$V_g(t) = K_e \omega(t) \quad (8)$$

เมื่อเราสมมติให้เส้นแรงของฟิลด์มีค่าคงที่ แรงบิดของแม่เหล็กไฟฟ้าซึ่งเกิดขึ้นแก่โรเตอร์ของมอเตอร์จะเป็นสัดส่วนกับกระแสอาร์เมเจอร์

$$T_g(t) = K_t I_a(t) \quad (9)$$

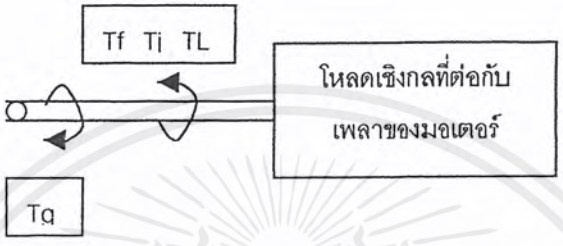
เมื่อ K_t คือ ค่าคงที่ของแรงบิดของมอเตอร์

กำลังงานเชิงกลที่เกิดขึ้นในโรเตอร์ คือผลคูณของแรงบิดที่เกิดขึ้นและความเร็วเชิงมุม

$$P_g = T_g(t)\omega(t) \quad (10)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โรเตอร์และอีสเตอรีซิส ดังแสดงในรูปที่ 2.3 แสดงให้เห็นถึงแรงบิดต่างๆที่เกิดขึ้นต่อโหลดมอเตอร์กำลังงานเชิงกลที่เกิดขึ้นในโรเตอร์ทั้งหมดนี้จะจ่ายไปยังโหลดที่ต่ออยู่กับเพลาของมอเตอร์ แต่กำลังงานนี้บางส่วนจะสูญเสียไปในมอเตอร์ การสูญเสียเนื่องมาจากแรงเสียดทาน หมายถึง ความหน่วงเนื่องจากลมที่มีต่อโรเตอร์ แรงเสียดทานตัวรองรับโรเตอร์ กระแสที่ไหลวนในเหล็กของมอเตอร์



รูปที่ 2.3 แสดงถึงแรงบิดต่างๆที่เกิดขึ้นต่อโหลดของมอเตอร์

$T_g(t)$: แรงบิดของมอเตอร์

$T_f(t)$: แรงบิดที่ต้องชนะการสูญเสียเนื่องจากการเสียดทาน

$T_j(t)$: แรงบิดเพื่อใช้เพิ่มอัตราเร่งแก่ ความเฉื่อยของโหลด

$T_L(t)$: แรงบิดโหลด

ในช่วงเวลาใดๆก็ตาม แรงบิดของมอเตอร์จะต้องเท่ากับและมีทิศทางตรงข้ามกับผลรวมของแรงบิด $T_g(t)$,$T_f(t)$ และ $T_L(t)$ ดังนั้น

$$T_g(t) = T_f(t) + T_L(t) + J\, d\omega(t)/dt \qquad (11)$$

เมื่อ J คือ ผลรวมของโมเมนต์แรงเฉื่อยของโรเตอร์และโหลดที่ต่ออยู่ที่เพลาของมอเตอร์ผลรวมของแรงบิดเสียดทานที่ประกอบกันขึ้นที่เพลาของมอเตอร์ ซึ่งเป็นลิเนียร์ฟังก์ชั่นกับความเร็วเชิงมุมของโรเตอร์เรียกว่า ส่วนประกอบของ วิสคอสฟริคชั่นและมักจะอยู่ในเทอมที่แยกออกจากฟริคชั่นอื่นๆซึ่งแสดงได้ด้วยสมการต่อไปนี้

$$T_g(t) = T_f(t) + T_L(t) + J\, d\omega(t)/dt + B\omega(t) \qquad (12)$$

เมื่อ B คือ สัมประสิทธิ์ของวิสคอสฟริคชั่นของมอเตอร์และโหลดที่ต่ออยู่กับเพลาของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$T_f(t)$ คือ ผลรวมของฟริคชั่นของโหลดและ ของมอเตอร์ทั้งหมด มีแรงต้านของลมและการสูญเสียกำลังในเหล็กของเพลามอเตอร์ ยกเว้นวิสคอสฟริคชั่น

สมการที่ (5), (7), (8) และ (12) เป็นชุดสมการพื้นฐานของดีซีมอเตอร์โมเดลและจากสมการเหล่านี้เราสามารถจะหาทรานเฟอร์ฟังชั่นของดีซีมอเตอร์ได้ โดยใส่ลาปลาสทรานฟอร์มทั้งสองข้างของชุดสมการพื้นฐานและเขียนได้ใหม่เป็น

$$V_i(S) - V_g(S) = (R_a + SL_a) I_a(S)$$

$$V_g(S) = K_e \omega(S)$$

$$T_g(S) = K_t I_a(S)$$

$$T_g(S) - T_f(S) - T_L(S) = (B + SJ) \omega(S)$$

2.2 เซ็นเซอร์และเอ็นโค้ดเดอร์ (Sensor and Encoder)

เซ็นเซอร์และเอ็นโค้ดเดอร์ เป็นส่วนประกอบสำคัญในระบบควบคุมที่มีการป้อนกลับในระบบควบคุมแบบลูปปิด ใช้สำหรับคุณสมบัติการทำงานของระบบเซ็นเซอร์และเอ็นโค้ดเดอร์ใช้เป็นตัวป้อนสัญญาณกลับเพื่อการควบคุม เซ็นเซอร์แบ่งออกเป็น 2 ลักษณะ คือ แบบอนาลอกเซ็นเซอร์ และแบบดิจิตอลเซ็นเซอร์ แบบอนาลอกได้แก่ ทาโคเยนเนอร์เรเตอร์ และซิงโครซ์ ส่วนเซ็นเซอร์แบบดิจิตอลได้แก่ เอ็นโค้ดเดอร์

2.2.1 ทาโคเยนเนอร์เรเตอร์ (Tacho Generator)

ทาโคเยนเนอร์เรเตอร์ เป็นเครื่องมือที่เราสามารถเปลี่ยนพลังงานกลไปเป็นพลังงานไฟฟ้าและให้กำเนิดเอาท์พุทโวลท์เตจ เป็นสัดส่วนกับแมกนิจูดของความเร็วเชิงมุม ในระบบการบังคับมอเตอร์ โดยทั่วไปจะใช้ทาโคมิเตอร์เป็นตัวแสดงความเร็วของเพลาของมอเตอร์หรือป้อนความเร็วกลับสำหรับการบังคับความเร็ว หรือสำหรับทำให้เสถียรภาพของระบบดีขึ้น

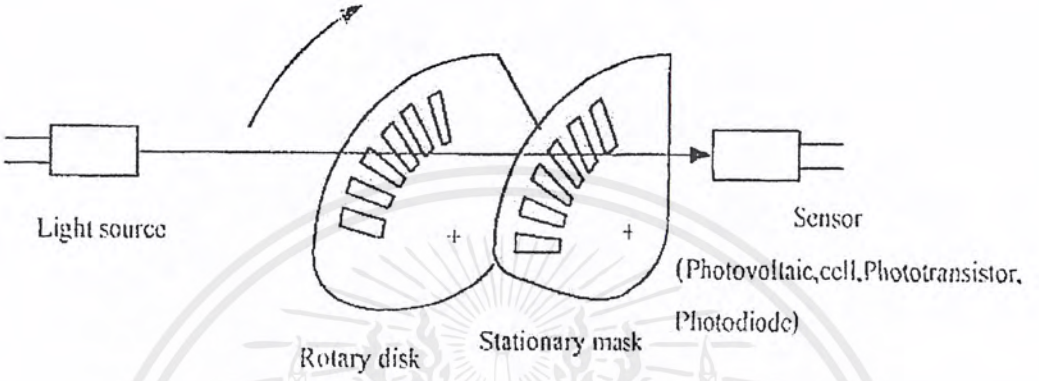
2.2.2 เอ็นโค้ดเดอร์ (Encoder)

ในระบบการบังคับตำแหน่งหรือความเร็วของเพลามอเตอร์ ต้องใช้เอ็นโค้ดเดอร์สำหรับรักษาตำแหน่ง และสร้างสัญญาณป้อนกลับโดยที่เอ็นโค้ดเดอร์จะสร้างสัญญาณพัลซ์ที่แปรผันตรงกับการหมุนของเพลา ซึ่งสามารถนำไปใช้การรับรู้ความเร็วและตำแหน่งเพลาของมอเตอร์ในรูปของอัตราจำนวนของพัลซ์ได้

เอ็นโค้ดเดอร์ประกอบด้วยส่วนที่สำคัญคือ ตัวกำเนิดแสง จานหมุน (rotary disk) จานที่อยู่กับที่ และบนแผ่นจานหมุนทำเป็นช่องโดยรอบดังแสดงในรูป ที่ใช้กันอยู่ทั่วไปมีด้วยกัน 2 ชนิดคือ แบบสมบูรณ์ในตัวเองและไม่สมบูรณ์ในตัวเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอ็นโค้ดเดอร์แบบสมบูรณ์ในตัวเองในรูปที่ 2.4 และบนแผ่นจานหมุนจะมีช่องสำหรับให้แสงผ่านข้ามไปยังตัวเซ็นเซอร์ ถ้าเป็นเอนโค้ดเดอร์ที่ใช้วัดความเร็วต่ำไม่ต้องมีแผ่นอยู่กับที่ก็ได้ ส่วนตัวกำเนิดแสงจะใสใช้หลอดไฟ หรือหลอดแอลซีดี



รูปที่ 2.4 ตัวอย่างกลไกของเอ็นโค้ดเดอร์

ความละเอียดของเอ็นโค้ดเดอร์ คือจำนวนคาบเวลาของสัญญาณเอาท์พุทต่อการหมุนของเพลา 1 รอบ ซึ่งบอกเป็นจำนวนพัลซ์ต่อรอบหรือ จำนวนไซเคิลต่อ 360 องศามุมทางเชิงกล หรือไซเคิลต่อองศา เอนโค้ดเดอร์ที่ใช้กันโดยทั่วไปมีความละเอียดตั้งแต่ 15 ถึง 10,000 พัลซ์ต่อรอบ

ในทางปฏิบัติเนื่องจากแสงที่ออกมาจากแหล่งกำเนิดเป็นลำแสงเดียวถ้าเราต้องการให้แสงที่ผ่านช่องไปยังเซ็นเซอร์เป็นเส้นตรงพร้อมๆกัน (collimation) ก็ทำได้โดยใช้เลนซ์หรือ พาราโบลิครีแฟลคเตอร์

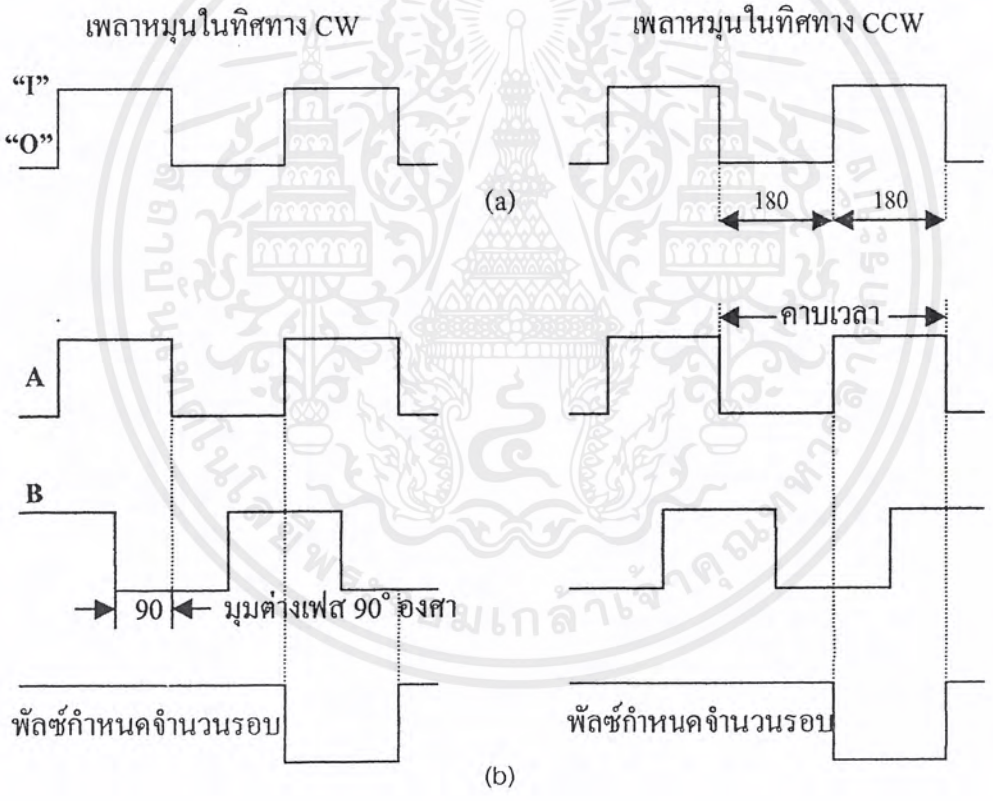
จำนวนพัลซ์ต่อ 1 รอบของสัญญาณที่เอนโค้ดเดอร์สร้างออกมาจะเท่ากับจำนวนช่องว่างบนแผ่นจานหมุน และความกว้างของช่องว่างกับความกว้างของแถบทึบระหว่างช่องว่างจะเท่ากัน เพราะฉะนั้นเราสามารถคำนวณหาความกว้างของช่องว่าง (ω) ได้จาก

$$\omega = \pi D/2N \qquad (13)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอาท์พุทของเอนโค้ดเดอร์ โดยทั่วๆไปแล้วสัญญาณเอาท์พุทที่ออกจากเอ็นโค้ดเดอร์โดยตรงจะมีระดับไม่เพียงพอในการควบคุมหรือสำหรับการประมวลผลสัญญาณ ดังนั้นจึงต้องมีวงจรขยายและเปลี่ยนแปลงรูปร่างลูกคลื่นสัญญาณต่อไว้ในเอ็นโค้ดเดอร์ด้วยเสมอสัญญาณที่ได้จากตัวเซ็นเซอร์ปกติแล้วจะมีรูปสัญญาณเป็นสามเหลี่ยม หรือรูปสัญญาณซายน์ขึ้นอยู่กับความละเอียดที่ต้องการ รูปสัญญาณเหล่านี้สามารถทำให้เป็นรูปสี่เหลี่ยมได้โดยการต่อคอมพาราเตอร์เข้ากับลิเนียร์แอมปลิไฟของเอ็นโค้ดเดอร์ก็จะได้เอาท์พุทเป็นรูปสี่เหลี่ยมที่ต้องการ

สัญญาณ 2 ชุดที่ได้จากเอ็นโค้ดเดอร์ชนิด 2 ช่องเฟสของสัญญาณ 2 ช่องนี้จะต่างกัน 90 องศาทางไฟฟ้า เราเรียกสัญญาณ 2 ช่องนี้ว่าเป็นควอดดราเจอร์ (quadrature) กันซึ่งเหมาะที่จะใช้ในการรับรู้ทิศทางการหมุนของเพลาหรือใช้ควบคุมระบบที่ซับซ้อนอื่นๆ



รูปที่ 2.5 (a) ตัวอย่างรูปคลื่นเอาท์พุทสี่เหลี่ยมของอุปกรณ์เอนโค้ดเดอร์ช่องเดียว (ไบไดเร็ก)
(b) ตัวอย่างสัญญาณเอนโค้ดเดอร์ 2 ช่องมีมุมเฟสต่างกัน 90° (สองทิศทาง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

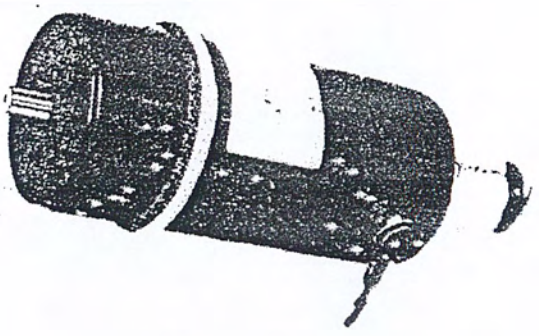
โดยที่ รูป (a) แสดงถึงลูกคลื่นเอาท์พุทสี่เหลี่ยมของเอนโค้ดเดอร์ชนิด 1 ช่องไม่ว่าเพลาจะหมุนในทิศทางใดก็ได้สัญญาณออกมาเหมือนกันจึงเหมาะที่จะใช้กับงานที่ไม่กำหนดทิศทางเท่านั้น ส่วนในรูป (b) แสดงสัญญาณ 2 ชุดที่ได้จากเอนโค้ดเดอร์ชนิด 2 ช่องเฟสของสัญญาณ 2 ช่องนี้จะต่างกัน 90 องศาทางไฟฟ้าเราเรียกสัญญาณ 2 ช่องนี้ว่าเป็น ควอดราเจอร์ (quadrature) กันซึ่งเหมาะจะใช้ในกรรับรู้ทิศทางการหมุนของเพลา หรือใช้ควบคุมระบบที่ซับซ้อนอื่นๆ จากสัญญาณ รูป(b) จะเห็นได้ว่าสัญญาณทั้ง 2 ช่องจะเริ่มจาก 0 ถึง 1 และ 1 ถึง 0 ขึ้นอยู่กับทิศทางการหมุนของแผ่นหมุนของเอนโค้ดเดอร์

ในเอนโค้ดเดอร์บางชนิดจะมีพัลซ์ที่แสดงถึงจำนวนรอบของการหมุนสำหรับใช้เป็นศูนย์ในการอ้างอิง พัลซ์ที่ใช้แสดงจำนวนรอบนี้จะเกิดขึ้น 1 พัลซ์ต่อ 1 รอบ โดยทั่วไปแล้วใช้บอกถึงตำแหน่งเชิงกลหรือใช้เป็นสัญญาณเคลียร์จำนวนที่นับไว้ในหน่วยเก็บข้อมูล

ต่อไปจะเป็นรูปตัวอย่างของดีซีเซอร์โวมอเตอร์ที่มีเซ็นเซอร์และเอ็นโค้ดเดอร์ ดังแสดงในรูปที่ 2.6,2.7 และรูปที่ 2.8

รูปที่ 2.6 เซอร์โวมอเตอร์ที่มีทาโคเยนเนอเรเตอร์ร่วมอยู่บนเพลาเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 ดีซีมอเตอร์ที่มีเอ็นโค้ดเดอร์และทาโคเยนเนอเรเตอร์

รูปที่ 2.8 ดีซีมอเตอร์ที่มีเกียร์บล็อค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

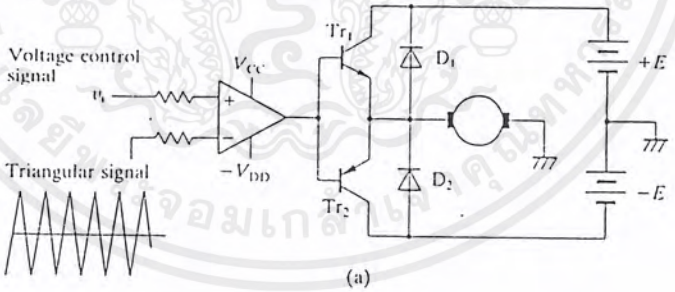
2.3 PWM Servo Amplifier

SERVO–AMPLIFIER คือ เป็นการเรียกวงจรอิเล็กทรอนิกส์ที่ดีไซน์ไว้ใช้สำหรับการขับ DC Servo - Motor ใช้ในการควบคุม ความเร็วหรือตำแหน่งมักถูกนำไปใช้ขับมากในงานอุตสาหกรรมหุ่นยนต์ และ อุปกรณ์ควบคุมเชิงเลข servo–amplifier มีแบ่งได้เป็น 2 ชนิดตามวิธีที่ใช้ขับอุปกรณ์โซลิดสเตท ชนิดแรกเรียกว่า เซอร์แอมปลิไฟร์ชนิดเชิงเส้น (Liner-Servo Amplifier) ซึ่งขับให้ทรานซิสเตอร์ทำงานอยู่ในย่านที่เป็นเชิงเส้น อีกชนิดหนึ่งเรียกว่า เซอร์โวแอมปลิไฟร์แบบพัลซ์วิดธ์มอดูเลชั่น (PWM Servo–Amplifier) ซึ่งจะขับทรานซิสเตอร์หรือมอสเฟตให้ทำงานแบบสวิตซ์ คือทำงาน (ON) และไม่ทำงาน (OFF) เหมือนกับใช้ในคอนเวอร์เตอร์ ซึ่งเซอร์แอมปลิไฟร์ทั้ง 2 แบบนี้ จะมีทั้งแบบ Voltage Control และแบบ Current Control ในที่นี้จะกล่าวแต่ PWM servo-amplifier ซึ่งมีข้อดีกว่าแบบ Linear คือ

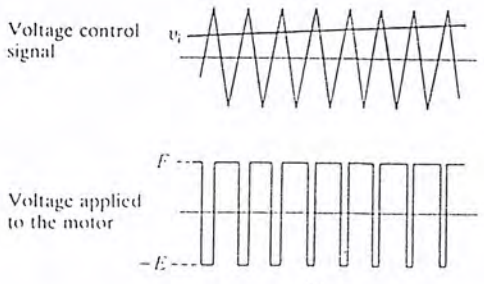
1.มีการสูญเสีย lossเนื่องจากความร้อน น้อยกว่า

2.มีประสิทธิภาพดีกว่า เพราะกำลังงานสูญเสียที่เกิดขึ้นน้อยมากจนแทบจะละเลยได้ แต่อาจต้องเอากำลังงสูญเสียมาคิดเมื่อความถี่ของการสวิตซ์มีค่าสูง

ระบบPWM ปกติจะใช้แหล่งจ่ายไฟกระแสตรง ซึ่งให้ซัพพลายโวลท์เตจให้ ON-OFF ที่ความถี่คงที่ และมีช่วงเวลาของการ ON ที่ปรับค่าได้ในหนึ่งคาบหรือที่เรียกว่าการปรับDuty Cycle ซึ่งเป็นการปรับค่าเฉลี่ยของโวลต์เตจที่จ่ายให้กับมอเตอร์ เมื่อโหลดเพิ่มขึ้นความถี่ของPWM Amplifier จะคงที่แต่เปลี่ยนค่า Duty Cycle มากขึ้น



(a)



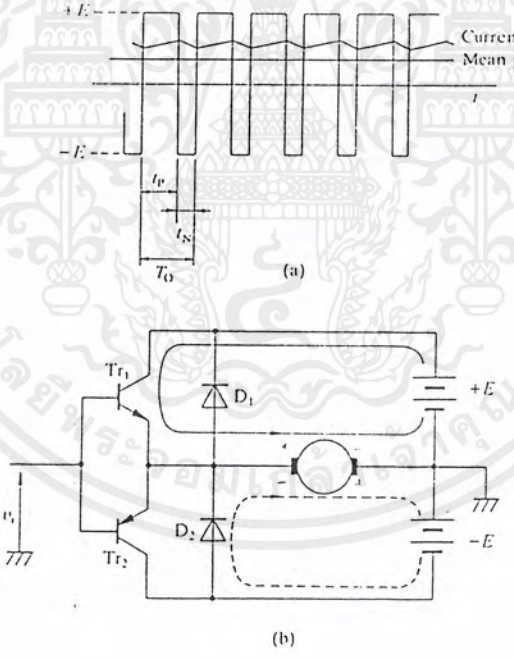
(b)

รูปที่ 2.9 เซอร์โวแอมปลิไฟร์ชนิด PWMแบบควบคุมแรงดันพื้นฐาน (a) วงจร (b) รูปคลื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 หลักการของเซอร์โวแอมปลิไฟร์ชนิด PWM แบบควบคุมแรงดัน

รูปที่ 2.9 แสดงถึงหลักการของเซอร์โวแอมปลิไฟร์ชนิด PWM แบบควบคุมแรงดัน ซึ่งคล้ายกับเซอร์โวแอมปลิไฟร์ชนิดเชิงเส้น ความแตกต่างของวงจรที่เห็นได้ชัดก็คือ วงจรมีไดโอดเรกติไฟร์สองตัวต่อขนานกับทรานซิสเตอร์แต่ละตัว หน้าที่ของไดโอดทั้งสองตัวคือ ป้องกันความเสียหายให้กับทรานซิสเตอร์ เนื่องจากแรงดันที่มากเกินขนาดจากการเปลี่ยนแปลงอย่างทันทันใดของกระแสมอเตอร์ ที่เพิ่มขึ้นมาในวงจรอีกตัวหนึ่งก็คือ ออปแอมป์ที่มาทำหน้าที่เป็นคอมพาราเตอร์ (ตัวเปรียบเทียบระดับแรงดันที่ขาอินพุททั้งสอง) ออปแอมป์ตัวนี้จะสร้างสัญญาณพัลส์วิดธ์มอดูเลต ให้กับทรานซิสเตอร์ คอมพาราเตอร์จะมีขาอินพุท 2 ขาสัญญาณควบคุมแรงดันป้อนเข้าที่ขาบวก และสัญญาณสามเหลี่ยมป้อนเข้าที่ขาลบ ความสัมพันธ์ระหว่างสัญญาณอินพุทและสัญญาณเอาท์พุทของคอมพาราเตอร์ ตลอดจนแรงดันที่มอเตอร์สรุปได้ดังนี้



รูปที่ 2.10 (a) รูปคลื่นและกระแสพัลส์วิดธ์มอดูเลชั่น
(b) ทางไหลของกระแสเมื่อแรงดันเฉลี่ยเป็นบวกกระแสไหลผ่าน Q1 เมื่อมันทำงาน และQ2 ไม่ทำงาน และจะไหลผ่าน D2 เมื่อ Q1 ไม่ทำงาน Q2 ทำงาน

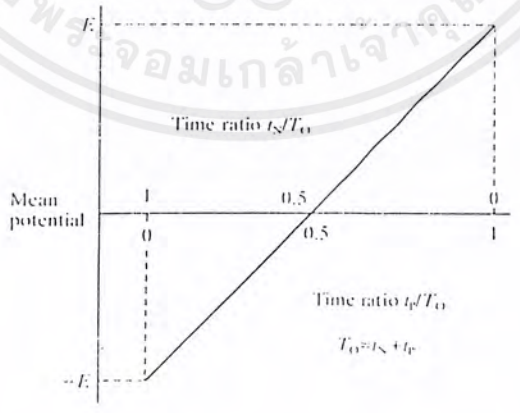
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(1) เมื่อ Vi มากกว่าสัญญาณรูปสามเหลี่ยมแรงดันเอาท์พุทของคอมพาราเตอร์จะเท่ากับ +Vcc ทรานซิสเตอร์ Q1จะทำงานเต็มที่ ส่วน Q2 จะไม่ทำงาน ดังนั้นแรงดันที่จ่ายให้กับมอเตอร์เท่ากับ E

(2) เมื่อ Vi น้อยกว่าสัญญาณรูปสามเหลี่ยมแรงดันเอาท์พุทของคอมพาราเตอร์จะเท่ากับ $-V_{DD}$ ทรานซิสเตอร์ Q2 ทำงานเต็มที่ ส่วน Q1 อยู่ในสภาวะไม่ทำงาน ดังนั้นแรงดันที่จ่ายให้มอเตอร์จะเท่ากับ –E

ความสัมพันธ์ระหว่างรูปคลื่นแรงดันคร่อมตัวมอเตอร์และค่าแรงดันเฉลี่ยแสดงให้เห็นดังรูปที่ 2.10 (a) จากรูปคาบเวลา t_p ที่ Q1 ทำงานจะนานกว่าคาบเวลา t_n ซึ่ง Q2 ทำงาน ทำให้ทิศทางการไหลของกระแสจากซ้ายไปขวา ดังรูปที่ 2.10 (b) เนื่องจากกระแสที่จ่ายโดยแหล่งจ่าย +E จึงกำหนดให้ทิศทางของกระแสเป็นทิศทางบวก ส่วนทางไหลของกรแสในคาบเวลา t_n แสดงด้วยเส้นประในรูปเดียวกัน กระแสไหลผ่าน D2 จ่ายพลังงานคืนให้แหล่งจ่าย –E สังเกตว่ากระแสช่วงนี้จะลดลงตามคาบเวลา เมื่อความถี่การสวิตช์สูงคาบเวลา t_p ต่อไปจะเกิดขึ้นก่อนที่กระแสจะลดลงถึงศูนย์ ทำให้ลดริปเปิ้ลของกระแสในมอเตอร์ลงได้อีกทางหนึ่ง ในเซอร์โวแอมป์ปลิไฟร์ชนิด PWM จะใช้ความถี่การสวิตช์สูงถึง 20 kHz หรือมากกว่า เพื่อป้องกันการสร้างสัญญาณรบกวนทางเสียงที่เป็นอันตราย

ในทางอุดมคติจะถือว่า ไม่มีความสูญเสียในการใช้วิธีพัลส์วิดธ์มอดูเลชั่นค่าแรงดันเฉลี่ยที่เอาท์พุทของเซอร์โวแอมปลิไฟร์ชนิด PWM จะเป็นฟังก์ชั่นเชิงเส้นของอัตราส่วนเวลา (t_p / T_o) ดังแสดงในรูปที่ 2.11



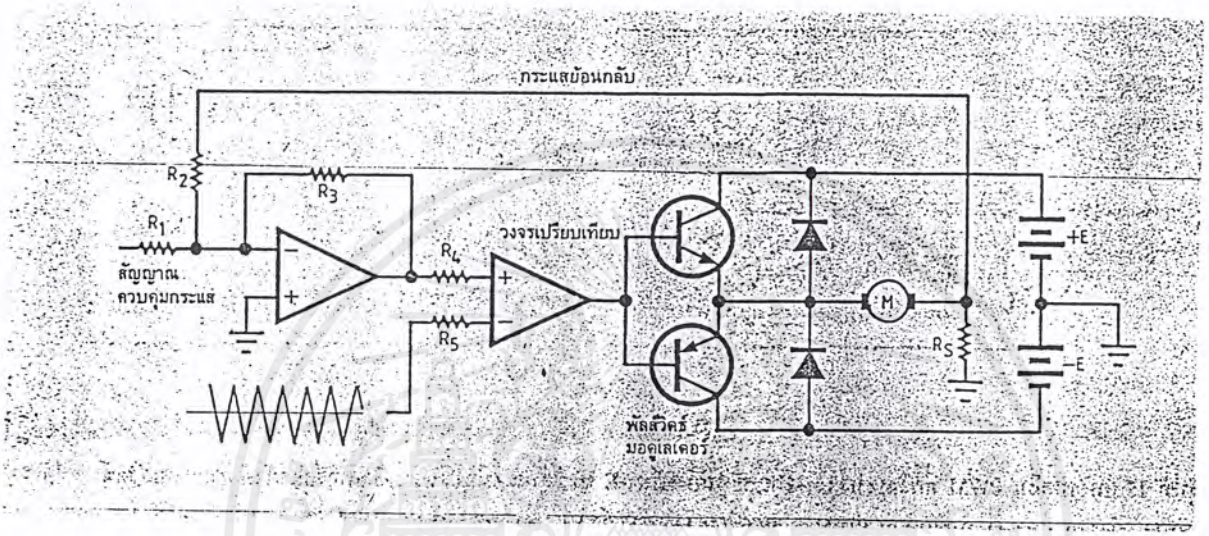
รูปที่ 2.11 แรงดันเอาท์พุทของเซอร์โวแอมปลิไฟร์ชนิด PWM
ในรูปที่ 2.10 เป็นฟังก์ชั่นของอัตราส่วนเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.3.2 เซอร์โวแอมปลิไฟร์ชนิด PWM แบบควบคุมกระแส

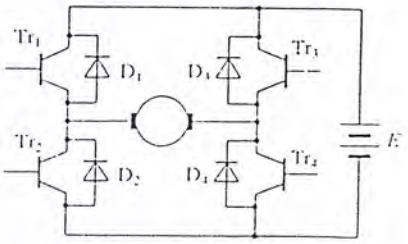
ในรูปที่ 2.12 เป็นเซอร์โวแอมปลิไฟร์ชนิด PWM แบบควบคุมกระแสออปแอมป์ที่อยู่หน้าคอมพาราเตอร์จะใช้เป็นตัวควบคุมกระแสของมอเตอร์ โดยใช้ตัวต้านทาน R_s ต่อเพื่อเป็นตัวตรวจจับและควบคุมกระแสให้กับมอเตอร์



รูปที่ 2.12 เซอร์โวแอมปลิไฟร์ชนิด PWM แบบความคุมกระแส

2.3.3 เซอร์โวแอมปลิไฟร์ชนิด PWM แบบบริดจ์

เซอร์โวแอมปลิไฟร์ที่ผ่านมาเป็นแบบใช้แหล่งจ่ายกำลังงานคู่บวกลบก็อาจจะลดแหล่งจ่ายกำลังงานลงให้เหลือเพียงแหล่งจ่ายเดียวได้โดยใช้วงจรบริดจ์รูปแบบของวงจรบริดจ์สามารถใช้ได้ทั้งเซอร์โวแอมปลิไฟร์ชนิดเชิงเส้น และชนิด PWM อย่างไรก็ตามจะเน้นการใช้กับ PWM เป็นสำคัญ ดังรูปที่ 2.13 เป็นวงจรเบื้องต้นของเซอร์โวแอมปลิไฟร์ชนิด PWM แบบบริดจ์ ใช้ทรานซิสเตอร์ทั้งหมดสี่ตัวควบคุมแรงดันที่จ่ายให้ดีซีมอเตอร์

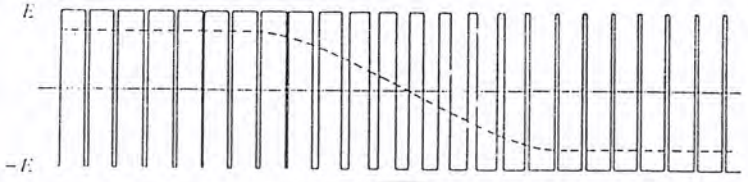


รูปที่ 2.13 วงจรพื้นฐานของเซอร์โวแอมปลิไฟร์ชนิด PWM แบบบริดจ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

42610

การใช้ทรานซิสเตอร์ทั้งสี่ตัวให้ทำงานมีอยู่ด้วยกัน สองวิธีในเบื้องต้นคือ วิธีแรกใช้ทรานซิสเตอร์ทุกตัวทำงานเป็นสวิตซ์ จะได้แรงดันตามรูปที่ 2.14



รูปที่ 2.14 รูปคลื่น PWM ที่สวิงระหว่าง +E และ –E เมื่อใช้ทรานซิสเตอร์ทั้งสี่ตัวทำงานแบบสวิตซ์

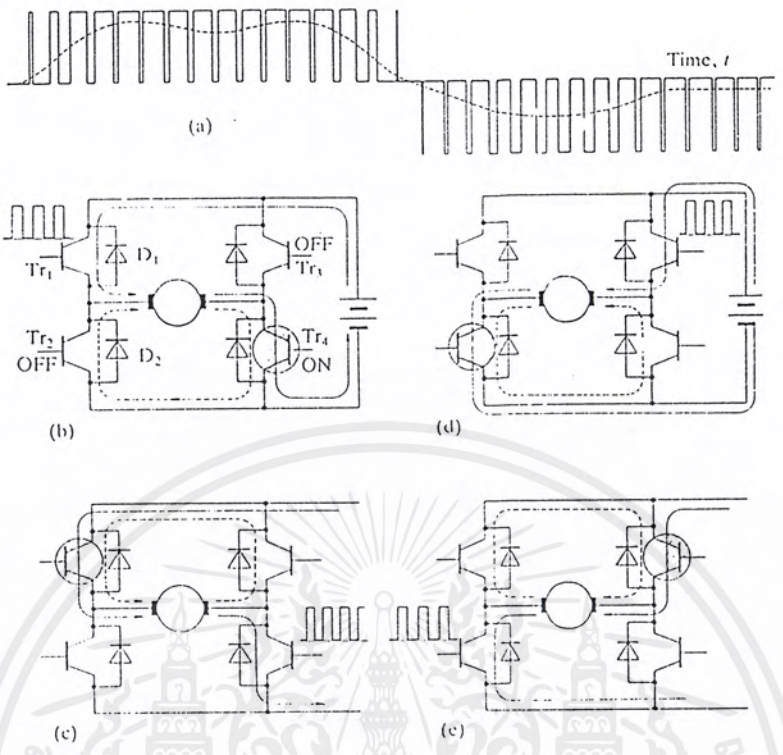
ซึ่งแรงดันชั่วขณะจะสวิงอยู่ระหว่าง +E ถึง –E และค่าแรงดันเฉลี่ยแสดงอยู่ในรูปเดียวกันตามเส้นประ ในวิธีนี้เมื่อ Q1 และ Q4 ทำงาน Q2 และ Q3 จะไม่ทำงาน แรงดัน +E จะคร่อมตัวมอเตอร์ ในทางกลับกันถ้าหาก Q1 และ Q4 ไม่ทำงาน แต่ Q2 และ Q3 ทำงาน แรงดัน –E จะคร่อมที่ตัวมอเตอร์ถ้าคาบเวลาบวก t_p เท่ากับคาบเวลาในช่วงลบ t_n ค่าแรงดันเฉลี่ยในช่วงนั้นจะเป็นศูนย์

อีกวิธีหนึ่งจะสร้างรูปคลื่นดังในรูปที่ 2.15 (a) การใช้ทรานซิสเตอร์ทั้งสี่ตัวให้ทำงานแสดงตามรูปที่ 2.15 (b) ถึง รูปที่ 2.15 (e) เมื่อแรงดันบวกคร่อมที่ตัวมอเตอร์ตามรูปที่ 2.15 (b) Q4 จะอยู่ในสภาวะทำงาน Q2 และ Q3 จะไม่ทำงาน ส่วน Q1 จะถูกขับด้วยสัญญาณ PWM เมื่อ Q1 ทำงานกระแสจะไหลตามเส้นทึบ และแรงดันจากแหล่งจ่ายจะจ่ายให้กับมอเตอร์และเมื่อ Q1 ไม่ทำงานมีเพียง Q4 เท่านั้นที่ทำงานกระแสในมอเตอร์จะวนเวียนในมอเตอร์, Q4 และ D2 เนื่องจากผลการเหนี่ยวนำของมอเตอร์

ในรูปที่ 2.15 (c) เป็นการขับอีกลักษณะหนึ่งใช้ทรานซิสเตอร์ Q1ทำงานตลอด ส่วน Q4 ถูกขับด้วยสัญญาณ PWM ส่วน รูปที่ 2.15 (d) และ 2.15 (e) เป็นกรณีที่แรงดันลบจ่ายให้มอเตอร์

เมื่อเปรียบเทียบข้อดีและข้อเสียของทั้ง 2 วิธี จะเห็นว่า วิธีแรกเปลี่ยนแรงดันบวกไปเป็นลบหรือจากลบไปเป็นบวกโดยผ่านจุดศูนย์เป็นไปอย่างราบเรียบกว่าซึ่งความราบเรียบนี้มีความสำคัญมากในเซอร์โวแอมปริไฟร์ ส่วนวิธีที่สองความไม่ราบเรียบเกิดจากคุณสมบัติในการสวิตซ์ ที่แตกต่างกันของทรานซิสเตอร์แต่ละตัว แต่วิธีที่สองจะลดค่าริบเปิ้ลลงครึ่งหนึ่งเมื่อเปรียบเทียบกับวิธีแรก เมื่อความถี่และกระแสเฉลี่ยเท่ากันในการใช้งานเป็นไปได้ว่าจะนำทั้งสองวิธีมารวมกัน เพื่อให้ได้รูปคลื่นใหม่ที่มีคุณสมบัติดีกว่า ดังรูปที่ 2.16 ซึ่งใช้ในวงจรอินเวอร์ชนิด PWM รุ่นใหม่ๆแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



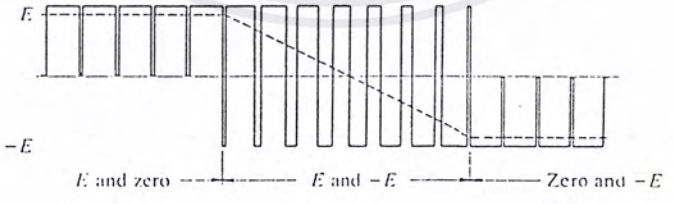
รูปที่ 2.15 (a) รูปคลื่น PWM ที่สวิงระหว่าง +E และศูนย์ หรือ –E และศูนย์ เมื่อใช้ทรานซิสเตอร์สองตัวทำงานแบบสวิตช์

(b) Q4 ทำงาน และ Q1 ถูกขับด้วย PWM

(c) Q1 ทำงาน และ Q4 ถูกขับด้วย PWM

(d) Q2 ทำงาน และ Q3 ถูกขับด้วย PWM

(e) Q3 ทำงาน และ Q2 ถูกขับด้วย PWM



รูปที่ 2.16 แสดงการรวมวิธีการในรูปที่ 2.14 และ 2.15 เข้าด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การเลือกใช้งานอุปกรณ์สวิตซ์ชิ่ง

สามารถเลือกใช้งานอุปกรณ์สวิตซ์ชิ่งในการขับมอเตอร์ได้หลายชนิด คือทั้ง เพาเวอร์ทรานซิสเตอร์ หรือเพาเวอร์มอสเฟต ในการสวิตช์ชิ่งก็ได้ ซึ่งจะกล่าวถึงโครงสร้าง และข้อดี ข้อเสียของอุปกรณ์ทั้งสอง ชนิด ดังต่อไปนี้

2.4.1 เพาเวอร์ทรานซิสเตอร์ (Power Transistor)

เพาเวอร์ทรานซิสเตอร์สามารถนำมาใช้เป็นอุปกรณ์ในการสวิตซ์ชิ่งได้ โดยนำมาต่อให้ทำงานในย่านอิ่มตัว(Saturation Region)โดยทำหน้าที่เหมือนสวิตซ์ปิด-เปิด แต่สัญญาณที่นำมาขับขาเบส (BASE) จะต้องมีซีกลบ เพื่อให้เพาเวอร์ทรานซิสเตอร์สามารถหยุดทำงานได้เร็วขึ้น และทำงานที่ความถี่สูงได้ และมีข้อดีอีกอย่างหนึ่งของเพาเวอร์ทรานซิสเตอร์คือ ไม่ต้องมีวงจรช่วยหยุดนำกระแสทำให้ควบคุมได้ไม่ยุ่งยาก เพียงแต่ควบคุมกระแสที่จ่ายแก่ขาเบสก็สามารถควบคุมการนำและหยุดนำกระแสของเพาเวอร์ทรานซิสเตอร์ได้ และนอกจากนี้ตัวมันยังมีฟรีวิลลิ่งไดโอดต่อขนานกับขาคอลเลคเตอร์ (Collector) และอีมิตเตอร์ (Emitter) เพื่อให้ทางเดินกระแสไหลอย่างต่อเนื่องเมื่อโหลดเป็นชนิดอินดั๊กทีฟโหลด (LOAD L)

2.4.2 เพาเวอร์มอสเฟต (Power Mosfet)

เพาเวอร์มอสเฟต (MOSFET Power Transitor) สามารถทำงานได้ดีที่ความถี่สูง ตั้งแต่ 50 กิโลเฮิรตซ์ ไปจนถึงประมาณ 200 หรือ 400 กิโลเฮิรตซ์ เนื่องจากมันใช้เวลาในการเปลี่ยนสถานะค่อนข้างสั้น และการพัฒนาในปัจจุบันมีแนวโน้มจะทำให้เพาเวอร์มอสเฟตทำงานได้ที่ความถี่สูงขึ้นไปอีก ซึ่งจะเป็นผลดีในการลดขนาดของคอนเวอร์เตอร์ เพาเวอร์มอสเฟตจึงเข้ามาแทนที่ไบโพลาร์เพาเวอร์ทรานซิลเตอร์อย่างรวดเร็วในปัจจุบัน

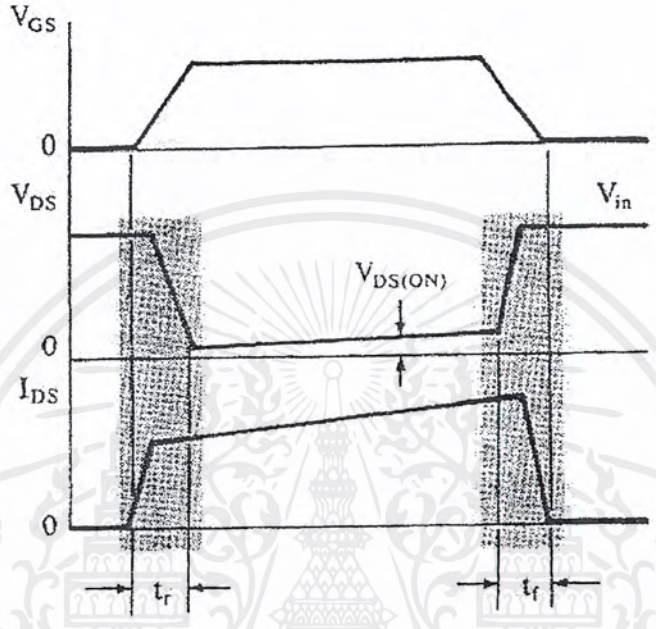
โครงสร้างของเพาเวอร์มอสเฟตจะเป็นไปได้ทั้งในลักษณะ N – channel และ P – channel อย่างไรก็ตาม ในบทนี้จะกล่าวถึงเฉพาะเพาเวอร์มอสเฟตชนิด N – channel เท่านั้น

เงื่อนไขของวงจรขับเพาเวอร์มอสเฟต

การขับเพาเวอร์มอสเฟตให้นั้นกระแสนั้นแตกต่างจากการขับกระแสไบแอสในไบโพลาร์เพาเวอร์ทรานซิลเตอร์ เนื่องจากมีเงื่อนไขการไบแอสที่แตกต่างกัน ที่ซึ่งสำหรับไบโพลาร์เพาเวอร์ทรานซิลเตอร์กระแสจะไหลผ่านคอลเล็กเตอร์และอีมิตเตอร์ได้ก็ต่อเมื่อมีกระแสไบแอสไหลผ่านที่เบสและอีมิตเตอร์แต่เพาเวอร์มอสเฟตจะมีกระแสไหลผ่านเดรนและซอร์สได้ก็ต่อเมื่อแรงดันตกคร่อมที่ขาเกตและซอร์สมีค่าอย่างต่ำเท่ากับค่าแรงดันขีดเริ่ม($V_{GH(th)}$; Threshold Voltage)ของมัน แต่ใช้กระแสต่ำ การขับเพาเวอร์มอสเฟตให้นำกระแสจึงทำได้ง่าย และยุ่งยากน้อยกว่าไบโพลาร์เพาเวอร์ทรานซิลเตอร์มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างไรก็ตาม เพื่อให้เพาเวอร์มอสเฟตทำงานได้อย่างมีประสิทธิภาพสูงสุด จึงจำเป็นต้องศึกษาเงื่อนไขต่างๆ สำหรับการบังคับให้เพาเวอร์มอสเฟตนำกระแส เป็นอันดับแรกเสียก่อน



รูปที่ 2.17 แสดงลักษณะของกระแสและแรงดันตกคร่อมมอสเฟตเมื่อนำกระแสและหยุดนำกระแส ซึ่งจะเกิดกำลังงานสูญเสียความร้อน

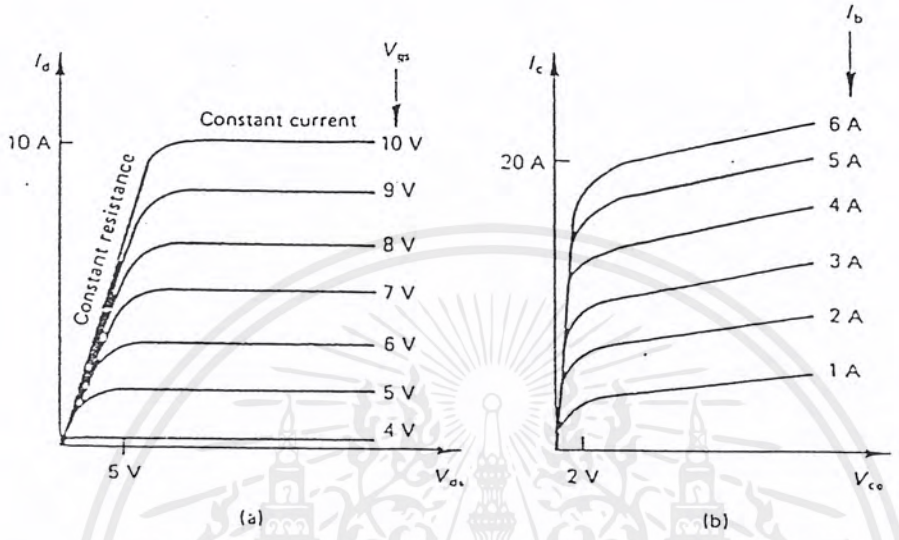
2.4.3 ข้อดีของมอสเฟตเมื่อเทียบกับทรานซิสเตอร์

- ควบคุมด้วยแรงดัน เมื่อพิจารณาถึงการนำไปใช้งานสวิตซ์ชิ่งของมอสเฟทกับทรานซิสเตอร์แล้ว สิ่งสำคัญที่จะต้องคำนึงถึงคือ ลักษณะสมบัติทางไฟฟ้าและลักษณะสมบัติทางเอาท์พุท จากรูปที่ 2.18 เป็นลักษณะสมบัติทางเอาท์พุทของทรานซิสเตอร์และมอสเฟท จะเห็นได้ว่าทรานซิสเตอร์เป็นอุปกรณ์ที่ควบคุมด้วยกระแส โดยการป้อนกระแสที่ขาเบสก็จะทำให้เกิดการไหลของกระแสคอลเลคเตอร์ได้

ส่วนมอสเฟทนั้นเป็นอุปกรณ์ที่ควบคุมด้วยแรงดัน สำหรับในการขับมอสเฟทชนิด N-channel enhancement ให้ทำงานได้นั้น จะต้องป้อนแรงดันบวกระหว่างเกทกับซอร์สจากการที่ทรานซิสเตอร์เป็นอุปกรณ์ที่ควบคุมด้วยกระแส ทำให้ต้องใช้วงจรขับที่สามารถจ่ายกระแสเบสได้อย่างเพียงพอ เพื่อให้ทรานซิสเตอร์ทำงานได้อย่างอิ่มตัว ในขณะที่มอสเฟทนั้นใช้วงจรขับที่จ่ายแรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดันในการควบคุมกระแสเดรน ดังนั้นในการออกแบบวงจรของมอสเฟทจึงทำได้ง่ายกว่าใช้กำลังไฟฟ้าที่น้อยกว่า ในอุปกรณ์ในการสร้างน้อยกว่า เมื่อเทียบกับวงจรขับของทรานซิสเตอร์

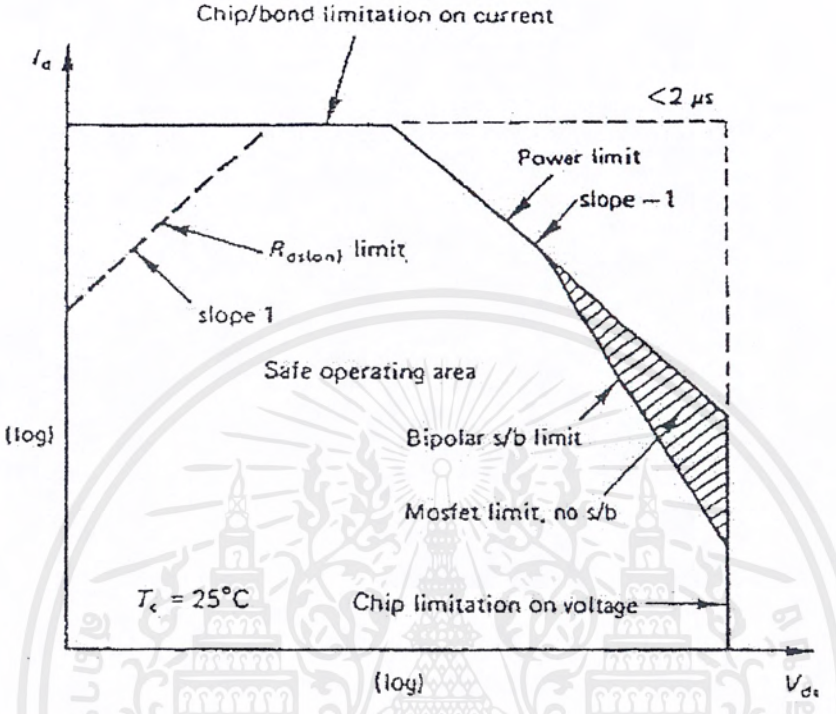


รูปที่ 2.18 (a) ลักษณะสมบัติทางเอาท์พุทของมอสเฟต
(b) ลักษณะสมบัติทางเอาท์พุทของทรานซิสเตอร์

- ความเร็วของการทำงานแบบสวิตชิ่งสูง มอสเฟทมีความเร็วในการทำงานแบบสวิตชิ่งสูงกว่าทรานซิสเตอร์เพราะเนื่องจากมอสเฟทเป็นอุปกรณ์ประเภท Unipolar ทำให้ไม่เกิดค่าเวลา Storage time เหมือนกับทรานซิสเตอร์ ซึ่งค่าเวลาดังกล่าวมีผลต่อความเร็วในสวิตชิ่งของทรานซิสเตอร์เป็นอย่างมาก

- ไม่เกิดปรากฏการณ์ Second breakdown พื้นที่ในการทำงานที่ปลอดภัยของมอสเฟทจะมากกว่าทรานซิสเตอร์ พื้นที่ในการทำงานของทรานซิสเตอร์นั้นจะถูกจำกัดด้วยปรากฏการณ์ Second breakdown ปรากฏการณ์ดังกล่าวเกิดขึ้นเมื่อเนื่องจากความหนาแน่นของกระแสที่มากจนเกินไป และทำให้เกิดความร้อนจนกระทั่งเกิดการ Breakdown ได้แต่มอสเฟทจะไม่เกิดปรากฏการณ์ดังกล่าว เนื่องจากมีความหนาแน่นของกระแสมากจนทำให้เกิดความร้อนแล้วจะทำให้เกิดการเคลื่อนที่ของพาหะที่ Channel นั้นลดลงเป็นผลให้กระแสที่ไหลนั้นลดลงด้วยจึงไม่เกิดปรากฏการณ์ Second breakdown

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 แสดงพื้นที่ปลอดภัยของมอสเฟตและทรานซิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การใช้งานไมโครคอนโทรลเลอร์

คำนำ

ในบทนี้จะกล่าวถึงคุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 ซึ่งเป็นที่นิยมใช้กันอย่างแพร่หลายเป็นอย่างมาก และการประยุกต์การใช้งานไมโครคอนโทรลเลอร์เชื่อมต่อกับอุปกรณ์ต่างๆ แล้วแต่วัตถุประสงค์ในการใช้งาน เช่น ส่วนของการรับค่าและการแสดงผล เพื่อเป็นแนวทางต่อไป

3.1 คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ตระกูล MCS-51

1. เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต
2. มีวงจรออสซิลเลเตอร์ และวงจรผลิตสัญญาณนาฬิกาภายในไอซี
3. มีขาสัญญาณเชื่อมต่อหน่วยความจำข้อมูลภายนอก (external data memory) โดยอ้างตำแหน่งแอดเดรสได้ถึง 64k
4. สามารถเชื่อมต่อหน่วยความจำโปรแกรมภายนอก (external program memory) โดยอ้างตำแหน่งแอดเดรสได้สูง 64k
5. มีหน่วยความจำโปรแกรมภายในตัว (on-chip program memory) ขนาด 4k โดยเฉพาะเบอร์ 8052 จะมีส่วนความจำในส่วนนี้ถึง 8k สำหรับเบอร์ 8031 และ8032 จะไม่มีหน่วยความจำในส่วนนี้
6. มีหน่วยความจำข้อมูลภายในตัว (on-chip data memory) ขนาด 128 ไบต์ โดยเฉพาะเบอร์ 8032 และ 8052 จะมีหน่วยความจำในส่วนนี้ถึง 256 ไบต์
7. หน่วยความจำข้อมูลภายในบางส่วนสามารถเข้าถึงข้อมูลระดับบิตได้ด้วย ทำให้การควบคุมหรือการตรวจสอบสถานะบิตทำได้ง่าย ส่งผลให้การเขียนโปรแกรมทำได้ง่ายมากขึ้น
8. มีไทม์เมอร์/เคาน์เตอร์ (timer/counter) ขนาด 16 บิต จำนวน 2ตัว โดยเฉพาะเบอร์ 8032 หรือ 8052 จะมีไทม์เมอร์/เคาน์เตอร์ จำนวน 3ตัว
9. การอินเตอร์รัปต์สามารถทำได้จาก 5 แหล่งกำเนิด โดยเฉพาะเบอร์ 8032 และ 8052 จะทำอินเตอร์รัปต์ได้ 6 แหล่งกำเนิด โดยการอินเตอร์รัปต์ยังสามารถจัดระดับความสำคัญได้เป็น 2 ระดับ
10. มีพอร์ทการสื่อสารอนุกรมภายในตัวเอง ซึ่งการทำงานเป็นแบบฟูลดูเพล็กซ์ (full duplex)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. มีคำสั่งคำนวณทางคณิตศาสตร์และทางตรรกศาสตร์
12. คำสั่งส่วนใหญ่ใช้เวลาการทำงานเพียง 1 ไมโครวินาที เมื่อใช้คริสตอลความถี่ 12 เมกะเฮิรตซ์
13. ต้องการแหล่งจ่ายไฟ 5 โวลท์เพียงชุดเดียว

เบอร์	หน่วยความจำภายใน		ตัวจับเวลา/ตัวนับ จำนวน	อินเทอร์รัพท์
	โปรแกรม	ข้อมูล		
8052 AH	8K×8 ROM	256×8 RAM	3×16 BIT	6
8051 AH	4K×8ROM	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 3.1 แสดงคุณสมบัติของไมโครคอนโทรลเลอร์แต่ละเบอร์ตระกูล MCS-51

3.1.1 โครงสร้างภายนอกของ MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทุกเบอร์จะมีตำแหน่งขาพื้นฐานที่เหมือนกัน ดังแสดงในรูปที่ 3.1 สำหรับหน้าที่ การใช้งานของแต่ละขา มีดังนี้

- ขา V_{CC} เป็นขาป้อนแรงดันไฟเลี้ยง +5 โวลท์
- ขา V_{SS} เป็นขา กราวนด์
- ขาพอร์ต 0 (Port 0) มี 8 ขา ได้แก่ P0.0-P0.7 แต่ละขาสามารถกำหนดให้เป็นได้ทั้งขาพอร์ตอินพุท และเอาท์พุท แบบ 2ทิศทาง สำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุทพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ตเพื่อกำหนดให้ขาพอร์ตเหล่านั้นอยู่ในสถานะปล่อยลอย (float) ซึ่งในสถานะนี้เองที่สามารถนำมาใช้เป็นพอร์ตอินพุทอิมพีแดนซ์สูงได้ นอกจากพอร์ตนี้จะใช้งานเป็นพอร์ตอินพุทเอาท์พุทแล้วมันยังถูกใช้งานในการติดต่อกับหน่วยความจำภายนอกด้วย โดยทำหน้าที่ในการกำหนดตำแหน่งแอดเดรสไบต์ต่ำ (A0-A7) ซึ่งจะใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานเป็นแบบมัลติเพล็กซ์กับการรับส่งข้อมูลขนาด 8 บิต (D0-D7)

- ขาพอร์ต 1 (Port 1) มี 8 ขาได้แก่ขา P1.0-P1.7 แต่ละขาสามารถกำหนดให้เป็นได้ทั้งขาพอร์ตอินพุท และเอาท์พุทแบบ 2 ทิศทาง สำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุทพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้เป็นอินพุท นอกจากนี้สำหรับเบอร์ 8032 และ 8052 ขาพอร์ต P1.0 และ P1.1 จะถูกนำมาใช้งานเป็นขา T2 และ T2EX ตามลำดับด้วย
- ขาพอร์ต 2 (Port 2) มี 8 ขา ได้แก่ขา P2.0-P2.7 แต่ละขาสามารถกำหนดให้เป็นได้ทั้งขาพอร์ตอินพุท และเอาท์พุทแบบ 2 ทิศทางสำหรับใช้งานทั่วไป โดยใช้งานเป็นอินพุทพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้เป็นพอร์ตอินพุท นอกจากพอร์ตนี้จะใช้งานเป็นพอร์ตอินพุท และเอาท์พุทแล้วมันยังถูกใช้งานในการติดต่อกับหน่วยความจำภายนอกด้วย โดยทำหน้าที่ในการกำหนดตำแหน่งแอดเดรสไบต์สูง (A8-A15)
- ขาพอร์ต 3 (Port 3) มี 8 ขา ได้แก่ขา P3.0-P3.7 แต่ละขาสามารถกำหนดให้เป็นได้ทั้งขาพอร์ตอินพุท และเอาท์พุทแบบ 2 ทิศทางสำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุทพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้เป็นพอร์ตอินพุท นอกจากพอร์ตนี้จะใช้งานเป็นพอร์ทอินพุทและเอาท์พุทแล้วมันยังถูกใช้งานในหน้าที่พิเศษต่าง ๆ ดังแสดงในตารางที่ 3.2
- ขารีเซต (RST) ใช้สำหรับการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยการรีเซตต้องคงสถานะเป็น 1 อย่างน้อยนาน 2 แมชีนไซเกิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างปกติ
- ขา ALE/PROG เป็นขาสัญญาณเพื่อทำหน้าที่ควบคุมการแลทซ์ค่าตำแหน่งแอดเดรสไบต์ต่ำ (Address Latch Enable หรือ พอร์ต 0) เมื่อต้องการติดต่อกับหน่วยความจำภายนอก นอกจากนี้ขานี้ยังทำหน้าที่เป็นอินพุทรับพัลส์ในการโปรแกรม (Program pulse input) ในส่วนของหน่วยความจำ EPROM สำหรับไมโครคอนโทรลเลอร์ในตระกูล MCS-51 ที่มีหน่วยความจำโปรแกรมภายในเป็น EPROM
- ขา PSEN (Program Store Enable) ทำหน้าในการส่งสัญญาณร้องขอเพื่ออ่านคำสั่งจากหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ประมวลผลคำสั่งจากหน่วยความจำภายนอก ขานี้จะส่งสัญญาณสโตรบออกมาจำนวน 2 ครั้งในแต่ละแมชีนไซเกิล แต่ในขณะที่ติดต่อกับหน่วยความจำข้อมูลภายนอกจะไม่มีการส่งสัญญาณสโตรบแต่อย่างใด
- ขา EA/V_{PP} (Extermal Access enable/V_{PP}) เป็นขาสำหรับการเลือกใช้หน่วยความจำโปรแกรมจากภายในหรือจากภายนอก โดยถ้ามีสถานะเป็น 0 จะหมายถึงให้ไมโคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนโทรลเลอร์รับคำสั่งจากหน่วยความจำภายนอกที่ตำแหน่งแอดเดรส 0-0FFFH (0-1FFFH ถ้าเป็นเบอร์8052) อย่างไรก็ตาม ถ้าบิตป้องกัน (security bit) ในหน่วยความจำ EPROM ถูกโปรแกรมไว้ ไมโครคอนโทรลเลอร์จะไม่รับคำสั่งจากหน่วยความจำภายนอกเลย นอกจากนี้ขานี้ยังทำหน้าที่รับแรงดันไฟสำหรับการโปรแกรม (V_{pp}) ขนาด 21 โวลต์ เพื่อใช้ในระหว่างการโปรแกรม EPROM

- ขา $XTAL_1$ และขา $XTAL_2$ เป็นขาอินพุทและเอาท์พุทของวงจรอินเวอร์ติ้งออสซิลเลเตอร์แอมปลิไฟเออร์ (inverting oscillator amplifier) สำหรับใช้ต่อร่วมกับคริสตอลภายนอก

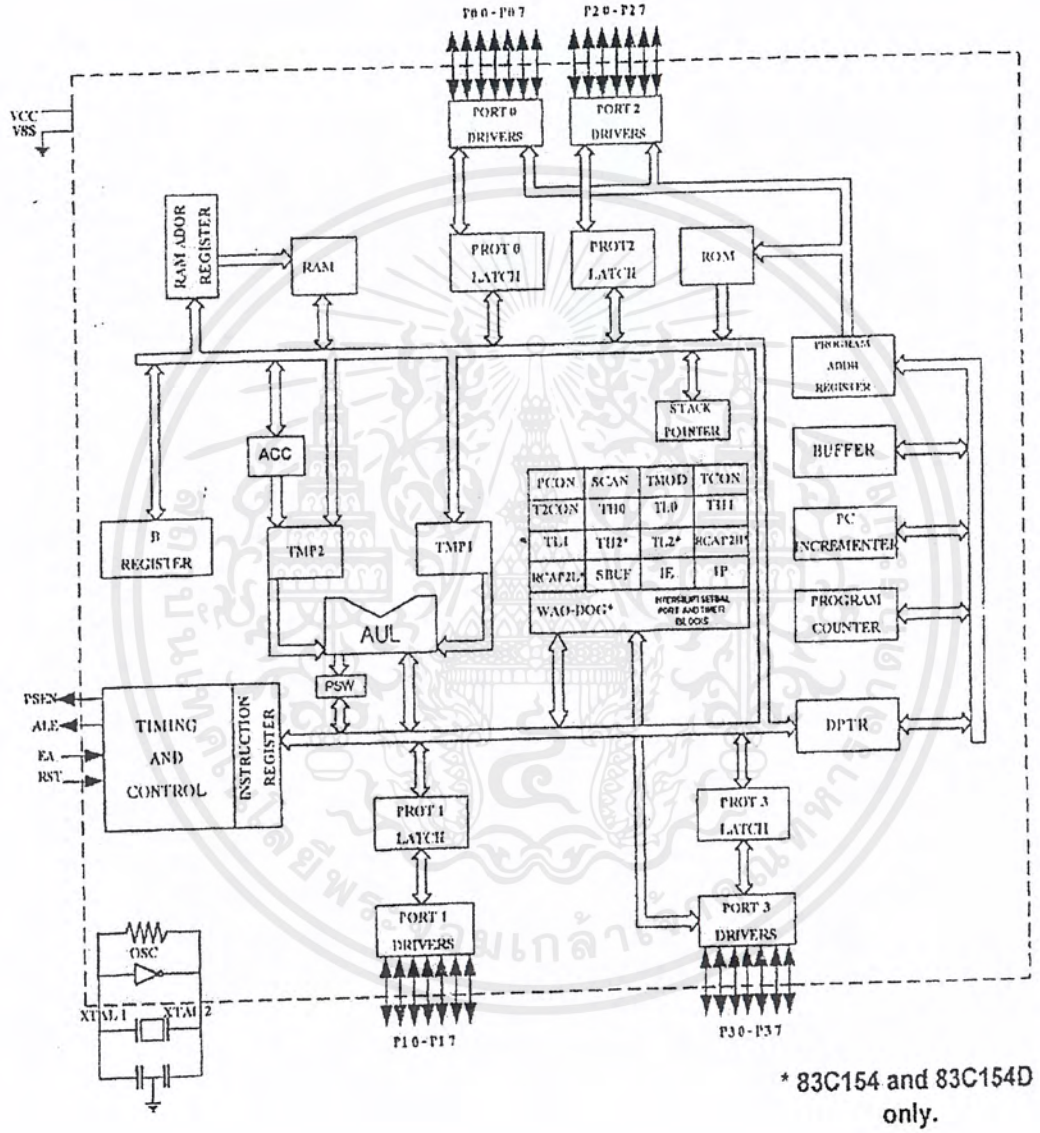
ขาพอร์ท	หน้าที่พิเศษ
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory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

ตารางที่ 3.2 แสดงหน้าที่พิเศษของแต่ละขาของพอร์ท P3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 โครงสร้างภายในของ MCS – 51

โครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51 ดังแสดงในรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมของ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การเชื่อมโยง 8253 กับ MCS-51

3.2.1 โครงสร้างของ 8253 ประกอบไปด้วย

- วงจรนับถอยหลัง 3 ตัว (#0,#1,#2) แบบ 16 บิต
- รีจิสเตอร์ควบคุม ใช้ควบคุมการทำงาน
- ลอจิกควบคุมการเขียนอ่าน

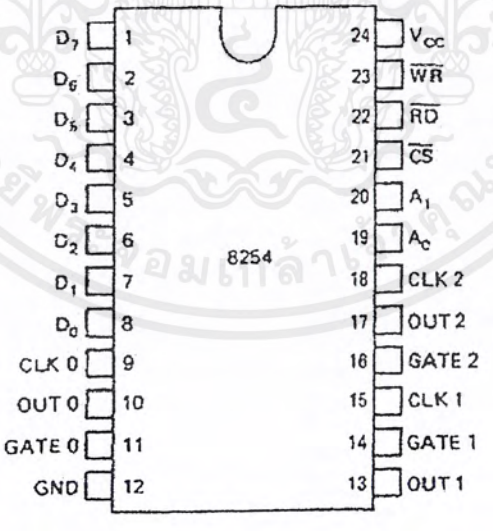
โหมดการทำงาน 8253 มีโหมดการทำงานถึง 6 แบบ (โหมด 0 – 5) ซึ่งทั้ง 6 โหมด นี้จำเป็นต้องโหลดค่านับเริ่มต้นให้ก่อนแล้วจะมีการนับถอยหลัง จนครบจะมีเอาท์พุทออก

การทำงานของวงจรนับ วงจรนับ (Counter) แต่ละตัว จะมีขา CLK , GATE ซึ่งเป็นขาอินพุท และเอาท์พุท ทั้ง 3 ขามี การทำงานดังนี้

CLK (Clock Input) คล้อคอินพุทที่ป้อนให้วงจรนับแบบ 16 บิต ความถี่สูงสุด 2 MHz (ถ้าเป็น CMOS จะใช้กับความถี่ได้สูงถึง 10 MHz)

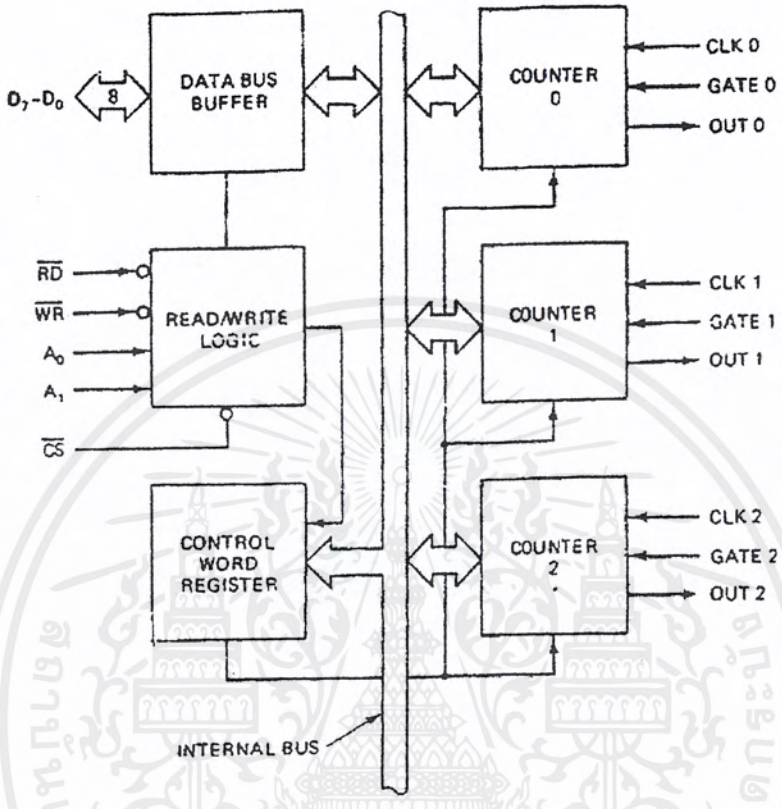
GATE เป็นขาอินพุทที่ใช้ควบคุมสัญญาณคล้อคที่ป้อนให้วงจรนับใช้เริ่ม การทำงานของวงจรนับ (รายละเอียดดูตารางที่ 3.1)

OUT (เอาท์พุท) เป็นเอาท์พุทวงจรนับ



รูปที่ 3.2 การวางขาของ 8253

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 บล็อกไดอะแกรมแสดงผังภายในของ 8253

จากบล็อกไดอะแกรมรูปที่ 3.3 ข้อมูลต่างๆที่จะติดต่อกับ 8253 จะกระทำผ่านบัสข้อมูลซึ่งมีบัฟเฟอร์อยู่ภายใน (Data bus buffer) บล็อกถัดมามาจะเป็นลอจิกที่ใช้ในการควบคุมการเขียนหรือการอ่าน รีจิสเตอร์ภายในของ8253 ให้สอดคล้องกับสัญญาณควบคุมจากไมโครคอนโทรลเลอร์ บล็อกถัดมาเป็นรีจิสเตอร์คำสั่งควบคุม (Control Word Register) ใช้สำหรับเก็บข้อมูลที่ควบคุมจากรีจิสเตอร์ควบคุมทั้งสิ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 รีจิสเตอร์ควบคุม (CONTROL WORD REGISTER)

ในรีจิสเตอร์ที่ใช้ควบการทำงาน ของ 8253 ตามคำสั่งที่จะส่งมาควบคุม เราสามารถที่จะติดต่อกับรีจิสเตอร์ตัวนี้ได้เมื่อให้เข้าขา A0,A1 เป็น 1 ทั้ง 2 ขารายละเอียด ดังตารางที่ 3.3

$\overline{CS}$	RD	WR	A1	A0	Write into Counter 0
0	1	0	0	0	Write into Counter 1
0	1	0	1	0	Write into Counter 0
0	1	0	1	0	Write into Counter 2
0	1	0	1	1	Write Control Word
0	0	1	0	0	Read from Counter 0
0	0	1	1	0	Read from Counter 1
0	0	1	1	0	Read from Counter 2
0	0	1	1	1	No-Operation (3 – State)
1	X	X	X	X	No-Operation (3 – State)
0	1	1	X	X	No-Operation (3 – State)

ตารางที่ 3.3 แสดงค่าความจริงของ 8253

การกำหนดค่าคำสั่งควบคุม

การจะใช้งานเคาน์เตอร์ได้นั้นจะต้องกำหนดสิ่งต่างๆ ดังนี้คือ เคาน์เตอร์ที่จะใช้ รูปแบบของการนับโหมดที่จะใช้งานซึ่งการกำหนดค่าเหล่านี้จะทำโดยโปรแกรมเข้าไปให้รีจิสเตอร์คำสั่งควบคุมจะมีรูปแบบดังนี้

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

3.2.3 โหมดและการทำงานของ 8253 ประกอบด้วย

โหมด 0 อินเตอร์รัพท์จากการนับ

โหมด 1 โมโนสเตเบิ้ลที่โปรแกรมได้

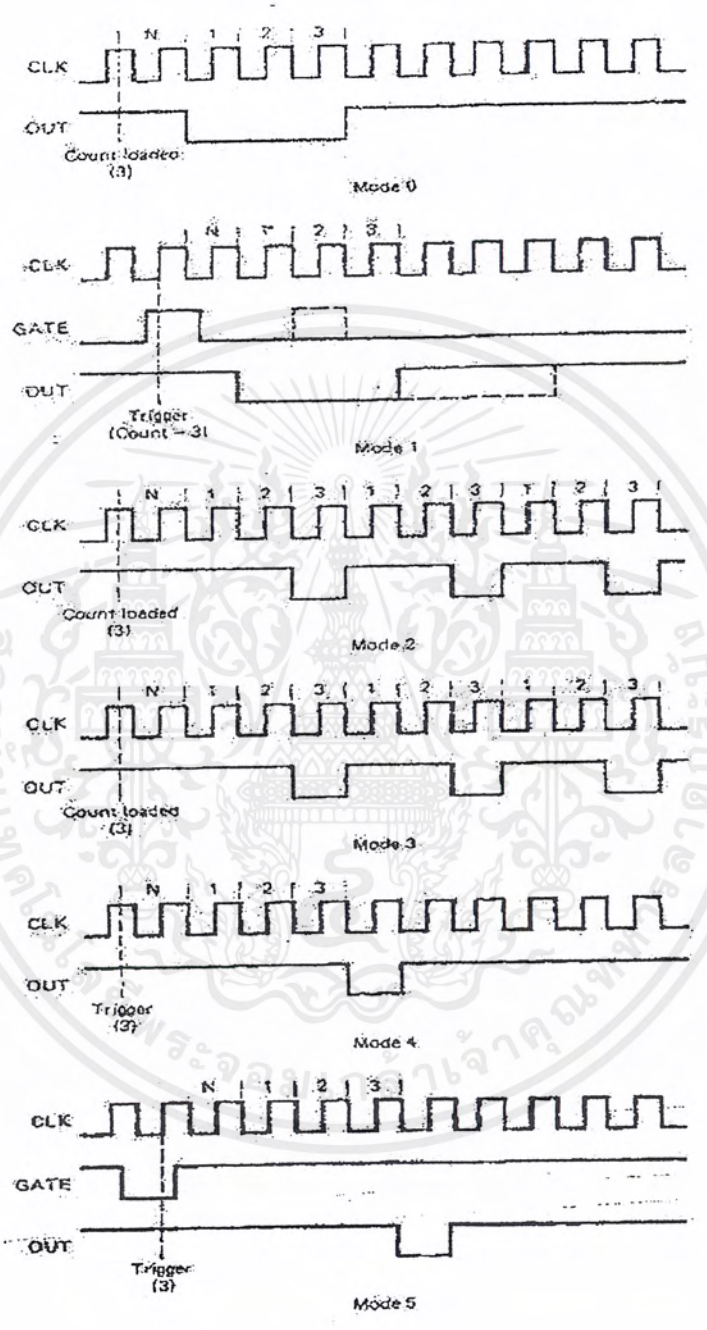
โหมด 2 Rate Generator

โหมด 3 สร้างรูปคลื่นสี่เหลี่ยม

โหมด 4 Software-trig stobe

โหมด 5 Hardware-trig stobe

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงการทำงานของ 8253 ทั้ง 6 โหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่การทำงานในแต่ละโหมด จะต้องควบคุมที่ขา Gate ดังรายละเอียดตาม ตารางที่ 3.4

โหมด	ลอจิก "0" หรือ ⊓	สัญญาณ ⊓	ลอจิก "1"
0	ดิสเอเบิ้ล วงจรนับ	-	อีนาเบิ้ล วงจรนับ
1	-	1.เริ่มนับ 2.รีเซ็ทเอาท์พุทหลังจากสัญญาณลูกต่อมา	-
2	1. ดิสเอเบิ้ล วงจรนับ 2. เซทเอาท์พุทให้เป็น "1" ทันที	1. โหลดค่านับใหม่ 2. เริ่มการนับ	อีนาเบิ้ล วงจรนับ
3	1. ดิสเอเบิ้ล วงจรนับ 2. เซ็ทเอาท์พุทให้เป็น "1" ทันที	เริ่มการนับ	อีนาเบิ้ล วงจรนับ
4	ดิสเอเบิ้ล วงจรนับ	-	อีนาเบิ้ล วงจรนับ
5	-	เริ่มการนับ	-

ตารางที่ 3.4 แสดงการทำงานของขา Gate ทั้ง 6 โหมด

***** ขา GATE ถ้าเป็น 1 จะเป็นการอีนาเบิ้ล Counter

0 จะเป็นการดิสเอเบิ้ล Counter

แต่บางโหมดจะใช้การไม่ได้(ดังตารางที่ 3.3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 การโปรแกรม 8253

ก่อนที่จะนำ 8253 ไปใช้งานได้นั้นต้องทำการโปรแกรม 8253 ก่อน โดยส่ง Control Word ไปยัง Control Port ของ 8253 รูปแบบ Control Word

SC1	SC0	COUNTER
0	0	เลือก COUNTER #0
0	1	เลือก COUNTER #1
1	0	เลือก COUNTER #2

ตารางที่ 3.5 แสดงวิธีการเลือก COUNTER

RL 1	RL 0	ความหมาย
0	1	อ่านหรือเขียนค่านับไบต์ต่ำเท่านั้น
1	0	อ่านหรือเขียนค่านับไบต์สูงเท่านั้น
1	1	อ่านหรือเขียนค่านับไบต์ต่ำก่อนแล้วตามด้วยไบต์สูง

ตารางที่ 3.6 แสดงวิธีการเลือก RL

M2	M1	M0	โหมด	ความหมาย
0	0	0	0	อินเทอร์รัพท์จากการนับ
0	0	1	1	โมโนสเตเบิ้ลที่โปรแกรมได้
X	1	0	2	Rate Generator
X	1	1	3	สร้างรูปคลื่นสี่เหลี่ยม
1	0	0	4	Software-trig stobe
1	0	1	5	Hardware-trig stobe

ตารางที่ 3.7 แสดงการเลือกโหมดการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากเคาน์เตอร์ของ 8253 มีจำนวน 3 ตัวด้วยกัน ฉะนั้นการโปรแกรมเคาน์เตอร์นั้น จำเป็นต้องกำหนดเคาน์เตอร์ ที่ต้องการจะโปรแกรมเสียก่อน การกำหนดทำได้โดยให้ลอจิกที่ถูกต้องกับบิต D7 และ D6 ซึ่งมีชื่อว่า SC1 และ SC0 ของตารางที่ 3.5 เมื่อได้เคาน์เตอร์ที่ต้องการแล้ว เคาน์เตอร์นั้นก็จะถูกเซ็ทและจะอยู่ในสภาพนั้นจนกว่าจะมีคำสั่งควบคุมอื่นมาทำให้เปลี่ยนแปลง

เมื่อเลือกเคาน์เตอร์จากบิต D7 และ D6 ได้แล้วต่อไปบิต D5 และ D4 จะเป็นตัวกำหนดว่าเคาน์เตอร์นี้จะใช้ใน READ หรือ LOAD MODE ซึ่งโหมดการอ่าน (READ MODE) เป็นโหมดที่ไมโครโปรเซสเซอร์ อ่านข้อมูลจากเคาน์เตอร์ ส่วนโหมดการโหลด (LOAD MODE) เป็นโหมดที่ไมโครโปรเซสเซอร์ เขียนข้อมูลเข้าไปให้เคาน์เตอร์

หมายเหตุ เคาน์เตอร์ใน 8253 จะมีขนาด 16 บิต โดยแบ่งเป็น Least Significant Byte และ Most Significant Byte

เมื่อ D5 และ D4 มีค่าเป็น 00H เคาน์เตอร์จะทำให้อยู่ในโหมดการแลทซ์ (Latch) ซึ่งเป็นโหมดที่ใช้สำหรับการอ่านค่าของเคาน์เตอร์ขณะที่เคาน์เตอร์ยังทำงานอยู่โหมดการเขียนนี้ให้กับรีจิสเตอร์ควบคุม จะทำให้ค่าที่อยู่ในเคาน์เตอร์ถูกแลทซ์ให้กับรีจิสเตอร์ภายใน 8253และเมื่อทำการอ่านค่าเคาน์เตอร์นี้จะถูกอ่านออกไป

การอ่านข้อมูลถ้าไม่อยู่ในโหมดการแลทซ์ อาจเกิดผิดผลาดขึ้นได้เพราะขณะที่ทำการอ่านข้อมูล นั้นขบวนการที่เกิดขึ้นในเคาน์เตอร์จะทำให้ข้อมูลที่อยู่เดิมเปลี่ยนไปเป็นผลทำให้ข้อมูลที่ป้อนเข้า CPUเกิดผิดผลาดขึ้น

บิต D3, D2, D1 เป็นบิตที่กำหนดโหมดการทำงานของเคาน์เตอร์ส่วน บิตสุดท้าย D0 ใช้กำหนดลักษณะการนับของเคาน์เตอร์ว่ามีลักษณะการนับเป็นอย่างไรนั่นคือ จะนับเป็น BCD หรือเป็น ไบนารี ถ้า D0 มีลอจิกเป็น “1” เคาน์เตอร์จะนับแบบ BCD ค่าการนับจะวนจาก 9999-0000 ถ้า D0 มีลอจิกเป็น “0” จะนับแบบไบนารีค่าการนับจะวนจาก FFFFH-0000H

เมื่อกำหนดวิธีการของเคาน์เตอร์แล้ว ต่อไปจะเป็นการกำหนดเป็นค่าที่เริ่มนับ (Count Value) ค่าที่เขียนลงไปนี้จะต้องสอดคล้องกับรูปแบบของการอ่านหรือการโหลดที่ได้เลือกไว้ด้วย เช่น เลือกการนับแบบ BCD ค่าการนับก็จะต้องเป็นลักษณะ BCD เป็นต้น ควรจำไว้ว่า ค่าของเคาน์เตอร์ที่อยู่ภายในจะถูกรีเซ็ทเป็น 0000H เมื่อมีการเขียนคำสั่งควบคุม

ในกรณีที่กำหนดวิธีการอ่านและการเขียนเป็นแบบ 2 ไบท์ (RL0,RL1 = 1,1) จะต้องส่งค่าไปให้เคาน์เตอร์ 2 ครั้ง โดยการเขียนค่าที่จะต้องส่ง LSB ก่อนและตามด้วย MSB เสมอต่อเนื่องกันไป

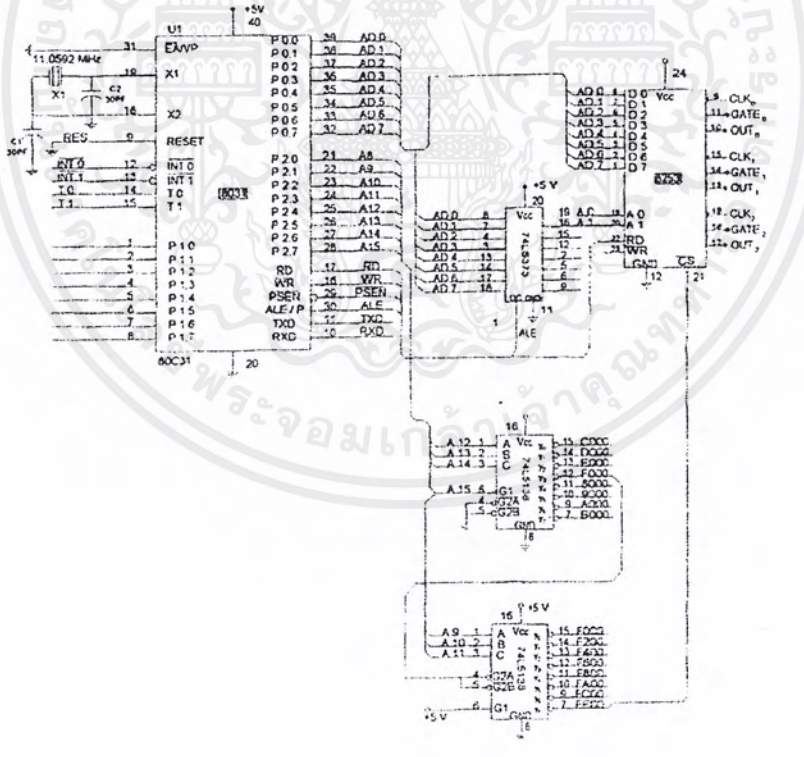
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 การต่อ 8253 เข้ากับ MCS-51

จากรูปที่ 3.5 74LS128 ทำการถอดรหัส ได้เบอร์เท่ากับ (FE00-FE03)H และขา A0, A1 ของ MCS-51 ต่อกับ A0,A1 8253 จะได้เบอร์พอร์ต ดังตารางที่ 3.8

พอร์ต	เลือก COUNTER
FE00H	COUNTER #0
FE01H	COUNTER #1
FE02H	COUNTER #2
FE03H	COUNTER #3

ตารางที่ 3.8 แสดงพอร์ตของ 8253 ตามรูปที่ 3.5



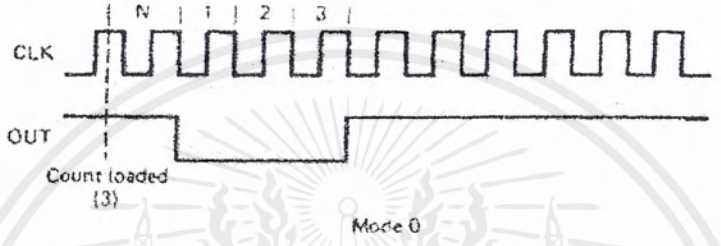
รูปที่ 3.5 แสดงการต่อ 8253 กับ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 การทำงานโหมดต่างๆ มีดังนี้

โหมด 0 (Event Counter)

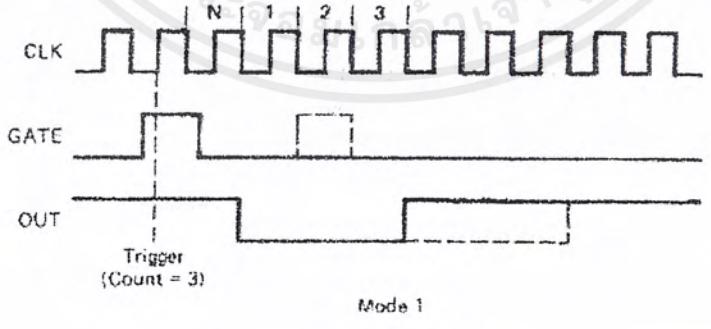
Event Counter ถ้าขา GATE มีลอจิก 1 วงจรจะเริ่มนับจากค่าที่ตั้งไว้ในช่วงขอบขาลง ของคล้อคลูกที่ 2 และจะมีเอาท์พุทเป็น 1 เมื่อวงจรนับลงถึง 0 เมื่อเรานำสัญญาณนี้ไปผ่าน อินเวอร์เตอร์ ก็จะส่งไปอินเตอร์รัพท์ CPU ได้



รูปที่ 3.6 ผังเวลาการทำงาน และการป้อนสัญญาณของ 8253 (โหมด 0)

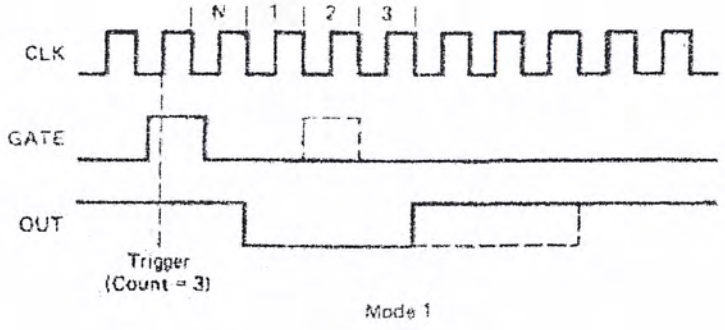
โหมด 1 โมโนสเตเบิ้ลที่โปรแกรมได้แบบทริกซ้ำได้ (Retriggerable Monostable Multivibrator)

ในโหมดนี้เป็นการสร้างพัลซ์ลอจิก 0 มีความกว้างตามค่าที่โปรแกรม การเริ่มการทำงานถูกควบคุมที่ขา GATE ช่วงขอบขาขึ้น ดังตารางที่ 3.3



รูปที่ 3.7 การทำงานของโหมด 1 เมื่อทริกครั้งเดียว

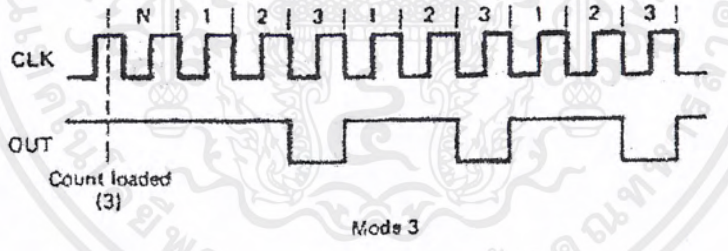
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 การทำงานของโหมด 1 เมื่อมีการทริกซ้ำ

โหมด 2 วงจรกำเนิดสัญญาณสโตรบแบบต่อเนื่อง (Strobe Generator)

ในโหมดนี้จะมีความกว้างดงที่คือเท่ากับ 1 คาบเวลาของคล้อคและช่วงที่เป็น 1 ขึ้นอยู่กับค่านับที่โหลดเข้า Counter ลบด้วยหนึ่ง เช่นถ้าโหลดค่านับเท่ากับ 03 คล้อคอินพุท มีความถี่ 1 MHz ก็จะได้เอาท์พุทช่วงที่เป็น 1 เท่ากับ 2 μs และช่วงที่เป็น 0 เท่ากับ 1 μs จะได้รูปเอาท์พุท ดังรูปที่3.9

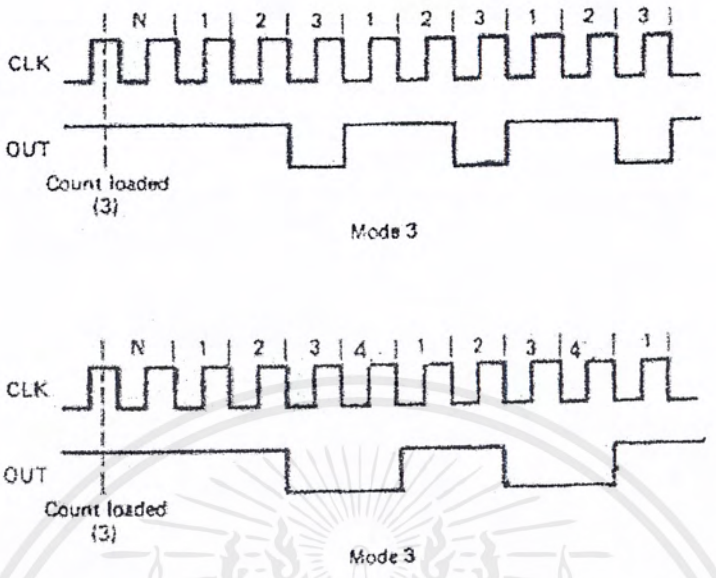


รูปที่ 3.9 เอาท์พุทของโหมด 2

โหมด 3 สร้างรูปคลื่นสี่เหลี่ยม (Square Wave Generator)

รูปคลื่นสี่เหลี่ยมที่ได้จะเป็นแบบ Duty Cycle = 1:1 ถ้าโปรแกรมค่านับเป็นเลขคู่ แต่ถ้าโปรแกรมค่านับเป็นเลขคี่จะได้ช่วงลอจิก 1 กว้างกว่าลอจิก 0 อยู่ เท่ากับความกว้างของCLK 1 ลูก ดังรูปที่ 3.10

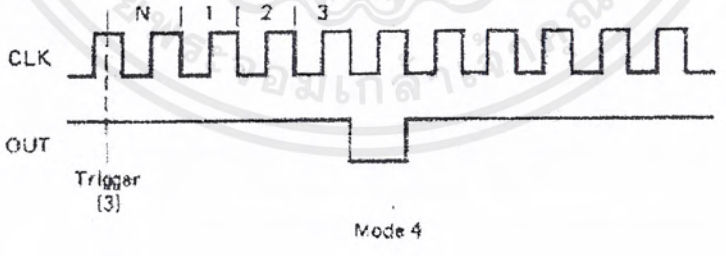
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 รูปคลื่นเอาท์พุทเมื่อใช้ค่านับเลขคู่ และเลขคี่ (โหมด 3)

โหมด 4 ซอฟท์แวร์ทริกสโตรบ (การสร้างสโตรบ โดยไม่ต้องทริกจากภายนอก)

โหมดนี้หลังจากส่งค่านับไบท์สุดท้ายเข้า Counter แล้วก็จะเริ่มนับตามค่าที่โปรแกรมไว้ เมื่อนับครบก็จะมีเอาท์พุทตกลง เป็นลอจิก 0 นาน (1 Clock Period) ดังรูป 3.11



รูปที่ 3.11 การทำงานของโหมด 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การใช้ LCD MODULE

ปัจจุบัน LCD เป็นที่นิยมกันอย่างมาก สำหรับการแสดงผลในเครื่องมือ เครื่องใช้ต่าง ๆ ทั้งนี้ เนื่องจากความเหมาะสมด้วยประการทั้งปวง ทั้งในด้านการกินกระแสต่ำสามารถแสดงผลเป็นตัวอักษรตัวเลข หรือแสดงเป็นกราฟฟิก (Graphic) ได้ (เฉพาะรุ่น) จะติดปัญหาก็คือในด้านวงจร ซึ่งมีระบบการทำงานที่ซับซ้อนและหาอุปกรณ์ได้ค่อนข้างยาก แต่ขณะนี้ผู้ผลิต LCD จะทำรุ่นที่เป็น LCD module ออกมาก็คือเป็น module ที่มีตัว LCD และวงจรควบคุมมาให้พร้อม เรียกว่า LCM ซึ่งทำให้ผู้ใช้สามารถต่อเข้ากับระบบไมโครโปรเซสเซอร์ ได้ง่ายและสำหรับการเขียนโปรแกรม รวมทั้งมีการจำหน่ายกันอย่างกว้างขวาง และมีราคาที่เหมาะสม ทำให้ผู้ใช้งานด้านไมโครโปรเซสเซอร์หันมาใช้แผงแสดงด้วย LCD module กันมากขึ้น

แผงของ LCD จะประกอบด้วย เซกเมนต์แสดงผลขนาดเล็กจำนวนมากในเซกเมนต์จะบรรจุชั้นของเหลวเป็นแผ่นบางๆอยู่ระหว่างชั้นของแก้ว ของเหลวนี้เป็นสารประกอบทำงานโดยอาศัยพลังงานไฟฟ้ามาควบคุมการทำงานหรือการแสดงผลของ LCD เกิดขึ้นเนื่องจากการควบคุมแรงดันที่ตกคร่อมตัวมัน เช่น ถ้าให้แรงดันตกคร่อมเซกเมนต์ ก็จะเกิดสีดำหรือทึบแสงแต่ถ้าเอาแรงดันนั้นออก เซกเมนต์นั้นก็จะสว่างหรือโปร่งแสงด้วยวิธีการจ่ายแรงดันและงดจ่ายแรงดันนี้ก็เพียงพอที่จะควบคุมการแสดงตัวเลข ตัวอักษรและสัญลักษณ์ต่างๆได้ และจากสาเหตุที่จอ LCD มีแรงดันควบคุมดังนั้นจึงกินกำลังงานต่ำและขนาดเล็กแบนราบ

โมดุลของ LCD บางรุ่นอาจจะมี1แถวหรือมากกว่าการแสดงผลของจอ LCD จะอยู่ในรูปเมตริกซ์เช่นบางรุ่นแสดงเมตริกซ์ที่มีขนาดกว้าง 5 เซกเมนต์ สูง 8 เซกเมนต์ และสำหรับรุ่น HD44780 สามารถควบคุมการแสดงผลได้สูงถึง 11 เซกเมนต์ ซึ่งเป็นผลดีกับการแสดงตัวอักษรบางตัว เช่น g , p และ q

จอแสดงผลLCD นั้นมีให้เลือกใช้หลายขนาดแต่ที่นิยมใช้กันมากก็เป็นแบบ 1x16 (1แถว 16 ตัวอักษร), 2x16 (2แถว 16ตัวอักษร) และ2x20 (2แถว 20ตัวอักษร) ส่วนถ้าเป็นจอแสดงผลขนาดใหญ่สามารถแสดงตัวอักษรได้ถึง 80 ตัวอักษร ซึ่งอาจจะต้องมีวงจรขับหรือ ชิปคอนโทรลเลอร์เพิ่มขึ้นเพื่อใช้ร่วมกับ HD44780 ที่ต่อสายสัญญาณ 14 เส้นได้ ตารางที่ 3.9 แสดงขาต่างๆที่ใช้ในการเชื่อมต่อกับไมโครคอนโทรลเลอร์

สำหรับไฟเลี้ยง LCDจอแสดงผล LCD จะใช้ไฟเลี้ยง+5 โวลต์ ป้อนให้ที่ขา 2 ซึ่งตัวมันกินกระแสเพียงไม่กี่มิลลิแอมป์ ส่วนขาที่ 3 ต่อเพื่อปรับมุมมองการแสดงผลให้เหมาะสม ทั้งนี้ก็ขึ้นอยู่กับผลของแสงในขณะนั้นด้วยรวมไปถึงตำแหน่งการติดตั้งและอุณหภูมิ

เมื่อเปรียบเทียบคุณสมบัติระหว่าง LED กับ LCD จะพบว่าในที่ที่มีแสงสว่างค่อนข้างสูง LED เกือบจะมองไม่เห็น ส่วน LCD นั้น สามารถอ่านในที่ที่มีแสงสว่างได้ เนื่องจากว่าการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของ LED นั้นจะปล่อยพลังงานแสงออกมาส่วน LCD นั้นจะใช้การหักเหแสง โดยให้แสงส่งผ่านตัวมัน ซึ่งบางสภาวะในที่มีแสงสว่างน้อยก็ไม่สามารถอ่านค่าจอแสดงผล LCD ได้วิธีแก้ก็คือการใช้จอ LCD ที่มีแบ็กไลต์ (Backlight) จึงเป็นการใช้ฉาก electrolumine (EL) ซึ่งมีความสามารถในการเรืองแสงได้นำไปติดตั้งไว้ด้านหลังของจอ LCD โดยที่แผง EL จะทำหน้าที่แพร่กระจายความสว่างจากด้านหลัง ทำให้ LCD มีความสว่างและทำให้เรามองเห็นได้

การที่จะนำสารเรืองแสง Eletrolumine มาใช้งานนั้น ที่ชุดโมดูล LCD ต้องมีแผง EL และชุดแปลงแรงดันเป็นสัญญาณไฟสลับแรงดันสูง ซึ่งจะเป็นอุปกรณ์แปลงแรงดันไฟดีซี 5 โวลต์เป็นไฟเอซี 100 โวลต์ ที่ความถี่ 400 เฮิรตซ์ อุปกรณ์แปลงแรงดันที่ต้องใช้กระแสหลายมิลลิแอมป์ในการทำงานจึงทำให้เป็นข้อเสียเปรียบของอุปกรณ์ตัวนี้

โมดูลของ LCD แบ่งออกเป็นแบบสะท้อนกลับ (Reflective) แบบนี้จะไม่ใช้แหล่งกำเนิดแสงทางด้านหลัง ส่วนอีกแบบหนึ่งคือแบบส่งผ่าน (transmissive) แบบนี้ต้องใช้แหล่งกำเนิดแสงด้านหลังและแบบสุดท้ายคือแบบผสม (transflective) แบบนี้จะใช้แหล่งกำเนิดแสงจากด้านหลังหรือไม่ใช้ก็ได้ โดยสามารถต่อสวิตช์เข้ากับแหล่งกำเนิดแสง เวลาจะใช้แหล่งกำเนิดแสงก็เปิด หรือถ้าไม่ต้องการใช้ก็ปิดตามต้องการ

3.3.1 คอนโทรลเลอร์และการควบคุม

การที่จะใช้โมดูลของแอลซีดี ในงานใดงานหนึ่งนั้น จะต้องทำความเข้าใจกับตัวควบคุมก่อน HD44780 เป็นตัวควบคุมขนาดเล็กที่คล้ายกับคอมพิวเตอร์โดยจะทำงานทั้งหมด 11 คำสั่ง แสดงดังในตารางที่ 3.9 เพื่อควบคุมการทำงานต่าง ๆ เช่นเคลียร์หน้าจอแสดงผล, เขียนตัวอักษร, เลือกตำแหน่งที่จะแสดงผลและอ่านข้อมูลจากจอแสดงผล

หน่วยความจำภายใน HD44780 มี 2 ชนิดคือ Character Generator (CG) ROM และ Character – Generator (CG) RAM

CGROM ใช้สำหรับเก็บตัวอักษรเกือบ 200 รูปแบบ เช่นตัวอักษรภาษาอังกฤษ, ตัวเลข เครื่องหมายทางคณิตศาสตร์, สัญลักษณ์พิเศษ และอักษรญี่ปุ่น ซึ่งจะกำหนดลงในรอมไว้แล้วไม่สามารถแก้ไขหรือเปลี่ยนแปลงได้

CGROM ใช้เก็บตัวอักษรที่ผู้ใช้ออกแบบขึ้นเองเช่น โลโก้ สัญลักษณ์พิเศษอักษรกราฟิกง่าย ๆ ที่สามารถออกแบบบนเมตริกซ์ขนาด 5x8 ได้ อักษรที่เขียนขึ้นนี้จะเขียนครั้งละ 5 บิต หลาย ๆ คำ แต่ละคำจะแทนรูปแบบเซกเมนต์ 1 แถว แล้วเก็บไว้ใน CGRAM รูปแบบอักษรนี้จะหายไปเมื่อปิดเครื่อง และเมื่อจะใช้งานต้องเรียกข้อมูลมาใหม่ หลังจากเปิดเครื่อง

อักษรใน CGROM และ CGRAM เป็นอักษรขนาด 8 บิต (0 ถึง FFh) ซึ่งบางตำแหน่งก็ไม่ได้ใช้ ตำแหน่งแอดเดรสที่ใช้กันมากจากช่วง 21H ถึง 7DH ซึ่งจะตรงกับตำแหน่งรหัสแอสกี้บน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์ เช่น " A " จะถูกเก็บไว้ที่ตำแหน่ง 41H และ " B" จะเก็บไว้ที่ตำแหน่ง 42H เป็นต้น โดยตำแหน่งแอดเดรสจะถูกเก็บอยู่ในเลขฐาน 16

ไอซี HD44780 มีรีจิสเตอร์ 2 ตัวคือ Instruction Register (IR) ซึ่งใช้สำหรับเก็บรหัสของคำสั่งและ Data Register (DR) ซึ่งใช้สำหรับเก็บรหัสอักษรเมื่อต้องการเขียนหรืออ่านข้อมูลจากไอซี จะต้องเลือกใช้รีจิสเตอร์ให้เหมาะสมเพื่อนำไปต่อกับขา 4 ของ LCD ตามหน้าที่การทำงานที่เราต้องการ

ขา	สัญลักษณ์	ฟังก์ชั่น
1	Vss	กราวด์
2	Vdd	+5 โวลต์
3	V0	ปรับความสว่างด้วยแรงดัน (0-5 โวลต์)
4	RS	เลือกรีจิสเตอร์ (0 = รีจิสเตอร์คำสั่งหรือแฟลกแสดงสภาวะการทำงานและตัวนับแอดเดรส ;1 = รีจิสเตอร์ดาต้า)
5	RW	เลือกการอ่านหรือเขียน (0 = เขียน ; 1 = อ่าน)
6	E	อีนาเบิลการอ่านหรือเขียน LCD
7	D0	ดาต้าอินพุต/เอาต์พุตบิตต่ำสุด
8	D1	ดาต้าอินพุต/เอาต์พุตบิตที่ 2
9	D2	ดาต้าอินพุต/เอาต์พุตบิตที่ 3
10	D3	ดาต้าอินพุต/เอาต์พุตบิตที่ 4
11	D4	ดาต้าอินพุต/เอาต์พุตบิตที่ 5
12	D5	ดาต้าอินพุต/เอาต์พุตบิตที่ 6
13	D6	ดาต้าอินพุต/เอาต์พุตบิตที่ 7
14	D7	ดาต้าอินพุต/เอาต์พุตบิตสูงสุด

ตารางที่ 3.9 แสดงตำแหน่งขาต่าง ๆ ที่ใช้เชื่อมต่อกับ LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำแรมเก็บข้อมูลแสดงผล Display Data (DD) RAM จะเก็บรหัสตัวอักษรขนาด 8 บิตได้มากกว่า 80 ตัวอักษร ไว้ในแอดเดรสแต่ละตำแหน่งโดยรหัสของตัวอักษรที่เก็บไว้ใน DDRAM จะกำหนดว่าให้แสดงอักษรที่ตำแหน่งไหน

ในกรณีที่ LCD มีสองแถว (ถ้าต้องการลบแถวแรก) การทำงานของมันจะทำการเลื่อนตำแหน่งซ้ายสุดของแถวบนซึ่งเป็นตำแหน่งแอดเดรสสูงก่อน และตำแหน่งต่อไปก็จะเลื่อนตามมา ตามลำดับจนกว่าจะหมดแถวที่ 1 แล้วแถวที่ 2 จะตามมา เช่นสมมติว่าแถวที่ 1 มีตำแหน่งแอดเดรส เริ่มที่0-39H พอหมดแถวที่ 1 แล้วแถวที่ 2 จะถูกลบต่อไปเป็น 40H, 41H,42H,43H และต่อๆไปจนครบ

แต่ละครั้งที่ทำการเขียนอักษรลงในแอดเดรสของ DDRAM แต่ละตำแหน่งแอดเดรสจะเพิ่มตำแหน่งขึ้นโดยอัตโนมัติตามลำดับของการเขียนตัวอักษร ยกเว้นกรณีที่ผู้ใช้กำหนดตำแหน่งแอดเดรสที่จะเขียนข้อมูลเองในบางครั้ง

อย่างไรก็ตามเนื่องจากว่าแถวที่ 2เริ่มที่ตำแหน่งแอดเดรส 40H ดังนั้นถ้าหากผู้ใช้ต้องการให้แสดงที่แถวที่ 2 จะต้องอ้างตำแหน่งแอดเดรสให้ถูกต้องด้วย เช่นถ้าจอแสดงผลมี 16 ตำแหน่ง แถวที่ 1 จะสิ้นสุดที่ตำแหน่ง 0FH และแถวที่ 2 จะเริ่มต้นที่ 40H ดังนั้นถ้าต้องการย้ายอักษรจากตำแหน่งขวาสุดของแถวที่ 1 ไปยังตำแหน่งซ้ายสุดของแถวที่ 2 ผู้ใช้จะต้องอ้างตำแหน่งแอดเดรสไปที่ 40H

นอกจากนี้จอแสดงผลบางรุ่น จะมีแถวแสดงอักษรเป็นแบบฟิสิกซ์คอล 1 แถว และแบบลอจิคอลอีก 2 แถว ซึ่งจะคล้ายกับจอแสดงผลแบบ 2 แถว ในกรณีที่เป็นจอแสดงผลแบบ 16 ตัวอักษร 8 ตัวอักษรแรกจะเริ่มที่แอดเดรส 0 ถึง 7 และ 8 ตัวอักษรหลังจะเริ่มที่แอดเดรส 40H ถึง 47H ดังนั้น ถ้าจะเขียนข้อมูลไปที่ตำแหน่ง 8 ตัวหลังต้องกำหนดตำแหน่งไปที่ 40H

อย่างไรก็ตาม ในจอแสดงผลขนาดเล็กจะไม่มี DDRAM เนื่องจากว่า DDRAM เป็นแรมที่ใช้สำรองไว้ให้ผู้ใช้เลือกใช้ตามวัตถุประสงค์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INSTRUCTION	RS	R/W	DATA BIT								EXE TIME
			7	6	5	4	3	2	1	0	(μS)
CLEAR DISPLAY	0	0	0	0	0	0	0	0	0	1	1640
CURSOR AT HOME	0	0	0	0	0	0	0	0	1	*	1640
ENTRY MODE SET	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
DISPLAY 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							1
CGRAM DDRAM WR	1	0	WRITE DATA								40
CGRAM DDRAM RD	1	1	READ DATA								40

I/D = 1 : เพิ่มค่า — I/D = 0 : ลดค่าลง

S = 1 : ตัวอักขระจะถูกเลื่อน

S/C = 1 : ตัวอักขระถูกเลื่อน — S/C = 0 : เคอร์เซอร์ถูกเลื่อน

R/L = 1 : เลื่อนไปทางขวา — R/L = 0 : เลื่อนไปทางซ้าย

DL = 1 : 8 บิต — DL = 0 : บิต

N = 1 : 2 บรรทัด — N = 0 : 1 บรรทัด

F = 1 : 5x10 จุด — F = 0 : 5x7 จุด

BF = 1 : การทำงานภายในยังไม่เสร็จสิ้น — BF = 0 : สามารถรับคำสั่งใหม่ได้

ตารางที่ 3.10 แสดงรายละเอียดของชุดคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 การอ่านและการเขียน

รูปที่ 3.12 จะแสดงไทมิ่งไดอะแกรมของการประมวลผลคำสั่ง ในการอ่านและเขียนข้อมูลกับ LCD ขั้นตอนของการเขียนเริ่มจากมีสัญญาณ RS เข้ามาและให้สัญญาณ R / $\overline{W}$ มีสถานะเป็น "LOW" หลังจากนั้นประมาณ 140 นาโนวินาที สัญญาณอีนาเบิลจะมีสถานะเป็น "HIGH" และคงสถานะอยู่อย่างน้อย 450 นาโนวินาทีเพื่อที่จะให้ขาดาต้า D_0-D_7 ส่งข้อมูลอย่างน้อย 195 นาโนวินาทีก่อนที่สัญญาณอีนาเบิลจะเป็น "LOW" อีกครั้ง

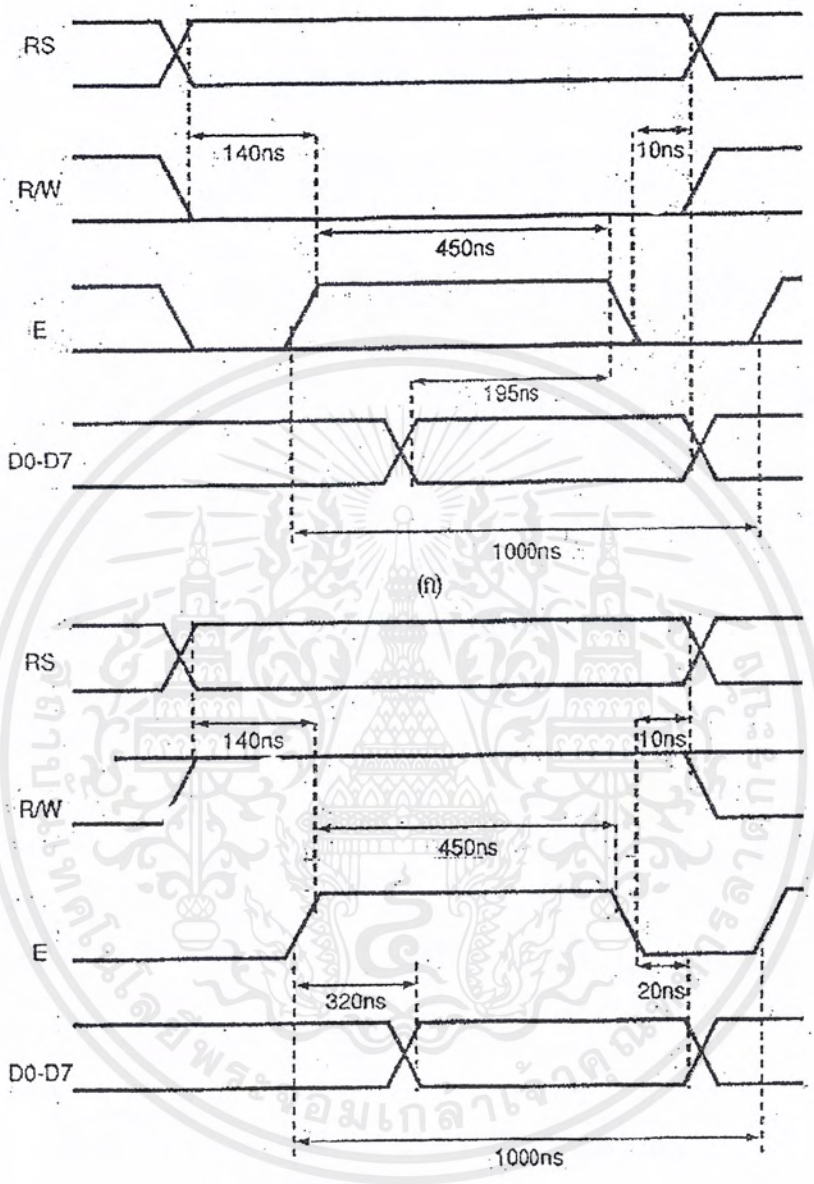
ส่วนขั้นตอนในการอ่านข้อมูลจะคล้ายกับการเขียนแต่สัญญาณ R/ $\overline{W}$ จะเป็น "HIGH" ส่วนสัญญาณข้อมูล D_0-D_7 จะทำงานหลังจากสัญญาณอีนาเบิลเป็น "HIGH" แล้วประมาณ 320 นาโนวินาที

ไอซี HD44780 จะไม่ทำคำสั่งใหม่ที่เข้ามาจนกว่าจะทำคำสั่งที่ทำงานอยู่ขณะนั้นเสร็จก่อน ซึ่งในกรอบแยกที่ 1 จะแสดงเวลาที่มากที่สุดที่แต่ละคำสั่งใช้ในการประมวลผล แต่ถ้าใช้ภาษาเบสิกหรือภาษาระดับสูงในการโปรแกรมค่าเวลาเหล่านี้อาจไม่ต้องใส่ใจกับมันมากนัก เพราะว่าตัวโปรแกรมจะเข้าถึงคำสั่งโดยอัตโนมัติอยู่แล้ว ถ้าหากต้องการใช้ชุดโมดูล LCD รับคำสั่งต่อมาอาจทำได้โดยเขียนโปรแกรมหน่วงเวลาหลังจากทำคำสั่งเหล่านั้นหรืออาจจะอ่านแฟลกว่าง (บิต 7) ก็ได้

คำสั่งในการแสดงการเชื่อมต่อของขนาดข้อมูลขนาด 8 บิต (D_0– D_7) ซึ่งถ้าเป็นคอนโทรลเลอร์ HD 44780 จะใช้ข้อมูลขนาด 4 บิตเท่านั้น บางครั้งถ้าเรามีไมโครโทรลเลอร์ขนาด 8 บิต เราสามารถที่จะต่อใช้ข้อมูลเพียง 4 บิต ก็ได้เพื่อเป็นการประหยัดฮาร์ดแวร์ การใช้ข้อมูลเพียง 4 บิตนี้จะใช้สัญญาณในการส่งข้อมูลทั้งหมดเพียง 7 เส้น เท่านั้น (D_0- D_7 , RS , R/ $\overline{W}$ และอีนาเบิ้ล) หรืออาจใช้เพียง 6 เส้น ถ้าให้สัญญาณ R/ $\overline{W}$ เป็น "LOW" ตลอด ก็จะสามารถอ่านข้อมูลบิต D_0- D_7 แสดงบนจอแสดงผลได้

อย่างไรตามการที่จะส่งคำสั่งเพื่อใช้กับข้อมูลบิต D_4 – D_7 ไปพร้อมกับสัญญาณ RS และ R/ $\overline{W}$ โดยไม่ต้องใช้บิต D_0-D_3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 ไทม์มิ่งไดอะแกรม

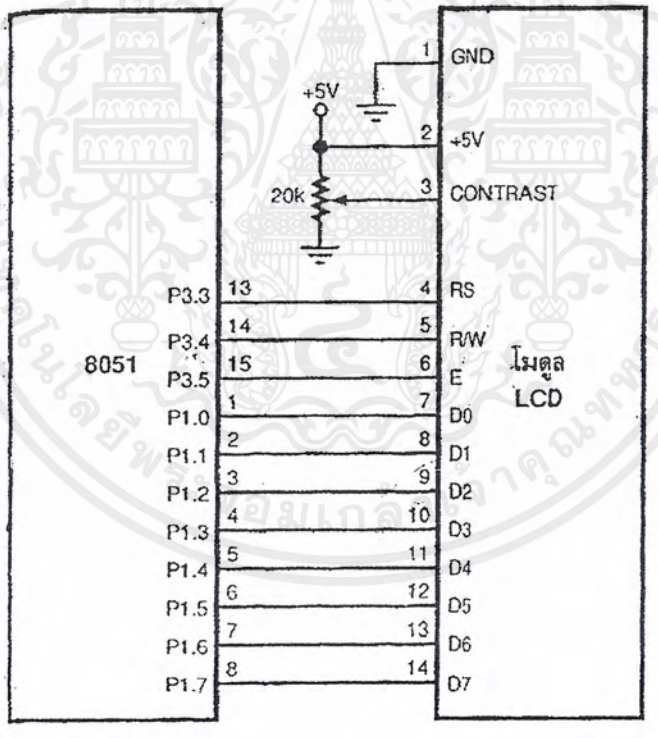
(ก) เป็นการเขียนข้อมูลลงบนโมดูล LCD

(ข) เป็นการอ่านข้อมูลจากโมดูล LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 การเชื่อมต่อ

โมดูลของ LCD สามารถที่เชื่อมต่อกับไมโครคอนโทรลได้หลายเบอร์รูปที่ 3.13 แสดงการเชื่อมต่อโมดูล LCD เข้ากับไมโครคอนโทรลเลอร์ 8051 โดยมีหน้าที่การทำงานของแต่ละขาดังนี้ คือ ขา 1 ต่อ กราวด์ขา 2 ต่อไฟ + 5 โวลต์ ขา 3 ต่อกับตัวต้านทานปรับคาได้แบบโพเทนชิโอมห์มิเตอร์ เมื่อปรับความเข้มและความสว่าง ขา 4,5 และ6 ใช้ต่อสัญญาณควบคุม ซึ่งในที่นี้จะต้องใช้พอร์ต 3 ของ 8051 เป็นตัวควบคุม ส่วนบัสของข้อมูล (Data Bus) ที่ขา 7 ถึงขา 14 จะต่อกับพอร์ต 1ของ 8051 เพื่อจะอ่านและเขียนข้อมูลไปบนโมดูล LCD การเชื่อมต่อสัญญาณที่แสดงในรูปที่ 3.13 นั้นไม่ได้หมายความว่าจะต่อในลักษณะนี้ได้เท่านั้น แต่สามารถที่จะใช้ไมโครคอนโทรลเลอร์เบอร์อื่นก็ได้ เช่น ตระกูล 6800 ซึ่งอาจจะต่อผ่านพอร์ตของไมโครคอนโทรลเลอร์โดยตรงหรืออาจจะต่อผ่าน 8255 หรือไอซี PPI เบอร์อื่นก็ได้

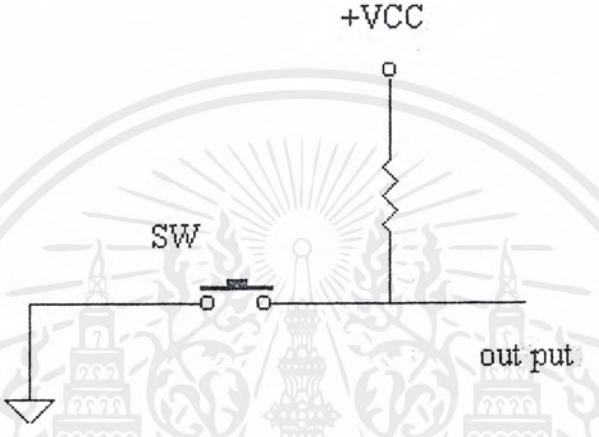


รูปที่ 3.13 แสดงการต่อโมดูล LCD ร่วมกับไมโครคอนโทรลเลอร์ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

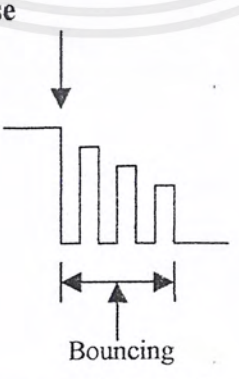
3.4 คีย์บอร์ด

คีย์บอร์ดเป็นอุปกรณ์ที่ใช้ในการส่งค่าข้อมูล โดยส่งข้อมูลจากหลักการของเมตริกซ์ กล่าวคือ ไมโครคอนโทรลเลอร์จะส่งค่าสัญญาณขนาด 8 บิต ออกมาที่พอร์ตซึ่งมีการต่อคีย์บอร์ดอยู่โดยค่าสัญญาณขนาด 8 บิตนี้ จะเปลี่ยนแปลงเมื่อกดคีย์ใดคีย์หนึ่ง โดยพื้นฐานของคีย์บอร์ดคือสวิตซ์ โดยมีวงจรดังรูปที่ 3.14



รูปที่ 3.14 แสดงวงจรของสวิตซ์

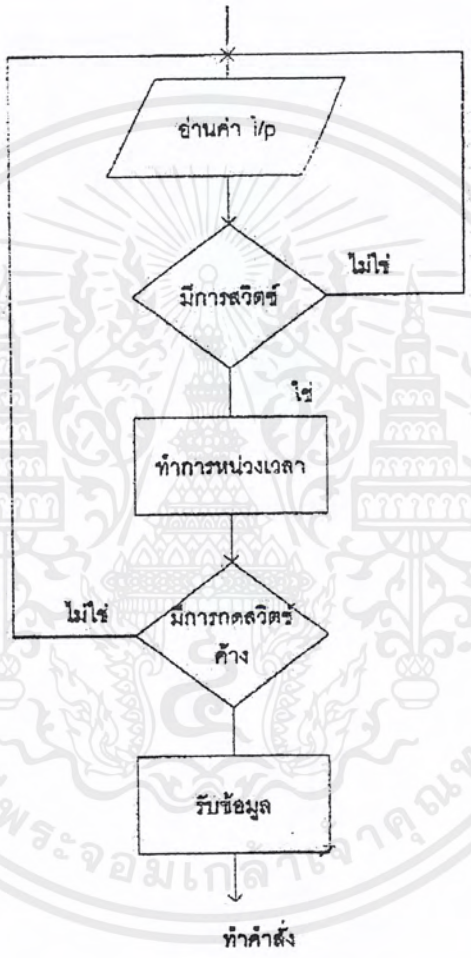
ขณะยังไม่กดสวิตซ์ เอาท์พุทจะมีค่าเท่ากับ +VCC และเมื่อกดสวิตซ์ เอาท์พุทจะมีค่าเท่ากับ 0 โวลท์ เนื่องจากในการกดสวิตซ์ โดยทั่วไปจะปรากฏพัลซ์ ที่ไม่ต้องการขึ้นมา ซึ่งเกิดจากการกระเด้งของหน้าสัมผัส ปรากฏการณ์นี้เรียกว่า การบราวซ์ (Bounce) ดังรูปที่ 3.15



รูปที่ 3.15 แสดงการเกิดบราวซ์ขณะกดสวิตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเกิดบราวซ์จะทำให้การรับข้อมูลผิดพลาดได้ ดังนั้นจึงต้องแก้สัญญาณบราวซ์โดยใช้ซอฟต์แวร์ โปรแกรมจะทำงานโดยการตรวจสอบการกดสวิตซ์ครั้งแรกแล้วรอสักระยะเวลาช่วงหนึ่งเพื่อที่จะให้ผ่านช่วงเวลาที่เกิดการบราวซ์ผ่านไปแล้ว ทำการอ่านค่าข้อมูลจากสวิตซ์อีกครั้งหนึ่งเพื่อจะนำไปเป็นข้อมูล การแก้สัญญาณบราวซ์นั้นสามารถนำมาเขียนโฟลว์ชาร์ท ดังรูปที่ 3.16

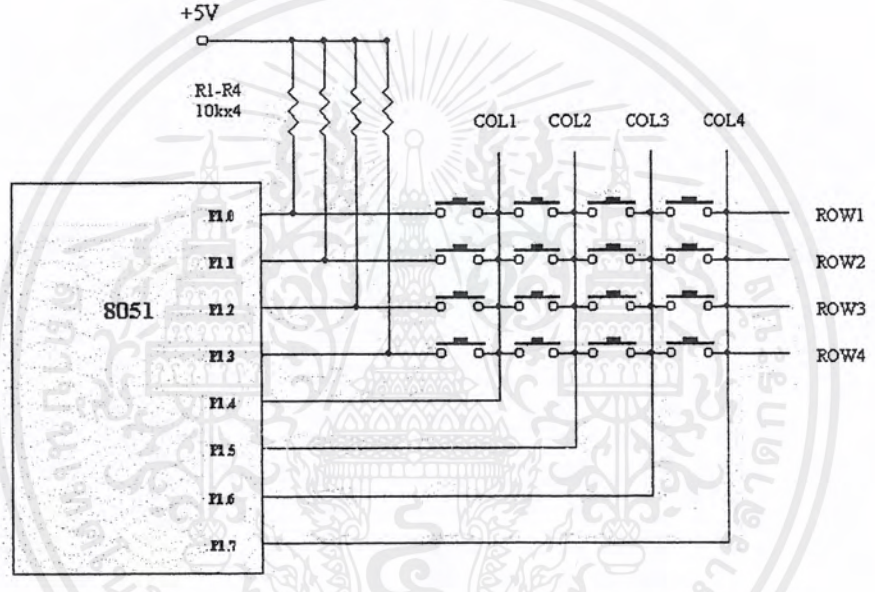


รูปที่ 3.16 โฟลว์ชาร์ทแสดงการแก้สัญญาณบราวซ์

จะเห็นว่าถ้าให้สวิตซ์หนึ่งตัวแทนคำสั่ง 1 คำสั่งจะทำให้เกิดการเปลืองสวิตซ์จำนวนมาก รวมทั้งเปลืองสายที่จะใช้ติดต่อกับบอร์ด ดังนั้นจึงใช้คีย์บอร์ดโทรศัพท์ ซึ่งเป็นคีย์บอร์ดแบบเมตริกซ์ ชนิด 4x4 ซึ่งมีปุ่มทั้งหมดเพียง 16 ปุ่ม ซึ่งเกือบทั้งหมดปุ่มจะทำหน้าที่รับคำสั่งได้ 2 คำสั่ง และใช้สายติดต่อกับบอร์ดทั้งหมดเพียง 8 เส้น ยังมีสวิตซ์เพิ่มเข้ามาอีกหนึ่งตัวเพื่อใช้ในการอินเตอร์รัพท์ เพื่อเข้าสู่สแกน (scan) คีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรับข้อมูลจากสวิตซ์หลายตัวแบบเมติกซ์ซึ่งเป็นการควบคุมการรับข้อมูลเข้ามาจากคีย์บอร์ด ซึ่งได้จัดวางไว้ในลักษณะแบบเมตริกซ์ โดยใช้พอร์ตแบบขนานของ 8051 โดยปกติแล้วการควบคุมการรับสัญญาณอินพุทจากกลุ่มของสวิตซ์ (คีย์บอร์ด) ก็มักจะใช้เส้นสัญญาณเดี่ยวในการจัดการ แต่อย่างไรก็ตามหากว่าจำนวนของสวิตซ์มีมากกว่า 8 ตัวแล้ว ก็จะต้องใช้เส้นสัญญาณจากพอร์ตมากกว่าหนึ่งพอร์ตเข้ามาทำหน้าที่ดูแล ดังนั้นหากว่าได้มีการจัดวางองค์ประกอบของสวิตซ์เป็นแบบเมตริกซ์แล้ว (ดังรูปที่ 3.17) ก็จะทำให้สามารถลดจำนวนของเส้นสัญญาณเหล่านี้ลงได้อย่างมาก



รูปที่ 3.17 แสดงการต่อวงจร 8051 กับคีย์บอร์ด แบบเมตริกซ์

หลักการทำงานของโปรแกรมจะใช้หลักการในการสแกน เพื่อหาว่าสวิตซ์ใดที่ถูกกด ซึ่งในการเขียนโปรแกรมสแกนคีย์บอร์ดนี้ จะใช้พอร์ต P1 เป็นพอร์ตอินพุท เอาท์พุท ในการใช้พอร์ต P1 เป็นพอร์ตอินพุทนั้นจะต้องส่งค่า 1 ไปยังแต่ละบิทของพอร์ตเสียก่อน เพื่อกำหนดให้เป็นพอร์ทอินพุทซึ่งในการรับรู้ของไมโครคอนโทรลเลอร์ว่าคีย์ใดถูกกดนั้นจำเป็นจะต้องมีตัวแก้สัญญาณบราวซ์ ซึ่งจะใช้ซอฟท์แวร์เป็นตัวแก้สัญญาณบราวซ์ โดยการตรวจสอบการกดคีย์ครั้งแรกแล้วรอสักระยะเวลาช่วงหนึ่ง เพื่อที่จะให้ผ่านช่วงเวลาที่เกิดบราวซ์ผ่านไป แล้วก็ทำการอ่านข้อมูลจากสวิตซ์อีกครั้งหนึ่งเพื่อจะนำไปเป็นข้อมูลต่อไป เวลาที่ใช้หน่วงในการแก้บราวซ์ประมาณ 10-40 ms ก็เพียงพอแล้ว

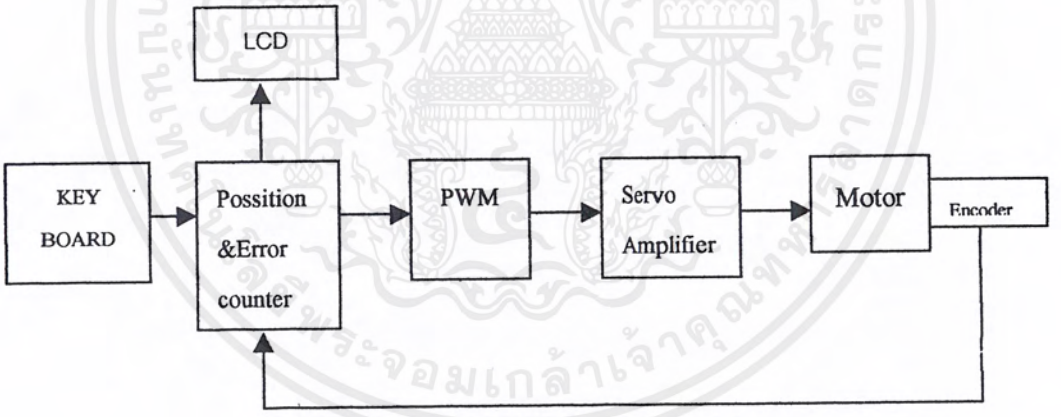
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
หลักการและขั้นตอนการทำงาน

บทนำ

หลักการทำงานของเซอร์โวมอเตอร์ จะถูกขับโดยการควบคุมจากไมโครคอนโทรลเลอร์ ซึ่งการขับจะใช้เทคนิคพัลซ์วิดท์มอดูเลชั่น PWM (Pulse Width Modulation) และใช้เพาเวอร์มอสเฟต (Power Mosfet) ซึ่งต่อแบบ H-bridge เป็นวงจรขับ ในส่วนของการจับตำแหน่งการเคลื่อนที่จะใช้เคาน์เตอร์ ภายในไมโครคอนโทรลเลอร์เป็นตัวจับสัญญาณพัลซ์จากเอ็นโค้ดเดอร์ซึ่งอยู่กับตัวเซอร์โวมอเตอร์

4.1 รูปแบบโครงงาน

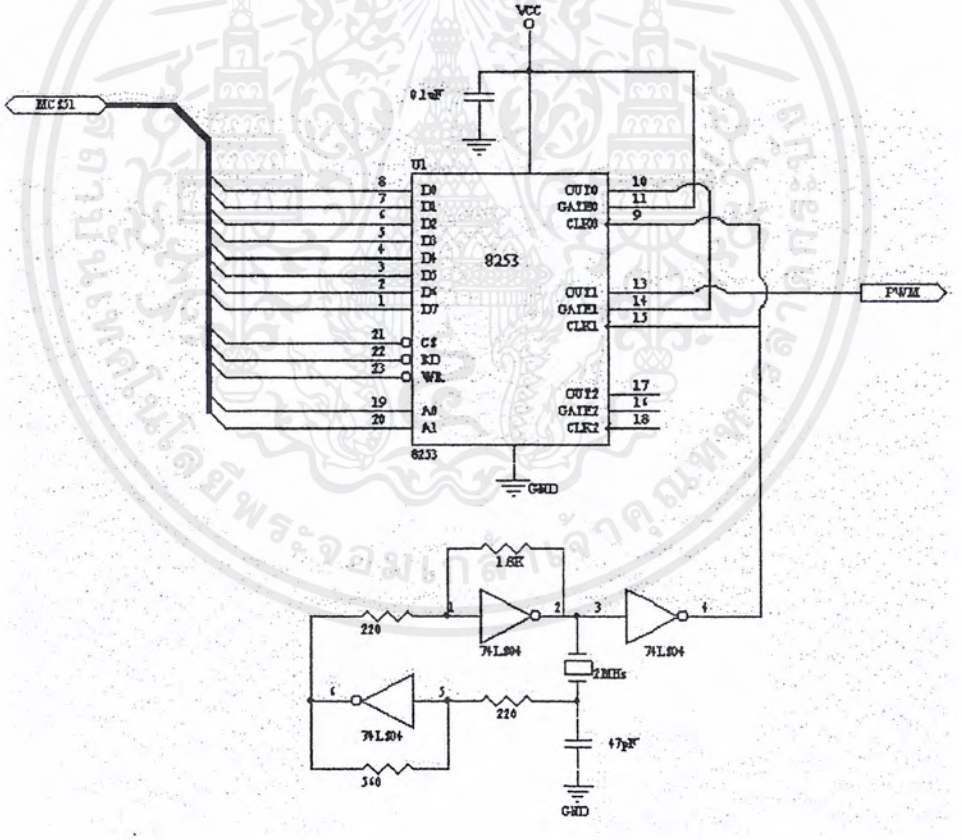


รูปที่ 4.1 แสดงระบบโดยรวมของโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 เทคนิคการสร้างสัญญาณPWM

ในการสร้างสัญญาณ PWM นั้นเราจะใช้ ไอซี 8253 ใช้เป็นตัวสร้างสัญญาณ PWMโดยที่ให้เคาน์เตอร์ #0 ถูกโปรแกรมให้ทำงานที่โหมด 3 คือเป็นตัวสร้าง Square Wave โดยที่ขาเกทถูกเซ็ทให้เป็น “i” และขาคล็อคต่อกับส่วนสร้างสัญญาณนาฬิกาความถี่ 2 MHz เพื่อเป็นฐานเวลาในการสร้างความถี่ของสัญญาณ PWM ที่ 20 kHz สัญญาณที่ได้จากเคาน์เตอร์ #0 จะถูกต่อเข้ากับขาเกทของเคาน์เตอร์ #1 โดยที่เคาน์เตอร์ตัวนี้จะถูกโปรแกรมให้ทำงานที่โหมด 1 ซึ่งเรียกว่า โมโนสเตเบิ้ลมัลติไวเบรเตอร์ จะเป็นการสร้างพัลซ์ให้มีความกว้างของคาบเวลา (Duty Cycle) ตามค่าที่โปรแกรม โดยที่ขาคล็อคของเคาน์เตอร์จะต้องป้อนสัญญาณนาฬิกาให้ด้วย ซึ่งจากหลักการดังกล่าวเราสามารถที่จะสร้างสัญญาณ PWM ได้

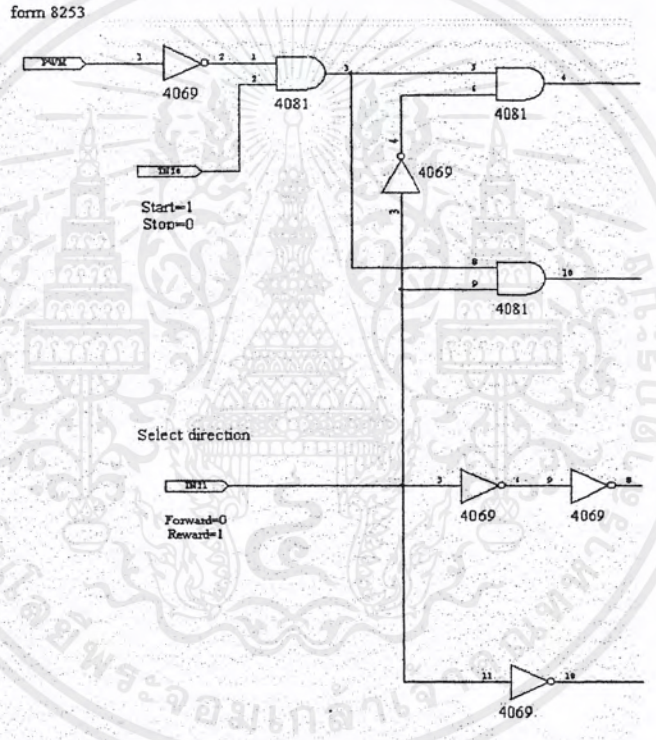


รูปที่ 4.2 แสดงการต่อวงจรสร้างพัลซ์วิดท์มอดูเลชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 วงจรควบคุมการหมุนของมอเตอร์ และขับเกทมอสเฟต

หลักการคือ การควบคุมการทำงานของมอเตอร์จะถูกควบคุมโดยไมโครคอนโทรลเลอร์ และการใช้ลอจิกในการควบคุม โดยที่จะใช้ขา INT 0 ควบคุมการ Start และ Stop ของมอเตอร์ โดย ถ้า INT0=1 จะทำให้มอเตอร์ทำงาน ถ้า INT0=0 จะทำให้มอเตอร์หยุดทำงาน ส่วนการควบคุมทิศทางการหมุนจะใช้ขา INT 1 ควบคุมทิศทางการหมุนของมอเตอร์ โดย ถ้า INT1=0 จะทำให้มอเตอร์หมุนไปในทิศทาง Forward ถ้า INT1=1 จะทำให้มอเตอร์หมุนในทิศทาง Reward ซึ่งจะใช้ลอจิกในการควบคุมมอเตอร์โดยใช้ CMOS 4069 และ CMOS 4081 ดังรูปที่ 4.3



รูปที่ 4.3 แสดงวงจรการควบคุมการหมุนของมอเตอร์

โดยที่ในการสวิตท์ชิ่งของเพาเวอร์มอสเฟตจะใช้หลักที่ให้คู่มอสเฟตทำงานเพียงคู่ใดคู่หนึ่ง ทำเพียงคู่เดียวต่อทิศทางการหมุนในทิศทางใดทิศทางหนึ่ง คือ เมื่อให้มอเตอร์มีทิศทาง Forward ก็จะให้ Q4 ทำงานตลอดเวลา และให้ Q1 ถูกขับด้วยสัญญาณ PWM ที่ซึ่งเมื่อ Q1 ทำงาน กระแสจะไหลผ่าน Q1 และ Q4 และเมื่อ Q1 ไม่ทำงาน กระแสจะไหลวนเวียนผ่าน Q4 และ D3 เนื่องจากผลของการเหนี่ยวนำของมอเตอร์ ส่วนในทิศทาง Reward การจะใช้หลักการเดียวกัน

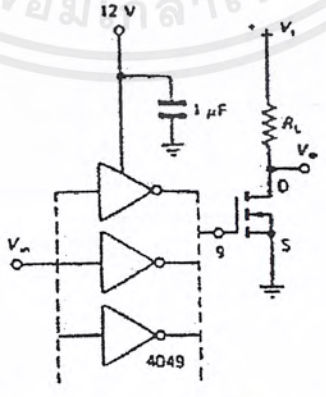
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปจะแสดงตารางค่าความจริงของในการควบคุมการทำงานของมอเตอร์ ดังตารางที่ 4.1

PWM	INTO	INT1	OUT1	OUT2	OUT3	OUT4	*
0	0	0	0	0	0	0	-
0	0	1	0	0	0	0	-
0	1	0	0	0	0	1	Forward
0	1	1	0	0	1	0	Reward
1	0	0	0	0	0	1	-
1	0	1	0	0	1	0	-
1	1	0	1	0	0	1	Forward
1	1	1	0	1	1	0	Reward

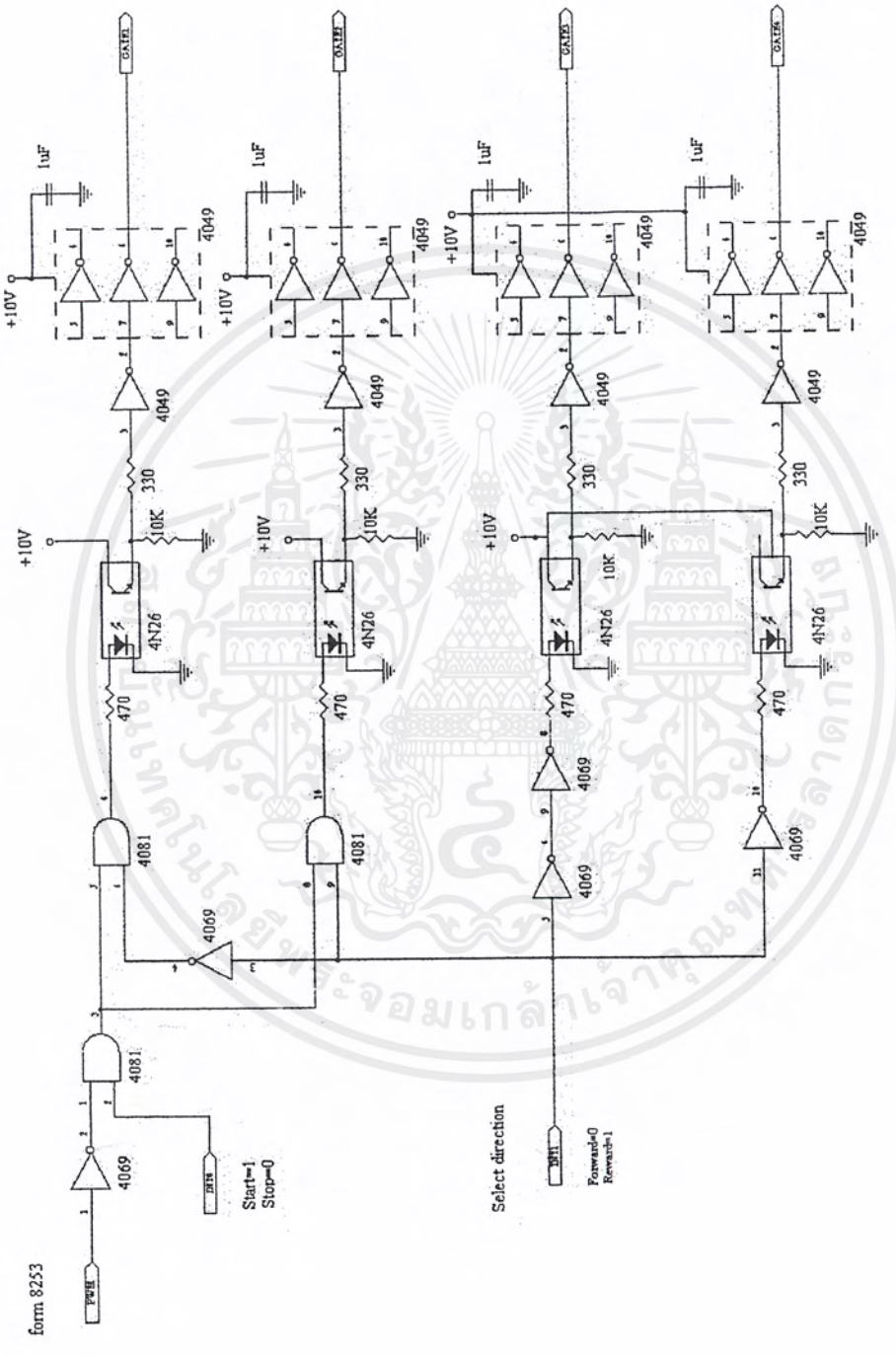
ตารางที่ 4.1 แสดงค่าความจริงในการควบคุมมอเตอร์

ส่วนที่ใช้ในการขับเกทของเพาเวอร์มอสเฟต จะใช้ออปโตคัปเปิล (opto – couple) ในการแยกกราวน์ระหว่างส่วนควบคุม กับส่วนวงจรกำลังของมอเตอร์เพื่อป้องกันการรบกวน และเป็นอันตรายจากไฟแรงดันสูง โดยปกติแล้ววงจรที่ใช้ในการขับอุปกรณ์สวิทชิ่งในวงจรกำลังซึ่งจะต้องใช้ทั้งหมด 4 ชุดและแต่ละชุดต้องแยกกราวน์กัน และจะต้องจ่ายกระแสได้เพียงพอที่อุปกรณ์สวิทชิ่งทำงานได้ในส่วนวงจรของโครงงานนี้ได้ ใช้ CMOS 4049 ซึ่งเป็น inverting ซึ่งภายในมี inverting อยู่6 ตัว นำมาต่อขนานกันเพื่อที่ขยายกระแสในการขับเกทมอสเฟต ให้มอสเฟตสามารถทำงานได้



รูปที่ 4.4 แสดงวงจรที่ใช้ขับมอสเฟต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

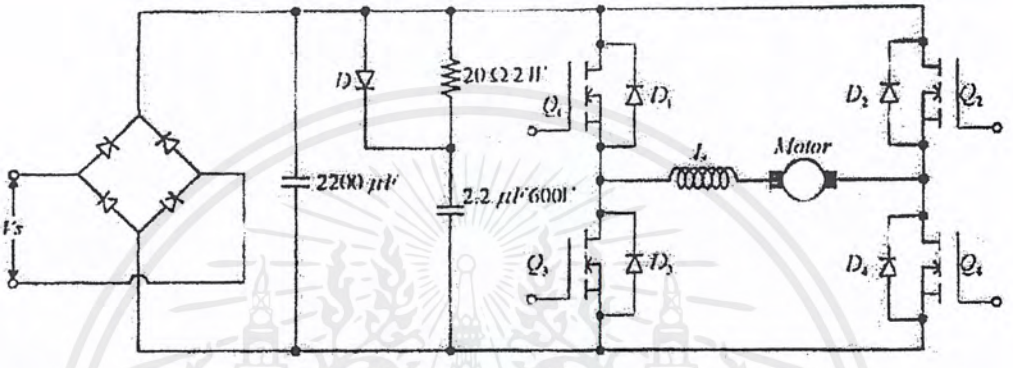


รูปที่ 4.5 แสดงวงจรที่ใช้ในโครงงานในการขับมอสเฟต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 วงจรขับเคลื่อนมอเตอร์

โดยจะใช้เพาเวอร์มอสเฟตเป็นอุปกรณ์สวิตช์ชิ่ง โดยที่วางตัวแบบ H-bridge และมีวงจรสนับเบอร์ต่อคร่อมที่ DC. Link เพื่อป้องกันเพาเวอร์มอสเฟตในขณะ Turn Off ดังแสดงในรูปที่ 4.6



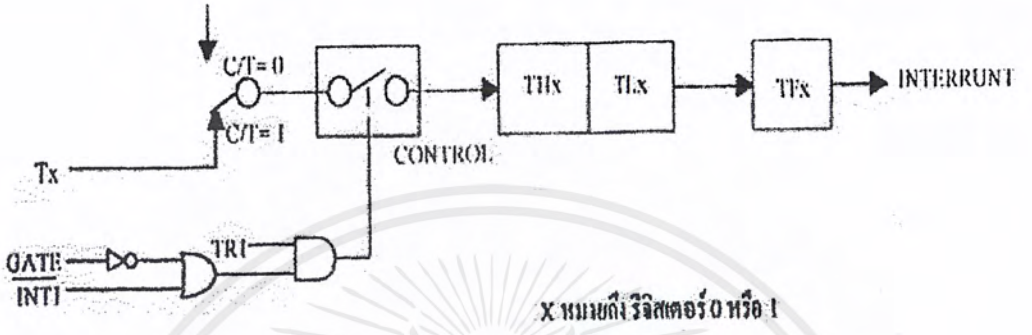
รูปที่ 4.6 แสดงวงจรที่ใช้ในการขับมอเตอร์

4.5 การตรวจจับตำแหน่ง

การตรวจจับตำแหน่งจะใช้ไทม์เมอร์/เคาน์เตอร์ ในไมโครคอนโทรลเลอร์เป็นตัวตรวจจับ ซึ่งในไมโครคอนโทรลเลอร์จะมีไทม์เมอร์/เคาน์เตอร์อยู่ 2 ชุด คือไทม์เมอร์/เคาน์เตอร์ 0 และ 1ในแต่ละชุดจะมีรีจีสเตอร์ขนาด 8 บิต อยู่ 2 ตัว เพื่อเก็บค่าการนับของไทม์เมอร์/เคาน์เตอร์ ได้สูงสุด 16 บิต ในไทม์เมอร์/เคาน์เตอร์ 0 และ 1 มีรีจีสเตอร์ คือ TH0,TL0 และ TH1,TL1 ตามลำดับ โดย TH0,TH1 จะเก็บค่าของการนับ 8 บิตบน ส่วนTL0,TL1 จะเก็บค่าของการนับ 8 บิตล่าง ซึ่งจะสามารถควบคุมการทำงานของไทม์เมอร์/เคาน์เตอร์ ในโหมดไทม์เมอร์ หรือโหมดของเคาน์เตอร์ได้ โดยการกำหนดในรีจีสเตอร์ชื่อ TMOD (Timer/Counter Mode Control Register) ซึ่งโครงงานนี้จะใช้ โหมดเคาน์เตอร์โดยที่โหมดเคาน์เตอร์จะถูกเพิ่มค่าทีละหนึ่ง เมื่อป้อนสัญญาณคล็อคจากภายนอกเข้ามาหนึ่งลูก (เข้ามาทางขา T0 หรือ T1 ของไมโครคอนโทรลเลอร์ เมื่อเคาน์เตอร์นับครบจะทำการเซ็ท TFx ภายในรีจีสเตอร์ TCON (Timer/Counter Control Register) เพื่อแสดงการเกิดโอเวอร์โฟลว์ของเคาน์เตอร์ และจะถูกเคลียร์เองเมื่อ CPU ย้ายไปที่โปรแกรมบริการอินเตอร์รัพท์ หรือใช้คำสั่ง CLR TFx (x หมายถึง 0 หรือ 1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมดการทำงานของเคาน์เตอร์ จะมีด้วยกัน 4 โหมดซึ่งในที่นี้จะเลือกใช้ในโหมดที่ 1 (เป็นโหมด 16 บิต)



รูปที่ 4.7 แสดงบล็อกไดอะแกรมการทำงานเคาน์เตอร์ (Mode 1)

4.6 ส่วนของโปรแกรมการทำงาน

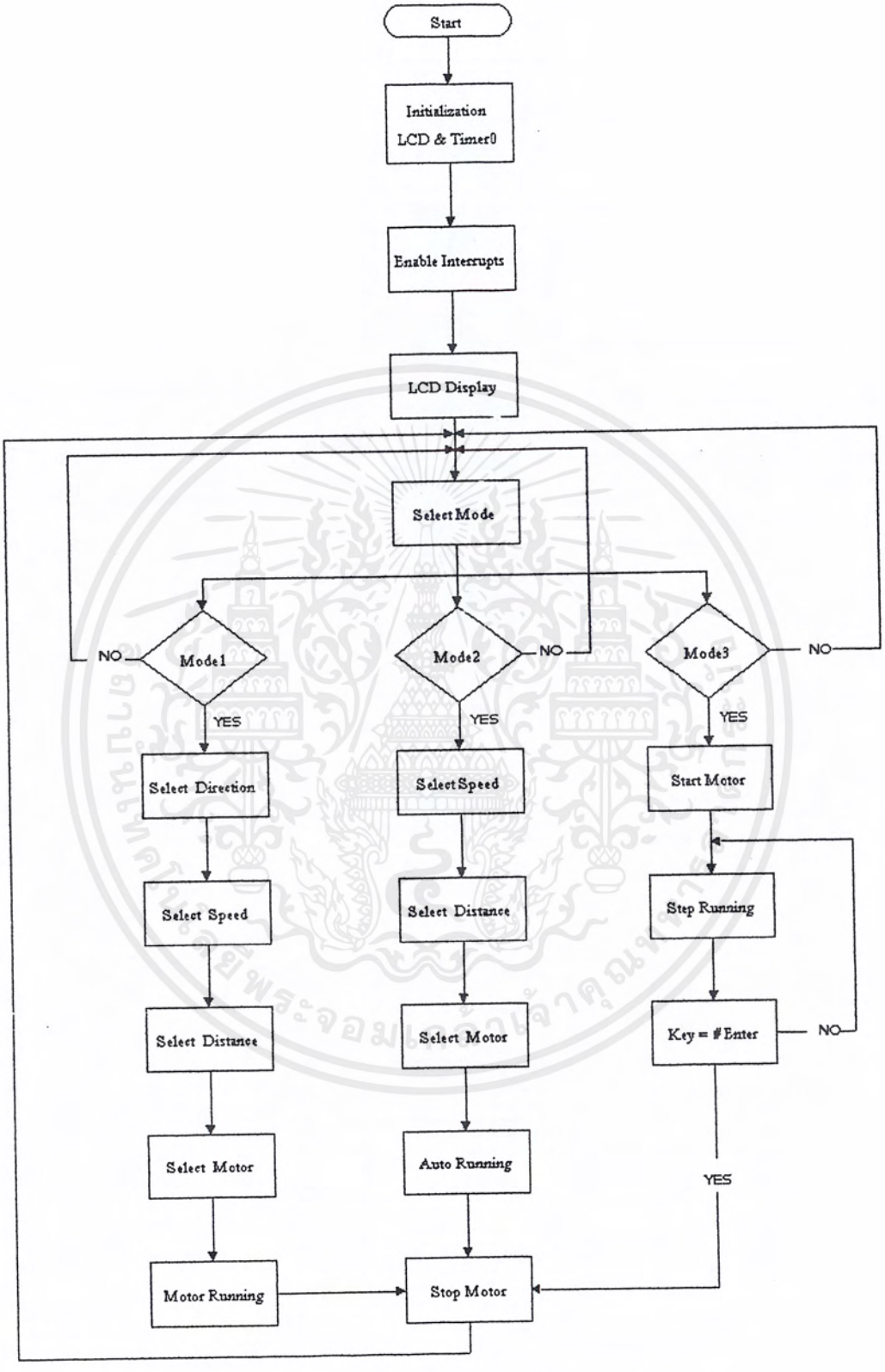
ในส่วนของโปรแกรมหลักที่ใช้ในโครงงานจะแสดงด้วยโฟล์วชาร์ท ดังรูปที่ 4.8 โดยเริ่มต้นจากการ Innitial คือการ Set ค่าเริ่มต้นของโปรแกรม และตามด้วย LCD แสดงผลของทางหน้าจอ จากนั้นจะมีการเลือกโหมดการทำงาน ซึ่งมีด้วยกันอยู่ 3 โหมด โดยที่การทำงานแต่ละโหมดมีดังนี้

โหมด 1 จะเป็นการสั่งให้มอเตอร์ทำงานในทิศทางเดียว ด้วยการเลือกทิศทาง (Forward,Reward) เลือกความเร็ว ซึ่งมี 3 ระดับ จากนั้นก็เลือกระยะทางตั้งแต่ 5-50 cm.

โหมด 2 จะเป็นการสั่งให้มอเตอร์ทำงาน ไป-กลับ โดยมีการเลือก ความเร็ว และระยะทาง

โหมด 3 จะเป็นการทำให้มอเตอร์หมุนเป็น Step โดยทำทีละ 10 cm ไปเรื่อยจนกว่าจะสั่งหยุดการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

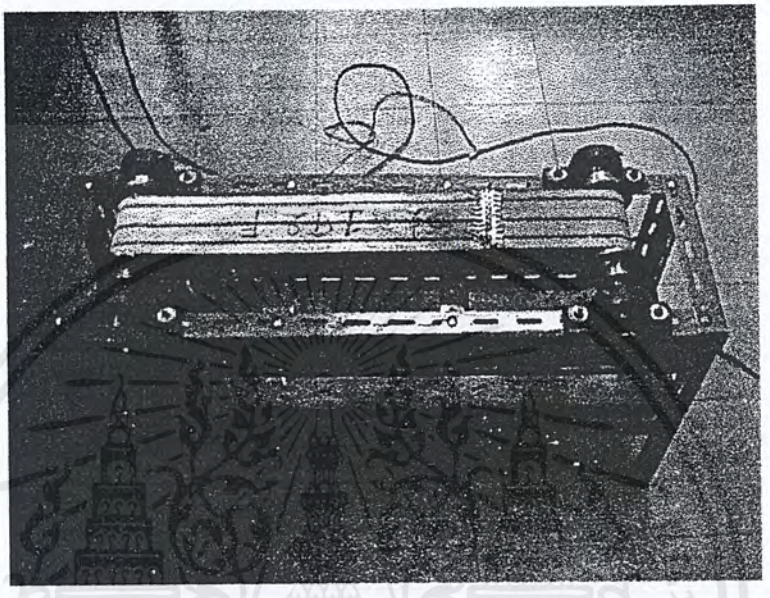


รูปที่ 4.8 แสดงโฟล์วชาร์ทการทำงานของโปรแกรมหลัก

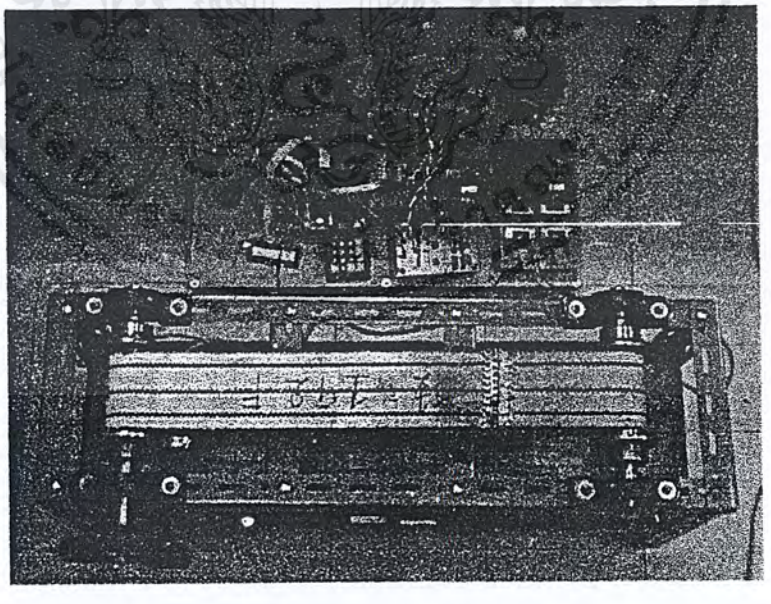
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 ชุดสายพานลำเลียง

เป็นการจำลองการทำงานของชุดสายพานโดยให้มีความยาวของสายพานประมาณ 50 cm และทำการสร้างชุดสายพานขึ้นมา ซึ่งแสดงดังรูปที่ 4.9 และ 4.10



รูปที่ 4.9 แสดงชุดสายพานลำเลียงที่ใช้ในโครงงาน



รูปที่ 4.10 แสดงภาพของโครงงานทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การทดสอบโครงงาน

5.1 การทดสอบหาค่าพารามิเตอร์ต่างๆ ของมอเตอร์

อุปกรณ์ที่ใช้

1. ดีซีเซอร์โวมอเตอร์ แบบสนามแม่เหล็กคงที่ (Tamagawa) พิกัด 80 W , 31 V , 3.3A ,ความเร็วรอบ 4,600 rpm ,เอ็นโค้ดเดอร์ แบบอินคริเมนทอล 432 c/t , แรงบิด 2 kg.cm.
2. แหล่งจ่ายไฟตรง 0 – 30 V ,1 A.
3. แอมป์มิเตอร์กระแสตรง 3A , โวลต์โอห์มมิเตอร์
4. ทาโคมิเตอร์ (Tachometer)
5. ออสซิลโลสโครบ (Oscilloscope)

วิธีการทดสอบ

1. ต่อแหล่งจ่ายไฟและแอมป์มิเตอร์เข้ากับมอเตอร์
2. ค่อยๆป้อนแรงดันไฟฟ้าในระดับต่างๆให้กับมอเตอร์ แล้วอ่านค่ากระแส และความเร็วรอบ พร้อมทั้งบันทึกค่า
3. วัดค่าความต้านทานของมอเตอร์ โดยใช้มัลติมิเตอร์วัดที่ขั้วของมอเตอร์

ผลการทดสอบ

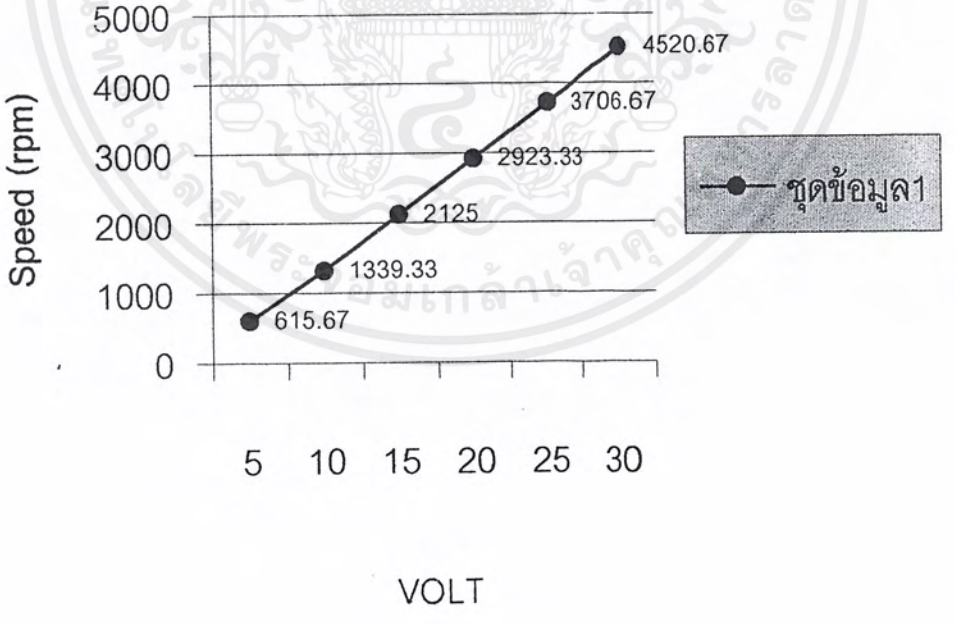
- สามารถวัดความต้านทานของมอเตอร์ที่ขั้วของมอเตอร์โดยใช้มัลติมิเตอร์ได้ประมาณ 2.5 โอห์ม
- เมื่อทำการป้อนแรงดันไฟฟ้าให้กับมอเตอร์ และทำการบันทึกค่ากระแสและความเร็วรอบ พร้อมทั้งคำนวณค่าพารามิเตอร์ต่างๆของมอเตอร์ ดังตารางที่ 5.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Vin (Volt)	I (Amp)	ความเร็วรอบ (รอบต่อนาที)			
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย
5	0.20	610	618	619	615.67
10	0.28	1340	1368	1310	1339.33
15	0.30	2123	2132	2120	2125
20	0.32	2925	2920	2925	2923.33
25	0.35	2700	3712	3708	3706.67
30	0.39	4562	4500	4500	4520.67

ตารางที่ 5.1 แสดงการบันทึกค่าความเร็วรอบ และกระแส

- จากตารางสามารถนำค่าในตารางมาแสดงความสัมพันธ์ระหว่างแรงดันและความเร็วรอบด้วยกราฟ ดังแสดงรูปที่ 5.1



รูปที่ 5.1 แสดงกราฟความสัมพันธ์ระหว่างแรงดันและความเร็วรอบของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จากนั้นนำค่าที่วัดได้มาคำนวณหาค่า Km ซึ่งเป็นค่าคงที่ ที่ใช้ในการออกแบบ

ตัวอย่างการคำนวณ

$E_g = V_t - I_a R_a$ $= 15 - (0.3)(2.5) = 14.25$ V.

$\omega = N \cdot 2\pi/60$ $= 2125 * 2\pi/60 = 222.53$ rps

$K_m = E_g/\omega$ $= 14.25/222.53 = 0.064036$ v-s/rad

Vin (Volt)	ω (rps)	Km (V-s/rad)	Eg (Volt)
5	64.47	0.0698	4.50
10	140.25	0.06631	9.30
15	222.53	0.064036	14.25
20	306.13	0.062718	19.20
25	388.16	0.062165	24.13
30	473.4	0.061322	29.03

ตารางที่ 5.2 แสดงค่าพารามิเตอร์ของมอเตอร์ที่ได้

ดังนั้น หาค่าเฉลี่ยได้

Km (เฉลี่ย) = (0.0698+0.06631+0.064036+0.062718+0.062165+0.061322)/6

= 0.064392 v-s/rad

- ทำการต่อวงจรดังรูปที่ 4.2 เพื่อทดสอบหาค่า Inductance ของมอเตอร์

กำหนดให้

Vi = 30 V

Rx = 0.47 Ω, 5 W

Ra = ค่าความต้านทานของมอเตอร์

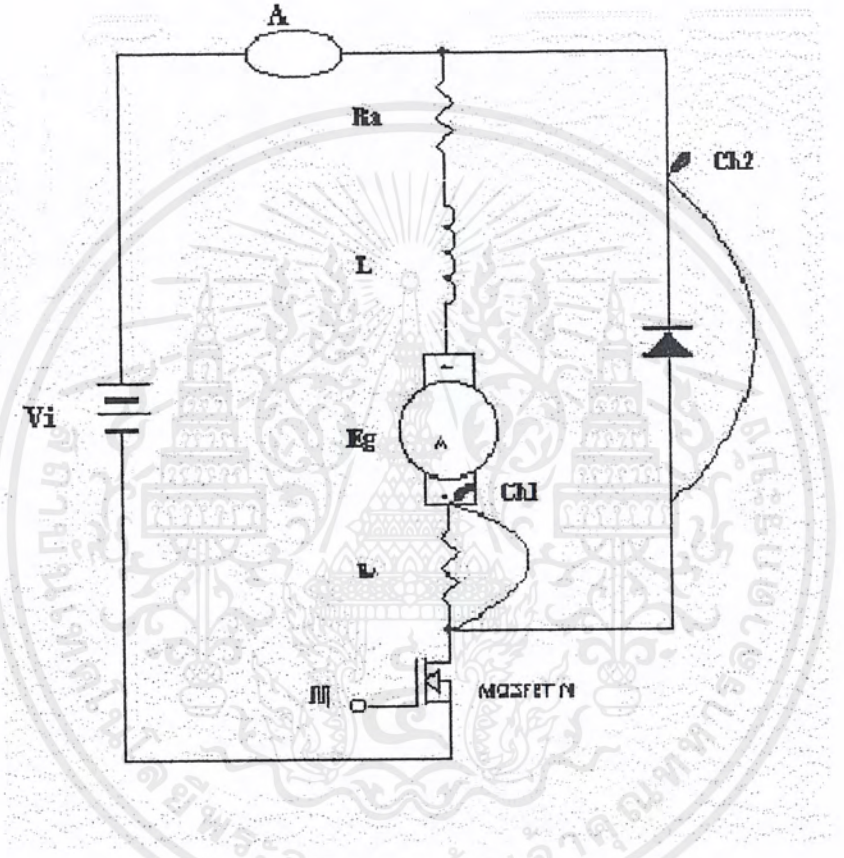
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

L = ค่า Inductance ของมอเตอร์

Ch1 = 20 mV/div*10, Ch2 = 2 V/div*10, Time/div = 10 μs/div

Pulse generator = 0-5 V, duty cycle 50% ความถี่ 20 kHz

Vsw คือ แรงดันที่ตกคร่อม มอสเฟต ขณะ ON ≈ 0 V



รูปที่ 5.2 แสดงวงจรการทดสอบค่า Inductance ของมอเตอร์

Vi (Volt)	Eg (Volt)	Ch1		Ch2	Vsw	Vt	L (mH)
		Imax (A)	Imin (A)	(peak)			
30	14.24	0.425	0.085	30	0	15	1.26

ตารางที่ 5.3 แสดงค่า Inductance ของมอเตอร์ที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณหาค่า Inductance หาได้จากสมการ

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

แทนค่าต่างๆ ลงไปในสมการ แล้วคำนวณหาค่า L_a ออกมาจะได้

$$L_a = 1.26\ \text{mH}$$

สรุปผลการทดลอง

จากการทดลองสามารถทำการหาค่าพารามิเตอร์ต่างๆของมอเตอร์ได้ตามต้องการ และจากการหาค่า L_a และค่า R_a ทำให้สามารถตัดค่ากระแสริปเปิลทิ้งได้จากสมการ $T < L_a/R_a$ ซึ่งความถี่ที่ใช้ในโครงงานนี้ จะใช้ความถี่ 20 kHz จะได้ว่า 50 μs < 504 μs ซึ่งที่ความถี่นี้กระแสยังคงมีความต่อเนื่องอยู่ เมื่อค่าDuty cycle เปลี่ยนไป และมอเตอร์ยังคงหมุนได้อย่างราบเรียบ

5.2 การทดสอบสัญญาณพัลซ์จากเอ็นโค้ดเดอร์

อุปกรณ์ที่ใช้

1. มอเตอร์ที่ใช้ในโครงงาน
2. แหล่งจ่ายไฟ 0-30V และ 5 V
3. Oscilloscope
4. โวลต์มิเตอร์

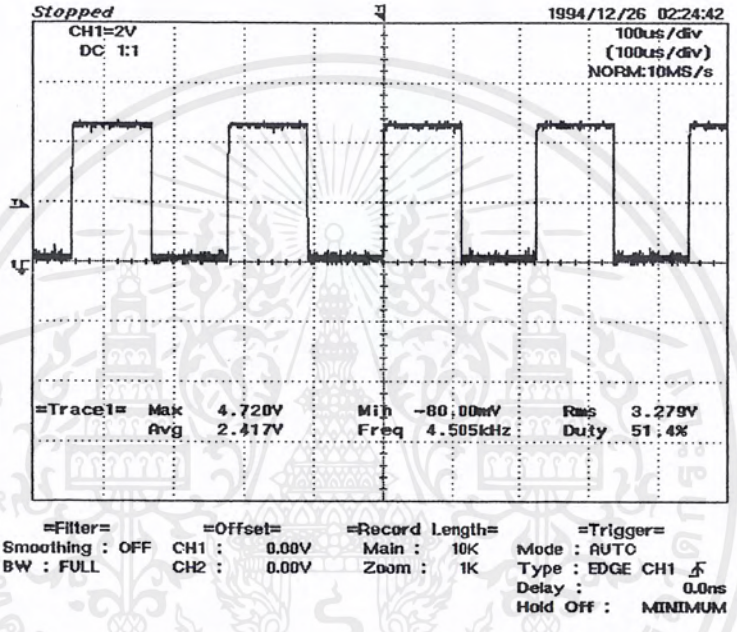
วิธีการทดสอบ

ต่อแหล่งจ่ายไฟ 5V เข้ากับเอ็นโค้ดเดอร์ที่ตัวมอเตอร์ และจ่ายไฟที่มีแรงดันน้อยๆ ให้กับมอเตอร์ และทำการวัดสัญญาณพัลซ์ที่ออกมาจากเอ็นโค้ดเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบ

เนื่องจากปัญหาที่มอเตอร์มีสภาพไม่สมบูรณ์ ทำให้สัญญาณพัลซ์ที่ออกมาจากเอ็นโค้เดอร์นั้นออกมามีเพียงเฟสเดียว และสัญญาณที่ได้มีระดับแรงดันต่ำเกินไปไม่สามารถที่จะนำไปใช้งานได้ จึงได้ต่อ ค่าความต้านทานเข้าไปเพื่อ Pull up สัญญาณ ให้มีระดับแรงดัน 5V ดังแสดงในรูปที่ 5.3



รูปที่ 5.3 แสดงสัญญาณพัลซ์ที่ได้จากเอ็นโค้ดเดอร์

สรุปผลการทดลอง

จากรูปสัญญาณที่ได้จะเห็นว่ามีระดับแรงดันเพียงพอที่จะนำไปใช้งาน (0-5V) โดยจะนำสัญญาณนี้ไปผ่านบัฟเฟอร์ก่อนที่จะเข้าไมโครคอนโทรลเลอร์โดยการนำไปใช้จะนำไปต่อเข้ากับขา Timer 0โดยให้ Timer 0 ทำงานเป็น Counter นับสัญญาณพัลซ์ที่มาจากเอนโค้ดเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 การทดสอบการสร้างสัญญาณพัลซ์วิดท์มอดูเลชั่น (PWM)

อุปกรณ์ที่ใช้

1. คอมพิวเตอร์
2. ชุดไมโครคอนโทรลเลอร์ กับ ชุดสร้างPWM
3. Oscilloscope

วิธีการทดสอบ

การทดลองโดยการเขียนโปรแกรมให้กับไมโครคอนโทรลเลอร์ไปควบคุม 8253 ให้สร้างสัญญาณพัลซ์ แล้วทำการจับสัญญาณที่ออกมาจาก Out 0 ของ Counter0 และ Out1 ของ Counter 1 จากตัว 8253 ดังแสดงในรูปที่ 5.4 และส่วนของโปรแกรมจะแสดง ดังนี้

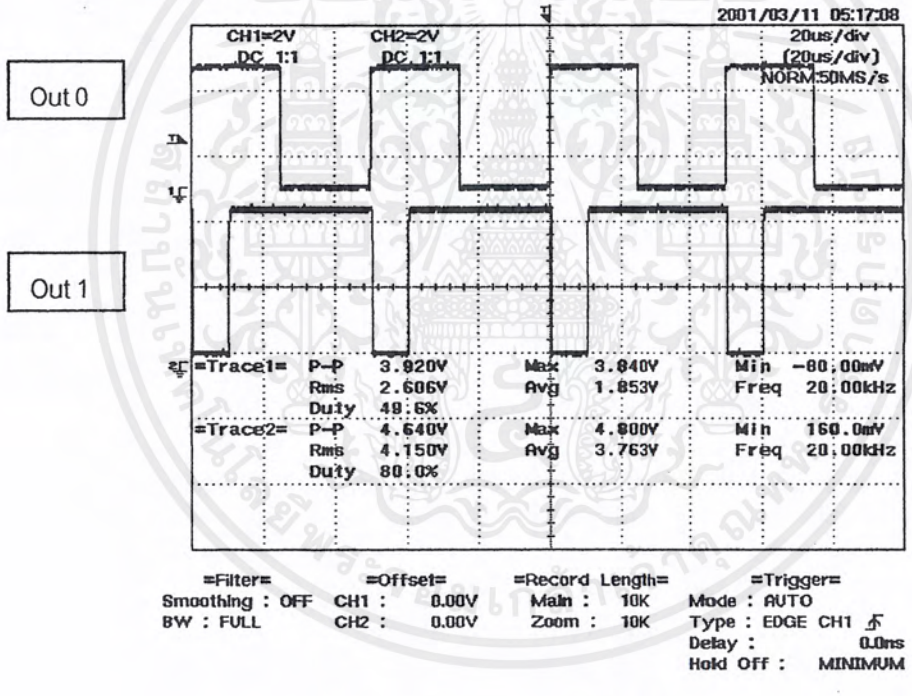
โปรแกรมทดสอบการสร้างสัญญาณพัลซ์วิดท์มอดูเลชั่น โดยใช้ 8253 ดังนี้

```
                ORG   0000H
CHN0            EQU   0FC00H
CHN1            EQU   0FC01H
CTRL            EQU   0FC03H

                MOV   A,#37H
                MOV   DPTR,#CTRL
                MOV   @DPTR,A       ;SQURE WAVE MODE
                MOV   DPTR,CHN0
                MOV   A,#00H
                MOVX  @DPTR,A
                MOV   A,#01H
                MOVX  @DPTR,A       ;20 kHz SQUARE WAVE
                MOV   A,#73H
                MOV   DPTR,#CTRL
                MOVX  @DPTR,A       ;MONOSATABLE MODE
                MOV   DPTR,#CHN1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOVX @DPTR,A
MOV  A,#50H
MOVX @DPTR,A
MOV  A,#00H
MOVX @DPTR,A

END
```

ผลการทดสอบ จากการทดสอบเราสามารถได้สัญญาณ ดังรูปที่ 5.4



รูปที่ 5.4 แสดงสัญญาณพัลซ์วิดท์มอดูเลชั่น

สรุปผลการทดลอง

จากโปรแกรมเราสามารถที่จะสร้างสัญญาณพัลซ์วิดท์มอดูเลชั่นได้ ซึ่งสามารถสร้างสัญญาณให้มีค่า ดิวตี้ไซเคิล ตั้งแต่ 0-100 % ได้ตามต้องการ ซึ่งจากการทดลอง โดยการสร้างสัญญาณให้มีค่า ดิวตี้ไซเคิล 80% และมีความถี่ 20kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 การทดสอบสัญญาณขับเกท และ การทดสอบภาคขับมอเตอร์

อุปกรณ์ที่ใช้

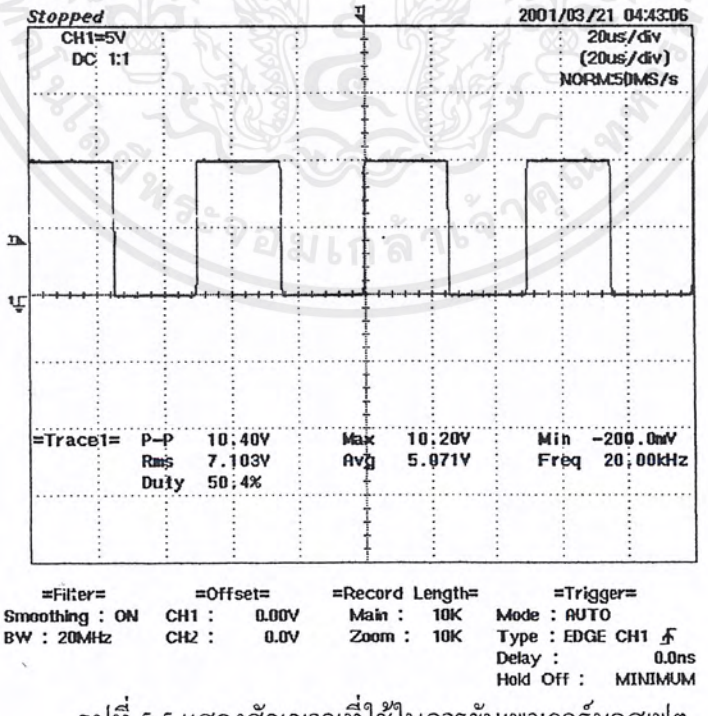
1. คอมพิวเตอร์
2. แผงวงจรที่ใช้
3. ชุดขับเคลื่อนสายพาน
4. Oscilloscope

วิธีการทดสอบ

ต่อชุดไมโครคอนโทรลเลอร์ผ่านอีพรอมอีมูเลเตอร์ และต่อภาคขับมอเตอร์เข้ากับชุดสร้างสัญญาณพัลซ์วิดมอดูเลชั่น และต่อชุดขับเคลื่อนสายพานเข้ากับแผงวงจรแล้วทำการวัดรูปคลื่นสัญญาณที่ใช้ในการขับเกทและสัญญาณของแรงดันที่จ่ายให้กับมอเตอร์โดยวัดคร่อมที่ขั้วของมอเตอร์

ผลการทดสอบ

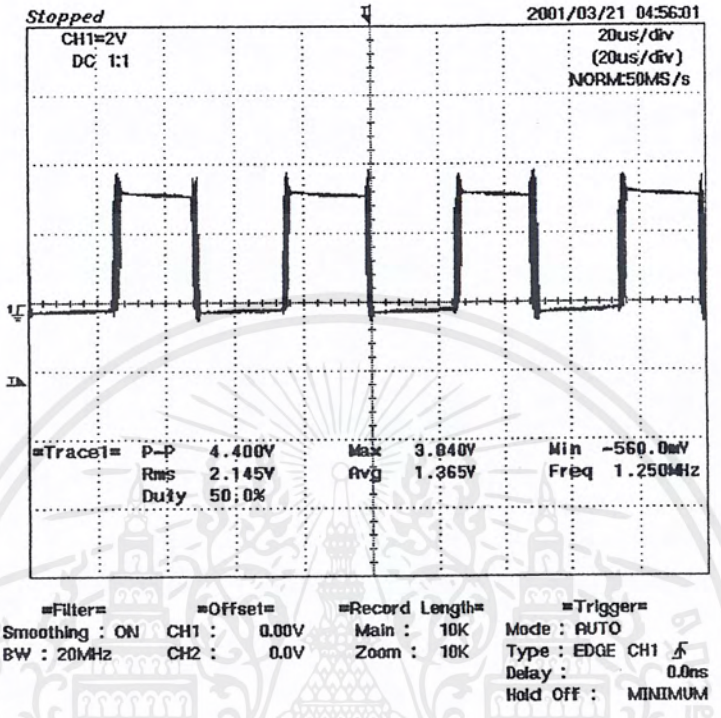
5.4.1 ทำการวัดสัญญาณที่ใช้ขับเกทของมอสเฟต โดยการสั่งให้สร้างสัญญาณขับที่มีค่า Duty cycle 50% ซึ่งสัญญาณที่ได้จะออกมาดังรูปที่ 5.5



รูปที่ 5.5 แสดงสัญญาณที่ใช้ในการขับเพาเวอร์มอสเฟต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.2 ทำการวัดแรงดันที่จ่ายให้กับมอเตอร์ ดังรูปที่ 5.6



รูปที่ 5.6 แสดงสัญญาณที่จ่ายให้กับตัวมอเตอร์

สรุปผลการทดลอง

หลักการที่ใช้ในการขับมอเตอร์สามารถที่จะทำให้ขับมอเตอร์ได้ทั้ง 2 ทิศทาง (Forward,Reward) ได้เป็นอย่างดี โดยให้Mosfet คู่ใดคู่หนึ่งทำงานสวิตท์ชิ่งตลอดเวลา ในการหมุนไปทิศทางหนึ่ง และ สัญญาณที่ได้จากชุดขับเกทของมอสเฟต ซึ่งมีค่า 10 Vก็ สามารถที่จะขับให้มอสเฟตทำงานได้ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 การทดสอบโครงงานทั้งหมด

อุปกรณ์ที่ใช้

1. แผงวงจรชุดขับและชุดขับเคลื่อนสายพาน
2. คอมพิวเตอร์
3. โหลดน้ำหนัก
4. ตลับเมตร

วิธีการทดสอบ

ทำการโหลดโปรแกรมจากคอมพิวเตอร์โดยผ่านอีพรอมอีมูเลเตอร์ และสั่งให้มอเตอร์ทำงานที่ความเร็ว และระยะทางต่างๆ เมื่อ อยู่ในสภาวะ No-Load และสภาวะที่มี Load พร้อมทั้งบันทึกค่า

ผลการทดสอบ

ตารางที่ 5.4 แสดงการทดสอบขณะ NO LOAD ที่ความเร็วสายพาน 3 เมตรต่อนาที

ระยะทางมาตรฐาน (cm)	ผลการทดลองครั้งที่			ระยะทางเฉลี่ย (cm)	ค่าความคลาดเคลื่อน (%)
	1	2	3		
5	5	5.05	5.05	5.03	0.6
10	10	10	10	10	0
15	14.95	15	15.05	15	0
20	20.1	20	19.8	19.97	-0.15
25	25	25.1	25.1	25.07	0.28
30	30.02	30	30.1	30.04	0.13
35	35.05	35	35	35.02	0.7
40	40	40	40.05	40.02	0.05
45	45.05	44.92	45.05	45.01	0.02
50	50	50.05	50.05	50.03	0.06

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.5 แสดงการทดสอบขณะ NO LOAD ที่ความเร็วสายพาน 6 เมตรต่อนาที

ระยะทางมาตรฐาน (cm)	ผลการทดลองครั้งที่			ระยะทางเฉลี่ย (cm)	ค่าความคลาดเคลื่อน (%)
	1	2	3		
5	5	5	5	5	0
10	10	10	10	10	0
15	14.95	14.9	15	14.95	-0.33
20	19.95	20	20	19.98	-0.1
25	24.95	24.8	25	24.92	-0.32
30	29.95	29.85	30	29.93	-0.23
35	35	34.85	34.85	34.9	-0.03
40	39.9	39.95	39.95	39.93	-0.18
45	44.9	45	44.9	44.93	-0.16
50	50	50	49.95	49.98	-0.04

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.6 แสดงการทดสอบขณะ NO LOAD ที่ความเร็วสายพาน 9 เมตรต่อนาที

ระยะทางมาตรฐาน (cm)	ผลการทดลองครั้งที่			ระยะทางเฉลี่ย (cm)	ค่าความคลาดเคลื่อน (%)
	1	2	3		
5	5	5	5	5	0
10	10	9.9	9.9	9.93	-0.7
15	15	14.85	15.1	14.98	-0.13
20	19.85	19.9	19.85	19.87	-0.65
25	25	24.85	25	24.95	-0.2
30	30	29.95	29.9	29.95	-0.17
35	35	35.05	34.9	34.98	-0.06
40	39.95	40	39.85	39.93	-0.18
45	44.9	44.9	44.8	44.87	-0.29
50	50	50	49.8	49.93	-0.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.7 แสดงการทดสอบขณะขับ LOAD ขนาด 0.5 kg ที่ความเร็วสายพาน 3 เมตรต่อนาที

ระยะทาง มาตรฐาน (cm)	ผลการทดลองครั้งที่			ระยะทางเฉลี่ย (cm)	ค่าความคลาดเคลื่อน (%)
	1	2	3		
5	5.1	5	5.1	5.07	1.4
10	10	10.1	10.05	10.05	0.5
15	14.96	14.92	15	14.96	-0.27
20	20	19.95	20.1	20.02	0.1
25	25	25.1	24.92	25.01	0.04
30	30.02	30	29.95	29.99	-0.03
35	34.9	35	35.05	34.98	-0.06
40	39.9	40	40.1	40	0
45	44.8	45	45.1	44.97	-0.07
50	49.9	50.1	49.8	49.93	-0.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.8 แสดงการทดสอบขณะขับ LOAD ขนาด 0.5 kg ที่ความเร็วสายพาน 6 เมตรต่อนาที

ระยะทาง มาตรฐาน (cm)	ผลการทดลองครั้งที่			ระยะทางเฉลี่ย (cm)	ค่าความคลาดเคลื่อน (%)
	1	2	3		
5	5	5	5	5	0
10	9.95	9.9	9.92	9.92	-0.8
15	14.93	15	14.95	14.96	-0.27
20	20	20	49.95	19.98	-0.1
25	24.9	25	24.93	24.94	-0.24
30	30	30	29.92	29.97	-0.1
35	35	34.9	34.95	34.95	-0.14
40	40	39.9	40	39.97	-0.08
45	45	44.9	45	44.97	-0.07
50	49.95	49.9	49.95	49.93	-0.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.9 แสดงการทดสอบขณะขับ LOAD ขนาด 0.5 kg ที่ความเร็วสายพาน 9 เมตรต่อนาที

ระยะทางมาตรฐาน (cm)	ผลการทดลองครั้งที่			ระยะทางเฉลี่ย (cm)	ค่าความคลาดเคลื่อน (%)
	1	2	3		
5	5	5	5	5	0
10	9.9	9.9	9.9	9.9	-1
15	14.9	15	14.8	14.9	-0.67
20	19.85	20	19.9	19.92	-0.4
25	25	24.9	24.8	24.9	-0.4
30	29.9	30	29.8	29.9	-0.33
35	35	34.95	35.05	35	0
40	39.85	39.8	40	39.88	-0.3
45	44.9	44.85	45	44.92	-0.18
50	49.8	50	50	49.93	-0.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.10 แสดงการทดสอบขณะขับ LOAD ขนาด 1 kg ที่ความเร็วสายพาน 3 เมตรต่อนาที

ระยะทางมาตรฐาน (cm)	ผลการทดลองครั้งที่			ระยะทางเฉลี่ย (cm)	ค่าความคลาดเคลื่อน (%)
	1	2	3		
5	5	5.05	5	5.02	0.4
10	9.9	10	10	9.97	-0.3
15	15	15	15.1	15.03	0.2
20	19.9	20	20	19.97	-0.15
25	25.1	25.15	25.1	25.12	0.48
30	30	30	35.05	30.02	0.07
35	35.1	35	35	35.03	0.09
40	40	40	40	40	0
45	44.95	45	45.2	45.05	0.11
50	50	50	50	50	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.11 แสดงการทดสอบขณะขับ LOAD ขนาด 1 kg ที่ความเร็วสายพาน 6 เมตรต่อนาที

ระยะทาง มาตรฐาน (cm)	ผลการทดลองครั้งที่			ระยะทางเฉลี่ย (cm)	ค่าความคลาดเคลื่อน (%)
	1	2	3		
5	5	5	5	5	0
10	9.95	10	9.9	9.95	-0.5
15	15	15	14.85	14.95	-0.33
20	20	19.85	20	19.95	-0.25
25	25	25	24.9	24.97	-0.12
30	29.9	30	30.1	30	0
35	35	35	35	35	0
40	40	40.05	39.9	39.98	-0.05
45	45	45	45	45	0
50	49.9	50	50.05	49.98	-0.04

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

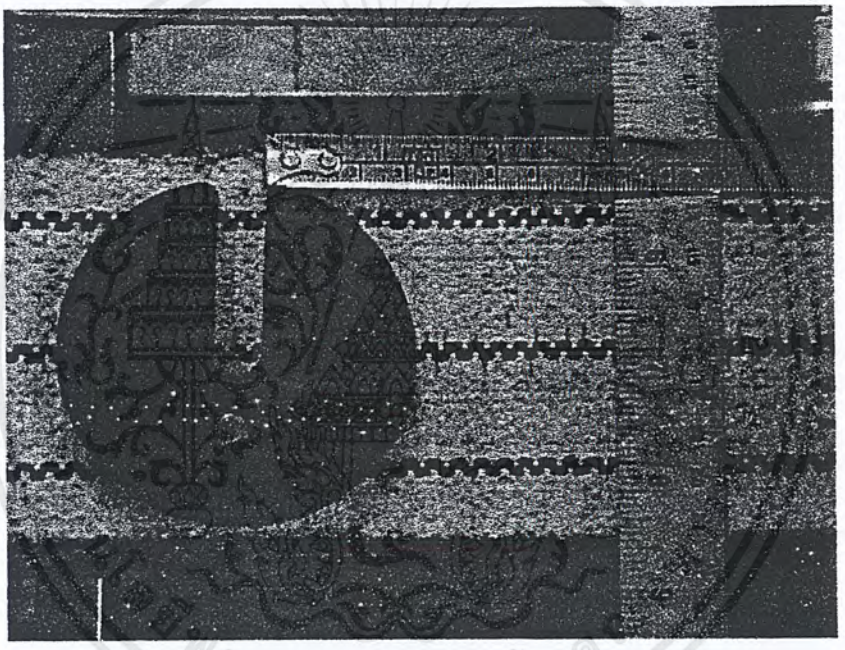
ตารางที่ 5.12 แสดงการทดสอบขณะขับ LOAD ขนาด 1 kg ที่ความเร็วสายพาน 9 เมตรต่อนาที

ระยะทางมาตรฐาน (cm)	ผลการทดลองครั้งที่			ระยะทางเฉลี่ย (cm)	ค่าความคลาดเคลื่อน (%)
	1	2	3		
5	5	5	5	5	0
10	9.9	9.95	9.9	9.92	-0.8
15	15	15.1	14.85	14.98	-0.13
20	19.85	20	20	19.95	-0.25
25	25	24.9	24.85	24.92	-0.32
30	30	30	30.1	30.03	0.1
35	34.8	35	34.85	34.88	-0.34
40	39.85	39.8	39.85	39.83	-0.43
45	45	44.9	45	44.97	-0.07
50	49.8	50.1	49.9	49.93	-0.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

จากการทดสอบการทำงานของโครงงานทั้งที่ไม่มีโหลด และเมื่อมีโหลดจะเห็นว่าการควบคุมตำแหน่งนั้นมีความแม่นยำมากพอควรทีเดียว แต่ค่าผิดพลาดที่เกิดขึ้นในการทดสอบที่ระยะทางและความเร็วเดียวกันอาจมีค่าไม่เท่ากัน อันเนื่องมาจากสาเหตุต่างๆ เช่น ช่วงตะเข็บรอยต่อของสายพานทำให้ขณะหมุนอาจมีการกระตุกได้ ความฝืดของสายพาน ค่าผิดผลาดของโปรแกรมในการสั่งเบรคมอเตอร์ เป็นต้น ซึ่งในการสั่งเบรคของมอเตอร์นั้นจะสร้างทอร์คในทิศทางตรงกันข้ามเพื่อหักล้างกับทอร์คเดิมในช่วงเวลาหนึ่ง



รูปที่ 5.7 แสดงวิธีการวัดระยะทางในขณะทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปผลและแนวทางในการพัฒนา

1. ส่วนของการสร้างสัญญาณพัลซ์วิดท์มอดูเลชั่น สามารถที่จะกำหนด DUTY CYCLE ต่างๆ ได้ โดยการโปรแกรมค่าการนับให้แก่ 8253 ซึ่งจะทำให้สามารถที่จะเปลี่ยนความถี่ในการขับเคลื่อนของมอเตอร์ได้

2. ส่วนของการควบคุมตำแหน่งที่ระยะต่างๆของแต่ละโหมดการทำงาน ค่าของความคลาดเคลื่อนน้อยมาก เนื่องจากโปรแกรมที่ใช้ควบคุม สามารถที่จะหยุดการเคลื่อนที่ของสายพานได้โดยใช้การกลับทิศแรงดันเพื่อหน่วงการเคลื่อนที่ จึงทำให้ไม่เกิดการไถลออกไปอีกเมื่อถึงระยะทางที่ต้องการ ค่าที่ใช้หน่วงในการเบรกสายพานของแต่ละระดับความเร็วจะไม่เท่ากัน ส่วนความคลาดเคลื่อนที่เกิดขึ้น เนื่องจากชุดสายพานช่วงตะเข็บรอยต่อของสายพาน สายพานมีความอ่อนตัวเมื่อถึงช่วงการหมุนที่แกนหมุนไม่มากนัก จึงทำให้เกิดการส่ายตัวของสะพาน ทำให้ระยะทางที่ต้องการเกิดความคลาดเคลื่อนขึ้น

3. ส่วนของการควบคุมการติดต่อระหว่างผู้ใช้กับส่วนประมวลผลกลางเป็นไปได้อย่างดี เข้าใจได้ง่าย

4. ส่วนของเซอร์โวมอเตอร์ที่มีเอ็นท์โค้ดเดอร์ในตัวมอเตอร์มีความละเอียดสูงจึงให้ผลของการควบคุมตำแหน่งที่แม่นยำ และให้แรงบิดในการขับโหลดได้ดี

5. เนื่องจากเซอร์โวมอเตอร์ที่ใช้ในโครงงานมีชุดเกียร์บล็อกติดอยู่กับตัวของมอเตอร์ทำให้ชุดสายพานมีอัตราการทดความเร็วรอบที่สูง ความเร็วของสายพานจึงไม่เร็วมากนัก

ข้อเสนอแนะสำหรับการปรับปรุงและแนวทางในการพัฒนา

1. ควรเปลี่ยนวิธีการหยุดหมุนมอเตอร์ โดยเมื่อไมโครคอนโทรลเลอร์นับสัญญาณพัลส์จากเอ็นโค้ดเดอร์ถึงตำแหน่งที่ต้องการจะหยุดแล้ว ให้ทำการสลับการจ่ายแรงดันบวกและลบให้กับมอเตอร์ เพื่อทำให้แรงดันไฟฟ้าที่คร่อมมอเตอร์มีค่าเฉลี่ยเป็นศูนย์ มอเตอร์ก็จะหยุดในตำแหน่งที่ต้องการได้อย่างแม่นยำมากยิ่งขึ้น และไม่มีการลื่นไถลเนื่องจากโหลดภายนอกมากระทำ

2. ควรเปลี่ยนรูปแบบการขับมอเตอร์ให้เป็นแบบยูนิโพล่าร์,ไบโพล่าร์ หรือ ลิมิตยูนิโพลาร์ ซึ่งในโครงงานใช้เพียงคู่เดียวในการขับมอเตอร์ไปในทิศทางใดทิศทางหนึ่ง

3. ควรออกแบบชุดขับเคลื่อนสายพานให้มีประสิทธิภาพที่ดีขึ้น ซึ่งจะส่งผลการทำงานที่ดีขึ้น โดยที่ทำให้การเคลื่อนที่ของสายพานนั้นมีความคล่องตัว ความตึงหย่อนของสายพานพอเหมาะ

4. ควรที่จะมีการควบคุมความเร็วและทอร์คด้วยเพื่อการทำงานที่ดียิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;******CHANNEL ADDRESS OF 8253*****
CHN0        EQU   0FC00H
CHN1        EQU   0FC01H
CTRL        EQU   0FC03H

;*****INTERNAL RAM VARIABLE******
KEYBUF          EQU   21H
POS             EQU   28H
COUNT           EQU   51H
DIRECT          EQU   52H   ;'F' OR 'R'
SPEED1          EQU   53H
SPEED2          EQU   54H
IDX_SPEED       EQU   55H
FLAG_DIS        EQU   TF0   ;INTERRUPT TIMER0
WAY             EQU   INT1
EN_MOTOR        EQU   INT0
STARTCOUNT          EQU   TR0   ;TIMER0

;*********************************
                ORG   0000H
ORIGIN:           SJMP   MAIN
;***TIMER0 INTERRUPT********
                ORG   ORIGIN+0BH
                LJMP   TIMER0ISR
MAIN:
;***INITIAL LCD SEGMENT***
                CLR   EN_MOTOR
                CALL   INIT_LCD
                CALL   CLRSCR
                CALL   CURSOR_ON
;***INITIAL TIMER0 SEGMENT***
                MOV   TMOD,#05H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MAIN1:          SETB   EA
                SETB   ET0
MAIN2:          CALL   MAIN_MENU
                SJMP   MAIN2

;*******************************
;MAIN MENU MODE1 OR 2
;*******************************
MAIN_MENU:      CALL   CLRSCR
                CALL   PRINTLCD
                DB     "DC SERVO",00
                MOV    A,#40H
                CALL   GOTOLCD
                CALL   PRINTLCD
                DB     " MOTOR",00
ENTER_MENU:     CALL DELAY
                CALL   SCANKEY
                CJNE   A,#HELP,ENTER_MENU
MAIN_MENU1:     CALL   CLRSCR
                CALL   PRINTLCD
                DB     "SELECT M",00
                MOV    A,#40H
                CALL   GOTOLCD
                CALL   PRINTLCD
                DB     "ODE1,2,3",00
LOOP1_MENU:     CALL   SCANKEY
                CJNE   A,#0FFH,CHECKMODE
                SJMP   LOOP1_MENU
CHECKMODE:      CJNE   A,#1,GOTOMODE2
                CALL   MODE1
                AJMP   MAIN_MENU1
GOTOMODE2:      CJNE   A,#2,GOTOMODE3
                CALL   MODE2
                AJMP   MAIN_MENU1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
GOTOMODE3:      CJNE   A,#3,GOTOERROR
                CALL   MODE3
                AJMP   MAIN_MENU1
GOTOERROR:      SJMP   LOOP1_MENU

;************************************************
;MODE 1  OPERATION FROM USER SELECTION
;************************************************
MODE1:          CALL   CLRSCR
                CALL   PRINTLCD
                DB     "MODE 1 O",00
                MOV    A,#40H
                CALL   GOTOLCD
                CALL   PRINTLCD
                DB     "PERATION",00
M1_ENTER:       CALL   SCANKEY
                CJNE   A,#HELP,M1_ENTER
                MOV    SPEED1,#0
                MOV    SPEED2,#0
                MOV    IDX_SPEED,#0
                CALL   DIRECTION     ;F OR R
                CALL   SPEED
                CALL   READ_STAB
                CALL   LOAD_SPEED    ;LOAD SPEED TO 8253
                CALL   DISTANCE
                CALL   CLRSCR
                CALL   PRINTLCD
                DB     "NOW  RUN",00
                MOV    A,#40H
                CALL   GOTOLCD
                CALL   PRINTLCD
                DB     "NING...",00
                CLR    FLAG_DIS
                CALL   START_MOTOR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CALL    COUNTER
                CALL   STOP_MOTOR
                MOV     IDX_SPEED,#3
                CALL    READ_STAB
                CALL    LOAD_SPEED
                CALL   CLRSCR
                CALL   PRINTLCD
                DB     "FINISH",00
END_M1:         CALL    DELAY
                CALL   SCANKEY
                CJNE   A,#HELP,END_M1
                RET

;***********************************************
:MODE 2  AUTOMATIC RUNNING
;***********************************************
MODE2:          CALL   CLRSCR
                CALL   PRINTLCD
                DB     "MODE 2 O",00
                MOV    A,#40H
                CALL   GOTOLCD
                CALL   PRINTLCD
                DB     "PERATION",00
M2_ENTER:       CALL   DELAY
                CALL   SCANKEY
                CJNE   A,#HELP,M2_ENTER
                MOV    SPEED1,#0
                MOV    SPEED2,#0
                MOV    IDX_SPEED,#0
                CALL    SPEED
                CALL   READ_STAB
                CALL   LOAD_SPEED
                CLR    WAY
                MOV    DIRECT,#'F'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CALL   DISTANCE
                CALL   CLRSCR
                CALL   PRINTLCD
                DB     "NOW  RUN",00
                MOV    A,#40H
                CALL   GOTOLCD
                CALL   PRINTLCD
                DB     "NING...",00
                CLR    FLAG_DIS
                CALL   START_MOTOR
                CALL    COUNTER
                CALL   STOP_MOTOR
                CALL   DELAYS
                CALL   DELAYS
                SETB   WAY
                MOV    DIRECT,#'R'
                CLR    FLAG_DIS
                CALL   START_MOTOR
                CALL    COUNTER
                CALL   STOP_MOTOR
                CALL   CLRSCR
                CALL   PRINTLCD
                DB     "FINISH",00
END_M2:         CALL    DELAY
                CALL   SCANKEY
                CJNE   A,#HELP,END_M2
                RET

;***********************************************
;MODE 3  STEP  RUNNING
;***********************************************
MODE3:          CALL   CLRSCR
                CALL   PRINTLCD
                DB     "MODE 3 O",00
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV    A,#40H
                CALL   GOTOLCD
                CALL   PRINTLCD
                DB     "PERATION",00

M3_ENTER:       CALL   DELAY
                CALL   SCANKEY
                CJNE   A,#HELP,M3_ENTER
                MOV    SPEED1,#0
                MOV    SPEED2,#0
                MOV    IDX_SPEED,#2
                CALL   READ_STAB
                CALL   LOAD_SPEED
                CLR    WAY
                MOV    DIRECT,#'F'
                CALL   CLRSCR
                CALL   PRINTLCD
                DB     "NOW  RUN",00
                MOV    A,#40H
                CALL   GOTOLCD
                CALL   PRINTLCD
                DB     "NING...",00
                CLR    FLAG_DIS
STEP_RUN:       CALL   START_MOTOR
                CALL   STEP
                CALL   STOP_MOTOR
                CALL   DELAYS
                CALL   DELAYS
PAUSE:          CALL   SCANKEY
                CJNE   A,#HELP,STEP_RUN
                CALL   CLRSCR
                CALL   PRINTLCD
                DB     "FINISH",00
END_M3:         CALL   DELAYS
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CALL   SCANKEY
                CJNE   A,#HELP,END_M3
                RET

;***********************************************
STEP:           MOV    R4,#HIGH 45400
                MOV    R3,#LOW 45400
                CALL   TWO_COMP
                MOV    TH0,R4
                MOV    TL0,R3
                JNB    FLAG_DIS,S
                CLR    FLAG_DIS
                RET

;***********************************************
COUNTER:        MOV    R7,COUNT
COUNT1:         MOV    R4,#HIGH 22700
                MOV    R3,#LOW 22700
                CALL   TWO_COMP
                MOV    TH0,R4
                MOV    TL0,R3
                JNB    FLAG_DIS,S
                CLR    FLAG_DIS
                DJNZ   R7,COUNT1
                RET

;***************************************************
;PRE: IDX_SPEED VARIABLE
;POST: SPEED1,SPEED2
;***************************************************
READ_STAB:      MOV    DPTR,#PULSE_TAB
                MOV    A,IDX_SPEED
                CALL   LOOKUPW
                MOV    DPH,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV   SPEED1,DPH
                MOV   SPEED2.DPL
                RET

PULSE_TAB:      DB    00H,40H   ;DUTY 40%
                DB    00H,60H   ;DUTY 60%
                DB    00H,80H   ;DUTY 80%
                DB    00H,00H   ;STOP

;************************************
;INITIAL 8253 SEGMENT
;************************************
LOAD_SPEED:           MOV   A,#73H
                MOV   DPTR,#CTRL
                MOVX  @DPTR,A      ;MONOSATABLE MODE
                MOV   DPTR,#CHN1
                MOV   A,SPEED2
                MOVX  @DPTR,A
                MOV   A,SPEED1
                MOVX  @DPTR,A
                MOV   A,#37H
                MOV   DPTR,#CTRL
                MOVX  @DPTR,A      ;SQUARE WAVE MODE
                MOV   DPTR,#CHN0
                MOV   A,#00H
                MOVX  @DPTR,A
                MOV   A,#01H
                MOVX  @DPTR,A      ;20 KHZ SQUARE WAVE
                RET

;*****************************************
;INTERRUPT ROUTINE TIMER0
;*****************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
TIMER0ISR:      CLR   FLAG_DIS
                RETI

;***********************************************
;MOTOR START
;***********************************************
START_MOTOR:          SETB   STARTCOUNT
                SETB   EN_MOTOR
                RET

;***********************************************
;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]
                SJMP  END_ST
END_ST:         CLR   EN_MOTOR
                RET

;***********************************************
DLYSTOP:        MOV   A,IDX_SPEED
                CJNE  A,#0,STOP1
                CALL  DELAY
                CALL  DELAY
                SJMP  OUT_STOP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
STOP1:      CJNE   A,#1,STOP2
            MOV    R3,#30
DELAYS1:    CALL   DELAY
            DJNZ   R3,DELAYS1
            SJMP   OUT_STOP
STOP2:      CJNE   A,#2,OUT_STOP
            MOV    R3,#100
DELAYS2:    CALL   DELAY
            DJNZ   R3,DELAYS2
OUT_STOP:   RET

;************************************************
DLYBREAK:     MOV   R0,#2
              NOP
              DJNZ  R0,S
              RET

;*********************************************
;SELECT SPEED LEVEL 1,2,3
;*********************************************
SPEED:         PUSH  ACC
SP0:           CALL  CLRSCR
               CALL  PRINTLCD
               DB    "SPEED(1,",00
               MOV   A,#40H
               CALL          GOTOLCD
               CALL    PRINTLCD
               DB      "2,3):?",00
               SJMP  GET_SPEED
GET_SPEED:      CALL DLY
               CALL  SCANKEY
               CJNE  A,#0FFH,SP1
               SJMP  GET_SPEED
SP1:           CJNE  A,#HELP,SP2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                SJMP    SP7
SP2:            CJNE    A,#CLEAR,SP3
                SJMP    SP0
SP3:            CJNE    A,#1,SP4
                MOV     A,#45H
                CALL    GOTOLCD
                CALL    PRINTLCD
                DB      "1",00
                MOV     R5,#1
                MOV     IDX_SPEED,#0
                SJMP    GET_SPEED
SP4:            CJNE    A,#2,SP5
                MOV     A,#45H
                CALL    GOTOLCD
                CALL    PRINTLCD
                DB      "2",00
                MOV     R6,#2
                MOV     IDX_SPEED,#1
                SJMP    GET_SPEED
SP5:            CJNE    A,#3,SP6
                MOV     A,#45H
                CALL    GOTOLCD
                CALL    PRINTLCD
                DB      "3",00
                MOV     R7,#3
                MOV     IDX_SPEED,#2
                SJMP    GET_SPEED
SP6:            CALL    CLRSCR
                CALL    PRINTLCD
                DB      "ERROR",00
                SJMP    SP_ERROR
SP_ERROR:       CALL    SCANKEY
                CJNE    A,#HELP,SP_ERROR
                AJMP    SP0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SP7:        CJNE    R5,#1,SP8
            MOV     R5,#0
            SJMP    SP10
SP8:        CJNE    R6,#2,SP9
            MOV     R6,#0
            SJMP    SP10
SP9:        CJNE    R7,#3,SP6
            MOV     R7,#0
SP10:       POP     ACC
            RET

;*************************************************************
;INPUT DIRECTION OF MOTOR & ACTIVE IMMEDIATELY
;PRE: --
;POST: MOTOR IS RUN IN USER DIRECTION DEFINED
;DIRECT VARIABLE='F' OR 'R'
;*************************************************************
DIRECTION:  PUSH    ACC
BEGIN_DIR:  CALL    CLRSCR
            CALL    PRINTLCD
            DB      "DIRECTIO",00
            MOV     A,#40H
            CALL    GOTOLCD
            CALL    PRINTLCD
            DB      "N MOTOR",00
ENTER1:             CALL    DELAYS
            CALL    SCANKEY
            CJNE    A,#HELP,ENTER1
            CALL    CLRSCR
GET_DIR:    CALL    DELAYS
            CALL    SCANKEY
            CJNE    A,#0FFH,ENTER2
            SJMP    GET_DIR
ENTER2:     CJNE    A,#HELP,FORWARD_DIR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                SJMP    SCAN_DIR
FORWARD_DIR:    CJNE    A,#ND,REVERSE_DIR
                CALL     CLRSCR
                CALL     PRINTLCD
                DB      "FORWARD",00
                MOV      DIRECT,#'F'
                MOV     R3,#'F'
                CLR      WAY
                SJMP    GET_DIR
REVERSE_DIR:            CJNE    A,#KEY_DOWN,GET_DIR
                CALL     CLRSCR
                CALL     PRINTLCD
                DB      "REVERSE",00
                MOV      DIRECT,#'R'
                MOV     R4,#'R'
                SETB     WAY
                SJMP    GET_DIR
DIR_MISS:       CALL    CLRSCR
                CALL    PRINTLCD
                DB      "ERROR",00
                SJMP    DIR_ERROR
DIR_ERROR:      CALL    SCANKEY
                CJNE    A,#HELP,DIR_ERROR
                SJMP    GET_DIR
SCAN_DIR:       CJNE    R3,#'F',SCAN_DRT
                MOV     R3,#0
                SJMP    END_DIR
SCAN_DRT:       CJNE    R4,#'R',DIR_MISS
                MOV     R4,#0
                SJMP    END_DIR
END_DIR:        POP      ACC
                RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*********************************************
;INPUT DISTANCE FROM USER
;PRE: -
;POST: COUNTH:COUNTL VARIABLE
;*********************************************
DISTANCE:       PUSH        ACC
BEGIN_DIS:      CALL        CLRSCR
                CALL    PRINTLCD
                DB      "DISTANCE",00
                MOV     A,#40H
                CALL    GOTOLCD
                CALL    PRINTLCD
                DB      "5-50(cm)",00

DIS_ENTER:      CALL    DLY
                CALL    SCANKEY
                CJNE    A,#HELP,DIS_ENTER
                CALL    READLN
                POP     ACC
                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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV   R4,A
        POP   ACC
        RET

;*******************************
; KEYBOARD SEGMENT
;*******************************
ENTER           EQU   0BH   ;ENTER KEY
CLEAR           EQU   01BH  ;CLEAR KEY
KEY_UP          EQU   011H
KEY_DOWN        EQU   012H
ND              EQU   013H
HELP            EQU   014H

;******** bit addressable***********
KEYPRS      EQU   8     ;INTERNAL RAM ADDR. 21H BIT

;*******************************
; KEYBOARD SECTION
;*******************************
ROW0        EQU   P1.0
ROW1        EQU   P1.1
ROW2        EQU   P1.2
ROW3        EQU   P1.3
COL0        EQU   P1.4
COL1        EQU   P1.5
COL2        EQU   P1.6
COL3        EQU   P1.7

;***********************************************
;SCANKEY BOARD 4 ROW* 4 COLUMN
;OUTPUT : ACC IF EQU 0FFH NOKEY
;***********************************************
SCANKEY:        PUSH  DPL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                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
ACTVCOL3:       SETB   COL2
                CLR    COL3   ;ACTIVE COLUMN 3
                MOV    B,#3
                CALL   WHAT_ROW
                CJNE   A,#0FFH,KEYCODE
                CLR    KEYPRS       ;KEY ALREADY RELEASED
                SJMP   OUT_SNK      ;NO KEY PRESSED
KEYCODE:        JNB    KEYPRS,KCOD1
                SJMP   OUT_SNK
KCOD1:          SETB   KEYPRS
OUT_SNK:        POP    DPH
                POP    DPL
                MOV    P1,#0FFH
                RET
KEYTAB:         DB     01H,02H,03H,11H ; 1  2  3   UP
                DB     04H,05H,06H,12H ; 4  5  6   DOWN
                DB     07H,08H,09H,13H ; 7  8  9   2ND
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                DB      1BH,00H,14H,0BH ;CLR 0 HELP ENTER

;***********************************************
;SEARCH ROW PRESSED & GET OFFSET
;INPUT: B=COLUMN
;OUTPUT: ACC=KEY CODE
;***********************************************
WHAT_ROW:       MOV     A,P1
                ORL     A,#11110000B
                XRL     A,#11111111B
                JNZ     SK      ;KEYPRESSED
                MOV     A,#0FFH ;KEY WASN'T PRESSED
                SJMP    OUT_ROW
SK:             CLR     COL0
                CLR     COL1
                CLR     COL2
                CLR     COL3
                JB      ROW0,WHRW1
                MOV     DPTR,#KEYTAB+0  ;ROW 0
                SJMP    KOFFSET
WHRW1:          JB      ROW1,WHRW2
                MOV     DPTR,#KEYTAB+4  ;ROW 1
                SJMP    KOFFSET
WHRW2:          JB      ROW2,WHRW3
                MOV     DPTR,#KEYTAB+8  ;ROW 2
                SJMP    KOFFSET
WHRW3:          MOV     DPTR,#KEYTAB+12 ;ROW 3
KOFFSET:        MOV     A,B
                MOVC            A,@A+DPTR
OUT_ROW:        RET


;***********************************************
;       READ KEY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;READ DISTANCE VALUE
;OUTPUT : ACCUMURATOR
;*********************************************
READLN:     CALL   CLRSCR
            CALL   WAITBF
RL0:        MOV    R4,#0
            MOV    R3,#0
            MOV    R2,#0
            MOV    COUNT,#0
            SJMP   RL1
RL1:        CALL   DELAYS
            CALL   SCANKEY
            CJNE   A,#0FFH,RL2
            SJMP   RL1
RL2:        CJNE   A,#HELP,RL3
            SJMP   RL8
RL3:        CJNE   A,#CLEAR,RL5
RL4:        CALL   CLRSCR
            SJMP   RL0
RL5:        CJNE   R2,#0FFH,RL6
            CALL   CLRSCR
            CALL   PRINTLCD
            DB     "ERROR",00
DIS_ERROR:  CALL   DLY
            CALL   SCANKEY
            CJNE   A,#HELP,DIS_ERROR
            SJMP   RL4
RL6:        CJNE   R2,#0FH,RL7
            MOV    R4,A
            ORL    A,#30H
            CALL   WRITELCD
            MOV    R2,#0FFH
            SJMP   RL1
RL7:        MOV    R3,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ORL   A,#30H
                CALL  WRITELCD
                MOV   R2,#0FH
                SJMP  RL1
RL8:            CJNE  R3,#0,RL10
                CJNE  R4,#0,RL9
                CALL  CLRSCR
                CALL  PRINTLCD
                DB    "NO DISTA",00
                MOV   A,#40H
                CALL  GOTOLCD
                CALL  PRINTLCD
                DB    "NCE!",00
NO_ENTER:       CALL  DLY
                CALL  SCANKEY
                CJNE  A,#HELP,NO_ENTER
                SJMP  RL4
RL9:            CJNE  R4,#5,RL5
                MOV   COUNT,#1
                SJMP  RL_OUT
RL10:           CJNE  R3,#1,RL12
                CJNE  R4,#0,RL11
                MOV   COUNT,#2
                SJMP  RL_OUT
RL11:           CJNE  R4,#5,RL5
                MOV   COUNT,#3
                SJMP  RL_OUT
RL12:           CJNE  R3,#2,RL14
                CJNE  R4,#0,RL13
                MOV   COUNT,#4
                SJMP  RL_OUT
RL13:           CJNE  R4,#5,RL19
                MOV   COUNT,#5
                SJMP  RL_OUT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
RL14:       CJNE   R3,#3,RL16
            CJNE   R4,#0,RL15
            MOV    COUNT,#6
            SJMP   RL_OUT
RL15:       CJNE   R4,#5,RL19
            MOV    COUNT,#7
            SJMP   RL_OUT
RL16:       CJNE   R3,#4,RL18
            CJNE   R4,#0,RL17
            MOV    COUNT,#8
            SJMP   RL_OUT
RL17:       CJNE   R4,#5,RL19
            MOV    COUNT,#9
            SJMP   RL_OUT
RL18:       CJNE   R3,#5,RL19
            CJNE   R4,#0,RL19
            MOV    COUNT,#10
            SJMP   RL_OUT
RL19:       LJMP   RL5
RL_OUT:     RET

;****************************************
: 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  ;READDATA FROM DD RAM

;**********************************
; WRITE ASCII TO LCD 16X1
; INPUT : ACC = ASCII
;**********************************
WRITELCD:   PUSH   DPL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                PUSH   DPH
                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,#0FFH
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  ;8 BIT,5x7 DOT
                MOVX   @DPTR,A
                LCALL  WAITBF
                MOV    A,#0CH
                MOVX   @DPTR,A
                LCALL  WAITBF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                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

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
                MOV    A,#1
                MOV    DPTR,#COMMAND
                MOVX   @DPTR,A
                CALL   WAITBF
                POP    ACC
                RET

;*************************************
;SET CURSOR ON AND BLINK
;*************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CURSOR_ON:    MOV    A,#0FH
              MOV    DPTR,#COMMAND
              MOVX   @DPTR,A
              RET

;*****************************************************
;WRITE CHARACTER STRING FOLLOWED BY THE
;HCALL  PRINTLCD INSTRUCTION TO LCD
:INPUT :-
;REGISTOR:DESTROY DPTR,ACC
;*****************************************************
PRINTLCD:     POP    DPH    ;GET POINTER(STACK)
              POP    DPL
STSD0:        CLR    A
              MOVC   A,@A+DPTR
              JZ     STSD2
STSD1:        CALL   WRITELCD
              CALL   WAITBF
STSD3:        INC    DPTR
              SJMP   STSD0
STSD2:        INC    DPTR
              PUSH   DPL
              PUSH   DPH
              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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MUL   AB
                ADD   A,DPL
                MOV   DPL,A
                MOV   A,DPH
                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   ACC   ;RETREIVE HI BYTE
                POP   B
                POP   PSW
                RET
;*************************
; GENERAL DELAY
;*************************
DELAY:          MOV   R0,#200 ;1ms
DELAY1:         NOP
                NOP
                NOP
                DJNZ  R0,DELAY1
                RET
DLY:            MOV   R1,#200 ;0.2s
DLY1:           CALL  DELAY
                DJNZ  R1,DLY1
                RET
DELAYS:         CALL  DLY     ;0.4s
                CALL  DLY
                RET
END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

intersil

82C54

March 1997

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 10µA
  - ICCOP ..... 10mA at 8MHz
- Operating Temperature Ranges
  - C82C54 ..... 0°C to +70°C
  - I82C54 ..... -40°C to +85°C
  - M82C54 ..... -55°C to +125°C

Description

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

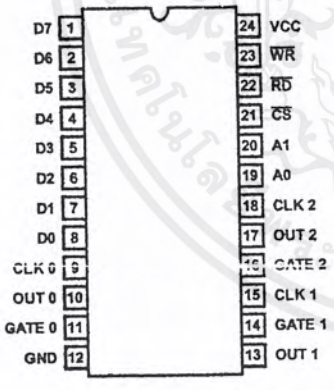
The 82C54 has three independently programmable and functional 16-bit counters, each capable of handling clock input frequencies of up to 8MHz (82C54) or 10MHz (82C54-10) or 12MHz (82C54-12).

The high speed and industry standard configuration of the 82C54 make it compatible with the Intersil 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.

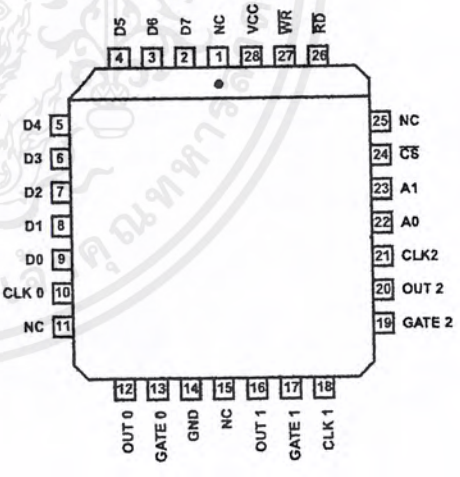
The Intersil advanced CMOS process results in a significant reduction in power with performance equal to or greater than existing equivalent products.

Pinouts

82C54 (PDIP, CERDIP, SOIC)
TOP VIEW



82C54 (PLCC/CLCC)
TOP VIEW



CAUTION: These devices are sensitive to electrostatic discharge; follow proper IC Handling Procedures.
http://www.intersil.com or 407-727-9207 | Copyright © Intersil Corporation 1999

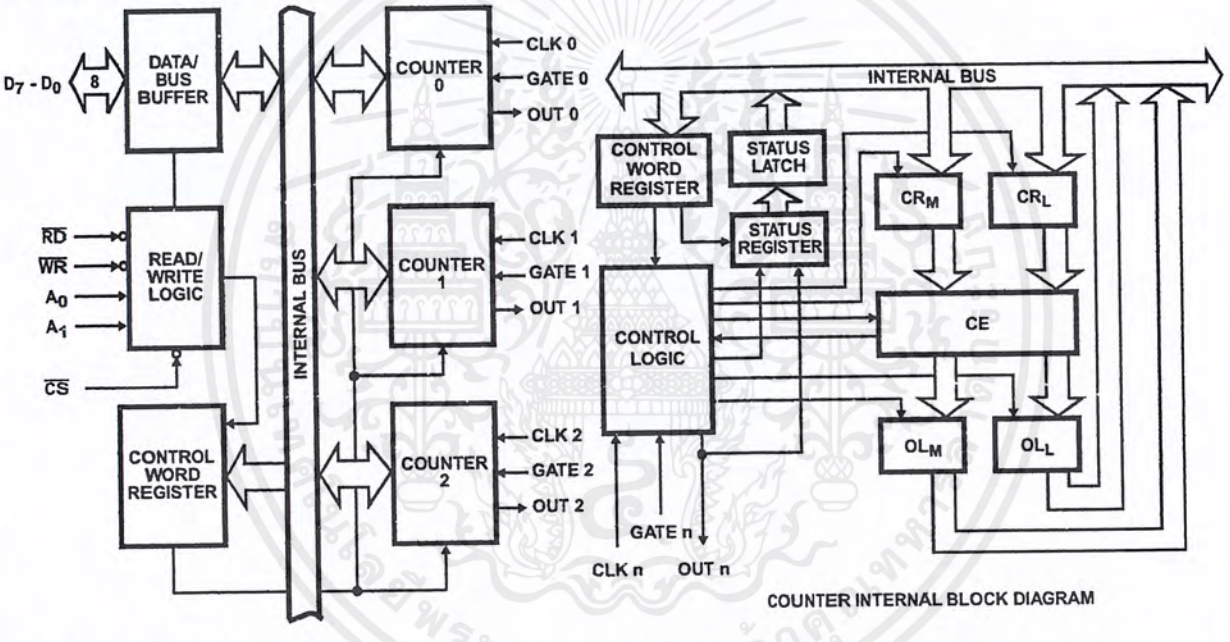
File Number **2970.1**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

PART NUMBERS			TEMPERATURE RANGE	PACKAGE	PKG. NO.
8MHz	10MHz	12MHz			
CP82C54	CP82C54-10	CP82C54-12	0°C to +70°C	24 Lead PDIP	E24.6
IP82C54	IP82C54-10	IP82C54-12	-40°C to +85°C	24 Lead PDIP	E24.6
CS82C54	CS82C54-10	CS82C54-12	0°C to +70°C	28 Lead PLCC	N28.45
IS82C54	IS82C54-10	IS82C54-12	-40°C to +85°C	28 Lead PLCC	N28.45
CD82C54	CD82C54-10	CD82C54-12	0°C to +70°C	24 Lead CERDIP	F24.6
ID82C54	ID82C54-10	ID82C54-12	-40°C to +85°C	24 Lead CERDIP	F24.6
MD82C54/B	MD82C54-10/B	MD82C54-12/B	-55°C to +125°C	24 Lead CERDIP	F24.6
MR82C54/B	MR82C54-10/B	MR82C54-12/B	-55°C to +125°C	28 Lead CLCC	J28.A
SMD # 8406501JA	-	8406502JA	-55°C to +125°C	24 Lead CERDIP	F24.6
SMD# 84065013A	-	84065023A	-55°C to +125°C	28 Lead CLCC	J28.A
CM82C54	CM82C54-10	CM82C54-12	0°C to +70°C	24 Lead SOIC	M24.3

Functional Diagram



COUNTER INTERNAL BLOCK DIAGRAM

Pin Description

SYMBOL	DIP PIN NUMBER	TYPE	DEFINITION
D7 - D0	1 - 8	I/O	DATA: Bi-directional three-state data bus lines, connected to system data bus.
CLK 0	9	I	CLOCK 0: Clock input of Counter 0.
OUT 0	10	O	OUT 0: Output of Counter 0.
GATE 0	11	I	GATE 0: Gate input of Counter 0.
GND	12		GROUND: Power supply connection.
OUT 1	13	O	OUT 1: Output of Counter 1.
GATE 1	14	I	GATE 1: Gate input of Counter 1.
CLK 1	15	I	CLOCK 1: Clock input of Counter 1.
GATE 2	16	I	GATE 2: Gate input of Counter 2.
OUT 2	17	O	OUT 2: Output of Counter 2.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Description (Continued)

SYMBOL	DIP PIN NUMBER	TYPE	DEFINITION
CLK 2	18	I	CLOCK 2: Clock input of Counter 2.
A0, A1	19 - 20	I	ADDRESS: Select inputs for one of the three counters or Control Word Register for read/write operations. Normally connected to the system address bus.
$\overline{CS}$	21	I	CHIP SELECT: A low on this input enables the 82C54 to respond to $\overline{RD}$ and $\overline{WR}$ signals. $\overline{RD}$ and $\overline{WR}$ are ignored otherwise.
$\overline{RD}$	22	I	READ: This input is low during CPU read operations.
$\overline{WR}$	23	I	WRITE: This input is low during CPU write operations.
V_{CC}	24		V_{CC}: The +5V power supply pin. A 0.1µF capacitor between pins VCC and GND is recommended for decoupling.

A1	A0	SELECTS
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2
1	1	Control Word Register

Functional Description

General

The 82C54 is a programmable interval timer/counter designed for use with microcomputer systems. It is a general purpose, multi-timing element that can be treated as an array of I/O ports in the system software.

The 82C54 solves one of the most common problems in any microcomputer system, the generation of accurate time delays under software control. Instead of setting up timing loops in software, the programmer configures the 82C54 to match his requirements and programs one of the counters for the desired delay. After the desired delay, the 82C54 will interrupt the CPU. Software overhead is minimal and variable length delays can easily be accommodated.

Some of the other computer/timer functions common to microcomputers which can be implemented with the 82C54 are:

- Real time clock
- Event counter
- Digital one-shot
- Programmable rate generator
- Square wave generator
- Binary rate multiplier
- Complex waveform generator
- Complex motor controller

Data Bus Buffer

This three-state, bi-directional, 8-bit buffer is used to interface the 82C54 to the system bus (see Figure 1).

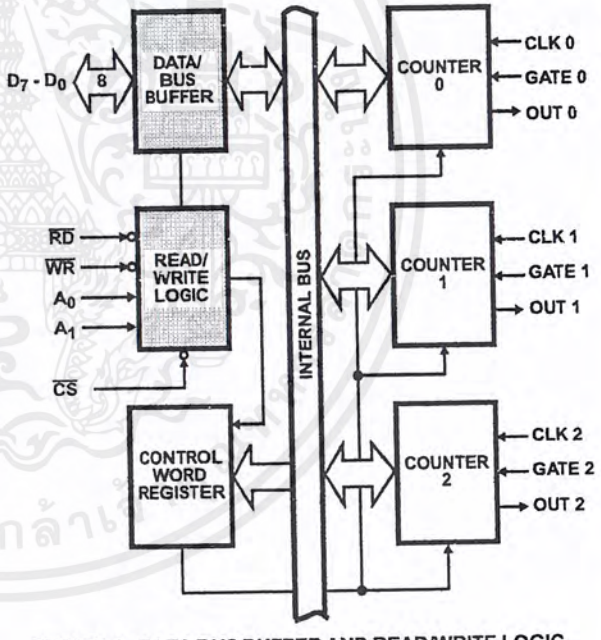


FIGURE 1. DATA BUS BUFFER AND READ/WRITE LOGIC FUNCTIONS

Read/Write Logic

The Read/Write Logic accepts inputs from the system bus and generates control signals for the other functional blocks of the 82C54. A1 and A0 select one of the three counters or the Control Word Register to be read from/written into. A "low" on the $\overline{RD}$ input tells the 82C54 that the CPU is reading one of the counters. A "low" on the $\overline{WR}$ input tells the 82C54 that the CPU is writing either a Control Word or an initial count. Both $\overline{RD}$ and $\overline{WR}$ are qualified by $\overline{CS}$; $\overline{RD}$ and $\overline{WR}$ are ignored unless the 82C54 has been selected by holding $\overline{CS}$ low.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Control Word Register

The Control Word Register (Figure 2) is selected by the Read/Write Logic when A1, A0 = 11. If the CPU then does a write operation to the 82C54, the data is stored in the Control Word Register and is interpreted as a Control Word used to define the Counter operation.

The Control Word Register can only be written to; status information is available with the Read-Back Command.

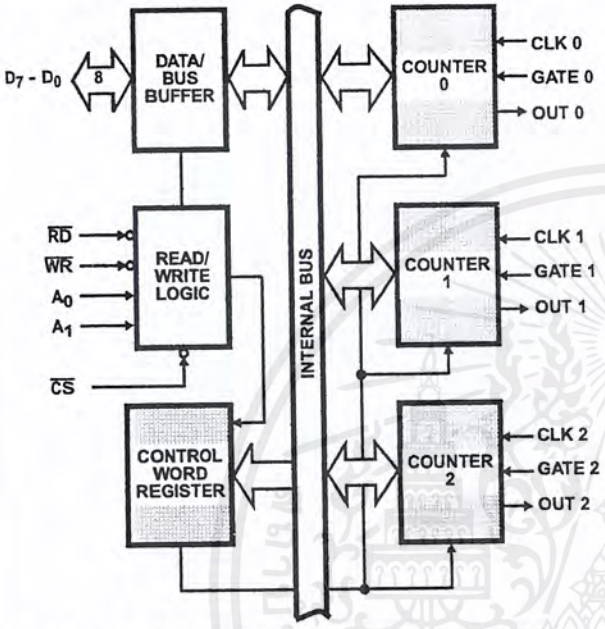


FIGURE 2. CONTROL WORD REGISTER AND COUNTER FUNCTIONS

Counter 0, Counter 1, Counter 2

These three functional blocks are identical in operation, so only a single Counter will be described. The internal block diagram of a signal counter is shown in Figure 3. The counters are fully independent. Each Counter may operate in a different Mode.

The Control Word Register is shown in the figure; it is not part of the Counter itself, but its contents determine how the Counter operates.

The status register, shown in the figure, when latched, contains the current contents of the Control Word Register and status of the output and null count flag. (See detailed explanation of the Read-Back command.)

The actual counter is labeled CE (for Counting Element). It is a 16-bit presettable synchronous down counter.

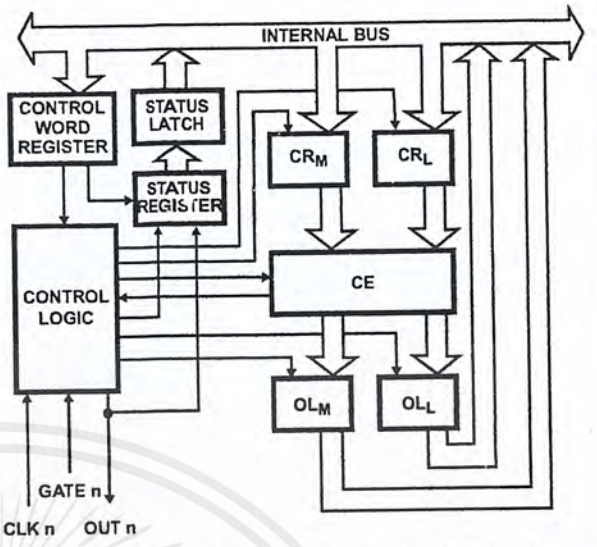


FIGURE 3. COUNTER INTERNAL BLOCK DIAGRAM

OLM and OLL are two 8-bit latches. OL stands for "Output Latch"; the subscripts M and L for "Most significant byte" and "Least significant byte", respectively. Both are normally referred to as one unit and called just OL. These latches normally "follow" the CE, but if a suitable Counter Latch Command is sent to the 82C54, the latches "latch" the present count until read by the CPU and then return to "following" the CE. One latch at a time is enabled by the counter's Control Logic to drive the internal bus. This is how the 16-bit Counter communicates over the 8-bit internal bus. Note that the CE itself cannot be read; whenever you read the count, it is the OL that is being read.

Similarly, there are two 8-bit registers called CRM and CRL (for "Count Register"). Both are normally referred to as one unit and called just CR. When a new count is written to the Counter, the count is stored in the CR and later transferred to the CE. The Control Logic allows one register at a time to be loaded from the internal bus. Both bytes are transferred to the CE simultaneously. CRM and CRL are cleared when the Counter is programmed for one byte counts (either most significant byte only or least significant byte only) the other byte will be zero. Note that the CE cannot be written into; whenever a count is written, it is written into the CR.

The Control Logic is also shown in the diagram. CLK n, GATE n, and OUT n are all connected to the outside world through the Control Logic.

82C54 System Interface

The 82C54 is treated by the system software as an array of peripheral I/O ports; three are counters and the fourth is a control register for MODE programming.

Basically, the select inputs A0, A1 connect to the A0, A1 address bus signals of the CPU. The $\overline{CS}$ can be derived directly from the address bus using a linear select method or it can be connected to the output of a decoder.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Operational Description

General

After power-up, the state of the 82C54 is undefined. The Mode, count value, and output of all Counters are undefined.

How each Counter operates is determined when it is programmed. Each Counter must be programmed before it can be used. Unused counters need not be programmed.

Programming the 82C54

Counters are programmed by writing a Control Word and then an initial count.

All Control Words are written into the Control Word Register, which is selected when A1, A0 = 11. The Control Word specifies which Counter is being programmed.

By contrast, initial counts are written into the Counters, not the Control Word Register. The A1, A0 inputs are used to select the Counter to be written into. The format of the initial count is determined by the Control Word used.

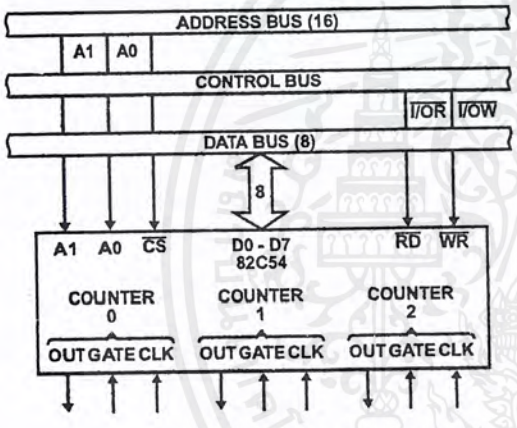


FIGURE 4. 82C54 SYSTEM INTERFACE

Write Operations

The programming procedure for the 82C54 is very flexible. Only two conventions need to be remembered:

1. For Each Counter, the Control Word must be written before the initial count is written.
2. The initial count must follow the count format specified in the Control Word (least significant byte only, most significant byte only, or least significant byte and then most significant byte).

Since the Control Word Register and the three Counters have separate addresses (selected by the A1, A0 inputs), and each Control Word specifies the Counter it applies to (SC0, SC1 bits), no special instruction sequence is required. Any programming sequence that follows the conventions above is acceptable.

Control Word Format

A1, A0 = 11; $\overline{CS}$ = 0; $\overline{RD}$ = 1; $\overline{WR}$ = 0

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC - Select Counter

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command (See Read Operations)

RW - Read/Write

RW1	RW0	
0	0	Counter Latch Command (See Read Operations)
0	1	Read/Write least significant byte only.
1	0	Read/Write most significant byte only.
1	1	Read/Write least significant byte first, then most significant byte.

M - Mode

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

BCD - Binary Coded Decimal

0	Binary Counter 16-bit
1	Binary Coded Decimal (BCD) Counter (4 Decades)

NOTE: Don't Care bits (X) should be 0 to insure compatibility with future products.

Possible Programming Sequence

	A1	A0
Control Word - Counter 0	1	1
LSB of Count - Counter 0	0	0
MSB of Count - Counter 0	0	0
Control Word - Counter 1	1.	1
LSB of Count - Counter 1	0	1
MSB of Count - Counter 1	0	1
Control Word - Counter 2	1	1
LSB of Count - Counter 2	1	0
MSB of Count - Counter 2	1	0

Possible Programming Sequence

	A1	A0
Control Word - Counter 0	1	1
Control Word - Counter 1	1	1
Control Word - Counter 2	1	1
LSB of Count - Counter 2	1	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Possible Programming Sequence (Continued)

	A1	A0
LSB of Count - Counter 1	0	1
LSB of Count - Counter 0	0	0
MSB of Count - Counter 0	0	0
MSB of Count - Counter 1	0	1
MSB of Count - Counter 2	1	0

Possible Programming Sequence

	A1	A0
Control Word - Counter 2	1	1
Control Word - Counter 1	1	1
Control Word - Counter 0	1	1
LSB of Count - Counter 2	1	0
MSB of Count - Counter 2	1	0
LSB of Count - Counter 1	0	1
MSB of Count - Counter 1	0	1
LSB of Count - Counter 0	0	0
MSB of Count - Counter 0	0	0

Possible Programming Sequence

	A1	A0
Control Word - Counter 1	1	1
Control Word - Counter 0	1	1
LSB of Count - Counter 1	0	1
Control Word - Counter 2	1	1
LSB of Count - Counter 0	0	0
MSB of Count - Counter 1	0	1
LSB of Count - Counter 2	1	0
MSB of Count - Counter 0	0	0
MSB of Count - Counter 2	1	0

NOTE: In all four examples, all counters are programmed to Read/Write two-byte counts. These are only four of many programming sequences.

A new initial count may be written to a Counter at any time without affecting the Counter's programmed Mode in any way. Counting will be affected as described in the Mode definitions. The new count must follow the programmed count format.

If a Counter is programmed to read/write two-byte counts, the following precaution applies. A program must not transfer control between writing the first and second byte to another routine which also writes into that same Counter. Otherwise, the Counter will be loaded with an incorrect count.

Read Operations

It is often desirable to read the value of a Counter without disturbing the count in progress. This is easily done in the 82C54.

There are three possible methods for reading the Counters. The first is through the Read-Back command, which is explained later. The second is a simple read operation of the Counter, which is selected with the A1, A0 inputs. The only requirement is that the CLK input of the selected Counter must be inhibited by using either the GATE input or external logic. Otherwise, the count may be in process of changing when it is read, giving an undefined result.

Counter Latch Command

The other method for reading the Counters involves a special software command called the "Counter Latch Command". Like a Control Word, this command is written to the Control Word Register, which is selected when A1, A0 = 11. Also, like a Control Word, the SC0, SC1 bits select one of the three Counters, but two other bits, D5 and D4, distinguish this command from a Control Word.

A1, A0 = 11; $\overline{CS}$ = 0; $\overline{RD}$ = 1; $\overline{WR}$ = 0

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	0	0	X	X	X	X

SC1, SC0 - specify counter to be latched

SC1	SC0	COUNTER
0	0	0
0	1	1
1	0	2
1	1	Read-Back Command

D5, D4 - 00 designates Counter Latch Command, X - Don't Care.

NOTE: Don't Care bits (X) should be 0 to insure compatibility with future products.

The selected Counter's output latch (OL) latches the count when the Counter Latch Command is received. This count is held in the latch until it is read by the CPU (or until the Counter is reprogrammed). The count is then unlatched automatically and the OL returns to "following" the counting element (CE). This allows reading the contents of the Counters "on the fly" without affecting counting in progress. Multiple Counter Latch Commands may be used to latch more than one Counter. Each latched Counter's OL holds its count until read. Counter Latch Commands do not affect the programmed Mode of the Counter in any way.

If a Counter is latched and then, some time later, latched again before the count is read, the second Counter Latch Command is ignored. The count read will be the count at the time the first Counter Latch Command was issued.

With either method, the count must be read according to the programmed format; specifically, if the Counter is programmed for two byte counts, two bytes must be read. The two bytes do not have to be read one right after the other; read or write or programming operations of other Counters may be inserted between them.

Another feature of the 82C54 is that reads and writes of the same Counter may be interleaved; for example, if the Counter is programmed for two byte counts, the following sequence is valid.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Read least significant byte.
2. Write new least significant byte.
3. Read most significant byte.
4. Write new most significant byte.

If a counter is programmed to read or write two-byte counts, the following precaution applies: A program MUST NOT transfer control between reading the first and second byte to another routine which also reads from that same Counter. Otherwise, an incorrect count will be read.

Read-Back Command

The read-back command allows the user to check the count value, programmed Mode, and current state of the OUT pin and Null Count flag of the selected counter(s).

The command is written into the Control Word Register and has the format shown in Figure 5. The command applies to the counters selected by setting their corresponding bits D3, D2, D1 = 1.

A0, A1 = 11; $\overline{CS}$ = 0; $\overline{RD}$ = 1; $\overline{WR}$ = 0

D7	D6	D5	D4	D3	D2	D1	D0
1	1	$\overline{COUNT}$	$\overline{STATUS}$	CNT 2	CNT 1	CNT 0	0

D5: 0 = Latch count of selected Counter (s)
D4: 0 = Latch status of selected Counter(s)
D3: 1 = Select Counter 2
D2: 1 = Select Counter 1
D1: 1 = Select Counter 0
D0: Reserved for future expansion; Must be 0

FIGURE 5. READ-BACK COMMAND FORMAT

The read-back command may be used to latch multiple counter output latches (OL) by setting the COUNT bit D5 = 0 and selecting the desired counter(s). This signal command is functionally equivalent to several counter latch commands, one for each counter latched. Each counter's latched count is held until it is read (or the counter is reprogrammed). That counter is automatically unlatched when read, but other counters remain latched until they are read. If multiple count read-back commands are issued to the same counter without reading the count, all but the first are ignored; i.e., the count which will be read is the count at the time the first read-back command was issued.

The read-back command may also be used to latch status information of selected counter(s) by setting STATUS bit D4 = 0. Status must be latched to be read; status of a counter is accessed by a read from that counter.

The counter status format is shown in Figure 6. Bits D5 through D0 contain the counter's programmed Mode exactly as written in the last Mode Control Word. OUTPUT bit D7 contains the current state of the OUT pin. This allows the user to monitor the counter's output via software, possibly eliminating some hardware from a system.

D7	D6	D5	D4	D3	D2	D1	D0
OUTPUT	NULL COUNT	RW1	RW0	M2	M1	M0	BCD

D7: 1 = Out pin is 1
0 = Out pin is 0
D6: 1 = Null count
0 = Count available for reading
D5 - D0 = Counter programmed mode (See Control Word Formats)

FIGURE 6. STATUS BYTE

NULL COUNT bit D6 indicates when the last count written to the counter register (CR) has been loaded into the counting element (CE). The exact time this happens depends on the Mode of the counter and is described in the Mode Definitions, but until the counter is loaded into the counting element (CE), it can't be read from the counter. If the count is latched or read before this time, the count value will not reflect the new count just written. The operation of Null Count is shown below.

THIS ACTION: **CAUSES:**

A. Write to the control word register:(1) . . . . . Null Count = 1
B. Write to the count register (CR):(2) . . . . . Null Count = 1
C. New count is loaded into CE (CR - CE) . . . . . Null Count = 0

(1) Only the counter specified by the control word will have its null count set to 1. Null count bits of other counters are unaffected.

(2) If the counter is programmed for two-byte counts (least significant byte then most significant byte) null count goes to 1 when the second byte is written.

If multiple status latch operations of the counter(s) are performed without reading the status, all but the first are ignored; i.e., the status that will be read is the status of the counter at the time the first status read-back command was issued.

COMMANDS								DESCRIPTION	RESULT
D7	D6	D5	D4	D3	D2	D1	D0		
1	1	0	0	0	0	1	0	Read-Back Count and Status of Counter 0	Count and Status Latched for Counter 0
1	1	1	0	0	1	0	0	Read-Back Status of Counter 1	Status Latched for Counter 1
1	1	1	0	1	1	0	0	Read-Back Status of Counters 2, 1	Status Latched for Counter 2, But Not Counter 1
1	1	0	1	1	0	0	0	Read-Back Count of Counter 2	Count Latched for Counter 2
1	1	0	0	0	1	0	0	Read-Back Count and Status of Counter 1	Count Latched for Counter 1, But Not Status
1	1	1	0	0	0	1	0	Read-Back Status of Counter 1	Command Ignored, Status Already Latched for Counter 1

FIGURE 7. READ-BACK COMMAND EXAMPLE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Both count and status of the selected counter(s) may be latched simultaneously by setting both COUNT and STATUS bits D5, D4 = 0. This is functionally the same as issuing two separate read-back commands at once, and the above discussions apply here also. Specifically, if multiple count and/or status read-back commands are issued to the same counter(s) without any intervening reads, all but the first are ignored. This is illustrated in Figure 7.

If both count and status of a counter are latched, the first read operation of that counter will return latched status, regardless of which was latched first. The next one or two reads (depending on whether the counter is programmed for one or two type counts) return latched count. Subsequent reads return unlatched count.

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A1	A0	
0	1	0	0	0	Write into Counter 0
0	1	0	0	1	Write into Counter 1
0	1	0	1	0	Write into Counter 2
0	1	0	1	1	Write Control Word
0	0	1	0	0	Read from Counter 0
0	0	1	0	1	Read from Counter 1
0	0	1	1	0	Read from Counter 2
0	0	1	1	1	No-Operation (Three-State)
1	X	X	X	X	No-Operation (Three-State)
0	1	1	X	X	No-Operation (Three-State)

FIGURE 8. READ/WRITE OPERATIONS SUMMARY

Mode Definitions

The following are defined for use in describing the operation of the 82C54.

CLK PULSE:

A rising edge, then a falling edge, in that order, of a Counter's CLK input.

TRIGGER:

A rising edge of a Counter's Gate input.

COUNTER LOADING:

The transfer of a count from the CR to the CE (See "Functional Description")

Mode 0: Interrupt on Terminal Count

Mode 0 is typically used for event counting. After the Control Word is written, OUT is initially low, and will remain low until the Counter reaches zero. OUT then goes high and remains high until a new count or a new Mode 0 Control Word is written to the Counter.

GATE = 1 enables counting; GATE = 0 disables counting. GATE has no effect on OUT.

After the Control Word and initial count are written to a Counter, the initial count will be loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not go high until N + 1 CLK pulses after the initial count is written.

If a new count is written to the Counter it will be loaded on the next CLK pulse and counting will continue from the new count. If a two-byte count is written, the following happens:

(1) Writing the first byte disables counting. Out is set low immediately (no clock pulse required).

(2) Writing the second byte allows the new count to be loaded on the next CLK pulse.

This allows the counting sequence to be synchronized by software. Again OUT does not go high until N + 1 CLK pulses after the new count of N is written.

If an initial count is written while GATE = 0, it will still be loaded on the next CLK pulse. When GATE goes high, OUT will go high N CLK pulses later; no CLK pulse is needed to load the counter as this has already been done.

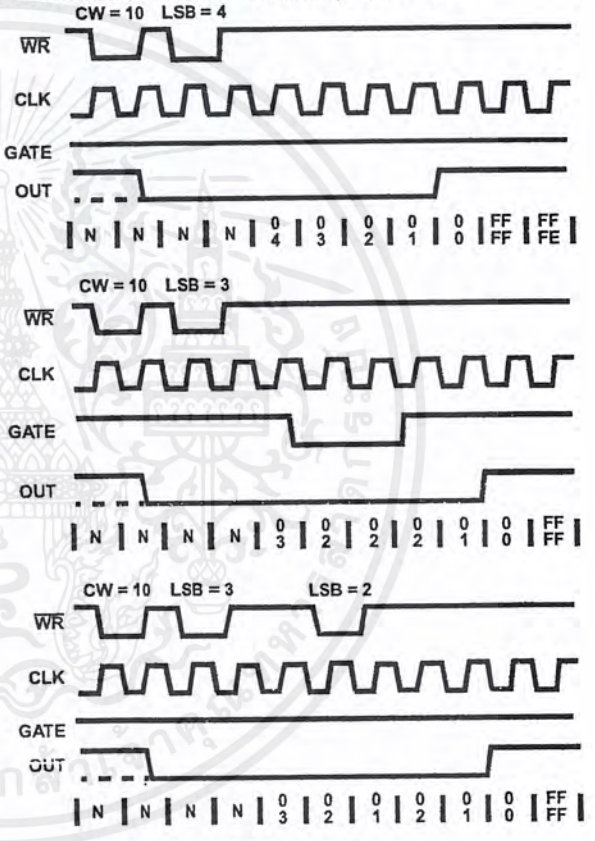


FIGURE 9. MODE 0

NOTES: The following conventions apply to all mode timing diagrams.

1. Counters are programmed for binary (not BCD) counting and for reading/writing least significant byte (LSB) only.
2. The counter is always selected ($\overline{CS}$ always low).
3. CW stands for "Control Word"; CW = 10 means a control word of 10, Hex is written to the counter.
4. LSB stands for Least significant "byte" of count.
5. Numbers below diagrams are count values. The lower number is the least significant byte. The upper number is the most significant byte. Since the counter is programmed to read/write LSB only, the most significant byte cannot be read.
6. N stands for an undefined count.
7. Vertical lines show transitions between count values.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mode 1: Hardware Retriggerable One-Shot

OUT will be initially high. OUT will go low on the CLK pulse following a trigger to begin the one-shot pulse, and will remain low until the Counter reaches zero. OUT will then go high and remain high until the CLK pulse after the next trigger.

After writing the Control Word and initial count, the Counter is armed. A trigger results in loading the Counter and setting OUT low on the next CLK pulse, thus starting the one-shot pulse N CLK cycles in duration. The one-shot is retriggerable, hence OUT will remain low for N CLK pulses after any trigger. The one-shot pulse can be repeated without rewriting the same count into the counter. GATE has no effect on OUT.

If a new count is written to the Counter during a one-shot pulse, the current one-shot is not affected unless the Counter is retriggerable. In that case, the Counter is loaded with the new count and the one-shot pulse continues until the new count expires.

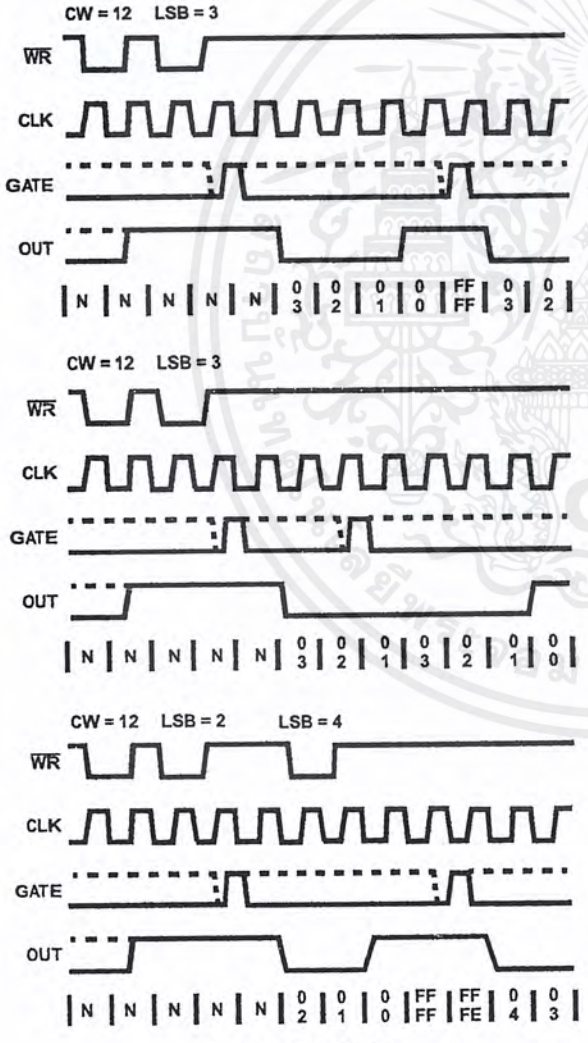


FIGURE 10. MODE 1

Mode 2: Rate Generator

This Mode functions like a divide-by-N counter. It is typically used to generate a Real Time Clock Interrupt. OUT will initially be high. When the initial count has decremented to 1, OUT goes low for one CLK pulse. OUT then goes high again, the Counter reloads the initial count and the process is repeated. Mode 2 is periodic; the same sequence is repeated indefinitely. For an initial count of N, the sequence repeats every N CLK cycles.

GATE = 1 enables counting; GATE = 0 disables counting. If GATE goes low during an output pulse, OUT is set high immediately. A trigger reloads the Counter with the initial count on the next CLK pulse; OUT goes low N CLK pulses after the trigger. Thus the GATE input can be used to synchronize the Counter.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. OUT goes low N CLK pulses after the initial count is written. This allows the Counter to be synchronized by software also.

Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count but before the end of the current period, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from the end of the current counting cycle.

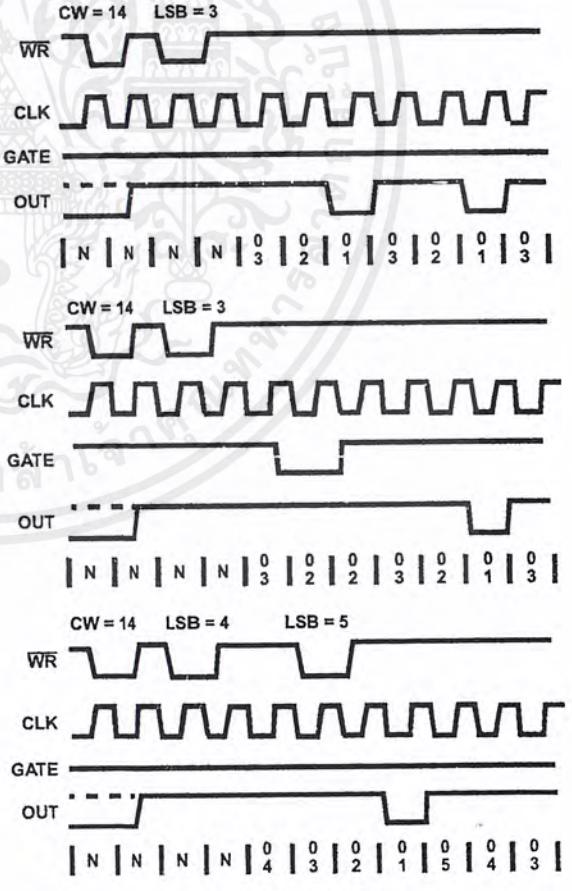


FIGURE 11. MODE 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mode 3: Square Wave Mode

Mode 3 is typically used for Baud rate generation. Mode 3 is similar to Mode 2 except for the duty cycle of OUT. OUT will initially be high. When half the initial count has expired, OUT goes low for the remainder of the count. Mode 3 is periodic; the sequence above is repeated indefinitely. An initial count of N results in a square wave with a period of N CLK cycles.

GATE = 1 enables counting; GATE = 0 disables counting. If GATE goes low while OUT is low, OUT is set high immediately; no CLK pulse is required. A trigger reloads the Counter with the initial count on the next CLK pulse. Thus the GATE input can be used to synchronize the Counter.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. This allows the Counter to be synchronized by software also.

Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count but before the end of the current half-cycle of the square wave, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from the new count. Otherwise, the new count will be loaded at the end of the current half-cycle.

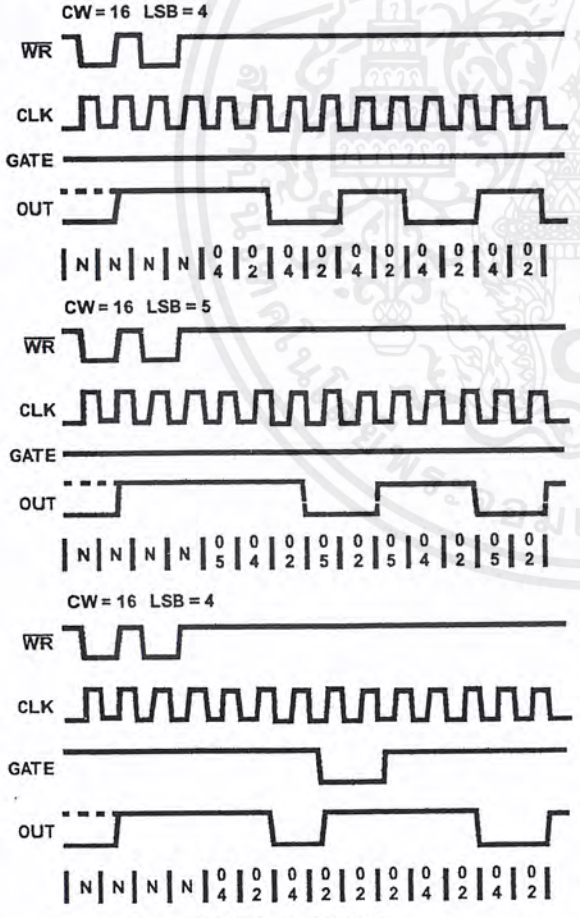


FIGURE 12. MODE 3

Mode 3 is Implemented as Follows:

EVEN COUNTS: OUT is initially high. The initial count is loaded on one CLK pulse and then is decremented by two on succeeding CLK pulses. When the count expires, OUT changes value and the Counter is reloaded with the initial count. The above process is repeated indefinitely.

ODD COUNTS: OUT is initially high. The initial count is loaded on one CLK pulse, decremented by one on the next CLK pulse, and then decremented by two on succeeding CLK pulses. When the count expires, OUT goes low and the Counter is reloaded with the initial count. The count is decremented by three on the next CLK pulse, and then by two on succeeding CLK pulses. When the count expires, OUT goes high again and the Counter is reloaded with the initial count. The above process is repeated indefinitely. So for odd counts, OUT will be high for (N + 1)/2 counts and low for (N - 1)/2 counts.

Mode 4: Software Triggered Mode

OUT will be initially high. When the initial count expires, OUT will go low for one CLK pulse then go high again. The counting sequence is "Triggered" by writing the initial count.

GATE = 1 enables counting; GATE = 0 disables counting. GATE has no effect on OUT.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of N, OUT does not strobe low until N + 1 CLK pulses after the initial count is written.

If a new count is written during counting, it will be loaded on the next CLK pulse and counting will continue from the new count. If a two-byte count is written, the following happens:

(1) Writing the first byte has no effect on counting.

(2) Writing the second byte allows the new count to be loaded on the next CLK pulse.

This allows the sequence to be "retriggered" by software. OUT strobes low N + 1 CLK pulses after the new count of N is written.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

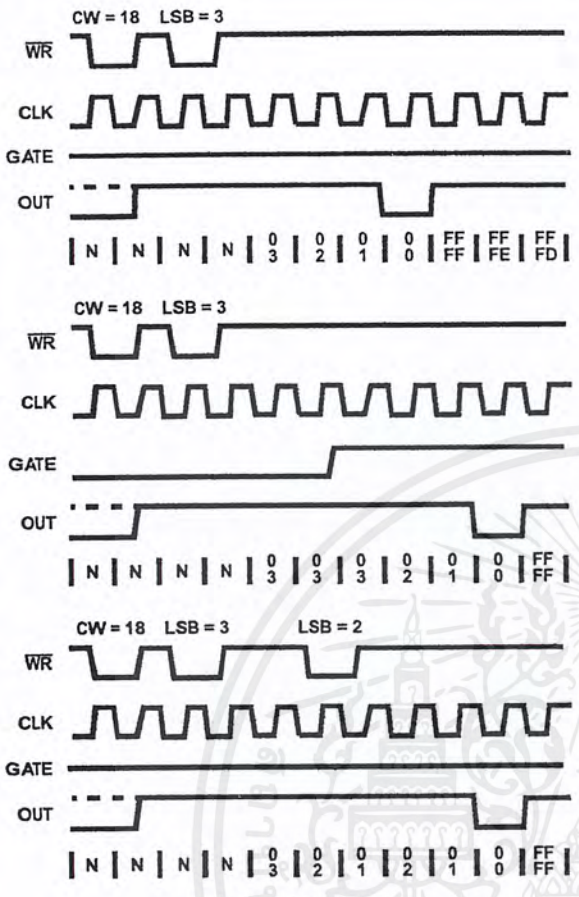


FIGURE 13. MODE 4

Mode 5: Hardware Triggered Strobe (Retriggerable)

OUT will initially be high. Counting is triggered by a rising edge of GATE. When the initial count has expired, OUT will go low for one CLK pulse and then go high again.

After writing the Control Word and initial count, the counter will not be loaded until the CLK pulse after a trigger. This CLK pulse does not decrement the count, so for an initial count of N. OUT does not strobe low until N + 1 CLK pulses after trigger.

A trigger results in the Counter being loaded with the initial count on the next CLK pulse. The counting sequence is triggerable. OUT will not strobe low for N + 1 CLK pulses after any trigger GATE has no effect on OUT.

If a new count is written during counting, the current counting sequence will not be affected. If a trigger occurs after the new count is written but before the current count expires, the Counter will be loaded with new count on the next CLK pulse and counting will continue from there.

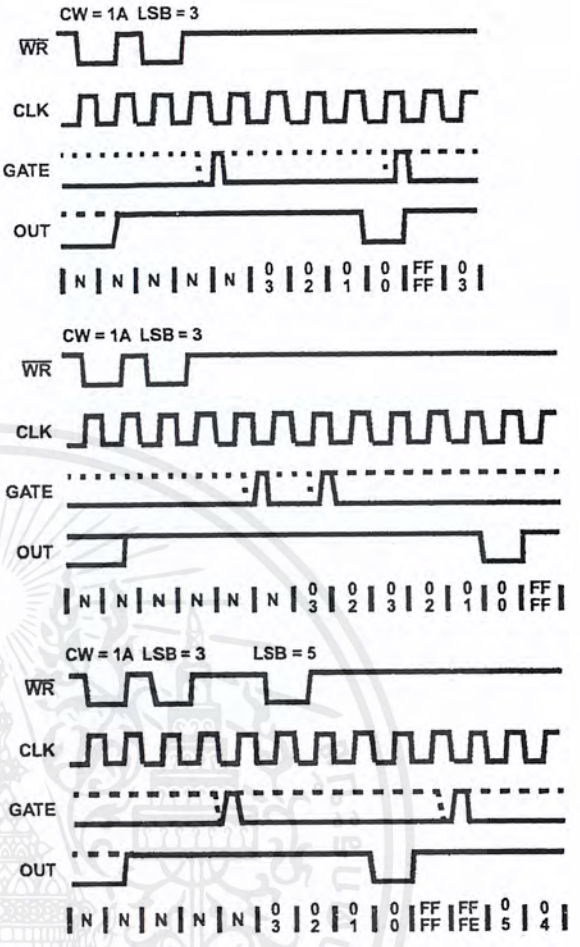


FIGURE 14. MODE 5

Operation Common to All Modes

Programming

When a Control Word is written to a Counter, all Control Logic, is immediately reset and OUT goes to a known initial state; no CLK pulses are required for this.

Gate

The GATE input is always sampled on the rising edge of CLK. In Modes 0, 2, 3 and 4 the GATE input is level sensitive, and logic level is sampled on the rising edge of CLK. In modes 1, 2, 3 and 5 the GATE input is rising-edge sensitive. In these Modes, a rising edge of Gate (trigger) sets an edge-sensitive flip-flop in the Counter. This flip-flop is then sampled on the next rising edge of CLK. The flip-flop is reset immediately after it is sampled. In this way, a trigger will be detected no matter when it occurs - a high logic level does not have to be maintained until the next rising edge of CLK. Note that in Modes 2 and 3, the GATE input is both edge- and level-sensitive.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Counter

New counts are loaded and Counters are decremented on the falling edge of CLK.

The largest possible initial count is 0; this is equivalent to 2^{16} for binary counting and 10^4 for BCD counting.

The counter does not stop when it reaches zero. In Modes 0, 1, 4, and 5 the Counter "wraps around" to the highest count, either FFFF hex for binary counting or 9999 for BCD counting, and continues counting. Modes 2 and 3 are periodic; the Counter reloads itself with the initial count and continues counting from there.

SIGNAL STATUS MODES	LOW OR GOING LOW	RISING	HIGH
0	Disables Counting	-	Enables Counting
1	-	1) Initiates Counting 2) Resets output after next clock	-
2	1) Disables counting 2) Sets output immediately high	Initiates Counting	Enables Counting
3	1) Disables counting 2) Sets output immediately high	Initiates Counting	Enables Counting
4	1) Disables Counting	-	Enables Counting
5	-	Initiates Counting	-

FIGURE 15. GATE PIN OPERATIONS SUMMARY

MODE	MIN COUNT	MAX COUNT
0	1	0
1	1	0
2	2	0
3	2	0
4	1	0
5	1	0

NOTE: 0 is equivalent to 2^{16} for binary counting and 10^4 for BCD counting.

FIGURE 16. MINIMUM AND MAXIMUM INITIAL COUNTS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings

Supply Voltage . . . . . +8.0V
Input, Output or I/O Voltage . . . . . GND-0.5V to V_{CC} +0.5V
ESD Classification . . . . . Class 1

Operating Conditions

Operating Voltage Range . . . . . +4.5V to +5.5V
Operating Temperature Range
C82C54, C82C54-10, -12 . . . . . 0°C to +70°C
I82C54, I82C54-10, -12 . . . . . -40°C to +85°C
M82C54, M82C54-10, -12 . . . . . -55°C to +125°C

Thermal Information

Thermal Resistance (Typical)	θ_{JA} (°C/W)	θ_{JC} (°C/W)
CERDIP Package	55	12
CLCC Package	65	14
PDIP Package	60	N/A
PLCC Package	65	N/A
SOIC Package	75	N/A

Storage Temperature Range . . . . . -65°C to +150°C
Maximum Junction Temperature Ceramic Package . . . . . +175°C
Maximum Junction Temperature Plastic Package . . . . . +150°C
Maximum Lead Temperature Package (Soldering 10s) . . . . +300°C
(PLCC and SOIC - Lean Tips Only)

Die Characteristics

Gate Count . . . . . 2250 Gates

CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

DC Electrical Specifications

V_{CC} = +5.0V ± 10%, T_A = 0°C to +70°C (C82C54, C82C54-10, C82C54-12)
T_A = -40°C to +85°C (I82C54, I82C54-10, I82C54-12)
T_A = -55°C to +125°C (M82C54, M82C54-10, M82C54-12

SYMBOL	PARAMETER	MIN	MAX	UNITS	TEST CONDITIONS
VIH	Logical One Input Voltage	2.0	-	V	C82C54, I82C54
		2.2	-	V	M82C54
VIL	Logical Zero Input Voltage	-	0.8	V	
VOH	Output HIGH Voltage	3.0	-	V	IOH = -2.5mA
		V_{CC} -0.4	-	V	IOH = -100µA
VOL	Output LOW Voltage	-	0.4	V	IOL = +2.5mA
II	Input Leakage Current	-1	+1	µA	VIN = GND or V_{CC} DIP Pins 9,11,14-16,18-23
IO	Output Leakage Current	-10	+10	µA	VOUT = GND or V_{CC} DIP Pins 1-8
ICCSB	Standby Power Supply Current	-	10	µA	V_{CC} = 5.5V, VIN = GND or V_{CC}, Outputs Open, Counters Programmed
ICCOP	Operating Power Supply Current	-	10	mA	V_{CC} = 5.5V, CLK0 = CLK1 = CLK2 = 8MHz, VIN = GND or V_{CC}, Outputs Open

Capacitance

T_A = +25°C; All Measurements Referenced to Device GND, Note 1

SYMBOL	PARAMETER	TYP	UNITS	TEST CONDITIONS
CIN	Input Capacitance	20	pF	FREQ = 1MHz
COUT	Output Capacitance	20	pF	FREQ = 1MHz
CI/O	I/O Capacitance	20	pF	FREQ = 1MHz

NOTE:

1. Not tested, but characterized at initial design and at major process/design changes.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Electrical Specifications V_{CC} = +5.0V ± 10%, T_A = 0°C to +70°C (C82C54, C82C54-10, C82C54-12)
T_A = -40°C to +85°C (I82C54, I82C54-10, I82C54-12)
T_A = -55°C to +125°C (M82C54, M82C54-10, M82C54-12)

SYMBOL		PARAMETER	82C54		82C54-10		82C54-12		UNITS	TEST CONDITIONS
			MIN	MAX	MIN	MAX	MIN	MAX		
READ CYCLE										
(1)	TAR	Address Stable Before $\overline{RD}$	30	-	25	-	25	-	ns	1
(2)	TSR	$\overline{CS}$ Stable Before $\overline{RD}$	0	-	0	-	0	-	ns	1
(3)	TRA	Address Hold Time After $\overline{RD}$	0	-	0	-	0	-	ns	1
(4)	TRR	$\overline{RD}$ Pulse Width	150	-	95	-	95	-	ns	1
(5)	TRD	Data Delay from $\overline{RD}$	-	120	-	85	-	85	ns	1
(6)	TAD	Data Delay from Address	-	210	-	185	-	185	ns	1
(7)	TDF	$\overline{RD}$ to Data Floating	5	85	5	65	5	65	ns	2, Note 1
(8)	TRV	Command Recovery Time	200	-	165	-	165	-	ns	
WRITE CYCLE										
(9)	TAW	Address Stable Before $\overline{WR}$	0	-	0	-	0	-	ns	
(10)	TSW	$\overline{CS}$ Stable Before $\overline{WR}$	0	-	0	-	0	-	ns	
(11)	TWA	Address Hold Time After $\overline{WR}$	0	-	0	-	0	-	ns	
(12)	TWW	$\overline{WR}$ Pulse Width	95	-	95	-	95	-	ns	
(13)	TDW	Data Setup Time Before $\overline{WR}$	140	-	95	-	95	-	ns	
(14)	TWD	Data Hold Time After $\overline{WR}$	25	-	0	-	0	-	ns	
(15)	TRV	Command Recovery Time	200	-	165	-	165	-	ns	
CLOCK AND GATE										
(16)	TCLK	Clock Period	125	DC	100	DC	80	DC	ns	1
(17)	TPWH	High Pulse Width	60	-	30	-	30	-	ns	1
(18)	TPWL	Low Pulse Width	60	-	40	-	30	-	ns	1
(19)	TR	Clock Rise Time	-	25	-	25	-	25	ns	
(20)	TF	Clock Fall Time	-	25	-	25	-	25	ns	
(21)	TGW	Gate Width High	50	-	50	-	50	-	ns	1
(22)	TGL	Gate Width Low	50	-	50	-	50	-	ns	1
(23)	TGS	Gate Setup Time to CLK	50	-	40	-	40	-	ns	1
(24)	TGH	Gate Hold Time After CLK	50	-	50	-	50	-	ns	1
(25)	TOD	Output Delay from CLK	-	150	-	100	-	100	ns	1
(26)	TODG	Output Delay from Gate	-	120	-	100	-	100	ns	1
(27)	TWO	OUT Delay from Mode Write	-	260	-	240	-	240	ns	1
(28)	TWC	CLK Delay for Loading	0	55	0	55	0	55	ns	1
(29)	TWG	Gate Delay for Sampling	-5	40	-5	40	-5	40	ns	1
(30)	TCL	CLK Setup for Count Latch	-40	40	-40	40	-40	40	ns	1

NOTE:

1. Not tested, but characterized at initial design and at major process/design changes.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Timing Waveforms

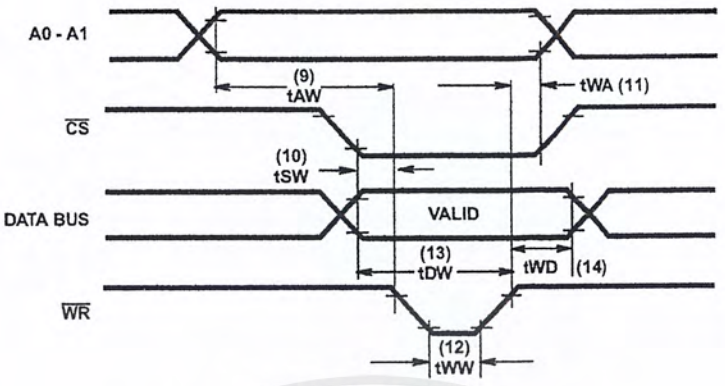


FIGURE 17. WRITE

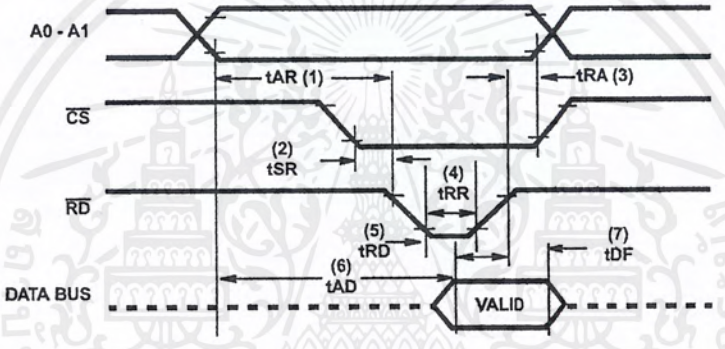


FIGURE 18. READ

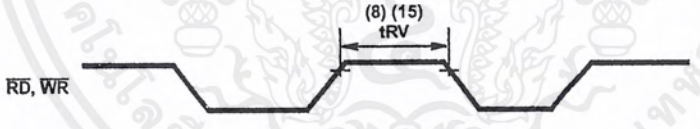


FIGURE 19. RECOVERY

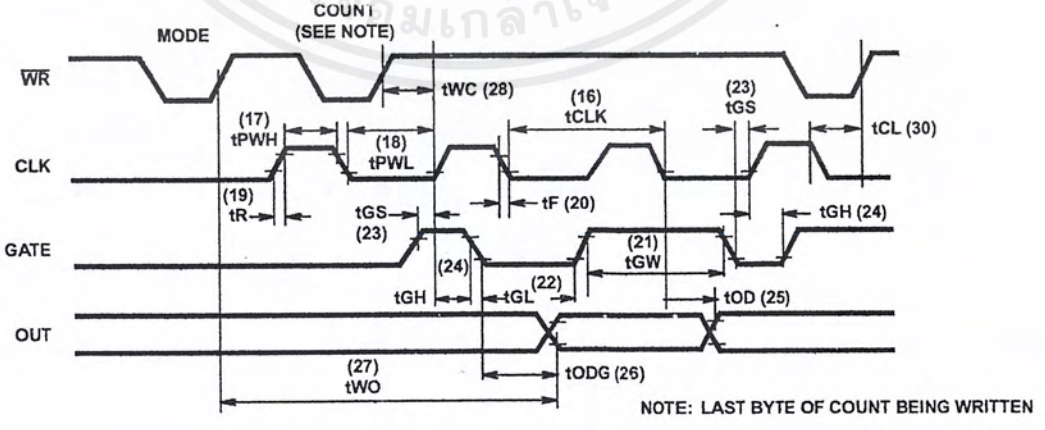
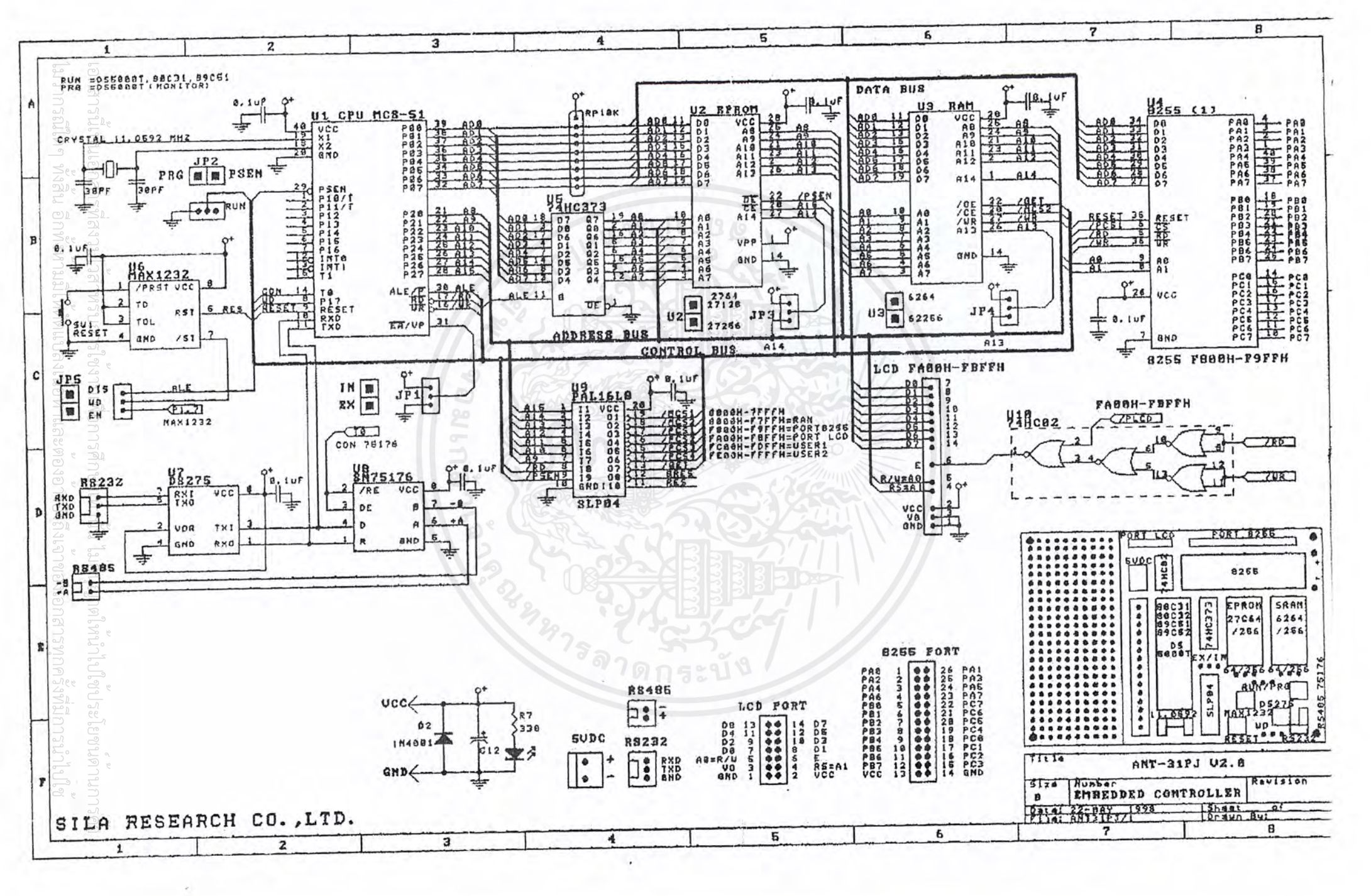


FIGURE 20. CLOCK AND GATE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


RUN =0D560001, 89C31, 89C51
PRG =0D560001 (MONITOR)
U1 CPU MCS-51
CRYSTAL 11.0592 MHZ
JP2
PRG
PSEN
RUN
U6 MAX1232
U5 74HC373
U2 EPROM
U3 RAM
U4 8255 (1)
DATA BUS
ADDRESS BUS
CONTROL BUS
U9 PAL16L8
SLP04
JP5
JP1
IN
EX
U7 DS275
U8 SN75176
RS232
RS485
LCD FA00H-FBFFH
U10 74HC02
FA00H-FBFFH
8255 F800H-F9FFH
8255 PORT
LCD PORT
VCC
R7
330
D2
IN4001
C12
GND
5VDC
RS232
RXD
TXD
GND
LCD PORT
8255 PORT
SILA RESEARCH CO.,LTD.
Title
ANT-31PJ V2.8
Size
B
Number
EMBEDDED CONTROLLER
Revision
Date: 22-MAY-998
Sheet of
File: ANT31PJ
Drawn By:

กิตติกรรมประกาศ

การที่รายงานฉบับนี้จะเสร็จสมบูรณ์ได้นั้น กระผมกราบขอขอบคุณท่าน อาจารย์ ผศ.ดร. วิจิตร กิณเรศ ซึ่งเป็นอาจารย์ที่ปรึกษา Project ท่านได้ให้คำแนะนำ คอยดูแล และให้คำปรึกษาวางแผนการปฏิบัติงานต่างๆ เกี่ยวกับการควบคุมเซอร์โวมอเตอร์ และขอขอบคุณพี่ ๆ ป.โท ที่ให้คำแนะนำที่ดี และเพื่อน ๆ น้อง ๆ ทุกคนที่ให้กำลังใจ คอยช่วยเหลือจนทำให้รายงานฉบับนี้สำเร็จสมบูรณ์ลงได้

สุดท้ายนี้ขอกราบขอบพระคุณ คุณพ่อ และคุณแม่ ผู้ซึ่งให้กำเนิด ให้การอบรมสั่งสอน และให้การสนับสนุนการศึกษามาโดยตลอด

นายทศพล ดีระกิตติสุวรรณ์
นายปรีชา คงถาวรธรรม
นายสุรัตน์ ตรีธนากร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. โยธิน เปรมปราณีรัตน์, "วิเคราะห์และออกแบบระบบการควบคุมมอเตอร์", สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, กรุงเทพมหานคร, 2526
2. สุวัฒน์ ดั่น, "เทคนิคและการออกแบบสวิตซ์ชิ่งเพาเวอร์ซัพพลาย" ,เอลเทลไทย จำกัด, กรุงเทพ, 2538
3. ธีรวัฒน์ ประกอบผล, "การประยุกต์ใช้งานไมโครโทรลเลอร์" , สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), กรุงเทพมหานคร,2542
4. Takasjhi Kenjo , "Power electronic for the Microprossor Age" , Oxford University Press , 1900
5. Rashid M.H. , "Prower electronic" , Prentice Hall,Ic, USA 1988, 585pp
6. สมยศ จุณณะปิยะ, "การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ ตระกูล MCS-51" สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, กรุงเทพมหานคร, 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้