

หุ่นยนต์ติดตามเป้าหมาย
ROBOT TRACKING OBJECT



โดย

นางสาวนริศา ลิทธิกันต์

เลขประจำตัว 40010368

นางสาวหนึ่งนุช เหลืองพิพัฒน์

เลขประจำตัว 40010929

4

เลขหมู่.....
เลขทะเบียน..... 42738
วัน, เดือน, ปี- 7 ส.ย. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

หุ่นยนต์ติดตามเป้าหมาย
ROBOT TRACKING OBJECT



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2543

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นยนต์ตรวจจำเป้าหมาย

ผู้จัดทำ

1. นางสาวนริศา สิริธิดานต์ เลขประจำตัว 40010368
2. นางสาวหนึ่งนุช เหลืองพิพัฒน์ เลขประจำตัว 40010929



(ผศ.ดร. สุรพันธ์ เอื้อไพบูลย์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์ตรวจจับเป้าหมาย

ROBOT TRACKING OBJECT

ผู้จัดทำ

- | | |
|---------------------------------|----------------------|
| 1. นางสาวนริศา สิทธิกานต์ | เลขประจำตัว 40010368 |
| 2. นางสาวหนึ่งนุช เหลืองพิพัฒน์ | เลขประจำตัว 40010929 |

โครงการได้รับการตรวจสอบแล้ว พร้อมทั้งจะทำการสอบได้



(ผศ.ดร. สุรพันธ์ เอื้อไพบูลย์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์ติดตามเป้าหมาย

นางสาวนริศา สิริทิถานต์

นางสาวหนึ่งนุช เหลืองพิพัฒน์

ผศ.ดร. สุรพันธ์ เอื้อไพบูลย์ อาจารย์ที่ปรึกษา

ปีการศึกษา 2543

บทคัดย่อ

ในการพัฒนาระบบการติดตามเป้าหมายโดยอัตโนมัติ ซึ่งสามารถนำมาใช้ในโรงงานอุตสาหกรรม หรือสถานที่ที่อันตรายที่มนุษย์ไม่สามารถเข้าไปได้ นับว่าเป็นโครงการที่น่าสนใจ แต่มีความยากในการประมวลผลภาพ ซึ่งในโครงการนี้เปรียบเสมือนการจำลองการทำงานของระบบ โดยขั้นตอนการทำงานเริ่มจากการรับภาพเป้าหมายที่ต้องการติดตามมาทางกล้องวิดีโอดิจิทัล เก็บข้อมูลเข้าสู่คอมพิวเตอร์ เมื่อวัตถุนั้นได้ทำการเคลื่อนที่ไป คอมพิวเตอร์จะทำการประมวลผลเพื่อหาตำแหน่งของวัตถุ แล้วส่งข้อมูลไปสั่งให้ไมโครคอนโทรลเลอร์ ทำการควบคุมการเคลื่อนที่ของรถให้ติดตามเป้าหมายนั้นไป

และในการทำโครงการนี้รถสามารถติดตามวัตถุที่ใช้เป็นเป้าหมายได้อย่างถูกต้อง แต่มีข้อจำกัดตรงที่ต้องใช้สีของวัตถุเป็นสีทึบ และควรมีสีเข้มกว่าพื้นหลังมาก รวมทั้งการเคลื่อนที่ของวัตถุต้องมีความเร็วในการเคลื่อนที่จำกัด เพื่อให้การประมวลผลภาพถูกต้องมากขึ้น

ROBOT TRACKING OBJECT

Narisa Sitthikan

Nuengnut Luengpipat

Dr.Surapan Airphaiboon Advisor

2000

ABSTRACT

Developing of a vision-based object detection system is very interesting but we are facing the difficult in a process to give the picture out. This project uses as a tracking object robot , by tracking the image of the object form digital video camera on vehicle. Then take the movement of the object image to process system so that a computer acknowledge situation. The vehicle movement is controller by microcontroller. This robot can fallow quite well, but we face the problems about colors between object and background. Object's colors should be stronger than background's color and the movement of the robot should not be so far better processing.

สารบัญ

	หน้า
บทคัดย่อ	
Abstract	
บทที่ 1 บทนำ	1
บทที่ 2	2
2.1 การใช้งานไมโครคอนโทรลเลอร์	5
2.1.1 ไทม์เมอร์ / เคาน์เตอร์	6
2.1.2 รีจิสเตอร์สำหรับใช้งานทั่วไปใน MCS-51	9
2.1.3 รีจิสเตอร์ใช้งานเฉพาะใน MCS-51	9
2.1.4 สถาปัตยกรรมของไมโครคอนโทรลเลอร์ 89C52	11
2.2 มอเตอร์กระแสตรง (Direct Current Motors)	12
2.2.1 หลักการทำงานของ MOTOR (MOTOR'S PRINCIPLE)	12
2.2.2 ความสำคัญของแรงเคลื่อนไฟสวน (SINIFICANCE OF THE BACK E.M.F.)	14
2.2.3 แรงบิด (Torque)	15
2.2.4 การแยกประเภทของมอเตอร์กระแสตรง	16
2.3 ความรู้เบื้องต้นเกี่ยวกับพอร์ทขนาน	18
2.3.1 ทำไมถึงเลือกใช้งานพอร์ทขนาน	19
2.3.2 ความรู้เบื้องต้นเกี่ยวกับพอร์ทขนาน	20
2.3.3 ลักษณะทางกายภาพของพอร์ทขนาน	20
2.3.4 การนำพอร์ทไปใช้งาน	25
2.3.5 การเขียนโปรแกรมติดต่อกับพอร์ทขนาน	25
2.4 กระบวนผลภาพ	26
2.4.1 การประมวลผลภาพเชิงตัวเลข (Digital Image Processing)	26
2.4.1.1 การแทนภาพด้วยข้อมูลแบบดิจิตอล	26
2.4.1.2 ลักษณะการจัดเก็บข้อมูลภาพแบบดิจิตอล	27
2.4.2 สัญญาณข้อมูลภาพจากดิจิตอลวิดีโอ	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3	หลักการทำงานของระบบ	29
3.1	โครงสร้างของระบบการติดตามเป้าหมาย	29
3.2	โครงสร้างของโปรแกรม	30
3.2.1	โปรแกรมการทำงานของ Delphi	30
3.2.2	โปรแกรมการทำงานของไมโครคอนโทรลเลอร์	34
3.3	วงจรขั้วมอเตอร์กระแสตรง	35
บทที่ 4	การทดลองและผลการทดลอง	37
4.1	การทดลองเพื่อทดสอบการทำงานของโปรแกรมเมื่อเป้าหมายมีการเคลื่อนที่	37
4.2	การทดลองและผลที่ได้จากการควบคุมมอเตอร์กระแสตรง	42
4.3	การทดลองในหัวข้ออื่น ๆ ที่มีผลต่อการทำงานของระบบ	42
บทที่ 5	สรุปและวิจารณ์	44
ภาคผนวก ก		
ภาคผนวก ข		
ภาคผนวก ค		
กิตติกรรมประกาศ		
เอกสารอ้างอิง		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

		หน้า
บทที่ 1	บทนำ	
บทที่ 2		
	รูปที่ 2.1 รีจิสเตอร์ใช้งานเฉพาะ TMOD	3
	รูปที่ 2.2 การทำงานของไทม์เมอร์/เคาน์เตอร์1 ในโหมด0(13 บิตเคาน์เตอร์)	4
	รูปที่ 2.3 แสดง โครงสร้างแต่ละบิตของรีจิสเตอร์ PSW	6
	รูปที่ 2.4 แสดง โครงสร้างแต่ละบิตของรีจิสเตอร์ TCON	7
	รูปที่ 2.5 แสดง โครงสร้างแต่ละบิตของรีจิสเตอร์ TMOD	7
	รูปที่ 2.6 แสดง โครงสร้างภายในชิพไมโครคอนโทรลเลอร์ 89C52	8
	รูปที่ 2.7 แสดง ไดอะแกรมขาของ 89C52	9
	รูปที่ 2.8 วงจรออสซิลเลเตอร์ภายใน	9
	รูปที่ 2.9 89C52 ที่ทำงานโดยสัญญาณที่มาจากภายนอก	10
	รูปที่ 2.10 แสดงวงจรการรีเซตภายหลังเริ่มจ่ายพลังงาน	10
	รูปที่ 2.11 แสดงภาพโครงสร้างแต่ละบิตของพอร์ทใน MCS-51	12
	รูปที่ 2.12 แสดงทิศทางของแรงที่เกิดขึ้นในตัวนำได้โดยใช้กฎมือซ้าย (left-hand-law) ของเฟลมมิ่ง	13
	รูปที่ 2.13 หลักการของมอเตอร์	14
	รูปที่ 2.14 (a) วงจรของมอเตอร์ไฟตรง	14
	(b) วงจรสมมูลย์ของมอเตอร์ที่เขียนแทนแรงเคลื่อนไฟสวน ได้ด้วยเบตเตอร์ E_b	
	รูปที่ 2.15 แสดงการเกิดแรงบิดในมอเตอร์กระแสตรง	15
	รูปที่ 2.16 มอเตอร์กระแสตรงแบบอามเจอร์ต่ออนุกรมกับสนามแม่เหล็ก	16
	รูปที่ 2.17 ความสัมพันธ์ระหว่างความเร็วและแรงบิดของมอเตอร์อนุกรม ที่แรงดันคงที่	16
	รูปที่ 2.18 โครงสร้างของมอเตอร์กระแสตรงแบบสนามแม่เหล็กแยกกระตุ้น	17
	รูปที่ 2.19 ความสัมพันธ์ระหว่างแรงบิดกับความเร็วของซันท์มอเตอร์ภายใน ได้สนามแม่เหล็กและแรงดันคงที่	17
	รูปที่ 2.20 มอเตอร์กระแสตรงแบบฟิวด์แม่เหล็กถาวร	18

รูปที่ 2.22	แสดงระบบบัสภายในของพอร์ทขนาน	22
รูปที่ 2.23	แสดงตำแหน่งขาภายในพอร์ทขนาน	24
รูปที่ 2.24	แผนภาพตัวอย่างการส่งสัญญาณวิธีโอแตร้า 24 เฟรมต่อวินาที	28
บทที่ 3 หลักการทำงานของระบบ		
รูปที่ 3.1	โครงสร้างของระบบ	29
รูปที่ 3.2	แผนภาพการทำงานหลักของโปรแกรม Delphi	31
รูปที่ 3.3	แผนภาพการประมวลผลเพื่อหาตำแหน่งที่วัตถุได้เคลื่อนที่ไป	32
รูปที่ 3.4	แสดงภาพสรุปตำแหน่งของเป้าหมายที่เคลื่อนที่ไปได้ในเฟรมภาพ	33
รูปที่ 3.5	แผนภาพการทำงานของไมโครคอนโทรลเลอร์	34
รูปที่ 3.6	แสดงวงจรขับมอเตอร์กระแสตรง	35
รูปที่ 3.7	แสดงขาการต่อวงจรขับมอเตอร์กับ IC เบอร์ 74LS373	35
รูปที่ 3.8	แสดงวงจรโครงสร้างภายใน IC เบอร์ LMD18245	36
รูปที่ 3.9	แสดงลักษณะการไหลของกระแสในวงจร DMOS H-Bridge	36
บทที่ 4		
รูปที่ 4.1	ภาพหน้าจอของโปรแกรมเมื่อเป้าหมายอยู่ตรงกลาง	37
รูปที่ 4.2	ภาพหน้าจอของโปรแกรมเมื่อเป้าหมายอยู่ด้านบน	38
รูปที่ 4.3	ภาพหน้าจอของโปรแกรมเมื่อเป้าหมายอยู่ด้านล่าง	39
รูปที่ 4.4	ภาพหน้าจอของโปรแกรมเมื่อเป้าหมายอยู่ด้านซ้าย	40
รูปที่ 4.5	ภาพหน้าจอของโปรแกรมเมื่อเป้าหมายอยู่ด้านขวา	41

บทที่ 1

บทนำ

เนื่องจากการศึกษาการติดตามเป้าหมายโดยอัตโนมัติ โดยอาศัยหลักการประมวลผลภาพ เป็นเทคโนโลยีที่น่าสนใจเพราะสามารถพัฒนาเพื่อใช้ประโยชน์ในโรงงานอุตสาหกรรมหรือการติดตามหาเป้าหมายในที่ที่อันตรายที่มนุษย์ไม่สามารถเข้าไปได้

วัตถุประสงค์

เพื่อให้รถติดตามเป้าหมายได้โดยอัตโนมัติ ซึ่ง การติดตามเป้าหมายนี้จะใช้กล้องวิดีโอ ดิจิตอลในการตรวจจับเป้าหมายแล้วประมวลผลภาพโดยใช้คอมพิวเตอร์เพื่อหาตำแหน่งที่เป้าหมาย ได้เคลื่อนที่ไป แล้วจึงส่งข้อมูลไปยังไมโครคอนโทรลเลอร์เพื่อควบคุมการทำงานของรถให้ เคลื่อนที่ไปยังหาเป้าหมายได้ถูกต้อง โดยระบบการขับเคลื่อนจะใช้มอเตอร์กระแสตรง

ขอบเขตของโครงการ

- ใช้วัตถุที่มีขนาดใหญ่พอสมควรเป็นเป้าหมาย โดยเคลื่อนที่อยู่บนพื้นสีขาว
- ต้องควบคุมสิ่งแวดล้อมรอบข้างให้มีความเหมาะสม เช่น แสง ความเร็วในการเคลื่อนที่ของเป้าหมายต้องไม่เร็วเกินไป เพื่อให้คอมพิวเตอร์สามารถประมวลผลภาพได้ทัน เพื่อให้ได้ตำแหน่งที่เป้าหมายเคลื่อนที่ไปได้อย่างแน่นอน

บทที่ 2

2.1 การใช้งานไมโครคอนโทรลเลอร์

MCS-51 เป็นไมโครคอนโทรลเลอร์แบบชิพเดี่ยวที่มีข้อดีเมื่อเทียบกับไมโครคอนโทรลเลอร์ขนาด 8 บิต ของตระกูลอื่น ๆ ดังนี้

1. มีหน่วยความจำสำหรับเก็บข้อมูลทั่วไป (RAM) บรรจุไว้ภายใน 128-256 ไบต์
2. มีหน่วยความจำสำหรับเก็บโปรแกรมควบคุมการทำงานอยู่ภายในจำนวน 4 กิโลไบต์
3. มีวงจรตั้งเวลาวางจรนับขนาด 16 บิต 2 ตัว อยู่ภายใน
4. มีสัญญาณนาฬิกาภายในตัว
5. มีวงจรรับส่งข้อมูลอนุกรมได้ 2 ทิศทาง
6. มีพอร์ตที่สามารถรับหรือส่งข้อมูลได้ 2 ทิศทางจำนวน 4 พอร์ต พอร์ตละ 8 บิต

นอกจากนี้ MCS-51 ยังมีคุณสมบัติอื่น ๆ ที่น่าสนใจ คือ

- ต้องการแหล่งจ่ายไฟ 5 โวลต์เพียงชุดเดียว
- มีหน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลที่อยู่ภายในชิพได้อย่างละ 64 กิโลไบต์
- มีคำสั่งคูณและหารเลขขนาด 8 บิต ในตัวเอง
- จัดลำดับความสำคัญของสัญญาณอินเตอร์รัปต์ได้ 2 ระดับ
- รับและส่งข้อมูลแบบอนุกรมได้ในตัว โดยสามารถกำหนดอัตราเร็วในการรับส่งข้อมูลได้ตั้งแต่ 300 ถึง 375 กิโลไบต์ต่อวินาที
- สามารถประมวลผลแบบบูลีนเพื่อใช้งานควบคุมโดยเฉพาะ
- มีรีจิสเตอร์สำหรับใช้งานเป็น ไทม์เมอร์หรือเคาน์เตอร์เพื่อนับจำนวนสัญญาณนาฬิกาภายในชิพ หรือนับการเปลี่ยนแปลงสถานะของสัญญาณนอกขนาด 16 บิต จำนวน 2 ตัว เพื่อใช้สำหรับนับจำนวนพัลส์ วัดความกว้างของพัลส์หรือใช้วัดช่วงเวลา
- หน่วยความจำสำหรับเก็บข้อมูลภายในบางส่วนสามารถเข้าถึงข้อมูลได้ทั้งระดับไบต์และระดับบิตเพื่อให้การออกแบบโปรแกรมและการควบคุมระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

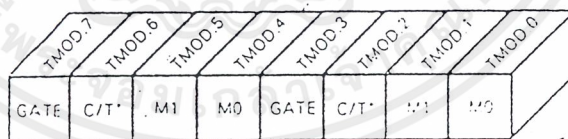
2.1.1 ไทม์เมอร์ / เคนต์เตอร์

ไมโครคอนโทรลเลอร์ในตระกูลมีรีจิสเตอร์พิเศษที่สามารถเลือกใช้งานเป็นไทม์เมอร์หรือ เคนต์เตอร์อย่างใดอย่างหนึ่ง รีจิสเตอร์ประเภทนี้มีอยู่ด้วยกัน 2 ตัว แต่ละตัวขนาด 16 บิต เรียก ไทม์เมอร์ 0 และ ไทม์เมอร์ 1 ตามลำดับ

- ไทม์เมอร์นั้นมีค่าในรีจิสเตอร์ที่ใช้เป็นไทม์เมอร์ที่ถูกเลือกใช้งานจะถูกเพิ่มขึ้นทุกแมกซ์ซิมัแซเคิล
- เคนต์เตอร์นั้นค่าในรีจิสเตอร์ที่ใช้เป็นเคนต์เตอร์ที่ถูกเลือกใช้งานจะถูกเพิ่มค่าทีละ 1 เมื่อมีการเปลี่ยนสถานะ

ไทม์เมอร์ 0 และ ไทม์เมอร์ 0

สามารถเลือกการทำงานเป็นไทม์เมอร์หรือเคนต์เตอร์ได้โดยการกำหนดค่าบิต ในรีจิสเตอร์ใช้งานเฉพาะ ดังรูปที่ 2.1 (ไทม์เมอร์ 0 ใช้บิต 2 ส่วน ไทม์เมอร์ 1 ใช้บิต 6) โดยหากบิตนี้มีค่าเป็น 0 หมายถึงเลือกใช้งานเป็นไทม์เมอร์ถ้าบิตนี้มีค่าเป็น 1 หมายถึงเลือกใช้งานเป็นเคนต์เตอร์

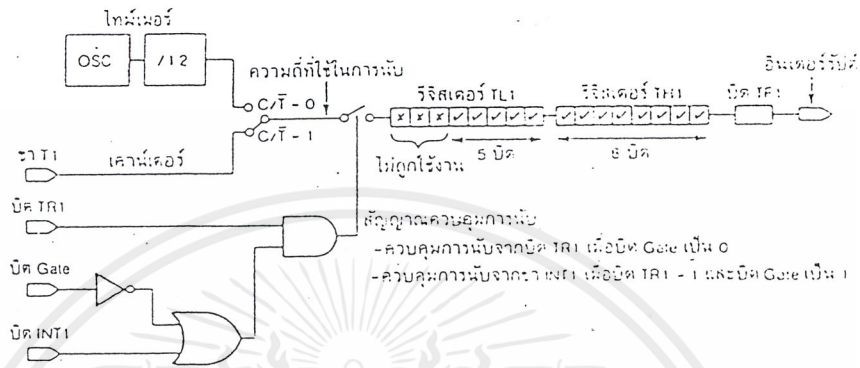


รูปที่ 2.1 รีจิสเตอร์ใช้งานเฉพาะ TMOD

นอกจากนี้จะเลือกการทำงานของรีจิสเตอร์ให้เป็นไทม์เมอร์หรือเคนต์เตอร์ได้แล้วในแต่ละการทำงานยังมีการทำงานย่อยอยู่อีก 4 แบบ ตามความเหมาะสมของการใช้งาน

โหมด 0 จะใช้รีจิสเตอร์ขนาด 8 บิตเป็นตัวนับโดยมีการเพิ่มค่าครั้งละ 1 ทุกครั้งนับสัญญาณได้ครบ 32 ครั้ง โดยในโหมดนี้รีจิสเตอร์ที่ใช้นับเพียง 13 บิต (8 บิตในรีจิสเตอร์ THx รวมกับ 5 บิตใน THx) ดังรูปที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 การทำงานของไทม์เมอร์/เคาน์เตอร์ 1 ในโหมด 0 (13 บิตเคาน์เตอร์)

โหมด 1 การทำงานเหมือนโหมด 0 เว้นแต่ค่าในรีจิสเตอร์ถูกใช้งานครบทั้ง 16 บิตนั่นเอง คือไทม์เมอร์หรือเคาน์เตอร์ในโหมดนี้มีขนาด 16 บิต

โหมด 2 ในโหมดนี้จะกำหนดรีจิสเตอร์ใช้งานในการนับเพียง 8 บิต (จากรีจิสเตอร์ TLx) ที่มีการโหลดค่าด้วยค่าในรีจิสเตอร์ THx การใช้งานโหมดนี้มีไว้เพื่อสร้างสัญญาณอินเตอร์รัปต์ที่มีความยาวคงที่

โหมด 3 ในโหมดนี้ไทม์เมอร์ 1 จะไม่มีการนับ แต่ไทม์เมอร์จะบังคับให้รีจิสเตอร์ TL0 ของไทม์เมอร์ 0 ถูกใช้เป็นไทม์เมอร์เพียงอย่างเดียว การทำงานโหมด 3 มีไว้เพื่อใช้งานที่ต้องการไทม์เมอร์หรือเคาน์เตอร์ขนาด 8 บิตเพิ่มมากขึ้น

การกำหนดการใช้ไทม์เมอร์ 0 หรือไทม์เมอร์ 1

ในแต่ละตารางสมมติว่าไทม์เมอร์ 0 หรือไทม์เมอร์ 1 ถูกใช้เพียงตัวเดียวเท่านั้น หากต้องใช้ไทม์เมอร์ 0 หรือไทม์เมอร์ 1 พร้อมกันให้นำค่าในตารางไทม์เมอร์ 0 (ตารางที่ 2.1 และ 2.2) มาบวกกับค่าในตารางไทม์เมอร์ 1 (ตารางที่ 2.3 และ 2.4) และหากใช้ไทม์เมอร์หรือเคาน์เตอร์ที่ควบคุมการทำงานจากโปรแกรม จะต้องเซตบิต TRx ในรีจิสเตอร์ใช้งานเฉพาะ TCON เพื่อให้ไทม์เมอร์เริ่มทำงานเอง

ตารางที่ 2.1 แสดงค่าในรีจิสเตอร์ TMOD ค่าต่าง ๆ สำหรับ Timer 0

โหมด	ฟังก์ชันไทม์เมอร์ 0	ควบคุมจากโปรแกรม	TMOD ควบคุมจากฮาร์ดแวร์ภายนอก
0	13 bit Timer	00H	08H
1	16 bit Timer	01H	09H
2	8 bit Auto Reload	02H	0AH
3	two 8 bit Timer	03H	0BH

ตารางที่ 2.2 แสดงค่าในรีจิสเตอร์ TMOD ค่าต่าง ๆ สำหรับ Counter 0

โหมด	ฟังก์ชันไทม์เมอร์ 0	ควบคุมจากโปรแกรม	TMOD ควบคุมจากฮาร์ดแวร์ภายนอก
0	13 bit Timer	04H	0CH
1	16 bit Timer	05H	0DH
2	8 bit Auto Reload	06H	0EH
3	two 8 bit Timer	07H	0FH

ตารางที่ 2.3 แสดงค่าในรีจิสเตอร์ TMOD ค่าต่าง ๆ สำหรับ Timer 1

โหมด	ฟังก์ชันไทม์เมอร์ 0	ควบคุมจากโปรแกรม	TMOD ควบคุมจากฮาร์ดแวร์ภายนอก
0	13 bit Timer	00H	80H
1	16 bit Timer	10H	90H
2	8 bit Auto Reload	20H	A0H
3	does not run	30H	B0H

ตารางที่ 2.4 แสดงค่าในรีจิสเตอร์ TMOD ค่าต่าง ๆ สำหรับ Counter 1

โหมด	ฟังก์ชันไทม์เมอร์ 0	ควบคุมจากโปรแกรม	TMOD ควบคุมจากฮาร์ดแวร์ภายนอก
0	13 bit Timer	40H	C0H
1	16 bit Timer	50H	D0H
2	8 bit Auto Reload	60H	E0H
3	not available	--	--

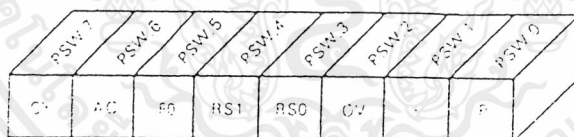
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 รีจิสเตอร์สำหรับใช้งานทั่วไปใน MCS-51

รีจิสเตอร์ A,B และรีจิสเตอร์ใช้งานทั่วไป R0-R7 ซึ่งอยู่ในหน่วยความจำสำหรับเก็บข้อมูลทั่วไปภายในชิพบริเวณ 128 ไบต์แรก รีจิสเตอร์ใช้งานทั่วไป R0-R7 ใน MCS-51 มีอยู่ด้วยกันทั้งหมด 4 กลุ่ม แต่ละกลุ่มประกอบด้วยรีจิสเตอร์จำนวน 8 ตัว (R0-R7) ซึ่งมีชื่อเรียกเหมือนกัน ดังนั้นรีจิสเตอร์ใช้งานทั่วไป R0-R7 ใน MCS-51 จึงมีทั้งหมด 32 ตัวในการทำงานขณะใด ๆ รีจิสเตอร์ทั้ง 4 กลุ่มใดกลุ่มหนึ่งใน 4 กลุ่ม กระทำโดยการเซตหรือเคลียร์บิต RS0,RS1 ในรีจิสเตอร์ใช้งานเฉพาะ PSW

2.1.3 รีจิสเตอร์ใช้งานเฉพาะใน MCS-51

รีจิสเตอร์ใช้งานเฉพาะ PSW (Program Status Word) เข้าถึงข้อมูลได้ในระดับบิต ดังแสดงในรูป 2.3

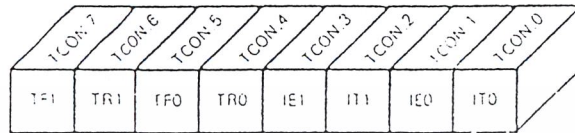


รูปที่ 2.3 แสดง โครงสร้างแต่ละบิตของรีจิสเตอร์ PSW

PSW.7 CY	carry flag
PSW.6 AC	auxillary carry flag
PSW.5 F0	flag 0 เป็นบิตบอกสถานะที่ผู้ใช้สามารถกำหนดการใช้งานได้เอง
PSW.4 RS1	ใช้เลือกกลุ่มรีจิสเตอร์ R1-R7
PSW.3 RS0	ใช้เลือกกลุ่มรีจิสเตอร์ R1-R7
PSW.2 OV	บิตแสดงการเกิด overflow
PSW.1 _	บิตที่ผู้ใช้กำหนดการใช้งานได้เอง
PSW.0 P	parity flag

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

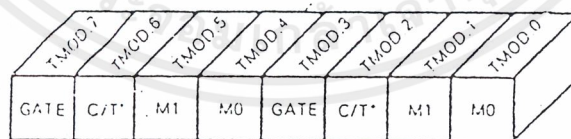
รีจิสเตอร์ใช้งานเฉพาะ TCON (Timer/Counter Control Register) เข้าถึงข้อมูลได้ในระดับ บิตดังรูปที่ 2.4



รูปที่ 2.4 แสดง โครงสร้างแต่ละบิตของรีจิสเตอร์ TCON

TCON.7	TF1	บิตแสดงการ iverflow ของโทม์เมอร์ 1
TCON.6	TR1	บิตควบคุมการนับของโทม์เมอร์ 1
TCON.5	TF0	บิตแสดงการ iverflow ของโทม์เมอร์ 0
TCON.4	TR0	บิตควบคุมการนับของโทม์เมอร์ 0
TCON.3	IE1	บิตแสดงสถานะสัญญาณอินเตอร์รัปต์ภายนอกชนิดที่ 1
TCON.2	IT1	บิตเลือกประเภทการตรวจสอบสัญญาณอินเตอร์รัปต์ที่เกิดขึ้นที่ขา INT1
TCON.1	IE0	บิตแสดงสถานะสัญญาณอินเตอร์รัปต์ภายนอกชนิดที่ 0
TCON.0	IT0	บิตเลือกประเภทการตรวจสอบสัญญาณอินเตอร์รัปต์ที่เกิดขึ้นที่ขา INT0

รีจิสเตอร์ใช้งานเฉพาะ TMOD (Timer/Counter Mode Control Register) ไม่สามารถเข้าถึงข้อมูลได้ในระดับบิตดังรูปที่ 2.5



รูปที่ 2.5 แสดง โครงสร้างแต่ละบิตของรีจิสเตอร์ TMOD

- GATE บิตเลือกการควบคุมให้รีจิสเตอร์สำหรับใช้เป็นโทม์เมอร์หรือเคาน์เตอร์ทำงานโดยควบคุมจากฮาร์ดแวร์หรือซอฟต์แวร์
- C/T บิตเลือกการทำงานของรีจิสเตอร์สำหรับใช้เป็นโทม์เมอร์หรือเคาน์เตอร์
- 0 ทำงานเป็นโทม์เมอร์
 - 1 ทำงานเป็นเคาน์เตอร์

M1 บิตสำหรับเลือกโหมดการทำงานของโทม์เมอร์ 0 หรือ โทม์เมอร์ 1
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

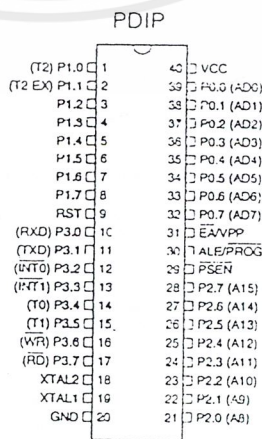
- M0 บิทสำหรับเลือกโหมดการทำงานของไทม์เมอร์ 0 หรือ ไทม์เมอร์ 1
- 00 โหมด 0
 - 01 โหมด 1
 - 10 โหมด 2
 - 11 โหมด 3

2.1.4 สถาปัตยกรรมของไมโครคอนโทรลเลอร์ 89C52

โครงสร้างภายในชิพไมโครคอนโทรลเลอร์ 89C52 ชิปเดี่ยวแสดงดังรูปที่ 2.7 ซึ่งอธิบายถึงส่วนย่อย ๆ ภายใน 89C52

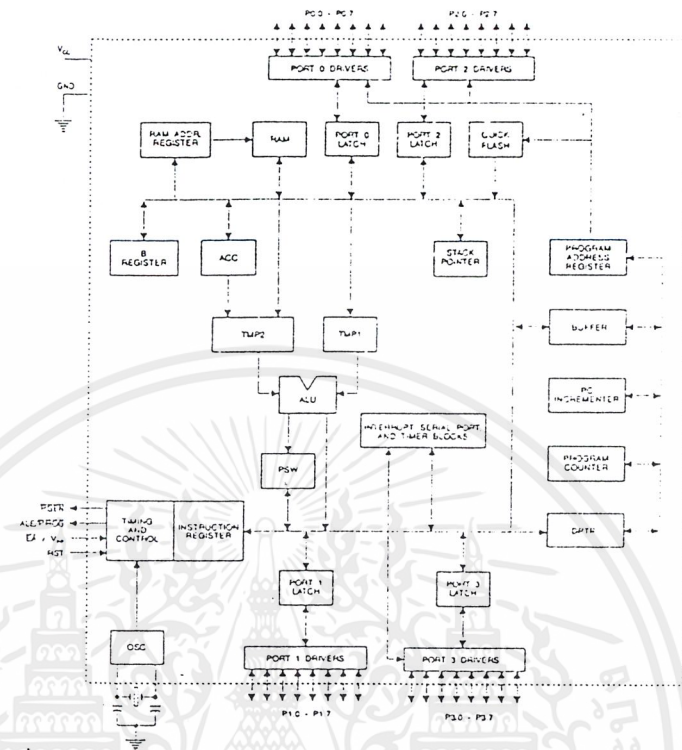
สัญญาณจากภายในจะต่อออกสู่ภายนอกทางขาของ 89C52 ที่มีอยู่ 40 ขา ดังรูปที่ 2.6 ซึ่งมีการใช้งานต่าง ๆ ดังนี้

- VCC ขา 40 เป็นขาที่ต้องป้อนไฟเลี้ยง +5 โวลต์ เข้าไปเพื่อให้วงจรสามารถทำงานได้
- VSS ขา 20 เป็นขาที่ต้องต่อกับกราวด์ของแหล่งจ่ายไฟ
- RST ขา 9 เป็นขาที่รีเซ็ตการทำงานของ 8051 ถ้าป้อนสัญญาณที่มีสถานะลอจิก 1 ที่ขานี้จะเป็นการรีเซ็ตการทำงานเพื่อกลับไปเริ่มการทำงานจากคำสั่งที่อยู่ในหน่วยความจำตำแหน่ง 0000
- ALE ขา 30 ใช้เป็นขาส่งสัญญาณออกไปภายนอก เพื่อควบคุมการแลตซ์ค่าตำแหน่งไบต์คำสั่งจากพอร์ท 0 ในระหว่างการติดต่อหน่วยความจำสำหรับเก็บโปรแกรมหรือข้อมูลภายนอก
- PSEN ขา 29 ใช้ส่งสัญญาณเพื่ออ่านคำสั่งจากโปรแกรมที่เก็บไว้ในหน่วยความจำภายในหน่วยความจำภายนอกชิพ
- XTAL1 ขา 19 ใช้ต่อคริสตอลภายนอก โดยเป็นอินพุทเข้าสู่วงจรรอสซิลเลเตอร์
- XTAL2 ขา 18 ใช้ต่อคริสตอลภายนอก โดยเป็นเอาต์พุทออกจากวงจรรอสซิลเลเตอร์ ดังรูปที่ 2.6

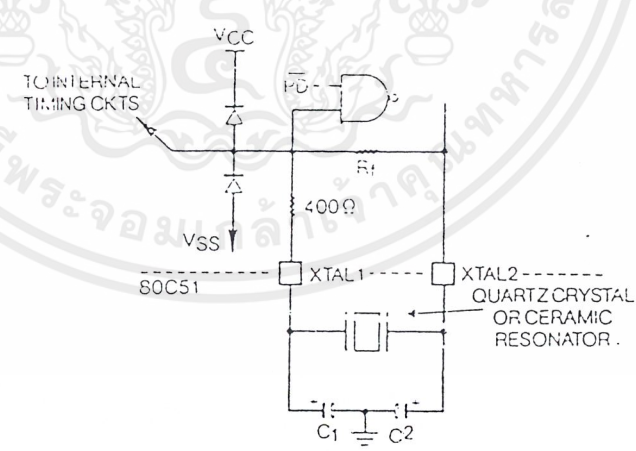


รูปที่ 2.6 แสดงไดอะแกรมขาของ 89C52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาก็เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

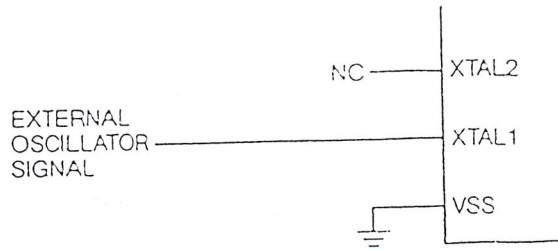


รูปที่ 2.7 แสดงโครงสร้างภายในชิปไมโครคอนโทรลเลอร์ 89C52



รูปที่ 2.8 วงจรออสซิลเลเตอร์ภายใน
ถ้าใช้สัญญาณนาฬิกาจากภายนอกมาเป็นสัญญาณนาฬิกาให้ปล่อยขา XTAL2 ลอยไว้แล้ว
ต่อสัญญาณนาฬิกาเข้าขา XTAL1 ดังรูปที่ 2.9

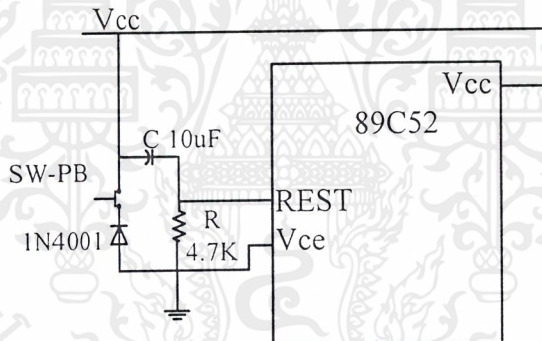
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 89C52 ที่ทำงานโดยสัญญาณที่มาจากภายนอก

การรีเซ็ต 89C52 ภายหลังเริ่มจ่ายพลังงาน

เมื่อเริ่มจ่ายพลังงานจะต้องมีการรีเซ็ตก่อนเสมอ การรีเซ็ตในขณะที่เริ่มจ่ายพลังงานจะกระทำโดยอัติโนมัตีด้วยการต่อวงจรแสดงดังรูป 2.10



รูปที่ 2.10 แสดงวงจรการรีเซ็ตภายหลังเริ่มจ่ายพลังงาน

การทำงานของวงจรรีเซ็ตมีดังนี้

เมื่อเริ่มจ่ายพลังงานให้แก่ MCS-51 จะทำให้ขา RST มีสถานะเป็นหนึ่งในช่วงเวลาหนึ่งซึ่งขึ้นกับค่าตัวเก็บประจุ ความเร็วในการประจุ เพื่อให้แน่ใจว่ามีการรีเซ็ตเกิดขึ้นจริงๆ ในช่วงเริ่มต้นให้ขา VCC จะต้องให้ขา RST มีสถานะเป็น 1 นานพอที่จะให้วงจรออสซิลเลเตอร์เริ่มต้นทำงานรวมกับช่วงเวลา 2 เมกซีไนซ์เคล็ด

ในช่วงเริ่มต้นให้ VCC แก่ MCS-51 (ช่วง power up) VCC ควรมีค่าตามที่กำหนดภายในเวลา 10 วินาที (rise time) โดยช่วงเวลาที่วงจรออสซิลเลเตอร์เริ่มต้นทำงานจะขึ้นอยู่กับความถี่ของคริสตัลที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- PORT 0 เป็นพอร์ตนานขนาด 8 บิตที่อยู่ที่ยา 39 ถึงยา 32 เริ่มจากบิต 0 ถึง บิต 7 ตามลำดับ พอร์ต 0 นี้ใช้ได้ทั้งการรับส่งตำแหน่งและข้อมูลกับหน่วยความจำหรือใช้รับส่งข้อมูลก็ได้ นอกจากนี้ยังใช้งานได้หลายอย่างดังนี้

1. ใช้สำหรับส่งค่าตำแหน่งหน่วยความจำที่ต้องการติดต่อกับ โดย 8 บิตล่างถูกส่งออกไปทางพอร์ต 0 และ 8 บิตบนถูกส่งออกไปทางพอร์ต 2
2. ใช้รับส่งข้อมูลกับ Data Memory หรือใช้รับข้อมูลจาก Program Memory
3. ใช้รับส่งข้อมูลออกจากพอร์ตโดยตรง

- PORT 1 เป็นพอร์ตนานขนาด 8 บิต อยู่ที่ยา 1 ถึง 8 เริ่มจากบิต 0 ถึงบิต 7 ตามลำดับ ใช้ทำหน้าที่เป็นตัวรับส่งข้อมูลเท่านั้น ไม่สามารถส่งตำแหน่งได้

- PORT 2 เป็นพอร์ตนานขนาด 8 บิตอยู่ที่ยา 21 ถึง 28 เริ่มจากบิต 0 ถึงบิต 7 ตามลำดับ ใช้งานเพียง 2 ลักษณะคือ

1. ใช้ส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อกับทำงานร่วมกับ พอร์ต 0
2. ใช้เป็นพอร์ตรับส่งข้อมูลกับภายนอก

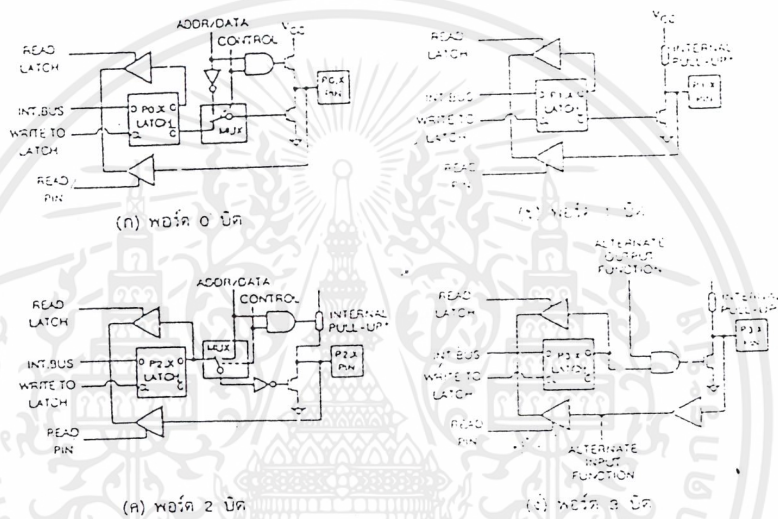
- PORT 3 เป็นพอร์ตนานขนาด 8 บิตอยู่ที่ยา 10 ถึง 17 เริ่มจากบิต 0 ถึงบิต 7 ตามลำดับ นอกจากนี้จะใช้งานเหมือนพอร์ตอื่น ๆ แล้วยังใช้งานอื่น โดยใช้คำสั่งควบคุมดังนี้

- | | |
|------------|--|
| P3.0(RxD) | เป็นขาที่รับข้อมูลแบบอนุกรม |
| P3.1(TxD) | เป็นขาที่รับข้อมูลแบบอนุกรม |
| P3.2(INT0) | ใช้รับสัญญาณขัดจังหวะจากภายนอก |
| P3.3(INT1) | ใช้รับสัญญาณขัดจังหวะจากภายใน |
| P3.4(TO) | ใช้เป็นขาจับสัญญาณให้เคาน์เตอร์ของไทม์เมอร์ 0 |
| P3.5(T1) | ใช้เป็นขาจับสัญญาณให้เคาน์เตอร์ของไทม์เมอร์ 1 |
| P3.6(WR) | ใช้เป็นขาจับสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก |
| P3.7(RD) | ใช้เป็นขาจับสัญญาณควบคุมการอ่านข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก |

การกำหนดการใช้งานอินพุท/เอาต์พุท (I/O Configuration)

พอร์ต 1, 2 และ 3 มีวงจรพูลอัพ (วงจรยกระดับแรงดันของสัญญาณ) ภายใน (Internal Pullups) พอร์ต 0 มีเอาต์พุทแบบ Open Drain และอินพุท/เอาต์พุทแต่ละเส้นสามารถใช้แยกอิสระจากกันสำหรับใช้เป็นอินพุทหรือเอาต์พุท เมื่อพอร์ตถูกกำหนดการใช้งานเป็นอินพุทค่าในบิตเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอสต์ของพอร์ตต้องมีค่าเป็น 1 ซึ่งจะ turn off เอาท์พุทไดร์เวอร์ FET (ลอจิก 0 ที่ขาเกตของ FET จะปิดช่องทางนำกระแสของ Field Effect Transistor เพราะเป็น Enhancement Mode n-Channel MOSFET ส่วนลอจิก 1 ที่ขาเกตจะเป็นตัวเปิดช่องทางนำกระแสและทำให้ FET มีคุณสมบัติเป็นตัวต้านทาน) ดังนั้นขาของพอร์ต 1, 2 และ 3 จะถูกยกระดับสัญญาณเป็น 1 โดยวงจรพูลอัพภายใน แต่สามารถถูกดึงระดับสัญญาณลงต่ำเป็น 0 ได้จากวงจรภายนอก



รูปที่ 2.11 แสดงภาพ โครงสร้างแต่ละบิตของพอร์ตใน MCS-51

ในกรณีของพอร์ต 0 ใช้ทรานซิสเตอร์ชนิด FET เป็นตัวยกระดับแรงดันแทน (pull up FET) ซึ่งจะถูกใช้งานเป็นเฉพาะเมื่อพอร์ต 0 ส่งค่า 1 ออกมาภายนอกเฉพาะระหว่างช่วงการติดต่อถูกใช้งานเป็นเอาท์พุทพอร์ตจะมีเอาท์พุทชนิด Open drain เพราะฉะนั้น การเขียนค่า 1 ไปยังแต่ละบิต แอสต์จะทำให้ FET ที่ทำหน้าที่เป็นเอาท์พุทถูกเปิดช่องทางนำกระแส ทำให้ขาของพอร์ตมีสถานะ high impedance เพื่อใช้พอร์ตเป็นอินพุทในขณะที่ทำหน้าที่เป็นสายรับส่งข้อมูล (Data Bus)

2.2 มอเตอร์กระแสตรง (Direct Current Motors)

2.2.1 หลักการทำงานของ MOTOR (MOTOR'S PRINCIPLE)

มอเตอร์ก็คือตัวเปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล เพื่อนำพลังงานกลที่ได้ไปขับเคลื่อนสิ่งต่าง ๆ ตามที่ต้องการ โดยอาศัยหลักการที่ว่าเมื่อมีกระแสไหลในตัวนำ ซึ่งอยู่ในสนามแม่เหล็กนั้น ย่อมทำให้เกิดแรงขึ้นในทิศทางที่หาได้จากกฎมือซ้ายของเฟลมมิ่ง ขนาดของแรงที่เกิดขึ้นหาได้

เอกสารจากสมาคมการสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$F = Bil \quad \dots\dots(2.1)$$

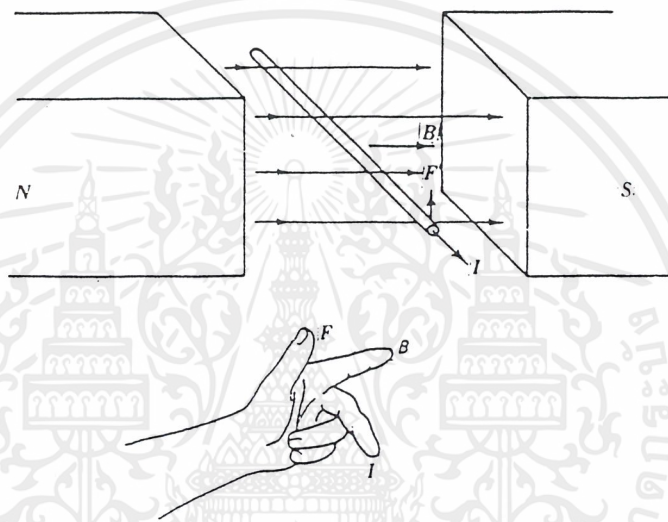
เมื่อ F = แรงที่เกิดขึ้นบนตัวนำหนึ่งตัว (นิวตัน)

B = ความหนาแน่นสนามแม่เหล็ก (เวเบอร์/เมตร²)

i = กระแสที่ไหลในตัวนำ (แอมป์)

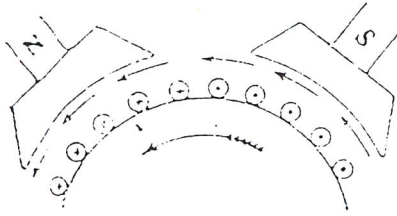
L = ความยาวของตัวนำ

แรง F ที่เกิดขึ้น จะอยู่ในแนวที่ตั้งฉากกับสนามแม่เหล็ก และกระแสที่ไหลผ่านในตัวนำนั้น



รูปที่ 2.12 แสดงทิศทางของแรงที่เกิดขึ้นในตัวนำได้โดยใช้กฎมือซ้าย (left-hand-law) ของเฟลมมิ่ง มอเตอร์กระแสไฟตรงที่มีหลายขั้วแม่เหล็กเมื่อใส่ไฟเข้าไปที่สนามกระตุ้น ข้อมทำให้เกิดอำนาจแม่เหล็กขึ้นที่ขั้วแม่เหล็ก และเมื่อป้อนกระแสให้ไหลผ่านในตัวนำที่อาร์เมเจอร์ ก็จะทำให้เกิดแรงขึ้นในตัวนำที่อยู่ในสนามแม่เหล็กนั่นคือ ทำให้อาร์เมเจอร์หมุน ในที่นี้จากรูปที่ 2.14 ได้กำหนดให้กระแสที่ไหลในตัวนำของอาร์เมเจอร์ที่อยู่ภายใต้ขั้วได้นั้น ให้กระแสที่ไหลในตัวนำของอาร์เมเจอร์ที่อยู่ภายใต้ขั้วเหนือ N มีทิศของกระแสพุ่งเข้าไปข้างในดังหางลูกศรที่กากบาท ส่วนตัวนำที่อยู่ภายใต้ขั้วใต้ S นั้น ให้กระแสพุ่งออกมาข้างนอกดังหัวลูกศรที่เป็นจุด เมื่อเป็นเช่นนี้จึงหาทิศทางของการเคลื่อนที่ของตัวนำทุก ๆ ตัว ที่อยู่ภายใต้ขั้วแม่เหล็กทั้ง N และ S ได้ โดยใช้กฎมือซ้ายของเฟลมมิ่ง จะพบว่าแรงที่เกิดขึ้นบนตัวนำทุก ๆ ตัว ภายใต้ขั้วแม่เหล็กเดียวกัน จะมีทิศไปในทางเดียวกัน และจะอยู่ในแนวสัมผัสกับเส้นรอบวงของอาร์เมเจอร์ นั่นคือ ภายใต้ขั้วแม่เหล็กแต่ละขั้วก็จะเกิดแรงลัพธ์ขึ้นแรงหนึ่ง ดังนั้นจึงให้มอเตอร์หมุนไปได้จากแรงที่เกิดขึ้นนี้

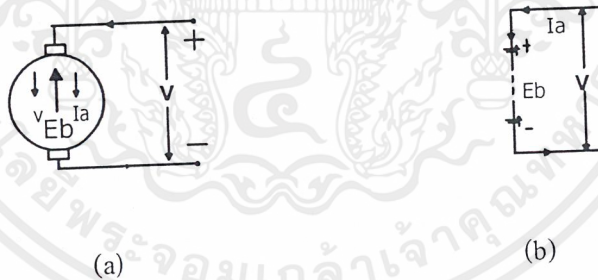
สำหรับ Commutator เป็นตัวที่ทำหน้าที่ให้กระแสไหลผ่านตัวนำไปในทิศทางเดียวตลอดเวลารายใต้ขั้วแม่เหล็กแต่ละขั้ว ซึ่งช่วยทำให้เกิดแรงบิดไปในทิศทางเดียวกันอย่างต่อเนื่อง



รูปที่ 2.13 หลักการของมอเตอร์

2.2.2 ความสำคัญของแรงเคลื่อนไฟสวน (SIGNIFICANCE OF THE BACK E.M.F.)

เมื่ออาร์เมเจอร์เริ่มหมุนจะทำให้ตัวนำที่อยู่ในอาร์เมเจอร์ตัดเส้นแรงแม่เหล็กที่มาจากสนามแม่เหล็กหลัก หรือสนามแม่เหล็กพัทธ์ที่เกิดอยู่ในมอเตอร์นั้น นั่นคือ เกิดการเหนี่ยวนำแรงเคลื่อนไฟขึ้นบนตัวนำภายในอาร์เมเจอร์ ซึ่งเป็นไปตามกฎของการเหนี่ยวนำแรงเคลื่อนแม่เหล็กไฟฟ้า แรงเคลื่อนที่เกิดขึ้นนี้ หาทิศทางของแรงเคลื่อนไฟฟ้าได้ โดยใช้กฎมือขวาของเฟลมมิ่งและพบว่า มีทิศสวนกับแรงเคลื่อนไฟที่ใส่เข้าไปให้กับมอเตอร์ ดังนั้น จึงเรียกรแรงเคลื่อนที่เกิดขึ้นนี้ว่า แรงเคลื่อนสวน (back e.m.f. = E_b) ดังรูปที่ 2.14 (a)



รูปที่ 2.14(a) วงจรของมอเตอร์ไฟตรง

(b) วงจรสมมูลย์ของมอเตอร์ที่เขียนแทนแรงเคลื่อนไฟสวนได้ด้วยแบตเตอรี่ E_b จากการที่เกิด E_b ขึ้นนี้ จึงเขียนวงจรสมมูลย์ของมอเตอร์ได้ดังรูปที่ 2.14(b) ซึ่งเหมือนกับว่ามีแบตเตอรี่ E_b ต่อคร่อมอยู่กับแรงเคลื่อนไฟ V ที่จ่ายให้กับมอเตอร์ นั่นคือในการที่จะให้ I_a ไหลได้ในอาร์เมเจอร์นั้น จะต้องมีการจ่ายไฟจำนวนหนึ่งที่ต้องเอาชนะการสวนนี้ กำลังไฟที่เอาชนะกำลังที่สวนนี้ได้คือ $E_b I_a$

ในมอเตอร์นั้น กำลังที่เอาชนะการสวนนี้ก็คือ กำลังที่เปลี่ยนไปเป็นกำลังกลนั่นเอง

นั่นคือ
$$\text{กระแส } I_a \text{ จะมีค่า} = \frac{V - E_b}{R_a}$$

โดยที่ $E_b = K_e \Phi S$ (2.2)
 เมื่อ K_e = ค่าคงที่ทางไฟฟ้า
 S = ความเร็วเป็น r.p.m.
 Φ = เส้นแรงแม่เหล็กต่อขั้ว (Wb)

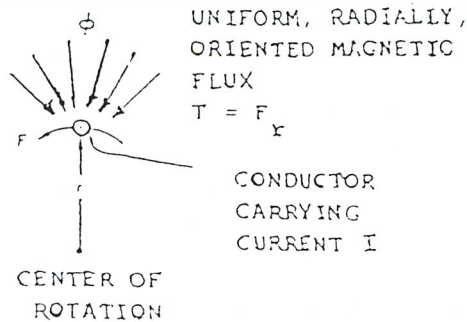
จากสมการที่ 2.2 E_b จะขึ้นอยู่กับค่า Φ และ S ฉะนั้นเพื่อให้ Φ คงที่จึงพบว่า E_b เป็นสัดส่วนโดยตรงกับความเร็ว S คือเมื่อ S มีค่ามาก จะทำให้ E_b มีค่ามาก และทำให้ I_a มีค่าน้อย (เมื่อ V คงที่ นั่นคือผลต่างของ V กับ E_b มีค่าน้อย เมื่อ R_a คงที่) จึงทำให้แรงบิด (torque) ที่มอเตอร์จ่ายออกมามีค่าน้อย เช่นเดียวกัน เมื่อ S มีค่าน้อย ทำให้ I_a มีค่ามาก ผลที่ได้คือ เกิดแรงบิดขึ้นมาจากมอเตอร์ ดังนั้นพบว่า E_b เป็นตัวทำให้มอเตอร์มีการควบคุมอยู่ตัวของมันเอง โดยการดึงกระแสเท่าที่จำเป็นต้องใช้ เพื่อให้มอเตอร์หมุนได้ในภาวะที่ต้องการ เพื่อให้เกิดการสมดุลย์ตามกฎการทรงมวล

2.2.3 แรงบิด (Torque)

มอเตอร์กระแสตรงเป็นเครื่องจักรกลไฟฟ้าที่ทำหน้าแปลงพลังงานไฟฟ้าเป็นแรงบิดทางกล โดยแรงบิดของแกนมอเตอร์จะขึ้นอยู่กับค่ากระแสในขดลวดอาเมเจอร์และเส้นแรงแม่เหล็กที่ตัดผ่านขดลวดอาเมเจอร์ดังแสดงในสมการที่ 2.3 โดยทิศทางของกระแส แรงบิด และสนามแม่เหล็กแสดงดังรูปที่ 2.15

$$T = K\Phi I \quad \dots\dots\dots(2.3)$$

เมื่อ T คือ แรงบิดที่แกนของมอเตอร์มีหน่วยเป็นนิวตันเมตร (N-m)
 K คือ ค่าคงที่ทางกายภาพของมอเตอร์ (Sec^2/m)
 Φ คือ เส้นแรงแม่เหล็กที่ตัดผ่านขดลวดอาเมเจอร์มีหน่วยเป็นเวเบอร์ (Weber)
 I คือ กระแสในขดลวดอาเมเจอร์มีหน่วยเป็นแอมแปร์ (Ampere)



รูปที่ 2.15 แสดงการเกิดแรงบิดในมอเตอร์กระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 การแยกประเภทของมอเตอร์กระแสตรง

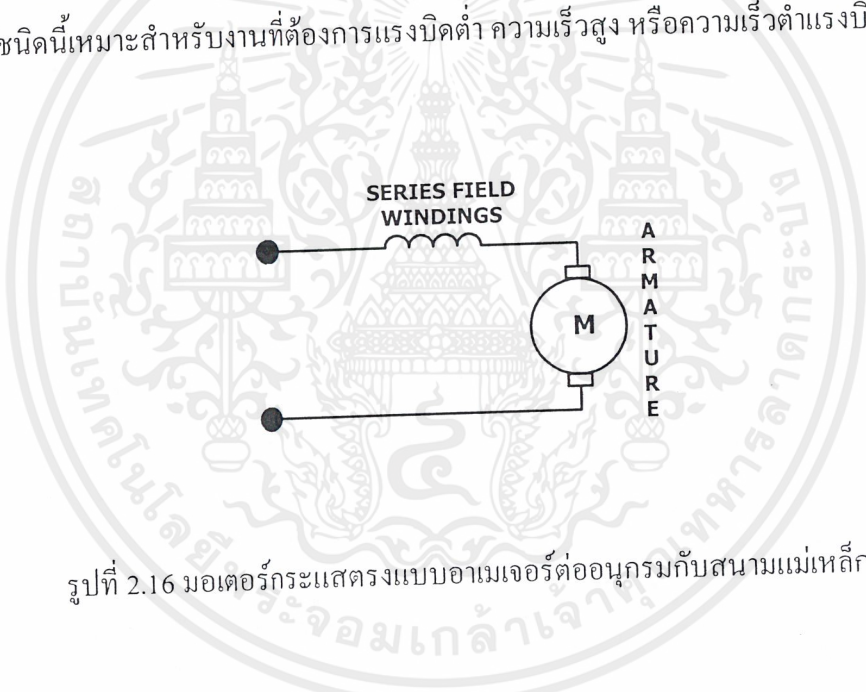
มอเตอร์กระแสตรงสามารถแบ่งได้หลายประเภทตามลักษณะการสร้างสนามแม่เหล็กและโครงสร้างของอาเมเจอร์ โดยเมื่อแบ่งตามลักษณะการสร้างสนามแม่เหล็กสามารถแบ่งได้สองประเภทคือ

1) มอเตอร์กระแสตรงแบบปรับเส้นแรงแม่เหล็กได้

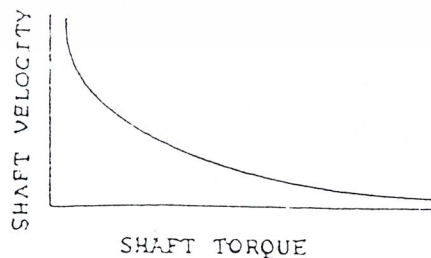
มอเตอร์กระแสตรงแบบปรับเส้นแรงแม่เหล็กได้สามารถแบ่งได้เป็น 2 ประเภท คือ

- แบบขดลวดสนามแม่เหล็กต่ออนุกรมกับขดลวดอาเมเจอร์

มีโครงสร้างดังรูปที่ 2.16 มอเตอร์แบบนี้มีเส้นแรงแม่เหล็กเป็นสัดส่วนกับกระแส จึงสามารถปรับค่าเส้นแรงแม่เหล็กได้ โดยมีความสัมพันธ์ในลักษณะที่ไม่เป็นเชิงเส้นดังรูปที่ 2.17 มอเตอร์ชนิดนี้เหมาะสำหรับงานที่ต้องการแรงบิดต่ำ ความเร็วสูง หรือความเร็วต่ำแรงบิดสูง



รูปที่ 2.16 มอเตอร์กระแสตรงแบบอาเมเจอร์ต่ออนุกรมกับสนามแม่เหล็ก



รูปที่ 2.17 ความสัมพันธ์ระหว่างความเร็วและแรงบิดของมอเตอร์อนุกรมที่แรงดันคงที่

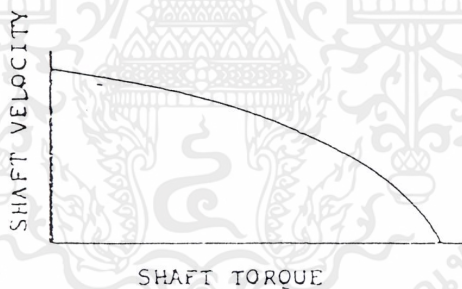
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มอเตอร์กระแสตรงแบบสนามแม่เหล็กแยกกระตุ้น

หรือเรียกว่าชันทมอเตอร์ (shunt motor) มอเตอร์แบบนี้สามารถปรับเส้นแรงแม่เหล็กได้ โดยอิสระไม่ขึ้นกับค่ากระแสในอาเมเจอร์ มักจะใช้งานที่ต้องการแรงบิดสูง โครงสร้างของมอเตอร์ชนิดนี้แสดงกังรูปที่ 2.18 ความสัมพันธ์ระหว่างแรงบิดกับความเร็วของชันทมอเตอร์ภายใต้สนามแม่เหล็กและแรงดันคงที่แสดงดังรูปที่ 2.19



รูปที่ 2.18 โครงสร้างของมอเตอร์กระแสตรงแบบสนามแม่เหล็กแยกกระตุ้น



รูปที่ 2.19 ความสัมพันธ์ระหว่างแรงบิดกับความเร็วของชันทมอเตอร์ ภายใต้สนามแม่เหล็กและแรงดันคงที่

2) มอเตอร์กระแสตรงแบบสนามแม่เหล็กคงที่

ระบบการกระตุ้นฟิลด์ของสนามแม่เหล็กโดยทั่วไป มักใช้แม่เหล็กถาวรดังรูปที่ 1.20

ในมอเตอร์กระแสตรงแบบนี้สนามแม่เหล็กจะมีค่าคงที่ ดังนั้นสามารถเขียนความสัมพันธ์ของแรงบิดและแรงดันย้อนกลับ ได้ดังนี้

$$T = K_t I \quad \dots\dots(2.4)$$

$$E = K_e \omega \quad \dots\dots(2.5)$$

โดยสมการทางไฟฟ้าของมอเตอร์กระแสตรงแบบสนามแม่เหล็กคงที่เป็นดังนี้

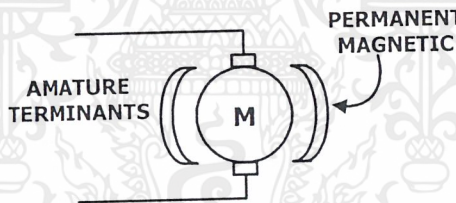
$$V = K_e \omega + L di/dt + iR \quad \dots\dots(2.6)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อ V คือ แรงดันที่ป้อนให้มอเตอร์มีหน่วยเป็น โวลต์ (volt)
 K_c คือ ค่าคงที่แรงดันย้อนกลับ
 L คือ ค่าความเหนี่ยวนำของมอเตอร์มีหน่วยเป็น เฮนรี่ (Henry)
 R คือ ความต้านทานของขั้วมอเตอร์มีหน่วยเป็น โอห์ม (Ohm)

$$T_g = Jdw/dt + BW + T_f + T_l \quad \dots\dots(2.7)$$

- เมื่อ T_g คือ แรงบิดที่เกิดจากอาร์เมเจอร์มีหน่วยเป็นนิวตัน-เมตร (N-m)
 J คือ ผลรวมของ โมเมนต์ความเฉื่อยของมอเตอร์และ โหลด
 W คือ ความเร็วการหมุนของมอเตอร์มีหน่วยเป็นเรเดียนต่อวินาที (Rad/sec)
 B คือ สัมประสิทธิ์ของวิสกอสแดมป์ปิ้ง
 T_f คือ แรงบิดเสียดทานมีหน่วยเป็นนิวตัน-เมตร (N-m)
 T_l คือ แรงบิดที่โหลดมีหน่วยเป็นนิวตัน-เมตร (N-m)



รูปที่ 2.20 มอเตอร์กระแสตรงแบบฟิวด์แม่เหล็กถาวร

2.3 ความรู้เบื้องต้นเกี่ยวกับพอร์ตขนาน

การประมวลผลเพื่องานควบคุมนั้น สิ่งแรกจะต้องมีส่วนของสัญญาณอินพุท ซึ่งอาจจะมาจากตัวตรวจจับต่างๆผ่านวงจรภาคหน้าเพื่อเปลี่ยนรูปแบบสัญญาณอินพุทให้เหมาะสมกับการเชื่อมต่อกับคอมพิวเตอร์ เมื่อข้อมูลอินพุทถูกส่งเข้าสู่คอมพิวเตอร์แล้ว คอมพิวเตอร์จะทำการประมวลผลข้อมูลที่ได้มาเหล่านั้นให้อยู่ในรูปแบบที่เหมาะสมก่อนที่จะส่งออกไปยังภายนอกผ่านอุปกรณ์เอาต์พุท ซึ่งอาจจะเป็นการส่งออกไปยังจอภาพ หรือส่งออกไปยังจุดเชื่อมต่ออื่นๆเพื่อควบคุมอุปกรณ์เอาต์พุตต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อระหว่างคอมพิวเตอร์กับอุปกรณ์ภายนอกทั้งส่วนของภาคินพุทและภาคเอาต์พุท สามารถทำได้หลายวิธีดังนี้

-เชื่อมต่อผ่านทางการ์ดอินพุทเอาต์พุท ซึ่งใช้วิธีเสียบการ์ดหรือติดตั้งการ์ดลงในสล๊อตภายในเครื่องคอมพิวเตอร์

-เชื่อมต่อผ่านทางพอร์ทอนุกรม

-เชื่อมต่อผ่านทางพอร์ทขนาน

-เชื่อมต่อผ่านระบบมาตรฐานอื่นๆเช่น พอร์ทUSB(Universal Serial Bus),พอร์ทSCSI หรือพอร์ทGAME เป็นต้น

2.3.1 ทำไมถึงเลือกใช้งานพอร์ทขนาน

เมื่อเทียบกับการใช้งานการ์ดอินพุทเอาต์พุทที่ติดตั้งตั้งอยู่ภายในเครื่องคอมพิวเตอร์แล้ว พอร์ทขนานมีข้อได้เปรียบอยู่หลายประการดังนี้

(1) ในด้านความปลอดภัย การที่ต้องถอดฝาเครื่องคอมพิวเตอร์ออกมาเพื่อเสียบการ์ดเชื่อมต่อลงในสล๊อตของคอมพิวเตอร์ อาจจะทำให้เกิดความเสียหายกับส่วนอื่นๆ ของคอมพิวเตอร์ได้ ถ้าผู้ใช้งาน ไม่มีความชำนาญหรือเกิดการต่อวงจรที่ผิดพลาด

(2) ในด้านการเข้ากันได้กับคอมพิวเตอร์ส่วนใหญ่ การเชื่อมต่อโดยใช้การ์ดที่เสียบลงในสล๊อต ไม่สามารถใช้กับคอมพิวเตอร์ในปัจจุบันได้ทุกรุ่น ยกตัวอย่าง คอมพิวเตอร์โน้ตบุ๊ก จะไม่มีสล๊อตเสียบ แต่จะมีที่เสียบการ์ด PCMCIAแทน ในขณะที่พอร์ทขนานจะมีติดตั้งอยู่ในคอมพิวเตอร์ทุกเครื่อง ทั้งนี้เพื่อใช้ในการติดต่อกับเครื่องพิมพ์

(3) ข้อจำกัดด้านพื้นที่ คอมพิวเตอร์บางเครื่องมีการเสียบการ์ดเชื่อมต่อตัวอื่นๆอยู่แล้ว อาทิการ์ดเสียง การ์ดโมเด็ม เป็นต้น จนไม่มีสล๊อตเหลือพอสำหรับการเสียบการ์ดเชื่อมต่อเพิ่มเติม

(4) ความสะดวกในการใช้งาน การเชื่อมต่อทางพอร์ทขนานสามารถทำได้ง่ายๆ เพียงต่อสายสำหรับเชื่อมต่อเข้ากับคอนเน็คเตอร์ DB-25 ของพอร์ทขนาน

(5) จำนวนช่องสัญญาณอินพุท/เอาต์พุท พอร์ทขนานมีจำนวนพอร์ทอินพุท/เอาต์พุทมากเพียงพอที่จะนำไปใช้งานต่างๆ และยังสามารถขยายให้มีจำนวนพอร์ทเพิ่มขึ้นได้ โดยพอร์ทขนานปกติมีจำนวนขาเอาต์พุท 12 ขา และอินพุท 5 ขา

(6) ความเร็วในการสื่อสารข้อมูลกับพอร์ทขนาน มีความเร็วเท่ากับการติดต่อกับระบบบัสโดยตรง และมีความเร็วมากกว่าการติดต่อผ่านทางพอร์ทอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(7) อะไหล่และชิ้นส่วนประกอบ คอนเน็กเตอร์และสายเชื่อมต่อต่างๆของการเชื่อมต่อผ่านทางพอร์ตนานหาได้ง่ายและราคาไม่แพง สร้างขึ้นเองก็สามารถทำได้อย่างง่ายดาย จากคุณสมบัติดังที่ได้กล่าวทำให้พอร์ตนานเหมาะที่จะนำมาใช้ในการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกเพื่อควบคุมหรือรับสัญญาณข้อมูล

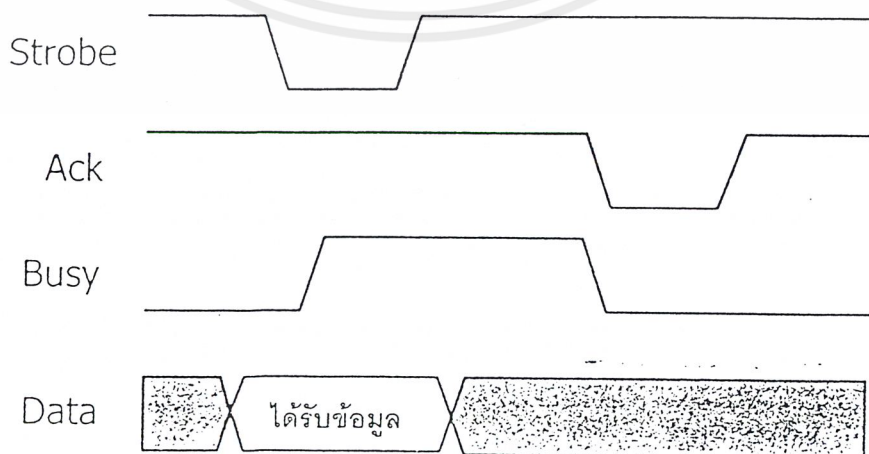
2.3.2 ความรู้เบื้องต้นเกี่ยวกับพอร์ตนาน

พอร์ตนาน (Parallel port) สาเหตุที่มีชื่อนี้ เนื่องจากการถ่ายถอดข้อมูลของพอร์ตนานนี้เป็นแบบขนาน สำหรับชื่อเรียกอีกชื่อหนึ่งของพอร์ตนานคือ พอร์ทเครื่องพิมพ์ (Printer port) เนื่องจากพอร์ตนานนี้ใช้สำหรับต่อเครื่องพิมพ์นั่นเอง

ด้วยการถ่ายถอดข้อมูลแบบขนานนี้เอง ทำให้พอร์ตนานมีอัตราการถ่ายถอดข้อมูลสูงกว่าการถ่ายถอดข้อมูลแบบอนุกรมประมาณ 8-10 เท่า และการประมวลผลข้อมูลส่วนใหญ่จะมีขนาด 8 บิต ดังนั้นพอร์ตนานจึงสามารถรองรับการถ่ายถอดข้อมูล 8 บิตได้โดยไม่ต้องต่อส่วนเพิ่มเติม

2.3.3 ลักษณะทางกายภาพของพอร์ตนาน

เพื่อให้เข้าใจถึงการนำเอาพอร์ตนานไปใช้งาน ก่อนอื่นต้องมาทำความเข้าใจก่อนว่า ปกตินั้น การส่งเครื่องพิมพ์งานจากคอมพิวเตอร์ไปยังพอร์ตนานนั้นมีรูปแบบการทำงานภายในอย่างไร



จากรูปที่ 2.21 จะเห็นได้ว่ามีสัญญาณที่ใช้งานจริงๆมีไม่มาก เริ่มจากสัญญาณพอร์ท Data ถูกส่งออกไปยังเครื่องพิมพ์ พร้อมทั้งส่งสัญญาณ Strobe ไปด้วย เพื่อให้เครื่องพิมพ์รับรู้ว่าการส่งข้อมูลใหม่มาที่ขา Data แล้ว จากนั้น คอมพิวเตอร์จะต้องรอการตอบกลับจากเครื่องพิมพ์ นั่นคือเครื่องพิมพ์จะสร้างสัญญาณ Busy เพื่อบอกว่าเครื่องพิมพ์ยังไม่พร้อมที่จะรับข้อมูลใหม่ จนกระทั่งเครื่องพิมพ์พร้อมทำงาน เครื่องพิมพ์จะสร้างสัญญาณ ACK ส่งไปยังคอมพิวเตอร์เพื่อแจ้งว่าพร้อมที่จะรับข้อมูลใหม่แล้ว

สัญญาณข้อมูลขนาด 8 บิต, สัญญาณ Strobe และสัญญาณ ACK (acknowledge) เป็นสัญญาณที่สำคัญในการส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์ นอกจากนี้ส่วนใหญ่การติดต่อกับเครื่องพิมพ์ยังต้องมีสัญญาณอื่นๆร่วมด้วย เนื่องจากเครื่องพิมพ์ต้องทำหน้าที่ถึง 3 อย่างคือ รับข้อมูลจากคอมพิวเตอร์, พิมพ์ข้อมูลที่รับเข้ามา และตอบสนองต่อการใช้งานของผู้ใช้ เช่น การเปลี่ยนฟอนต์ เป็นต้น บางครั้งอาจเกิดเหตุการณ์ไม่ปกติ เช่น บัฟเฟอร์สำหรับข้อมูลเต็ม(เนื่องจากเครื่องพิมพ์เป็นอุปกรณ์ที่ทำงานทางกลย่อมทำงานได้ช้ากว่าการส่งข้อมูลของคอมพิวเตอร์) เครื่องพิมพ์ต้องแจ้งไปยังคอมพิวเตอร์ว่าให้หยุดส่งข้อมูลชั่วคราว เนื่องจากรับข้อมูลมากกว่านี้ไม่ได้แล้ว สัญญาณที่ส่งไปยังคอมพิวเตอร์คือสัญญาณ Busy และเมื่อเครื่องพิมพ์เกิดข้อผิดพลาด เช่น กระดาษติด จะต้องแจ้งไปยังคอมพิวเตอร์เช่นกัน โดยส่งสัญญาณที่เรียกว่าสัญญาณ Error นอกจากนี้เมื่อคอมพิวเตอร์ต้องการรีเซ็ตเครื่องพิมพ์ คอมพิวเตอร์ต้องส่งสัญญาณ Reset ไปยังเครื่องพิมพ์ จึงสามารถสรุปหาสัญญาณที่จะเป็นสำหรับการติดต่อดังในตารางที่ 2.5

จากตารางที่ 2.5 จะเห็นได้ว่าพอร์ทขนานของคอมพิวเตอร์ยังแยกย่อยออกเป็นอีก 3 พอร์ทได้แก่ พอร์ทเอาต์พุตที่ทำหน้าที่ส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์ พอร์ทเอาต์พุตสำหรับสัญญาณ Strobe และ Reset พอร์ทอินพุตสำหรับการอ่านค่าสัญญาณ Acknowledge, Busy และสัญญาณ Error จากเครื่องพิมพ์

ตารางที่ 2.5 สัญญาณสำคัญของพอร์ทขนานที่ใช้ติดต่อเครื่องพิมพ์

สัญญาณ	หน้าที่การทำงาน	ทิศทาง
ข้อมูล 8 บิต	ข้อมูลที่ส่งจากคอมพิวเตอร์ไปยังเครื่องพิมพ์	คอมพิวเตอร์
Strobe	แจ้งเครื่องพิมพ์ถึงข้อมูลที่ส่งมาใหม่	คอมพิวเตอร์
Acknowledge	เครื่องพิมพ์แจ้งมายังคอมพิวเตอร์ว่าได้รับข้อมูลแล้ว	เครื่องพิมพ์
Busy	แจ้งสถานะว่าเครื่องพิมพ์ไม่ว่างที่จะรับข้อมูลใหม่	เครื่องพิมพ์
Error	แจ้งสถานะว่าเครื่องพิมพ์เกิดข้อผิดพลาด	เครื่องพิมพ์
Reset	รีเซ็ตเครื่องพิมพ์	คอมพิวเตอร์

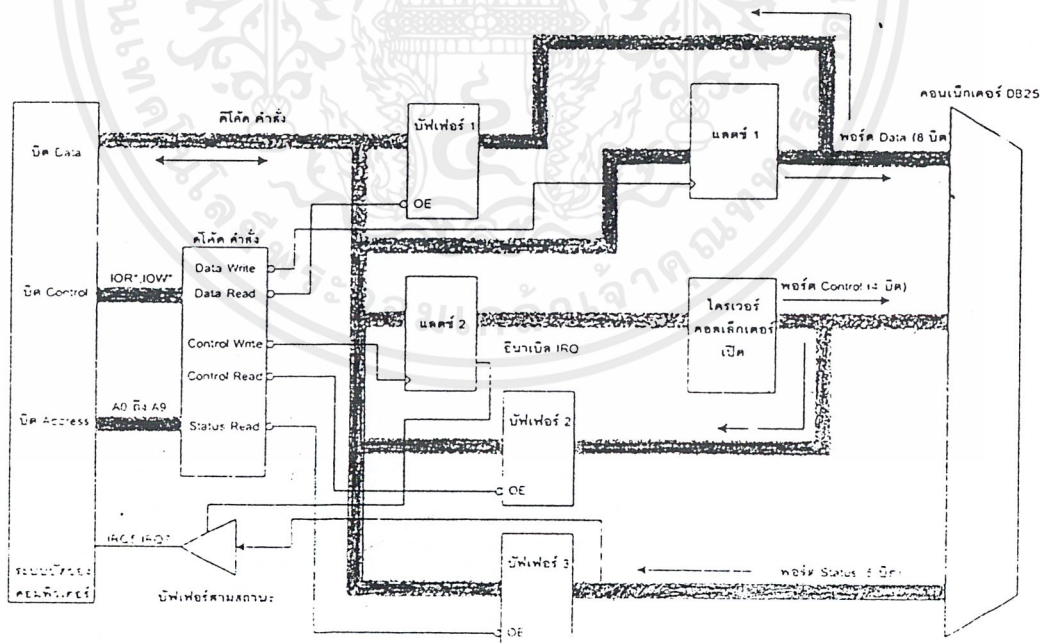
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปตีพิมพ์หรือเผยแพร่

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยปกติพอร์ทขนานออกแบบมาให้มีสายสัญญาณอยู่ทั้งหมด 17 เส้น สายสัญญาณเหล่านี้จะมีรีจิสเตอร์ 3 ตัวควบคุมการทำงานดังนี้

- 1) พอร์ทเอาต์พุตสำหรับสัญญาณข้อมูล 8 เส้น มีรีจิสเตอร์ Data ควบคุม
- 2) พอร์ทอินพุตสำหรับการอ่านค่าสถานะต่างๆ จากภายนอกมีอยู่ด้วยกัน 5 เส้น ใช้รีจิสเตอร์ Status ในการควบคุม
- 3) พอร์ทเอาต์พุตสำหรับส่งสัญญาณควบคุมไปยังอุปกรณ์ภายนอก มีอยู่ด้วยกัน 4 เส้น ใช้รีจิสเตอร์ Control ในการควบคุม

บล็อกไดอะแกรมในรูปที่ 2.22 แสดงระบบบัสของคอมพิวเตอร์สำหรับการติดต่อกับพอร์ทขนาน สัญญาณเอาต์พุตจากพอร์ทขนานจะถูกส่งไปยังคอนเน็คเตอร์แบบ DB-25 สำหรับคอมพิวเตอร์ส่วนใหญ่ในปัจจุบันพอร์ทขนานจะมีพร้อมกับเมนบอร์ด ไม่จำเป็นต้องใช้การ์ดเสียบเพิ่มเติมเหมือนในอดีต พร้อมทั้งมีฟังก์ชันการทำงานที่ซับซ้อนขึ้น แต่ยังคงสนับสนุนการทำงานของพอร์ทขนานในรูปแบบมาตรฐาน (SPP) อยู่



รูปที่ 2.22 แสดงระบบบัสภายในของพอร์ทขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อรูปที่ 2.21 เทียบการทำงานโดยทั่วไปกับการเชื่อมต่อผ่านการ์ดที่เสียบลงในสล็อตของคอมพิวเตอร์แล้ว พอร์ตขนานจะมีลักษณะใกล้เคียงกัน โดยการติดต่อกับพอร์ตขนานจะต้องมีการอ้างแอดเดรส ตำแหน่งแอดเดรสที่ใช้อ้างอิงจะเป็นตำแหน่ง A0-A9 และใช้ขา IOR และ IOW สำหรับเป็นตัวเลือกว่าต้องรอ่านหรือเขียนรีจิสเตอร์ตัวใด จากการตีโค้ดแอดเดรส A0-A9 นี้เองทำให้ได้สัญญาณออกมาเพื่อไปควบคุมหรืออีนาเบิลวงจรบัฟเฟอร์ต่างๆดังนี้

Data Write สัญญาณอีนาเบิลสำหรับนำข้อมูลที่อยู่ในบิต Data ไปออกที่ขา Data ของพอร์ตขนาน

Data Read สัญญาณอีนาเบิลสำหรับอ่านข้อมูลจากขา Data ของพอร์ตขนานไปเก็บไว้ในบิต Data

Control Write สัญญาณอีนาเบิลสำหรับนำข้อมูลที่อยู่ในบิต Data ไปออกที่ขา Control ของพอร์ตขนาน สำหรับพอร์ตนี้นอกจากจะส่งข้อมูลออกไปยังพอร์ตขนานแล้ว ยังทำหน้าที่อีนาเบิลการอินเตอร์รัปต์ของการเปลี่ยนแปลงสัญญาณที่พอร์ต Status อีกด้วย

Control Read สัญญาณอีนาเบิลสำหรับอ่านค่าข้อมูลจากขา Control มาเก็บไว้ในบิต Data

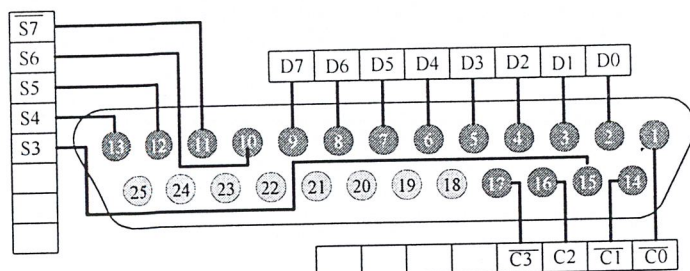
Status Read สัญญาณอีนาเบิลสำหรับอ่านค่าข้อมูลจากพอร์ต Status มาเก็บไว้ในบิต Data ตารางที่ 2.6 แสดงแอดเดรสของพอร์ตขนาน ซึ่งกำหนดไว้ 3 ตำแหน่ง คือ LTP1, LTP2 และ LTP3 ส่วนในตารางที่ 2.7 แสดงชื่อและหน้าที่การทำงานของตำแหน่งขาต่างๆ บนพอร์ตขนาน

ตารางที่ 2.6 แสดงแอดเดรสของพอร์ตขนาน

ชื่อพอร์ต	LTP1:		LTP2:		LTP3:	
	ฐานสิบ	ฐานสิบหก	ฐานสิบ	ฐานสิบหก	ฐานสิบ	ฐานสิบหก
DATA	888	378H	956	3BCH	632	278H
STATUS	889	379H	957	3BDH	633	279H
CONTROL	890	37AH	958	3BEH	634	27AH

ตารางที่ 2.7 แสดงสัญญาณทั้งหมดที่อยู่บนพอร์ทขนาน

DB-25	รีจิสเตอร์	ทิศทาง	ตำแหน่งบิต	ชื่อขาสัญญาณ	หน้าที่การทำงาน
1	Control	Out	\overline{CD}	\overline{STROBE}	แอกทีฟ "0" ส่งค่าออกไปเพื่อบอกเวลาที่ขาคาดำมีข้อมูลแล้ว
2-9	Data	Out	D1-D8	DATA1-DATA8	รับและส่งข้อมูล
10	Status	In	S6	NACK	เป็นพัลส์ลอจิก "0" ที่ส่งมาจากเครื่องพิมพ์บอกว่าเครื่องพิมพ์พร้อมแล้ว
11	Status	In	$\overline{s7}$	\overline{BUSY}	เป็นสัญญาณแจ้งมาจากเครื่องพิมพ์ว่ายังไม่พร้อมรับข้อมูล
12	Status	In	S5	PE	แจ้งกระดาษหมด
13	Status	In	S4	SELECT	แจ้งว่าเครื่องพิมพ์ต่ออยู่
14	Control	Out	$\overline{c1}$	$\overline{AUTOFEED}$	สั่งเครื่องพิมพ์ให้เลื่อนบรรทัด
15	Status	In	S3	\overline{ERROR}	แสดงข้อผิดพลาดจากเครื่องพิมพ์
16	Control	Out	C2	\overline{INT}	รีเซ็ตเครื่องพิมพ์โดย "0"
DB-25	รีจิสเตอร์	ทิศทาง	ตำแหน่งบิต	ชื่อขาสัญญาณ	หน้าที่การทำงาน
17	Control	Out	$\overline{c3}$	$\overline{SELECT-IN}$	สัญญาณแจ้งเครื่องพิมพ์ว่าต้องการเลือกเครื่องนี้
18-25				GND	กราวนด์



รูปที่ 2.23 แสดงตำแหน่งขากภายในพอร์ทขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 การนำพอร์ทไปใช้งาน

สำหรับพอร์ทขนานแบบมาตรฐาน ผู้ใช้งานสามารถนำพอร์ทอินพุต 5 บิต (พอร์ท Status) พอร์ทเอาต์พุต 4 บิต (พอร์ท Control) และพอร์ทเอาต์พุตอีก 8 บิต (พอร์ท Data) ไปใช้งานได้โดยตรง โดยที่ 4 บิตของพอร์ทเอาต์พุตหรือพอร์ท Control นั้นสามารถดัดแปลงให้ใช้งานเป็นพอร์ทอินพุตขนาด 4 บิต ได้ด้วยดังนั้นผู้ใช้งานจึงสามารถนำสัญญาณจากพอร์ทขนานที่มีมากถึง 17 เส้นไปใช้งานในการควบคุมโดยใช้ระดับสัญญาณ TTL

2.3.5 การเขียนโปรแกรมติดต่อกับพอร์ทขนาน

พอร์ทขนานของคอมพิวเตอร์จะมีลักษณะเช่นเดียวกับอุปกรณ์อินพุตเอาต์พุตตัวอื่นๆคือ เมื่อต้องการติดต่อก็จะต้องกำหนดแอดเดรสที่ต้องการติดต่อกับ ตารางที่ 2.6 แสดงแอดเดรสของพอร์ทขนาน โดยแบ่งออกเป็น 3 ตำแหน่งคือ แอดเดรสของรีจิสเตอร์ Data ,รีจิสเตอร์ Status และรีจิสเตอร์ Control โดยแอดเดรสจะมีอยู่ทั้งหมด 3 ชุด สำหรับพอร์ทขนาน 3 ชุด คือ LTP1,LTP2 และ LTP3

เมื่อต้องการติดต่อกับพอร์ทขนานในตำแหน่งใด ก็ให้ส่งค่าข้อมูลออกไปที่พอร์ทขนานในตำแหน่งนั้นๆ ยกตัวอย่างการเขียนโปรแกรมด้วย แอสเซมบลีเพื่อส่งค่าลอจิก "1" ออกไปทุกบิตของพอร์ท Data ของ LTP1 จะต้องเขียน โปรแกรมดังนี้

การส่งค่าข้อมูลออกไปยังพอร์ทขนาน

```
MOV DX,378H
```

```
MOV AL,FFH
```

```
OUT DX,AL
```

การอ่านค่าข้อมูลจากพอร์ทขนาน

```
MOV DX,379H
```

```
IN AL,DX
```

2.4 กระบวนการผลภาพ

2.4.1 การประมวลผลภาพเชิงตัวเลข (Digital Image Processing)

การประมวลผลภาพเชิงตัวเลข หมายถึง การนำภาพที่พบทั่วไปมาประมวลผลด้วยเครื่องคอมพิวเตอร์ โดยภาพที่นำมาประมวลผลด้วยเครื่องคอมพิวเตอร์จะถูกแทนที่ด้วยตัวเลขให้อยู่ในรูปของเมตริกซ์ แต่ภาพที่ได้โดยส่วนมากแล้วเป็นภาพที่ได้จากตัวรับสัญญาณ ซึ่งอยู่ในรูปของฟังก์ชัน $f(x,y)$ ต่อเนื่องในระนาบสองมิติ (คือแกน X และแกน Y) โดยจะเป็นสัดส่วนกับความสว่างหรือความเข้มของภาพ ที่ตำแหน่ง (x,y)

2.4.1.1 การแทนภาพด้วยข้อมูลแบบดิจิทัล

ภาพข้อมูลแบบดิจิทัล (Digital Image) เป็นภาพที่ถูกแปลงมาจากอนาลอก อยู่ในรูปของตัวเลข โดยภาพอนาลอกถูกแบ่งเป็นพื้นที่สี่เหลี่ยมเล็ก ๆ ที่เรียกว่า พิกเซล (pixel) ในแต่ละพิกเซล จะถูกระบุตำแหน่งโดย (x,y) โดยเราสามารถแปลงภาพเป็นภาพข้อมูลดิจิทัลได้ โดยมีขั้นตอนและวิธีการดังนี้

เมื่อนำสัญญาณอนาลอกที่ต้องการประมวลผลมาผ่านส่วนที่เรียกว่า ดิจิไตเซอร์ (Digitizer) ซึ่งทำหน้าที่ในการเปลี่ยนสัญญาณอนาลอกให้เป็นสัญญาณดิจิทัล อุปกรณ์ส่วนนี้ได้แก่ กล้องวิดีโอดิจิทัล จากนั้นทำการควอนไทซ์ (Quantizing) เพื่อที่จะประมวลผลสัญญาณด้วยระบบคอมพิวเตอร์ ฟังก์ชันของภาพ $f(x,y)$ จะถูกทำให้เป็นสัญญาณไม่ต่อเนื่องทั้งระนาบภาพ ซึ่งเราเรียกว่า การสุ่มภาพ (Image Sampling) ของฟังก์ชันที่ได้เรียกว่า การควอนไทเซชันระดับสีเทา (Gray Level Quantization) ก็จะได้ข้อมูลที่เป็นดิจิทัล

สมมติว่าสัญญาณภาพต่อเนื่อง $f(x,y)$ ถูกดิจิไตซ์ในระนาบ X และ Y เป็นช่วงเท่า ๆ กัน เราสามารถจัด $f(x,y)$ ให้อยู่ในรูปของเมตริกซ์ขนาด $N \times N$ ได้ดังสมการที่

$$f(x,y) = \begin{matrix} f(0,0) & f(0,1) & f(0,2) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \dots & f(1,N-1) \\ \dots & \dots & \dots & \dots & \dots \\ f(0,0) & f(0,1) & f(0,2) & \dots & f(0,N-1) \end{matrix} \dots\dots\dots(2.8)$$

ซึ่งทางขวาของสมการ จะเรียกได้ว่า ภาพดิจิทัล และทุก ๆ สมาชิกของเมตริกซ์ จะเรียกว่า พิกเซล จากขบวนการสร้างภาพดิจิทัลข้างต้น จะเห็นได้ว่าเราสามารถทราบขนาดของความละเอียดของภาพ $N \times N$ พิกเซล

2.4.1.2 ลักษณะการจัดเก็บข้อมูลภาพแบบดิจิทัล

โดยทั่วไปแล้ว ข้อมูลภาพจะมีความเข้มตั้งแต่ 2 ระดับขึ้นไป แต่ที่ใช้กันที่ค่าระดับความเข้มของจุดภาพเท่ากับ 256 ระดับ ซึ่งจะทำให้ค่าของจุดภาพอยู่ในช่วง (0-255) โดยใช้เนื้อที่การเก็บข้อมูลภาพขนาด 1 ไบต์ หรือ 8 บิต สำหรับข้อมูล 1 จุดภาพ ($2^8 = 256$) ในกรณีที่ต้องการภาพที่มีความละเอียดของระดับความเข้มสูง ๆ อาจจะต้องการจำนวนบิตสำหรับการเก็บข้อมูลมากกว่า 8 บิต ก็อาจจะเป็น 16 หรือ 24 บิต โดยค่าความเข้มของจุดภาพจะเท่ากับ 2^{16} และ 2^{24} โดยจะแยกให้เห็นชัดเจนดังนี้

1. ภาพ 2 ระดับ คือมีเพียงจุดขาวกับดำเท่านั้น โดยแต่ละจุดภาพเป็นข้อมูลขนาด 1 บิต
 2. ภาพ 16 ระดับ คือ ในแต่ละจุดภาพจะมีขนาดของข้อมูล 4 บิต ซึ่งทำให้สามารถแสดงได้ 16 ระดับสี หรือ 16 เกรย์สเกล ขึ้นอยู่กับภาพนั้นเป็นภาพสีหรือภาพขาวดำ
 3. ภาพ 256 ระดับ คือ ในแต่ละจุดภาพจะมีขนาดของข้อมูล 8 บิต ซึ่งทำให้สามารถแสดงภาพได้ 256 ระดับสี หรือ 256 เกรย์สเกล ขึ้นอยู่กับภาพนั้นเป็นภาพสีหรือภาพขาวดำ
 4. ภาพทิวทัศน์ (True color) คือในแต่ละจุดภาพมีขนาดของข้อมูล 24 บิต ทำให้สามารถแสดงภาพได้เหมือนภาพจริงที่สุด เพราะสามารถแสดงสีได้ถึง 16,777,216 สี ภาพทิวทัศน์สามารถแสดงได้เฉพาะภาพสีเท่านั้น ไม่สามารถแสดงภาพขาวดำได้
- การแสดงผลนี้จะใช้วิธีตั้งค่าของแม่สีในตารางสี โดยอาจเลือกเป็นแบบใดก็ได้ ซึ่งจะแสดงผลโดยการส่งค่าสี RGB ผ่าน D/A สีละ 8 บิต ออกไป ความแตกต่างของการแสดงผลสีและขาวดำ คือ ภาพขาวดำจะต้องตั้งให้แม่สีทั้งสามสี มีค่าเท่ากัน เนื่องจาก VGA กำหนดให้แม่สีแต่ละสีใช้ได้เพียง 64 ระดับเท่านั้น หากต้องการเห็นจริงทั้ง 256 ระดับ ต้องแสดงในโหมดทิวทัศน์แล้วให้ RGB มีค่าเท่ากัน ซึ่งในโหมดนี้จะสามารถใช้รีจิสเตอร์ได้ 8 บิต สำหรับแต่ละแม่สี

โดยทั่วไปวิธีการประมวลผลภาพเชิงตัวเลขที่ทำให้คอมพิวเตอร์สามารถรู้จักวัตถุภายในภาพได้นั้น พองจะแบ่งได้สองระดับด้วยกันคือ การประมวลผลภาพในระดับต่ำ (Low Level Image Processing) และการประมวลผลภาพในระดับสูง (High Level Image Processing)

การประมวลผลภาพในระดับต่ำจะเป็นการประมวลผลเชิงตัวเลขเกือบทั้งหมด เพื่อหาตัวแปรต่าง ๆ มาอธิบายข้อมูลภาพ โดยมีจุดประสงค์เพื่อนำตัวแปรเหล่านั้นไปใช้ในการประมวลผล

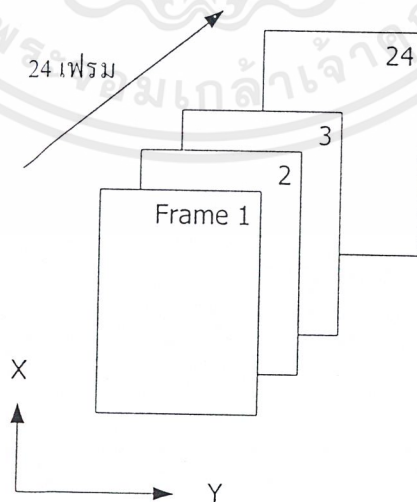
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระดับสูงต่อไป โดยทั่วไปแล้วการประมวลผลภาพระดับต่ำจะประกอบไปด้วย การประมวลผลภาพก่อน (Preprocessing) การกำจัดสัญญาณรบกวน หรือทำให้ภาพคมชัด หางอบของภาพ เป็นต้น

การประมวลผลระดับสูง เป็นการนำผลลัพธ์ หรือสัญลักษณ์ที่ได้จากการประมวลผลระดับต่ำมาตีความหรือประมวลผลเพื่อให้คอมพิวเตอร์สามารถรู้จัก และเข้าใจภาพได้ สำหรับความแตกต่างของการประมวลผลภาพระดับต่ำและการประมวลผลภาพระดับสูงนั้นคือ ข้อมูลที่นำมาใช้ในการประมวลผลภาพ โดยที่การประมวลผลภาพระดับต่ำจะใช้ค่าความสว่างของจุดโดยตรง ส่วนการประมวลผลภาพระดับสูงนั้นข้อมูลภาพที่นำมาประมวลผลจะถูกแสดงในรูปของสัญลักษณ์ ซึ่งสัญลักษณ์ เหล่านี้จะแสดงถึงสิ่งต่าง ๆ ที่อยู่ในภาพ เช่น ขนาดของวัตถุ รูปร่าง และความสัมพันธ์กันระหว่างวัตถุภายในภาพ

2.4.2 สัญญาณข้อมูลภาพจากดิจิตอลวิดีโอ

การส่งสัญญาณข้อมูลภาพจากวิดีโอ จะมีลักษณะการส่งที่เป็นลำดับภาพเดี่ยวหรือเฟรม (Frame) ที่ฉายต่อเนื่องกันดังรูปที่ 2.24 เช่น ภาพยนตร์ใช้อัตรา 24 เฟรมต่อวินาที หรือวิดีโอระบบ NTSC จะส่งด้วยความเร็ว 30 เฟรมต่อวินาที โดยดิจิตอลวิดีโอแต่ละเฟรมจะเป็นข้อมูลภาพดิจิตอลในลักษณะของเมตริกซ์ (Matrix) ซึ่งแต่ละจุดเรียกว่า พิกเซล (Pixel) มีค่าของระดับความเข้มสี โดยทั่วไปจะใช้เกรย์สเกลสากลที่มีค่าตัวเลขตั้งแต่ 0 ถึง 255 โดย 0 แทนความมืดมากที่สุด ส่วนตัวเลข 255 จะแทนความสว่างมากที่สุด



รูปที่ 2.24 แผนภาพตัวอย่างการส่งสัญญาณวิดีโออัตรา 24 เฟรมต่อวินาที

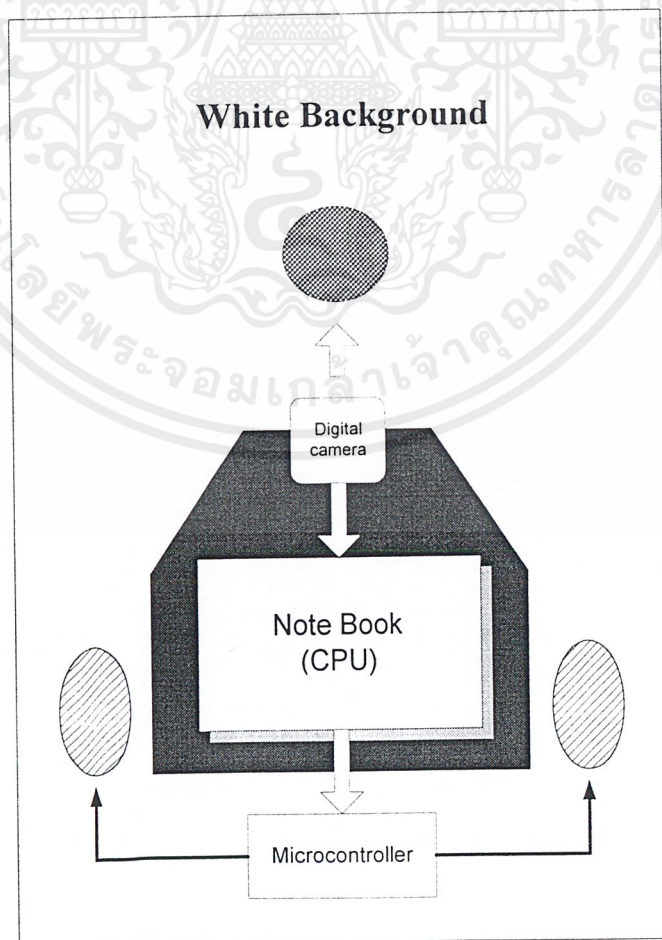
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

หลักการทำงานของระบบ

3.1 โครงสร้างของระบบการติดตามเป้าหมาย

ระบบการติดตามเป้าหมายนี้ ประกอบด้วยส่วนสำหรับการรับภาพ ส่วนของโปรแกรมการประมวลผลภาพ และส่วนการควบคุมการเคลื่อนที่ของรถในการติดตามเป้าหมาย ในส่วนของการรับภาพจะประกอบด้วยกล้องวีดีโอดิจิตอลสำหรับจับภาพของเป้าหมายที่อยู่ด้านหน้าของรถ สัญญาณจากกล้องวีดีโอดิจิตอลจะต่อผ่านการ์ดวีดีโอเพื่อแปลงสัญญาณจากกล้องให้เป็นสัญญาณดิจิตอล แล้วส่งให้กับคอมพิวเตอร์เพื่อทำการประมวลผล เมื่อเป้าหมายมีการเคลื่อนที่ คอมพิวเตอร์จะทำการประมวลผลเพื่อหาตำแหน่งที่เป้าหมายได้เคลื่อนที่ไป แล้วส่งข้อมูลมาสั่งให้ไมโครคอนโทรลเลอร์เพื่อทำการควบคุมการเคลื่อนที่ของรถให้เคลื่อนที่ตามเป้าหมายได้ถูกต้อง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานรูปที่ 3.1 โครงสร้างของระบบ อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 โครงสร้างของโปรแกรม

ในส่วนของโปรแกรมที่ใช้ในการประมวลผลภาพในระบบการติดตามเป้าหมายนี้ เขียนขึ้นโดยใช้โปรแกรม Delphi 5 ซึ่งอยู่บนพื้นฐานของภาษาปาสคาล โดยในการเขียนโปรแกรมเพื่อทำการประมวลผลภาพนี้ได้แบ่งออกเป็นส่วนของโปรแกรมหลัก และส่วนของการคำนวณเวกเตอร์ของการเคลื่อนที่ของเป้าหมายไป

ในส่วนของ การติดตามเป้าหมายจะใช้ไมโครคอนโทรลเลอร์เป็นส่วนควบคุมการเคลื่อนที่ของรถ โดยจะควบคุมลักษณะการหมุนของมอเตอร์กระแสตรง ซึ่งจะใช้ภาษาแอสเซมบลี

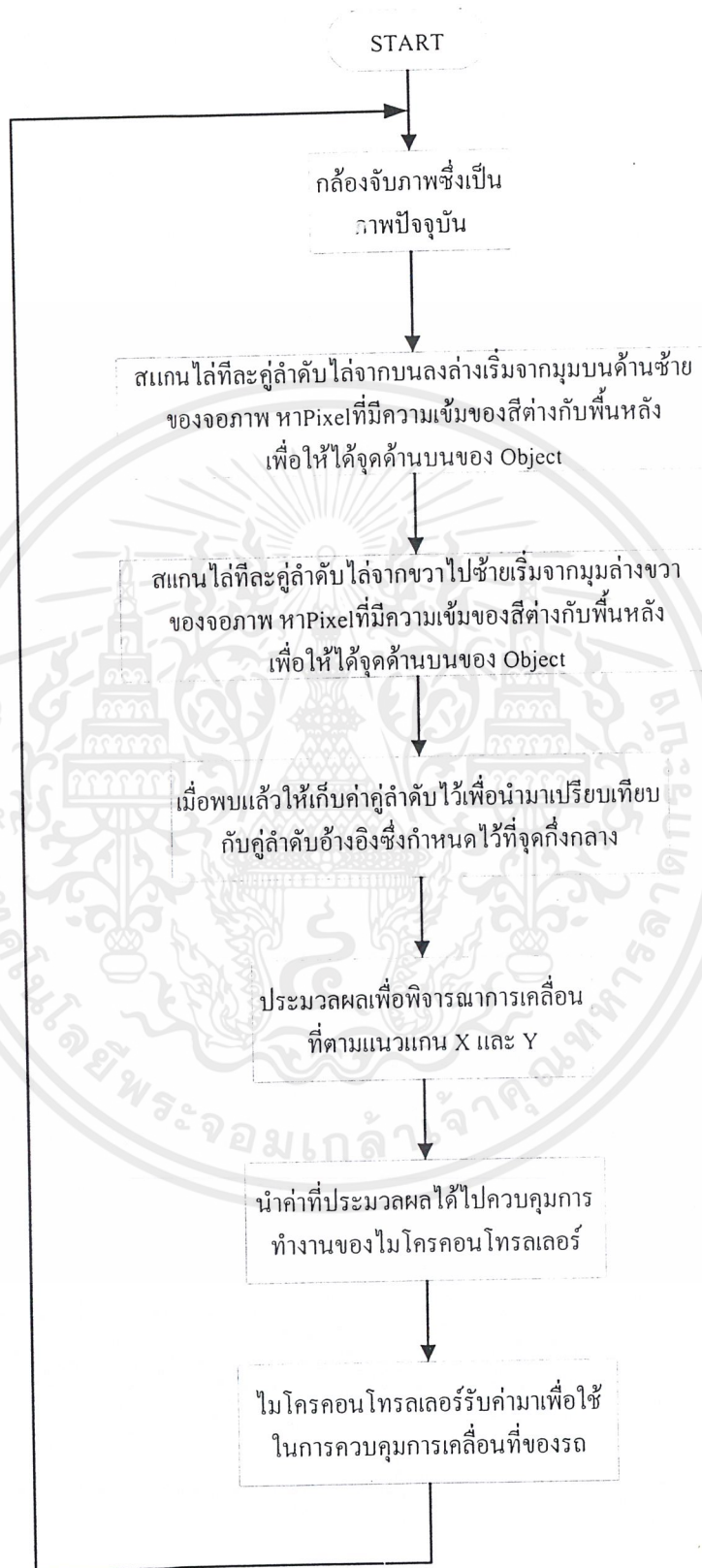
3.2.1 โปรแกรมการทำงานของ Delphi

ในส่วนการทำงานของโปรแกรม Delphi จะแบ่งออกเป็น 2 ส่วนหลักใหญ่ ๆ คือส่วนที่ใช้ในการประมวลผลภาพ และส่วนในการหาเวกเตอร์ของการเคลื่อนที่ของภาพของเป้าหมาย

โปรแกรมในส่วนของการประมวลผลภาพ

จากรูปที่ 3.2 แสดงแผนการทำงานของ การประมวลผลภาพของคอมพิวเตอร์ โดยหลักการการทำงานเริ่มจาก กล้องวีดีโอดิจิตอลรับภาพของเป้าหมายมา ซึ่งเป็นภาพปัจจุบัน โดยขนาดของเฟรมที่แสดงภาพของเป้าหมายมีขนาด 160 x 120 pixels แล้วจึงทำการเปรียบเทียบค่าความเข้มของสีในคู่พิกเซลที่อยู่ติดกัน ซึ่งเริ่มจากมุมบนด้านซ้ายของเฟรมรูปภาพ ไล่แสดกนในแนวแกน X ก่อน แล้วจึงไล่ลงมาตามแนวแกน Y จนพบคู่ลำดับที่มีค่าผลต่างของความเข้มของสีเท่ากับที่ได้กำหนดไว้ ซึ่งจะได้จุดบนสุดของเป้าหมาย เก็บค่าตำแหน่งของจุดนี้ไว้ในคู่ลำดับ (X11,Y11) แล้วเปลี่ยนโดยอาศัยหลักการเดิมแต่เปลี่ยนมาเริ่มจากมุมบนขวาของเฟรมรูปภาพ โดยไล่แสดกนในแนวแกน Y ก่อน แล้วจึงไล่มาตามแนวแกน X จนพบคู่ลำดับที่มีค่าผลต่างของความเข้มของสีเท่ากับค่าที่ได้กำหนดไว้ ซึ่งจะได้จุดด้านขวาของเป้าหมาย เก็บค่าตำแหน่งของจุดนี้ไว้ในคู่ลำดับ (X21,Y21)

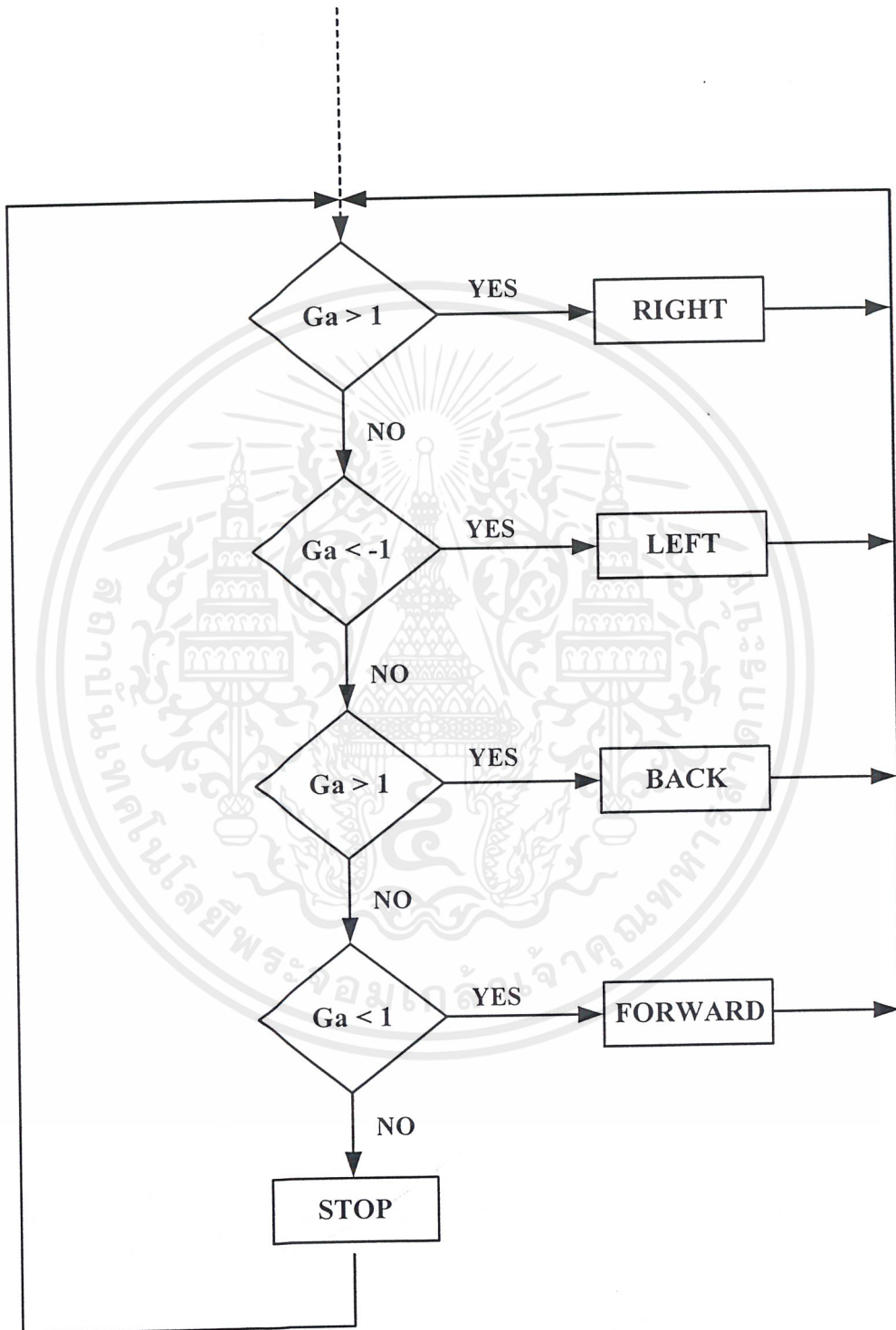
จากการทำตามขั้นตอนดังกล่าวมาแล้วข้างต้นจะได้ตำแหน่งที่อยู่ของเป้าหมายในปัจจุบัน ซึ่งเมื่อนำค่าตำแหน่งดังกล่าวมาเปรียบเทียบกับค่าตำแหน่งอ้างอิงที่กำหนดไว้ ก็คือตำแหน่งกึ่งกลางของภาพ 80 x 60 เราก็จะได้ตำแหน่งที่วัตถุนั้นได้เคลื่อนที่ไป ซึ่งในการหาตำแหน่งที่วัตถุได้เคลื่อนที่ไปนั้นจะกล่าวในหัวข้อถัดไป



รูปที่ 3.2 แผนภาพการทำงานหลักของโปรแกรม Delphi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมในส่วนของการติดตามเป้าหมาย



รูปที่ 3.3 แผนภาพการประมวลผลเพื่อหาตำแหน่งที่วัตถุได้เคลื่อนที่ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการประมวลผลภาพจะได้ตำแหน่งของวัตถุคือ คู่ลำดับ (X11,Y11) และ คู่ลำดับ (X21,Y21) นำค่าคู่ลำดับนี้มาเปรียบเทียบกับตำแหน่งกึ่งกลางของเฟรมภาพ เพื่อให้ได้เวกเตอร์ของเป้าหมายที่เคลื่อนที่ไป โดยทำดังสมการ

$$Xa = X11-80 \quad \dots\dots\dots 3.1$$

$$Ya = Y21-60 \quad \dots\dots\dots 3.2$$

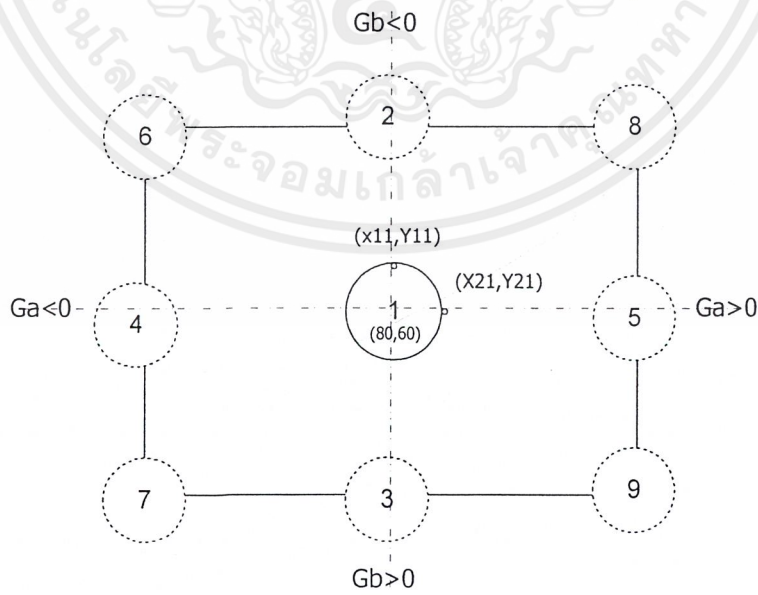
จากค่าที่ได้มาจะทราบระยะที่เป้าหมายได้เคลื่อนที่ไป แต่ในหลักความเป็นจริง การเคลื่อนที่ของเป้าหมายในเฟรมภาพ จะมีค่ามากกว่าเมื่อเทียบกับการเคลื่อนที่ของเป้าหมายจริง ๆ และอาจมีค่าผิดพลาดที่จะเกิดจากการสั้นของการเคลื่อนที่ของรถ หรืออาจเกิดจากการสั้นของแสงทำให้เงาของเป้าหมายสั้นไหว มีผลทำให้การสั่งงานหรือค่าที่ใช้ในการสั่งการเคลื่อนที่ของรถมีค่าไม่คงที่ ทำให้มีการเคลื่อนที่ที่ผิดพลาดได้ ดังนั้นจึงนำค่าผลต่างที่ได้มาทำการหารค่าคงที่ค่าหนึ่งเพื่อลดปัญหาที่อาจเกิดขึ้นดังกล่าวมาแล้ว โดยใช้สมการดังนี้

$$Ga = Xa / 8 \quad \dots\dots\dots 3.3$$

$$Gb = Xb / 6 \quad \dots\dots\dots 3.4$$

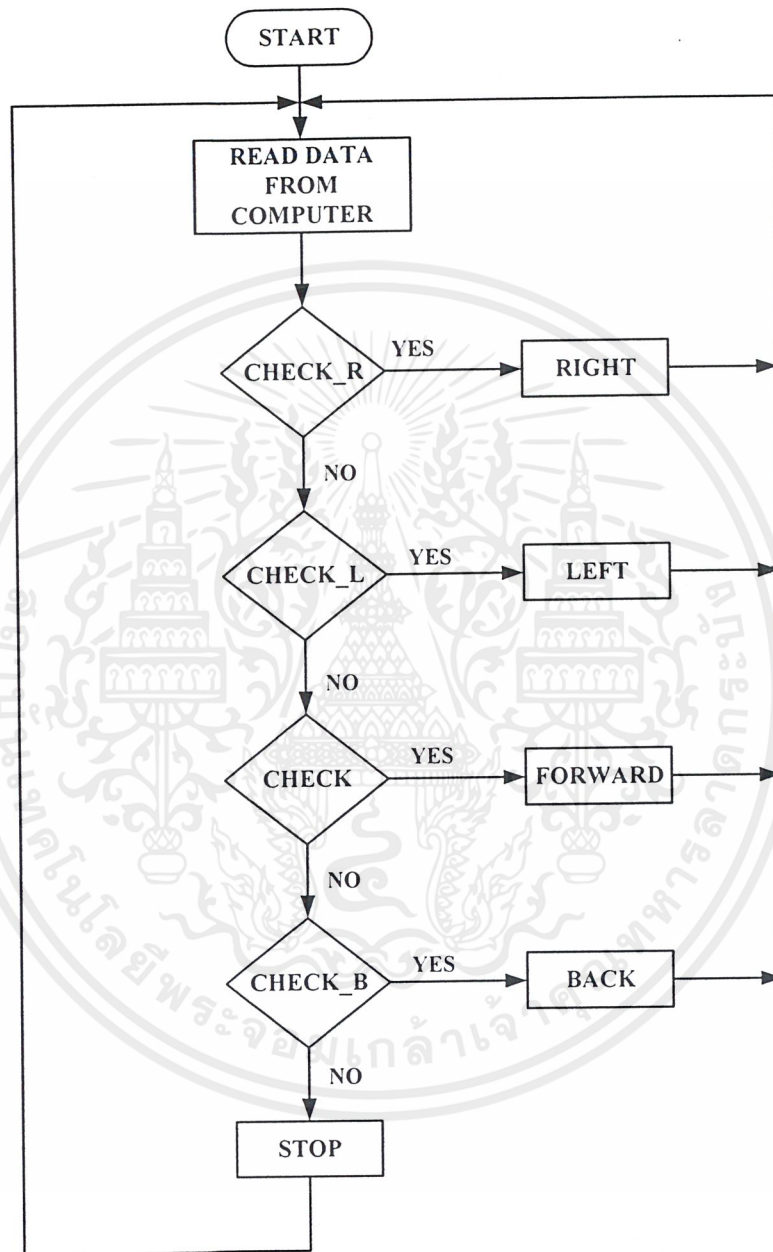
แล้วนำค่าที่ได้จากการคำนวณมาเป็นค่าอ้างอิงในการสั่งการเคลื่อนที่ของรถ โดยจะพิจารณาในการเคลื่อนที่ในแนวแกน X ก่อน แล้วจึงจะพิจารณาการเคลื่อนที่ในแนวแกน Y ดังแสดงแผนภาพในรูปที่ 3.3

อาจจะสรุปการเคลื่อนที่ของเป้าหมายที่สามารถจะเป็นไปได้ดังรูปที่ 3.4



รูปที่ 3.4 แสดงภาพสรุปตำแหน่งของเป้าหมายที่เคลื่อนที่ไปได้ในเฟรมภาพ

3.2.2 โปรแกรมการทำงานของไมโครคอนโทรลเลอร์

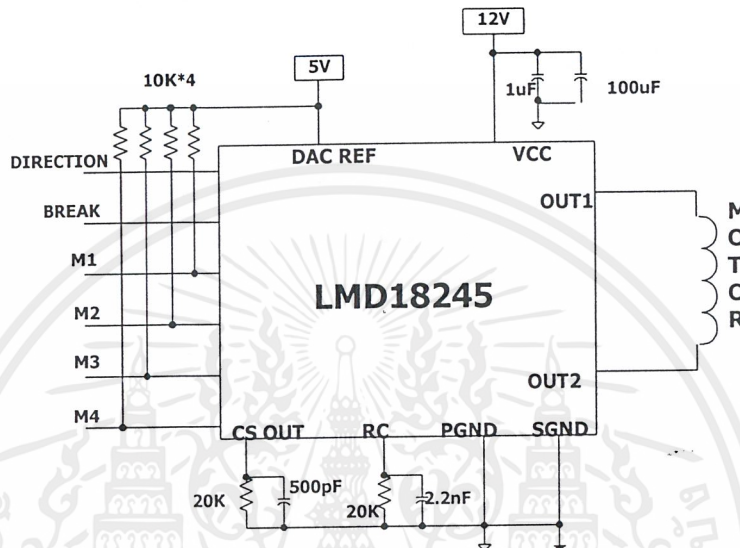


รูปที่ 3.5 แผนภาพการทำงานของไมโครคอนโทรลเลอร์

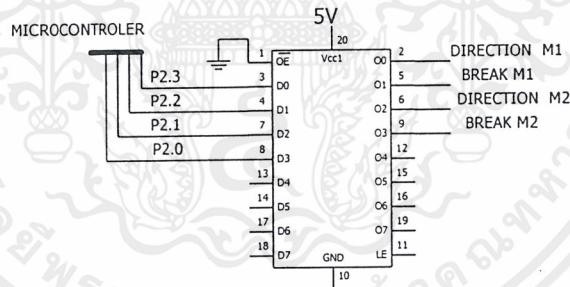
โปรแกรมในส่วนของคอนโทรลเลอร์จะใช้ในการควบคุมการเคลื่อนที่ของรถ โดยจะรับคำสั่งจากคอมพิวเตอร์ ดังนั้นโปรแกรมการทำงานของไมโครคอนโทรลเลอร์จึงเริ่มต้นจากการรับคำสั่งจากคอมพิวเตอร์ผ่านทางพอร์ทขนาน ซึ่งเป็นข้อมูลขนาด 8 บิต แล้วทำการตรวจสอบบิตข้อเอกสารนี้ข้อมูลที่ได้รับมาจากคอมพิวเตอร์ เพื่อนำมาเป็นคำสั่งในการเคลื่อนที่ของรถ นั่นคือการควบคุมทิศทางไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของการเคลื่อนที่ของมอเตอร์กระแสตรง โดยพิจารณาแนวการเคลื่อนที่ในแนวแกน X ก่อน แล้วจึงพิจารณาในแนวแกน Y

3.3 วงจรขับมอเตอร์กระแสตรง



รูปที่ 3.6 แสดงวงจรขับมอเตอร์กระแสตรง



รูปที่ 3.7 แสดงขาการต่อวงจรขับมอเตอร์กับ IC เบอร์ 74LS373

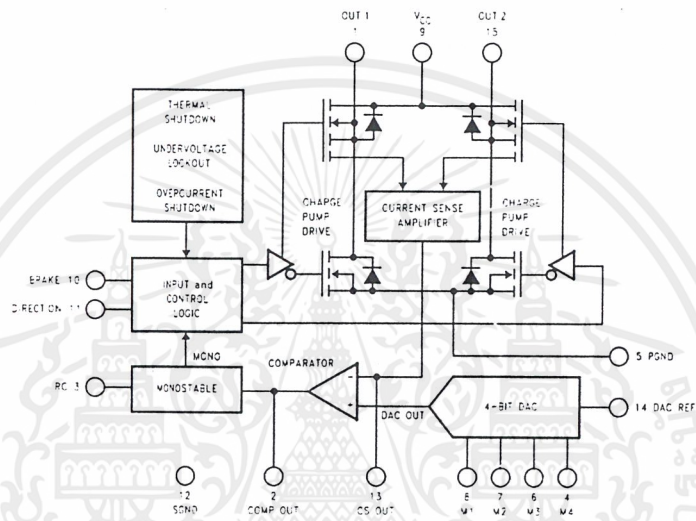
จากรูปวงจรจะใช้ IC เบอร์ LMD18245 เป็นตัวขับมอเตอร์กระแสตรง ซึ่งใช้ไมโครคอนโทรลเลอร์เป็นตัวกำหนดทิศทางเคลื่อนที่ และการหยุดของมอเตอร์ โดยส่งผ่านมายัง IC เบอร์ 74LS373 ทำหน้าที่เป็นบัฟเฟอร์ก่อนที่จะเข้ามาสั่ง IC ขับมอเตอร์

สำหรับการทำงานของ IC เบอร์ LMD18245 นี้ จะสามารถศึกษาการทำงานได้โดยดูจากโครงสร้างภายใน IC ดังรูปที่ 3.8 ซึ่งจะมีวงจรบริดจ์ (H-Bridge) ที่ใช้อุปกรณ์โซลิตสแตทเป็นสวิตช์ และมีไดโอดช่วยในการป้องกันกระแสย้อนกลับ (back e.m.s.) และช่วยในการหยุดมอเตอร์ ซึ่งสามารถอธิบายหลักการทำงานของวงจร H-Bridge ได้คือ เมื่อทำการ ON S1 และ S4 พบว่าจะมีกระแสไหลดังรูปที่ 3.9(a) และเมื่อทำการ OFF S4 พบว่ากระแสจะไหลผ่านไดโอดที่ต่อคร่อม S2

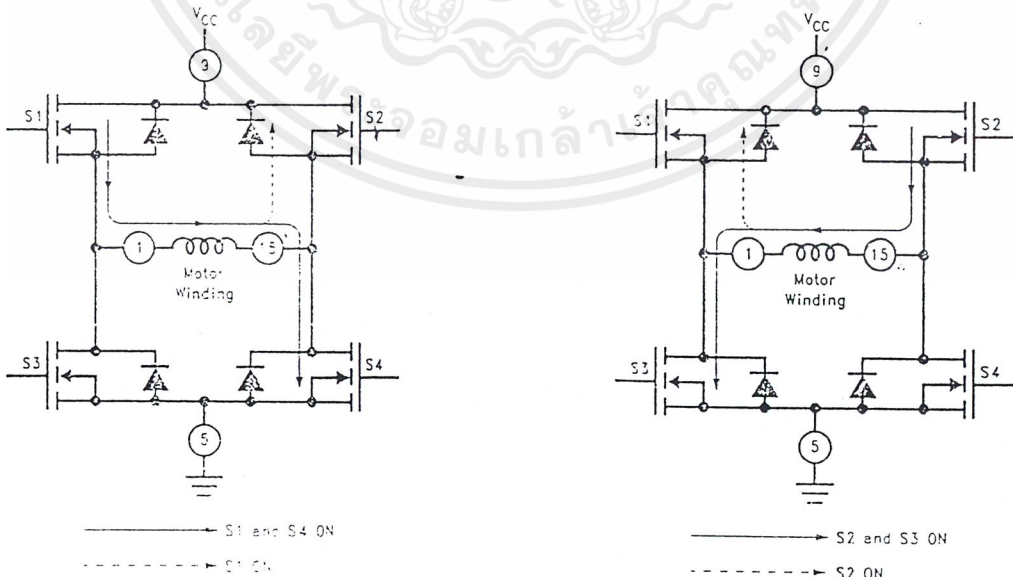
ซึ่งเปรียบเหมือนเป็นการ Short circuit ทำให้กระแสไหลในมอเตอร์มีค่าประมาณ 0 ซึ่งทำให้

เอกสารนี้เป็นทรัพย์สินทางปัญญาของบริษัทฯ เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาตในการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเปรียบเหมือนเป็นการShort circuit ทำให้กระแสไหลในมอเตอร์มีค่าประมาณ 0 ซึ่งทำให้มอเตอร์เคลื่อนที่ช้าลงจนหยุด และจากรูปที่ 3.9 (b) เมื่อทำการ ON S2 และ S3 จะมีผลให้มอเตอร์เคลื่อนที่ในทิศตรงกันข้ามกับกรณีแรก และในลักษณะเดียวกันเมื่อ OFF S3 พบว่ากระแสจะไหลผ่านไดโอดที่ต่อคร่อม S1 ซึ่งจะทำให้มอเตอร์เคลื่อนที่ช้าลงจนหยุด



รูปที่ 3.8 แสดงวงจร โครงสร้างภายใน IC เบอร์ LMD18245



(a)

(b)

เอกสารนี้เป็นเอกสารที่รูปที่ 3.9 แสดงลักษณะการไหลของกระแสในวงจร DMOS H-Bridge โดยใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

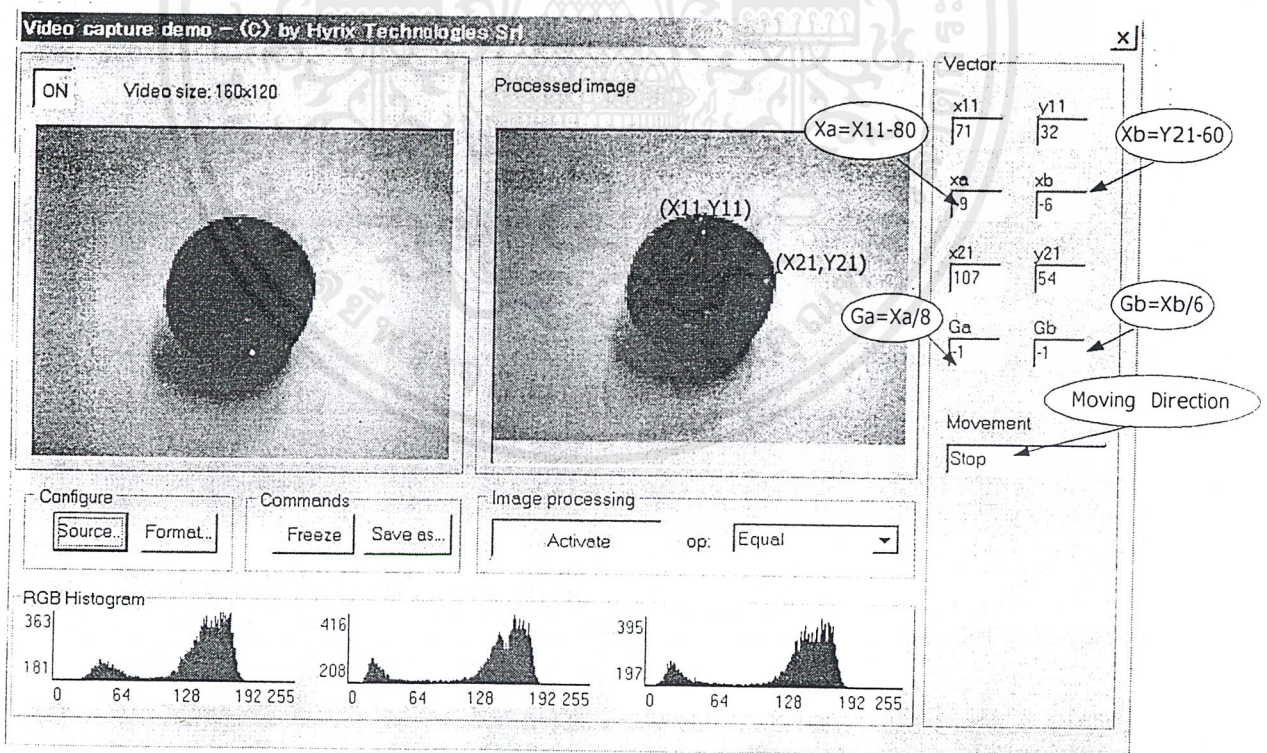
การทดลองและผลการทดลอง

4.1 การทดลองเพื่อทดสอบการทำงานของโปรแกรมเมื่อเป้าหมายมีการเคลื่อนที่

สำหรับการทดลองจะทำการศึกษาการทำงานของโปรแกรม เมื่อมีการเคลื่อนที่ของเป้าหมายไปในตำแหน่งต่าง ๆ จะได้ผลเป็นไปตามเงื่อนไขของโปรแกรมที่เขียนไว้หรือไม่

โดยการทดลองจะทดสอบให้เป้าเคลื่อนที่ไปในตำแหน่งต่าง ๆ 5 ตำแหน่ง แล้วสังเกตผลการทงงาน และการสั่งการของโปรแกรมว่าถูกต้องหรือไม่ โดยสามารถดูผลได้จากการแสดงผลออกทางหน้าจอแสดงผลของโปรแกรม ดังนี้

- เมื่อให้เป้าหมายอยู่ตรงกลางของเฟรมภาพ จะแสดงผลดังนี้

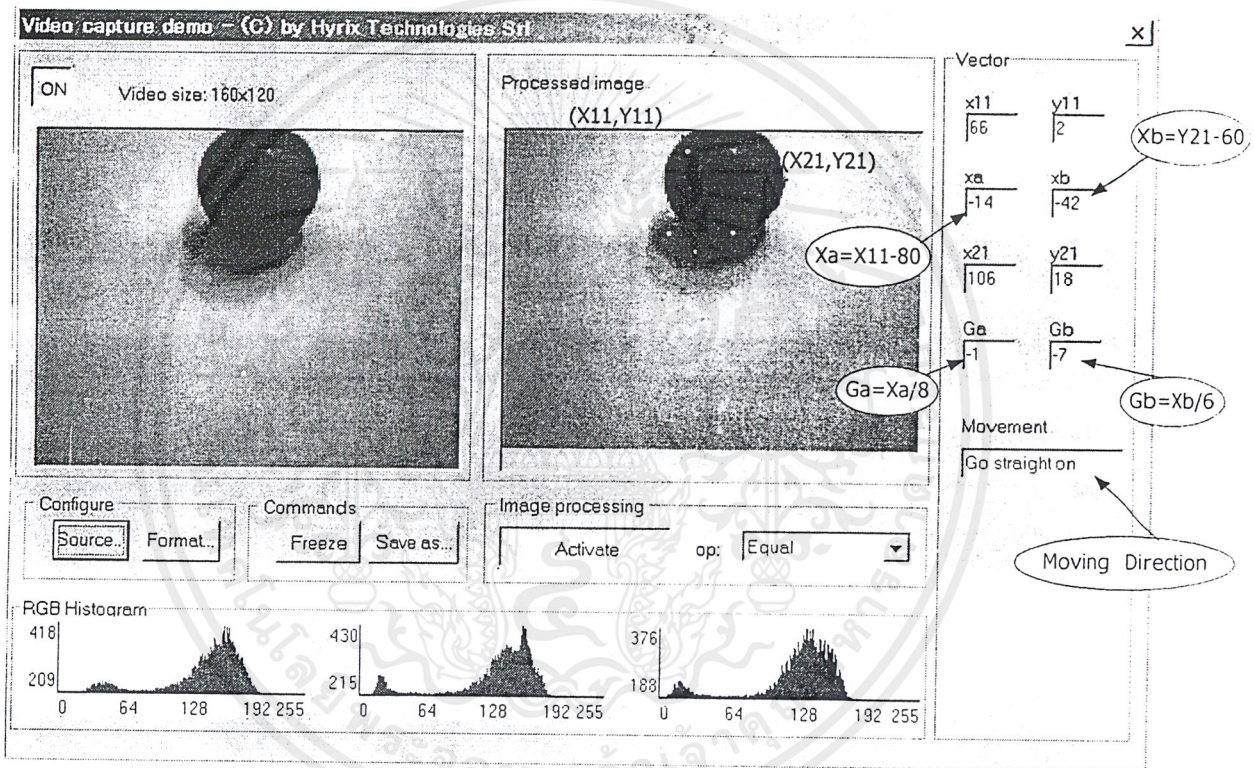


รูปที่ 4.1 ภาพหน้าจอของโปรแกรมเมื่อเป้าหมายอยู่ตรงกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.1 จะพบว่าการประมวลผลภาพให้ผลที่ถูกต้องคือ เมื่อเป้าหมายอยู่ตรงกลาง เฟรมภาพ คอมพิวเตอร์จะสั่งให้หยุดการเคลื่อนที่ของรถ ซึ่งจะแสดงผลออกทางช่อง Movement โดยจะส่งค่าออกทางพอร์ทขนานเพื่อไปส่งค่าให้กับไมโครคอนโทรลเลอร์ ให้สั่งหยุดการเคลื่อนที่ของมอเตอร์กระแสตรง และจากการทดลองพบว่ารถได้หยุดการเคลื่อนที่ดังที่ได้สั่งการ

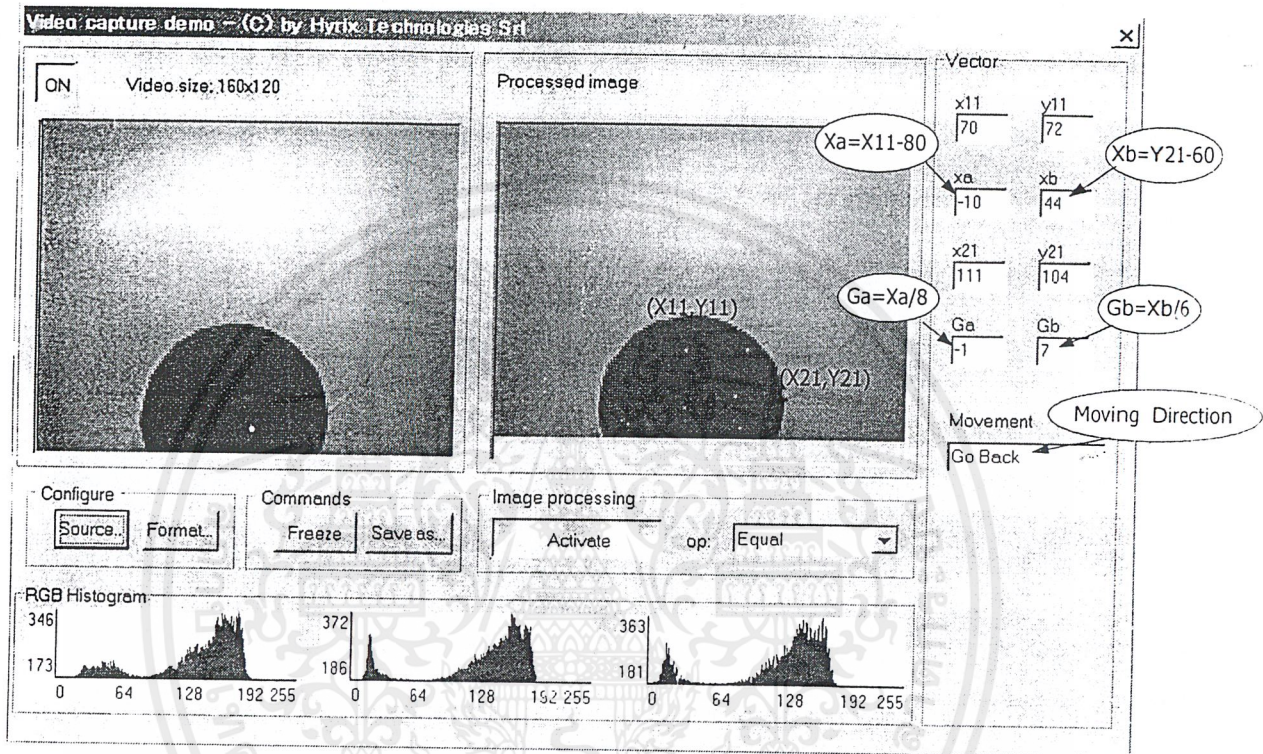
- เมื่อเราเคลื่อนที่เป้าหมายไปด้านบน จะแสดงผลดังนี้



รูปที่ 4.2 ภาพหน้าจอของโปรแกรมเมื่อเป้าหมายอยู่ด้านบน

จากรูปที่ 4.2 จะพบว่าการประมวลผลภาพให้ผลที่ถูกต้องคือ เมื่อเป้าหมายเคลื่อนที่ไปอยู่ด้านบนของเฟรมภาพ คอมพิวเตอร์จะสั่งให้รถเคลื่อนที่ไปทางด้านหน้า ซึ่งจะแสดงผลออกทางช่อง Movement โดยจะส่งค่าออกทางพอร์ทขนานเพื่อไปส่งค่าให้กับไมโครคอนโทรลเลอร์ ให้สั่งมอเตอร์กระแสตรงให้เคลื่อนที่ไปด้านหน้า และจากการทดลองพบว่ารถได้เคลื่อนที่ไปด้านหน้าดังที่ได้สั่งการ

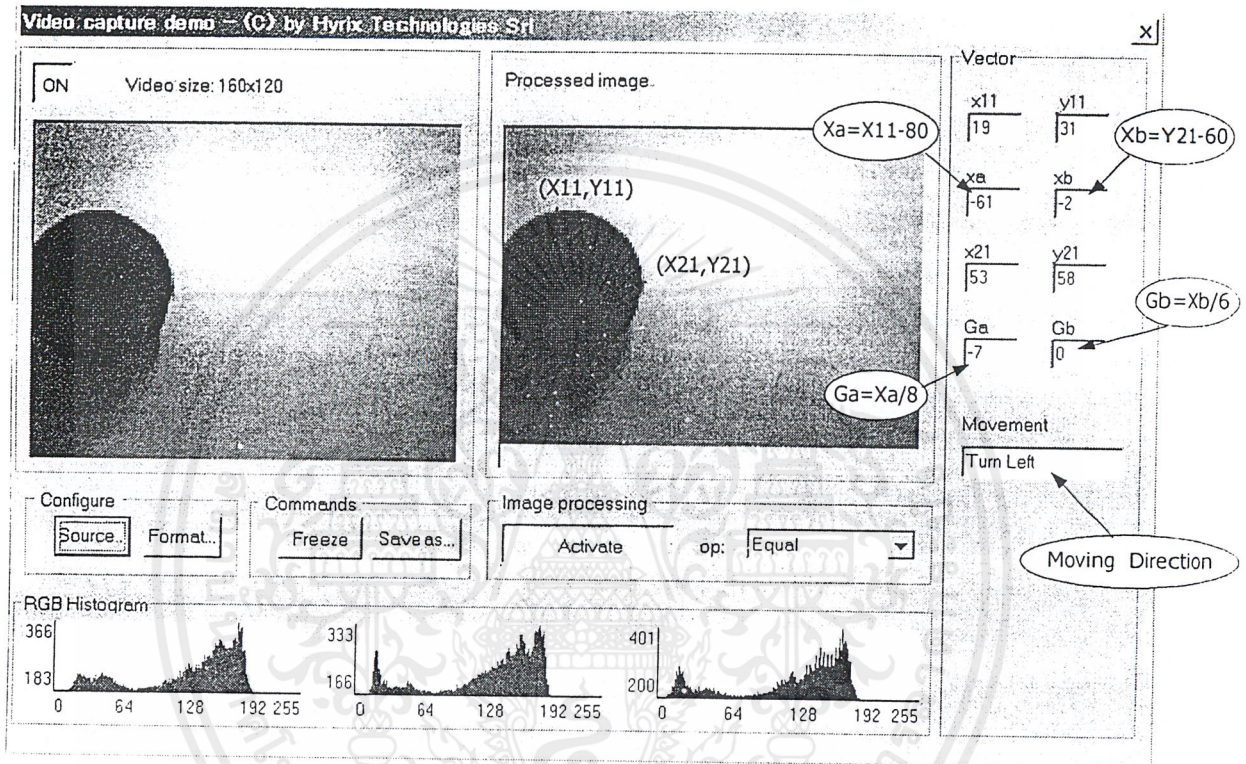
- เมื่อเราทำการเคลื่อนที่เป้าหมายไปทางด้านล่าง จะแสดงผลดังนี้



รูปที่ 4.3 ภาพหน้าจอของโปรแกรมเมื่อเป้าหมายอยู่ด้านล่าง

จากรูปที่ 4.3 จะพบว่าการประมวลผลภาพให้ผลที่ถูกต้องคือ เมื่อเป้าหมายเคลื่อนที่ไปอยู่ด้านล่างของเฟรมภาพ คอมพิวเตอร์จะสั่งให้รถเคลื่อนที่ถอยไปด้านหลัง ซึ่งจะแสดงผลออกทางช่อง Movement โดยจะส่งค่าออกทางพอร์ทขนานเพื่อไปส่งค่าให้กับไมโครคอนโทรลเลอร์ ให้สั่งมอเตอร์กระแสตรงให้เคลื่อนที่ไปด้านหลัง และจากการทดลองพบว่ารถได้เคลื่อนที่ไปด้านหลังดังที่ได้สั่งการ

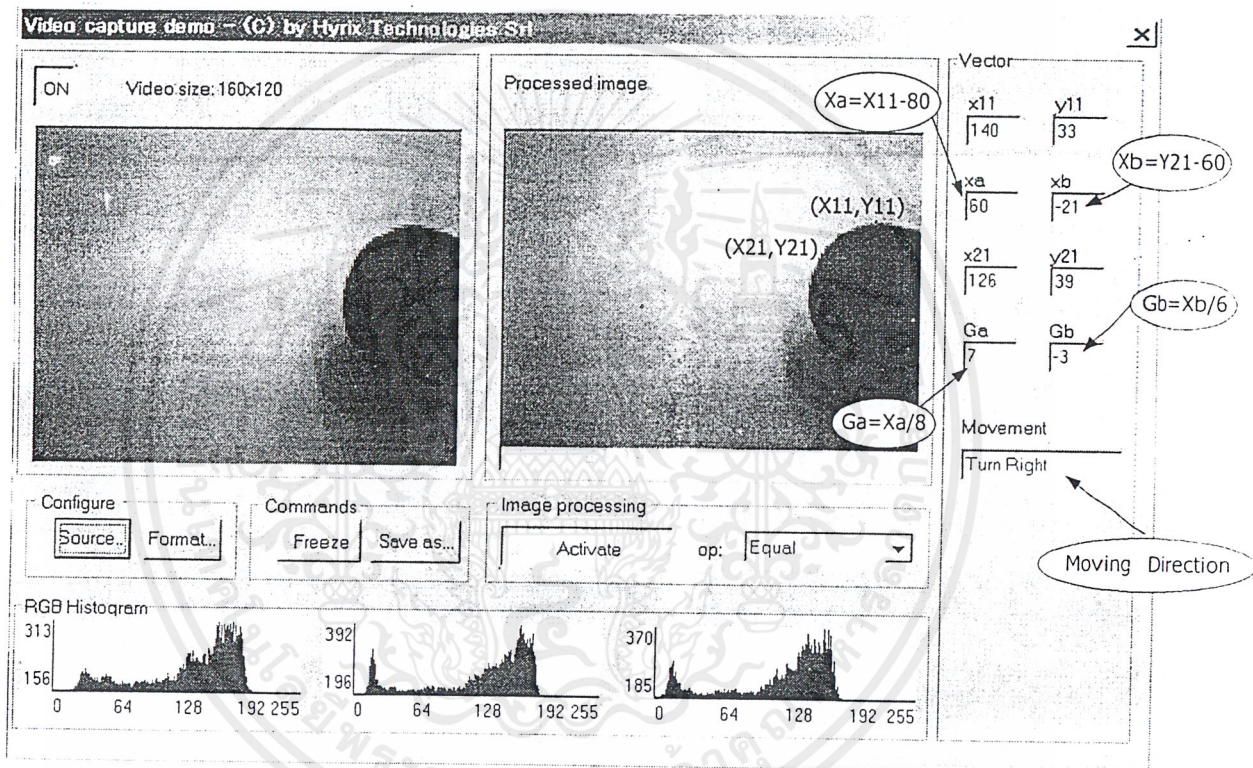
- เมื่อเราเคลื่อนที่เป้าหมายไปทางด้านซ้าย จะแสดงผลดังนี้



รูปที่ 4.4 ภาพหน้าจอของโปรแกรมเมื่อเป้าหมายอยู่ด้านซ้าย

จากรูปที่ 4.4 จะพบว่าการประมวลผลภาพให้ผลที่ถูกต้องคือ เมื่อเป้าหมายเคลื่อนที่ไปอยู่ด้านซ้ายของเฟรมภาพ คอมพิวเตอร์จะสั่งให้รถเคลื่อนที่ถอยไปด้านซ้าย ซึ่งจะแสดงผลออกทางช่อง Movement โดยจะส่งค่าออกทางพอร์ทขนานเพื่อไปส่งค่าให้กับไมโครคอนโทรลเลอร์ ให้สั่งมอเตอร์กระแสตรงให้เคลื่อนที่โดยจะสับการเคลื่อนที่กัน คือล้อทางด้านซ้ายจะเคลื่อนที่ไปทางด้านหลัง ส่วนล้อทางด้านขวาจะเคลื่อนที่ไปด้านหน้า เพื่อให้รถเคลื่อนที่ไปทางด้านซ้าย และจากการทดลองพบว่ารถได้เคลื่อนที่ไปด้านหลังดังที่ได้สั่งการ

- เมื่อเราเคลื่อนที่เป้าหมายไปทางด้านขวา จะแสดงผลดังนี้



รูปที่ 4.5 ภาพหน้าจอของโปรแกรมเมื่อเป้าหมายอยู่ด้านขวา

จากรูปที่ 4.5 จะพบว่าการประมวลผลภาพให้ผลที่ถูกต้องคือ เมื่อเป้าหมายเคลื่อนที่ไปอยู่ด้านขวาของเฟรมภาพ คอมพิวเตอร์จะสั่งให้รถเคลื่อนที่ถอยไปด้านขวา ซึ่งจะแสดงผลออกทางช่อง Movement โดยจะส่งค่าออกทางพอร์ตขนานเพื่อไปส่งค่าให้กับไมโครคอนโทรลเลอร์ให้สั่งมอเตอร์กระแสตรงให้เคลื่อนที่ โดยจะสับการเคลื่อนที่กัน คือล้อทางด้านซ้ายจะเคลื่อนที่ไปทางด้านหน้า ส่วนล้อทางด้านขวาจะเคลื่อนที่ไปด้านหลัง เพื่อให้รถเคลื่อนที่ไปทางด้านขวา และจากเอกสารนี้เราทดลองพบว่ารถได้เคลื่อนที่ไปด้านหลังดังที่ได้สั่งการนั้น ไม่นานนักให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะพบว่าจากการเคลื่อนที่ของเป้าหมายไปในตำแหน่งต่าง ๆ การทำงานของโปรแกรมได้ทำงานได้อย่างถูกต้อง และสั่งการให้รถเคลื่อนที่ไปได้อย่างถูกต้อง พบว่าถ้าเป้าหมายอยู่ในตำแหน่งอื่น ๆ รถจะเคลื่อนไปตามแนวแกน X ก่อน แล้วจึงเคลื่อนที่ไปในแนวแกน Y ซึ่งการเคลื่อนที่ในแนวต่างๆ นี้ ก็ทำงานตามผลการทดลองที่ได้ทำการทดลองไปแล้วข้างต้นนั่นเอง

จากการทดลองดังกล่าว เราจะสั่งให้รถเคลื่อนที่ในทิศทางต่าง ๆ ได้โดยการส่งไปที่ขา 10 และ 11 ของไอซีเบอร์ LMD 18245 โดยจะในหัวข้อถัดไปนี้

4.2 การทดลองและผลที่ได้จากการควบคุมมอเตอร์กระแสตรง

จากการศึกษาการทำงานของวงจรขับมอเตอร์กระแสตรง โดยใช้ไอซีเบอร์ LMD18245 ซึ่งความเร็วในการเคลื่อน ทิศทางของการเคลื่อนที่ และการหยุดของมอเตอร์กระแสตรง จะใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุม ซึ่งให้ผลการควบคุมทิศทางการเคลื่อนที่ของมอเตอร์กระแสตรงแสดงดังตารางที่ 4.1

ตารางที่ 4.1 การควบคุมทิศทางและการหยุดของมอเตอร์กระแสตรง

การสั่งงานด้วยไมโครคอนโทรลเลอร์	ผลที่ได้	
ควบคุมทิศทาง (ขา11ของLMD18245)	H	มอเตอร์หมุนไปข้างหน้า
	L	มอเตอร์หมุนไปข้างหลัง
ควบคุมการเคลื่อนที่ (ขา10ของLMD18245)	H	มอเตอร์หยุดหมุน
	L	มอเตอร์หมุน

4.3 การทดลองในหัวข้ออื่น ๆ ที่มีผลต่อการทำงานของระบบ

จากการทดลองในด้านปัจจัยต่าง ๆ ที่มีผลต่อการทำงานของระบบ สามารถแบ่งออกเป็นหัวข้อต่าง ๆ และได้ผลการทดลอง ดังจะได้อธิบายโดยสรุปตามตารางดังต่อไปนี้

ตารางที่ 4.2 แสดงผลการทดลองในการทดลองปัจจัยต่าง ๆ ที่มีผลต่อการทำงานของระบบ

หัวข้อการทดลอง	ผลการทดลอง
สภาพของแสง	ต้องมีความสว่างพอเพียง
ดีเลย์ใหม่ในการประมวลผล	มีค่าพอสมควร ซึ่งจะขึ้นอยู่กับความเร็วของคอมพิวเตอร์ด้วย
ความเร็วในการเคลื่อนที่ของเป้าหมาย	ต้องไม่เร็วเกินไป เพื่อให้สามารถประมวลผลภาพได้ทัน และทำให้ระบบมีการทำงานที่ถูกต้องมากขึ้น
ความผิดพลาดของ Pixel	ในแนวแกน X 0-8 Pixel ในแนวแกน Y 0-6 Pixel
ขนาดของเป้าหมาย	ต้องไม่เล็กเกินไป อย่างน้อยต้องมองเห็นเป็นสี่ทีมากกว่าพื้นหลังมาก
สีของพื้นหลังเป้าหมาย	ต้องมีสีอ่อนมาก และเป็นสีที่สว่างกว่าสีของเป้าหมายมาก ซึ่งควรใช้สีขาวจะดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์การทดลอง

โครงการนี้เป็นแบบกำลังการทำงานของระบบการติดตามเป้าหมายโดยอัตโนมัติ ซึ่งใช้หลักการของการประมวลผลภาพที่ได้จากกล้องวิดีโอดิจิทัล เราสามารถนำไปประยุกต์ใช้ในโรงงานอุตสาหกรรมหรือสถานที่ที่มีอันตรายที่มนุษย์ไม่สามารถเข้าไปได้

ระบบการทำงานที่ได้สร้างขึ้นมานี้เริ่มจากการรับภาพเป้าหมายที่ต้องการติดตามมาทางกล้องดิจิทัล หาคำแหน่งที่แน่นอนที่เป้าหมายอยู่ ถ้าเป้าหมายมีการเคลื่อนที่ คอมพิวเตอร์จะทำการประมวลผลเพื่อให้ได้ทิศทางที่เป้าหมายนั้นได้เคลื่อนที่ไป แล้วส่งข้อมูลผ่านพอร์ตขนานไปสั่งให้ไมโครคอนโทรลเลอร์ทำการควบคุมการเคลื่อนที่ของรถให้สามารถติดตามเป้าหมายไปได้ถูกทิศทาง

จากการทำโครงการนี้ได้พบปัญหาต่างๆ มากมาย รวมทั้งยังมีข้อจำกัดต่าง ๆ ที่จะต้องมีการแก้ไข และพัฒนาให้ดีขึ้น คือ บริเวณที่รถนั้นได้เคลื่อนที่ไปต้องมีแสงสว่างเพียงพอ ไม่มีสิ่งกีดขวางสีพื้นที่กล้องจับได้ต้องเป็นสีขาว เป้าหมายที่ใช้ต้องมีสีเข้มกว่าสีพื้นมาก ความเร็วในการเคลื่อนที่ของรถ และความเร็วของเป้าหมายที่เคลื่อนที่ไปต้องถูกจำกัดไม่ให้เร็วเกินไป เนื่องมาจากข้อจำกัดของความเร็วของคอมพิวเตอร์ที่สามารถประมวลผลได้ทัน และยังมีปัญหาอื่นๆ

อย่างไรก็ดีจากการทำโครงการนี้ทำให้ได้ศึกษาภาษาแอสซีและแอปพลิเคชันต่างๆ วงจรการควบคุมมอเตอร์กระแสตรง การทำงานของไมโครคอนโทรลเลอร์ การติดต่อสื่อสารระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์โดยอาศัยพอร์ตขนาน รวมทั้งยังได้แนวความคิดการประยุกต์ใช้อิมเมจโปรเซสซึ่งด้วย

สำหรับแนวทางการแก้ไขที่อยากเสนอแนะเพื่อใช้ในการพัฒนาให้ระบบมีการทำงานที่ดีขึ้น ควรจะพัฒนาทางด้านการเขียนโปรแกรมให้ดีขึ้นอาจใช้เทคโนโลยีทางด้านโครงข่ายนิเวศ (Neural Network) เพื่อช่วยในการจดจำวัตถุ ทำให้สามารถประมวลผลเพื่อติดตามวัตถุได้โดยไม่ต้องจำกัดสิ่งแวดล้อมมากนัก

หรืออาจพัฒนาทางด้านฮาร์ดแวร์เพื่อควบคุมการทำงานของรถ และการรับส่งภาพแบบไร้สายจากระยะไกลได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

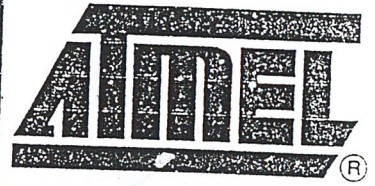
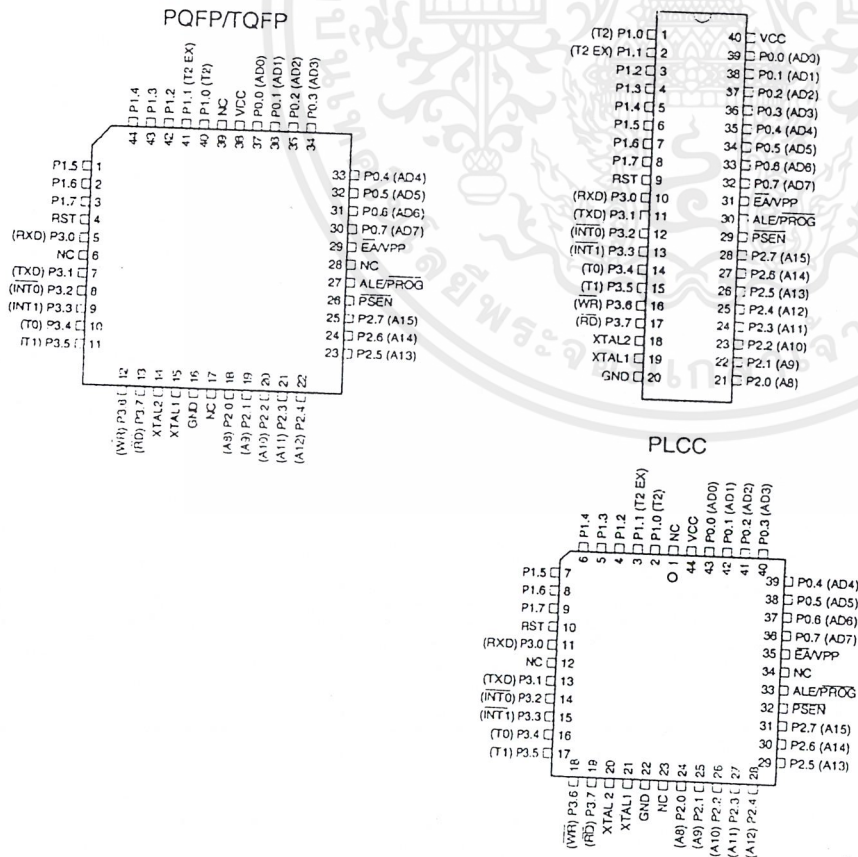
Features

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

Description

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

Pin Configurations



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

AT89C52

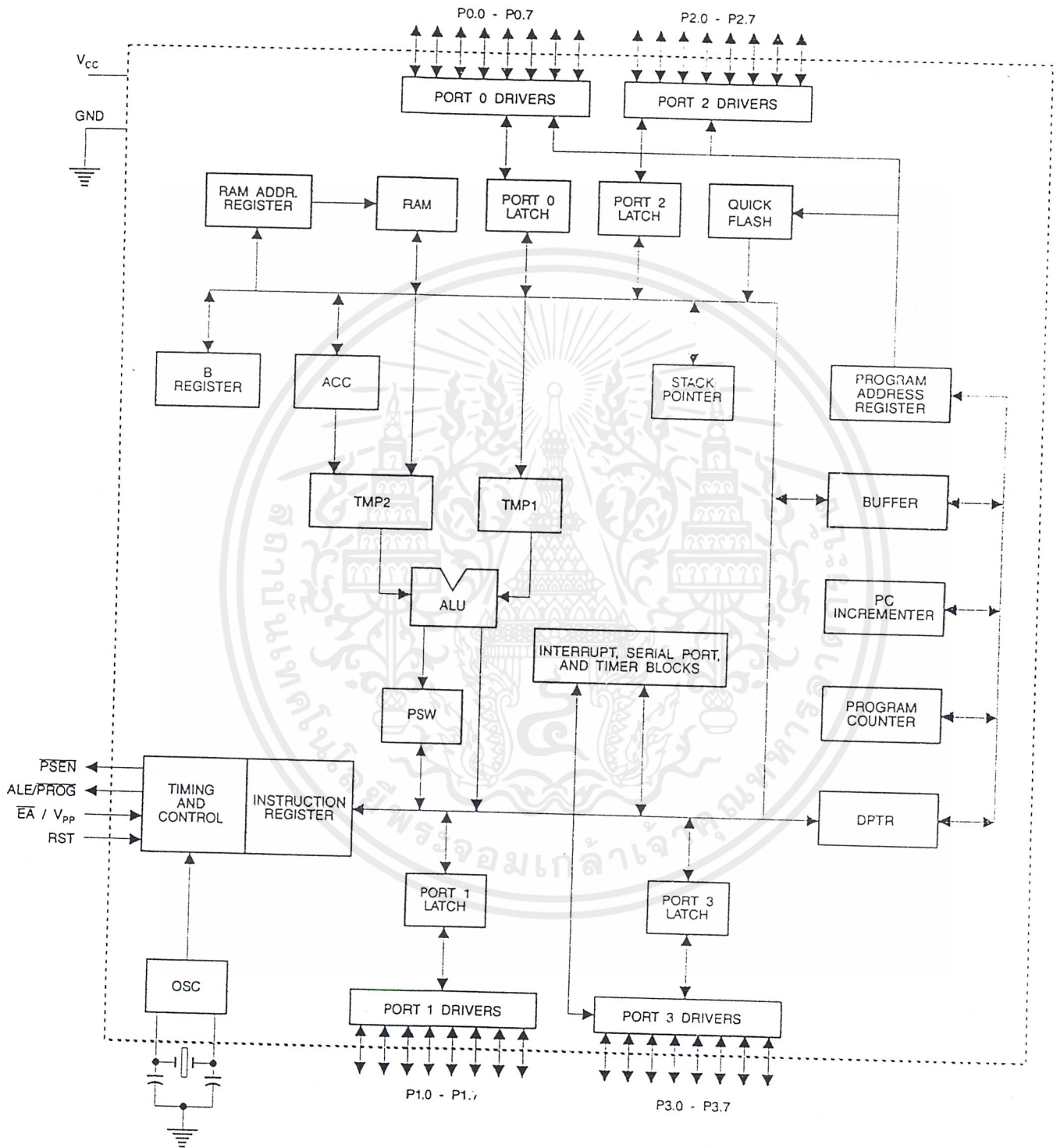
Rev. 0313H-02/00



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Block Diagram



AT89C52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

Pin Description

VCC

Supply voltage.

GND

Ground.

Port 0

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

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

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

Port 1

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

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

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

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

Port 2

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

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

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

Port 3

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

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

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

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (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)

RST

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

ALE/PROG

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

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external





timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

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

PSEN

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

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

EA/VPP

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

EA should be strapped to VCC for internal program executions.

This pin also receives the 12-volt programming enable voltage (VPP) during Flash programming when 12-volt programming is selected.

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

Table 1. AT89C52 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		0CFH
0C0H								0C7H
0B8H	IP XX000000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0X000000							0AFH
0A0H	P2 11111111							0A7H
98H	SCON 00000000	SBUF XXXXXXXX						9FH
90H	P1 11111111							97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		8FH
80H	P0 11111111	SP 00001111	DPL 00000000	DPH 00000000			PCON 0XX00000	87H

AT89C52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Function Registers

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

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

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke

new features. In that case, the reset or inactive values of the new bits will always be 0.

Timer 2 Registers Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 4) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16-bit capture mode or 16-bit auto-reload mode.

Interrupt Registers The individual interrupt enable bits are in the IE register. Two priorities can be set for each of the six interrupt sources in the IP register.

Table 2. T2CON – Timer/Counter 2 Control Register

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

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

Data Memory

The AT89C52 implements 256 bytes of on-chip RAM. The upper 128 bytes occupy a parallel address space to the Special Function Registers. That means the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the address mode used in the instruction

specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions that use direct addressing access SFR space.

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

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





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

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

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

Timer 0 and 1

Timer 0 and Timer 1 in the AT89C52 operate the same way as Timer 0 and Timer 1 in the AT89C51.

Timer 2

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit $C/\overline{T}2$ in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 3. Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

Table 3. Timer 2 Operating Modes

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

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external

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

Capture Mode

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

Auto-reload (Up or Down Counter)

Timer 2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 4). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

Figure 1. Timer in Capture Mode

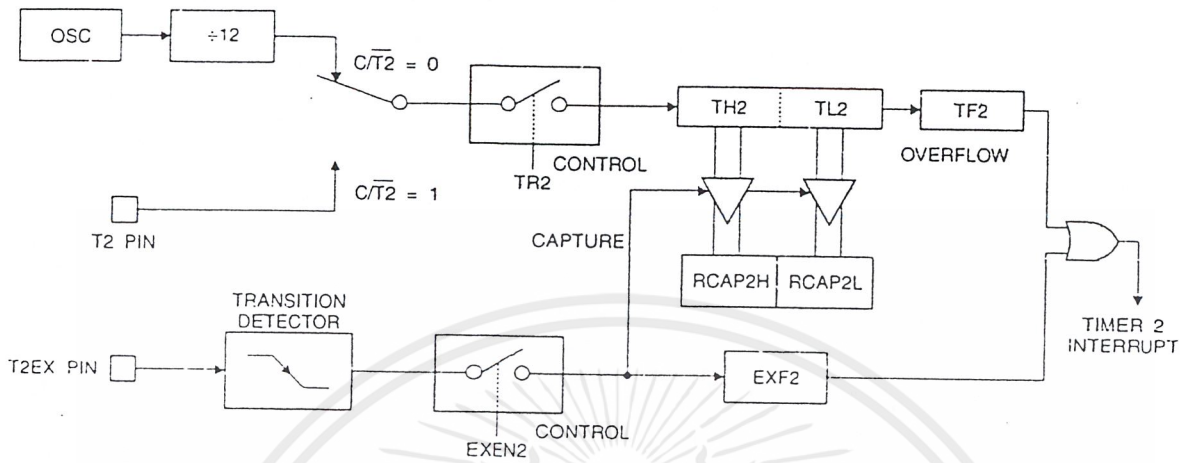


Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L. The values in Timer in Capture Mode RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled. Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 3. In this mode, the T2EX pin controls

the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16-bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

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

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

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

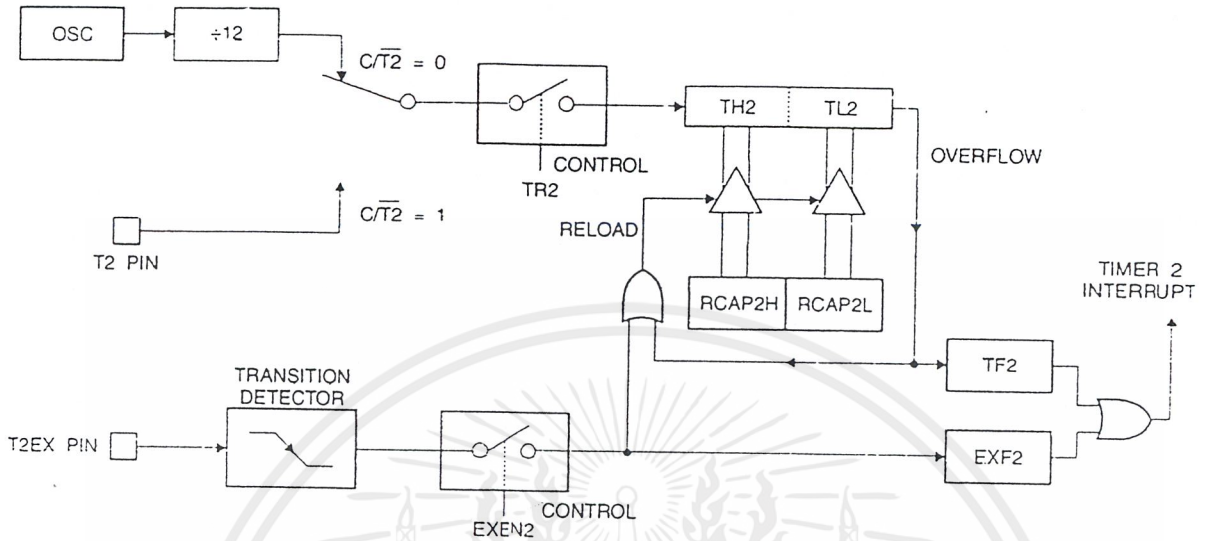


Table 4. T2MOD – Timer 2 Mode Control Register

T2MOD Address = 0C9H						Reset Value = XXXX XX00B		
Not Bit Addressable								
Bit	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	T2OE	DCEN

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

AT89C52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนสื่อออนไลน์
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

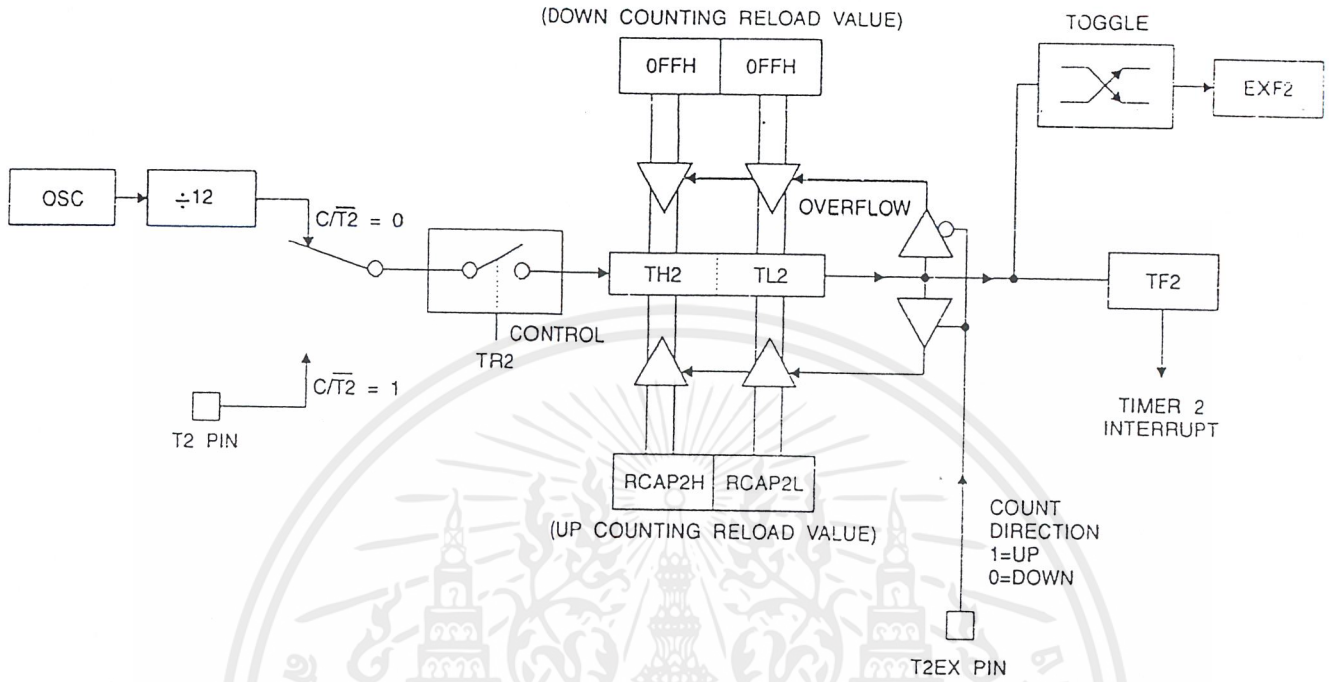
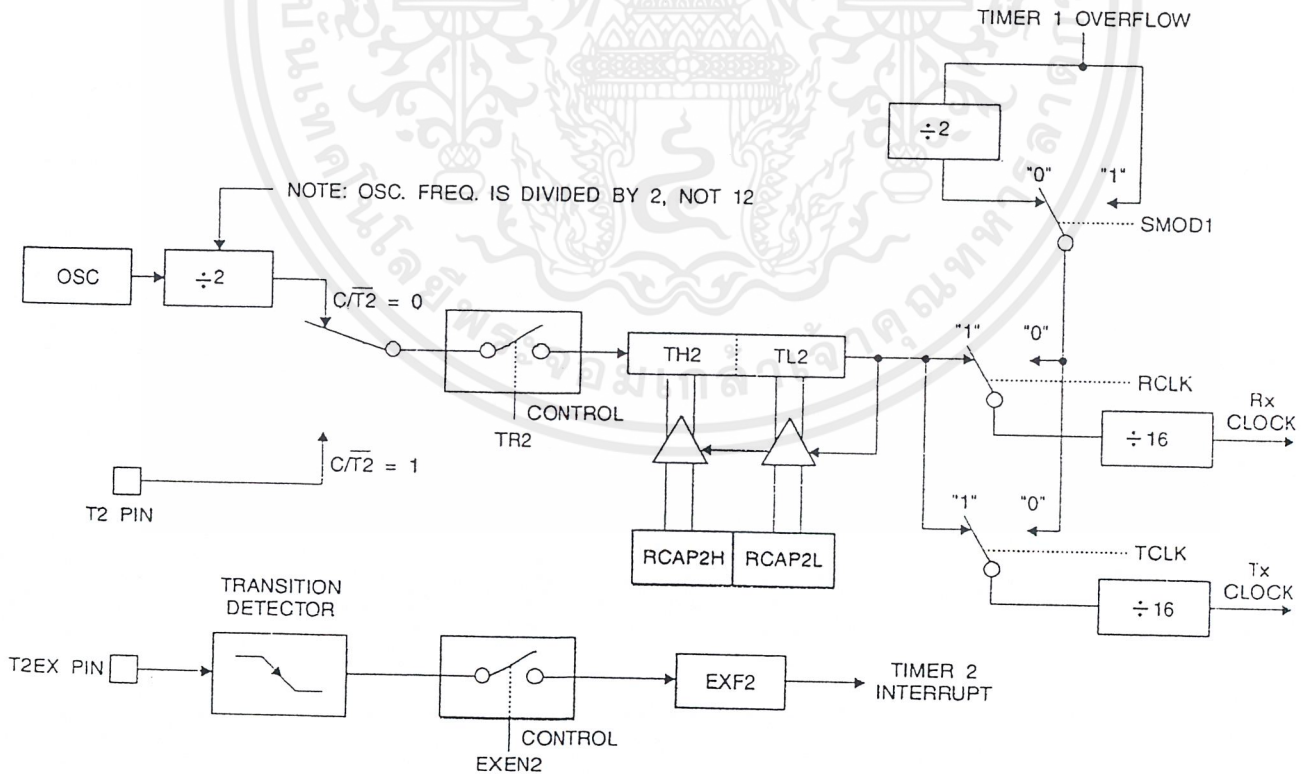


Figure 4. Timer 2 in Baud Rate Generator Mode



Baud Rate Generator

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

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

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

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

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

increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

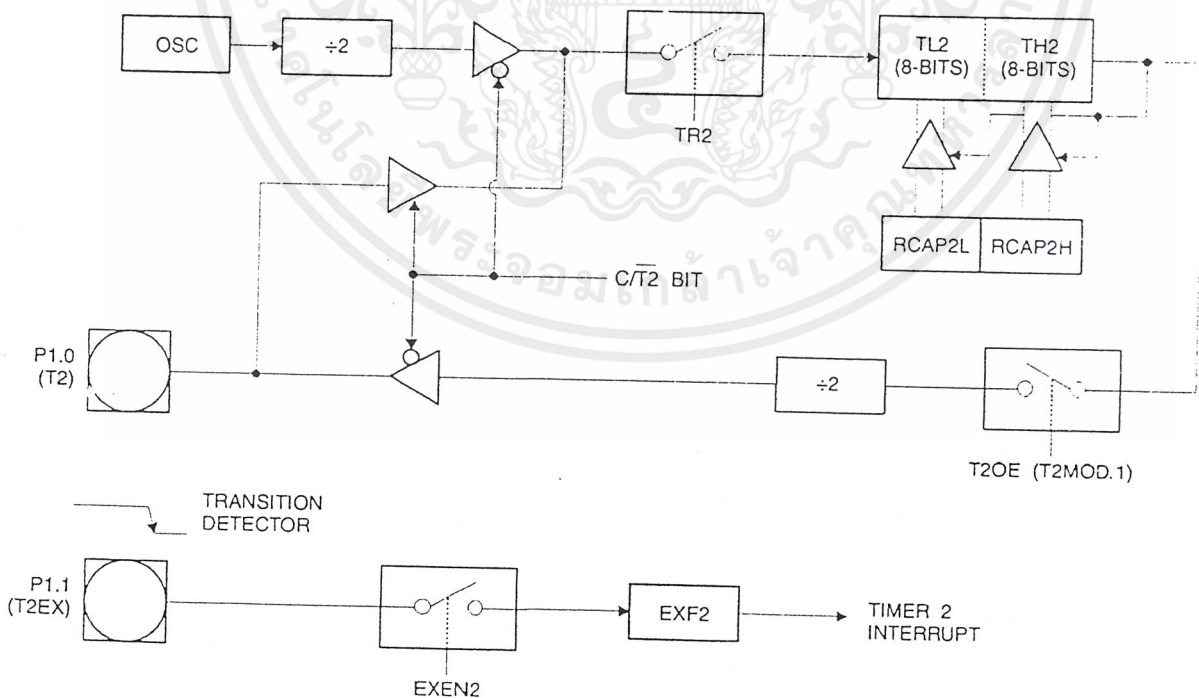
$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]}$$

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

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

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

Figure 5. Timer 2 in Clock-out Mode



Programmable Clock Out

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

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

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

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

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

UART

The UART in the AT89C52 operates the same way as the UART in the AT89C51.

Interrupts

The AT89C52 has a total of six interrupt vectors: two external interrupts (INT0 and INT1), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 6.

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

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

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

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However,

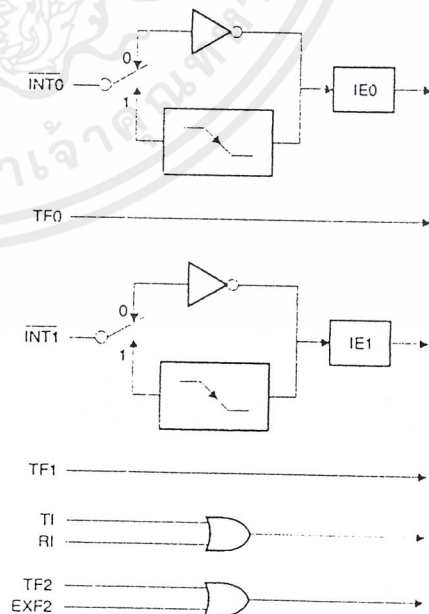
the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

Table 5. Interrupt Enable (IE) Register

(MSB)								(LSB)
EA	-	ET2	ES	ET1	EX1	ET0	EX0	
Enable Bit = 1 enables the interrupt.								
Enable Bit = 0 disables the interrupt.								

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
-	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	Serial Port interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.
User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.		

Figure 6. Interrupt Sources





Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 7. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 8. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

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

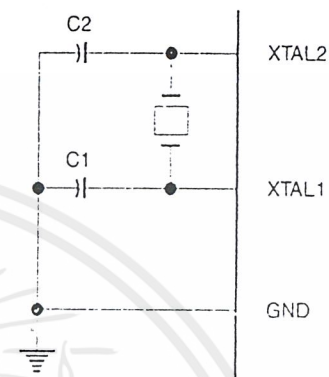
Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

Power-down Mode

In the power-down mode, the oscillator is stopped, and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power-down mode is terminated. The only exit from power-down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC}

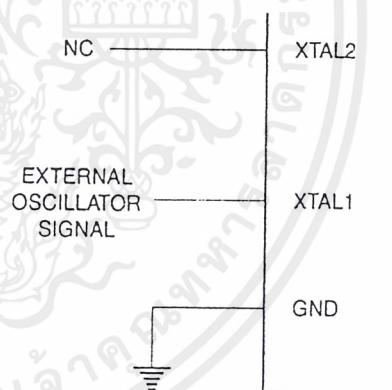
is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Figure 7. Oscillator Connections



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

Figure 8. External Clock Drive Configuration



Status of External Pins During Idle and Power-down Modes

Mode	Program Memory	ALE	$\overline{\text{PSEN}}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

Program Memory Lock Bits

The AT89C52 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

	V _{PP} = 12V	V _{PP} = 5V
Signature	(030H) = 1EH (031H) = 52H (032H) = FFH	(030H) = 1EH (031H) = 52H (032H) = 95H

Lock Bit Protection Modes

Program Lock Bits				Protection Type
LB1	LB2	LB3		
1	U	U	U	No program lock features.
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, \overline{EA} is sampled and latched on reset, and further programming of the Flash memory is disabled.
3	P	P	U	Same as mode 2, but verify is also disabled.
4	P	P	P	Same as mode 3, but external execution is also disabled.

When lock bit 1 is programmed, the logic level at the \overline{EA} pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value and holds that value until reset is activated. The latched value of \overline{EA} must agree with the current logic level at that pin in order for the device to function properly.

Programming the Flash

The AT89C52 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The Low-voltage programming mode provides a convenient way to program the AT89C52 inside the user's system, while the high-voltage programming mode is compatible with conventional third-party Flash or EPROM programmers.

The AT89C52 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	V _{PP} = 12V	V _{PP} = 5V
Top-side Mark	AT89C52 xxxx yyww	AT89C52 xxxx - 5 yyww

The AT89C52 code memory array is programmed byte-by-byte in either programming mode. To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.

Programming Algorithm Before programming the AT89C52, the address, data and control signals should be set up according to the Flash programming mode table and Figure 9 and Figure 10. To program the AT89C52, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise \overline{EA}/V_{PP} to 12V for the high-voltage programming mode.
5. Pulse ALE/ \overline{PROG} once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

Data Polling The AT89C52 features Data Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on PO.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

Ready/Busy The progress of byte programming can also be monitored by the RDY/ \overline{BSY} output signal. P3.4 is pulled low after ALE goes high during programming to indicate \overline{BUSY} . P3.4 is pulled high again when programming is done to indicate READY.

Program Verify If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/ \overline{PROG} low for 10 ms. The code array is written with all 1s. The chip erase operation must be executed before the code memory can be reprogrammed.





Reading the Signature Bytes The signature bytes are read by the same procedure as a normal verification of locations 030H, 031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

- (030H) = 1EH indicates manufactured by Atmel
- (031H) = 52H indicates 89C52
- (032H) = FFH indicates 12V programming
- (032H) = 05H indicates 5V programming

Programming Interface

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

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

Flash Programming Modes

Mode	RST	PSEN	ALE/PROG	EA/V _{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data	H	L		H/12V	L	H	H	H
Read Code Data	H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H	L		H/12V	H	H	H
	Bit - 2	H	L		H/12V	H	H	L
	Bit - 3	H	L		H/12V	H	L	H
Chip Erase	H	L	(1)	H/12V	H	L	L	L
Read Signature Byte	H	L	H	H	L	L	L	L

Note: 1. Chip Erase requires a 10 ms PROG pulse.

Figure 9. Programming the Flash Memory

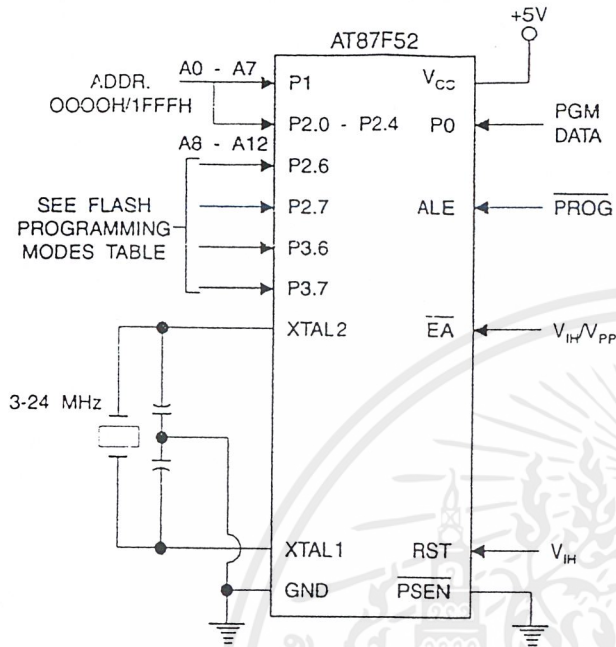
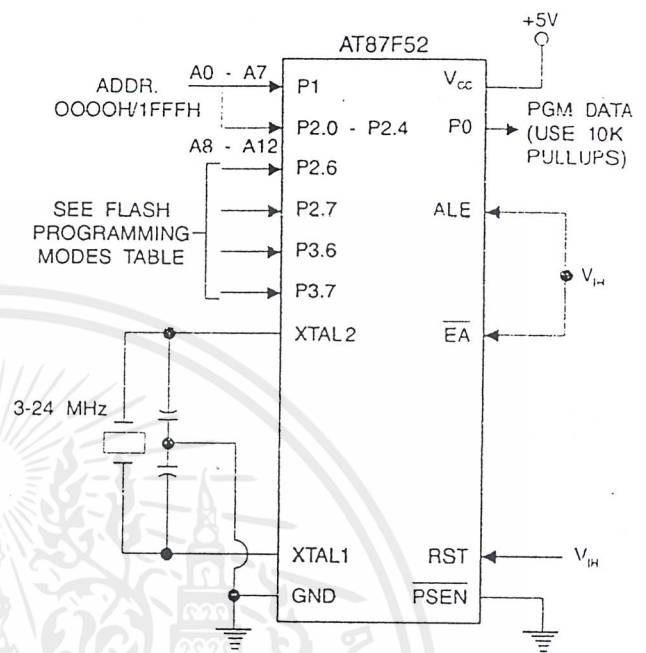


Figure 10. Verifying the Flash Memory



Flash Programming and Verification Characteristics

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

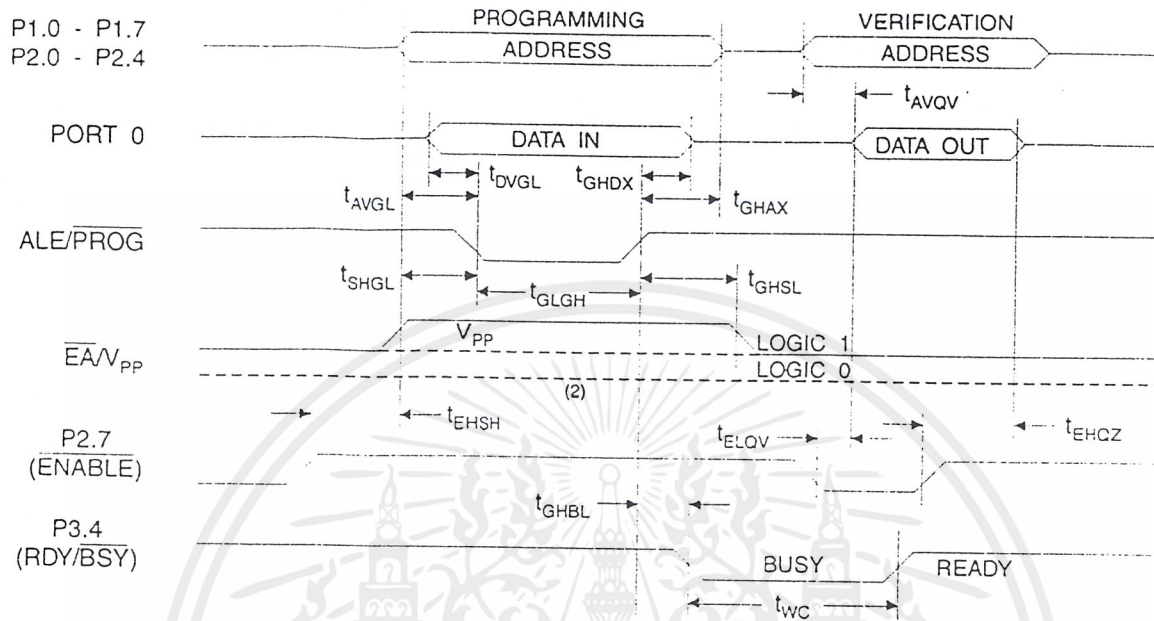
Symbol	Parameter	Min	Max	Units
$V_{PP}^{(1)}$	Programming Enable Voltage	11.5	12.5	V
$I_{PP}^{(1)}$	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	3	24	MHz
t_{AVGL}	Address Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold after $\overline{\text{PROG}}$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

Note: 1. Only used in 12-volt programming mode.

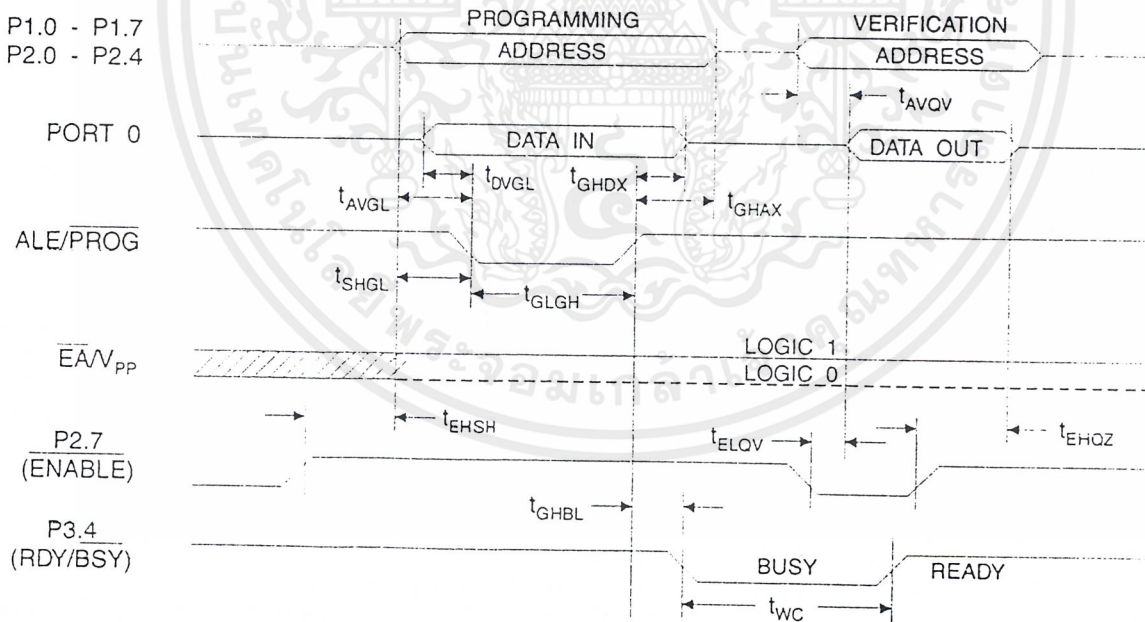




Flash Programming and Verification Waveforms - High-voltage Mode ($V_{PP}=12V$)



Flash Programming and Verification Waveforms - Low-voltage Mode ($V_{PP}=5V$)



Absolute Maximum Ratings*

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

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

DC Characteristics

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

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

- Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
 Maximum I_{OL} per port pin: 10 mA
 Maximum I_{OL} per 8-bit port:
 Port 0: 26 mA Ports 1, 2, 3: 15 mA
 Maximum total I_{OL} for all output pins: 71 mA
 If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power-down is 2V.





AC Characteristics

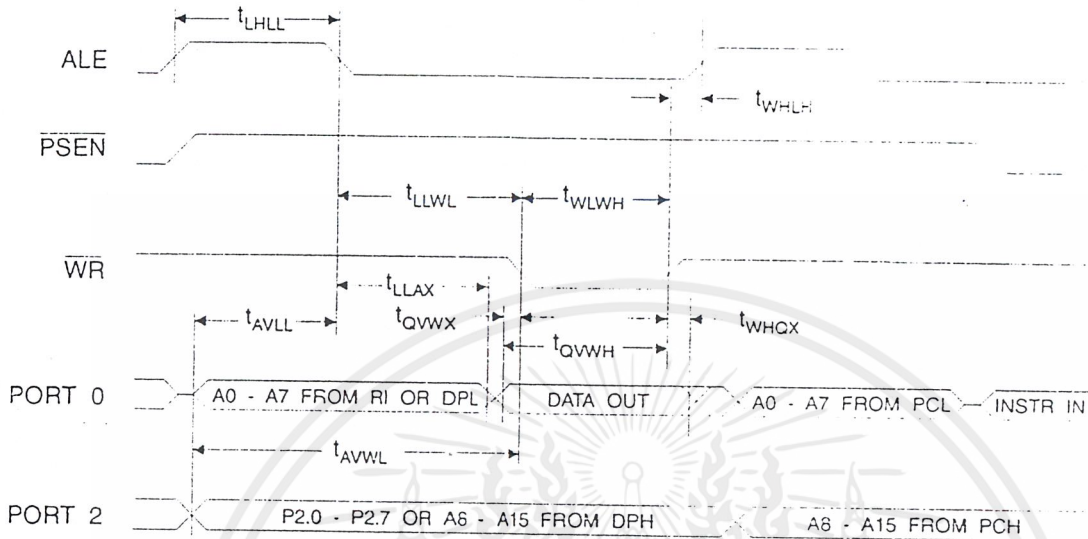
Under operating conditions, load capacitance for Port 0, ALE/PROG, and PSEN = 100 pF; load capacitance for all other outputs = 80 pF.

External Program and Data Memory Characteristics

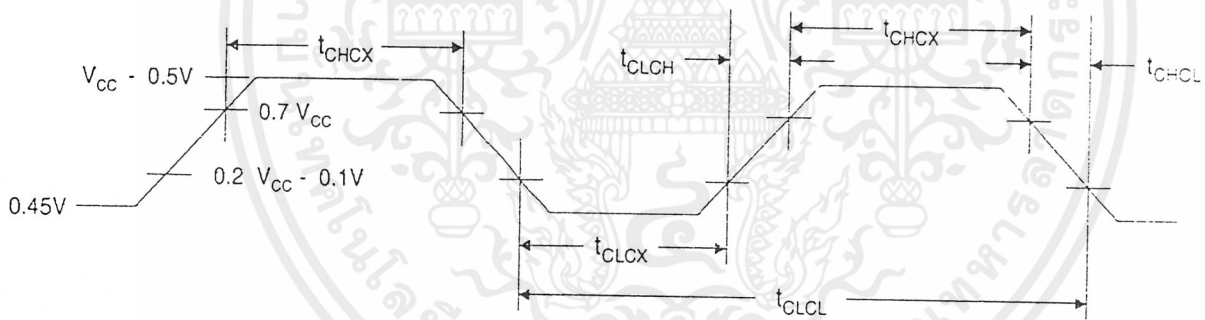
Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHL}	ALE Pulse Width	127		$2t_{CLCL}-40$		ns
t_{AVLL}	Address Valid to ALE Low	43		$t_{CLCL}-13$		ns
t_{LLAX}	Address Hold After ALE Low	48		$t_{CLCL}-20$		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL}-65$	ns
t_{LLPL}	ALE Low to PSEN Low	43		$t_{CLCL}-13$		ns
t_{PLPH}	PSEN Pulse Width	205		$3t_{CLCL}-20$		ns
t_{PLIV}	PSEN Low to Valid Instruction In		145		$3t_{CLCL}-45$	ns
t_{PXIX}	Input Instruction Hold after PSEN	0		0		ns
t_{PXIZ}	Input Instruction Float after PSEN		59		$t_{CLCL}-10$	ns
t_{PXAV}	PSEN to Address Valid	75		$t_{CLCL}-8$		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}-55$	ns
t_{PLAZ}	PSEN Low to Address Float		10		10	ns
t_{RLRH}	RD Pulse Width	400		$6t_{CLCL}-100$		ns
t_{WLWH}	WR Pulse Width	400		$6t_{CLCL}-100$		ns
t_{RLDV}	RD Low to Valid Data In		252		$5t_{CLCL}-90$	ns
t_{RHDZ}	Data Hold After RD	0		0		ns
t_{RHDZ}	Data Float After RD		97		$2t_{CLCL}-28$	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CLCL}-150$	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CLCL}-165$	ns
t_{LLWL}	ALE Low to RD or WR Low	200	300	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
t_{AVWL}	Address to RD or WR Low	203		$4t_{CLCL}-75$		ns
t_{QVWX}	Data Valid to WR Transition	23		$t_{CLCL}-20$		ns
t_{QVWH}	Data Valid to WR High	433		$7t_{CLCL}-120$		ns
t_{WHQX}	Data Hold After WR	33		$t_{CLCL}-20$		ns
t_{RLAZ}	RD Low to Address Float		0		0	ns
t_{WHLH}	RD or WR High to ALE High	43	123	$t_{CLCL}-20$	$t_{CLCL}+25$	ns



External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
t_{CLCL}	Clock Period	41.6		ns
t_{CHCX}	High Time	15		ns
t_{CLCX}	Low Time	15		ns
t_{CLCH}	Rise Time		20	ns
t_{CHCL}	Fall Time		20	ns



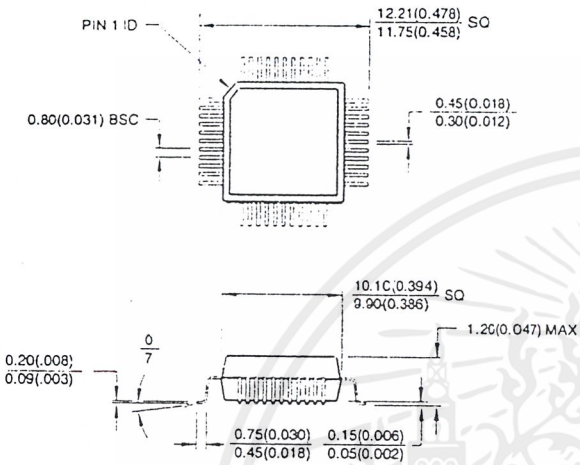
Ordering Information

Speed (MHz)	Power Supply	Code	Package	Operation Range				
12	5V ± 20%	AC	44A	Commercial (0°C to 70°C)				
		JC	44J					
		PC	40P6					
		QC	44Q					
		AI	II	PI	QI	Industrial (-40°C to 85°C)		
							AI	44A
							II	44J
							PI	40P6
16	5V ± 20%	AC	44A	Commercial (0°C to 70°C)				
		JC	44J					
		PC	40P6					
		QC	44Q					
		AI	II	PI	QI	Industrial (-40°C to 85°C)		
							AI	44A
							II	44J
							PI	40P6
20	5V ± 20%	AC	44A	Commercial (0°C to 70°C)				
		JC	44J					
		PC	40P6					
		QC	44Q					
		AI	II	PI	QI	Industrial (-40°C to 85°C)		
							AI	44A
							II	44J
							PI	40P6
24	5V ± 20%	JC	44A	Commercial (0°C to 70°C)				
		C	44J					
		JC	40P6					
		C	44Q					
		II	II	II	II	Industrial (-40°C to 85°C)		
							II	44A
							II	44J
							II	40P6
			44Q					

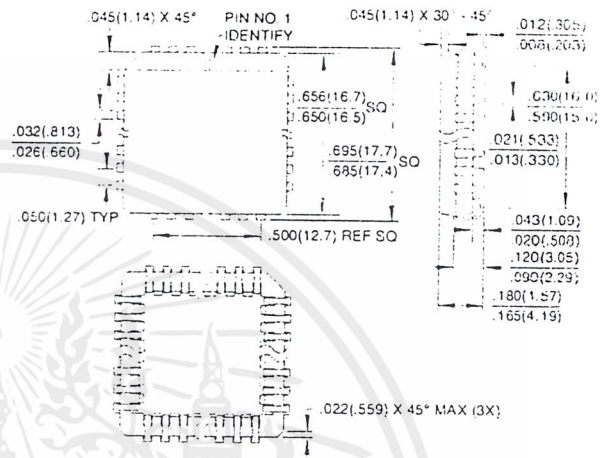
Package Type	
44A	44-lead, Thin Plastic Quad Flatpack (TQFP)
44J	44-lead, Plastic J-Leader (PLCC)
40P6	40-lead, 0.600" Wide In-Line Package (PDIP)
44Q	44-lead, Plastic Quad Flatpack (PQFP)

Packaging Information

44A, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flatpack (TQFP)
 Dimensions in Millimeters and (Inches)*
 JEDEC STANDARD MS-026 ACB

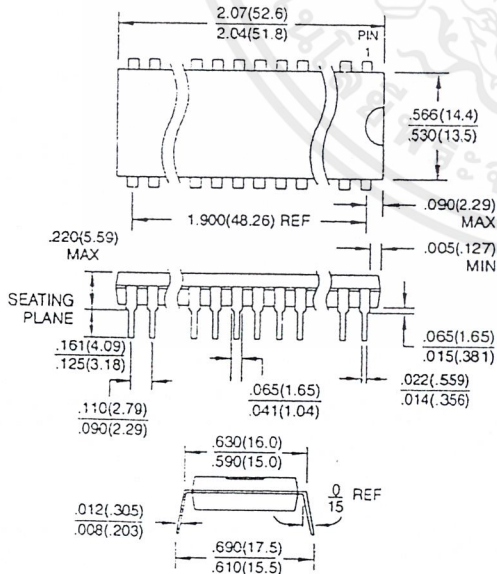


44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)
 Dimensions in Inches and (Millimeters)
 JEDEC STANDARD MS-018 AC

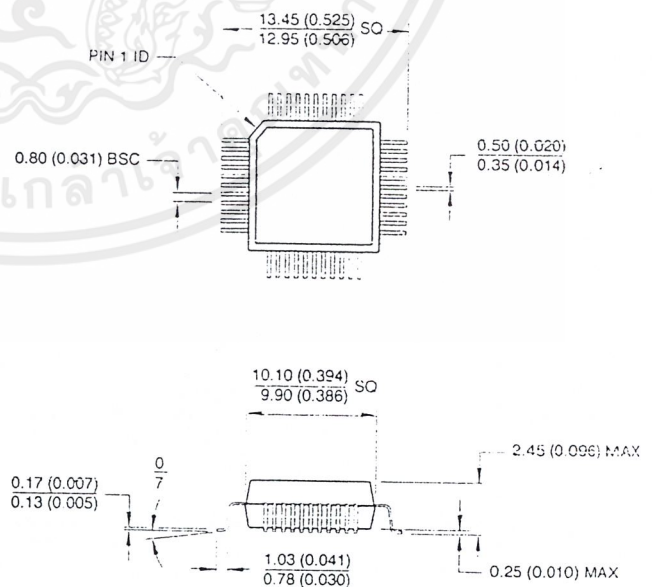


Controlling dimension: millimeters

40P6, 40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
 Dimensions in Inches and (Millimeters)



44Q, 44-lead, Plastic Quad Flat Package (PQFP)
 Dimensions in Millimeters and (Inches)*
 JEDEC STANDARD MS-022 AB



Controlling dimension: millimeters





Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe

Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686-677
FAX (44) 1276-686-697

Asia

Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan

Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset

Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Fax-on-Demand

North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail

literature@atmel.com

Web Site

<http://www.atmel.com>

BBS

1-(408) 436-4309

© Atmel Corporation 1999.

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Marks bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

Terms and product names in this document may be trademarks of others.



Printed on recycled paper.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใด 0313H-02/00x14

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LMD18245 3A, 55V DMOS Full-Bridge Motor Driver

General Description

The LMD18245 full-bridge power amplifier incorporates all the circuit blocks required to drive and control current in a brushed type DC motor or one phase of a bipolar stepper motor. The multi-technology process used to build the device combines bipolar and CMOS control and protection circuitry with DMOS power switches on the same monolithic structure. The LMD18245 controls the motor current via a fixed off-time chopper technique.

An all DMOS H-bridge power stage delivers continuous output currents up to 3A (6A peak) at supply voltages up to 55V. The DMOS power switches feature low $R_{DS(ON)}$ for high efficiency, and a diode intrinsic to the DMOS body structure eliminates the discrete diodes typically required to clamp bipolar power stages.

An innovative current sensing method eliminates the power loss associated with a sense resistor in series with the motor. A four-bit digital-to-analog converter (DAC) provides a digital path for controlling the motor current, and, by extension, simplifies implementation of full, half and microstep stepper motor drives. For higher resolution applications, an external DAC can be used.

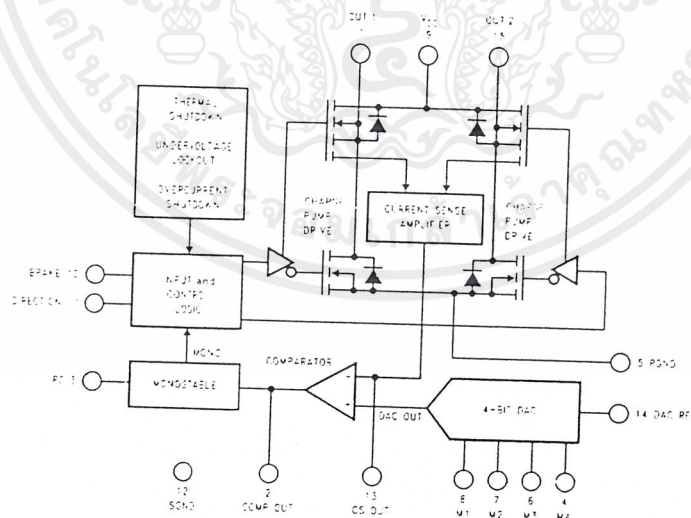
Features

- DMOS power stage rated at 55V and 3A continuous
- Low $R_{DS(ON)}$ of typically 0.3 Ω per power switch
- Internal clamp diodes
- Low-loss current sensing method
- Digital or analog control of motor current
- TTL and CMOS compatible inputs
- Thermal shutdown (outputs off) at $T_J = 155\text{ C}$
- Overcurrent protection
- No shoot-through currents
- 15-lead TO-220 molded power package

Applications

- Full, half and microstep stepper motor drives
- Stepper motor and brushed DC motor servo drives
- Automated factory, medical and office equipment

Functional Block and Connection Diagram (15-Lead TO-220 Molded Power Package (T))



Order Number LMD18245T
See NS Package Number TA15A

DS011878-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings (Note 1)

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

DC Voltage at:	
OUT 1, V _{CC} , and OUT 2	+60V
COMP OUT, RC, M4, M3, M2, M1, BRAKE, DIRECTION, CS OUT, and DAC REF	+12V
DC Voltage PGND to SGND	±400mV
Continuous Load Current	3A
Peak Load Current (Note 2)	6A
Junction Temperature (T _{J(max)})	+150°C

Power Dissipation (Note 3):

TO-220 (T _A = 25°C, Infinite Heatsink)	3.5W
TO-220 (T _A = 25°C, Free Air)	1500V
ESD Susceptibility (Note 4)	-40°C to +150°C
Storage Temperature Range (T _S)	300°C
Lead Temperature (Soldering, 10 seconds)	

Operating Conditions (Note 1)

Temperature Range (T _J) (Note 3)	-10°C to +125°C
Supply Voltage Range (V _{CC})	+12V to +55V
CS OUT Voltage Range	0V to +5V
DAC REF Voltage Range	0V to +5V
MONOSTABLE Pulse Range	10 μs to 100 ms

Electrical Characteristics (Note 2)

The following specifications apply for V_{CC} = +42V, unless otherwise stated. Boldface limits apply over the operating temperature range, -40°C ≤ T_J ≤ +125°C. All other limits apply for T_A = T_J = 25°C.

Symbol	Parameter	Conditions	Typical (Note 5)	Limit (Note 5)	Units (Limits)
I _{CC}	Quiescent Supply Current	DAC REF = 0V, V _{CC} = +20V	8	15	mA (max)
POWER OUTPUT STAGE					
R _{DS(ON)}	Switch ON Resistance	I _{LOAD} = 3A	0.3	0.4	Ω (max)
		I _{LOAD} = 6A	0.3	0.4	Ω (max)
V _{DIODE}	Body Diode Forward Voltage	I _{DIODE} = 3A	1.0	0.6	V (max)
T _{rr}	Diode Reverse Recovery Time	I _{DIODE} = 1A	80	1.5	ns
Q _{rr}	Diode Reverse Recovery Charge	I _{DIODE} = 1A	40		nC
t _{D(ON)}	Output Turn ON Delay Time				
	Sourcing Outputs	I _{LOAD} = 3A	5		μs
	Sinking Outputs	I _{LOAD} = 3A	900		ns
t _{D(OFF)}	Output Turn OFF Delay Time				
	Sourcing Outputs	I _{LOAD} = 3A	600		ns
	Sinking Outputs	I _{LOAD} = 3A	400		ns
t _{ON}	Output Turn ON Switching Time				
	Sourcing Outputs	I _{LOAD} = 3A	40		μs
	Sinking Outputs	I _{LOAD} = 3A	1		μs
t _{OFF}	Output Turn OFF Switching Time				
	Sourcing Outputs	I _{LOAD} = 3A	200		ns
	Sinking Outputs	I _{LOAD} = 3A	80		ns
t _{pw}	Minimum Input Pulse Width	Pins 10 and 11	2		μs
t _{DB}	Minimum Dead Band	(Note 6)	40		ns
CURRENT SENSE AMPLIFIER					
	Current Sense Output	I _{LOAD} = 1A (Note 7)	250	200 175 300 325	μA (min) μA (min) μA (max) μA (max)
	Current Sense Linearity Error	0.5A ≤ I _{LOAD} ≤ 3A (Note 7)	±6	±9	% (max)
	Current Sense Offset	I _{LOAD} = 0A	5		μA (max)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Characteristics (Note 2) (Continued)

The following specifications apply for $V_{CC} = +42V$, unless otherwise stated. Boldface limits apply over the operating temperature range, $-40^{\circ}C \leq T_J \leq +125^{\circ}C$. All other limits apply for $T_A = T_J = 25^{\circ}C$.

Symbol	Parameter	Conditions	Typical (Note 5)	Limit (Note 5)	Units (Limits)
DIGITAL-TO-ANALOG CONVERTER (DAC)					
	Resolution			4	Bits (min)
	Monotonicity			4	Bits (min)
	Total Unadjusted Error		0.125	0.25 0.5	LSB (rmax) LSB (max)
	Propagation Delay		50		ns
I_{REF}	DAC REF Input Current	DAC REF = +5V	-0.5	± 10	μA μA (max)
COMPARATOR AND MONOSTABLE					
	Comparator High Output Level ¹		6.27		V
	Comparator Low Output Level ¹		88		mV
	Comparator Output Current				
	Source		0.2		mA
	Sink		3.2		mA
t_{DELAY}	Monostable Turn OFF Delay	(Note 8)	1.2	2.0	μs μs (max)
PROTECTION AND PACKAGE THERMAL RESISTANCES					
	Undervoltage Lockout, V_{CC}			5 8	V (min) V (max)
T_{JSD}	Shutdown Temperature, T_J		155		$^{\circ}C$
θ_{JC}	Package Thermal Resistances				
	Junction-to-Case, TO-220		1.5		C/W
θ_{JA}	Junction-to-Ambient, TO-220		35		C/W
LOGIC INPUTS					
V_{IL}	Low Level Input Voltage			-0.1 0.8	V (min) V (max)
V_{IH}	High Level Input Voltage			2 12	V (min) V (max)
I_{IN}	Input Current	$V_{IN} = 0V$ or 12V		± 10	μA (max)

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Electrical specifications do not apply when operating the device outside the rated Operating Conditions.

Note 2: Unless otherwise stated, load currents are pulses with widths less than 2 ms and duty cycles less than 5%.

Note 3: The maximum allowable power dissipation at any ambient temperature is $P_{MAX} = (125 - T_A)/\theta_{JA}$, where 125 C is the maximum junction temperature for operation. T_A is the ambient temperature in $^{\circ}C$, and θ_{JA} is the junction-to-ambient thermal resistance in $^{\circ}C/W$. Exceeding P_{MAX} voids the Electrical Specifications by forcing T_J above 125 C. If the junction temperature exceeds 155 C, internal circuitry disables the power bridge. When a heatsink is used, θ_{JA} is the sum of the junction-to-case thermal resistance of the package, θ_{JC} , and the case-to-ambient thermal resistance of the heatsink.

Note 4: ESD rating is based on the human body model of 100 pF discharged through a 1.5 k Ω resistor. M1, M2, M3 and M4, pins 8, 7, 6 and 4 are protected to 800V.

Note 5: All limits are 100% production tested at 25 C. Temperature extreme limits are guaranteed via correlation using accepted SQC (Statistical Quality Control) methods. All limits are used to calculate AOQL (Average Outgoing Quality Level). Typical values are at $T_J = 25^{\circ}C$ and represent the most likely parametric norm.

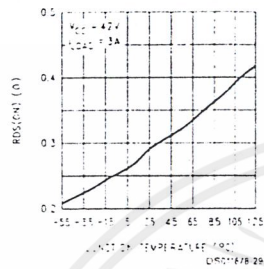
Note 6: Asymmetric turn OFF and ON delay times and switching times ensure a switch turns OFF before the other switch in the same half H-bridge begins to turn ON (preventing momentary short circuits between the power supply and ground). The transitional period during which both switches are OFF is commonly referred to as the dead band.

Note 7: (I_{LOAD} , I_{SENSE}) data points are taken for load currents of 0.5A, 1A, 2A and 3A. The current sense gain is specified as I_{SENSE}/I_{LOAD} for the 1A data point. The current sense linearity is specified as the slope of the line between the 0.5A and 1A data points minus the slope of the line between the 2A and 3A data points, all divided by the slope of the line between the 0.5A and 1A data points.

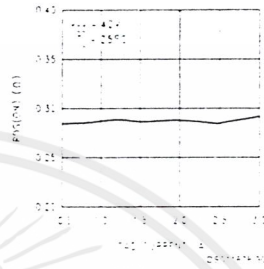
Note 8: Turn OFF delay, t_{DELAY} , is defined as the time from the voltage at the output of the current sense amplifier reaching the DAC output voltage to the lower DMOS switch beginning to turn OFF. With $V_{CC} = 32V$, DIRECTION high, and 200 Ω connected between OUT1 and V_{CC} , the voltage at RC is increased from 0V to 5V at 1.2V/ μs and t_{DELAY} is measured as the time from the voltage at RC reaching 2V to the time the voltage at OUT 1 reaches 3V.

Typical Performance Characteristics

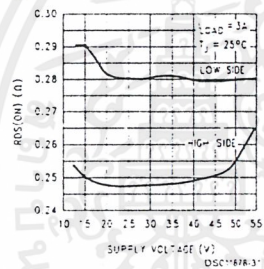
RDS(ON) vs Temperature



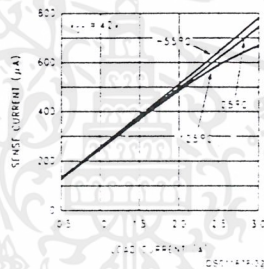
RDS(ON) vs Load Current



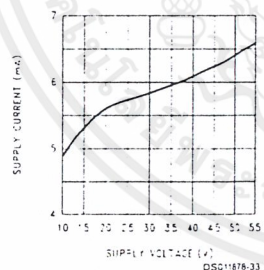
RDS(ON) vs Supply Voltage



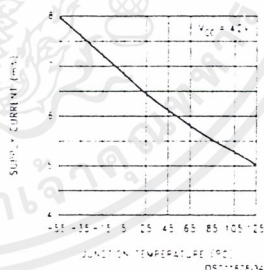
Current Sense Output vs Load Current



Supply Current vs Supply Voltage

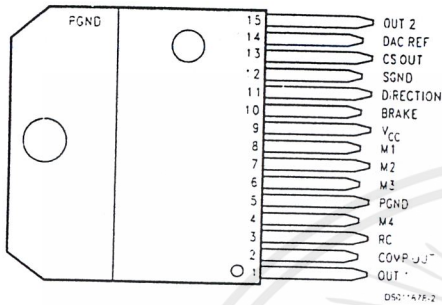


Supply Current vs Temperature



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Connection Diagram



Top View
15-Lead TO-220 Molded Power Package
Order Number LMD18245T
See NS Package Number TA15A

Pinout Descriptions (See Functional Block and Connection Diagrams)

- Pin 1, OUT 1:** Output node of the first half H-bridge.
- Pin 2, COMP OUT:** Output of the comparator. If the voltage at CS OUT exceeds that provided by the DAC, the comparator triggers the monostable.
- Pin 3, RC:** Monostable timing node. A parallel resistor-capacitor network connected between this node and ground sets the monostable timing pulse at about 1.1 RC seconds.
- Pin 5, PGND:** Ground return node of the power bridge. Bond wires (internal) connect PGND to the tab of the TO-220 package.
- Pins 4 and 6 through 8, M4 through M1:** Digital inputs of the DAC. These inputs make up a four-bit binary number with M4 as the most significant bit or MSB. The DAC provides an analog voltage directly proportional to the binary number applied at M4 through M1.
- Pin 9, V_{CC}:** Power supply node.
- Pin 10, BRAKE:** Brake logic input. Pulling the BRAKE input logic-high activates both sourcing switches of the power bridge—effectively shorting the load. See *Table 1*. Shorting the load in this manner forces the load current to recirculate and decay to zero.
- Pin 11, DIRECTION:** Direction logic input. The logic level at this input dictates the direction of current flow in the load. See *Table 1*.
- Pin 12, SGND:** Ground return node of all signal level circuits.

Pin 13, CS OUT: Output of the current sense amplifier. The current sense amplifier sources 250 μ A (typical) per ampere of total forward current conducted by the upper two switches of the power bridge.

Pin 14, DAC REF: Voltage reference input of the DAC. The DAC provides an analog voltage equal to $V_{DAC\ REF} \times D/16$, where D is the decimal equivalent (0–15) of the binary number applied at M4 through M1.

Pin 15, OUT 2: Output node of the second half H-bridge.

TABLE 1. Switch Control Logic Truth Table

BRAKE	DIRECTION	MONO	Active Switches
H	X	X	Source 1, Source 2
L	H	L	Source 2
L	H	H	Source 2, Sink 1
L	L	L	Source 1
L	L	H	Source 1, Sink 2

X = don't care

MONO is the output of the monostable.

Functional Descriptions

TYPICAL OPERATION OF A CHOPPER AMPLIFIER

Chopper amplifiers employ feedback driven switching of a power bridge to control and limit current in the winding of a motor (*Figure 1*). The bridge consists of four solid state power switches and four diodes connected in an H configuration. Control circuitry (not shown) monitors the winding current and compares it to a threshold. While the winding current remains less than the threshold, a source switch and a sink switch in opposite halves of the bridge force the supply voltage across the winding, and the winding current increases rapidly towards V_{CC}/R (*Figure 1a* and *Figure 1d*). As the winding current surpasses the threshold, the control circuitry turns OFF the sink switch for a fixed period or *off-time*. During the off-time, the source switch and the opposite upper diode short the winding, and the winding current recirculates and decays slowly towards zero (*Figure 1b* and *Figure 1e*). At the end of the off-time, the control circuitry turns back ON the sink switch, and the winding current again increases rapidly towards V_{CC}/R (*Figure 1a* and *Figure 1d* again). The above sequence repeats to provide a current chopping action that limits the winding current to the threshold (*Figure 1g*). Chopping only occurs if the winding current reaches the threshold. During a change in the direction of the winding current, the diodes provide a decay path for the initial winding current (*Figure 1c* and *Figure 1f*). Since the bridge shorts the winding for a fixed period, this type of chopper amplifier is commonly referred to as a *fixed off-time chopper*.

Functional Descriptions (Continued)

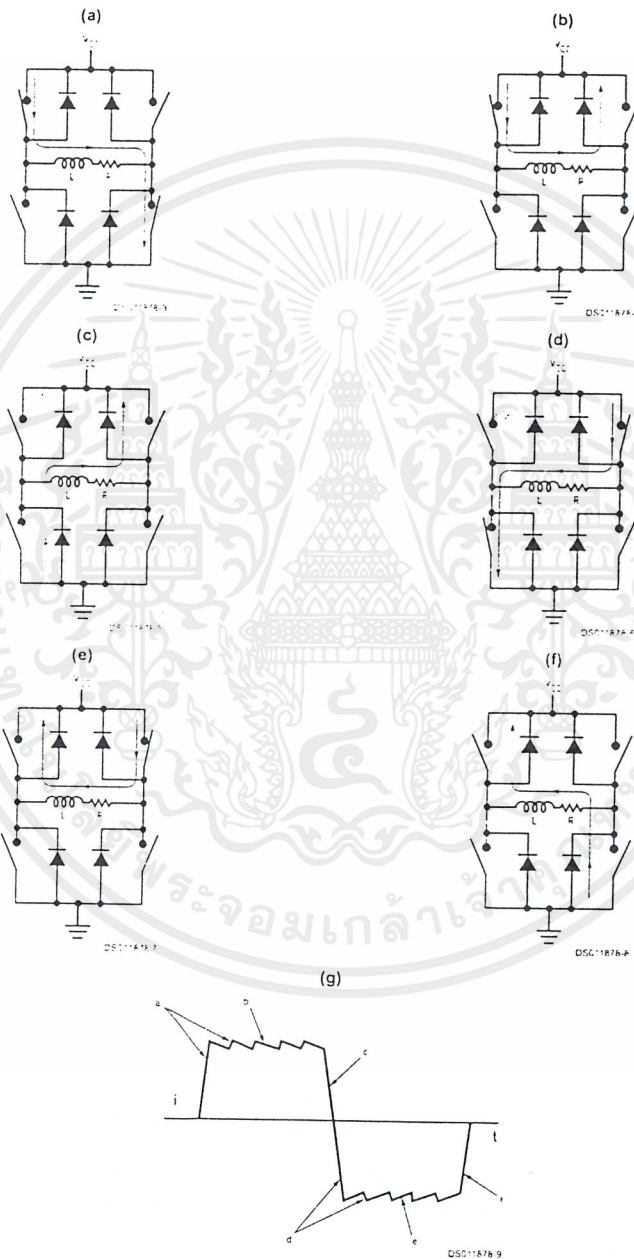


FIGURE 1. Chopper Amplifier Chopping States: Full V_{CC} Applied Across the Winding (a) and (d), Shorted Winding (b) and (e), Winding Current Decays During a Change in the Direction of the Winding Current (c) and (f), and the Chopped Winding Current (g)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Descriptions (Continued)

THE LMD18245 CHOPPER AMPLIFIER

The LMD18245 incorporates all the circuit blocks needed to implement a fixed off-time chopper amplifier. These blocks include: an all DMOS, full H-bridge with clamp diodes, an amplifier for sensing the load current, a comparator, a monostable, and a DAC for digital control of the chopping threshold. Also incorporated are logic, level shifting and drive blocks for digital control of the direction of the load current and braking.

THE H-BRIDGE

The power stage consists of four DMOS power switches and associated body diodes connected in an H-bridge configuration (Figure 2). Turning ON a source switch and a sink

switch in opposite halves of the bridge forces the full supply voltage less the switch drops across the motor winding. While the bridge remains in this state, the winding current increases exponentially towards a limit dictated by the supply voltage, the switch drops, and the winding resistance. Subsequently turning OFF the sink switch causes a voltage transient that forward biases the body diode of the other source switch. The diode clamps the transient at one diode drop above the supply voltage and provides an alternative current path. While the bridge remains in this state, it essentially shorts the winding and the winding current recirculates and decays exponentially towards zero. During a change in the direction of the winding current, both the switches and the body diodes provide a decay path for the initial winding current (Figure 3)

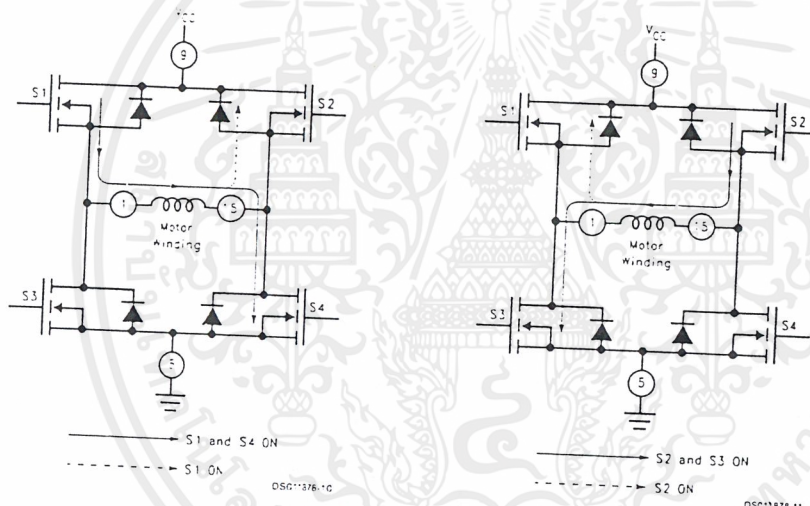


FIGURE 2. The DMOS H-Bridge

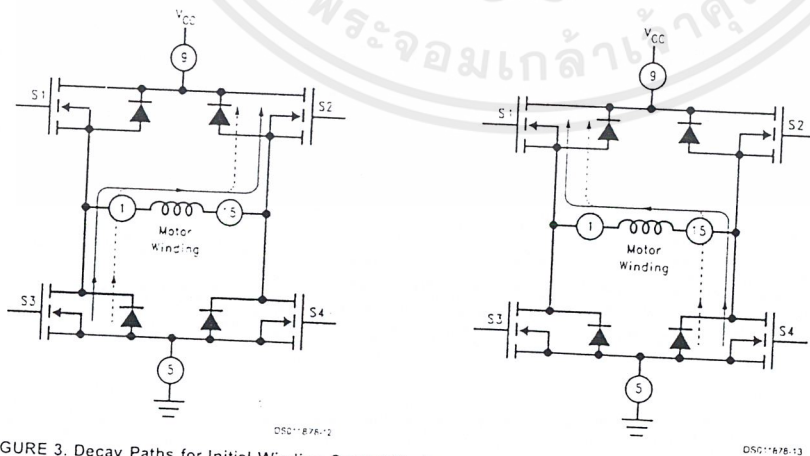


FIGURE 3. Decay Paths for Initial Winding Current During a Change in the Direction of the Winding Current

Functional Descriptions (Continued)

THE CURRENT SENSE AMPLIFIER

Many transistor cells in parallel make up the DMOS power switches. The current sense amplifier (Figure 4) uses a small fraction of the cells of both upper switches to provide a unique, low-loss means for sensing the load current. In practice, each upper switch functions as a 1x sense device in parallel with a 4000x power device. The current sense amplifier forces the voltage at the source of the sense device to equal that at the source of the power device; thus, the devices share the total drain current in proportion to the 1:4000 cell ratio. Only the current flowing from drain to source, the forward current, registers at the output of the current sense amplifier. The current sense amplifier, therefore, sources 250 μ A per ampere of total forward current conducted by the upper two switches of the power bridge.

The sense current develops a potential across R_S that is proportional to the load current; for example, per ampere of load current, the sense current develops one volt across a 4 k Ω resistor (the product of 250 μ A per ampere and 4 k Ω). Since chopping of the load current occurs as the voltage at CS OUT surpasses the threshold (the DAC output voltage), R_S sets the gain of the chopper amplifier; for example, a 2 k Ω resistor sets the gain at two amperes of load current per volt of the threshold (the reciprocal of the product of 250 μ A per ampere and 2 k Ω). A quarter watt resistor suffices. A low value capacitor connected in parallel with R_S filters the effects of switching noise from the current sense signal.

While the specified maximum DC voltage compliance at CS OUT is 12V, the specified operating voltage range at CS OUT is 0V to 5V.

THE DIGITAL-TO-ANALOG CONVERTER (DAC)

The DAC sets the threshold voltage for chopping at $V_{DAC REF} \times D/16$, where D is the decimal equivalent (0–15) of the binary number applied at M4 through M1, the digital inputs of the DAC. M4 is the MSB or most significant bit. For applications that require higher resolution, an external DAC can drive the DAC REF input. While the specified maximum DC voltage compliance at DAC REF is 12V, the specified operating voltage range at DAC REF is 0V to 5V.

THE COMPARATOR, MONOSTABLE AND WINDING CURRENT THRESHOLD FOR CHOPPING

As the voltage at CS OUT surpasses that at the output of the DAC, the comparator triggers the monostable, and the monostable, once triggered, provides a timing pulse to the control logic. During the timing pulse, the power bridge shorts the motor winding, causing current in the winding to recirculate and decay slowly towards zero (Figure 1b and Figure 1e again). A parallel resistor-capacitor network connected between RC (pin #3) and ground sets the timing pulse or off-time at about 1.1 RC seconds.

Chopping of the winding current occurs as the voltage at CS OUT exceeds that at the output of the DAC; so chopping occurs at a winding current threshold of about

$$(V_{DAC REF} \times D/16) \div ((250 \times 10^{-6}) \times R_S) \text{ amperes.}$$

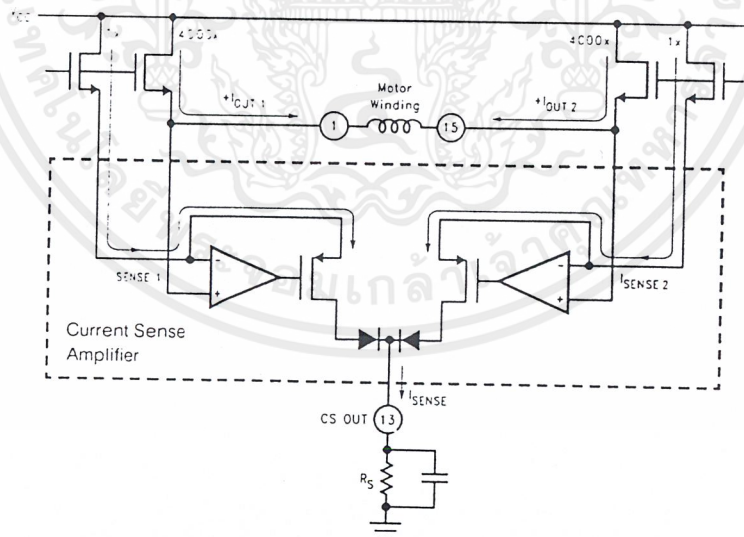


FIGURE 4. The Source Switches of the Power Bridge and the Current Sense Amplifier

DS91187F-1-2

Applications Information

POWER SUPPLY BYPASSING

Step changes in current drawn from the power supply occur repeatedly during normal operation and may cause large voltage spikes across inductance in the power supply line. Care must be taken to limit voltage spikes at V_{CC} to less than the 60V Absolute Maximum Rating. At a change in the direction of the load current, the initial load current tends to raise the voltage at the power supply rail (Figure 3) again. Current transients caused by the reverse recovery of the clamp diodes tend to pull down the voltage at the power supply rail.

Bypassing the power supply line at V_{CC} is required to protect the device and minimize the adverse effects of normal operation on the power supply rail. Using both a 1 μF high frequency ceramic capacitor and a large-value aluminum electrolytic capacitor is highly recommended. A value of 100 μF per ampere of load current usually suffices for the aluminum electrolytic capacitor. Both capacitors should have short leads and be located within one half inch of V_{CC} .

OVERCURRENT PROTECTION

If the forward current in either source switch exceeds a 12A threshold, internal circuitry disables both source switches, forcing a rapid decay of the fault current (Figure 5). Approximately 3 μs after the fault current reaches zero, the device restarts. Automatic restart allows an immediate return to normal operation once the fault condition has been removed. If the fault persists, the device will begin cycling into and out of thermal shutdown. Switching large fault currents may cause potentially destructive voltage spikes across inductance in the power supply line; therefore, the power supply line must be properly bypassed at V_{CC} for the motor driver to survive an extended overcurrent fault.

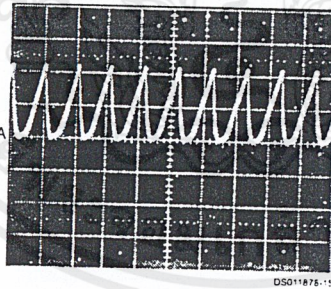
In the case of a locked rotor, the inductance of the winding tends to limit the rate of change of the fault current to a value easily handled by the protection circuitry. In the case of a low inductance short from either output to ground or between outputs, the fault current could surge past the 12A shutdown threshold, forcing the device to dissipate a substantial amount of power for the brief period required to disable the source switches. Because the fault power must be dissipated by only one source switch, a short from output to ground represents the worst case fault. Any overcurrent fault is potentially destructive, especially while operating with high supply voltages ($\geq 30\text{V}$), so precautions are in order. Sinking V_{CC} for heat with 1 square inch of 1 ounce copper on the printed circuit board is highly recommended. The sink switches are not internally protected against shorts to V_{CC} .

THERMAL SHUTDOWN

Internal circuitry senses the junction temperature near the power bridge and disables the bridge if the junction temperature exceeds about 155°C. When the junction temperature cools past the shutdown threshold (lowered by a slight hysteresis), the device automatically restarts.

UNDERVOLTAGE LOCKOUT

Internal circuitry disables the power bridge if the power supply voltage drops below a rough threshold between 8V and 5V. Should the power supply voltage then exceed the threshold, the device automatically restarts.



Trace: Fault Current at 5A/div
Horizontal: 20 $\mu\text{s}/\text{div}$

FIGURE 5. Fault Current with $V_{CC} = 30\text{V}$, OUT 1 Shorted to OUT 2, and CS OUT Grounded

The Typical Application

Figure 6 shows the typical application, the power stage of a chopper drive for bipolar stepper motors. The 20 k Ω resistor and 2.2 nF capacitor connected between RC and ground set the off-time at about 48 μs , and the 20 k Ω resistor connected between CS OUT and ground sets the gain at about 200 mA

per volt of the threshold for chopping. Digital signals control the thresholds for chopping, the directions of the winding currents, and, by extension, the drive type (full step, half step, etc.). A $\mu\text{processor}$ or $\mu\text{controller}$ usually provides the digital control signals.

The Typical Application (Continued)

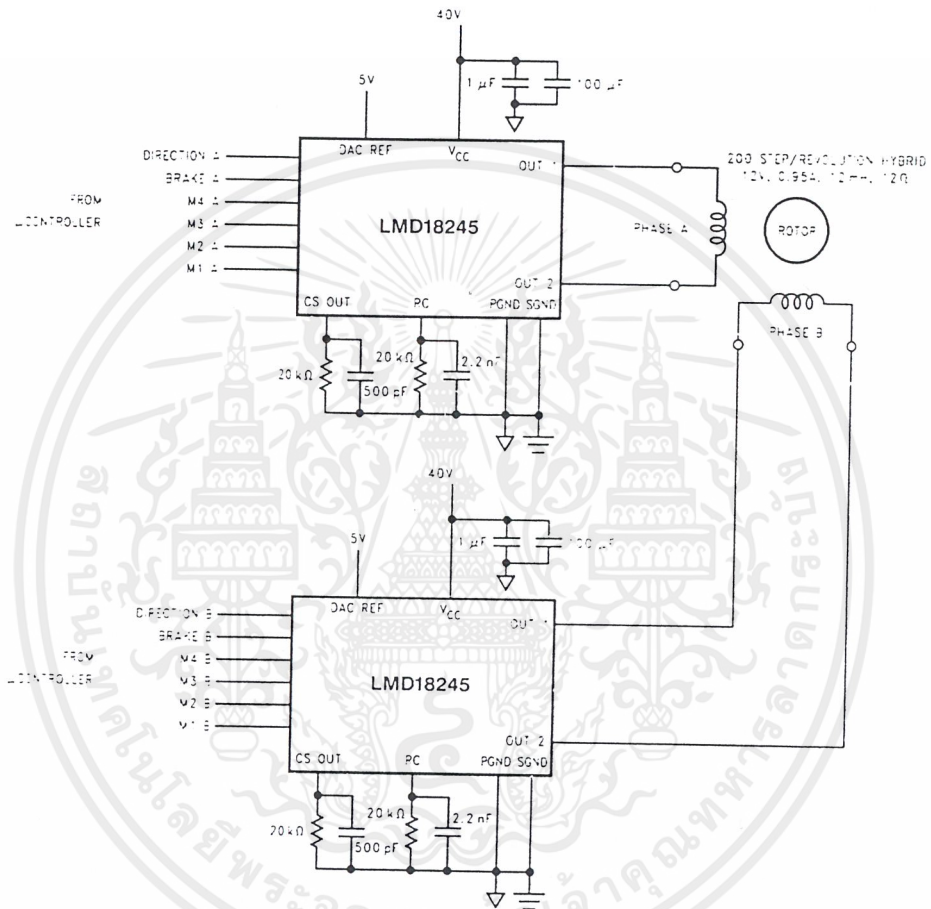


FIGURE 6. Typical Application Circuit for Driving Bipolar Stepper Motors

ONE-PHASE-ON FULL STEP DRIVE (WAVE DRIVE)

To make the motor take full steps, windings A and B can be energized in the sequence

$$A \rightarrow B \rightarrow A^* \rightarrow B^* \rightarrow A \rightarrow \dots$$

where A represents winding A energized with current in one direction and A* represents winding A energized with current in the opposite direction. The motor takes one full step each time one winding is de-energized and the other is energized. To make the motor step in the opposite direction, the order of the above sequence must be reversed. Figure 7 shows the winding currents and digital control signals for a wave drive application of the typical application circuit.

TWO-PHASE-ON FULL STEP DRIVE

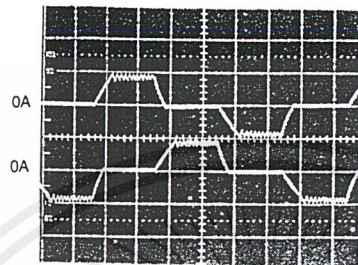
To make the motor take full steps, windings A and B can also be energized in the sequence

$$AB \rightarrow A^*B \rightarrow A^*B^* \rightarrow AB^* \rightarrow AB \rightarrow \dots$$

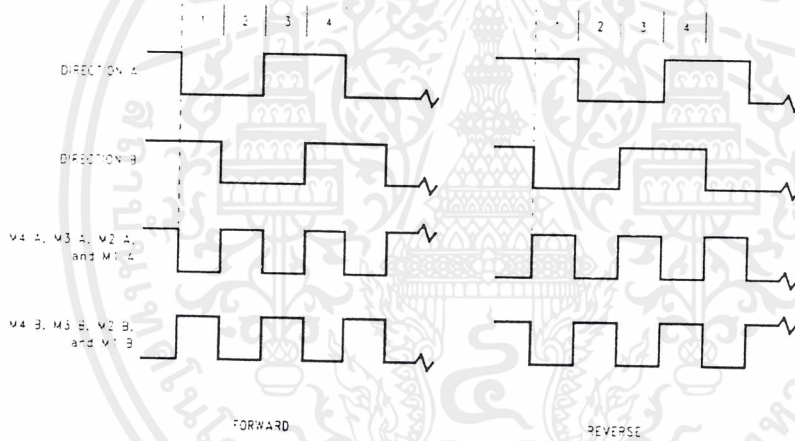
and because both windings are energized at all times, this sequence produces more torque than that produced with wave drive. The motor takes one full step at each change of direction of either winding current. Figure 8 shows the winding currents and digital control signals for this application of the typical application circuit, and Figure 9 shows, for a single phase, the winding current and voltage at the output of the associated current sense amplifier.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The Typical Application (Continued)



Top Trace: Phase A Winding Current at 1A/div
 Bottom Trace: Phase B Winding Current at 1A/div
 Horizontal: 1 ms/div
 Vertical: 1500 steps/second

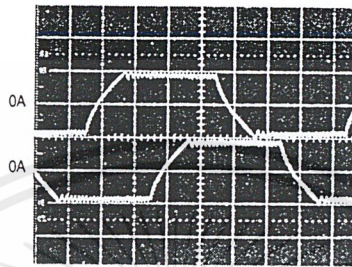


BRAKE A = BRAKE B = 0

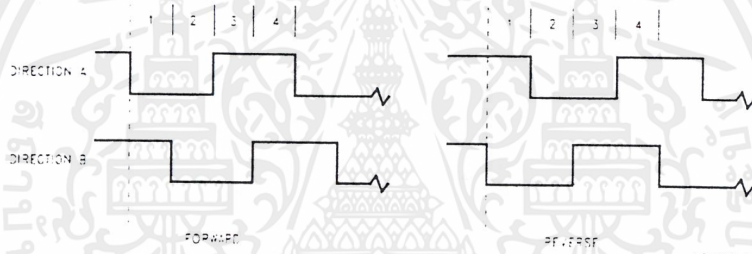
FIGURE 7. Winding Currents and Digital Control Signals for One-Phase-On Drive (Wave Drive)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The Typical Application (Continued)

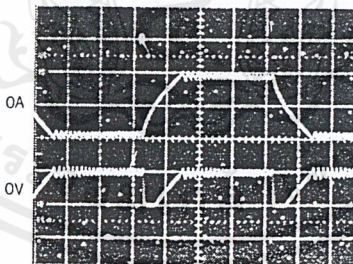


Top Trace: Phase A Winding Current at 1A/div
Bottom Trace: Phase B Winding Current at 1A/div
Horizontal: 1 ms/div
1500 steps/second



M4 A through M1 A = M4 B through M1 B = 1
BRAKE A = BRAKE B = 0

FIGURE 8. Winding Currents and Digital Control Signals for Two-Phase-On Drive



Top Trace: Phase A Winding Current at 1A/div
Bottom Trace: Phase A Sense Voltage at 5V/div
Horizontal: 1 ms/div
1500 steps/second

FIGURE 9. Winding Current and Voltage at the Output of the Associated Current Sense Amplifier

HALF STEP DRIVE WITHOUT TORQUE COMPENSATION

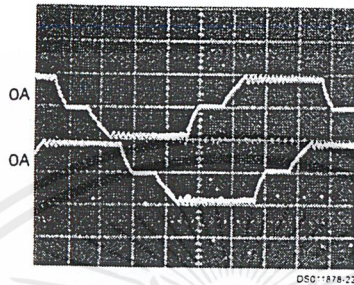
To make the motor take half steps, windings A and B can be energized in the sequence:

$$A \rightarrow AB \rightarrow B \rightarrow A^*B \rightarrow A^* \rightarrow A^*B^* \rightarrow B^* \rightarrow AB^* \rightarrow A \rightarrow \dots$$

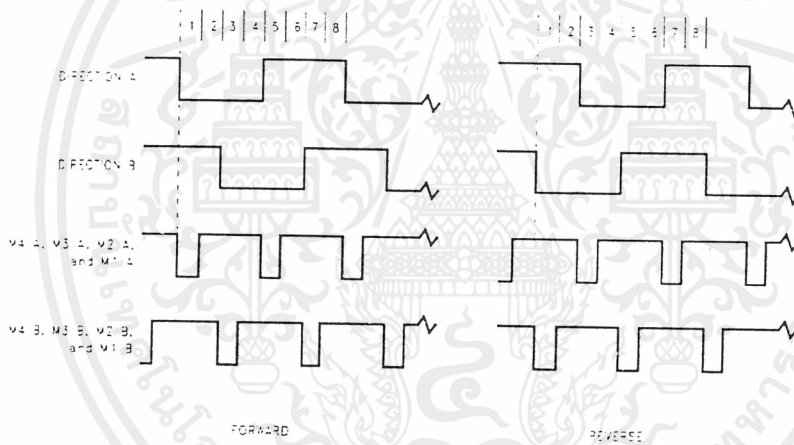
The motor takes one half step each time the number of energized windings changes. It is important to note that al-

though half stepping doubles the step resolution, changing the number of energized windings from two to one decreases (one to two increases) torque by about 40%, resulting in significant torque ripple and possibly noisy operation. Figure 10 shows the winding currents and digital control signals for this half step application of the typical application circuit.

The Typical Application (Continued)



Top Trace Phase A Winding Current at 1A/div
 Bottom Trace Phase B Winding Current at 1A/div
 Horizontal 1 ms/div
 1500 steps/second



BRAKE A = BRAKE B = 0

FIGURE 10. Winding Currents and Digital Control Signals for Half Step Drive without Torque Compensation

HALF STEP DRIVE WITH TORQUE COMPENSATION

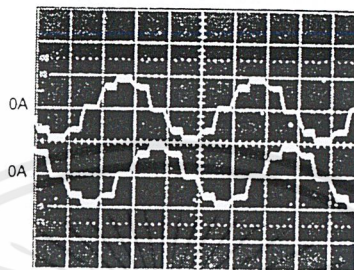
To make the motor take half steps, the windings can also be energized with sinusoidal currents (Figure 11). Controlling the winding currents in the fashion shown doubles the step resolution without the significant torque ripple of the prior drive technique. The motor takes one half step each time the level of either winding current changes. Half step drive with torque compensation is microstepping drive. Along with the obvious advantage of increased step resolution, microstepping reduces both full step oscillations and resonances that occur as the motor and load combination is driven at its natural resonant frequency or subharmonics thereof. Both of

these advantages are obtained by replacing full steps with bursts of microsteps. When compared to full step drive, the motor runs smoother and quieter.

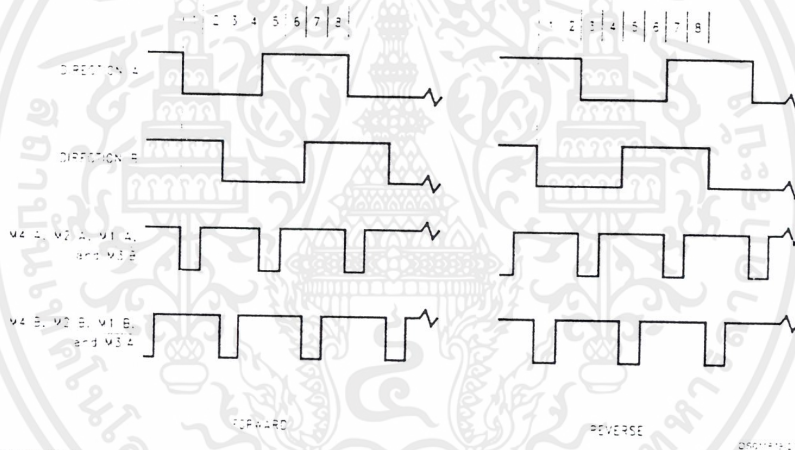
Figure 12 shows the lookup table for this application of the typical application circuit. Dividing 90 electrical per full step by two microsteps per full step yields 45 electrical per microstep μ , therefore, increases from 0 to 315 in increments of 45. Each full 360 cycle comprises eight half steps. Rounding $|\cos(x)|$ to four bits gives D A, the decimal equivalent of the binary number applied at M4 A through M1 A. DIRECTION A controls the polarity of the current in winding A. Figure 11 shows the sinusoidal winding currents.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The Typical Application (Continued)



Top Trace: Phase A Winding Current at 1A/div
 Bottom Trace: Phase B Winding Current at 1A/div
 Horizontal: 2 ms/div
 1500 steps/second



BRAKE A = BRAKE B = 0

90 ELECTRICAL/FULL STEP = 2 MICROSTEPS/FULL STEP = 45 ELECTRICAL/MICROSTEP

FIGURE 11. Winding Currents and Digital Control Signals for Half Step Drive with Torque Compensation

	α	$ \cos(\alpha) $	D A	DIRECTION A	$ \sin(\alpha) $	D B	DIRECTION B
FORWARD	0	1	15	1	0	0	1
↓	45	0.707	11	1	0.707	11	1
	90	0	0	0	1	15	1
	135	0.707	11	0	0.707	11	1
↑	180	1	15	0	0	0	0
REVERSE	225	0.707	11	0	0.707	11	0
↓	270	0	0	1	1	15	0
	315	0.707	11	1	0.707	11	0
	REPEAT						

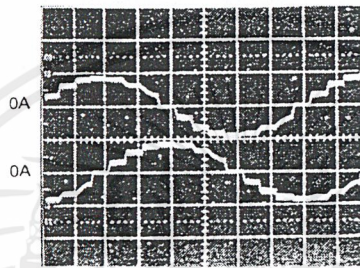
FIGURE 12. Lookup Table for Half Step Drive with Torque Compensation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The Typical Application (Continued)

QUARTER STEP DRIVE WITH TORQUE COMPENSATION

Figure 13 shows the winding currents and lookup table for a quarter step drive (four microsteps per full step) with torque compensation.



Top Trace: Phase A Winding Current at 1A/div
Bottom Trace: Phase B Winding Current at 1A/div
Horizontal: 2ms/div
1250 steps/second

90 ELECTRICAL/FULL STEP = 4 MICROSTEPS; FULL STEP = 22.5° ELECTRICAL/MICROSTEP

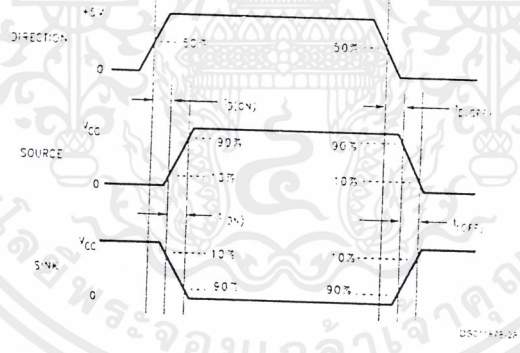
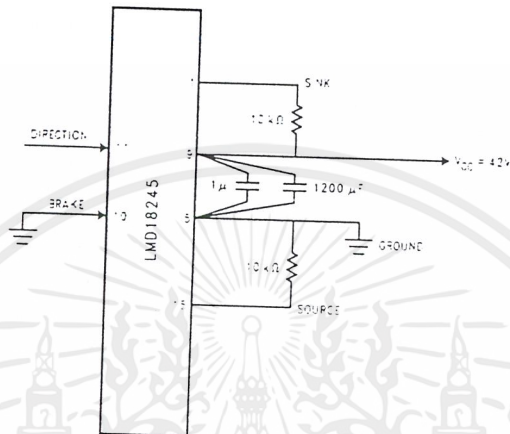
	α	$ \cos(\alpha) $	D A	DIRECTION A	$ \sin(\alpha) $	D B	DIRECTION B
	0	1	15	1	0	0	1
FORWARD	22.5	0.924	14	1	0.383	6	1
	45	0.707	11	1	0.707	11	1
	67.5	0.383	6	1	0.924	14	1
REVERSE	90	0	0	0	1	15	1
	112.5	0.383	6	0	0.924	14	1
	135	0.707	11	0	0.707	11	1
	157.5	0.924	14	0	0.383	6	1
	180	1	15	0	0	0	0
	202.5	0.924	14	0	0.383	6	0
	225	0.707	11	0	0.707	11	0
	247.5	0.383	6	0	0.924	14	0
	270	0	0	1	1	15	0
	292.5	0.383	6	1	0.924	14	0
	315	0.707	11	1	0.707	11	0
	337.5	0.924	14	1	0.383	6	0
	REPEAT						

BRAKE A = BRAKE B = 1

FIGURE 13. Winding Currents and Lookup Table for Quarter Step Drive with Torque Compensation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

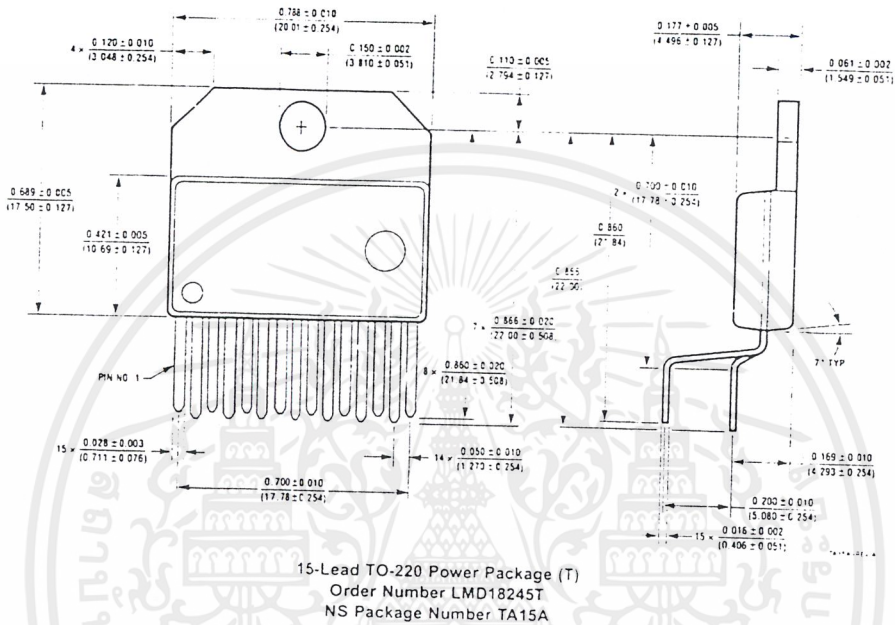
Test Circuit and Switching Time Definitions



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LMD18245 3A, 55V DMOS Full-Bridge Motor Driver

Physical Dimensions inches (millimeters) unless otherwise noted



LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
 Americas
 Tel: 1-800-272-9959
 Fax: 1-855-737-7018
 Email: support@nsc.com

National Semiconductor Europe
 Fax: +49 (0) 1 80-530 85 86
 Email: europe.support@nsc.com
 Division Tel: +49 (0) 1 80-530 85 85
 English Tel: +49 (0) 1 80-532 78 32
 Français Tel: +49 (0) 1 80-532 93 56
 Italiano Tel: +49 (0) 1 80-534 16 80

National Semiconductor Asia Pacific Customer Response Group
 Tel: 65-2544486
 Fax: 65-2514486
 Email: asa.support@nsc.com

National Semiconductor Japan Ltd.
 Tel: 81-3-5620-6175
 Fax: 81-3-5620-6176

www.national.com

National does not assume any responsibility for use of any circuit described herein without patent licenses as implemented in any product. This document is subject to change without notice and specifications are subject to change without notice.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

unit Unit11;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
StdCtrls, ImageEnView, VideoCap, ExtCtrls, Buttons, ImageEnProc, Menus,
ImageEnIO, ComCtrls, IEOpenSaveDlg, iedefs, HistogramBox;
procedure Portout(data:byte);

type

TForm1 = class(TForm)
 GroupBox1: TGroupBox;
 SpeedButton2: TSpeedButton;
 ImageEnProc1: TImageEnProc;
 ImageEnProc2: TImageEnProc;
 GroupBox2: TGroupBox;
 ImageEnIO1: TImageEnIO;
 ImageEnVideoView1: TImageEnVideoView;
 Label5: TLabel;
 Label6: TLabel;
 SaveImageEnDialog1: TSaveImageEnDialog;
 GroupBox3: TGroupBox;
 ImageEnView1: TImageEnView;
 GroupBox4: TGroupBox;
 Button5: TButton;
 Button6: TButton;
 GroupBox5: TGroupBox;
 SpeedButton1: TSpeedButton;
 Button11: TButton;

GroupBox6: TGroupBox;
SpeedButton5: TSpeedButton;
Label7: TLabel;
ComboBox1: TComboBox;
Label1: TLabel;
Label8: TLabel;
HistogramBox1: THistogramBox;
HistogramBox2: THistogramBox;
HistogramBox3: THistogramBox;
GroupBox7: TGroupBox;
Edit4: TEdit;
Edit5: TEdit;
Edit6: TEdit;
Edit7: TEdit;
Edit8: TEdit;
Edit9: TEdit;
Label4: TLabel;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Edit10: TEdit;
Edit11: TEdit;
Label15: TLabel;
Label16: TLabel;
Edit12: TEdit;
Label17: TLabel;

procedure SpeedButton2Click(Sender: TObject);

procedure SpeedButton1Click(Sender: TObject);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับเอาไว้ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure SpeedButton5Click(Sender: TObject);
procedure Button11Click(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure ImageEnVideoView1Job(Sender: TObject; job: TIEJob;
    per: Integer);

private
    { Private declarations }
public
    { Public declarations }
procedure ImageEnVideoView1VideoFrame(Sender: TObject; Bitmap: TBitmap);
procedure DisplayVideoSize;
end;

var
    Form1: TForm1;

implementation

uses giflzw, tiffzlw;
    {SR *.DFM}

    // Input ON
procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
    ImageEnVideoView1.ShowVideo:=SpeedButton2.Down;
    DisplayVideoSize;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

// freeze

procedure TForm1.SpeedButton1Click(Sender: TObject);

begin

ImageEnVideoView1.Frozen:=SpeedButton1.Down;

end;

// Configure source

procedure TForm1.Button5Click(Sender: TObject);

begin

if not ImageEnVideoView1.DoConfigureSource **then**

MessageDlg('Configure Source dialog not available',mtInformation,[mbOK],0)

else

DisplayVideoSize;

end;

// Configure Format

procedure TForm1.Button6Click(Sender: TObject);

begin

if not ImageEnVideoView1.DoConfigureFormat **then**

MessageDlg('Configure Format dialog not available',mtInformation,[mbOK],0)

else

DisplayVideoSize;

end;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Frames to ImageEnView1 (Activate button)
```

```
procedure TForm1.SpeedButton5Click(Sender: TObject):
```

```
begin
```

```
    if SpeedButton5.Down then
```

```
        ImageEnVideoView1.OnVideoFrame:=ImageEnVideoView1.VideoFrame
```

```
    else begin
```

```
        ImageEnVideoView1.OnVideoFrame:=nil;
```

```
        portout(32);
```

```
        Edit12.Text:='Stop';
```

```
    end;
```

```
end;
```

```
// out to port
```

```
procedure Portout(data:byte);
```

```
begin
```

```
    asm
```

```
        push dx
```

```
        push ax
```

```
        mov dx,$378
```

```
        mov al,data
```

```
        out dx,al
```

```
        pop ax
```

```
        pop dx
```

```
    end;
```

```
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Frames to ImageEnView1 - OnVideoFrame
```

```
procedure TForm1.ImageEnVideoView1.VideoFrame(Sender: TObject; Bitmap: TBitmap);
```

```
type
```

```
    frame2=array[1..3,1..150,1..150] of integer;
```

```
    frame1=array[1..160,1..120] of integer;
```

```
var
```

```
    x1:TColor;
```

```
    x2:TRGB;
```

```
    pixel2:frame2;
```

```
    pixel:frame1;
```

```
    x,y,a,b,m,xa,xb,n,ga,gb:integer;
```

```
    a1,b1,a2,b2,a3:integer;
```

```
    x11,y11,y21,x21:integer;
```

```
begin
```

```
    ImageEnView1.assign(Bitmap);
```

```
    case ComboBox1.ItemIndex of
```

```
        1: ImageEnProc2.ConvertToBWOrdered;
```

```
        2: ImageEnProc2.ConvertToBWThreshold(-1);
```

```
        3: ImageEnProc2.ConvertToGray;
```

```
    end;
```

```
    ImageEnView1.Fit;
```

```
    HistogramBox1.Update();
```

```
    HistogramBox2.Update();
```

```
    HistogramBox3.Update();
```

```
// image processing //
```

```
for a:=1 to 160 do // keep RGB value
```

```
begin
```

```
    for b:=1 to 120 do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
begin
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

x1:=ImageEnView1.Bitmap.Canvas.Pixels[a,b];
x2.r:=x1 and S000000FF;
pixel[a,b]:=x2.r;
end;
end;

```

```

//..... ..deteck object^.....//

```

```

#1 TOP

```

```

a1:=2; b1:=2;
for x:=2 to 158 do
begin
for y:=2 to 118 do
begin
m:=pixel[x,y]-pixel[x,y+1];
if m<0 then m:= m*(-1);
if m>70 then
begin
pixel2[1,a1,1]:=x;
pixel2[1,1,b1]:=y;
if pixel2[1,1,b1]<pixel2[1,1,2] then
begin
pixel2[1,2,1]:=x;
pixel2[1,1,2]:=y;
end;
end;
a1:=a1+1;
b1:=b1+1;
break;
break;
end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
x11:=pixel2[1,2,1];  
y11:=pixel2[1,1,2];
```

```
#2 RIGHT
```

```
a3:=2;
```

```
for y:=1 to 118 do
```

```
begin
```

```
for x:=158 downto 4 do
```

```
begin
```

```
m:=pixel[x,y]-pixel[(x-1),y];
```

```
if m<0 then m:= m*(-1);
```

```
if m>70 then
```

```
begin
```

```
pixel2[2,a3,1]:=y;
```

```
pixel2[2,1,a3]:=x;
```

```
if pixel2[2,1,a3]>pixel2[2,1,2] then
```

```
begin
```

```
pixel2[2,1,2]:=x;
```

```
pixel2[2,2,1]:=y;
```

```
end;
```

```
a3:=a3+1;
```

```
break;
```

```
break;
```

```
end;
```

```
end;
```

```
end;
```

```
ImageEnView1.Bitmap.Canvas.Pixels[pixel2[2,1,2],pixel2[2,2,1]]:=clred;
```

```
x21:=pixel2[2,1,2];
```

```
y21:=pixel2[2,2,1];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

# case at top
if y21-y11 < 10 then
begin
a1:=2; b1:=2;
for x:=2 to 158 do
begin
for y:=118 downto 2 do
begin
m:=pixel[x,y]-pixel[x,(y-1)];
if m<0 then m:= m*(-1);
if m>70 then
begin
pixel2[1,a1,1]:=x;
pixel2[1,1,b1]:=y;
if pixel2[1,1,b1]>pixel2[1,1,2] then
begin
pixel2[1,2,1]:=x;
pixel2[1,1,2]:=y;
end;
a1:=a1+1;
b1:=b1+1;
break;
break;
end;
end;
end;
x11:=pixel2[1,2,1];
y11:=pixel2[1,1,2];
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Edit9.Text:=InttoStr(y21);
Edit8.Text:=InttoStr(x21);
Edit4.Text:=InttoStr(x11);
Edit5.Text:=InttoStr(y11);
ImageEnView1.Bitmap.Canvas.Pixels[x11,y11]:=clblue;

```

```
// control robot
```

```

xa:=x11-80;
xb:=y21-60;
ga:=xa div 8;
gb:=xb div 6;
Edit6.Text:=InttoStr(xa);
Edit7.Text:=InttoStr(xb);
Edit10.Text:=InttoStr(ga);
Edit11.Text:=InttoStr(gb);

```

```
// scan x
```

```

if ga<-1 then // Left
  begin
    portout(8); //for turn left
    Edit12.Text:='Turn Left';
  end
else
  if ga>1 then // right
    begin
      portout(2); //for turn right
      Edit12.Text:='Turn Right';
    end
  else

```

```
// scan y
```

```
if gb <-1 then
```

```
begin
```

```
portout(1); // for forward
```

```
Edit12.Text:='Go straight on';
```

```
end
```

```
else
```

```
if gb > 1 then
```

```
begin
```

```
portout(16); // for go back
```

```
Edit12.Text:='Go Back';
```

```
end
```

```
else
```

```
begin
```

```
portout(32); // for stop
```

```
Edit12.Text:='Stop';
```

```
end;
```

```
end;
```

```
// Save as...
```

```
procedure TForm1.Button11Click(Sender: TObject);
```

```
begin
```

```
if SaveImageEnDialog1.Execute then
```

```
ImageEnIO1.SaveToFile(SaveImageEnDialog1.filename);
```

```
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// DisplayVideoSize
procedure TForm1.DisplayVideoSize;
var
    r:TRect;
begin
    r:=ImageEnVideoView1.GetVideoSize;
    Label6.caption:=inttostr(r.right+1)+'x'+inttostr(r.bottom+1);
end;

procedure TForm1.FormActivate(Sender: TObject);
begin
    DefGIF_LZWDECOMPFUNC:=GIFLZWDecompress;
    DefGIF_LZWCOMPFUNC:=GIFLZWCompress;
    DefTIFF_LZWDECOMPFUNC:=TIFFLZWDecompress;
    DefTIFF_LZWCOMPFUNC:=TIFFLZWCompress;
    ComboBox1.ItemIndex:=0;
end;

procedure TForm1.ImageEnVideoView1Job(Sender: TObject; job: TIEJob;
per: Integer);
begin
    case job of
        iejNOTHING: Label8.Caption:="";
        iejVIDEOCAP_CONNECTING: Label8.Caption:='Connecting...';
    end;
    Application.ProcessMessages;
end;
end.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมนี้เป็นโปรแกรมที่วิเคราะห์และประยุกต์ภาพที่รับเข้ามาจากกล้องวีดีโอดิจิทัล โดยส่วนหนึ่งมาจากการดาวน์โหลด และอีกส่วนหนึ่งได้เขียนเพิ่มขึ้นเอง

ส่วนที่เขียนเพิ่มขึ้นเป็นส่วนที่หาเป้าหมายที่ต้องการบนฉาก โดยจะบันทึกคู่ลำดับของ ตำแหน่งของเป้าหมายไว้ตลอดการเคลื่อนที่

จากนั้นจะเปรียบเทียบคู่ลำดับที่ได้กับจัดอ้างอิงที่กึ่งกลางของเฟรมภาพ เพื่อส่งข้อมูลไปยัง ไมโครคอนโทรลเลอร์ให้ทำการควบคุมการเคลื่อนที่ของรถ ไม่ว่าเป้าหมายจะเคลื่อนที่ไปที่ใดก็ตาม รถจะเคลื่อนที่ไปตามเป้าหมายเพื่อตั้งให้ภาพคู่ศูนย์กลางจอเสมอ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ORG 0000
JMP MAIN
ORG 0020H
MAIN: MOV P2,#0FH ; (Stop and check commands from
CALL CHECK_R ; computer by pooling method.
JZ LEFT ; "Turn Right" is the command that it will
; consider first if it's not turn right it will check
; if it is "Turn Left")

RIGHT: MOV P2,#1000B ; (If the command is "Turn Right".
CALL DE_1 ; Microcontroller will set DC Motor to turn
MOV P2,#0FH ; right until the command from computer has
CALL DELAY ; change then it wil jump to MAIN)
CALL CHECK_R
JZ MAIN
JMP RIGHT

LEFT: CALL CHECK_L ; (Check command if it is not "Turn Left" it
JZ FW ; will jump to FW either Microcontroller will set
MOV P2,#0010B ; DC motor to turn left LL: until the command
CALL DE_1 ; from computer has change then it will jump
MOV P2,#0FH ; to MAIN)
CALL DELAY
CALL CHECK_L
JNZ MAIN
JMP LL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

FW:      CALL CHECK      ; (Check command if it is not "Forward" it will
        JZ    BACK      jump to BACK either Microcontroller will
                                set DC motor to go straight on until command
                                from computer has change then it will
                                jump to MAIN)

```

```

FF:      MOV    P2,#1010B
        CALL   DE_1
        MOV    P2,#0FH
        CALL   DELAY
        CALL   CHECK
        JNZ    FF
        LJMP   MAIN

```

```

BACK:    CALL   CHECK_B   ; ( Check command if it is not "Back" it will
        JZ    STOP      jump to STOP either Microcontroller will
                                set DC motor to go backward until command
                                from computer has change then it will jump
                                to MAIN)

```

```

BB:      MOV    P2,#00H
        CALL   DE_1
        MOV    P2,#0FH
        CALL   DELAY
        CALL   CHECK_B
        JNZ    BACK
        LJMP   MAIN

```

```

STOP:    MOV    P2,#0FH   ; ( Stop until command from computer has change
        CALL   CHECK_S   then it will jump to MAIN)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
JNZ STOP
LJMP MAIN
```

```
CHECK_R: MOV P3,#0FFH      ;( Compare and Store the result)
          MOV A,P3
          ANL A,#02H
          RET
```

```
CHECK_L: MOV P3,#0FFH
          MOV A,P3
          ANL A,#08H
          RET
```

```
CHECK:   MOV P3,#0FFH
          MOV A,P3
          ANL A,#01H
          RET
```

```
CHECK_B: MOV P3,#0FFH
          MOV A,P3
          ANL A,#10H
          RET
```

```
CHECK_S: MOV P3,#0FFH
          MOV A,P3
          ANL A,#20H
          RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DE_1:    MOV    R0,#0FFH    ; ( Make pulse for controlling DC motor)
         DJNZ   R0,S
         RET
```

```
DELAY:   MOV    R1,#0FFH
```

```
DELAY1:  MOV    R2,#02H
         DJNZ   R2,S
         DJNZ   R1,DELAY1
         RET
```



กิตติกรรมประกาศ

หลังจากการทำโครงการนี้ทำให้ผู้จัดทำได้รับความรู้มากมาย ทั้งทางเรื่องของคลื่นอุตสาหกรรมวิทยุ โชนิก มอเตอร์ไฟกระแสตรง และด้านไมโครคอนโทรลเลอร์ ซึ่งความรู้ดังกล่าวนี้ส่วนใหญ่มาจากคำแนะนำของผู้มีอุปการะหลาย ๆ ท่าน ตั้งแต่อาจารย์ที่ปรึกษา ผศ.ดร. สุรพันธ์ เอื้อไพบูลย์ สำหรับคำแนะนำ และผู้อำนวยการสถาน เครื่องมือที่ใช้ในการทำโครงการชิ้นนี้เป็นอย่างดี

ขอขอบคุณเพื่อน ๆ ในภาคอิเล็กทรอนิกส์หลาย ๆ ท่านที่ช่วยให้คำปรึกษา และยังคงคอยให้กำลังใจเสมอมา

ขอขอบคุณเพื่อน ๆ ภาคไฟฟ้ากำลังที่ให้ทั้งคำปรึกษา ข้อเสนอแนะ รวมทั้งยังช่วยหาอุปกรณ์ในการจัดทำที่ทางผู้จัดทำไม่ทราบว่าจะหาได้อย่างไร ทั้งยังช่วยในการจัดทำในหลายๆ ด้าน

ขอขอบคุณเพื่อน ๆ ชุมชนอิเล็กทรอนิกส์ที่ให้คำปรึกษา และผู้อำนวยการ IC ที่ใช้ในโครงการนี้
คุณค่าและประโยชน์อันพึงมีจากโครงการนี้ ผู้จัดทำขอขอบแต่ผู้มีอุปการะทุกท่าน

บริษัท สิวดีทาคอนส์

นางสาวนริศา สิทธิกานต์

หนังสือ เพลิดเพลิน

นางสาวหนึ่งนุช เหลืองพิพัฒน์

เอกสารอ้างอิง

1. พิพัฒน์ เลหาสงคราม “ไมโครคอนโทรลเลอร์ MCS-51” ภาควิชาเทคโนโลยีการวัดคุมอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. พฤษภาคม 2537
2. ประเมษฐ์ ประณยานันท์, ปิยะพงศ์ เผ่าณิข. “คู่มือและการประยุกต์ใช้งานไมโครคอนโทรลเลอร์ MCS-51” , บริษัทซีเอ็ดดูเคชั่น
3. มงคล ทองสงคราม. “เครื่องกลไฟฟ้ากระแสตรง” , บริษัทรามการพิมพ์ จำกัด . เมษายน 2533
4. รศ.สัมพันธ์ หาญเขต. “เครื่องกลไฟฟ้า 1” ,ภาควิชาวิศวกรรมไฟฟ้า สถาบันเทคโนโลยีพระจอมเกล้าธนบุรี บางมด, พฤษภาคม 2519
5. สัจจะ จรัสรุ่งเรือง,จักรพงษ์ สุขประเสริฐ. “Delphi 5” ,สำนักพิมพ์อินโฟเพรส .กุมภาพันธ์ 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้