

หุ่นยนต์สำรวจเคลื่อนที่ควบคุมการทำงานด้วยไมโครคอมพิวเตอร์

SURVEYING MOBILE ROBOTIC CONTROLLED BY
MICROCOMPUTER



นายภัทรพันธ์ ภัตสรศิริ
นายพีระพล เย่าตัก

เลขหมู่ 217
เลขทะเบียน 4 2522
วัน, เดือน, ปี 24 พ.ศ. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีนานาชาติพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b11220160

**SURVEYING MOBILE ROBOTIC CONTROLLED BY
MICROCOMPUTER**




**A THESIS SUBMITTED IN PARTIAL FULLFILMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING
DEPARTMENT OF INDUSTRIAL INSTRUMENTATION TECHNOLOGY
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2000

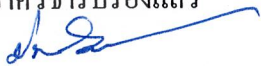
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาเทคโนโลยีวัดคุมทางอุตสาหกรรม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาโท

หัวข้อปริญญาโท หุ่นยนต์สำรวจเคลื่อนที่ควบคุมการทำงานด้วยไมโครคอมพิวเตอร์
SURVEYING MOBILE ROBOTIC CONTROLLED BY
MICROCOMPUTER
นักศึกษาผู้จัดทำ นายภัทรพันธ์ ภัสสรศิริ รหัสประจำตัว 40010549
นายพีระพล เขาคัด รหัสประจำตัว 40010527
ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2543

อาจารย์ผู้ควบคุมปริญญาโท	ลายมือชื่อ
อ.ทวีพล ชื้อสตัย	

วัน/เดือน/ปี ที่สอบ วันพฤหัสบดีที่ 15 มีนาคม พ.ศ.2544
สถานที่สอบ ณ ห้องสอบปริญญาโท ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

ภาควิชารับรองแล้ว

(ผศ.ประสิทธิ์ จุลเสรีวงศ์)
หัวหน้าภาควิชา ฯ

หัวข้อปริญญานิพนธ์

หุ่นยนต์สำรวจเคลื่อนที่ควบคุมการทำงานด้วยไมโครคอมพิวเตอร์
SURVEYING MOBILE ROBOTIC CONTROLLED BY
MICROCOMPUTER

นักศึกษาผู้จัดทำ

นายภัทรพันธ์ ภัตสรศิริ
นายพีระพล เย่าตัก

อาจารย์ที่ปรึกษา

อาจารย์ทวีพล ชื้อสตัย

ปีการศึกษา

2543

บทคัดย่อ

โครงการนี้นำเสนอการออกแบบหุ่นยนต์ที่ใช้ในการสำรวจ ควบคุมการทำงานด้วยคอมพิวเตอร์ โดยจะรับคำสั่งของเส้นทางการเคลื่อนที่จากโปรแกรมคอมพิวเตอร์ จากนั้นจะส่งผ่านข้อมูลผ่านพอร์ตอนุกรม RS-232C ไปยังไมโครโปรเซสเซอร์ที่ตัวหุ่นยนต์ และในตัวโปรแกรมจะมีการวาดแผนที่แสดงการเคลื่อนที่ของตัวหุ่นยนต์ มีการลากเส้นตามเส้นทางการเคลื่อนที่พร้อมไปกับการเคลื่อนที่ของตัวหุ่น ในการเที่ยวของตัวหุ่นยนต์ได้มีการนำกลไกเข้ามาช่วยเพื่อให้ได้ความแม่นยำในการหมุนตัวเวลาเลี้ยว เป็นการออกแบบให้มีการเคลื่อนที่ในรูปแบบใหม่ ๆ ของหุ่นยนต์ในรูปแบบที่ต่างไปจากทั่วไป เพื่อให้เกิดแนวทาง ในการศึกษาพัฒนาการควบคุมระหว่างคอมพิวเตอร์และกลไก

Thesis Title	Surveying Mobile Robotic Controlled By Microcomputer	
Authers	MR. Pattarapan	Passornsiri
	MR. Peerapon	Yowtak
Thesis Advisor	MR. Taweepol	Suesat
Year	2000	

ABSTRACT

This project presents the designing of Surveying Mobile Robotic Controlled by Microcomputer. The routine of robot is controlled by computer program. Then the movement will send information through serial port RS-232 to microprocessor at the robot. And we can see the routine of robot when it moves anyway by map at program, the line will move follow the routine of robot, so the program can make a map. The robot movement designs for exactly rotate by use mechanical system for turn body of robot. This movement design is different from ordinary robot that created for development of control between computer and mechanic.

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้คงไม่สามารถดำเนินการจนสำเร็จลงได้ ถ้าปราศจากการสนับสนุนจากผู้ที่มิพระคุณเหล่านี้

1. อาจารย์ทวีพล ชื้อสัคย์ - อาจารย์ที่ปรึกษา
2. ชุมนุมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ - ให้ยืมเครื่องมืออุปกรณ์ และสถานที่ทำปริญญานิพนธ์
3. คุณศุภชัย ชนะฤทธิชัย - ผู้อนุเคราะห์เครื่องโปรแกรมไอซี 8051 ให้ใช้ตั้งแต่เริ่มต้นจนโครงการนี้สำเร็จ
4. อาจารย์ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรมทุกท่านที่ให้คำแนะนำอันเป็นประโยชน์ต่อการทำปริญญานิพนธ์ฉบับนี้



คณะผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
บทที่ 2 ทฤษฎีและหลักการ.....	2
2.1 ไมโครคอนโทรลเลอร์ MCS-51.....	2
2.1.1 สถาปัตยกรรมของไมโครคอนโทรลเลอร์ 8051.....	2
2.1.2 พอร์ตอินพุต/เอาต์พุต.....	4
2.1.3 รีจิสเตอร์สำหรับใช้งานทั่วไปใน MCS-51.....	5
2.1.4 ไทม์เมอร์/เคาน์เตอร์.....	6
2.1.5 การอินเตอร์รัプト.....	7
2.1.6 การรับส่งข้อมูลอนุกรม (UART) ของ MCS-51.....	7
2.2 ความรู้เบื้องต้นเกี่ยวกับพอร์ตอนุกรม.....	8
2.2.1 การสื่อสารแบบอนุกรม.....	8
2.2.2 การสื่อสารข้อมูลแบบอะซิงโครนัส.....	9
2.2.3 มาตรฐานพอร์ตอนุกรมแบบ RS-232.....	10
2.2.4 คอนเน็คเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ.....	11
2.2.5 UART (Universal Asynchronous Receiver Transmitter).....	13
บทที่ 3 การออกแบบและการสร้าง.....	14
3.1 รายละเอียดเกี่ยวกับโครงงาน.....	14
3.2 การออกแบบหุ่นยนต์.....	15
3.2.1 แนวคิดในการออกแบบ.....	15

สารบัญ(ต่อ)

	หน้า
3.2.2 การเปลี่ยนทิศทางในลักษณะการเดี่ยว.....	15
3.2.3 การเปลี่ยนทิศทางในลักษณะการหมุน.....	16
3.3 ลักษณะของหุ่นยนต์ที่ออกแบบ.....	17
3.3.1 ฮาร์ดแวร์.....	17
3.3.1.1 โครงสร้างหุ่นยนต์.....	18
3.3.1.2 ชุดขับเคลื่อน.....	19
3.3.1.3 วงจรควบคุม.....	20
3.3.1.3.1 วงจรไมโครคอนโทรลเลอร์.....	21
3.3.1.3.2 วงจรขับเคลื่อนมอเตอร์.....	22
3.3.1.3.3 วงจรนับรอบ หรือวงจรเข้ารหัส.....	24
3.3.2 ซอฟต์แวร์.....	25
3.3.2.1 โปรแกรมที่ทำงานบนไมโครคอนโทรลเลอร์หรือบนตัวหุ่นยนต์.....	25
3.3.2.2 โปรแกรมที่ทำงานบนไมโครคอมพิวเตอร์.....	26
3.3.2.3 การทำงานของโปรแกรม.....	27
บทที่ 4 การทดลองและผลสรุป.....	32
4.1 การทดลองการเคลื่อนที่ในแนวตรง.....	32
4.2 การทดลองการหมุนตัว.....	36
บทที่ 5 สรุปผลงานวิจัยและแนวทางในการพัฒนาต่อไป.....	40
บรรณานุกรม.....	42
ภาคผนวก.....	43
ภาคผนวก ก.	44
ภาคผนวก ข.	48
ภาคผนวก ค.	61

สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงลำดับความสำคัญของการอินเทอร์เน็ต.....	7
2.2 การจัดขาสัญญาของพอร์ทอนุกรมในแบบต่างๆ และหน้าที่การทำงาน.....	12
4.1 ผลการทดลองเมื่อสั่งให้เคลื่อนที่ไปข้างหน้าและย้อนกลับเป็นระยะทาง 1000 หน่วย.....	33
4.2 ผลการทดลองเมื่อสั่งให้เคลื่อนที่ไปข้างหน้าและย้อนกลับเป็นระยะทาง 2000 หน่วย.....	33
4.3 ผลการทดลองเมื่อสั่งให้เคลื่อนที่ไปข้างหน้าและย้อนกลับเป็นระยะทาง 4000 หน่วย.....	34
4.4 การทดสอบการหมุนที่ 30 องศา.....	37
4.5 การทดสอบการหมุนที่ 90 องศา.....	37
4.6 การทดสอบการหมุนที่ 180 องศา.....	38
4.7 การทดสอบการหมุนที่ 360 องศา.....	38



สารบัญรูป

รูปที่	หน้า
2.1 โครงสร้างภายในชิพไมโครคอนโทรลเลอร์ 8051.....	3
2.2 ไคอะแกรมของ 8051.....	4
2.3 รีจิสเตอร์ใช้งานเฉพาะ TMOD.....	6
2.4 ไคอะแกรมเวลาของการสื่อสารข้อมูลแบบซิงโครนัส.....	8
2.5 รูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส.....	9
2.6 ขาสัญญาณและการเชื่อมต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ผ่าน RS-232.....	13
3.1 แสดงการหมุนตัวของหุ่นยนต์ที่ออกแบบไว้.....	17-18
3.2 แสดงโครงสร้างพื้นฐานของหุ่นยนต์.....	18
3.3 มอเตอร์ที่ใช้ในการขับเคลื่อนและเฟืองทดรอบเพิ่มแรงบิดซึ่งติดตั้งเอ็นโคดเดอร์ไว้ด้วย.....	19
3.4 การยึดแกนหมุนเข้ากับเฟืองตัวใหญ่ ซึ่งนับการหมุนได้ละเอียด 0.523 องศา.....	19
3.5 แสดงชุดมอเตอร์ขับเคลื่อนเมื่อเชื่อมต่อเข้ากับเพลาล้อขับเคลื่อน.....	20
3.6 แสดงวงจรของไมโครคอนโทรลเลอร์.....	21
3.7 แสดงการจับวางขาของ LMD18200 และ LMD18245.....	22
3.8 โครงสร้างภายในของ LMD18200.....	22
3.9 วงจรใช้งานของ LMD18200.....	23
3.10 โครงสร้างภายในของ LMD18245.....	23
3.11 วงจรใช้งานของ LMD18245.....	24
3.12 แสดงชุดเอ็นโคดเดอร์และวงจรของเอ็นโคดเดอร์.....	24
3.13 แสดงการทำงานของโปรแกรมบนไมโครคอนโทรลเลอร์.....	25
3.14 แผนภูมิการทำงานของโปรแกรมควบคุม.....	27
3.15 ส่วนประกอบต่างๆของโปรแกรมควบคุมบนไมโครคอมพิวเตอร์.....	28
3.16 ลักษณะของการสั่งให้หุ่นยนต์เคลื่อนที่.....	29
3.17 แสดงการสร้างแผนที่ของโปรแกรมเมื่อมีการเคลื่อนที่ของหุ่นยนต์เกิดขึ้น.....	29
3.18 แสดงผลเมื่อมีการเลี้ยว ในส่วนแผนที่จะมีการบอกทิศทางว่าเลี้ยวไปแล้ว.....	30
3.19 การแสดงผลเมื่อใช้คำสั่งระยะทางระหว่างจุดสองจุด.....	31
4.1 การทดลองการเคลื่อนที่โดยให้หุ่นยนต์เคลื่อนที่เข้าไปมา.....	32
4.2 กราฟระยะทางของการเคลื่อนที่ที่หุ่นยนต์เคลื่อนที่ไปจากการทดลอง.....	34

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.3	กราฟระยะทางของการเคลื่อนที่ที่หุ่นยนต์เคลื่อนที่กลับมายังจุดเริ่มต้นจากการทดลอง.....35
4.4	กระดาษซึ่งตีเส้นตัดกันเป็นมุม 15 องศา เพื่อใช้ในการทดสอบการหมุนตัว.....36



บทที่ 1

บทนำ

ในปัจจุบันเทคโนโลยีมีการพัฒนาไปอย่างก้าวหน้าในหลายด้าน ไม่ว่าจะเป็นการพัฒนาทางด้านการสื่อสารโทรคมนาคม เครื่องจักรกล อินเทอร์เน็ต ฯลฯ โดยเฉพาะอย่างยิ่งวิทยาการทางด้านคอมพิวเตอร์มีความก้าวหน้าเป็นอย่างยิ่ง เป็นส่วนที่สำคัญที่ทำให้วิทยาการของมนุษย์มีการพัฒนาไปได้อย่างรวดเร็ว สามารถนำมาใช้กับชีวิตประจำวันช่วยอำนวยความสะดวกให้แก่มนุษย์ได้มากขึ้น เรานำคอมพิวเตอร์มาใช้สำหรับควบคุมเครื่องจักรกลทำให้เกิดประสิทธิภาพในการทำงาน สามารถตัดแปลงและติดต่อกับเครื่องจักรโดยผ่านคอมพิวเตอร์ จึงเป็นเหตุผลที่มาของการศึกษาโครงการนี้

หุ่นยนต์สำรวจการเคลื่อนที่นี้จะถูกควบคุมผ่านทางคอมพิวเตอร์ โดยคอมพิวเตอร์จะทำหน้าที่รับคำสั่งการเคลื่อนที่ทางมาตัสและคีย์บอร์ด ทำการประมวลผลคำสั่งที่ได้รับ แล้วส่งข้อมูลขนาด 8 ไบต์ ผ่านพอร์ตอนุกรม RS-232 มาที่ไมโครคอนโทรลเลอร์บนตัวหุ่นยนต์ โดยส่วนของไมโครคอนโทรลเลอร์นี้จะทำหน้าที่เพียงแต่ควบคุมการขับของมอเตอร์ที่ใช้บังคับการเคลื่อนที่ของหุ่นยนต์เท่านั้น ในการนำคอมพิวเตอร์มาใช้ในการควบคุมหุ่นยนต์นี้จะช่วยทำให้ภาระในการประมวลผลของไมโครโปรเซสเซอร์มารวมกันที่คอมพิวเตอร์ และช่วยให้ง่ายต่อการเขียนโปรแกรม, แก้ไข โปรแกรมการทำงานของหุ่นยนต์ นอกจากนี้รูปแบบการเคลื่อนที่ของหุ่นยนต์ในโครงการนี้ยังนำเสนอในแบบที่ต่างออกไปเพื่อที่จะให้ได้รับความแม่นยำในการเคลื่อนที่มากยิ่งขึ้น ในการควบคุม และได้มีการเขียนโปรแกรมรองรับการทำงานเพื่อให้ใช้ประโยชน์ได้หลาย ๆ อย่าง เช่น การเขียนแผนที่ตามเส้นทางเคลื่อนที่ของหุ่นยนต์ และการควบคุมการนับรอบของมอเตอร์ เป็นต้น สำหรับปริญญาโทฉบับนี้จะนำเสนอถึงเรื่องการสร้างและออกแบบหุ่นยนต์สำรวจการเคลื่อนที่ ซึ่งควบคุมการทำงานด้วยไมโครคอมพิวเตอร์ ซึ่งหวังว่าจะสามารถนำมาประยุกต์ใช้โดยติดตั้งอุปกรณ์ตรวจวัดต่าง ๆ เพื่อให้นำไปใช้งานได้อย่างมีประสิทธิภาพ

บทที่ 2

ทฤษฎีและหลักการ

2.1 ไมโครคอนโทรลเลอร์ MCS-51

MCS-51 เป็นไมโครคอนโทรลเลอร์ลักษณะแบบชิพเดี่ยวที่มีข้อดีเมื่อนำมาเปรียบเทียบกับไมโครโปรเซสเซอร์ขนาด 8 บิตตระกูลอื่น ดังนี้

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

นอกจากนี้ MCS-51 ยังมีคุณสมบัติอื่น ๆ ที่น่าสนใจ คือ

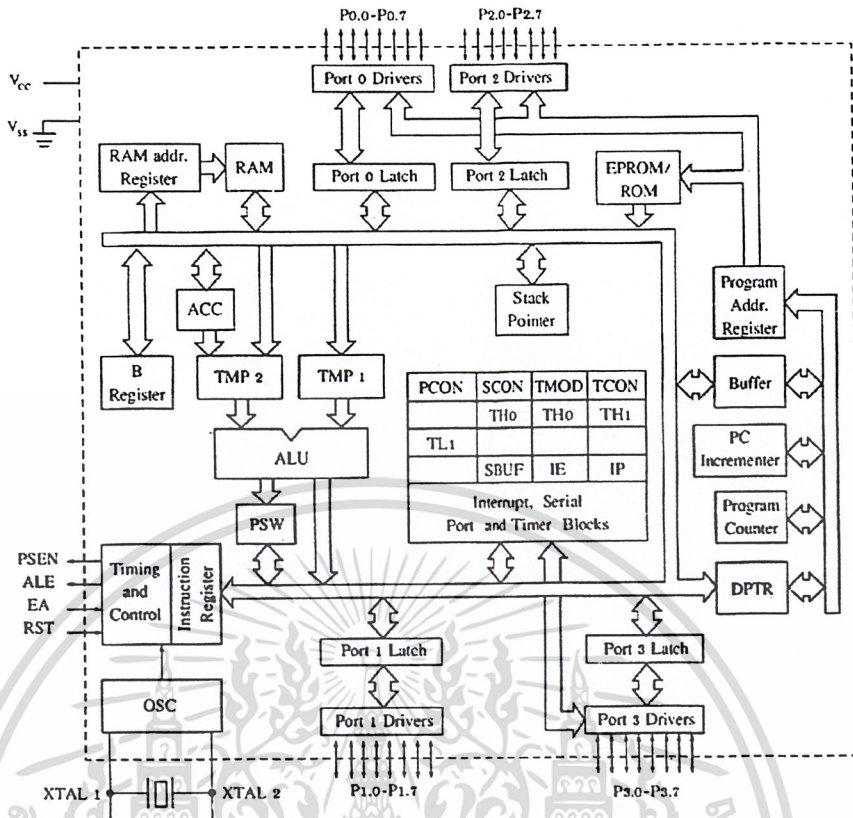
- ต้องการแหล่งจ่ายไฟ 5 โวลต์เพียงชุดเดียว
- มีหน่วยความจำสำหรับเก็บโปรแกรมควบคุมการทำงานอยู่ภายในชิพ
- สามารถที่ใช้หน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลที่อยู่ภายนอกชิพได้อย่างละ 64 กิโลไบต์
- มีคำสั่งคูณและหารเลขขนาด 8 บิตในตัวเอง
- จัดลำดับความสำคัญของสัญญาณอินเทอร์รัปต์ได้ 2 ระดับ
- รับและส่งข้อมูลแบบอนุกรมได้ในตัวโดยที่สามารถกำหนดอัตราเร็วในการรับและการส่ง ข้อมูลได้ตั้งแต่ 300 ถึง 375 กิโลบิตต่อวินาที
- สามารถประมวลผลแบบบูลีนเพื่อใช้ในงานควบคุมโดยเฉพาะ
- มีรีจิสเตอร์สำหรับใช้งานเป็น ไทม์เมอร์หรือเคาน์เตอร์เพื่อที่จะใช้นับจำนวนสัญญาณนาฬิกาภายในชิพ หรือนับการเปลี่ยนแปลงสถานะของสัญญาณภายนอกขนาด 16 บิต จำนวน 2 ตัว เพื่อใช้สำหรับนับจำนวนพัลส์ วัดความกว้างของพัลส์หรือใช้วัดช่วงเวลา
- หน่วยความจำสำหรับเก็บข้อมูลภายในบางส่วนก็สามารถเข้าถึงข้อมูลได้ทั้งระดับไบต์และบิตเพื่อให้การออกแบบโปรแกรมและการควบคุมระบบงานทำได้ง่ายขึ้น

2.1.1 สถาปัตยกรรมของไมโครคอนโทรลเลอร์ 8051

โครงสร้างภายในชิพไมโครคอนโทรลเลอร์ 8051 ชิพเดี่ยวแสดงดังรูปที่ 2.1

ซึ่งอธิบายถึงส่วนย่อยๆ ภายใน 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

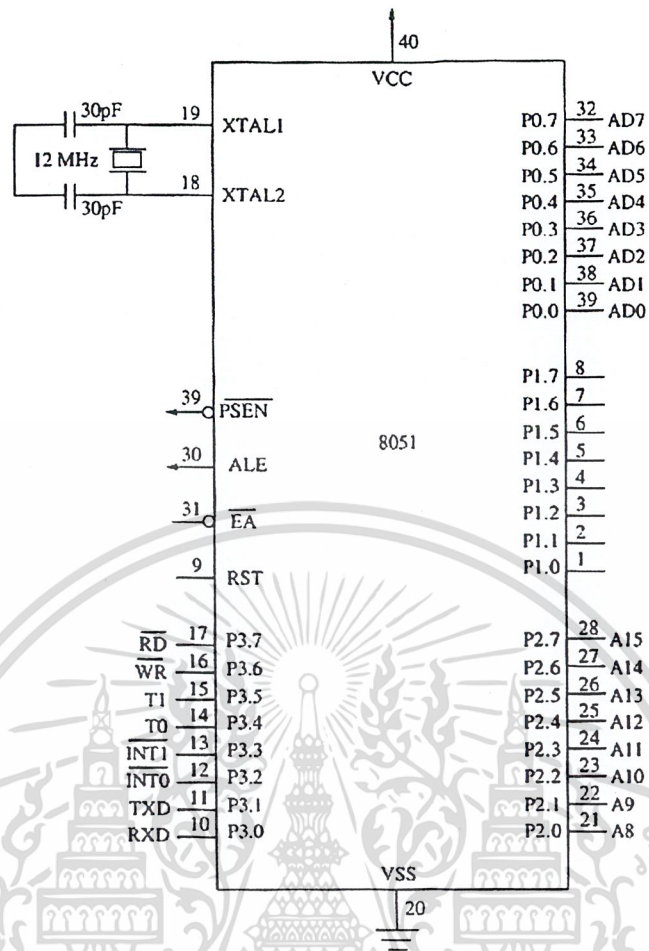


รูปที่ 2.1 โครงสร้างภายในชิปไมโครคอนโทรลเลอร์ 8051

สัญญาณภายในจะต่อออกสู่ภายนอกทางขาของ 8051 ที่มีอยู่ 40 ขา ดังรูป 2.2 ซึ่งมีการใช้งานต่าง ๆ ที่สำคัญดังนี้

- VCC ขา 40 เป็นขาต้องป้อนไฟเลี้ยง +5 โวลต์ เข้าไปเพื่อให้วงจรสามารถทำงานได้
- VSS ขา 20 เป็นขาที่ต้องต่อกับกราวด์ของแหล่งจ่ายไฟ
- RST ขา 9 ขาริเซตนี้จะริเซตการทำงานของ 8051 ถ้าป้อนสัญญาณที่มีสถานะลอจิก 1 ที่ขานี้จะเป็นการริเซตการทำงานกลับไปเริ่มทำงานจากคำสั่งที่อยู่ในหน่วยความจำตำแหน่ง 0000
- PSEN ขา 29 ใช้ส่งสัญญาณเพื่ออ่าน โปรแกรมซึ่งจะเก็บไว้ในหน่วยความจำภายนอกชิป
- XTAL1 ขา 19 ใช้ต่อคริสตอลภายนอก โดยที่จะใช้เพื่อเป็นอินพุตเข้าสู่วงจรออสซิลเลเตอร์
- XTAL2 ขา 18 ใช้ต่อคริสตอลภายนอก โดยที่จะใช้เพื่อเป็นเอาต์พุตออกจากวงจรออสซิลเลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 โค้ดแอมของ 8051

เมื่อเริ่มจ่ายพลังงานจะต้องมีการรีเซ็ตก่อนเสมอ ซึ่งการรีเซ็ตในขณะที่เริ่มจ่ายพลังงานจะกระทำโดยอัตโนมัติ โดยการทำงานของวงจรรีเซ็ตก็คือ เมื่อเริ่มมีการจ่ายพลังงานให้แก่ MCS-51 จะทำให้ขา RTS มีสถานะเป็นหนึ่งในช่วงเวลาหนึ่งซึ่งขึ้นอยู่กับค่าตัวเก็บประจุ ความเร็วในการประจุ เพื่อให้แน่ใจว่ามีการรีเซ็ตเกิดขึ้นจริง ในช่วงเวลาเริ่มต้นให้ขา VCC จะต้องให้ขา รีเซ็ต มีสถานะเป็น 1 นานพอที่จะให้วงจรรอสจิลเลเตอร์เริ่มต้นทำงานร่วมกับช่วงเวลา 2 เมกซ์ซินไซเคิล

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

2.1.2 พอร์ตอินพุท/เอาต์พุท

PORT 0 เป็นพอร์ตขนานขนาด 8 บิต อยู่ที่ขา 39 ถึง 32 เริ่มจากบิต 0 ถึงบิต 7 ตามลำดับ พอร์ต 0 นี้ยังใช้ได้ทั้งการรับส่งตำแหน่งและข้อมูลกับหน่วยความจำหรือใช้รับส่งข้อมูลก็ได้ นอกจากนี้ยังใช้งานได้หลายอย่างดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ใช้สำหรับส่งค่าตำแหน่งหน่วยความจำที่ต้องการติดต่อด้วย โดย 8 บิตล่าง ถูกส่งออกไปทางพอร์ท 0 และ 8 บิตบนถูกส่งออกไปทางพอร์ท 2
2. ใช้รับส่งข้อมูลกับ Data Memory หรือใช้รับข้อมูลจาก Program Memory
3. ใช้รับส่งข้อมูลออกจากพอร์ทโดยตรง

PORT 1 เป็นพอร์ทขนานขนาด 8 บิต อยู่ที่ขา 1 ถึง 8 เริ่มจากบิต 0 ถึง บิต 7 ตามลำดับใช้ทำหน้าที่เป็นตัวรับส่งข้อมูลเท่านั้น ไม่สามารถส่งตำแหน่งได้

PORT 2 เป็นพอร์ทขนานขนาด 8 บิต อยู่ที่ขา 21 ถึง 28 เริ่มจากบิต 0 ถึง บิต 7 ตามลำดับใช้งานเพียง 2 ลักษณะคือ

1. ใช้ส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องติดต่อกับพอร์ท 0
2. ใช้เป็นพอร์ทรับส่งข้อมูลภายนอก

PORT 3 เป็นพอร์ทขนานขนาด 8 บิตอยู่ที่ขา 10 ถึง 17 เริ่มจากบิต 0 ถึงบิต 7 ตามลำดับ นอกจากจะใช้งานเหมือนพอร์ทอื่นแล้วยังใช้งานอื่นโดยใช้คำสั่งควบคุมดังนี้

- P3.0 (RxD) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม
- P3.1 (TxD) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม
- P3.2 (INT0) ใช้รับสัญญาณขัดจังหวะจากภายนอก
- P3.3 (INT1) ใช้รับสัญญาณขัดจังหวะจากภายใน
- P3.4 (T0) ใช้เป็นขารับสัญญาณให้เคาเตอร์ของไทม์เมอร์ 0
- P3.5 (T1) ใช้เป็นขารับสัญญาณให้เคาเตอร์ของไทม์เมอร์ 1
- P3.6 (WR) ใช้เป็นขาสัญญาณควบคุมการเขียนข้อมูลยังหน่วยความจำสำหรับข้อมูลภายนอก
- P3.7 (RD) ใช้เป็นขาสัญญาณควบคุมการเขียนข้อมูลยังหน่วยความจำสำหรับข้อมูลภายนอก

2.1.3 รีจิสเตอร์สำหรับใช้งานทั่วไปใน MCS-51

รีจิสเตอร์ A, B และ รีจิสเตอร์ใช้งานทั่วไป 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 ตัว ในการทำงานขณะใด ๆ รีจิสเตอร์ทั้ง 4 กลุ่ม (R0-R7) นั้นจะถูกเลือกใช้งานเพียงกลุ่มเดียวเท่านั้น การเลือกใช้งานของตัวรีจิสเตอร์ R0-R7 กลุ่มใดกลุ่มหนึ่งใน 4 กลุ่ม นั้นจะกระทำโดยการเซตหรือเคลียร์บิต RS0, RS1

ในรีจิสเตอร์ใช้งานเฉพาะ PSW

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 ไทม์เมอร์/เคาน์เตอร์

ไมโครคอนโทรลเลอร์ในตระกูล 8051 มีรีจิสเตอร์พิเศษที่สามารถเลือกใช้งานเป็นไทม์เมอร์หรือเคาน์เตอร์อย่างใดอย่างหนึ่ง รีจิสเตอร์ประเภทนี้มีอยู่ด้วยกัน 2 ตัว แต่ละตัวขนาด 16 บิต เรียกไทม์เมอร์ 0 และไทม์เมอร์ 1 ตามลำดับ

ไทม์เมอร์นั้นค่าในรีจิสเตอร์ที่ใช้เป็นไทม์เมอร์ที่ถูกเลือกใช้งานจะถูกเพิ่มขึ้นทุกเมกซ์ซินไซเคิล

เคาเตอร์นั้นค่าในรีจิสเตอร์ที่ใช้เป็นเคาเตอร์ที่ถูกเลือกใช้งานจะถูกเพิ่มค่าทีละ 1 เมื่อมีการเปลี่ยนแปลงสถานะ

ไทม์เมอร์ 0 และไทม์เมอร์ 1

สามารถเลือกการทำงานให้เป็นไทม์เมอร์หรือเคาเตอร์ได้โดยการใ้การกำหนดค่าบิต ในรีจิสเตอร์ใช้งานเฉพาะ ดังรูปที่ 2.3 (ไทม์เมอร์ 0 ใช้บิต 2 ส่วนไทม์เมอร์ 1 ใช้บิต 6) โดยหากบิตนี้มีค่าเป็น 0 หมายถึงเลือกใช้งานเป็นไทม์เมอร์ถ้าบิตนี้มีค่าเป็น 1 หมายถึงเลือกใช้งานเป็นเคาเตอร์

GATE1	C/T1	M1	M0	GATE0	C/T0	M1	M0
-------	------	----	----	-------	------	----	----

รูปที่ 2.3 รีจิสเตอร์ใช้งานเฉพาะ TMOD

นอกจากจะเลือกการทำงานของรีจิสเตอร์ให้ไทม์เมอร์หรือเคาน์เตอร์ได้แล้วในแต่ละการทำงานยังมีการทำงานย่อยอยู่อีก 4 แบบ ตามความเหมาะสมของการใช้งาน

โหมด 0 จะใช้รีจิสเตอร์ขนาด 8 บิต เป็นตัวนับโดยมีการเมค่าครั้งละ 1 ทุกครั้งนับสัญญาณได้ครบ 32 ครั้ง โดยในโหมดนี้มีรีจิสเตอร์ที่ใช้นับเพียง 13 บิต (8 บิตในรีจิสเตอร์ TLx รวมกับ 5 บิตใน THx)

โหมด 1 การทำงานเหมือนโหมด 0 เว้นแต่ค่าในรีจิสเตอร์ถูกใช้งานครบทั้ง 16 บิต นั่นเองคือไทม์เมอร์หรือเคาน์เตอร์ในโหมดนี้มีขนาด 16 บิต


โหมด 2 ในโหมดนี้จะกำหนดรีจิสเตอร์ใช้งานในการนับบิตเพียง 8 บิต (จากรีจิสเตอร์ TLx) ที่มีการโหลดค่าด้วยค่าในรีจิสเตอร์ THx การที่จะใช้งานในโหมดนี้นั้นมีไว้เพื่อสร้างสัญญาณอินเตอร์รัปต์ที่มีคาบเวลาคงที่

โหมด 3 ในโหมดไทม์เมอร์ 1 จะไม่มีการยับแต่ไทม์เมอร์จะบังคับให้รีจิสเตอร์ TLO ของไทม์เมอร์ 0 ถูกใช้เป็นไทม์เมอร์เพียงอย่างเดียว ซึ่งการทำงานโหมด 3 นี้มีไว้เพื่อใช้งานที่ต้องการไทม์เมอร์หรือเคาน์เตอร์ขนาด 8 บิตเพิ่มขึ้น

2.1.5 การอินเทอร์รัปต์

คือการขัดจังหวะโปรแกรมชั่วคราวแล้วมาทำโปรแกรมบริการอินเทอร์รัปต์ (Interrupt Service Routine : ISR) อินเทอร์รัปต์ของ MCS-51 นั้นได้มาจาก 8 แหล่ง แต่จะมีเพียง 6 เวกเตอร์ (TI และ RI ใช้ตำแหน่งเดียวกัน) ตำแหน่งของอินเทอร์รัปต์เวกเตอร์แสดงดังตารางที่ 2.1

ตารางที่ 2.1 แสดงลำดับความสำคัญของการอินเทอร์รัปต์

ลำดับ	ชื่อสัญญาณอินเทอร์รัปต์	Vector Address	Priority
1	$\overline{\text{INT0}}$	003H	Highest  Lowest
2	TF0	000BH	
3	$\overline{\text{INT1}}$	0013H	
4	TF1	001BH	
5	TI+RI	0023H	
6	TF2+EXF2	002BH	

2.1.6 การรับส่งข้อมูลอนุกรม (UART) ของ MCS-51

พอร์ตสื่อสารอนุกรมของ MCS-51 มีโครงสร้างการทำงานของการทำงานสื่อสารที่เป็นแบบที่เรียกว่า ฟูลดูเพล็กซ์ (Full Duplex) ในการรับและส่งข้อมูลอนุกรมได้ในเวลาเดียวกัน โดยทางด้านส่งใช้ขา TxD (พอร์ต 3.1) ทางด้านรับใช้ขา Rx D (พอร์ต 3.0) SBUF ใช้เป็นบัฟเฟอร์สำหรับรับและส่งข้อมูลอนุกรม

พอร์ตสื่อสารอนุกรมของ MCS-51 สามารถโปรแกรมการทำงานได้หลายโหมดด้วยกัน โดยเลือกที่บิต SM1 และ SM0 ซึ่งอยู่ในรีจิสเตอร์ควบคุม (SCON) การทำงานทั้ง 4 โหมดของพอร์ตสื่อสารอนุกรมมีดังนี้

โหมด 0 พอร์ตสื่อสารอนุกรม 8 บิต โดยการส่งจะเลื่อนออกทีละบิต โดยส่งบิต D0 ออกไปก่อนทางขา Rx D และไม่มีการส่งบิตเริ่มต้น แต่จะส่งสัญญาณนาฬิกาเลื่อนทางขา Tx D

โหมด 1 พอร์ตสื่อสารอนุกรม 10 บิต ข้อมูล 8 บิต 1 บิตเริ่มต้น และ 1 บิตหยุด และสามารถเปลี่ยนแปลงความเร็วในการส่งข้อมูลได้ โดยขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โพล์ของไทม์เมอร์ 1

โหมด 2 พอร์ตสื่อสารอนุกรม 11 บิต ข้อมูล 9 บิต 1 บิตเริ่มต้น และ 1 บิตหยุด

โหมด 3 พอร์ตสื่อสารอนุกรม 11 บิต UART โดยส่งข้อมูล 9 บิต 1 บิตเริ่มต้น และ 1 บิตหยุด อัตราเร็วในการรับส่งข้อมูลที่ ไทม์เมอร์ 1, 2 และ SMOD

2.2 ความรู้เบื้องต้นเกี่ยวกับพอร์ทอนุกรม

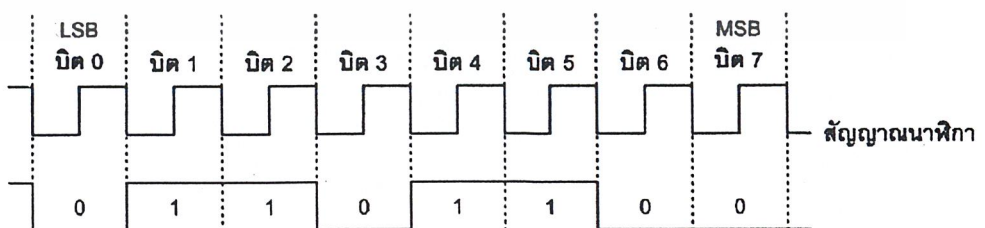
การเคลื่อนที่ย้ายข้อมูลจากคอมพิวเตอร์ไปยังอุปกรณ์ต่อพ่วงภายนอกหรือคอมพิวเตอร์ด้วยกัน มีด้วยกัน 2 รูปแบบคือ รับส่งข้อมูลแบบขนานและรับส่งข้อมูลแบบอนุกรม

การรับส่งข้อมูลแบบขนานเป็นการรับหรือส่งข้อมูลคราวละ 4 หรือ 8 บิตในเวลาเดียวกัน ทำให้การรับและส่งข้อมูลมีความเร็วสูง ทว่าจำนวนของรีจิสเตอร์ที่ใช้ในการถ่ายทอดข้อมูลต้องมามากเท่ากับจำนวนบิตของข้อมูลที่ทำการถ่ายทอดด้วย นอกจากนี้ยังมีสายที่ใช้สำหรับควบคุมและตรวจสอบการรับส่งข้อมูลด้วย ซึ่งอาจต้องใช้สายมากเป็น 2 เท่าของจำนวนบิตข้อมูลก็ได้ ส่งผลให้ราคาของสายที่ใช้ในการเชื่อมต่อแบบขนานมักจะมีราคาแพง และอีกข้อจำกัดหนึ่งของการถ่ายทอดข้อมูลแบบขนานคือ ระยะทางในการถ่ายทอดข้อมูล โดยปกติจะอยู่ที่ประมาณ 10-15 ฟุต

ในขณะที่การรับส่งข้อมูลแบบอนุกรมจะเป็นการรับส่งข้อมูลครั้งละ 1 บิต โดยมีรูปแบบการรับเป็นมาตรฐาน ต้องมีการตรวจสอบความพร้อมในการรับส่งข้อมูลของตัวรับและตัวส่ง ซึ่งการรับส่งข้อมูลแบบอนุกรมนั้นจะมีข้อดีในเรื่องของจำนวนสายสัญญาณที่น้อยมากและไม่แปรผันตามจำนวนบิตของข้อมูล ระยะทางการรับส่งสูงกว่าแบบขนานมาก โดยปกติถ้าเป็นพอร์ทอนุกรมแบบ RS-232 จะสามารถต่อสายได้ยาว 50 ฟุตโดยประมาณ

2.2.1 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมแบ่งได้เป็น 2 แบบคือ การสื่อสารอนุกรมแบบซิงโครนัสและการสื่อสารอนุกรมแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิกาอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือคีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้น คือสัญญาณนาฬิกามีข้อมูลและกราวด์ รูปที่ 2.4 แสดงให้เห็น ไคอะแกรมเวลาของการสื่อสารข้อมูลแบบซิงโครนัส



รูปที่ 2.4 ไคอะแกรมเวลาของการสื่อสารข้อมูลแบบซิงโครนัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 การสื่อสารข้อมูลแบบอะซิงโครนัส

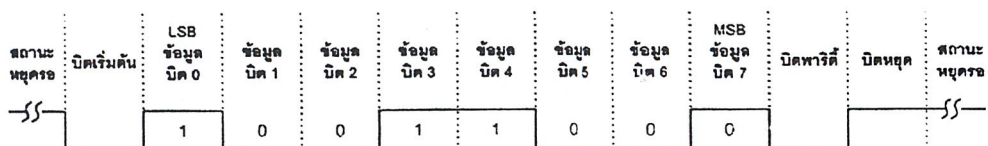
การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับส่งข้อมูลโดยไม่จำเป็นต้องมีสัญญาณนาฬิกาพร้อมด้วย แต่จะใช้การกำหนดค่าอัตราเร็วในการรับและส่งข้อมูลให้มีค่าเท่ากัน ซึ่งเรียกอัตราเร็วนี้ว่าอัตราบอดหรือบอดเรต (Baud rate) ซึ่งจะมีหน่วยเป็นบิตต่อวินาที (bit per second: BPS)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกันคือ

1. บิตเริ่มต้น (Start bit) มีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรม มีขนาด 5,6,7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี (Parity bit) มีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้ายหรือบิตหยุด (Stop bit) มีขนาด 1,1.5 หรือ 2 บิต

รูปที่ 2.5 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส เมื่อไม่มีการส่งข้อมูล ขา DATA จะมีสถานะลอจิก “1” เรียกสถานะนี้ว่า สถานะหยุดรอ (waiting stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการ ให้ขา DATA มีลอจิก “0” ด้วยช่วงระยะเวลา 1 บิต ซึ่งเรียก 1 บิตนี้ว่าเป็นบิตเริ่มต้น (start bit) จากนั้นบิตข้อมูลจะถูกส่งออกไปโดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุดหรือก็คือบิต LSB ก่อนซึ่งข้อมูลที่ต้องการส่งอาจมีจำนวน 5,6,7 หรือ 8 บิตก็ได้ จากนั้นตามด้วยบิตพาริตี (parity bit) ซึ่งใช้ในการตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่จะส่งคือบิตปิดท้ายหรือบิตหยุด โดยเป็นการทำให้ขา DATA มีสถานะลอจิก “1” อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต,1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัสหรืออัตราบอดหรือบอดเรตที่ใช้สำหรับพอร์ทอนุกรม RS-232 มีด้วยกันหลายค่า ได้แก่ 110 ,150 ,300 ,600 , 1200 ,2400 ,4800 ,9600 และ 19200 บิตต่อวินาที โดยมีค่าเพิ่มขึ้นตามเทคโนโลยีของคอมพิวเตอร์ เนื่องจากบอดเรต คือค่าของจำนวนบิตที่สามารถส่งได้ใน 1 วินาที สมมุติว่าข้อมูลอนุกรมมีขนาด 8 บิต ไม่มีการตรวจสอบพาริตี มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวที่ข้อมูล 1 ไบต์จะมีความยาวเท่ากับ 10 บิต ถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที



รูปที่ 2.5 รูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการเชิงงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจสอบพาริตีสามารถกำหนดให้เป็นแบบคี่ (Odd) แบบคู่ (Even) หรือ ไม่มีการตรวจสอบพาริตีก็ได้ พาริตีคี่หรือพาริตีคู่แสดงถึงจำนวนลอจิก “1” ทั้งหมดภายในข้อมูลที่ส่งไป 1 ไบต์ รวมบิตพาริตีว่าจำนวนเลขคู่หรือเลขคี่ ยกตัวอย่าง ข้อมูลที่จะทำการส่งมีขนาด 8 บิต มีค่าเท่ากับ 99H หรือ 10011001B จะเห็นว่าข้อมูลในไบต์นี้มีจำนวนลอจิก “1” จำนวน 4 ตัวซึ่งเป็นเลขคู่ ดังนั้น ถ้ากำหนดค่าพาริตีเป็นคู่ ค่าของพาริตีจะต้องมีลอจิกเป็น “0” แต่ถ้ากำหนดพาริตีเป็นคี่ ค่าของบิตพาริตีจะต้องเป็น “1” เพื่อให้ข้อมูล 1 ไบต์รวมทั้งพาริตีเป็นคี่

บิตพาริตีถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART (Universal Asynchronous Receiver Transmitter: เป็นอุปกรณ์ที่ใช้ในการรับส่งข้อมูลอนุกรม ซึ่งจะกล่าวถึงในรายละเอียดนี้ ภายหลัง) ซึ่งทางภาครับจะต้องกำหนดคุณสมบัติในการตรวจสอบพาริตีที่ตรงกันไว้ว่าจะตรวจสอบพาริตีคี่หรือพาริตีคู่ จากนั้นภาครับของ UART จะทำการตรวจสอบค่าพาริตีที่เกิดขึ้นว่าเป็นคู่หรือคี่ โดยการนับจำนวนลอจิก “1” ทั้งหมดรวมทั้งบิตพาริตีด้วย ถ้ากำหนดพาริตีไว้เป็นคู่แต่อ่านค่าในการนับตัวเลขออกมาเป็นคี่ ทางภาครับจะแสดงข้อผิดพลาดออกมาให้ผู้ใช้ทราบ กระบวนการดังกล่าวเป็นวิธีการตรวจสอบความผิดพลาดที่เกิดขึ้นในการรับส่งข้อมูลที่ง่ายที่สุด แต่มันสามารถตรวจได้เมื่อมีบิตข้อมูลที่ทำให้การรับส่งผิดพลาดเพียงบิตเดียวเท่านั้น ถ้าข้อมูลที่ทำให้การส่งมีบิตที่ผิดพลาดมากกว่า 1 บิต การตรวจสอบด้วยวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตีบิตเป็น NONE นั้นทั้งภาครับและภาคส่ง จะไม่มีการตรวจสอบพาริตี

คอมพิวเตอร์ในรุ่น AT เกือบทั้งหมดจะใช้ไอซี UART เบอร์ 16450 และ 16550 ส่วนคอมพิวเตอร์ในรุ่น XT ใช้ไอซี UART เบอร์ 8250 ไอซี UART เหล่านี้มีระดับแรงดันของลอจิกเป็นแบบทีทีแอล (+5V) แต่เพื่อให้มีแรงดันเป็นไปตามมาตรฐาน RS-232 และเพื่อให้การรับส่งข้อมูลสามารถทำได้ในระยะทางไกลมากขึ้น ระดับแรงดันที่ทีทีแอลจะถูกแปลงเป็นระดับแรงดันที่สูงขึ้น โดยลอจิก “0” จะมีระดับแรงดัน -3V ถึง -12V และลอจิก “1” มีระดับแรงดัน +3V ถึง +12V

2.2.3 มาตรฐานพอร์ทอนุกรมแบบ RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลแบบอะซิงโครนัส 2 ทิศทาง โดยมาตรฐาน RS-232 ในอดีตนั้นถูกออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจากโมเด็มนี้ส่งผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดซึ่งอยู่ห่างไกลกัน โดยสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronics Industries Association: EIA) ได้วางมาตรฐานที่มีชื่อเรียกกันว่า EIA RS-232 มาตรฐานนี้ในช่วงแรกจะใช้คอนเน็กเตอร์เป็นแบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ -3V ถึง -12V แสดงว่ามีข้อมูล(mark) และ +3V ถึง +12V แสดงว่าเป็นช่องว่าง (space)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรฐาน RS-232 ถูกใช้ในการกำหนดรูปแบบการสื่อสารข้อมูลกันระหว่าง อุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment: DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating: DCE) ซึ่งในพวกอุปกรณ์ DTE นั้นจะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัว เช่น ไมโครคอนโทรลเลอร์หรือไมโครคอมพิวเตอร์ ซึ่งจะมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE ทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น

ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่เราเห็นได้ชัดคือ คอนเน็กเตอร์ ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็กเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ทอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็กเตอร์ที่อยู่โมเด็มจะเป็นแบบ DCE สำหรับการใช้งานในคอมพิวเตอร์ พอร์ทอนุกรม RS-232 ถูกใช้เพื่อเชื่อมต่อกับ โมเด็ม, แมส และเครื่องพิมพ์ที่สามารถติดต่อทางพอร์ทอนุกรมได้

2.2.4 คอนเน็กเตอร์สำหรับพอร์ท RS-232 และการเชื่อมต่อ

มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็กเตอร์แบบ DB-25 ตัวผู้ หรือ DB-9 ตัวผู้ ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาต่อใช้งาน เพียง 9 เส้นเช่นเดียวกับคอนเน็กเตอร์แบบ DB-9 เนื่องจากขาอื่น ๆ ที่เคยมีการใช้งานมาในอดีตไม่ค่อยมีความสำคัญมากนักจึงถูกยกเลิกไป โดยแสดงรูปร่างและตำแหน่งขา ดังรูปที่ 2.6

ขา Data Carrier Detect: DCD หรืออาจเรียกว่า Carrier Detect: CD ขานี้จะ แยกดีฟเมื่อมีการส่งสัญญาณพาร์จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม

ขา Receive Data: RD หรือ RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายัง คอมพิวเตอร์ โดยจะนำข้อมูลที่อ่านได้ไปเก็บไว้ในรีจิสเตอร์บัฟเฟอร์

ขา Transmitted: TD หรือ TxD ขานี้ใช้เพื่อส่งข้อมูลอนุกรมออกจาก คอมพิวเตอร์ โดยการนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลออกไป

ขา Data Terminal Ready: DTR เป็นขาเอาท์พุทที่ใช้สำหรับส่งสัญญาณออก จากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าการติดต่อกับอุปกรณ์ปลายทาง โดย ขา DTR นี้จะต้องเชื่อมต่อกับขา DDSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์และถ้าใช้การเชื่อมต่อแบบ 3 สาย ต้องเชื่อมต่อกับขา DTR และ DSR ของพอร์ทอนุกรมเข้าด้วยกัน และจะต้องต่อเชื่อมกับขา DCD ด้วยในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาร์

ขา Signal Ground: GND เป็นขากราวด์ของสัญญาณ

ขา Data Set Ready: DSR ขานี้จะใช้ควบคุม DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับข้อมูลจาก ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา Request To Send: RTS เป็นขาเอาต์พุตสำหรับส่งสัญญาณร้องขอให้อุปกรณ์ปลายทางส่งข้อมูลมาให้คอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ซึ่งในกรณีที่มีการเชื่อมต่อแบบ 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS เข้าด้วยกัน เพื่อให้การรับและส่งข้อมูลเกิดขึ้นได้ตลอดเวลา

ขา Clear To Send: CTS เป็นขาอินพุตทำหน้าที่รอรับสัญญาณที่ส่งเข้ามา เมื่อมีการส่งสัญญาณเข้ามาที่ขานี้ ข้อมูลที่ขา TxD จะถูกส่งออกไป ขานี้ใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่รับข้อมูลแล้วหรือยัง

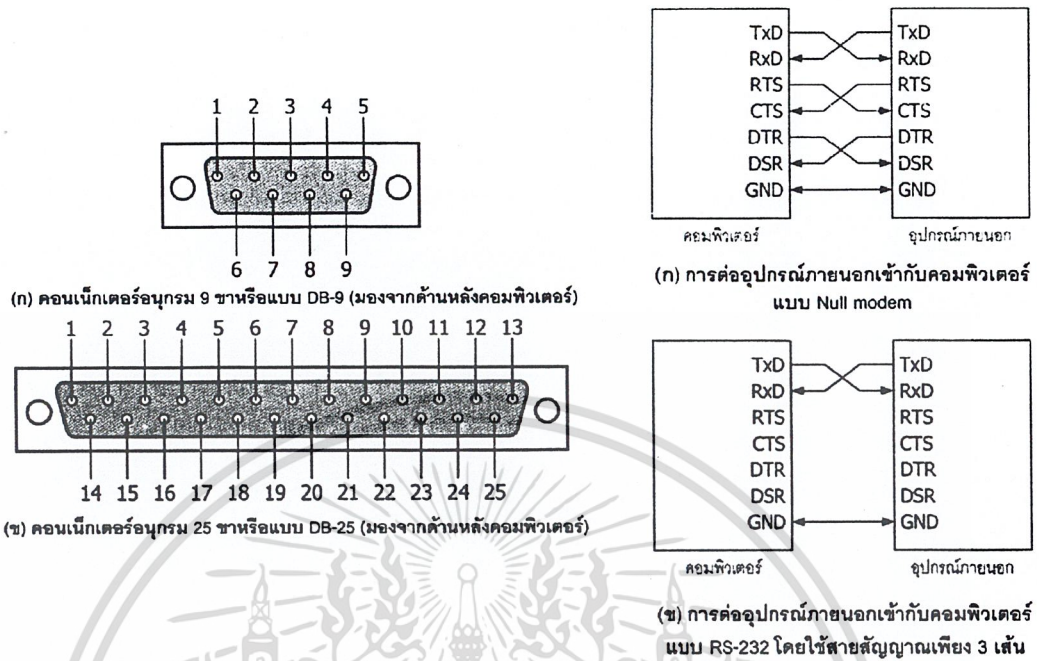
ขา Ring Indicator: RI ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็มแล้วยังมีความต้องการตรวจสอบสัญญาณเรียกจากสายโทรศัพท์

สำหรับการเชื่อมต่อสายระหว่างคอมพิวเตอร์กับอุปกรณ์ภายนอกแสดงผังรูปที่ 2.6 ลูกศรในรูปแสดงถึงทิศทางของข้อมูล การเชื่อมต่อในรูปที่ 2.6 เป็นการเชื่อมต่อแบบ NULL MODEM หรือการเชื่อมต่อโดยใช้สายสัญญาณน้อยที่สุดเพียง 3 เส้น โดยเส้นหนึ่งสำหรับส่งข้อมูล อีกเส้นสำหรับรับข้อมูลและอีกเส้นเป็นกราวด์

ตารางที่ 2.2 การจัดขาสัญญาณของพอร์ตอนุกรมในแบบต่างๆ และหน้าที่การทำงาน

คอนเน็กเตอร์ DB-25	คอนเน็กเตอร์ DB-25	ชื่อของสายสัญญาณ	ชนิดของสาย สัญญาณ
1	8	Data Carrier Detect: DCD	อินพุต
2	3	Received DATA: TxD	อินพุต
3	2	Transmitted Data: TxD	เอาต์พุต
4	20	Data Terminal Ready: DTR	เอาต์พุต
5	7	Signal Ground: GND	-
6	6	Data Set Ready: DSR	อินพุต
7	4	Request To Send: RTS	เอาต์พุต
8	5	Clear To Send: CTS	อินพุต
9	22	Ring Indicator: RI	อินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 ขาสัญญาณและการเชื่อมต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ผ่าน RS-232

2.2.5 UART (Universal Asynchronous Receiver Transmitter)

UART มาจากคำว่า Universal Asynchronous Receiver Transmitter ซึ่งจะหมายถึงอุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัสนั่นเอง สำหรับการสื่อสารอนุกรมบนคอมพิวเตอร์แล้ว UART ถือว่าเป็นหัวใจสำคัญของการสื่อสารอนุกรม

หน้าที่หลักของ UART คือแปลงข้อมูลที่อยู่ในรูปแบบขนานจากซีพียูให้อยู่ในรูปแบบอนุกรมแบบอะซิงโครนัส แล้วจะทำการส่งออกไป และทำการแปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนานก่อนที่จะส่งเข้าซีพียู ซึ่งนอกจาก UART จะส่งข้อมูลไปยังซีพียูแล้ว ยังแจ้งรายละเอียดอื่น ๆ ของข้อมูลให้คอมพิวเตอร์รับทราบอีกด้วย อาทิ อัตราเร็วในการรับส่งข้อมูลหรือบอดเรต, รูปแบบการส่งข้อมูล, ความผิดพลาดที่เกิดระหว่างการส่งข้อมูล เช่น ผิดพลาดจากพาริตี, เฟรมข้อมูล, โอเวอร์รัน เป็นต้น

ภายใน UART จะมีวงจรสร้างบอดเรตโปรแกรมได้ (Programmable Baudrate Generator) โดยกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของของ UART โดยตัวหารนี้จะมีขนาด 16 บิต ดังนั้นจะสามารถกำหนดตัวหารอยู่ในช่วง 1-65,535

บทที่ 3

การออกแบบและการสร้าง

3.1 รายละเอียดเกี่ยวกับโครงการ

โครงการนี้จะออกแบบและสร้างหุ่นยนต์สำรวจที่สามารถเคลื่อนที่ได้อย่างแม่นยำ โดยสามารถระบุตำแหน่งที่แน่นอนของการเคลื่อนที่ได้ สามารถควบคุมการเคลื่อนที่ได้ง่าย สามารถสั่งให้เคลื่อนที่ไปข้างหน้า หรือว่าถอยหลังได้ในความละเอียดของหน่วยเป็นมิลลิเมตร สามารถสั่งให้เลี้ยวได้ในความละเอียดของหน่วยเป็นองศา มีการเก็บข้อมูลของการเคลื่อนที่ที่ผ่านมาอย่างละเอียด พร้อมทั้งสร้างแผนที่ของเส้นทางการเคลื่อนที่ที่หุ่นยนต์เคลื่อนที่ผ่านไปได้

ลักษณะของการเคลื่อนที่โดยเฉพาะในส่วนของกาเลี้ยวที่ออกแบบมาจะแตกต่างจากหุ่นยนต์เคลื่อนที่โดยล้อที่พบเห็นได้ทั่วไป เนื่องจากออกแบบโดยคำนึงถึงความแม่นยำในการเคลื่อนที่เป็นสำคัญ โดยจะใช้การหมุนของหุ่นยนต์แทนการหมุนเลี้ยวของล้อหรือการหมุนสวนทางของล้อ ซึ่งจากสมมุติฐานในการออกแบบนั้นจะสามารถเคลื่อนที่ได้ละเอียดมากกว่า แต่อาจจะต้องแลกกับเวลาในการเคลื่อนที่ที่เพิ่มมากขึ้นซึ่งจะยังไม่สนใจในขณะนี้

การเคลื่อนที่ของหุ่นยนต์จะขับเคลื่อนโดยมอเตอร์ไฟฟ้ากระแสตรง ส่งกำลังผ่านเฟืองทดไปยังเพลาชักขับเคลื่อน ตรวจสอบและควบคุมตำแหน่งโดยใช้การเอ็นโคดเดอร์ (Encoder) โดยจะนับจำนวนของรอบการหมุนของมอเตอร์แล้วส่งข้อมูลไปยังไมโครคอนโทรลเลอร์ (Microcontroller) ที่อยู่บนตัวหุ่นยนต์ เมื่อถึงตำแหน่งที่ต้องการก็จะหยุดมอเตอร์ โดยจะมีการเพิ่มความแม่นยำโดยการทำการทดลองหาค่าความผิดพลาดเนื่องจากการความถี่ของมอเตอร์ในขณะที่หยุดหมุนและนำค่าที่ได้ไปชดเชยความผิดพลาดด้วยโปรแกรม ซึ่งจะทำให้หุ่นยนต์สามารถเคลื่อนที่ได้ในละเอียดสูงถึง 1 มิลลิเมตร และสามารถหมุนได้ในความละเอียดถึง 1 องศา

การควบคุมจะควบคุมด้วยคอมพิวเตอร์ซอฟต์แวร์บนไมโครคอมพิวเตอร์ โดยจะติดต่อกันกับหุ่นยนต์ผ่านทางพอร์ทอนุกรม RS-232 ด้วยข้อมูลสื่อสารขนาดความยาว 8 ตัวอักษร ด้วยซอฟต์แวร์ที่สร้างขึ้นด้วยภาษาวิซวลเบสิก (Visual Basic) ซึ่งทำงานบนระบบปฏิบัติการไมโครซอฟท์วินโดวส์ (Microsoft Windows) โดยในส่วนตัวโปรแกรมจะทำหน้าที่ควบคุมระบบการเคลื่อนที่ทั้งหมดของหุ่นยนต์ สามารถสั่งการเคลื่อนที่ของหุ่นยนต์ได้ง่าย สามารถระบุหน่วยของการเคลื่อนที่ได้โดยการเคลื่อนที่ที่จะสามารถระบุได้ว่าจะให้เคลื่อนที่ไปที่เซ็นติเมตร กี่นิ้ว หรืออื่นๆ โดยที่ความละเอียดสูงสุดคือจำนวนซีของเอ็นโคดเดอร์ที่นับได้ในการเคลื่อนที่ของมอเตอร์ โดยโปรแกรมจะเก็บรายละเอียดของการเคลื่อนที่ที่ผ่านมาพร้อมทั้งสร้างเป็นแผนที่ให้เห็นภาพรวมของการเคลื่อนที่ เมื่อต้องการเคลื่อนที่ไปยังจุดใดของการเคลื่อนที่ สามารถสั่งการได้ทันทีโดยการ

ใช้เมาส์คลิกที่แผนที่ ซึ่งตัวโปรแกรมก็จะบอกพิกัดการเคลื่อนที่มาให้ ซึ่งเราสามารถสั่งให้หุ่นยนต์
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เคลื่อนที่กลับไปยังจุดที่ต้องการได้อย่างแม่นยำเป็นเส้นตรงโดยไม่ต้องอ้อมกลับไปตามเส้นทางทางเดิมที่เคลื่อนที่ผ่านมา

3.2 การออกแบบหุ่นยนต์

โครงการ “หุ่นยนต์สำรวจเคลื่อนที่ควบคุมการทำงานด้วยไมโครโปรเซสเซอร์” เป็นโครงการที่พัฒนาต่อเนื่องมาจากโครงการของนักศึกษาภาควิชาวิศวกรรมการวัดคุม ในปีการศึกษา ก่อนหน้านี้ โดยการพัฒนาให้ความสำคัญกับการควบคุมการเคลื่อนที่ให้ละเอียดมากขึ้น และพัฒนา ให้โปรแกรมควบคุมสามารถใช้งานได้ง่ายพร้อมทั้งมีประสิทธิภาพสูงยิ่งขึ้น การออกแบบจึงแบ่ง เป็นสองส่วนคือ

1. ฮาร์ดแวร์ คือ โครงสร้างของหุ่นยนต์ ชุดขับเคลื่อน รวมไปถึงวงจรทั้งหมด
2. ซอฟต์แวร์ คือ โปรแกรมควบคุมตัวหุ่นยนต์ต่างๆ ไม่ว่าจะเป็น โปรแกรมที่ทำงาน บนไมโครคอมพิวเตอร์ หรือว่าโปรแกรมที่ทำงานบนตัวหุ่นยนต์

3.2.1 แนวคิดในการออกแบบ

คำว่า “หุ่นยนต์” หมายถึงเครื่องจักรที่สามารถเคลื่อนไหวได้โดยจะสามารถทำงานได้ตามความต้องการของผู้สร้าง ซึ่งในการออกแบบหุ่นยนต์สำหรับโครงการนี้นั้นต้องการให้มีความสามารถในการเคลื่อนที่ได้แม่นยำเป็นสำคัญ จึงได้ศึกษาการเคลื่อนที่ของหุ่นยนต์ที่เคลื่อนที่ด้วยล้อ พบว่าการเคลื่อนที่ด้วยล้อแต่ละชนิดนั้นจะแตกต่างกันที่วิธีการเปลี่ยนทิศทางของการเคลื่อนที่ ซึ่งจากการศึกษาการเคลื่อนที่ลักษณะต่างๆ สามารถสรุปลักษณะของการเปลี่ยนทิศทางในการเคลื่อนที่ได้เป็น 2 ลักษณะ คือ การเปลี่ยนทิศทางในลักษณะการเลี้ยว และการเปลี่ยนทิศทางในลักษณะการหมุน

3.2.2 การเปลี่ยนทิศทางในลักษณะการเลี้ยว

การเลี้ยวจะเปลี่ยนทิศทางเคลื่อนที่โดยอาศัยการเคลื่อนที่เป็นวิถีโค้ง ซึ่งวิธีการเลี้ยวอาจจะแตกต่างออกไปได้ตามการออกแบบใช้งาน โดยการเลี้ยวแบ่งเป็นสองวิธีคือ ใช้การหมุนล้อ และใช้ความแตกต่างของความเร็วระหว่างล้อสองล้อ

1. การหมุนล้อ โดยพื้นฐานนั้นจะเป็นลักษณะเดียวกับการเคลื่อนที่ของรถยนต์ ซึ่งมักจะใช้ล้อ 4 ล้อ ซึ่งจะมีล้อ 1 คู่เป็นล้อที่จะหมุนเลี้ยวไปตามทิศทางที่ต้องการเคลื่อนที่ไป (ล้อเป็น) และอีก 2 คู่ จะถูกยึดอยู่กับที่ (ล้อตาย) โดยการขับเคลื่อนจะส่งกำลังไปที่ล้อใดก็ได้ ขึ้นอยู่กับ การออกแบบ ข้อดีของวิธีนี้คือใช้ชุดมอเตอร์ขับเคลื่อนเพียงชุดเดียว จึงเหมาะกับงานที่ต้องมีการ ยก หรือบรรทุกสิ่งของที่อาจจะมีน้ำหนักค่อนข้างมาก ซึ่งต้องใช้ชุดมอเตอร์ที่มีขนาดใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ใช้ความแตกต่างของความเร็วระหว่างล้อสองล้อ วิธีนี้จะใช้ล้อขับเคลื่อน 2 ล้อซึ่งจะมีการขับเคลื่อนแยกกัน เมื่อต้องการเลี้ยวก็จะให้ล้อหนึ่งมีความเร็วมากกว่าอีกล้อหนึ่ง เมื่อความเร็วแตกต่างกันก็จะเกิดการเคลื่อนที่เป็นเส้นโค้ง วิธีนี้จะให้ความแม่นยำมากกว่าวิธีแรก แต่จะต้องใช้ชุดขับเคลื่อนแยกกันสองชุด

ข้อดีการเคลื่อนที่ลักษณะนี้คือ สามารถทำงานได้อย่างรวดเร็ว เพราะไม่ต้องหยุดการเคลื่อนหรือลดความเร็วเมื่อจะทำการเปลี่ยนเส้นทาง แต่จะมีข้อเสียคือ ต้องใช้พื้นที่ในการเลี้ยวมาก และความแม่นยำในการเคลื่อนที่นั้นก็จะต่ำ เพราะการเคลื่อนที่ในแนวโค้งนั้นจะทำการคำนวณได้ยาก และจะมีความผิดพลาดจากการลื่นไถล (Slip) ของล้อตายหรือเพราะการที่ล้อที่มีความเร็วต่างกันขณะที่กำลังเลี้ยว ทำให้คาดการณ์องศาที่เลี้ยวไปได้ยาก จึงเหมาะกับการควบคุมด้วยมือ (Manual) มากกว่าการควบคุมแบบอัตโนมัติ (Automatic) ซึ่งต้องควบคุมการเคลื่อนที่ด้วยตัวเลข

3.2.3 การเปลี่ยนทิศทางในลักษณะการหมุน

การหมุนจะใช้เปลี่ยนทิศทางโดยการหมุนตัวของหุ่นยนต์ ซึ่งวิธีนี้จะแบ่งวิธีการหมุนตัวเป็นสองประเภทคือ การหมุนโดยใช้ความแตกต่างระหว่างความเร็วระหว่างล้อสองล้อ และการใช้กลไกส่วนอื่นมาช่วยในการหมุนตัว

1. การหมุนโดยใช้ความแตกต่างระหว่างความเร็วระหว่างล้อสองล้อ วิธีนี้จะใช้ล้อขับเคลื่อน 2 ล้อซึ่งจะมีการขับเคลื่อนแยกกันซึ่งจะมีลักษณะคล้ายคลึงกับวิธีการเคลื่อนที่ของรถตีนตะขาบ เมื่อต้องการหมุนตัวจะหยุดการเคลื่อนที่ และหมุนล้อทั้งสองสวนทางกัน ก็จะทำให้เกิดการหมุนขึ้น โดยตำแหน่งที่เป็นแกนหมุนก็คือจุดกึ่งกลางระหว่างล้อทั้งสอง วิธีนี้นั้นก็จะให้ความแม่นยำพอสมควร ซึ่งขึ้นอยู่กับชุดขับเคลื่อนของทั้งสองล้อ ว่ามีความเร็วแตกต่างกันหรือไม่ ถ้ามีความเร็วแตกต่างกัน ก็จะทำให้แกนหมุนเคลื่อนไปจากจุดกึ่งกลาง

2. การใช้กลไกส่วนอื่นมาช่วยในการหมุนตัว จะใช้การหมุนของโครงสร้างหุ่นยนต์โดยตรง โดยจะมีส่วนที่จะมาสัมผัสกับพื้นเมื่อต้องการหมุน แล้วทำการหมุน ซึ่งก็จะทำให้หุ่นยนต์หมุนรอบแกนหมุนนั้นโดยตรง ทำให้มีความเที่ยงตรงมาก แต่จะมีข้อเสียก็คือจะมีกลไกที่ซับซ้อน และจะเสียเวลาในการหมุนมาก เพราะต้องมีกระบวนการในการเคลื่อนที่ให้ชุดกลไกในการหมุนมาสัมผัสกับพื้น และยกตัวขับเคลื่อนขึ้นให้สูงขึ้นจากพื้น เมื่อหมุนเสร็จก็ต้องเคลื่อนที่กลับให้ล้อมาสัมผัสกับพื้นเหมือนเดิม

ข้อดีของการเคลื่อนที่ในลักษณะนี้คือจะให้ความละเอียดที่สูงกว่า ควบคุมตำแหน่งได้ง่ายเหมาะกับการเคลื่อนที่ที่ต้องการความแน่นอน แต่ความเร็วจะต่ำและเคลื่อนที่ได้ช้า

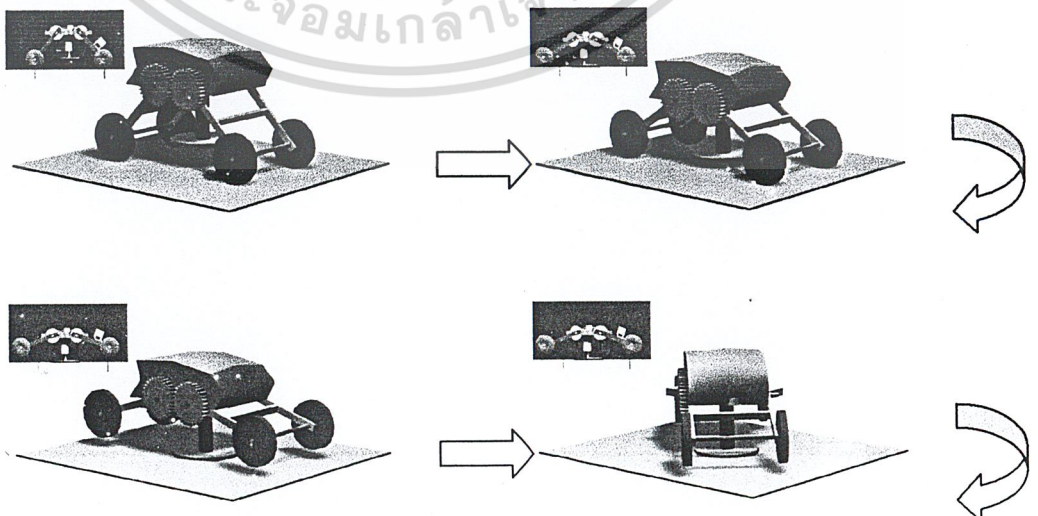
กว่าลักษณะแรก

3.3 ลักษณะของหุ่นยนต์ที่ออกแบบ

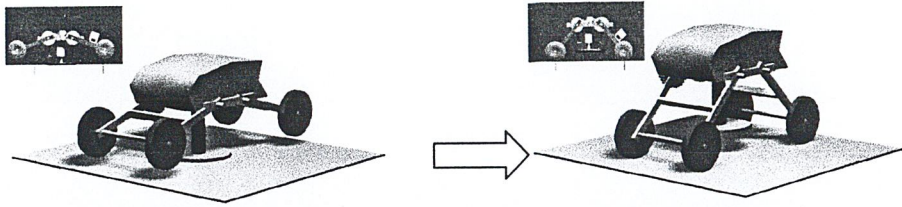
เนื่องจากหุ่นยนต์ที่ออกแบบนั้น ต้องการให้มีความแม่นยำในการเคลื่อนที่สูงสุด จึงเลือกที่จะออกแบบให้หุ่นยนต์มีการเปลี่ยนทิศทางของการเคลื่อนที่ในลักษณะการหมุน และจากการปรึกษากับอาจารย์ที่ปรึกษา และในกลุ่ม หน่วยงานมีความเห็นตรงกันว่า จะใช้วิธีหมุนแบบที่ 2 คือ การใช้กลไกอื่นมาช่วยหมุนตัว เนื่องจากมีความแม่นยำกว่า และไม่ค่อยพบเห็นการเคลื่อนที่ลักษณะนี้ซักเท่าไร จึงเป็นความท้าทายที่จะสร้างและออกแบบขึ้นมา ถ้าสามารถใช้งานได้ดีและได้รับการยอมรับ ก็จะเป็นการชี้แนวทางในการพิจารณาสร้างหุ่นยนต์ที่มีความละเอียดในการเคลื่อนที่สูงต่อไป โดยในการออกแบบนั้นแบ่งการออกแบบเป็นสองส่วน คือ ฮาร์ดแวร์ และซอฟต์แวร์ โดยการดำเนินการจะทำการออกแบบและสร้างส่วนฮาร์ดแวร์ให้สำเร็จก่อน จากนั้นจะทำการเขียนซอฟต์แวร์ขึ้น เนื่องจากซอฟต์แวร์ที่ดีจะต้องทำงานเข้ากับฮาร์ดแวร์ และการพัฒนาจะต้องคำนึงถึงข้อจำกัดต่างๆของฮาร์ดแวร์ ซึ่งจะช่วยให้ลดข้อผิดพลาดที่เกิดจากฮาร์ดแวร์ลงได้

3.3.1 ฮาร์ดแวร์

ในการออกแบบทางฮาร์ดแวร์นั้นจะประกอบไปด้วยสามส่วนคือ ส่วนของโครงสร้างหุ่นยนต์ ชุดขับเคลื่อน และส่วนของวงจรทั้งหมด ซึ่งวิธีการยกตัวและระบบการหมุนตัวที่ออกแบบมาจะทำงานในลักษณะเมื่อวิ่งแล้วในตอนเดียวของตัวหุ่นยนต์จะมีลักษณะที่ต่างออกจากรูปแบบของหุ่นยนต์โดยทั่วไปคือจะมีการหยุดตัวของหุ่นยนต์แล้วยกตัวหุ่นในส่วนของล้อขึ้นมาและหมุนตัวในการเลี้ยวตามแต่องศาที่ต้องการ หลังจากหมุนตัวแล้วตัวหุ่นยนต์ก็จะยกตัวรถขึ้นมาและแล่นต่อไปได้ ดังรูปที่ 3.1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

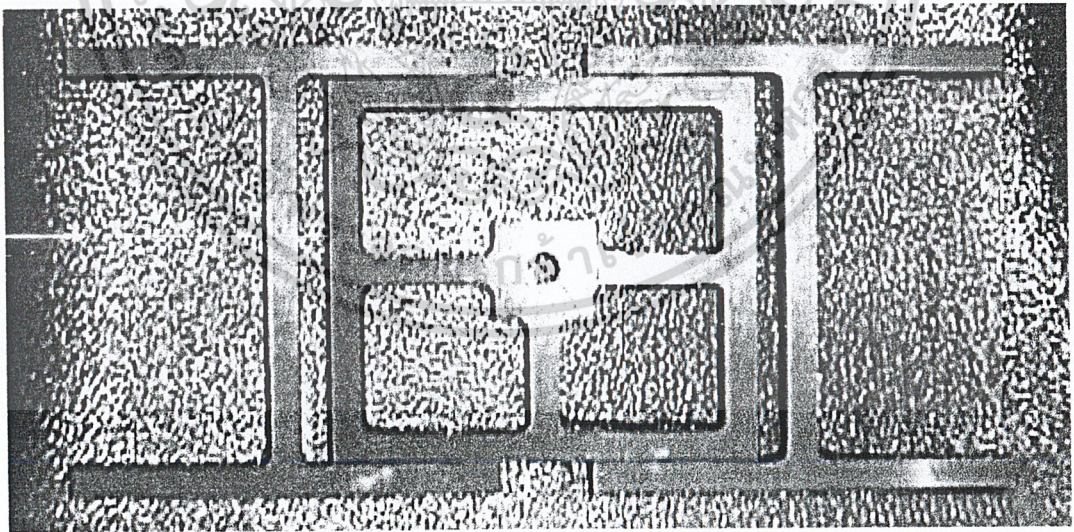


รูปที่ 3.1 แสดงการหมุนตัวของหุ่นยนต์ที่ออกแบบไว้

3.3.1.1 โครงสร้างหุ่นยนต์

การออกแบบโครงสร้างนั้นจะต้องมีความแข็งแรง เพื่อที่จะไม่ให้เกิดการคดงอ หรือว่าโค้งงอของตัวหุ่นยนต์ซึ่งอาจจะเป็นผลให้ตำแหน่งของล้อบิดเบี้ยวไปได้ จะทำให้การเคลื่อนที่เกิดความผิดพลาดของตำแหน่งขึ้น รูปร่างที่เจาะจะต้องมีตำแหน่งที่ถูกต้อง จุดหมุนที่เป็นแกนหรือเพลาดังกล่าวต้องสวมต่อกันอย่างพอดี ไม่มีการคลอนเกิดขึ้น เพราะความผิดพลาดเพียงเล็กน้อยอาจจะทำให้เกิดข้อผิดพลาดขึ้นได้

การออกแบบโครงสร้างเลือกใช้วัสดุที่เป็นแผ่นพีวีซี (PVC) ซึ่งมีน้ำหนักเบาและแข็งแรง โดยทำการกัดขึ้นรูปและเจาะรูให้เป็นโครงสร้างที่ต้องการโดยใช้เครื่องมือกลึง (Milling Machine) ซึ่งโครงสร้างหลักจะประกอบไปด้วยส่วนประกอบ 3 ชั้น ดังรูปที่ 3.2

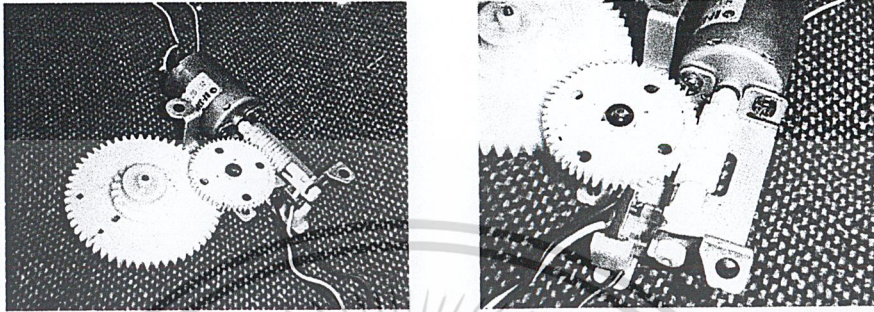


รูปที่ 3.2 แสดงโครงสร้างพื้นฐานของหุ่นยนต์

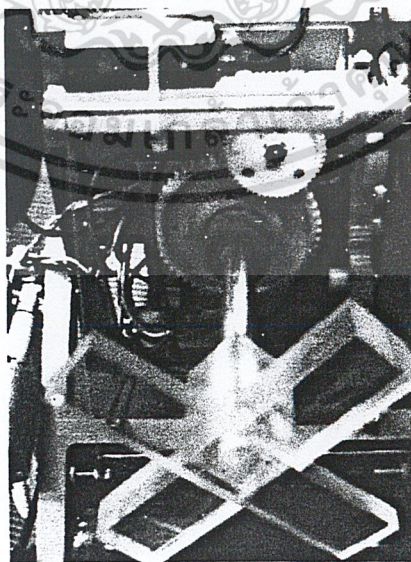
โครงสร้างแต่ละชั้นจะเชื่อมกันด้วยแกนเพลาลูกปืนเพื่อให้สามารถยกล้อขึ้นให้ส่วนกลไกในการหมุนเคลื่อนที่ลงมาสัมผัสกับพื้นในการหมุนดังแบบจำลอง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.2 ชุดขับเคลื่อน

ชุดขับเคลื่อนจะประกอบไปด้วยมอเตอร์ขับเคลื่อน เฟืองทด สายพาน และล้อ โดยในการออกแบบจะใช้ชุดมอเตอร์พร้อมเฟืองทดทั้งหมด 3 ชุด ซึ่งประกอบไปด้วยชุดสำหรับขับเคลื่อน ชุดสำหรับยกตัว และชุดสำหรับหมุน ซึ่งจะใช้มอเตอร์ชนิดเดียวกัน

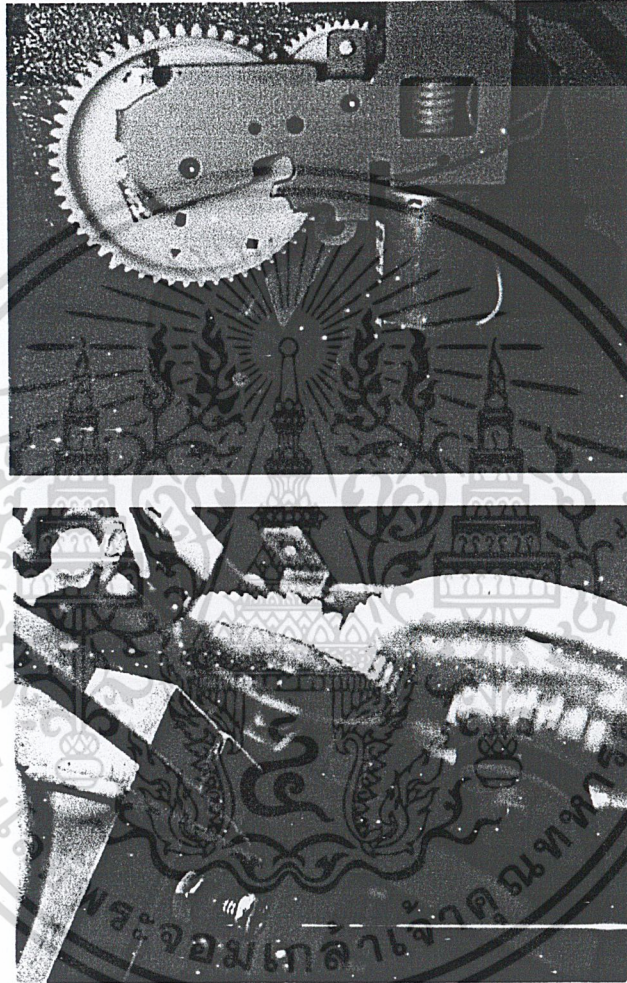


รูปที่ 3.3 มอเตอร์ที่ใช้ในการขับเคลื่อนและเฟืองทดรอบเพิ่มแรงบิดซึ่งติดตั้งเอ็นโคเดอร์ไว้ด้วย เฟืองทดจะเป็นแบบตัวหนอนและจะมาทดด้วยเฟืองธรรมดาอีกครั้ง โดยให้อัตราทดรวมคือ $1 : 229.5$ โดยจะมีชุดเข้าห้สำหรับนับจำนวนรอบของการหมุนของมอเตอร์โดยมอเตอร์หมุนไป 1 รอบจะสามารถนับได้ 3 หน่วย ดังนั้นความละเอียดของการหมุนตัวนั้นจะมีตรวจนับได้สูงสุดเท่ากับ $229.5 \times 3 = 688.3$ หน่วย ในการหมุน 1 รอบ (360 องศา) หรือจะเท่ากับความละเอียดที่ตรวจนับได้สูงสุด = 0.523 องศา และถ้าใช้ล้อที่มีเส้นผ่านศูนย์กลาง 5 เซนติเมตร ก็จะให้ความละเอียดในการเคลื่อนที่เท่ากับ 0.23 มิลลิเมตร ซึ่งเป็นค่าที่คำนวณโดยยังไม่คำนึงถึงค่าผิดพลาดจากความเฉื่อยของมอเตอร์และปัจจัยอื่น ๆ ที่อาจจะเกิดขึ้นได้



รูปที่ 3.4 การยึดแกนหมุนเข้ากับเฟืองตัวใหญ่ ซึ่งนับการหมุนได้ละเอียด 0.523 องศา เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับวิธีในการหมุนตัวนั้น จะมีชุดมอเตอร์ยึดติดอยู่กับส่วนแกนกลางของแกนหมุนอยู่ เมื่อต้องการจะหมุนตัว หุ่นยนต์ก็จะทำการย่อตัวให้ฐานสัมผัสกับพื้น และล้อขับเคลื่อนก็จะลอยขึ้นจากพื้น จากนั้นมอเตอร์ก็จะหมุนตัว ทำให้ตัวหุ่นยนต์หมุนรอบตัวเอง เมื่อถึงตำแหน่งที่ต้องการก็จะหยุดและยกตัวขึ้น ล้อขับเคลื่อนก็จะสัมผัสกับพื้นตามปกติ พร้อมทั้งจะเคลื่อนที่ต่อไป



รูปที่ 3.5 แสดงชุดมอเตอร์ขับเคลื่อนเมื่อเชื่อมต่อเข้ากับเพลาล้อขับเคลื่อน

3.3.1.3 วงจรควบคุม

การควบคุมหุ่นยนต์จะควบคุมโดยโปรแกรมคอมพิวเตอร์ผ่านไมโครคอนโทรลเลอร์เบอร์ 8051 จากนั้น ไมโครคอนโทรลเลอร์จะทำการส่งสัญญาณควบคุมไปยังวงจรวงจรถับเคลื่อนมอเตอร์ เมื่อมอเตอร์หมุนก็จะถูกตรวจนับรอบโดยวงจรเ็นโคดเดอร์ ซึ่งจะส่งสัญญาณพัลส์ไปยังไมโครคอนโทรลเลอร์ เมื่อถึงตำแหน่งที่ต้องการ ไมโครคอนโทรลเลอร์ก็จะส่งสัญญาณหยุดทำงานไปให้กับวงจรถับเคลื่อนมอเตอร์

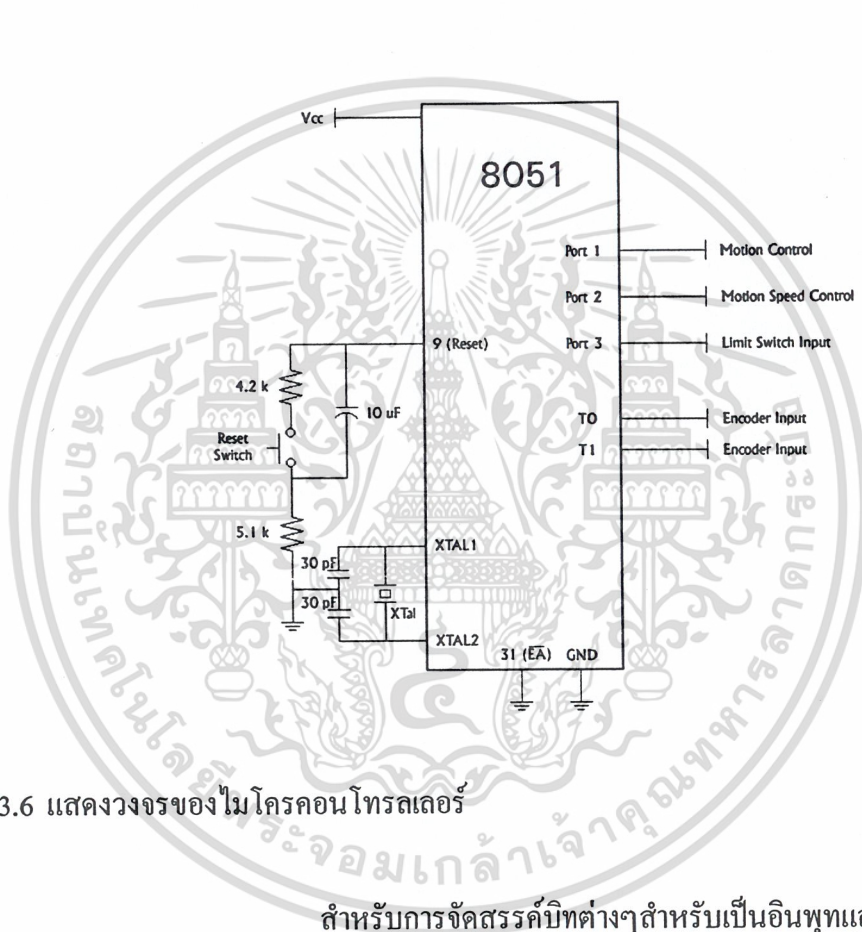
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่กล่าวมาจะพบว่า วงจรที่สำคัญจะมีทั้งหมด 3 ส่วนด้วยกัน คือ

1. วงจรไมโครคอนโทรลเลอร์
2. วงจรขับเคลื่อนมอเตอร์
3. วงจรนับรอบ หรือวงจรเอ็นโคเดอร์

3.3.1.3.1 วงจรไมโครคอนโทรลเลอร์

วงจรของไมโครคอนโทรลเลอร์



รูปที่ 3.6 แสดงวงจรของไมโครคอนโทรลเลอร์

สำหรับการจัดสรรบิตต่างๆสำหรับเป็นอินพุทและเอาต์พุทในแต่ละพอร์ตจะเป็นไปดังต่อไปนี้

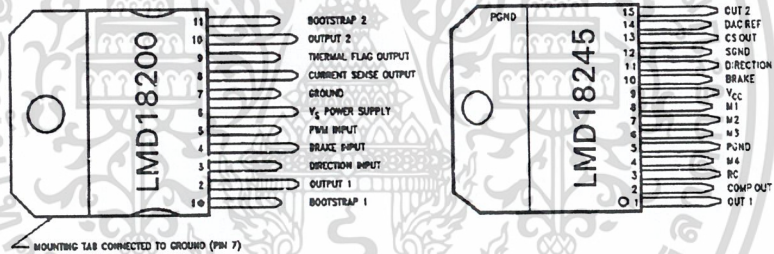
พอร์ต 0	บิต 0 - Run/Brake มอเตอร์ขับเคลื่อน
	บิต 1 - Run/Brake มอเตอร์ยกตัว
	บิต 2 - Run/Brake มอเตอร์หมุนตัว
	บิต 3 - ไม่ได้ใช้
	บิต 4 - หมุนซ้าย/ขวา มอเตอร์ขับเคลื่อน
	บิต 5 - หมุนซ้าย/ขวา มอเตอร์ยกตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

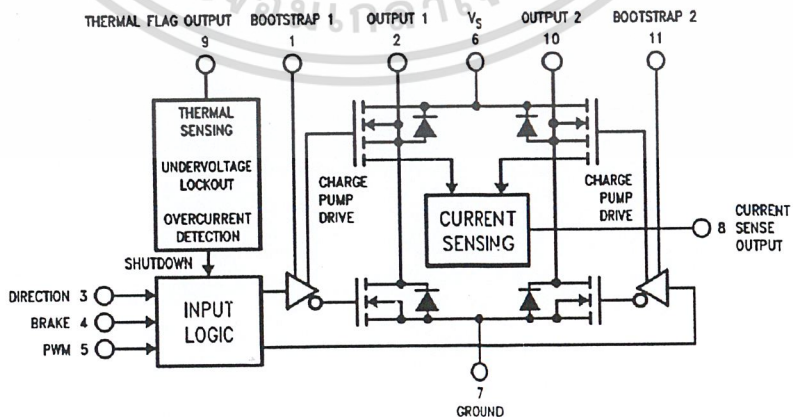
- บิต 6 - หมุนซ้าย/ขวา มอเตอร์หมุนตัวไม่ได้ใช้
- บิต 7 - ไม่ได้ใช้
- พอร์ท 1**
 - บิต 0 - PWM ควบคุมความเร็วมอเตอร์หมุนตัว
 - บิต 2 ถึง 7 - ไม่ได้ใช้
- พอร์ท 2**
 - บิต 0 - Limit Switch ยกตัว Up State
 - บิต 1 - Limit Switch ยกตัว Down State
 - บิต 4 ถึง 7 - ไม่ได้ใช้

3.3.1.3.2 วงจรขับเคลื่อนมอเตอร์

วงจรถับเคลื่อนมอเตอร์จะใช้ IC เบอร์ LMD18200 และ LMD18245 ของเนชั่นแนล ซึ่งเป็น IC ขับเคลื่อนมอเตอร์แบบ H-Bridge ที่มีประสิทธิภาพสูงและต่อใช้งานได้ง่าย สามารถตั้งควบคุมการหมุนและทิศทางของการหมุนได้จากสัญญาณดิจิทัลที่ออกมาจากตัวไมโครคอนโทรลเลอร์ได้โดยตรง อีกทั้งยังสามารถควบคุมความเร็วในการหมุนได้ง่ายอีกด้วย

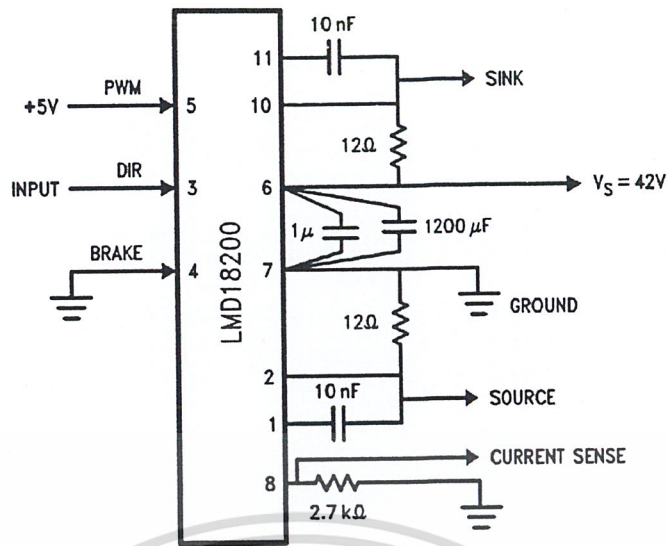


รูปที่ 3.7 แสดงการจัดวางขาของ LMD18200 และ LMD18245



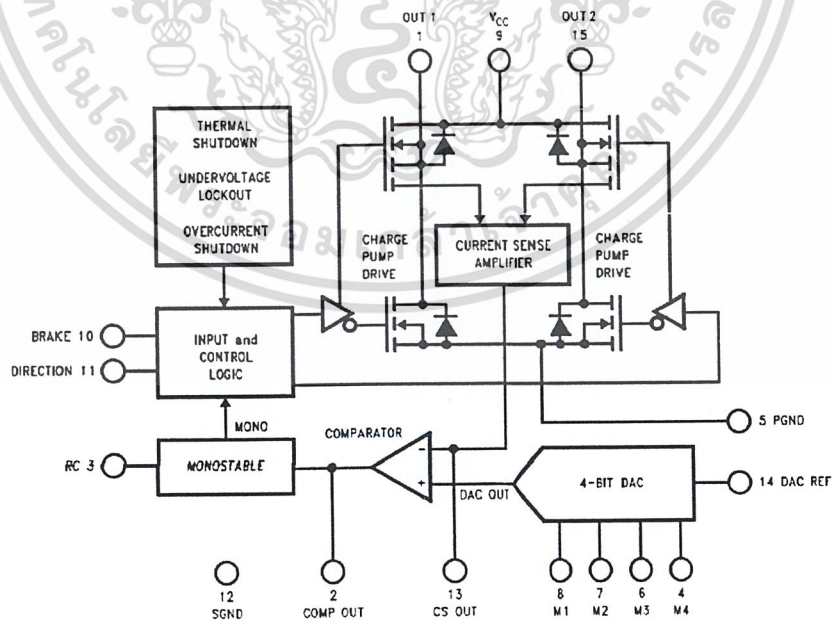
รูปที่ 3.8 โครงสร้างภายในของ LMD18200

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



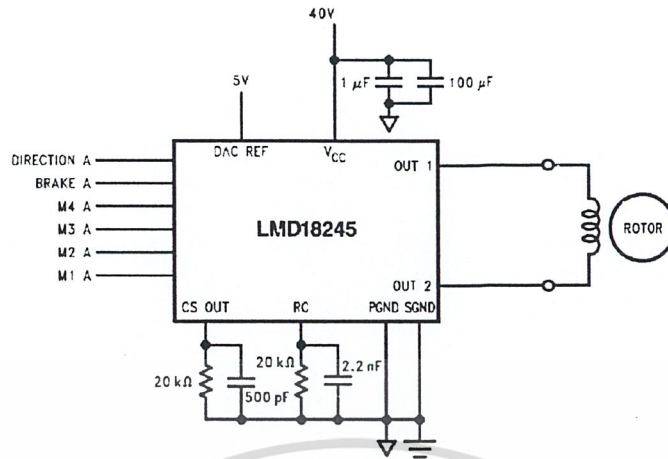
รูปที่ 3.9 วงจรใช้งานของ LMD18200

การใช้งาน LMD18200 จะใช้ในการควบคุมมอเตอร์ไว้สำหรับการขับเคลื่อน เนื่องจากระบบมีการปรับความเร็วด้วย PWM สามารถปรับความเร็วได้สะดวกกว่า LMD18245 ในขณะที่ LMD18245 ถูกนำไปใช้ในการขับเคลื่อนตัวและมอเตอร์ชุดยกตัว



รูปที่ 3.10 โครงสร้างภายในของ LMD18245

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



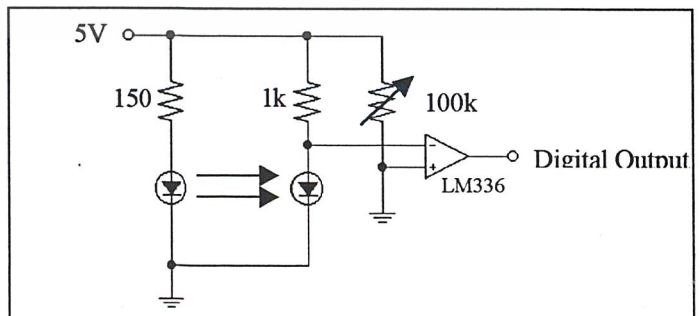
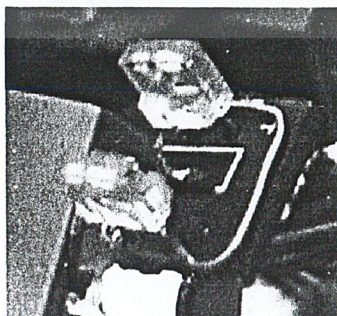
รูปที่ 3.11 วงจรใช้งานของ LMD18245

สำหรับ LMD18245 จะแตกต่างกับ LMD18200 ที่การควบคุมความเร็วจะใช้ D/A ขนาด 4 บิตแทนการใช้ PWM ซึ่งจากการทดลองใช้งานนั้นจะตอบสนองต่อแรงดัน Output ที่ต่ำกว่า 15V ไม่ค่อยเห็นผลซักเท่าไร จึงนำมาใช้งานในการยกและหมุนตัวซึ่งจะไม่มี การควบคุมความเร็ว

วงจรรวมตระกูล LMD ทั้งสองเบอร์นี้นั้นได้ช่วยอำนวยความสะดวกในการออกแบบวงจรขับเคลื่อนมอเตอร์เป็นอย่างมาก อีกทั้งยังทำให้วงจรมีขนาดเล็กและมีเสถียรภาพที่ดีอีกด้วย

3.3.1.3.3 วงจรนับรอบ หรือวงจรเข้ารหัส

วงจรมับรอบจะนับรอบในการหมุนของมอเตอร์โดยใช้แสงอินฟราเรด ซึ่งได้จากอินฟราเรดไดโอด โดยจะมีตัวโฟโตไดโอดเป็นตัวตรวจจับแสงที่ส่งออกมาเมื่อมอเตอร์หมุน จะมีครีบบันทึกอยู่กับตัวมอเตอร์มาตัดกับลำแสงอินฟราเรด ทำให้เกิดความแตกต่างของความเข้มแสงขึ้นที่ตัวโฟโตไดโอด ก็จะทำให้สามารถตรวจจับนับรอบของการหมุนได้โดยใช้วงจรรวมพาราเตอร์



รูปที่ 3.12 แสดงชุดเอ็นโคดเดอร์และวงจรของเอ็นโคดเดอร์

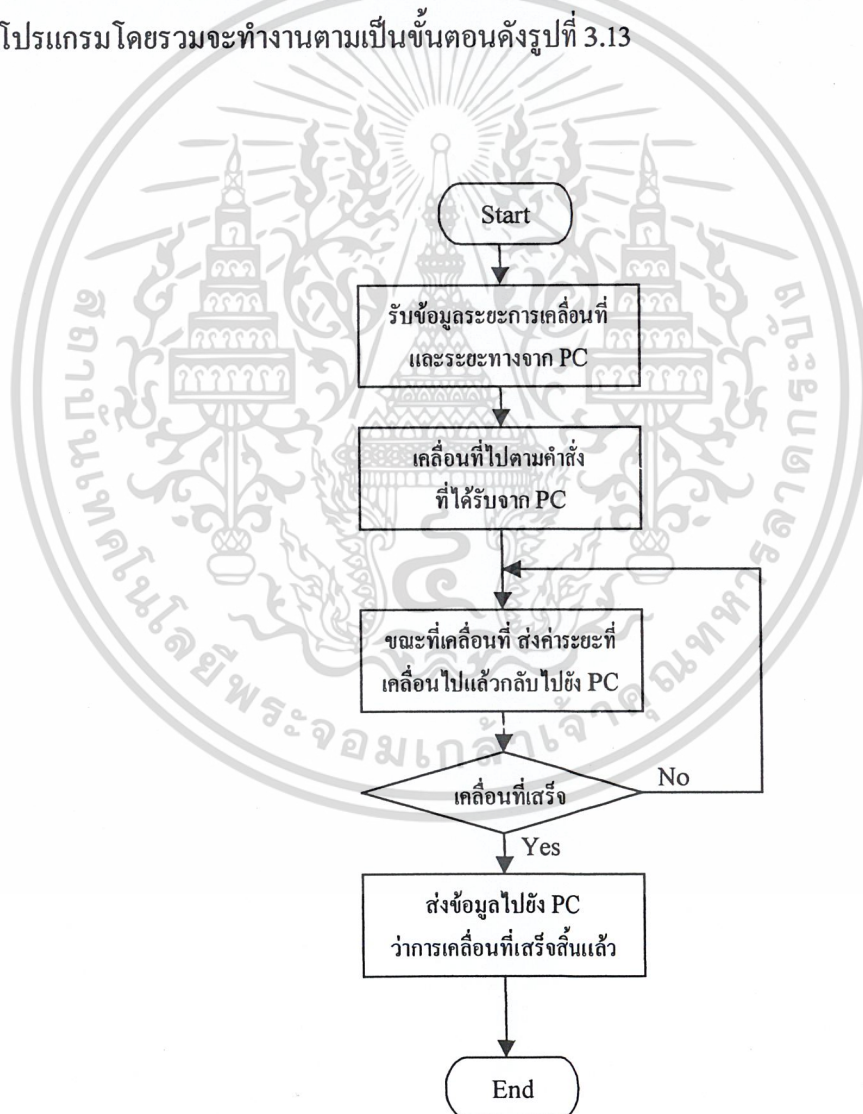
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 ซอฟต์แวร์

ส่วนของซอฟต์แวร์หรือโปรแกรมจะมีทั้งหมดสองส่วนด้วยกันคือ โปรแกรมที่ทำงานบนไมโครคอนโทรลเลอร์หรือบนตัวหุ่นยนต์ และส่วนที่ทำงานบนไมโครคอมพิวเตอร์

3.3.2.1 โปรแกรมที่ทำงานบนไมโครคอนโทรลเลอร์หรือบนตัวหุ่นยนต์

โปรแกรมส่วนนี้จะเขียนด้วยภาษาแอสเซมบลีของ MSC-51 ซึ่งจะเขียนและพัฒนาขึ้นบนไมโครคอมพิวเตอร์ แล้วป้อนโปรแกรมเข้าสู่ไมโครคอนโทรลเลอร์เพื่อให้โปรแกรมทำงานจริงผ่านเครื่องเขียนโปรแกรมหรือที่เรียกกันทั่วไปว่าเครื่องเบิร์น ซึ่งสำหรับตัวของโปรแกรมโดยรวมจะทำงานตามเป็นขั้นตอนดังรูปที่ 3.13



รูปที่ 3.13 แสดงการทำงานของโปรแกรมบนไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมจะทำหน้าที่ควบคุมการทำงานของหุ่นยนต์ให้ทำงานตามคำสั่งของโปรแกรมอีกส่วนหนึ่งซึ่งทำงานอยู่บน ไมโครคอนโทรลเลอร์ผ่านทางสายสัญญาณ RS-232 โดยผ่านชุดข้อมูลขนาด 8 ตัวอักษรซึ่งแสดงรายละเอียดไว้ดังนี้

“T”	Code	Port1	CT (hi)	CT (lo)	PWM	Reserve	“.”
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8

Byte 1 : จะเป็นตัวอักษร “T” เป็นส่วนหัวของข้อมูลให้โปรแกรมทราบว่าจะมีการสื่อสารเกิดขึ้น

Byte 2 : สั่งควบคุมการเคลื่อนที่จะเป็น “x” โดยข้อมูลการเคลื่อนที่จะอยู่ใน Byte ที่ 3 ถึง 7
อ่านค่าข้อมูลสถานะจะเป็น “y” โดยที่ Byte ที่ 3 ถึง 7 จะเป็นอะไรก็ได้

Byte 3 : เป็นค่าของบิตต่างๆใน Port 1 ของไมโครคอนโทรลเลอร์ ซึ่งใช้ในการควบคุมวงจรจับ

Byte 4 : ค่าของ Counter บิตสูง ซึ่งทำงานในโหมด 1 ใช้ในการกำหนดระยะทางการเคลื่อนที่

Byte 5 : ค่าของ Counter บิตต่ำ ซึ่งทำงานในโหมด 1 ใช้ในการกำหนดระยะทางการเคลื่อนที่

Byte 6 : ค่าความเร็วของมอเตอร์ เพื่อกำหนดความเร็วในการเคลื่อนที่

Byte 7 : สงวนไว้เพื่อมีการพัฒนาโปรแกรมเพิ่มเติม

Byte 8 : เป็นตัวอักษร “.” หรือจุดฟูลสตอป แสดงถึงการจบการส่งข้อมูล

เมื่อได้รับข้อมูลการเคลื่อนที่มาแล้ว โปรแกรมก็จะไปตั้งค่าตัวนับ (Counter) ตามข้อมูลที่ส่งมาเพื่อกำหนดระยะทางการเคลื่อนที่ จากนั้นก็จะเช็คบิตต่างๆในพอร์ต 1 ของไมโครคอนโทรลเลอร์ตาม Byte ที่ 3 ของข้อมูลที่ได้มา จากนั้นหุ่นยนต์ก็จะเริ่มเคลื่อนที่ (ไม่ว่าจะหมุนตัวหรือว่าเคลื่อนที่เดินหน้า/ถอยหลัง) เมื่อนับค่าใน Counter ได้ตามที่ตั้งมาแล้ว ก็แสดงว่าถึงตำแหน่งที่ต้องการ โปรแกรมก็จะทำการตั้งค่าในพอร์ต 1 ใหม่ เป็นค่าตัวเลขฐานสอง 11110000 ซึ่งเป็นการสั่งให้มอเตอร์ทุกตัวอยู่ในสภาวะหยุดหมุน แล้วโปรแกรมก็จะหยุดรอคำสั่งต่อไป

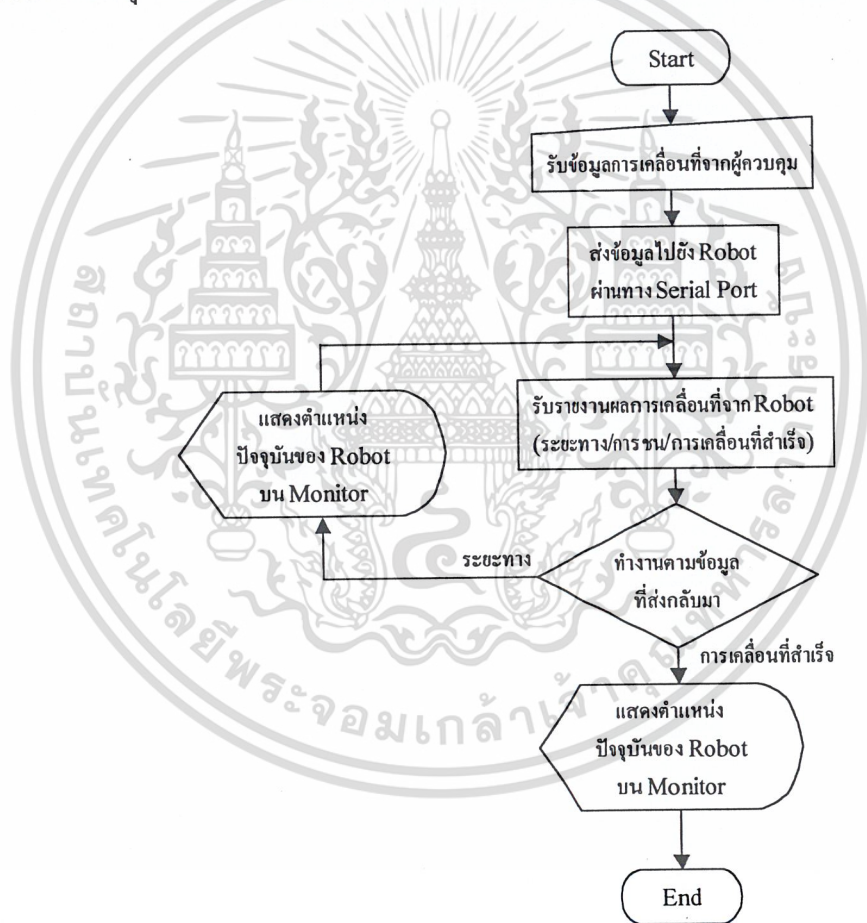
3.3.2.2 โปรแกรมที่ทำงานบนไมโครคอมพิวเตอร์

เป็นโปรแกรมควบคุมที่ทำหน้าที่รับข้อมูลการเคลื่อนที่จากผู้ควบคุมแล้วส่งไปยังไมโครคอนโทรลเลอร์ โดยเมื่อส่งการเคลื่อนที่ไปแล้วจะสร้างแผนที่การเคลื่อนที่ขึ้นมาให้ผู้ควบคุมทราบแนวว่าหุ่นยนต์เคลื่อนที่ไปในทิศทางที่สั่งการไปด้วยระยะทางเท่าไร โปรแกรมจะมีการรายงานการสื่อสารข้อมูลด้วยว่าขณะนี้ได้ส่งข้อมูลออกไปทางพอร์ตอนุกรมเป็นข้อมูลอะไรบ้าง มีข้อมูลอะไรที่หุ่นยนต์ตอบกลับมาบ้าง โดยการทำงานของโปรแกรมจะให้เป็นไปตามแผนภูมิการทำงานดังรูปที่ 3.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมนี้พัฒนาขึ้นบนภาษา Visual Basic ซึ่งเขียน โปรแกรมขึ้นมาได้ง่าย และมี Component ให้เลือกใช้มากมาย เพียงเราต้องการ Component ใดมาใช้งานบนโปรแกรม ก็เพียงแค่ลากมาวางบน Form ของโปรแกรม ก็สามารถใช้งานได้แล้ว และสำหรับในส่วนของการสื่อสารอนุกรมนี้ Visual Basic ก็มี Component ซึ่งช่วยในการจัดการกับการสื่อสารอนุกรม คือ MSCOMM Control ซึ่งทำให้การพัฒนาโปรแกรมเป็นเรื่องง่ายมากขึ้น

การออกแบบโปรแกรมเน้นให้มีความสะดวกสบายในการใช้งาน กล่าวคือ มีส่วนอำนวยความสะดวกต่างๆ เช่น มีแผนที่ในการเคลื่อนที่ที่แสดงให้เห็นและเข้าใจง่าย มีการจดจำไว้ว่าได้เคลื่อนที่ผ่านไปตำแหน่งใดแล้วบ้าง มีการแสดงรายละเอียดของข้อมูลสื่อสารระหว่างไมโครคอมพิวเตอร์ กับตัวหุ่นยนต์ สามารถที่จะกำหนดหน่วยในการเคลื่อนที่ได้ และในการเคลื่อนที่ไปยังจุดใดบนแผนที่ สามารถคลิกเมาส์ลงไปบนตำแหน่งในแผนที่ได้โดยตรง



รูปที่ 3.14 แผนภูมิการทำงานของโปรแกรมควบคุม

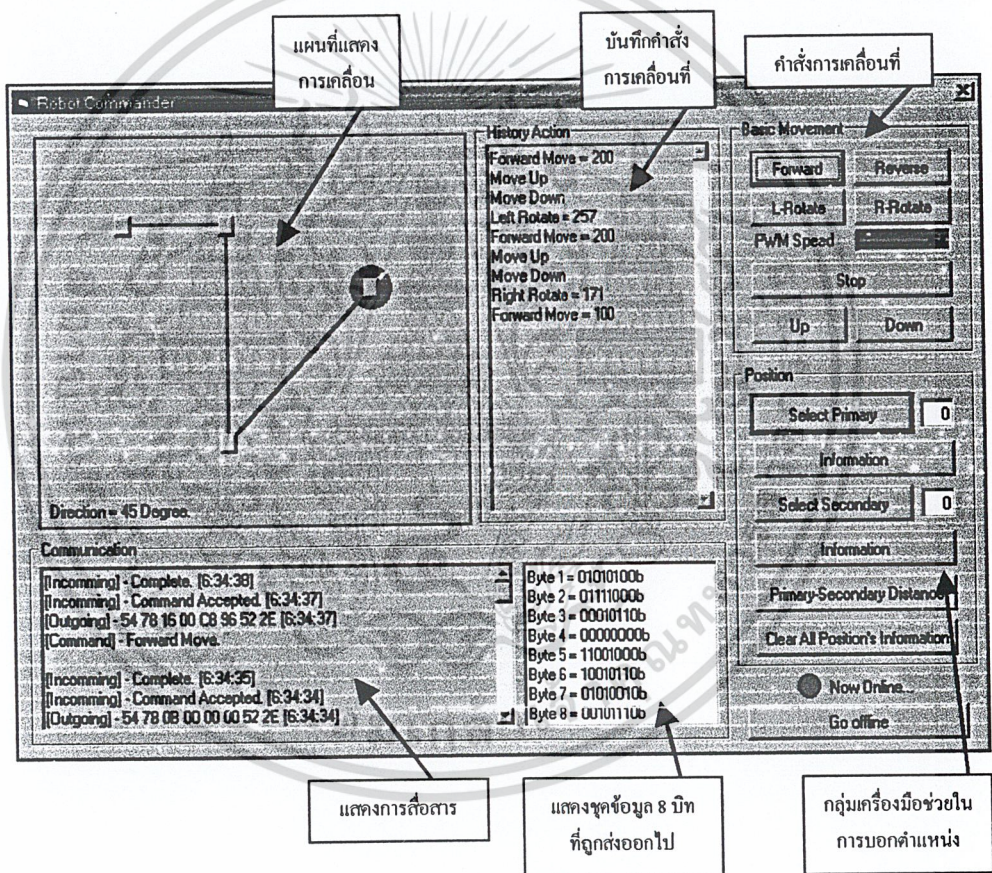
3.3.2.3 การทำงานของโปรแกรม

เมื่อต้องการเชื่อมการติดต่อกับตัวหุ่นยนต์ต้องทำการคลิกที่ปุ่ม Go online ทุกครั้งที่เปิด โปรแกรม โดยตัวโปรแกรมจะมีคำสั่งต่างๆ ให้ใช้งานเพื่อการควบคุมคอมพิวเตอร์ โดยที่คำสั่งสำหรับควบคุมหลักจะอยู่บนมุมมองขวามือ จะมีคำสั่งสำหรับการเคลื่อนที่ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปข้างหน้า/ถอยหลัง และคำสั่งสำหรับการหมุนตัวเพื่อเปลี่ยนทิศทางการเคลื่อนที่ สำหรับด้านซ้ายล่าง (Communication) จะเป็นส่วนแสดงรายละเอียดการสื่อสารอนุกรม โดยถ้ามีการสื่อสารเกิดขึ้น ก็จะมีการรายงานขึ้นมาตลอดเวลาว่ามีข้อมูลอะไรรับ/ส่ง ไปแล้วบ้าง พร้อมกับบอกเวลาที่ใช้ในการสื่อสารที่เกิดขึ้น

เมื่อมีการตั้งคำสั่งให้หุ่นยนต์เคลื่อนที่ที่เกิดขึ้น โปรแกรมก็จะสร้างรหัสข้อมูลขนาด 8 ตัวอักษร แล้วส่งออกไปทางพอร์ตอนุกรม เมื่อหุ่นยนต์รับทราบก็จะเคลื่อนที่ไปตามคำสั่งนั้น โดยโปรแกรมจะสร้างแผนที่สำหรับการเคลื่อนที่ขึ้นมาด้วย โดยที่บริเวณด้านซ้ายบน จะเป็นแผนที่ในการเคลื่อนที่

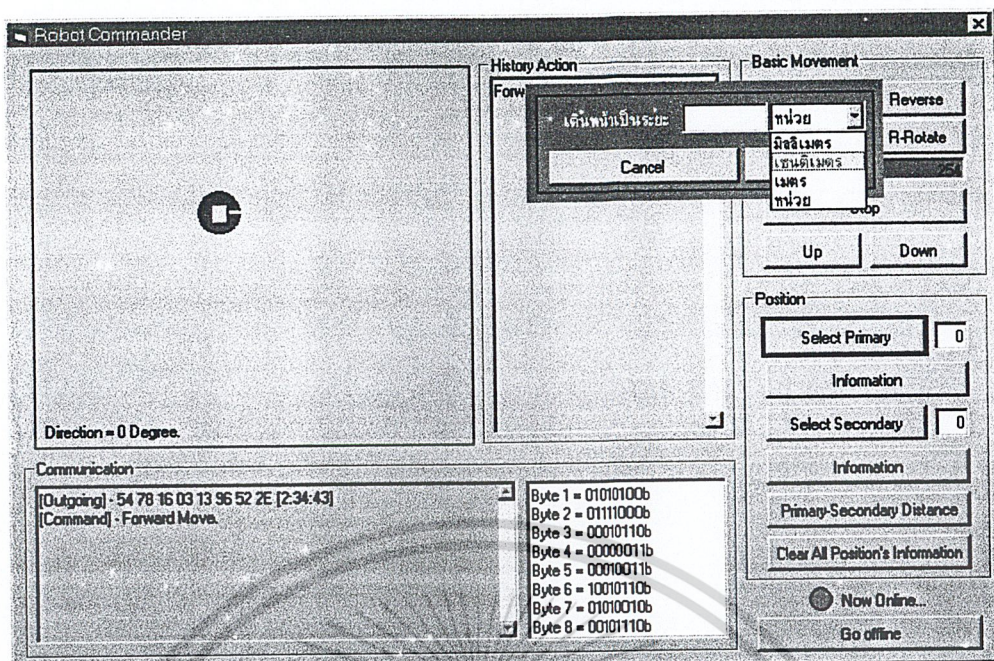
สำหรับลักษณะการทำงานของโปรแกรมจะเป็นไปตามตัวอย่างการทำงานดังต่อไปนี้



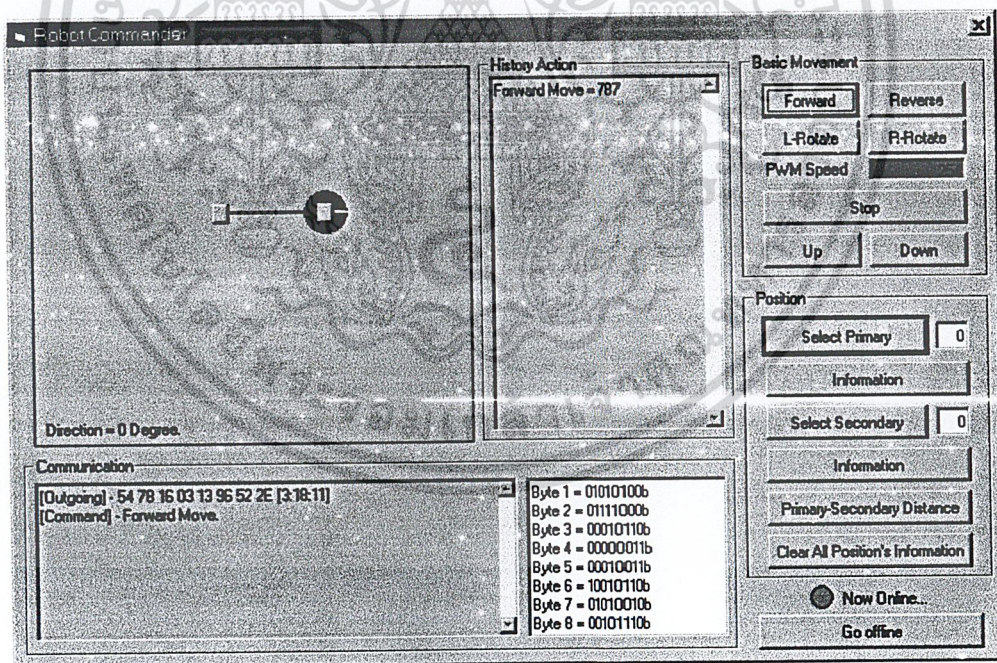
รูปที่ 3.15 ส่วนประกอบต่างๆของโปรแกรมควบคุมบนไมโครคอมพิวเตอร์

เมื่อต้องการสั่งให้หุ่นเคลื่อนที่ สามารถคลิกได้ที่ปุ่มคำสั่งด้านบนขวาของโปรแกรม ก็จะปรากฏส่วนของการป้อนระยะการเคลื่อนที่ขึ้นมา จากนั้นใส่ค่าที่ต้องการเข้าไป โดยสามารถเลือกหน่วยของการเคลื่อนที่ได้ เมื่อเกิดการเคลื่อนที่ขึ้น โปรแกรมก็จะสร้างแผนที่ของการเคลื่อนที่ขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



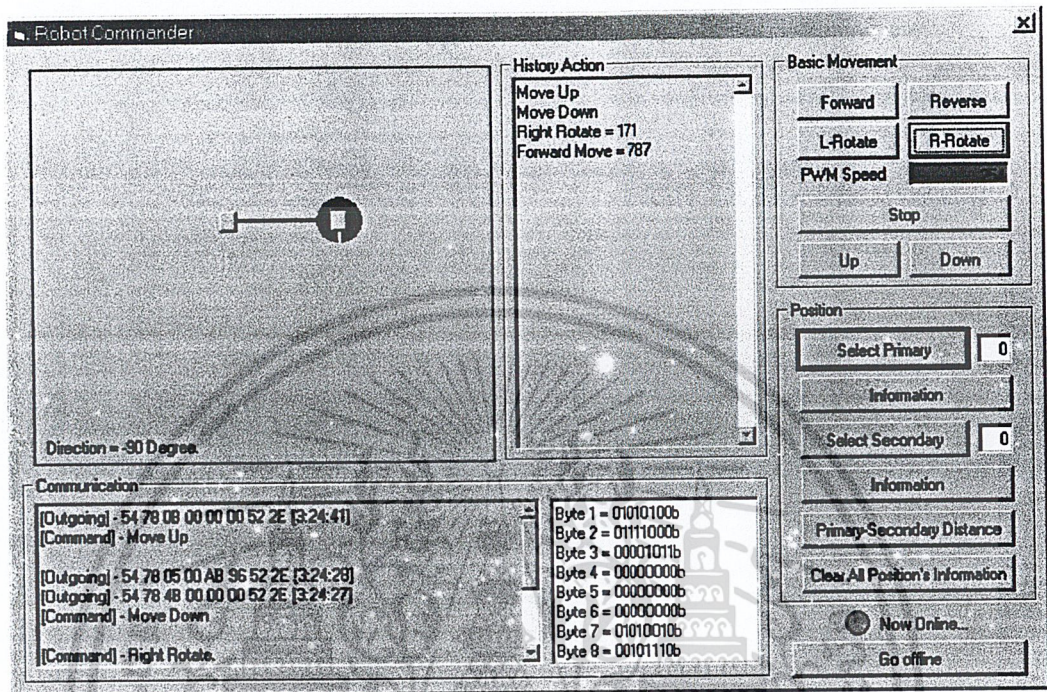
รูปที่ 3.16 ลักษณะของการสั่งให้หุ่นยนต์เคลื่อนที่



รูปที่ 3.17 แสดงการสร้างแผนที่ของโปรแกรมเมื่อมีการเคลื่อนที่ของหุ่นยนต์เกิดขึ้น

ในส่วนของกรอบ Communication จะแสดงถึงข้อมูลที่ส่งไปยัง หุ่นยนต์ซึ่งจะเป็นจะพบข้อมูลการเคลื่อนที่ขนาด 8 ตัวอักษร ซึ่งแปลงเป็นเลขฐาน 16 ซึ่งถูกส่งออก เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปยังตัวหุ่นตามเวลาที่อยู่ในวงเล็บด้านท้ายบรรทัด นอกจากนั้นยังบอกเป็นข้อมูลเลขฐานสองไว้ที่กรอบสีขาวด้วย ซึ่งข้อมูลของการเคลื่อนที่แต่ละครั้งก็จะถูกบันทึกไว้ในกรอบของ History Action

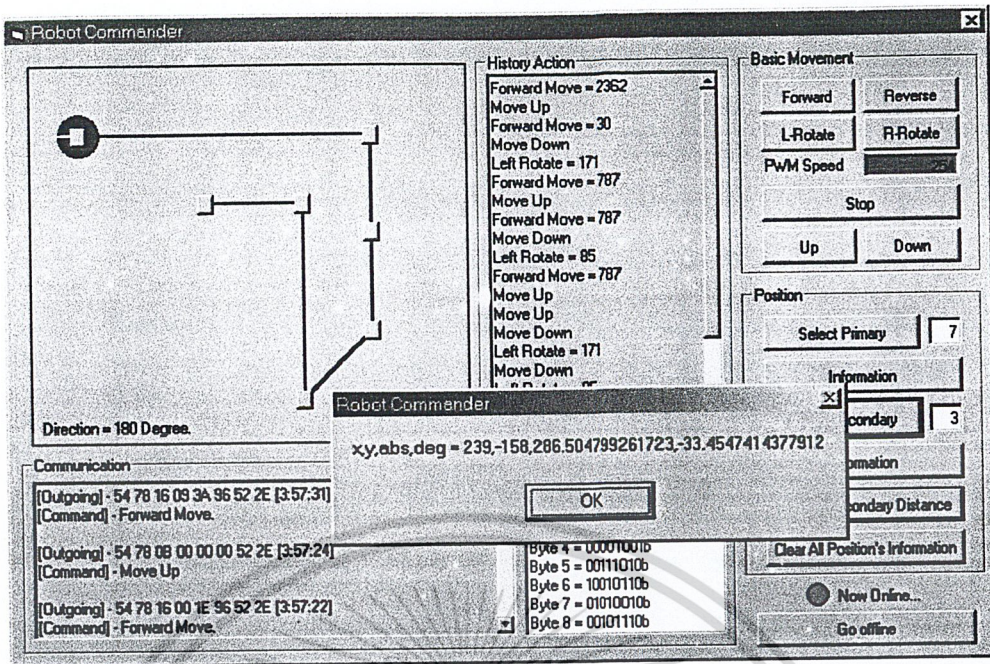


รูปที่ 3.18 แสดงผลเมื่อมีการเคลื่อน ในส่วนแผนที่จะมีการบอกทิศทางว่าเคลื่อนไปแล้ว

สำหรับการหมุนเคลื่อน วิธีการส่งการก็คล้ายคลึงกับการเคลื่อนที่ โดยสามารถระบุหน่วยในการเคลื่อนเป็นองศาได้ เมื่อมีการเคลื่อนเกิดขึ้น สัญลักษณ์ที่แสดงแทนตัวหุ่นยนต์ในกรอบแผนที่ก็จะหมุนตามไปในทิศทางของการเคลื่อนที่ และมีการบอกเป็นค่าองศาไว้ที่ตรงมุมขวาล่างของกรอบแผนที่อีกด้วย

สำหรับในกรอบ Position จะเป็นเครื่องมือที่ใช้ช่วยในการหาพิกัดของหุ่นยนต์ เมื่อหุ่นยนต์เคลื่อนที่ และแผนที่มีการสร้างเส้นทางการเคลื่อนที่พร้อมกับจุดที่เชื่อมต่อ ซึ่งเป็นจุดที่ใช้ในการอ้างอิงพิกัด เมื่อต้องการทราบตำแหน่งเทียบกับจุดเริ่มต้นที่จุดใด สามารถเลือกที่จุดนั้นและดูข้อมูลของการเคลื่อนที่ได้โดยปุ่มคำสั่ง Information โดยจะสามารถเลือกจุดตำแหน่งได้พร้อมกัน 2 จุด (Primary และ Secondary) นอกจากนั้นยังสามารถดูพิกัดระยะทางและมุมของการเคลื่อนที่ระหว่างจุด Primary และ Secondary ได้โดยใช้ปุ่มคำสั่ง Primary-Secondary Distance โปรแกรมก็จะแสดงข้อมูลของการเคลื่อนที่ขึ้นมาดังรูปที่ 3.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 การแสดงผลเมื่อใช้คำสั่งระยะทางระหว่างจุดสองจุด

เมื่อมีการสั่งให้มีการเคลื่อนที่ต่อไปหลายๆครั้ง ก็จะมีข้อมูลการเคลื่อนที่เก็บเพิ่มขึ้นตามจำนวนการเคลื่อนที่ และโปรแกรมก็จะทำการสร้างแผนที่เพิ่มไปเรื่อยๆ ถ้าต้องการลบแผนที่เพื่อนเริ่มต้นการเคลื่อนที่ใหม่โดยถือว่าจุดปัจจุบันเป็นจุดเริ่มต้นก็สามารถทำได้ โดยการใช้ปุ่มคำสั่ง Clear All Position's Information โปรแกรมก็จะทำการล้างแผนที่เก่าออกเพื่อเริ่มต้นใหม่

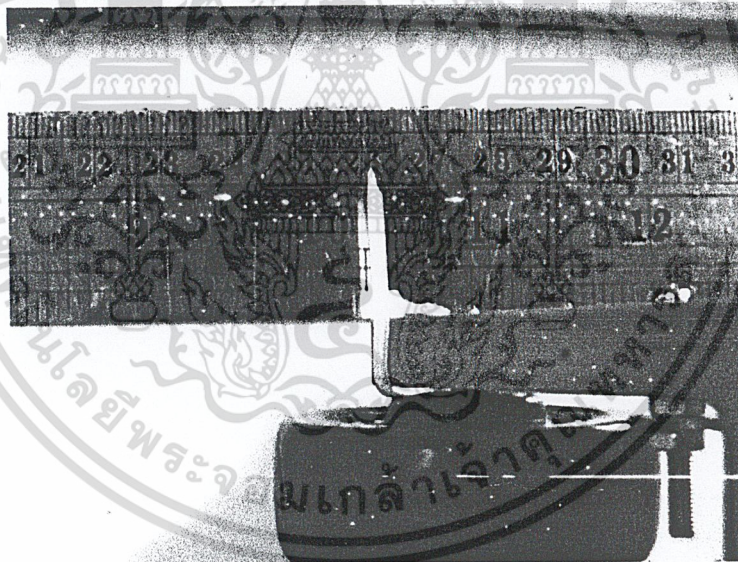
บทที่ 4

การทดลองและผลสรุป

เมื่อสร้างตัวหุ่นยนต์สำเร็จและเคลื่อนที่ได้จริงแล้ว จากนั้นจะเป็นการทดลองว่าหุ่นยนต์สามารถเคลื่อนที่ได้เพียงตรงตามการทดลองหรือไม่ โดยการสั่งให้หุ่นยนต์เคลื่อนที่เข้าไปเข้ามาเป็นระยะทางขนาดเท่าเดิมหลายๆครั้ง ในระยะทางหลายๆช่วง และให้หมุนตัวเป็นมุมองศาเดิมหลายๆครั้ง แล้วบันทึกค่าไว้เป็นสถิติ จากนั้นนำมาคำนวณหาค่าความผิดพลาดจากการเคลื่อนที่ คัดออกมาเป็นเปอร์เซ็นต์เทียบกับระยะทาง

4.1 การทดลองการเคลื่อนที่ในแนวตรง

สำหรับการวัดความเที่ยงตรงของระยะการเคลื่อนที่ จะโดยใช้ไม้บรรทัดที่มีระยะของสเกลที่น่าเชื่อถือได้มาวัดระยะทาง โดยทำจุดอ้างอิงตำแหน่งยื่นออกมาจากตัวหุ่นยนต์ซึ่งอยู่บนสเกลของไม้บรรทัดดังรูปที่ 4.1



รูปที่ 4.1 การทดลองการเคลื่อนที่โดยให้หุ่นยนต์เคลื่อนที่เข้าไปมา

การทดลองจะให้หุ่นยนต์เคลื่อนที่ไปเป็นระยะ 1000 หน่วย, 2000 หน่วย และ 4000 หน่วย ของเอ็น โคเดอร์ (1000 หน่วยของเอ็น โคเดอร์เท่ากับ 128 มิลลิเมตร จากการคำนวณ) เมื่อเคลื่อนที่ไปจนเสร็จก็จะบันทึกค่าระยะทางจริงที่เคลื่อนที่ได้ไว้ จากนั้นก็จะสั่งให้หุ่นยนต์เคลื่อนที่ย้อนกลับมาที่จุดเริ่มต้นด้วยระยะทางเท่าเดิม แล้วบันทึกค่าการเคลื่อนที่อีกครั้ง จากนั้นจะเลื่อนสเกลไม้บรรทัดไปให้ตรงกับจุด 0 ใหม่ ทำการเคลื่อนที่เช่นนี้ทั้งหมด 20 ครั้ง ในการทดลองแต่ละครั้ง ซึ่งผลจากการทดลองจะเป็นไปตามตารางที่ 4.1 ถึง 4.3 และรูปที่ 4.2 และ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ผลการทดลองเมื่อสั่งให้เคลื่อนที่ไปข้างหน้าและย้อนกลับเป็นระยะทาง 1000 หน่วย
ระยะการเคลื่อนที่ 1000 หน่วย (12.8 ซม.)

ครั้งที่	ระยะไป	Error	ระยะกลับ	Error
1	12.3	-0.5	0	0
2	12.2	-0.6	0	0
3	12.5	-0.3	0.15	-0.15
4	12.2	-0.6	0.1	-0.1
5	12.5	-0.3	0.1	-0.2
6	12.1	-0.7	0.05	-0.05
7	12.3	-0.5	0.1	-0.1
8	12.5	-0.3	0.2	-0.2
9	12.1	-0.7	0	0
10	12.1	-0.7	0.1	-0.1

ครั้งที่	ระยะไป	Error	ระยะกลับ	Error
1	12.1	-0.7	0	0
2	12.2	-0.6	0.1	-0.1
3	12.2	-0.6	0.05	-0.05
4	12.1	-0.7	0	0
5	12.3	-0.5	0.1	-0.1
6	12.1	-0.7	0	0
7	12.1	-0.7	0	0
8	12.2	-0.6	0.05	-0.05
9	12.2	-0.6	0.1	-0.1
10	12.1	-0.7	0	0

$$X_{\text{ไป}} = 12.22 \text{ (S.D. = 0.14)}$$

$$X_{\text{กลับ}} = 0.06 \text{ (S.D. = 0.06)}$$

ตารางที่ 4.2 ผลการทดลองเมื่อสั่งให้เคลื่อนที่ไปข้างหน้าและย้อนกลับเป็นระยะทาง 2000 หน่วย
ระยะการเคลื่อนที่ 2000 หน่วย (25.6 ซม.)

ครั้งที่	ระยะไป	Error	ระยะกลับ	Error
1	23.6	-2	0.4	-0.4
2	23.8	-1.8	0.5	-0.5
3	24.2	-1.4	0.5	-0.5
4	24	-1.6	0.8	-0.8
5	23.8	-1.8	0.6	-0.6
6	23.7	-1.9	0.5	-0.5
7	24.05	-1.55	0.5	-0.5
8	24.05	-1.55	0.7	-0.7
9	23.8	-1.8	0.5	-0.5
10	23.7	-1.9	0.4	-0.4

ครั้งที่	ระยะไป	Error	ระยะกลับ	Error
1	24	-1.6	0.5	-0.5
2	24.2	-1.4	0.5	-0.5
3	24	-1.6	0.5	-0.5
4	24	-1.6	0.5	-0.5
5	23.7	-1.9	0.4	-0.4
6	23.9	-1.7	0.5	-0.5
7	24	-1.6	0.5	-0.5
8	24.3	-1.3	0.6	-0.6
9	23.8	-1.8	0.6	-0.6
10	23.7	-1.9	0.6	-0.6

$$X_{\text{ไป}} = 23.9 \text{ (S.D. = 0.2)} \quad X_{\text{กลับ}} = 0.53 \text{ (S.D. = 0.1)}$$

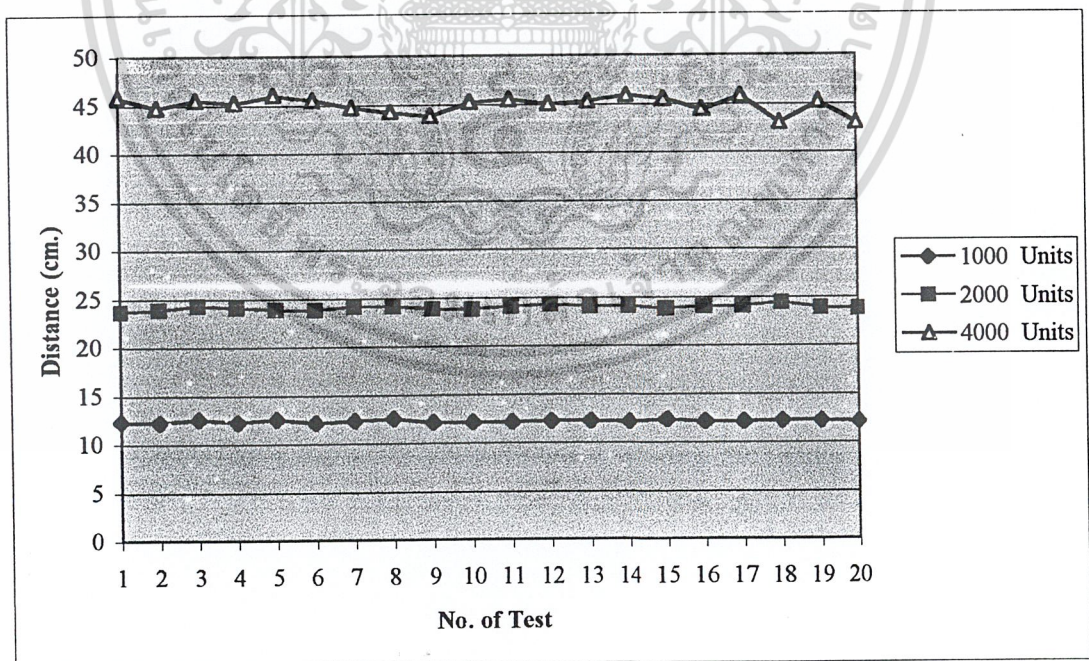
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 ผลการทดลองเมื่อสั่งให้เคลื่อนที่ไปข้างหน้าและย้อนกลับเป็นระยะทาง 4000 หน่วย
ระยะการเคลื่อนที่ 4000 หน่วย (51.2 ซม.)

ครั้งที่	ระยะไป	Error	ระยะกลับ	Error	ครั้งที่	ระยะไป	Error	ระยะกลับ	Error
1	45.8	-5.4	0.9	-0.9	1	45.5	-5.7	1.2	-1.2
2	44.7	-6.5	1.2	-1.2	2	45	-6.2	1.2	-1.2
3	45.5	-5.7	1.1	-1.1	3	45.3	-5.9	0.9	-0.9
4	45.2	-6	0.9	-0.9	4	45.9	-5.3	0.8	-0.8
5	46	-5.2	0.9	-0.9	5	45.5	-5.7	1.2	-1.2
6	45.5	-5.7	0.9	-0.9	6	44.4	-6.8	0.9	-0.9
7	44.7	-6.5	0.9	-0.9	7	45.8	-5.4	1.5	-1.5
8	44.2	-7	1.2	-1.2	8	43	-8.2	1.2	-1.2
9	43.8	-7.4	0.9	-0.9	9	45.2	-6	1.2	-1.2
10	45.2	-6	1.3	-1.3	10	43	-8.2	0.9	-0.9

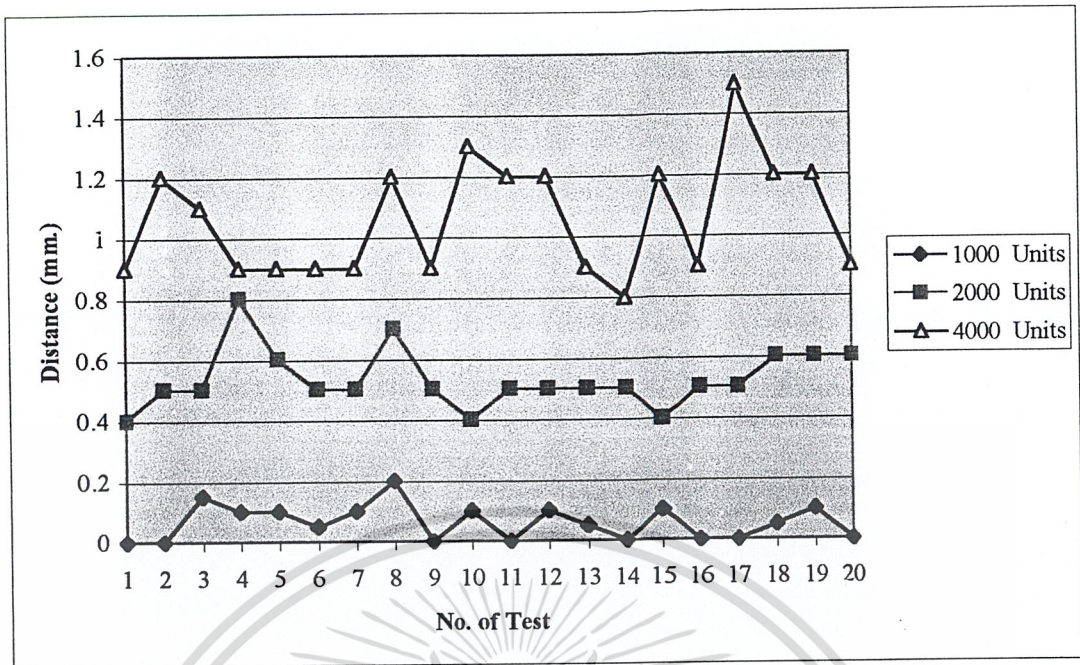
$$X_{\text{ไป}} = 44.96 \text{ (S.D. = 0.88)}$$

$$X_{\text{กลับ}} = 1.06 \text{ (S.D. = 0.2)}$$



รูปที่ 4.2 กราฟระยะทางของการเคลื่อนที่ที่หุ่นยนต์เคลื่อนที่ไปจากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 กราฟระยะทางของการเคลื่อนที่ที่หุ่นยนต์เคลื่อนที่กลับมายังจุดเริ่มต้นจากการทดลอง

จากการทดลองการเคลื่อนที่ในแนวตรงนั้นพบว่า ค่าความผิดพลาดจะแปรผันตามระยะทางการเคลื่อนที่ เมื่อมีการเคลื่อนที่ไกลมากขึ้น ค่าความผิดพลาดก็จะยิ่งเพิ่มมากขึ้น แต่ระยะการเคลื่อนที่ที่ได้ก็มีการเกาะกลุ่มกันดี ถึงจะไม่ได้ค่าตรงตามการคำนวณ แต่ค่าที่ได้ในแต่ละครั้งก็มีความใกล้เคียงกันโดยสังเกตได้จากค่าความเบี่ยงเบนมาตรฐาน โดยความคลาดเคลื่อนที่เกิดขึ้นก็วิเคราะห์ได้ว่ามาจาก 3 ส่วนสำคัญคือ

1. ความคลาดเคลื่อนเนื่องจากการสลิปของสายพาน จากการสังเกตในการเคลื่อนที่หลายๆครั้ง จะเกิดการสลิปของสายพานขึ้นเสมอ โดยเฉพาะในเวลาที่หุ่นยนต์ออกตัว และหยุดการเคลื่อนที่ เนื่องจากในช่วงแรกของการเคลื่อนที่ จะมีทอร์ก (Torque) จากมอเตอร์ที่ทำให้เกิดความเร็วสูงกว่าเวลาที่เคลื่อนที่ปกติ และจุดเดียวที่สามารถถ่วงได้บนตัวหุ่นยนต์ก็คือตำแหน่งของการคล้องสายพาน และเมื่อหุ่นยนต์เคลื่อนที่ไปพบกับสภาพพื้นผิวที่ไม่เรียบนักเช่น เม็ดทรายหรือหลุมเล็กๆ ก็จะเกิดทอร์กเพิ่มขึ้น ส่งผลให้เกิดการสลิปขึ้นที่สายพานด้วย ซึ่งการสลิปจากสายพานนี้เป็นสาเหตุของการคลาดเคลื่อนที่ใหญ่ที่สุด เนื่องจากจะเกิดไปตลอดระยะเวลาการเคลื่อนที่ ทั้งในช่วงของการเริ่มต้นการเคลื่อนที่ และในระก่วงการเคลื่อนที่ ทำให้เกิดความคลาดเคลื่อนเพิ่มขึ้นตามระยะเวลาการเคลื่อนที่ ที่เพิ่มขึ้น

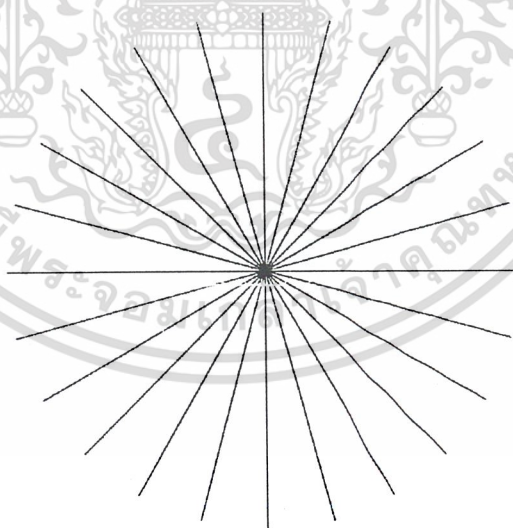
2. ความคลาดเคลื่อนจากการสลิปของล้อกับพื้น กรณีนี้ก็เกิดขึ้นเช่นเดียวกับการสลิปของสายพาน เมื่อมีความเร็วเกิดขึ้นที่ล้อ ก็จะทำให้เกิดการสลิปขึ้น แต่จะเกิดขึ้นเฉพาะช่วงเริ่มต้นและสิ้นสุดของการเคลื่อนที่เท่านั้น การสลิปของล้อมีผลค่อนข้างน้อยเมื่อเทียบกับการสลิปของสายพาน

3. ความคลาดเคลื่อนจากฮิสเทอรีซิสของสายพานและเฟืองทด ในการเคลื่อนที่กลับไปกลับมา ฮิสเทอรีซิส (Hysteresis) เป็นสาเหตุของการคลาดเคลื่อนอยู่เสมอ ซึ่งในส่วนสายพานที่มีการยืดหยุ่นนั้น เมื่อมีการเคลื่อนที่ย้อนกลับ การหมุนกลับทิศทางของมอเตอร์จะสูญเสียไปในการทำให้สายพานเกิดความตึงเครียดส่วนหนึ่งในช่วงเริ่มต้น ในการทดลองครั้งแรก มีการใช้สายพานที่มีความยืดหยุ่นค่อนข้างสูง ซึ่งความผิดพลาดจากฮิสเทอรีซิสของสายพาน มีค่าประมาณ 2 ถึง 4 มิลลิเมตร และเมื่อทราบถึงปัญหานี้ ก็ได้ทดลองเปลี่ยนสายพาน เป็นสายพานที่มีความยืดหยุ่นต่ำลง ก็สามารถลดความผิดพลาดในส่วนนี้ได้เหลือไม่ถึง 1 มิลลิเมตร

สำหรับค่าฮิสเทอรีซิสของชุดเฟืองทด จากการศึกษาคูณสมบัติของชุดเฟืองทดแล้ว มีค่าฮิสเทอรีซิสน้อยมาก ซึ่งจากการทดลองชุดเฟืองทด พบว่าจะมีผลของค่าฮิสเทอรีซิสประมาณ 0.1 มิลลิเมตร ซึ่งน้อยมากจนตัดทิ้งได้เมื่อเทียบกับความผิดพลาดจากฮิสเทอรีซิสของสายพาน

4.2 การทดลองการหมุนตัว

การทดลองการหมุนตัวได้ใช้กระดาษซึ่งตีเป็นเส้นระบุงศาที่แน่นอนมาช่วยเป็นเครื่องอำนวยความสะดวกในการทดลอง ดังรูปที่ 4.4 ซึ่งในแต่ละเส้นที่ใกล้เคียงกัน จะทำมุม 15 องศา



รูปที่ 4.4 กระดาษซึ่งตีเส้นตัดกันเป็นมุม 15 องศา เพื่อใช้ในการทดสอบการหมุนตัว

ในการทดสอบการหมุนตัว จะทดลองให้หุ่นยนต์หมุนตัวเป็นมุม 30 องศา, 90 องศา, 180

องศา และ 360 องศา ซึ่งการทดลองในแต่ละค่า จะทำการทดลองทั้งหมด 20 ครั้งกลับทิศทางไปมา เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 การทดสอบการหมุนที่ 30 องศา
ขนาดของการหมุน 30 องศา

ครั้งที่	ระยะที่วัดได้	ความผิดพลาด	ครั้งที่	ระยะที่วัดได้	ความผิดพลาด
1	30	0	11	30	0
2	30	0	12	30*	+ น้อยกว่า 1
3	30*	+ น้อยกว่า 1	13	30	0
4	30	0	14	30	0
5	30	0	15	30	0
6	30	0	16	30	0
7	30	0	17	30	0
8	30	0	18	30	0
9	31	+1	19	30	0
10	30*	+ น้อยกว่า 1	20	30	0

ตารางที่ 4.5 การทดสอบการหมุนที่ 90 องศา
ขนาดของการหมุน 90 องศา

ครั้งที่	ระยะที่วัดได้	ความผิดพลาด	ครั้งที่	ระยะที่วัดได้	ความผิดพลาด
1	90	0	11	90	0
2	90	0	12	90	0
3	90	0	13	90	0
4	90	0	14	90	0
5	90	0	15	90	0
6	90	0	16	90	0
7	90	0	17	90	0
8	90	0	18	90	0
9	90	0	19	90	0
10	90	0	20	90	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.6 การทดสอบการหมุนที่ 180 องศา
ขนาดของการหมุน 180 องศา

ครั้งที่	ระยะที่วัดได้	ความผิดพลาด	ครั้งที่	ระยะที่วัดได้	ความผิดพลาด
1	180	0	11	180	0
2	180	0	12	180	0
3	180	0	13	180	0
4	180	0	14	180	0
5	180	0	15	180*	+ น้อยกว่า 1
6	180	0	16	180	0
7	180	0	17	180*	+ น้อยกว่า 1
8	180	0	18	180	0
9	180	0	19	180	0
10	180	0	20	180	0

ตารางที่ 4.7 การทดสอบการหมุนที่ 360 องศา
ขนาดของการหมุน 360 องศา

ครั้งที่	ระยะที่วัดได้	ความผิดพลาด	ครั้งที่	ระยะที่วัดได้	ความผิดพลาด
1	360	0	11	360	0
2	360	0	12	360	0
3	361	+1	13	360*	+ น้อยกว่า 1
4	360	0	14	360	0
5	360	0	15	360	0
6	360*	+ น้อยกว่า 1	16	360	0
7	360	0	17	360	0
8	360	0	18	360	0
9	360	0	19	360	0
10	360	+ น้อยกว่า 1	20	360	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดสอบการหมุน พบว่ามีความเที่ยงตรงและแม่นยำดีมาก เนื่องจากการออกแบบหุ่นยนต์นี้ไม่มีส่วนที่อาจทำให้เกิดความผิดพลาดได้เช่นของการเคลื่อนที่ในแนวตรง ทั้งในส่วนของกลไกของการหมุน และตัวชุดเฟืองต่างๆก่อนข้างจะมีเสถียรภาพที่ดีมาก แต่จะมีข้อแม้อยู่ที่ตรงจุดสัมผัสกับพื้นต้องมีความหนึ่ดกับพื้นผิวมากเพียงพอที่จะไม่ทำให้เกิดการสลิปของฐานหมุนขณะทำการหมุน ซึ่งจากผลการทดลองนั้น มีค่าความผิดพลาดน้อยมาก จนไม่สามารถอ่านค่าได้ จึงต้องใช้เครื่องหมาย * แสดงกำกับไว้ที่ค่าการหมุนที่อ่านได้ ซึ่งหมายถึงมีการคลาดเคลื่อนเกิดขึ้นแต่น้อยกว่า 1 องศา ซึ่งเครื่องมือวัดที่มีอยู่ไม่สามารถระบุเจาะจงได้แน่ชัดว่าเป็นส่วนย่อยไปเท่าไร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลงานวิจัยและแนวทางในการพัฒนาต่อไป

จากการออกแบบและสร้างหุ่นยนต์สำรวจการเคลื่อนที่ ที่ใช้การควบคุม โดยคอมพิวเตอร์ และใช้วิธีการเลี้ยวแบบหมุนตัวพบว่า การเคลื่อนที่ในส่วนของการเลี้ยวหมุนตัวให้ผลเป็นที่น่าพอใจเป็นอย่างยิ่ง โดยแทบจะไม่มีอาการคลาดเคลื่อนเกิดขึ้นเลย ผิดกับส่วนของการเคลื่อนที่ในแนวตรง ซึ่งกลับให้ผลไม่เป็นที่น่าพอใจซักเท่าไร

เมื่อมองไปถึงการออกแบบในส่วนของการหมุนตัวก็พบว่า โครงสร้างทั้งหมดนี้มีความแข็งแรงมาก ปราศจากระยะคลอนและการลื่นสลลิด ผิดกับการออกแบบในส่วนของการเคลื่อนที่ในแนวตรง ที่มีจุดอ่อนอยู่ตรงการใช้สายพานเป็นตัวส่งถ่ายกำลัง ซึ่งถือว่าเป็นจุดที่ผิดพลาดที่สุดในการออกแบบหุ่นยนต์นี้ เนื่องจากค่าการสลลิดจากสายพานนั้นมีผลทำให้เกิดความผิดพลาดมากกว่าที่จะยอมรับได้ ซึ่งจากการทดลองมีค่าความเบี่ยงเบนมาตรฐานสูงสุดที่การเคลื่อนที่ 4000 หน่วย (ซึ่งมีค่าเฉลี่ยของการเคลื่อนที่คือ 44.69 เซนติเมตร) จะเท่ากับ 0.88 เซนติเมตร ซึ่งหมายถึงมีความผิดพลาดจากค่ากลางไปเกือบ 1 เซนติเมตร และยังมีแนวโน้มเพิ่มมากขึ้นอีกคือเมื่อมีระยะทางในการเคลื่อนที่เพิ่มขึ้น ซึ่งอาจจะยังไม่สามารถที่จะยอมรับได้ในกรณีที่ต้องการเน้นความแม่นยำในการเคลื่อนที่สูงๆ ดังนั้นการออกแบบพัฒนาเพื่อให้ความแม่นยำสูงมากขึ้นกว่านี้นั้น ควรจะถ่ายกำลังโดยการเชื่อมเฟืองเข้ากับล้อโดยตรง เช่นเดียวกับกลไกของการหมุน โดยการเชื่อมต่อเฟืองเข้ากับแกนหมุนโดยตรง ซึ่งจะทำให้ไม่มีการสลลิดเนื่องจากสายพานเกิดขึ้น

อีกส่วนหนึ่งที่ที่น่าสนใจนั้นเป็นจุดอ่อนในการออกแบบส่วนการยกตัวของหุ่นยนต์ ซึ่งจะทำงานในลักษณะย่อตัวหุ่นยนต์ลงไป โดยยึดล้อคู่หนึ่งไว้ที่ตำแหน่งเดิม แล้วทำการย่อตัวพร้อมกับเคลื่อนล้ออีกคู่หนึ่งออกไป ทำให้หุ่นยนต์มีจุดศูนย์กลางของการเคลื่อนที่เปลี่ยนแปลงไป ทำให้ต้องมีการวัดคำนวณเพื่อที่จะชดเชยระยะที่เลื่อนไป ซึ่งทำให้มีความยุ่งยากมากขึ้นในการออกแบบ ซึ่งในการออกแบบพัฒนาเพื่อให้ความสมบูรณ์กว่านี้อาจใช้การเคลื่อนที่ของแกนหมุนขึ้น-ลงในแนวตั้งแทน ซึ่งทำให้หุ่นยนต์มีแกนหมุนอยู่กับที่ โดยสามารถใช้จุดหมุนเป็นจุดอ้างอิงตำแหน่งได้โดยตรง

ในส่วนของการควบคุมการทำงานของหุ่นยนต์ด้วยไมโครคอมพิวเตอร์นั้น มีข้อดีหลายประการ ถ้านับตัวหุ่นยนต์มีการติดตั้งอุปกรณ์บันทึกภาพหรืออุปกรณ์เครื่องมือวัดต่างๆ ก็สามารถรับข้อมูลมาแสดงผลบนจอภาพได้โดยตรง สำหรับในส่วนของการเขียนโปรแกรม ก็สามารถเขียนโปรแกรมให้มีฟังก์ชันการทำงานได้มากมาย ไม่มีข้อจำกัด สามารถเก็บข้อมูลและแสดงผลออกมาได้โดยตรง เนื่องจากไมโครคอมพิวเตอร์ในปัจจุบันมีขีดความสามารถสูงกว่าในอดีตมาก แต่อาจจะมีข้อเสียในเรื่องของขนาด ซึ่งอาจจะลำบากในการพกพา และในกรณีที่น่ามาใช้กับงานที่ไม่มีความสำคัญมากอาจจะมียุติราคาแพงเกินไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการพัฒนาส่วนของการควบคุมนั้น ยังขาดในส่วนของการทำงานอัตโนมัติ เช่นระบบจดจำเส้นทางสำหรับการเคลื่อนที่ในงานที่ซ้ำๆกัน ระบบการกำหนดเส้นทางการเคลื่อนที่ ซึ่งจะสะดวกขึ้นมากหากสามารถสั่งการ โดยคลิกเมาส์ลงบนแผนที่โดยตรง และอาจจะเพิ่มความทันสมัย โดยการเขียนโปรแกรมแล้วควบคุมผ่านเครื่องคอมพิวเตอร์มือถือหรือพ็อกเก็ตพีซี เช่นเครื่องที่ใช้ระบบปฏิบัติการปาล์ม (Palm OS) ซึ่งมีขนาดเล็ก พกพาง่าย และมีราคาที่ถูกกว่าไมโครคอมพิวเตอร์ ซึ่งเป็นคอมพิวเตอร์มือถือชนิดนี้กำลังได้รับความนิยมมากขึ้นทุกวัน ซึ่งในอนาคตอาจจะกลายมาเป็นส่วนหนึ่งในชีวิตประจำวันและกลายเป็นมาตรฐานที่ใช้ในการบังคับและควบคุมในอนาคต



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- กฤษดา ใจเย็น, อรรถพล บุญยะโกศา และชัยวัฒน์ ลิ้มพรจิตรวิไล, "เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอก ผ่านพอร์ตอนุกรม", บริษัท อิน โนเวตีฟ เอ็กเพอริเมนต์ จำกัด, หน้า 1-34
- รศ.ดร. สมยศ จุณณะปิยะ, "การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51", สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, พ.ศ.2543



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

- ภาคผนวก ก. โปรแกรมควบคุมการทำงานของหุ่นยนต์ในส่วนไมโครคอนโทรลเลอร์
- ภาคผนวก ข. โปรแกรมควบคุมการทำงานของหุ่นยนต์ในส่วนไมโครคอมพิวเตอร์
- ภาคผนวก ค. DATASHEET



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

โปรแกรมควบคุมการทำงานของหุ่นยนต์ในส่วนไมโครคอนโทรลเลอร์

```

ORG      0000H
LJMP     MAIN

ORG      0003H      ; Interrupt 0
LJMP     INTR0

ORG      000BH      ; Timer/Counter 0
LJMP     TIMR0

ORG      0023H      ; Serial Port Interrupt
LJMP     S_INT

MAIN:    MOV      P0,#11111111B ; Clear Port
         MOV      P1,#00001111B
         MOV      P2,#00000000B

         MOV      PCON,#00H ; Set serial port
         MOV      SCON,#50H
         MOV      TMOD,#25H
         MOV      TH1,#0FDH

         SETB     EA ; Enable Interrupt
         SETB     ES ; Enable Serial Interrupt
         SETB     ET0 ; Enable Timer 0 Interrupt
         SETB     EX0 ; Enable Interrupt 0
         SETB     IT0 ; INT0 Edge Detect
         SETB     TR0 ; Start Encoder Counter.
         SETB     TR1 ; Timer 1 Run (Serial)

         MOV      DPTR,#ONLINE
         LCALL    WRITE

         MOV      R7,#00H ; Clr R7 (Set CMD Received)
         MOV      R6,#00H ; Clr R6 (Running Status)
         MOV      R1,#10H

;--- [Register Usage] -----
; R0 = Main program addressing
; R1 = Pointer using in Serial Interrupt
; R2 = Use in PWM Calculation (Active time).
; R3 = Use in PWM Calculation (Deactive time).
; R4 = Run-time distance status.
; R5 = No used

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;      R6 = Running Status
;      R7 = Command Received Status
;      20h = In Command Pointer PUSH.
;--- [Main program] -----
M_LOOP:  DJNZ      R2,COUNT      ;   PWM Calculation.
         CPL      P1.3
         MOV      A,R3
         MOV      R2,A
         CPL      A
         MOV      R3,A

COUNT:  JB       P1.0,SKIPm1
         JNB      P0.6,m1CLR17
         SETB     P1.7
         SJMP     SKIPm1
m1CLR17: CLR      P1.7

SKIPm1:  JB       P1.1,SKIPm2
         JNB      P0.7,m2CLR17
         SETB     P1.7
         SJMP     SKIPm2
m2CLR17: CLR      P1.7

SKIPm2:  NOP

X_COUNT: NOP
;---[Lifting Limit Switch Sensing] -----
LIFT:    JB       P1.2,X_LIFT
         JB       P1.6,LIFT_U
         JNB      P2.1,X_LIFT
         MOV      P1,#00001111B
         MOV      DPTR,#CMPLTE
         LCALL   WRITE
         SJMP     X_LIFT
LIFT_U:  JNB      P2.0,X_LIFT
         MOV      P1,#00001111B
         MOV      DPTR,#CMPLTE
         LCALL   WRITE
X_LIFT:  NOP

C_CHK:   CJNE     R7,#0FFH,M_LOOP ;   If No Command Stay Loop.
         MOV      DPTR,#CMD_OK   ;   Tell PC to know accepted
         LCALL   WRITE
         CLR      TR0
         MOV      A,11H          ;   If B2 = 'r', Send T0 Value.
         CJNE     A,#72H,R_CMND
         CLR      ES
         MOV      A,TH0
         MOV      SBUF,A

```

```

INC          DPTR
JNB          TI,$
CLR          TI
MOV          A,TL0
MOV          SBUF,A
INC          DPTR
JNB          TI,$
CLR          TI
SETB        ES
AJMP        M_LOOP

R_CMND:     MOV          R7,#00H          ; If Commanded Clear Status.
            MOV          R6,#0FFH       ; Set Running Status.

            XCH          A,13H          ; Set HiByte Counter.
            CPL          A
            MOV          TH0,A

            XCH          A,14H          ; Set LoByte Counter.
            CPL          A
            MOV          TL0,A

            MOV          A,15H          ; Set PWM Variable.
            MOV          R2,A
            CPL          A
            MOV          R3,A
            MOV          P0,#

            SETB        TR0             ; Enable Counter.
            MOV          P1,12H         ; Set Port 1 Value.

            AJMP        M_LOOP          ; Loop Program.

;--- [End Main program] -----
;--- [Serial Interrupt] -----
S_INT:      PUSH        ACC
            JNB          TI,S_RECV      ; If no transmit, go to receive.
            CLR          TI
            LJMP        S_REXIT

S_RECV:     MOV          A,SBUF          ; Move SBuf to Acc.
            CLR          RI             ; Clear data waiting flag.
            MOV          @R1,A         ; Move A to table (10h-17h).
            MOV          A,10H
            CJNE        A,#54H,ERR_IN  ; Check header. (=‘T’?)
            INC          R1             ; Increase pointer.
            CJNE        R1,#18H,S_REXIT ; If not complete, wait next.
            SETB        P2.7           ; Set P2.7 (Command
            ; received status)
            MOV          R7,#0FFH      ; Set Commanded Status
            MOV          R1,#10H       ; Reset pointer
ERR_IN:     MOV          R7,#0FFH
S_REXIT:    POP          ACC

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                RETI
;--- [End Serial Interrupt] -----

;--- [Interrupt 0] -----
INTR0:    PUSH    ACC
          MOV     20H,R1
          MOV     R1,#10H
          MOV     P0,#00H
INT0_L:   MOV     SBUF,@R1
          JNB    TI,$
          CLR    TI
          INC    R1
          CJNE   R1,#18H,INT0_L
          MOV    R1,20H
          POP    ACC
          RETI

;--- [End Interrupt 0] -----
;--- [Timer 0 Interrupt] -----
TIMR0:    PUSH    ACC
          MOV     P1,#00001111B
          MOV     R6,#00H
          MOV     DPTR,#CMPLTE
          LCALL  WRITE
          POP    ACC
          RETI

;--- [End Timer 0 Interrupt] -----
;--- [WRITE PROCEDURE] -----
WRITE:    PUSH    ACC
          CLR    ES
W_LOOP:   MOV     A,#00H
          MOVC   A,@A+DPTR
          MOV    SBUF,A
          INC    DPTR
          JNB    TI,$
          CLR    TI
          CJNE   A,#!,W_LOOP
W_END:    SETB   ES
          POP    ACC
          RET

;--- [Text Database] -----
CMPLTE:   DB      'Complete.',0AH
CMD_OK:   DB      'Command Accepted.',0AH
ONLINE:   DB      'Online.',0AH
;--- [END WRITE PROCEDURE] -----
                END                                ;      End Program.

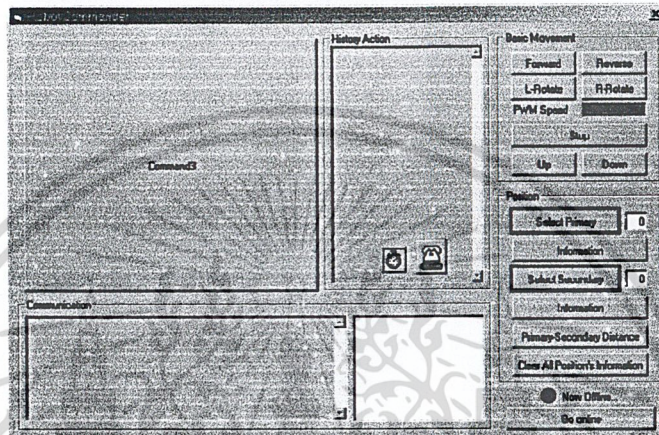
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

โปรแกรมควบคุมการทำงานของหุ่นยนต์ในส่วนไมโครคอมพิวเตอร์

Form1 programming code



```

Dim X As Long, Y As Long
Public MapMoveX As Integer, MapMoveY As Integer, MapRotate As Integer
Public PosSelect As Integer
Public Direction As Long
Public Rotate As Boolean
Public ReportIn As String
Public InBuf As String
Public MapX As Long, MapY As Long
Dim StoreBuf, DataEnd, EndPos, DataIn, DataOut, Out8Bit, PWM, XXX
Dim GraphN As Integer, Degree As Integer, MapMove As Boolean, cx, cy, ox, oy As Long
Dim MoveX(150) As Long, MoveY(150) As Long, NegX As Long, NegY As Long
Dim RealX(150) As Long, RealY(150) As Long, RealDir(150) As Long
Dim Distance As Long, MaxX, MaxY, MinX, MinY, AbsX, AbsY As Integer
Dim ABSDistance As Integer
Dim XScale, YScale, ScaleValue
Public Drawed As Boolean
Const MStop = "11110000", MRotateR = "10100000", MRotateL = "10100100", MReverse =
"01100000", MForward = "01101000"
Const DegConst = 685 / 360

Public Sub MapRedraw(X As Long, Y As Long)
    GraphN = GraphN + 1
    RealDir(GraphN) = Direction
    While RealDir(GraphN) > 359
        RealDir(GraphN) = RealDir(GraphN) - 360
    Wend
    While RealDir(GraphN) < 0
        RealDir(GraphN) = RealDir(GraphN) + 360
    Wend

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RealX(GraphN) = X
RealY(GraphN) = Y
MoveX(GraphN) = X + MoveX(GraphN - 1)
MoveY(GraphN) = Y + MoveY(GraphN - 1)
If MoveX(GraphN) < -NegX Then NegX = -MoveX(GraphN)
If MoveY(GraphN) < -NegY Then NegY = -MoveY(GraphN)
XScale = 1
YScale = 1
For a = 0 To GraphN
    MoveX(a) = MoveX(a) + NegX + 1
    MoveY(a) = MoveY(a) + NegY + 1
    If Abs(312 / MoveX(a)) < XScale Then XScale = 312 / MoveX(a)
    If Abs(264 / MoveY(a)) < YScale Then YScale = 264 / MoveY(a)
Next
If XScale > YScale Then ScaleValue = YScale Else ScaleValue = XScale
Cls
For a = 1 To GraphN
    MapX = cx + ((MoveX(a)) * ScaleValue)
    MapY = cy + ((MoveY(a)) * ScaleValue)
    Line (cx + ((MoveX(a - 1)) * ScaleValue), cy + ((MoveY(a - 1)) * ScaleValue))-(MapX, MapY)
    Circle (cx + ((MoveX(a - 1)) * ScaleValue), cy + ((MoveY(a - 1)) * ScaleValue)), 3, RGB(255,
255, 0)
    ABSDistance = Round(Sqr((RealX(a) ^ 2) + (RealY(a) ^ 2)))
Next
For a = 0 To GraphN
    MapX = cx + ((MoveX(a)) * ScaleValue)
    MapY = cy + ((MoveY(a)) * ScaleValue)
    PositionKey(a).ToolTipText = CStr(ABSDistance) + "," + CStr(RealDir(a))
    PositionKey(a).Width = 12
    PositionKey(a).Height = 12
    PositionKey(a).left = MapX - 6
    PositionKey(a).Top = MapY - 6
    PositionKey(a).Visible = True
Next
End Sub

Private Sub MapClear()
    Command3.Visible = True
    Command3.Visible = False
End Sub

Private Sub ClearPosition_Click()
    For a = 1 To GraphN
        PositionKey(a).Visible = False
    Next
    NegX = 0
    NegY = 0
    Direction = 0
    ScaleValue = 1
    XScale = 1
    YScale = 1
    MoveX(0) = 156
    MoveY(0) = 123
    RealX(0) = MoveX(0)
    RealY(0) = MoveY(0)
    GraphN = 0
    Cls
    Drawed = False
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub GotoPrimary_Click()
    UpButton_Click
    disx = -(MoveX(GraphN) - MoveX(CInt(PriNo.Text)))
    disy = MoveY(GraphN) - MoveY(CInt(PriNo.Text))
    absdis = Sqr((disx ^ 2) + (disy ^ 2))

    If disx <> 0 Then deg = Atn(disy / disx) * 7 * 180 / 22

    If disx = 0 Then
        If disy > 0 Then deg = -90
        If disy < 0 Then deg = 90
        If disy = 0 Then deg = 0
    End If
    deg = deg - Direction
    If deg >= 360 Then deg = deg - 360
    If deg <= -360 Then deg = deg + 360
    If disx >= 0 Then
        If disy > 0 Then
            deg = deg
        Else
            deg = 360 + deg
        End If
    Else
        deg = 180 + deg
    End If
    If deg >= 360 Then deg = deg - 360
    If deg <= -360 Then deg = deg + 360
    If deg >= 180 Then deg = deg - 360
    If deg <= -180 Then deg = deg + 360

    MapMoveX = disx
    MapMoveY = disy
    MapRotate = deg * DegConst

    Distance = MapRotate
    If Distance <> 0 Then

        Form1.DownButton_Click
        ReportIn = "X"
        While ReportIn <> "Complete."
            DoEvents
        Wend
        For aaa = 1 To 1000
            DoEvents
        Next
        ReportIn = ""

        If Distance > 0 Then
            Out8Bit = "Tx" + Chr(Bin2Dec(StrReverse(MRotateL))) + HiByte(Distance) +
            LoByte(Distance) + Chr(150) + "R" + "."
            Direction = Direction + (Distance / DegConst)
        End If
        If Distance < 0 Then
            Out8Bit = "Tx" + Chr(Bin2Dec(StrReverse(MRotateR))) + HiByte(Distance) +
            LoByte(Distance) + Chr(150) + "R" + "."
            Direction = Direction - (Distance / DegConst)
        End If
        SendDATA (Out8Bit)
        While ReportIn <> "Complete."
            DoEvents

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Wend

ReportIn = "X"
While ReportIn <> "Complete."
    DoEvents
Wend
For aaa = 1 To 10000
    DoEvents
Next
UpButton_Click
End If

Distance = absdis
If Distance <> 0 Then
    Out8Bit = "Tx" + Chr(Bin2Dec(StrReverse(MForward))) + HiByte(Distance) +
LoByte(Distance) + Chr(150) + "R" + "."
    SendDATA (Out8Bit)
    X = Round(Cos((Direction / 180) * (22 / 7)) * Distance)
    Y = -Round(Sin((Direction / 180) * (22 / 7)) * Distance)
    MapRedraw X, Y
End If
End Sub

Private Sub PrimaryInfo_Click()
    Index = CInt(PriNo.Text)
    disx = -(MoveX(CInt(PriNo.Text)) - MoveX(0))
    disy = MoveY(CInt(PriNo.Text)) - MoveY(0)
    absdis = Sqr((disx ^ 2) + (disy ^ 2))
    If disx = 0 Then
        If disy > 0 Then deg = 90
        If disy < 0 Then deg = -90
        If disy = 0 Then deg = 0
    End If
    If disx <> 0 Then deg = Atn(disy / disx) * 7 * 180 / 22
    If disx = 0 Then
        If disy > 0 Then deg = -90
        If disy < 0 Then deg = 90
        If disy = 0 Then deg = 0
    End If
    If deg >= 360 Then deg = deg - 360
    If deg <= -360 Then deg = deg + 360
    If disx >= 0 Then
        If disy > 0 Then
            deg = deg
        Else
            deg = 360 + deg
        End If
    Else
        deg = 180 + deg
    End If
    If deg >= 360 Then deg = deg - 360
    If deg <= -360 Then deg = deg + 360
    If deg >= 180 Then deg = deg - 360
    If deg <= -180 Then deg = deg + 360
    AbsX = disx
    AbsY = disy
    OriginDEG = deg
    MsgBox "Primary Position" + Eol + "Position Number " + CStr(Index) + Eol + "X axis distance
from origin = " + CStr(AbsX) + Eol + "Y axis distance from origin = " + CStr(AbsY) + Eol +
"Absolute distane from origin = " + CStr(Sqr((AbsX ^ 2) + (AbsY ^ 2))) + ", " + CStr(OriginDEG)

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่ในที่สาธารณะ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

Private Sub PriToSec_Click()

```
disx = -(MoveX(CInt(PriNo.Text)) - MoveX(CInt(SecNo.Text)))
disy = MoveY(CInt(PriNo.Text)) - MoveY(CInt(SecNo.Text))
absdis = Sqr((disx ^ 2) + (disy ^ 2))
```

```
If disx <> 0 Then deg = Atn(disy / disx) * 7 * 180 / 22
```

```
If disx = 0 Then
```

```
    If disy > 0 Then deg = -90
```

```
    If disy < 0 Then deg = 90
```

```
    If disy = 0 Then deg = 0
```

```
End If
```

```
    If deg >= 360 Then deg = deg - 360
```

```
    If deg <= -360 Then deg = deg + 360
```

```
If disx >= 0 Then
```

```
    If disy > 0 Then
```

```
        deg = deg
```

```
    Else
```

```
        deg = 360 + deg
```

```
    End If
```

```
Else
```

```
    deg = 180 + deg
```

```
End If
```

```
    If deg >= 360 Then deg = deg - 360
```

```
    If deg <= -360 Then deg = deg + 360
```

```
    If deg >= 180 Then deg = deg - 360
```

```
    If deg <= -180 Then deg = deg + 360
```

```
MsgBox "x,y,abs,deg = " + CStr(disx) + "," + CStr(disy) + "," + CStr(absdis) + "," + CStr(deg)
```

End Sub

Private Sub SecondaryInfo_Click()

```
Index = CInt(SecNo.Text)
```

```
disx = -(MoveX(CInt(SecNo.Text)) - MoveX(0))
```

```
disy = MoveY(CInt(SecNo.Text)) - MoveY(0)
```

```
absdis = Sqr((disx ^ 2) + (disy ^ 2))
```

```
If disx <> 0 Then deg = Atn(disy / disx) * 7 * 180 / 22
```

```
If disx = 0 Then
```

```
    If disy > 0 Then deg = 90
```

```
    If disy < 0 Then deg = -90
```

```
    If disy = 0 Then deg = 0
```

```
End If
```

```
    If deg >= 360 Then deg = deg - 360
```

```
    If deg <= -360 Then deg = deg + 360
```

```
If disx >= 0 Then
```

```
    If disy > 0 Then
```

```
        deg = deg
```

```
    Else
```

```
        deg = 360 + deg
```

```
    End If
```

```
Else
```

```
    deg = 180 + deg
```

```
End If
```

```
    If deg >= 360 Then deg = deg - 360
```

```
    If deg <= -360 Then deg = deg + 360
```

```
    If deg >= 180 Then deg = deg - 360
```

```
    If deg <= -180 Then deg = deg + 360
```

```
AbsX = disx
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

AbsY = disy
OriginDEG = deg
MsgBox "Primary Position" + Eol + "Position Number " + CStr(Index) + Eol + "X axis distance
from origin = " + CStr(AbsX) + Eol + "Y axis distance from origin = " + CStr(AbsY) + Eol +
"Absolute distane from origin = " + CStr(Sqr((AbsX ^ 2) + (AbsY ^ 2))) + ", " + CStr(OriginDEG)
End Sub

```

```

Private Sub SelectPrimary_Click()
    PosSelect = 1
    PriStatus.Visible = True
    SecStatus.Visible = False
End Sub

```

```

Private Sub SelectSecondary_Click()
    PosSelect = 2
    SecStatus.Visible = True
    PriStatus.Visible = False
End Sub

```

```

Public Sub UpButton_Click()
    Form1.Text3.Text = "Move Up" + Eol + Form1.Text3.Text
    Form1.Text1.Text = "[Command] - Move Up" + Eol + Eol + Form1.Text1.Text
    Out8Bit = "Tx" + Chr(Bin2Dec(StrReverse("11010000"))) + HiByte(0) + LoByte(0) + Chr(0) + "R"
    + "."
    SendDATA (Out8Bit)
End Sub

```

```

Public Sub DownButton_Click()
    Form1.Text3.Text = "Move Down" + Eol + Form1.Text3.Text
    Form1.Text1.Text = "[Command] - Move Down" + Eol + Eol + Form1.Text1.Text
    Out8Bit = "Tx" + Chr(Bin2Dec(StrReverse("11010010"))) + HiByte(0) + LoByte(0) + Chr(0) + "R"
    + "."
    SendDATA (Out8Bit)
End Sub

```

```

Private Sub Command1_Click()

End Sub

```

```

Private Sub Form_Load()
    Drawed = False

```

```

    Form1.Visible = True
    ButtonActive (False)
    PWM = CInt(Text4.Text)
    For a = 0 To 150
        MoveX(a) = 9999
        DoEvents
    Next
    NegX = 0
    NegY = 0
    Direction = 0
    ScaleValue = 1
    XScale = 1
    YScale = 1
    MoveX(0) = 156
    MoveY(0) = 123
    RealX(0) = MoveX(0)
    RealY(0) = MoveY(0)

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

GraphN = 0
SecStatus.Visible = False
PosSelect = 1
End Sub

```

```

Private Sub PositionKey_Click(Index As Integer)
    If PosSelect = 1 Then PriNo.Text = CInt(Index)
    If PosSelect = 2 Then SecNo.Text = CInt(Index)
End Sub

```

```

Private Sub StopButton_Click()
    Form1.Text3.Text = "Stop !!!" + Eol + Form1.Text3.Text
    Form1.Text1.Text = "[Command] - Stop." + Eol + Eol + Form1.Text1.Text
    Out8Bit = "Tx" + Chr(Bin2Dec(StrReverse("11110000"))) + HiByte(0) + LoByte(0) + Chr(0) + "R"
    + "."
    SendDATA (Out8Bit)
End Sub

```

```

Private Sub RRotate_Click()
    Form1.Text1.Text = "[Command] - Rotate Right." + Eol + Form1.Text1.Text
    Form1.Enabled = False
    Dialog.Combo1.Clear
    Dialog.Combo1.AddItem ("องศา")
    Dialog.Combo1.AddItem ("หน่วย")
    Dialog.Combo1.ListIndex = 0
    a = InputDialog(" หมุนไปทางขวา", 7200, 1900, MRotateR)
End Sub

```

```

Private Sub LRotate_Click()
    Form1.Enabled = False
    Dialog.Combo1.Clear
    Dialog.Combo1.AddItem ("องศา")
    Dialog.Combo1.AddItem ("หน่วย")
    Dialog.Combo1.ListIndex = 0
    a = InputDialog(" หมุนไปทางซ้าย", 6200, 1900, MRotateL)
End Sub

```

```

Private Sub Reverse_Click()
    Form1.Enabled = False
    a = InputDialog(" เดินหน้าเป็นระยะ", 7200, 1600, MReverse)
End Sub

```

```

Private Sub Forward_Click()
    Form1.Enabled = False
    a = InputDialog(" เดินหน้าเป็นระยะ", 6200, 1600, MForward)
End Sub

```

```

Private Sub SendButton_Click()
    SendDATA (Text2.Text)
End Sub

```

```

Private Sub Command9_Click()

```

```

    Text1.Text = ""

```

```

    Text2.Text = ""

```

เอกสารนี้เป็นเอกสารต้นฉบับสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Text3.Text = ""
End Sub

Private Sub Form_Unload(Cancel As Integer)
    If MSComm1.PortOpen = True Then MSComm1.PortOpen = False
End Sub

Private Sub MSComm1_OnComm()
Dim TextXX As String
    If MSComm1.CommEvent > 900 Then MsgBox (CStr(MSComm1.CommEvent))

    TextXX = "L"
    If MSComm1.CommEvent = 1 Then

        End If

    If MSComm1.CommEvent = 2 Then
        InBuf = InBuf + MSComm1.Input
        DataEnd = InStr(1, InBuf, ".")
        While DataEnd <= 0
            ReportIn = InBuf
            Text1.Text = "[Incomming] - " + Trim(left(InBuf, DataEnd)) + "[" + CStr(Time) + "]" + Eol +
Text1.Text
            InBuf = Mid(InBuf, DataEnd + 1)
            DataEnd = InStr(1, InBuf, ".")
        Wend
    End If

End Sub

Private Sub Text4_Change()
    PWM = CInt(Text4.Text)
End Sub

Private Sub Timer1_Timer()
    If Direction > 180 Then Direction = Direction - 360
    If Direction < -180 Then Direction = Direction + 360
    If Len(Text3.Text) > 2000 Then
        Text3.Text = left(Text3.Text, 2000)
    End If

    If Drawed = False Then
        MapX = MoveX(GraphN) * ScaleValue
        MapY = MoveY(GraphN) * ScaleValue
        Circle (MapX, MapY), 15
        Line (MapX, MapY)-((MapX + (15 * Cos((-Direction * 22) / (7 * 180))))), (MapY + (15 * Sin((-
Direction * 22) / (7 * 180))))), QBColor(14)
        Line (MapX, MapY)-(MapX, MapY)
        Drawed = True
    End If
    Label2.Caption = "Direction = " + CStr(Direction) + " Degree."
End Sub

Private Sub Toggle_Click()
    If Toggle.Caption = "Go online" Then
        MSComm1.PortOpen = True
        Toggle.Caption = "Go offline"
        Shape1.FillColor = vbGreen

```

```

Label1.Caption = "Now Online..."
ButtonActive (True)
Else
MSComm1.PortOpen = False
Toggle.Caption = "Go online"
Shapel.FillColor = vbRed
Label1.Caption = "Now Offline..."
ButtonActive (False)
End If
End Sub

Private Sub ButtonActive(Status As Boolean)

```

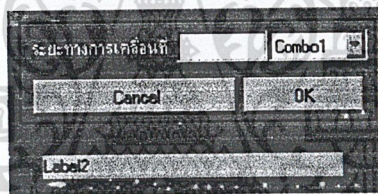
```

Forward.Enabled = Status
Reverse.Enabled = Status
RRotate.Enabled = Status
LRotate.Enabled = Status
StopButton.Enabled = Status
UpButton.Enabled = Status
DownButton.Enabled = Status

```

```
End Sub
```

Dialog1 programming code.



```
Public Out8Bit As String
```

```
Private Sub CancelButton_Click()
Unload Dialog
End Sub
```

```
Private Sub Form_Load()
Combo1.ListIndex = 3
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
Dim b, c As Integer
c = Dialog.Height
For b = c To 1 Step -150
Dialog.Height = b
DoEvents
Next b
Form1.Enabled = True
End Sub
```

```
Private Sub OKButton_Click()
Const MStop = "11110000", MRotateR = "10100000", MRotateL = "10100100", MReverse =
"01100000", MForward = "01101000"
Dim Out8Bit As String
Dim Distance As Long, PWM
Dim X As Long, Y As Long
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Const DegConst = 684 / 360
Distance = CLng(Text1.Text)
Select Case Combol.Text
Case "องศา"
Distance = Distance * DegConst
Case "เซนติเมตร"
Distance = Distance / 0.0127
Case "มิลลิเมตร"
Distance = Distance / 0.127
End Select
Select Case Port1.Caption
Case MRotateR
Form1.Text1.Text = "[Command] - Right Rotate." + Eol + Eol + Form1.Text1.Text
Form1.Text3.Text = "Right Rotate = " + CStr(Distance) + Eol + Form1.Text3.Text
Case MRotateL
Form1.Text1.Text = "[Command] - Left Rotate." + Eol + Eol + Form1.Text1.Text
Form1.Text3.Text = "Left Rotate = " + CStr(Distance) + Eol + Form1.Text3.Text
Case MReverse
Form1.Text1.Text = "[Command] - Reverse Move." + Eol + Eol + Form1.Text1.Text
Form1.Text3.Text = "Reverse Move = " + CStr(Distance) + Eol + Form1.Text3.Text
Case MForward
Form1.Text1.Text = "[Command] - Forward Move." + Eol + Eol + Form1.Text1.Text
Form1.Text3.Text = "Forward Move = " + CStr(Distance) + Eol + Form1.Text3.Text
End Select

If Text1.Text <> "" Then
If (Port1.Caption = MRotateL) Or (Port1.Caption = MRotateR) Then
Form1.DownButton_Click
Form1.ReportIn = "X"
While Form1.ReportIn <> "Complete."
DoEvents
Wend
For aaa = 1 To 1000
DoEvents
Next
Form1.ReportIn = ""
End If

Out8Bit = "Tx" + Chr(Bin2Dec(StrReverse(Port1.Caption))) + HiByte(Distance) +
LoByte(Distance) + Chr(150) + "R" + "."
SendDATA (Out8Bit)
If (Port1.Caption = MRotateL) Or (Port1.Caption = MRotateR) Then Form1.Rotate = True
Select Case Port1.Caption
Case MRotateR
Form1.Direction = Form1.Direction - (Distance / DegConst)
Case MRotateL
Form1.Direction = Form1.Direction + (Distance / DegConst)
End Select
If Port1.Caption <> MRotateR And Port1.Caption <> MRotateL And Port1.Caption <> MStop
Then
If Port1.Caption = MReverse Then Distance = -Distance
X = Round(Cos((Form1.Direction / 180) * (22 / 7)) * Distance) / 10
Y = -Round(Sin((Form1.Direction / 180) * (22 / 7)) * Distance) / 10
Form1.MapRedraw X, Y
End If

If (Port1.Caption = MRotateL) Or (Port1.Caption = MRotateR) Then
Form1.ReportIn = "X"
While Form1.ReportIn <> "Complete."

```

```

    DoEvents
Wend
For aaa = 1 To 10000
    DoEvents
Next
Form1.UpButton_Click
End If
    Unload Dialog
End If
End Sub

```

```

Private Sub Text1_Change()
Dim TextLen
    TextLen = Len(Text1.Text)
    If TextLen < 0 Then
        If Asc(Right(Text1.Text, 1)) > Asc("9") Or Asc(Right(Text1.Text, 1)) < Asc("0") Then
            Text1.Text = ""
        End If
    End If
End Sub

```

```

Private Sub Text1_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = 13 Then OKButton_Click
End Sub

```

Function module programming code.

```

Public Sub SendDATA(DOut As String)
Dim DataOut, K
    DataOut = DOut
    Form1.MSComm1.Output = DataOut
    Form1.Text5.Text = " "
    For K = Len(DataOut) To 1 Step -1
        Form1.Text5.Text = "Byte " + CStr(K) + " = " + Str2Bin(Mid(DataOut, K, 1)) + "b" + Eol +
Form1.Text5.Text
        Next K
        For a = 1 To Len(DOut)
            If Len(Hex(Asc(Mid(DOut, a, 1)))) = 2 Then
                TextXX = TextXX + " " + Hex(Asc(Mid(DOut, a, 1)))
            Else
                TextXX = TextXX + " 0" + Hex(Asc(Mid(DOut, a, 1)))
            End If
        Next
        TextXX = Right(TextXX, Len(TextXX) - 1) + " "
        Form1.Text1.Text = "[Outgoing] - " + TextXX + "[" + CStr(Time) + "]" + Eol +
Form1.Text1.Text
        Form1.Drawed = False
    End Sub

```

```

Public Function InputDialog(Quest As String, X, Y As Integer, Control As String) As Integer
Dim left As Integer
    Dialog.left = X
    Dialog.Top = Y
    Dialog.Caption = Head
    Dialog.Label1.Caption = Quest
    Dialog.Show
    For a = 1 To 1350 Step 40
        Dialog.Height = a
        DoEvents
    Next a

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dialog.Port1.Caption = Control
End Function
```

```
Public Function Offline() As Boolean
  Offline = False
  If Form1.MSComm1.PortOpen = False Then
    MsgBox ("You are currently offline.")
    Offline = True
  End If
End Function
```

```
Public Function Eol()
  Eol = Chr(13) + Chr(10)
End Function
```

```
Public Function Str2Bin(CIn As String) As String
  Dim n, InAsc As Integer
  InAsc = Asc(CIn)
  For n = 7 To 0 Step -1
    If InAsc / (2 ^ n) >= 1 Then
      Str2Bin = Str2Bin + "1"
      InAsc = InAsc - (2 ^ n)
    Else
      Str2Bin = Str2Bin + "0"
    End If
  Next n
End Function
```

```
Public Function Bin2Dec(CIn As String) As Integer
  Dim n As Integer
  CIn = StrReverse(CIn)
  Bin2Dec = 0
  For n = Len(CIn) To 1 Step -1
    Bin2Dec = Bin2Dec + (2 ^ ((n - 1) * CInt(Mid(CIn, n, 1))))
  Next n
End Function
```

```
Public Function HiByte(Value As Long) As String
  Dim n As Integer, vin As Long, out As String
  out = ""
  vin = Value
  For n = 15 To 0 Step -1
    If vin / (2 ^ n) >= 1 Then
      out = out + "1"
      vin = vin - (2 ^ n)
    Else
      out = out + "0"
    End If
  Next n
  HiByte = Chr(Bin2Dec(Mid(out, 1, 8)))
End Function
```

```
Public Function LoByte(Value As Long) As String
  Dim n As Integer, vin As Long, out As String
  out = ""
  vin = Value
  For n = 15 To 0 Step -1
    If vin / (2 ^ n) >= 1 Then
      out = out + "1"
      vin = vin - (2 ^ n)
    Else
```

```
    out = out + "0"  
  End If  
Next n  
LoByte = Chr(Bin2Dec(Mid(out, 9, 18)))  
End Function
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

DATASHEET



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



LMD18200 3A, 55V H-Bridge

General Description

The LMD18200 is a 3A H-Bridge designed for motion control applications. The device is built using a multi-technology process which combines bipolar and CMOS control circuitry with DMOS power devices on the same monolithic structure. Ideal for driving DC and stepper motors, the LMD18200 accommodates peak output currents up to 6A. An innovative circuit which facilitates low-loss sensing of the output current has been implemented.

Features

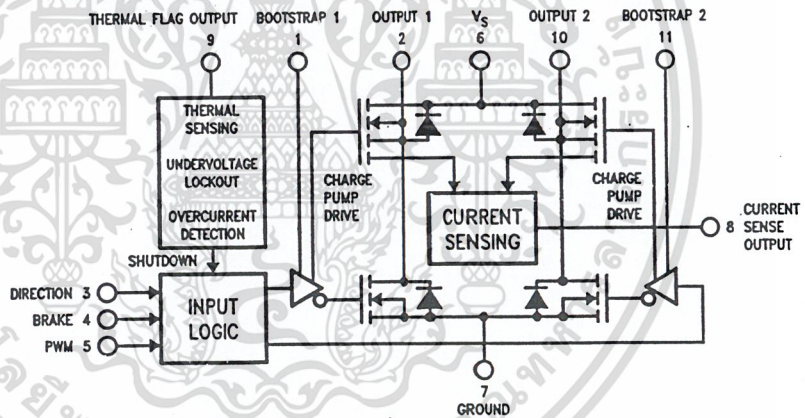
- Delivers up to 3A continuous output
- Operates at supply voltages up to 55V
- Low R_{DS(ON)} typically 0.3Ω per switch

- TTL and CMOS compatible inputs
- No "shoot-through" current
- Thermal warning flag output at 145°C
- Thermal shutdown (outputs off) at 170°C
- Internal clamp diodes
- Shorted load protection
- Internal charge pump with external bootstrap capability

Applications

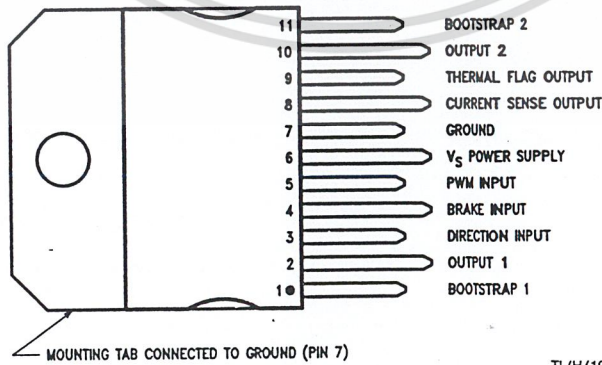
- DC and stepper motor drives
- Position and velocity servomechanisms
- Factory automation robots
- Numerically controlled machinery
- Computer printers and plotters

Functional Diagram



TL/H/10568-1

Connection Diagram and Ordering Information



Order Number LMD18200T
See NS Package TA11B

Top View

TL/H/10568-2

Absolute Maximum Ratings (Note 1)

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

Total Supply Voltage (V _S , Pin 6)	60V
Voltage at Pins 3, 4, 5, 8 and 9	12V
Voltage at Bootstrap Pins (Pins 1 and 11)	V _{OUT} + 16V
Peak Output Current (200 ms)	6A
Continuous Output Current (Note 2)	3A
Power Dissipation (Note 3)	25W

Power Dissipation (T _A = 25°C, Free Air)	3W
Junction Temperature, T _{J(max)}	150°C
ESD Susceptibility (Note 4)	1500V
Storage Temperature, T _{STG}	-65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	300°C

Operating Ratings (Note 1)

Junction Temperature, T _J	-40°C to +125°C
V _S Supply Voltage	+12V to +55V

Electrical Characteristics

The following specifications apply for V_S = 42V, unless otherwise specified. Boldface limits apply over the entire operating temperature range, -40°C ≤ T_J ≤ +125°C, all other limits are for T_A = T_J = 25°C. (Note 5)

Symbol	Parameter	Conditions	Typ	Limit	Units
R _{DS(ON)}	Switch ON Resistance	Output Current = 3A (Note 6)	0.33	0.4/0.6	Ω (max)
R _{DS(ON)}	Switch ON Resistance	Output Current = 6A (Note 6)	0.33	0.4/0.6	Ω (max)
V _{CLAMP}	Clamp Diode Forward Drop	Clamp Current = 3A (Note 6)	1.2	1.5	V (max)
V _{IL}	Logic Low Input Voltage	Pins 3, 4, 5		-0.1 0.8	V (min) V (max)
I _{IL}	Logic Low Input Current	V _{IN} = -0.1V, Pins = 3, 4, 5		-10	μA (max)
V _{IH}	Logic High Input Voltage	Pins 3, 4, 5		2 12	V (min) V (max)
I _{IH}	Logic High Input Current	V _{IN} = 12V, Pins = 3, 4, 5		10	μA (max)
	Current Sense Output	I _{OUT} = 1A (Note 8)	377	325/300 425/450	μA (min) μA (max)
	Current Sense Linearity	1A ≤ I _{OUT} ≤ 3A (Note 7)	±6	±9	%
	Undervoltage Lockout	Outputs turn OFF		9 11	V (min) V (max)
T _{JW}	Warning Flag Temperature	Pin 9 ≤ 0.8V, I _L = 2 mA	145		°C
V _{F(ON)}	Flag Output Saturation Voltage	T _J = T _{JW} , I _L = 2 mA	0.15		V
I _{F(OFF)}	Flag Output Leakage	V _F = 12V	0.2	10	μA (max)
T _{JSD}	Shutdown Temperature	Outputs Turn OFF	170		°C
I _S	Quiescent Supply Current	All Logic Inputs Low	13	25	mA (max)
t _{Don}	Output Turn-On Delay Time	Sourcing Outputs, I _{OUT} = 3A Sinking Outputs, I _{OUT} = 3A	300 300		ns ns
t _{On}	Output Turn-On Switching Time	Bootstrap Capacitor = 10 nF Sourcing Outputs, I _{OUT} = 3A Sinking Outputs, I _{OUT} = 3A	100 80		ns ns
t _{DoFF}	Output Turn-Off Delay Times	Sourcing Outputs, I _{OUT} = 3A Sinking Outputs, I _{OUT} = 3A	200 200		ns ns
t _{off}	Output Turn-Off Switching Times	Bootstrap Capacitor = 10 nF Sourcing Outputs, I _{OUT} = 3A Sinking Outputs, I _{OUT} = 3A	75 70		ns ns
t _{pw}	Minimum Input Pulse Width	Pins 3, 4 and 5	1		μs
t _{cpr}	Charge Pump Rise Time	No Bootstrap Capacitor	20		μs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Characteristics Notes

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its rated operating conditions.

Note 2: See Application Information for details regarding current limiting.

Note 3: The maximum power dissipation must be derated at elevated temperatures and is a function of $T_{J(max)}$, θ_{JA} , and T_A . The maximum allowable power dissipation at any temperature is $P_{D(max)} = (T_{J(max)} - T_A)/\theta_{JA}$, or the number given in the Absolute Ratings, whichever is lower. The typical thermal resistance from junction to case (θ_{JC}) is 1.0°C/W and from junction to ambient (θ_{JA}) is 30°C/W . For guaranteed operation $T_{J(max)} = 125^\circ\text{C}$.

Note 4: Human-body model, 100 pF discharged through a 1.5 k Ω resistor. Except Bootstrap pins (pins 1 and 11) which are protected to 1000V of ESD.

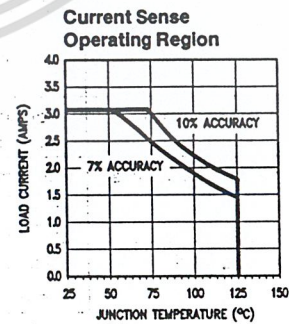
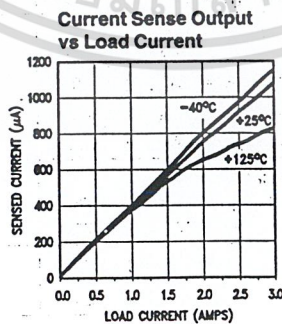
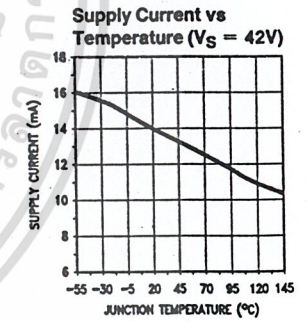
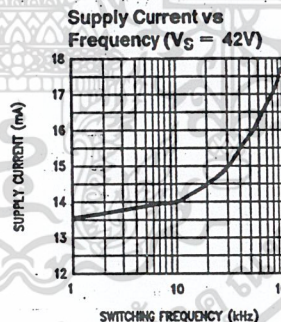
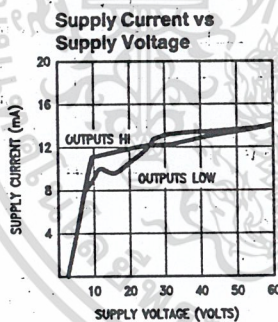
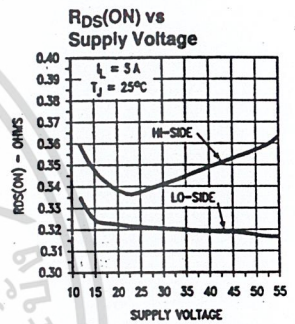
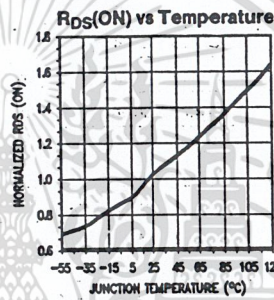
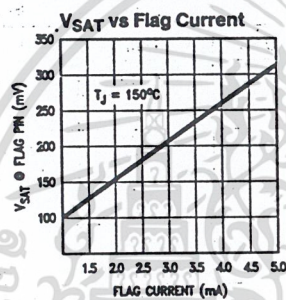
Note 5: All limits are 100% production tested at 25°C. Temperature extreme limits are guaranteed via correlation using accepted SQC (Statistical Quality Control) methods. All limits are used to calculate AOQL, (Average Outgoing Quality Level).

Note 6: Output currents are pulsed ($t_W < 2$ ms, Duty Cycle $< 5\%$).

Note 7: Regulation is calculated relative to the current sense output value with a 1A load.

Note 8: Selections for tighter tolerance are available. Contact factory.

Typical Performance Characteristics



TL/H/10568-3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pinout Description (See Connection Diagram)

Pin 1, BOOTSTRAP 1 Input: Bootstrap capacitor pin for half H-bridge number 1. The recommended capacitor (10 nF) is connected between pins 1 and 2.

Pin 2, OUTPUT 1: Half H-bridge number 1 output.

Pin 3, DIRECTION Input: See Table I. This input controls the direction of current flow between OUTPUT 1 and OUTPUT 2 (pins 2 and 10) and, therefore, the direction of rotation of a motor load.

Pin 4, BRAKE Input: See Table I. This input is used to brake a motor by effectively shorting its terminals. When braking is desired, this input is taken to a logic high level and it is also necessary to apply logic high to PWM input, pin 5. The drivers that short the motor are determined by the logic level at the DIRECTION input (Pin 3): with Pin 3 logic high, both current sourcing output transistors are ON; with Pin 3 logic low, both current sinking output transistors are ON. All output transistors can be turned OFF by applying a logic high to Pin 4 and a logic low to PWM input Pin 5; in this case only a small bias current (approximately -1.5 mA) exists at each output pin.

Pin 5, PWM Input: See Table I. How this input (and DIRECTION input, Pin 3) is used is determined by the format of the PWM Signal.

Pin 6, Vs Power Supply

Pin 7, GROUND Connection: This pin is the ground return, and is internally connected to the mounting tab.

Pin 8, CURRENT SENSE Output: This pin provides the sourcing current sensing output signal, which is typically 377 $\mu\text{A}/\text{A}$.

Pin 9, THERMAL FLAG Output: This pin provides the thermal warning flag output signal. Pin 9 becomes active-low at 145°C (junction temperature). However the chip will not shut itself down until 170°C is reached at the junction.

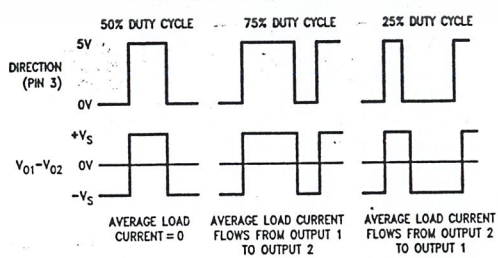
Pin 10, OUTPUT 2: Half H-bridge number 2 output.

Pin 11, BOOTSTRAP 2 Input: Bootstrap capacitor pin for Half H-bridge number 2. The recommended capacitor (10 nF) is connected between pins 10 and 11.

TABLE I. Logic Truth Table

PWM	Dir	Brake	Active Output Drivers
H	H	L	Source 1, Sink 2
H	L	L	Sink 1, Source 2
L	X	L	Source 1, Source 2
H	H	H	Source 1, Source 2
H	L	H	Sink 1, Sink 2
L	X	H	NONE

Locked Anti-Phase PWM Control



TL/H/10568-4

Application Information

TYPES OF PWM SIGNALS

The LMD18200 readily interfaces with different forms of PWM signals. Use of the part with two of the more popular forms of PWM is described in the following paragraphs.

Simple, locked anti-phase PWM consists of a single, variable duty-cycle signal in which is encoded both direction and amplitude information. A 50% duty-cycle PWM signal represents zero drive, since the net value of voltage (integrated over one period) delivered to the load is zero. For the LMD18200, the PWM signal drives the direction input (pin 3) and the PWM input (pin 5) is tied to logic high.

Sign/magnitude PWM consists of separate direction (sign) and amplitude (magnitude) signals. The (absolute) magnitude signal is duty-cycle modulated, and the absence of a pulse signal (a continuous logic low level) represents zero drive. Current delivered to the load is proportional to pulse width. For the LMD18200, the DIRECTION input (pin 3) is driven by the sign signal and the PWM input (pin 5) is driven by the magnitude signal.

USING THE CURRENT SENSE OUTPUT

The CURRENT SENSE output (pin 8) has a sensitivity of 377 μA per ampere of output current. For optimal accuracy and linearity of this signal, the value of voltage generating resistor between pin 8 and ground should be chosen to limit the maximum voltage developed at pin 8 to 5V, or less. The maximum voltage compliance is 12V.

It should be noted that the recirculating currents (free wheeling currents) are ignored by the current sense circuitry. Therefore, only the currents in the upper sourcing outputs are sensed.

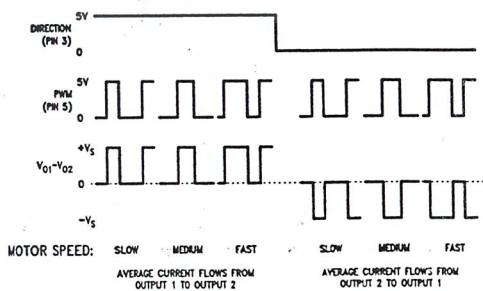
USING THE THERMAL WARNING FLAG

The THERMAL FLAG output (pin 9) is an open collector transistor. This permits a wired OR connection of thermal warning flag outputs from multiple LMD18200's, and allows the user to set the logic high level of the output signal swing to match system requirements. This output typically drives the interrupt input of a system controller. The interrupt service routine would then be designed to take appropriate steps, such as reducing load currents or initiating an orderly system shutdown. The maximum voltage compliance on the flag pin is 12V.

SUPPLY BYPASSING

During switching transitions the levels of fast current changes experienced may cause troublesome voltage transients across system stray inductance.

Sign/Magnitude PWM Control



TL/H/10568-5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Application Information (Continued)

It is normally necessary to bypass the supply rail with a high quality capacitor(s) connected as close as possible to the V_S Power Supply (Pin 6) and GROUND (Pin 7). A 1 μF high-frequency ceramic capacitor is recommended. Care should be taken to limit the transients on the supply pin below the Absolute Maximum Rating of the device. When operating the chip at supply voltages above 40V a voltage suppressor (transorb) such as P6KE62A is recommended from supply to ground. Typically the ceramic capacitor can be eliminated in the presence of the voltage suppressor. Note that when driving high load currents a greater amount of supply bypass capacitance (in general at least 100 μF per Amp of load current) is required to absorb the recirculating currents of the inductive loads.

CURRENT LIMITING

Current limiting protection circuitry has been incorporated into the design of the LMD18200. With any power device it is important to consider the effects of the substantial surge currents through the device that may occur as a result of shorted loads. The protection circuitry monitors this increase in current (the threshold is set to approximately 10 Amps) and shuts off the power device as quickly as possible in the event of an overload condition. In a typical motor driving application the most common overload faults are caused by shorted motor windings and locked rotors. Under these conditions the inductance of the motor (as well as any series inductance in the V_{CC} supply line) serves to reduce the magnitude of a current surge to a safe level for the LMD18200. Once the device is shut down, the control circuitry will periodically try to turn the power device back on. This feature allows the immediate return to normal operation in the event that the fault condition has been removed. While the fault remains however, the device will cycle in and out of thermal shutdown. This can create voltage transients on the V_{CC} supply line and therefore proper supply bypassing techniques are required.

The most severe condition for any power device is a direct, hard-wired ("screwdriver") long term short from an output to ground. This condition can generate a surge of current through the power device on the order of 15 Amps and require the die and package to dissipate up to 500 Watts of power for the short time required for the protection circuitry to shut off the power device. This energy can be destructive, particularly at higher operating voltages (>30V) so

some precautions are in order. Proper heat sink design is essential and it is normally necessary to heat sink the V_{CC} supply pin (pin 6) with 1 square inch of copper on the PCB.

INTERNAL CHARGE PUMP AND USE OF BOOTSTRAP CAPACITORS

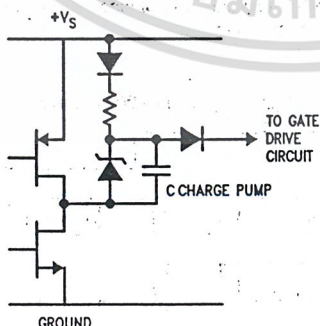
To turn on the high-side (sourcing) DMOS power devices, the gate of each device must be driven approximately 8V more positive than the supply voltage. To achieve this an internal charge pump is used to provide the gate drive voltage. As shown in Figure 1, an internal capacitor is alternately switched to ground and charged to about 14V, then switched to V_S supply thereby providing a gate drive voltage greater than V_S supply. This switching action is controlled by a continuously running internal 300 kHz oscillator. The rise time of this drive voltage is typically 20 μs which is suitable for operating frequencies up to 1 kHz.

For higher switching frequencies, the LMD18200 provides for the use of external bootstrap capacitors. The bootstrap principle is in essence a second charge pump whereby a large value capacitor is used which has enough energy to quickly charge the parasitic gate input capacitance of the power device resulting in much faster rise times. The switching action is accomplished by the power switches themselves (Figure 2). External 10 nF capacitors, connected from the outputs to the bootstrap pins of each high-side switch provide typically less than 100 ns rise times allowing switching frequencies up to 500 kHz.

INTERNAL PROTECTION DIODES

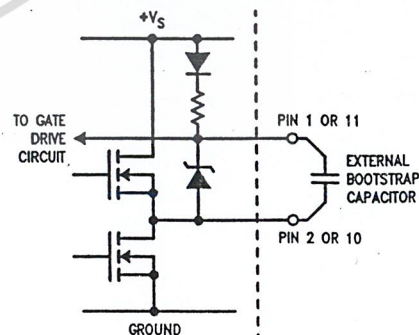
A major consideration when switching current through inductive loads is protection of the switching power devices from the large voltage transients that occur. Each of the four switches in the LMD18200 have a built-in protection diode to clamp transient voltages exceeding the positive supply or ground to a safe diode voltage drop across the switch.

The reverse recovery characteristics of these diodes, once the transient has subsided, is important. These diodes must come out of conduction quickly and the power switches must be able to conduct the additional reverse recovery current of the diodes. The reverse recovery time of the diodes protecting the sourcing power devices is typically only 70 ns with a reverse recovery current of 1A when tested with a full 6A of forward current through the diode. For the sinking devices the recovery time is typically 100 ns with 4A of reverse current under the same conditions.



TL/H/10568-6

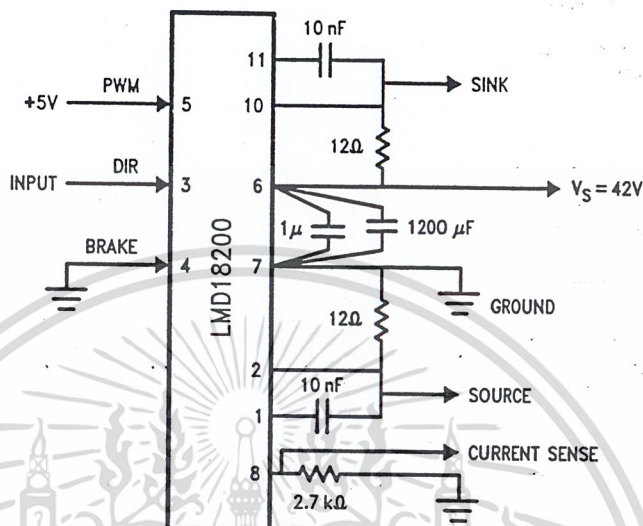
FIGURE 1. Internal Charge Pump Circuitry



TL/H/10568-7

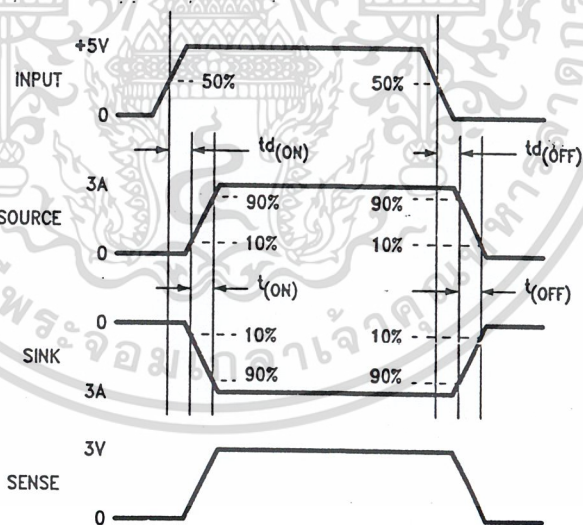
FIGURE 2. Bootstrap Circuitry

Test Circuit



TL/H/10568-8

Switching Time Definitions



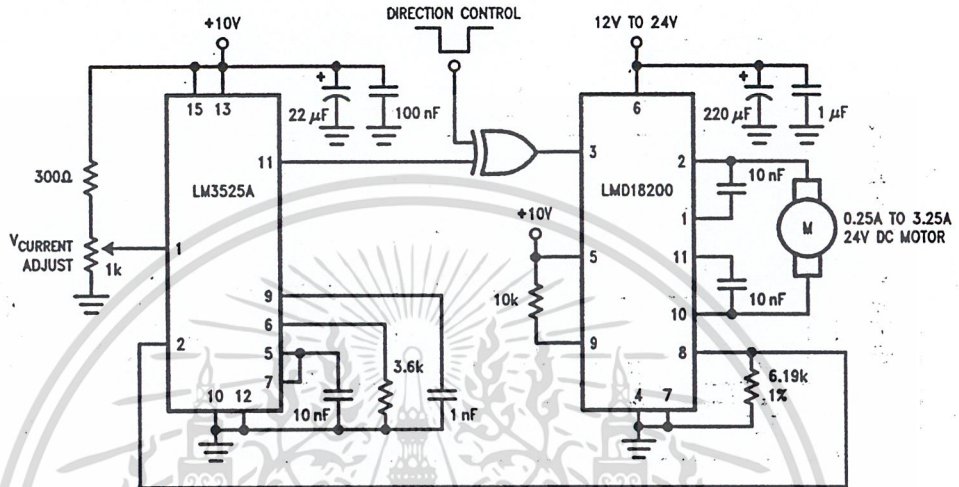
TL/H/10568-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Applications (Continued)

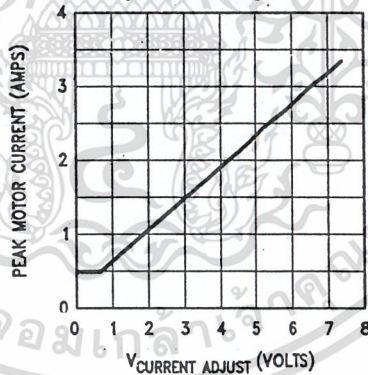
TORQUE REGULATION

Locked Anti-Phase Control of a brushed DC motor. Current sense output of the LMD18200 provides load sensing. The LM3525A is a general purpose PWM controller.



TL/H/10568-12

Peak Motor Current vs Adjustment Voltage



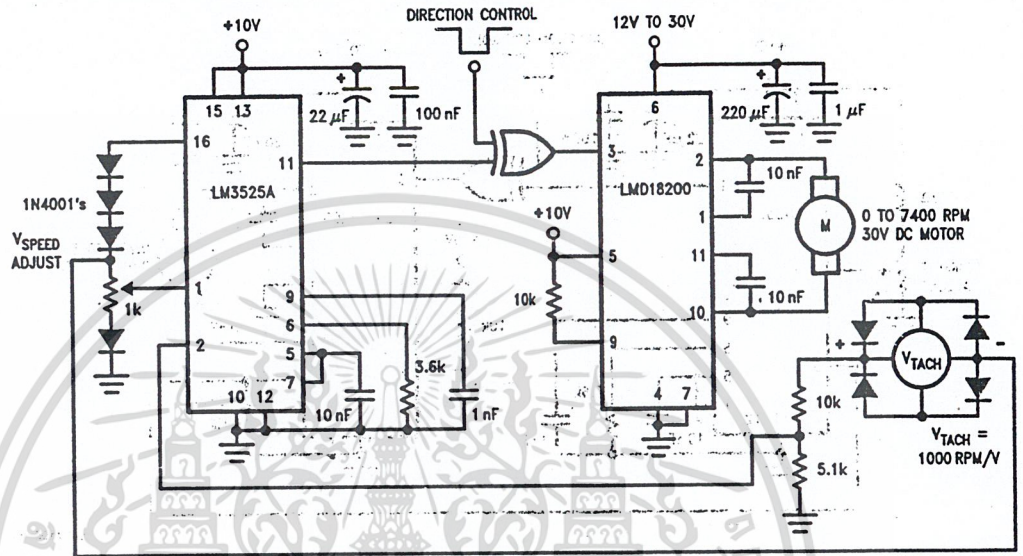
TL/H/10568-13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

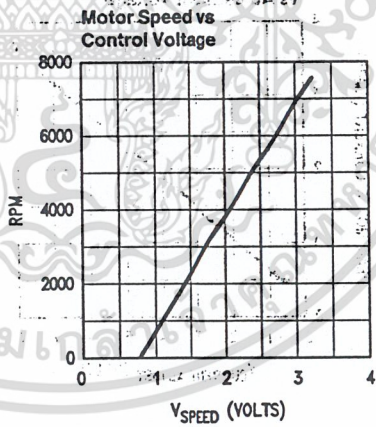
Typical Applications (Continued)

VELOCITY REGULATION

Utilizes tachometer output from the motor to sense motor speed for a locked anti-phase control loop.



TL/H/10568-14



TL/H/10568-15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

quires
power
This
within
rticu-
ween
(in 8)
nally
1 this
ernal
apply
slight
the
re in-
ween
pikes
have
f the



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 choppe 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.

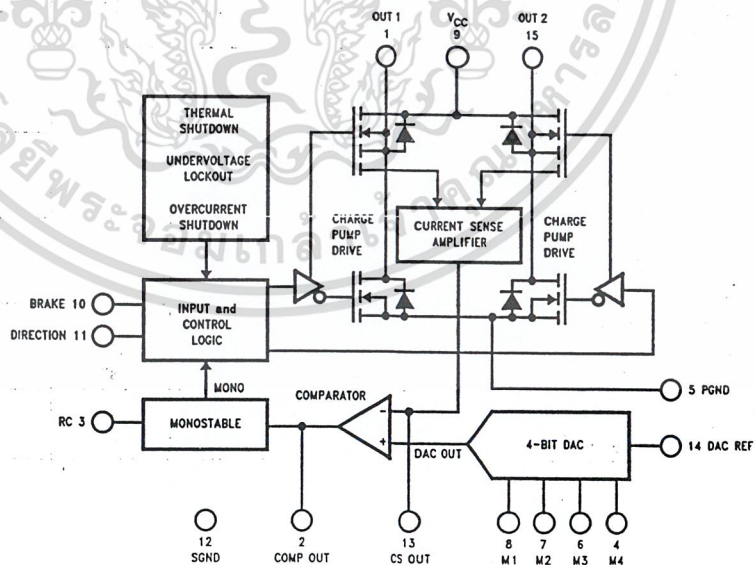
Features

- DMOS power stage rated at 55V and 3A continuous
- Low $R_{DS(ON)}$ of typically 0.3Ω 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 $T_J = 155^\circ\text{C}$
- 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

TL/H/11878-1

23-10

LMD18245

4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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)	25W
TO-220 (T _A = 25°C, Free Air)	3.5W

ESD Susceptibility (Note 4)

1500V

Storage Temperature Range (T_S)

-65°C to +150°C

Lead Temperature (Soldering, 10 seconds)

300°C

Operating Conditions (Note 1)Temperature Range (T_J) (Note 3)

-40°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 The following specifications apply for V_{CC} = +42V, unless otherwise stated. **Bold-face limits apply over the operating temperature range, -40°C ≤ T_J ≤ +125°C.** All other limits apply for T_A = T_J = 25°C. (Note 2)

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 mA (max)
POWER OUTPUT STAGE					
R _{DS(ON)}	Switch ON Resistance	I _{LOAD} = 3A	0.3	0.4 0.6	Ω (max) Ω (max)
		I _{LOAD} = 6A	0.3	0.4 0.6	Ω (max) Ω (max)
V _{DIODE}	Body Diode Forward Voltage	I _{DIODE} = 3A	1.0	1.5	V V(max)
T _{rr}	Diode Reverse Recovery Time	I _{DIODE} = 1A	80		ns
Q _{rr}	Diode Reverse Recovery Charge	I _{DIODE} = 1A	40		nC
t _{D(ON)}	Output Turn ON Delay Time Sourcing Outputs Sinking Outputs	I _{LOAD} = 3A	5		μs
		I _{LOAD} = 3A	900		ns
t _{D(OFF)}	Output Turn OFF Delay Time Sourcing Outputs Sinking Outputs	I _{LOAD} = 3A	600		ns
		I _{LOAD} = 3A	400		ns
t _{ON}	Output Turn ON Switching Time Sourcing Outputs Sinking Outputs	I _{LOAD} = 3A	40		μs
		I _{LOAD} = 3A	1		μs
t _{OFF}	Output Turn OFF Switching Time Sourcing Outputs Sinking Outputs	I _{LOAD} = 3A	200		ns
		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

Electrical Characteristics The following specifications apply for $V_{CC} = +42V$, unless otherwise stated. **Bold-face limits apply over the operating temperature range, $-40^{\circ}C \leq T_J \leq +125^{\circ}C$.** All other limits apply for $T_A = T_J = 25^{\circ}C$. (Note 2) (Continued)

Symbol	Parameter	Conditions	Typical (Note 5)	Limit (Note 5)	Units (Limits)
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	$0.5A \leq I_{LOAD} \leq 3A$ (Note 7)	± 6	± 9	% %(max)
	Current Sense Offset	$I_{LOAD} = 0A$	5	20	μA μA (max)
DIGITAL-TO-ANALOG CONVERTER (DAC)					
	Resolution			4	Bits (min)
	Monotonicity			4	Bits (min)
	Total Unadjusted Error		0.125	0.25 0.5	LSB (max) 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		6.27		V
	Comparator Low Output Level		88		mV
	Comparator Output Current Source		0.2		mA
	Comparator Output Current 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		$^{\circ}C/W$
θ_{JA}	Junction-to-Ambient, TO-220		35		$^{\circ}C/W$
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)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Characteristics The following specifications apply for $V_{CC} = +42V$, unless otherwise stated. **Bold-face limits apply over the operating temperature range, $-40^{\circ}C \leq T_J \leq +125^{\circ}C$.** All other limits apply for $T_A = T_J = 25^{\circ}C$. (Note 2) (Continued)

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 $125^{\circ}C$ 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 $125^{\circ}C$. If the junction temperature exceeds $155^{\circ}C$, 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 $25^{\circ}C$. 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.

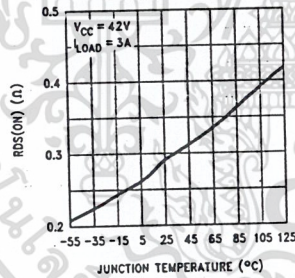
Note 6: Asymmetric turn OFF and ON delay times and switching times ensure a switch turns OFF before the other switch in the same half H-bridge begins to turn ON (preventing momentary short circuits between the power supply and ground). The transitional period during which both switches are OFF is commonly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

Note 7: (LOAD, ISENSE) 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.

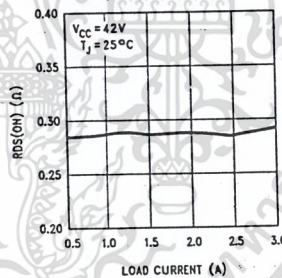
Note 8: Turn OFF delay, t_{DELAY} , is defined as the time from the voltage at the output of the current sense amplifier reaching the DAC output voltage to the lower DMOS switch beginning to turn OFF. With $V_{CC} = 32V$, DIRECTION high, and 200 Ω connected between OUT1 and V_{CC} , the voltage at RC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

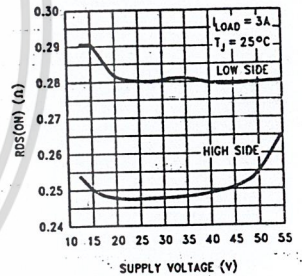
RDS(ON) vs Temperature



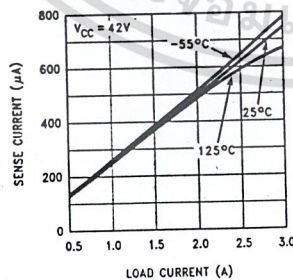
RDS(ON) vs Load Current



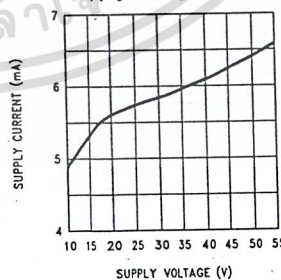
RDS(ON) vs Supply Voltage



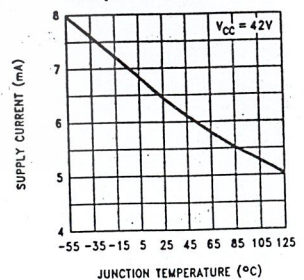
Current Sense Output vs Load Current



Supply Current vs Supply Voltage

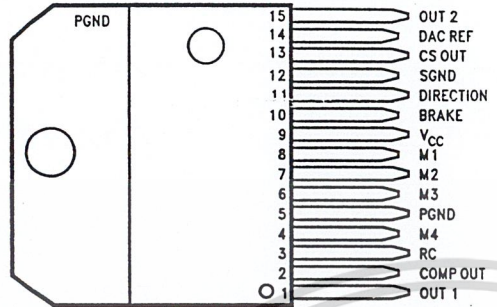


Supply Current vs Temperature



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Connection Diagram



Top View

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

TL/H/11878-2

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 I. 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 I.
- 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 I. 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 (Figures 1a and 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 (Figures 1b and 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 (Figures 1a and 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 (Figures 1c and 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)

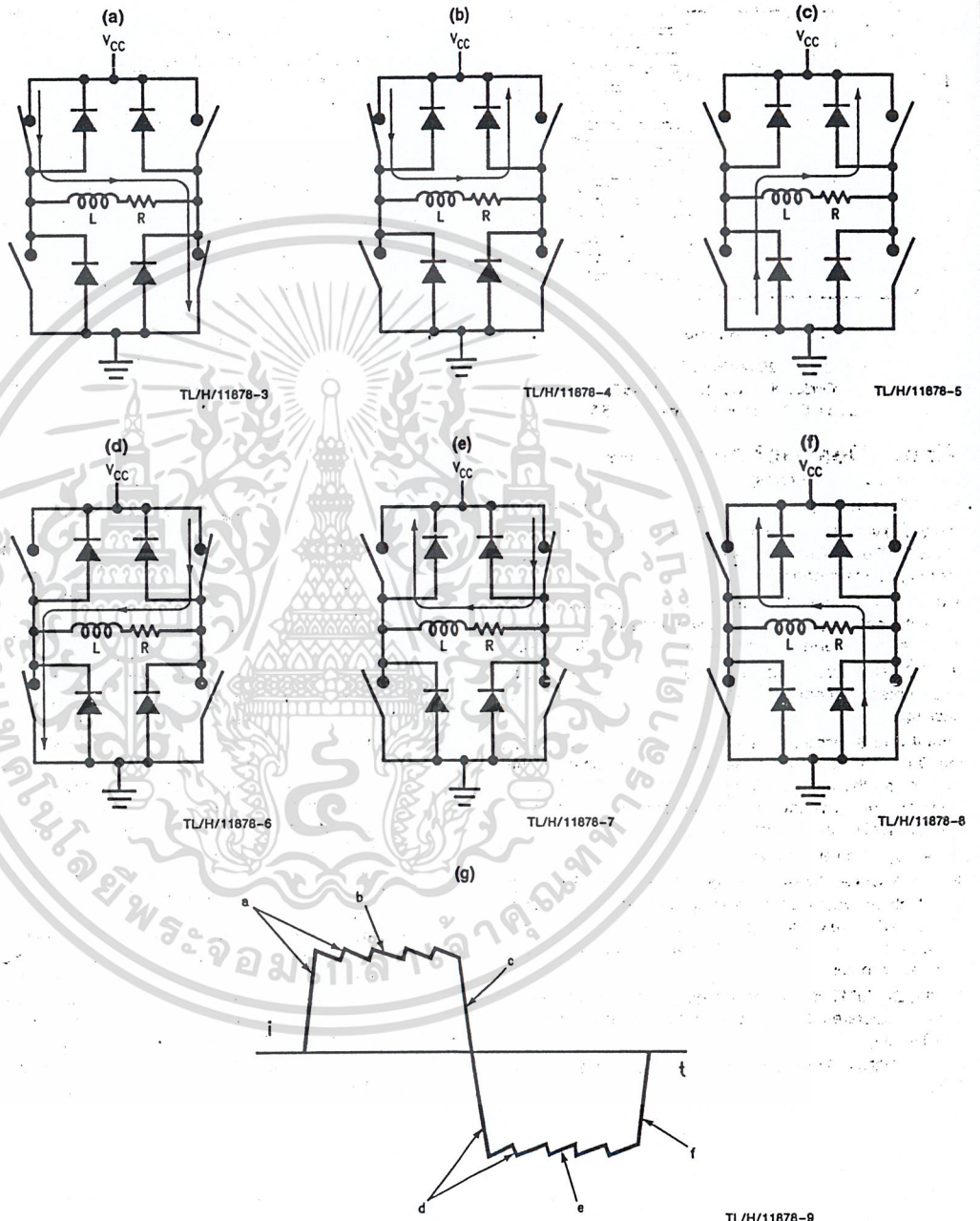


FIGURE 1. Chopper Amplifier Chopping States: Full V_{CC} Applied Across the Winding (a) and (d), Shorted Winding (b) and (e), Winding Current Decays During a Change in the Direction of the Winding Current (c) and (f), and the Chopped Winding Current (g)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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).

178-5

78-B

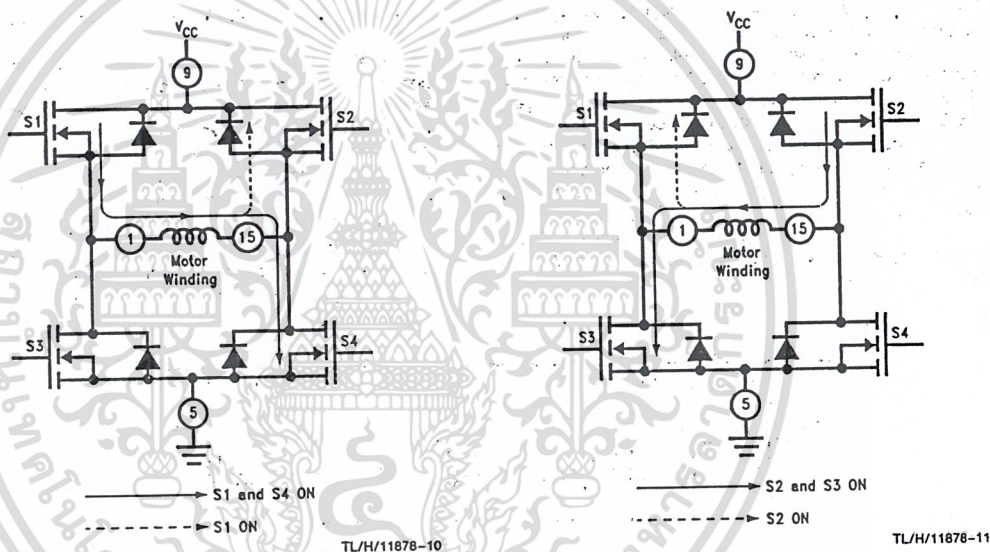


FIGURE 2. The DMOS H-Bridge

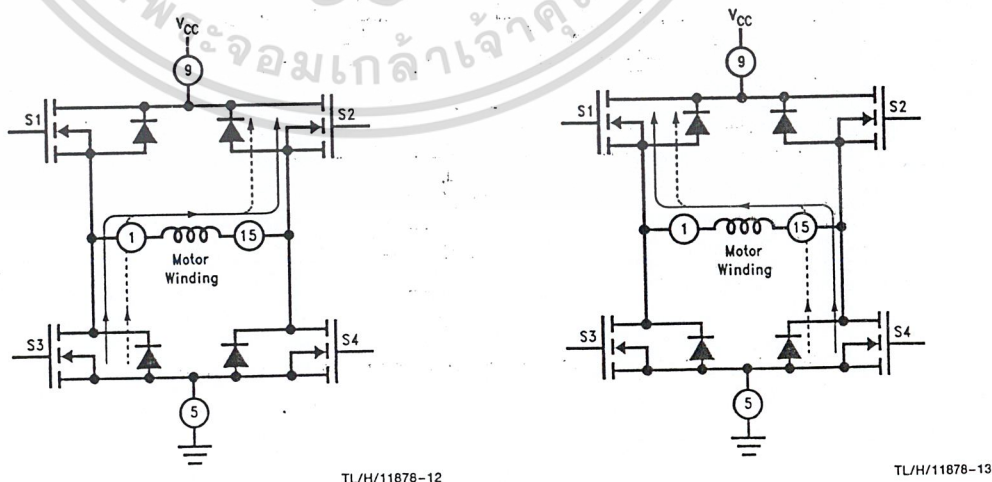


FIGURE 3. Decay Paths for Initial Winding Current During a Change in the Direction of the Winding Current

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 μA 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 μA 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 μA 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 $V_{\text{DAC REF}} \times D/16$, 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 (Figures 1b and 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

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

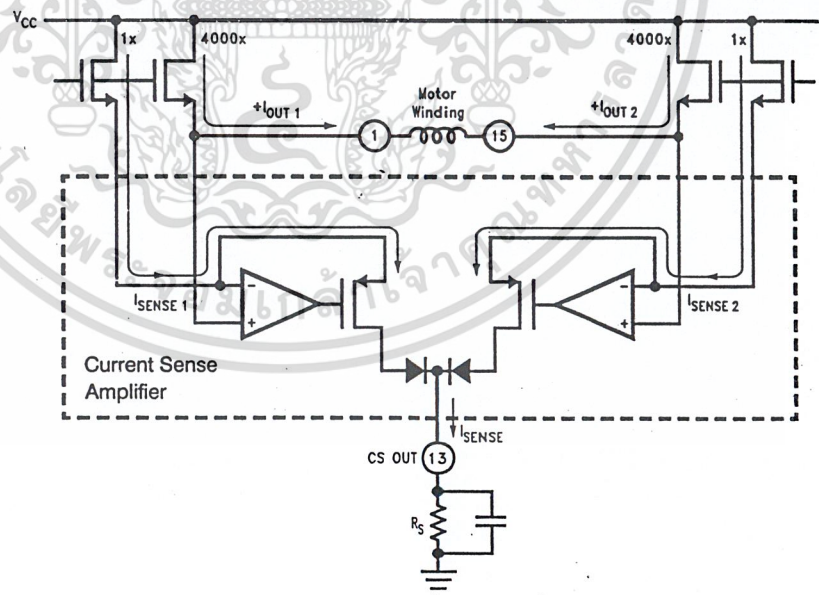


FIGURE 4. The Source Switches of the Power Bridge and the Current Sense Amplifier

TL/H/11878-14

CS
CS

at
(0-
the
ant
ter-
fied
the
5V.

it of
the
idge
g to
and
cted
e or

CS
ping

as.

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 μ F high frequency ceramic capacitor and a large-value aluminum electrolytic capacitor is highly recommended. A value of 100 μ F 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 μ s 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.

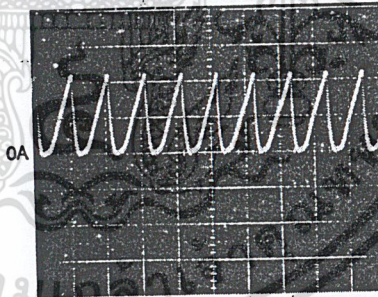
In the case of a locked rotor, the inductance of the winding tends to limit the rate of change of the fault current to a value easily handled by the protection circuitry. In the case of a low inductance short from either output to ground or between outputs, the fault current could surge past the 12A 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 ($\geq 30V$), 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.



TL/H/11878-15

Trace: Fault Current at 5A/div
Horizontal: 20 μ s/div

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The Typical Application

Figure 6 shows the typical application, the power stage of a chopper drive for bipolar stepper motors. The 20 kΩ resistor and 2.2 nF capacitor connected between RC and ground set the off-time at about 48 μs, and the 20 kΩ 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 μprocessor or μcontroller usually provides the digital control signals.

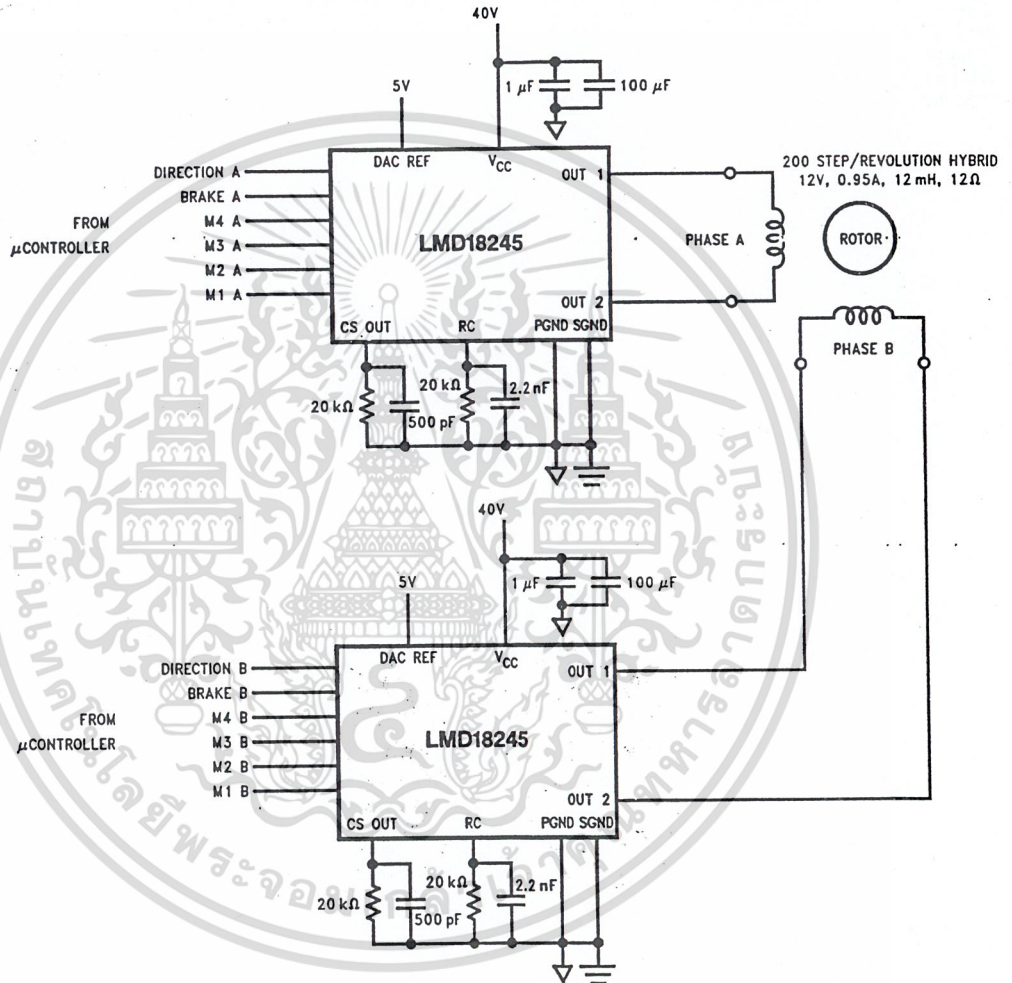


FIGURE 6. Typical Application Circuit for Driving Bipolar Stepper Motors

TL/H/11878-16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The Typical Application (Continued)

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

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

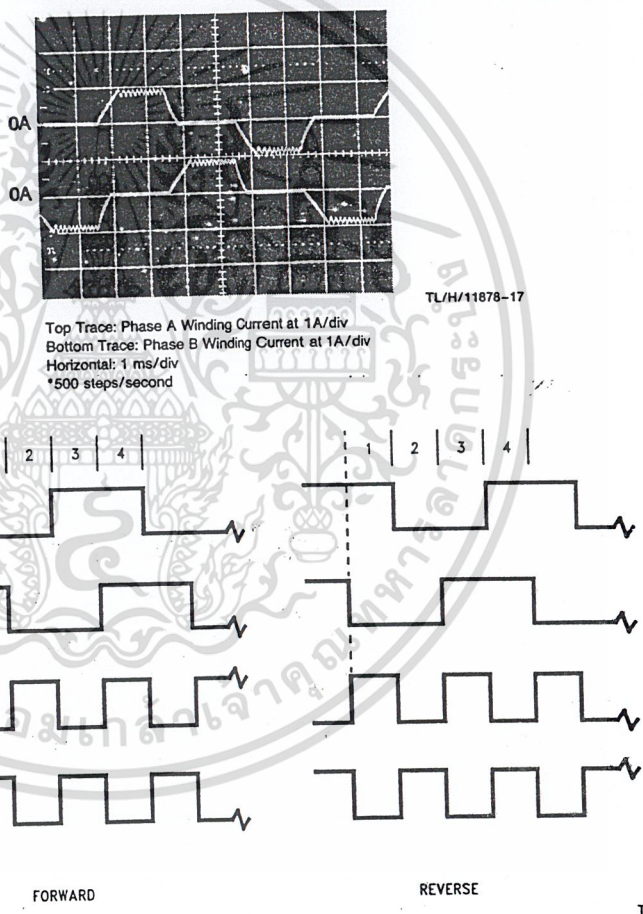
where A represents winding A energized with current in one direction and A* represents winding A energized with current in the opposite direction. The motor takes one full step each time one winding is de-energized and the other is energized. To make the motor step in the opposite direction, the order 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.



TL/H/11878-17

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

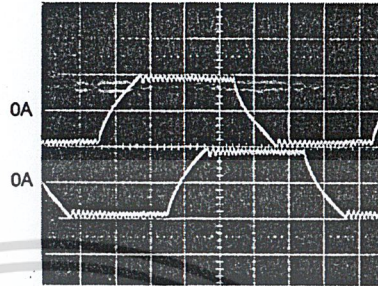
BRAKE A = BRAKE B = 0

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

TL/H/11878-18

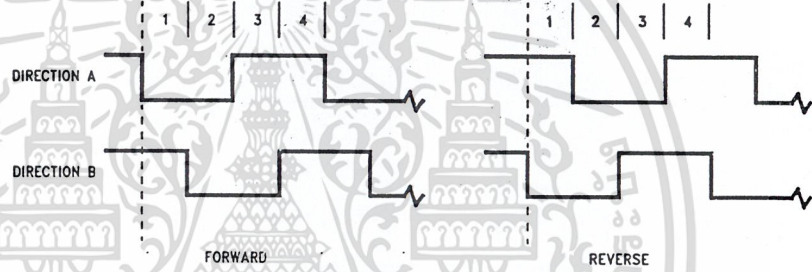
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The Typical Application (Continued)



TL/H/11878-19

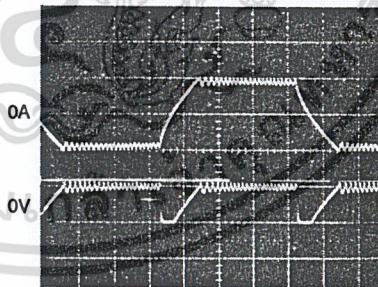
Top Trace: Phase A Winding Current at 1A/div
 Bottom Trace: Phase B Winding Current at 1A/div
 Horizontal: 1 ms/div
 *500 steps/second



TL/H/11878-20

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



TL/H/11878-21

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

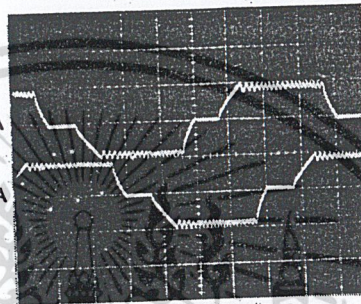
The Typical Application (Continued)

HALF STEP DRIVE WITHOUT TORQUE COMPENSATION

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

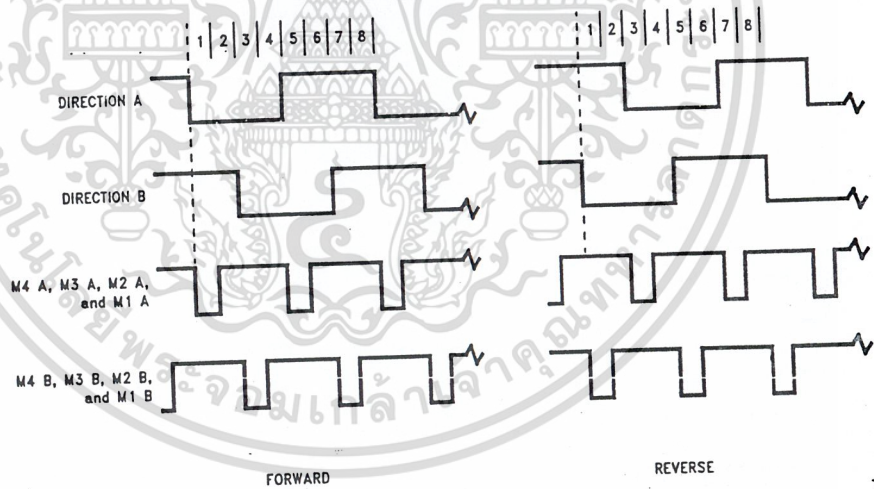
A → AB → B → A*B → A* →
 A*B* → B* → AB* → A → ...

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



TL/H/11878-22

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



TL/H/11878-23

BRAKE A = BRAKE B = 0

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

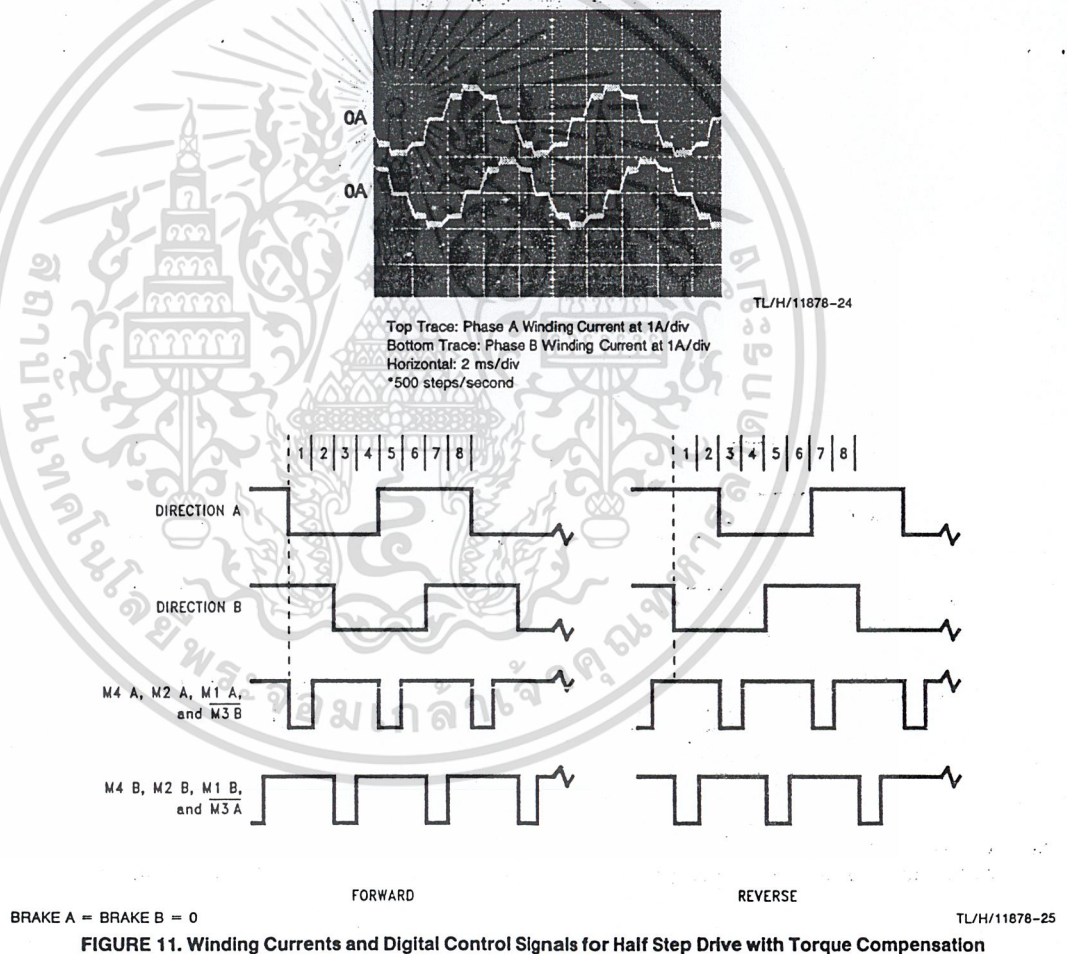
The Typical Application (Continued)

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. α , therefore, increases from 0 to 315° in increments of 45°. Each full 360° cycle comprises eight half steps. Rounding $[\cos\alpha]$ 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.



The Typical Application (Continued)

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

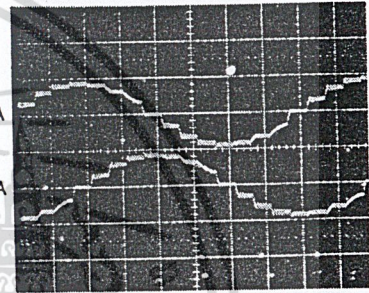
	α	$ \cos(\alpha) $	D A	DIRECTION A	$ \sin(\alpha) $	D B	DIRECTION B
FORWARD ↓	0	1	15	1	0	0	1
	45	0.707	11	1	0.707	11	1
	90	0	0	0	1	15	1
REVERSE ↑	135	0.707	11	0	0.707	11	1
	180	1	15	0	0	0	0
	225	0.707	11	0	0.707	11	0
	270	0	0	1	1	15	0
	315	0.707	11	1	0.707	11	0

REPEAT

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

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.



TL/H/11878-26

Top Trace: Phase A Winding Current at 1A/div
Bottom Trace: Phase B Winding Current at 1A/div
Horizontal: 2ms/div
*250 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
FORWARD ↓	0	1	15	1	0	0	1
	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

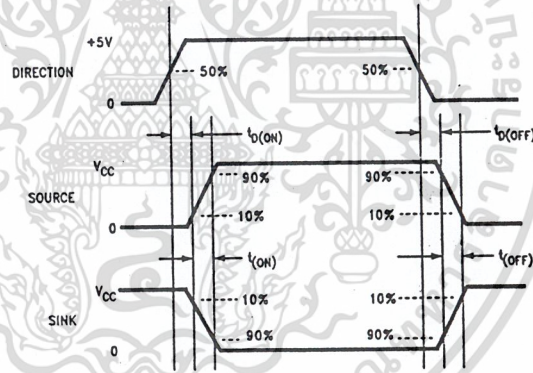
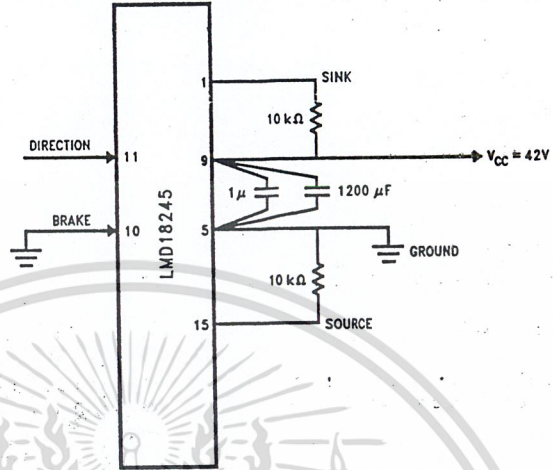
REPEAT

BRAKE A = BRAKE B = 0

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Test Circuit and Switching Time Definitions



TL/H/11878-28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้