



การควบคุมมอเตอร์ไฟฟ้ากระแสตรงแบบไร้แปรงถ่าน

Brushless DC Motor of Control



นายนทวรรณ	กันไชยตะ	รหัส 45380150
นายแสงเพชร	นาคจีน	รหัส 45380188
นายเจษฎากร	ไชยมงคล	รหัส 45380264

15675592

ห้องสมุดคณะวิศวกรรมศาสตร์	
วันที่รับ.....	1, 5 พ.ย. 2549,
เลขทะเบียน.....	4900169
เลขเรียกหนังสือ.....	๗5.
มหาวิทยาลัยนเรศวร	น 1570

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2548

หัวข้อโครงการ	การควบคุมมอเตอร์ไฟฟ้ากระแสตรงแบบไร้แปรงถ่าน (Brushless DC Motor of Control)		
ผู้ดำเนินโครงการ	นายนนทวรรณ	กันไชยดี	รหัส 45380150
	นายแสงเพชร	นาคจีน	รหัส 45380188
	นายเจษฎากร	ไชยมงคล	รหัส 45380264
อาจารย์ที่ปรึกษา	ดร.สมพร	เรืองสินชัชวานิช	
อาจารย์ที่ปรึกษาร่วม	ดร. สมยศ	เกียรติวินชวีไล	
สาขาวิชา	วิศวกรรมไฟฟ้า		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2548		

บทคัดย่อ

โครงการนี้นำเสนอการควบคุมมอเตอร์ไฟฟ้ากระแสตรง โดยได้ใช้กับมอเตอร์ไร้แปรงถ่าน และควบคุมให้มอเตอร์สามารถควบคุมความเร็ว สามารถหมุนซ้าย สามารถหมุนขวา และทำการหยุดหมุนได้ ซึ่งการทำงานจะอาศัยการป้อนค่าของความเร็วรอบให้กับอุปกรณ์สวิตช์กำลัง โดยแสดงผลของความเร็วรอบออกมาทางหน้าจอแอล ซี ดี ของวงจรกำลังอย่างเหมาะสมเพื่อให้ทำงานตามต้องการได้ ข้อดีของมอเตอร์ไร้แปรงถ่านนี้คือไม่ต้องบำรุงรักษาบ่อยเมื่อเทียบกับมอเตอร์ทั่วไป

ในโครงการนี้ได้ทดลองควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรง โดยได้ใช้มอสเฟตเป็นอุปกรณ์สวิตซ์ในการควบคุมความเร็วของมอเตอร์ และการป้อนค่าความเร็วรอบให้อุปกรณ์สวิตช์กำลังโดยแสดงผลของความเร็วรอบออกมาทางหน้าจอแอล ซี ดี ผลที่ได้จากการทดลองทำให้สามารถนำไปพัฒนาต่อไปในระบบการขับเคลื่อนมอเตอร์กระแสตรงที่มีขนาดใหญ่ได้

Project Title Brushless DC Motor of Control
Name Mr. Nonhawath Kanchaita ID 45380150
Mr. Sangpech Nakjeen ID 45380188
Mr. Jetsadagorn Chaimongkol ID 45380264
Project Advisor Dr. Somporn Ruangsinchaiwanich
Co-Project Advisor Dr. Somyot Kiatwanidvilai
Major Electrical Engineering
Department Electrical and Computer Engineering
Academic Year 2005

ABSTRACT

This project presents the controlling of dc motor. By this techniques can control the operation of brushless motor and can speed control, can turn the left motor in motoring, can turn the right motor in motoring and can turn stop motoring of motor. Which the worker will live the values feeder of the speed round the power switch equipment. By indicate values feeder of the speed round to the monitor (LCD) of power circuit appropriately for can work to follow to want. The merit of brushless motor is must not often maintain when it compares with the general motor

The project is experiment to control the speed of dc motor. By have used mosfet to the switch equipment in the power circuit and control the speed of motor. And the value feeder of the speed round give power switch equipment. By indicate values feeder of the speed round to the monitor (LCD). The results get from the experiment doing for can lead to develop in the large size dc motor drive system.

กิตติกรรมประกาศ

ในการทำปริญญาบัตรในครั้งนี้ สำเร็จล่วงไปด้วยดีโดยได้รับความช่วยเหลือและให้คำแนะนำจาก คร. สมพร เรืองสินชัยวานิช ที่เป็นอาจารย์ที่ปรึกษาปริญญาบัตรในครั้งนี้ และขอขอบคุณ คร. สมยศ เกียรติวนิชวิไล และคร. สุรเชษฐ์ กานต์ประชา ที่ให้ความกรุณาเป็นผู้ตรวจสอบปริญญาบัตรและคอยให้คำแนะนำในส่วนของคุณวุฒิที่เกี่ยวข้องกับปริญญาบัตรในครั้งนี้ด้วย

ท้ายสุดขอขอบพระคุณ คุณพ่อ คุณแม่ และญาติพี่น้องที่คอยดูแล คอยเป็นกำลังใจและเป็นผู้สนับสนุนในด้านต่างๆ มาโดยตลอดในการทำปริญญาบัตรนี้ และขอขอบคุณบุคคลต่างๆ ที่ไม่ได้กล่าวถึงรวมถึงแหล่งข้อมูลที่เกี่ยวข้องต่อการทำปริญญาบัตรในครั้งนี้ด้วย

คณะผู้จัดทำ



สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญรูป.....	ฉ
บทที่ 1 บทนำ	
1.1 ที่มาและความสำคัญของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบข่ายของโครงการ.....	2
1.4 ขั้นตอนกิจกรรมการดำเนินงาน.....	2
1.5 ตารางกิจกรรมการดำเนินงาน.....	3
1.6 ผลที่คาดว่าจะได้รับ.....	4
1.7 งบประมาณที่ใช้.....	4
บทที่ 2 ทฤษฎีและหลักการควบคุมมอเตอร์	
2.1 มอเตอร์ไฟฟ้ากระแสตรง.....	5
2.2 มอสเฟตกำลัง.....	18
2.3 อินเวอร์เตอร์.....	23
บทที่ 3 การออกแบบการทดลอง	
3.1 รูปแบบการทำงานของวงจรควบคุมการทำงานของมอเตอร์.....	26
3.2 การออกแบบการเขียนโปรแกรม.....	26
3.3 การออกแบบวงจรควบคุมและวงจรกำลัง.....	28
3.4 การออกแบบการทดลอง.....	28

สารบัญ (ต่อ)

	หน้า
บทที่ 4 ผลการทดลอง.....	31
บทที่ 5 บทสรุป	
5.1 สรุปผลการทดลอง.....	41
5.2 ประเมินผลการทดลอง.....	43
5.3 ปัญหา ข้อเสนอแนะ และแนวทางแก้ไข.....	43
5.4 แนวทางในการพัฒนาต่อไป.....	44
เอกสารอ้างอิง.....	45
ภาคผนวก.....	46
ภาคผนวก ก.....	47
ภาคผนวก ข.....	54
ภาคผนวก ค.....	72
ประวัติผู้เขียนโครงการ.....	102

สารบัญรูป

รูปที่	หน้า
2.1	คลื่นแรงดันที่ได้จากการปิด-เปิดไฟฟ้ากระแสตรง.....10
2.2 (ก)	วงจรขอปเปอ์ลดค่าลงโดยใช้มอสเฟตกำลัง.....12
(ข)	รูปคลื่นแรงดันและกระแสของวงจรขอปเปอ์ลดค่าลง.....12
2.3 (ก)	วงจรขอปเปอ์ยกค่าขึ้นโดยใช้ทรานซิสเตอร์กำลังแบบไบ โพลาร์.....13
(ข)	รูปคลื่นแรงดันและกระแสของวงจรขอปเปอ์ยกค่าขึ้น.....13
2.4	การขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงแบบไร้แปรงถ่าน.....16
2.5	ภาพตัดขวางของมอเตอร์ไร้แปรงถ่าน.....16
2.6	คลื่นสัญญาณควบคุม.....18
2.7 (ก)	มอสเฟตชนิดดีพีทีซีเอ็น แชนแนลเอ็น.....19
(ข)	สัญลักษณ์ของมอสเฟตชนิดดีพีทีซีเอ็น แชนแนลเอ็น.....19
2.8 (ก)	มอสเฟตชนิดดีพีทีซีเอ็น แชนแนลพี.....19
(ข)	สัญลักษณ์ของมอสเฟตชนิดดีพีทีซีเอ็น แชนแนลพี.....19
2.9 (ก)	มอสเฟตชนิดเอนเทนซ์เมนต์ แชนแนลเอ็น.....20
(ข)	สัญลักษณ์ของมอสเฟตชนิดเอนเทนซ์เมนต์ แชนแนลเอ็น.....20
2.10 (ก)	มอสเฟตชนิดเอนเทนซ์เมนต์ แชนแนลพี.....20
(ข)	สัญลักษณ์ของมอสเฟตชนิดเอนเทนซ์เมนต์ แชนแนลพี.....20
2.11 (ก)	คุณลักษณะกราฟการทำงานของดีมอสเฟต แชนแนลเอ็นและแชนแนลพี.....21
(ข)	คุณลักษณะกราฟการทำงานของอิมอสเฟต แชนแนลเอ็นและแชนแนลพี.....21
(ค)	คุณลักษณะเอาต์พุตของอิมอสเฟตกำลังแบบแชนแนลเอ็น.....21
2.12	พื้นที่ทำงานปลอดภัยของมอสเฟตกำลัง.....22
2.13	วงจรอินเวอร์เตอร์ 3 เฟส.....27
2.14	คลื่นสัญญาณควบคุมอินเวอร์เตอร์แบบนำกระแส 120° และรูปคลื่นแรงดันเฟส.....25
3.1	บล็อกไดอะแกรมของวงจรควบคุมมอเตอร์.....26
3.2	โฟลว์ชาร์ตแสดงการทำงานของโปรแกรมที่ใช้ควบคุมมอเตอร์.....27
3.3	วงจรขั้วมอเตอร์ลงอุปกรณ์เรียบร้อยแล้ว.....28
3.4	การต่อมอเตอร์และแหล่งจ่ายไปเข้ากับวงจรควบคุมมอเตอร์.....29
3.5	วงจรที่ต่อเสร็จแล้วพร้อมทำการทดลอง.....30

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.1 กราฟสัญญาณสร้างอินพุต ที่ความเร็วรอบ 5,000 rpm.....	31
4.2 กราฟสัญญาณสร้างอินพุต ที่ความเร็วรอบ 8,000 rpm.....	31
4.3 กราฟสัญญาณสร้างอินพุต ที่ความเร็วรอบ 12,000 rpm.....	31
4.4 กราฟสัญญาณสร้างอินพุต ที่ความเร็วรอบ 15,000 rpm.....	31
4.5 กราฟสัญญาณที่ได้จากการ Trigger ของมอเตอร์ตัวที่ 1 และ 2 ที่ความเร็วรอบ 5,000 rpm	32
4.6 กราฟสัญญาณที่ได้จากการ Trigger ของมอเตอร์ตัวที่ 3 และ 4 ที่ความเร็วรอบ 5,000 rpm	32
4.7 กราฟสัญญาณที่ได้จากการ Trigger ของมอเตอร์ตัวที่ 5 และ 6 ที่ความเร็วรอบ 5,000 rpm	32
4.8 กราฟสัญญาณที่ได้จากการ Trigger ของมอเตอร์ตัวที่ 1 และ 2 ที่ความเร็วรอบ 8,000 rpm	33
4.9 กราฟสัญญาณที่ได้จากการ Trigger ของมอเตอร์ตัวที่ 3 และ 4 ที่ความเร็วรอบ 8,000 rpm	33
4.10 กราฟสัญญาณที่ได้จากการ Trigger ของมอเตอร์ตัวที่ 5 และ 6 ที่ความเร็วรอบ 8,000 rpm	33
4.11 กราฟสัญญาณที่ได้จากการ Trigger ของมอเตอร์ตัวที่ 1 และ 2 ที่ความเร็วรอบ 12,000 rpm.....	34
4.12 กราฟสัญญาณที่ได้จากการ Trigger ของมอเตอร์ตัวที่ 3 และ 4 ที่ความเร็วรอบ 12,000 rpm.	34
4.13 กราฟสัญญาณที่ได้จากการ Trigger ของมอเตอร์ตัวที่ 5 และ 6 ที่ความเร็วรอบ 12,000 rpm	34
4.14 กราฟสัญญาณที่ได้จากการ Trigger ของมอเตอร์ตัวที่ 1 และ 2 ที่ความเร็วรอบ 15,000 rpm.....	35
4.15 กราฟสัญญาณที่ได้จากการ Trigger ของมอเตอร์ตัวที่ 3 และ 4 ที่ความเร็วรอบ 15,000 rpm	35

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.16 กราฟสัญญาณที่ได้จากการ Trigger ของมอเตอร์ตัวที่ 5 และ 6 ที่ความเร็วรอบ 15,000 rpm	35
4.17 กราฟสัญญาณที่ได้จากเฟส AB ที่ความเร็วรอบ 5,000 rpm.....	36
4.18 กราฟสัญญาณที่ได้จากเฟส BC ที่ความเร็วรอบ 5,000 rpm.....	36
4.19 กราฟสัญญาณที่ได้จากเฟส CA ที่ความเร็วรอบ 5,000 rpm.....	36
4.20 กราฟสัญญาณที่ได้จากเฟส AB ที่ความเร็วรอบ 8,000 rpm.....	37
4.21 กราฟสัญญาณที่ได้จากเฟส BC ที่ความเร็วรอบ 8,000 rpm.....	37
4.22 กราฟสัญญาณที่ได้จากเฟส CA ที่ความเร็วรอบ 8,000 rpm.....	37
4.23 กราฟสัญญาณที่ได้จากเฟส AB ที่ความเร็วรอบ 12,000 rpm.....	38
4.24 กราฟสัญญาณที่ได้จากเฟส BC ที่ความเร็วรอบ 12,000 rpm.....	38
4.25 กราฟสัญญาณที่ได้จากเฟส CA ที่ความเร็วรอบ 12,000 rpm.....	38
4.26 กราฟสัญญาณที่ได้จากเฟส AB ที่ความเร็วรอบ 15,000 rpm.....	39
4.27 กราฟสัญญาณที่ได้จากเฟส BC ที่ความเร็วรอบ 15,000 rpm.....	39
4.28 กราฟสัญญาณที่ได้จากเฟส CA ที่ความเร็วรอบ 15,000 rpm.....	39
4.29 ความเร็วรอบกับค่ากระแสที่มอเตอร์หมุน	40
5.1 กราฟสัญญาณ Gate ตัวที่ 1 กับ Gate ตัวที่ 4 ที่ความเร็วรอบ 5,000 rpm.....	41
5.2 กราฟสัญญาณ Gate ตัวที่ 1 กับ Gate ตัวที่ 4 ที่ความเร็วรอบ 8,000 rpm.....	41
5.3 กราฟสัญญาณ Gate ตัวที่ 1 กับ Gate ตัวที่ 4 ที่ความเร็วรอบ 12,000 rpm.....	41
5.4 กราฟสัญญาณ Gate ตัวที่ 1 กับ Gate ตัวที่ 4 ที่ความเร็วรอบ 15,000 rpm.....	41
5.5 กราฟสัญญาณที่ได้จากเฟส A ที่ความเร็วรอบ 5,000 rpm.....	42
5.6 กราฟสัญญาณที่ได้จากเฟส A ที่ความเร็วรอบ 8,000 rpm.....	42
5.7 กราฟสัญญาณที่ได้จากเฟส A ที่ความเร็วรอบ 12,000 rpm.....	42
5.8 กราฟสัญญาณที่ได้จากเฟส A ที่ความเร็วรอบ 15,000 rpm.....	42

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ในปัจจุบันการคอนโทรลมอเตอร์ไฟฟ้ากระแสตรงจะพบได้ในงานอุตสาหกรรมส่วนมาก เนื่องจากความก้าวหน้าของเทคโนโลยีทางไมโครอิเล็กทรอนิกส์และไมโครโปรเซสเซอร์ ดังนั้นการออกแบบระบบคอนโทรลมอเตอร์ไฟฟ้ากระแสตรง และการวิเคราะห์จึงเป็นเทคนิคใหม่ที่มีความสำคัญและน่าสนใจยิ่ง

ระบบคอนโทรลแบบดั้งเดิมพลังจักรกลที่สำคัญในระบบมักได้แก่ พวкмอเตอร์ไฟฟ้าต่างๆ เบรกและคลัทช์ เป็นต้น แต่ด้วยความเจริญก้าวหน้าทางเทคโนโลยีการสร้างแม่เหล็กถาวรให้มอเตอร์ไฟฟ้ากระแสตรงมีคุณภาพสูง และได้กลายเป็นพลังจักรกลที่สำคัญในระบบคอนโทรลเกือบทุกชนิด และในเวลาเดียวกันบวกกับความก้าวหน้าของไมโครโปรเซสเซอร์ได้สร้างการประยุกต์งานใหม่ๆ ขึ้นมา ทำให้มอเตอร์ไฟฟ้ากระแสตรงได้รับความนิยมในการใช้งานอุตสาหกรรมในปัจจุบันอย่างกว้างขวาง

โดยปกติแล้วความเร็วของมอเตอร์ไฟฟ้ามีขีดจำกัดไว้ตายตัวอยู่แล้ว ขึ้นอยู่กับการออกแบบสร้างของมอเตอร์และได้แนวคิดที่ทำโครงการนี้ขึ้นมาเพื่อที่จะควบคุมความเร็วของมอเตอร์, ควบคุมการหมุนของมอเตอร์ และสั่งให้มอเตอร์หยุดหมุนได้โดยใช้มอสเฟตเป็นสวิทช์ในการควบคุมมอเตอร์และมอเตอร์ไฟฟ้ากระแสตรงที่ใช้จะเป็นมอเตอร์ไฟฟ้ากระแสตรงชนิดพิเศษคือ แบบไม่มีแปรงถ่าน (Brushless dc Motor) ซึ่งหลักการทางเทคโนโลยีเหมือนกับมอเตอร์ไฟฟ้ากระแสตรงชนิดมีแปรงถ่านยกเว้น การคอมมิวเตชัน กระทำโดยเทคนิคทางอิเล็กทรอนิกส์แทนที่จะกระทำโดยวิธีการทางเชิงกล

1.2 วัตถุประสงค์

1. เพื่อศึกษาการทำงานของการควบคุมมอเตอร์ไฟฟ้ากระแสตรงได้
2. เพื่อควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรงได้
3. เพื่อศึกษาโปรแกรมและนำโปรแกรมไปประยุกต์ใช้งานกับมอเตอร์ไฟฟ้ากระแสตรงได้

1.3 ขอบข่ายของงาน

สามารถควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรง และสามารถสั่งให้มอเตอร์หมุนซ้าย หมุนขวาและทำการหยุดหมุนได้ โดยใช้มอสเฟตเป็นอุปกรณ์สวิตซ์ในการควบคุมมอเตอร์ และป้องกันค่าของความเร็วรอบให้กับอุปกรณ์สวิตซ์ โดยให้แสดงผลทางหน้าจอ แอล ซี ดี และใช้มอเตอร์ไร้แปรงถ่านเพื่อที่จะไม่ต้องบำรุงรักษามอเตอร์บ่อยๆ

1.4 ขั้นตอนกิจกรรมการดำเนินงาน

1. ศึกษาข้อมูลเกี่ยวกับอุปกรณ์และการทำงานของอุปกรณ์ที่ต้องใช้ในการควบคุมมอเตอร์
2. ศึกษาและค้นคว้าเกี่ยวกับโปรแกรมที่ใช้สำหรับการควบคุมการทำงานของมอเตอร์
3. เขียน โปรแกรมควบคุมการทำงานของมอเตอร์และทดสอบโปรแกรม
4. นำส่วนต่างๆ มาประกอบและทดสอบการทำงาน
5. บันทึกผลการทดลองและวิเคราะห์ผลการทดลอง
6. สรุปผลการทดลองและจัดทำรูปเล่ม โครงงาน



1.6 ผลที่คาดว่าจะได้รับ

1. สามารถเข้าใจหลักการทำงานของมอเตอร์กระแสตรงได้
2. สามารถควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรงได้
3. สามารถนำไปประยุกต์ใช้ในส่วนที่เกี่ยวกับการควบคุมได้
4. สามารถใช้ไมโครคอนโทรลเลอร์ในการควบคุมมอเตอร์ไฟฟ้ากระแสตรงได้
5. สามารถนำโครงการนี้ไปประยุกต์ใช้กับงานที่ใช้มอเตอร์ไฟฟ้ากระแสตรงในชีวิตประจำวันได้

1.7 งบประมาณ

1. ค่าอุปกรณ์อิเล็กทรอนิกส์
 2. ค่าถ่ายเอกสาร
 3. ค่าพิมพ์เอกสาร
 4. กระดาษ
 5. หนังสือประกอบการทำโครงการ
- รวมเป็นเงิน 3,000 บาท (สามพันบาทถ้วน)



บทที่ 2

ทฤษฎีและหลักการควบคุมมอเตอร์

2.1 มอเตอร์ไฟฟ้ากระแสตรง

2.1.1 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรง เป็นทรานส์ดิวเซอร์แรงบิดซึ่งมีการออกแบบให้มีคุณลักษณะพิเศษ คือ แรงบิดของเพลลาของมอเตอร์ไฟฟ้ากระแสตรงจะเป็นสัดส่วนโดยตรงกับกระแสอาร์-เมเจอร์ แรงบิดของเพลลาของมอเตอร์ไฟฟ้ากระแสตรงจะได้จากผลต่างสนามแม่เหล็กและขดลวดตัวนำ ในที่นี้กระแสไหลในขดลวดตัวนำจะสร้างฟิลด์ที่ประกอบด้วยเส้นแรงแม่เหล็ก ϕ และขดลวดตัวนำเหล่านั้นอยู่ห่างจากศูนย์กลางการหมุน เราจะได้ความสัมพันธ์ระหว่างแรงบิดของเพลลาและกระแสเท่ากับ

$$T = k\phi I \quad (2.1)$$

เมื่อ	T	คือแรงบิดของเพลลา (นิวตัน-เมตร)
	ϕ	คือเส้นแรงแม่เหล็ก (เวเบอร์)
	I	คือกระแส (แอมแปร์)
	k	คือค่าคงที่

ดังนั้นแรงบิดของเพลลาจะเป็นสัดส่วนโดยตรงกับผลคูณของเส้นแรงแม่เหล็กและกระแส เมื่อขดลวดตัวนำเคลื่อนที่ในสนามแม่เหล็กก็จะทำให้เกิดโวลต์เตจตกคร่อมตัวมันเอง โวลต์เตจนี้จะเป็นสัดส่วนกับความเร็วของเพลลาของมอเตอร์และด้านการไหลของกระแส ความสัมพันธ์ระหว่างโวลต์เตจย้อนกลับนี้และความเร็วของเพลลามอเตอร์คือ

$$E = k\phi\omega \quad (2.2)$$

เมื่อ	E	คือโวลต์เตจย้อนกลับ emf (โวลต์)
	ϕ	คือเส้นแรงแม่เหล็ก (เวเบอร์)
	ω	คือความเร็วของมอเตอร์ (เรเดียน/วินาที)

2.1.2 การแยกประเภทของมอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรงสามารถแบ่งออกได้เป็นหลายประเภท ขึ้นอยู่กับลักษณะวิธีการสร้างสนามแม่เหล็กของตัวมอเตอร์และยังขึ้นอยู่กับพื้นฐานการออกแบบโครงสร้างของอาร์เมเจอร์ การแบ่งประเภทตามลักษณะการจ่ายสนามแม่เหล็กแยกออกได้เป็น 2 แบบ คือ

1. มอเตอร์ไฟฟ้ากระแสตรงแบบปรับเส้นแรงแม่เหล็กได้
2. มอเตอร์ไฟฟ้ากระแสตรงแบบเส้นแรงแม่เหล็กมีค่าคงที่

ถ้าเราพิจารณาแยกประเภทตามลักษณะการออกแบบโครงสร้างของอาร์เมเจอร์สามารถแยกออกได้เป็น 3 แบบคือ

1. มอเตอร์ไฟฟ้ากระแสตรงแบบอาร์เมเจอร์เป็นแกนเหล็ก
2. มอเตอร์ไฟฟ้ากระแสตรงแบบอาร์เมเจอร์ที่มีขดลวดพันอยู่บนพื้นผิว
3. มอเตอร์ไฟฟ้ากระแสตรงแบบอาร์เมเจอร์เป็นขดลวดหมุน

นอกจากนี้ยังมีมอเตอร์ไฟฟ้ากระแสตรงชนิดพิเศษอีกแบบหนึ่งคือ **แบบไม่มีแปรงถ่าน (Brushless dc Motor)** ซึ่งมีหลักการทางเทคโนโลยีเหมือนกับมอเตอร์ไฟฟ้ากระแสตรงชนิดมีแปรงถ่านยกเว้นการคอมมิวเตชัน กระทำโดยเทคนิคทางอิเล็กทรอนิกส์แทนที่จะทำโดยวิธีการทางเชิงกล

2.1.3 ข้อดีและข้อเสียของมอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรงมีข้อดี คือ

1. มีคุณสมบัติเหมาะสมในการควบคุมอัตราเร็วของการขับเคลื่อน โดยมีพิสัยของอัตราเร็ว กว้างมาก ซึ่งเราสามารถเพิ่มอัตราเร็วให้สูงหรือให้ต่ำกว่าอัตราเร็วปกติได้
2. มีแรงหมุนขณะสตาร์ทสูงมาก ซึ่งเหมาะกับงานบางประเภท เช่น ยกของ คุุดลาก และขับเคลื่อน
3. วิธีการควบคุมของมอเตอร์ไฟฟ้ากระแสตรงง่ายกว่า เรียบร้อยกว่ามอเตอร์ไฟฟ้ากระแสสลับที่ทำงานคล้ายกัน

อย่างไรก็ตามมอเตอร์ไฟฟ้ากระแสตรงยังมีข้อเสียหลายประการ คือ

1. ต้องหาแหล่งจ่ายไฟฟ้ากระแสตรงไว้ใช้งานเป็นพิเศษ เนื่องจากไฟฟ้าทั่วไปเป็นไฟฟ้ากระแสสลับ
2. สำหรับขนาดพิกัดกำลังของมอเตอร์เท่ากัน มอเตอร์ไฟฟ้ากระแสตรงจะมีขนาดใหญ่กว่าและมีราคาสูงกว่ามอเตอร์ไฟฟ้ากระแสสลับ

3. ในการสตาร์ทมอเตอร์กระแสตรงขนาดใหญ่ต้องอาศัยวิธีพิเศษและยุ่งยากกว่ามอเตอร์ไฟฟ้ากระแสสลับ ยกเว้นสำหรับมอเตอร์ขนาดเล็กเท่านั้นที่ไม่ต้องการวิธีสตาร์ทพิเศษ

4. ต้องการการซ่อมบำรุงรักษามากกว่ามอเตอร์ไฟฟ้ากระแสสลับ เนื่องจากคอมมิวเตเตอร์ (Commutator) มีการสึกหรอที่เกิดจากอาร์ค ประกายไฟ และการขัดสีระหว่างแปรงถ่านกับคอมมิวเตเตอร์

จากข้อดีและข้อเสียที่กล่าวมาข้างต้นจะเห็นว่าแม้มอเตอร์ไฟฟ้ากระแสตรงจะมีข้อเสียมาก แต่ข้อดีของมอเตอร์ไฟฟ้ากระแสตรงยังเป็นข้อที่มีความสำคัญมาก ดังนั้นจึงยังมีการใช้งานมอเตอร์ไฟฟ้ากระแสตรงในปัจจุบันนี้

2.1.4 การควบคุมอัตราเร็วและแรงหมุนของมอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรงมีหลายชนิด อาทิเช่น มอเตอร์แบบแม่เหล็กถาวร (PM Motor หรือ Permanent Magnet Motor) ซึ่งในปัจจุบันนี้มีการประดิษฐ์แม่เหล็กที่มีความเข้มของสนามแม่เหล็กไฟฟ้าสูง และแม่เหล็กเซรามิก (Ceramic Magnet) เป็นต้น ซึ่งนิยมนำมาประดิษฐ์มอเตอร์ขนาดเล็กที่มีแรงม้าเป็นเศษส่วน (Fractional Horse Power) โดยนิยมนำไปใช้ในรถยนต์ เครื่องยกกระชกหน้าต่าง เครื่องยกหรือเลื่อนที่นั่ง เป็นต้น

มอเตอร์แบบแม่เหล็กถาวรมีลักษณะสมบัติคล้ายกับของมอเตอร์ไฟฟ้ากระแสตรง แบบขนาน โดยมีอัตราเร็วเกือบคงที่ แต่มีค่าลดลงบ้างเล็กน้อยตามค่าของภาระหรือแรงหมุนของภาระ ดังนั้นโค้งลักษณะสมบัติระหว่างอัตราเร็วกับแรงหมุนโหลด ซึ่งมอเตอร์ไฟฟ้ากระแสสลับแบบขนานมีสูตรการคำนวณหาอัตราเร็วดังนี้

$$\omega = \frac{E - I_a R_a}{K_\omega I_f} \quad (2.3)$$

เมื่อ	ω	เป็นค่าอัตราเร็ว (รอบต่อนาที)
	E	เป็นค่าแรงดันที่ขั้วไฟฟ้าของมอเตอร์ (โวลต์)
	I_a	เป็นค่ากระแสไหลในวงจรอาร์เมเจอร์ (แอมแปร์)
	R_a	เป็นค่าความต้านทานในวงจรอาร์เมเจอร์ (โอห์ม)
	K_ω	เป็นตัวคงที่สัดส่วนของอัตราเร็ว (โวลต์-นาทิต่อแอมแปร์-รอบ)
	I_f	เป็นค่ากระแสไหลในวงจรฟิลด์ (แอมแปร์)

และแรงหมุนที่เกิดขึ้นได้จาก

$$T = K_t I_a \quad (2.4)$$

เมื่อ T เป็นแรงหมุนที่เกิดขึ้น (นิวตัน-เมตร)
 K_t เป็นตัวคงที่สัดส่วนของแรงหมุน (นิวตัน-เมตรต่อแอมแปร์)

มอเตอร์อีกชนิดหนึ่งคือ มอเตอร์แบบผสม (Compound) จะมีอัตราเร็วสูงขึ้นหรือช้าลงเมื่อเปรียบเทียบกับมอเตอร์แบบขนาน ขึ้นอยู่กับการต่อฟิลด์อนุกรม (Series Field) ให้สนามแม่เหล็กเข้าหักล้างกันหรือเสริมกับสนามของฟิลด์ขนาน (Shunt Field) ซึ่งเรียกว่ามอเตอร์ผสมแบบหักล้าง (Differential Compound Motor) และมอเตอร์ผสมแบบสะสม (Cumulative Compound Motor) ตามลำดับ แต่อย่างไรก็ตามมอเตอร์ผสมแบบหักล้างไม่ค่อยนิยมใช้กัน

มอเตอร์แบบอนุกรมเป็นมอเตอร์ที่มีความสำคัญมาก เนื่องจากสามารถปรับเปลี่ยนอัตราเร็วเป็นพิสัยกว้าง และมีแรงหมุนเกิดขึ้นตอนสตาร์ทที่สูงมาก โดยเราสามารถคำนวณหาค่าอัตราเร็ว

$$\omega = \frac{E - I_a R_a}{K_\omega I_a^2} \quad (2.5)$$

และแรงบิดที่ได้

$$T = K_t I_a^2 \quad (2.6)$$

จากสมการที่ 2.6 จะเห็นได้ว่าคุณค่าแรงบิดมีค่าสูง เนื่องจากขึ้นกับกำลังสองของกระแสไฟฟ้า

การควบคุมอัตราเร็วของมอเตอร์ไฟฟ้ากระแสตรงแทบทุกแบบจะทำได้คล้ายกัน ซึ่งได้จากสมการที่ 2.3 และสมการที่ 2.5 ที่ได้กล่าวมาข้างต้น

จะเห็นได้ว่า วิธีการควบคุมอัตราเร็วมี 4 วิธี คือ

1. ควบคุมแรงดันป้อนเข้าที่ขั้วของมอเตอร์
2. ควบคุมจำนวนกระแสที่ไหลเข้าอาร์เมเจอร์

3. ควบคุมจำนวนกระแสไหลเข้าฟิลต์

4. ควบคุมแบบผสม

วิธีการควบคุมอัตราเร็วของมอเตอร์ที่ง่าย ๆ และมีราคาถูกทั้งยังสามารถใช้ในการสตาร์ทมอเตอร์ได้ด้วยคือ การใช้ตัวต้านทานเข้าต่อในวงจรอาร์เมเจอร์และวงจรฟิลต์ การปรับเปลี่ยนค่าความต้านทานจะเป็นการปรับเปลี่ยนค่าแรงดันและค่ากระแสที่ป้อนเข้าวงจรอาร์เมเจอร์และวงจรฟิลต์ ทำให้เราสามารถควบคุมความเร็วให้ได้ตามต้องการแล้วยังสามารถควบคุมค่ากระแสตอนสตาร์ทให้อยู่ในค่าที่ปลอดภัยและมีแรงหมุนตอนสตาร์ทสูงได้ เมื่อขณะมอเตอร์กำลังหมุนอยู่เราก็ยังสามารถใช้ตัวต้านทานเหล่านั้นในการควบคุมอัตราเร็วด้วย แต่วิธีนี้มีข้อเสียเปรียบคือ มีกำลังสูญเสียในตัวต้านทานเหล่านั้น ทำให้มีประสิทธิภาพต่ำ นอกจากนี้ขณะมอเตอร์กำลังหมุนที่คอมมิวเตเตอร์จึงมีอาร์ค ทำให้เป็นหลุมเป็นบ่อที่หน้าสัมผัสและมีการสึกหรอเกิดขึ้นต้องการการซ่อมบำรุงรักษาบ่อยๆ

เราจะจัดข้อเสียเปรียบที่กล่าวมาโดยนำเอาเพาเวอร์อิเล็กทรอนิกส์เข้ามาควบคุมแทน ซึ่งอาจแบ่งออกเป็น 2 วิธีใหญ่ๆ ตามแบบของแหล่งจ่ายไฟฟ้า คือ

1. แหล่งจ่ายไฟฟ้ากระแสตรง ในวงจรชอปเปอร์ (Chopper) เปลี่ยนไฟฟ้ากระแสตรงจากแหล่งจ่ายให้เป็นไฟฟ้ากระแสตรง ที่ปรับเปลี่ยนแรงดันไฟฟ้ากระแสตรงเอาต์พุตได้
2. แหล่งจ่ายไฟฟ้ากระแสสลับ ใช้วงจรคอนเวอร์เตอร์เข้าเรียงกระแสเปลี่ยนให้เป็นไฟฟ้ากระแสตรง

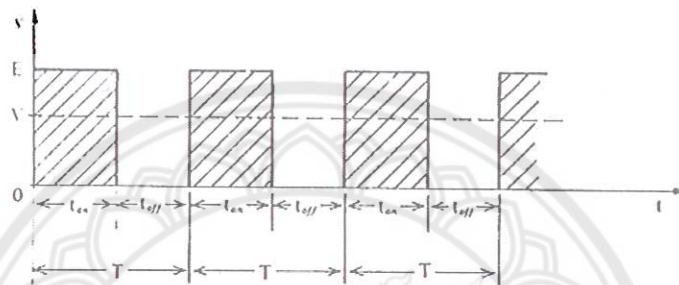
2.1.5 วงจรควบคุมแรงดันไฟฟ้ากระแสตรง

วงจรควบคุมแรงดันไฟฟ้ากระแสตรงสามารถนำไปใช้ควบคุมแรงดันไฟฟ้ากระแสตรงได้และได้จากการปิดและเปิดไฟฟ้ากระแสตรงเป็นจังหวะ แรงเคลื่อนไฟฟ้ากระแสตรงที่ส่งไปยังภาระจะขาดออกเป็นช่วงๆ ดังแสดงในรูปที่ 2.1 ซึ่งเราสามารถควบคุมค่าเฉลี่ยของแรงดันเฉลี่ยวงจรให้สูงหรือต่ำลงด้วยการควบคุมเวลาเปิด (t_{on}) , เวลาปิด (t_{off}) หรือค่าของเวลาปิดเปิด ($T = t_{on} + t_{off}$) ให้แปรค่าต่างๆ เป็นจังหวะซึ่งสามารถเขียนเป็นสมการได้ดังนี้

$$v(t) = \begin{cases} E & \text{เมื่อ } t = t_{on} \\ 0 & \text{เมื่อ } t = t_{off} \end{cases} \quad (2.7)$$

เมื่อกระจายออกเป็นอนุกรม Fourier เราจะได้ค่าสัมประสิทธิ์

$$\begin{aligned}
 a_0 &= \frac{1}{T/2} \int_0^T E dt = 2 \frac{t_{on}}{T} E \\
 a_n &= \frac{1}{T/2} \int_0^T E \cos\left(\frac{n\pi t}{T/2}\right) dt = \frac{2E}{T} \left[\int_0^{t_{on}} \cos \frac{2n\pi t}{T} dt + \int_{t_{on}}^T (0) \cdot \cos \frac{2n\pi t}{T} dt \right] \\
 &= \frac{E}{n\pi} \sin \frac{2n\pi t_{on}}{T}, n = 1, 2, 3, \dots
 \end{aligned} \tag{2.8}$$



รูปที่ 2.1 รูปคลื่นแรงดันที่ได้จากการปิดเปิดไฟฟ้ากระแสตรง
[ที่มา ความคุมเครื่องกลไฟฟ้าด้วยอิเล็กทรอนิกส์กำลัง 1 : หน้า 249]

และ

$$\begin{aligned}
 b_n &= \frac{1}{T/2} \int_0^T E \sin\left(\frac{n\pi t}{T/2}\right) dt = \frac{2E}{T} \left[\int_0^{t_{on}} \sin \frac{2n\pi t}{T} dt + \int_{t_{on}}^T (0) \sin \frac{2n\pi t}{T} dt \right] \\
 &= \frac{E}{n\pi} \left(1 - \cos \frac{2n\pi t_{on}}{T}\right), n = 1, 2, 3, \dots
 \end{aligned} \tag{2.9}$$

ทำให้เราได้อนุกรม Fourier ดังนี้

$$v(t) = E \left[\frac{t_{on}}{T} + \frac{1}{\pi} \sum_{n=1}^{\infty} \frac{1}{n} \left\{ \left(\sin \frac{2n\pi t_{on}}{T} \right) \left(\cos \frac{2n\pi t}{T} \right) + \left(1 - \cos \frac{2n\pi t_{on}}{T} \right) \left(\sin \frac{2n\pi t}{T} \right) \right\} \right] \tag{2.10}$$

หมายเหตุ : สมการที่ 2.7 – 2.11 [ที่มา ความคุมเครื่องกลไฟฟ้าด้วยอิเล็กทรอนิกส์กำลัง 1 :
หน้า 250]

เราสามารถคำนวณหาค่าแรงดันเฉลี่ยได้ดังนี้

$$V_{AVG} = \frac{t_{on}}{T} E = \frac{t_{on}}{t_{on} + t_{off}} E \quad (2.11)$$

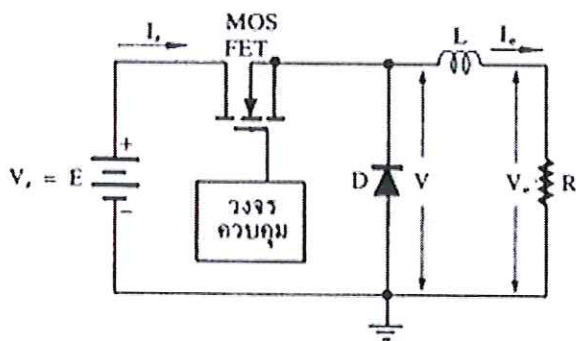
โดยไฟฟ้าแต่ละช่วงมีความสูงเท่ากับ ความสูงของแรงดันของแหล่งจ่ายไฟฟ้า E แต่แรงดันเอาต์พุต V จะเป็นค่าแรงดันเฉลี่ยซึ่งอาจปรับเปลี่ยนค่าได้ 3 วิธี คือ

1. ปรับเปลี่ยนค่า t_{on} แต่รักษาคาบ T ให้คงที่ เรียกว่า “วิธีมอดูเลตตามความกว้างของพัลส์” (Pulse Width Modulation)
2. รักษาค่า t_{on} ให้คงที่ แต่ปรับเปลี่ยนค่าของคาบ T เรียกว่า “วิธีมอดูเลตตามความถี่” (Frequency Modulation)
3. วิธีผสมของมอดูเลตตามความกว้างของพัลส์และตามความถี่

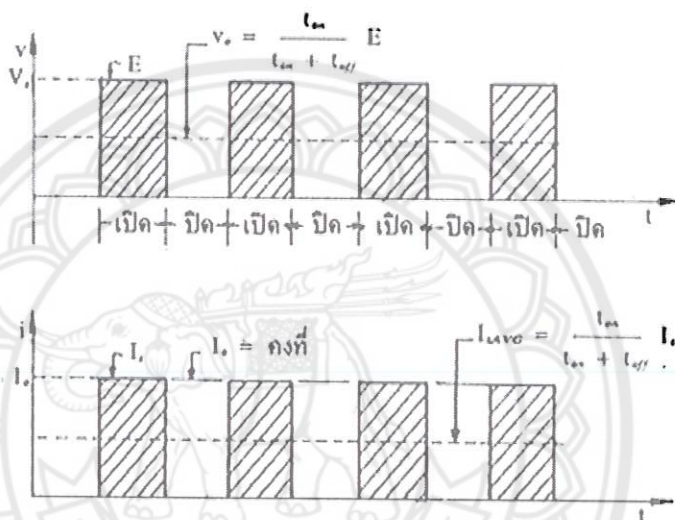
วงจรที่จะสามารถให้แรงดันขาดตอนเป็นช่วงๆ ที่สามารถเปลี่ยนค่าแรงดันเฉลี่ยได้ดังรูปคลื่นที่แสดงในรูปที่ 2.1 เป็นวงจรที่เรียกว่า “วงจรชอปเปอร์” (Chopper) ซึ่งทำขึ้นได้หลายแบบ วงจรแบบที่นิยมใช้ในปัจจุบันเป็นวงจรที่ใช้ทรานซิสเตอร์และทรานซิสเตอร์กำลังแบบไบโพลาร์ เป็นส่วนประกอบสำคัญแต่ในที่นี้จะขอใช้เพียงมอสเฟตกำลังซึ่งสามารถสร้างขึ้นได้เป็นวงจรสองแบบคือ วงจรชอปเปอร์ลดค่าลง (Step-Down Chopper) และวงจรชอปเปอร์ยกค่าขึ้น (Step-Up Chopper)

1. วงจรชอปเปอร์ลดค่าลง (Step Down Chopper)

วงจรชอปเปอร์ลดค่าลง โดยการนำเอาทรานซิสเตอร์กำลังแบบไบโพลาร์หรือมอสเฟตกำลังแต่ในที่นี้จะขอใช้เพียงมอสเฟตกำลังเท่านั้น เข้าต่อในวงจรเป็นอนุกรมและคั่นกลางระหว่างแหล่งจ่ายไฟฟ้ากับภาระ ดังรูปที่ 2.2 (ก) ซึ่งเมื่อมอสเฟตกำลังทำงานปิด-เปิดกระแส จะทำให้ได้แรงดันและกระแสมีรูปคลื่นดังแสดงในรูปที่ 2.2 (ข) การปรับเปลี่ยนแรงดันเฉลี่ยทำได้ด้วยการควบคุมวงจรเบสของมอสเฟตกำลัง ให้ทำงานปิด-เปิดวงจรเป็นจังหวะโดยการแปรเวลาปิด เวลาเปิด หรือทั้งคู่ แรงดันเฉลี่ยที่ได้รับจะมีค่าต่ำกว่าค่าแรงดันเฉลี่ยวงจรเสมอ จึงเรียกวงจรนี้ว่า “วงจรชอปเปอร์ลดค่าลง”



(ก)



(ข)

รูปที่ 2.2 (ก) วงจรขอปเปอร์ลดค่าลงโดยใช้มอสเฟตกำลัง

(ข) รูปคลื่นแรงดันและกระแสของวงจรขอปเปอร์ลดค่าลง

[ที่มา ควบคุมเครื่องกลไฟฟ้าด้วยอิเล็กทรอนิกส์กำลัง 1 : หน้า 251]

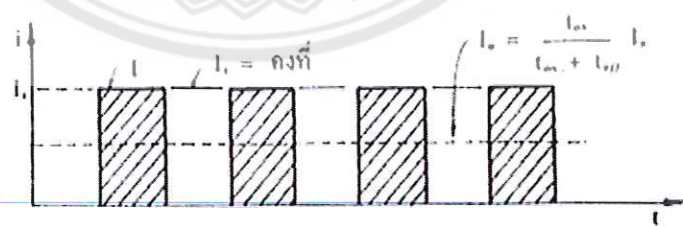
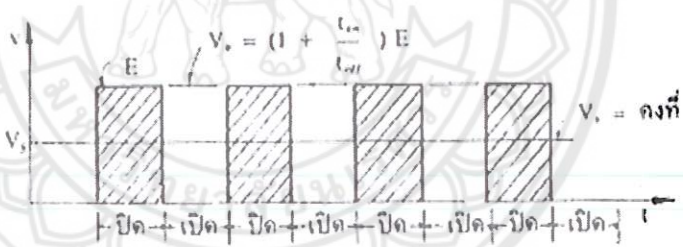
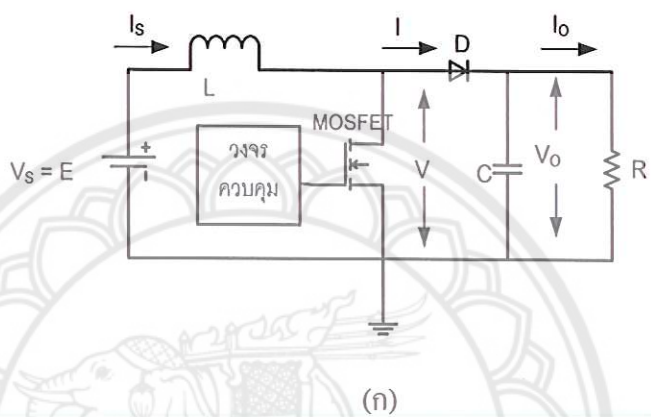
ในรูปที่ 2.2 (ก) ตัวเหนี่ยวนำ L ทำหน้าที่จัดรูปคลื่นแรงเอาต์พุตให้ราบเรียบลง ฟรีวีลลิงไดโอด D ทำหน้าที่เป็นเส้นทางให้กระแสไหลในระหว่างที่ปิดกระแส ถึงแม้ว่าวงจรนี้จะทำให้แรงดันเฉลี่ยลดค่าลง แต่แหล่งจ่ายไฟฟ้าจะจ่ายกระแส I_a ที่มียอดสูงสุดมีค่าเท่ากับกระแสภาระ I_o ดังนั้นกระแสเฉลี่ยจากแหล่งจ่ายไฟฟ้า I_{sAVG} จะมีค่าน้อยกว่ากระแสภาระซึ่งมีความสัมพันธ์ดังนี้

$$I_{sAVG} = \frac{t_{on}}{t_{on} + t_{off}} I_o \tag{2.12}$$

โดยที่กระแสภาระ I_o มีค่าคงที่ เมื่อเราแปรเวลาปิด เวลาเปิด หรือทั้งคู่ จะทำให้เราสามารถปรับค่ากระแสเฉลี่ยให้สูงขึ้นขณะที่แรงดันเฉลี่ยลดต่ำลง

2. วงจรชอปเปอร์ยกค่าขึ้น (Step Up Chopper)

โดยการเปลี่ยนแปลงวงจรในรูปที่ 2.2 ด้วยการนำเอาตัวเหนี่ยวนำ L มาต่อติดกับแหล่งจ่ายไฟฟ้าแบบอนุกรม และนำเอาออสเฟตกำลังมาต่อลงในวงจรแบบขนานดังแสดงในรูปที่ 2.3 (ก) เมื่อกระแสไหลเป็นจังหวะในรูปคลื่นกระแสและแรงดันดังแสดงในรูปที่ 2.3 (ข) ดังนี้



รูปที่ 2.3 (ก) วงจรชอปเปอร์ยกค่าขึ้นโดยใช้มอสเฟตกำลัง
 (ข) รูปคลื่นกระแสและแรงดันของวงจรชอปเปอร์ยกค่าขึ้น
 [ที่มา ควบคุมเครื่องกลไฟฟ้าด้วยอิเล็กทรอนิกส์กำลัง 1 : หน้า 252]

โดยขณะที่เปิดกระแส ตัวเหนี่ยวนำ L จะเก็บพลังงานไว้ในรูปสนามแม่เหล็ก ครั้งเมื่อปิดกระแส ตัวเหนี่ยวนำ L จะปล่อยพลังงานที่เก็บไว้ ออกแล้วทำให้เกิดมีแรงดันขึ้นบวกเพิ่มเข้ากับแรงดันขึ้นบวกเพิ่มเข้ากับแรงดันเฉลี่ยวงจรป้อนเข้าสู่ภาระผ่านไดโอด โดยที่กระแส I_s จากแหล่งจ่ายไฟฟ้ามีค่าคงที่ และพลังงานมีค่าคงที่ ดังนั้นเราสามารถเขียนสมการได้ดังนี้

$$V_0 t_{off} = E(t_{on} + t_{off}) \quad (2.13)$$

ทำให้เราได้ความสัมพันธ์

$$V_0 = \left(1 + \frac{t_{on}}{t_{off}}\right) E \quad (2.14)$$

ซึ่งแสดงว่า เราอาจปรับค่าแรงดันเอาต์พุตเฉลี่ยให้สูงขึ้นได้ แต่กระแสภาระเฉลี่ยมีค่าลดลงตามสมการข้างล่างนี้

$$I_0 = \frac{t_{on}}{t_{on} + t_{off}} I_s \quad (2.15)$$

ถ้าหับ L ทำหน้าที่เฉลี่ยค่าแรงดัน ขณะที่ C ทำหน้าที่เฉลี่ยกระแส ถ้าหากว่า L และ C มีค่าใหญ่ทั้งคู่จะทำให้รูปคลื่นที่ได้รับมีลักษณะราบเรียบอย่างสมบูรณ์ เราอาจลดขนาดของ L และ C ให้เล็กลงได้โดยการเพิ่มความถี่ในการปฏิบัติงานของวงจรขอปเปอร์ให้มีค่าสูงขึ้น แต่วิธีนี้จะทำให้เกิดมีกำลังสูญเสียเพิ่มขึ้น เพราะฉะนั้นเราต้องออกแบบวงจรให้ใช้ความถี่สูงพอประมาณที่ไม่ให้กำลังสูญเสียเกิดขึ้นมากเกินไป

วงจรขอปเปอร์จะทำหน้าที่คล้ายๆ กับเป็นหม้อแปลงไฟฟ้ากระแสตรงที่สามารถลดแรงดันลงหรือยกแรงดันให้สูงขึ้น ด้วยการปรับเปลี่ยนเวลาปิด เวลาเปิด หรือทั้งคู่ ซึ่งเป็นการปรับเปลี่ยนที่ง่ายกว่าการเปลี่ยนจำนวนรอบ (Turn Ratio) ของหม้อแปลงไฟฟ้าต่างๆ ไป

เรานิยมใช้วงจรขอปเปอร์นี้ในการควบคุมอัตราเร็วของมอเตอร์ไฟฟ้ากระแสตรง ที่ใช้เป็นต้นกำลังในการขับเคลื่อนของยานพาหนะขนาดเล็ก เช่น รถยนต์ไฟฟ้า รถไฟฟ้าที่ใช้ในการเล่นกอล์ฟ รถยกของแบบส้อม เป็นต้น

หมายเหตุ : สมการ 2.13 – 2.15 [ที่มา ความคุมเครื่องกลไฟฟ้าด้วยอิเล็กทรอนิกส์กำลัง 1 : หน้า 253]

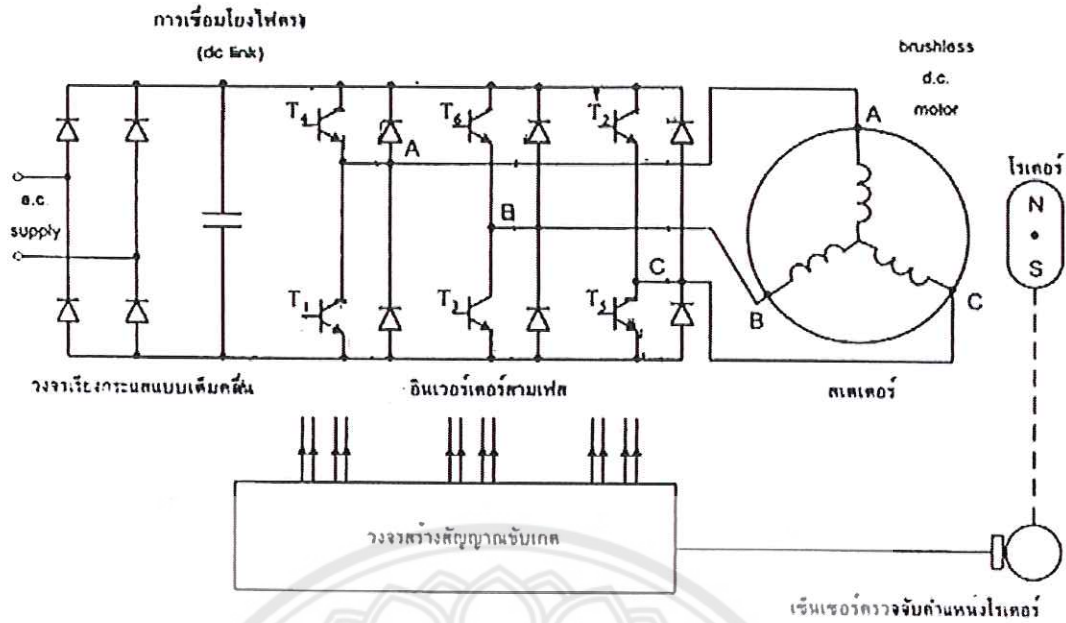
2.1.6 มอเตอร์ไฟฟ้ากระแสตรงแบบไร้แปรงถ่าน

Self-Controlled Synchronous Motor หรือนิยมเรียกอีกชื่อว่า “Brushless Synchronous Motor” อย่างไรก็ตาม สำหรับชื่อ “Brushless Motor” ยังมีความคลุมเครืออยู่มาก เพราะซิงโครนัสมอเตอร์แบบแม่เหล็กถาวรและซิงโครนัสรีลักแตนซ์มอเตอร์ ก็ไม่มีแปรงถ่านเช่นกัน ทำให้ซิงโครนัสมอเตอร์ทั้งสองแบบที่กล่าวข้างต้นก็อาจจะหมายถึง “Brushless Motor” ได้เช่นกัน

สำหรับการขับเคลื่อนมอเตอร์ไฟฟ้ามักจะนิยมเรียกว่า “Self-Controlled Synchronous Motor” ซึ่งจะมีย่านความถี่ในการทำงานและสามารถปรับความถี่ของแหล่งจ่ายอาร์เมเจอร์ที่สเตเตอร์ได้ โดยจะมีค่าแปรผันตามความเร็วของตัวโรเตอร์ เป็นผลทำให้สนามแม่เหล็กที่เกิดจากอาร์เมเจอร์ (Filed Armature) จะหมุนที่ความเร็วเดียวกันกับโรเตอร์ตลอดเวลา ส่วนสนามแม่เหล็กหมุนที่สเตเตอร์หรืออาร์เมเจอร์ และโรเตอร์จะหมุนที่ความเร็วซิงโครนัสตลอดเวลา ทำให้ได้แรงบิดอยู่ตัว (Steady Torque) ที่ทุกค่าความเร็ว ทำให้มีความคล้ายกับมอเตอร์ไฟฟ้ากระแสตรง ซึ่งอาร์เมเจอร์และสนามกระตุ้นจะซิงโครนัส (Synchronous) กัน แตกต่างกันเพียงแต่ความสัมพันธ์ของสนามที่เกิดจาก อาร์เมเจอร์และสนามกระตุ้นของมอเตอร์ไฟฟ้ากระแสตรง สนามแม่เหล็กจะคงที่ที่ทุกค่าความเร็วเท่านั้น

อย่างไรก็ตาม Self-Controlled Synchronous Motor จะต้องใช้ตัวตรวจจับตำแหน่งและความเร็วของตัวโรเตอร์ (Rotor Position Sensor) และยังต้องการการปรับความถี่ที่ป้อนให้กับสเตเตอร์ที่มีความแม่นยำสูง โดยตัวตรวจจับตำแหน่งของตัวโรเตอร์จะทำหน้าที่ในการแปลงข้อมูล (Encoder) ที่ได้จากตัวโรเตอร์ เพื่อนำไปใช้ในการควบคุมการทำงานของสวิตซ์ในวงจรอินเวอร์เตอร์ที่ป้อนให้แก่ขดลวดอาร์เมเจอร์

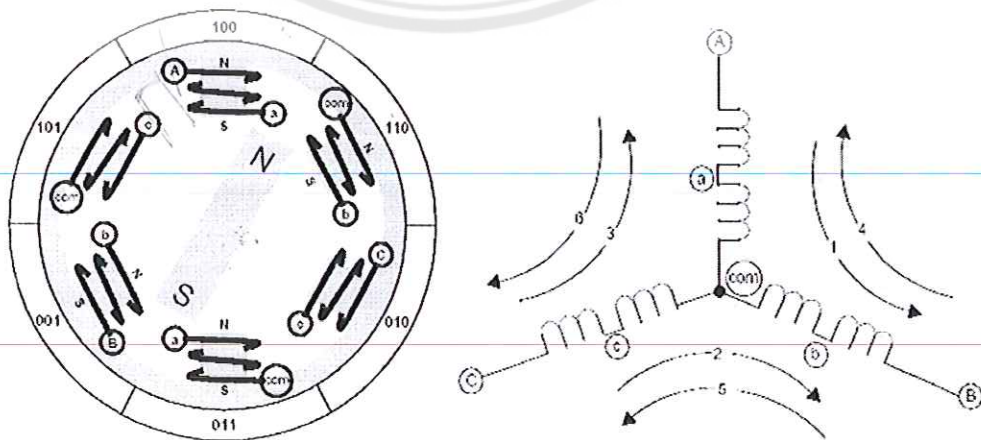
โดยทั่วไป การรวมกันของซิงโครนัสมอเตอร์ที่มีโรเตอร์แบบแม่เหล็กถาวร วงจรอินเวอร์เตอร์แบบสามเฟสและตัวตรวจจับตำแหน่งของตัวโรเตอร์ จะนิยมเรียกว่า “Brushless D.C Motor” โดยตัวตรวจจับตำแหน่งโรเตอร์ (Rotor Position Sensor) และอินเวอร์เตอร์จะทำหน้าที่เสมือนเป็นแปรงถ่านและคอมมิวเตเตอร์ที่ใช้ในมอเตอร์ไฟฟ้ากระแสตรง แต่ถ้าในกรณีที่ใช้ร่วมกับแหล่งจ่ายไฟฟ้ากระแสสลับหรือรับไฟโดยตรงจากการไฟฟ้า ผ่านวงจรไซโคลคอนเวอร์เตอร์ หรือดิซิงค์อินเวอร์เตอร์ ดังแสดงในรูปที่ 2.4 บางครั้งก็จะเรียกว่า “Brushless A.C. Motor”



รูปที่ 2.4 การขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงแบบไร้แปรงถ่าน
[ที่มา อิเล็กทรอนิกส์กำลัง : หน้า 765]

2.1.7 การทำงานของมอเตอร์ไร้แปรงถ่าน

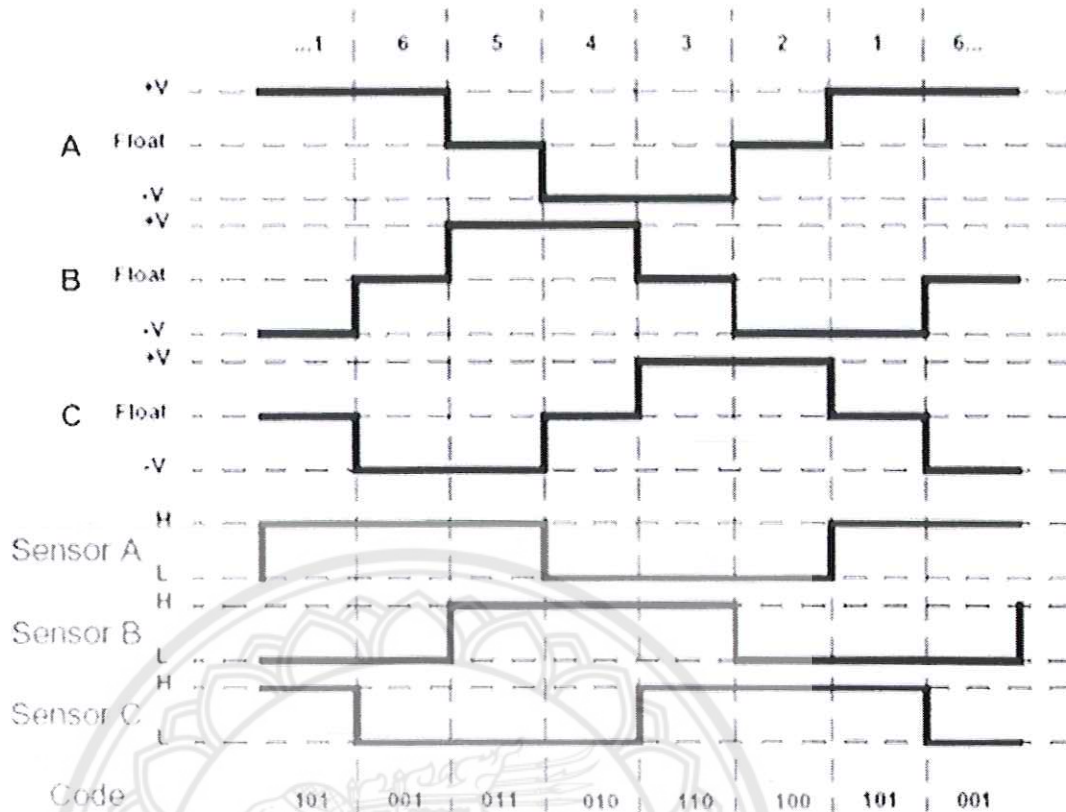
ในการทำงานของมอเตอร์ไร้แปรงถ่าน จะทำงานแบบง่ายๆ โดยมีแปรงถ่านเป็นตัวคอย สับเปลี่ยนการจ่ายกระแสไฟให้กับขดลวด เพื่อสร้างสนามแม่เหล็กที่เหมาะสม มอเตอร์ชนิดนี้จึงมี สายเพียง 2 เส้น เท่านั้น แต่ในมอเตอร์ไร้แปรงถ่านจะมีขดลวดอยู่ 3 ชุด และจะทำงานครั้งละ 2 ชุด สับเปลี่ยนกัน เพื่อสร้างสนามแม่เหล็กที่เหมาะสม จะเห็นได้ว่ามอเตอร์ไร้แปรงถ่านจะมีสาย จำนวน 3 เส้น ต่อกออกมา ดังรูปที่ 2.5 คือ ขั้ว A, B และ C



รูปที่ 2.5 ภาพตัดของมอเตอร์ไร้แปรงถ่าน

จากรูปที่ 2.5 ด้านขวาแสดงภาคตัดมอเตอร์ จะเห็นตัวหมุน (Rotor) ที่เป็นแม่เหล็กถาวร อยู่ตรงกลาง และขดลวด A, B และ C อยู่รอบๆ แต่หากเป็นมอเตอร์ที่เปลือกหมุน (Out-Runner) จะกลับกันคือขดลวดจะอยู่ตรงกลางแล้วมีเปลือกแม่เหล็กเป็นวงกลมอยู่รอบนอกแทน

ในรูปที่ 2.5 จะมีกระแสไหลผ่านขดลวดจาก A ไป B ตามเส้นทางที่ 1 ทำให้เกิดสนามแม่เหล็กสำหรับดึงดูดแกนหมุนให้อยู่ระหว่างขั้วของ A และ B หรือมอเตอร์จะวางตัวอยู่ตามรูปที่ 2.5 และหากให้กระแสไหลตามเส้นทางที่ 6 คือ จากขั้ว A และ C จะทำให้เกิดสนามแม่เหล็กระหว่างขั้ว A และ C จะทำให้ตัวหมุนเคลื่อนไปในทิศทางทวนเข็มนาฬิกาไปอยู่ระหว่างขั้ว C และ A แทน หรือขั้ว N ประมาณเลข 11 บนหน้าปัดนาฬิกา และหากเปลี่ยนกระแสให้ไหลในเส้นทางที่ 5 คือ จาก B ไปยัง C ต่อไปอีก ตัวหมุนจะหมุนทวนเข็มนาฬิกาต่อไปอีก ขั้ว N จะอยู่ที่ระหว่างขั้ว B และ C หรือที่ประมาณ 9 นาฬิกา ดังนั้นหากให้เกิดการหมุนต่อเนื่อง จะต้องทำการสับขั้วให้กระแสไหลในจังหวะที่เหมาะสมตามลำดับไป ดูจากรูปที่ 2.6 และเมื่อกระทำต่อเนื่องไปจนครบ 6 ขั้ว จะเกิดการหมุน 1 รอบพอดี ในระหว่างขั้นต่างๆ นี้ ขั้ว A, B และ C จะถูกสับเปลี่ยนไปมาระหว่างบวกและลบ ทำให้ตัวควบคุมจะต้องใช้ FET ต่อกันเป็น Bridge อย่างน้อย 6 ตัว (3 PFET + 3 NFET) เพื่อให้สามารถสับเปลี่ยนขั้วได้ จึงไม่แปลกใจเมื่อพบว่า Speed จะมี FET ตัวเล็กๆ ต่อกันอยู่เป็นชุดๆ อยู่จำนวนมากน้อยตามกระแสที่จ่ายได้



รูปที่ 2.6 คลื่นสัญญาณควบคุม

จากหลักการทำงานข้างต้น จะเห็นได้ว่าการสับเปลี่ยนขั้วมีความสัมพันธ์กับตำแหน่งของตัวหมุนเป็นอย่างมาก การสับเปลี่ยนขั้วเพื่อให้เกิดสนามแม่เหล็กที่เหมาะสม จะทำให้ได้แรงบิดที่ดี โดยแรงบิดจะเกิดขึ้นสูงสุด เมื่อตัวหมุนทำมุมกับสนามแม่เหล็ก 90 องศา ดังนั้นในการควบคุมมอเตอร์ไร้แปรงถ่าน เราจะได้ยินคำว่า Sensorless อีกคำหนึ่ง

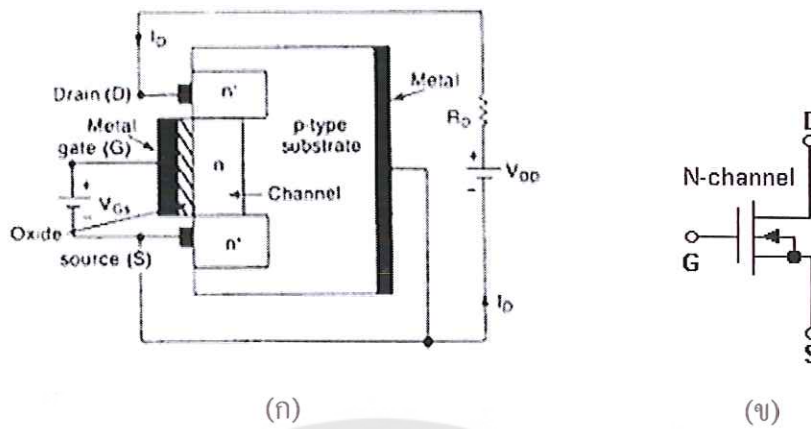
2.2 มอสเฟตกำลัง

มอสเฟตกำลังจะแบ่งออกเป็น 2 ชนิด คือ ดีพลีชัน (Depletion) และ เอนเทนซ์เมนต์ (Enhancement) แต่ละประเภทยังแบ่งออกเป็น 2 แบบ คือแบบแซนแนลเอ็น (N - Channel) และแบบแซนแนลพี (P - Channel)

2.2.1 ประเภทของมอสเฟตกำลัง

มอสเฟตกำลังเป็นอุปกรณ์ที่มี 3 ขั้ว คือขั้วซอร์ส (Source ; S), เดรน (Drain ; D) และเกท (Gate ; G) แบ่งออกเป็น 2 ประเภทคือ

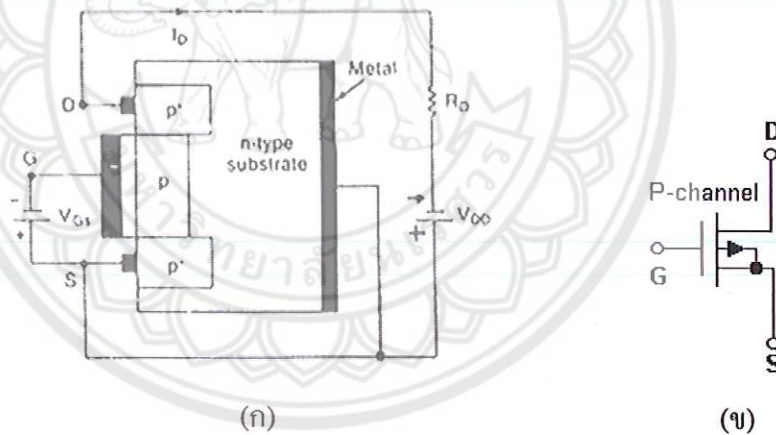
1. มอสเฟตแบบดีพลีชัน (Depletion MOSFET) เรียกสั้นๆ ว่า “ดีมอสเฟต”



รูปที่ 2.7 (ก) มอสเฟตชนิดดีพลีชัน แชนแนลเอ็น

(ข) สัญลักษณ์ของมอสเฟตชนิดดีพลีชัน แชนแนลเอ็น

[ที่มา POWER ELECTRONIC CIRCUITS, DEVICES AND APPLICATION : หน้า 138]



รูปที่ 2.8 (ก) มอสเฟตชนิดดีพลีชัน แชนแนลพี

(ข) สัญลักษณ์ของมอสเฟตชนิดดีพลีชัน แชนแนลพี

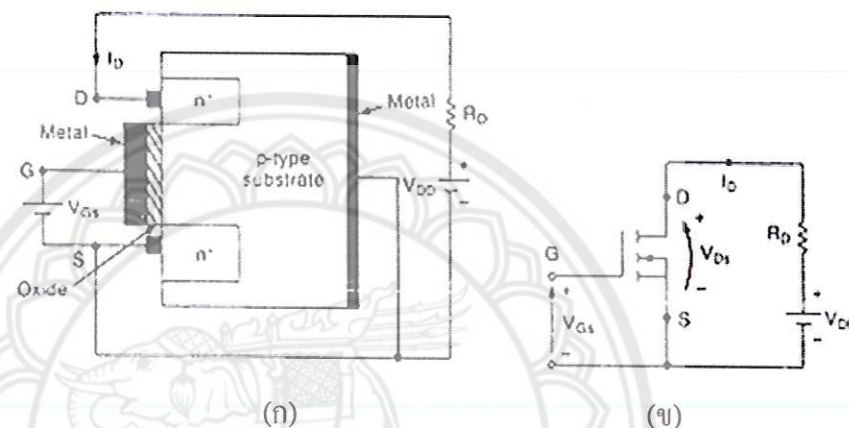
[ที่มา POWER ELECTRONIC CIRCUITS, DEVICES AND APPLICATION : หน้า 138]

ดีมอสเฟตแบบแชนแนลเอ็น (N) ประกอบขึ้นจากแผ่นผลึกฐานพี (P-Substrate) ที่เป็นสารกึ่งตัวนำทำจากซิลิกอนขั้ว D และขั้ว S ต่อกับบริเวณที่มีการกระตุ้นหรือโด๊ปให้เป็นบริเวณสารกึ่งตัวนำเอ็น (N-Doped Region) ทั้งสองส่วนนี้จะเชื่อมกับแชนแนลเอ็น สำหรับขั้ว G จะต่อกับวัสดุผิวนอกที่เป็นโลหะโดยมีซิลิกอนไดออกไซด์ (SiO₂) กั้นแชนแนลเอ็น (n) กับขั้ว G (ซิลิกอน-

ไดออกไซด์เป็นฉนวนประเภทไดอิเล็กทริก) เมื่อมีสนามไฟฟ้าจ่ายเข้ามาที่ชั้นของ SiO₂ ก็จะสร้างสนามไฟฟ้าด้านและสร้างชั้นฉนวนขึ้นภายในตัวเอง เพื่อกั้นขั้วเกตกับเซนแนล แสดงว่าไม่มีการต่อโดยตรงระหว่างขั้ว G กับเซนแนลของมอสเฟตชั้นที่เป็นฉนวน SiO₂

หลักการที่กล่าวมานี้ ใช้ได้กับดีมอสเฟตแบบเซนแนลพี (P) ดังรูปที่ 2.8 โดยขั้วของ V_{DS}, I_{DS} และ V_{GS} เปลี่ยนเป็นขั้วตรงข้ามกับกรณีดีมอสเฟตแบบเซนแนลเอ็น (N)

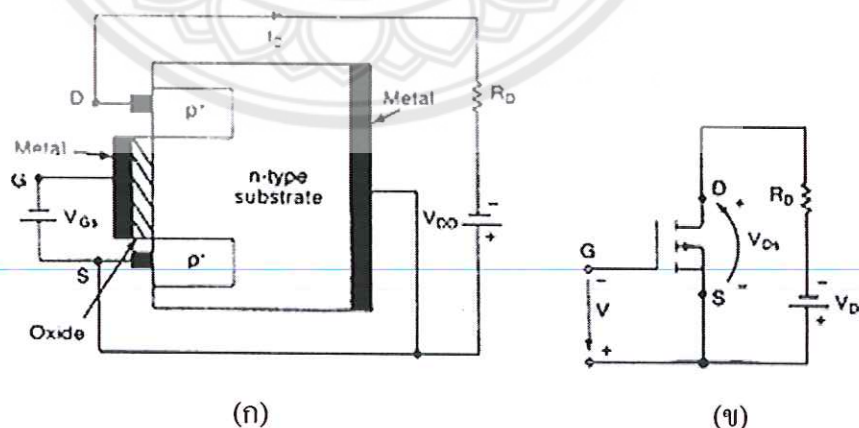
2. มอสเฟตแบบเอนเทนซ์เมนต์ (Enhancement MOSFET) หรือเรียกว่า “อีมอสเฟต”



รูปที่ 2.9 (ก) มอสเฟตชนิดเอนเทนซ์เมนต์ เซนแนลเอ็น

(ข) สัญลักษณ์มอสเฟตชนิดเอนเทนซ์เมนต์ เซนแนลเอ็น

[ที่มา POWER ELECTRONIC CIRCUITS, DEVICES AND APPLICATION : หน้า 139]



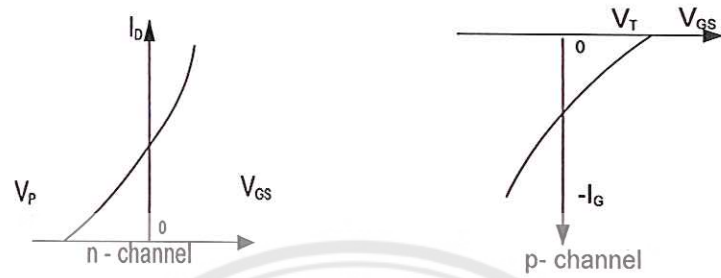
รูปที่ 2.10 (ก) มอสเฟตชนิดเอนเทนซ์เมนต์ เซนแนลพี

(ข) สัญลักษณ์มอสเฟตชนิดเอนเทนซ์เมนต์ เซนแนลพี

[ที่มา POWER ELECTRONIC CIRCUITS, DEVICES AND APPLICATION : หน้า 139]

2.2.2 คุณสมบัติของมอสเฟตกำลัง

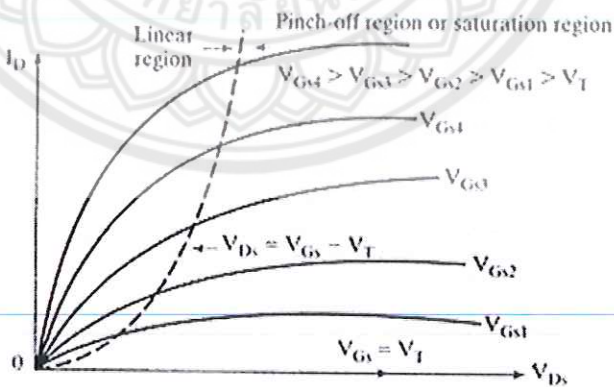
มอสเฟตกำลังเป็นอุปกรณ์ประเภทใช้แรงดันในการควบคุมและมีอิมพีแดนซ์อินพุตสูงมาก ขั้วเกตจะดึงกระแสที่ทำให้จากการทำงานต่ำมากแค่นาโนแอมแปร์ และมีค่าความนำภายในทรานซิสเตอร์เรียกว่าทรานส์คอนดักแตนซ์ (Transconductance ; g_m) โดยเป็นอัตราส่วนของ I_D ต่อแรงดันเกต (V_G) เป็นตัวกำหนดคุณลักษณะการทำงานของมอสเฟต



(ก) Depletion – type MOSFET



(ข) Enhancement – type MOSFET



(ค)

- รูปที่ 2.11 (ก) คุณสมบัติกราฟการทำงานของดีมอสเฟต แชนแนลเอ็น และ แชนแนลพี
- (ข) คุณสมบัติกราฟการทำงานของอีมอสเฟต แชนแนลเอ็น และ แชนแนลพี
- (ค) คุณสมบัติเอาต์พุตของอีมอสเฟตกำลังแบบแชนแนลเอ็น

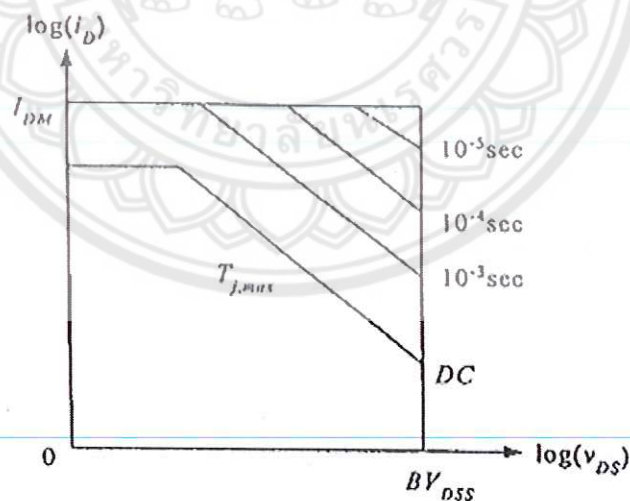
คุณลักษณะกราฟการทำงานของดีมอสเฟตและอีมอสเฟตแบบแซนแนลเอ็น (n) และแบบแซนแนลพี (p) เป็นดังรูปที่ 2.11 (ก) และ 2.11 (ข) ส่วนรูปที่ 2.11 (ค) แสดงคุณลักษณะเอาต์พุตของอีมอสเฟตแบบแซนแนลเอ็น ซึ่งมีบริเวณการทำงาน 3 บริเวณคือ

1. บริเวณคัตออฟ (Cut Off Region) เกิดขึ้นเมื่อแรงดัน V_{GS} มีค่าน้อยกว่าหรือเท่ากับแรงดันจุดเปลี่ยน (Threshold Voltage, V_T)
2. บริเวณพินชออฟหรือบริเวณอิ่มตัว (Pinch – Off or Saturation Region) เกิดขึ้นเมื่อแรงดัน V_{DS} มีค่าน้อยกว่าหรือเท่ากับแรงดัน $V_{GS} - V_T$
3. บริเวณที่เป็นเชิงเส้น (Linear Region) เกิดขึ้นเมื่อแรงดัน V_{DS} มีค่ามากกว่าหรือเท่ากับแรงดัน $V_{GS} - V_T$

ในบริเวณที่เป็นเชิงเส้น I_D จะเปลี่ยนแปลงตามสัดส่วนของ V_{DS} และเนื่องจาก I_D มีค่าสูงขณะ V_D มีค่าต่ำ อีมอสเฟตจึงใช้บริเวณที่เป็นเชิงเส้นนี้สำหรับแสดงปฏิกริยาในการสวิตช์

2.2.3 พื้นที่ทำงานปลอดภัยของมอสเฟตกำลัง

พื้นที่ทำงานปลอดภัยของมอสเฟตจะมีขีดจำกัดด้าน T_{jmax} พื้นที่ทำงานปลอดภัยจะจำกัดโดยกระแสตรงสูงสุด I_{DM} กำลังสูญเสียหรืออุณหภูมิที่รอยต่อสูงสุด T_{jmax} และแรงดันทลาย BV_{DSS} ดังรูปที่ 2.12 แสดงพื้นที่ทำงานปลอดภัยของมอสเฟตกำลังที่ใช้งานเป็นสวิตช์ ซึ่งพื้นที่จะมีลักษณะเป็นสี่เหลี่ยม พื้นที่ทำงานเมื่อไบอัสย้อนกลับจะไม่แตกต่างกัน



รูปที่ 2.12 พื้นที่ทำงานปลอดภัยของมอสเฟตกำลัง

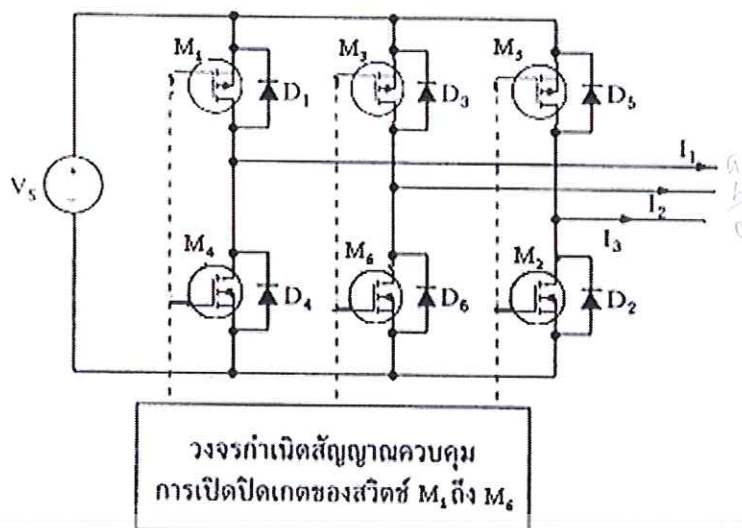
[ที่มา อิเล็กทรอนิกส์กำลัง : หน้า 335]

2.3 อินเวอร์เตอร์

อินเวอร์เตอร์คือวงจรแปลงผันพลังงานไฟฟ้ากระแสตรงให้เป็นไฟฟ้ากระแสสลับ เรียกอีกอย่างหนึ่งว่า เครื่องผกผัน เพื่อให้ได้แรงดันไฟฟ้ากระแสสลับที่มีค่าแรงดันคงที่หรือปรับค่าได้ หรือเพื่อให้ได้ความถี่ไฟฟ้ากระแสสลับเอาต์พุตให้มีค่าคงที่หรือปรับค่าได้ โดยใช้หลักการปรับค่าแรงดันไฟฟ้ากระแสตรงด้านอินพุต หรือใช้หลักการควบคุมเทคนิคการสวิตช์ภายในตัวอินเวอร์เตอร์ เช่น ใช้เทคนิคแบบมอดูเลตความกว้างพัลส์ (PWM) เป็นต้น หรืออาจใช้วิธีปรับค่าแรงดันไฟฟ้ากระแสตรงด้านอินพุตของอินเวอร์เตอร์ก็ได้เช่นกัน สำหรับอินเวอร์เตอร์ที่ใช้ในงานควบคุมความเร็วของมอเตอร์ 3 เฟส จะใช้หลักการควบคุมอัตราส่วนของแรงดันเอาต์พุตและความถี่เอาต์พุตให้คงที่ (V/f Constant) อีกด้วย เพื่อให้สามารถควบคุมการหมุนให้คงที่ได้ แต่สำหรับอินเวอร์เตอร์ที่ใช้ในงานอื่นๆ ที่ต้องการความถี่เอาต์พุตคงที่ ก็อาจใช้อินเวอร์เตอร์ที่ควบคุมการเปิด-ปิดสวิตช์อิเล็กทรอนิกส์กำลังด้วยวิธีที่ไม่ซับซ้อน เช่น วิธีมอดูเลตแบบ 180° หรือแบบ 120° เป็นต้น สำหรับรูปคลื่นของแรงดันเอาต์พุตไฟสลับของอินเวอร์เตอร์ในอุดมคติคือรูปคลื่นไซน์ แต่ในทางปฏิบัตินั้นจะไม่ใช่รูปคลื่นไซน์ แต่จะเป็นรูปสี่เหลี่ยมหรือกึ่งสี่เหลี่ยมซึ่งมีฮาร์มอนิกส์ปะปนอยู่มาก ซึ่งด้วยเทคนิคการสวิตช์แบบ PWM จะทำให้ฮาร์มอนิกส์ดังกล่าวลดลงและทำให้ประสิทธิภาพของอินเวอร์เตอร์มีค่าสูงขึ้น

2.3.1 อินเวอร์เตอร์ 3 เฟส

อินเวอร์เตอร์ 3 เฟส ประกอบไปด้วยสวิตช์อิเล็กทรอนิกส์กำลัง 6 ตัว คือ สวิตช์ M_1 และ M_4 ต่ออยู่กับเอาต์พุตเฟสที่ 1 สวิตช์ M_3 และ M_6 ต่ออยู่กับเอาต์พุตเฟสที่ 2 และสวิตช์ M_5 และ M_2 ต่อเข้ากับเอาต์พุตเฟสที่ 3 ดังรูปที่ 2.13 สัญญาณสำหรับควบคุมการเปิด-ปิดสวิตช์แต่ละตัวทำงานได้ 2 แบบ คือแบบนำกระแส 180° และแบบการนำกระแส 120° สวิตช์อิเล็กทรอนิกส์ที่ใช้ในวงจรในรูปที่ 2.13 คือ มอสเฟต (M_1 ถึง M_6) และ ไดโอด (D_1 ถึง D_6)

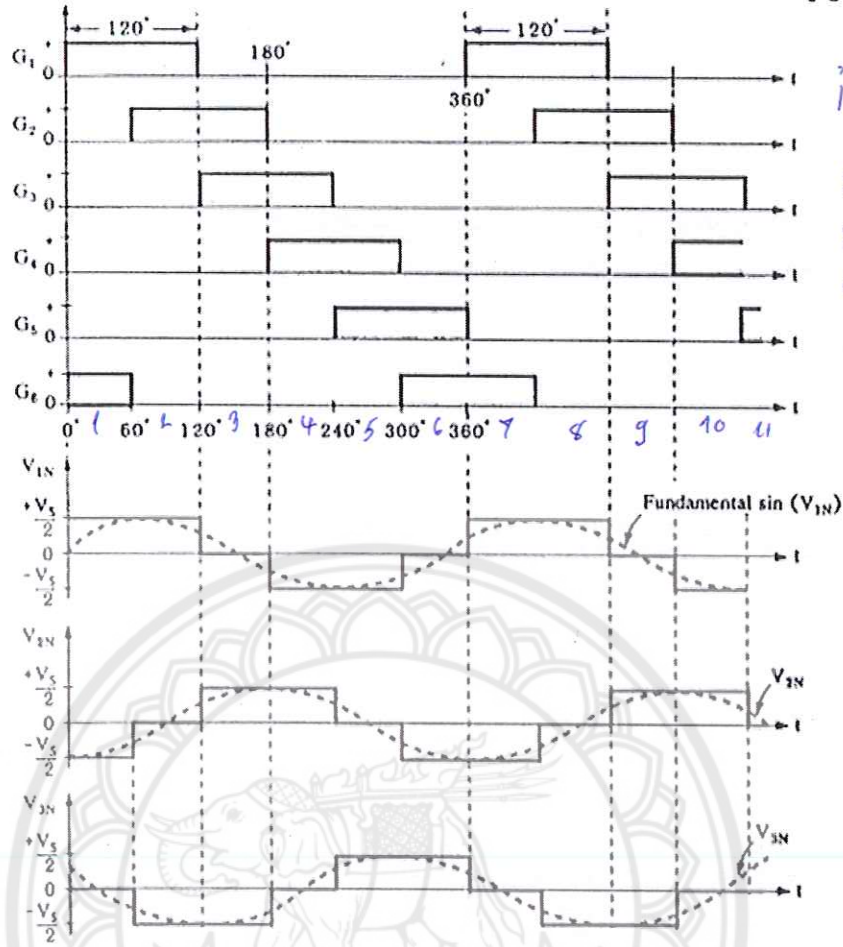


รูปที่ 2.13 วงจรอินเวอร์เตอร์ 3 เฟส

[ที่มา อิเล็กทรอนิกส์กำลัง 2 : หน้า 367]

อินเวอร์เตอร์ 3 เฟส แบบนำกระแส 120° จากวงจรในรูปที่ 2.13 เมื่อควบคุมให้สวิตช์-อิเล็กทรอนิกส์กำลังแต่ละตัวนำกระแสตัวละ 120° โดยใช้สัญญาณควบคุมการเปิด-ปิดเกตของมอสเฟต ตามลำดับ G_1 ถึง G_6 ดังรูปที่ 2.14 จะพบว่ามอสเฟต 2 ตัวทำงานพร้อมกันในแต่ละช่วงตามลำดับ ดังนี้คือ M_6M_1 , M_1M_2 , M_2M_3 , M_3M_4 , M_4M_5 , M_5M_6 และ M_6M_1 จึงได้รูปคลื่นแรงดันคร่อมโหลดระหว่างเฟส V_{1N} , V_{2N} และ V_{3N} ดังแสดงในรูปที่ 2.14

4900169



15075592
 2/5,
 17570
 2548
 c.v.

รูปที่ 2.14 คลื่นสัญญาณควบคุมอินเวอร์เตอร์แบบนำกระแส 120° และรูปคลื่นแรงดันเฟส [ที่มา อิเล็กทรอนิกส์กำลัง 2 : หน้า 368]

2.3.2 การควบคุมค่าแรงดันเอาต์พุตของอินเวอร์เตอร์ด้วยวิธีมอดูเลตความกว้างพัลส์

อินเวอร์เตอร์ 3 เฟสแบบนำกระแส 180° เป็นอินเวอร์เตอร์ชนิดปรับค่าความถี่เอาต์พุตได้ แต่แรงดันเอาต์พุตมีค่าคงที่ ส่วนอินเวอร์เตอร์ 3 เฟสแบบนำกระแส 120° ก็เป็นชนิดแรงดันคงที่ ความถี่ปรับค่าได้เช่นเดียวกัน เมื่อนำอินเวอร์เตอร์ 3 เฟสไปขับเคลื่อนมอเตอร์ 3 เฟส จึงต้องควบคุมให้อัตราส่วนของ V/f ให้มีค่าคงที่ตลอดย่านการควบคุม ดังนั้นจึงต้องใช้เทคนิคการปรับปรุงรูปแบบการสวิตช์มาควบคุมค่าแรงดันเอาต์พุต เทคนิคดังกล่าวคือวิธีการมอดูเลตความกว้างพัลส์ (Pulse Width Modulation) จุดประสงค์ของการมอดูเลตความกว้างพัลส์ก็เพื่อปรับค่าแรงดันเอาต์พุตและลดฮาร์มอนิกส์ที่ไม่ต้องการ เพื่อปรับปรุงให้คลื่นแรงดันและกระแสเอาต์พุตของอินเวอร์เตอร์ใกล้เคียงรูปคลื่นไซน์มากที่สุด

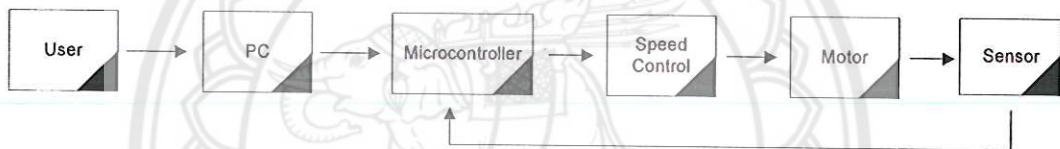
บทที่ 3

การออกแบบการทดลอง

ในบทที่ 2 ได้กล่าวถึงรายละเอียดและหลักการรวมถึงทฤษฎีต่างๆ มาแล้ว ทางคณะผู้จัดทำจึงได้เริ่มการดำเนินการออกแบบและสร้างวงจรควบคุมการทำงานของมอเตอร์ตามทฤษฎี แล้วได้นำไปทดลองขับมอเตอร์จริง ดังรายละเอียดต่อไปนี้

3.1 ระบบการทำงานของวงจรควบคุมการทำงานของมอเตอร์

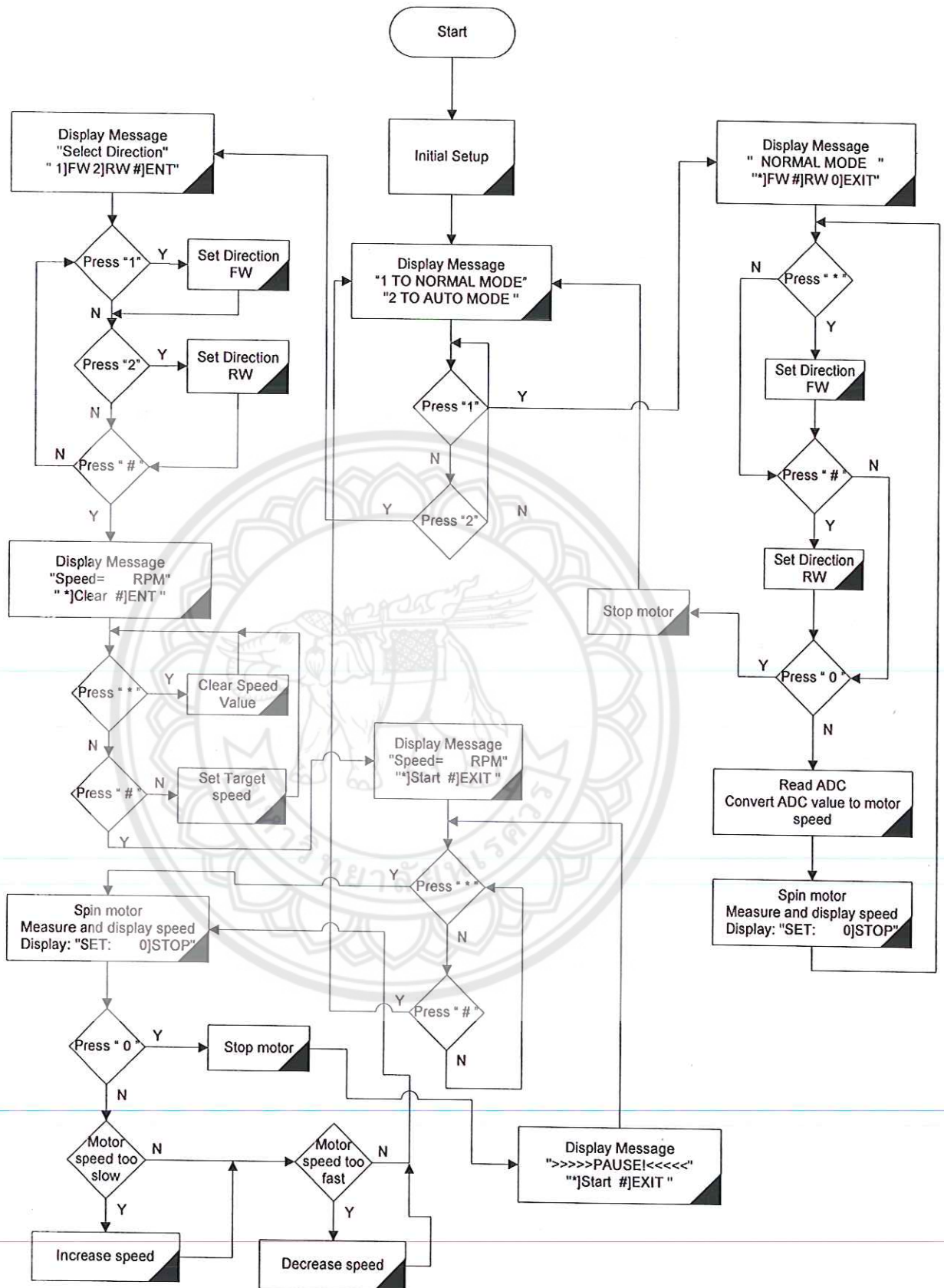
ระบบการทำงานของวงจรควบคุมการทำงานของมอเตอร์ที่ได้ออกแบบและสร้างขึ้นมานั้น ประกอบด้วยส่วนต่างๆ โดยการทำงานของวงจรควบคุมการทำงานของมอเตอร์นั้นสามารถอธิบายได้ดังรูปที่ 3.1 ซึ่งแสดงบล็อกไดอะแกรมของการทำงานของวงจรควบคุมการทำงานของมอเตอร์



รูปที่ 3.1 บล็อกไดอะแกรมของวงจรควบคุมมอเตอร์

3.2 การออกแบบการเขียนโปรแกรม

การเขียนโปรแกรมโดยการรับค่าจากคีย์บอร์ด (Key Board) หรือคีย์แพด (Key Pad) แล้วเอาต์พุตออกทางพอร์ตขนาน เพื่อใช้ในการควบคุมติดต่อกับอุปกรณ์และวงจรอิเล็กทรอนิกส์ ซึ่งสามารถดูโปรแกรมการควบคุมการทำงานได้ที่ภาคผนวก ข



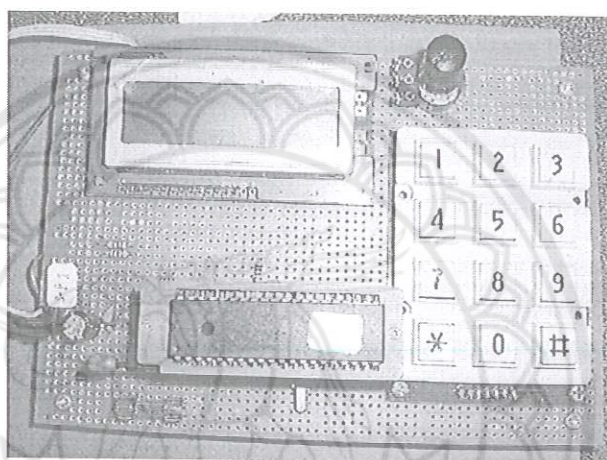
รูปที่ 3.2 โฟลว์ชาร์ตแสดงการทำงานของโปรแกรมที่ใช้ควบคุม

3.3 การออกแบบวงจรควบคุมและวงจรกำลัง

ในวงจรที่ใช้ในโครงการนี้เป็นวงจรที่ใช้มอสเฟต ซึ่งเป็นอุปกรณ์สวิตช์โดยใช้ไอซีเบอร์ LM2940C เป็นตัวขับมอสเฟต และใช้มอสเฟตเบอร์ IRF9540 และเบอร์ IRFZ44 โดยวงจรวงจรขับเคลื่อนมอเตอร์ ที่ใช้ไมโครคอนโทรลเลอร์ PIC16F877 ควบคุมการทำงานของมอเตอร์ซึ่งจะอยู่ในภาคผนวก ก

3.4 การออกแบบการทดลอง

3.4.1 วงจรขับเคลื่อนมอเตอร์



รูปที่ 3.3 วงจรขับเคลื่อนมอเตอร์ ลงอุปกรณ์เรียบร้อยแล้ว

การต่อวงจรขับเคลื่อนมอเตอร์ โดยต่อตามรูปที่ 3.3 และวงจรขับเคลื่อนมอเตอร์จะมีวงจรภาคจ่ายไฟต่อรวมอยู่ด้วย โดยมี ไอซีเบอร์ LM 2940C ทำหน้าที่รักษาระดับแรงดันไฟฟ้าในวงจรให้เท่ากับ 5 V จ่ายไฟให้กับวงจร

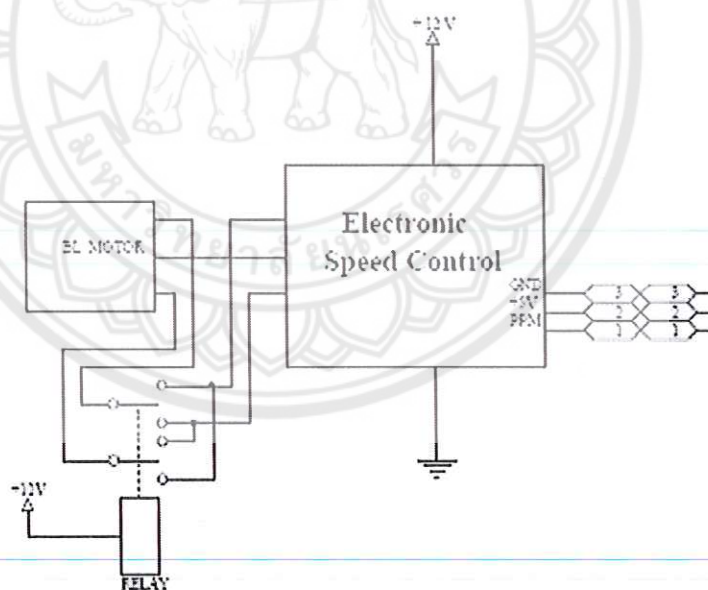
3.4.2 วงจรควบคุมมอเตอร์

วงจรควบคุมมอเตอร์จะมีมอสเฟต ($M_1, M_2, M_3, M_4, M_5, M_6$) ที่คอยทำหน้าที่เป็นสวิทช์ในการควบคุมมอเตอร์ รูปของวงจรควบคุมมอเตอร์จะอยู่ในภาคผนวก ก

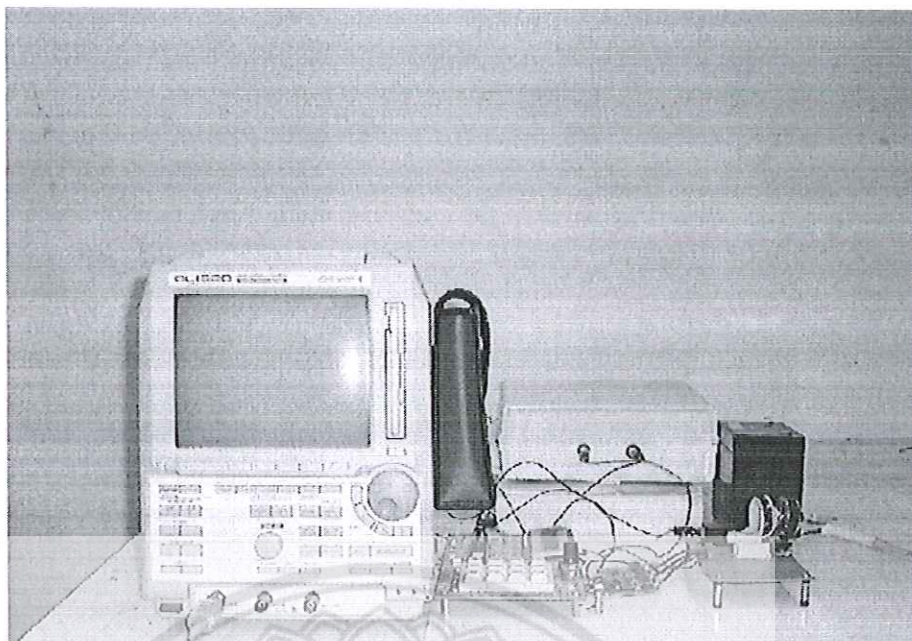
3.4.3 ทำการทดสอบกับวงจรขับมอเตอร์

เมื่อต่อวงจรแต่ละส่วนการทำงานเสร็จเรียบร้อยแล้ว นำวงจรทั้งหมดมาต่อรวมกันดังรูปที่ 3.5 โดยนำมอเตอร์และแหล่งจ่ายไฟมาต่อเข้ากับวงจรควบคุมมอเตอร์ ดังรูปที่ 3.4 แล้วจ่ายไฟ 12 VDC ให้กับวงจรควบคุมมอเตอร์ และใช้ไฟ 5 V DC จ่ายให้กับวงจรขับมอเตอร์ ดังรูปที่ 3.5 จากนั้นเริ่มทำการทดลองโดยปรับค่าของความเร็วรอบจากน้อยไปหามาก ที่ใช้ควบคุมมอเตอร์ โดยที่ไมโครคอนโทรลเลอร์จะสร้างสัญญาณควบคุมให้วงจรขับมอเตอร์ทำงานและวงจรขับมอเตอร์จะไปสั่งให้วงจรควบคุมมอเตอร์ทำงานตาม โปรแกรมที่ป้อนเข้ามา แล้วได้ค่าของกระแสไฟฟ้าแสดงดังกราฟในบทที่ 4

และได้มีการวัดสัญญาณของการ Trigger ของมอสเฟตแต่ละตัวและเฟส A B และ C โดยใช้สโคปวัดกราฟสัญญาณว่าจะได้ตรงกับทฤษฎีหรือคล้อยคลึงกันมากน้อยเพียงใดดังแสดงในบทที่ 4



รูปที่ 3.4 การต่อมอเตอร์และแหล่งจ่ายไฟเข้ากับวงจรควบคุมมอเตอร์



รูปที่ 3.5 วงจรที่ต่อเสร็จแล้วพร้อมทำการทดลอง

ในบทนี้ได้กล่าวถึงการออกแบบและสร้างวงจรควบคุมการทำงานของมอเตอร์ โดยใน ส่วนของโปรแกรมควบคุมการทำงาน ได้เริ่มจากการเขียนโฟลว์ชาร์ตแสดงการทำงานโดยรวมของ มอเตอร์และเขียนโปรแกรมควบคุมการทำงานของมอเตอร์ คือ มอเตอร์หมุนเร็วขึ้น หมุนช้าลง หมุนขวา และหยุดหมุน โดยมีการป้อนค่าของความเร็วรอบด้วยมือและแบบอัตโนมัติ

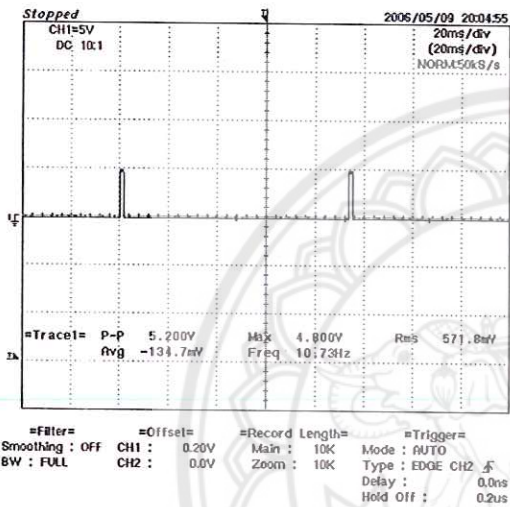
และในส่วนของวงจรขั้วมอเตอร์ได้เริ่มทำการศึกษาการทำงานของอุปกรณ์อิเล็กทรอนิกส์ กำลังต่างๆ ที่ใช้ในการออกแบบวงจรขั้วมอเตอร์ จากนั้นนำอุปกรณ์อิเล็กทรอนิกส์กำลังต่างๆ มา ประกอบรวมกันเป็นวงจรขั้วมอเตอร์ แล้วนำวงจรขั้วมอเตอร์ที่ได้นี้ไปทดลองขั้วมอเตอร์จริง ซึ่ง จะได้ผลการทดลองที่จะกล่าวต่อไปในบทที่ 4

บทที่ 4

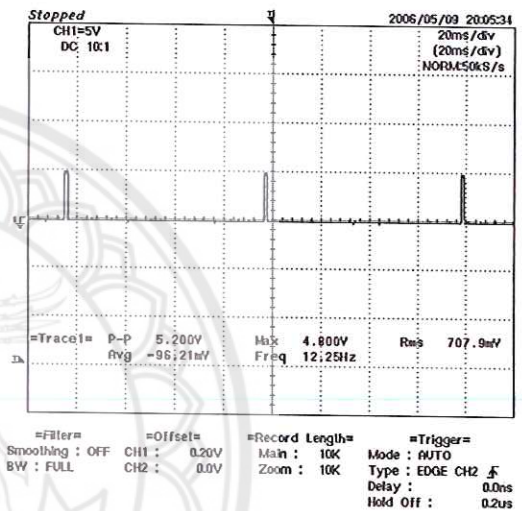
ผลการทดลอง

จากบทที่ 3 ได้ทำการออกแบบวงจร เพื่อใช้ในการทดลองและทำการทดลองแล้ว ในบทนี้จะนำเสนอผลจากการทดลองที่ได้จากการทดลองดังต่อไปนี้

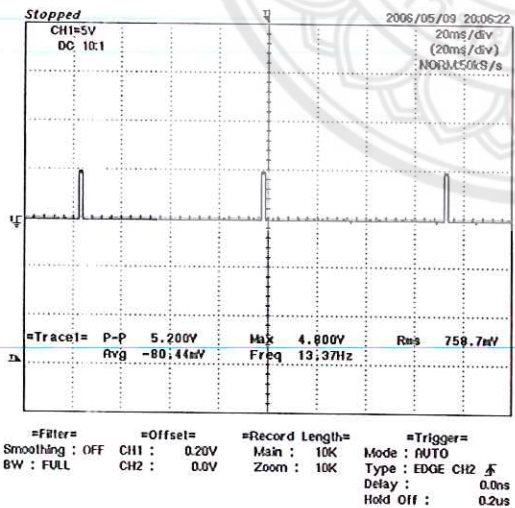
4.1 สัญญาณที่สร้างอินพุต



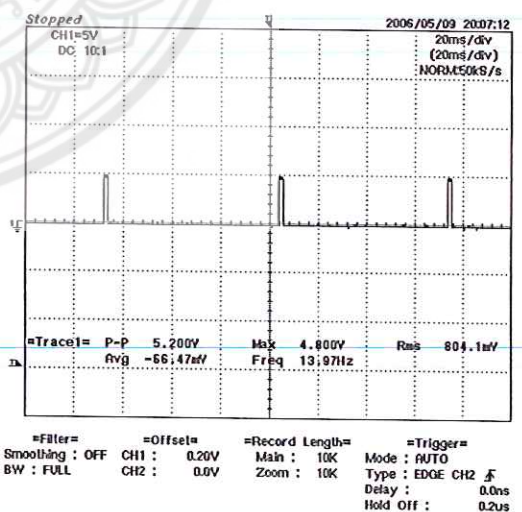
รูปที่ 4.1 ความเร็วรอบ 5,000 rpm



รูปที่ 4.2 ความเร็วรอบ 8,000 rpm



รูปที่ 4.3 ความเร็วรอบ 12,000 rpm

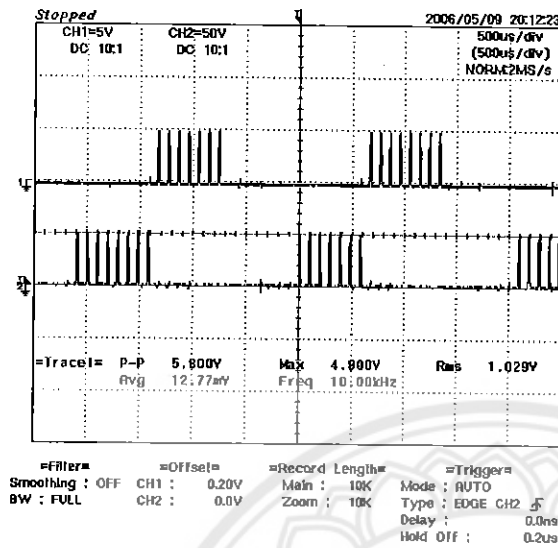


รูปที่ 4.4 ความเร็วรอบ 15,000 rpm

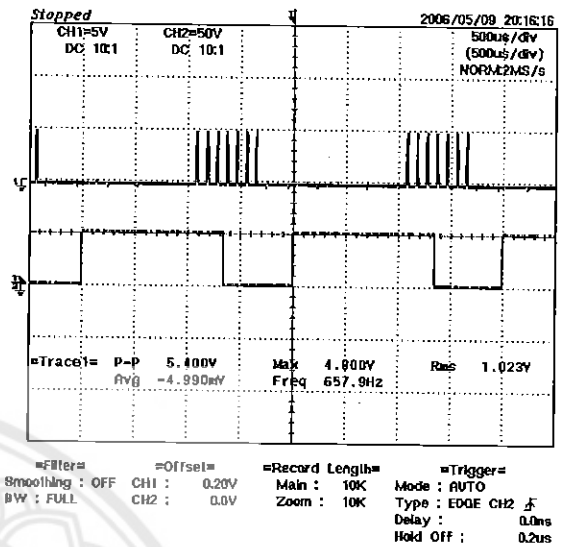
จากหัวข้อที่ 4.1 จะเห็นได้ว่าที่ความเร็วรอบ 5,000 rpm ช่วงเวลาของการ on - off ของสัญญาณแรงดันจะกว้างกว่าที่ความเร็วรอบ 8,000 rpm 12,000 rpm และ 15,000 rpm ตามลำดับ

4.2 สัญญาณที่ได้จากการ Trigger ของมอเตอร์

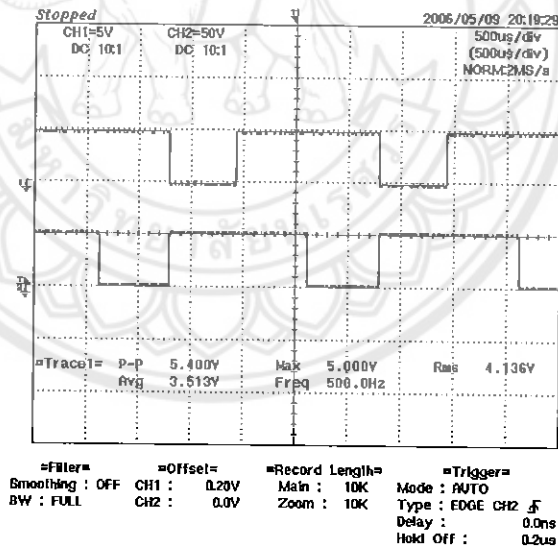
4.2.1 ที่ความเร็วรอบ 5,000 rpm



รูปที่ 4.5 มอเตอร์ตัวที่ 1 และ 2



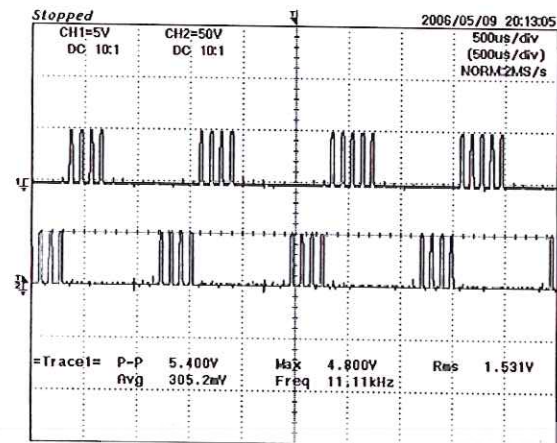
รูปที่ 4.6 มอเตอร์ตัวที่ 3 และ 4



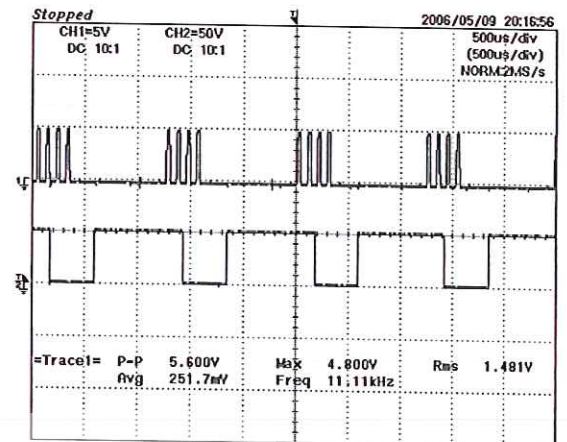
รูปที่ 4.7 มอเตอร์ตัวที่ 5 และ 6

จากรูปที่ 4.5 , 4.6 และรูปที่ 4.7 ซึ่งเป็นการวัดสัญญาณของแรงดันที่ได้จากการ Trigger ของมอเตอร์ ที่ความเร็วรอบ 5,000 rpm ซึ่งจะดูลักษณะการทำงานในช่วงของการ on – off ของมอเตอร์ทั้ง 6 ตัว โดยมอเตอร์ตัวที่ 1 จะทำงานก่อนแล้วตามด้วยมอเตอร์ตัวที่ 2 , 3 , 4 , 5 และตัวที่ 6 ตามลำดับ

4.2.2 ที่ความเร็วรอบ 8,000 rpm



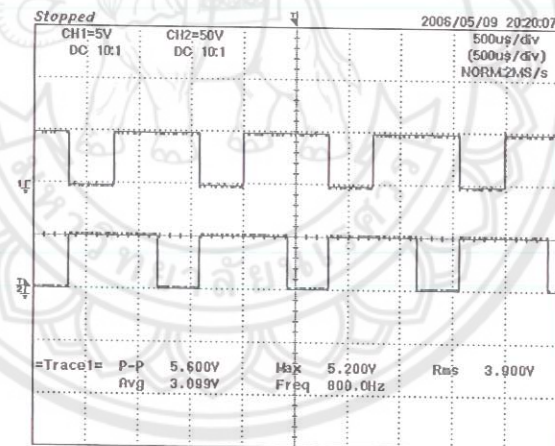
=Filter= Smoothing : OFF BW : FULL
=Offset= CH1 : 0.20V CH2 : 0.0V
=Record Length= Main : 10K Zoom : 10K
=Trigger= Mode : AUTO Type : EDGE CH2 \downarrow Delay : 0.0ns Hold Off : 0.2us



=Filter= Smoothing : OFF BW : FULL
=Offset= CH1 : 0.20V CH2 : 0.0V
=Record Length= Main : 10K Zoom : 10K
=Trigger= Mode : AUTO Type : EDGE CH2 \downarrow Delay : 0.0ns Hold Off : 0.2us

รูปที่ 4.8 มอสเฟตตัวที่ 1 และ 2

รูปที่ 4.9 มอสเฟตตัวที่ 3 และ 4

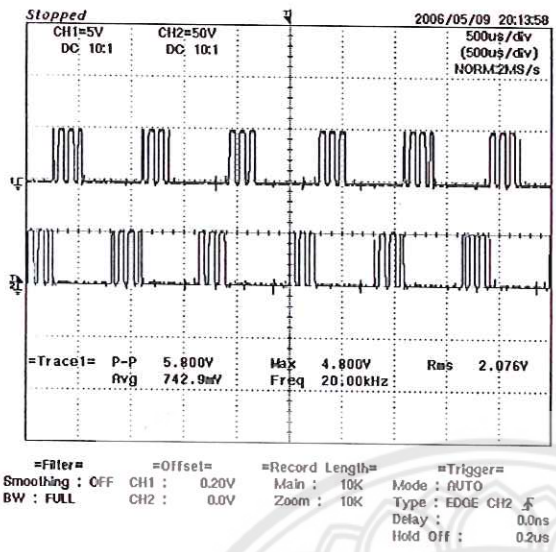


=Filter= Smoothing : OFF BW : FULL
=Offset= CH1 : 0.20V CH2 : 0.0V
=Record Length= Main : 10K Zoom : 10K
=Trigger= Mode : AUTO Type : EDGE CH2 \downarrow Delay : 0.0ns Hold Off : 0.2us

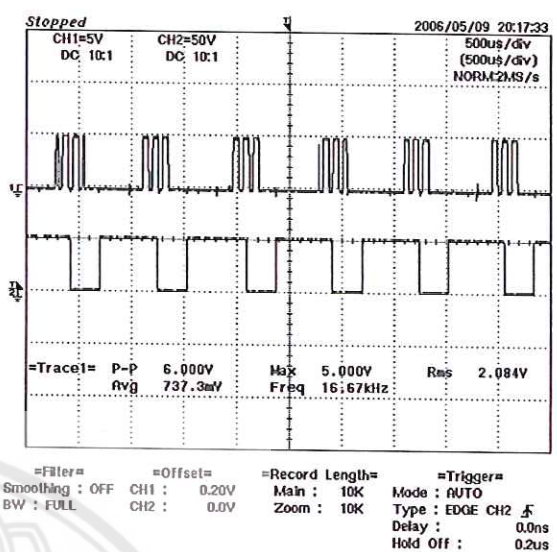
รูปที่ 4.10 มอสเฟตตัวที่ 5 และ 6

จากรูปที่ 4.8 , 4.9 และรูปที่ 4.10 เป็นการวัดสัญญาณของแรงดันที่ได้จากการ Trigger ของมอสเฟต ที่ความเร็วรอบ 8,000 rpm ซึ่งจะดูลักษณะการทำงานในช่วงของการ on – off ของมอสเฟตทั้ง 6 ตัว โดยมอสเฟตตัวที่ 1 จะทำงานก่อนแล้วตามด้วยมอสเฟตตัวที่ 2 , 3 , 4 , 5 และตัวที่ 6 ตามลำดับ

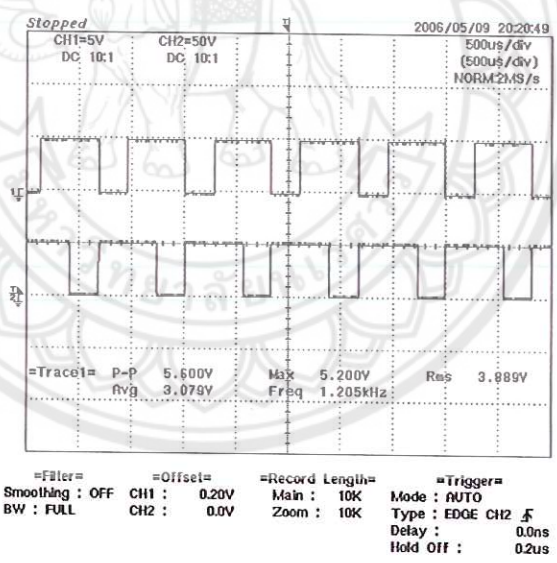
4.2.3 ที่ความเร็วรอบ 12,000 rpm



รูปที่ 4.11 มอสเฟตตัวที่ 1 และ 2



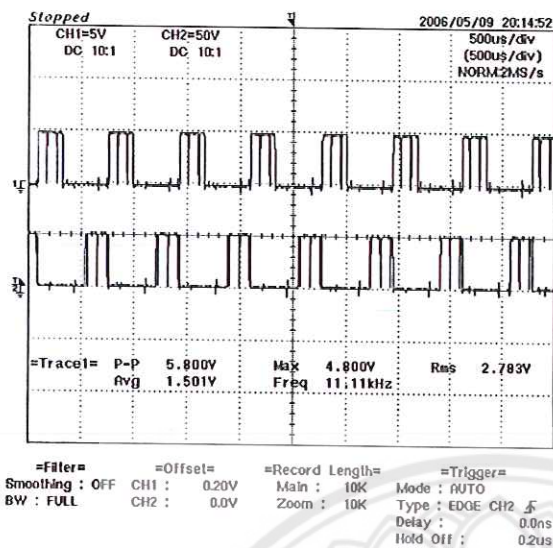
รูปที่ 4.12 มอสเฟตตัวที่ 3 และ 4



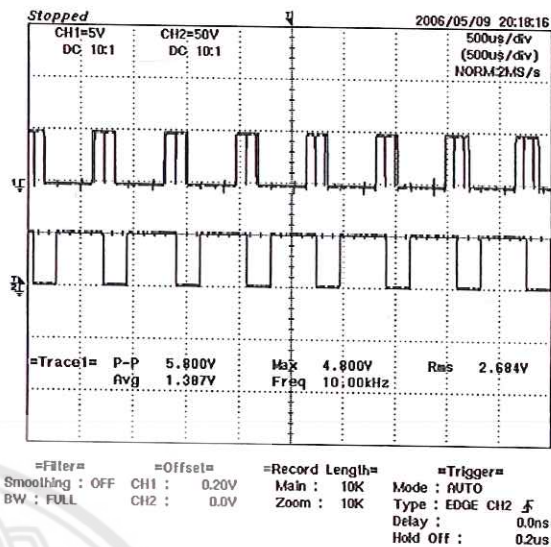
รูปที่ 4.13 มอสเฟตตัวที่ 5 และ 6

จากรูปที่ 4.11 , 4.12 และรูปที่ 4.13 เป็นการวัดสัญญาณของแรงดันที่ได้จากการ Trigger ของมอสเฟต ที่ความเร็วรอบ 12,000 rpm ซึ่งจะดูลักษณะการทำงานในช่วงของการ on - off ของมอสเฟตทั้ง 6 ตัว โดยมอสเฟตตัวที่ 1 จะทำงานก่อนแล้วตามด้วยมอสเฟตตัวที่ 2 , 3 , 4 , 5 และตัวที่ 6 ตามลำดับ

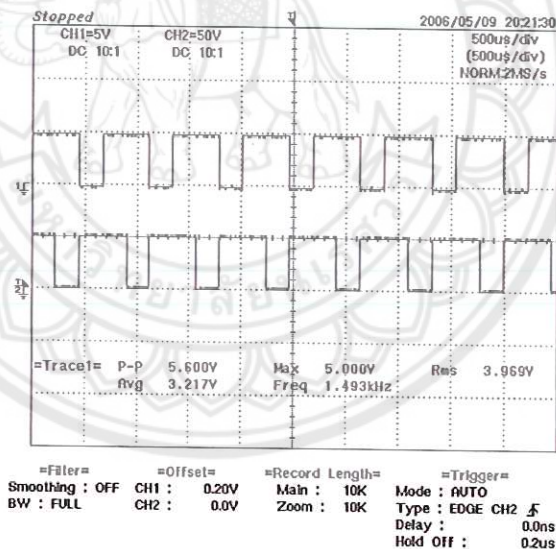
4.2.4 ที่ความเร็วรอบ 15,000 rpm



รูปที่ 4.14 มอสเฟตตัวที่ 1 และ 2



รูปที่ 4.15 มอสเฟตตัวที่ 3 และ 4

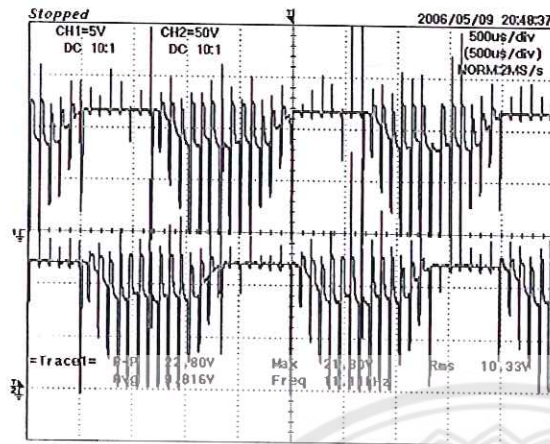


รูปที่ 4.16 มอสเฟตตัวที่ 5 และ 6

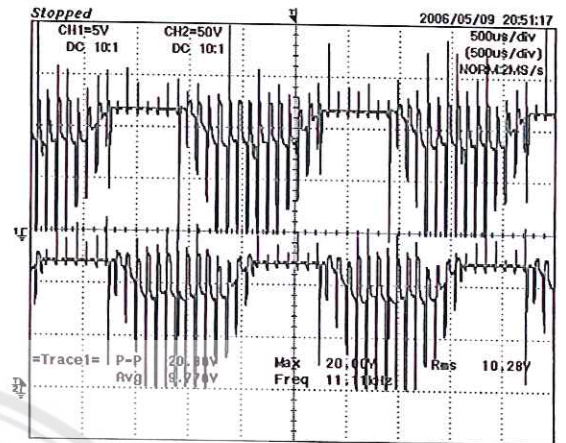
จากรูปที่ 4.14 , 4.15 และรูปที่ 4.16 เป็นการวัดสัญญาณของแรงดันที่ได้จากการ Trigger ของมอสเฟต ที่ความเร็วรอบ 15,000 rpm ซึ่งจะดูลักษณะการทำงานในช่วงของการ on - off ของมอสเฟตทั้ง 6 ตัว โดยมอสเฟตตัวที่ 1 จะทำงานก่อนแล้วตามด้วยมอสเฟตตัวที่ 2 , 3 , 4 , 5 และตัวที่ 6 ตามลำดับ

4.3 กราฟสัญญาณที่ได้จาก PHASE

4.3.1 ที่ความเร็วรอบ 5,000 rpm



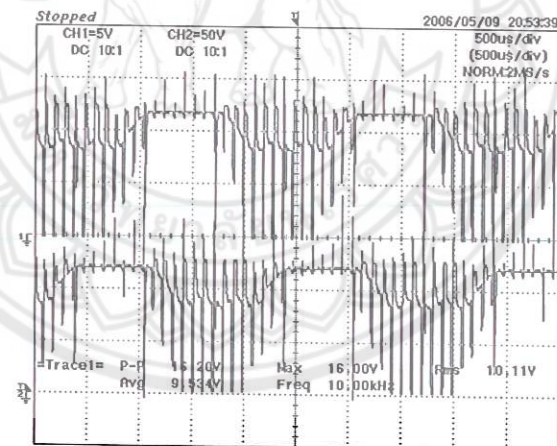
=Filter= Smoothing : OFF BW : FULL
=Offset= CH1 : 0.20V CH2 : 0.0V
=Record Length= Main : 10K Zoom : 10K
=Trigger= Mode : AUTO Type : EDGE CH2 \neq Delay : 0.0ns Hold Off : 0.2us



=Filter= Smoothing : OFF BW : FULL
=Offset= CH1 : 0.20V CH2 : 0.0V
=Record Length= Main : 10K Zoom : 10K
=Trigger= Mode : AUTO Type : EDGE CH2 \neq Delay : 0.0ns Hold Off : 0.2us

รูปที่ 4.17 เฟส AB

รูปที่ 4.18 เฟส BC

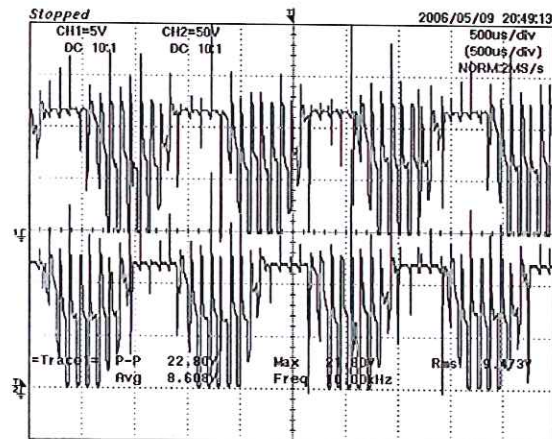


=Filter= Smoothing : OFF BW : FULL
=Offset= CH1 : 0.20V CH2 : 0.0V
=Record Length= Main : 10K Zoom : 10K
=Trigger= Mode : AUTO Type : EDGE CH2 \neq Delay : 0.0ns Hold Off : 0.2us

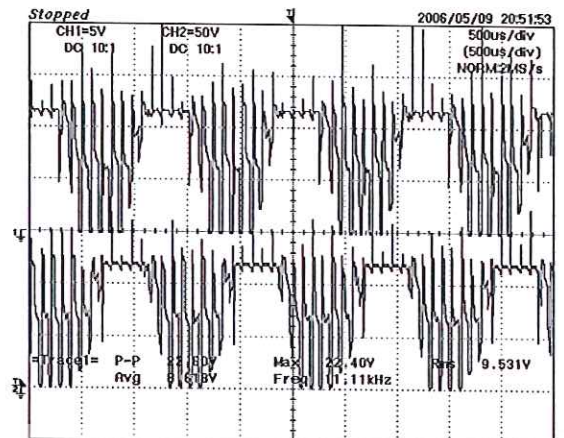
รูปที่ 4.19 เฟส CA

จากรูปที่ 4.17 , 4.18 และรูปที่ 4.19 เป็นการเปรียบเทียบกราฟสัญญาณของแรงดันที่ได้จากเฟส ที่ความเร็วรอบ 5,000 rpm ระหว่างเฟส AB , BC และ CA ซึ่งเฟส A จะนำหน้าเฟส B , เฟส B จะนำหน้าเฟส C และเฟส C นำหน้าเฟส A เป็นมุม 120 องศาตามลำดับ (ตามทฤษฎีภาคผนวก ก)

4.3.2 ที่ความเร็วรอบ 8,000 rpm



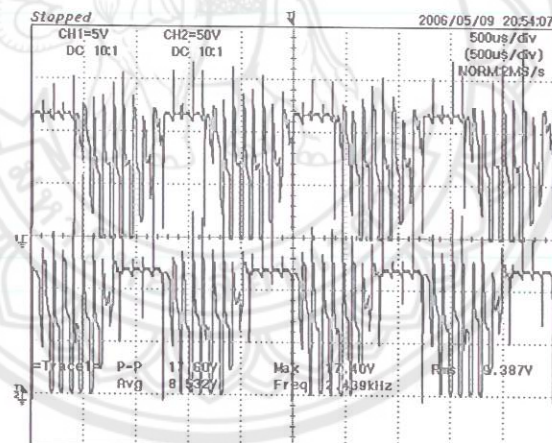
=Filter= =Offset= =Record Length= =Trigger=
Smoothing : OFF CH1 : 0.20V Main : 10K Mode : AUTO
BW : FULL CH2 : 0.0V Zoom : 10K Type : EDGE CH2 $\frac{f}{s}$
Delay : 0.0ns
Hold Off : 0.2µs



=Filter= =Offset= =Record Length= =Trigger=
Smoothing : OFF CH1 : 0.20V Main : 10K Mode : AUTO
BW : FULL CH2 : 0.0V Zoom : 10K Type : EDGE CH2 $\frac{f}{s}$
Delay : 0.0ns
Hold Off : 0.2µs

รูปที่ 4.20 เฟส AB

รูปที่ 4.21 เฟส BC

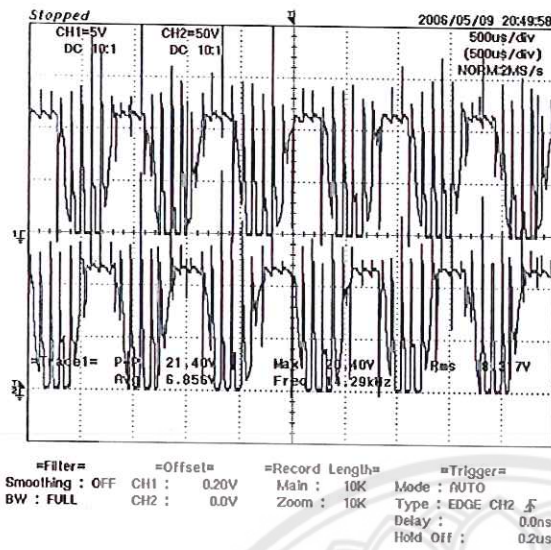


=Filter= =Offset= =Record Length= =Trigger=
Smoothing : OFF CH1 : 0.20V Main : 10K Mode : AUTO
BW : FULL CH2 : 0.0V Zoom : 10K Type : EDGE CH2 $\frac{f}{s}$
Delay : 0.0ns
Hold Off : 0.2µs

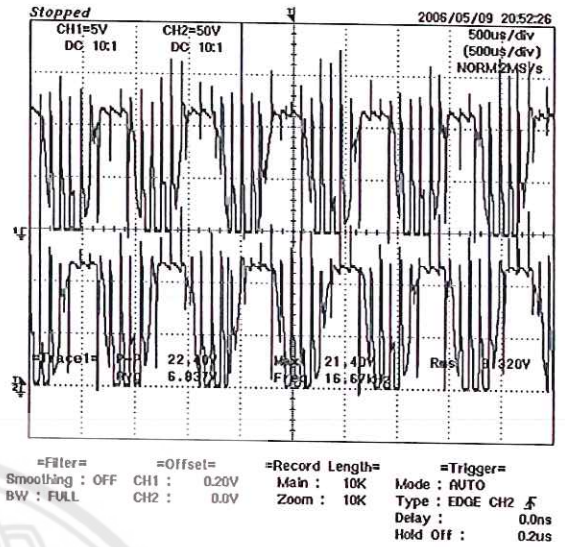
รูปที่ 4.22 เฟส CA

จากรูปที่ 4.20 , 4.21 และรูปที่ 4.22 เป็นการเปรียบเทียบกราฟสัญญาณของแรงดันที่ได้จากเฟส ที่ความเร็วรอบ 8,000 rpm ระหว่างเฟส AB , BC และ CA ซึ่งเฟส A จะนำหน้าเฟส B , เฟส B จะนำหน้าเฟส C และเฟส C นำหน้าเฟส A เป็นมุม 120 องศาตามลำดับ (ตามทฤษฎีภาคผนวก ค)

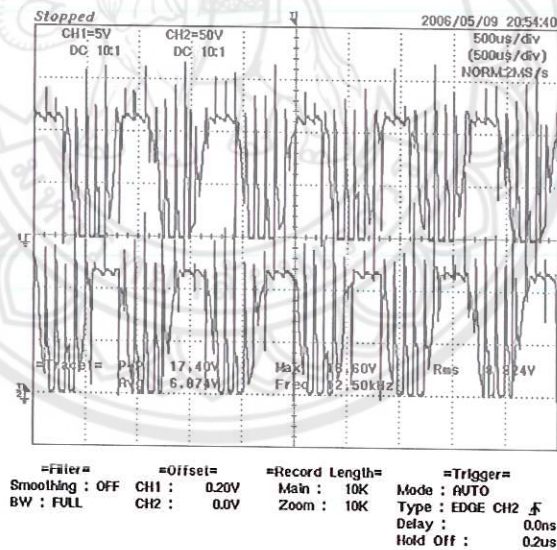
4.3.3 ที่ความเร็วรอบ 12,000 rpm



รูปที่ 4.23 เฟส AB



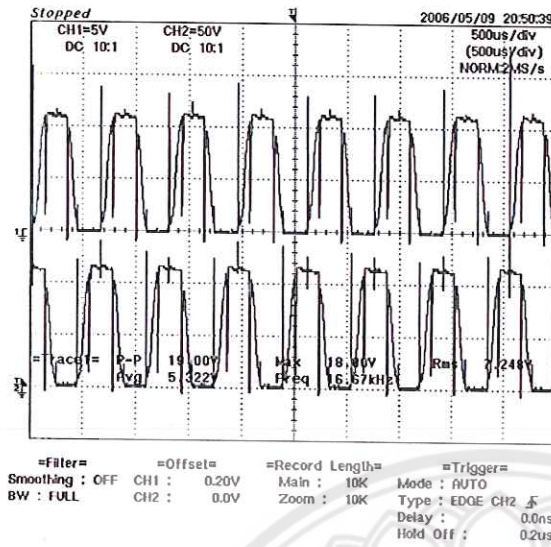
รูปที่ 4.24 เฟส BC



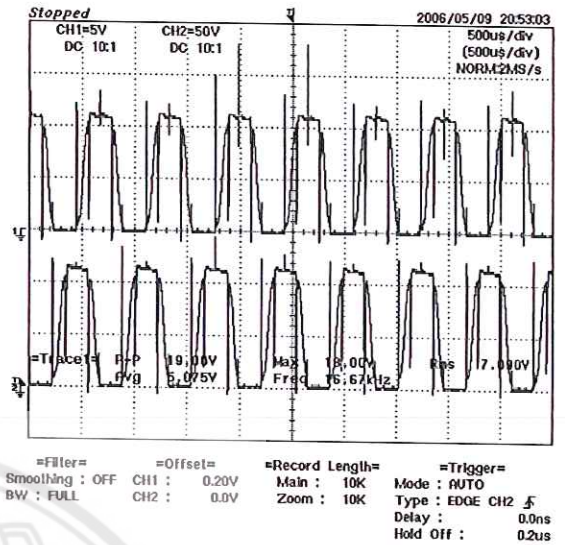
รูปที่ 4.25 เฟส CA

จากรูปที่ 4.23 , 4.24 และรูปที่ 4.25 เป็นการเปรียบเทียบกราฟสัญญาณของแรงดันที่ได้จากเฟส ที่ความเร็วรอบ 12,000 rpm ระหว่างเฟส AB , BC และ CA ซึ่งเฟส A จะนำหน้าเฟส B , เฟส B จะนำหน้าเฟส C และเฟส C นำหน้าเฟส A เป็นมุม 120 องศาตามลำดับ (ตามทฤษฎีภาคผนวก ค)

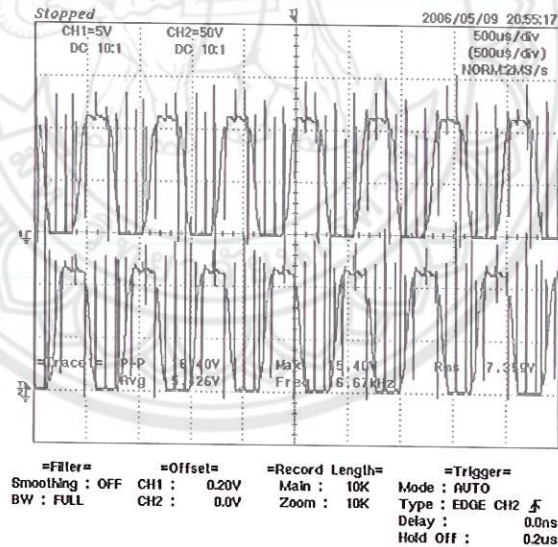
4.3.4 ที่ความเร็วรอบ 15,000 rpm



รูปที่ 4.26 เฟส AB



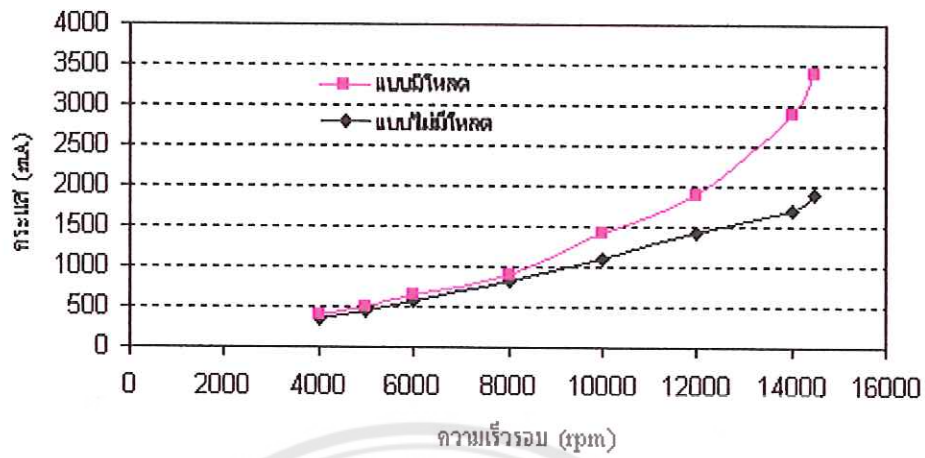
รูปที่ 4.27 เฟส BC



รูปที่ 4.28 เฟส CA

จากรูปที่ 4.26 , 4.27 และรูปที่ 4.28 เป็นการเปรียบเทียบกราฟสัญญาณของแรงดันที่ได้จากเฟส ที่ความเร็วรอบ 15,000 rpm ระหว่างเฟส AB , BC และ CA ซึ่งเฟส A จะนำหน้าเฟส B , เฟส B จะนำหน้าเฟส C และเฟส C นำหน้าเฟส A เป็นมุม 120 องศาตามลำดับ (ตามทฤษฎีภาคผนวก ค)

4.4 กราฟผลการทดลองของมอเตอร์ในขณะที่มีโหลดกับไม่มีโหลด



รูปที่ 4.29 ความเร็วรอบกับค่ากระแสที่มอเตอร์หมุนแบบมีโหลดและไม่มีโหลด

เมื่อเรานำกราฟที่ได้ดังรูปที่ 4.29 มาเปรียบเทียบกันระหว่างมอเตอร์หมุนขณะที่มีโหลดกับมอเตอร์หมุนขณะที่ไม่มีโหลด จะเห็นได้ว่าเมื่อเราปรับค่าความเร็วรอบของมอเตอร์ให้สูงขึ้นค่าของความเร็วรอบของมอเตอร์ขณะมีโหลดจะมีความชันมากกว่ามอเตอร์ขณะที่ไม่มีโหลด ดังนั้นเรากล่าวได้ว่ามอเตอร์ขณะที่มีโหลดจะกินกระแสสูงกว่ามอเตอร์ขณะที่ไม่มีโหลด

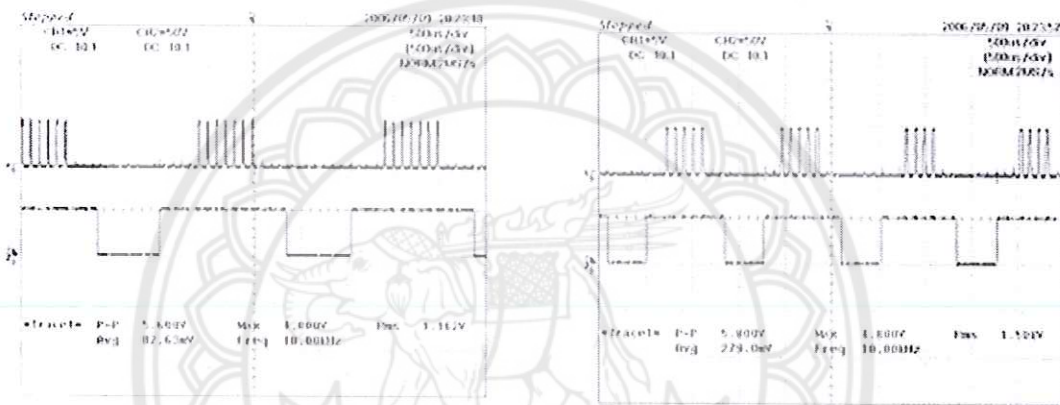
บทที่ 5

บทสรุป

จากการทดลอง มอเตอร์ไฟฟ้ากระแสตรงสามารถควบคุมความเร็วรอบได้ 2 วิธี คือการปรับค่าความเร็วรอบด้วยมือ และการปรับค่าความเร็วรอบอัตโนมัติ

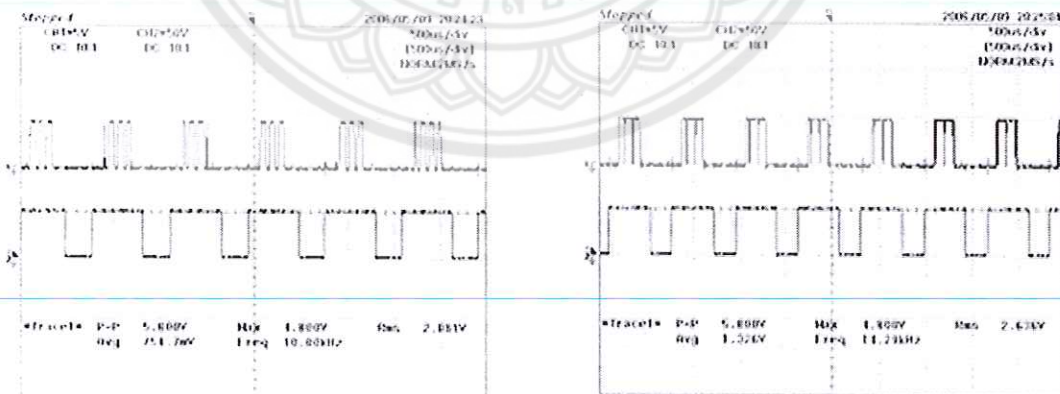
5.1 สรุปผลการทดลอง

5.1.1 ยกตัวอย่าง สัญญาณที่ได้จาก Gate ตัวที่ 1 กับ Gate ตัวที่ 4



รูปที่ 5.1 ความเร็วรอบ 5,000 rpm

รูปที่ 5.2 ความเร็วรอบ 8,000 rpm



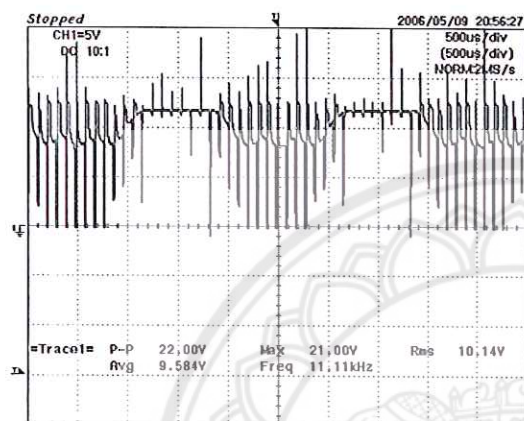
รูปที่ 5.3 ความเร็วรอบ 12,000 rpm

รูปที่ 5.4 ความเร็วรอบ 15,000 rpm

จากรูปที่ 5.1 , 5.2 , 5.3 และ 5.4 เป็นการวัดสัญญาณของแรงดันที่ได้จาก Gate ตัวที่ 1 กับ 4 ที่ความเร็วรอบต่างๆ ในช่วงเวลาของการ on – off ของสัญญาณพัลส์แรงดัน

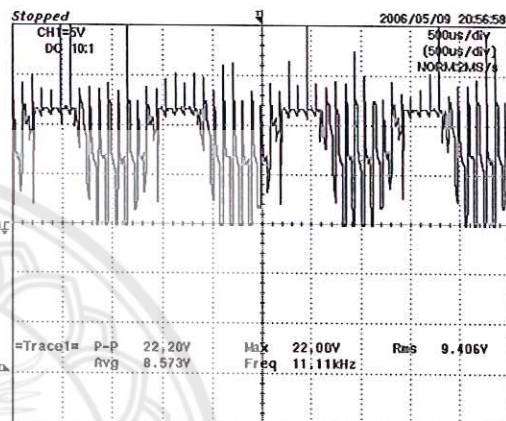
กราฟสัญญาณของแรงดันที่ได้จากการวัดการทรักของมอเตอร์ทั้ง 6 ตัว ที่หนึ่งคาบเวลา (T) ที่ความเร็วรอบต่างๆ ของมอเตอร์ เราสามารถสรุปได้ว่าสัญญาณการทรักของความเร็วรอบ 5,000 รอบ จะมีช่วงเวลาของการ on-off นานกว่า 8,000 รอบ, 12,000 รอบ และ 15,000 รอบ (ตาม-ทฤษฎีภาคผนวก ก)

5.1.2 เฟส A ที่ความเร็วรอบต่างๆ



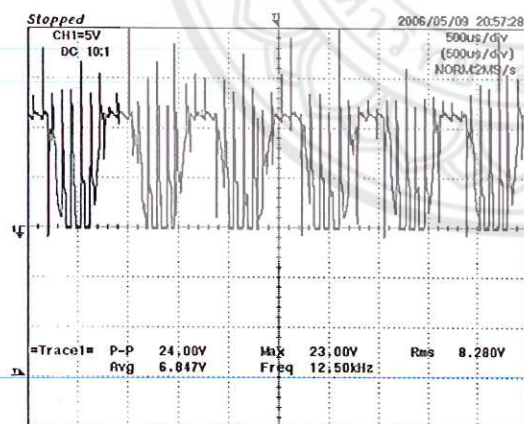
=Filter= Smoothing : OFF BW : FULL
 =Offset= CH1 : 0.20V CH2 : 0.0V
 =Record Length= Main : 10K Zoom : 10K
 =Trigger= Mode : AUTO Type : EDGE CH2 Delay : 0.0ns Hold Off : 0.2us

รูปที่ 5.5 ความเร็วรอบ 5,000 rpm



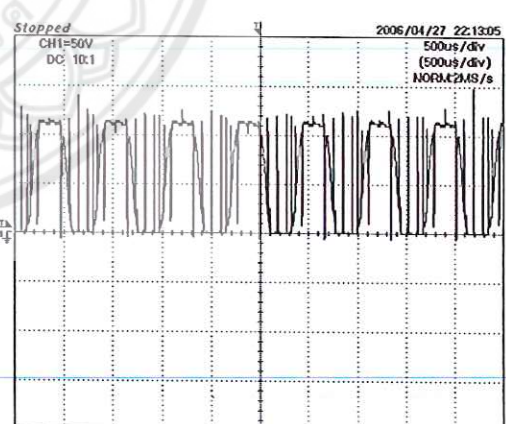
=Filter= Smoothing : OFF BW : FULL
 =Offset= CH1 : 0.20V CH2 : 0.0V
 =Record Length= Main : 10K Zoom : 10K
 =Trigger= Mode : AUTO Type : EDGE CH2 Delay : 0.0ns Hold Off : 0.2us

รูปที่ 5.6 ความเร็วรอบ 8,000 rpm



=Filter= Smoothing : OFF BW : FULL
 =Offset= CH1 : 0.20V CH2 : 0.0V
 =Record Length= Main : 10K Zoom : 10K
 =Trigger= Mode : AUTO Type : EDGE CH2 Delay : 0.0ns Hold Off : 0.2us

รูปที่ 5.7 ความเร็วรอบ 12,000 rpm



=Filter= Smoothing : OFF BW : FULL
 =Offset= CH1 : 0.0V CH2 : 0.00V
 =Record Length= Main : 10K Zoom : 100
 =Trigger= Mode : AUTO Type : EDGE CH1 Delay : 0.0ns Hold Off : MINIMUM

รูปที่ 5.8 ความเร็วรอบ 15,000 rpm

จากรูปที่ 5.5 , 5.6 , 5.7 และ 5.8 เป็นการนำสัญญาณของแรงดันที่เฟส A ที่ความเร็วรอบต่างๆ มาเปรียบเทียบกับกัน เพื่อดูสัญญาณแรงดันที่ได้ ที่ความเร็วรอบ 5,000 , 8,000 , 12,000 และ

15,000 รอบ เราจะเห็นได้ว่าค่าของความเร็วรอบที่ 5,000 rpm จะมีช่วงเวลาในการ on-off นานกว่า 8,000 รอบ, 12,000 รอบ และ 15,000 รอบ เช่นเดียวกับการ Trigger ของมอเตอร์ (ตามทฤษฎีภาคผนวก ก)

เมื่อทำการเปรียบเทียบกันระหว่างมอเตอร์หมุนขณะที่มีโหลดกับมอเตอร์หมุนขณะที่ไม่มีโหลด และหาค่าของความชัน จะเห็นได้ว่าเมื่อเราปรับค่าความเร็วรอบของมอเตอร์ให้สูงขึ้น ค่าของความถี่รอบของมอเตอร์ขณะที่มีโหลดจะมีความชันมากกว่ามอเตอร์ขณะที่ไม่มีโหลด ดังนั้นเรากล่าวได้ว่ามอเตอร์ขณะที่มีโหลดจะกินกระแสสูงกว่ามอเตอร์ที่ไม่มีโหลด

การหมุนของมอเตอร์แบบใช้มือปรับความเร็วรอบกับแบบอัตโนมัติอย่างไรจะดีกว่ากัน ซึ่งจะอยู่ที่การนำไปใช้งาน ถ้าเรานำไปใช้งานกับรถไฟฟ้า การหมุนของมอเตอร์แบบใช้มือปรับความเร็วรอบจะมีความละเอียดกว่าแบบอัตโนมัติ เนื่องจากการแบบอัตโนมัติจะมีความคลาดเคลื่อนของการหมุนของมอเตอร์ได้ และแบบปรับความเร็วรอบด้วยมือสามารถกำหนดความเร็วรอบไว้ได้คือ จะต้องมีการเร่งและผ่อนของมอเตอร์ แต่ถ้าการหมุนของมอเตอร์แบบอัตโนมัติมีความละเอียดกว่าแบบปรับด้วยมือ จะนิยมใช้กันในโรงงานที่ต้องการกำหนดความเร็วรอบของมอเตอร์ได้ เช่น เครื่องเจาะ CNC ในโรงงาน

5.2 ประเมินผลการทดลอง

จากการดำเนินงาน โครงการ เมื่อเปรียบเทียบกับวัตถุประสงค์แล้วได้ผลดังนี้

- 5.2.1 สามารถออกแบบวงจรควบคุมการทำงานของมอเตอร์และควบคุมการทำงานของสวิทช์อิเล็กทรอนิกส์ เพื่อให้มอเตอร์ทำงานตามที่ต้องการได้
- 5.2.2 สามารถนำไปประยุกต์กับการควบคุมมอเตอร์ที่มีขนาดใหญ่ได้
- 5.2.3 สามารถใช้โปรแกรม MicroCode Studio ในการจำลองการทำงานของวงจรขับเคลื่อนมอเตอร์ได้

5.3 ปัญหา ข้อเสนอแนะ และแนวทางแก้ไข

5.3.1 ปัญหาเกิดจากการต่อวงจรผิดพลาด ทำให้อุปกรณ์ได้รับความเสียหาย จึงต้องทำการจัดซื้ออุปกรณ์มาใหม่ ทำให้งานที่ได้ออกมาช้ากว่ากำหนด

5.3.2 ปัญหาเกิดจากการจ่ายแรงดันไฟฟ้าให้กับอุปกรณ์อิเล็กทรอนิกส์ ถ้าจ่ายแรงดันไม่เหมาะสมสัญญาณต่างๆ จะไม่ได้ตามที่ต้องการ จะต้องจ่ายแรงดันไฟฟ้าที่เหมาะสมให้กับอุปกรณ์อิเล็กทรอนิกส์ถึงจะทำงานคงที่ได้

5.4 แนวทางในการพัฒนาต่อไป

ผลที่ได้จากการทดลองในโครงการนี้ สามารถนำไปประยุกต์ใช้กับงานที่ใช้มอเตอร์ไฟฟ้า กระแสตรงที่มีลักษณะการทำงานคล้ายๆ กัน เช่น เครื่องบินบังคับ , ใช้ในการ Test ของมอเตอร์ใน โรงงานทำมอเตอร์ , มอเตอร์แอร์ และเหมาะที่จะนำไปพัฒนาต่อให้สามารถใช้ในยานพาหนะ เช่น รถไฟฟ้าขนาดใหญ่ต่อไปได้

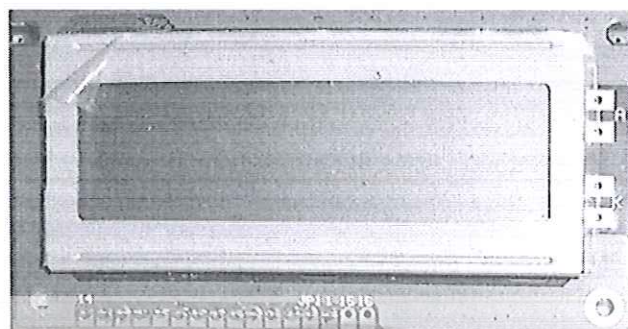


เอกสารอ้างอิง

- [1] วิชัย ศังขจันทรานนท์. ควบคุมเครื่องกลไฟฟ้าด้วยอิเล็กทรอนิกส์กำลัง 1. พิมพ์ครั้งที่ 5.
กรุงเทพมหานคร : สำนักพิมพ์สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น). 2537
- [2] ผศ.โยธิน เปรมปราณีรัชต์. วิเคราะห์และออกแบบระบบควบคุมมอเตอร์. กรุงเทพมหานคร :
สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตเจ้าคุณทหารลาดกระบัง. 2526
- [3] มงคล ทองสงคราม. อิเล็กทรอนิกส์กำลัง. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : พรศิวิการพิมพ์.
2536
- [4] รศ.ดร. วีระเชษฐ ชันเงิน. อิเล็กทรอนิกส์กำลัง. กรุงเทพมหานคร : ห้างหุ้นส่วนจำกัด วี.เจ.
พรินติ้ง. 2547
- [5] นภัทร วัจนเทพินทร์. อิเล็กทรอนิกส์กำลัง 2. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : สกายบุ๊กส์.
2544
- [6] กฤษดา ใจเย็น และคณะ. เรียนรู้และใช้งาน PICBASIC PRO คอมไพเลอร์.
กรุงเทพมหานคร : อินโนเวตีฟ เอ็กเพอริเมนต์ กรุงเทพ. 2521
- [7] Muhammad H. Rashid. **Power Electronic Circuits, Devices and Application**. 2nd
Edition, Prentice – Hall. Inc. 1993
- [8] Peter. “Speedy BL.” [Online]. Available : [Http://www.aerode.sign.de/LRK350/index.htm](http://www.aerode.sign.de/LRK350/index.htm).
1998
- [9] En. App Notes. “Speeds Control.” [Online]. Available : [Http://www.microchip.com
/downloads/00857a/index.pdf](http://www.microchip.com/downloads/00857a/index.pdf). 1994



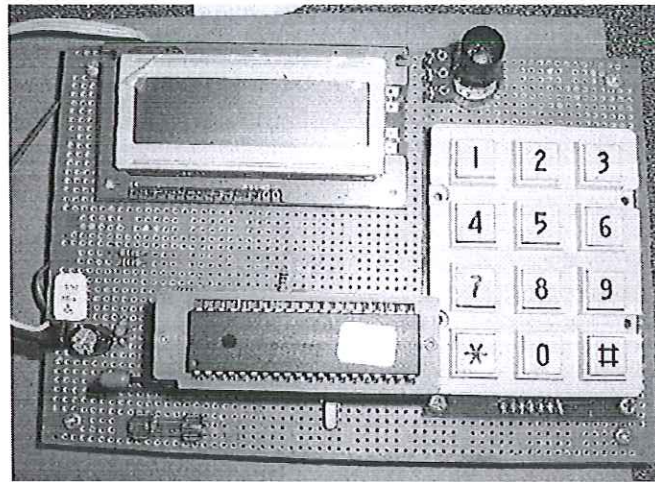
ภาคผนวก ก



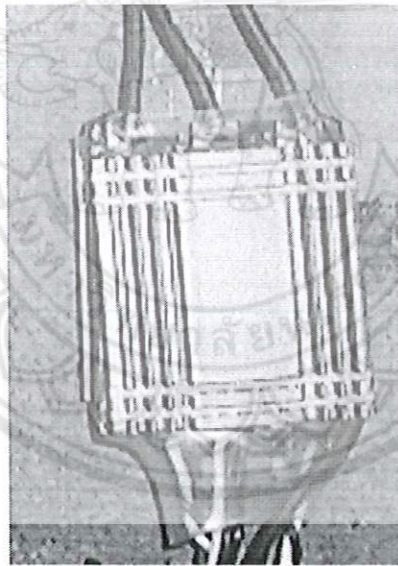
รูปที่ ก.1 จอแสดงผล แอล ซี ดี



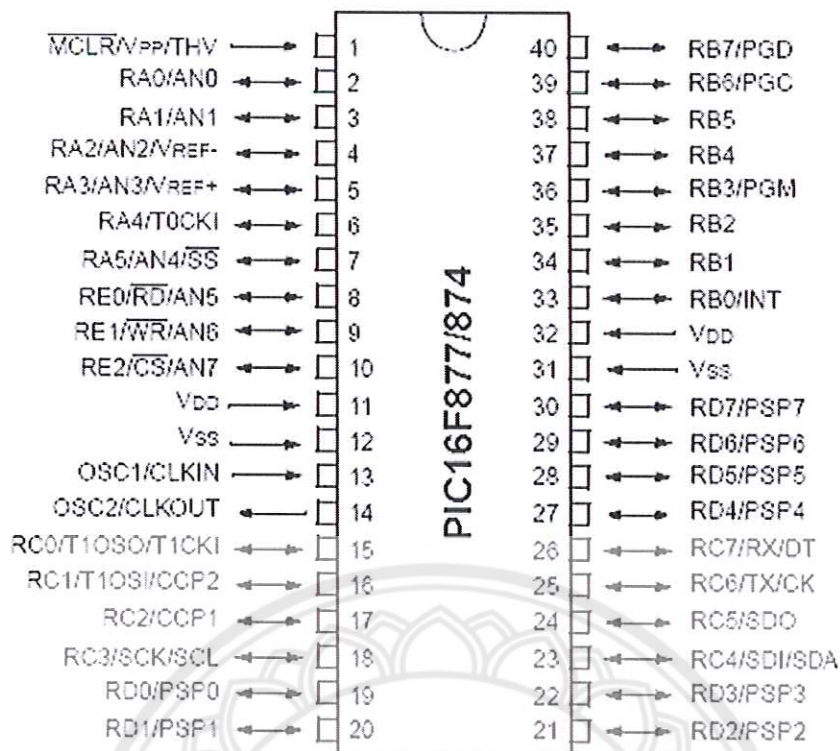
รูปที่ ก.2 คีย์บอร์ดหรือคีย์แพด



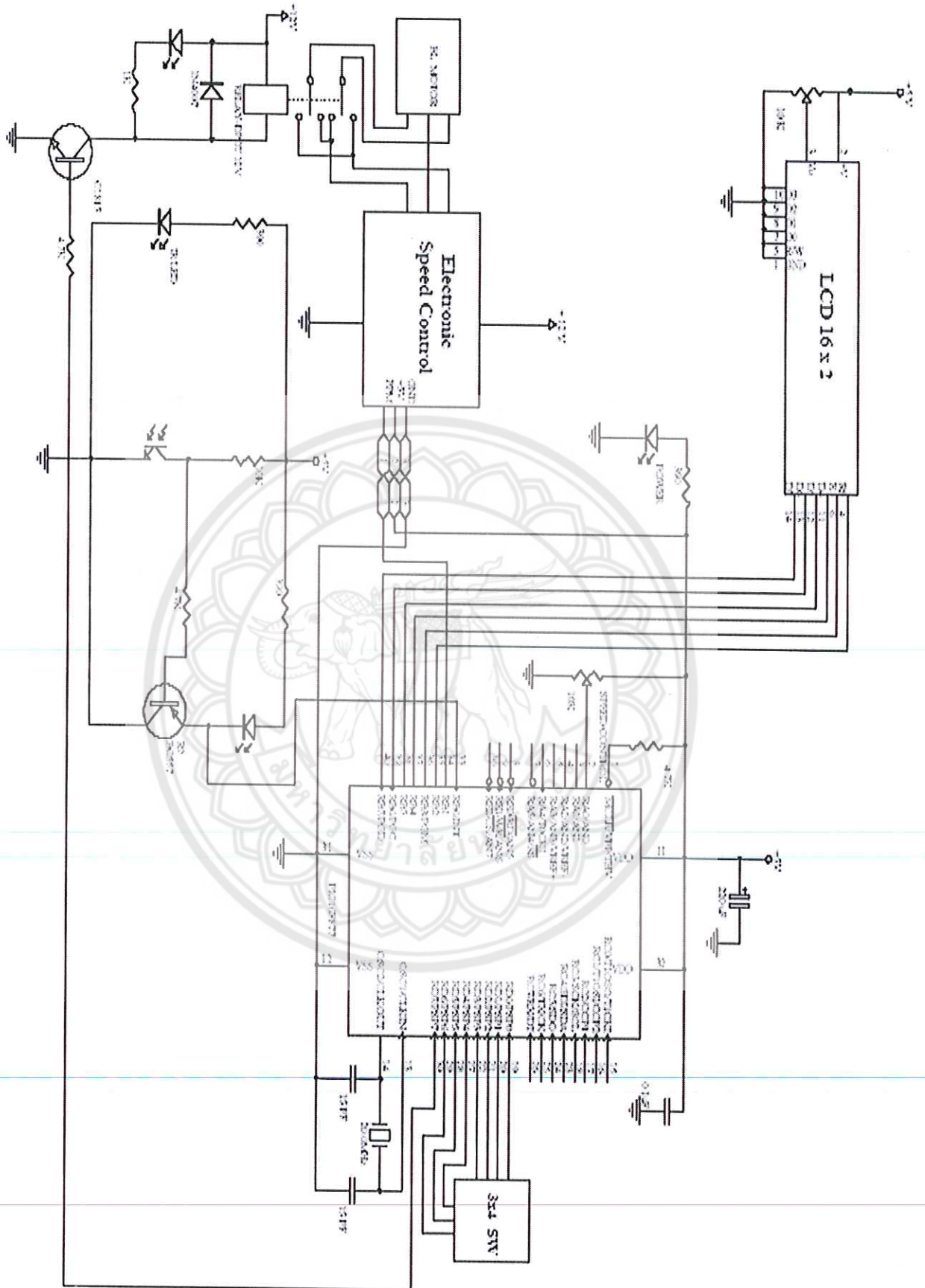
รูปที่ ก.3 วงจรขับมอเตอร์และไมโครคอนโทรลเลอร์ PIC16F877



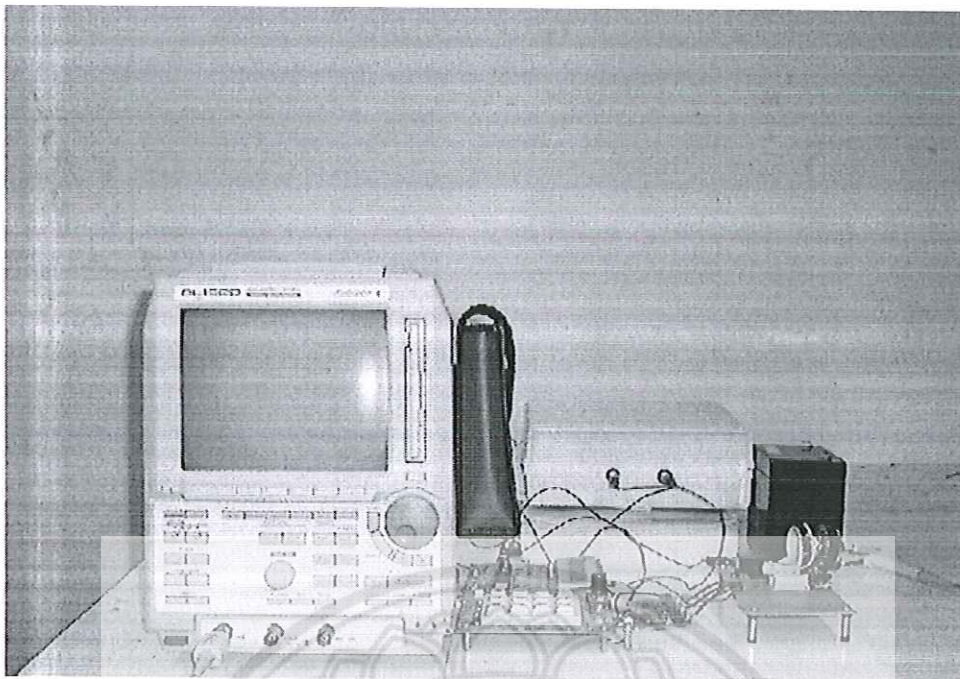
รูปที่ ก.4 วงจรควบคุมมอเตอร์



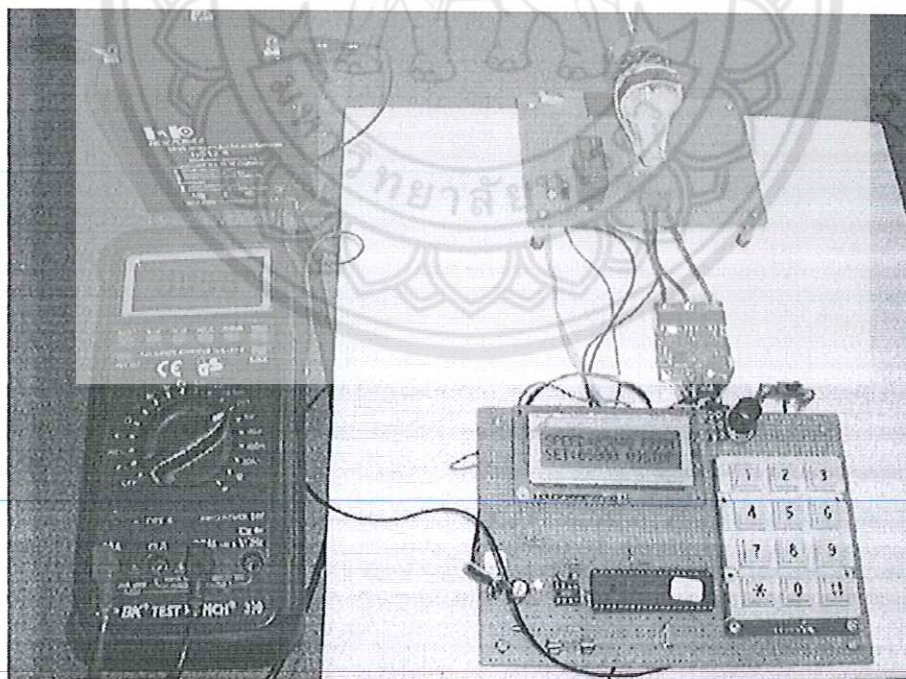
รูปที่ ก.5 ตำแหน่งขาของไมโครคอนโทรลเลอร์ PIC16F877



รูปที่ ก.6 วงจรรวมทั้งหมด



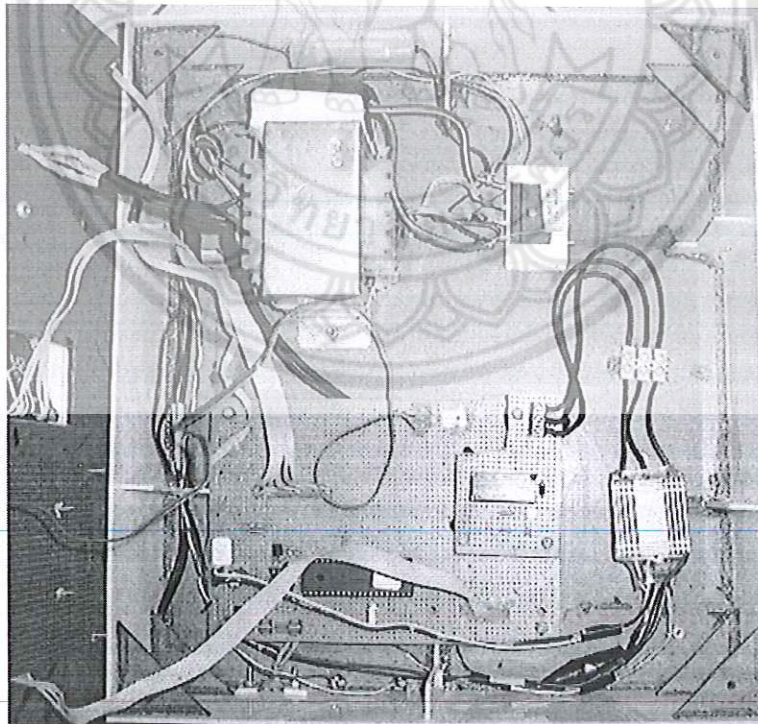
รูปที่ ก. 7 การต่อวงจรเพื่อทำการทดลอง



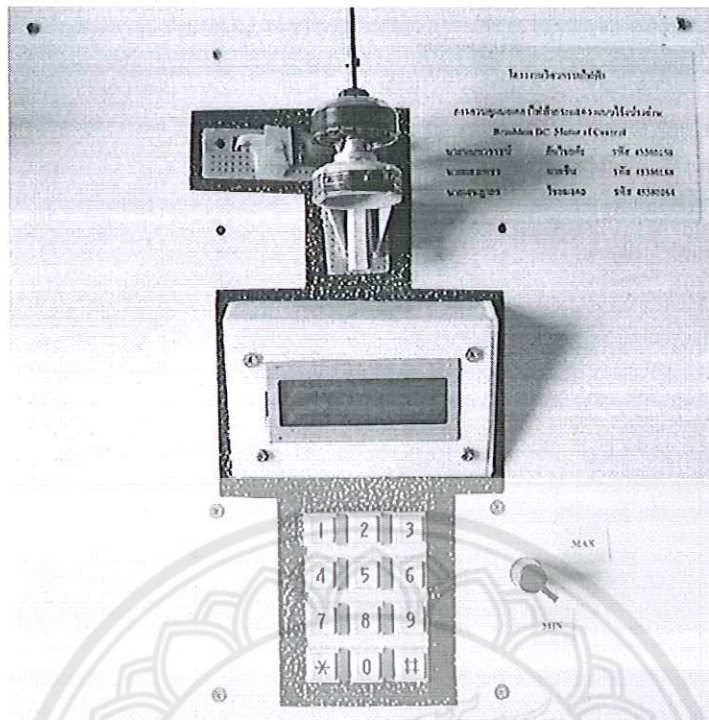
รูปที่ ก. 8 การต่อวงจรและทำการวัดกระแส



รูปที่ ก.9 การต่อวงจรขณะทำการทดลอง



รูปที่ ก.10 การต่อวงจรลงอุปกรณ์เรียบร้อยแล้ว



รูปที่ ก. 11 ด้านหน้าของกล่องอุปกรณ์ชิ้นงาน



ภาคผนวก ข

โปรแกรมการควบคุมการทำงานของมอเตอร์

```

*****
'* Name : UNTITLED.BAS *
'* Author : [set under view...options] *
'* Notice : Copyright (c) 2006 [set under view...options] *
'* : All Rights Reserved *
'* Date : 10/3/2006 *
'* Version : 1.0 *
'* Notes : *
'* : *
*****
@ DEVICE PIC16F877,HS_OSC,WDT_OFF
define osc 20

DEFINE ADC_BITS 10
DEFINE ADC_CLOCK 3
DEFINE ADC_SAMPLEUS 500

define LCD_BITS 4
define LCD_LINES 2
define LCD_DREG PORTB
define LCD_DBIT 4
define LCD_RSREG PORTB
define LCD_RSBIT 2
define LCD_EREG PORTB
define LCD_EBIT 3
Define LCD_COMMANDUS 10000
Define LCD_DATAUS 250
Define LCD_INITMS 3000

```



```
define PULSIN_MAX 40000
```

```
TARGET var word
```

```
RESULT var word
```

```
ESC var PORTB.1
```

```
TRIG VAR PORTB.0
```

```
ADCON1 = %00000000
```

```
TRISA = %11111111
```

```
TRISE = %00000111
```

```
TRISD = %11110000
```

```
ADCON1.7 = 1
```

```
input trig
```

```
LOW ESC
```

```
Row1 var PORTD.0
```

```
Row2 var PORTD.1
```

```
Row3 var PORTD.2
```

```
Row4 var PORTD.3
```

```
Col1 var PORTD.4
```

```
Col2 var PORTD.5
```

```
Col3 var PORTD.6
```

```
DIRECTION var PORTD.7
```

```
il var byte
```

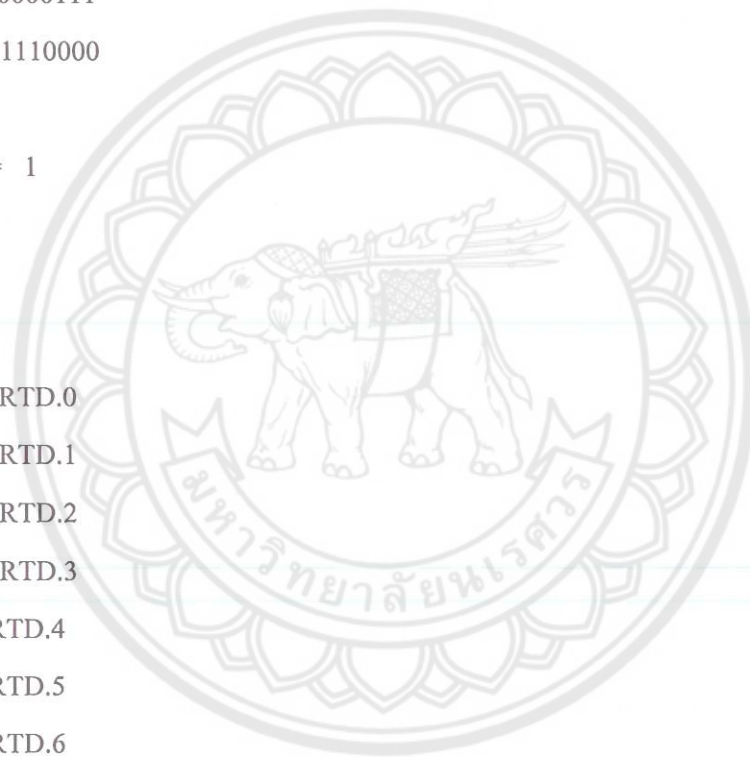
```
KEY var byte
```

```
KEY_PRESS var bit
```

```
RPM var word
```

```
TH var word
```

```
TL var word
```



PULSE_WIDTH var word

FREQ var word

TARGET_RPM var word

AUTO_RPM_MIN var word

AUTO_RPM_MAX var word

AUTO_RPM_CUR var word

MEM1 var word

ERROR_COUNT var Byte

DIG_COUNT var byte

PAUSE 1000

AUTO_RPM_MIN=500

AUTO_RPM_MAX=1000

AUTO_RPM_CUR=AUTO_RPM_MIN

LCDOUT \$FE,1

LCDOUT \$FE,%00001100

LCDOUT \$FE,\$80,"BRUSHLESS MOTOR"

LCDOUT \$FE,\$C0," SPEED CONTROL "

for i1 = 0 to 100

PULSOUT ESC,400

pause 20

next

AUTO_MENU:

SETUP_MENU:

LCDOUT \$FE,\$1

LCDOUT \$FE,\$80,"1 TO NORMAL MODE"

```
LCDOUT $FE,$C0,"2 TO AUTO MODE "
```

```
MENU_LOOP1:
```

```
call ReadKey
```

```
if KEY=1 then
```

```
goto NORMAL_MODE
```

```
endif
```

```
if KEY=2 then
```

```
goto AUTO_MODE
```

```
endif
```

```
PULSOUT ESC,400
```

```
pause 20
```

```
goto MENU_LOOP1
```

```
ENTER_VALUE:
```

```
TARGET_RPM=0
```

```
DIG_COUNT=0
```

```
rpm=0
```

```
NORMAL_MODE:
```

```
LCDOUT $FE,$80," NORMAL MODE "
```

```
LCDOUT $FE,$C0," "
```

```
'pause 1000
```

```
PULSOUT ESC,400
```

```
pause 20
```

```
LCDOUT $FE,$C0,"*]FW #]RW 0]EXIT"
```

NORMAL_MODE2:

call ReadKey

if rpm > 0 then

goto NORMAL_MODE3

endif

if key = 10 then '*'

low DIRECTION

LCDOUT \$FE,\$80,"Select : FORWARD"

endif

if key = 11 then '#'

high DIRECTION

LCDOUT \$FE,\$80,"Select : REWARD "

endif

NORMAL_MODE3:

'endif

if key = 0 then '0'

for i1=1 to 100

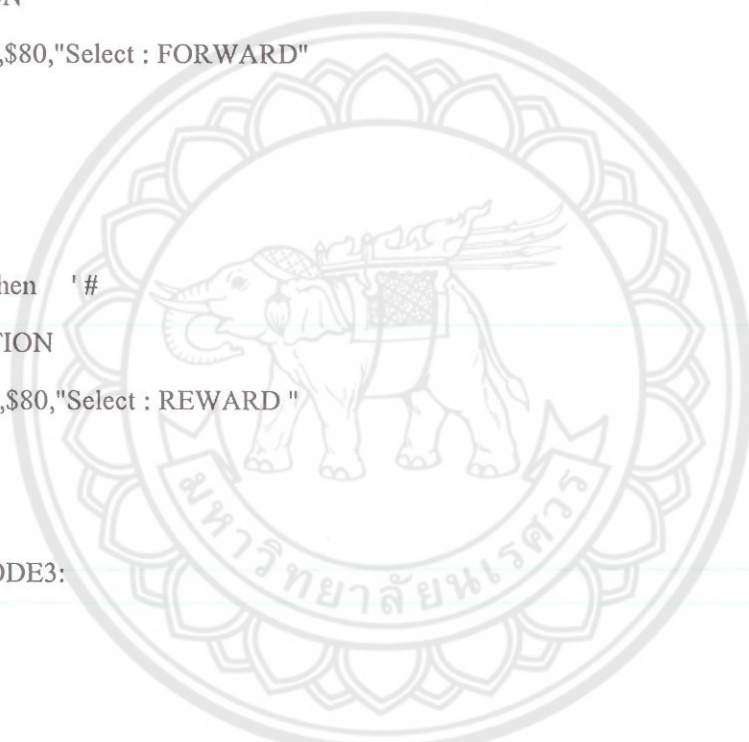
PULSOUT ESC,AUTO_RPM_CUR

pause 20

next

goto SETUP_MENU

endif



```
ADCIN 0,RESULT
result=result+350
if result < 400 then
Result=400
endif
if result > 1000 then
Result=1000
endif

PULSOUT ESC,result
pause 20

pulsin trig,1,th
pulsin trig,0,tl
PULSE_WIDTH=th+tl
PULSE_WIDTH=PULSE_WIDTH+PULSE_WIDTH
RPM = 1000
RPM = RPM * RPM      ' Preload internal registers with 1,000,000
RPM = DIV32 PULSE_WIDTH ' 1,000,000 / Pulse_Total

RPM = RPM * 60      ' Per minute

if RPM > 60000 then
rpm=0
endif

LCDOUT $FE,$80,"SPEED= ",dec5 rpm ," RPM"

goto NORMAL_MODE2
```

AUTO_MODE:

```
LCDOUT $FE,$80," AUTO MODE "
```

```
LCDOUT $FE,$C0," SETUP "
```

```
pause 1000
```

```
LCDOUT $FE,$80,"Select Direction"
```

```
LCDOUT $FE,$C0," 1]FW 2]RW #]ENT"
```

SET_DIRECTION:

```
call ReadKey
```

```
if KEY=1 then
```

```
low direction
```

```
LCDOUT $FE,$80,"Select : FORWARD"
```

```
endif
```

```
if KEY=2 then
```

```
high direction
```

```
LCDOUT $FE,$80,"Select : REWARD "
```

```
endif
```

```
if KEY=11 then
```

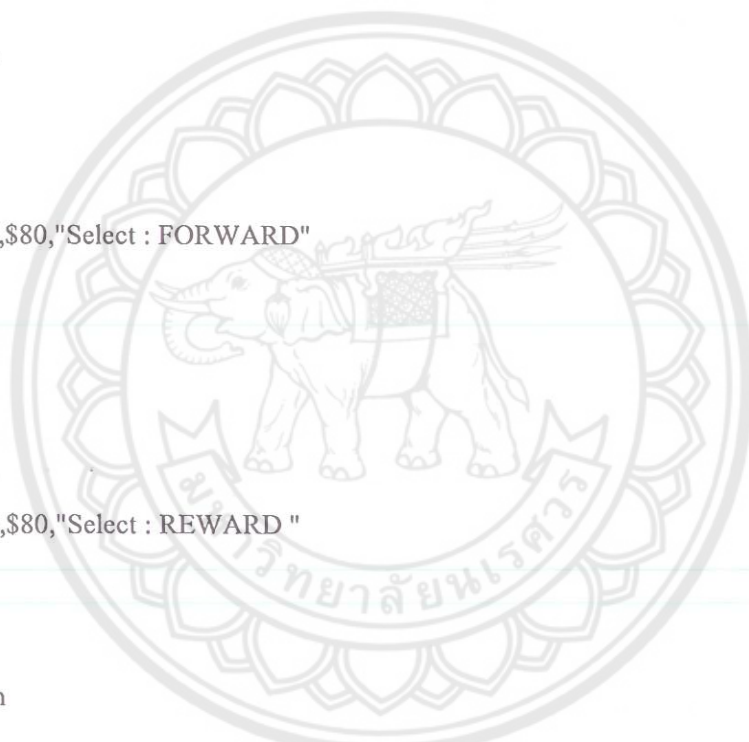
```
goto SET_SPEED
```

```
endif
```

```
PULSOUT ESC,400
```

```
pause 20
```

```
goto SET_DIRECTION
```



SET_SPEED:

LCDOUT \$FE,\$80,"Speed= RPM"

LCDOUT \$FE,\$C0,"*]Clear #]ENT "

pause 1000

SET_SPEED2:

call ReadKey

target=0

LCDOUT \$FE,\$80,"Speed= ",dec5 target," RPM"

RE_ENTER:

LCDOUT \$FE,\$80,"Speed= RPM"

LCDOUT \$FE,\$C0,"*]Clear #]ENT "

KEY_DG1:

PULSOUT ESC,400

pause 20

call ReadKey

if KEY_PRESS=0 then

goto KEY_DG1

endif

if KEY=11 then

goto START_AUTO

endif



```
if KEY=10 then
goto RE_ENTER
endif
```

```
TARGET=key
LCDOUT $FE,$80,"Speed= ",dec5 target," RPM"
pause 200
```

```
KEY_DG2:
PULSOUT ESC,400
pause 20
call ReadKey
```

```
if KEY_PRESS=0 then
goto KEY_DG2
endif
```

```
if KEY=11 then
goto START_AUTO
endif
```

```
if KEY=10 then
goto RE_ENTER
endif
```

```
target=target*10
target=target+key
LCDOUT $FE,$80,"Speed= ",dec5 target," RPM"
pause 200
```




```
KEY_DG3:
PULSOUT ESC,400
pause 20
call ReadKey

if KEY_PRESS=0 then
goto KEY_DG3
endif

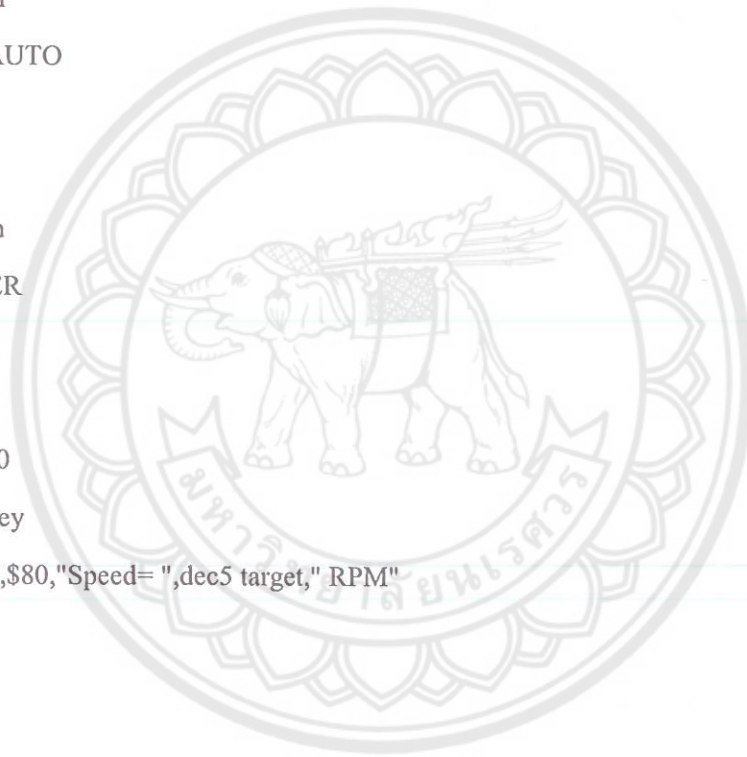
if KEY=11 then
goto START_AUTO
endif

if KEY=10 then
goto RE_ENTER
endif

target=target*10
target=target+key
LCDOUT $FE,$80,"Speed= ",dec5 target," RPM"
pause 200

KEY_DG4:
PULSOUT ESC,400
pause 20
call ReadKey

if KEY_PRESS=0 then
goto KEY_DG4
endif
```



```
if KEY=11 then
```

```
goto START_AUTO
```

```
endif
```

```
if KEY=10 then
```

```
goto RE_ENTER
```

```
endif
```

```
target=target*10
```

```
target=target+key
```

```
LCDOUT $FE,$80,"Speed= ",dec5 target," RPM"
```

```
pause 200
```

```
KEY_DG5:
```

```
PULSOUT ESC,400
```

```
pause 20
```

```
call ReadKey
```

```
if KEY_PRESS=0 then
```

```
goto KEY_DG5
```

```
endif
```

```
if KEY=11 then
```

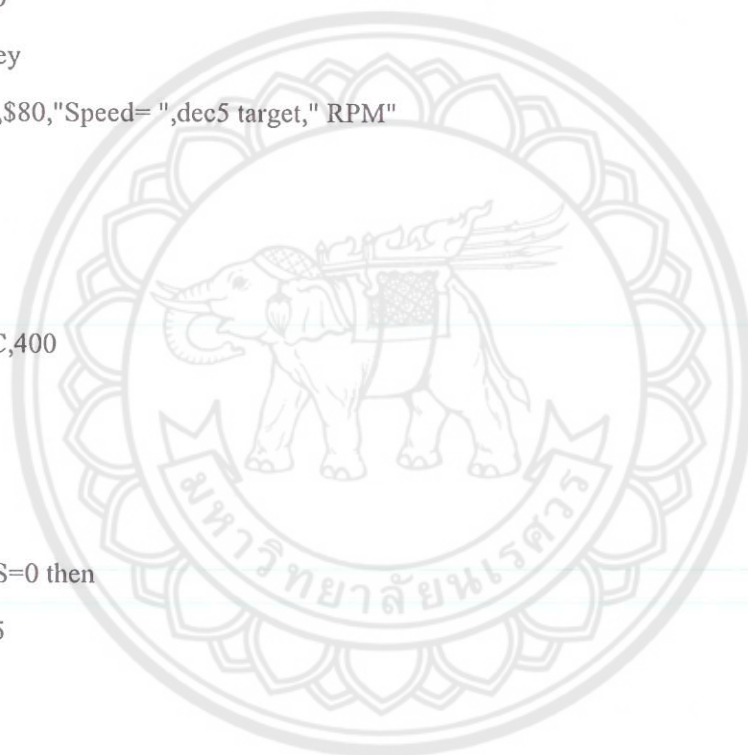
```
goto START_AUTO
```

```
endif
```

```
if KEY=10 then
```

```
goto RE_ENTER
```

```
endif
```



```
target=target*10
target=target+key
LCDOUT $FE,$80,"Speed= ",dec5 target," RPM"
PULSOUT ESC,400
pause 20
pause 200
```

```
PULSOUT ESC,400
pause 20
```

```
START_AUTO:
```

```
if target > 15000 then
LCDOUT $FE,$80,"Can't spin at"
LCDOUT $FE,$C0,"Speed > 15000RPM"
pause 1000
goto RE_ENTER
endif
```

```
if target < 4000 then
LCDOUT $FE,$80,"Can't spin at"
LCDOUT $FE,$C0,"Speed < 4000RPM "
pause 1000
goto RE_ENTER
endif
```

```
LCDOUT $FE,$80,"Speed= ",dec5 target," RPM"
LCDOUT $FE,$C0,"*]Start #]EXIT "
```

```
pause 1000
START_AUTO2:
PULSOUT ESC,400
pause 20
call ReadKey
if KEY=11 then
goto SETUP_MENU
endif

if KEY=10 then
goto START_AUTO_MODE
endif

goto START_AUTO2

START_AUTO_MODE:
START_PROGRAM:

TARGET_RPM=target

LCDOUT $FE,$C0,"SET:",dec5 TARGET_RPM," #]STOP"
```

Main:

```
PULSOUT ESC,AUTO_RPM_CUR
pause 50
pulsin trig,1,th
pulsin trig,0,tl
PULSE_WIDTH=th+tl
PULSE_WIDTH=PULSE_WIDTH+PULSE_WIDTH
RPM = 1000
```

```
RPM = RPM * RPM      ' Preload internal registers with 1,000,000
```

```
RPM = DIV32 PULSE_WIDTH ' 1,000,000 / Pulse_Total
```

```
RPM = RPM * 60      ' Per minute
```

```
if RPM > 60000 then
```

```
rpm=0
```

```
endif
```

```
if RPM < TARGET_RPM then
```

```
mem1 = TARGET_RPM - rpm
```

```
if mem1 > 2000 then
```

```
AUTO_RPM_CUR=AUTO_RPM_CUR+5
```

```
endif
```

```
if mem1 > 1000 then
```

```
AUTO_RPM_CUR=AUTO_RPM_CUR+3
```

```
endif
```

```
'if mem1 > 500 then
```

```
'AUTO_RPM_CUR=AUTO_RPM_CUR+2
```

```
'endif
```

```
if mem1 > 100 then
```

```
AUTO_RPM_CUR=AUTO_RPM_CUR+1
```

```
endif
```

```
endif
```

```
if RPM > TARGET_RPM then
```

```
mem1 = rpm-TARGET_RPM
```

```
if mem1 > 100 then
```

```
AUTO_RPM_CUR=AUTO_RPM_CUR-1
endif
endif

LCDOUT $FE,$80,"SPEED=",dec5 rpm ," RPM"
LCDOUT $FE,$C0,"SET:",dec5 TARGET_RPM," 0]STOP"
call ReadKey

if KEY=0 then
PULSOUT ESC,AUTO_RPM_CUR
goto MOTOR_PAUSE
endif

goto main

MOTOR_PAUSE:
LCDOUT $FE,$80,">>>>PAUSE!<<<<"
LCDOUT $FE,$C0,"*]Start #]EXIT "

PAUSE_LOOP:
call ReadKey
if KEY=11 then
goto SETUP_MENU
endif

if KEY=10 then
goto main
endif

if AUTO_RPM_CUR > 400 then
AUTO_RPM_CUR=AUTO_RPM_CUR-1
```

```
endif
```

```
PULSOUT ESC,AUTO_RPM_CUR
```

```
pause 20
```

```
goto PAUSE_LOOP
```

```
ReadKey:
```

```
KEY_PRESS=0
```

```
high row1
```

```
high row2
```

```
high row3
```

```
high row4
```

```
low row1
```

```
if col1=0 then
```

```
key=1
```

```
goto endreadkey
```

```
endif
```

```
if col2=0 then
```

```
key=2
```

```
goto endreadkey
```

```
endif
```

```
if col3=0 then
```

```
key=3
```

```
goto endreadkey
```

```
endif
```

```
high row1
```

```
low row2
```

```
if col1=0 then
```



```
key=4
goto endreadkey
endif
if col2=0 then
key=5
goto endreadkey
endif
if col3=0 then
key=6
goto endreadkey
endif
```

```
high row2
low row3
if col1=0 then
key=7
goto endreadkey
endif
if col2=0 then
key=8
goto endreadkey
endif
if col3=0 then
key=9
goto endreadkey
endif
```

```
high row3
low row4
if col1=0 then
key=10
```




```
goto endreadkey
endif
if col2=0 then
key=0
goto endreadkey
endif
if col3=0 then
key=11
goto endreadkey
endif
```

```
high row4
key=$FF
```

EndReadKey:

```
if key=$FF then
KEY_PRESS=0
else
KEY_PRESS=1
endif
```

```
high row1
high row2
high row3
high row4
```

```
return
```

```
end
```



ภาคผนวก ก

1. PIC16F87X 28/40-pin 8-Bit CMOS FLASH Microcontrollers

Devices Included in this Data Sheet: Microcontroller Core Features:

- High-performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory, Up to 368 x 8 bytes of Data Memory (RAM)
- Up to 256 x 8 bytes of EEPROM data memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low-power, high-speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial and Industrial temperature ranges
- Low-power consumption:

- < 2 mA typical @ 5V, 4 MHz
- 20 mA typical @ 3V, 32 kHz
- < 1 mA typical standby current

Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit periodregister, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPIä (MasterMode) and I2Cä (Master/Slave)
- Universal Synchronous Asynchronous ReceiverTransmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

DEVICE OVERVIEW

This document contains device-specific information. Additional information may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip website. The Reference Manual should be considered a complementary document to this data sheet, and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules. There are four devices (PIC16F873, PIC16F874, PIC16F876 and PIC16F877) covered by this datasheet. The PIC16F876/873 devices come in 28-pin packages and the PIC16F877/874 devices come in 40-pin packages. The 28-pin devices do not have a Parallel

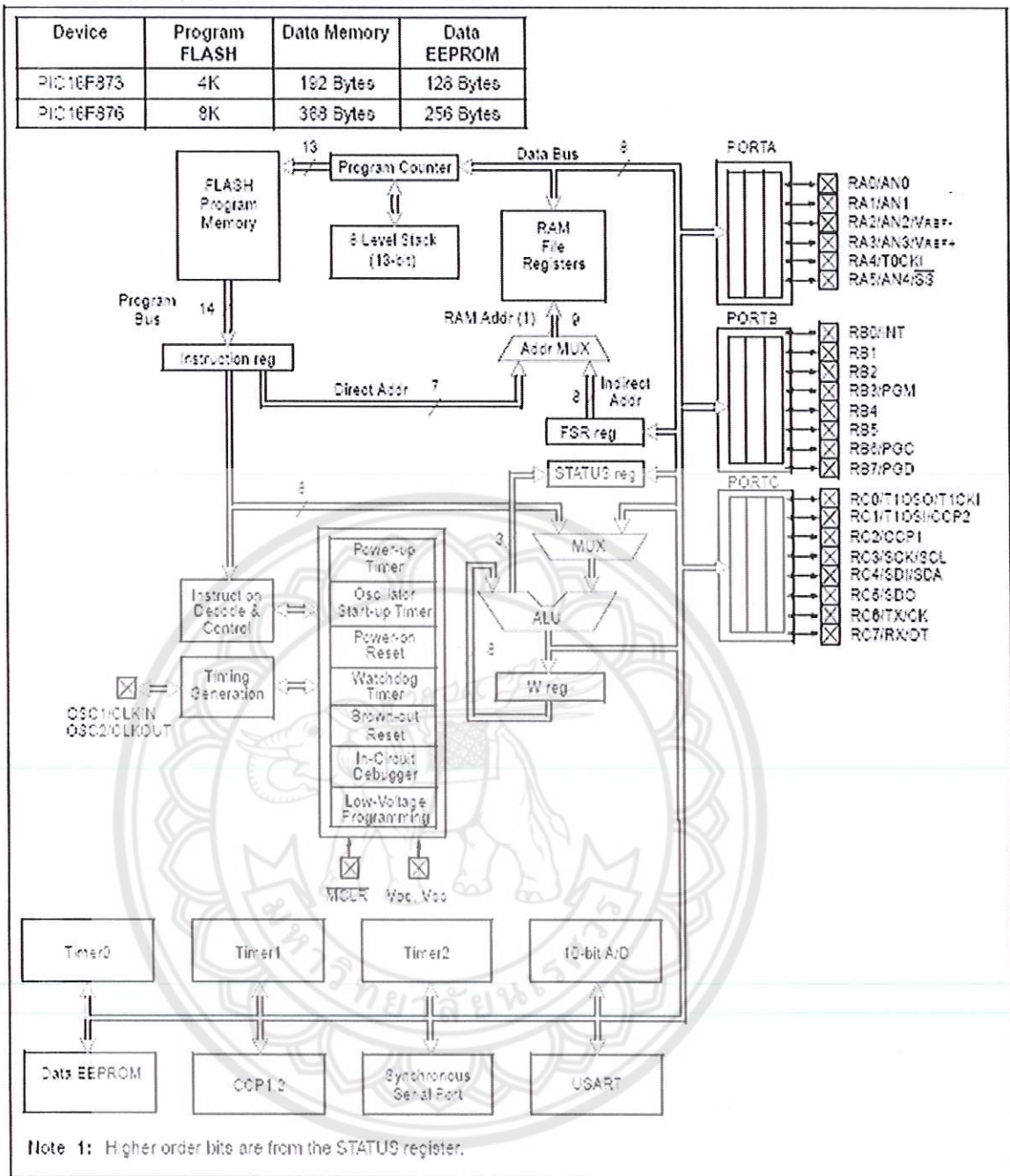


FIGURE 1-1: PIC16F873 AND PIC16F876 BLOCK DIAGRAM

Program Memory Organization

The PIC16F87X devices have a 13-bit program counter capable of addressing an 8K x 14 program memory space. The PIC16F877/876 devices have 8K x 14 words of FLASH program memory and the PIC16F873/874 devices have 4K x 14. Accessing a location above the physically implemented address will cause a wraparound. The reset vector is at 0000h and the interrupt vector is at 0004h.

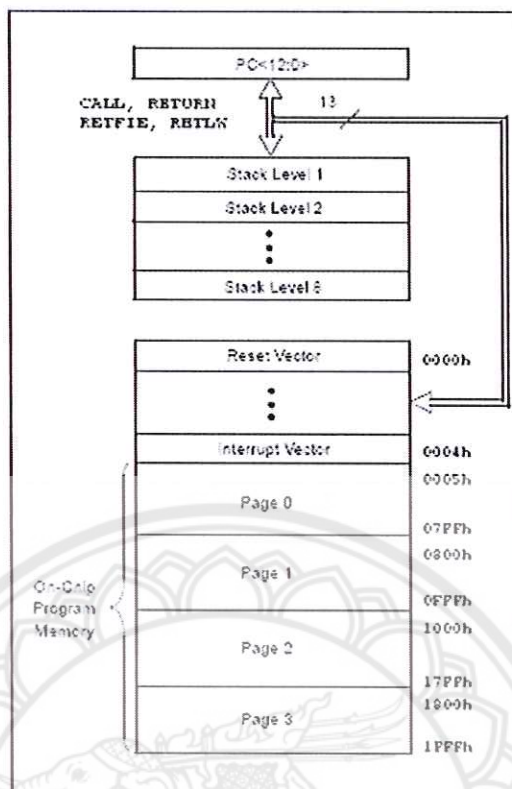


FIGURE 1-3: PIC16F877/876 PROGRAM MEMORY MAP AND STACK

Data Memory Organization

The data memory is partitioned into multiple banks which contain the General Purpose Registers and the Special Function Registers. Bits RP1(STATUS<6>) and RP0 (STATUS<5>) are the bank select bits.

RP1:RP0	Bank
00	0
01	1
10	2
11	3

Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers, implemented as static RAM. All implemented banks contain Special Function Registers. Some “high use” Special Function Registers from one bank may be mirrored in another bank for code reduction and quicker access.

OPTION_REG REGISTER

The OPTION_REG Register is a readable and writable register, which contains various control bits to configure the TMR0 prescaler/WDT postscaler (single assignable register known also as the prescaler), the External INT Interrupt, TMR0 and the weak pull-ups on PORTB.

INTCON REGISTER

The INTCON Register is a readable and writable register, which contains various enable and flag bits for the TMR0 register overflow, RB Port change and External RB0/INT pin interrupts.

I/O PORTS

Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin. Additional information on I/O ports may be found in the PIC micro™ Mid-Range Reference Manual, (DS33023).

PORTA and the TRISA Register

PORTA is a 6-bit wide bi-directional port. The corresponding data direction register is TRISA. Setting a TRISA bit (=1) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a hi-impedance mode). Clearing a TRISA bit (=0) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin). Reading the PORTA register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, the value is modified and then written to the port data latch. Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA4/T0CKI pin is a Schmitt Trigger input and an open drain output. All other PORTA pins have TTL input levels and full CMOS output drivers. Other PORTA pins are multiplexed with analog inputs and analog VREF input. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register1). The TRISA register controls the direction of the RA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

TIMER1 MODULE

The Timer1 module is a 16-bit timer/counter consisting of two 8-bit registers (TMR1H and TMR1L), which are readable and writable. The TMR1 Register pair (TMR1H:TMR1L) increments from 0000h to FFFFh and rolls over to 0000h. The TMR1 Interrupt, if enabled, is generated on overflow, which is latched in interrupt flag bit TMR1IF (PIR1<0>). This interrupt can be enabled/disabled by setting/clearing TMR1 interrupt enable bit TMR1IE (PIE1<0>). Timer1 can operate in one of two modes:

- As a timer
- As a counter

The operating mode is determined by the clock select bit, TMR1CS (T1CON<1>). In timer mode, Timer1 increments every instruction cycle. In counter mode, it increments on every rising edge of the external clock input. Timer1 can be enabled/disabled by setting/clearing control bit TMR1ON (T1CON<0>). Timer1 also has an internal “reset input”. This reset can be generated by either of the two CCP modules (Section 8.0). Register 6-1 shows the Timer1 control register. When the Timer1 oscillator is enabled (T1OSCEN is set), the RC1/T1OSI/CCP2 and RC0/T1OSO/T1CKI pins become inputs. That is, the TRISC<1:0> value is ignored. Additional information on timer modules is available in the PIC micro™ Mid-range MCU Family Reference Manual (DS33023).

Timer mode is selected by clearing the TMR1CS (T1CON<1>) bit. In this mode, the input clock to the timer is FOSC/4. The synchronize control bit T1SYNC (T1CON<2>) has no effect since the internal clock is always in sync.

Timer1 Counter Operation

Timer1 may operate in asynchronous or usynchronous mode depending on the setting of the TMR1CS bit. When Timer1 is being incremented via an external source, increments occur on a rising edge. After Timer1 is enabled in counter mode, the module must first have a falling edge before the counter begins to increment.

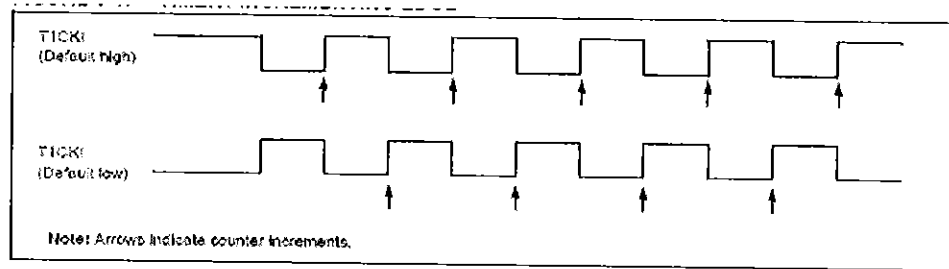


FIGURE 1-5: TIMER1 INCREMENTING EDGE

Timer1 Operation in Synchronized Counter Mode

Counter mode is selected by setting bit TMR1CS. In this mode, the timer increments on every rising edge of clock input on pin RC1/T1OSI/CCP2, when bit T1OSCEN is set, or on pin RC0/T1OSO/T1CKI, when bit T1OSCEN is cleared. If T1SYNC is cleared, then the external clock input is synchronized with internal phase clocks. The synchronization is done after the prescaler stage. The prescaler stage is an asynchronous ripple-counter. In this configuration, during SLEEP mode, Timer1 will not increment even if the external clock is present, since the synchronization circuit is shut off. The prescaler however will continue to increment.

TIMER2 MODULE

Timer2 is an 8-bit timer with a prescaler and a postscaler. It can be used as the PWM time-base for the PWM mode of the CCP module(s). The TMR2 register is readable and writable, and is cleared on any device reset. The input clock (FOSC/4) has a prescale option of 1:1, 1:4 or 1:16, selected by control bits T2CKPS1:T2CKPS0 (T2CON<1:0>). The Timer2 module has an 8-bit period register PR2. Timer2 increments from 00h until it matches PR2 and then resets to 00h on the next increment cycle. PR2 is a readable and writable register. The PR2 register is initialized to FFh upon reset. The match output of TMR2 goes through a 4-bit posts caler (which gives a 1:1 to 1:16 scaling inclusive) to generate a TMR2 interrupt (latched in flag bit TMR2IF, (PIR1<1>)). Timer2 can be shut off by clearing control bit TMR2ON (T2CON<2>) to minimize power consumption. Register 7-1 shows the Timer2 control register. Additional information on timer modules is available in the PIC micro™ Mid-Range MCU Family Reference Manua (DS33023).

Timer2 Prescaler and Postscaler

The prescaler and postscaler counters are cleared when any of the following occurs:

- a write to the TMR2 register
- a write to the T2CON register
- any device reset (POR, MCLR reset, WDT reset or BOR) TMR2 is not cleared

when T2CON is written.

Output of TMR2

The output of TMR2 (before the postscaler) is fed to the SSPort module, which optionally uses it to generate shift clock.

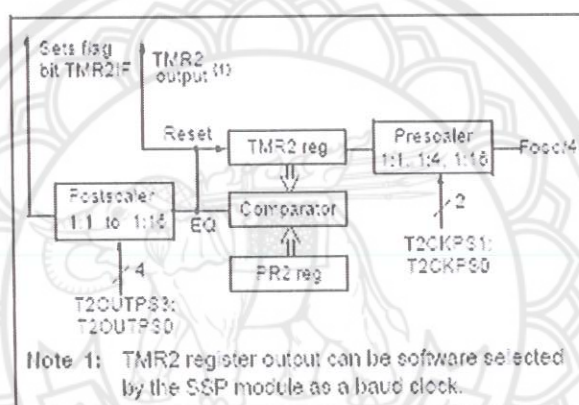


FIGURE 1-6: TIMER2 BLOCK DIAGRAM

2. LM2940/LM2940C 1A Low Dropout Regulator

General Description

The LM2940/LM2940C positive voltage regulator features the ability to source 1A of output current with a dropout voltage of typically 0.5V and a maximum of 1V over the entire temperature range. Furthermore, a quiescent current reduction circuit has been included which reduces the ground current when the differential between the input voltage and the output voltage exceeds approximately 3V. The quiescent current with 1A of output current and an input-output differential of 5V is therefore only 30 mA. Higher quiescent currents only exist when the regulator is in the dropout mode ($V_{IN} - V_{OUT} \leq 3V$).

Designed also for vehicular applications, the LM2940/ LM2940C and all regulated circuitry are protected from reverse battery installations or 2-battery jumps. During line

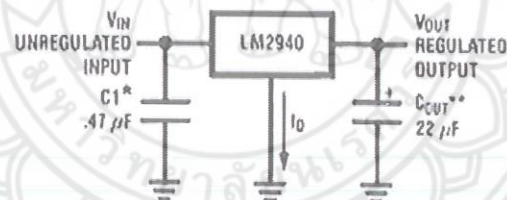
transients, such as load dump when the input voltage can momentarily exceed the specified maximum operating voltage, the regulator will automatically shut down to protect both the internal circuits and the load. The LM2940/ LM2940C cannot be harmed by temporary mirror-image insertion. Familiar regulator features such as short circuit and thermal overload protection are also provided.

Features

Dropout voltage typically 0.5V @ $I_o = 1A$

- Output current in excess of 1A
- Output voltage trimmed before assembly
- Reverse battery protection
- Internal short circuit current limit
- Mirror image insertion protection
- P+ Product Enhancement tested

Typical Application



*Required if regulator is located far from power supply filter.

**COUT must be at least 22 μF to maintain stability. May be increased without bound to maintain regulation during transients. Locate as close as possible to the regulator. This capacitor must be rated over the same operating temperature range as the regulator and the ESR is critical; see curve.

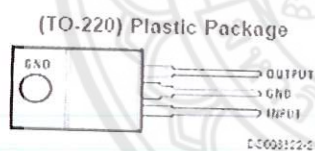
Ordering Information

Temperature Range	Output Voltage						Package
	5.0	8.0	9.0	10	12	15	
$0^{\circ}\text{C} \leq T_J \leq 125^{\circ}\text{C}$	LM2940CT-5.0		LM2940CT-9.0		LM2940CT-12	LM2940CT-15	TO-220
	LM2940CS-5.0		LM2940CS-9.0		LM2940CS-12	LM2940CS-15	TO-263
$-40^{\circ}\text{C} \leq T_J \leq 125^{\circ}\text{C}$	LM2940T-5.0	LM2940T-8.0	LM2940T-9.0	LM2940T-10	LM2940T-12		TO-220
	LM2940S-5.0	LM2940S-8.0	LM2940S-9.0	LM2940S-10	LM2940S-12		TO-263
$-40^{\circ}\text{C} \leq T_J \leq 85^{\circ}\text{C}$	LM2940IMP-5.0	LM2940IMP-8.0	LM2940IMP-9.0	LM2940IMP-10	LM2940IMP-12	LM2940IMP-15	SOT-223
	LM2940IMPX-5.0	LM2940IMPX-8.0	LM2940IMPX-9.0	LM2940IMPX-10	LM2940IMPX-12	LM2940IMPX-15	SOT-223 in Tape and Reel
SOT-223 Package Marking	L53B	L54B	L0EB	L55B	L56B	L70B	

The physical size of the SOT-223 is too small to contain the full device part number. The package markings indicated are what will appear on the actual device.

Temperature Range	Output Voltage				Package
	5.0	8.0	12	15	
$-55^{\circ}\text{C} \leq T_J \leq 125^{\circ}\text{C}$	LM2940J-5.0/883 5962-8958701EA	LM2940J-8.0/883 5962-9088301QEA	LM2940J-12/883 5962-9088401QEA	LM2940J-15/883 5962-9088501QEA	J16A
	LM2940WG5.0/883 5962-8958701XA				WG16A

Connection Diagrams



Front View

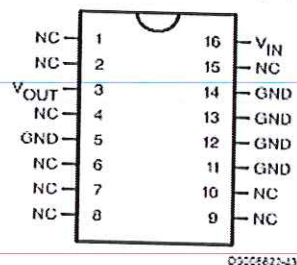
Order Number LM2940CT-5.0, LM2940CT-9.0,
LM2940CT-12, LM2940CT-15, LM2940T-5.0,
LM2940T-8.0, LM2940T-9.0,
LM2940T-10 or LM2940T-12
See NS Package Number TO3B



Front View

Order Part Number LM2940IMP-5.0,
LM2940IMP-8.0, LM2940IMP-9.0,
LM2940IMP-10, LM2940IMP-12 or LM2940IMP-15
See NS Package Number MP04A

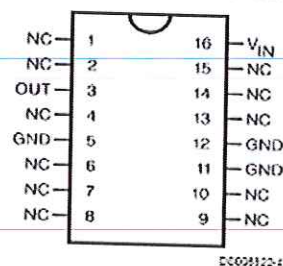
16-Lead Dual-in-Line Package (J)



Top View

Order Number LM2940J-5.0/883 (5962-8958701EA),
LM2940J-8.0/883 (5962-9088301QEA),
LM2940J-12/883 (5962-9088401QEA),
LM2940J-15/883 (5962-9088501QEA)
See NS Package Number J16A

16-Lead Ceramic Surface-Mount Package (WG)



Top View

Order Number LM2940WG5.0/883 (5962-8958701XA)
See NS Package Number WG16A

3. Brushless DC Motor Control

INTRODUCTION

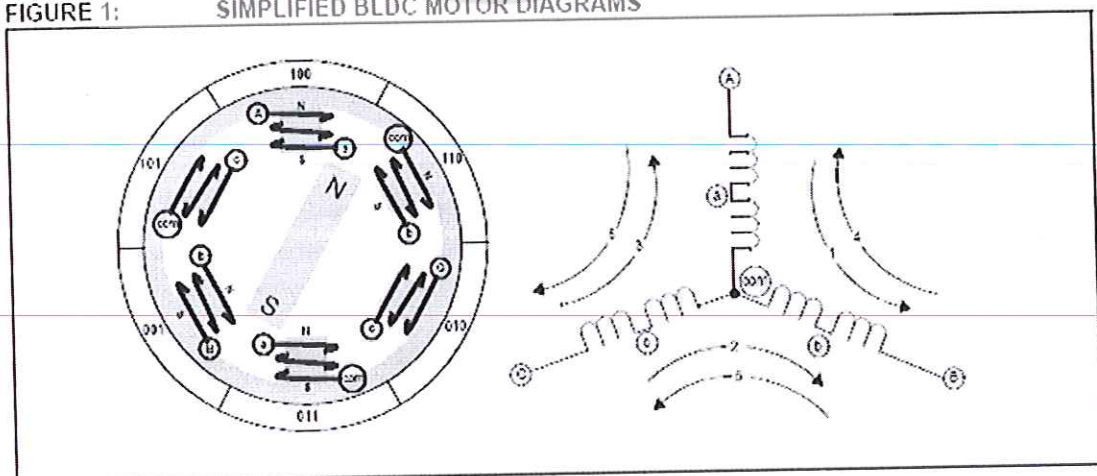
This application note discusses the steps of developing several controllers for brushless motors. We cover sensed, sensorless, open loop, and closed loop design. There is even a controller with independent voltage and speed controls so you can discover your motor's characteristics empirically.

The code in this application note was developed with the Microchip PIC16F877 PICmicro® Microcontroller, in conjunction with the In-Circuit Debugger (ICD). This combination was chosen because the ICD is inexpensive, and code can be debugged in the prototype hardware without need for an extra programmer or emulator. As the design develops, we program the target device and exercise the code directly from the MPLAB® environment. The final code can then be ported to one of the smaller, less expensive, PICmicro microcontrollers. The porting takes minimal effort because the instruction set is identical for all PICmicro 14-bit core devices.

It should also be noted that the code was bench tested and optimized for a Pittman N2311A011 brushless DC motor. Other motors were also tested to assure that the code was generally useful. Anatomy of a BLDC Figure 1 is a simplified illustration of BLDC motor construction.

A brushless motor is constructed with a permanent magnet rotor and wire wound stator poles. Electrical energy is converted to mechanical energy by the magnetic attractive forces between the permanent magnet rotor and a rotating magnetic field induced in the wound stator poles.

FIGURE 1: SIMPLIFIED BLDC MOTOR DIAGRAMS

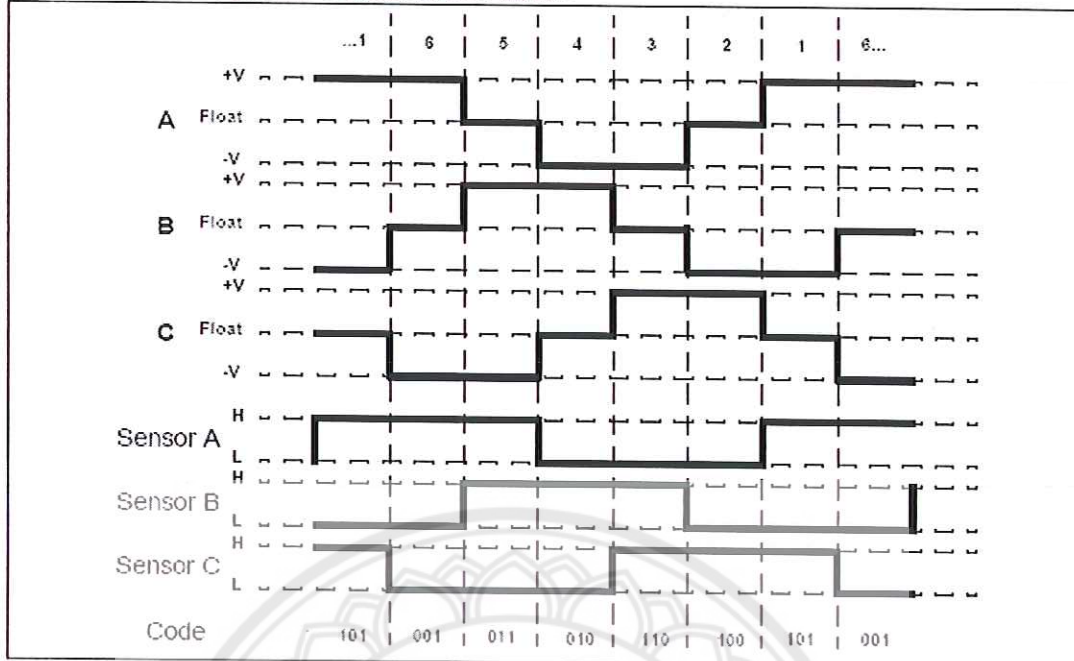


In this example there are three electromagnetic circuits connected at a common point. Each electromagnetic circuit is split in the center, thereby permitting the permanent magnet rotor to move in the middle of the induced magnetic field. Most BLDC motors have a three-phase winding topology with star connection. A motor with this topology is driven by energizing 2 phases at a time. The static alignment shown in Figure 2, is that which would be realized by creating an electric current flow from terminal A to B, noted as path 1 on the schematic in Figure 1. The rotor can be made to rotate clockwise 60 degrees from the A to B alignment by changing the current path to flow from terminal C to B, noted as path 2 on the schematic. The suggested magnetic alignment is used only for illustration purposes because it is easy to visualize. In practice, maximum torque is obtained when the permanent magnet rotor is 90 degrees away from alignment with the stator magnetic field.

The key to BLDC commutation is to sense the rotor position, then energize the phases that will produce the most amount of torque. The rotor travels 60 electrical degrees per commutation step. The appropriate stator current path is activated when the rotor is 120 degrees from alignment with the corresponding stator magnetic field, and then deactivated when the rotor is 60 degrees from alignment, at which time the next circuit is activated and the process repeats. Commutation for the rotor position, shown in Figure 1, would be at the completion of current path 2 and the beginning of current path 3 for clockwise rotation. Commutating the electrical connections through the six possible combinations, numbered 1 through 6, at precisely the right moments will pull the rotor through one electrical revolution. In the simplified motor of Figure 1, one electrical revolution is the same as one mechanical revolution. In actual practice, BLDC motors have more than one of the electrical circuits shown, wired in parallel to each other, and a corresponding multi-pole permanent magnetic rotor. For two circuits there are two electrical revolutions per mechanical revolution, so for a two circuit motor, each electrical commutation phase would cover 30 degrees of mechanical rotation. **Sensored Commutation**

The easiest way to know the correct moment to commutate the winding currents is by means of a position sensor. Many BLDC motor manufacturers supply motors with a three-element Hall effect position sensor. Each sensor element outputs a digital high level for 180 electrical degrees of electrical rotation, and a low level for the other 180 electrical degrees. The three sensors are offset from each other by 60 electrical degrees so that each sensor output is in alignment with one of the electromagnetic circuits. A timing diagram showing the relationship between the sensor outputs and the required motor drive voltages is shown in Figure 2.

FIGURE 2: SENSOR VERSUS DRIVE TIMING



The numbers at the top of Figure 2 correspond to the current phases shown in Figure 1. It is apparent from Figure 2 that the three sensor outputs overlap in such a way as to create six unique three-bit codes corresponding to each of the drive phases. The numbers shown around the peripheral of the motor diagram in Figure 1 represent the sensor position code. The north pole of the rotor points to the code that is output at that rotor position. The numbers are the sensor logic levels

where the Most Significant bit is sensor C and the Least Significant bit is sensor A.

Each drive phase consists of one motor terminal driven high, one motor terminal driven low, and one motor terminal left floating. A simplified drive circuit is shown in Figure 3. Individual drive controls for the high and low drivers permit high drive, low drive, and floating drive at each motor terminal. One precaution that must be taken with this type of driver circuit is that both high side and low side drivers must never be activated at the same time. Pull-up and pull-down resistors must be placed at the driver inputs to ensure that the drivers are off immediately after a microcontroller RESET, when the microcontroller outputs are configured as high impedance inputs.

Another precaution against both drivers being active at the same time is called dead time control. When an output transitions from the high drive state to the low drive state, the proper amount of time for the high side driver to turn off must be allowed to elapse before the low side

driver is activated. Drivers take more time to turn off than to turn on, so extra time must be allowed to elapse so that both drivers are not conducting at the same time. Notice in Figure 3 that the high drive period and low drive period of each output, is separated by a floating drive phase period. This dead time is inherent to the three phase BLDC drive scenario, so special timing for dead time control is not necessary. The BLDC commutation sequence will never switch the high-side device and the low-side device in a phase, at the same time.

At this point we are ready to start building the motor commutation control code. Commutation consists of linking the input sensor state with the corresponding drive state. This is best accomplished with a state table and a table offset pointer. The sensor inputs will form the table offset pointer, and the list of possible output drive codes will form the state table. Code development will be performed with a PIC16F877 in an ICD. I have arbitrarily assigned PORTC as the motor drive port and PORTE as the sensor input port. PORTC was chosen as the driver port because the ICD demo board also has LED indicators on that port so we can watch the slow speed commutation drive signals without any external test equipment. Each driver requires two pins, one for high drive and one for low drive, so six pins of PORTC will be used to control the six motor drive MOSFETS. Each sensor requires one pin, so three pins of PORTE will be used to read the current state of the motor's three-output sensor. The sensor state will be linked to the drive state by using the sensor input code as a binary offset to the drive table index. The sensor states and motor drive states from Figure 2 are tabulated in Table 1.

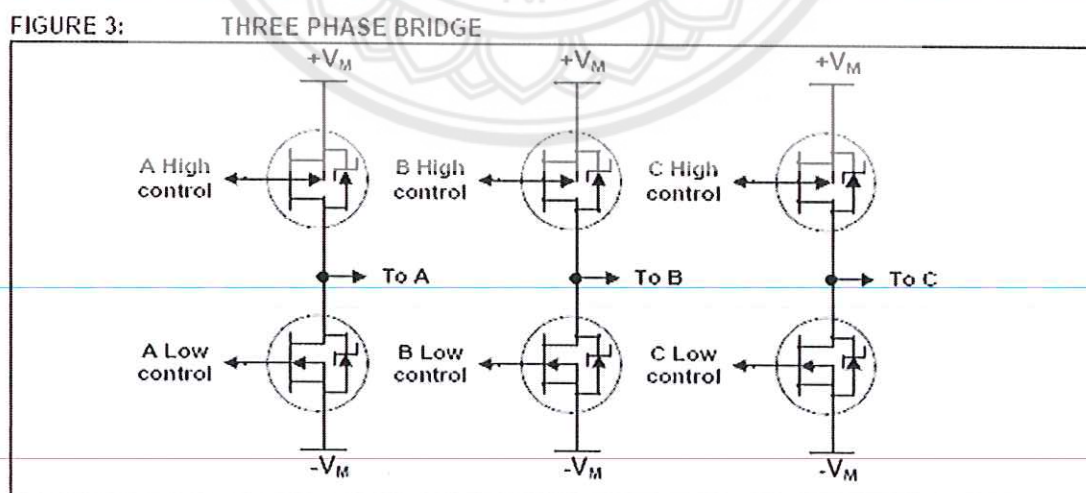


TABLE 1: CW SENSOR AND DRIVE BITS BY PHASE ORDER

Pin	RE2	RE1	RE0	RC5	RC4	RC3	RC2	RC1	RC0
Phase	Sensor C	Sensor B	Sensor A	C High Drive	C Low Drive	B High Drive	B Low Drive	A High Drive	A Low Drive
1	1	0	1	0	0	0	1	1	0
2	1	0	0	1	0	0	1	0	0
3	1	1	0	1	0	0	0	0	1
4	0	1	0	0	0	1	0	0	1
5	0	1	1	0	1	1	0	0	0
6	0	0	1	0	1	0	0	1	0

Sorting Table 1 by sensor code binary weight results in Table 2. Activating the motor drivers, according to a state table built from Table 2, will cause the motor of Figure 1 to rotate clockwise.

TABLE 2: CW SENSOR AND DRIVE BITS BY SENSOR ORDER

Pin	RE2	RE1	RE0	RC5	RC4	RC3	RC2	RC1	RC0
Phase	Sensor C	Sensor B	Sensor A	C High Drive	C Low Drive	B High Drive	B Low Drive	A High Drive	A Low Drive
6	0	0	1	0	1	0	0	1	0
4	0	1	0	0	0	1	0	0	1
5	0	1	1	0	1	1	0	0	0
2	1	0	0	1	0	0	1	0	0
1	1	0	1	0	0	0	1	1	0
3	1	1	0	1	0	0	0	0	1

Counter clockwise rotation is accomplished by driving current through the motor coils in the direction opposite of that for clockwise rotation. Table 3 was constructed by swapping all the high and low drives of Table 2. Activating the motor coils, according to a state table built from Table 3, will cause the motor to rotate counter clockwise. Phase numbers in Table 3 are preceded by a slash denoting that the EMF is opposite that of the phases in Table 2.

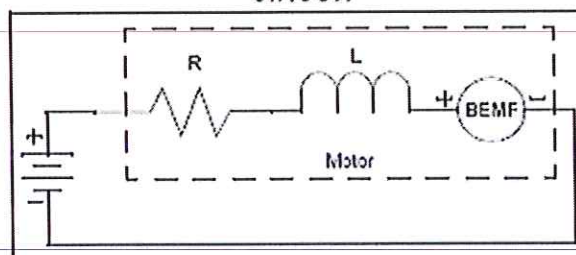
TABLE 3: CCW SENSOR AND DRIVE BITS

Pin	RE2	RE1	RE0	RC5	RC4	RC3	RC2	RC1	RC0
Phase	Sensor C	Sensor B	Sensor A	C High Drive	C Low Drive	B High Drive	B Low Drive	A High Drive	A Low Drive
/6	0	0	1	1	0	0	0	0	1
/4	0	1	0	0	0	0	1	1	0
/5	0	1	1	1	0	0	1	0	0
/2	1	0	0	0	1	1	0	0	0
/1	1	0	1	0	0	1	0	0	1
/3	1	1	0	0	1	0	0	1	0

The code segment for determining the appropriate drive word from the sensor inputs is shown in Figure 4.

Before we try the commutation code with our motor, let's consider what happens when a voltage is applied to a DC motor. A greatly simplified electrical model of a DC motor is shown in Figure 5.

FIGURE 5: DC MOTOR EQUIVALENT CIRCUIT



When the rotor is stationary, the only resistance to current flow is the impedance of the electromagnetic coils. The impedance is comprised of the parasitic resistance of the copper in the windings, and the parasitic inductance of the windings themselves. The resistance and inductance are very small by design, so start-up currents would be very large, if not limited.

When the motor is spinning, the permanent magnet rotor moving past the stator coils induces an electrical potential in the coils called Back Electromotive Force, or BEMF. BEMF is directly proportional to the motor speed and is determined from the motor voltage constant K_V .

EQUATION 1:

$$\begin{aligned} \text{RPM} &= K_V \times \text{Volts} \\ \text{BEMF} &= \text{RPM} / K_V \end{aligned}$$

In an ideal motor, R and L are zero, and the motor will spin at a rate such that the BEMF exactly equals the applied voltage. The current that a motor draws is directly proportional to the torque load on the motor shaft. Motor current is determined from the motor torque constant K_T .

EQUATION 2:

$$\text{Torque} = K_T \times \text{Amps}$$

An interesting fact about K_T and K_V is that their product is the same for all motors. Volts and Amps are expressed in MKS units, so if we also express K_T in MKS units, that is N-M/Rad/Sec, then the product of K_V and K_T is 1.

EQUATION 3:

$$K_V \times K_T = 1$$

This is not surprising when you consider that the units of the product are $[1/(V \cdot A)] \cdot [(N \cdot M)/(Rad/Sec)]$, which is the same as mechanical power divided by electrical power.

If voltage were to be applied to an ideal motor from an ideal voltage source, it would draw an infinite amount of current and accelerate instantly to the speed dictated by the applied

voltage and KV. Of course no motor is ideal, and the start-up current will be limited by the parasitic resistance and inductance of the motor windings, as well as the current capacity of the power source. Two detrimental effects of unlimited start-up current and voltage are excessive torque and excessive current. Excessive torque can cause gears to strip, shaft couplings to slip, and other undesirable mechanical problems. Excessive current can cause driver MOSFETS to blow out and circuitry to burn.

We can minimize the effects of excessive current and torque by limiting the applied voltage at start-up with pulse width modulation (PWM). Pulse width modulation is effective and fairly simple to do. Two things to consider with PWM are, the MOSFET losses due to switching, and the effect that the PWM rate has on the motor. Higher PWM frequencies mean higher switching losses, but too low of a PWM frequency will mean that the current to the motor will be a series of high current pulses instead of the desired average of the voltage waveform. Averaging is easier to attain at lower frequencies if the parasitic motor inductance is relatively high, but high inductance is an undesirable motor characteristic. The ideal frequency is dependent on the characteristics of your motor and power switches. For this application, the PWM frequency will be approximately 10 kHz.

We are using PWM to control start-up current, so why not use it as a speed control also? We will use the analog- to-digital converter (ADC), of the PIC16F877 to read a potentiometer and use the voltage reading as the relative speed control input. Only 8 bits of the ADC are used, so our speed control will have 256 levels. We want the relative speed to correspond to the relative potentiometer position. Motor speed is directly proportional to applied voltage, so varying the PWM duty cycle linearly from 0% to 100% will result in a linear speed control from 0% to 100% of maximum RPM. Pulse width is determined by continuously adding the ADC result to the free running Timer0 count to determine

When the drivers should be on or off. If the addition results in an overflow, then the drivers are on, otherwise they are off. An 8-bit timer is used so that the ADC to timer additions need no scaling to cover the full range. To obtain a PWM frequency of 10 kHz Timer0 must be running at 256 times that rate, or 2.56 MHz. The minimum prescale value for Timer0 is 1:2, so we need an input frequency of 5.12 MHz. The input to Timer0 is $F_{OSC}/4$. This requires an FOSC of 20.48 MHz. That is an odd frequency, and 20 MHz is close enough, so we will use 20 MHz resulting in a PWM frequency of 9.77 kHz.

There are several ways to modulate the motor drivers. We could switch the high and low side drivers together, or just the high or low driver while leaving the other driver on. Some high side MOSFET drivers use a capacitor charge pump to boost the gate drive above the drain voltage. The charge pump charges when the driver is off and discharges into the MOSFET gate when the driver is on. It makes sense then to switch the high side driver to keep the charge pump refreshed. Even though this application does not use the charge pump type drivers, we will modulate the high side driver while leaving the low side driver on. There are three high side drivers, any one of which could be active depending on the position of the rotor. The motor drive word is 6-bits wide, so if we logically AND the drive word with zeros in the high driver bit positions, and 1's in the low driver bit positions, we will turn off the active high driver regardless which one of the three it is. We have now identified 4 tasks of the control loop:

- Read the sensor inputs
- Commutate the motor drive connections
- Read the speed control ADC
- PWM the motor drivers using the ADC and Timer0

addition results

At 20 MHz clock rate, control latency, caused by the loop time, is not significant so we will construct a simple polled task loop. The control loop flow chart is shown in Figure 6

FIGURE 6: SENSORED DRIVE FLOWCHART

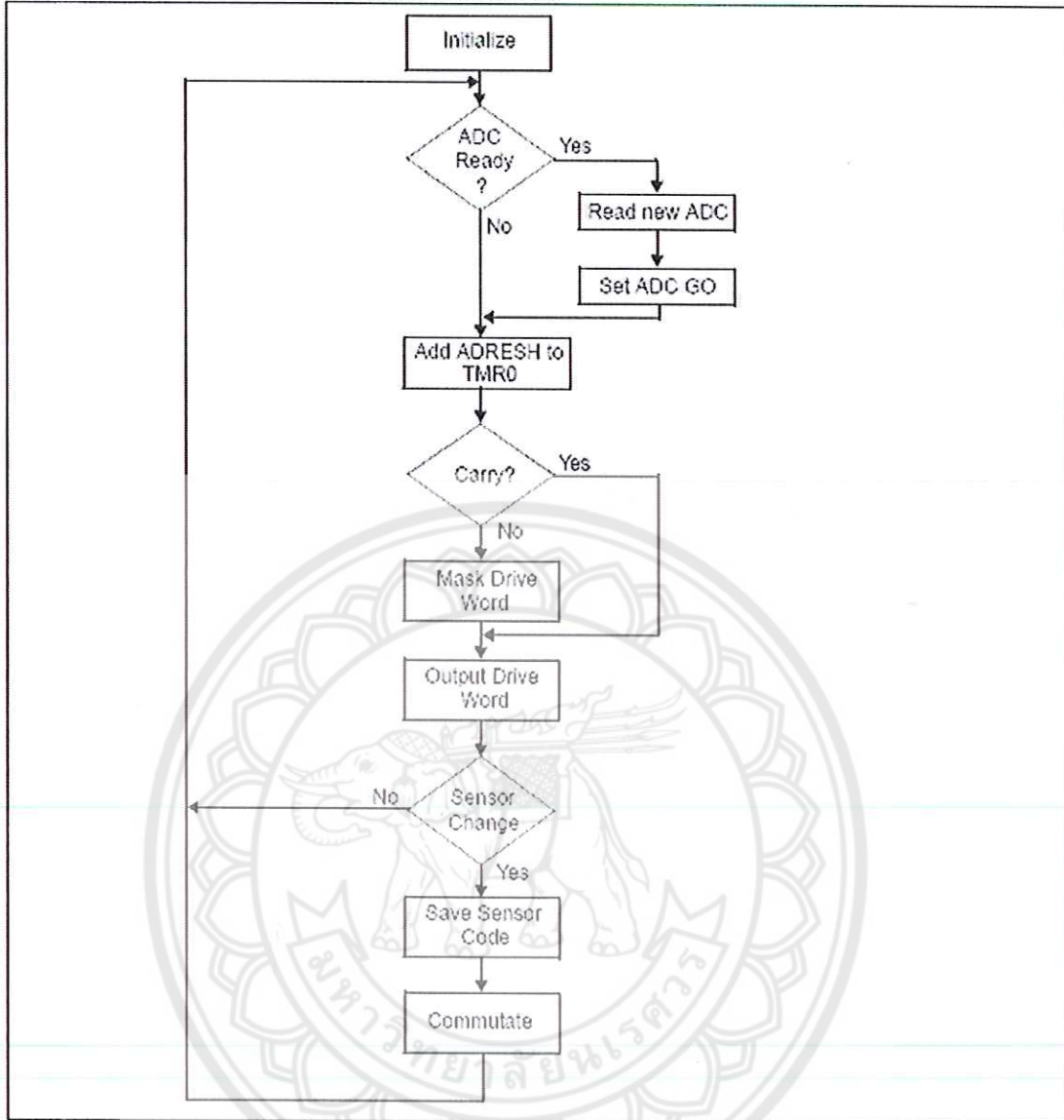
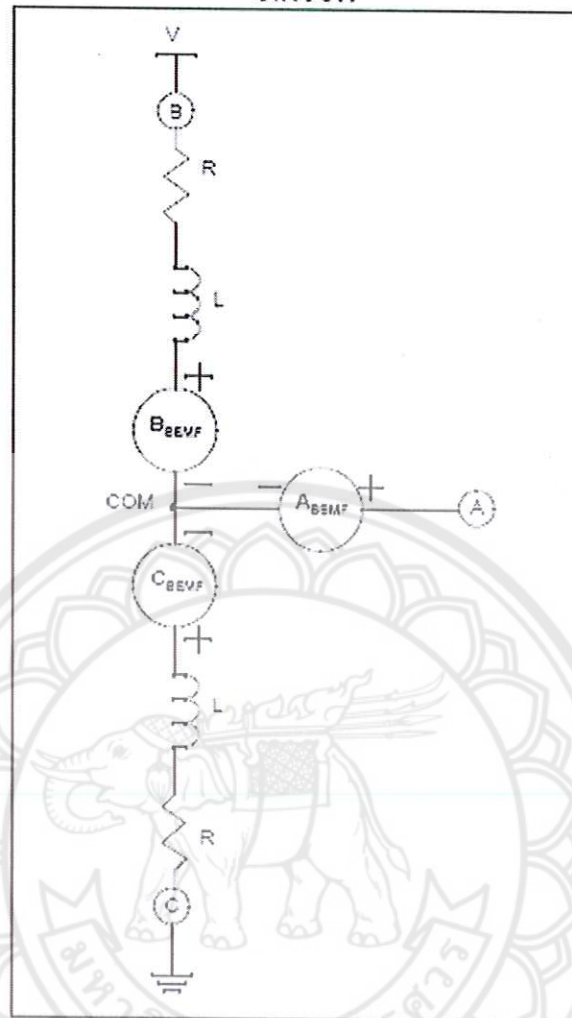


Figure 7 shows the equivalent circuit of the motor with coils B and C driven while coil A is undriven and available for BEMF measurement. At the commutation frequency the L's are negligible. The R's are assumed to be equal. The L and R components are not shown in the A branch since no significant current flows in this part of the circuit so those components can be ignored.

FIGURE 7: BEMF EQUIVALENT CIRCUIT



The BEMF generated by the B and C coils in tandem, as shown in Figure 7, can be expressed as shown in Equation 7.

EQUATION 7:

$$BEMF_{BD} = B_{BEMF} - C_{BEMF}$$

The sign reversal of CBEMF is due to moving the reference point from the common connection to ground. Recall that there are six drive phases in one electrical revolution. Each drive phase occurs +/- 30 degrees around the peak back EMF of the two motor windings being driven during that phase. At full speed the applied DC voltage is equivalent to the RMS BEMF

voltage in that 60 degree range. In terms of the peak BEMF generated by any one winding, the RMS BEMF voltage across two of the windings can be expressed as shown in Equation 8.

EQUATION 8:

$$BEMF_{RMS} = \sqrt{\frac{3}{\pi} \int_{\frac{\pi}{6}}^{\frac{\pi}{2}} \left(\sin(\alpha) - \sin\left(\alpha - \frac{2\pi}{3}\right) \right)^2 d\alpha}$$

$$BEMF_{RMS} = \sqrt{\frac{3}{\pi} \left(\frac{\pi}{2} + \frac{3\pi}{4} \right)}$$

$$BEMF_{RMS} = 1.6554$$

We will use this result to normalize the BEMF diagrams presented later, but first lets consider the expected BEMF at the undriven motor terminal. Since the applied voltage is pulse width modulated, the drive alternates between on and off throughout the phase time. The BEMF, relative to ground, seen at the A terminal when the drive is on, can be expressed as shown in Equation 9.

EQUATION 9:

$$BEMF_A = \frac{(V - (B_{BEMF} - C_{BEMF}))R}{2R} - C_{BEMF} + A_{BEMF}$$

$$BEMF_A = \frac{V - B_{BEMF} + C_{BEMF}}{2} - C_{BEMF} + A_{BEMF}$$

Notice that the winding resistance cancels out, so resistive voltage drop, due to motor torque load, is not a factor when measuring BEMF.

The BEMF, relative to ground, seen at the A terminal when the drive is off can be expressed as shown in Equation 10.

EQUATION 10:

$$BEMF_A = A_{BEMF} - C_{BEMF}$$

PWM Load Operations

When generating sine waves to a motor, an interrupt routine typically is used to step through a sine table in memory, scale that sine value, and output the result from the PWM generator. The rate at which the sine table is scanned can be derived from an interrupt from the PWM generator. The PWMMC module can be programmed to provide an interrupt rate of every one, two, four, or eight PWM reload cycles.

Direct Output Control

In some cases, the user may desire to bypass the PWM generator and directly control the PWM outputs. A mechanism exists to disconnect PWM generator from its outputs and directly control the six PWM outputs. When this mode is used, the PWM generator continues to run; however, its output is isabled, overridden by direct output. The PWMOUT register can also be used for BLDC motor control to selectively disable the PWM channels. This method permits a single 8-bit pattern to control the BLDC motor commutation.

Dead-Time Insertion and Dead-time Compensation

When the PWM generator is used in complementary mode, the PWMMC module provides dead-time insertion. This provides a short delay time between turning off the top power transistor and turning on the lower transistor in one phase leg. Dead-time is essential to safe switching and low-power losses. Dead-time can be specified in the dead-time writeonce register. This 8-bit value specifies the number of CPU clock cycles to use for the dead-time

During the dead-time, the current might be either positive or negative. The voltage on the motor phase will depend on the direction of the motor current. This leads to distortion from the desired voltage waveform. The MR32 provides three current sense inputs which can be used for deadtime compensation. Dead-time compensation provides smooth sine wave currents for ac induction motors even at low-speed operation. Most BLDC motors do not require complementary PWMs, dead-time insertion, or dead-time compensation.

The software presented here does not use these features. The only BLDC applications that might require these features are high-performance BLDC servomotors, sine wave excited BLDC motors, brushless ac, or ac synchronous motors. These types of motors account for less than 1 percent of BLDC motor drives today, but might find applications in electric vehicles, electric power steering, and robotics in the near future.

Brushless dc Motor Control Algorithm

Many different control algorithms have been used to provide sensorless control of BLDC motors. Most hard disk drive controllers use a linear control IC (integrated circuit) to provide sensorless control of the spindle drive. Typically, the motor voltage is controlled using a power transistor operating as a linear voltage regulator. This is not practical when driving higher-power motors. High-power motors must use PWM control and require a microcontroller to provide starting and control functions. The control algorithm must provide three things:

- PWM voltage which controls the motor speed
- Mechanism to commutate the motor
- Some method to estimate the rotor position using the back-EMF (electromotive force)

of the motor

Pulse-width modulation is used to apply a variable voltage to the motor windings. The effective voltage is proportional to the PWM duty cycle. When properly commutated, the torque-speed characteristics of the BLDC motor are identical to a dc motor. The variable voltage can be used to control the speed of the motor and the available torque.

The commutation of the power transistors energizes the appropriate windings in the stator to provide optimum torque generation depending on the rotor position. In a conventional dc motor, this task is performed by the brushes and commutator. In a BLDC motor, the MCU must know the position of the rotor and commutate at the appropriate time. When the rotor is turning, it acts as a generator. The moving magnets induce an ac voltage into the stator windings. This ac voltage is always present when the motor is turning. The applied voltage must be greater than this induced voltage to provide torque-generating current. In effect, the motor is pushing back with its own voltage. The induced voltage is called the back-EMF. EMF stands for electromotive force, a physics term for voltage. BLDC motor drive circuits are designed so that one of the three phases is always open, with both upper and lower transistors in the off state. The open phase is used to sense the back-EMF of the motor.

This provides useful information on the position of the rotor. Commutation A 3-phase brushless dc motor has six power transistors arranged in a 3-phase bridge (see Figure 3). The power transistors are switched according to a predefined commutation pattern. Several different patterns are used for BLDC motors. To use sensorless control, one phase must always be open. The motor has three windings, typically in a wye (Y) configuration. The center point is floating. So the open phase voltage is relative to the center point. The center point is affected by the voltage applied to the other two phases. Many BLDC motor drives modulate only the bottom transistors. This causes the center point to shift upward and the PWM voltage is actively coupled to the open phase. This commutation pattern, therefore, is not suitable for sensorless BLDC motors using PWM. A delta winding is electrically equivalent to a wye connection and behaves in a similar fashion relative to a virtual center point.

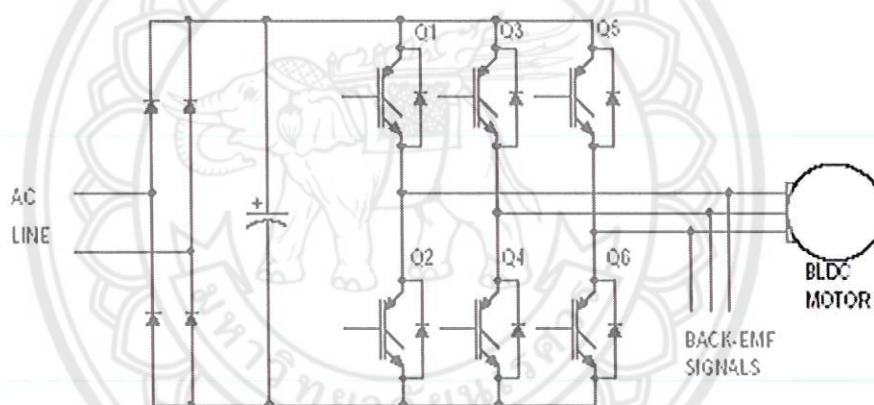


Figure 8 BLDC Motor System

The sensorless control algorithm developed for the MR32 uses a symmetrical PWM commutation pattern (see Table 1). The PWM signal always is applied to diagonally opposite transistors. This keeps the center point of the wye in the center of the dc supply voltage. The two active windings balance so that the PWM voltage is not coupled directly to the open winding. There still is some noise due to winding mismatch and capacitive coupling which is filtered out easily.

Table 4 BLDC Motor Commutation

	Top			Bottom		
	A	B	C	A	B	C
0	—	PWM	—	PWM	—	—
1	—	PWM	—	—	—	PWM
2	PWM	—	—	—	—	PWM
3	PWM	—	—	—	PWM	—
4	—	—	PWM	—	PWM	—
5	—	—	PWM	PWM	—	—
	Q1	Q3	Q5	Q2	Q4	Q6

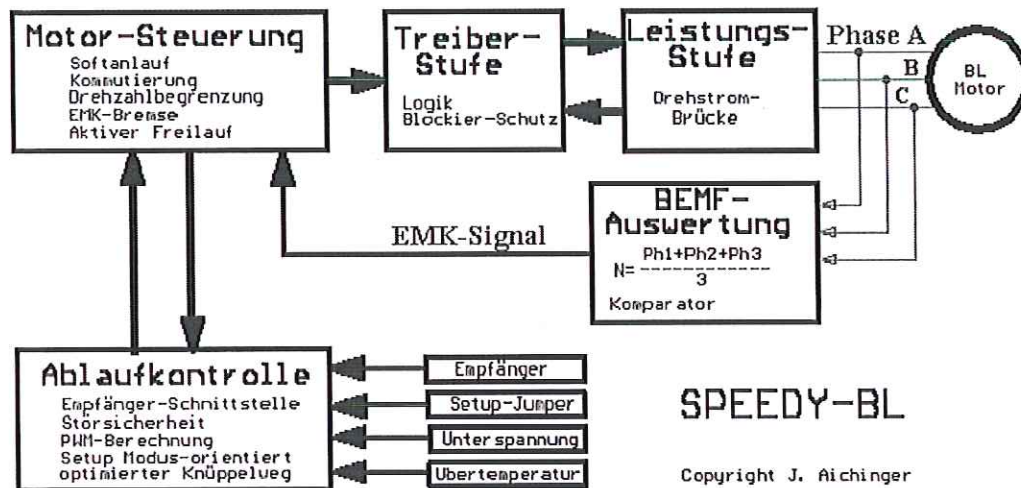
This commutation pattern requires only a single PWM signal. Identical PWM signals are applied to two of the six transistors. The software uses a trick to force the unused PWM channels to the off state. The unused PWM channels are disabled by setting the most significant bit (bit 15) of the PWM value register. This also corresponds to the sign bit for a 16-bit signed integer. ThePWMMCmodule is designed so that a number equal to or less than 0 will force the channel off. This feature is useful for ac induction motors and provides saturation for data values outside the normal operating range. Alternatively, the unused channels could be disabled by writing a 0 to the PWM value register. This commutation pattern also could be accomplished using a single PWM with a hardware multiplexer. However, this is not the only acceptable PWM strategy for BLDC motors. Applications which require servo-positioning or regenerative braking might require two PWMs with dead-time. Large motors greater than 750W might require three PWMs to minimize torque ripple. The MR32 PWMMC module is flexible enough to be used with most BLDC PWM strategies.

4. Speed Control

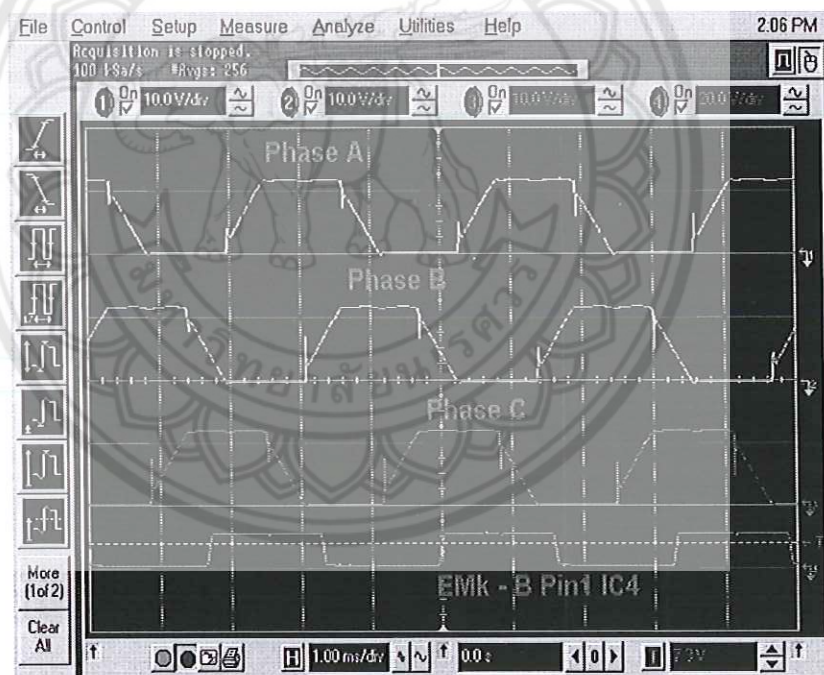
Reglerarbeitsweise bei Vollast

Das ist das Blockschaltbild, wie es auf der HomePage von Jo zu sehen ist:

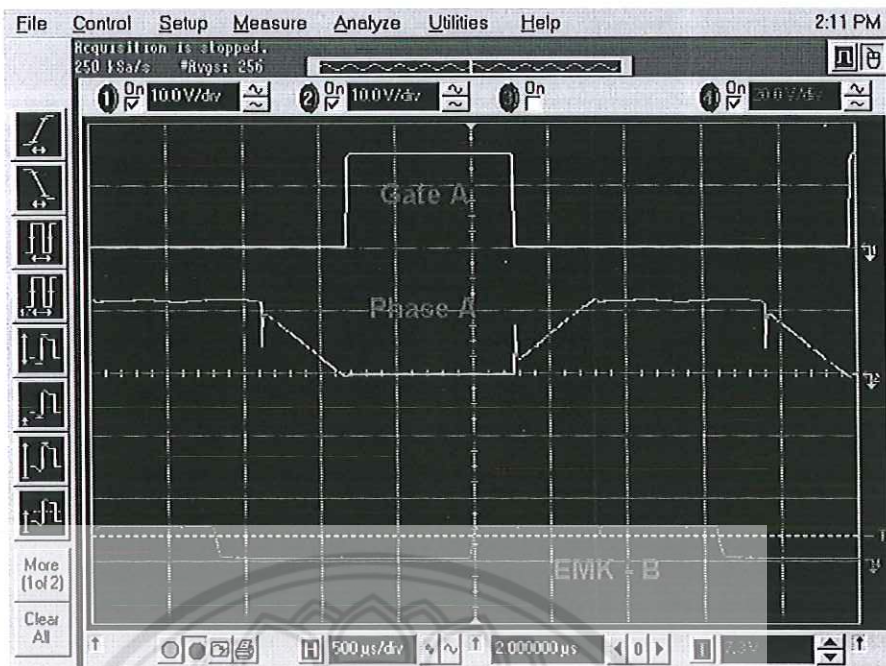
Wenn der Motor dreht, entstehen 3 Positionssignale, sogenannte EMK-Signale. Sie zeigen dem Prozessor (Motor-Steuerung im Bild) die Position des Rotors. Auf dem Scope-Bilder, die unterste Kurve, violett.



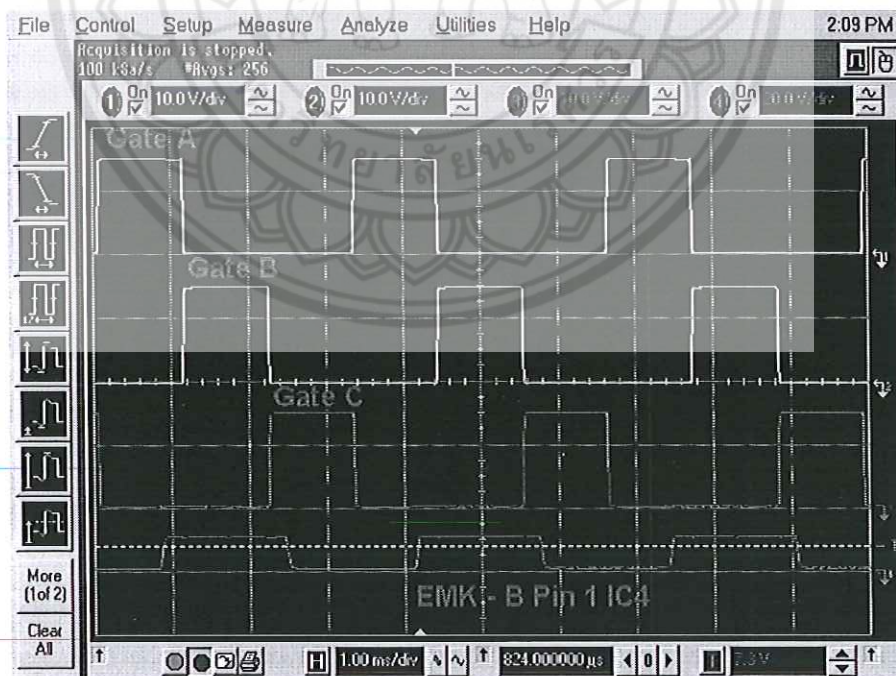
Die Phasen A, B, C geben mit 120 Grad Versatz die Akkuspannung auf den BL-Motor weiter, wie sich in einer 3-Phasen-Ansteuerung gehört. Das sieht folgend aus: (Bitte klicke auf das Bild, wenn Du reinzoomen willst)



Um die Leitung der Phase A mit Minus zu verbinden, muss man die Gate des unteren MOSFETs (es gibt 6 MOSFETs in der 3-Phasen H-Brücke) mit 15 V ansteuern (gelbe Linie, Gate A):



Um diese schöne 3-Phasen-Spannungen zu erreichen sehen die MOSFET-Gates-Ansteuerungen folgend aus, (wieder draufklicken, wenn Du zeitliche Vergrößerung Dir wünschst)

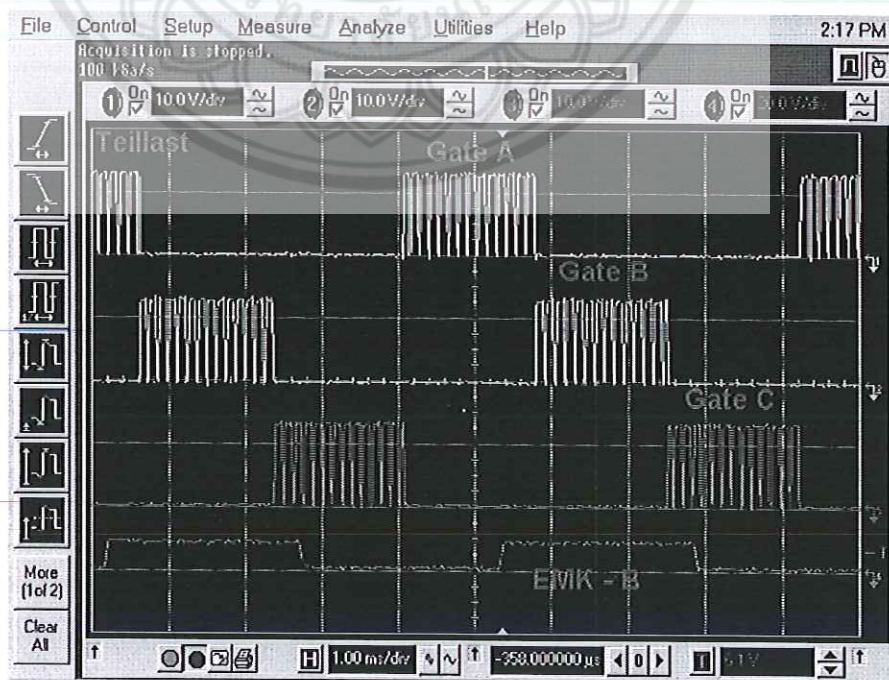


Reglerarbeitsweise bei Anlauf und Teillast

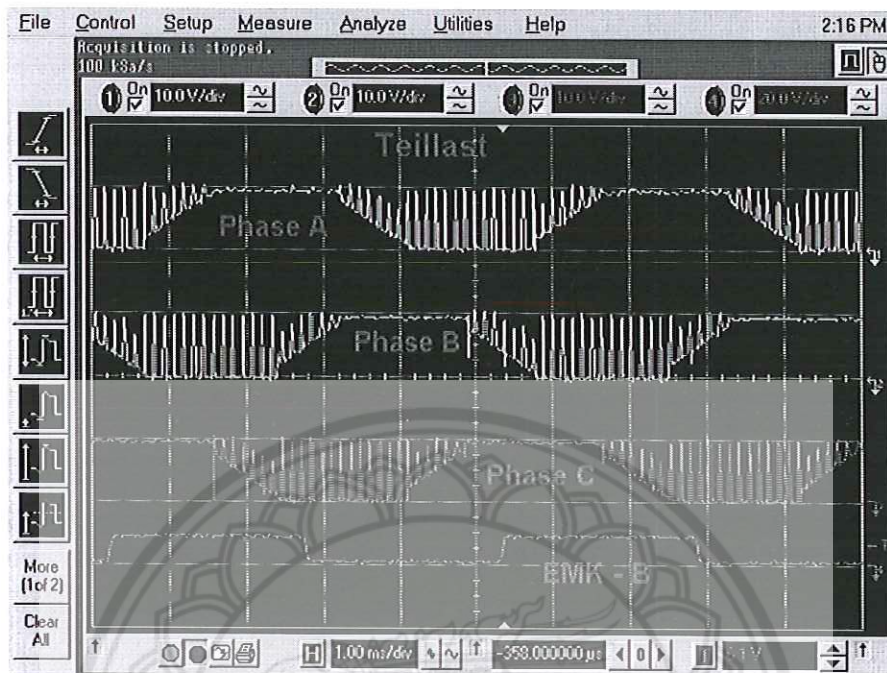
Bei Anlauf kann man dem stehenden Motor nicht die volle Akkuspannung draufknallen. Sonst ist seine Wicklung sofort verbrannt. Der Controller muss sehr zaghaft eine sehr schwache Teillastansteuerung ihm anbieten und über die EMK-Signale nachschauen, ob der Partner sich überhaupt drehen will. Erst wenn die Positionssignale (EMK) richtig dem aufgezwungenem Drehfeld folgen, kann man das Drehfeld beschleunigen, Teillast reduzieren, um am Ende in die Vollast überzugehen. Somit ist die Teillast eine sehr wichtige und schwierige Aufgabe für den Controller (und auch für den Entwickler dessen). Bei Speedy-BL wird zusätzlich ein sogenannter Freilauf im Teillast realisiert, was dramatisch die Erwärmung des Controllers reduziert. Darüber reden wir hier aber nicht, um den klaren Gedankenfluß nicht zu stören.

Aber kommen wir zurück zu der Teillast:

Da der BL-Motor bei Teillast weniger Strom bei gleicher Akkuspannung kriegen soll, müssen wir die schönen Rechtecke der Ansteuerung der MOSFETs (siehe oben) richtig zerhacken, damit im Schnitt die jeweilige Phase weniger Strom kriegt. Es sieht dann so aus (klicke drauf wenn Du reinzoomen willst):



Bei dieser Zerhackarbeit (engl. chopping) kriegt der BL-Motor, was er braucht - eben eine zerhackte Spannung, die aber weiter der Trapezform folgt:



Warum jetzt die EMK-Signale nicht zerhackt werden, obwohl sie direkt von den Phasen A, B, C abgeleitet sind, bleibt ein süßes Entwicklergeheimnis (und auch meiner). Der Prozessor (Motor-Steuerung) wäre ziemlich ratlos, würde er eine zerhackte Position bekommen

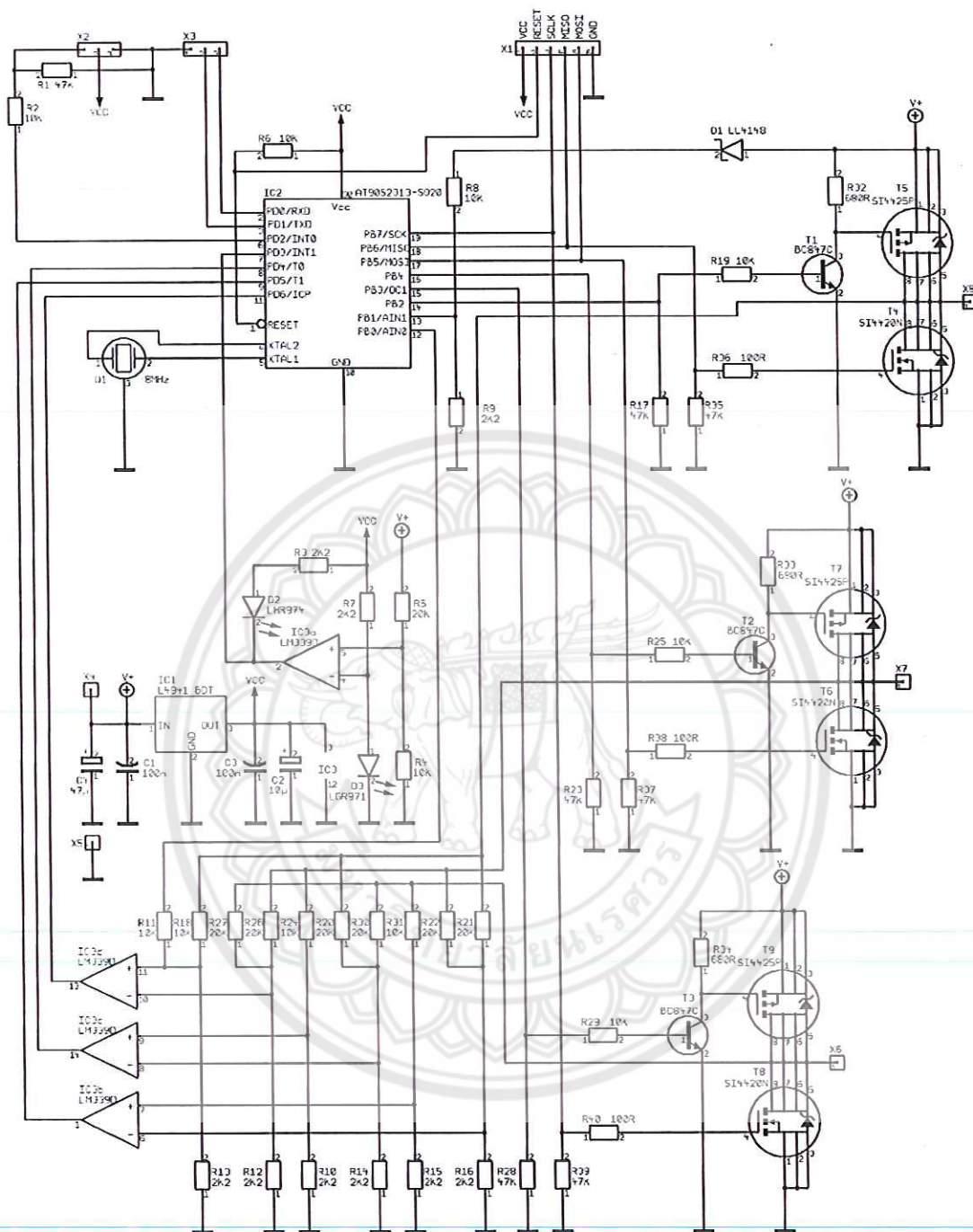


Figure 9 วงจร Speed Control

(ที่มา : <http://www.acrodesign.de/peter/2001/LRK350/SPEEDY-BL.html>)

(ที่มา : <http://www.microchip.com/downloads/en.AppNotes/00857a.pdf>)

ประวัติผู้เขียนโครงการ



ชื่อ นายนทวรรณ ก้นไชยดี
 ภูมิลำเนา 26/1 ม.13 ต.ห้วยซ้อ อ.เจียงของ จ.เชียงราย
 ประวัติการศึกษา
 - จบระดับมัธยมศึกษาจาก โรงเรียนห้วยซ้อวิทยาคม-
 รัชมังคลาภิเษก
 - ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์
 มหาวิทยาลัยนเรศวร
 E-mail : tum_n4@hotmail.com



ชื่อ นายแสงเพชร นาคจีน
 ภูมิลำเนา 170/4 ม.9 ต.สารจิตร อ.ศรีสัชนาลัย จ. สุโขทัย
 ประวัติการศึกษา
 - จบระดับมัธยมศึกษาจาก โรงเรียนบ้านแก่งวิทยา
 - ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์
 มหาวิทยาลัยนเรศวร
 E-mail : pech_1234@hotmail.com



ชื่อ นายเจษฎากร ไชยมงคล
 ภูมิลำเนา 42/1 ม.5 ต.ห้วยข้าวดำ อ.จุน จ.พะเยา
 ประวัติการศึกษา
 - จบระดับมัธยมศึกษาจาก โรงเรียนจุนวิทยาคม
 - ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์
 มหาวิทยาลัยนเรศวร
 E-mail : jet_0264@hotmail.com