

เครื่องปั่นเหวี่ยงเลือด

BLOOD CENTRIFUGE



เลขที่.....
เลขทะเบียน..... 81994
วัน,เดือน,ปี..... - 4 ก.ค. 2551

b. 11911213
i.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมระบบควบคุม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทบริหารศึกษา 2550

ภาควิชาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องปั่นเหวี่ยงเลือด
BLOOD CENTRIFUGE

ผู้จัดทำ นายนันท์วัฒน์ สถิตคุณากร 47010378
นายบรรพลาญ ทักษิณีแก้ว 47010403
นางสาวบัณฑิตา ชาญณรงค์ 47010406



[Signature] อาจารย์ที่ปรึกษา
(อาจารย์ สONGเมือง นันทขว้าง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องปั้นห้วยงเลียด

โดย

นายนันทวัฒน์ สติคุณากร 47010378

นายบรรหาญ ทัพศรีแก้ว 47010403

นางสาวบัณฑิตา ชาบุญรงค์ 47010406

อาจารย์ที่ปรึกษา

อาจารย์ สองเมือง นันทขว้าง

ปีการศึกษา 2550

บทคัดย่อ

ปฏิญานิพนธ์ฉบับนี้ นำเสนอการออกแบบเครื่องปั้นห้วยงเลียดขึ้นมาให้สามารถใช้งานได้ จุดมุ่งหมายคือลดการนำเข้าจากต่างประเทศ ซึ่งมีราคาแพง ขั้นแรกคือการออกแบบโครงสร้างภายนอกและภายในของเครื่องปั้นห้วยงเลียด โดยละเอียดผ่านทางโปรแกรมออโต้แคด จากโครงสร้างภายในเห็นว่ามอเตอร์ไฟฟ้าเป็นโครงสร้างหลักที่ทำให้เกิดแรงหมุนห้วยงขึ้น ทางโครงการนี้ได้เลือกใช้มอเตอร์เหนี่ยวนำแบบเฟสเดียว ขั้นตอนต่อมาคือการสร้างระบบขับเคลื่อนมอเตอร์ และระบบควบคุมความเร็วรอบของมอเตอร์ เพื่อให้เครื่องปั้นห้วยงเลียดนี้สามารถทำงานได้โดยสามารถปั้นห้วยงได้ด้วยความเร็วรอบค่าต่าง ๆ กัน เท่ากับค่าที่เราได้ตั้งค่าไว้ ซึ่งการควบคุมความเร็วรอบของมอเตอร์นี้จะถูกควบคุมโดยไมโครคอนโทรลเลอร์ ขอบเขตของโครงการนี้คือสามารถควบคุมความเร็วรอบของมอเตอร์ แบบวงปิดได้ โดยการทำงานร่วมกันของวงจรถ่ายค่าตั้ง และวงจรถ่ายค่าควบคุมจากไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BLOOD CENTRIFUGE

By

Mr. Nuntawat Satitkunakorn

Mr. Bunharn Tupsrigaew

Miss. Bantita Channarong

Advisor

Mr. Songmoung Nundrakwang

Academic Year 2007

ABSTRACT

This thesis presents design about blood centrifuge is applicable, objective is reduce importation from foreignism that expensive. First step, design detailed an inner and outside structure of blood centrifuge by program Auto-cad. From an inner structure, motor is main structure that generates centrifugal force. This project chooses single phase induction motor, next step is construction about motor driver system and speed control motor system for blood centrifuge can operational by rotate with varied speed that match with set-point value, speed control motor is controlled by microcontroller. Bound of this project is control speed of motor with the close loop type by co-operation of the power circuit and the control circuit from microcontroller.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำปริยฐานิพนธ์ฉบับนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความช่วยเหลือเป็นอย่างดี จาก อ.สองเมือง นันทขว้าง ที่ได้กรุณาให้คำปรึกษาแนะนำที่ดีมาโดยตลอดตั้งแต่ต้น และคอยสอบถาม ให้ความช่วยเหลืออื่นๆที่เป็นประโยชน์ต่อโครงการ ผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูง

ขอขอบพระคุณ พี่ๆนักศึกษาปริญญาโทที่ให้คำแนะนำ คอยช่วยเหลือ ให้คำอธิบาย และเสนอแนะแนวทางแก้ไขปัญหาให้ในบางปัญหาที่พบในการทำโครงการ

ขอบคุณเพื่อนๆ ทุกคนที่ให้กำลังใจ ช่วยสนับสนุนอุปกรณ์ที่ขาดเหลือ คอยกระตุ้นเตือน และถามไถ่ความคืบหน้าของโครงการอยู่เสมอ

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่เป็นกำลังใจที่ดีให้เสมอมา และรวมถึงการให้ความสนับสนุนในเรื่องของงบประมาณที่ขาดเหลือไป ตลอดจนเป็นแรงบันดาลใจที่ดีที่สุดที่ทำให้โครงการนี้สำเร็จลุล่วงไปได้ด้วยดี



ผู้จัดทำ

นายนันท์วัฒน์ สถิตคุณากร

นายบรรหาญ ทัพสิริแก้ว

นางสาวบัณฑิตา ชาญณรงค์

สารบัญ

	หน้า
บทคัดย่อ	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VII
บทที่ 1 บทนำ	1
1.1 กล่าวนำ	1
1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์	2
1.3 ขั้นตอนการศึกษาและบริหารจัดการโครงการ	2
1.4 รายละเอียดของปริญญานิพนธ์	3
บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง	4
2.1 ทฤษฎีมอเตอร์เหนี่ยวนำแบบเฟสเดียว	4
2.1.1 คำนำ	4
2.1.2 ทฤษฎีสถาปัตยกรรมแม่เหล็กขวางของมอเตอร์เหนี่ยวนำแบบเฟสเดียว	4
2.2 วงจรเรียงกระแส	7
2.2.1 วงจรเรียงกระแสแบบครึ่งคลื่น	7
2.2.2 วงจรเรียงกระแสแบบเต็มคลื่น	8
2.2.3 วงจรเรียงกระแสแบบบริดจ์	9
2.3 อินเวอร์เตอร์เฟสเดียว	10
2.4 รูปแบบการสวิตซ์	12
2.4.1 การสวิตซ์แรงดันไฟฟ้าแบบสแควร์	12
2.4.2 การสวิตซ์แรงดันแบบควอซีสแควร์	14
2.4.3 การสวิตซ์แรงดันไฟฟ้าแบบไบโพลาร์	17
2.4.4 การสวิตซ์แรงดันไฟฟ้าแบบยูนิโพลาร์	18
2.5 การคำนวณหาค่าฮาร์มอนิกส์	19
2.6 การควบคุมความเร็วรอบของมอเตอร์	20
2.6.1 การควบคุมความเร็วโดยการควบคุมความถี่	21
2.7 ประเภทของการควบคุม	21
2.7.1 ระบบควบคุมแบบวงเปิด	22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.7.2 ระบบควบคุมแบบวงปิด	23
2.7.3 ตัวควบคุมแบบ Proportional (แบบ P)	24
บทที่ 3 หลักการออกแบบ	25
3.1 การออกแบบโครงสร้างภายนอกและภายในของเครื่องปั้นเหวี่ยงเลือด	25
3.1.1 การออกแบบโครงสร้างภายนอกของเครื่องปั้นเหวี่ยงเลือด	25
3.1.2 การออกแบบโครงสร้างภายในของเครื่องปั้นเหวี่ยงเลือด	26
3.1.3 การออกแบบที่ใส่หลอดเลือด	27
3.2 การออกแบบระบบขับเคลื่อนมอเตอร์เหนี่ยวนำแบบเฟสเดียว	28
3.2.1 วงจรเรียงกระแส	28
3.2.2 วงจรสวิตซ์ชิ่ง	30
3.2.3 การควบคุมการทำงานของ ไอจีบีที	30
3.2.4 วงจรควบคุมการทำงานของ ไอจีบีที	31
3.2.5 วงจร ออปโต คัปเปิล	33
3.2.6 การทำงานของไมโครคอนโทรลเลอร์	34
บทที่ 4 การทดลอง	36
4.1 ผลการทดลองที่ค่าความเร็วรอบต่างๆ	36
4.1.1 ที่ความเร็วรอบ 1000 rpm	36
4.1.2 ที่ความเร็วรอบ 1400 rpm	38
4.1.3 ที่ความเร็วรอบ 2000 rpm	40
4.1.4 ที่ความเร็วรอบ 3000 rpm	42
บทที่ 5 บทวิจารณ์และสรุป	44
5.1 สรุปผลการทดลอง	44
5.2 ปัญหาที่พบและแนวทางแก้ไข	44
5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา	45
ภาคผนวก ก แสดงรูปส่วนประกอบของวงจร	46
ภาคผนวก ข โปรแกรมการติดต่อและโปรแกรมควบคุม	49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
ภาคผนวก ก เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์	
ก.1 เอกสารคู่มือการใช้งาน GT60M303	81
ก.2 เอกสารคู่มือการใช้งาน IR2111	84
ก.3 เอกสารคู่มือการใช้งาน TLP250	89
เอกสารอ้างอิง	93



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่	หน้า
1.1 หลักการทำงานของเครื่องปั่นเหวี่ยง	1
2.1 แสดงทิศทางการเกิดของกระแสไฟฟ้าและเส้นแรงแม่เหล็กในโรเตอร์ของมอเตอร์	5
2.2 แสดงรูปคลื่นของกระแสไฟฟ้าที่สเตเตอร์และ โรเตอร์ของมอเตอร์	6
2.3 ตำแหน่งการวางขดลวดของมอเตอร์เหนี่ยวนำเฟสเดียว	7
2.4 วงจรเรียงกระแสแบบครึ่งคลื่น	8
2.5 วงจรเรียงกระแสแบบเต็มคลื่น	9
2.6 วงจรเรียงกระแสแบบบริดจ์	10
2.7 อินเวอร์เตอร์เฟสเดียว แบบฮาล์ฟบริดจ์	11
2.8 อินเวอร์เตอร์เฟสเดียว แบบฟูลบริดจ์	11
2.9 วงจรอินเวอร์เตอร์ฟูลบริดจ์	12
2.10 วงจรอินเวอร์เตอร์สมมูลจากรูปที่ 2.9	12
2.11 สัญญาณขาออกการสวิตซ์แรงดันแบบสแควร์ ช่วง $0 - \pi$	13
2.12 สัญญาณขาออกการสวิตซ์แรงดันไฟฟ้าแบบสแควร์ช่วง $\pi - 2\pi$	13
2.13 สัญญาณแบบสแควร์	14
2.14 สัญญาณแบบควอซีสแควร์	14
2.15 การทำพีวีลิ่ง	15
2.16 สัญญาณแบบควอซีสแควร์ จาก $0 - \pi$	15
2.17 สัญญาณแบบควอซีสแควร์ จาก $\pi - 2\pi$	16
2.18 สัญญาณแบบควอซีสแควร์	16
2.19 การสวิตซ์แรงดันแบบไบโพลาร์	18
2.20 การสวิตซ์แรงดันแบบยูนิโพลาร์	19
2.21 แสดงบล็อกไดอะแกรมของระบบควบคุมแบบวงเปิด	22
2.22 แสดงบล็อกไดอะแกรมของระบบควบคุมความเร็วมอเตอร์แบบวงเปิด	22
2.23 แสดงบล็อกไดอะแกรมของระบบควบคุมแบบวงปิด	23
2.24 แสดงบล็อกไดอะแกรมของระบบควบคุมความเร็วมอเตอร์แบบวงปิด	23
2.25 แสดงบล็อกไดอะแกรมการควบคุมแบบสัดส่วน	24
3.1 แสดงโครงสร้างภายนอกของเครื่องปั่นเหวี่ยงเลือด	25
3.2 แสดงโครงสร้างภายในของเครื่องปั่นเหวี่ยงเลือด	26
3.3 แสดงที่ใส่หลอดเลือด	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

รูปที่	หน้า
3.4 วงจรเรียงกระแสพร้อมตัวกรองแรงดัน โดยใช้ตัวเก็บประจุ	28
3.5 กราฟแสดงแรงดันด้านเอาต์พุทเมื่อผ่านไดโอดบริดจ์และตัวกรอง	29
3.6 วงจรป้องกันกระแสพุ่งสูงในช่วงเริ่มต้นทำงาน	29
3.7 วงจรสวิตซ์	30
3.8 วงจรควบคุมการทำงานของไอจีบีที โดยใช้ไอซีเบอร์ IR2111	31
3.9 ไทม์มิ่ง ไดอะแกรม ของไอซีเบอร์ IR2111	32
3.10 วงจรอินเวอร์เตอร์เฟสเดียว แบบฟูลบริดจ์	32
3.11 วงจรออปโต คัปเปิล	33
3.12 โพล์ชาร์ตแสดงการทำงาน โดยรวมของไมโครคอนโทรลเลอร์	35
4.1 กราฟแสดงผลความเร็วรอบที่ตั้งไว้ 1000 rpm	36
4.2 กราฟแสดงแรงดันที่ ความเร็วรอบ 1000 rpm	37
4.3 กราฟแสดงกระแสที่ ความเร็วรอบ 1000 rpm	37
4.4 กราฟแสดงผลความเร็วรอบที่ตั้งไว้ 1400 rpm	38
4.5 กราฟแสดงแรงดันที่ ความเร็วรอบ 1400 rpm	39
4.6 กราฟแสดงกระแสที่ ความเร็วรอบ 1400 rpm	39
4.7 กราฟแสดงผลความเร็วรอบที่ตั้งไว้ 2000 rpm	40
4.8 กราฟแสดงแรงดันที่ ความเร็วรอบ 2000 rpm	41
4.9 กราฟแสดงกระแสที่ ความเร็วรอบ 2000 rpm	41
4.10 กราฟแสดงผลความเร็วรอบที่ตั้งไว้ 3000 rpm	42
4.11 กราฟแสดงแรงดันที่ ความเร็วรอบ 3000 rpm	43
4.12 กราฟแสดงกระแสที่ ความเร็วรอบ 3000 rpm	43
ก.1 แสดงวงจรเรียงกระแส	46
ก.2 แสดงวงจรป้องกันกระแสพุ่ง	46
ก.3 แสดงวงจรสวิตซ์	47
ก.4 แสดงวงจรขับเคลื่อนมอเตอร์เหนี่ยวนำแบบเฟสเดียว	47
ก.5 แสดงการต่อเอนโคเดอร์เข้ากับมอเตอร์เหนี่ยวนำแบบเฟสเดียว	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 กล่าวนำ

เครื่องปั่นเหวี่ยง เป็นเครื่องมือสำหรับใช้แยกอนุภาคต่างๆออกจากสารละลาย หรือสารแขวนลอย โดยอาศัยความแตกต่างของความหนาแน่น และขนาดของสาร หรืออนุภาคนั้นๆ เครื่องปั่นเหวี่ยง นี้มักใช้ในกระบวนการเตรียมสำหรับวิเคราะห์ ในงานวิจัยต่างๆทางด้านชีวเคมี ชีววิทยา และด้านการแพทย์ มักใช้แยกตัวอย่างส่วนที่เป็นของแข็งออกจากตัวอย่างส่วนที่เป็นของเหลว หรือใช้เพื่อแยกของเหลวหลายๆจำเพาะต่างกันให้เกิดการแยกชั้นกันขึ้น ใช้เร่งการตกตะกอนของอนุภาคที่ไม่ละลายออกจากของเหลว

ในโครงการนี้เลือกใช้ตัวอย่างเครื่องปั่นเหวี่ยงที่เป็นต้นแบบศึกษา คือ เครื่องปั่นเหวี่ยงที่ออกแบบมาเพื่อใช้กับงานเฉพาะทาง คือ ชนิดที่ใช้สำหรับแยกเม็ดเลือดแดงออกจากน้ำเลือด โดยใช้หลักการสร้างแรงดึงดูดเหวี่ยง หรือสร้างแรงหนีศูนย์กลางที่เกิดจากการหมุนรอบจุดหมุน โดยอนุภาคที่มีมวลต่างกัน จะมีกัการเรียงตัวในหลอดขณะหมุนเหวี่ยง ในอัตราที่ต่างกันตามการตอบสนองต่อแรงโน้มถ่วง ซึ่งแรงหมุนเหวี่ยงที่ใช้เพื่อเพิ่มอัตราการจัดเรียงตัวนี้ จะทำให้ของแข็งเคลื่อนที่ผ่านของเหลวออกมาเป็นแนวเส้นตรงออกจากศูนย์กลางการหมุน จึงแยกของแข็งและของเหลวออกจากกันได้ โดยมีแกนหมุนเป็นมอเตอร์ไฟฟ้า เมื่อมีกระแสไหลเข้ามอเตอร์จะเหนี่ยวนำให้เกิดคลื่นแม่เหล็กไฟฟ้า และหมุนไป ด้วยความเร็วรอบในการหมุนที่มีหน่วยเป็น รอบต่อนาที และความเร็วยุโรปนี้จะถูกควบคุมด้วยวงจรไฟฟ้า



รูปที่ 1.1 หลักการทำงานของเครื่องปั่นเหวี่ยง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์

1. ทำการออกแบบโครงสร้างภายในและภายนอกของเครื่องปั่นเหวี่ยงเลือดโดยละเอียด
2. ทำการศึกษาเรื่องมอเตอร์เหนี่ยวนำแบบเฟสเดียว(Single Phase Induction Motor)
3. ทำการศึกษาและออกแบบระบบขับเคลื่อนมอเตอร์เหนี่ยวนำแบบเฟสเดียว ซึ่งประกอบด้วยวงจรภาคกำลัง คือ วงจรเรียงกระแส วงจรสวิตชิง(Switching) วงจรควบคุมการทำงานของไอจีบีที(IGBT) และวงจรอปโต คัปเปิล(Opto Couple)
4. ทำการศึกษาและออกแบบระบบควบคุมความเร็วรอบของมอเตอร์เหนี่ยวนำแบบเฟสเดียว โดยไมโครคอนโทรลเลอร์(Microcontroller) dsPIC30F4011 และเป็นการควบคุมแบบวงปิด

1.3 ขั้นตอนการศึกษาและการจัดทำโครงงาน

การศึกษาเรื่องเครื่องปั่นเหวี่ยงเลือด จำเป็นต้องเรียนรู้เพื่อให้เกิดความเข้าใจในองค์ประกอบและหลักการทำงานของมันก่อน หลังจากที่ได้ทำการศึกษาแล้วจึงมาคิดค้นออกแบบโครงสร้างเครื่องปั่นเหวี่ยงเลือดขึ้นเอง โดยให้ความสำคัญที่สามารถใช้งานได้มีประสิทธิภาพที่สุด และจากการศึกษาโครงสร้างหลักภายในพบว่า โครงสร้างหลักที่สำคัญของเครื่องปั่นเหวี่ยงเลือดคือมอเตอร์ไฟฟ้า ซึ่งเป็นอุปกรณ์สำคัญที่ทำหน้าที่ผลิตให้เกิดการหมุนรอบแกน มีทั้งชนิดใช้แปรงถ่านและไม่ใช้แปรงถ่าน มอเตอร์ที่ใช้งานกันนั้นใช้ได้ทั้งแบบไฟฟ้ากระแสสลับและ ไฟฟ้ากระแสตรง และมีโรเตอร์(Rotor) เป็นส่วนสำหรับบรรจุภาชนะใส่ตัวอย่าง ซึ่งในที่นี้คือหลอดเลือด เมื่อเกิดการหมุนโดยการผลัดของมอเตอร์ จะเกิดแรงหนีศูนย์กลางขึ้น

สำหรับมอเตอร์ที่เป็น โครงสร้างหลักของเครื่องปั่นเหวี่ยงเลือด ทางโครงงานได้เลือกใช้มอเตอร์เหนี่ยวนำแบบเฟสเดียวเนื่องจากข้อได้เปรียบดังต่อไปนี้ คือ ราคาถูกกว่ามอเตอร์ชนิดไฟฟ้ากระแสตรงที่ขนาดพิกัดกำลังเท่ากัน ขนาดเล็ก โครงสร้างไม่ซับซ้อน มีความแข็งแรงทนทาน ต้องการการบำรุงรักษาน้อยมาก แต่มอเตอร์เหนี่ยวนำแบบเฟสเดียนี้ ต้องใช้อุปกรณ์ทางอิเล็กทรอนิกส์ ซึ่งเป็นวงจรภาคกำลัง อันประกอบด้วย วงจรเรียงกระแส วงจรสวิตชิง โดยมีไมโครคอนโทรลเลอร์เป็นตัวส่งสัญญาณไปทำการสวิตชิง

หลังจากที่สามารถขับเคลื่อนมอเตอร์ได้แล้ว จึงทำการออกแบบระบบควบคุมความเร็วรอบของมอเตอร์ ชนิดการควบคุมแบบวงปิด ซึ่งมีไมโครคอนโทรลเลอร์เป็นตัวควบคุม โดยเป็นการนำวงจรภาคกำลัง และภาคควบคุมจากไมโครคอนโทรลเลอร์มาทำงานร่วมกันในการควบคุมความเร็วรอบของมอเตอร์เหนี่ยวนำแบบเฟสเดียว ซึ่งเป็น โครงสร้างหลักที่สำคัญของเครื่องปั่นเหวี่ยงเลือดนี้ได้

1.4 รายละเอียดของปฏิญญานิพนธ์

เนื้อหาที่จะกล่าวในปฏิญญานิพนธ์ฉบับนี้ประกอบด้วย

บทที่ 1 บทนำ กล่าวถึงวัตถุประสงค์ หลักการ ขั้นตอนการศึกษา และการจัดทำโครงการ พร้อมทั้งรายละเอียดของปฏิญญานิพนธ์ของแต่ละบท

บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง กล่าวถึงหลักการและทฤษฎีที่เกี่ยวข้องในเรื่องของมอเตอร์เหนี่ยวนำแบบเฟสเดียว การขับเคลื่อนมอเตอร์ วงจรอิเล็กทรอนิกส์ที่เกี่ยวข้อง การควบคุมความเร็วรอบของมอเตอร์ และนำเอาความรู้ไปประยุกต์ใช้ในการจัดทำโครงการ

บทที่ 3 หลักการออกแบบ นำเสนอการออกแบบ โครงสร้างของเครื่องปั้นเหนี่ยวนำเฟสเดียว ภายในและภายนอกโดยละเอียด และ โครงสร้างของระบบขับเคลื่อนมอเตอร์เหนี่ยวนำแบบเฟสเดียว รวมถึงแนวคิดในการออกแบบระบบควบคุมความเร็วรอบของมอเตอร์

บทที่ 4 การทดลอง เป็นส่วนการทดสอบระบบขับเคลื่อนมอเตอร์เหนี่ยวนำแบบเฟสเดียว ตลอดจนการทดลองระบบควบคุมความเร็วรอบของมอเตอร์

บทที่ 5 บทวิจารณ์และสรุป จะสรุปผลการดำเนินงาน ปัญหาที่เกิดขึ้น และแนวทางการปรับปรุงพัฒนาโครงการนี้ต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและความรู้ที่เกี่ยวข้อง

จากที่ได้กล่าวในบทที่ 1 แล้วว่า ก่อนที่จะมีการออกแบบสร้างเครื่องปั้นหึ่งเลือกขึ้นมาต้องทำการศึกษาโครงสร้างหลักภายในเสียก่อน จากการศึกษาพบว่ามอเตอร์ไฟฟ้าเป็นโครงสร้างหลักภายในที่สำคัญของเครื่องปั้นหึ่งเลือกนี้ ที่ทำให้เครื่องปั้นหึ่งเลือกสามารถใช้งานได้จริงและทางโครงการได้เลือกใช้มอเตอร์เหนี่ยวนำแบบเฟสเดียว จึงจำเป็นต้องศึกษาองค์ประกอบต่างๆที่จำเป็นของระบบขับเคลื่อนมอเตอร์เหนี่ยวนำแบบเฟสเดียวให้เข้าใจ ในบทนี้จะศึกษาและอธิบายถึงมอเตอร์เหนี่ยวนำแบบเฟสเดียว องค์ประกอบต่างๆ ของระบบขับเคลื่อนมอเตอร์ รวมถึงรายละเอียดของระบบควบคุมความเร็วรอบมอเตอร์

2.1 ทฤษฎีมอเตอร์เหนี่ยวนำแบบเฟสเดียว

2.1.1 คำนำ

มอเตอร์เหนี่ยวนำแบบเฟสเดียวมีโครงสร้างคล้ายกับมอเตอร์ 3 เฟส แต่มีข้อแตกต่างกันตรงที่สเตเตอร์(Stator) ของแบบเฟสเดียวมีการพันขดลวดในลักษณะแบบเฟสเดียว และต่อเข้ากับแหล่งจ่ายไฟฟ้ากระแสสลับเพียง 1 เฟส ดังนั้นสนามแม่เหล็กที่เกิดขึ้นที่สเตเตอร์จึงไม่ใช่สนามแม่เหล็กที่หมุนด้วยความเร็วซิงโครนัส(Synchronous) เหมือนกับในกรณีของมอเตอร์ 2 เฟส หรือ 3 เฟสที่ได้รับแรงดันไฟฟ้ากระแสสลับ 2 เฟส หรือ 3 เฟส

จึงเป็นเหตุให้สนามแม่เหล็กที่เกิดการกลับไปกลับมามีที่สเตเตอร์นั้นไม่สามารถทำให้มอเตอร์เกิดแรงบิด และหมุนขึ้นได้ในขณะที่โรเตอร์นั้นยังหยุดอยู่ ซึ่งเป็นสาเหตุว่ามอเตอร์เหนี่ยวนำ เฟสเดียวไม่สามารถเริ่มหมุนได้ด้วยตัวมันเอง แต่จะหมุนได้ก็ต่อเมื่อโรเตอร์ถูกทำให้หมุนด้วยกรรมวิธีใดวิธีหนึ่ง ซึ่งหมายถึงต้องทำให้เริ่มหมุนด้วยมือหรืออุปกรณ์อื่นๆ ก่อน และเมื่อมอเตอร์ถูกช่วยทำให้หมุนไปทิศทางใดทิศทางหนึ่งก่อนแล้ว จะทำให้เกิดแรงบิดและอัตราเร่งขึ้นในโรเตอร์นั้นจนกระทั่งได้ความเร็วเต็มพิกัดของมัน

2.1.2 ทฤษฎีสันนามแม่เหล็กขวางของมอเตอร์เหนี่ยวนำแบบเฟสเดียว

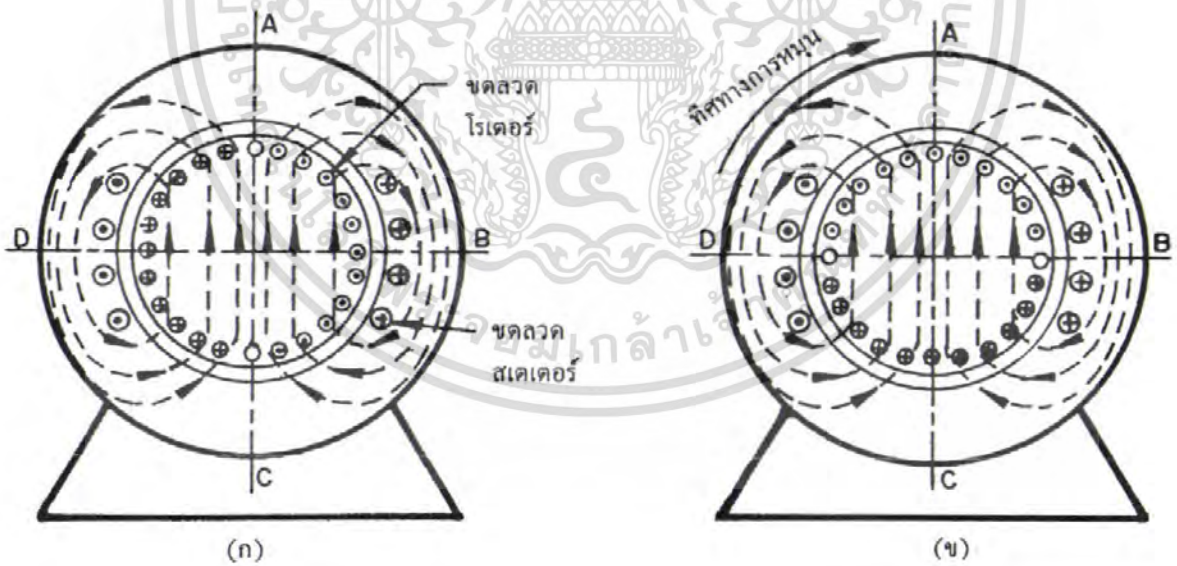
จากรูปที่ 2.1(ก) ถ้าโรเตอร์อยู่ในสนามแม่เหล็กที่เกิดขึ้นเนื่องจากไฟฟ้ากระแสสลับ จะทำให้เกิดแรงดันไฟฟ้าเหนี่ยวนำขึ้นในโรเตอร์ (ลักษณะคล้ายกับกรณีของหม้อแปลงไฟฟ้า) พร้อมกับส่งผลให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกิดกระแสไฟฟ้าไหลในโรเตอร์และเกิดสนามแม่เหล็กในโรเตอร์ โดยที่สนามแม่เหล็กที่เกิดขึ้นจากการเหนี่ยวนำนี้จะมีทิศทางตรงกันข้ามกับสนามแม่เหล็กที่สเตเตอร์และมีค่าเท่ากัน ผลลัพธ์ที่เกิดขึ้นที่โรเตอร์นั้นคือโรเตอร์จะไม่หมุน เนื่องจากการหักล้างกันระหว่างสนามแม่เหล็กที่ สเตเตอร์กับ โรเตอร์ ดังนั้นจึงไม่เกิดการหมุนหรือไม่เกิดแรงบิดเริ่มหมุนขึ้นในมอเตอร์ แต่แรงบิดจะเกิดขึ้นในโรเตอร์และโรเตอร์หมุนได้

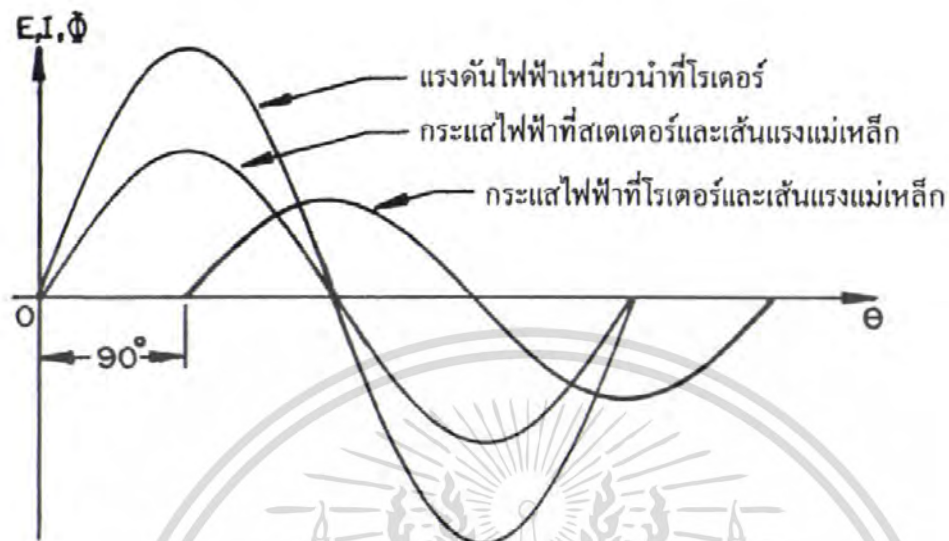
โดยการที่ตัวนำ ในโรเตอร์ตัดกับสนามแม่เหล็กซึ่งทำให้เกิดแรงดันไฟฟ้าเหนี่ยวนำขึ้น กระแสไฟฟ้าจะไหลอยู่ในแท่งตัวนำในโรเตอร์ ถ้าตัวนำที่โรเตอร์ถูกทำให้หมุนไปจากตำแหน่ง A ไปยังตำแหน่ง B ซึ่งทำมุมฉากกับตำแหน่งเดิม ตัวนำโรเตอร์จะเกิดการตัดผ่านสนามแม่เหล็กที่สเตเตอร์ ดังแสดงในรูปที่ 2.1(ข) ในขณะที่เดียวกันสนามแม่เหล็กก็จะแยกออกเป็น 2 ระบบคือต่างเฟสกันอยู่ 90 องศา และกระแสไฟฟ้าในสเตเตอร์และโรเตอร์ก็จะมีเฟสที่ต่างกัน

อย่างไรก็ตามในความเป็นจริงแล้วกระแสไฟฟ้าที่โรเตอร์จะต่างเฟสกับกระแสไฟฟ้าที่สเตเตอร์ เพราะเนื่องจากว่าที่โรเตอร์นั้นความถี่เปลี่ยนไป โดยที่ความถี่ของกระแสไฟฟ้าที่โรเตอร์จะขึ้นอยู่กับความเร็วรอบของโรเตอร์ ซึ่งผลลัพธ์ที่ได้ของเส้นแรงแม่เหล็กทั้งสองนี้แสดงได้ดังรูปที่ 2.2



รูปที่ 2.1 แสดงทิศทางของการเกิดของกระแสไฟฟ้าและเส้นแรงแม่เหล็กใน โรเตอร์ของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงรูปคลื่นของกระแสไฟฟ้าที่สเตเตอร์และโรเตอร์ของมอเตอร์

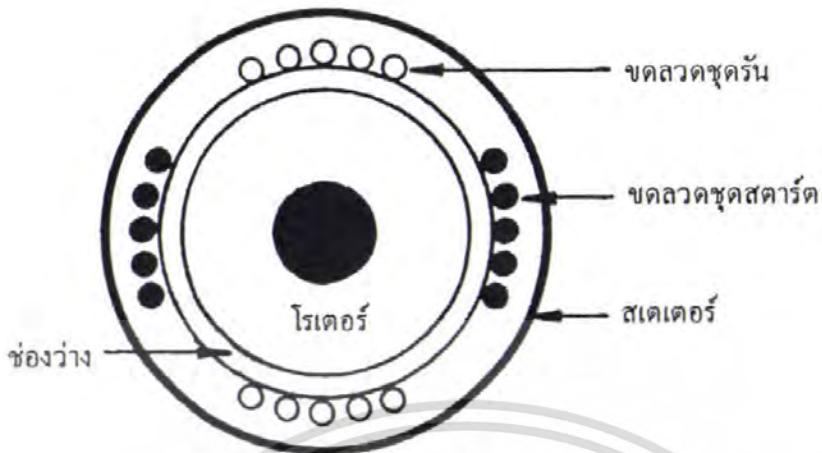
มอเตอร์เหนี่ยวนำกระแสสลับชนิด เฟสเดียว เป็นมอเตอร์ชนิดทำงาน โดยอาศัยแหล่งจ่ายไฟฟ้ากระแสสลับเฟสเดียว ซึ่งสามารถแบ่งได้เป็น 2 แบบ

ก. เครื่องกลชนิดกรงกระรอก ได้แก่ สปลิตเฟสมอเตอร์ (Split Phase Motor) คาปาซิเตอร์มอเตอร์ (Capacitor Motor) และเชดเคดโพลมอเตอร์ (Shaded Pole Motor)

ข. เครื่องกลชนิดโรเตอร์พันขดลวด ได้แก่ ซีรีส์มอเตอร์ (Series Motor) และรีพัลชันมอเตอร์ (Repulsion Motor)

มอเตอร์เหนี่ยวนำเฟสเดียวจะประกอบด้วยขดลวด 2 ชุดคือ ขดลวดชุดสตาร์ทหรือขดลวดช่วย (Starting Winding or Auxiliary Winding) และขดลวดชุดรันหรือขดลวดหลัก (Running Winding or Main Winding) ขดลวดทั้งสองวางทำมุมกัน 90 องศาทางไฟฟ้า และต่อक्रमเข้ากับแหล่งจ่ายไฟฟ้าเฟสเดียว ลักษณะตำแหน่งของขดลวดแสดงในรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 ตำแหน่งการวางขดลวดของมอเตอร์เหนี่ยวนำเฟสเดียว

มอเตอร์เหนี่ยวนำเฟสเดียวนี้ อาศัยความต่างเฟสระหว่างกระแสไฟฟ้าที่ไหลในขดลวดทั้งสองบนสเตเตอร์ที่วางทำมุมกัน 90 องศา ดังนั้นมอเตอร์จะมีคุณสมบัติหรืออาการเช่นเดียวกับมอเตอร์ 2 เฟส กระแสไฟฟ้าที่ไหลในขดลวดทั้งสองจะทำให้เกิดสนามแม่เหล็กหมุน จึงทำให้มอเตอร์สามารถเริ่มหมุนด้วยตัวเองได้

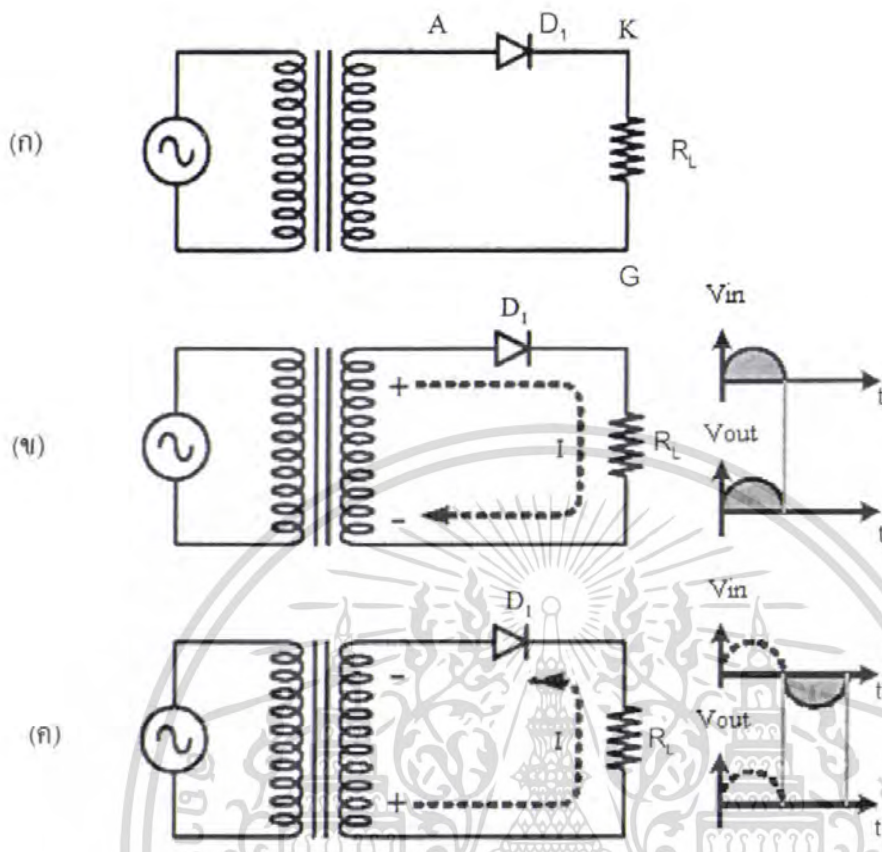
2.2 วงจรเรียงกระแส

วงจรเรียงกระแส เป็นวงจรเปลี่ยนแรงดันไฟฟ้ากระแสสลับเป็นไฟฟ้ากระแสตรง แบ่งออกเป็น 3 ประเภท คือ วงจรเรียงกระแสแบบครึ่งคลื่น, วงจรเรียงกระแสแบบเต็มคลื่น และวงจรเรียงกระแสแบบบริดจ์(Bridge)

2.2.1 วงจรเรียงกระแสแบบครึ่งคลื่น

เมื่อต่อเป็นวงจรไฟฟ้าดังรูป 2.4 (ก) พิจารณารูป 2.4 (ข) ถ้าอินพุท(Input) เป็น แรงดันไฟฟ้ากระแสสลับครึ่งบวก กระแสไหลจากขั้วบวก ผ่านไดโอด(Diode) แบบไบอัส(Bias)ตรงไปถึงขั้วลบ ดังนั้นสามารถวัดแรงดันเอาต์พุท(Output) ที่ R_L ได้ พิจารณารูป 2.4 (ค) ถ้าอินพุทเป็น แรงดันไฟฟ้ากระแสสลับครึ่งลบ กระแสไหลจากขั้วบวกผ่านไดโอดแบบไบอัสกลับ ทำให้ไม่มีกระแสไฟฟ้าไหลไปถึงขั้วลบ ดังนั้นจึงไม่สามารถวัดแรงดันเอาต์พุทที่ R_L ได้ ดังนั้นวงจรเรียงกระแสแบบครึ่งคลื่นจึงมีแรงดันเอาต์พุทเป็นไฟฟ้ากระแสตรงแค่ครึ่งเดียวเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

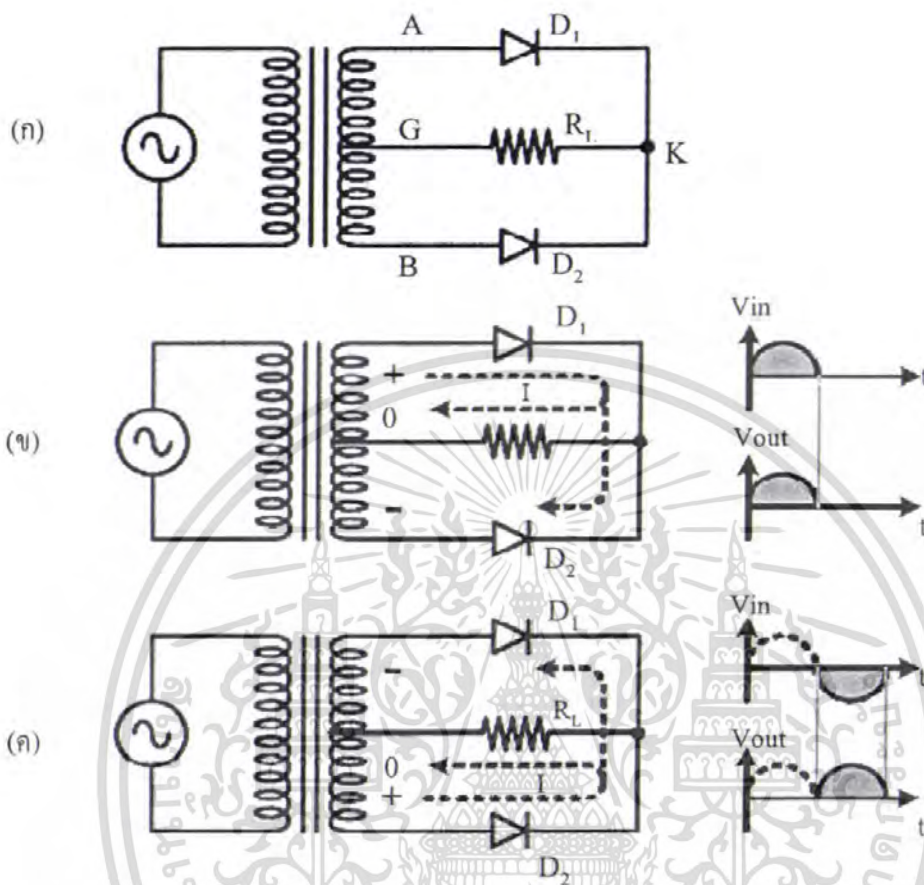


รูปที่ 2.4 วงจรเรียงกระแสแบบครึ่งคลื่น

2.2.2 วงจรเรียงกระแสแบบเต็มคลื่น

เมื่อต่อเป็นวงจรไฟฟ้าดังรูป 2.5 (ก) พิจารณารูป 2.5 (ข) ถ้าอินพุตเป็น แรงดันไฟฟ้ากระแสสลับครึ่งบวก กระแสไหลจากขั้ว บวกผ่านไดโอด D_1 แบบไบอัสตรงไปถึงกราวด์ ขณะที่กระแสไม่สามารถผ่านไดโอด D_2 ได้ เนื่องจากเป็นไบอัสกลับ ดังนั้นจึงสามารถวัดแรงดันเอาต์พุตที่ R_L ได้ ด้านขวาของ R_L เป็นแรงดันไฟฟ้าบวกเมื่อเทียบกับทางด้านซ้าย พิจารณารูป 2.5 (ค) ถ้าอินพุตเป็นแรงดันไฟฟ้ากระแสสลับครึ่งลบ กระแสไหลจากขั้ว บวกผ่านไดโอด D_2 แบบไบอัสตรงไปถึงกราวด์ ขณะที่กระแสไม่สามารถผ่านไดโอด D_1 ได้ เนื่องจากเป็นไบอัสกลับ ดังนั้นจึงสามารถวัดแรงดันเอาต์พุตที่ R_L ได้ ด้านขวาของ R_L เป็นแรงดันไฟฟ้าบวกเมื่อเทียบกับทางด้านซ้ายเช่นเดียวกัน จะเห็นได้ว่า ไม่ว่าขั้วของไฟฟ้ากระแสสลับจะเปลี่ยนแปลงไปอย่างไรก็ตาม ทางด้านขวาของตัวต้านทาน R_L เป็นบวกเสมอเมื่อเทียบกับทางด้านซ้ายดังนั้นแรงดันไฟฟ้าที่วัดได้จาก R_L จึงเป็นแรงดันไฟฟ้ากระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

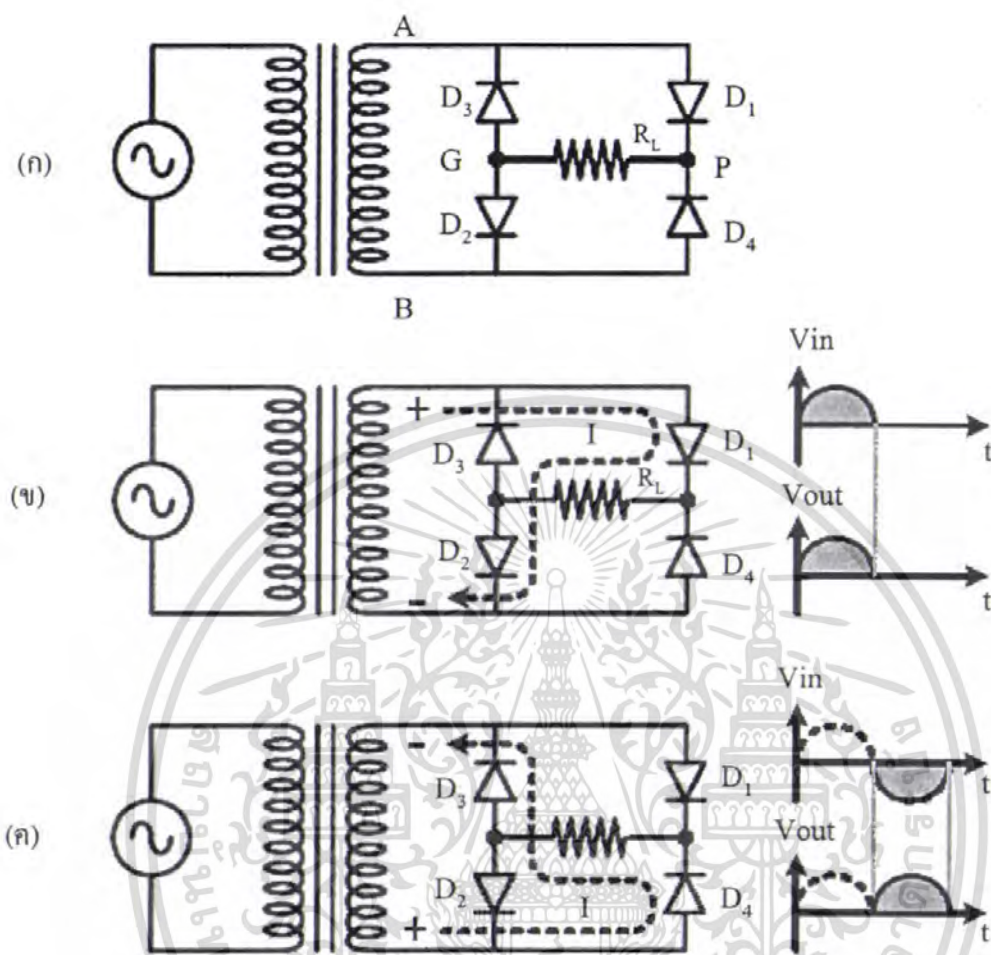


รูปที่ 2.5 วงจรเรียงกระแสแบบเต็มคลื่น

2.2.3 วงจรเรียงกระแสแบบบริดจ์

เมื่อต่อเป็นวงจรไฟฟ้าดังรูป 2.6 (ก) พิจารณารูป 2.6 (ข) ถ้าอินพุตเป็นแรงดันไฟฟ้ากระแสสลับครึ่งบวก กระแสไหลจากขั้วบวก ไบอัสตรงผ่านไดโอด D_1 และ D_2 ไปถึงขั้วลบ ดังนั้นจึงสามารถวัดแรงดันเอาต์พุตที่ R_L ได้ ด้านขวาของ R_L เป็นแรงดันไฟฟ้าบวกเมื่อเทียบกับทางด้านซ้าย พิจารณารูป 2.6 (ค) ถ้าอินพุตเป็นแรงดันไฟฟ้ากระแสสลับครึ่งลบ กระแสไหลจากขั้ว บวก ไบอัสตรงผ่านไดโอด D_4 และ D_3 ไปถึงขั้วลบ ดังนั้นจึงสามารถวัดแรงดันเอาต์พุตที่ R_L ได้ ด้านขวาของ R_L เป็นแรงดันไฟฟ้าบวกเมื่อเทียบกับทางด้านซ้ายเช่นเดียวกัน จะเห็นได้ว่า ไม่ว่าขั้วของไฟฟ้ากระแสสลับจะเปลี่ยนแปลงไปอย่างไรก็ตาม ทางด้านขวาของตัวต้านทาน R_L เป็น บวกเสมอเมื่อเทียบกับทางด้านซ้าย ดังนั้น แรงดันไฟฟ้าที่วัดได้จาก R_L จึงเป็นแรงดันไฟฟ้ากระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

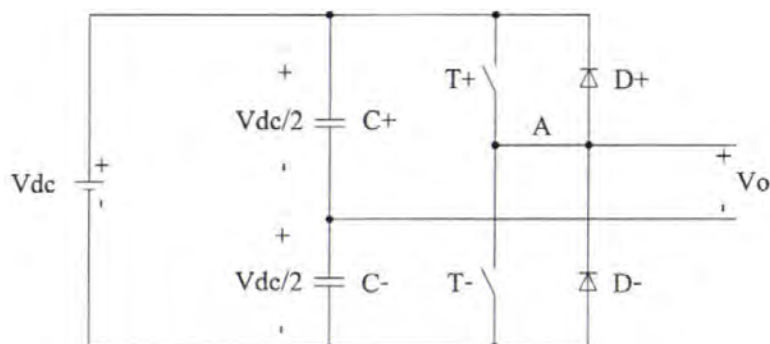


รูปที่ 2.6 วงจรเรียงกระแสแบบบริดจ์

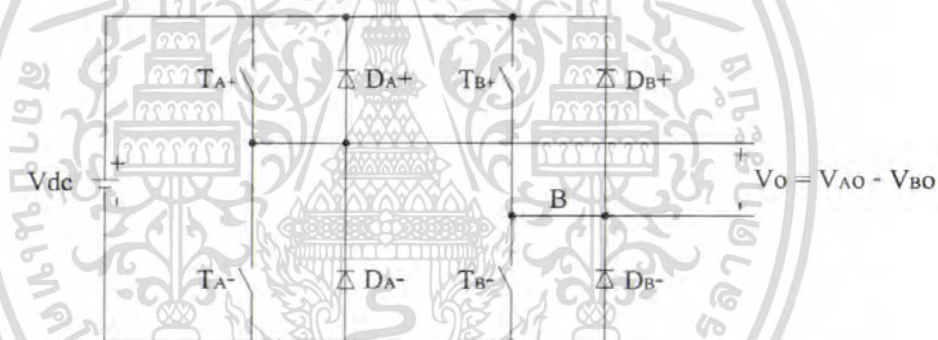
2.3 อินเวอร์เตอร์เฟสเดียว

อินเวอร์เตอร์(Inverter) เฟสเดียว แบ่งออกเป็น 2 ชนิด คือ แบบฮาล์ฟบริดจ์ (Half Bridge) และแบบฟูลบริดจ์ (Full Bridge) ในแบบฮาล์ฟบริดจ์ จะมีตัวเก็บประจุสองตัวต่อลำดับกันอยู่ระหว่างแหล่งจ่ายแรงดันไฟฟ้ากระแสตรง และหากกำหนดให้ค่าตัวเก็บประจุสองตัวมีค่าเท่ากัน จะทำให้แรงดันไฟฟ้าตกคร่อมตัวเก็บประจุแต่ละตัวมีค่าเท่ากันคือ V_{d2} จุดกึ่งกลางแรงดันไฟฟ้า จุด o ในรูปที่ 2.7 จะมีค่าคงที่เมื่อเทียบกับบัสลบ (N) ดังแสดงในรูป 2.7 ส่วนวงจรอินเวอร์เตอร์แบบฟูลบริดจ์เฟสเดียวจะประกอบไปด้วย สองกึ่งคือ กิ่ง A และ กิ่ง B โดยฟูลบริดจ์จะมีกำลังไฟฟ้าสูงกว่าแบบฮาล์ฟบริดจ์สองเท่าจึงเหมาะที่จะเลือกใช้เมื่อต้องการจ่ายกำลังไฟฟ้าโหลดสูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 อินเวอร์เตอร์เฟสเดียว แบบฮาล์ฟบริดจ์



รูปที่ 2.8 อินเวอร์เตอร์เฟสเดียว แบบฟูลบริดจ์

เงื่อนไขสำคัญของอินเวอร์เตอร์เฟสเดียวแบบฮาล์ฟบริดจ์และแบบฟูลบริดจ์ คือ การทำงานของสวิตช์ T_{A+} และ T_{A-} ต้องไม่ทำงานพร้อมกันในทุกช่วงเวลา มิฉะนั้นแล้วจะเกิดการลัดวงจรระหว่างบัสบวกกับบัสลบ ในทางปฏิบัติจะต้องการช่วงเวลาที่ยาวขึ้นที่สวิตช์ทั้งคู่ไม่นำกระแส ซึ่งจะเรียกว่า เดดไทม์ (Deadtime) โดยเดดไทม์จะอยู่ในช่วงเวลา ก่อนที่จะเปลี่ยนสถานะการนำกระแสจากนำกระแสเป็นไม่นำกระแส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

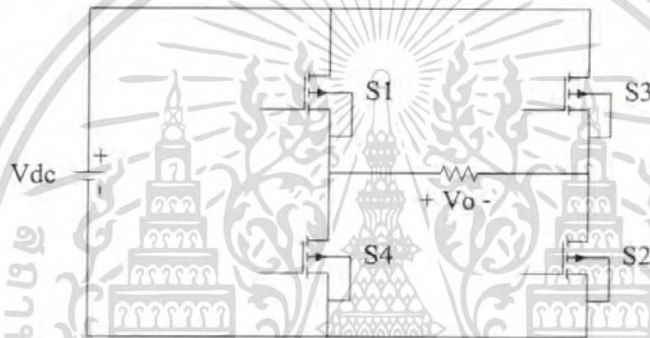
2.4 รูปแบบการสวิตชิง

2.4.1 การสวิตชิงแรงดันไฟฟ้าแบบสแควร์

การสวิตชิงแรงดันแบบสแควร์(square voltage switching) คือการควบคุมให้สวิตช์แบบบริดจ์ทำงานพร้อมกันเป็นคู่ เช่นในรูป 2.9 การทำงานของสวิตช์ S1 และ S2 จะถูกควบคุมให้ทำงานพร้อมกันในแต่ละช่วงเวลา อีกคู่หนึ่งคือการทำงานของสวิตช์ S3 และ S4 ดังนั้นแรงดันไฟฟ้าขาออก V_o จะเท่ากับ

$$V_o = V_{dc}, \quad \text{สวิตช์ S1 และ S2 จะนำกระแส}$$

$$V_o = -V_{dc}, \quad \text{สวิตช์ S3 และ S4 จะนำกระแส}$$



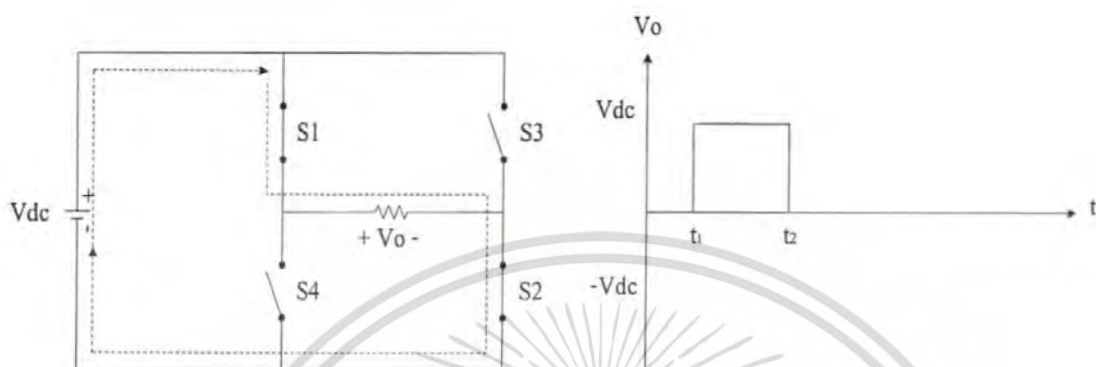
รูปที่ 2.9 วงจรอินเวอร์เตอร์ฟูลบริดจ์



รูปที่ 2.10 วงจรอินเวอร์เตอร์สมมูลจากรูปที่ 2.9

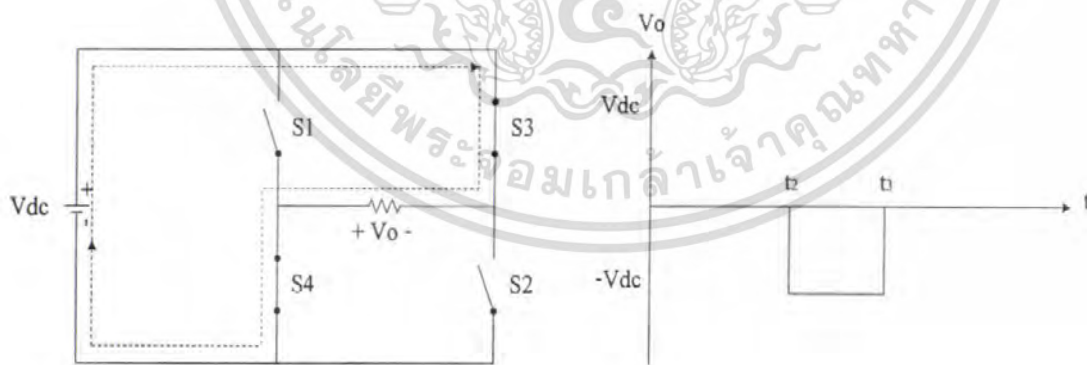
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ S1,S2 ปิด และ S3,S4 เปิด สำหรับ $t_1 < t < t_2$



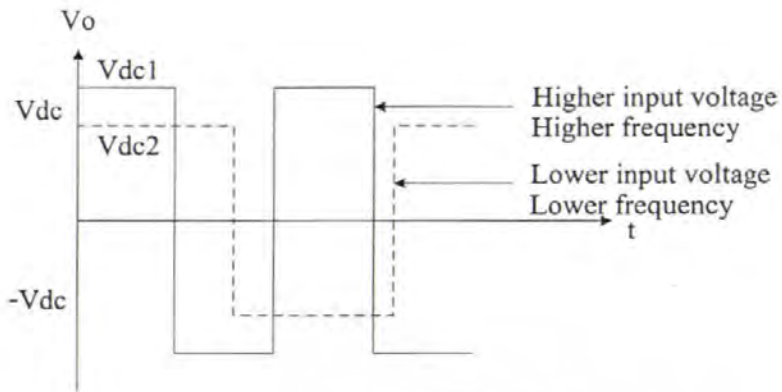
รูปที่ 2.11 สัญญาณขาออกการสวิตซิ่งแรงดันแบบสแควร์ ช่วง $0 - \pi$

เมื่อ S3,S4 ปิด และ S1,S2 เปิด สำหรับ $t_2 < t < t_3$



รูปที่ 2.12 สัญญาณขาออกการสวิตซิ่งแรงดันไฟฟ้าแบบสแควร์ช่วง $\pi - 2\pi$

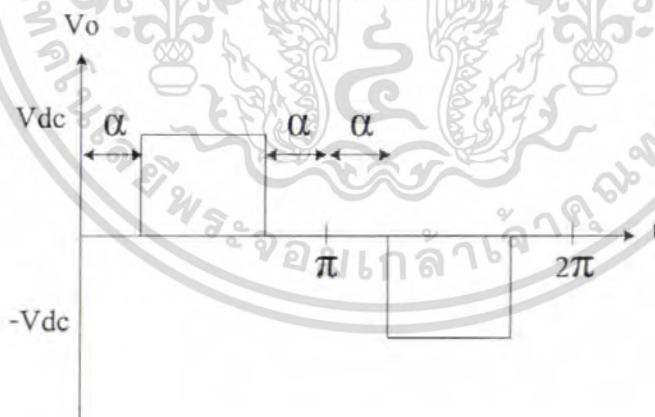
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 สัญญาณแบบสแควร์

2.4.2 การสวิตซ์แรงดันไฟฟ้าแบบควอซีสแควร์

การสวิตซ์แรงดันแบบควอซีสแควร์ (Quasi-square) คือการควบคุมการทำงานของสวิตซ์แบบบริดจ์ให้ทำงานพร้อมกันเป็นคู่เช่นเดียวกับ การสวิตซ์แบบสแควร์ แต่จะแตกต่างกันที่ มุมเฟสเริ่มต้นและสิ้นสุดทุกครั้งคาบจะหน่วงทำมุมอัลฟา (α) ในรูปที่ 2.13 ซึ่งมุมอัลฟานี้จะทำให้สามารถปรับค่ากระแสไฟฟ้าทางด้านออก ได้



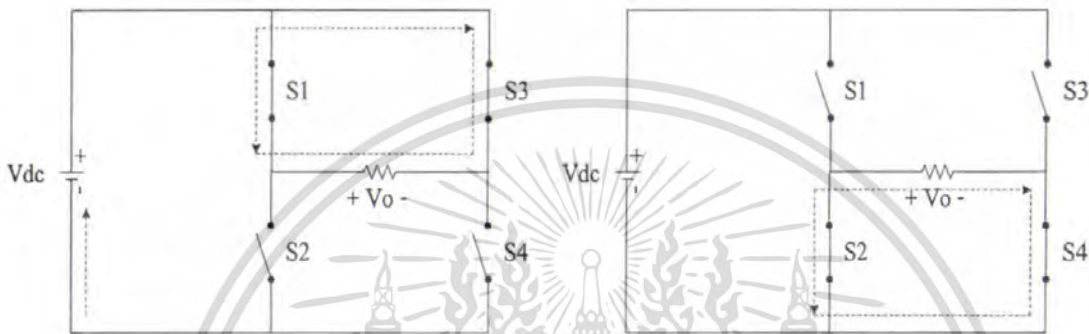
รูปที่ 2.14 สัญญาณแบบควอซีสแควร์

จะเห็นได้ว่าช่วงอัลฟานี้แรงดันเป็นศูนย์ ซึ่งเป็นเหตุที่ทำให้เราสามารถปรับแรงดันขาออกได้ในขณะเดียวกัน ขดลวดเหนี่ยวนำจะเหนี่ยวนำให้เกิดแรงดันย้อนกลับซึ่งจะทำให้รูปคลื่นขาออกผิดเพี้ยน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

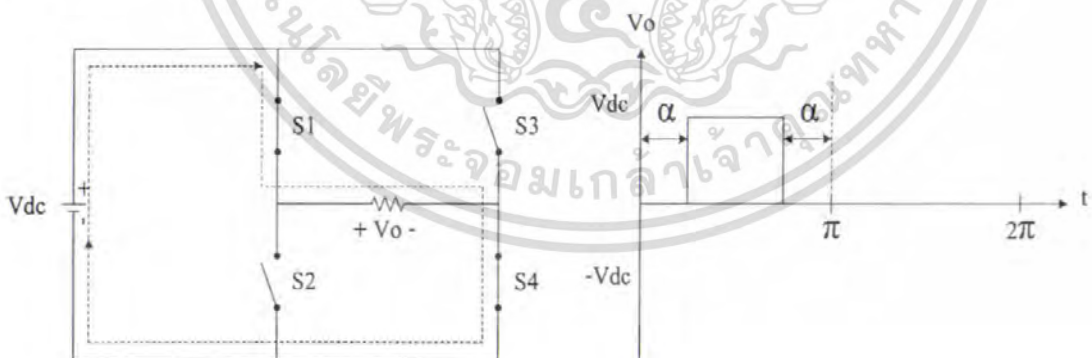
ไป ดังนั้นในขณะที่อยู่ในช่วงอัตรากำลังจึงต้องมีการสวิตช์ในรูปแบบ ฟรีวีลิ่ง (Free Wheeling) เพื่อป้องกันกระแสไฟฟ้าเหนี่ยวนำจากขดลวดเหนี่ยวนำ

เมื่อ S1,S3 ปิด และ S2,S4 เปิด กับ S2,S4 ปิด และ S1,S3 เปิด สำหรับทุก α แรงดันด้านออก $V_o = 0$



รูปที่ 2.15 การทำฟรีวีลิ่ง

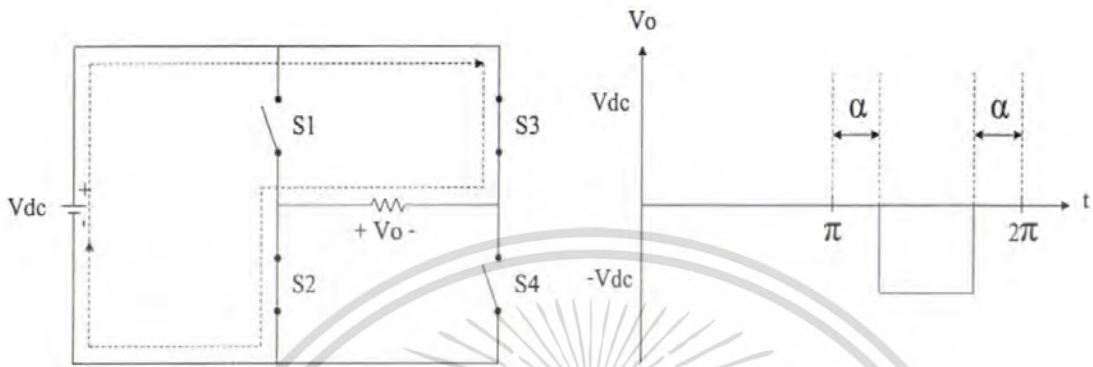
เมื่อ S1,S4 ปิด และ S2,S3 เปิด สำหรับ $\alpha < t < (\pi - \alpha)$



รูปที่ 2.16 สัญญาณแบบควอซีสแควร์ จาก $0 - \pi$

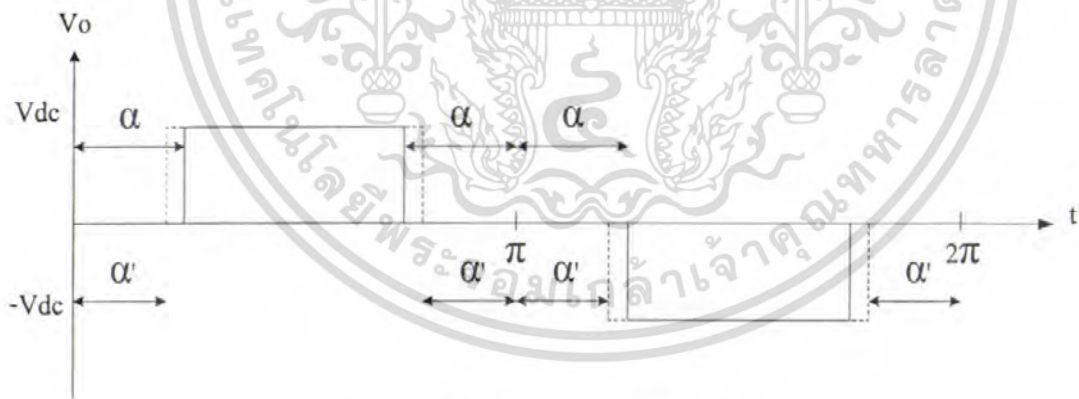
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ S2,S3 ปิด และ S1,S4 เปิด สำหรับ $(\pi + \alpha) < t < (2\pi - \alpha)$



รูปที่ 2.17 สัญญาณแบบควอดซี่สแควร์ จาก $\pi - 2\pi$

รูปคลื่นด้านออกของการสวิตช์แรงดันแบบควอดซี่สแควร์



รูปที่ 2.18 สัญญาณแบบควอดซี่สแควร์

- กระแสไฟฟ้าทางด้านขาออกสูง α น้อย
- กระแสไฟฟ้าทางด้านขาออกต่ำ α มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดี

- การสร้างรูปสัญญาณขาออกง่าย
- มีเสถียรภาพ
- สามารถปรับกระแสไฟฟ้าทางด้านขาออกได้

ข้อด้อย

- มีฮาร์โมนิกส์(Harmonic)อันดับต่ำๆ ที่อยู่ใกล้กับฮาร์โมนิกอันดับหนึ่ง

2.4.3 การสวิตซ์แรงดันไฟฟ้าแบบไบโพลาร์

การสวิตซ์แรงดันไฟฟ้าแบบไบโพลาร์(Bipolar) คือการควบคุมให้สวิตซ์แบบบริดจ์ทำงานพร้อมกันเป็นคู่ เช่น รูปที่ 2.16 การทำงานของสวิตซ์ S1 และ S3 จะถูกควบคุมให้ทำงานพร้อมกันในแต่ละช่วงเวลา อีกคู่หนึ่งคือการทำงานของสวิตซ์ S2 และ S4 ดังนั้นแรงดันไฟฟ้าด้านออก V_o

$V_o = V_d$ เมื่อ $V_{control} > V_{tri}$, สวิตซ์ S1 และ S3 จะนำกระแส

$V_o = -V_d$ เมื่อ $V_{control} < V_{tri}$, สวิตซ์ S2 และ S4 จะนำกระแส

เมื่อคิดจากกึ่งหนึ่งของอินเวอร์เตอร์แบบบริดจ์ แรงดันไฟฟ้าด้านออกของกิ่ง B จะเท่ากับค่าลบของแรงดันไฟฟ้าด้านออกของกิ่ง A คือ $V_{Bo} = -V_{Ao}$ ดังนั้นแรงดันไฟฟ้าด้านออกของอินเวอร์เตอร์หรือแรงดันไฟฟ้าระหว่างกิ่ง A กับกิ่ง B คือ

$$V_o = V_{Ao} = V_{Bo} = 2V_{Ao}$$

$$\hat{V}_{ol} = m_a V_d \text{ เมื่อ } m_a < 1.0 \quad (A)$$

$$V_d < \hat{V}_{ol} < (4/\pi)V_d \text{ เมื่อ } m_a > 1.0 \quad (B)$$

หรืออาจจะสรุปได้ว่าแรงดันไฟฟ้ายอดด้านออกจะเท่ากับสมการที่ A เมื่อ m_a อยู่ในย่านเชิงเส้นและแรงดันไฟฟ้ายอดด้านออกจะเท่ากับสมการที่ B อยู่ในช่วงโอเวอร์มอดูเลชัน

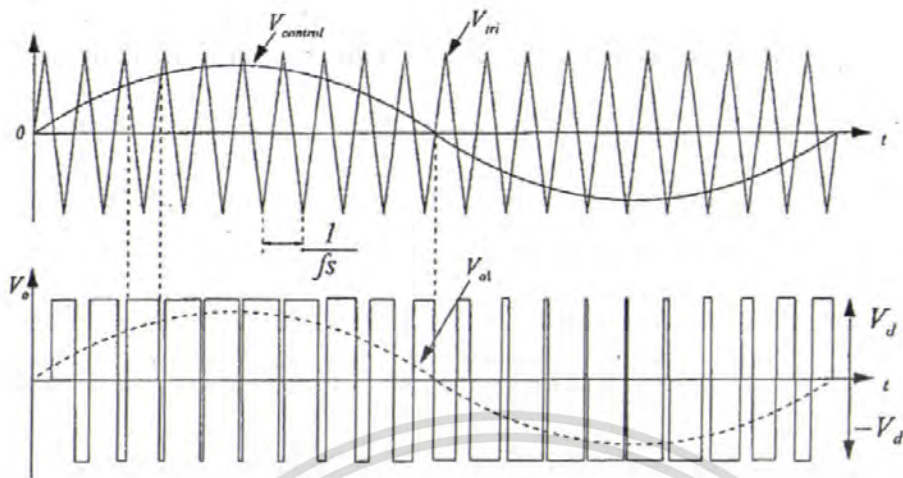
โดยที่แรงดันไฟฟ้าด้านออกจะสวิตซ์อยู่ระหว่าง $+V_d$ กับ $-V_d$ ส่วนไซด์แบนฮาร์โมนิกจะเกิดขึ้นรอบๆ $m_r, 2m_r, 3m_r, \dots$ เช่น หากความถี่สวิตซ์เท่ากับ 5 kHz ไซด์แบนฮาร์โมนิกก็จะเกิดขึ้นที่ 5 kHz, 40 kHz และ 60 kHz เป็นต้น ดังเช่นที่ได้อธิบายในสมการ $h = j(m_r) \pm k$

เมื่อ $h = 1$ คือความถี่หลักมูล

ถ้าค่า j เป็นเลขคี่ ค่าฮาร์โมนิกจะเท่ากับค่า k ที่เป็นเลขคู่

ถ้าค่า j เป็นเลขคู่ ค่าฮาร์โมนิกจะเท่ากับค่า k ที่เป็นเลขคี่

81994



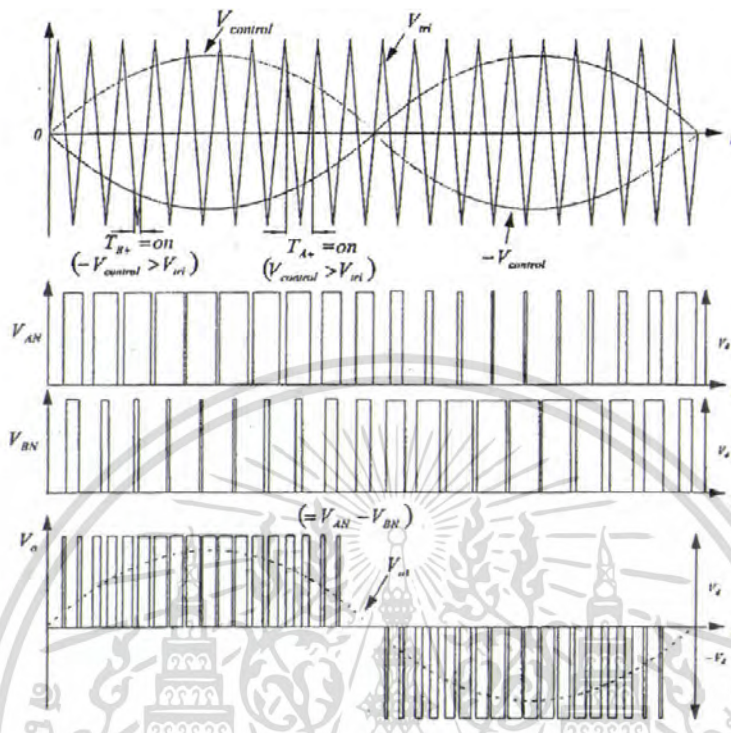
รูปที่ 2.19 การสวิตช์แรงดันแบบไบโพลาร์

สิ่งที่ระบุว่าเป็นการสวิตช์แบบไบโพลาร์ คือ

1. มีการสวิตช์ของแรงดันระหว่างสาย (V_o หรือ V_{ab}) ระหว่างขั้วหรือบัสบวกกับลบ
2. ความถี่ของพัลส์ที่โหลดจะเท่ากับความถี่ของ V_{tri}
3. จะเกิดความถี่จากสเปกตรัม (Spectra) คือ เริ่มต้นที่รอบๆ f_s และจำนวนเท่าของ f_s

2.4.4 การสวิตช์แรงดันไฟฟ้าแบบยูนิโพลาร์

จุดเด่นของยูนิโพลาร์(Unipolar) คือ การเกิดความถี่ด้านออกจะมีความถี่เป็นสองเท่าของความถี่สวิตช์ของแต่ละเฟส เช่น ความถี่สวิตช์ของอุปกรณ์เป็น 10 kHz หากเป็นการทำงานแบบยูนิโพลาร์ ค่าแรงดันออกจะประกอบไปด้วยสัญญาณพัลส์เบิ้ลยูเอ็ม ที่มีความถี่หลักมูลและความถี่ฮาร์มอนิกที่ 2 เท่าของแบบไบโพลาร์ที่ 20 kHz ผลของความถี่ที่โหลดเพิ่มขึ้นเป็นสองเท่า ทำให้การออกแบบวงจรกรองความถี่ เล็ก ง่าย และ ประหยัด



รูปที่ 2.20 การสวิตชิงแรงดันแบบยูนิโพลาร์

2.5 การคำนวณหาค่าฮาร์มอนิกส์

การวิเคราะห์อนุกรมฟูเรียร์ (Fourier Series) เป็นวิธีที่ใช้มากที่สุดใน การวิเคราะห์ ค่าความถี่ฮาร์มอนิกส์ของค่าแรงดันขาออกของอินเวอร์เตอร์ ค่าแรงดันขาออกจะอธิบายในรูปแบบของอนุกรมฟูเรียร์ ด้วยความถี่มูลฐานและ ฮาร์มอนิกส์ ผลที่ได้จะอยู่ในรูปของ ขนาดและเฟสแต่ละสัญญาณ โดยส่วนประกอบที่ได้ ง่ายต่อการนำไปใช้งาน อนุกรมฟูเรียร์สำหรับสัญญาณคาบแสดงได้ดังนี้

$$f(v) = \frac{1}{2} a_0 + \sum_{n=1}^{\infty} (a_n \cos n\theta + b_n \sin n\theta)$$

$$\text{where } \theta = \omega t$$

เมื่อ $f(v)$ เป็นสัญญาณคาบ ที่มีคาบเป็น 2π ค่า a_n , b_n สามารถคำนวณ โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$a_n = \frac{1}{\pi} \int_0^{2\pi} f(v) \cos(n\theta) d\theta$$

$$b_n = \frac{1}{\pi} \int_0^{2\pi} f(v) \sin(n\theta) d\theta$$

องค์ประกอบความผิดเพี้ยนของฮาร์มอนิกส์รวม (Total Harmonic Distortion) (THD) เป็นที่นิยมในการใช้เป็นตัวชี้วัดคุณภาพของสัญญาณแรงดันไฟฟ้า และกระแสไฟฟ้าขาออก องค์ประกอบฮาร์มอนิกส์รวมแสดง จำนวนของสัญญาณที่ผิดเพี้ยนของแรงดันไฟฟ้าขาออกดังนี้

$$THD_v = \frac{\sqrt{\sum_{n=2}^{\infty} (V_{n,RMS})^2}}{V_{1,RMS}} = \frac{\sqrt{\sum_{n=2}^{\infty} (V_{RMS})^2 - (V_{1,RMS})^2}}{V_{1,RMS}}$$

$$THD_i = \frac{\sqrt{\sum_{n=2}^{\infty} (I_{n,RMS})^2}}{I_{1,RMS}}$$

เมื่อ $(V_{n,RMS})$ คือ ค่ารากที่สองเฉลี่ยของ ฮาร์มอนิกส์แต่ละอันดับของ $V_o(t)$ ขณะที่ $(V_{1,RMS})$ คือ ค่ารากที่สองเฉลี่ยของส่วนประกอบความถี่มูลฐาน เพื่อที่จะลดฮาร์มอนิกส์ที่ไม่ต้องการของ สัญญาณขาออกของอินเวอร์เตอร์ การสังเคราะห์สัญญาณ ที่ดับเบิ้ลยูเอ็มจึงสามารถหาขนาดของแต่ละฮาร์มอนิกส์ ด้วยวิธีการวิเคราะห์ฟูเรียร์

2.6 การควบคุมความเร็วรอบของมอเตอร์

เนื่องจากความเร็วรอบของมอเตอร์เหนี่ยวนำ จะเปลี่ยนแปลงสัมพันธ์กับสมการความเร็วรอบหรือสมการซิง โกรนัส-สปีด ดังต่อไปนี้

$$\text{Synchronous speed } (N_s) = (120 \times f) / P$$

โดยกำหนดให้

f คือ ความถี่กระแสไฟฟ้า

P คือ จำนวนขั้วแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการสมชิ่งโครน์ส-สปีด จะเห็นว่าความเร็วรอบของมอเตอร์สามารถปรับเปลี่ยนได้ 2 ทาง คือ เปลี่ยนจำนวนขั้วแม่เหล็ก (P) และเปลี่ยนแปลงความถี่ของกระแสไฟฟ้าที่จ่ายให้กับมอเตอร์ไฟฟ้า (f)

วิธีการควบคุมความเร็วรอบด้วยการเปลี่ยนจำนวนขั้วแม่เหล็กนั้น ความเร็วจะเปลี่ยนแปลงไปครั้งละมาก ๆ ซึ่งการเปลี่ยนแปลงความเร็วรอบในลักษณะนี้ความเร็วรอบที่เปลี่ยนแปลงจะไม่ละเอียด และที่สำคัญคือต้องใช้มอเตอร์ที่ออกแบบพิเศษที่สามารถเปลี่ยนแปลงจำนวนขั้วแม่เหล็กได้เท่านั้น ทำให้ไม่เหมาะสมกับความต้องการของงานในหลายๆ ประเภทที่ต้องการควบคุมความเร็วรอบในขณะที่มีโหลดเพื่อให้ความเร็วเหมาะสมกับความเร็วของกระบวนการผลิต ดังนั้นในกระบวนการผลิตทั่วไปจึงนิยมใช้อินเวอร์เตอร์ในการควบคุมความเร็วรอบของมอเตอร์มากกว่าเนื่องจากสามารถควบคุมให้มอเตอร์หมุนด้วยความเร็วคงที่ ปรับความเร็วรอบไปที่ความเร็วต่างๆ ได้อย่างรวดเร็วและมีความเที่ยงตรงมากกว่า

2.6.1 การควบคุมความเร็วโดยการควบคุมความถี่

ในการควบคุมความเร็วโดยวิธีนี้จะกล่าวถึงเพียงการควบคุมแบบ V/f คงที่ กล่าวคือ เมื่อเราลดความถี่ของแรงดันสเตเตอร์ลง เราจะต้องลดแรงดันของสเตเตอร์ตามไปด้วยเพื่อให้อัตราส่วนของแรงดันกับความถี่ (V/f) มีค่าคงที่ ถ้าไม่เช่นนั้นจะทำให้แกนเหล็กในสเตเตอร์อิ่มตัว ซึ่งจะทำให้มีกระแสไหลมากผิดปกติและอาจจะทำให้มอเตอร์เสียหายได้ การทำให้อัตราส่วน V/f คงที่ จะทำให้สนามแม่เหล็กในช่องอากาศระหว่างสเตเตอร์และโรเตอร์มีค่าคงที่ ซึ่งจะทำให้แรงบิดสูงสุดของมอเตอร์มีค่าคงที่ด้วย

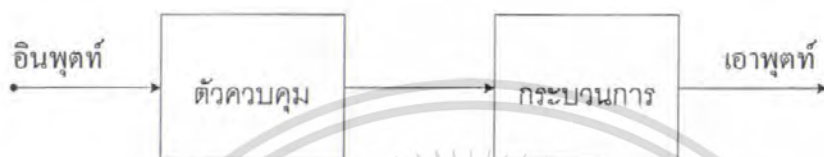
อย่างไรก็ดีที่ความถี่ต่ำเราจำเป็นต้องเพิ่มอัตราส่วนของ V/f เพื่อชดเชยความต้านทานแรงดันที่ตกคร่อมค่าอิมพีแดนซ์ (Impedance) ของขดลวดสเตเตอร์ สำหรับในย่านความถี่สูงกว่าค่ากำหนด เราจะไม่เพิ่มแรงดันของสเตเตอร์ขึ้นไปซึ่งจะทำให้สนามแม่เหล็กในช่องอากาศระหว่างสเตเตอร์และโรเตอร์มีค่าลดลง แรงบิดของมอเตอร์ก็จะลดลงตามด้วย สำหรับอุปกรณ์ที่ใช้ในการควบคุมความเร็วมอเตอร์ด้วยวิธีนี้จะใช้อุปกรณ์ที่เรียกว่า "อินเวอร์เตอร์"

2.7 ประเภทของการควบคุม

ประเภทของการควบคุมแบ่งออกได้เป็น 2 ลักษณะใหญ่ๆ คือ ระบบควบคุมแบบวงเปิด และระบบควบคุมแบบวงปิด

2.7.1 ระบบควบคุมแบบวงเปิด

ระบบควบคุมแบบวงเปิด หรือเรียกอีกอย่างหนึ่งว่าระบบควบคุมแบบไม่มีการป้อนกลับ ระบบนี้เป็นระบบที่ง่ายที่สุดและมีอุปกรณ์ไม่ยุ่งยาก ค่าเอาต์พุตที่ได้จะไม่มีผลต่อการควบคุมกระบวนการของระบบ หมายความว่า ไม่มีการนำเอาค่าเอาต์พุตที่ได้กลับมาเปรียบเทียบกับค่าอินพุต (ค่าอ้างอิง) ที่ป้อนให้กับระบบ ซึ่งสามารถแสดงลักษณะบล็อกไดอะแกรมของระบบควบคุมแบบเปิด



รูปที่ 2.21 แสดงบล็อกไดอะแกรมของระบบควบคุมแบบวงเปิด

ในที่นี้จะขอยกตัวอย่างการควบคุมแบบวงเปิด ซึ่งเป็นการควบคุมความเร็วมอเตอร์ ซึ่งเขียนบล็อกไดอะแกรมได้ดังรูป



รูปที่ 2.22 แสดงบล็อกไดอะแกรมของระบบควบคุมความเร็วมอเตอร์แบบวงเปิด

จากรูป แสดงถึงความสัมพันธ์ของระบบระหว่างอินพุต คือ ค่าความเร็วรอบที่ต้องการ และเอาต์พุต คือ ค่าความเร็วรอบจริง ทิศทางของลูกศรที่พุ่งเข้าหาระบบควบคุมแสดงถึงอินพุต ส่วนลูกศรที่มีทิศพุ่งออก แสดงถึงเอาต์พุต จากบล็อกไดอะแกรม จะเห็นว่าไม่มีสัญญาณใดๆ ของระบบ ป้อนกลับไปยังอินพุตเลย ดังนั้นจึงเรียกระบบควบคุมลักษณะนี้ว่า ระบบควบคุมแบบเปิด

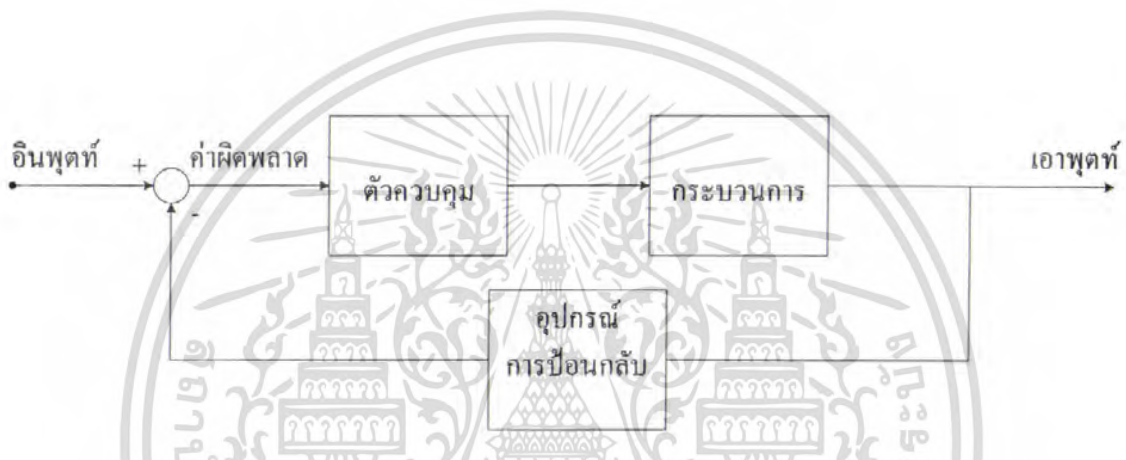
ข้อสังเกตของระบบควบคุมแบบวงเปิด คือ

- ไม่มีสัญญาณทางด้านเอาต์พุตป้อนกลับมายังอินพุต
- ระบบไม่มีความซับซ้อน
- ระบบใช้กับงานที่ไม่ต้องการความแม่นยำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.2 ระบบควบคุมแบบวงปิด

เมื่อระบบควบคุมแบบเปิดไม่สามารถให้ความแม่นยำในการควบคุมได้ ดังนั้นจึงมีการพิจารณาที่จะนำสัญญาณเอาต์พุตซึ่งมีผลโดยตรงกับระบบควบคุมมาใช้งาน โดยการนำสัญญาณเอาต์พุตของระบบป้อนกลับมาเปรียบเทียบกับสัญญาณอินพุตที่ป้อนให้กับระบบ ซึ่งผลต่างระหว่างสัญญาณทั้งสองที่นำมาเปรียบเทียบนั้นจะเป็นสัญญาณค่าผิดพลาด (error) เพื่อที่จะใช้เป็นสัญญาณป้อนเข้าตัวควบคุมให้ตัวควบคุมนำไปสร้างสัญญาณควบคุมใหม่เพื่อลดความผิดพลาดที่เกิดขึ้นในระบบและทำให้เอาต์พุตของระบบเข้าสู่ค่าที่เราต้องการ (set point) แสดงได้ ดังรูป



รูปที่ 2.23 แสดงบล็อกไดอะแกรมของระบบควบคุมแบบวงปิด

ตัวอย่างของระบบควบคุมแบบวงปิด ในที่นี้จะยกตัวอย่าง การควบคุมความเร็วรอบของมอเตอร์ ซึ่งสามารถเขียนเป็นบล็อกไดอะแกรม ได้ดังรูป



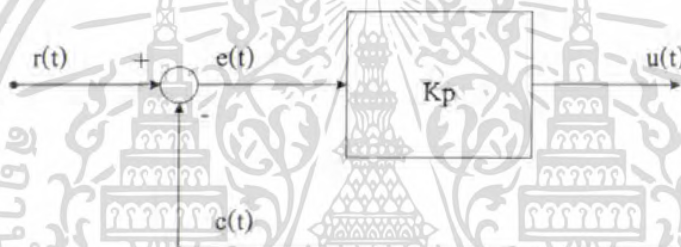
รูปที่ 2.24 แสดงบล็อกไดอะแกรมของระบบควบคุมความเร็วมอเตอร์แบบวงปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปเป็นระบบควบคุมความเร็วรอบของมอเตอร์แบบวงปิด โดยมีเอนโคเดอร์ทำหน้าที่เป็น ตัว feedback ค่าความเร็วรอบจริงกลับมาเปรียบเทียบกับค่าความเร็วรอบที่ต้องการ เกิดเป็นค่า error ที่ป้อนให้กับตัวควบคุม จากนั้นตัวควบคุมจะประมวลผลและส่งสัญญาณไปสั่งให้วงจรขับมอเตอร์ทำงานจนกว่า จะได้ค่าความเร็วรอบจริงเท่ากับค่าความเร็วรอบที่ต้องการ ซึ่งก็คือค่า error เป็นศูนย์

2.7.3 ตัวควบคุมแบบ Proportional (แบบ P)

ค่าเอาต์พุตของตัวควบคุมจะแปรผันตรงกับค่าความคลาดเคลื่อน คือ ค่าความคลาดเคลื่อนมีค่า มากขึ้น ค่าเอาต์พุตของตัวควบคุมจะมีค่ามากขึ้นตาม และถ้าค่าความคลาดเคลื่อนมีค่าน้อยลง ค่า เอาต์พุตของตัวควบคุมก็จะมีค่าน้อยลงตาม ตัวควบคุมแบบ P สามารถแสดงความสัมพันธ์ในรูปของ บล็อกไดอะแกรม ได้ดังรูป



รูปที่ 2.25 แสดงบล็อกไดอะแกรมการควบคุมแบบสัดส่วน

และสามารถเขียนเป็นสมการความสัมพันธ์ระหว่างสัญญาณควบคุม (หรือเอาต์พุตของตัวควบคุม) $U(t)$ กับสัญญาณค่าความคลาดเคลื่อน $e(t)$ คือ

$$U(t) = K_p e(t)$$

$$\text{หรือ } \frac{U(s)}{E(s)} = K_p$$

เมื่อ $U(t)$ = ค่าเอาต์พุตของตัวควบคุมแบบ P

K_p = อัตราการขยายตัวควบคุมแบบ P

ข้อดีของตัวควบคุมแบบ P คือ ใช้งานง่ายไม่ยุ่งยาก

ข้อเสียของตัวควบคุมแบบ P คือ ไม่สามารถกำจัดค่า Offset ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

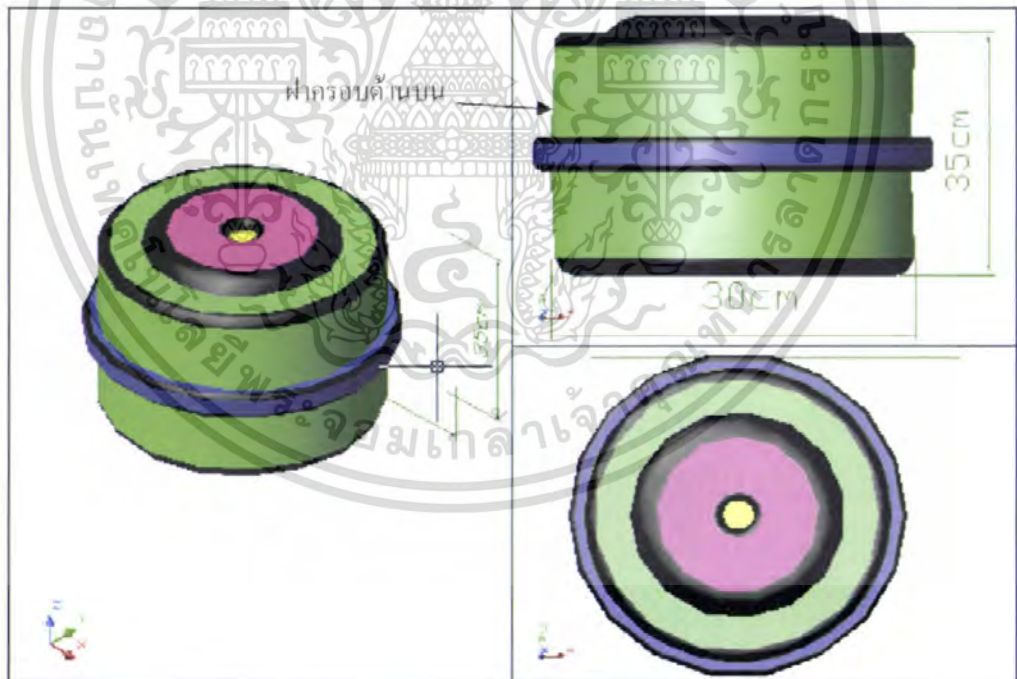
หลักการออกแบบ

3.1 การออกแบบโครงสร้างภายนอกและภายในของเครื่องบินเหวี่ยงเลือด

การออกแบบเครื่องบินเหวี่ยงเลือดนี้ ทางโครงการได้เลือกใช้โปรแกรมอโต้แคดสำหรับการออกแบบ โดยทำการออกแบบโครงสร้างภายนอก โครงสร้างภายใน และที่ใส่หลอดเลือดของเครื่องบินเหวี่ยงเลือดในรูปแบบ 3 มิติ ซึ่งมีรายละเอียดของการออกแบบ ดังต่อไปนี้

3.1.1 การออกแบบโครงสร้างภายนอกของเครื่องบินเหวี่ยงเลือด

ลักษณะโครงสร้างภายนอกนั้น ได้ทำการออกแบบให้มีลักษณะเป็นรูปทรงกระบอก มีฝาครอบสามารถเปิดได้ ตัวเครื่องบินเหวี่ยงเลือดมีขนาดเส้นผ่านศูนย์กลางเท่ากับ 30 เซนติเมตร และมีความสูงเท่ากับ 35 เซนติเมตร ดังรูปที่ 3.1

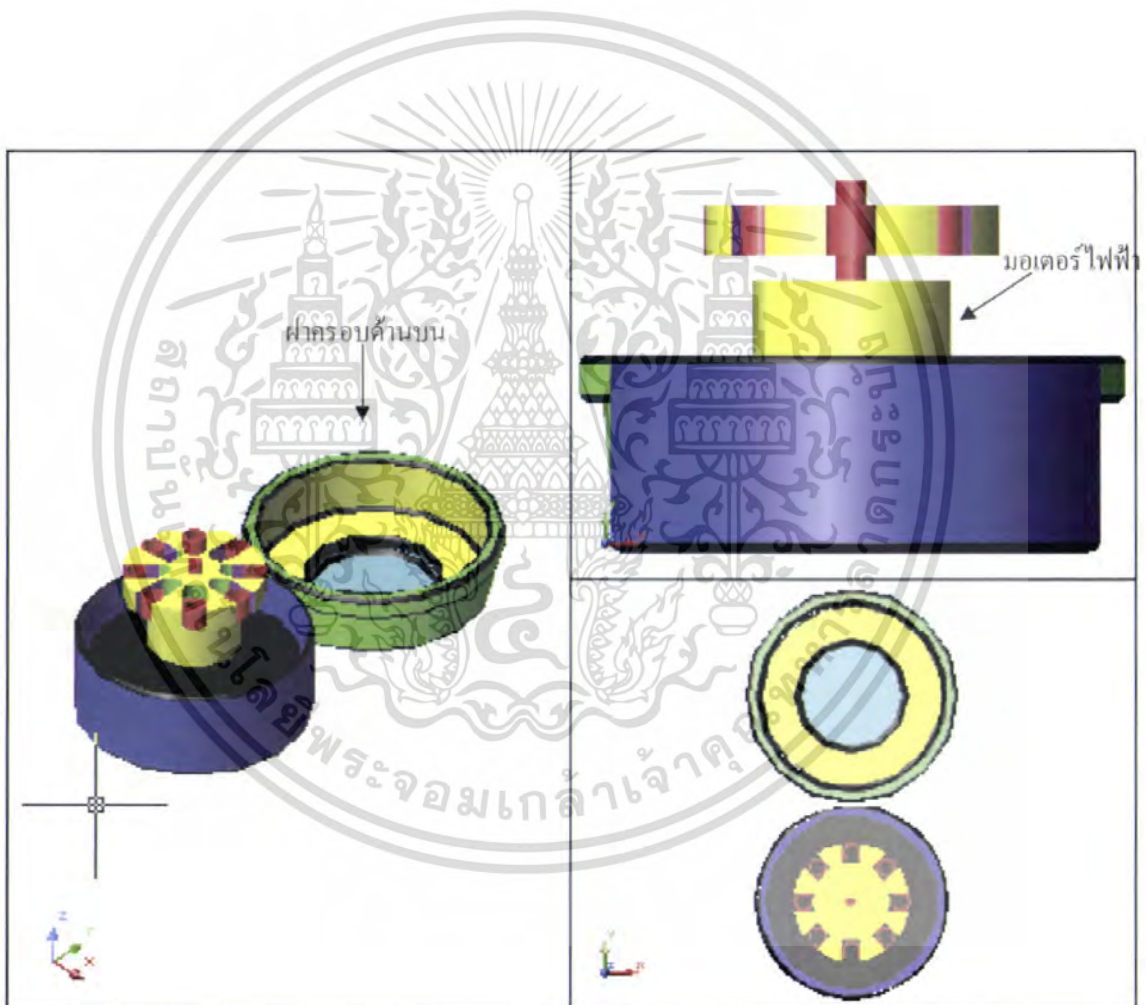


รูปที่ 3.1 แสดงโครงสร้างภายนอกของเครื่องบินเหวี่ยงเลือด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 การออกแบบโครงสร้างภายในของเครื่องปั้นหึ่งเลือด

ลักษณะโครงสร้างภายในนั้น ได้ทำการออกแบบในลักษณะที่เมื่อทำการเปิดฝาครอบด้านบน ออก จะมองเห็น โครงสร้างภายในของเครื่องปั้นหึ่งเลือด มีส่วนประกอบที่สำคัญ คือ ที่ใส่หลอดเลือด ซึ่งติดอยู่บนแกนที่ต่อกับมอเตอร์ไฟฟ้าซึ่งอยู่ด้านล่างของเครื่องปั้นหึ่งเลือด ดังรูปที่ 3.2 โดยที่ใส่หลอดเลือดนี้จะหมุนด้วยความเร็วรอบที่มอเตอร์หมุนไป ทำให้เกิดการหมุนหึ่งขึ้นของเครื่องปั้นหึ่งเลือด

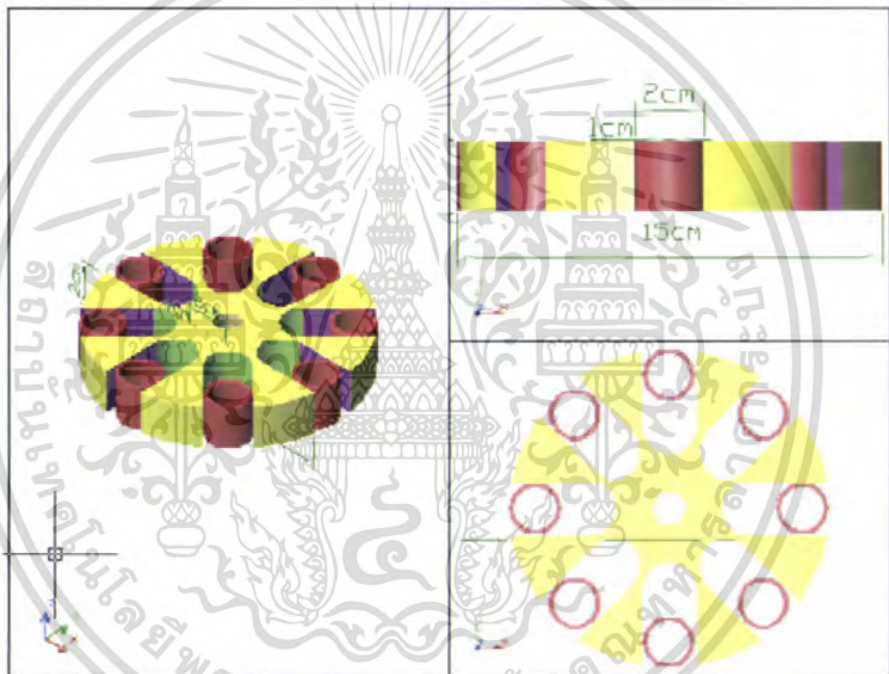


รูปที่ 3.2 แสดงโครงสร้างภายในของเครื่องปั้นหึ่งเลือด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 การออกแบบที่ใส่หลอดเลือด

ที่ใส่หลอดเลือด คือสิ่งที่ใช้สำหรับบรรจุภาชนะตัวอย่าง ซึ่งก็คือหลอดเลือดนั่นเอง โดยได้ทำการออกแบบให้สามารถบรรจุหลอดเลือดได้จำนวน 8 หลอด และมีลักษณะเป็นแบบวงกลม มีขนาดของเส้นผ่านศูนย์กลาง 15 เซนติเมตร ตรงกลางเจาะเป็นรู สำหรับใช้ประกอบเข้ากับแกนที่ต่อกับตัวมอเตอร์ไฟฟ้า โดยมีขนาดเส้นผ่านศูนย์กลางเท่ากับ 1 เซนติเมตร ส่วนช่องที่ใช้สำหรับใส่หลอดเลือดทั้ง 8 ช่องนั้น ในแต่ละช่องมีเส้นผ่านศูนย์กลางเท่ากับ 2 เซนติเมตร



รูปที่ 3.3 แสดงที่ใส่หลอดเลือด

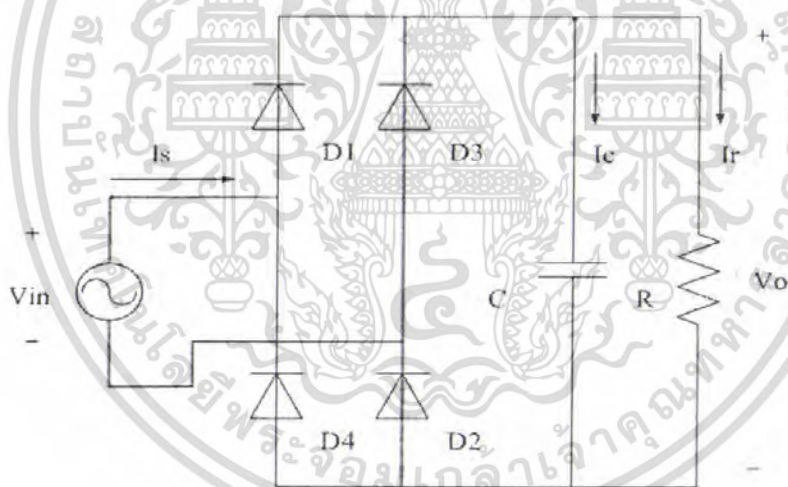
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบระบบขับเคลื่อนมอเตอร์เหนี่ยวนำแบบเฟสเดียว

การทำงานของระบบขับเคลื่อนมอเตอร์เหนี่ยวนำแบบเฟสเดียว ประกอบไปด้วย วงจรเรียงกระแส วงจรสวิตซ์ซึ่ง วงจรควบคุมการทำงานของไอจีบีที และวงจรอุปโต คัปเปิล ซึ่งทางโครงการได้นำไมโครคอนโทรลเลอร์ มาประยุกต์ใช้สำหรับขับนำ ไอจีบีที โดยไมโครคอนโทรลเลอร์ นี้จะสร้างสัญญาณเลียนแบบสัญญาณจากวงจรรอนาล็อก(Analog) ซึ่งไมโครคอนโทรลเลอร์ที่จะสร้างสัญญาณนี้ได้ ต้องมีโมดูล(Module) ไว้ทำหน้าที่สร้างสัญญาณนี้โดยเฉพาะ ซึ่งทางโครงการได้เลือกใช้ ไมโครคอนโทรลเลอร์ dsPIC30F4011 และใช้เป็นตัวควบคุมความเร็วรอบของมอเตอร์ด้วย

3.2.1 วงจรเรียงกระแส

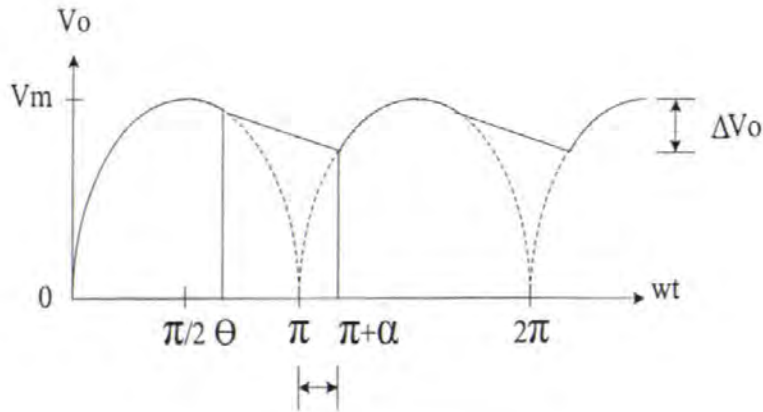
วงจรเรียงกระแสเป็นวงจรที่เปลี่ยนแรงดันไฟฟ้ากระแสสลับให้เป็นแรงดันไฟฟ้ากระแสตรง สร้างได้จากการนำไดโอดสี่ตัว ที่สามารถทนแรงดันที่พิกัดที่เราทำการทดลองได้มาต่อแบบบริดจ์ แต่ในการทดลองใช้ไดโอดบริดจ์แบบสำเร็จรูป เพื่อง่ายต่อการนำมาใช้งานและเพื่อความคงทน



รูปที่ 3.4 วงจรเรียงกระแสพร้อมตัวกรองแรงดัน โดยใช้ตัวเก็บประจุ

เมื่อกระแสไหลผ่านไดโอดบริดจ์และผ่านตัวกรอง ซึ่งก็คือตัวเก็บประจุที่อยู่ในวงจร สัญญาณเอาท์พุท ที่ได้จะเป็นสัญญาณแรงดันไฟฟ้ากระแสตรง ที่มีริบเปิ้ล(Ripple) อยู่ด้วย ซึ่งขนาดของริบเปิ้ลจะเล็กหรือใหญ่ขึ้นอยู่กับค่าตัวเก็บประจุที่เราเลือกใช้ โดยกราฟของแรงดันเมื่อผ่านไดโอดบริดจ์และผ่านตัวกรองแสดงตามรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



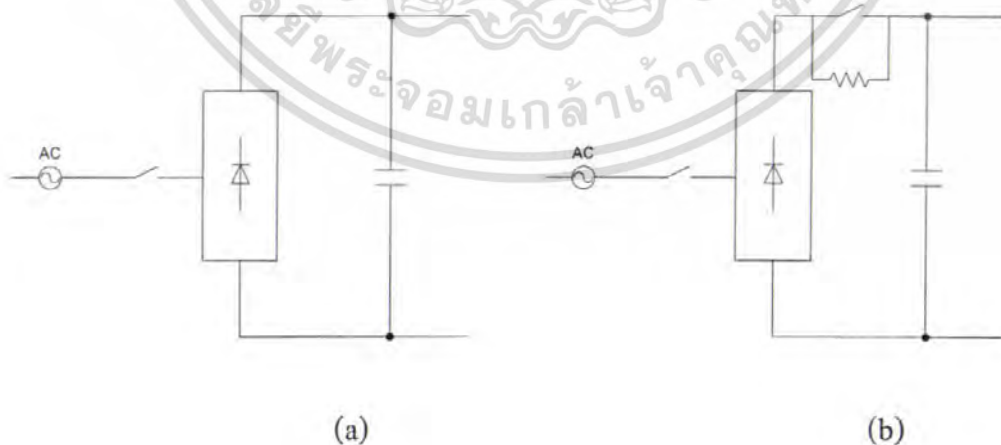
รูปที่ 3.5 กราฟแสดงแรงดันด้านเอาต์พุตเมื่อผ่าน ไดโอดบริดจ์และตัวกรอง

สูตรที่ใช้ในการคำนวณค่าเก็บประจุที่ค่าที่สุดที่นำมาใช้ในวงจรเรียงกระแสได้ คือ

$$C_{\min} = [P_{\text{out}} (1 - 2 f_{\text{net}} t_{\text{on}})] / [2 f_{\text{net}} V_{\text{dc(max)}}^2 \Delta V_0]$$

P_{out} คือ กำลังด้าน output ของมอเตอร์

ข้อควรระวังของการทำวงจรเรียงกระแส คือ การหลีกเลี่ยงกระแสพุ่ง วงจรเรียงกระแสจะเกิดกระแสในช่วงเริ่มต้นทำงานมีค่าสูงมากเนื่องจาก ขณะนั้นตัวเก็บประจุจะดึงกระแสในปริมาณสูงมาก ดังรูปที่ 3.3(a) ถ้าไม่มีการป้องกันก็จะเกิดอันตรายได้ ซึ่งวิธีป้องกันสามารถทำได้โดยต่อตัวต้านทานเพื่อต้านกระแสในช่วงเริ่มต้น ดังรูปที่ 3.3(b) และเมื่อเวลาผ่านไปจึงชอร์ต(Short) ตัวต้านทานนั้นทิ้ง ก็จะสามารถป้องกันการเกิดกระแสพุ่งสูงในช่วงเริ่มต้นของการทำงานได้

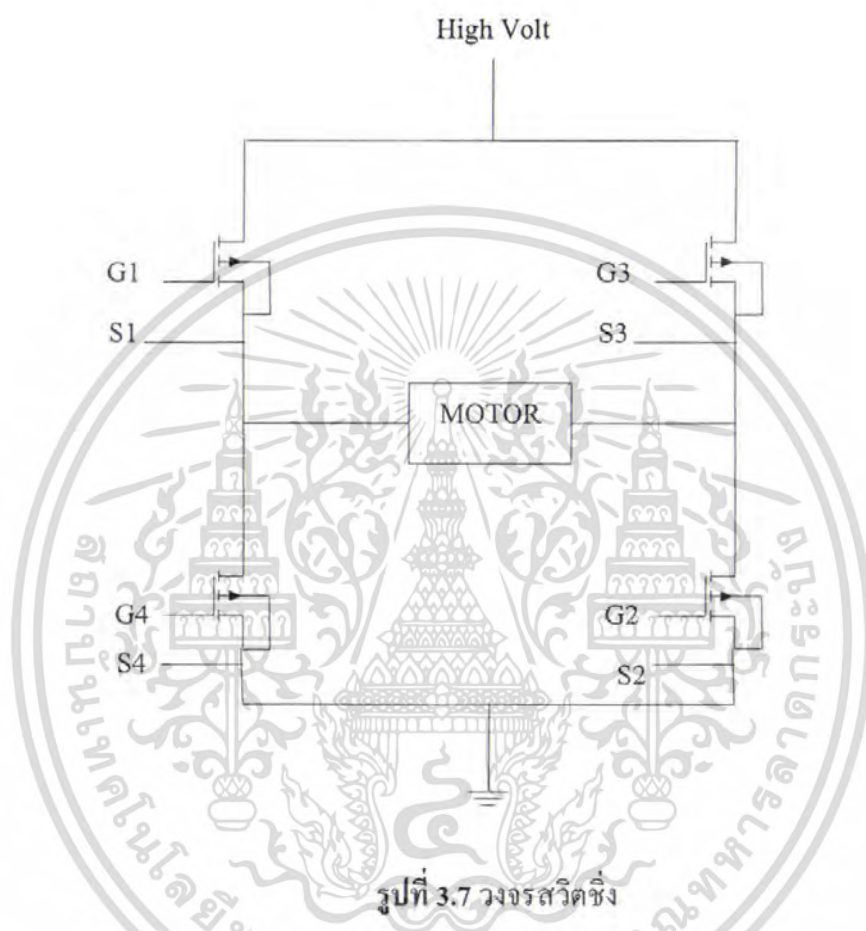


รูปที่ 3.6 วงจรป้องกันกระแสพุ่งสูงในช่วงเริ่มต้นทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 วงจรสวิตซ์

วงจรสวิตซ์ เป็นวงจรที่เปลี่ยนแรงดันไฟฟ้ากระแสตรงที่รับมาจากวงจรเรียงกระแส โดยเปลี่ยนเป็นแรงดันไฟฟ้ากระแสสลับ โดยอาศัยการสวิตซ์ของ ไอจีบีที ที่ทำหน้าที่เป็นสวิตซ์เปิดและปิดการทำงาน โดยนำ ไอจีบีที มาต่อแบบฟูลบริดจ์ ดังรูปที่ 3.4



รูปที่ 3.7 วงจรสวิตซ์

การทำงานของ ไอจีบีที จะทำงานเป็นคู่ G1,G2 จะทำงานพร้อมกัน และ G3,G4 ก็จะทำงนพร้อมกันดังที่กล่าวไว้แล้วในบทที่ 2 โดยที่ ไอจีบีที ในกิ่งเดียวกันห้ามทำงานพร้อมกัน เพราะจะทำให้เกิดการลัดวงจร และวงจรเกิดความเสียหายได้

3.2.3 การควบคุมการทำงานของ ไอจีบีที

การควบคุมการทำงานของ ไอจีบีที เราต้องควบคุมแรงดันระหว่างขา G และขา E ของ ไอจีบีที ให้มีค่าเท่ากับ 10-15 โวลต์ ตามที่คาด้าชีท(Datasheet) แนะนำให้ใช้ ซึ่งหมายความว่า G1,G2,G3,G4 จะรับแรงดันขาเข้าจากวงจรขับเท่ากับ 15 โวลต์ ส่วน S1,S2,S3,S4 ต้องมีค่าใกล้เคียงกับกราวด์(Ground) คือ 0 โวลต์ ซึ่ง S2 กับ S4 ไม่ใช่ปัญหา เพราะว่ามันต่อเทียบกราวด์

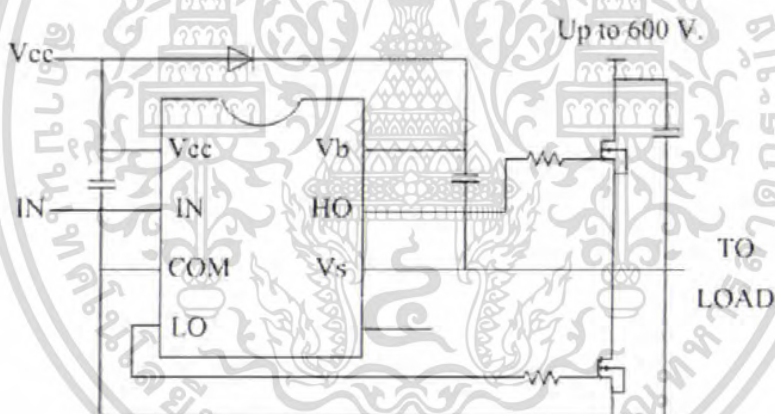
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อยู่แล้ว ปัญหาอยู่ที่ S1 กับ S3 ที่เราต้องทำให้มันมีค่าใกล้เคียงกับกราวด์ จึงจะสามารถควบคุมแรงดันระหว่างขา G กับขา E ของ ไอจีบีที ให้มีค่าเท่ากับ 15 โวลต์ ได้

โครงการนี้เลือกใช้ไอซี(IC) เบอร์ IR2111 ซึ่งมีคุณสมบัติที่สามารถรองรับปัญหาที่เกิดขึ้นนี้ได้ โดยจะอธิบายการทำงานของไอซี เบอร์ IR2111 ในหัวข้อต่อไป

3.2.4 วงจรควบคุมการทำงานของไอจีบีที

การควบคุมการทำงานของไอจีบีที มีไอซีหลายตัวที่สามารถนำมาควบคุมการทำงานของ ไอจีบีทีได้ ซึ่งในการทดลองนี้เลือกใช้ไอซีเบอร์ IR2111 ซึ่งเป็นไอซีที่ใช้ควบคุม ไอจีบีที หรือ พาวเวอร์ มอสเฟต(Power Mosfet) แต่เป็นการควบคุมแบบฮาล์ฟบริดจ์ ซึ่งจะควบคุม ไอจีบีที ด้าน ไฮไซด์(High Side) และ โลว์ไซด์(Low Side) ในกึ่งเดียวกันไม่ให้ทำงานพร้อมกันเนื่องจาก IR2111 มีการหน่วงเวลาในตัวให้ เพื่อป้องกันไอจีบีที ในกึ่งเดียวกันทำงานพร้อมกัน โครงการนี้ทำการต่อวงจรแบบฟูลบริดจ์ ดังนั้นต้องใช้ IR2111 สองตัวเพื่อจะควบคุม ไอจีบีทีสี่ตัว ซึ่งวงจร IR2111 แสดงดังรูปที่ 3.5

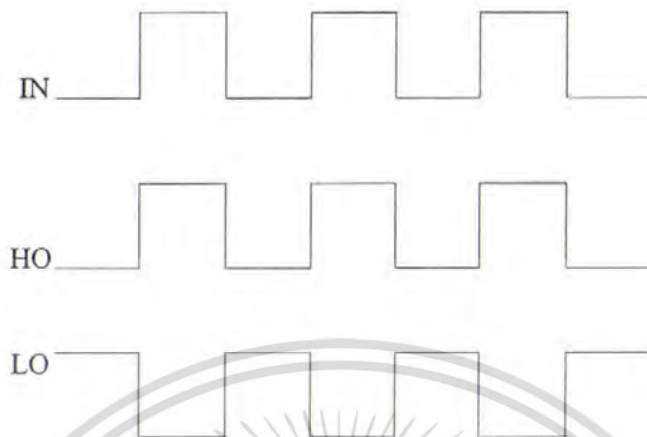


รูปที่ 3.8 วงจรควบคุมการทำงานของไอจีบีที โดยใช้ไอซีเบอร์ IR2111

การทำงานของไอซี เบอร์ IR2111 อธิบายตามรูปที่ 3.5 จากรูปจะพบว่าเมื่อรับสัญญาณเข้ามาด้าน อินพุต เป็นค่าไฮ(High) ซึ่งรับมาจากวงจรออปโต คัปเปิล ในโครงการนี้สัญญาณไฮ ที่รับจากวงจรออปโต คัปเปิลมีค่าเท่ากับ 15 โวลต์ เข้ามายังขา IN ของ IR2111 จะทำให้ เอาท์พุทออกจากขา HO ของ IR2111 มีค่าเท่ากับ 15 โวลต์ ส่วนขา V_s จะเป็นเสมือนกราวด์ ซึ่งมีค่าใกล้เคียงกราวด์คือ ประมาณ 0.3 โวลต์ ดังนั้นแรงดันที่ขา G กับขา E ของ ไอจีบีที ตัวบนจะมีค่าประมาณ 15 โวลต์ นั่นคือไอจีบีทีตัวบนทำงาน ในทางกลับกันถ้า IR2111 รับสัญญาณเข้าขา IN เป็น 0 โวลต์ ขา HO

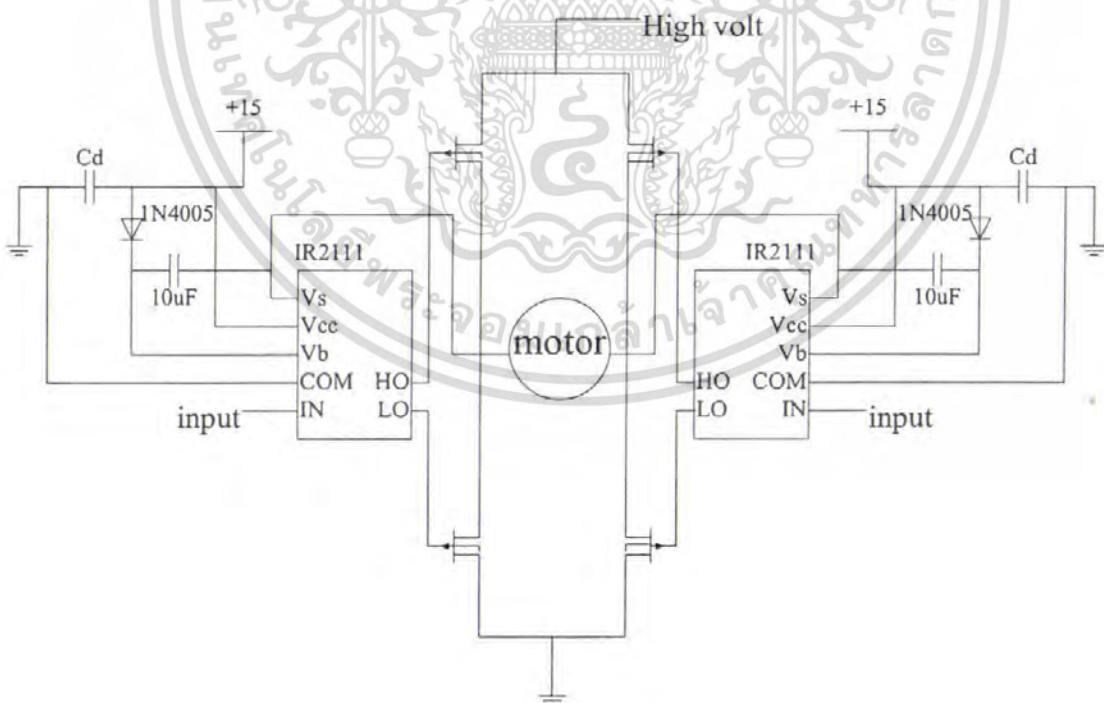
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะไม่มีเอาต์พุตออกมา แต่ขา LO จะมีเอาต์พุตเท่ากับ 15 โวลต์ หมายความว่า ถ้า IR2111 รับ สัญญาณเข้าขา IN เป็น 0 โวลต์ ไอจีบีที ตัวล่างจะทำงาน



รูปที่ 3.9 ไทม์มิ่ง ไดอะแกรม ของไอซีเบอร์ IR2111

จากวงจรสวิตชิ่ง และวงจรควบคุมการทำงานของ ไอจีบีที จะได้วงจรรวมตามรูปที่ 3.7 ซึ่งเป็น วงจรแบบฟูลบริดจ์ที่แปลงแรงดันไฟฟ้ากระแสตรงเป็นแรงดันไฟฟ้ากระแสสลับแบบเฟสเดียว

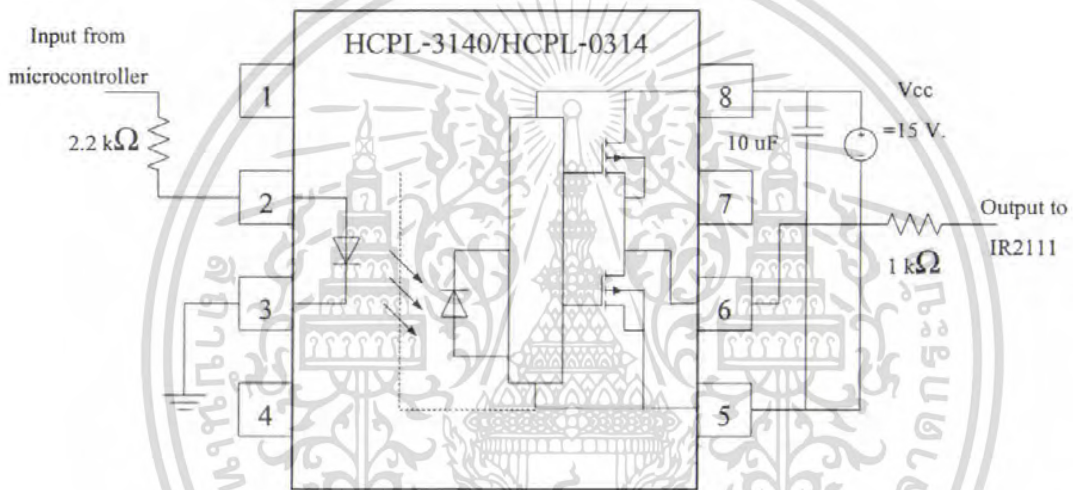


รูปที่ 3.10 วงจรอินเวอร์เตอร์เฟสเดียว แบบฟูลบริดจ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 วงจร ออปโต คัปเปิล

วงจรออปโต คัปเปิล เป็นวงจรที่ใช้แยกกราวด์ของไมโครคอนโทรลเลอร์ ออกจากกราวด์ของระบบเพื่อป้องกันแรงดันและกระแสย้อนกลับ ที่อาจทำให้ไมโครคอนโทรลเลอร์พังได้ ถ้าไม่มีการแยกกราวด์ออกจากกัน ซึ่งนอกจากจะใช้แยกกราวด์แล้ว วงจรออปโต คัปเปิลนี้ ยังทำหน้าที่เป็นตัวขยายสัญญาณจาก 5 โวลต์ ที่รับมาจากไมโครคอนโทรลเลอร์ ให้ขยายเป็น 15 โวลต์ เพื่อที่จะไปเข้าขา IN ของ IR2111



รูปที่ 3.11 วงจรออปโต คัปเปิล

ข้อควรระวัง

1. เนื่องจากเป็นการทดลองแรงดันไฟฟ้าสูงถึง 300 โวลต์ ดังนั้นจึงต้องเลือกอุปกรณ์ที่สามารถทนแรงดันและกระแสที่พิกัดโหลดที่เราใช้ได้
2. ในการทดลองต้องต่อวงจรให้ครบก่อนจ่ายสัญญาณจากไมโครคอนโทรลเลอร์เพราะถ้าต่อไม่ครบสัญญาณเอาท์พุท จาก IR2111 ก็จะไม่มีการทำงาน
3. ถ้าจะทดลองกับโหลดมอเตอร์ให้ใช้ วาริเอ็ท (Variac) ค่อยๆเพิ่มแรงดัน เพื่อป้องกันกระแสกระชากในช่วงเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 การทำงานของไมโครคอนโทรลเลอร์

การทำงานของโปรแกรมจะเริ่มจากการตั้งค่าโมดูลต่างๆเช่น

1. โมดูลสำหรับการสื่อสารข้อมูลอนุกรม(UART) โดยตั้งค่าไว้ที่ 57600 บิตต่อวินาที
2. โมดูลเชื่อมต่อตัวเข้ารหัสแบบควอดราเจอร์(QEI) เนื่องจากตัวเข้ารหัสที่ใช้งาน ให้ค่าพัลส์ 250 พัลส์ต่อรอบ จึงตั้งค่าการนับ พัลส์ A และ B ไว้ที่ 1000 พัลส์ ต่อการหมุนตัวเข้ารหัสหนึ่งรอบ เมื่อโมดูลนับค่าถึงค่าที่กำหนด จะเกิดอินเทอร์พท์ เพื่อทำงาน รีเซต ค่าการนับ เป็นศูนย์ โดยขั้นตอนการตั้งค่า โมดูล QEI มีดังนี้

2.1 ทำการ Enable Interrupt

2.2 ทำการกำหนด ค่าตำแหน่งสูงสุดในการนับ ซึ่งเท่ากับ 1000

2.3 ทำการกำหนด โหมดทิศทางการรับค่า อินพุต A และ B

3. โมดูล Motor Control PWM โดยมีขั้นตอนดังนี้

3.1 ทำการ Enable Interrupt

3.2 กำหนดค่า Prescaler และ Postscaler

3.3 กำหนดค่า Duty Cycle

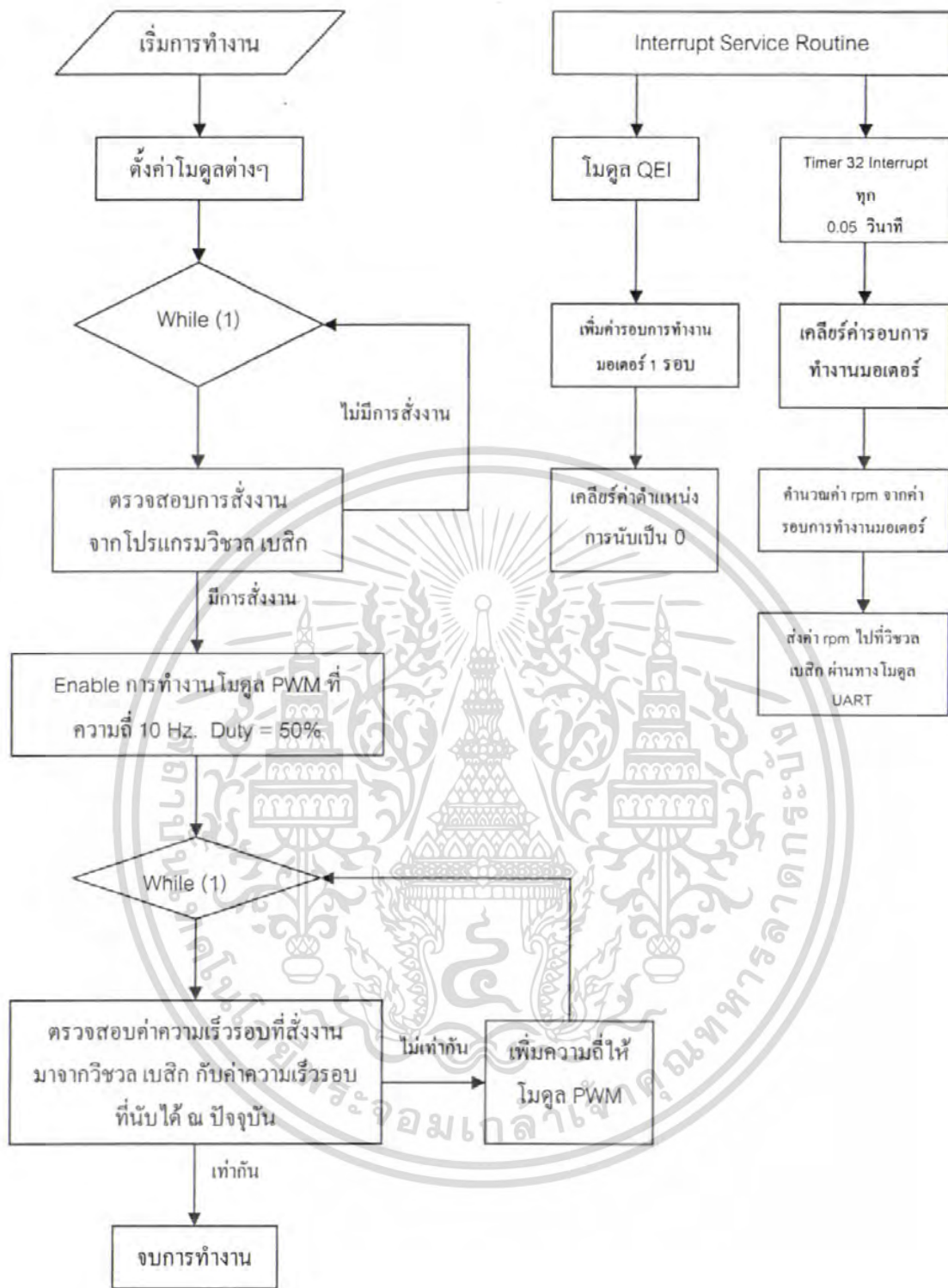
3.4 กำหนดค่า Period

3.5 กำหนดโหมดแบบปรับสัญญาณกึ่งกลาง(Center Aligned Mode)

3.6 กำหนด Channel 1 ทำงานแบบ Complementary Mode

3.7 ทำการ Enable PWM Function

การตั้งค่าของโมดูล Motor Control PWM นี้เมื่อ ทำการ Enable โมดูลจะส่งสัญญาณเอาต์พุตออกมาทันที โดยในโครงการนี้ได้ย้ยส่วนของ การกำหนดค่า Period , Duty Cycle และ การ Enable ไปไว้หลังจาก ได้รับ สัญญาณ ให้ทำงานจาก โมดูลสื่อสารอนุกรม ซึ่งสั่งการโดย โปรแกรม วิชาล เบสิก (Visual Basic) การทำงานของ ไมโครคอนโทรลเลอร์โดยรวม แสดงตามโฟลว์ชาร์ต (Flow Chart) ดังนี้



รูปที่ 3.12 โฟลว์ชาร์ตแสดงการทำงาน โดยรวมของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง

ในบทนี้ จะกล่าวถึงการทดลองและผลการทดลองของวงจรและภาคขับเคลื่อนเหนี่ยวนำแบบเฟสเดียว การทำงานของระบบควบคุมความเร็วรอบแบบวงปิด โดยการเปลี่ยนความถี่ มีรายละเอียดของผลการทดลองดังต่อไปนี้

4.1 ผลการทดลองที่ค่าความเร็วรอบต่างๆ

4.1.1 ที่ความเร็วรอบ 1000 rpm

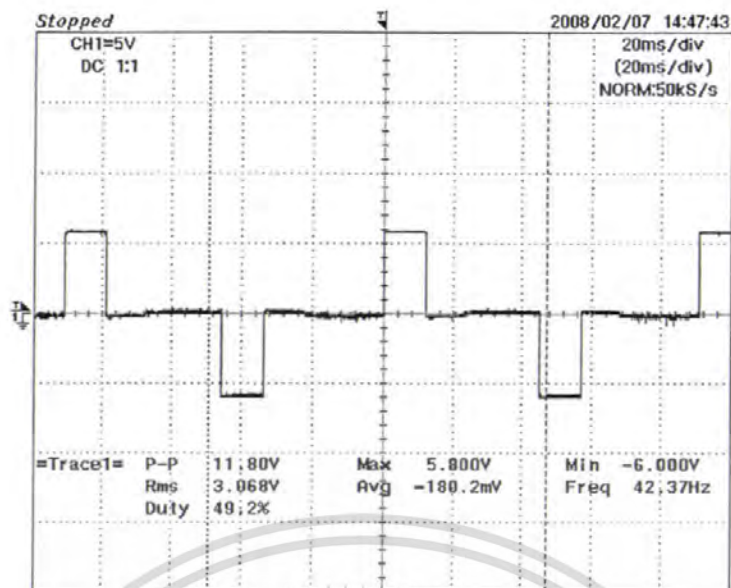
เมื่อทำการตั้งค่าความเร็วรอบเท่ากับ 1000 rpm ผลของความเร็วรอบเอาต์พุตที่ได้แสดงดังรูป

รูปที่ 4.1



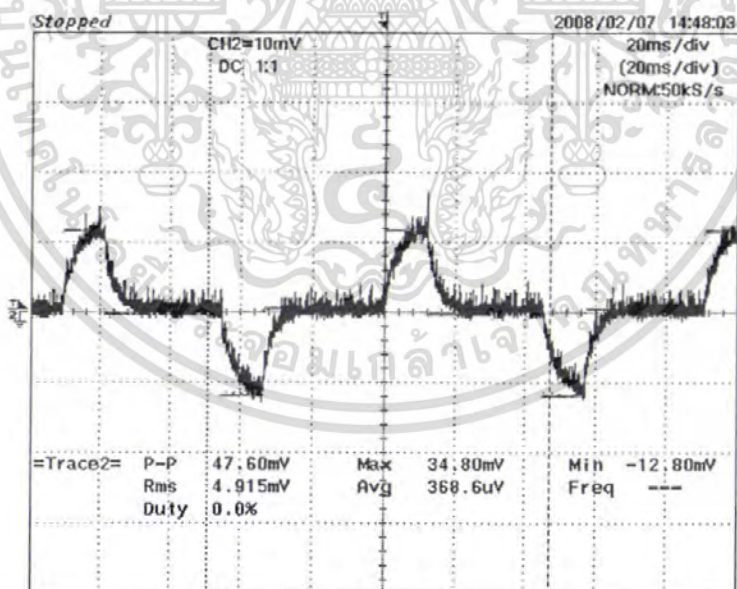
รูปที่ 4.1 กราฟแสดงผลความเร็วรอบที่ตั้งไว้ 1000 rpm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



=Filter= =Offset= =Record Length= =Trigger=
Smoothing : OFF CH1 : 0.00V Main : 10K Mode : AUTO
BW : FULL CH2 : 0.0000V Zoom : 10K Type : EDGE CH1 \uparrow
Delay : 0.0ns
Hold Off : MINIMUM

รูปที่ 4.2 กราฟแสดงแรงดันที่ ความเร็วรอบ 1000 rpm



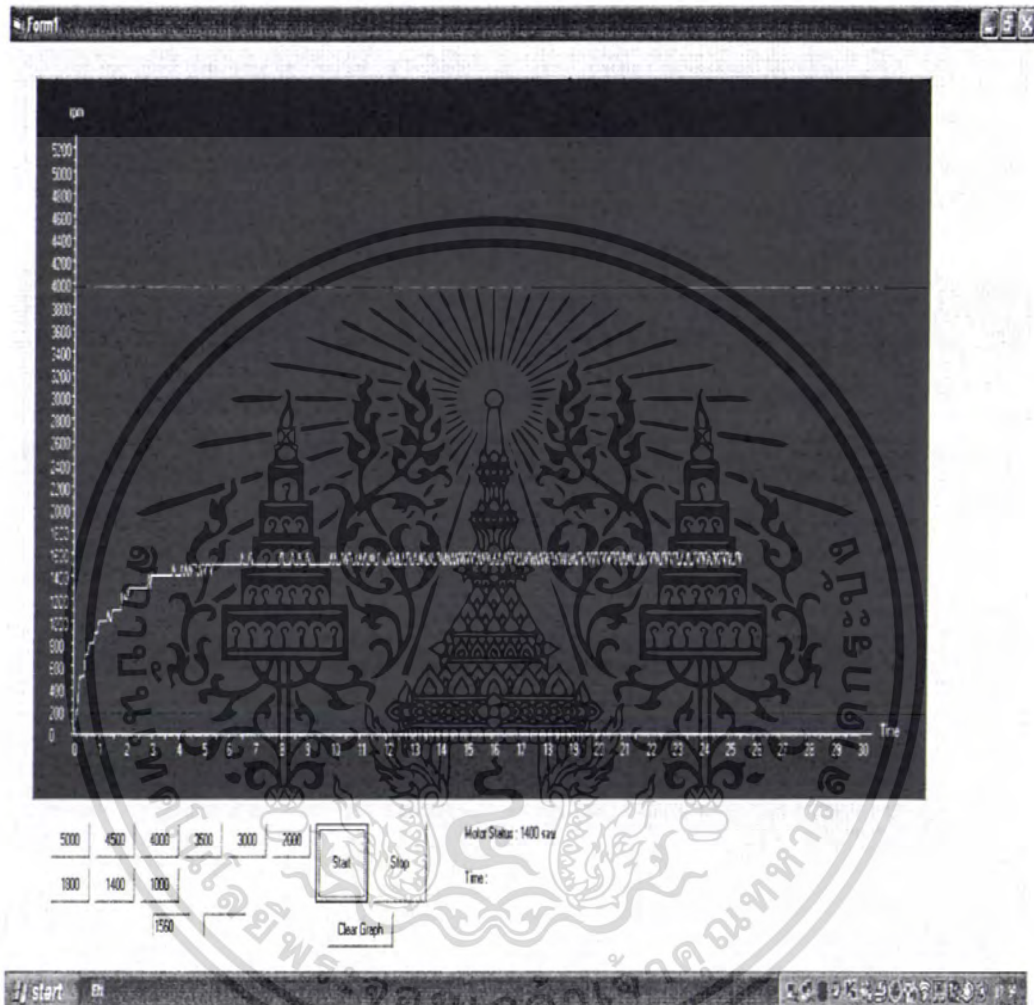
=Filter= =Offset= =Record Length= =Trigger=
Smoothing : OFF CH1 : 0.00V Main : 10K Mode : AUTO
BW : FULL CH2 : 0.0000V Zoom : 10K Type : EDGE CH1 \uparrow
Delay : 0.0ns
Hold Off : MINIMUM

รูปที่ 4.3 กราฟแสดงกระแสที่ ความเร็วรอบ 1000 rpm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

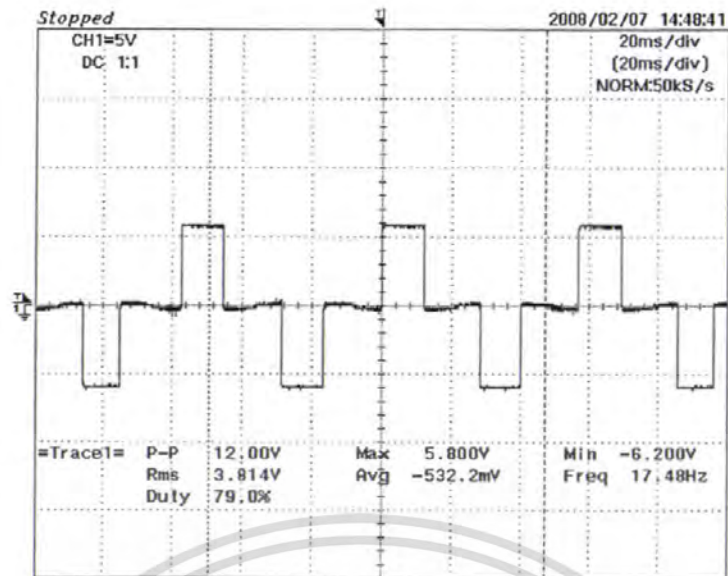
4.1.2 ที่ความเร็วรอบ 1400 rpm

เมื่อทำการตั้งค่าความเร็วรอบเท่ากับ 1400 rpm ผลของความเร็วรอบเอาต์พุตที่ได้แสดงดังรูปที่ 4.4



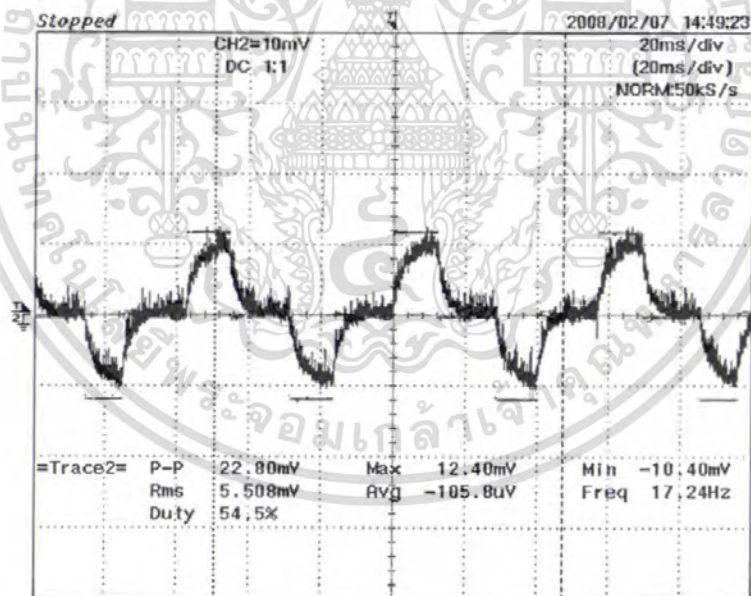
รูปที่ 4.4 กราฟแสดงผลความเร็วรอบที่ตั้งไว้ 1400 rpm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



=Filter= Smoothing : OFF BW : FULL
=Offset= CH1 : 0.00V CH2 : 0.0000V
=Record Length= Main : 10K Zoom : 10K
=Trigger= Mode : AUTO Type : EDGE CH1 \uparrow Delay : 0.0ns Hold Off : MINIMUM

รูปที่ 4.5 กราฟแสดงแรงดันที่ ความเร็วรอบ 1400 rpm



=Filter= Smoothing : OFF BW : FULL
=Offset= CH1 : 0.00V CH2 : 0.0000V
=Record Length= Main : 10K Zoom : 10K
=Trigger= Mode : AUTO Type : EDGE CH1 \uparrow Delay : 0.0ns Hold Off : MINIMUM

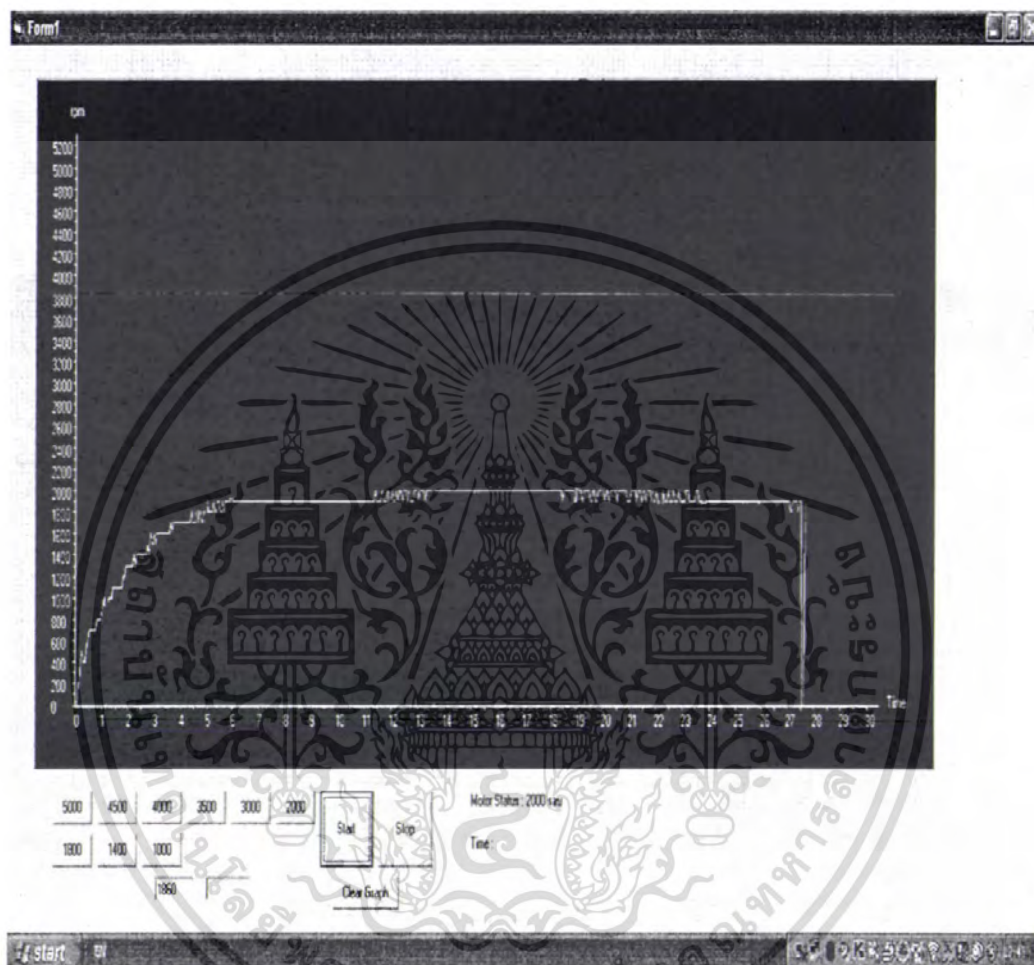
รูปที่ 4.6 กราฟแสดงกระแสที่ ความเร็วรอบ 1400 rpm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 ที่ความเร็วรอบ 2000 rpm

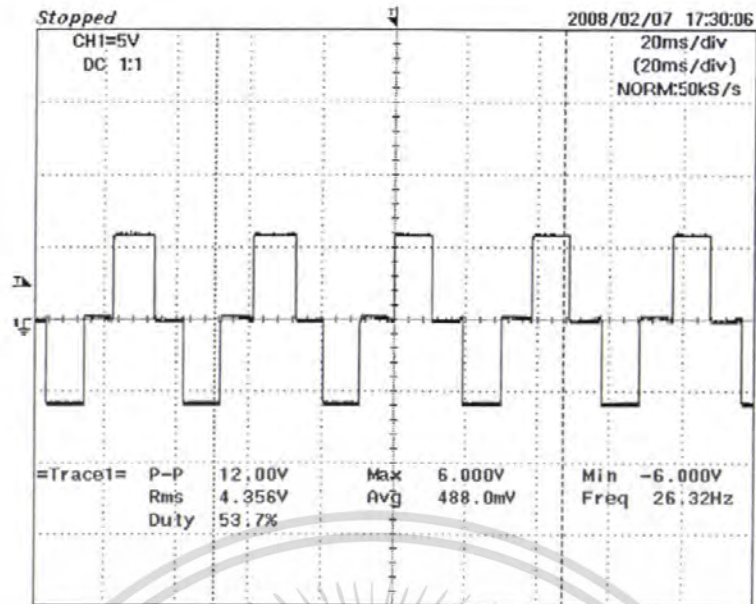
เมื่อทำการตั้งค่าความเร็วรอบเท่ากับ 2000 rpm ผลของความเร็วรอบเอาท์พุทที่ได้แสดงดังรูป

ที่ 4.7



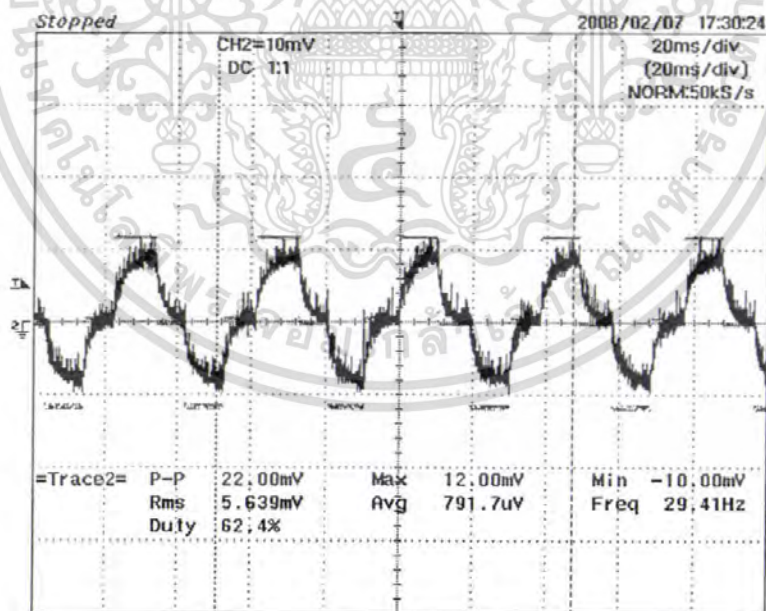
รูปที่ 4.7 กราฟแสดงผลความเร็วรอบที่ตั้งไว้ 2000 rpm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



=Filter= =Offset= =Record Length= =Trigger=
Smoothing : OFF CH1 : 0.00V Main : 10K Mode : AUTO
BW : FULL CH2 : 0.0000V Zoom : 10K Type : EDGE CH1 \uparrow
Delay : 0.0ns
Hold Off : MINIMUM

รูปที่ 4.8 กราฟแสดงแรงดันที่ ความเร็วรอบ 2000 rpm



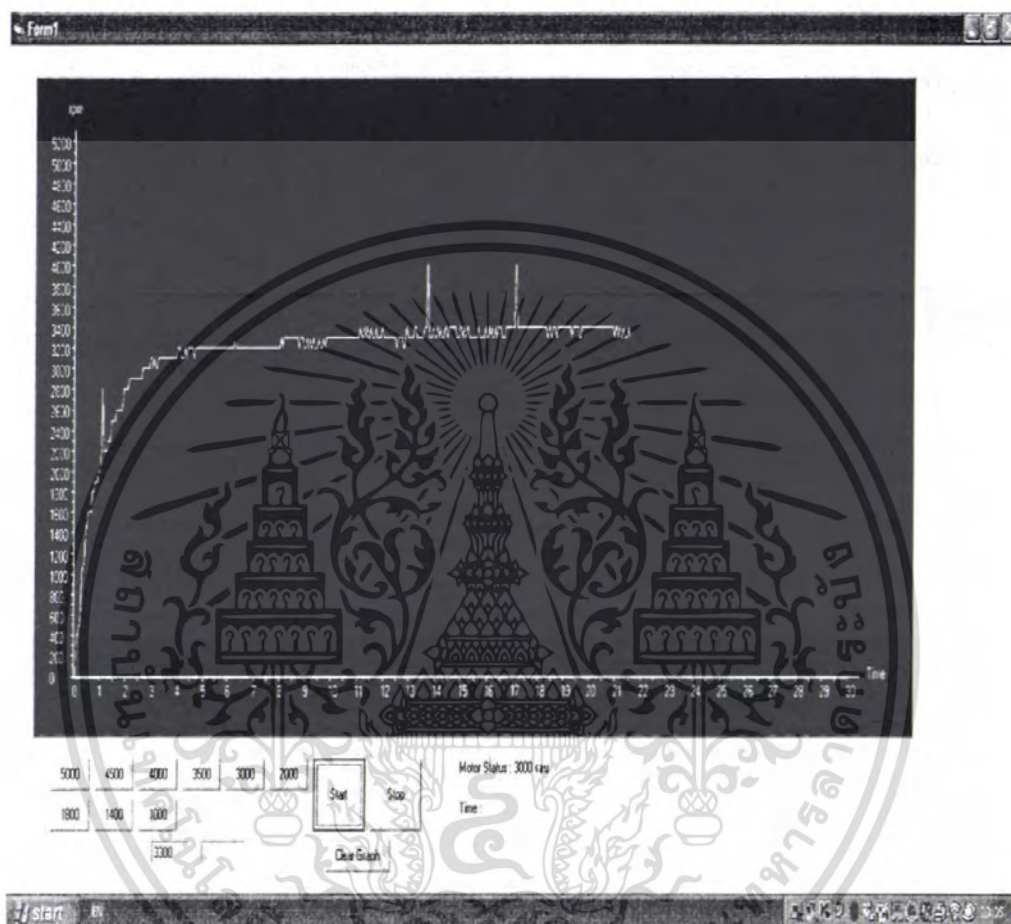
=Filter= =Offset= =Record Length= =Trigger=
Smoothing : OFF CH1 : 0.00V Main : 10K Mode : AUTO
BW : FULL CH2 : 0.0000V Zoom : 10K Type : EDGE CH1 \uparrow
Delay : 0.0ns
Hold Off : MINIMUM

รูปที่ 4.9 กราฟแสดงกระแสที่ ความเร็วรอบ 2000 rpm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

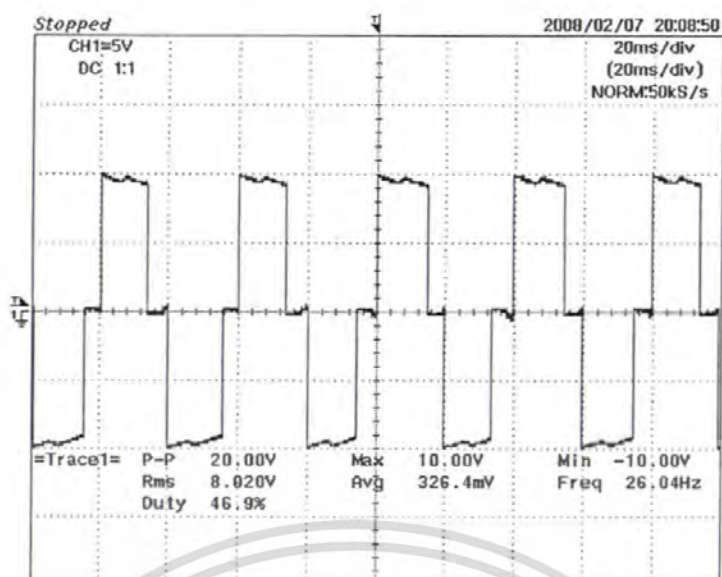
4.1.4 ที่ความเร็วรอบ 3000 rpm

เมื่อทำการตั้งค่าความเร็วรอบเท่ากับ 3000 rpm ผลของความเร็วรอบเอาท์พุทที่ได้แสดงดังรูปที่ 4.10



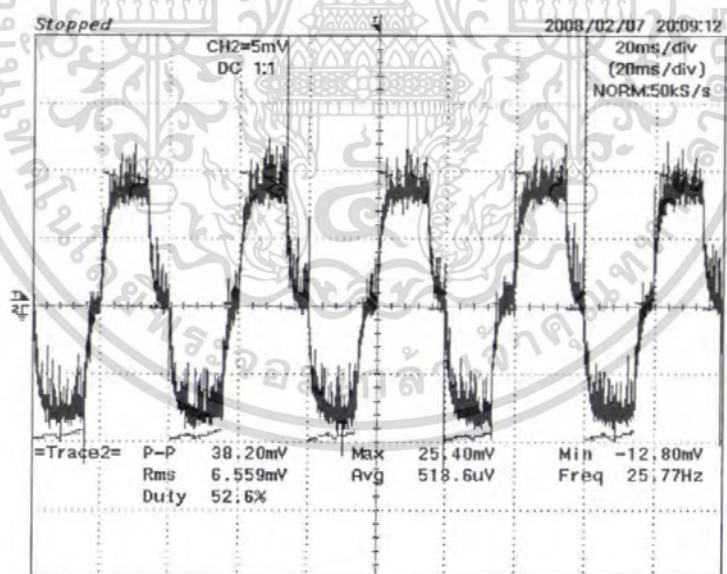
รูปที่ 4.10 กราฟแสดงผลความเร็วรอบที่ตั้งไว้ 3000 rpm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



=Filter= Smoothing : OFF BW : FULL
=Offset= CH1 : 0.00V CH2 : 0.0000V
=Record Length= Main : 10K Zoom : 10K
=Trigger= Mode : AUTO Type : EDGE CH1 Delay : 0.0ns Hold Off : MINIMUM

รูปที่ 4.11 กราฟแสดงแรงดันที่ ความเร็วรอบ 3000 rpm



=Filter= Smoothing : OFF BW : FULL
=Offset= CH1 : 0.00V CH2 : 0.0000V
=Record Length= Main : 10K Zoom : 10K
=Trigger= Mode : AUTO Type : EDGE CH1 Delay : 0.0ns Hold Off : MINIMUM

รูปที่ 4.12 กราฟแสดงกระแสที่ ความเร็วรอบ 3000 rpm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

5.1 สรุปผลการทดลอง

จากผลการทดลองพบว่า การควบคุมความเร็วรอบของมอเตอร์สามารถควบคุมได้โดยการเปลี่ยนแปลงค่าความถี่ โดยถ้าเพิ่มความถี่ให้มีค่ามากขึ้น ค่าความเร็วรอบก็จะเพิ่มขึ้น และในทางตรงกันข้ามถ้าลดความถี่ลง ค่าความเร็วรอบก็จะลดลง โดยที่แรงดันคงที่

ค่าความเร็วรอบสามารถวัดได้จากการที่เอนโคเดอร์(Encoder) รับค่าแล้วส่งค่าไปยังไมโครคอนโทรลเลอร์ ให้ไมโครคอนโทรลเลอร์ทำการประมวลผล ถ้าค่าความเร็วรอบยังไม่ถึงค่าที่เราได้ตั้งค่าเอาไว้ ไมโครคอนโทรลเลอร์จะทำการสั่งเพิ่มค่าความถี่ไปเรื่อยๆ จนกว่าจะถึงค่าความเร็วรอบที่เราตั้งค่าไว้ โดยใช้หลักการควบคุมแบบวงปิด

5.2 ปัญหาที่พบและแนวทางแก้ไข

เนื่องจากทางโครงการได้เลือกใช้มอเตอร์เหนี่ยวนำแบบเฟสเดียว ขนาดแรงดัน 220 โวลต์ ซึ่งเป็นแรงดันไฟฟ้าบ้าน การทดลองจึงค่อนข้างอันตรายเพราะเป็นแรงดันสูง และอุปกรณ์ในการทดลองที่สามารถรองรับด้านความปลอดภัยนั้น ไม่ได้มีไว้ให้ ดังนั้นจึงต้องทำการทดลองด้วยความระมัดระวัง และแก้ปัญหาโดยการขอใช้อุปกรณ์จากทางภาคไฟฟ้ากำลัง

การทดลองที่ปลอดภัยนั้น ต้องเริ่มต้นทดลองที่ค่าแรงดันต่ำๆก่อน แล้วจึงค่อยเพิ่มค่าแรงดันให้สูงขึ้น เพื่อป้องกันการกระชากของกระแส และวงจรควรมีฟิวส์(Fuse) เพื่อป้องกันการเสกเกินไว้ด้วย ทางด้านห้องทดลองควรมีเบรกเกอร์(Breaker) ที่ใช้งานได้อย่างมีประสิทธิภาพ เพื่อว่าเมื่อเกิดกระแสไหลมากเกินไป เบรกเกอร์ต้องสามารถตัดไฟได้ทันทีทันใด

ในการออกแบบด้านฮาร์ดแวร์(Hardware) ซึ่งเป็นส่วนวงจรขมอมอเตอร์นั้น การเลือกใช้ไอซีมีส่วนสำคัญมาก เพราะสามารถเลือกใช้ไอซีเบอร์อะไรก็ได้ แต่ต้องคำนึงด้วยว่า ไอซีเบอร์นั้นมีจำหน่ายในประเทศไทยหรือไม่ ราคาถูกหรือแพงอย่างไร และมีความเหมาะสมกับตัวชิ้นงานของโครงการมากน้อยเพียงใด นอกจากนี้การใช้งานไอซี ควรศึกษาาคาดำชี้ท ให้ละเอียด ให้เข้าใจเสียก่อน รวมถึงรายละเอียดส่วนย่อยที่คาดำชี้ท ไม่ได้บอกมา จำเป็นต้องศึกษาให้มากที่สุดเท่าที่จะสามารถหาข้อมูลได้ เพราะในบางครั้งคาดำชี้ท ก็ไม่ได้บอกส่วนสำคัญ ที่จะต้องใช้ในวงจรของจริง ซึ่งในบางกรณีถ้านำมาต่อใช้งานจริงแล้ว วงจรนั้นอาจไม่ทำงานก็เป็นได้

5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา

จากการที่ทางโครงการนี้ได้ทำการออกแบบโครงสร้างของเครื่องบินเหวี่ยงเลือดไว้ก่อนหน้านี้และสามารถควบคุมความเร็วรอบของมอเตอร์ได้ ซึ่งถือเป็นส่วนสำคัญที่จำเป็นในการทำงานของเครื่องบินเหวี่ยงเลือด แต่โครงสร้างภายนอกของตัวเครื่องก็ถือเป็นองค์ประกอบที่สำคัญไม่แพ้กัน ดังนั้น ส่วนที่ควรพัฒนาต่อไปคือ การสร้างชิ้นงานเครื่องบินเหวี่ยงเลือดให้ออกมาเหมาะสมในด้านรูปทรง ขนาด และน้ำหนัก ตามที่ได้ออกแบบไว้ให้สามารถนำไปใช้ประโยชน์ได้จริง ตามวัตถุประสงค์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] นคร กักศิชาติ, ชัยวัฒน์ ลิ้มพรจิตรวิไล. คู่มือการทดลอง dsPIC Microcontroller เบื้องต้น ด้วยโปรแกรมภาษา C กับ MPLAB C30. กรุงเทพมหานคร : อินโนเวทีฟ เอ็กเพอริเมนต์. 2547.
- [2] รศ.ดร.วีรเชษฐ์ ชันเงิน, วุฒิพล ธาราธิรเศรษฐ์. อิเล็กทรอนิกส์กำลัง. กรุงเทพมหานคร : สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2547.
- [3] ศิวะ หงษ์นภา. ระบบขับเคลื่อนมอเตอร์ไฟฟ้ากระแสสลับ. กรุงเทพมหานคร : สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น). 2543.
- [4] ชัยทัต มณีอินทร์, เลิศชาย โสกา, สิริชัย งามโชคพิภพ. “อินเวอร์เตอร์ 1 เฟส.” ปรินญาณิพนธ์วิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2538.
- [5] Dr. Zainal Salam. “DC to AC Conversion (INVERTER).” [Online]. Available : <http://encon.fke.utm.my/notes/Introduction-2002.pdf>. 2002.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

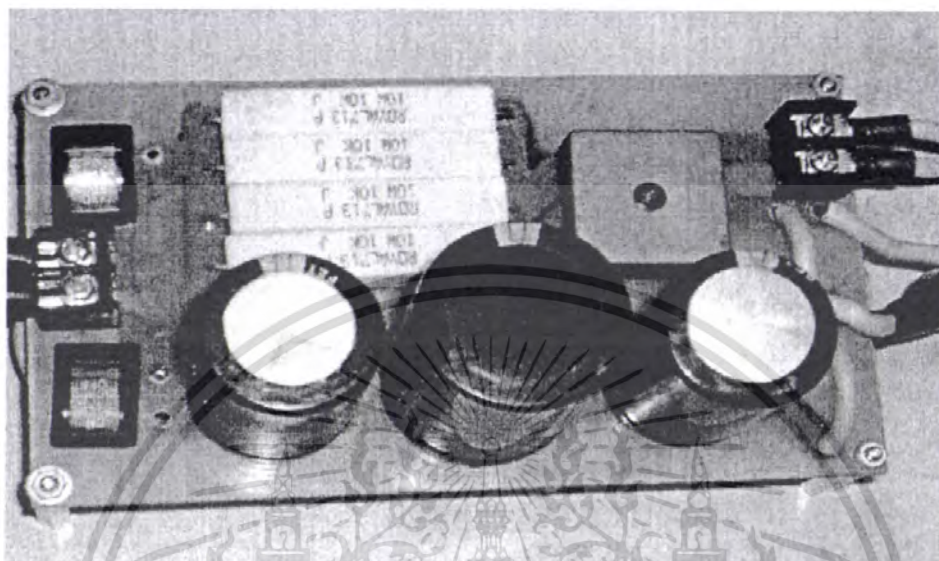


ภาคผนวก

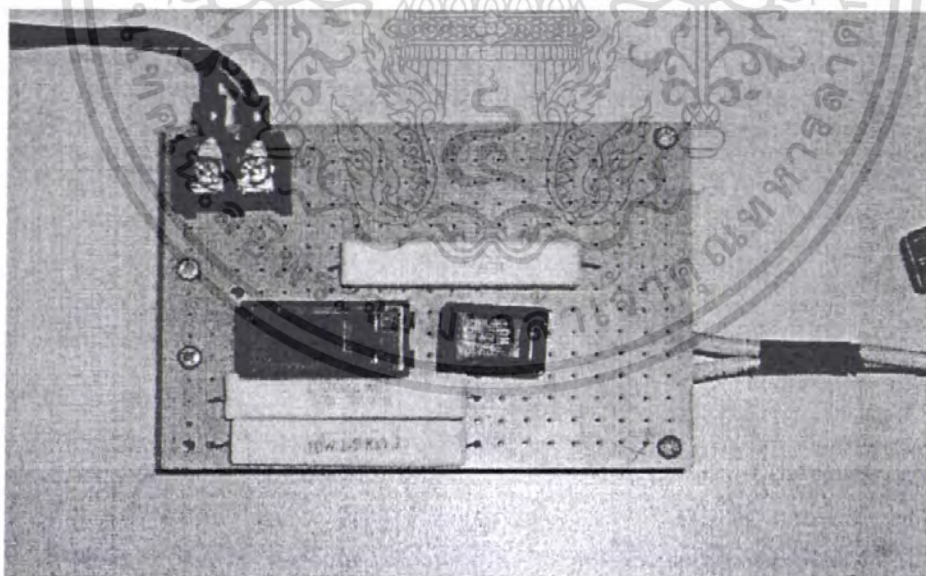
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

แสดงรูปส่วนประกอบของวงจร

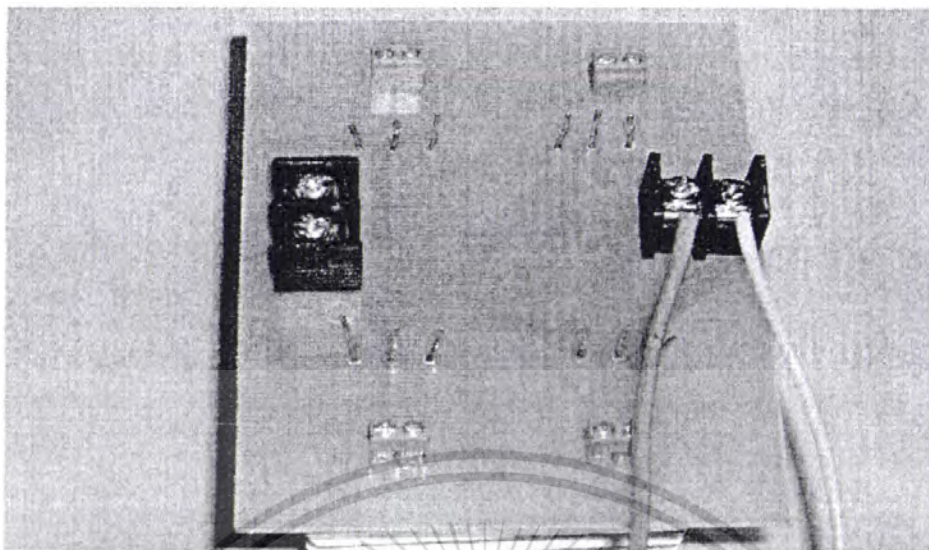


รูปที่ ก.1 แสดงวงจรเรียงกระแส

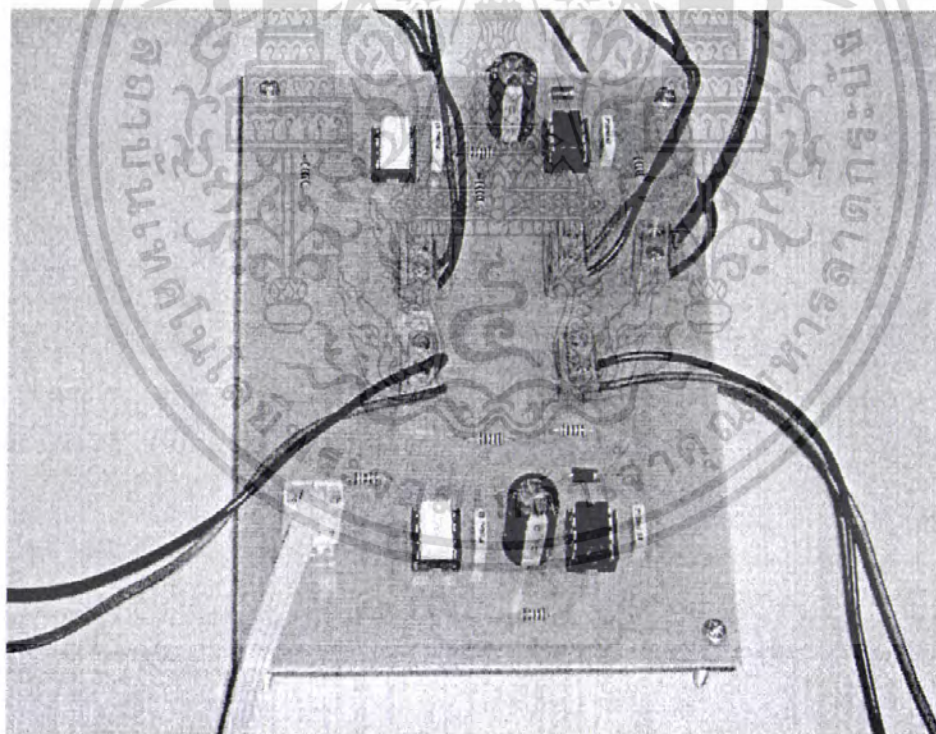


รูปที่ ก.2 แสดงวงจรป้องกันกระแสพุ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

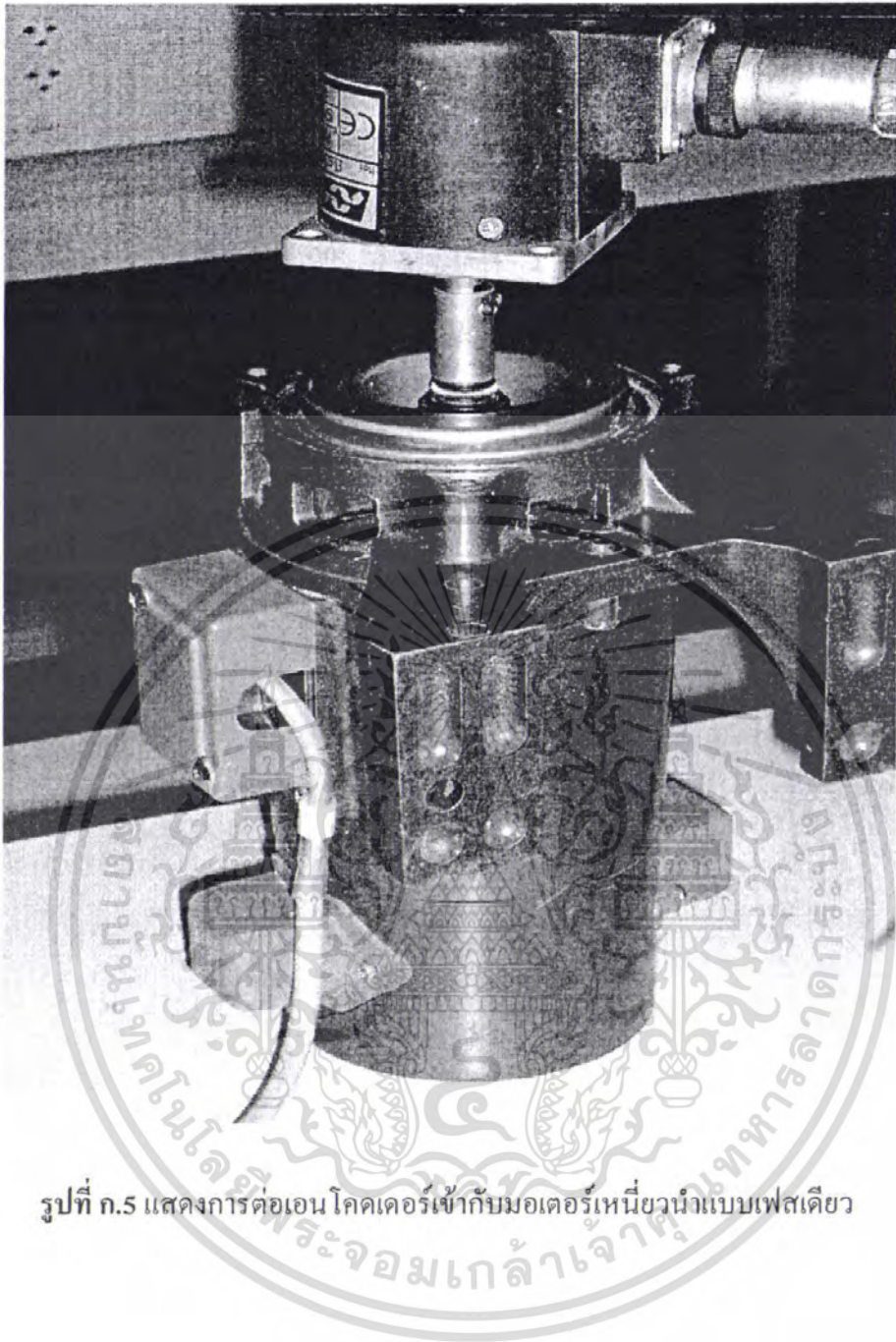


รูปที่ ก.3 แสดงวงจรสวิตชิง



รูปที่ ก.4 แสดงวงจรขับเคลื่อนมอเตอร์เหนี่ยวนำแบบเฟสเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.5 แสดงการต่อเอนโคเดอร์เข้ากับมอเตอร์เหนี่ยวนำแบบเฟสเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

โปรแกรมการติดต่อและโปรแกรมควบคุม

โปรแกรมวิซวลเบสิก ซึ่งเขียนเพื่อใช้ในการติดต่อส่งข้อมูลจากเอ็น โคดเดอร์เพื่อนำผลไปพล็อตกราฟ และส่งผลไปให้ไมโครคอนโทรลเลอร์ทำการประมวลผล

Option Explicit

Dim Dspicinput, index_str, round_str, dat, Hz As String

Dim round_new, round_old, index_old, x, y, a, c, time1, time2, rpm1, rpm2, round1, Index As Variant

Private Sub Command1_Click()

Hz = "1"

Label1.Caption = "Motor Status : 5000 ???"

End Sub

Private Sub Command6_Click()

Hz = "2"

Label1.Caption = "Motor Status : 4500 ???"

End Sub

Private Sub Command7_Click()

Hz = "3"

Label1.Caption = "Motor Status : 4000 ???"

End Sub

Private Sub Command8_Click()

Hz = "4"

Label1.Caption = "Motor Status : 3500 ???"

End Sub

Private Sub Command9_Click()

Hz = "5"

Label1.Caption = "Motor Status : 3000 ???"

End Sub

Private Sub Command10_Click()

Hz = "6"

Label1.Caption = "Motor Status : 2000 ???"

End Sub

Private Sub Command11_Click()

Hz = "7"

Label1.Caption = "Motor Status : 1800 ???"

End Sub

Private Sub Command12_Click()

Hz = "8"

Label1.Caption = "Motor Status : 1400 ???"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End Sub

Private Sub Command4_Click()
    Hz = "9"
    Label1.Caption = "Motor Status : 1000 ????"
End Sub

Private Sub Command2_Click()
    Timer1.Enabled = True
    MSComm1.Output = Hz
End Sub

Private Sub Command3_Click()
    Timer1.Enabled = False
    MSComm1.Output = "0"
End Sub

Private Sub Command5_Click()
    Picture1.Cls
    Call Drawgraph
    time1 = 30
    time2 = 30
End Sub

Private Sub Form_LOAD()
    MSComm1.PortOpen = True
    Picture1.AutoRedraw = True
    Call Drawgraph
    time1 = 30
    time2 = 30
    rpm1 = 58
    rpm2 = 58
    round_new = 0
    round_old = 0
    index_old = 0
    x = 0
    y = 0
    a = 0
    c = 0
    round1 = 0
    Index = 0

End Sub

Private Sub Timer1_Timer()
    Dspicinput = MSComm1.Input
    round_new = Val(Dspicinput)
    Text1.Text = round_new
    time2 = time2 + 1

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

rpm2 = round_new / 100
If (round_new < 0 Or round_new > 5500) Then
rpm2 = round_old / 100
End If
rpm2 = Round(rpm2, 0)
round_old = round_new
rpm2 = (rpm2 - 58) * -1
Picture1.Line (time1, rpm1)-(time2, rpm2)
time1 = time2
rpm1 = rpm2
End Sub
Sub Drawgraph()
c = 1
Picture1.Scale (0, 0)-(680, 64)
Picture1.BackColor = vbBlack
Picture1.ForeColor = vbWhite
Picture1.Line (30, 58)-(640, 58)
Picture1.Line (30, 5)-(30, 58)
For x = 30.1 To 630 Step 1
Picture1.Line (x, 58)-(x, 58.3)
Next x
For x = 40 To 630 Step 10
Picture1.Line (x, 58)-(x, 58.5)
Picture1.CurrentY = 58.7
Picture1.CurrentX = x - 3
Next x

For x = 50 To 630 Step 20
Picture1.CurrentY = 58.7
Picture1.CurrentX = x - 3
Picture1.Print c
c = c + 1
Next x

Picture1.CurrentY = 58.7
Picture1.CurrentX = 29
Picture1.Print "0"

For y = 58 To 6 Step -1
Picture1.Line (30, y)-(28, y)
Next y

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

For y = 58 To 6 Step -2
  Picture1.Line (30, y)-(27.5, y)
  Picture1.CurrentY = y - 0.5
  Picture1.CurrentX = 10
  c = (y - 58) * 100
  Picture1.Print c * -1
Next y

```

```

Picture1.CurrentX = 645
Picture1.CurrentY = 57
Picture1.Print "Time"
Picture1.CurrentX = 25
Picture1.CurrentY = 2
Picture1.Print "rpm"
End Sub

```

โปรแกรมภาษาซี

```

//-----//
// Program      : library for LCD display
// Description   : Module function for LCD display mode 4 bit
//               : operate clock 7.3738 MHz at PI1_4x
// Filename      : lcd.h
// C compiler    : C30 Compiler by Microchip Technology
//-----//

#define NUMBER_OF_DIGITS 32
#define LINE1 0x80 // Define constant
#define LINE2 0xC0 // Define constant
#define delay_ms(x) lcd_delay(x)
#define lcd_clear() lcd_command(1)
#define lcd_origin() lcd_command(2)
#define RS LATFbits.LATF0 // Bit control command for LCD
#define E LATFbits.LATF1 // Bit control pulse command for LCD
//-----//
//-----// Function delay 1 ms per unit -----//
//-----//

void lcd_delay(unsigned int ms)
{
    unsigned int x,a; // Keep for counter loop
    for(x=0;x<ms;x++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    for(a=0;a<439;a++); // Loop for delay 1 millisecc per unit(value = 816 @7.3728 MHz PLL4x)
    }
}
//-----Function send command to LCD format 4 Bit -----//
//-----Function send data to LCD format 4 Bit -----//

void lcd_command(unsigned char com)
{
    unsigned char buff; // For Keep command send to LCD
    buff = (com & 0xF0)>>4; // Keep 4 bit high byte before send to LCD
    RS = 0; // Select send command to LCD

    E = 1; // Start generate pulse clock
LCD
    LATD = (LATD & 0xF0)|buff; // Send data to LCD port
    lcd_delay(1); // Delay 1 ms
    E = 0; // Stop generate pulse clock LCD
    lcd_delay(1); // Delay 1 ms
    buff = (com & 0x0F); // Keep 4 bit low byte before send to LCD
    RS = 0; // Select send command to LCD

    E = 1; // start generate pulse clock
LCD
    LATD = (LATD & 0xF0)|buff; // Send data to LCD port
    lcd_delay(1); // Delay 1 ms
    E = 0; // Stop generate pulse clock LCD
    lcd_delay(1); // Delay 1 ms
}
//-----Function send data to LCD format 4 Bit -----//
//-----Function send data to LCD format 4 Bit -----//

void lcd_text(char text)
{
    unsigned char buff; // For Keep data send to LCD
    buff = (text & 0xF0)>>4; // Keep 4 bit high byte before send to LCD
    RS = 1; // Select send data to LCD

    E = 1; // start generate pulse clock
LCD
    LATD = (LATD & 0xF0)|buff; // Send data to LCD port
    lcd_delay(1); // Delay 1 ms

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

E = 0;          // Stop generate pulse clock LCD
lcd_delay(1);  // Delay 1 ms

buff = (text & 0x0F); // Keep 4 bit low byte before send to LCD
RS = 1;                                               // Write Command to LCD

E = 1;                                               // start generate pulse clock
LCD
LATD = (LATD & 0xF0) | buff; // Send data to LCD port
lcd_delay(1); // Delay 1 ms
E = 0;          // Stop generate pulse clock LCD
lcd_delay(1);  // Delay 1 ms
}
//-----
//----- Function show string message -----
//-----
void lcd_puts(unsigned char line, char *p)
{
    lcd_origin(); // Set origin address of LCD
    lcd_command(line); // Set address 001H of LCD
    while(*p) // Check data pointer != 0?
    {
        lcd_text(*p); // Send data to LCD
        p++; // Increase address 1 time
    }
}
//-----
//----- Convert long integer to ascii for display on LCD -----
//-----
void _ultoa(unsigned long value, char* string, unsigned char radix)
{
    unsigned char index; // Counter of digit
    char buffer[NUMBER_OF_DIGITS]; // Data buffer
    index = NUMBER_OF_DIGITS; // Load counter by digit count

    do
    {
        buffer[--index] = '0' + (value % radix); // Convert configuration by radix(10 or 16)
        if (buffer[index] > '9') buffer[index] += 'A' - '9' - 1; // For base over 10(base 16)
        value /= radix; // Div by to calculate into next digit
    } while (value != 0); // End for convert?
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
    *string++ = buffer[index++]; // Load convert value to string buffer
} while ( index < NUMBER_OF_DIGITS ); // Over of digit count?

*string = 0; // Place null for end string
}

//-----
//----- Convert long integer to ascii for display on LCD -----
//-----

void _ltoa(long value_1, char * string_1, unsigned char radix_1)
{
    if (value_1 < 0 && radix_1 == 10) // For value < 0 (base 10)
    {
        *string_1++ = '-'; // Load sign '-' for display
        _ultoa(-value_1, string_1, radix_1); // Convert long integer
    }
    else
    {
        _ultoa(value_1, string_1, radix_1); // Convert long integer
    }
}
//-----
//----- Convert long integer to ascii for display on LCD -----
//-----

/*void _ltoa(unsigned int value_1, char * string_1, unsigned char radix_1)
{
    if (value_1 >= 0x8000 && radix_1 == 10) // For value < 0 (base 10)
    {
        *(string_1++) = '-'; // Load sign '-' for display
        //value_1 = (value_1 - 0xFFFFE)+1;
        _ultoa(-value_1, string_1, radix_1); // Convert long integer
    }
    else
    {
        _ultoa(value_1, string_1, radix_1); // Convert long integer
    }
}
*/

//-----
//----- Convert integer to ascii for display on LCD -----
//-----

void inttoled(unsigned char posi, int value)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char buff[12]; // For keep string send to LCD
    _ltoa(value,&buff[0],10); // Convert value base 10
    lcd_puts(posi,&buff[0]); // Send integer to LCD
}
//-----
//----- Function set initial format LCD 4 Bit -----
//-----
void lcd_init()
{
    TRISD = 0;
    TRISFbits.TRISF0 = 0;
    TRISFbits.TRISF1 = 0;
    lcd_delay(500); // Delay 500 ms
    lcd_command(0x33); // Command set format 4 bit
    lcd_command(0x32); // Command set format 4 bit
    lcd_command(0x28); // 2 line ,5*7 dot
    lcd_command(0x0C); // Display ON , none cursor
    lcd_command(0x01); // Clear screen
}
/*-----
/* Header for QEI module library functions
/*-----

#ifndef __QEI_H
#define __QEI_H

/*QEI is defined in following variants */
#ifdef __dsPIC30F2010__ || defined(__dsPIC30F3010__) || defined(__dsPIC30F4012__) ||
    defined(__dsPIC30F3011__) || defined(__dsPIC30F4011__) || defined(__dsPIC30F6010__) ||
    defined(__dsPIC30F5015__) || defined(__dsPIC30F6010A__) || defined(__dsPIC30F5016__) ||
    defined(__dsPIC30F6015__)

/* List of SFRs for QEI */
/* This list contains the SFRs with default (POR) values to be used for configuring QEI */
/* The user can modify this based on the requirement */

#define QEICON_VALUE      0x0000
#define DF1.TCON_VALUE    0x0000
#define POSCNT_VALUE      0x0000
#define MAXCNT_VALUE      0xFFFF

/* QEICON Configuration Bit Definitions */

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในเชิงวิชาการเท่านั้น มิใช่เพื่อเผยแพร่ทั่วไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define QEI_DIR_SEL_CNTRL      0xFFFE /* Control/Status Bit, QEICON<11>, Defines Timer Counter (POSCNT)
Direction */

#define QEI_EXT_CLK           0xFFFF /* External clock from pin TQCK1 (on the rising edge) */
#define QEI_INT_CLK          0xFFFD /* Internal clock (FOSC/4) */

#define QEI_INDEX_RESET_ENABLE 0xFFFF /* QEI Index Pulse resets Position Counter */
#define QEI_INDEX_RESET_DISABLE 0xFFFB /* Index Pulse does not reset Position Counter */

#define QEI_CLK_PRESCALE_1    0xFFE7 /* QEI Timer Input Clock Prescale Select Bits */
#define QEI_CLK_PRESCALE_8    0xFFEF
#define QEI_CLK_PRESCALE_64   0xFFFF
#define QEI_CLK_PRESCALE_256  0xFFFE /* QEI 1:256 prescale value */

#define QEI_GATED_ACC_ENABLE  0xFFFF /* QEI Timer gated time accumulation enabled */
#define QEI_GATED_ACC_DISABLE 0xFFDF /* Timer gated time accumulation disabled */

#define QEI_LOGIC_CONTROL_IO  0xFFFF /* QEI Position Counter Direction Status Output Enable(QEI logic controls
state of I/O pin) */
#define QEI_NORMAL_IO         0xFFBF /* QEI Position Counter Direction Status Output Disabled(Normal I/O pin
operation) */

#define QEI_INPUTS_SWAP      0xFFE7 /* QEI Phase A and Phase B inputs swapped */
#define QEI_INPUTS_NOSWAP    0xFF7F /* QEI Phase A and Phase B inputs not swapped */

#define QEI_MODE_x4_MATCH     0x1FFF /* QEI Quadrature Encoder Interface enabled (x4 mode) with position counter
reset by match (MAXCNT) */
#define QEI_MODE_x4_PULSE     0x1EFF /* QEI Quadrature Encoder Interface enabled (x4 mode) with Index Pulse reset
of position counter */
#define QEI_MODE_x2_MATCH     0x1DFF /* QEI Quadrature Encoder Interface enabled (x2 mode) with position counter
reset by match (MAXCNT) */
#define QEI_MODE_x2_PULSE     0x1CFF /* QEI Quadrature Encoder Interface enabled (x2 mode) with Index Pulse reset
of position counter */
#define QEI_MODE_TIMER        0xF9FF /* QEI Starts 16-bit Timer */
#define QEI_MODE_OFF          0xF8FF /* QEI Quadrature Encoder Interface/Timer off */

#define QEI_UP_COUNT          0xFFFF /* QEI Position Counter Direction +VE */
#define QEI_DOWN_COUNT        0xF7FF /* QEI Position Counter Direction -VE */

#define QEI_IDLE_STOP         0xFFFF /* QEI Discontinue module operation when device enters a idle mode. */
#define QEI_IDLE_CON          0xDFFF /* QEI Continue module operation in idle mode */

/* defines for the DFLLTCN register */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define QEI_QE_CLK_DIVIDE_1_1    0xFF8F /* QEI QEA/QEB Digital Filter Clock Divide Select Bits */
#define QEI_QE_CLK_DIVIDE_1_2    0xFF9F
#define QEI_QE_CLK_DIVIDE_1_4    0xFFAF
#define QEI_QE_CLK_DIVIDE_1_16   0xFFBF
#define QEI_QE_CLK_DIVIDE_1_32   0xFFCF
#define QEI_QE_CLK_DIVIDE_1_64   0xFFDF
#define QEI_QE_CLK_DIVIDE_1_128  0xFFEF
#define QEI_QE_CLK_DIVIDE_1_256  0xFFFF /* QEI QEA/QEB Digital Filter Clock Divide Select Bits */

#define QEI_QE_OUT_ENABLE        0xFFFF /* QEI QEA/QEB Digital Filter Clock Divide Select Bits */
#define QEI_QE_OUT_DISABLE      0xFF7F /* QEI QEA/QEB Digital Filter Clock Divide Select Bits */

#define POS_CNT_ERR_INT_ENABLE   0xFFFF /* Enable interrupt due to position count errors */
#define POS_CNT_ERR_INT_DISABLE 0xFEFF /* Disable interrupt due to position count errors */

#define MATCH_INDEX_PHASEB_HIGH 0xEFFF
#define MATCH_INDEX_PHASEB_LOW  0xFBFF

#define MATCH_INDEX_PHASEA_HIGH 0xFFFF
#define MATCH_INDEX_PHASEA_LOW  0xFDFE

#define MATCH_INDEX_INPUT_PHASEB 0xF1FF
#define MATCH_INDEX_INPUT_PHASEA 0xFBFF

#define MATCH_INDEX_INPUT_HIGH   0xEFFF
#define MATCH_INDEX_INPUT_LOW    0xFDFE

/* Setting the priority of QEI interrupt */
#define QEI_INT_PRI_0            0xFF8
#define QEI_INT_PRI_1            0xFF9
#define QEI_INT_PRI_2            0xFFA
#define QEI_INT_PRI_3            0xFFB
#define QEI_INT_PRI_4            0xFFC
#define QEI_INT_PRI_5            0xFFD
#define QEI_INT_PRI_6            0xFFE
#define QEI_INT_PRI_7            0xFFFF

/* Enable / Disable QEI interrupt */
#define QEI_INT_ENABLE           0xFFFF /* Set the Interrupt enable bit */
#define QEI_INT_DISABLE          0xFF7F /* Clear the Interrupt enable bit */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* Macros to Enable/Disable interrupts and set Interrupt priority of QEI */
#define EnableIntQEI      asm("BSET IEC2,#8")
#define DisableIntQEI    asm("BCLR IEC2,#8")
#define SetPriorityIntQEI(priority)  (IPC10bits.QEIP = priority)

/* QEI Function Prototypes */

void OpenQEI(unsigned int config1, unsigned int config2) __attribute__((section(".libperi"))); /* Configure QEI */

void CloseQEI(void) __attribute__((section(".libperi"))); /* Disables the QEI module */

void ConfigIntQEI(unsigned int) __attribute__((section(".libperi"))); /* QEI interrupt configuration */

unsigned int ReadQEI(void) __attribute__((section(".libperi"))); /* Read QEI result */

void WriteQEI(unsigned int position) __attribute__((section(".libperi"))); /* Write QEI result */

#endif

#endif /* _QEI_H */
.....
/* Header for Timer module library functions
.....
#ifndef TIMER_H
#define TIMER_H

/* List of SFRs for Timers */
/* This list contains the SFRs with default (POR) values to be used for configuring Timers */
/* The user can modify this based on the requirement */
#define TMRx_VALUE      0x0000
#define PRx_VALUE       0xFFFF
#define TxCON_VALUE     0x0000

/* Timer1 Control Register (T1CON) Bit Defines */

#define T1_ON           0xffff /* Timer1 ON */
#define T1_OFF          0x7fff /* Timer1 OFF */

#define T1_IDLE_CON     0xdfff /* operate during sleep */
#define T1_IDLE_STOP    0xffff /* stop operation during sleep */

#define T1_GATE_ON      0xffff /* Timer Gate time accumulation enabled */
#define T1_GATE_OFF     0xffff /* Timer Gate time accumulation disabled */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการวิจัยเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define T1_PS_1_1    0xffcf /* Prescaler 1:1 */
#define T1_PS_1_8    0xffdf /* 1:8 */
#define T1_PS_1_64   0xffef /* 1:64 */
#define T1_PS_1_256  0xffff /* 1:256 */

#define T1_SYNC_EXT_ON  0xf0ff /* Synch external clk input */
#define T1_SYNC_EXT_OFF 0xf0fb /* Do not synch external clk input */

#define T1_SOURCE_EXT  0xf0ff /* External clock source */
#define T1_SOURCE_INT  0xf0fd /* Internal clock source */

/* defines for Timer Interrupts */

#define T1_INT_PRIOR_7 0xffff /* 111 = Interrupt is priority 7 */
#define T1_INT_PRIOR_6 0xfffe /* 110 = Interrupt is priority 6 */
#define T1_INT_PRIOR_5 0xfffd /* 101 = Interrupt is priority 5 */
#define T1_INT_PRIOR_4 0xfffc /* 100 = Interrupt is priority 4 */
#define T1_INT_PRIOR_3 0xfffb /* 011 = Interrupt is priority 3 */
#define T1_INT_PRIOR_2 0xfffa /* 010 = Interrupt is priority 2 */
#define T1_INT_PRIOR_1 0xfff9 /* 001 = Interrupt is priority 1 */
#define T1_INT_PRIOR_0 0xfff8 /* 000 = Interrupt is priority 0 */

#define T1_INT_ON      0xf0ff /* Interrupt Enable */
#define T1_INT_OFF     0xf0f7 /* Interrupt Disable */

/* Macros to Enable/Disable interrupts and set Interrupt priority of Timers 1 */
#define EnableIntT1    asm("BSET IEC0,#3")
#define DisableIntT1   asm("BCLR IEC0,#3")
#define SetPriorityIntT1(priority) (IPC0bits.T1IP = priority)

/* Timer1 Function Prototypes */

/* OpenTimer1 */
void OpenTimer1( unsigned int config, unsigned int period) __attribute__((section
(".libperi")));

/* CloseTimer1 */
void CloseTimer1(void) __attribute__((section(".libperi")));

/* ReadTimer1 */
unsigned int ReadTimer1(void) __attribute__((section(".libperi")));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* WriteTimer1 */
void WriteTimer1( unsigned int timer) __attribute__((section (".libperi")));

/* Config Int Timer1 */
void ConfigIntTimer1(unsigned int config) __attribute__((section (".libperi")));

/* Timer2 Control Register (T2CON) Bit Defines */

#define T2_ON          0xffff /* Timer2 ON */
#define T2_OFF        0x7fff /* Timer2 OFF */

#define T2_IDLE_CON   0xdfff /* operate during sleep */
#define T2_IDLE_STOP  0xffff /* stop operation during sleep */

#define T2_GATE_ON    0xffff /* Timer2 Gate time accumulation enabled */
#define T2_GATE_OFF   0xffbf /* Timer2 Gate time accumulation disabled */

#define T2_PS_1_1     0xffcf /* Prescaler 1:1 */
#define T2_PS_1_8     0xffdf /* Prescaler 1:8 */
#define T2_PS_1_64    0xffef /* Prescaler 1:64 */
#define T2_PS_1_256   0xffff /* Prescaler 1:256 */

#define T2_32BIT_MODE_ON  0xffff /* Timer 2 and Timer 3 form a 32 bit Timer */
#define T2_32BIT_MODE_OFF 0xffff

#define T2_SOURCE_EXT  0xffff /* External clock source */
#define T2_SOURCE_INT  0xffff /* Internal clock source */

/* defines for Timer Interrupts */
#define T2_INT_PRIOR_7  0xffff /* 111 = Interrupt is priority 7 */
#define T2_INT_PRIOR_6  0xffff /* 110 = Interrupt is priority 6 */
#define T2_INT_PRIOR_5  0xffff /* 101 = Interrupt is priority 5 */
#define T2_INT_PRIOR_4  0xffff /* 100 = Interrupt is priority 4 */
#define T2_INT_PRIOR_3  0xffff /* 011 = Interrupt is priority 3 */
#define T2_INT_PRIOR_2  0xffff /* 010 = Interrupt is priority 2 */
#define T2_INT_PRIOR_1  0xffff /* 001 = Interrupt is priority 1 */
#define T2_INT_PRIOR_0  0xffff /* 000 = Interrupt is priority 0 */

#define T2_INT_ON       0xffff /* Interrupt Enable */
#define T2_INT_OFF      0xffff /* Interrupt Disable */

/* Macros to Enable/Disable interrupts and set Interrupt priority of Timers 2*/
#define EnableIntT2(asm("BSET IFC0,#6"))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define DisableIntT2      asm("BCLR IEC0,#6")
#define SetPriorityIntT2(priority)  (IPC1bits.T2IP = priority)

/* Timer2 Function Prototypes */

/* OpenTimer2 */
void OpenTimer2(unsigned int config, unsigned int period) __attribute__((section(
".libperi")));

/* CloseTimer2 */
void CloseTimer2(void) __attribute__((section (".libperi")));

/* ReadTimer2 */
unsigned int ReadTimer2(void) __attribute__((section (".libperi")));

/* WriteTimer2 */
void WriteTimer2( unsigned int timer) __attribute__((section (".libperi")));

/* ConfigIntTimer2 */
void ConfigIntTimer2(unsigned int i) __attribute__((section (".libperi")));

/* Timer3 Control Register (T3CON) Bit Defines */

#define T3_ON      0xffff /* Timer3 ON */
#define T3_OFF     0x7fff /* Timer3 OFF */

#define T3_IDLE_CON  0x0fff /* operate during sleep */
#define T3_IDLE_STOP 0xffff /* stop operation during sleep */

#define T3_GATE_ON   0xffff /* Timer3 Gate time accumulation enabled */
#define T3_GATE_OFF  0xffbf /* Timer Gate time accumulation disabled */

#define T3_PS_1_1    0xffcf /* Prescaler 1:1 */
#define T3_PS_1_8    0xffdf /* 1:8 */
#define T3_PS_1_64    0xffef /* 1:64 */
#define T3_PS_1_256  0xffff /* 1:256 */
#define T3_SOURCE_EXT 0xffff /* External clock source */
#define T3_SOURCE_INT 0xffff /* Internal clock source */

/* defines for Timer Interrupts */
#define T3_INT_PRIOR_7  0xffff /* 111 = Interrupt is priority 7 */
#define T3_INT_PRIOR_6  0xffff /* 110 = Interrupt is priority 6 */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้เพื่อใช้สำหรับงานวิจัยเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define T3_INT_PRIOR_5  0xffff /* 101 = Interrupt is priority 5 */
#define T3_INT_PRIOR_4  0xfffc /* 100 = Interrupt is priority 4 */
#define T3_INT_PRIOR_3  0xfffb /* 011 = Interrupt is priority 3 */
#define T3_INT_PRIOR_2  0xfffa /* 010 = Interrupt is priority 2 */
#define T3_INT_PRIOR_1  0xffff /* 001 = Interrupt is priority 1 */
#define T3_INT_PRIOR_0  0xffff /* 000 = Interrupt is priority 0 */

#define T3_INT_ON      0xffff /* Interrupt Enable */
#define T3_INT_OFF     0xffff /* Interrupt Disable */

/* Macros to Enable/Disable interrupts and set Interrupt priority of Timers 3*/
#define EnableIntT3      asm("BSET IEC0,#7")
#define DisableIntT3     asm("BCLR IEC0,#7")
#define SetPriorityIntT3(priority)  ((IPC1bits.T3IP = priority)

/* Timer3 Function Prototypes */

/* OpenTimer3 */
void OpenTimer3(unsigned int config, unsigned int timer) __attribute__((section(
".libperi")));

/* CloseTimer3 */
void CloseTimer3(void) __attribute__((section(".libperi")));

/* ReadTimer3 */
unsigned int ReadTimer3(void) __attribute__((section(".libperi")));

/* WriteTimer3 */
void WriteTimer3(unsigned int timer) __attribute__((section(".libperi")));

/* ConfigIntTimer3 */
void ConfigIntTimer3(unsigned int config) __attribute__((section(".libperi")));

/* Timer 32 mode using Timer 2 and Timer3*/

/* Timer 32 bit mode Prototypes*/

/* OpenTimer23 */
void OpenTimer23(unsigned int config, unsigned long period) __attribute__((section(
".libperi")));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void CloseTimer23(void) __attribute__((section(".libperi")));

/* ReadTimer23 */
unsigned long ReadTimer23(void) __attribute__((section(".libperi")));

/* WriteTimer23 */
void WriteTimer23(unsigned long timer) __attribute__((section(".libperi")));

/* Config Int Timer 23*/
void ConfigIntTimer23(unsigned int config) __attribute__((section(".libperi")));

/* These devices support Timers 4 and 5 */
#if defined(__dsPIC30F3010__) || defined(__dsPIC30F4012__) || defined(__dsPIC30F3011__) || \
    defined(__dsPIC30F4011__) || defined(__dsPIC30F6010__) || defined(__dsPIC30F5011__) || \
    defined(__dsPIC30F6011__) || defined(__dsPIC30F6012__) || defined(__dsPIC30F5013__) || \
    defined(__dsPIC30F6013__) || defined(__dsPIC30F6014__) || defined(__dsPIC30F4013__) || \
    defined(__dsPIC30F5015__) || defined(__dsPIC30F6010A__) || defined(__dsPIC30F6011A__) || \
    defined(__dsPIC30F6012A__) || defined(__dsPIC30F6013A__) || defined(__dsPIC30F6014A__) || \
    defined(__dsPIC30F5016__) || defined(__dsPIC30F6015__)

/* Timer4 Control Register (T4CON) Bit Defines */
#define T4_ON          0xffff /* Timer4 ON */
#define T4_OFF        0x7fff /* Timer4 OFF */

#define T4_IDLE_CON    0xdfff /* operate during sleep */
#define T4_IDLE_STOP  0xffff /* stop operation during sleep */

#define T4_GATE_ON    0xffff /* Timer Gate time accumulation enabled */
#define T4_GATE_OFF   0xffff /* Timer Gate time accumulation disabled */

#define T4_PS_1_1     0xffcf /* Prescaler 1:1 */
#define T4_PS_1_8     0xffdf /* 1:8 */
#define T4_PS_1_64    0xffef /* 1:64 */
#define T4_PS_1_256   0xffff /* 1:256 */

#define T4_SOURCE_EXT  0xffff /* External clock source */
#define T4_SOURCE_INT  0xfffd /* Internal clock source */

#define T4_32BIT_MODE_ON  0xffff /* Timer 4 and Timer 5 form a 32 bit Timer */
#define T4_32BIT_MODE_OFF 0xffff

/* defines for Timer Interrupts */
#define T4_INT_PRIOR_7 0xffff /* 111 Interrupt is priority 7 */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define T4_INT_PRIOR_6  0xfffe  /* 110 = Interrupt is priority 6 */
#define T4_INT_PRIOR_5  0xfffd  /* 101 = Interrupt is priority 5 */
#define T4_INT_PRIOR_4  0xfffc  /* 100 = Interrupt is priority 4 */
#define T4_INT_PRIOR_3  0xfffb  /* 011 = Interrupt is priority 3 */
#define T4_INT_PRIOR_2  0xfffa  /* 010 = Interrupt is priority 2 */
#define T4_INT_PRIOR_1  0xfff9  /* 001 = Interrupt is priority 1 */
#define T4_INT_PRIOR_0  0xfff8  /* 000 = Interrupt is priority 0 */

#define T4_INT_ON        0xffff  /* Interrupt Enable */
#define T4_INT_OFF       0xffff  /* Interrupt Disable */

/* Macros to Enable/Disable interrupts and set Interrupt priority of Timer 4 */
#define EnableIntT4      asm("BSL1 IEC1,#5")
#define DisableIntT4    asm("BCLR IEC1,#5")
#define SetPriorityIntT4(priority)  (IPC5bits.T4IP = priority)

/* Timer4 Function Declarations */

/* OpenTimer4
 * Configures Timer4
 */
void OpenTimer4(unsigned int config, unsigned int timer) __attribute__((section(
(".libperi"))));

/* CloseTimer4
 * Disables Timer4
 */
void CloseTimer4(void) __attribute__((section(".libperi")));

/* ReadTimer4
 * Reads Timer4
 */
unsigned int ReadTimer4(void) __attribute__((section(".libperi")));

/* WriteTimer4
 * Writes Timer4
 */
void WriteTimer4(unsigned int timer) __attribute__((section(".libperi")));

/* Config Int Timer 4*/
void ConfigIntTimer4(unsigned int config) __attribute__((section(".libperi")));

/* Timer5 Config Bit Defines */

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define T5_ON      0xffff /* Timer5 ON */
#define T5_OFF     0x7fff /* Timer5 OFF */

#define T5_IDLE_CON  0xdfff /* operate during sleep */
#define T5_IDLE_STOP 0xffff /* stop operation during sleep */

#define T5_GATE_ON   0xffff /* Timer Gate time accumulation enabled */
#define T5_GATE_OFF  0xffff /* Timer Gate time accumulation disabled */

#define T5_PS_1_1    0xffcf /* Prescaler 1:1 */
#define T5_PS_1_8    0xffdf /* 1:8 */
#define T5_PS_1_64   0xffef /* 1:64 */
#define T5_PS_1_256  0xffff /* 1:256 */

#define T5_SOURCE_EXT 0xffff /* External clock source */
#define T5_SOURCE_INT 0xffff /* Internal clock source */

/* defines for Timer Interrupts */
#define T5_INT_PRIOR_7 0xffff /* 111 = Interrupt is priority 7 */
#define T5_INT_PRIOR_6 0xffff /* 110 = Interrupt is priority 6 */
#define T5_INT_PRIOR_5 0xffff /* 101 = Interrupt is priority 5 */
#define T5_INT_PRIOR_4 0xffff /* 100 = Interrupt is priority 4 */
#define T5_INT_PRIOR_3 0xffff /* 011 = Interrupt is priority 3 */
#define T5_INT_PRIOR_2 0xffff /* 010 = Interrupt is priority 2 */
#define T5_INT_PRIOR_1 0xffff /* 001 = Interrupt is priority 1 */
#define T5_INT_PRIOR_0 0xffff /* 000 = Interrupt is priority 0 */

#define T5_INT_ON     0xffff /* Interrupt Enable */
#define T5_INT_OFF    0xffff /* Interrupt Disable */

/* Macros to Enable/Disable interrupts and set Interrupt priority of Timer 5 */
#define EnableIntT5      asm("BSET IEC1,#6")
#define DisableIntT5     asm("BCLR IEC1,#6")
#define SetPriorityIntT5(priority) (IPC5bits.T5IP = priority)

/* Timer5 Function Declarations */

/* OpenTimer5
 * Configures Timer5
 */
void OpenTimer5(unsigned int config, unsigned int timer) __attribute__((section

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* CloseTimer5
 * Disables Timer5
 */
void CloseTimer5(void) __attribute__((section(".libperi")));

/* ReadTimer5
 * Reads Timer5
 */
unsigned int ReadTimer5(void) __attribute__((section(".libperi")));

/* WriteTimer5
 * Writes Timer5
 */
void WriteTimer5(unsigned int timer) __attribute__((section(".libperi")));

/* Config Int Timer 5*/
void ConfigIntTimer5(unsigned int config) __attribute__((section(".libperi")));

/* Timer 45 mode using Timer 4 and Timer 5 */

/* OpenTimer45 */
void OpenTimer45(unsigned int config, unsigned long timer) __attribute__((section(
(".libperi")));

/* CloseTimer45 */
void CloseTimer45(void) __attribute__((section(".libperi")));

/* ReadTimer45 */
unsigned long ReadTimer45(void) __attribute__((section(".libperi")));

/* WriteTimer45 */
void WriteTimer45(unsigned long timer) __attribute__((section(".libperi")));

/* Config Int Timer 45*/
void ConfigIntTimer45(unsigned int config) __attribute__((section(".libperi")));
#endif

#endif

//-----//
// Program : Centrifuge

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Frequency           : 4.00 MHz at PLL 4x
// Filename            : Centrifuge_Dspic30F4011.c
// C compiler          : C30 Compiler by Microchip Technology
//-----//
#include<p30f4011.h>    // Header file for dsPIC30F4011
#include<timer.h>      // Module function for Timer
#include<lcd4011.h>    // Module function for LCD display 4 bit mode
#include<qei.h>
#define PWM_H LATEbits.LATE0
#define PWM_L LATEbits.LATE1
#define button1 PORTFbits.RF2
#define button2 PORTFbits.RF3
#define button3 PORTFbits.RF4
unsigned char phase=0,frequency_range=0;
unsigned int timer1_value,timer2_value,periods; // Keep interval for timer1
unsigned int max_value=0,d,b;
unsigned int round_new=0;
unsigned int y=0,x=0;
unsigned int rpm,value,rpm_fix,n=0;
unsigned char buff[4],buff2[2],buff3[2];
float frequency;
//-----//
//-----// Interrupt service routine for timer1
//-----//
void_ISR_T2Interrupt(void)
{
    value = ReadQEI();
    if ((value >= 0) && (value <= 100)) {x = 60;}
    else if ((value > 100) && (value <= 200)) {x = 120;}
    else if ((value > 200) && (value <= 300)) {x = 180;}
    else if ((value > 300) && (value <= 400)) {x = 240;}
    else if ((value > 400) && (value <= 500)) {x = 300;}
    else if ((value > 500) && (value <= 600)) {x = 360;}
    else if ((value > 600) && (value <= 700)) {x = 420;}
    else if ((value > 700) && (value <= 800)) {x = 480;}
    else if ((value > 800) && (value <= 900)) {x = 540;}
    else if ((value > 900) && (value <= 1000)) {x = 600;}
    rpm = (round_new*600)+(x-y);
    round_new = 0;
    y = x;
    WriteTimer2(0);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IFS0bits.T2IF = 0;
}
void _ISR_QEIInterrupt(void)
{
    round_new++;
    POSCNT = 0;
    IFS2bits.QE1IF = 0;
}
void qei_init(void)
{
    unsigned int config1;

    ADPCFG = 0xFFFF;

    ConfigIntQEI(QEI_INT_PRI_3 & QEI_INT_ENABLE);

    POSCNT = 0;
    MAXCNT = 999;

    config1
    (QEI_DIR_SEL_QEB& // QEB Pin State Defines Position Counter
    QEI_INT_CLK& // Internal clock (FOSC/4)
    QEI_INDEX_RESET_DISABLE& // Index Pulse does not reset Position
    QEI_CLK_PRESCALE_1 & // QEI Timer Input Clock Prescale Select Bits
    QEI_GATED_ACC_DISABLE& // Timer gated time accumulation
    QEI_NORMAL_IO& // QEI Position Counter Direction
    QEI_INPUTS_NOSWAP& // QEI Phase A and Phase B inputs not swapped
    QEI_MODE_x4_MATCH& // QEI Quadrature Encoder Interface enabled (x4
    // with position
    counter reset by match (MAXCNT)
    QEI_UP_COUNT& // QEI Position Counter Direction +VE
    QEI_IDLE_CON); // QEI Continue module operation in idle mode

    OpenQEI(config1,0);
}
void timer2_init(void)
{
    ConfigIntTimer2(T2_INT_PRIOR_2 & T2_INT_ON);
}

```

เอกสารนี้เป็นเอกสารที่เผยแพร่ในนามของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

WriteTimer2(0);
timer2_value = 25000;
OpenTimer2(T2_ON &                                // Start timer2
            T2_GATE_OFF &                          // Disable gate pin for timer2
            T2_IDLE_STOP &                          // Stop timer2 in idle mode
            T2_PS_1_8 &                             // Prescaler 1:8
            T2_32BIT_MODE_ON &                      // Enable mode 32 bit Timer2
            T2_SOURCE_INT,timer2_value);            // Wait till the timer matches with the
period value
}
void_ISR_T1Interrupt(void)
{
    WriteTimer1(0);                                // Clear count value at TMR1 register
    IFS0bits.T1IF = 0;                              // Clear Timer interrupt flag
    switch(frequency_range)
    {
    case 0:
    {
        switch(phase)
        {
        case 0: {PWM_H = 1,PWM_L = 0,phase = 1; }break;
        case 1: {PWM_H = 1,PWM_L = 1,phase = 2; }break;
        case 2: {PWM_H = 1,PWM_L = 1,phase = 3; }break;
        case 3: {PWM_H = 1,PWM_L = 1,phase = 4; }break;
        case 4: {PWM_H = 1,PWM_L = 1,phase = 5; }break;
        case 5: {PWM_H = 1,PWM_L = 1,phase = 6; }break;
        case 6: {PWM_H = 1,PWM_L = 1,phase = 7; }break;
        case 7: {PWM_H = 1,PWM_L = 1,phase = 8; }break;
        case 8: {PWM_H = 1,PWM_L = 1,phase = 9; }break;
        case 9: {PWM_H = 1,PWM_L = 1,phase = 10; }break;
        case 10: {PWM_H = 1,PWM_L = 1,phase = 11; }break;
        case 11: {PWM_H = 1,PWM_L = 1,phase = 12; }break;
        case 12: {PWM_H = 1,PWM_L = 1,phase = 13; }break;
        case 13: {PWM_H = 1,PWM_L = 1,phase = 14; }break;
        case 14: {PWM_H = 1,PWM_L = 1,phase = 15; }break;
        case 15: {PWM_H = 1,PWM_L = 1,phase = 16; }break;
        case 16: {PWM_H = 1,PWM_L = 1,phase = 17; }break;
        case 17: {PWM_H = 1,PWM_L = 1,phase = 18; }break;
        case 18: {PWM_H = 0,PWM_L = 1,phase = 19; }break;
        case 19: {PWM_H = 0,PWM_L = 0,phase = 20; }break;
        case 20: {PWM_H = 0,PWM_L = 0,phase = 21; }break;
        case 21: {PWM_H = 0,PWM_L = 0,phase = 22; }break;
        case 22: {PWM_H = 0,PWM_L = 0,phase = 23; }break;

```

เอกสารนี้เป็นเอกสารที่เผยแพร่ฟรี หรือมีค่าใช้จ่ายเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 23: {PWM_H = 0,PWM_L = 0,phase = 24; }break;
case 24: {PWM_H = 0,PWM_L = 0,phase = 25; }break;
case 25: {PWM_H = 0,PWM_L = 0,phase = 26; }break;
case 26: {PWM_H = 0,PWM_L = 0,phase = 27; }break;
case 27: {PWM_H = 0,PWM_L = 0,phase = 28; }break;
case 28: {PWM_H = 0,PWM_L = 0,phase = 29; }break;
case 29: {PWM_H = 0,PWM_L = 0,phase = 30; }break;
case 30: {PWM_H = 0,PWM_L = 0,phase = 31; }break;
case 31: {PWM_H = 0,PWM_L = 0,phase = 32; }break;
case 32: {PWM_H = 0,PWM_L = 0,phase = 33; }break;
case 33: {PWM_H = 0,PWM_L = 0,phase = 34; }break;
case 34: {PWM_H = 0,PWM_L = 0,phase = 35; }break;
case 35: {PWM_H = 0,PWM_L = 0,phase = 0; }break;
}
break;
}
case 1:
{
switch(phase)
{
case 0: {PWM_H = 1,PWM_L = 0,phase = 1; }break;
case 1: {PWM_H = 1,PWM_L = 0,phase = 2; }break;
case 2: {PWM_H = 1,PWM_L = 0,phase = 3; }break;
case 3: {PWM_H = 1,PWM_L = 0,phase = 4; }break;
case 4: {PWM_H = 1,PWM_L = 1,phase = 5; }break;
case 5: {PWM_H = 1,PWM_L = 1,phase = 6; }break;
case 6: {PWM_H = 1,PWM_L = 1,phase = 7; }break;
case 7: {PWM_H = 1,PWM_L = 1,phase = 8; }break;
case 8: {PWM_H = 1,PWM_L = 1,phase = 9; }break;
case 9: {PWM_H = 1,PWM_L = 1,phase = 10; }break;
case 10: {PWM_H = 1,PWM_L = 1,phase = 11; }break;
case 11: {PWM_H = 1,PWM_L = 1,phase = 12; }break;
case 12: {PWM_H = 1,PWM_L = 1,phase = 13; }break;
case 13: {PWM_H = 1,PWM_L = 1,phase = 14; }break;
case 14: {PWM_H = 1,PWM_L = 1,phase = 15; }break;
case 15: {PWM_H = 1,PWM_L = 1,phase = 16; }break;
case 16: {PWM_H = 1,PWM_L = 1,phase = 17; }break;
case 17: {PWM_H = 1,PWM_L = 1,phase = 18; }break;
case 18: {PWM_H = 0,PWM_L = 1,phase = 19; }break;
case 19: {PWM_H = 0,PWM_L = 1,phase = 20; }break;
case 20: {PWM_H = 0,PWM_L = 1,phase = 21; }break;
case 21: {PWM_H = 0,PWM_L = 1,phase = 22; }break;
case 22: {PWM_H = 0,PWM_L = 0,phase = 23; }break;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของบริษัทฯ เพื่อใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 23: {PWM_H = 0,PWM_L = 0,phase = 24; }break;
case 24: {PWM_H = 0,PWM_L = 0,phase = 25; }break;
case 25: {PWM_H = 0,PWM_L = 0,phase = 26; }break;
case 26: {PWM_H = 0,PWM_L = 0,phase = 27; }break;
case 27: {PWM_H = 0,PWM_L = 0,phase = 28; }break;
case 28: {PWM_H = 0,PWM_L = 0,phase = 29; }break;
case 29: {PWM_H = 0,PWM_L = 0,phase = 30; }break;
case 30: {PWM_H = 0,PWM_L = 0,phase = 31; }break;
case 31: {PWM_H = 0,PWM_L = 0,phase = 32; }break;
case 32: {PWM_H = 0,PWM_L = 0,phase = 33; }break;
case 33: {PWM_H = 0,PWM_L = 0,phase = 34; }break;
case 34: {PWM_H = 0,PWM_L = 0,phase = 35; }break;
case 35: {PWM_H = 0,PWM_L = 0,phase = 0; }break;
}
break;
}
case 2:
{
switch(phase)
{
case 0: {PWM_H = 1,PWM_L = 0,phase = 1; }break;
case 1: {PWM_H = 1,PWM_L = 0,phase = 2; }break;
case 2: {PWM_H = 1,PWM_L = 0,phase = 3; }break;
case 3: {PWM_H = 1,PWM_L = 0,phase = 4; }break;
case 4: {PWM_H = 1,PWM_L = 0,phase = 5; }break;
case 5: {PWM_H = 1,PWM_L = 0,phase = 6; }break;
case 6: {PWM_H = 1,PWM_L = 1,phase = 7; }break;
case 7: {PWM_H = 1,PWM_L = 1,phase = 8; }break;
case 8: {PWM_H = 1,PWM_L = 1,phase = 9; }break;
case 9: {PWM_H = 1,PWM_L = 1,phase = 10; }break;
case 10: {PWM_H = 1,PWM_L = 1,phase = 11; }break;
case 11: {PWM_H = 1,PWM_L = 1,phase = 12; }break;
case 12: {PWM_H = 1,PWM_L = 1,phase = 13; }break;
case 13: {PWM_H = 1,PWM_L = 1,phase = 14; }break;
case 14: {PWM_H = 1,PWM_L = 1,phase = 15; }break;
case 15: {PWM_H = 1,PWM_L = 1,phase = 16; }break;
case 16: {PWM_H = 1,PWM_L = 1,phase = 17; }break;
case 17: {PWM_H = 1,PWM_L = 1,phase = 18; }break;
case 18: {PWM_H = 0,PWM_L = 1,phase = 19; }break;
case 19: {PWM_H = 0,PWM_L = 1,phase = 20; }break;
case 20: {PWM_H = 0,PWM_L = 1,phase = 21; }break;
case 21: {PWM_H = 0,PWM_L = 1,phase = 22; }break;
case 22: {PWM_H = 0,PWM_L = 1,phase = 23; }break;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของบริษัทฯ เพื่อใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 23: {PWM_H = 0,PWM_L = 1,phase = 24; }break;
case 24: {PWM_H = 0,PWM_L = 0,phase = 25; }break;
case 25: {PWM_H = 0,PWM_L = 0,phase = 26; }break;
case 26: {PWM_H = 0,PWM_L = 0,phase = 27; }break;
case 27: {PWM_H = 0,PWM_L = 0,phase = 28; }break;
case 28: {PWM_H = 0,PWM_L = 0,phase = 29; }break;
case 29: {PWM_H = 0,PWM_L = 0,phase = 30; }break;
case 30: {PWM_H = 0,PWM_L = 0,phase = 31; }break;
case 31: {PWM_H = 0,PWM_L = 0,phase = 32; }break;
case 32: {PWM_H = 0,PWM_L = 0,phase = 33; }break;
case 33: {PWM_H = 0,PWM_L = 0,phase = 34; }break;
case 34: {PWM_H = 0,PWM_L = 0,phase = 35; }break;
case 35: {PWM_H = 0,PWM_L = 0,phase = 0; }break;
}
break;
}
case 3:
{
switch(phase)
{
case 0: {PWM_H = 1,PWM_L = 0,phase = 1; }break;
case 1: {PWM_H = 1,PWM_L = 0,phase = 2; }break;
case 2: {PWM_H = 1,PWM_L = 0,phase = 3; }break;
case 3: {PWM_H = 1,PWM_L = 0,phase = 4; }break;
case 4: {PWM_H = 1,PWM_L = 0,phase = 5; }break;
case 5: {PWM_H = 1,PWM_L = 0,phase = 6; }break;
case 6: {PWM_H = 1,PWM_L = 0,phase = 7; }break;
case 7: {PWM_H = 1,PWM_L = 0,phase = 8; }break;
case 8: {PWM_H = 1,PWM_L = 1,phase = 9; }break;
case 9: {PWM_H = 1,PWM_L = 1,phase = 10; }break;
case 10: {PWM_H = 1,PWM_L = 1,phase = 11; }break;
case 11: {PWM_H = 1,PWM_L = 1,phase = 12; }break;
case 12: {PWM_H = 1,PWM_L = 1,phase = 13; }break;
case 13: {PWM_H = 1,PWM_L = 1,phase = 14; }break;
case 14: {PWM_H = 1,PWM_L = 1,phase = 15; }break;
case 15: {PWM_H = 1,PWM_L = 1,phase = 16; }break;
case 16: {PWM_H = 1,PWM_L = 1,phase = 17; }break;
case 17: {PWM_H = 1,PWM_L = 1,phase = 18; }break;
case 18: {PWM_H = 0,PWM_L = 1,phase = 19; }break;
case 19: {PWM_H = 0,PWM_L = 1,phase = 20; }break;
case 20: {PWM_H = 0,PWM_L = 1,phase = 21; }break;
case 21: {PWM_H = 0,PWM_L = 1,phase = 22; }break;
case 22: {PWM_H = 0,PWM_L = 1,phase = 23; }break;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของบริษัทฯ ซึ่งเนื้อหาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 23: {PWM_H = 0,PWM_L = 1,phase = 24; }break;
case 24: {PWM_H = 0,PWM_L = 1,phase = 25; }break;
case 25: {PWM_H = 0,PWM_L = 1,phase = 26; }break;
case 26: {PWM_H = 0,PWM_L = 0,phase = 27; }break;
case 27: {PWM_H = 0,PWM_L = 0,phase = 28; }break;
case 28: {PWM_H = 0,PWM_L = 0,phase = 29; }break;
case 29: {PWM_H = 0,PWM_L = 0,phase = 30; }break;
case 30: {PWM_H = 0,PWM_L = 0,phase = 31; }break;
case 31: {PWM_H = 0,PWM_L = 0,phase = 32; }break;
case 32: {PWM_H = 0,PWM_L = 0,phase = 33; }break;
case 33: {PWM_H = 0,PWM_L = 0,phase = 34; }break;
case 34: {PWM_H = 0,PWM_L = 0,phase = 35; }break;
case 35: {PWM_H = 0,PWM_L = 0,phase = 0; }break;
}
break;
}
case 4:
{
switch(phase)
{
case 0: {PWM_H = 1,PWM_L = 0,phase = 1; }break;
case 1: {PWM_H = 1,PWM_L = 0,phase = 2; }break;
case 2: {PWM_H = 1,PWM_L = 0,phase = 3; }break;
case 3: {PWM_H = 1,PWM_L = 0,phase = 4; }break;
case 4: {PWM_H = 1,PWM_L = 0,phase = 5; }break;
case 5: {PWM_H = 1,PWM_L = 0,phase = 6; }break;
case 6: {PWM_H = 1,PWM_L = 0,phase = 7; }break;
case 7: {PWM_H = 1,PWM_L = 0,phase = 8; }break;
case 8: {PWM_H = 1,PWM_L = 0,phase = 9; }break;
case 9: {PWM_H = 1,PWM_L = 1,phase = 10; }break;
case 10: {PWM_H = 1,PWM_L = 1,phase = 11; }break;
case 11: {PWM_H = 1,PWM_L = 1,phase = 12; }break;
case 12: {PWM_H = 1,PWM_L = 1,phase = 13; }break;
case 13: {PWM_H = 1,PWM_L = 1,phase = 14; }break;
case 14: {PWM_H = 1,PWM_L = 1,phase = 15; }break;
case 15: {PWM_H = 1,PWM_L = 1,phase = 16; }break;
case 16: {PWM_H = 1,PWM_L = 1,phase = 17; }break;
case 17: {PWM_H = 1,PWM_L = 1,phase = 18; }break;
case 18: {PWM_H = 0,PWM_L = 1,phase = 19; }break;
case 19: {PWM_H = 0,PWM_L = 1,phase = 20; }break;
case 20: {PWM_H = 0,PWM_L = 1,phase = 21; }break;
case 21: {PWM_H = 0,PWM_L = 1,phase = 22; }break;
case 22: {PWM_H = 0,PWM_L = 1,phase = 23; }break;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 23: {PWM_H = 0,PWM_L = 1,phase = 24; }break;
case 24: {PWM_H = 0,PWM_L = 1,phase = 25; }break;
case 25: {PWM_H = 0,PWM_L = 1,phase = 26; }break;
case 26: {PWM_H = 0,PWM_L = 1,phase = 27; }break;
case 27: {PWM_H = 0,PWM_L = 0,phase = 28; }break;
case 28: {PWM_H = 0,PWM_L = 0,phase = 29; }break;
case 29: {PWM_H = 0,PWM_L = 0,phase = 30; }break;
case 30: {PWM_H = 0,PWM_L = 0,phase = 31; }break;
case 31: {PWM_H = 0,PWM_L = 0,phase = 32; }break;
case 32: {PWM_H = 0,PWM_L = 0,phase = 33; }break;
case 33: {PWM_H = 0,PWM_L = 0,phase = 34; }break;
case 34: {PWM_H = 0,PWM_L = 0,phase = 35; }break;
case 35: {PWM_H = 0,PWM_L = 0,phase = 0; }break;
}
break;
}
}
}
//-----
//----- config timer1 for PWM signal
//-----
void closepwm(void)
{
CloseTimer1();
PWM_H = 0,PWM_L = 0;
}
//-----
//----- config timer1 for PWM signal
//-----

void openpwm(void)
{
ConfigIntTimer1(T1_INT_PRIOR_1 &
                T1_INT_ON);

WriteTimer1(0);

periods = 2777/frequency; //periodPWM(1/f) = PreScaler(64)*4MHZ*6(plate)*timer1_value
timer1_value = periods; // Load value Interval 50 ms (Prescaler(8)*timer1_value(x) /Fcy(4 MHz) = Period of
PWM)

OpenTimer1(T1_ON & // Start timer1
            TE_GATE_OFF & // Disable gate pin for timer1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

T1_IDLE_STOP & // Stop timer in idle mode
T1_PS_1_8 & // Prescaler 1:64
T1_SYNC_EXT_OFF & // Disable sync external source
T1_SOURCE_INT,timer1_value); // Wait till the timer matches with the period value
}
//-----//
//-----delay for 1 milisec per ms-----//
//-----//

```

```
void delay_motor(unsigned int ms)
```

```

{
    unsigned int x,a;
    for(x=0;x<ms;x++)
    {
        for(a=0;a<443;a++);
    }
}
//-----//
//-----Main Program-----//
//-----//

```

```
int main(void)
```

```

{
    TRISEbits.TRISE0 = 0;
    TRISEbits.TRISE1 = 0;
    TRISFbits.TRISF2 = 1;
    TRISFbits.TRISF3 = 1;
    TRISFbits.TRISF4 = 1;
    qei_init();
    lcd_init();
    while(1)
    {
        while(1)
        {
            if(button1 == 0)
            {
                delay_motor(100);
                if(button1 == 0)
                {
                    lcd_puts(0x80,"SpeedSet:");
                    lcd_puts(0xC0,"Sp: F: Hz");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

buff[0] = ((rpm_fix%10000)/1000) | 0x30;
buff[1] = (((rpm_fix%10000)%1000)/100) | 0x30;
buff[2] = (((rpm_fix%10000)%1000)%100)/10 | 0x30;
buff[3] = (((rpm_fix%10000)%1000)%100)%10 | 0x30;
lcd_puts(0x89,buff);
lcd_puts(0x8D,"rpm");

timer2_init();
break;
}
}
if(button2 == 0)
{
delay_motor(100);
if(button2 == 0)
{
lcd_puts(0x80,"SpeedSet.");
lcd_puts(0xC0,"Sp: F: Hz");
rpm_fix = 1500;
buff[0] = ((rpm_fix%10000)/1000) | 0x30;
buff[1] = (((rpm_fix%10000)%1000)/100) | 0x30;
buff[2] = (((rpm_fix%10000)%1000)%100)/10 | 0x30;
buff[3] = (((rpm_fix%10000)%1000)%100)%10 | 0x30;
lcd_puts(0x89,buff);
lcd_puts(0x8D,"rpm");

timer2_init();
break;
}
}

frequency = 1;
frequency_range = 0;

openpwm();
delay_motor(500);

while(1)
{
if ((rpm < rpm_fix)&&(frequency < 50))
{
if((rpm_fix-rpm <= 3500 )&&(rpm_fix-rpm > 3000 ))
{
frequency = frequency + 1;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ทำงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
else if ((rpm_fix-rpm <= 3000 )&&(rpm_fix-rpm > 2500 ))
{
    frequency = frequency + 1.5;
}
else if ((rpm_fix-rpm <= 2500 )&&(rpm_fix-rpm > 2000 ))
{
    frequency = frequency + 1.25;
}
else if ((rpm_fix-rpm <= 1500 )&&(rpm_fix-rpm > 1000 ))
{
    frequency = frequency + 1;
}
else if ((rpm_fix-rpm <= 1000 )&&(rpm_fix-rpm > 500 ))
{
    frequency = frequency + 0.75;
}
else if ((rpm_fix-rpm <= 500 )&&(rpm_fix-rpm > 250 ))
{
    frequency = frequency + 0.5;
}
else if ((rpm_fix-rpm <= 250 )&&(rpm_fix-rpm > 100 ))
{
    frequency = frequency + 0.25;
}
else if ((rpm_fix-rpm <= 100 )&&(rpm_fix-rpm > 0 ))
{
    frequency = frequency + 0.1;
}
}

//Condition for v/f
if ((frequency > 0)&&(frequency <=10))
{
    frequency_range = 0;
}
else if ((frequency > 10 )&&(frequency <=20))
{
    frequency_range = 1;
}
else if ((frequency > 20)&&(frequency <= 30))
{
    frequency_range = 2;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        frequency_range = 3;
    }
else if (frequency >= 40)
    {
        frequency_range = 4;
    }

    openpwm();
        delay_motor(500);
        buf1[0] = ((rpm%10000)/1000) | 0x30;
        buf1[1] = (((rpm%10000)%1000)/100) | 0x30;
        buf1[2] = (((rpm%10000)%1000)%100)/10) | 0x30;
        buf1[3] = (((rpm%10000)%1000)%100)%10) | 0x30;
        lcd_puts(0xC3,buf1);
    }

if ((rpm >= rpm_fix) || (frequency >= 50))
    {
        buf1[0] = ((rpm%10000)/1000) | 0x30;
        buf1[1] = (((rpm%10000)%1000)/100) | 0x30;
        buf1[2] = (((rpm%10000)%1000)%100)/10) | 0x30;
        buf1[3] = (((rpm%10000)%1000)%100)%10) | 0x30;
        lcd_puts(0xC3,buf1);

        d = frequency/1;
        b = (frequency-(frequency/1))*100;

        buf2[0] = (d/10) | 0x30;
        buf2[1] = ((d%10)/1) | 0x30;
        lcd_puts(0xC9,buf2);
        lcd_puts(0xC7,"F:");
        lcd_puts(0xCB,"");
        buf3[0] = (b/10) | 0x30;
        buf3[1] = ((b%10)/1) | 0x30;
        lcd_puts(0xCC,buf3);
        lcd_puts(0xCE,"Hz");

        while(1)
        {
            if(button3 == 0)
            {
                delay_motor(100);
                if(button3 == 0)
                {
                    closepwm();

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CloseTimer2();
phase=0;

rpm= 0;

value=0;
rpm_fix=0;
x = 0;
y = 0;

frequency_range=0;

lcd_clear();

lcd_puts(0x80,"SpeedSet: rpm");
lcd_puts(0xC0,"Sp: F: Hz");

break;
}
}
break;
}
}
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์

ค.1 เอกสารคู่มือการใช้งาน GT60M303

ไอจีบีที GT60M303 เป็นอุปกรณ์ที่ทำหน้าที่เป็นสวิตช์เปิด-ปิดการทำงานของวงจรสวิตซ์ซึ่งมีรายละเอียดต่างๆดังนี้

TOSHIBA

GT60M303

TOSHIBA INSULATED GATE BIPOLAR TRANSISTOR SILICON N CHANNEL MOS TYPE

GT60M303

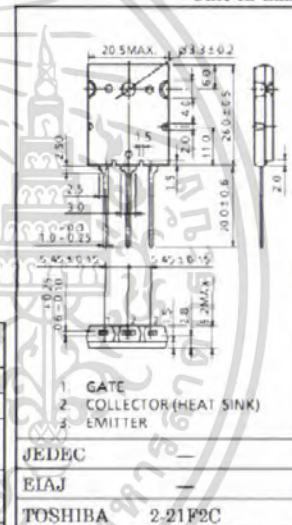
HIGH POWER SWITCHING APPLICATIONS

Unit in mm

- The 4th Generation
- FRD Included Between Emitter and Collector
- Enhancement-Mode
- High Speed IGBT : $t_{tr} = 0.25 \mu s$ (TYP.)
FRD : $t_{rr} = 0.7 \mu s$ (TYP.)
- Low Saturation Voltage : $V_{CE(sat)} = 2.1V$ (TYP.)

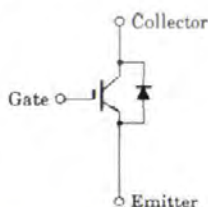
MAXIMUM RATINGS ($T_a = 25^\circ C$)

CHARACTERISTIC	SYMBOL	RATING	UNIT
Collector-Emitter Voltage	V_{CES}	900	V
Gate-Emitter Voltage	V_{GES}	± 25	V
Collector Current	DC	I_C	60
	1ms	I_{CP}	120
Emitter-Collector Forward Current	DC	I_{ECP}	15
	1ms	I_{ECP}	120
Collector Power Dissipation ($T_c = 25^\circ C$)	P_C	170	W
Junction Temperature	T_j	150	$^\circ C$
Storage Temperature Range	T_{stg}	$-55 \sim 150$	$^\circ C$
Screw Torque	τ	0.8	N·m



Weight : 9.75g

EQUIVALENT CIRCUIT



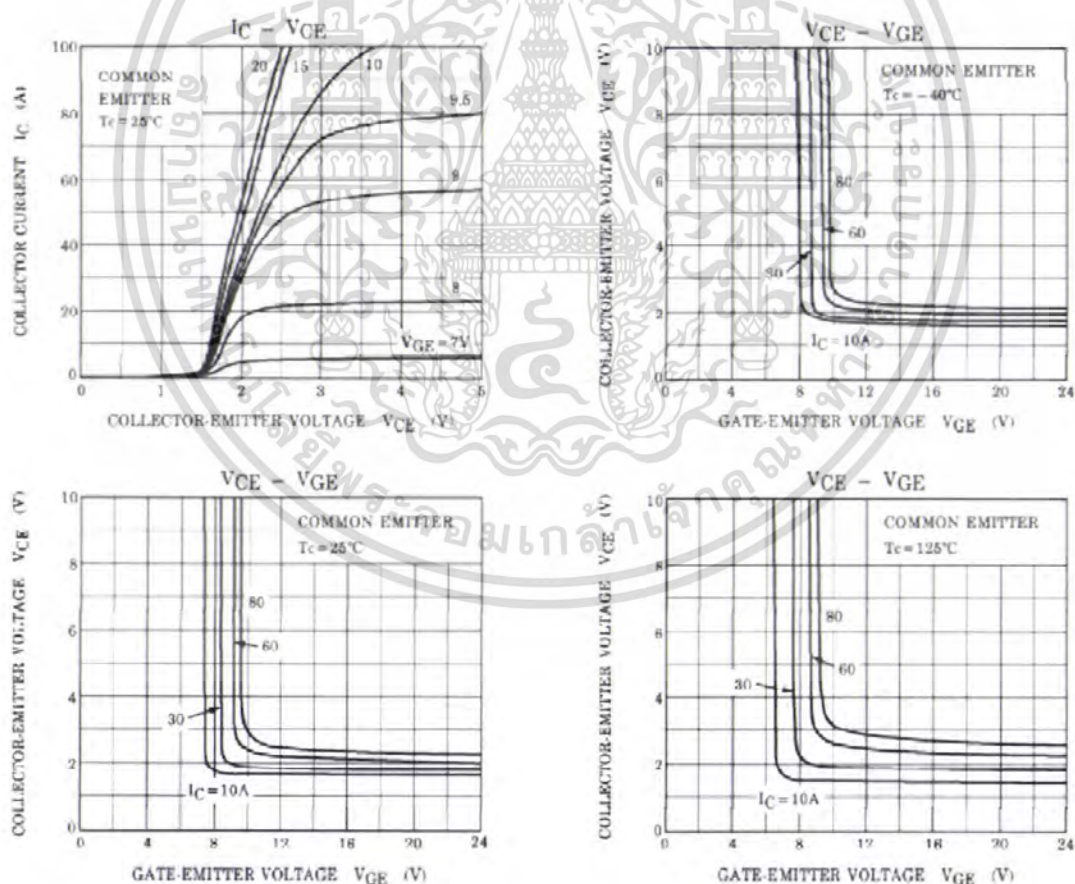
961001EAA1

- TOSHIBA is continually working to improve the quality and the reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to observe standards of safety, and to avoid situations in which a malfunction or failure of a TOSHIBA product could cause loss of human life, bodily injury or damage to property. In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent products specifications. Also, please keep in mind the precautions and conditions set forth in the TOSHIBA Semiconductor Reliability Handbook.
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA CORPORATION for any infringements of intellectual property or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any intellectual property or other rights of TOSHIBA CORPORATION or others.
- The information contained herein is subject to change without notice.

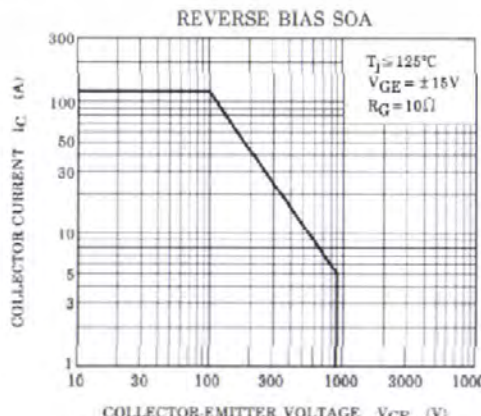
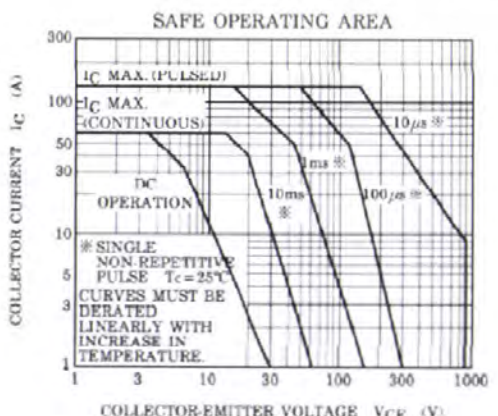
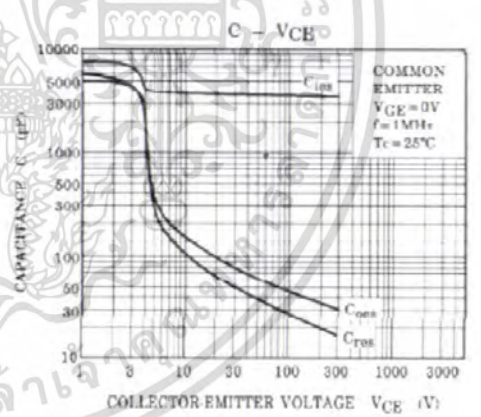
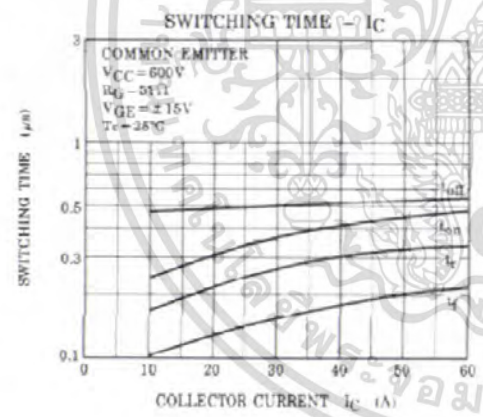
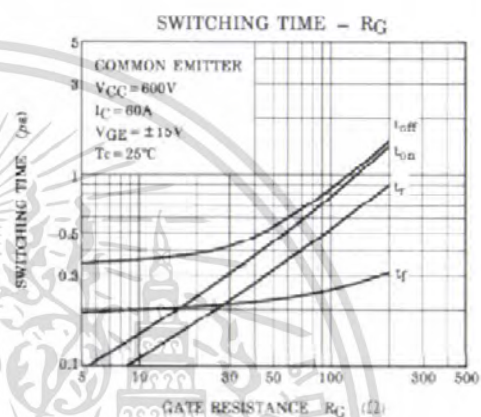
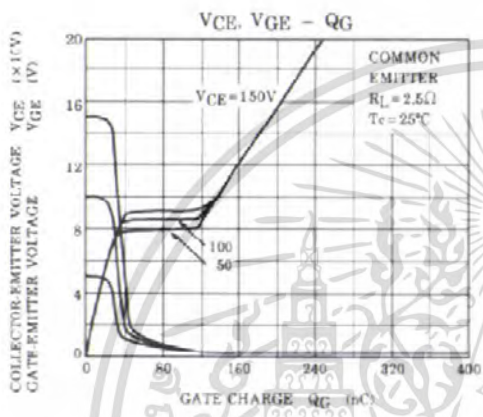
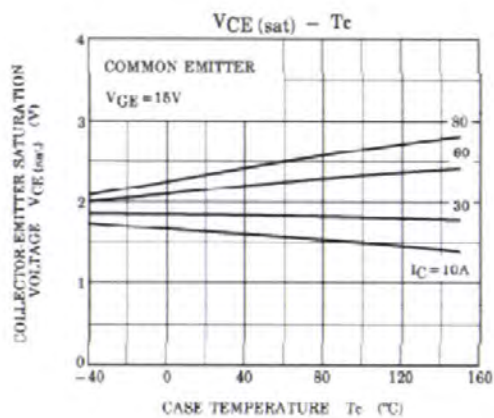
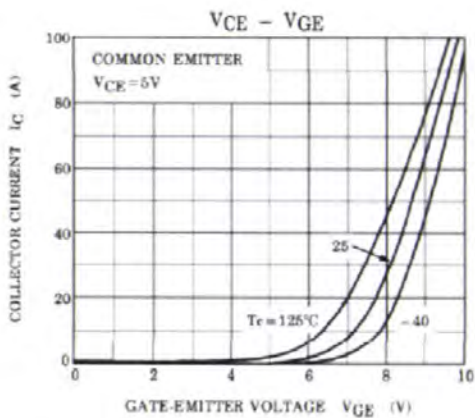
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการค้าเท่านั้น ไม่อนุญาตให้ผู้อื่นไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS ($T_a = 25^\circ\text{C}$)

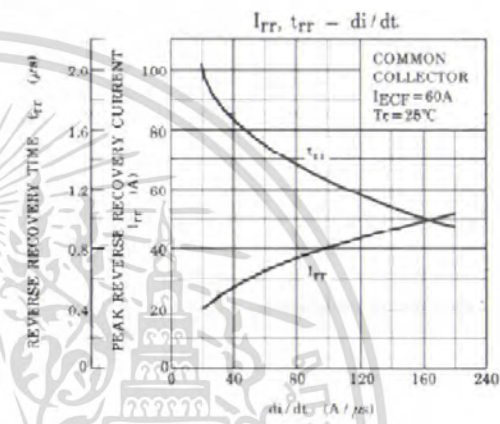
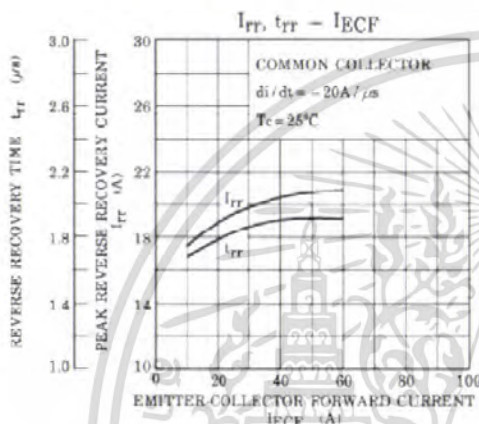
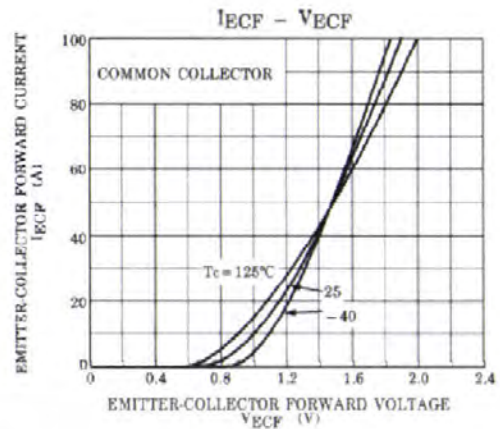
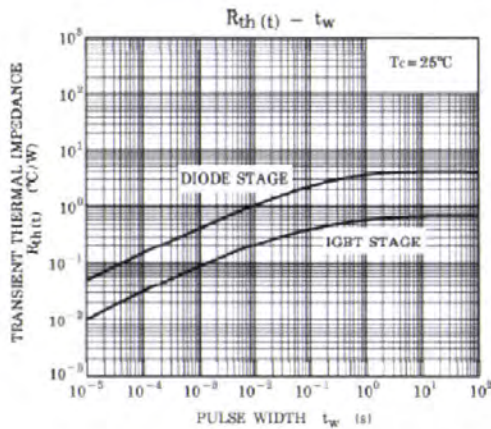
CHARACTERISTIC	SYMBOL	TEST CONDITION	MIN.	TYP	MAX.	UNIT
Gate Leakage Current	I_{GES}	$V_{GE} = \pm 25\text{V}, V_{CE} = 0$	—	—	± 500	nA
Collector Cut-off Current	I_{CES}	$V_{CE} = 900\text{V}, V_{GE} = 0$	—	—	1.0	mA
Gate-Emitter Cut-off Voltage	$V_{GE(OFF)}$	$I_C = 60\text{mA}, V_{CE} = 5\text{V}$	3.0	—	6.0	V
Collector-Emitter Saturation Voltage	$V_{CE(sat)(1)}$	$I_C = 10\text{A}, V_{GE} = 15\text{V}$	—	1.6	2.2	V
Collector-Emitter Saturation Voltage	$V_{CE(sat)(2)}$	$I_C = 60\text{A}, V_{GE} = 15\text{V}$	—	2.1	2.7	V
Input Capacitance	C_{ies}	$V_{CE} = 10\text{V}, V_{GE} = 0, f = 1\text{MHz}$	—	3800	—	pF
Switching Time	Rise Time	t_r	—	0.35	0.60	μs
	Turn-on Time	t_{on}	—	0.46	0.75	
	Fall Time	t_f	—	0.25	0.40	
	Turn-off Time	t_{off}	—	0.60	0.70	
Emitter-Collector Forward Voltage	V_{ECF}	$I_{EC} = 15\text{A}, V_{GE} = 0$	—	1.5	2.0	V
Reverse Recovery Time	t_{rr}	$I_F = 15\text{A}, V_{CE} = 0$ $di/dt = -20\text{A}/\mu\text{s}$	—	0.7	2.5	μs
Thermal Resistance	$R_{th(j-c)}$	IGBT	—	—	0.74	$^\circ\text{C}/\text{W}$
Thermal Resistance	$R_{th(j-e)}$	Diode	—	—	4.0	$^\circ\text{C}/\text{W}$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก.2 เอกสารคู่มือการใช้งาน IR2111

ไอซี IR2111 เป็นอุปกรณ์ที่ใช้ควบคุม ไอจีบีที ด้าน ไฮไซด์ และ โลว์ไซด์ ในกึ่งตัวนำกันไม่ให้ทำงานพร้อมกัน มีรายละเอียดต่างๆดังนี้

International
IR Rectifier

Data Sheet No. PD60028-M

IR2111(S) & (PbF)

HALF-BRIDGE DRIVER

Features

- Floating channel designed for bootstrap operation Fully operational to +600V Tolerant to negative transient voltage dV/dt immune
- Gate drive supply range from 10 to 20V
- Undervoltage lockout for both channels
- CMOS Schmitt-triggered inputs with pull-down
- Matched propagation delay for both channels
- Internally set deadtime
- High side output in phase with input
- Also available LEAD-FREE

Product Summary

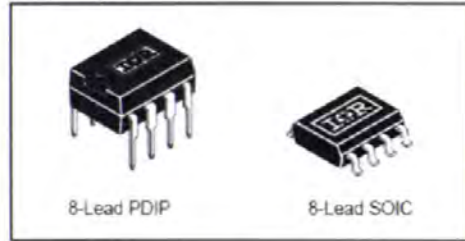
V _{OFFSET}	600V max.
I _{O+/-}	200 mA / 420 mA
V _{OUT}	10 - 20V
ton/off (typ.)	750 & 150 ns
Deadtime (typ.)	650 ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

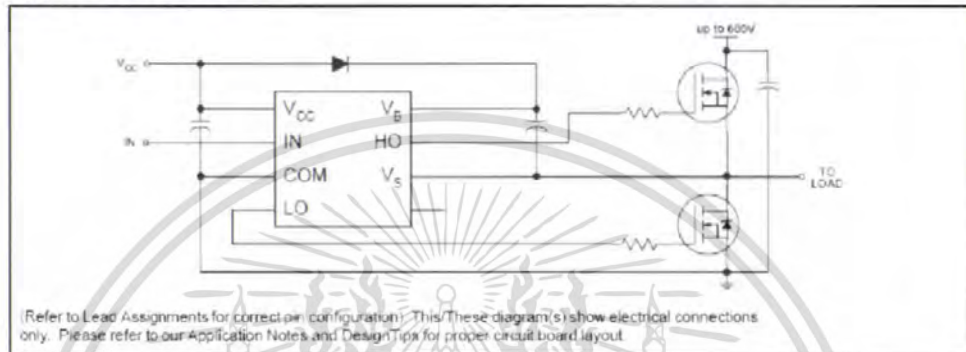
Description

The IR2111(S) is a high voltage, high speed power MOSFET and IGBT driver with dependent high and low side referenced output channels designed for half-bridge applications. Proprietary HVIC and latch immune CMOS technologies enable ruggedized monolithic construction. Logic input is compatible with standard CMOS outputs. The output drivers feature a high pulse current buffer stage designed for minimum driver cross-conduction. Internal deadtime is provided to avoid shoot-through in the output half-bridge. The floating channel can be used to drive an N-channel power MOSFET or IGBT in the high side configuration which operates up to 600 volts.

Packages



Typical Connection



Absolute Maximum Ratings

Absolute maximum ratings indicate sustained limits beyond which damage to the device may occur. All voltage parameters are absolute voltages referenced to COM. The thermal resistance and power dissipation ratings are measured under board mounted and still air conditions. Additional information is shown in figures 7 through 10.

Symbol	Definition	Min.	Max.	Units
V_B	High side floating supply voltage	-0.3	625	V
V_S	High side floating supply offset voltage	$V_B - 25$	$V_B + 0.3$	
V_{HO}	High side floating output voltage	$V_S - 0.3$	$V_B + 0.3$	
V_{CC}	Low side and logic fixed supply voltage	-0.3	25	
V_{LO}	Low side output voltage	-0.3	$V_{CC} + 0.3$	
V_{IN}	Logic input voltage	-0.3	$V_{CC} + 0.3$	
dV_S/dt	Allowable offset supply voltage transient (figure 2)	—	50	V/ns
P_D	Package power dissipation @ $T_A \leq +25^\circ\text{C}$	—	1.0	W
	(8 Lead PDIP)	—	0.625	
R_{thJA}	Thermal resistance, junction to ambient	—	125	$^\circ\text{C/W}$
	(8 lead SOIC)	—	200	
T_J	Junction temperature	—	150	$^\circ\text{C}$
T_S	Storage temperature	-55	150	
T_L	Lead temperature (soldering, 10 seconds)	—	300	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Recommended Operating Conditions

The input/output logic timing diagram is shown in figure 1. For proper operation the device should be used within the recommended conditions. The V_S offset rating is tested with all supplies biased at 15V differential.

Symbol	Definition	Min.	Max.	Units
V_B	High side floating supply absolute voltage	$V_S + 10$	$V_S + 20$	V
V_S	High side floating supply offset voltage	Note 1	600	
V_{HO}	High side floating output voltage	V_S	V_B	
V_{CC}	Low side and logic fixed supply voltage	10	20	
V_{LO}	Low side output voltage	0	V_{CC}	
V_{IN}	Logic input voltage	0	V_{CC}	
T_A	Ambient temperature	-40	125	°C

Note 1: Logic operational for V_S of -5 to +600V. Logic state held for V_S of -5V to $-V_B$. (Please refer to the Design Tip DT97-3 for more details).

Dynamic Electrical Characteristics

V_{BIAS} (V_{CC} , V_{BS}) = 15V, C_L = 1000 pF and T_A = 25°C unless otherwise specified. The dynamic electrical characteristics are measured using the test circuit shown in figure 3.

Symbol	Definition	Min.	Typ.	Max.	Units	Test Conditions
t_{on}	Turn-on propagation delay	550	750	950	ns	$V_S = 0V$
t_{off}	Turn-off propagation delay	—	150	180		$V_S = 600V$
t_r	Turn-on rise time	—	80	130		
t_f	Turn-off fall time	—	40	65		
DT	Deadtime, LS turn-off to HS turn-on & HS turn-off to LS turn-on	480	650	820		
MT	Delay matching, HS & LS turn-on/off	—	30	—		

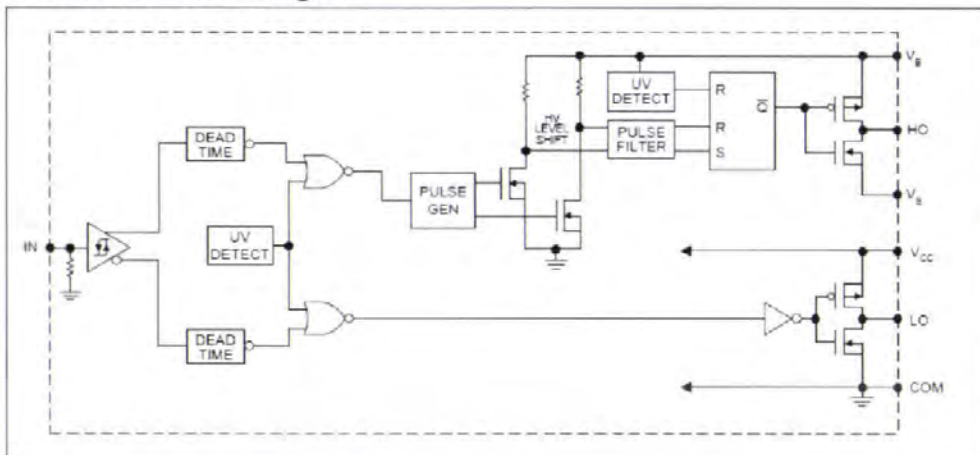
Static Electrical Characteristics

V_{BIAS} (V_{CC} , V_{BS}) = 15V and T_A = 25°C unless otherwise specified. The V_{IH} , V_{TH} and I_{IN} parameters are referenced to COM. The V_O and I_O parameters are referenced to COM and are applicable to the respective output leads, HO or LO.

Symbol	Definition	Min.	Typ.	Max.	Units	Test Conditions
V_{IH}	Logic "1" input voltage for HO & logic "0" for LO	6.4	—	—	V	$V_{CC} = 10V$
		9.5	—	—		$V_{CC} = 15V$
		12.6	—	—		$V_{CC} = 20V$
V_{IL}	Logic "0" input voltage for HO & logic "1" for LO	—	—	3.8	V	$V_{CC} = 10V$
		—	—	6.0		$V_{CC} = 15V$
		—	—	8.3		$V_{CC} = 20V$
V_{OH}	High level output voltage, $V_{BIAS} - V_O$	—	—	100	mV	$I_O = 0A$
V_{OL}	Low level output voltage, V_O	—	—	100	mV	$I_O = 0A$
I_{LK}	Offset supply leakage current	—	—	50	µA	$V_B = V_S = 600V$
I_{OBS}	Quiescent V_{BS} supply current	—	50	100		$V_{IN} = 0V$ or V_{CC}
I_{OCC}	Quiescent V_{CC} supply current	—	70	180		$V_{IN} = 0V$ or V_{CC}
I_{IN+}	Logic "1" input bias current	—	30	50		$V_{IN} = V_{CC}$
I_{IN-}	Logic "0" input bias current	—	—	1.0		$V_{IN} = 0V$
V_{BSUV+}	V_{BS} supply undervoltage positive going threshold	7.6	8.6	9.6	V	
V_{BSUV-}	V_{BS} supply undervoltage negative going threshold	7.2	8.2	9.2		
V_{CCUV+}	V_{CC} supply undervoltage positive going threshold	7.6	8.6	9.6		
V_{CCUV-}	V_{CC} supply undervoltage negative going threshold	7.2	8.2	9.2		
I_{O+}	Output high short circuit pulsed current	200	250	—	mA	$V_O = 0V$, $V_{IN} = V_{CC}$ $PW \leq 10 \mu s$
I_{O-}	Output low short circuit pulsed current	420	500	—		$V_O = 15V$, $V_{IN} = 0V$ $PW \leq 10 \mu s$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Block Diagram



Lead Definitions

Symbol	Description
IN	Logic input for high side and low side gate driver outputs (HO & LO), in phase with HO
V _B	High side floating supply
HO	High side gate drive output
V _S	High side floating supply return
V _{CC}	Low side and logic fixed supply
LO	Low side gate drive output
COM	Low side return

Lead Assignments

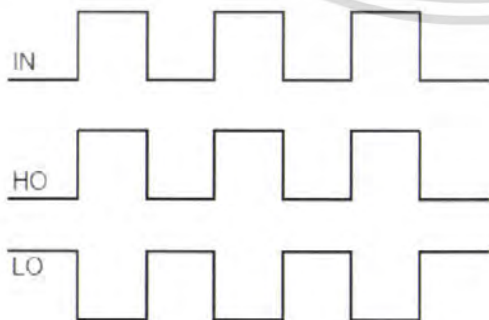
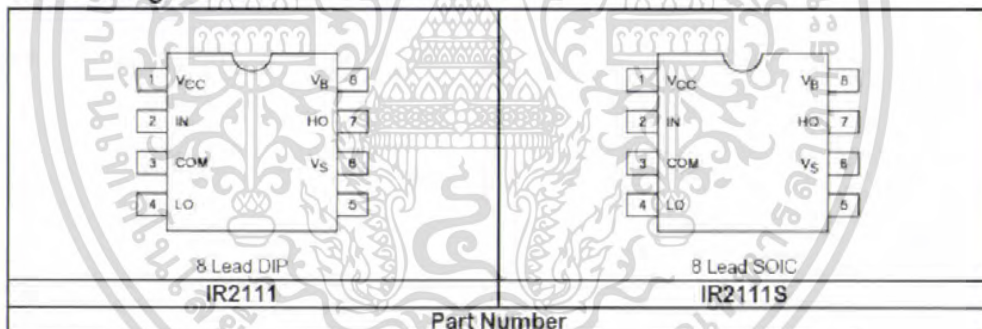


Figure 1. Input/Output Timing Diagram

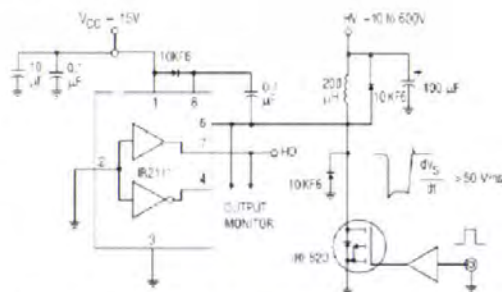


Figure 2. Floating Supply Voltage Transient Test Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

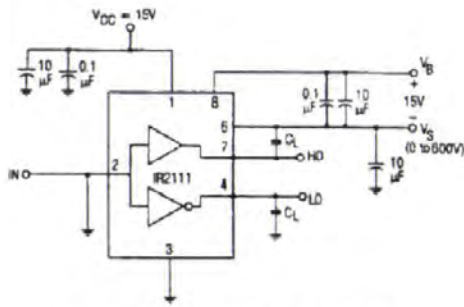


Figure 3. Switching Time Test Circuit

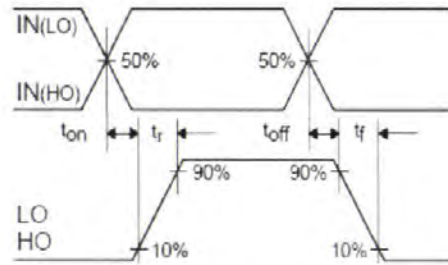


Figure 4. Switching Time Waveform Definition

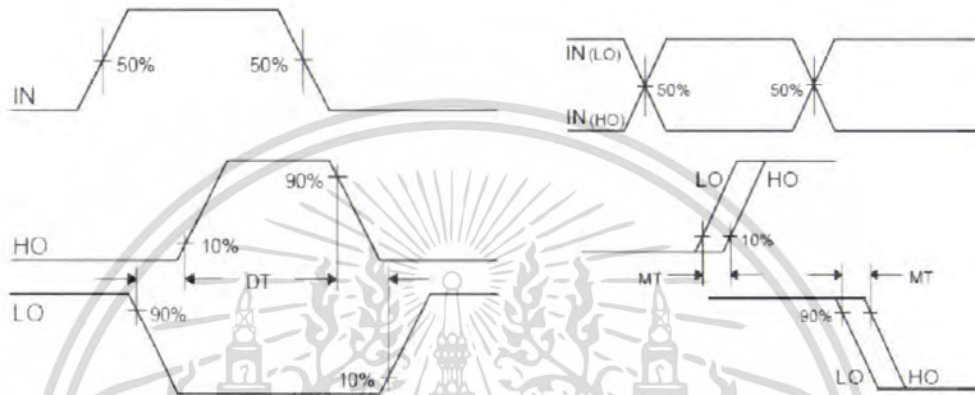


Figure 5. Deadtime Waveform Definitions

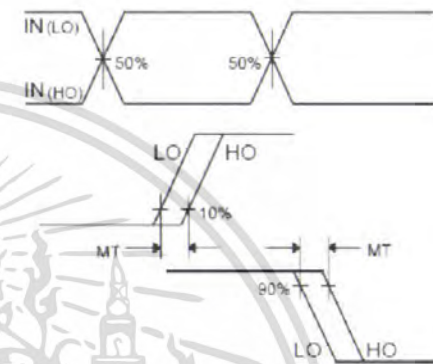


Figure 6. Delay Matching Waveform Definitions

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.3 เอกสารคู่มือการใช้งาน TLP250

ไอซี TLP250 ทำหน้าที่เป็นตัวขยายสัญญาณจาก 5 โวลต์ ที่รับมาจากไมโครคอนโทรลเลอร์ ขยายเป็น 15 โวลต์ เพื่อไปเข้าขา IN ของ IR2111 มีรายละเอียดต่างๆดังนี้

TOSHIBA

TLP250(INV)

TOSHIBA PHOTOCOUPLER GaAlAs IRED & PHOTO-IC

TLP250(INV)

TRANSISTOR INVERTER
 INVERTERS FOR AIR CONDITIONER
 IGBT GATE DRIVE
 POWER MOS FET GATE DRIVE

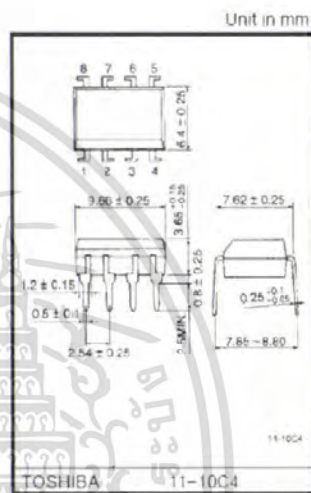
The TOSHIBA TLP250(INV) consists of a GaAlAs light emitting diode and a integrated photodetector.
 This unit is 8-lead DIP
 TLP250(INV) is suitable for gate driving circuit of IGBT or power MOS FET

- Input Threshold Current $I_F=5\text{mA(MAX)}$
- Supply Current(I_{CC}) 11mA(MAX)
- Supply Voltage(V_{CC}) 10-35V
- Output Current(I_O) $\pm 2.0\text{A(MAX)}$
- Switching Time(t_{LH}/t_{PH}) 0.5 $\mu\text{s(MAX)}$
- Isolation Voltage 2500Vrms
- UL Recognized UL1577-File No E67349
- Option(D4)

VDE Approved DIN VDE0884/06-92 Certificate No 76823
 Maximum Operating Insulation Voltage 630V_{PK}
 Highest Permissible Over Voltage 4000V_{PK}

(Note):When a VDE0884 approved type is needed,
 Please designate the "Option(D4)"

- Creepage Distance 6.4mm(MIN)
- Clearance 6.4mm(MIN)

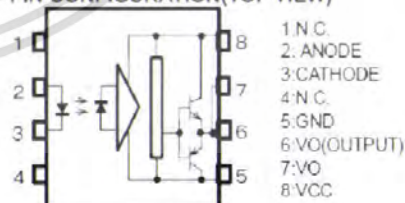


Weight 0.54 g

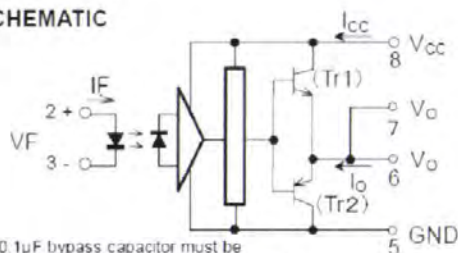
TRUTH TABLE

		Tr 1	Tr 2
INPUT LED	ON	ON	OFF
	OFF	OFF	ON

PIN CONFIGURATION(TOP VIEW)



SCHEMATIC



A 0.1 μF bypass capacitor must be connected between pin 8 and 5(See Note 5).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAXIMUM RATINGS (Ta=25°C)

CHARACTERISTIC		SYMBOL	RATING	UNIT	
LED	Forward Current	I_F	20	mA	
	Forward Current Derating (Ta≥70°C)	$\Delta I_F / \Delta T_a$	-0.36	mA/°C	
	Peak Transient Forward Current (Note 1)	I_{FPT}	1	A	
	Reverse Voltage	V_R	5	V	
	Junction Temperature	T_J	125	°C	
DETECTOR	"H" Peak Output Current	(Note 2)	PW ≤ 2.5μs, fs 15 kHz	-1.5	A
			PW ≤ 1.0μs, fs 15 kHz	-2.0	
	"L" Peak Output Current	(Note 2)	PW ≤ 2.5μs, fs 15 kHz	+1.5	A
			PW ≤ 1.0μs, fs 15 kHz	+2.0	
	Output Voltage	(Ta ≤ 70°C)	V_O	35	V
		(Ta = 85°C)		24	
	Supply Voltage	(Ta ≤ 70°C)	V_{CC}	35	V
		(Ta = 85°C)		24	
	Output Voltage Derating (Ta ≥ 70°C)		$\Delta V_O / \Delta T_a$	-0.73	V/°C
	Supply Voltage Derating (Ta ≥ 70°C)		$\Delta V_{CC} / \Delta T_a$	-0.73	V/°C
	Junction Temperature		T_J	125	°C
	Operating Frequency (Note 3)		f	25	kHz
Operating Temperature Range		T_{OP}	-20-85	°C	
Storage Temperature Range		T_{STG}	-55-125	°C	
Lead Soldering Temperature(10s)		T_{SOL}	260	°C	
Isolation Voltage (AC 1min., R.H. ≤ 60%, Ta=25°C)	(Note 4)	BV_S	2500	Vrms	

(Note 1) Pulse width PW ≤ 1μs, 300pps

(Note 2) Exponential Waveform

(Note 3) Exponential Waveform $I_{OH} \leq 1.0A$ (≤ 2.5μs), $I_{OL} \leq 1.0A$ (≤ 2.5μs)

(Note 4) Device considered a two terminal device, pins 1, 2, 3 and 4 shorted together and pins 5, 6, 7 and 8 shorted together.

(Note 5) A ceramic capacitor(0.1μF) should be connected from pin 8 to pin 5 to stabilize the operation of the high gain linear amplifier. Failure to provide the bypassing may impair the switching property. The total lead length between capacitor and coupler should not exceed 1cm.

RECOMMENDED OPERATING CONDITIONS

CHARACTERISTIC	SYMBOL	MIN	TYP	MAX	UNIT	
Input Current, ON	$I_{F(ON)}$	7	8	10	mA	
Input Voltage, OFF	$V_{F(OFF)}$	0	—	0.8	V	
Supply Voltage	V_{CC}	15	—	30	20	V
Peak Output Current	I_{OPH} / I_{OPL}	—	—	±0.5	A	
Operating Temperature	T_{OP}	-20	25	70	85	°C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS (Ta = -20~70°C, Unless otherwise specified)

CHARACTERISTIC		SYMBOL	TEST CIRCUIT	TEST CONDITION	MIN	TYP	MAX	UNIT	
Input Forward Voltage		V_f	—	$I_f = 10 \text{ mA}$, $T_a = 25^\circ\text{C}$	—	1.6	1.8	V	
Temperature Coefficient of Forward Voltage		$\Delta V_f / \Delta T_a$	—	$I_f = 10 \text{ mA}$	—	-2.0	—	mV / °C	
Input Reverse Current		I_R	—	$V_R = 5 \text{ V}$, $T_a = 25^\circ\text{C}$	—	—	10	μA	
Input Capacitance		C_T	—	$V = 0$, $f = 1 \text{ MHz}$, $T_a = 25^\circ\text{C}$	—	45	250	pF	
Output Current	"H" Level	I_{OH}	2	$V_{CC} = 30 \text{ V}$ (*1)	$I_f = 10 \text{ mA}$ $V_{b-e} = 4 \text{ V}$	-1.0	-1.5	—	A
	"L" Level	I_{OL}	1			$I_f = 0$ $V_{e-s} = 2.5 \text{ V}$	1.0	2	
Output Voltage	"H" Level	V_{OH}	3	$V_{CC1} = +15 \text{ V}$ $V_{EE1} = -15 \text{ V}$ $R_L = 200\Omega$, $I_f = 5 \text{ mA}$	11	12.8	—	V	
	"L" Level	V_{OL}	4	$V_{CC1} = +15 \text{ V}$ $V_{EE1} = -15 \text{ V}$ $R_L = 200\Omega$, $V_f = 0.8 \text{ V}$	—	-14.2	-12.5		
Supply Current	"H" Level	I_{cCH}	—	$V_{CC} = 30 \text{ V}$	$I_f = 10 \text{ mA}$ $T_a = 25^\circ\text{C}$	—	7	—	mA
					$I_f = 10 \text{ mA}$	—	—	11	
	"L" Level	I_{cCL}			$I_f = 0 \text{ mA}$ $T_a = 25^\circ\text{C}$	—	7.5	—	mA
					$I_f = 0 \text{ mA}$	—	—	11	
Threshold Input Current	L → H	I_{IH}	—	$V_{CC1} = +15 \text{ V}$ $V_{EE1} = -15 \text{ V}$ $R_L = 200\Omega$, $V_i > 0 \text{ V}$	—	1.2	5	mA	
Threshold Input Voltage	H → L	V_{IH}	—	$V_{CC1} = +15 \text{ V}$ $V_{EE1} = -15 \text{ V}$ $R_L = 200\Omega$, $V_i < 0 \text{ V}$	0.8	—	—	V	
Supply Voltage		V_{CC}	—	—	10	—	35	V	
Capacitance (Input-Output)		C_S	—	$V_S = 0$, $f = 1 \text{ MHz}$, $T_a = 25^\circ\text{C}$	—	1.0	2.0	pF	
Resistance (Input-Output)		R_S	—	$V_S = 500 \text{ V}$, $T_a = 25^\circ\text{C}$ $R_H \leq 60\%$	1×10^{10}	10^{12}	—	Ω	

(*): All typical values are at $T_a = 25^\circ\text{C}$

(*1): Duration of IO time $\leq 50 \mu\text{s}$

SWITCHING CHARACTERISTICS (Ta = -20~70°C, Unless otherwise specified)

CHARACTERISTIC		SYMBOL	TEST CIRCUIT	TEST CONDITION	MIN	TYP	MAX	UNIT
Propagation Delay Time	L → H	t_{PLH}	5	$I_f = 8 \text{ mA}$ $V_{CC} = 15 \text{ V}$ $R_L = 20\Omega$, $C_L = 10 \text{ nF}$	0.05	0.15	0.5	μs
	H → L	t_{PLH}			0.05	0.15	0.5	
Switching Time Dispersion between ON and OFF		$ t_{pHL} - t_{pLH} $	—	—	—	—	0.45	
Output Rise Time		t_r	—	—	—	—	—	
Output Fall Time		t_f	—	—	—	—	—	
Common Mode Transient Immunity at High Level Output		CM_H	6	$V_{CM} = 1000 \text{ V}$, $I_f = 8 \text{ mA}$ $V_{CC} = 30 \text{ V}$, $T_a = 25^\circ\text{C}$	-15000	—	—	V / μs
Common Mode Transient Immunity at Low Level Output		CM_L		$V_{CM} = 1000 \text{ V}$, $I_f = 0 \text{ mA}$ $V_{CC} = 30 \text{ V}$, $T_a = 25^\circ\text{C}$	15000	—	—	V / μs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Fig.1 I_{OPL} TEST CIRCUIT

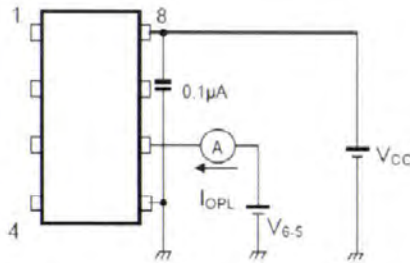


Fig.2 I_{OPH} TEST CIRCUIT

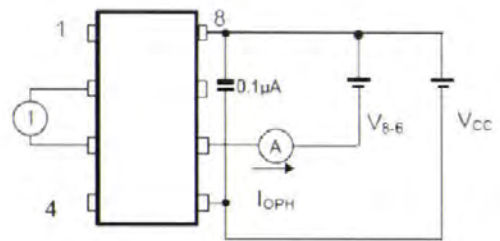


Fig.3 V_{OH} TEST CIRCUIT

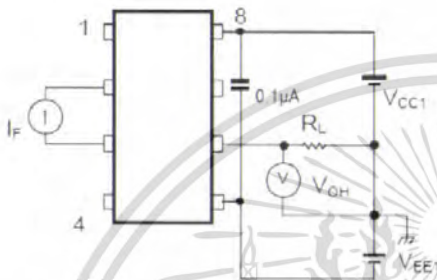


Fig.4 V_{OL} TEST CIRCUIT

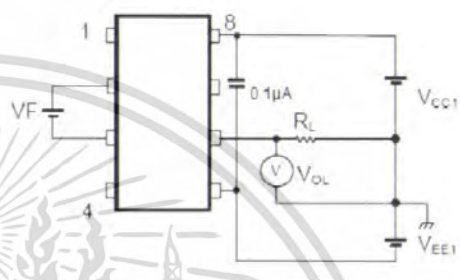


Fig.5 t_{pLH} , t_{pHL} , t_r , t_f TEST CIRCUIT

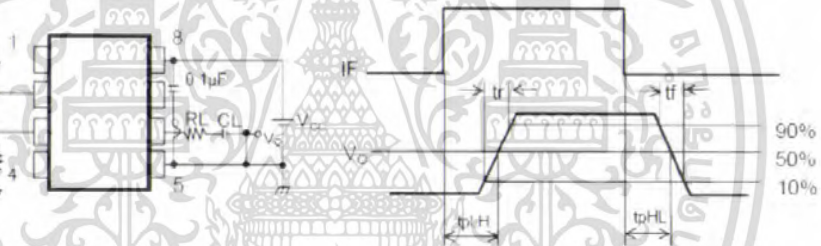
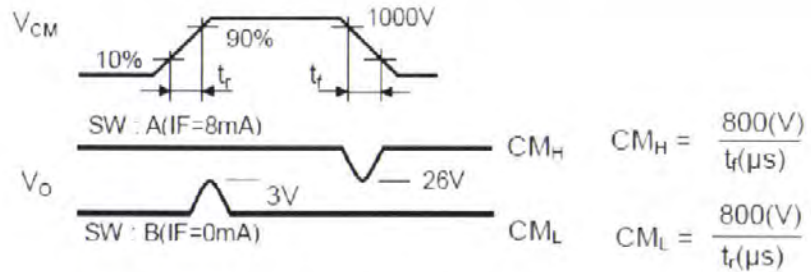
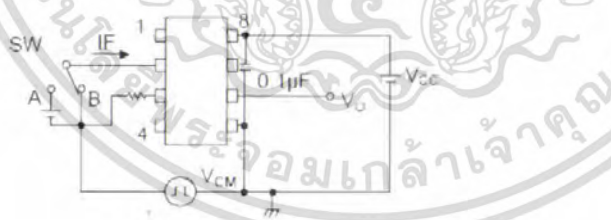


Fig.6 CM_H , CM_L TEST CIRCUIT



CML(CMH) is the maximum rate of rise(fall) of the common mode voltage that can be sustained with the output voltage in the low(high)state.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้