

**สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง**

**ไมโครสเตปมอเตอร์**

**MICROSTEP MOTOR**



โดย  
นางสาวพัศตรพิมล สุรวาทศิลป์  
นายพีรศักดิ์ วงศ์วิวัฒน์กิจ  
นายวิษณุ โรจนภิญญา

261.  
ท151921  
2550

เลขหาญ.....  
เลขทะเบียน..... **83067**  
วันเดือนปี...-5...ค.ศ. 2551

b. 11964223  
i. ....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


# ปริญญาโทบริหารศึกษาศาสตร์ 2550


ภาควิชาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ไมโครสเตปมอเตอร์  
MICROSTEP MOTOR

ผู้จัดทำ นางสาวพัชกรพิมล สุรวาทศิลป์ 47010498  
นายพีรศักดิ์ วงศ์วิวัฒน์กิจ 47010534  
นายวิษณุ โจรนภิญญ 47010723

  
.....อาจารย์ที่ปรึกษา  
(รศ.ดร.วันชัย รีรุธา)

  
.....อาจารย์ที่ปรึกษา  
(อ.ธวัชชัย คำศรี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ไมโครสเตปมอเตอร์

โดย

นางสาวพัทธกรพิมล สุรวาทศิลป์ 47010498  
นายพีรศักดิ์ วงศ์วิวัฒน์กิจ 47010534  
นายวิษณุ โรจนภิญญา 47010732

อาจารย์ที่ปรึกษา

รศ.ดร.วันชัย รุ่งรุจา

อ.ธวัชชัย คำศรี

ปีการศึกษา 2550

## บทคัดย่อ

สเตปมอเตอร์โดยทั่วไปจะมีขนาดสเตปเล็กที่สุดประมาณ 0.5 องศาต่อสเตป ซึ่งจะใหญ่เกินไปที่จะนำไปใช้ในบางประเภท เช่น ระบบแขนกล ระบบเครื่องพิมพ์และระบบพล็อตเตอร์ เป็นต้น เนื่องจากความละเอียดของจำนวนสเตปต่อรอบส่งผลโดยตรงกับความราบเรียบในการหมุนของสเตปมอเตอร์ ปรินูญานิพนธ์ฉบับนี้ได้ทำการควบคุมการขับสเตปมอเตอร์แบบไมโครสเตปโดยใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมการจ่ายกระแสไฟฟ้าของวงจรขับร่วมกับส่วนควบคุมแรงดัน เพื่อควบคุมหรือแบ่งระดับแรงดันไฟฟ้าให้มีระดับต่างกันหลายระดับเพื่อนำสัญญาณที่ได้ นั้นไปกระตุ้นเฟสของสเตปมอเตอร์ให้เกิดการหมุนเป็นสเตป ซึ่งการควบคุมการขับสเตปมอเตอร์แบบไมโครสเตปนั้นจะเป็นการเพิ่มความละเอียดของจำนวนสเตปต่อรอบของสเตปมอเตอร์ ทำให้การหมุนของสเตปมอเตอร์มีความราบเรียบมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# MICROSTEP MOTOR

By

Miss Pakpimol Surawatasilp

Mr. Pirasak Wongvivatkit

Mr. Wisanu Rojanapinyoo

Advisor

Assoc. Prof. Dr.Vanchai Riewruja

Mr.Thawatchai Kamsri

Academic Year 2007

## ABSTRACT

Generally, step motor has produced the smallest step about 0.5 degrees per step turn. So, it is too large for certain applications such as robot arms, printers or plotters and etc. Resolution of step per revolution is one of the important and effect to controlling smooth rotation. Therefore, to increasing the resolution of step motor, this thesis demonstrated the driving step motor in microstep excitation mode. Microcontroller is used to control current feeding to the driving circuit work with control voltage part to separate many different voltage levels. Then, transmit the signal voltage activates the step motor phase for step turns. The driving of microstep motor is increasing of resolution and number of step per revolution and smoother rotation.

## กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความช่วยเหลือเป็นอย่างดี จาก รศ.ดร.วันชัย ธีร์รุจา ที่ได้กรุณาให้คำแนะนำที่คิมา โดยตลอดและความช่วยเหลืออื่นๆ ที่เป็น ประโยชน์ต่อโครงการ

ขอขอบพระคุณ อาจารย์ธวัชชัย คำศรี ที่ได้กรุณาให้คำปรึกษาและคอยถามถึงความคืบหน้า อยู่ตลอดเวลา รวมทั้งสนับสนุนอุปกรณ์ที่ขาดเหลือ ผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณ อย่างสูง

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่คอยเป็นกำลังใจตลอดมา รวมถึงการสนับสนุนในเรื่องของงบประมาณ ตลอดจนเป็นแรงบันดาลใจที่ดีที่สุดที่ทำให้โครงการ นี้สำเร็จสมบูรณ์



### III

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VI
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมา	1
1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์	1
1.3 รายละเอียดของปริญญานิพนธ์	2
บทที่ 2 ทฤษฎีพื้นฐานและหลักการทำงานของสเตปมอเตอร์	3
2.1 บทนำ	3
2.2 โครงสร้างและหลักการทำงานของสเตปมอเตอร์	4
2.2.1 โครงสร้างและหลักการทำงานของสเตปมอเตอร์ชนิดต่างๆ	5
2.2.1.1 สเตปมอเตอร์แบบแม่เหล็กถาวร	5
2.2.1.2 สเตปมอเตอร์แบบแปรค่ารีลักแตนซ์	7
2.2.1.3 สเตปมอเตอร์แบบไฮบริด	8
2.2.2 การกระตุ้นเฟสของสเตปมอเตอร์	10
2.2.2.1 การกระตุ้นเฟสแบบพูลสเตปมอเตอร์	10
2.2.2.2 การกระตุ้นเฟสแบบฮาล์ฟสเตปมอเตอร์	11
2.3 การควบคุมการทำงานของสเตปมอเตอร์	11
2.4 วงจรขับสเตปมอเตอร์โดยทั่วไป	13
2.4.1 วงจรขับแบบโวลเตจสองระดับ	13
2.4.2 วงจรขับแบบชอปเปอร์	14
2.4.3 วงจรขับแบบบริดจ์	15
2.5 สรุป	17

## สารบัญ(ต่อ)

	หน้า
<b>บทที่ 3 หลักการออกแบบและการสร้าง</b>	18
3.1 บทนำ	18
3.2 ส่วนประกอบของระบบควบคุมการขับเคลื่อนมอเตอร์แบบไมโครสเตป	18
3.2.1 ไมโครคอนโทรลเลอร์	19
3.2.2 ส่วนควบคุมแรงดัน โดยใช้วิธีพัลส์วิดท์มอดูเลชัน	19
3.2.3 วงจรขับเคลื่อนมอเตอร์แบบไมโครสเตป	20
3.2.3.1 ไมโครคอนโทรลเลอร์	21
3.2.3.2 สวิตซ์ 4 ตัว	21
3.2.3.3 แลคเกอร์รีซิสเตอร์	21
3.2.3.4 ทรานซิสเตอร์	22
3.3 โปรแกรมที่ใช้ในการขับเคลื่อนมอเตอร์	23
3.4 สรุป	24
<b>บทที่ 4 ผลการทดลอง</b>	25
4.1 ผลการทดลอง	25
4.2 สรุปผลการทดลอง	31
<b>บทที่ 5 บทวิจารณ์และสรุป</b>	32
5.1 สรุปผลการดำเนินงาน	32
5.2 ปัญหาที่พบและแนวทางการแก้ไข	32
5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา	33
<b>ภาคผนวก ก แสดงรูปส่วนประกอบของวงจร</b>	34
<b>ภาคผนวก ข โปรแกรมที่ใช้ในการทดลอง</b>	37
<b>ภาคผนวก ค คู่มืออุปกรณ์อิเล็กทรอนิกส์</b>	64
ค.1 เอกสารคู่มือการใช้งาน PIC16F876A	64
ค.2 เอกสารคู่มือการใช้งาน TIP122	65
ค.3 เอกสารคู่มือการใช้งาน BC337	66
<b>เอกสารอ้างอิง</b>	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ

รูปที่	หน้า
2.1 แสดงบล็อกโคโอะแกรมของระบบควบคุมการขับเคลื่อนมอเตอร์	3
2.2 แสดงสนามแม่เหล็กที่เกิดขึ้นในลักษณะต่างๆ	3
2.3 แสดงแรงคึงคูดทำให้เกิดทอร์คที่หมุนโรเตอร์ให้ไปอยู่ในตำแหน่งสมดุล	4
2.4 แสดงโครงสร้างของสเตปมอเตอร์แบบแม่เหล็กถาวร	5
2.5 แสดงการทำงานของสเตปมอเตอร์แบบแม่เหล็กถาวรขนาดสี่เฟส	6
2.6 แสดงโครงสร้างของสเตปมอเตอร์แบบแปรค่ารีลักแตนซ์	7
2.7 แสดงการทำงานของสเตปมอเตอร์แบบแปรค่ารีลักแตนซ์ขนาดสามเฟส	7
2.8 แสดงตำแหน่งสมดุลของสเตปมอเตอร์เมื่อเฟสใดเฟสหนึ่งถูกกระตุ้น	8
2.9 แสดงการเปลี่ยนลำดับสเตปในการเคลื่อนที่ของสเตปมอเตอร์แบบแปรค่ารีลักแตนซ์	8
2.10 แสดงโครงสร้างของสเตปมอเตอร์แบบไฮบริด	9
2.11 แสดงระบบควบคุมการขับเคลื่อนมอเตอร์	12
2.12 แสดงวงจรขับเคลื่อนมอเตอร์แบบ โวลเตจสองระดับ	14
2.13 แสดงวงจรขับเคลื่อนมอเตอร์แบบชอปเปอร์	15
2.14 แสดงวงจรขับเคลื่อนมอเตอร์แบบบริดจ์สำหรับการขับเคลื่อนแบบไบโพลาร์	15
2.15 แสดงอีกรูปแบบหนึ่งของวงจรขับเคลื่อนแบบบริดจ์ซึ่งใช้แหล่งจ่ายไฟเลี้ยงเดียว	16
3.1 แสดงบล็อกโคโอะแกรมของระบบควบคุมการขับเคลื่อนมอเตอร์แบบไมโครสเตป	18
3.2 แสดงการปรับความกว้างของสัญญาณพัลส์และระดับแรงดันไฟฟ้าเฉลี่ย	20
3.3 แสดงวงจรขับเคลื่อนมอเตอร์แบบไมโครสเตป	20
3.4 แสดงไฟลิวชาร์ตการควบคุมการขับเคลื่อนมอเตอร์แบบไมโครสเตป	23
3.4 (ต่อ) แสดงไฟลิวชาร์ตการควบคุมการขับเคลื่อนมอเตอร์แบบไมโครสเตป	24
4.1 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณอินพุทของเฟส A+, A- 24 กับสัญญาณอินพุทของเฟส B+, B- โดยให้การกระตุ้นเฟสแบบฟูลสเตป 2 เฟส	25
4.2 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณอินพุทของเฟส A+, A- 26 กับสัญญาณเอาต์พุทของเฟส A+, A- โดยให้การกระตุ้นเฟสแบบฟูลสเตป 2 เฟส	26
4.3 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณเอาต์พุทของเฟส A+, A- 26 กับสัญญาณเอาต์พุทของเฟส B+, B- โดยให้การกระตุ้นเฟสแบบฟูลสเตป 2 เฟส	26
4.4 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณอินพุทของเฟส A+, A- 27 กับสัญญาณอินพุทของเฟส B+, B- โดยให้การกระตุ้นเฟสแบบฮาล์ฟสเตป	27

## สารบัญภาพ(ต่อ)

รูปที่	หน้า
4.5 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณอินพุทของเฟส A+, A- กับสัญญาณเอาต์พุทของเฟส A+, A- โดยใช้การกระตุ้นเฟสแบบฮาล์ฟสเตป	27
4.6 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณเอาต์พุทของเฟส A+, A- กับสัญญาณเอาต์พุทของเฟส B+,B-โดยใช้การกระตุ้นเฟสแบบฮาล์ฟสเตป	28
4.7 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณอินพุทของเฟส A+, A- กับสัญญาณอินพุทของเฟส B+, B- โดยจับในโหมด 1200 สเตป	28
4.8 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณอินพุทของเฟส A+, A- กับสัญญาณเอาต์พุทของเฟส A+, A- โดยจับในโหมด 1200 สเตป	29
4.9 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณเอาต์พุทของเฟส A+, A- กับสัญญาณเอาต์พุทของเฟส B+,B-โดยจับใน โหมด 1200 สเตป	29
4.10 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณอินพุทของเฟส A+, A- กับสัญญาณอินพุทของเฟส B+, B- โดยจับในโหมด 3600 สเตป	30
4.11 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณอินพุทของเฟส A+, A- กับสัญญาณเอาต์พุทของเฟส A+, A- โดยจับในโหมด 3600 สเตป	30
4.12 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณเอาต์พุทของเฟส A+, A- กับสัญญาณเอาต์พุทของเฟส B+,B-โดยจับใน โหมด 3600 สเตป	31
ก.1 แสดงวงจรชุดทดสอบการส่งสัญญาณพัลส์	34
ก.2 แสดงวงจรขับสเตปมอเตอร์	34
ก.3 แสดงระบบควบคุมและการทดสอบการขับสเตปมอเตอร์แบบไมโครสเตป	35
ก.4 แสดงสเกลวัดมุมสเตป	35
ก.5 แสดงวงจรขับสเตปมอเตอร์และวงจรชุดทดสอบการส่งสัญญาณพัลส์	36

# สารบัญตาราง

ตารางที่	หน้า
2.1 การกระตุ้นเฟสแบบฟูสเตป 1 เฟส	10
2.2 การกระตุ้นเฟสแบบฟูสเตป 2 เฟส	10
2.3 การกระตุ้นเฟสแบบฮาล์ฟสเตปมอเตอร์	11
3.1 แสดงสถานะของสวิทช์ในการเลือกโหมดการทำงานของสเตปมอเตอร์	21
3.2 แสดงระดับแรงดันไฟฟ้าที่จู่คร่วมของแอสเคเตอร์รีซิสเตอร์จากการทดลอง	22



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมา

สเตปมอเตอร์คืออุปกรณ์ที่มีลักษณะการเคลื่อนเป็นสเตป โดยอาศัยวิธีการทางแม่เหล็กไฟฟ้า ซึ่งเปลี่ยนสัญญาณพัลซ์อินพุตซึ่งเป็นสัญญาณดิจิทัลไปเป็นการเคลื่อนที่แบบอนาลอก สเตปมอเตอร์บางครั้งถูกเรียกว่าสเตปปีงมอเตอร์หรือสเตปเปอร์มอเตอร์ เมื่อสเตปมอเตอร์ถูกกระตุ้นโดยโวลเตจหรือกระแสไฟฟ้าซึ่งโดยมากจะเป็นไฟฟ้ากระแสตรงจะเกิดการเคลื่อนที่เชิงมุมหรือหมุนเป็นสเตป ซึ่งจำนวนสเตปเอาท์พุทของสเตปมอเตอร์จะมีจำนวนเท่ากับจำนวนของสัญญาณพัลซ์อินพุตที่ป้อนให้กับสเตปมอเตอร์ โดยเมื่อป้อนสัญญาณพัลซ์ 1 ลูก สเตปมอเตอร์จะมีการเคลื่อนที่ไป 1 สเตป

สมัยก่อนการใช้งานสเตปมอเตอร์มีประสิทธิภาพต่ำตู้เอซีมอเตอร์ (A.C Motor) และดีซีมอเตอร์ (D.C Motor) ไม่ได้ แต่เมื่อมีการนำดิจิทัลคอมพิวเตอร์มาใช้จึงได้เปลี่ยนรูปแบบการควบคุมสเตปมอเตอร์ใหม่ โดยใช้การควบคุมโดยไมโคร โปรเซสเซอร์หรือไมโครคอนโทรลเลอร์ ทำให้การใช้งานสเตปมอเตอร์สะดวกขึ้น ปัจจุบันสเตปมอเตอร์ได้ถูกนำมาใช้ในอุปกรณ์ที่ใช้งานร่วมกับคอมพิวเตอร์ เช่น พรินเตอร์ (Printer) ดิสก์ไดรฟ์ (Disk Drive) นอกจากนี้ยังถูกนำไปใช้งานในอุปกรณ์ที่ใช้การควบคุมเชิงตัวเลข (Numerical Control) การควบคุมกระบวนการ (Process Control) และการควบคุมอุปกรณ์ทางเครื่องมือกล (Machine Tool Control) เป็นต้น

การควบคุมการทำงานของสเตปมอเตอร์โดยทั่วไปจะใช้วงจรขับซึ่งให้สเตปมอเตอร์ตอบสนองต่อสัญญาณพัลซ์ซึ่งทำให้การเคลื่อนที่ของสเตปมอเตอร์เป็นจังหวะต่อเนื่องกัน ในที่นี้เป็นการศึกษาถึงวิธีการควบคุมการขับสเตปมอเตอร์แบบไมโครสเตป ซึ่งให้ความละเอียดของสเตปเชิงมุมสูงขึ้น เพื่อเป็นการเพิ่มความราบเรียบในการหมุนของสเตปมอเตอร์

### 1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์

- 1) เพื่อศึกษาวิธีการควบคุมการขับสเตปมอเตอร์แบบต่างๆ เพื่อนำสเตปมอเตอร์มาใช้งานได้อย่างมีประสิทธิภาพ
- 2) เพื่อศึกษาวิธีการขับสเตปมอเตอร์แบบไมโครสเตป ซึ่งให้ความละเอียดของสเตปเชิงมุมหรือจำนวนสเตปต่อรอบสูงขึ้น เพื่อเป็นการเพิ่มความราบเรียบในการหมุนของสเตปมอเตอร์
- 3) เพื่อสร้างวงจรขับสเตปมอเตอร์ที่มีประสิทธิภาพ สามารถนำไปประยุกต์ใช้ในงานด้านการควบคุมตำแหน่งได้

4) สามารถเขียน โปรแกรมติดต่อกับไมโครคอนโทรลเลอร์ เพื่อควบคุมการขับเคลื่อนมอเตอร์แบบไมโครสเตป

### 1.3 รายละเอียดของปฏิญานิพนธ์

เนื้อหาที่จะกล่าวในปฏิญานิพนธ์ฉบับนี้ประกอบด้วย

บทที่ 1 กล่าวถึงความเป็นมา วัตถุประสงค์ในการทำปฏิญานิพนธ์ และรายละเอียดของปฏิญานิพนธ์

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐาน โครงสร้าง หลักการทำงานของสเตปมอเตอร์ชนิดต่างๆ และการกระตุ้นเฟสสเตปมอเตอร์ นอกจากนี้ยังกล่าวถึงวงจรควบคุมและวงจรขับเคลื่อนมอเตอร์โดยทั่วไป

บทที่ 3 กล่าวถึงหลักการออกแบบและการสร้างระบบควบคุมการขับเคลื่อนมอเตอร์แบบไมโครสเตป

บทที่ 4 แสดงการทดลองและผลการทดลองการขับเคลื่อนมอเตอร์แบบไมโครสเตป

บทที่ 5 บทวิจารณ์และสรุป ปัญหาที่พบและแนวทางแก้ไข รวมทั้งข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา



## บทที่ 2

# ทฤษฎีพื้นฐานและหลักการทำงานของสเตปมอเตอร์

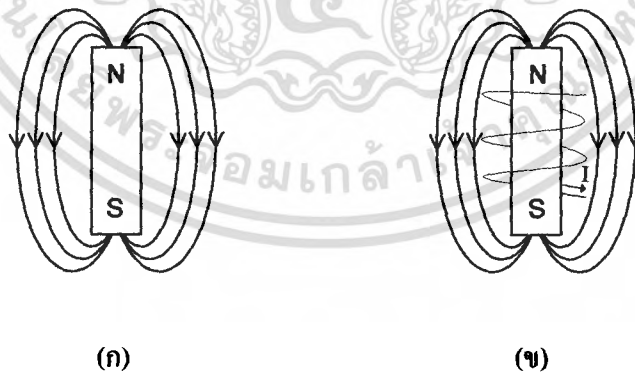
### 2.1 บทนำ

สเตปมอเตอร์แบ่งออกได้เป็นหลายชนิดด้วยกัน เช่น สเตปมอเตอร์แบบแม่เหล็กถาวร (Permanent Magnet) สเตปมอเตอร์แบบแปรค่ารีลัคแตนซ์ (Variable Reluctance) สเตปมอเตอร์แบบไฮบริด (Hybrid) และสเตปมอเตอร์แบบลิเนียร์ (Linear) เป็นต้น แต่การใช้งานส่วนใหญ่จะนิยมใช้สเตปมอเตอร์แบบแปรค่ารีลัคแตนซ์และสเตปมอเตอร์แบบไฮบริด

ระบบควบคุมการขับเคลื่อนสเตปมอเตอร์ ซึ่งแสดงในบล็อกไดอะแกรมรูปที่ 2.1 นั้น ประกอบด้วย ไมโครคอนโทรลเลอร์ วงจรขับ และสเตปมอเตอร์ โดยไมโครคอนโทรลเลอร์จะเป็นตัวควบคุมการจ่ายกระแสไฟฟ้าของวงจรขับซึ่งเป็นการส่งสัญญาณควบคุมเข้าสู่ส่วนของวงจรขับ เพื่อนำสัญญาณที่ได้นั้น ไปกระตุ้นเฟสของสเตปมอเตอร์แต่ละเฟสทำให้สเตปมอเตอร์เกิดการเคลื่อนที่เป็นสเตป โดยลักษณะการเหนี่ยวนำสนามแม่เหล็กแสดงได้ดังรูปที่ 2.2



รูปที่ 2.1 แสดงบล็อกไดอะแกรมของระบบควบคุมการขับเคลื่อนสเตปมอเตอร์



รูปที่ 2.2 แสดงสนามแม่เหล็กที่เกิดขึ้นในลักษณะต่างๆ

- ก) สนามแม่เหล็กที่เกิดจากแม่เหล็กถาวร
- ข) สนามแม่เหล็กของแม่เหล็กไฟฟ้าที่เกิดจากกระแส I

## 2.2 โครงสร้างและหลักการทำงานของสเตปมอเตอร์

จากรูปที่ 2.3 แม่เหล็กถาวรติดอยู่บนเพลาลงและสามารถหมุนได้อย่างอิสระในขณะที่ไม่มีขั้วไฟฟ้าที่สเตเตอร์ ถ้าก่อนมีขั้วไฟฟ้าตำแหน่งของโรเตอร์แม่เหล็กอยู่ตามแกน a-a' ซึ่งทำมุม  $\theta$  องศา กับแกนของขั้วแม่เหล็กไฟฟ้า ดังนั้นจะได้แรงแม่เหล็กที่เกิดจากการดึงดูดของขั้วแม่เหล็กที่ต่างกัน ทำให้เกิดส่วนของแรงบิดปกติ

$$F_n = F \cos \theta^\circ \text{ (แรงนี้ตั้งฉากกับแกน a-a')} \quad (2.1)$$

จะได้ทอร์ก (Torque) ผลรวม

$$T = F_n R \quad (2.2)$$

(ทำให้โรเตอร์หมุนไปในทิศทางตามเข็มนาฬิกา จนกว่าแกนของโรเตอร์ a-a' อยู่ในแนวเดียวกับสเตเตอร์)



รูปที่ 2.3 แสดงแรงดึงดูดทำให้เกิดทอร์กที่หมุนโรเตอร์ให้ไปอยู่ในตำแหน่งสมดุล

ถ้าหากมีขั้วแม่เหล็กไฟฟ้าหลายๆ ขั้วรอบสเตเตอร์และขั้วเหล่านั้นถูกกระตุ้นด้วยกระแสพัลส์ในรูปแบบที่เรียงลำดับกันไป โรเตอร์ก็จะหมุนในลักษณะเป็นสเตปที่เป็นไปตามการหมุนของสนามแม่เหล็กที่เกิดจากการสวิตซ์ที่เรียงลำดับของขดลวดขั้วแม่เหล็กไฟฟ้าของสเตเตอร์

## 2.2.1 โครงสร้างและหลักการทำงานของสเตปมอเตอร์ชนิดต่างๆ

ในการนำสเตปมอเตอร์ไปใช้งาน ก่อนอื่นควรจะต้องทราบถึงโครงสร้างและหลักการทำงานของสเตปมอเตอร์ชนิดต่างๆ แล้วเปรียบเทียบถึงความสามารถและขีดจำกัดการใช้งาน เพื่อที่จะสามารถเลือกใช้สเตปมอเตอร์ได้อย่างเหมาะสม

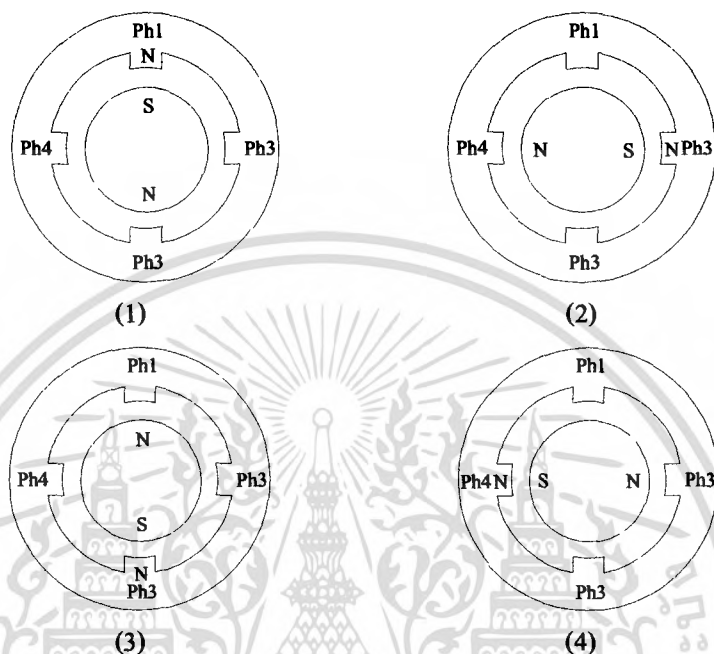
### 2.2.1.1 สเตปมอเตอร์แบบแม่เหล็กถาวร

โครงสร้างของสเตปมอเตอร์แบบแม่เหล็กถาวร ดังรูปที่ 2.4 มีโรเตอร์ทำจากแม่เหล็กถาวร ส่วนสเตเตอร์มีลักษณะเป็นซี่ฟันล้อมรอบ ซี่ฟันของสเตเตอร์ถูกพันด้วยขดลวดสำหรับสร้างสนามแม่เหล็ก เมื่อต้องการใช้งานสเตปมอเตอร์แบบแม่เหล็กถาวรที่มีขนาดมุมสเตปเล็กจะต้องเพิ่มจำนวนขั้วแม่เหล็กของโรเตอร์และจำนวนซี่ฟันของสเตเตอร์ แต่ก็มีข้อจำกัดในการเพิ่มจำนวนขั้วแม่เหล็กของโรเตอร์ เนื่องจากการสร้างแม่เหล็กถาวรให้มีโครงสร้างแบบมีขั้วแม่เหล็กหลายขั้วทำได้ยาก



รูปที่ 2.4 แสดงโครงสร้างของสเตปมอเตอร์แบบแม่เหล็กถาวร

ตัวอย่างการทำงานของสเตปมอเตอร์แบบแม่เหล็กถาวร สมมติว่าสเตปมอเตอร์แบบแม่เหล็กถาวรขนาดสี่เฟสโดยโครงสร้างของโรเตอร์เป็นแม่เหล็กถาวรทรงกระบอกเรียบและสเตเตอร์มีจำนวนซี่ฟัน 4 ซี่ฟันซึ่งรอบๆ พันด้วยขดลวด เมื่อสร้างสัญญาณกระตุ้นตามลำดับเฟส โรเตอร์จะหมุนไปตามทิศทางของการกระตุ้นดังแสดงในรูปที่ 2.5

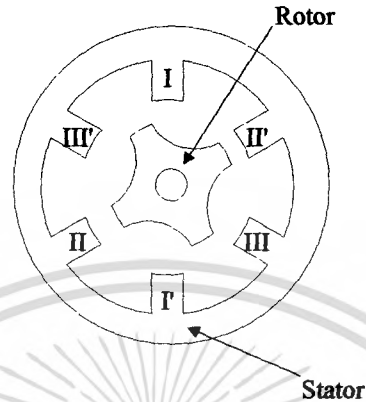


รูปที่ 2.5 แสดงการทำงานของสเตปมอเตอร์แบบแม่เหล็กถาวรขนาดสี่เฟส

ข้อเสียของสเตปมอเตอร์แบบแม่เหล็กถาวรคือมีขนาดมุมแม่เหล็กใหญ่ ทำให้มีความละเอียดของจำนวนสเตปต่อรอบน้อยเนื่องจากโครงสร้างของโรเตอร์เป็นแม่เหล็กถาวร การสร้างแม่เหล็กถาวรให้มีหลายขั้วทำได้ยากทำให้ไม่สามารถสร้างสเตปขนาดเล็กได้ สเตปมอเตอร์แบบแม่เหล็กถาวรส่วนใหญ่จะมีโครงสร้างขนาดเล็กทำให้ค่าทอร์ค ที่ได้ต่อหน่วยปริมาตรมีค่าต่ำ ถ้าต้องการปรับปรุงประสิทธิภาพในเรื่องทอร์ค แม่เหล็กถาวรที่ใช้จะต้องทำมาจากสารแม่เหล็กที่มีสภาพความเป็นแม่เหล็กสูง แต่สารแม่เหล็กที่มีสภาพความเป็นแม่เหล็กสูงมีราคาแพง สารแม่เหล็ก เช่น เฟอร์ไรท์ มีราคาถูกแต่สภาพความเป็นแม่เหล็กต่ำทำให้ค่าทอร์คต่ำ

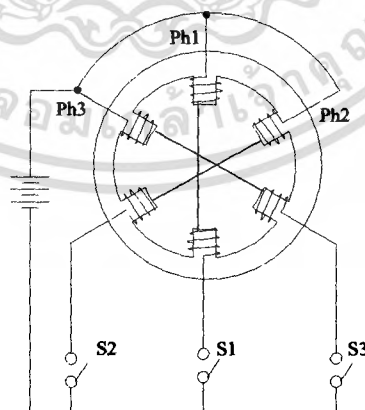
### 2.2.1.2 สเตปมอเตอร์แบบแปรค่ารีล็กแทนซ์

โครงสร้างของสเตปมอเตอร์แบบแปรค่ารีล็กแทนซ์ ดังรูปที่ 2.6 มีโรเตอร์และสเตเตอร์ทำมาจากโลหะซิลิกอน ซึ่งเป็นสารแม่เหล็กที่มีสภาพซึมซับทางแม่เหล็กสูงและยอมให้สนามแม่เหล็กจำนวนมากไหลผ่านได้

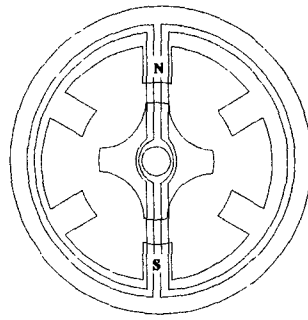


รูปที่ 2.6 แสดงโครงสร้างของสเตปมอเตอร์แบบแปรค่ารีล็กแทนซ์

ตัวอย่างการทำงานของสเตปมอเตอร์แบบแปรค่ารีล็กแทนซ์ จากรูปที่ 2.7 สมมติสเตปมอเตอร์แบบแปรค่ารีล็กแทนซ์ขนาดสามเฟสโดยโครงสร้างของโรเตอร์มี 4 ซี่ฟันและสเตเตอร์มี 6 ซี่ฟัน แต่ละเฟสของสเตเตอร์มี 2 ซี่ฟันที่อยู่ตรงข้ามกัน ดังนั้นซี่ฟัน I, II, III จะเป็นซี่ฟันหัวและ I', II', III' จะเป็นซี่ฟันหาง เมื่อสเตปมอเตอร์ได้รับการกระตุ้น เฟสที่ได้รับการกระตุ้นจะเกิดสนามแม่เหล็ก ถ้าโรเตอร์ I และ I' วางตัวอยู่ในแนวเดียวกันซึ่งจะมีค่ารีล็กแทนซ์น้อยที่สุดสถานะนั้นเป็นสถานะที่จุดสมดุลเป็นจุดซึ่งเส้นแรงแม่เหล็กสั้นที่สุดแสดงดังรูปที่ 2.8

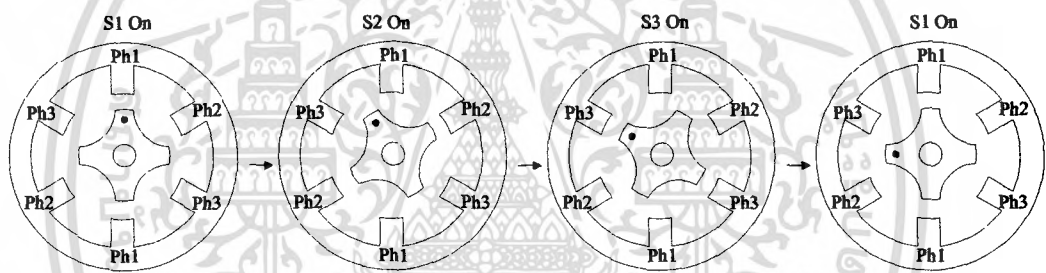


รูปที่ 2.7 แสดงการทำงานของสเตปมอเตอร์แบบแปรค่ารีล็กแทนซ์ขนาดสามเฟส



รูปที่ 2.8 แสดงตำแหน่งสมมูลของสเตปมอเตอร์เมื่อเฟสใดเฟสหนึ่งถูกกระตุ้น

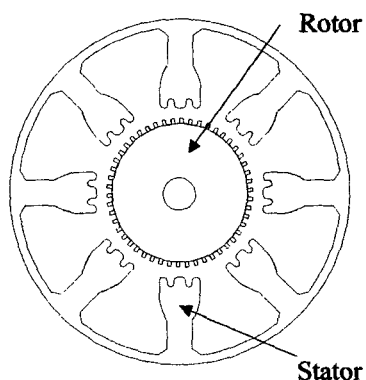
จากรูปที่ 2.8 สมมติว่าโรเตอร์และสเตเตอร์ไม่ได้อยู่ในแนวเดียวกับเส้นแรงของเฟสที่ถูกกระตุ้นจะมีค่ารีลักแตนซ์มาก สเตปมอเตอร์แบบแปรค่ารีลักแตนซ์จะทำให้มีค่ารีลักแตนซ์ต่ำที่สุด โดยวางตัวอยู่ในแนวเดียวกับแนวของเส้นแรงแม่เหล็ก ส่วนการเปลี่ยนลำดับสเตปในการเคลื่อนที่ของสเตปมอเตอร์แบบแปรค่ารีลักแตนซ์แสดงได้ดังรูปที่ 2.9



รูปที่ 2.9 แสดงการเปลี่ยนลำดับสเตปในการเคลื่อนที่ของสเตปมอเตอร์แบบแปรค่ารีลักแตนซ์

### 2.2.1.3 สเตปมอเตอร์แบบไฮบริด

สเตปมอเตอร์แบบไฮบริด ดังรูปที่ 2.10 เป็นรูปแบบหนึ่งของสเตปมอเตอร์ที่มีโรเตอร์เป็นแม่เหล็กถาวร ไฮบริด หมายถึง การทำงานร่วมกันของมอเตอร์แบบแม่เหล็กถาวรและมอเตอร์แบบแปรค่ารีลักแตนซ์ สเตปมอเตอร์แบบไฮบริดมีโครงสร้างของสเตเตอร์คล้ายกับโครงสร้างของสเตปมอเตอร์แบบแปรค่ารีลักแตนซ์ แต่การต่อขดลวดต่างกัน โดยในหนึ่งเฟสของสเตปมอเตอร์แบบแปรค่ารีลักแตนซ์จะมีสองขดลวดและขดลวดทั้งสองจะพันอยู่ที่ขั้วเดียวกัน โดยที่หนึ่งจะมีค่าแอมแปร์หนึ่งเฟส ขณะที่สเตปมอเตอร์แบบไฮบริดจะมีขดลวดของสองเฟสถูกพันอยู่ที่ขั้วเดียวกัน โดยที่ขั้วหนึ่งจะไม่เป็นของเฟสใดเฟสหนึ่ง โครงสร้างของขดลวดสองขดลวดที่ถูกพันที่ขั้วเดียวกันถูกเรียกว่าเป็นรูปแบบของไบไฟลาร์ (Bifilar) ซึ่งจะให้ขั้วแม่เหล็กที่แตกต่างกันในการกระตุ้นแต่ละครั้ง



รูปที่ 2.10 แสดงโครงสร้างของสเตปมอเตอร์แบบไฮบริด

คุณสมบัติที่สำคัญของสเตปมอเตอร์แบบไฮบริดคือ โครงสร้างของมอเตอร์มีแม่เหล็กถาวรอยู่ที่กึ่งกลางระหว่างเฟสทั้งสอง การเหนี่ยวนำสนามแม่เหล็กทำได้โดยใช้สนามแม่เหล็กซึ่งสร้างจากสเตเตอร์ซึ่งเป็นสนามแม่เหล็กแบบเฮเทอโพลาร์ (Heteropolar Field) ดังนั้นทอร์คเกิดจากการทำงานร่วมกันของสนามแม่เหล็กสองชนิดคือ สนามจากแม่เหล็กถาวร และสนามแม่เหล็กเหนี่ยวนำที่เกิดจากการกระตุ้นขดลวดแต่ละขด โครงสร้างซี่ฟันของสเตเตอร์จะใหญ่กว่าซี่ฟันของโรเตอร์เล็กน้อย เพื่อเพิ่มความถูกต้องแม่นยำทางตำแหน่งของการเคลื่อนที่

หลักการทำงานของสเตปมอเตอร์แบบไฮบริดที่แตกต่างจากสเตปมอเตอร์แบบแปรค่ารีลักแตนซ์คือ แรงบิดที่เกิดจากสนามแม่เหล็กจะไม่ขึ้นอยู่กับกระแสที่ไหลผ่านขดลวดอย่างเดียวแต่ขึ้นอยู่กับโครงสร้างของซี่ฟันด้วยซึ่งซี่ฟันถูกออกแบบเพื่อให้ได้โครงสร้างขนาดเล็ก และใช้แม่เหล็กถาวรเป็นแกนกลางเพื่อลดผลของการฮอสซิลเลททางแมคคานิกส์

ข้อดีของสเตปมอเตอร์แบบไฮบริดคือ ขนาดสเตปมีขนาดเล็ก มีความละเอียดต่อรอบสูง และมีค่าทอร์คสูงกว่าสเตปมอเตอร์แบบแปรค่ารีลักแตนซ์ แต่สเตปมอเตอร์แบบแปรค่ารีลักแตนซ์มีแรงเฉื่อยทางแมคคานิกส์น้อยกว่าสเตปมอเตอร์แบบไฮบริด

นอกจากสเตปมอเตอร์ทั้งสามที่กล่าวมาแล้วยังมีสเตปมอเตอร์ชนิดอื่นๆ ที่ไม่ได้กล่าวถึง เช่น ลิเนียร์สเตปมอเตอร์ ซึ่งเป็นสเตปมอเตอร์ที่ได้รับการออกแบบให้มีการเคลื่อนที่แบบเชิงเส้น อิเลคโตรไฮดรอลิกสเตปมอเตอร์ ซึ่งเป็นสเตปมอเตอร์กำลังสูงสำหรับการใช้ในอุตสาหกรรม เป็นต้น

## 2.2.2 การกระตุ้นเฟสของสเตปมอเตอร์

การกระตุ้นเฟสของสเตปมอเตอร์เป็นวิธีการทำให้สเตปมอเตอร์หมุน โดยการจ่ายกระแสไฟฟ้าหรือแรงดันไฟฟ้าเข้าไปกระตุ้นเฟสของสเตปมอเตอร์ สามารถแบ่งออกเป็น 2 วิธี ดังนี้

### 2.2.2.1 การกระตุ้นเฟสแบบฟูลสเตปมอเตอร์

การกระตุ้นเฟสแบบฟูลสเตปมอเตอร์ สามารถแบ่งออกเป็น 2 วิธี ดังนี้

#### 1. การกระตุ้นเฟสแบบฟูลสเตป 1 เฟส

การกระตุ้นเฟสแบบฟูลสเตป 1 เฟส ในแต่ละสเตปจะทำการกระตุ้นเฟสทีละเฟส เรียงลำดับกันไป ดังตารางที่ 2.1 ซึ่งการกระตุ้นเฟสด้วยวิธีนี้จะให้แรงบิดน้อย

ตารางที่ 2.1 การกระตุ้นเฟสแบบฟูลสเตป 1 เฟส

สเตปที่	เฟส 1	เฟส 2	เฟส 3	เฟส 4
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

#### 2. การกระตุ้นเฟสแบบฟูลสเตป 2 เฟส

การกระตุ้นเฟสแบบฟูลสเตป 2 เฟส ในแต่ละสเตปจะทำการกระตุ้นเฟสทีละ 2 เฟสที่อยู่ติดกันเรียงลำดับกันไป ดังตารางที่ 2.2 ซึ่งการกระตุ้นเฟสด้วยวิธีนี้ให้แรงบิดมากกว่าแบบฟูลสเตป 1 เฟส แต่ใช้กำลังไฟฟ้ามากกว่า เพราะจะต้องกระตุ้นทีละ 2 เฟสต่อ 1 สเตป

ตารางที่ 2.2 การกระตุ้นเฟสแบบฟูลสเตป 2 เฟส

สเตปที่	เฟส 1	เฟส 2	เฟส 3	เฟส 4
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

### 2.2.2.2 การกระตุ้นเฟสแบบฮาล์ฟสเตปมอเตอร์

การกระตุ้นเฟสแบบฮาล์ฟสเตปมอเตอร์ เป็นการรวมกันระหว่างการกระตุ้นแบบฟูลสเตป 1 เฟส และ 2 เฟส โดยการกระตุ้นเฟสเรียงสลับกันไปเรื่อยๆ ดังตารางที่ 2.3 ซึ่งการกระตุ้นเฟสด้วยวิธีนี้จะให้แรงบิดน้อยกว่าแบบฟูลสเตปทั้ง 2 ชนิดแต่จะให้ความละเอียดของจำนวนสเตปต่อรอบการหมุนมากกว่า เพราะการกระตุ้นแบบฮาล์ฟสเตป 2 สเตปจะเท่ากับการกระตุ้นแบบฟูลสเตปเพียง 1 สเตป

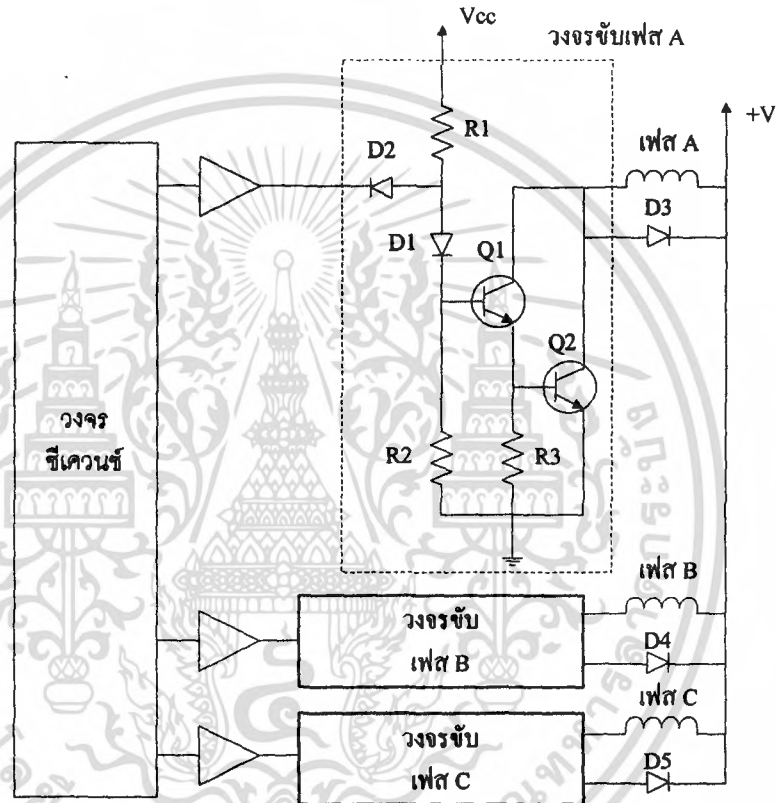
ตารางที่ 2.3 การกระตุ้นเฟสแบบฮาล์ฟสเตปมอเตอร์

สเตปที่	เฟส 1	เฟส 2	เฟส 3	เฟส 4
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

### 2.3 การควบคุมการทำงานของสเตปมอเตอร์

วงจรที่ใช้ควบคุมสเตปมอเตอร์ส่วนใหญ่จะเป็นวงจรลอจิกและวงจรภาคกำลัง ซึ่งใช้สำหรับสเตปมอเตอร์แบบแม่เหล็กถาวร สเตปมอเตอร์แบบแปรค่ารีลักแตนซ์ และสเตปมอเตอร์แบบไฮบริด ประสิทธิภาพของสเตปมอเตอร์ขึ้นอยู่กับวงจรขับที่ใช้ เช่น ไดนามิกทอร์ค ความเร็ว การตอบสนองต่อความถี่และการแดมป์ (Damping) เป็นต้น นอกจากนั้นยังมีพารามิเตอร์บางอย่าง เช่น ประสิทธิภาพ (Efficiency) และการสูญเสียกำลัง (Power Dissipation) ก็ขึ้นอยู่กับวงจรขับเช่นกัน

รูปแบบการควบคุมสเตปมอเตอร์โดยทั่วไปเป็นการควบคุมภาควงจรถ้าลังและลำดับการจ่ายพลังงานให้กับขดลวดในเฟสต่างๆ ของสเตปมอเตอร์ วงจรถ้าลังจะทำงานในทิศทางเดียวหรือสองทิศทางขึ้นอยู่กับวงจรถ้าลังลำดับลอจิก ซึ่งใช้เป็นสัญญาณขับขาเบสของทรานซิสเตอร์ถ้าลังซึ่งจะเพิ่มขนาดเพื่อจ่ายกระแสให้ขดลวดของมอเตอร์และต้องใช้วงจรถ้าลังจำกัดกระแสเมื่อใช้การกระตุ้นแบบใช้แรงดันเกิน (Over Excitation) เพื่อรักษาเสถียรภาพการทำงานของสเตปมอเตอร์ และใช้วงจรถ้าลังจำกัดแรงดันเกิน (Suppression Circuit) เพื่อกำจัดสไปค์โวลเตจ (Spike Voltage) ให้อยู่ในระดับที่ปลอดภัยของการใช้งานทรานซิสเตอร์ถ้าลัง



รูปที่ 2.11 แสดงระบบควบคุมการขับสเตปมอเตอร์

จากรูปที่ 2.11 ในส่วนของวงจรถ้าลังลำดับลอจิก (Ring Counter) ซึ่งมีขนาดเท่ากับจำนวนเฟสของสเตปมอเตอร์ การใช้วงจรถ้าลังลำดับลอจิกจะกระตุ้นวงจรถ้าลังได้ครั้งละหนึ่งเฟสเรียงลำดับกันไป สัญญาณจากวงจรถ้าลังลำดับลอจิกจะต้องถูกขยายให้เพียงพอที่จะจ่ายให้กับสเตปมอเตอร์ ดังนั้นจึงต้องป้อนสัญญาณนี้ให้กับวงจรถ้าลังทรานซิสเตอร์ถ้าลัง (Power Transistor) อีกทีหนึ่งเพื่อที่จะเพิ่มค่ากระแสให้เพียงพอที่จะจ่ายให้กับขดลวดของสเตปมอเตอร์

เฟสของมอเตอร์ หมายถึง จำนวนขดลวดที่พันบนสเตเตอร์ที่แยกออกจากกันอย่างอิสระ ซึ่งการพันขดลวดของสเตปมอเตอร์แบ่งเป็น 2 แบบ คือ แบบไบโพลาร์ (Bipolar) ขดลวดที่สเตเตอร์แต่ละชุดจะไม่มีจุดร่วม (Common) โดยการพันขดลวดจะพันแบบยูนิโพลาร์ (Unifilar) ใช้ลวดเส้นเดียว และแบบยูนิโพลาร์ (Unipolar) ขดลวดที่สเตเตอร์แต่ละชุดจะมีจุดร่วม (Common) โดยการพันขดลวดพันแบบไบโพลาร์ ใช้ลวดเส้น 2 เส้นพันไปด้วยกัน

การขยายวงจรจับทำได้โดยใช้วงจรแบบคาร์ลิงตัน (Darlington) เมื่อสัญญาณจากวงจรจัดลำดับลอจิกเท่ากับหนึ่ง จะเป็นสัญญาณขับวงจรขับให้วงจรคาร์ลิงตันทำงาน กระแสไหลจากแหล่งจ่ายไฟเลี้ยงผ่านสเตปมอเตอร์ลงสู่กราวด์ และเมื่อสัญญาณจากวงจรจัดลำดับลอจิกเท่ากับศูนย์ จะหยุดการทำงานของทรานซิสเตอร์กำลัง

แม้ว่าวงจรคาร์ลิงตันจะมีโวลเตจอิ่มตัว (Saturation Voltage) ที่สูงกว่าขนาดวงจรที่ใช้ทรานซิสเตอร์ตัวเดียว แต่วงจรคาร์ลิงตันนี้ไม่ต้องการสัญญาณขับเบสเป็นโวลเตจสูง ซึ่งใช้ลอจิกสูงของทีทีแอลก็เพียงพอที่จะขับเบสของวงจรคาร์ลิงตันให้นำกระแสได้ ดังนั้นจึงต่อไดโอด D1 อนุกรมเข้ากับขาเบสของ Q1 เพื่อแยกโวลเตจต่ำและโวลเตจสูงออกจากกัน การไอโซเลชัน (Isolation) แบบนี้จะช่วยป้องกันการเบรคควาน์ของคอลเลคเตอร์และเบสของทรานซิสเตอร์กำลัง

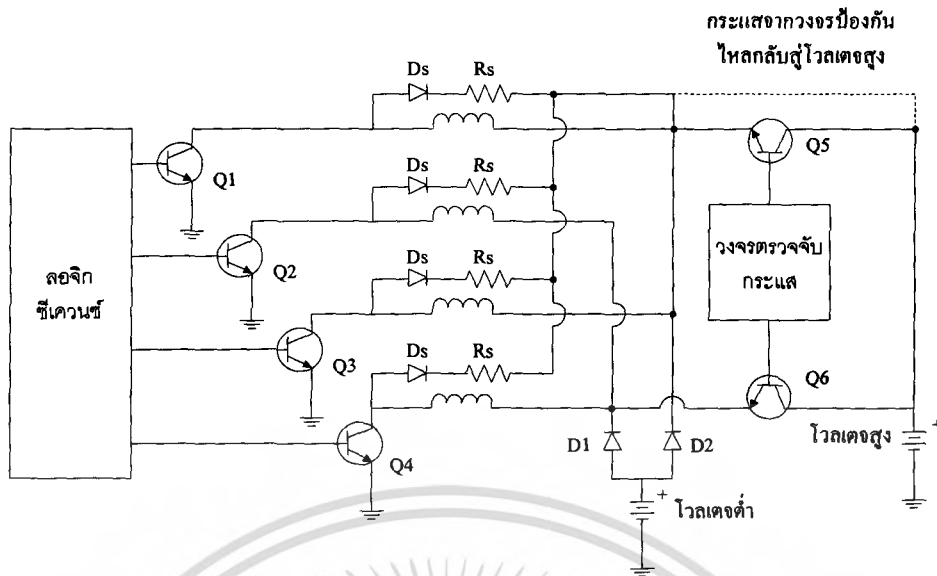
## 2.4 วงจรขับสเตปมอเตอร์โดยทั่วไป

ในการนำสเตปมอเตอร์ไปใช้งานได้อย่างมีประสิทธิภาพนั้น นอกจากจะต้องพิจารณาโครงสร้างและหลักการทำงานของสเตปมอเตอร์ชนิดต่างๆ แล้วยังต้องพิจารณาเลือกใช้วงจรขับที่มีความเหมาะสมเพื่อนำสเตปมอเตอร์ไปใช้งานได้อย่างมีประสิทธิภาพสูงสุด

### 2.4.1 วงจรขับแบบโวลเตจสองระดับ

วิธีการนี้เรียกว่าการขับแบบโวลเตจคู่หรือการขับแบบสองระดับ มีการทำงานสวิตช์ไปที่โวลเตจสูงเมื่อกระแสมาถึงระดับที่ต้องการและเมื่อถึงเวลาที่ตั้งไว้โวลเตจสูงจะสวิตช์ออฟ กระแสที่อยู่ในขดลวดนี้จะไหลผ่านไดโอด D1 โดยโวลเตจระดับต่ำจะทำงานแทน

เมื่อพิจารณาการลดลงของเวลาในการเปลี่ยนระดับของกระแสของระบบการขับสองระดับ โวลเตจคร่อมขดลวดจะยังคงอยู่ทั้งหมดจนกระทั่งกระแสเพิ่มขึ้นจนถึงระดับที่ต้องการ ค่าการยกระดับกระแสก่อนข้างจะมีลักษณะเป็นเชิงเส้น หลังจากโวลเตจสูงสวิตช์ไปสู่โวลเตจต่ำกระแสจะตกลงและเกิดแรงดันย้อนกลับ



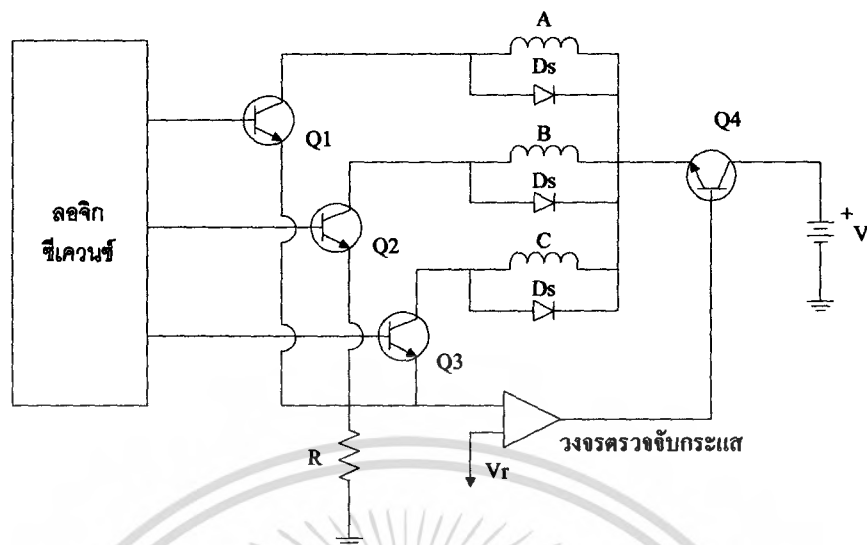
รูปที่ 2.12 แสดงวงจรขับเคลื่อนมอเตอร์แบบ โวลเตจสองระดับ

จากรูปที่ 2.12 ขณะที่ทรานซิสเตอร์ Q5 และ Q6 ไม่ทำงาน ไดโอด D1 และ D2 ของระดับ โวลเตจต่ำนำกระแสอยู่ เมื่อโวลเตจสูงทำงานทรานซิสเตอร์ Q5 และ Q6 นำกระแส ไดโอด D1 ต้องหยุดนำกระแสทันทีและแหล่งจ่ายโวลเตจต่ำจะกลายเป็นระบบภายนอก การทำงานจะเป็นลักษณะนี้สลับกันไปตลอดช่วงการทำงาน ไดโอดกระแสสูงโดยทั่วไปจะมีค่าเวลาดำเนินตัวช้ามักจะอยู่ในระดับ  $\mu\text{sec}$  ดังนั้นในกรณีนี้จึงมีความจำเป็นต้องใช้ไดโอดชนิดคืนตัวเร็ว (Fast Recovery Diode) เพื่อให้เกิดการเกิดสไปค์ของกระแส (Current Spike) ในขณะที่มีแหล่งจ่ายไฟเลี้ยงอยู่ในระดับที่ปลอดภัย โดยที่แหล่งจ่ายไฟเลี้ยงที่มีระดับโวลเตจต่ำควรออกแบบให้มีระดับกระแสที่เพียงพอในการต้านกระแสย้อนกลับ

#### 2.4.2 วงจรขับเคลื่อนขอลเปอร์

เป็นอีกวิธีการหนึ่งที่จะสามารถใช้งานมอเตอร์ได้อย่างมีประสิทธิภาพสูง คือการใช้พัลส์วิดท์มอดูเลชัน (Pulse Width Modulation) หรือฟริควเอนซีมอดูเลชัน (Frequency Modulation) ควบคุมกระแสเฉลี่ยที่จะให้กับสเตรปมอเตอร์ วงจรง่าย ๆ ของวงจรขับเคลื่อนขอลเปอร์แสดงดังรูปที่ 2.13 เมื่อเฟลตไคเฟสหนึ่งของมอเตอร์ออน ทรานซิสเตอร์ Q4 จะสวิทช์ไปสู่โวลเตจระดับสูงออนและออฟตลอดเวลา การใช้ไฟเลี้ยงที่มีระดับโวลเตจสูงเป็นสาเหตุให้กระแสในแต่ละเฟสเพิ่มระดับอย่างทันทีทันใด เหมือนในวงจรขับเคลื่อนโวลเตจสองระดับ มีวงจรตรวจระดับกระแสเพื่อจะเซ็ทค่าการออฟของสวิทช์ทรานซิสเตอร์ Q4 เมื่อมีกระแสมาถึงระดับที่ต้องการ และเมื่อกระแสลดต่ำกว่าค่าเทรชโฮลด์ (threshold) ทรานซิสเตอร์ Q4 จะออนอีกครั้งหนึ่ง การทำงานของวงจรมีจะยอมให้

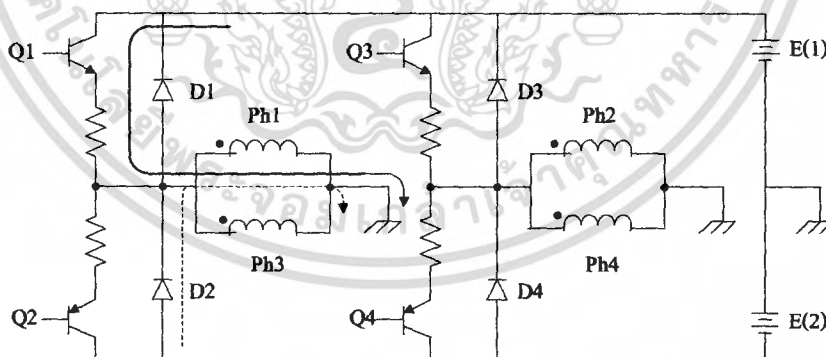
เอกสารนี้กระแสวิกฤตอยู่ในระดับที่ต้องการ การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 แสดงวงจรจับสเตปมอเตอร์แบบซอปเปอร์

2.4.3 วงจรขับแบบบริดจ์

ประสิทธิภาพในการทำงานของสเตปมอเตอร์จะสูงขึ้นถ้าควบคุมการไหลของกระแสให้ไหลได้ในสองทิศทาง ซึ่งช่วยให้ทอร์คเพิ่มขึ้น 20-36 % ของการใช้วงจรขับแบบมีกระแสไหลทางเดียว วงจรขับสเตปมอเตอร์แบบบริดจ์มีสองชนิด คือ แบบมีแหล่งจ่ายไฟเลี้ยงเดี่ยว แบบมีสองไฟเลี้ยงเดี่ยว (Single Supply) และแบบมีแหล่งจ่ายไฟเลี้ยงสองแหล่ง (Double Supplies)



รูปที่ 2.14 แสดงวงจรขับสเตปมอเตอร์แบบบริดจ์สำหรับการขับแบบไบโพลาร์

จากรูปที่ 2.14 ซึ่งแสดงวงจรขับเคลื่อนมอเตอร์แบบบริดจ์สำหรับการขับเคลื่อนแบบไบโพลาร์ มีคุณสมบัติการทำงานดังต่อไปนี้

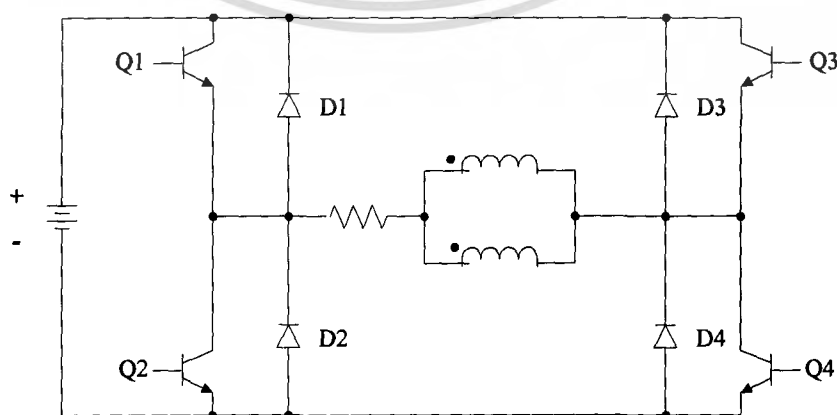
1. การต่อขดลวดของมอเตอร์ ถ้ามอเตอร์มี 8 ขั้ว กระแสกลับจะสามารถไหลผ่านได้เหมือนมอเตอร์สองเฟส

2. การป้องกันอันตรายอันเกิดจากทรานซิสเตอร์ร้อน ในขณะที่อีกตัวหนึ่งยังนำกระแสอยู่ ทำได้โดยการใส่วงจรหน่วงเวลาคือความต้านทานเข้าไประหว่างทรานซิสเตอร์ทั้งสอง

3. การใช้ไดโอดเพื่อป้องกันสไปค์โวลเตจ โดยไดโอดจะถูกใส่ขนานเข้าไปกับทรานซิสเตอร์กำลังเพื่อกำจัดสไปค์โวลเตจ เมื่อทรานซิสเตอร์ "Off" การทำงานของไดโอดจะแตกต่างจากแบบยูนิโพลาร์ ถ้า Q1 นำกระแสซึ่งแสดงโดยเส้นทึบ เมื่อ Q1 "Off" และ Q2 "On" กระแสไม่ได้ไหลกลับทิศ แต่จะไหลผ่านไดโอด D2 และแบตเตอรี่ E2 ซึ่งแสดงโดยเส้นประ ดังนั้นจึงอาจกล่าวได้ว่า D2 ตอนนี้จะถูกประจุ หรือถ้าพิจารณาในเทอมของพลังงาน พลังงานจากสนามแม่เหล็กจะถูกป้อนกลับเข้าสู่แหล่งจ่ายไฟเลี้ยง ในขณะที่การขับเคลื่อนมอเตอร์แบบยูนิโพลาร์ พลังงานจากสนามแม่เหล็กจะถูกใช้ไปในขดลวด ไดโอด ความต้านทานภายนอกและซีเนอร์ไดโอด ซึ่งทำให้การขับเคลื่อนไบโพลาร์มีข้อดีกว่า

4. กระแสในทิศทางกลับสำหรับการขับเคลื่อนยูนิโพลาร์ต้องใช้วงจรจำกัดแรงดันเกิน เพื่อแค้มพ์กระแสหลังจากออฟทรานซิสเตอร์ แต่สำหรับการขับเคลื่อนไบโพลาร์ จะให้โวลเตจเพื่อให้กระแสไหลกลับทิศ หลังจากกระแสที่เหลืออยู่หลังจากทรานซิสเตอร์ออฟกลายเป็นศูนย์ และจะมีกระแสชุดใหม่เพิ่มขึ้นผ่านทรานซิสเตอร์ Q2 ในทิศทางตรงข้าม ดังนั้นรูปคลื่นของกระแสจะไม่เป็นเอกซ์โปเนนเชียล ขึ้นอยู่กับแรงเคลื่อนไฟฟ้าที่เกิดจากการเคลื่อนที่ของโรเตอร์

อีกรูปแบบหนึ่งของการขับเคลื่อนไบโพลาร์ ดังแสดงในรูป 2.15 ซึ่งกรณีนี้จะใช้แหล่งจ่ายไฟเลี้ยงเดียว แต่ใช้ทรานซิสเตอร์สี่ตัวสำหรับแต่ละเฟส ความต้านทานจะถูกต่อขนานกับขดลวดเพื่อการเพิ่มกระแสที่เร็วขึ้น วงจรหน่วงเวลาใช้เพื่อป้องกันทรานซิสเตอร์นำกระแสคาบเกี่ยวกัน



รูปที่ 2.15 แสดงอีกรูปแบบหนึ่งของวงจรขับเคลื่อนแบบบริดจ์ซึ่งใช้แหล่งจ่ายไฟเลี้ยงเดียว

## 2.5 สรุป

เนื้อหาในบทนี้กล่าวถึงทฤษฎีพื้นฐาน โครงสร้าง หลักการทำงานของสเตปมอเตอร์ชนิดต่างๆ และการทำให้สเตปมอเตอร์เคลื่อนที่โดยการกระตุ้นเฟส นอกจากนี้ยังกล่าวถึงวงจรควบคุมและวงจรที่ใช้ในการจับสเตปมอเตอร์โดยทั่วไป ซึ่งต้องทำการพิจารณาถึงความสามารถ และข้อจำกัดของสเตปมอเตอร์และวงจรจับ เพื่อการนำสเตปมอเตอร์ไปใช้งานได้อย่างมีประสิทธิภาพ



### บทที่ 3

## หลักการออกแบบและการสร้าง

### 3.1 บทนำ

โดยทั่วไปการควบคุมการขับเคลื่อนมอเตอร์จะใช้วงจรขับแบบโวลเตจคองท์ ซึ่งควบคุมการเคลื่อนที่ของสเตปมอเตอร์ได้เฉพาะการขับแบบฟูลสเตปและแบบฮาล์ฟสเตปเท่านั้น ไม่สามารถควบคุมการขับเคลื่อนที่ละเอียดขึ้นได้จึงได้มีการสร้างชุดควบคุมขึ้นมาเพื่อควบคุมการขับเคลื่อนมอเตอร์แบบไมโครสเตป โดยการสร้างสัญญาณควบคุมการจ่ายกระแสไฟของแต่ละเฟสให้เป็นสัดส่วนต่อกัน โดยการทำงานของวงจรสร้างสัญญาณควบคุม อาศัยหลักการควบคุมระดับแรงดันไฟฟ้าเฉลี่ยโดยวิธีพัลส์วidthมอดูเลชัน

### 3.2 ส่วนประกอบของระบบควบคุมการขับเคลื่อนมอเตอร์แบบไมโครสเตป



รูปที่ 3.1 แสดงบล็อกไดอะแกรมของระบบควบคุมการขับเคลื่อนมอเตอร์แบบไมโครสเตป

ระบบควบคุมการขับเคลื่อนมอเตอร์แบบไมโครสเตป ซึ่งแสดงในบล็อกไดอะแกรมรูปที่ 3.1 นั้น ประกอบด้วยไมโครคอนโทรลเลอร์ ส่วนควบคุมแรงดัน วงจรขับ และสเตปมอเตอร์ ไมโครคอนโทรลเลอร์จะเชื่อมต่อกับคอมพิวเตอร์ เพื่อความสะดวกในการแก้ไขข้อมูลและการสร้างสัญญาณควบคุมได้ตามต้องการ ในการขับเคลื่อนมอเตอร์แบบไมโครสเตปนั้น ไมโครคอนโทรลเลอร์จะเป็นตัวควบคุมการจ่ายกระแสไฟของแต่ละเฟส เป็นการส่งสัญญาณควบคุมให้กับส่วนควบคุมแรงดันเพื่อควบคุมหรือแบ่งระดับแรงดันไฟฟ้าให้ได้ตามที่ต้องการ เข้าสู่ส่วนของวงจรขับ เพื่อนำสัญญาณที่ได้นั้น ไปกระตุ้นเฟสของสเตปมอเตอร์แต่ละเฟสให้สเตปมอเตอร์เกิดการเคลื่อนที่เป็นสเตป

### 3.2.1 ไมโครคอนโทรลเลอร์

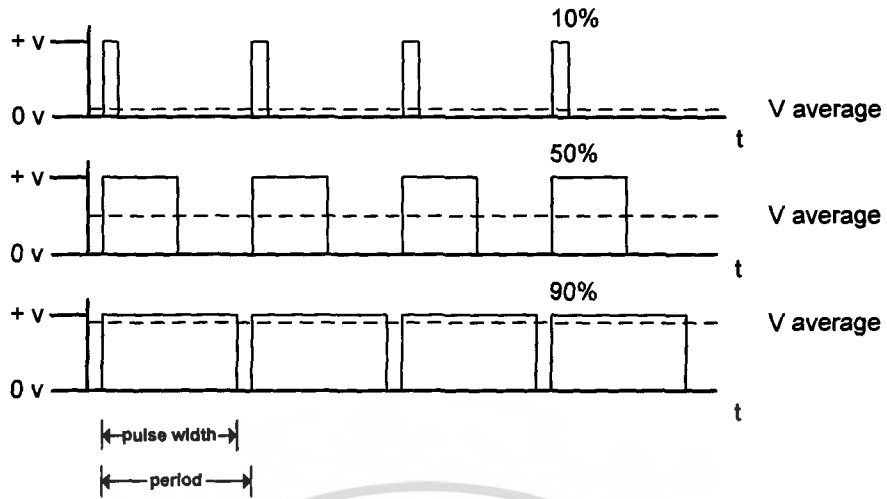
การควบคุมการขับเคลื่อนมอเตอร์แบบไมโครสเตป จะมีความสะดวกในการควบคุมมากยิ่งขึ้น ถ้าใช้ไมโครคอนโทรลเลอร์มาเป็นตัวควบคุมการทำงานของระบบแทนที่จะใช้ระบบเป็นส่วนของฮาร์ดแวร์ทั้งหมด เนื่องจากการใช้ไมโครคอนโทรลเลอร์สามารถเปลี่ยนแปลงหรือสร้างสัญญาณควบคุมได้ตามความต้องการ ไมโครคอนโทรลเลอร์ที่นำมาใช้คือเบอร์ PIC16F876A

ในการควบคุมสเตปมอเตอร์ โดยใช้สัญญาณควบคุมจากไมโครคอนโทรลเลอร์ ถ้าหากเราขับเคลื่อนมอเตอร์ในโหมดสเตปธรรมดา (Full step , Half step ) นั้น สัญญาณควบคุมจากไมโครคอนโทรลเลอร์จะควบคุมในส่วนของการกระตุ้นเฟสเพียงอย่างเดียว และถ้าขับเคลื่อนมอเตอร์ในแบบไมโครสเตป (1200 steps , 3600 steps ) จะมีสัญญาณควบคุมในส่วนของการแบ่งแรงดันด้วย

### 3.2.2 ส่วนควบคุมแรงดันโดยใช้วิธีพัลส์วิดท์มอดูเลชัน

พัลส์วิดท์มอดูเลชันเป็นวิธีการหนึ่งที่ใช้ในการแบ่งระดับแรงดันไฟฟ้า โดยอาศัยการปรับความกว้างของสัญญาณพัลส์ที่จ่ายออกมาจากแหล่งจ่าย เพื่อให้ระดับแรงดันไฟฟ้าเฉลี่ยที่ได้ออกมาในช่วงเวลาหนึ่งมีระดับที่แตกต่างกัน ลักษณะของการทำงานจะเป็นการทำงานแบบสวิทช์ คือแบบเปิด-ปิด ซึ่งจะทำให้กระแสไฟฟ้าที่ไหลผ่านอุปกรณ์อิเล็กทรอนิกส์กำลังมีการสูญเสียพลังงานในรูปแบบของพลังงานความร้อนน้อยลง แต่การสูญเสียที่เกิดการสวิทช์มีเพิ่มมากขึ้น ในการปรับความกว้างของสัญญาณพัลส์เพื่อปรับระดับแรงดันไฟฟ้าเฉลี่ยดังรูปที่ 3.2 สามารถปรับได้จาก %Duty Cycle

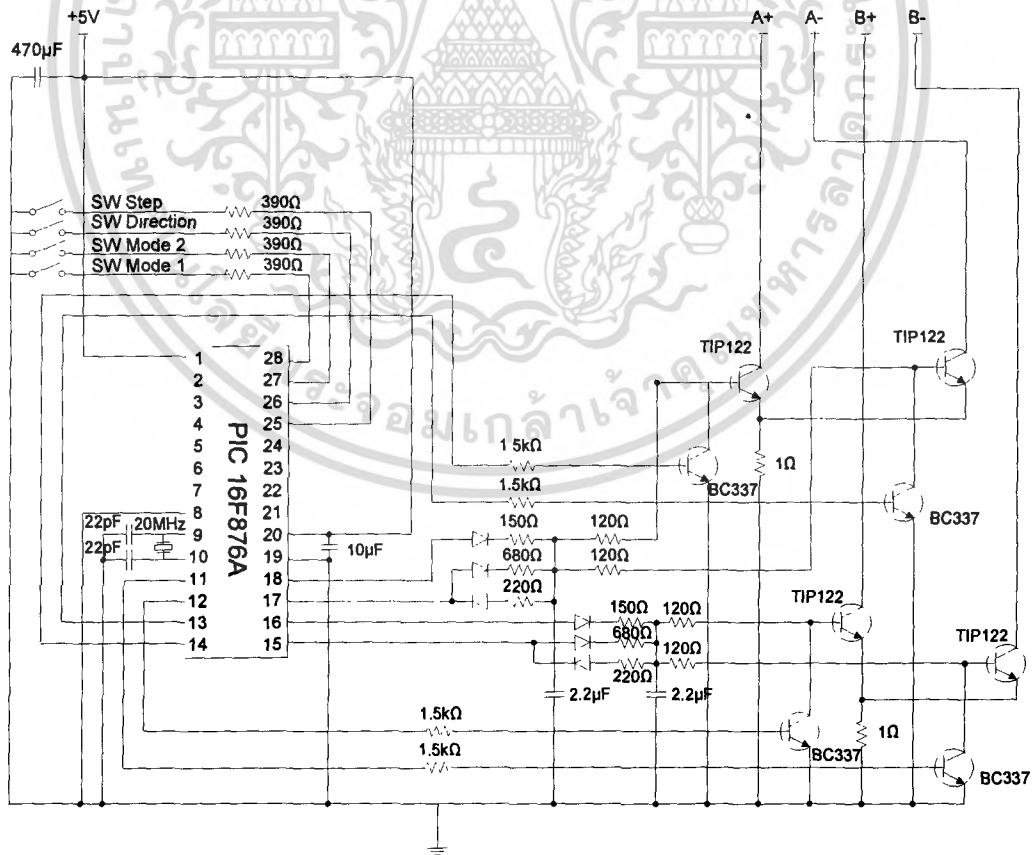
$$\%Duty Cycle = \frac{T_{on}}{T_{on} + T_{off}} \times 100\% \quad (3.1)$$



รูปที่ 3.2 แสดงการปรับความกว้างของสัญญาณพัลส์และระดับแรงดันไฟฟ้าเฉลี่ย

### 3.2.3 วงจรขับสเตปมอเตอร์แบบไมโครสเตป

วงจรที่ใช้ขับสเตปมอเตอร์แบบไมโครสเตปในโครงงานนี้ แสดงได้ดังรูปที่ 3.3



รูปที่ 3.3 แสดงวงจรขับสเตปมอเตอร์แบบไมโครสเตป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในเพื่อการศึกษาเท่านั้น เมื่อผู้ผู้ใดเห็นไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.2.3.1 ไมโครคอนโทรลเลอร์ ใช้เบอร์ PIC 16F876A** ซึ่งมีการเชื่อมต่อกับอุปกรณ์ต่างๆดังนี้

1. ออสซิลเลเตอร์เป็นตัวกำเนิดความถี่ 20 MHz ต่อเข้ากับขา OSC1 และ OSC2
2. แรงดันไฟฟ้าเลี้ยงไอซี +5V ต่อเข้ากับขา Vpp และขา MCLR ส่วนกราวด์ ต่อเข้ากับขา Vss
3. พาวเวอร์ทรานซิสเตอร์ TIP122 จำนวน 4 ตัว และ ทรานซิสเตอร์ BC337 จำนวน 4 ตัว ต่อเข้ากับพอร์ท C ทั้ง 8 ขา (RC0 - RC7)
4. สวิตช์ 3 ตัว ต่อเข้ากับพอร์ท B 3 ขา (RB5,RB6,RB7)

**3.2.3.2 สวิตช์ 4 ตัว** สวิตช์ Mode 1 และ Mode 2 ทำหน้าที่เลือกโหมดในการทำงานของสเตปมอเตอร์ แสดงดังตารางที่ 3.1

ตารางที่ 3.1 แสดงสถานะของสวิตช์ในการเลือกโหมดการทำงานของสเตปมอเตอร์

สถานะของสวิตช์		โหมดการทำงาน
Mode 1	Mode 2	
0	0	Full step
0	1	Half step
1	0	1200 steps
1	1	3600 steps

สวิตช์ไคเรคชัน (Direction Switch) ทำหน้าที่เลือกทิศทางในการหมุนของสเตปมอเตอร์ ถ้า สวิตช์ OFF มอเตอร์จะหมุนตามเข็มนาฬิกา และถ้าสวิตช์ ON มอเตอร์จะหมุนทวนเข็มนาฬิกา

สวิตช์สเตป (Step Switch) ทำหน้าที่เลือกโหมดการทำงานของไมโครคอนโทรลเลอร์ตัวควบคุมการทำงาน ให้เป็นแบบรับสัญญาณจากแหล่งจ่ายสัญญาณพัลส์ ในการหมุน 1 สเตป เมื่อได้รับสัญญาณพัลส์ 1 พัลส์

### 3.2.3.3 แลคเคอร์รี่ซิสเตอร์

ทำหน้าที่เป็นส่วนของการแบ่งแรงดันไฟฟ้า โดยใช้ตัวต้านทาน และไดโอด ซึ่งคล้ายกับหลักการของการแปลงสัญญาณดิจิทัลเป็นอนาลอก โดยต่อกับขาเอาต์พุทพอร์ท ของไมโครคอนโทรลเลอร์ชุดละ 2 ขา ซึ่งจะเป็นตัวกำหนดระดับแรงดันไฟฟ้าที่จุดร่วม ระดับแรงดันไฟฟ้าที่จุดร่วมของแลคเคอร์รี่ซิสเตอร์ จากการทดลอง แสดงดังตารางที่ 3.2

ตารางที่ 3.2 แสดงระดับแรงดัน ไฟฟ้าที่จุ่มร่วมของแลคเคอร์รีซิสเตอร์ จากการทดลอง

สถานะของขาเอาต์พุทพอร์ท		แรงดัน ไฟฟ้าที่จุ่มร่วม (volt)
RC7,RC5	RC6,RC4	
0	0	0
0	1	0.8
1	0	1.0
1	1	1.6

ในส่วนนี้ยังมีคาปาซิเตอร์ ที่เป็นตัวช่วยให้ระดับแรงดัน ไฟฟ้าที่จุ่มร่วมมีการเปลี่ยนแปลงในลักษณะเป็นทางลาด หรือ ไม่ให้มีการเปลี่ยนแปลงแบบทันทีทันใดเมื่อมีการเปลี่ยนแปลงระดับแรงดันไฟฟ้า เรียกว่า แรมป์ปิ้ง คาปาซิเตอร์ (Ramping Capacitor)

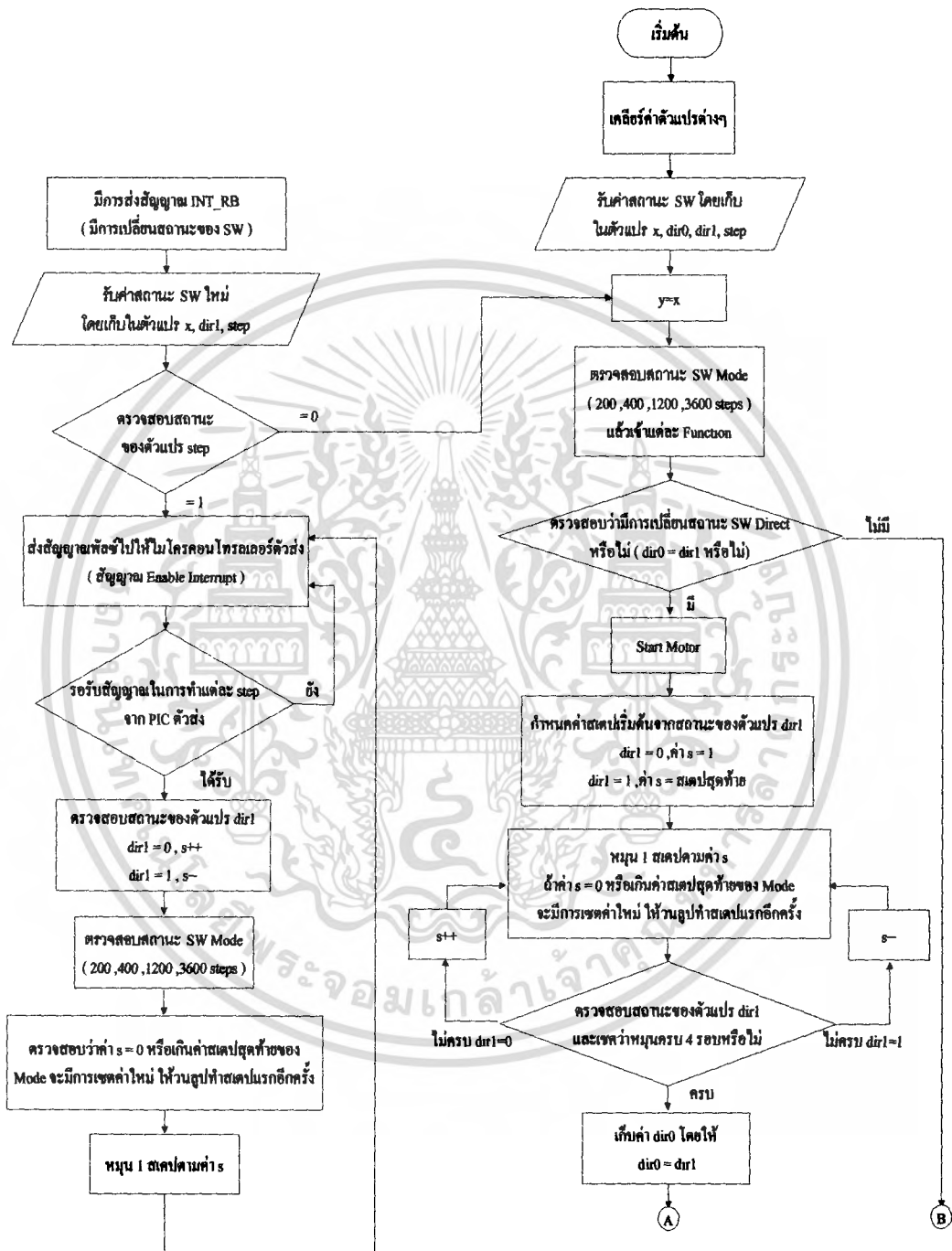
#### 3.2.3.4 ทรานซิสเตอร์

มีการเชื่อมต่อยุบรวมกันคล้ายกับการต่อแบบคาร์ลิงตัน โดยที่ขาคอลเลคเตอร์ ของ TIP122 จะต่อเข้ากับแต่ละเฟสของสเตปมอเตอร์ ที่ขาเบสจะต่อกับจุ่มร่วมของแลคเคอร์รีซิสเตอร์ และขาคอลเลคเตอร์ ของ BC337 และ ขาอีมิเตอร์ ต่อลงกราวด์ เช่นเดียวกับขาอีมิเตอร์ ของ BC337 ส่วนขาเบสของ BC337 จะต่อกับ เอาต์พุทพอร์ทของไมโครคอนโทรลเลอร์

ลักษณะการทำงานของส่วนนี้จะมีพาวเวอร์ทรานซิสเตอร์TIP122 ทำหน้าที่ควบคุมกระแสที่จะผ่านเฟสของสเตปมอเตอร์แต่ละเฟส โดยจะรับแรงดัน ไฟฟ้าที่ขาเบสจากจุ่มร่วมของ แลคเคอร์รีซิสเตอร์ ส่วนทรานซิสเตอร์ BC337 ทำหน้าที่เป็นตัวเลือกที่จะกระตุ้นเฟสใดของสเตปมอเตอร์ โดยรับสัญญาณควบคุมจากไมโครคอนโทรลเลอร์ ถ้าต้องการกระตุ้นที่เฟสใดก็ให้ค่าลอจิกที่ออกจากขาเอาต์พุทพอร์ทที่เชื่อมเข้ากับขาเบสของ BC337 มีค่าเป็น “0” และเฟสใดที่ไม่ต้องการกระตุ้นก็ให้ค่าลอจิก เท่ากับ “1”

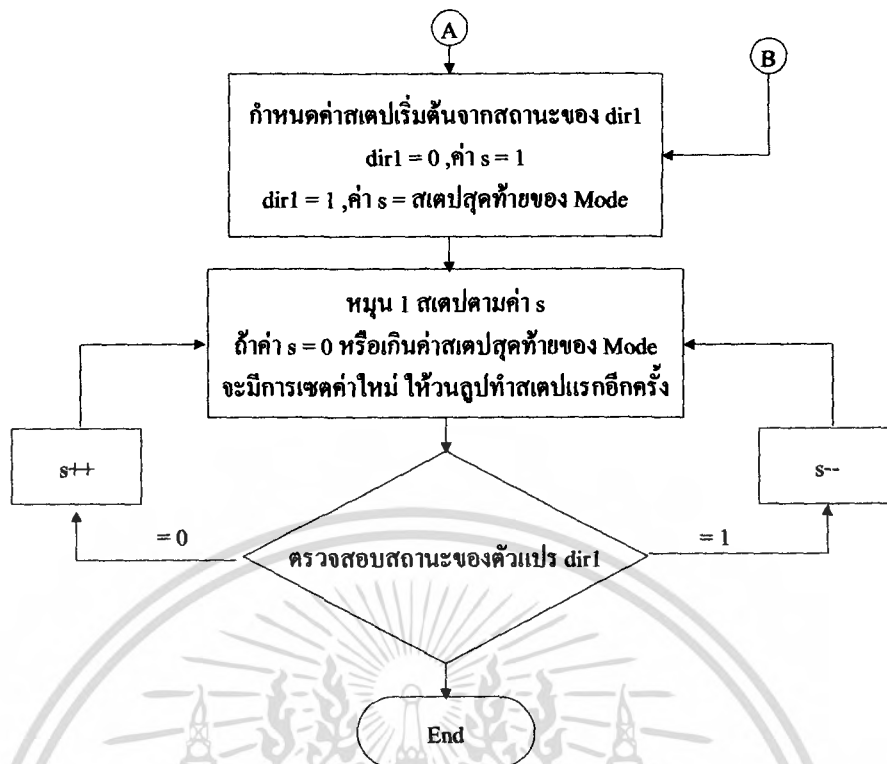
### 3.3 โปรแกรมที่ใช้ในการขับเคลื่อนมอเตอร์

โฟลว์ชาร์ตดังแสดงในรูปที่ 3.4 เป็นโปรแกรมที่ใช้ในการควบคุมการขับเคลื่อนมอเตอร์แบบปกติและแบบไมโครสเตป



รูปที่ 3.4 แสดงโฟลว์ชาร์ตการควบคุมการขับเคลื่อนมอเตอร์แบบไมโครสเตป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 (ต่อ) แสดงโฟลว์ชาร์ตการควบคุมการขับเคลื่อนมอเตอร์แบบไมโครสเตป

### 3.4 สรุป

ระบบควบคุมการขับเคลื่อนมอเตอร์แบบไมโครสเตป โครงสร้างประกอบด้วยส่วนของไมโครคอนโทรลเลอร์ซึ่งเป็นตัวควบคุม ไมโครคอนโทรลเลอร์จะเชื่อมต่อกับคอมพิวเตอร์เพื่อความสะดวกในการแก้ไขข้อมูลและการสร้างสัญญาณควบคุมได้ตามต้องการ ส่วนควบคุมแรงดันวงจรขับ และสแตปมอเตอร์ ในการขับเคลื่อนมอเตอร์แบบไมโครสเตปนั้น ไมโครคอนโทรลเลอร์จะเป็นตัวควบคุมการจ่ายกระแสไฟฟ้าของวงจรขับ เป็นการส่งสัญญาณควบคุมให้กับส่วนควบคุมแรงดันเพื่อควบคุมหรือแบ่งระดับแรงดันไฟฟ้าให้ได้ตามที่ต้องการ เข้าสู่ส่วนของวงจรขับ เพื่อนำสัญญาณที่ได้ขึ้นไปกระตุ้นเฟสของสแตปมอเตอร์แต่ละเฟสให้สแตปมอเตอร์เกิดการเคลื่อนที่เป็นสเตป สแตปมอเตอร์ที่ใช้ทดสอบในโครงการนี้เป็นสแตปมอเตอร์ชนิดไฮบริด ใช้กระแส 1 แอมป์ และใช้แรงดัน 5 โวลต์ มีขนาด 1.8 องศาต่อสเตป หรือจำนวนสเตป 200 สเตปต่อรอบ

## บทที่ 4

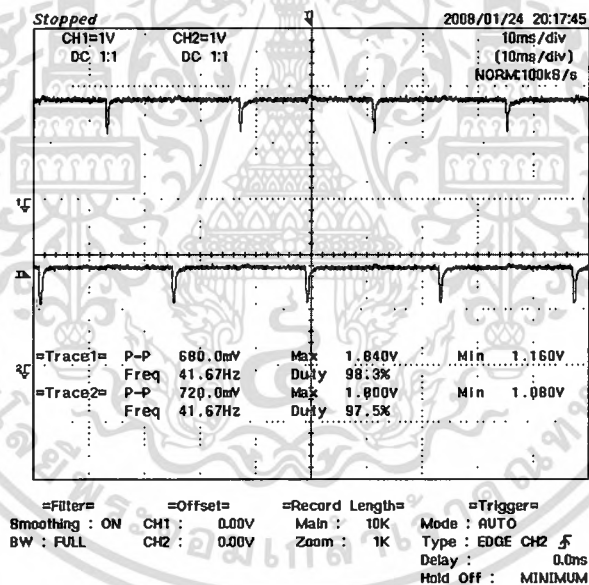
### ผลการทดลอง

#### 4.1 ผลการทดลอง

การทดสอบการควบคุมการขับเคลื่อนมอเตอร์ในโครงการนี้ จะทดสอบการขับเคลื่อนมอเตอร์แบบปกติและการขับเคลื่อนมอเตอร์แบบไมโครสเตป โดยการวัดสัญญาณอินพุทจากจตุรร่วมของ Ladder Resister เปรียบเทียบกับสัญญาณเอาต์พุทที่ทำการวัดจากขาเบสของ Power Transistor (TIP122) ของเฟส A+, A- และ เฟส B+, B- ของสเตรปมอเตอร์

สัญญาณช่องบน : โวลเตจที่วัดจาก Ladder Resister ของเฟส A+, A-

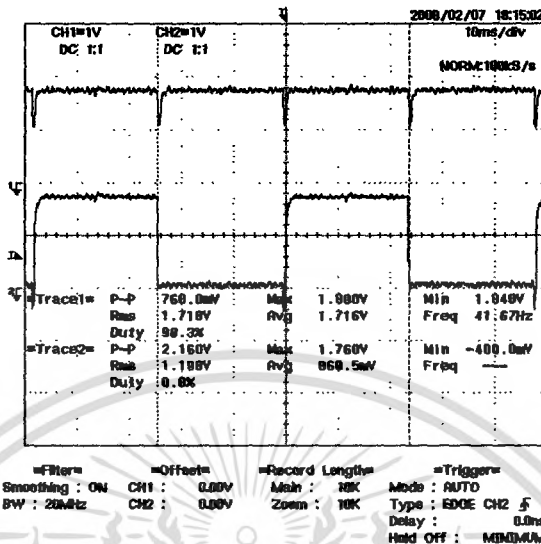
สัญญาณช่องล่าง : โวลเตจที่วัดจาก Ladder Resister ของเฟส B+, B-



รูปที่ 4.1 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณอินพุทของเฟส A+, A- กับสัญญาณอินพุทของเฟส B+, B- โดยใช้การกระตุ้นเฟสแบบพูลสเตป 2 เฟส

สัญญาณช่องบน : โวลเตจที่วัดจาก Ladder Resister ของเฟส A+, A-

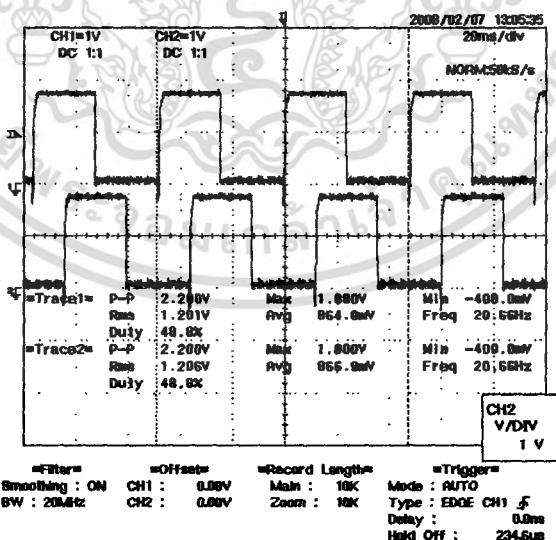
สัญญาณช่องล่าง : โวลเตจที่วัดจากขาเบสของ Power Transistor ของเฟส A+, A-



รูปที่ 4.2 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณอินพุทของเฟส A+, A- กับสัญญาณเอาต์พุทของเฟส A+, A- โดยใช้การกระตุ้นเฟสแบบพูลสเตป 2 เฟส

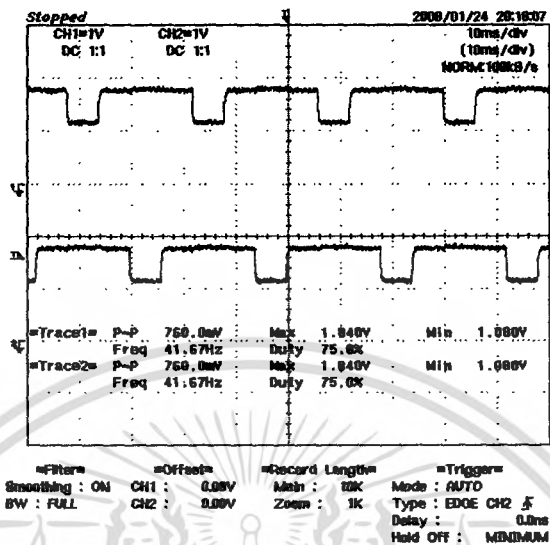
สัญญาณช่องบน : โวลเตจที่วัดจากขาเบสของ Power Transistor ของเฟส A+, A-

สัญญาณช่องล่าง : โวลเตจที่วัดจากขาเบสของ Power Transistor ของเฟส B+,B-



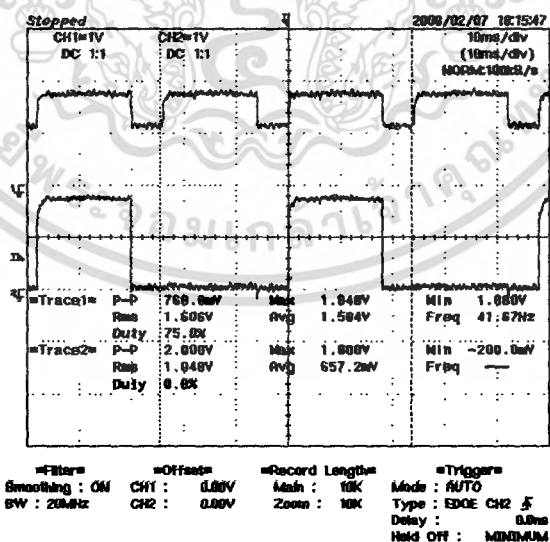
รูปที่ 4.3 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณเอาต์พุทของเฟส A+, A- กับสัญญาณเอาต์พุทของเฟส B+,B- โดยใช้การกระตุ้นเฟสแบบพูลสเตป 2 เฟส

สัญญาณช่องบน : โวลเตจที่วัดจาก Ladder Resister ของเฟส A+, A-  
 สัญญาณช่องล่าง : โวลเตจที่วัดจาก Ladder Resister ของเฟส B+, B-



รูปที่ 4.4 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณอินพุทของเฟส A+, A- กับสัญญาณอินพุทของเฟส B+, B- โดยใช้การกระตุ้นเฟสแบบฮาล์ฟสเตรป

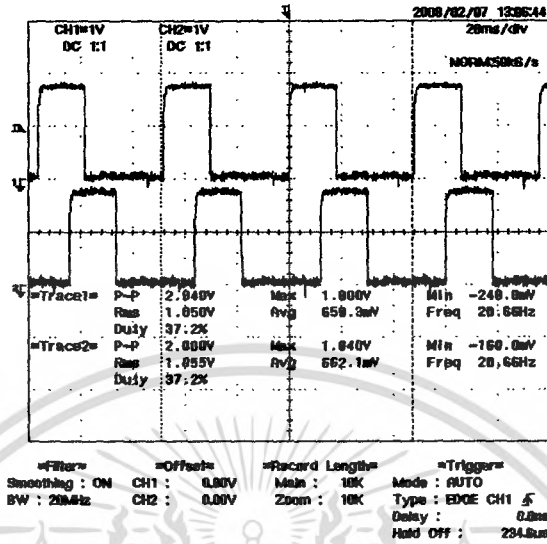
สัญญาณช่องบน : โวลเตจที่วัดจาก Ladder Resister ของเฟส A+, A-  
 สัญญาณช่องล่าง : โวลเตจที่วัดจากขาเบสของ Power Transistor ของเฟส A+, A-



รูปที่ 4.5 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณอินพุทของเฟส A+, A- กับสัญญาณเอาต์พุทของเฟส A+, A- โดยใช้การกระตุ้นเฟสแบบฮาล์ฟสเตรป

สัญญาณช่องบน : โวลเตจที่วัดจากขาเบสของ Power Transistor ของเฟส A+, A-

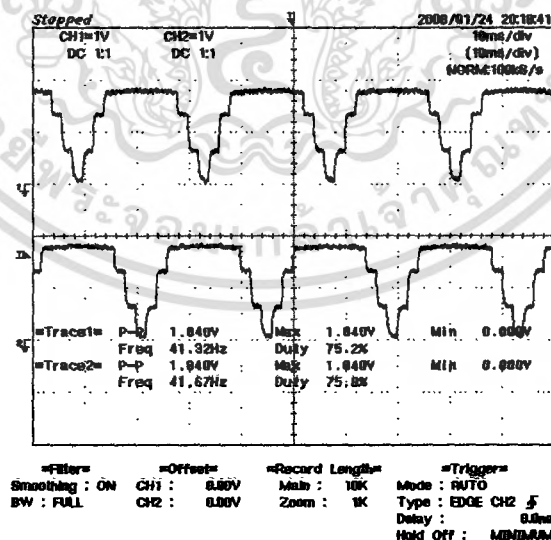
สัญญาณช่องล่าง : โวลเตจที่วัดจากขาเบสของ Power Transistor ของเฟส B+,B-



รูปที่ 4.6 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณเอาต์พุตของเฟส A+, A- กับสัญญาณเอาต์พุตของเฟส B+,B- โดยใช้การกระตุ้นเฟสแบบฮาล์ฟสแตป

สัญญาณช่องบน : โวลเตจที่วัดจาก Ladder Resister ของเฟส A+, A-

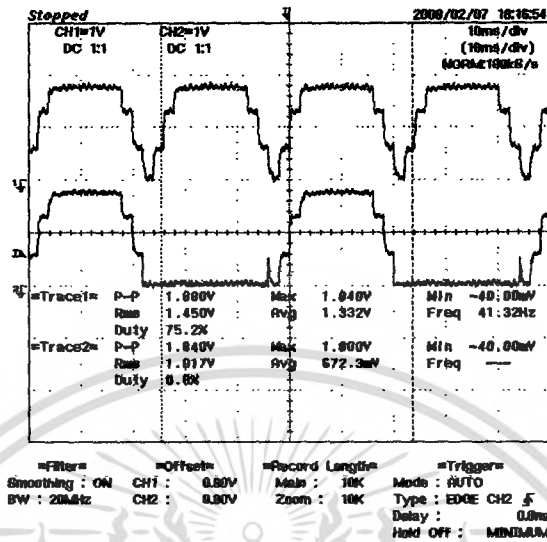
สัญญาณช่องล่าง : โวลเตจที่วัดจาก Ladder Resister ของเฟส B+, B-



รูปที่ 4.7 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณอินพุตของเฟส A+, A- กับสัญญาณอินพุตของเฟส B+, B- โดยขับในโหมด 1200 สเตป

สัญญาณช่องบน : โวลเตจที่วัดจาก Ladder Resister ของเฟส A+, A-

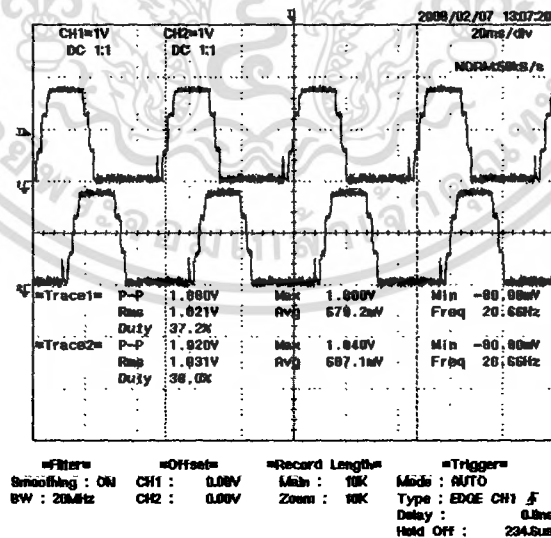
สัญญาณช่องล่าง : โวลเตจที่วัดจากขาเบสของ Power Transistor ของเฟส A+, A-



รูปที่ 4.8 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณอินพุทของเฟส A+, A- กับสัญญาณเอาต์พุทของเฟส A+, A- โดยจับในโหมด 1200 สเตป

สัญญาณช่องบน : โวลเตจที่วัดจากขาเบสของ Power Transistor ของเฟส A+, A-

สัญญาณช่องล่าง : โวลเตจที่วัดจากขาเบสของ Power Transistor ของเฟส B+, B-

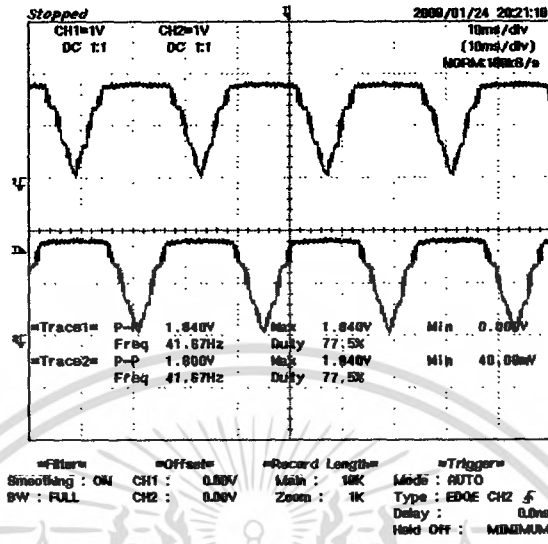


รูปที่ 4.9 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณเอาต์พุทของเฟส A+, A- กับสัญญาณเอาต์พุทของเฟส B+, B- โดยจับในโหมด 1200 สเตป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณช่องบน : โวลเตจที่วัดจาก Ladder Resister ของเฟส A+, A-

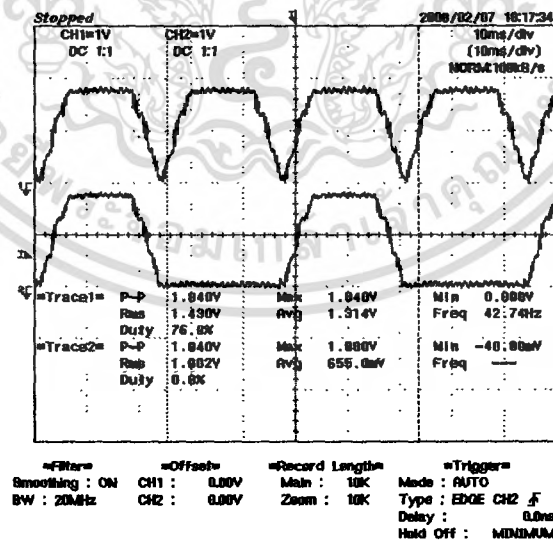
สัญญาณช่องล่าง : โวลเตจที่วัดจาก Ladder Resister ของเฟส B+, B-



รูปที่ 4.10 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณอินพุทของเฟส A+, A- กับสัญญาณอินพุทของเฟส B+, B- โดยจับใน โหมด 3600 สเตป

สัญญาณช่องบน : โวลเตจที่วัดจาก Ladder Resister ของเฟส A+, A-

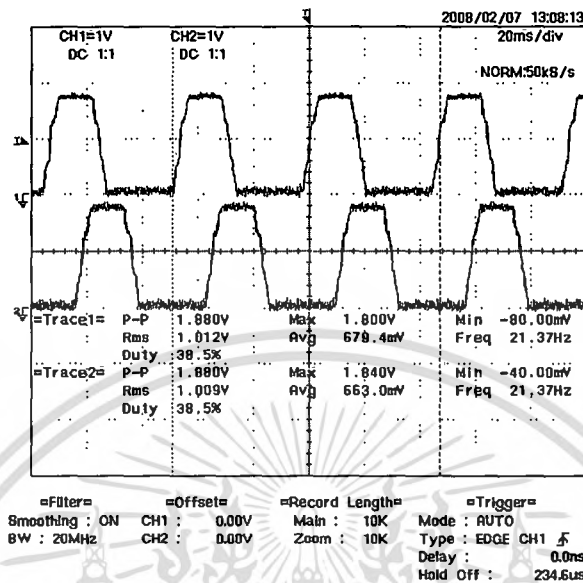
สัญญาณช่องล่าง : โวลเตจที่วัดจากขาเบสของ Power Transistor ของเฟส A+, A-



รูปที่ 4.11 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณอินพุทของเฟส A+, A- กับสัญญาณเอาต์พุทของเฟส A+, A- โดยจับใน โหมด 3600 สเตป

สัญญาณช่องบน : โวลเตจที่วัดจากขาเบสของ Power Transistor ของเฟส A+, A-

สัญญาณช่องล่าง : โวลเตจที่วัดจากขาเบสของ Power Transistor ของเฟส B+, B-



รูปที่ 4.12 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณเอาต์พุตของเฟส A+, A- กับสัญญาณเอาต์พุตของเฟส B+, B- โดยจับใน โหมด 3600 สเตป

#### 4.2 สรุปผลการทดลอง

จากการทดลองการขับเคลื่อนมอเตอร์แบบไมโครสเตป โดยนำสเตปมอเตอร์ที่มีความละเอียดในการหมุน 200 สเตปต่อรอบ มาทำการควบคุมโดยใช้ไมโครคอนโทรลเลอร์ จะสามารถแบ่งลักษณะการหมุนออกเป็น 4 แบบ คือ Full step , Half step , 1200 steps และ 3600 steps ซึ่งจากการทดลองจะพบว่าในการหมุนแบบ Full step ที่มีความละเอียดในการหมุน 1.8 องศาต่อสเตป ( 200 สเตปต่อรอบ ) และแบบ Half step ที่มีความละเอียดในการหมุน 0.9 องศาต่อสเตป ( 400 สเตปต่อรอบ ) จะมีการจ่ายสัญญาณเพื่อควบคุมการกระตุ้นเฟสแต่ละเฟสของสเตปมอเตอร์เป็นลักษณะสัญญาณสี่เหลี่ยม ( สัญญาณพัลส์ ) ซึ่งถ้าทำการเปรียบเทียบกับขับเคลื่อนมอเตอร์แบบไมโครสเตป 1200 steps ที่มีความละเอียดในการหมุน 0.3 องศาต่อสเตป ( 1200 สเตปต่อรอบ ) และแบบ 3600 steps ที่มีความละเอียดในการหมุน 0.1 องศาต่อสเตป ( 3600 สเตปต่อรอบ ) จะมีการจ่ายสัญญาณควบคุมในลักษณะคล้ายกับสัญญาณระดับขั้นบันได ที่เกิดจากการแบ่งระดับแรงดันไฟฟ้าที่ใช้ในการกระตุ้นเฟสแต่ละเฟสของสเตปมอเตอร์ จะพบว่าลักษณะการหมุนในแบบไมโครสเตปจะมีลักษณะการหมุนที่ราบเรียบกว่าการขับเคลื่อนมอเตอร์แบบธรรมดา ซึ่งถ้าเปรียบเทียบระหว่างการหมุนแบบ Full step กับแบบ 3600 steps แล้ว จะได้ว่าในการหมุนแบบ Full step จำนวน 1 สเตป จะเท่ากับการหมุนในแบบ 3600 steps จำนวน 18 สเตป ดังนั้นการขับเคลื่อนมอเตอร์แบบไมโครสเตป

จึงสามารถกำหนดตำแหน่งที่ต้องการ ได้ละเอียดกว่าการขับเคลื่อนมอเตอร์ในแบบธรรมดา ด้านการคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

# บทวิจารณ์และสรุป

### 5.1 สรุปผลการดำเนินงาน

ปริญญานิพนธ์ฉบับนี้ได้นำเสนอไมโครสเตปมอเตอร์ ซึ่งระบบควบคุมประกอบด้วย ไมโครคอนโทรลเลอร์ ส่วนควบคุมแรงดัน วงจรขับ และสเตปมอเตอร์ ในการขับสเตปมอเตอร์แบบไมโครสเตปนั้น ไมโครคอนโทรลเลอร์จะเป็นตัวควบคุมการจ่ายกระแสไฟฟ้าของวงจรขับ เป็นการส่งสัญญาณควบคุมให้กับส่วนควบคุมแรงดันเพื่อควบคุมหรือแบ่งระดับแรงดันไฟฟ้าให้ได้ตามที่ต้องการ เข้าสู่ส่วนของวงจรขับ เพื่อนำสัญญาณที่ได้นั้น ไปกระตุ้นเฟสของสเตปมอเตอร์แต่ละเฟส ให้สเตปมอเตอร์เกิดการเคลื่อนที่เป็นสเตป ส่วนของไมโครคอนโทรลเลอร์ ผู้จัดทำได้ใช้ PIC16F876A หน่วยความจำภายใน 8 กิโลไบต์ ส่วนของการควบคุมแรงดันใช้วิธีพัลส์วิดท์มอดูเลชันและเขียนโปรแกรมควบคุม ส่วนของวงจรขับได้ออกแบบและสร้างให้สามารถขับสเตปมอเตอร์ได้ทั้งแบบพูล แบบฮาล์ฟสเตปและแบบไมโครสเตป จากนั้นได้ทำการทดสอบและได้ผลการทดลองดังที่ได้กล่าวมาแล้ว สุดท้ายได้ทำการทดสอบร่วมกับวงจรชุดทดสอบการส่งสัญญาณพัลส์ และใช้สเกลวัดมุมสเตปเพื่อแสดงตำแหน่งและความราบเรียบในการหมุนของสเตปมอเตอร์

### 5.2 ปัญหาที่พบและแนวทางการแก้ไข

จากการศึกษาและทำโครงการในการทดลองการขับสเตปมอเตอร์แบบไมโครสเตปนั้น ในช่วงแรกประสบปัญหาในเรื่องของการรวบรวมข้อมูล และการทำความเข้าใจถึงวิธีที่จะใช้ในการขับสเตปมอเตอร์ให้สามารถหมุนได้โดยมีจำนวนสเตปต่อรอบที่มีความละเอียดมากขึ้น นอกจากนี้ยังมีปัญหาในส่วนของการเขียนโปรแกรมเพื่อควบคุมการขับสเตปมอเตอร์ให้สามารถหมุนได้ทั้งในแบบพูลสเตป, ฮาล์ฟสเตป, 1200 สเตปและ 3600 สเตป ดังนั้นในส่วนของการเลือกตัวไมโครคอนโทรลเลอร์ที่จะนำมาใช้ในงานจึงต้องมีการเปลี่ยนแปลง เพื่อให้มีความเหมาะสมที่จะนำมาใช้ในการเขียนโปรแกรม โดยตัวไมโครคอนโทรลเลอร์ที่ได้นำมาใช้ในครั้งแรกจะเป็นไมโครคอนโทรลเลอร์เบอร์ PIC16F84A แต่เนื่องจากหน่วยความจำภายในของไมโครคอนโทรลเลอร์เบอร์ PIC16F84A มีเพียง 2 กิโลไบต์ จึงทำให้การเขียนโปรแกรมเพื่อใช้ในการขับสเตปมอเตอร์ในโหมดของไมโครสเตปมอเตอร์มีหน่วยความจำไม่เพียงพอที่จะนำไปใช้งาน จึงได้มีการเปลี่ยนตัวไมโครคอนโทรลเลอร์ใหม่ เพื่อให้สามารถรองรับในส่วนของการเขียนโปรแกรมได้เพียงพอจึงได้มีการนำตัวไมโครคอนโทรลเลอร์เบอร์ PIC16F628A และ PIC16F876A มาใช้ในการทดลองตามลำดับ ซึ่งสุดท้ายได้เลือกตัว PIC16F876A ที่มีหน่วยความจำภายใน 8 กิโลไบต์ มาเป็น

ตัวไมโครคอนโทรลเลอร์ที่ใช้ในการส่งสัญญาณควบคุมสเตปมอเตอร์ในโหมดของไมโครสเตปต่อไป

### 5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา

ความเที่ยงตรงของขนาดไมโครสเตปที่ได้ขึ้นอยู่กับปัจจัยหลายประการด้วยกัน ได้แก่ ชนิดและโครงสร้างของสเตปมอเตอร์ที่ใช้ ประสิทธิภาพของวงจรที่ใช้ควบคุมระดับกระแสในแต่ละเฟสของสเตปมอเตอร์ รวมทั้งลักษณะการควบคุมการขับเคลื่อนสเตปมอเตอร์ ซึ่งในโครงการนี้จะอยู่ในลักษณะลูปเปิด (Open-loop) ซึ่งสามารถพัฒนาต่อไปได้โดยควบคุมในลักษณะลูปปิด (Close-loop) เพื่อนำมาใช้ในการตรวจสอบตำแหน่งที่สเตปมอเตอร์หมุนไป ซึ่งจะทำให้มีความเที่ยงตรงของขนาดไมโครสเตปสูงขึ้น อีกทั้งการควบคุมการขับเคลื่อนสเตปมอเตอร์ให้ทำงานแบบไมโครสเตปนั้น จะมีต้นทุนในการสร้างวงจรควบคุมของระบบจะสูงกว่าวงจรควบคุมการขับเคลื่อนสเตปมอเตอร์แบบปกติ ดังนั้นหากสามารถออกแบบวงจรควบคุมของไมโครสเตปมอเตอร์ให้มีประสิทธิภาพโดยใช้ต้นทุนต่ำได้จะเป็นประโยชน์อย่างมากต่อการนำไปประยุกต์ใช้งานต่อไป

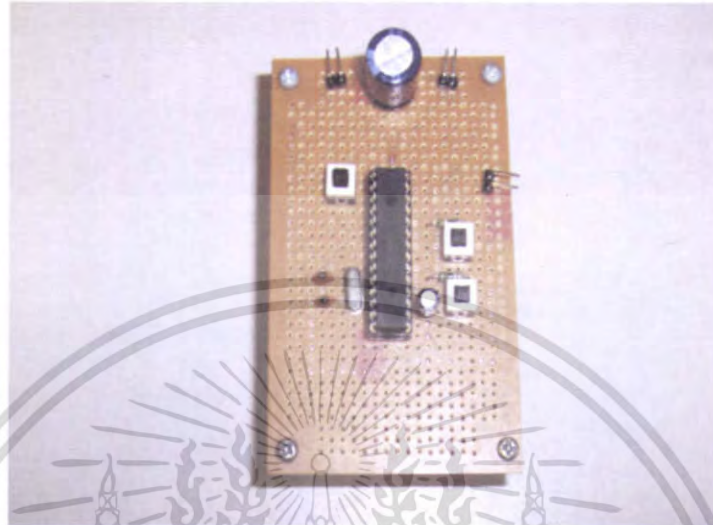


## เอกสารอ้างอิง

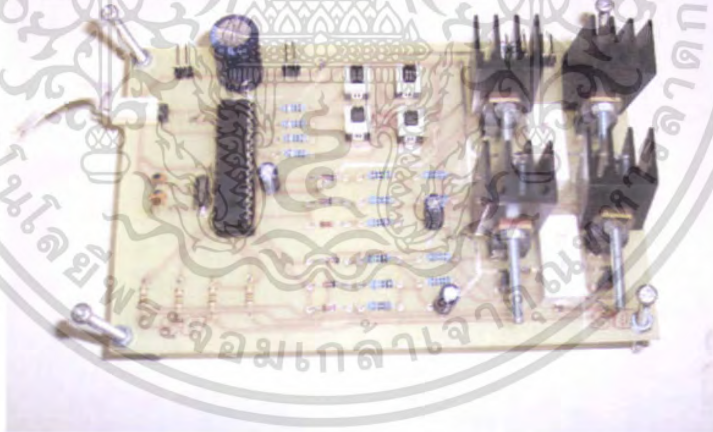
- [1] พีรศักดิ์ ยอดทอง, วัฒนา อุณหวนิชย์, สุรัชชัย แก้วใจจง.  
“การขับเคลื่อนมอเตอร์ในโหมดไมโครสเตปแบบวงจรถูกเปิด.” ปรินญาณีพนธ์วิศวกรรม-  
ศาสตรบัณฑิต สาขาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระ-  
จอมเกล้าเจ้าคุณทหารลาดกระบัง. 2546.
- [2] ประจัน พลึงสันติกุล. **เรียนรู้และใช้งาน CCS C คอมพิวเตอร์ เขียนโปรแกรมภาษา C  
ควบคุมไมโครคอนโทรลเลอร์ PIC**. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : อินโนเวตีฟ เอ็กเพอริ  
เมนต์. 2547.
- [3] Muhammaed F.Rahman, Aun-Neow Poo. “An Application Oriented Test Procedure for  
Designing Microstepping Step Motor Controller.” **IEEE Transaction on industrial  
Electronics**, vol. 35, no. 4 , November 1988.
- [4] Muhammaed F.Rahman, Aun-Neow Poo. “Approaches to Design of Ministepping Step  
Motor Controllers and Their Accuracy Concideration.” **IEEE Transaction on industrial  
Electronics**, vol. IE-32, August, 1985.
- [5] P.P. Acarnley, “Stepping motors : a guild to modern theory and practice.” **IEEE control  
Engineering Service 19**, Peregrius Ltd., 1985.
- [5] P.P. Acarnley, “Stepping motors : a guild to modern theory and practice.” **IEEE control  
Engineering Service 19**, Peregrius Ltd., 1985.
- [6] James Newton. “Linistepper.” [Online]. Available :  
<http://www.piclist.com/techref/io/stepper/linistep/index.htm>

## ภาคผนวก ก

## แสดงรูปส่วนประกอบของวงจร

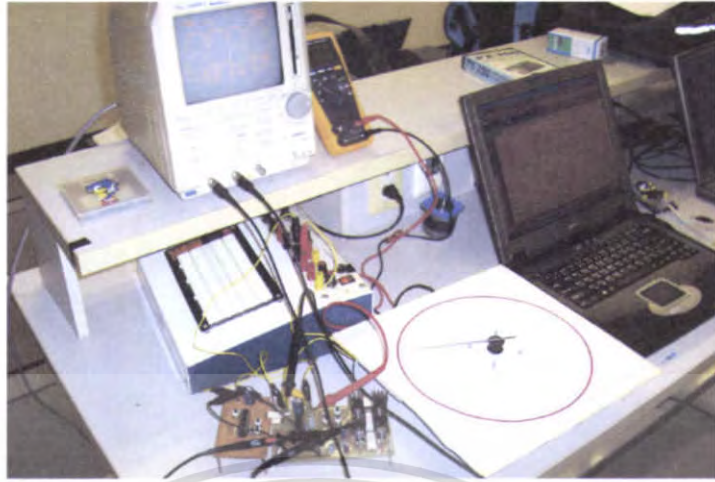


รูปที่ ก.1 แสดงวงจรชุดทดสอบการส่งสัญญาณพัลส์

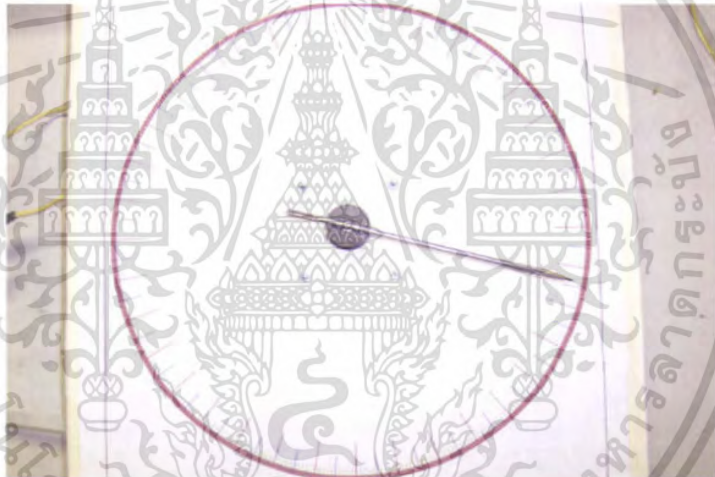


รูปที่ ก.2 แสดงวงจรขับสเตปมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.3 แสดงระบบควบคุมและการทดสอบการขับเคลื่อนมอเตอร์แบบไมโครสเตป



รูปที่ ก.4 แสดงสเตปมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข  
โปรแกรมที่ใช้ในการทดลอง

โปรแกรมสำหรับขับเคลื่อนมอเตอร์

```
#include <test_int_EXT.h>
```

```
#byte port_b = 0x06
```

```
#byte port_c = 0x07
```

```
#define t 12
```

```
#define t1 6000
```

```
#define t2 2000
```

```
#define time 255
```

```
#define time1 150
```

```
int x,y,n,s;
```

```
short dir0,dir1,step;
```

```
void start_motor();
```

```
void full_step();
```

```
void half_step();
```

```
void m1200();
```

```
void m3600();
```

```

#int_EXT
void EXT_isr()
{
    if(step == 1)
    {
        switch(dir1)
        {
            case 0: s++;
                break;
            case 1: s--;
                break;
        }
    }
}
#int_RB
void RB_isr()
{
    x = input_b0;
    dir1 = input(PIN_B5);
    step = input(PIN_B4);
    if(step == 1)
    {
        delay_ms(100);
        output_high(PIN_A0); delay_ms(10);
        output_low(PIN_A0); delay_ms(10);
    }
}

```

```

void main()
{
    port_b_pullups(TRUE);
    setup_adc_ports(NO_ANALOGS);
    setup_adc(ADC_OFF);
    setup_spi(FALSE);
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    enable_interrupts(INT_RB);
    enable_interrupts(INT_EXT);
    enable_interrupts(GLOBAL);

    set_tris_b(0xFF);
    set_tris_c(0x00);

    port_b = 0x0E; delay_ms(1);
    output_low(PIN_A0); delay_ms(1);
    x = 0x00; y = 0; n = 0; s = 0;

    x = input_b();
    y = x;

    dir0 = input(PIN_B5);
    dir1 = input(PIN_B5);
    step = input(PIN_B4);

```

```

start_motor();

while(1)
{
    if(input(PIN_B7)==0 & input(PIN_B6)==0)
    {
        full_step();
    }

    else if(input(PIN_B7)==0 & input(PIN_B6)==1)
    {
        half_step();
    }

    else if(input(PIN_B7)==1 & input(PIN_B6)==0)
    {
        m1200;
    }

    else if(input(PIN_B7)==1 & input(PIN_B6)==1)
    {
        m3600;
    }

    y = x;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void start_motor()
{
    switch(dir1)
    {
        case 0: s = 1;
            break;
        case 1: s = 4;
            break;
    }
    while(x == y)
    {
        switch(s)
        {
            case 0: s = 5;
                break;
            case 1: port_c = 0xF5; delay_ms(t); //1111 0101
                break;
            case 2: port_c = 0xF9; delay_ms(t); //1111 1001
                break;
            case 3: port_c = 0xFA; delay_ms(t); //1111 1010
                break;
            case 4: port_c = 0xF6; delay_ms(t); //1111 0110
                break;
            case 5: s = 0;
                break;
        }
    switch(dir1)

```

```

{
  case 0: s++;
    break;
  case 1: s--;
    break;
}
n++;
if(n == 19)
{
  n = 0;
  break;
}
}
}
void full_step()
{
  if((dir0 != dir1) && (step == 0))
  {
    start_motor();
    dir0 = input(PIN_B5);
  }
  if(step == 0)
  {
    switch(dir1)
    {
      case 0: s = 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break;
    case 1: s = 4;
        break;
    }
}
else
{
    switch(s)
    {
        case 0: s = 4;
            break;
        case 5: s = 1;
            break;
    }
}
while(x == y)
{
    switch(s)
    {
        case 0: s = 5;
            break;
        case 1: port_c = 0xF5;    delay_ms(t); //1111 0101
            break;
        case 2: port_c = 0xF9;    delay_ms(t); //1111 1001
            break;
        case 3: port_c = 0xFA;    delay_ms(t); //1111 1010
            break;
    }
}

```

```

case 4: port_c = 0xF6; delay_ms(t); //1111 0110
        break;
case 5: s = 0;
        break;
    }
if(step == 1)
    {
        output_high(PIN_A0); delay_ms(10);
        output_low(PIN_A0); delay_ms(10);
        break;
    }
switch(dir1)
    {
        case 0: s++;
            break;
        case 1: s--;
            break;
    }
}
void half_step()
{
    if((dir0 != dir1) && (step == 0))
    {
        start_motor();
        dir0 = input(PIN_B5);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(step == 0)
{
    switch(dir1)
    {
        case 0: s = 1;
            break;
        case 1: s = 8;
            break;
    }
}
else
{
    switch(s)
    {
        case 0: s = 8;
            break;
        case 9: s = 1;
            break;
    }
}
while(x == y)
{
    switch(s)
    {
        case 0: s = 9;
            break;
        case 1: port_c = 0xF7; delay_us(t1); //1111 0111

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break;
    case 2: port_c = 0xF5; delay_us(t1); //1111 0101
        break;
    case 3: port_c = 0xFD; delay_us(t1); //1111 1101
        break;
    case 4: port_c = 0xF9; delay_us(t1); //1111 1001
        break;
    case 5: port_c = 0xFB; delay_us(t1); //1111 1011
        break;
    case 6: port_c = 0xFA; delay_us(t1); //1111 1010
        break;
    case 7: port_c = 0xFE; delay_us(t1); //1111 1110
        break;
    case 8: port_c = 0xF6; delay_us(t1); //1111 0110
        break;
    case 9: s = 0;
        break;
    }
    if(step == 1)
    {
        output_high(PIN_A0); delay_ms(10);
        output_low(PIN_A0); delay_ms(10);
        break;
    }
    switch(dir1)
    {
    case 0: s++;

```

```

        break;
    case 1: s--;
        break;
    }
}
}

```

```
void m1200()
```

```

{
    if((dir0 != dir1) && (step == 0))
    {
        start_motor();
        dir0 = input(PIN_B5);
    }
    if(step == 0)
    {
        switch(dir1)
        {
            case 0: s = 1;
                break;
            case 1: s = 24;
                break;
        }
    }
}
else
{
    switch(s)

```

```

{
    case 0: s = 24;
        break;
    case 25: s = 1;
        break;
}
}
while(x == y)
{
    switch(s)
    {
        case 0: s = 25;
            break;
        case 1: port_c = 0xC5;    delay_us(t2);
            break;
        case 2: port_c = 0xD5;    delay_us(t2);
            break;
        case 3: port_c = 0xE5;    delay_us(t2);
            break;
        case 4: port_c = 0xF5;    delay_us(t2);
            break;
        case 5: port_c = 0xB5;    delay_us(t2);
            break;
        case 6: port_c = 0x75;    delay_us(t2);
            break;
        case 7: port_c = 0x39;    delay_us(t2);
            break;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 8: port_c = 0x79; delay_us(t2);
break;
case 9: port_c = 0xB9; delay_us(t2);
break;
case 10: port_c = 0xF9; delay_us(t2);
break;
case 11: port_c = 0xE9; delay_us(t2);
break;
case 12: port_c = 0xD9; delay_us(t2);
break;
case 13: port_c = 0xCA; delay_us(t2);
break;
case 14: port_c = 0xDA; delay_us(t2);
break;
case 15: port_c = 0xEA; delay_us(t2);
break;
case 16: port_c = 0xFA; delay_us(t2);
break;
case 17: port_c = 0xBA; delay_us(t2);
break;
case 18: port_c = 0x7A; delay_us(t2);
break;
case 19: port_c = 0x36; delay_us(t2);
break;
case 20: port_c = 0x76; delay_us(t2);
break;
case 21: port_c = 0xB6; delay_us(t2);

```

```

        break;
    case 22: port_c = 0xF6; delay_us(t2);
        break;
    case 23: port_c = 0xE6; delay_us(t2);
        break;
    case 24: port_c = 0xD6; delay_us(t2);
        break;
    case 25: s = 0;
        break;
    }
    if(step == 1)
    {
        output_high(PIN_A0); delay_ms(10);
        output_low(PIN_A0); delay_ms(10);
        break;
    }
    switch(dir1)
    {
    case 0: s++;
        break;
    case 1: s--;
        break;
    }
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void m3600()
{
    if((dir0 != dir1) && (step == 0))
    {
        start_motor();
        dir0 = input(PIN_B5);
    }
    if(step == 0)
    {
        switch(dir1)
        {
            case 0: s = 1;
                    break;
            case 1: s = 72;
                    break;
        }
        else
        {
            switch(s)
            {
                case 0: s = 72;
                        break;
                case 73: s = 1;
                        break;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(x == y)
{
    switch(s)
    {
        case 0: s = 73;
            break;

        case 1: port_c = 0xC5; delay_us(time); port_c = 0xC5;
delay_us(time1); port_c = 0xC5; delay_us(time); //0
            break;

        case 2: port_c = 0xC5; delay_us(time); port_c = 0xD5;
delay_us(time1); port_c = 0xC5; delay_us(time); //1
            break;

        case 3: port_c = 0xD5; delay_us(time); port_c = 0xC5;
delay_us(time1); port_c = 0xD5; delay_us(time); //2
            break;

        case 4: port_c = 0xD5; delay_us(time); port_c = 0xD5;
delay_us(time1); port_c = 0xD5; delay_us(time); //3
            break;

        case 5: port_c = 0xD5; delay_us(time); port_c = 0xE5;
delay_us(time1); port_c = 0xD5; delay_us(time); //4
            break;

        case 6: port_c = 0xE5; delay_us(time); port_c = 0xD5;
delay_us(time1); port_c = 0xE5; delay_us(time); //5
            break;

        case 7: port_c = 0xE5; delay_us(time); port_c = 0xE5;
delay_us(time1); port_c = 0xE5; delay_us(time); //6
            break;
    }
}

```

```

case 8: port_c = 0xE5; delay_us(time); port_c = 0xF5;
delay_us(time1); port_c = 0xE5; delay_us(time); //7
break;

case 9: port_c = 0xF5; delay_us(time); port_c = 0xE5;
delay_us(time1); port_c = 0xF5; delay_us(time); //8
break;

case 10: port_c = 0xF5; delay_us(time); port_c = 0xF5;
delay_us(time1); port_c = 0xF5; delay_us(time); //9
break;

case 11: port_c = 0xF5; delay_us(time); port_c = 0xB5;
delay_us(time1); port_c = 0xF5; delay_us(time); //10
break;

case 12: port_c = 0xB5; delay_us(time); port_c = 0xF5;
delay_us(time1); port_c = 0xB5; delay_us(time); //11
break;

case 13: port_c = 0xB5; delay_us(time); port_c = 0xB5;
delay_us(time1); port_c = 0xB5; delay_us(time); //12
break;

case 14: port_c = 0xB5; delay_us(time); port_c = 0x75;
delay_us(time1); port_c = 0xB5; delay_us(time); //13
break;

case 15: port_c = 0x75; delay_us(time); port_c = 0xB5;
delay_us(time1); port_c = 0x75; delay_us(time); //14
break;

case 16: port_c = 0x75; delay_us(time); port_c = 0x75;
delay_us(time1); port_c = 0x75; delay_us(time); //15
break;

```

```

case 17: port_c = 0x75; delay_us(time); port_c = 0x35;
delay_us(time1); port_c = 0x75; delay_us(time); //16
break;

case 18: port_c = 0x35; delay_us(time); port_c = 0x75;
delay_us(time1); port_c = 0x35; delay_us(time); //17
break;

case 19: port_c = 0x39; delay_us(time); port_c = 0x39;
delay_us(time1); port_c = 0x39; delay_us(time); //18
break;

case 20: port_c = 0x39; delay_us(time); port_c = 0x79;
delay_us(time1); port_c = 0x39; delay_us(time); //19
break;

case 21: port_c = 0x79; delay_us(time); port_c = 0x39;
delay_us(time1); port_c = 0x79; delay_us(time); //20
break;

case 22: port_c = 0x79; delay_us(time); port_c = 0x79;
delay_us(time1); port_c = 0x79; delay_us(time); //21
break;

case 23: port_c = 0x79; delay_us(time); port_c = 0xB9;
delay_us(time1); port_c = 0x79; delay_us(time); //22
break;

case 24: port_c = 0xB9; delay_us(time); port_c = 0x79;
delay_us(time1); port_c = 0xB9; delay_us(time); //23
break;

case 25: port_c = 0xB9; delay_us(time); port_c = 0xB9;
delay_us(time1); port_c = 0xB9; delay_us(time); //24
break;

```

```

    case 26: port_c = 0xB9; delay_us(time); port_c = 0xF9;
delay_us(time1); port_c = 0xB9; delay_us(time); //25
    break;

    case 27: port_c = 0xF9; delay_us(time); port_c = 0xB9;
delay_us(time1); port_c = 0xF9; delay_us(time); //26
    break;

    case 28: port_c = 0xF9; delay_us(time); port_c = 0xF9;
delay_us(time1); port_c = 0xF9; delay_us(time); //27
    break;

    case 29: port_c = 0xF9; delay_us(time); port_c = 0xE9;
delay_us(time1); port_c = 0xF9; delay_us(time); //28
    break;

    case 30: port_c = 0xE9; delay_us(time); port_c = 0xF9;
delay_us(time1); port_c = 0xE9; delay_us(time); //29
    break;

    case 31: port_c = 0xE9; delay_us(time); port_c = 0xE9;
delay_us(time1); port_c = 0xE9; delay_us(time); //30
    break;

    case 32: port_c = 0xE9; delay_us(time); port_c = 0xD9;
delay_us(time1); port_c = 0xE9; delay_us(time); //31
    break;

    case 33: port_c = 0xD9; delay_us(time); port_c = 0xE9;
delay_us(time1); port_c = 0xD9; delay_us(time); //32
    break;

    case 34: port_c = 0xD9; delay_us(time); port_c = 0xD9;
delay_us(time1); port_c = 0xD9; delay_us(time); //33
    break;

```

```

case 35: port_c = 0xD9; delay_us(time); port_c = 0xC9;
delay_us(time1); port_c = 0xD9; delay_us(time); //34
break;

case 36: port_c = 0xC9; delay_us(time); port_c = 0xD9;
delay_us(time1); port_c = 0xC9; delay_us(time); //35
break;

case 37: port_c = 0xCA; delay_us(time); port_c = 0xCA;
delay_us(time1); port_c = 0xCA; delay_us(time); //36
break;

case 38: port_c = 0xCA; delay_us(time); port_c = 0xDA;
delay_us(time1); port_c = 0xCA; delay_us(time); //37
break;

case 39: port_c = 0xDA; delay_us(time); port_c = 0xCA;
delay_us(time1); port_c = 0xDA; delay_us(time); //38
break;

case 40: port_c = 0xDA; delay_us(time); port_c = 0xDA;
delay_us(time1); port_c = 0xDA; delay_us(time); //39
break;

case 41: port_c = 0xDA; delay_us(time); port_c = 0xEA;
delay_us(time1); port_c = 0xDA; delay_us(time); //40
break;

case 42: port_c = 0xEA; delay_us(time); port_c = 0xDA;
delay_us(time1); port_c = 0xEA; delay_us(time); //41
break;

case 43: port_c = 0xEA; delay_us(time); port_c = 0xEA;
delay_us(time1); port_c = 0xEA; delay_us(time); //42
break;

```

```

case 44: port_c = 0xEA; delay_us(time); port_c = 0xFA;
delay_us(time1); port_c = 0xEA; delay_us(time); //43
break;

case 45: port_c = 0xFA; delay_us(time); port_c = 0xEA;
delay_us(time1); port_c = 0xFA; delay_us(time); //44
break;

case 46: port_c = 0xFA; delay_us(time); port_c = 0xFA;
delay_us(time1); port_c = 0xFA; delay_us(time); //45
break;

case 47: port_c = 0xFA; delay_us(time); port_c = 0xBA;
delay_us(time1); port_c = 0xFA; delay_us(time); //46
break;

case 48: port_c = 0xBA; delay_us(time); port_c = 0xFA;
delay_us(time1); port_c = 0xBA; delay_us(time); //47
break;

case 49: port_c = 0xBA; delay_us(time); port_c = 0xBA;
delay_us(time1); port_c = 0xBA; delay_us(time); //48
break;

case 50: port_c = 0xBA; delay_us(time); port_c = 0x7A;
delay_us(time1); port_c = 0xBA; delay_us(time); //49
break;

case 51: port_c = 0x7A; delay_us(time); port_c = 0xBA;
delay_us(time1); port_c = 0x7A; delay_us(time); //50
break;

case 52: port_c = 0x7A; delay_us(time); port_c = 0x7A;
delay_us(time1); port_c = 0x7A; delay_us(time); //51
break;

```

```

case 53: port_c = 0x7A; delay_us(time); port_c = 0x3A;
delay_us(time1); port_c = 0x7A; delay_us(time); //52
break;

case 54: port_c = 0x3A; delay_us(time); port_c = 0x7A;
delay_us(time1); port_c = 0x3A; delay_us(time); //53
break;

case 55: port_c = 0x36; delay_us(time); port_c = 0x36;
delay_us(time1); port_c = 0x36; delay_us(time); //54
break;

case 56: port_c = 0x36; delay_us(time); port_c = 0x76;
delay_us(time1); port_c = 0x36; delay_us(time); //55
break;

case 57: port_c = 0x76; delay_us(time); port_c = 0x36;
delay_us(time1); port_c = 0x76; delay_us(time); //56
break;

case 58: port_c = 0x76; delay_us(time); port_c = 0x76;
delay_us(time1); port_c = 0x76; delay_us(time); //57
break;

case 59: port_c = 0x76; delay_us(time); port_c = 0xB6;
delay_us(time1); port_c = 0x76; delay_us(time); //58
break;

case 60: port_c = 0xB6; delay_us(time); port_c = 0x76;
delay_us(time1); port_c = 0xB6; delay_us(time); //59
break;

case 61: port_c = 0xB6; delay_us(time); port_c = 0xB6;
delay_us(time1); port_c = 0xB6; delay_us(time); //60
break;

```

```

    case 62: port_c = 0xB6; delay_us(time); port_c = 0xF6;
delay_us(time1); port_c = 0xB6; delay_us(time); //61
    break;

    case 63: port_c = 0xF6; delay_us(time); port_c = 0xB6;
delay_us(time1); port_c = 0xF6; delay_us(time); //62
    break;

    case 64: port_c = 0xF6; delay_us(time); port_c = 0xF6;
delay_us(time1); port_c = 0xF6; delay_us(time); //63
    break;

    case 65: port_c = 0xF6; delay_us(time); port_c = 0xE6;
delay_us(time1); port_c = 0xF6; delay_us(time); //64
    break;

    case 66: port_c = 0xE6; delay_us(time); port_c = 0xF6;
delay_us(time1); port_c = 0xE6; delay_us(time); //65
    break;

    case 67: port_c = 0xE6; delay_us(time); port_c = 0xE6;
delay_us(time1); port_c = 0xE6; delay_us(time); //66
    break;

    case 68: port_c = 0xE6; delay_us(time); port_c = 0xD6;
delay_us(time1); port_c = 0xE6; delay_us(time); //67
    break;

    case 69: port_c = 0xD6; delay_us(time); port_c = 0xE6;
delay_us(time1); port_c = 0xD6; delay_us(time); //68
    break;

    case 70: port_c = 0xD6; delay_us(time); port_c = 0xD6;
delay_us(time1); port_c = 0xD6; delay_us(time); //69
    break;

```

```

    case 71: port_c = 0xD6; delay_us(time); port_c = 0xC6;
delay_us(time1); port_c = 0xD6; delay_us(time); //70
        break;

    case 72: port_c = 0xC6; delay_us(time); port_c = 0xD6;
delay_us(time1); port_c = 0xC6; delay_us(time); //71
        break;

    case 73: s = 0;
        break;
}
if(step == 1)
{
    output_high(PIN_A0); delay_ms(10);
    output_low(PIN_A0); delay_ms(10);
    break;
}
switch(dir1)
{
    case 0: s++;
        break;
    case 1: s--;
        break;
}
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โปรแกรมสำหรับขับเคลื่อนมอเตอร์ร่วมกับชุดทดสอบการส่งสัญญาณพัลส์

```

#include "PIC_send_pulse_full_option.h"

#define port_b = 0x06
#define port_c = 0x07

int16 n;
void step();

#ifdef EXT
void EXT_isr()
{
    if(input(PIN_B3)==0 & input(PIN_B2)==0)
    {
        if(n < 200)
        {
            step();
        }
    }
    else if(input(PIN_B3)==0 & input(PIN_B2)==1)
    {
        if(n < 400)
        {
            step();
        }
    }
}
#endif

```

```

else if(input(PIN_B3)==1 & input(PIN_B2)==0)
{
    if(n < 1200)
    {
        step();
    }
}
else if(input(PIN_B3)==1 & input(PIN_B2)==1)
{
    if(n < 3600)
    {
        step();
    }
}
}

void main()
{
    port_b_pullups(TRUE);
    setup_adc_ports(NO_ANALOGS);
    setup_adc(ADC_OFF);
    setup_spi(FALSE);
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);
}

```

```
enable_interrupts(INT_EXT);
enable_interrupts(GLOBAL);

set_tris_b(0x0D);

n = 0;

while(1)
{
    port_b = 0xF0; delay_ms(5);
}

void step0
{
    output_low(PIN_B1); delay_ms(10);
    output_high(PIN_B1); delay_ms(10);
    n++;
}
```

## ภาคผนวก ก

## เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์

## ก.1 เอกสารคู่มือการใช้งาน PIC16F876A

PIC16F876A เป็นไมโครคอนโทรลเลอร์ที่มีหน่วยความจำภายใน 8 กิโลไบต์ มีรายละเอียดต่างๆ ดังนี้



MICROCHIP

PIC16F87X

## 28/40-Pin 8-Bit CMOS FLASH Microcontrollers

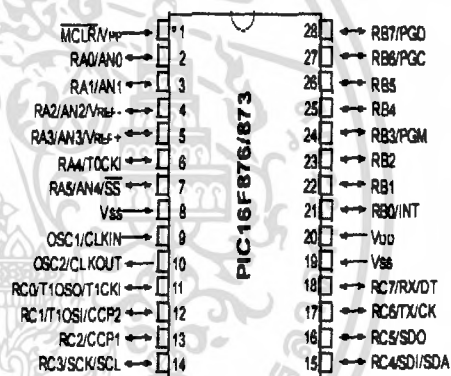
## Devices Included in this Data Sheet:

- PIC16F873
- PIC16F876
- PIC16F874
- PIC16F877

## Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input  
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,  
Up to 368 x 8 bytes of Data Memory (RAM)  
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and  
Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC  
oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM  
technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two  
pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature  
ranges
- Low-power consumption.
  - < 0.6 mA typical @ 3V, 4 MHz
  - 20 µA typical @ 3V, 32 kHz
  - < 1 µA typical standby current

## PDIP, SOIC




## Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,  
can be incremented during SLEEP via external  
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period  
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master  
mode) and I<sup>2</sup>C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver  
Transmitter (USART/SCI) with 9-bit address  
detection
- Parallel Slave Port (PSP) 8-bits wide, with  
external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for  
Brown-out Reset (BOR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ก.2 เอกสารคู่มือการใช้งาน TIP122


TIP122 เป็นทรานซิสเตอร์ชนิดหนึ่ง มีรายละเอียดต่างๆ ดังนี้



### TIP120/121/122

**Medium Power Linear Switching Applications**

- Complementary to TIP125/126/127



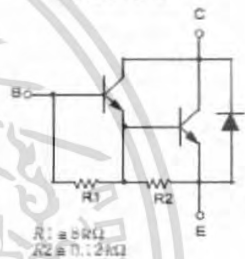
TO-220  
1. Base 2. Collector 3. Emitter

### NPN Epitaxial Darlington Transistor

**Absolute Maximum Ratings**  $T_C=25^\circ\text{C}$  unless otherwise noted

Symbol	Parameter	Value	Units
$V_{CB0}$	Collector-Base Voltage	TIP120	60 V
		TIP121	80 V
		TIP122	100 V
$V_{CEO}$	Collector-Emitter Voltage	TIP120	60 V
		TIP121	80 V
		TIP122	100 V
$V_{EBO}$	Emitter-Base Voltage	5	V
$I_C$	Collector Current (DC)	5	A
$I_{CP}$	Collector Current (Pulse)	8	A
$I_B$	Base Current (DC)	120	mA
$P_C$	Collector Dissipation ( $T_C=25^\circ\text{C}$ )	2	W
		65	W
$T_J$	Junction Temperature	150	$^\circ\text{C}$
$T_{STG}$	Storage Temperature	-65 ~ 150	$^\circ\text{C}$

Equivalent Circuit



**Electrical Characteristics**  $T_C=25^\circ\text{C}$  unless otherwise noted


Symbol	Parameter	Test Condition	Min.	Max.	Units	
$V_{CE(sus)}$	Collector-Emitter Sustaining Voltage	TIP120	$I_C = 100\text{mA}, I_B = 0$	60	V	
		TIP121		80	V	
		TIP122		100	V	
$I_{CEO}$	Collector Cut-off Current	TIP120	$V_{CE} = 30\text{V}, I_B = 0$	0.5	mA	
		TIP121		$V_{CE} = 40\text{V}, I_B = 0$	0.5	mA
		TIP122		$V_{CE} = 50\text{V}, I_B = 0$	0.5	mA
$I_{CBO}$	Collector Cut-off Current	TIP120	$V_{CB} = 60\text{V}, I_E = 0$	0.2	mA	
		TIP121		$V_{CB} = 80\text{V}, I_E = 0$	0.2	mA
		TIP122		$V_{CB} = 100\text{V}, I_E = 0$	0.2	mA
$I_{EBO}$	Emitter Cut-off Current	$V_{BE} = 5\text{V}, I_C = 0$		2	mA	
$h_{FE}$	* DC Current Gain	$V_{CE} = 3\text{V}, I_C = 0.5\text{A}$ $V_{CE} = 3\text{V}, I_C = 3\text{A}$	1000			
$V_{CE(sat)}$	* Collector-Emitter Saturation Voltage	$I_C = 3\text{A}, I_B = 12\text{mA}$		2.0	V	
		$I_C = 5\text{A}, I_B = 20\text{mA}$		4.0	V	
$V_{BE(on)}$	* Base-Emitter ON Voltage	$V_{CE} = 3\text{V}, I_C = 3\text{A}$		2.5	V	
$C_{ob}$	Output Capacitance	$V_{CB} = 10\text{V}, I_E = 0, f = 0.1\text{MHz}$		200	pF	

\* Pulse Test : PW=300 $\mu$ s, Duty cycle  $\leq$ 2%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


### ก.3 เอกสารคู่มือการใช้งาน BC337

BC337 เป็นทรานซิสเตอร์ชนิดหนึ่ง มีรายละเอียดต่างๆ ดังนี้



**FAIRCHILD**  
SEMICONDUCTOR\*

## BC337/338



**Switching and Amplifier Applications**

- Suitable for AF-Driver stages and low power output stages
- Complement to BC327/BC328

1. Collector 2. Base 3. Emitter

**NPN Epitaxial Silicon Transistor**

**Absolute Maximum Ratings**  $T_J=25^{\circ}\text{C}$  unless otherwise noted

Symbol	Parameter	Value	Units
$V_{CES}$	Collector-Emitter Voltage		
	: BC337	50	V
	: BC338	30	V
$V_{CEO}$	Collector-Emitter Voltage		
	: BC337	45	V
	: BC338	25	V
$V_{EBO}$	Emitter-Base Voltage	5	V
$I_C$	Collector Current (DC)	800	mA
$P_C$	Collector Power Dissipation	825	mW
$T_J$	Junction Temperature	150	$^{\circ}\text{C}$
$T_{STG}$	Storage Temperature	-55 - 150	$^{\circ}\text{C}$

**Electrical Characteristics**  $T_J=25^{\circ}\text{C}$  unless otherwise noted

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
$BV_{CEO}$	Collector-Emitter Breakdown Voltage	$I_C=10\text{mA}, I_B=0$	: BC337	45		V
			: BC338	25		V
$BV_{CES}$	Collector-Emitter Breakdown Voltage	$I_C=0, I_B=1\text{mA}, V_{BE}=0$	: BC337	50		V
			: BC338	30		V
$BV_{EBO}$	Emitter-Base Breakdown Voltage	$I_E=0, I_B=1\text{mA}, I_C=0$	5		V	
$I_{CES}$	Collector Cut-off Current	$V_{CE}=45\text{V}, I_B=0$ $V_{CE}=25\text{V}, I_B=0$	: BC337	2	100	nA
			: BC338	2	100	nA
$h_{FE1}$ $h_{FE2}$	DC Current Gain	$V_{CE}=1\text{V}, I_C=100\text{mA}$	100			
		$V_{CE}=1\text{V}, I_C=300\text{mA}$	60		630	
$V_{CE(sat)}$	Collector-Emitter Saturation Voltage	$I_C=500\text{mA}, I_B=50\text{mA}$			0.7	V
$V_{BE(on)}$	Base Emitter On Voltage	$V_{CE}=1\text{V}, I_C=300\text{mA}$			1.2	V
$f_T$	Current Gain Bandwidth Product	$V_{CE}=5\text{V}, I_C=10\text{mA}, f=50\text{MHz}$		100		MHz
$C_{ob}$	Output Capacitance	$V_{CB}=10\text{V}, I_E=0, f=1\text{MHz}$		12		pF

**$h_{FE}$  Classification**

Classification	16	25	40
$h_{FE1}$	100 - 250	160 - 400	250 - 630
$h_{FE2}$	60-	100-	170-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้