สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การขับสเตปมอเตอร์แบบมินิสเตปที่สามารถโปรแกรมได้

PROGRAMMABLE MINISTEP DRIVE

นายทวีศักดิ์ เพียรพิทักษ์วงศ์
นายชาญณรงค์ ยุทธวารี
นายเชิดศักดิ์ สุดใจเพ็ชร์
นายปิตินันต์ ไชยเสน

วท.
ท ม๔๔ก
ร๖๑๕

เลขหมู่.....
เลขทะเบียน 50237
วัน,เดือน,ปี 2 8 มิ.ย. 2547

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAMMABLE MINISTEP DRIVE

TAWEESAK PAINPITACHVONG

CHANNARONG YUTTAWARREE

CHERDSAK SUDJAITETCH

PITINUN CHAIYESAN

A THESIS SUBMITTLD IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEEING IN INSTRUMENTATION ENGINEEING

DEPARTMENT OF IN INSTRUMENTATION ENGINEEING

FACULTY OF ENGINEEING

KING MONGKUT' INSTITUTE OF TECHNOLOLY LADKRABANG

2002

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

.....

หัวข้อปริญญานิพนธ์	การขับสเตปมอเตอร์แบบมินิสเตปที่สามารถโปรแกรมได้			
	PROGRAMMABLE MINISTEP DRIVE			
นักศึกษาผู้จัดทำ	นายทวีศักดิ์	เพียรพิทักษ์วงศ์	รหัสประจำตัว	42010127
	นายชาญณรงค์	ยุทธวงรี	รหัสประจำตัว	43015561
	นายเชิดศักดิ์	สุดใจเพ็ชร์	รหัสประจำตัว	43015562
	นายปิตินันต์	ไชยเสน	รหัสประจำตัว	43015572
ปริญญา	วิศวกรรมศาสตรบัณฑิต			
สาขาวิชา	วิศวกรรมการวัดคุม			
ปีการศึกษา	2545			

อาจารย์ผู้ควบคุมปริญญานิพนธ์	**ลายมือชื่อ**
ผศ. ไสว พงศ์สวัสดิ์ ผศ. ประภาษ อุคคกิมาพันธุ์	

วัน/เดือน/ปี ที่สอบ วันอังคารที่ 22 เมษายน พ.ศ.2546

สถานที่สอบ ณ ห้องสอบปริญญานิพนธ์ ภาควิชาวิศวกรรมการวัดคุม

ภาควิชารับรองแล้ว

(ผศ.ประสิทธ์ จุลเสรีวงศ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช

หัวข้อปริญญานิพจน์	การขับสเตปมอเตอร์แบบมินิสเตปที่สามารถโปรแกรมได้	
	PROGRAMMABLE MINISTEP DRIVE	
นักศึกษาผู้จัดทำ	นายทวีศักดิ์	เพียรพิทักษ์วงศ์
	นายชาญณรงค์	ยุทธวารี
	นายเชิดศักดิ์	สุดใจเพ็ชร์
	นายปิตินันต์	ไชยเสน
อาจารย์ที่ปรึกษา	ผศ. ไสว	พงศ์สวัสดิ์
	ผศ. ประภาษ	อุดคกิมาพันธุ์
ปีการศึกษา	2545	

บทคัดย่อ

การขับสเตปปิงมอเตอร์แบบมินิสเตปนั้น สามารถทำให้สเตปในการหมุนของมอเตอร์เกิดความราบเรียบและสามารถหมุนได้อย่างต่อเนื่อง ให้ผลตอบสนองทางความเร็วที่ดี

ในโครงงานนี้ได้ทำการออกแบบและสร้างชุดการขับเคลื่อนสเตปปิงมอเตอร์แบบมินิสเตปที่สามารถโปรแกรมได้ความละเอียดของมินิสเตปได้ตามความต้องการด้วยการกำหนดค่าความของละเอียดในชุดวงจรขับที่เป็น PMW เพื่อศึกษาการทำงานของมอเตอร์และผลตอบสนองในการหมุน

จากผลการทดลองแสดงให้เห็นถึงว่า การขับแบบมินิสเตปนั้นสามารถทำให้การหมุนของสเตปปิงมอเตอร์มีความราบเรียบมากขึ้นและให้ผลตอบสนองทางความเร็วที่สูงมากขึ้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis title	Programmable Ministep Drive
Authors	Mr. Taweesak Painpitachvong
	Mr. Channarong Yuttawrree
	Mr. Cherdsak Sudjaitetch
	Mr. Pitinun Chaiyesan
Thesis Advisor	Asst.Prof. Sawai Pongsawat
	Asst.Prof. Prapas Ukkakimapun
Year	2002

ABSTRACT

Using **THE MINISTEP TECHNIQUE** with sinusoidal currentto drive stepping motor the revolution of stepping motor is smooth and continuously ,while the speed response is good

This is the project has designed and made the stepping motor driver with programable ministep technique, By designating the driver as PWM control in order to study the stepping motor operating and the revolution response.

From the test results of this motor, the result of the ministep technique can be made the revolution of stepping motor is smoother, while faster speed response is also obtained

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จสมบูรณ์มีเหตุผลใหญ่ๆ 2ประการ ประการแรกเรื่องทางด้านวิชาการ และประการที่สองคือทางด้านจิตใจ

ทางด้านวิชาซึ่งเป็นผู้ให้ค่อยให้คำแนะนำเป็นอย่างดีเสมอมาจากอาจารย์ที่ปรึกษาทั้ง 2ท่าน คือผศ. ไสว พงศ์สวัสดิ์ และผศ. ประภาษ อุดคกิมาพันธุ์ ซึ่งอาจารย์ทั้ง 2ท่านยังเป็นคนกระตุ้นให้งานวิจัยก้าวหน้าตั้งแต่การเริ่มต้นทำปริญญานิพนธ์จนกระทั่งเสร็จสมบูรณ์

ทางด้านจิตใจนั้นซึ่งได้จากคุณพ่อและคุณแม่ของผู้ทำวิจัยโดยเริ่มตั้งแต่เกิดจนถึง ณ นาทีนี้ให้มีกำลังใจในการทำงานจนกระทั่งปริญญานิพนธ์เสร็จสมบูรณ์

อนึ่งบุคคลต่างๆที่เกี่ยวข้องที่ทำให้งานวิจัยนี้เสร็จสมบูรณ์ ทั้งที่ไม่ได้กล่าวมาในข้างต้น ไม่ว่าจะเป็นเพื่อนๆ ในภาควิชาภาควิชาวิศวกรรมการวัดคุมหรืออาจารย์ที่ค่อยอบรมสั่งสอนเรื่องวิชาการและเรื่องจริยธรรมต่างๆ

จงขอให้สิ่งศักดิ์สิทธิ์ทุกท่านที่นับถือช่วย จงคลบรรดาให้ท่านเหล่านั้นประสบความสำเร็จในชีวิตต่อๆไป

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VII

สารบัญภาพ.....VIII

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและเหตุจูงใจของการทำวิจัย.....1

1.2 วัตถุประสงค์ของปริญญานิพนธ์.....1

1.3 ขอบเขตของปริญญานิพนธ์.....1

1.2 รายละเอียดของโครงงาน.....1

บทที่ 2 ทฤษฎีเบื้องต้นเกี่ยวกับมอเตอร์พร้อมกับหลักการขับเคลื่อนมอเตอร์และไมโครคอนโทรลเลอร์.....3

2.1 กล่าวนำ.....3

2.2 มอเตอร์ไฟฟ้าเบื้องต้น.....3

2.2.1 สเตปปิงมอเตอร์.....5

2.2.2 หลักการทำงานของสเตปปิงมอเตอร์.....6

2.2.2.1 คุณลักษณะทางสถิตย์.....7

2.2.2.2 คุณลักษณะทางพลวัต.....9

2.2.3 ประเภทของสเตปปิงมอเตอร์.....10

2.2.3.1 สเตปปิงมอเตอร์แบบแม่เหล็กถาวร.....10

2.2.3.2 สเตปปิงมอเตอร์แบบแปรค่าความต้านทานของสนามแม่เหล็กได้.....11

2.2.3.2 สเตปปิงแบบไฮบริดจ์.....11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.2.3 การกระตุ้นสเตปปิงมอเตอร์.....12

2.2.4 วงจรขับสำหรับสเตปปิงมอเตอร์.....13

2.2.5.1 การใช้ตัวต้านต่ออนุกรมกับมอเตอร์.....14

2.2.5.2 การใช้วงจรขับแบบใช้แรงดันไฟฟ้าสองระดับ.....15

2.2.5.3 การใช้วงจรขับแบบชอปเปอร์.....16

2.2.5.4 การใช้วงจรขับแบบบริดจ์.....17

2.3 ไมโครคอนโทรลเลอร์ MCS-51.....18

2.3.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูลMCS-51อนุกรมAT89xx.....19

2.3.2 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51.....20

2.3.3 โครงสร้างและทำงานของพอร์ต.....24

2.3.3.1 การใช้งานเป็นพอร์ตอินพุต.....25

2.3.3.2 การใช้งานเป็นพอร์ตเอาต์พุต.....26

บทที่ 3 การออกแบบและพัฒนาชุดอุปกรณ์ที่ต่อร่วมที่ใช้กับมอเตอร์.....28

3.1 หลักการออกแบบชุดขับเคลื่อนสเตปปิงมอเตอร์.....28

3.1.1 การออกแบบการป้อนแบบฟูลสเตป.....28

3.1.1.1 ลักษณะแบบกระตุ้นเฟสเดียว.....28

3.1.1.2 แบบกระตุ้นสองเฟส.....29

3.1.2 การออกแบบการป้อนฮาล์ฟสเตป.....30

3.2.3 การออกแบบการป้อนแบบมินิสเตป.....31

3.2 หลักการทำงานชุดของระบบขับเคลื่อน.....32

3.2.1 ส่วนของโปรแกรมและวงจรแปลงดิจตอลเป็นอนาล็อก.....32

3.2.2 วงจรขับเคลื่อนมอเตอร์.....33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 4 ผลการทดลอง.....36

4.1 การทดลองผลการตอบสนองทางความเร็ว.....36

4.1.1.1 ผลการทดลองป้อนแบบฟูลสเตป.....37

4.1.1.2 ผลการทดลองป้อนแบบฮาล์ฟสเตป.....39

4.2.1.3 ผลการทดลองการป้อนมินิสเตป.....42

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....44

5.1 สรุปผลการวิจัย.....44

5.2 ข้อเสนอแนะ.....44

5.3 ปัญหาสำหรับงานวิจัย.....44

บรรณานุกรม.....45

ภาคผนวก.....46

ภาคผนวก ก.....47

ภาคผนวก ข.....86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 แสดงการกระตุ้นที่ละเฟสจะป้อนไฟกระตุ้นให้กับสเตปปิงมอเตอร์ที่ละเฟส.....13
2.2 แสดงการกระตุ้นแบบย่อยสเตปจะป้อนไฟกระตุ้นให้กับสเตปปิงมอเตอร์ 13
2.3 แสดงรายละเอียดโดยสรุปบางส่วนของไมโครคอนโทรลเลอร์ MCS-51.....21
2.4 แสดงหน้าที่พิเศษของพอร์ต1 ในไมโครคอนโทรลเลอร์ MCS-51.....25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ หน้า

2.1 การประยุกต์ใช้งานระบบควบคุมทั่วไปๆกับการควบคุมมอเตอร์
ที่นำมาใช้ในโครงงานนี้.....3
2.2 โครงสร้างพื้นฐานของมอเตอร์ไฟฟ้ากระแสตรง.....4
2.3 โครงสร้างพื้นฐานของมอเตอร์ไฟฟ้ากระแสสลับ.....4
2.4 โครงสร้างพื้นฐานของมอเตอร์ไฟฟ้ากระแสสลับที่โรเตอร์เป็นแม่เหล็กถาวร.....5
2.5 โครงสร้างของสเตปปิงมอเตอร์.....5
2.6 แสดงการหมุนเป็นสเตป.....6
2.7 แสดงการหแบบต่อเนื่อง.....6
2.8 กราฟคุณลักษณะทางสถิตย์ของสเตปปิงขนาดสามเฟส.....7
2.9 กราฟคุณลักษณะของแรงบิดต่อมุมสเตป.....8
2.10 กราฟคุณลักษณะของแรงบิดต่อกระแส.....8
เส้น (ก.) แทนค่าของสเตปปิงแบบแปรค่าความต้านทาน
เส้น (ข.) แทนค่าของสเตปปิงแบบไฮบริดจ์
2.11 กราฟคุณลักษณะทางพลวัตของสเตปปิงมอเตอร์.....9
2.12 โครงสร้างของสเตปปิงมอเตอร์แบบแม่เหล็กถาวรสี่เฟส.....11
2.13 โครงสร้างของสเตปปิงมอเตอร์แบบแม่เหล็กถาวร.....12
2.14 แสดงวงจรสมมูลย์ของสเตปปิงมอเตอร์.....14
2.15 แสดงการใช้ความต้านทานต่ออนุกรมกับขดลวดของสเตปปิงมอเตอร์.....15
2.16 แสดงวงจรขับสเตปปิงมอเตอร์แบบแรงดันไฟฟ้าสองระดับ.....16
2.17 แสดงวงจรขับแบบชอปเปอร์.....17
2.18 แสดงวงจรขับแบบบริดจ์ ที่ใช้สองแหล่งจ่าย.....17
2.19 แสดงวงจรขับแบบบริดจ์ที่ใช้แหล่งจ่ายเดียว.....18
2.20 แสดงโครสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 18
2.21 แสดงโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 21
2.22 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชของ Atmel.....22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ หน้า

2.23 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89C5x.....23

2.24 วงจรภายในของพอร์ตทุกพอร์ตในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช.....26

2.25 วงจรพูลอัปภายในพอร์ตไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช.....27

3.1 สัญญาณที่ใช้ป้อนชุดขับเคลื่อนมอเตอร์โดยการแบบพัลส์เฟสเดียว.....29

3.2 แสดงลักษณะของการหมุนมอเตอร์แบบพัลส์เฟสเดียว.....29

3.3 สัญญาณที่ใช้ป้อนชุดขับเคลื่อนมอเตอร์โดยการแบบพัลส์สองเฟส.....29

3.4 แสดงลักษณะของการหมุนมอเตอร์แบบพัลส์สองเฟส.....30

3.5 สัญญาณที่ใช้ป้อนชุดขับเคลื่อนมอเตอร์โดยการแบบฮาล์ฟสเตป.....30

3.6 แสดงลักษณะของการหมุนมอเตอร์แบบฮาล์ฟสเตป.....31

3.7 แสดงสัญญาณที่ใช้ป้อนชุดขับเคลื่อนมอเตอร์โดยการแบบฟังชั่นแบบมินิเตป.....31

3.8 แสดงลักษณะของการหมุนมอเตอร์แบบมินิสเตป.....32

3.9 แสดงบล็อกการทำงานของระบบขับเคลื่อน.....33

3.10 แสดงวงจรขับเคลื่อนแบบสองแกนขวาง.....34

3.11 แสดงตัวอย่างสัญญาณ PMW ที่ใช้ป้อนชุดขับเคลื่อนมอเตอร์โดยการแบบมินิสเตป.....34

3.12 แสดงโฟลชาร์ตการทำงานของโปรแกรมที่นำมาใช้.....35

4.1 แสดงลักษณะของระบบควบคุมโดยรวมที่ได้ออกแบบไว้.....36

4.2 แสดงลักษณะรูปคลื่นสัญญาณอ้างอิงของแบบฟูลสเตป.....37

4.3 แสดงลักษณะรูปคลื่นสัญญาณเครื่องหมาย.....37

4.4 แสดงรูปคลื่นสัญญาณอ้างอิงกับสัญญาณเครื่องหมาย.....38

4.5 แสดงรูปคลื่นขณะที่จ่ายให้กับมอเตอร์.....38

4.6 แสดงกราฟความสัมพันธ์ความถี่กับความเร็วเมื่อการป้อนแบบฟูลสเตป.....39

4.7 แสดงรูปคลื่นของสัญญาณอ้างอิงของการป้อนแบบฮาล์ฟสเตป.....39

4.8 แสดงรูปคลื่นของสัญญาณเครื่องหมายของการป้อนแบบฮาล์ฟสเตป.....40

4.9 แสดงรูปคลื่นของแสดงรูปคลื่นสัญญาณอ้างอิงกับสัญญาณเครื่องหมาย
ของการป้อนรูปคลื่น แบบฮาล์ฟสเตป.....40

4.10 แสดงรูปคลื่นขณะที่จ่ายให้กับมอเตอร์.....41

4.11 แสดงกราฟความสัมพันธ์ความถี่กับความเร็วเมื่อการป้อนแบบฮาล์ฟสเตป.....41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ หน้า

4.12 แสดงรูปคลื่นของสัญญาณอ้างอิงของการป้อนแบบมินิเตป.....42
4.13 แสดงรูปคลื่นของสัญญาณเครื่องหมายของการป้อนแบบมินิสเตป.....42
4.14 ลักษณะรูปคลื่นขณะจ่ายให้แก่มอเตอร์.....43
4.15 แสดงกราฟความสัมพันธ์ความถี่กับความเร็วเมื่อการป้อนแบบมินิสเตป.....43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและเหตุจูงใจ

การใช้งานสเตปปิงมอเตอร์โดยทั่วไปนั้นสามารถใช้งานได้ง่าย และทำงานได้ดี แต่ยังมีข้อเสียอีกอย่างหนึ่งคือการหมุนของสเตปแบบไม่ต่อเนื่อง และเกิดแรงดันเหนี่ยวนำที่เกิดจากการหมุนที่เพลาของมอเตอร์

ซึ่งการกระตุ้นสเตปปิงมอเตอร์ในรูปแบบมินิสเตปนั้น จะทำให้สเตปปิงมอเตอร์หมุนได้อย่างราบเรียบและต่อเนื่องเทียบเคียงได้ราวกับมอเตอร์ไฟฟ้ากระแสตรง ขณะที่เราสามารถควบคุมตำแหน่งของการหมุนได้อย่างแม่นยำอีกด้วยซึ่งเป็นข้อดีควบคู่กันไป

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาการควบคุมและการขับเคลื่อนของมอเตอร์กระแสสลับโดยการใช้สเตปปิงมอเตอร์เป็นกรณีศึกษา
2. เขียนโปรแกรมการควบคุมมอเตอร์ โดยใช้ไมโครคอนโทรลเลอร์
3. ศึกษาผลกระทบต่างๆ ที่มีต่อมอเตอร์หลักจากนำโครงงานนี้มาขับเคลื่อนมอเตอร์

1.3 ขอบเขตของปริญญานิพนธ์

ปริญญาฉบับนี้นิพนธ์นี้ได้กล่าวถึงการพัฒนาและออกแบบวงจรขับสเตปปิงมอเตอร์โดยทำการป้อนกระแสแบบมินิสเตป เพื่อนำมาแก้ปัญหาการหมุนแบบไม่ต่อเนื่องของสเตปปิงมอเตอร์

ในกรณีของที่การหมุนแบบต่อเนื่อง โดยการให้กำลังงานในรูปแบบแบบมินิสเตปนั้นเป็นการแก้ปัญหาแรงดันเหนี่ยวนำที่เกิดจากการหมุนที่เพลาของมอเตอร์ เพื่อให้แรงยึดเหนี่ยวระหว่างโรเตอร์และสเตเตอร์มีค่าคงที่ตลอดย่านความเร็วตอบสนอง จะทำให้สเตปปิงมอเตอร์นั้นมีคุณสมบัติเทียบเคียงได้ราวกับมอเตอร์ไฟฟ้ากระแสตรง

1.4 รายละเอียดของปริญญานิพนธ์

ในโครงงานฉบับนี้แบ่งออกเป็นดังนี้

1. ในส่วนบทที่1 จะกล่าวถึงความเป็นมาของการทำวิจัย วัตถุประสงค์ของปริญญานิพจน์ ขอบเขตของปริญญานิพจน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ในส่วนที่บทที่2จะกล่าวถึงทฤษฎีเกี่ยวกับสเตปปิงมอเตอร์และหลักการขับเคลื่อนของมอเตอร์ และโครงสร้างของไมโครคอนโทรลเลอร์

3. ในส่วนบทที่3 จะกล่าวถึงการออกแบบและพัฒนาชุดขับเคลื่อนมอเตอร์

4. ในส่วนบทที่4 จะกล่าวถึงผลการทดลอง

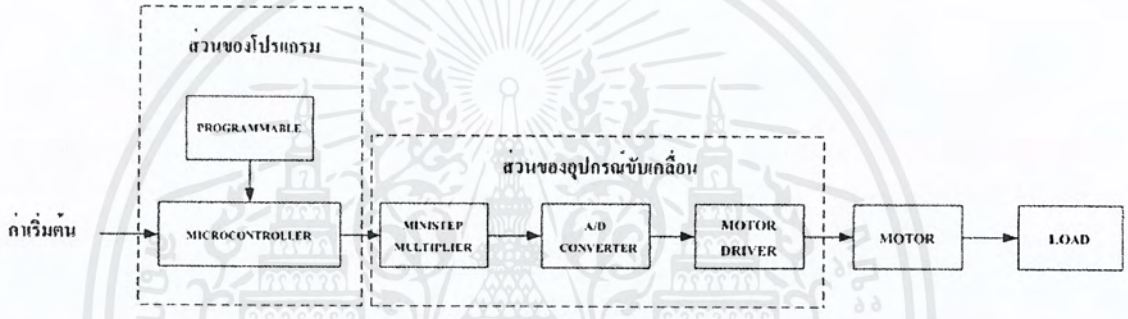
5. ในส่วนบทที่5 จะกล่าวถึงสรุปผลการทดลองและข้อเสนอแนะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีเบื้องต้นเกี่ยวกับมอเตอร์พร้อมกับ หลักการขับเคลื่อนมอเตอร์และไมโครคอนโทรลเลอร์

2.1 กล่าวนำ

โดยทั่วไประบบการควบคุมที่ดีคือ นั้นการรักษาสภาพระบบให้ดีหรือการควบคุมค่าต่างๆ ให้ได้ตามค่าที่ต้องการ เราจึงสามารถประยุกต์นำระบบควบคุมมาใช้กับการขับเคลื่อนมอเตอร์



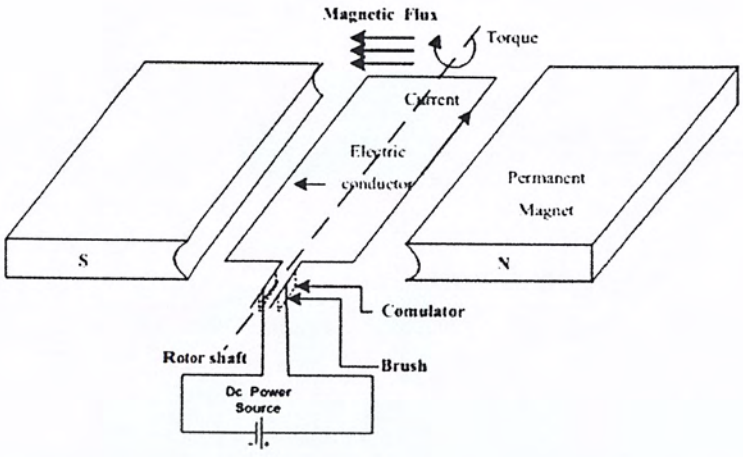
ภาพที่ 2.1 การประยุกต์ใช้งานระบบควบคุมทั่วไปๆกับการควบคุมมอเตอร์

จากภาพที่2.1 ประกอบด้วยชุดควบคุมแบบพื้นฐานซึ่งสามารถนำมาประยุกต์กับระบบที่ใช้ควบคุมตำแหน่งของตัวโรเตอร์เพื่อให้คอนโทรลเลอร์ทำการควบคุมการหมุนของสเตปปิงมอเตอร์ได้อย่างต่อเนื่องเทียบเคียงได้ราวเหมือนกับมอเตอร์ไฟฟ้ากระแสตรง โดยขณะที่เราสามารถควบคุมตำแหน่งของการหมุนได้อย่างแม่นยำอีกด้วย

2.2 มอเตอร์ไฟฟ้าเบื้องต้น

มอเตอร์ไฟฟ้าแบ่งออกเป็น 2ประเภทใหญ่ ๆ คือ มอเตอร์ไฟฟ้ากระแสตรงและมอเตอร์ไฟฟ้ากระแสสลับ ดังแสดงในรูป ซึ่งเป็นโครงสร้างพื้นฐานของมอเตอร์ไฟฟ้ากระแสตรง ซึ่งการหมุนและการเปลี่ยนแปลงความเร็วรอบในการหมุนมอเตอร์ไฟฟ้ากระแสตรง จะใช้หลักการเปลี่ยนแปลงแรงดันไฟฟ้ากระแสตรงที่ป้อนให้กับอาร์เมเจอร์ของมอเตอร์ ซึ่งความเร็วรอบของมอเตอร์จะแปรค่าแรงดันไฟฟ้าที่ป้อนให้กับอาร์เมเจอร์

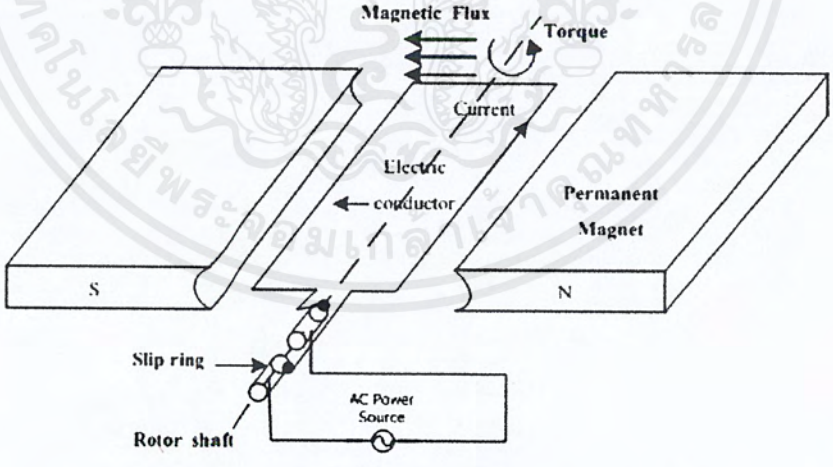
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.2 โครงสร้างพื้นฐานของมอเตอร์ไฟฟ้ากระแสตรง

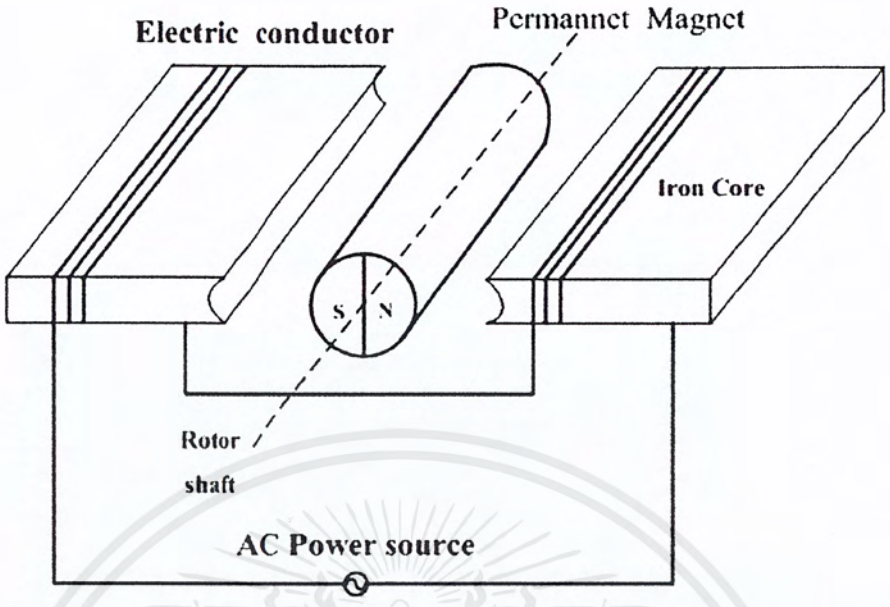
มอเตอร์ไฟฟ้ากระแสสลับจะป้อนไฟกระแสสลับเข้าที่ตัวมอเตอร์และการเปลี่ยนแปลงความเร็วรอบในการหมุนของมอเตอร์ไฟฟ้ากระแสสลับจะใช้หลักการเปลี่ยนแปลงความถี่ของแรงดันไฟฟ้ากระแสสลับที่ป้อนให้กับมอเตอร์ (AC Power Source) โดยที่ค่าความเร็วรอบของมอเตอร์ไฟฟ้ากระแสสลับจะเป็นสัดส่วนโดยตรงกับความถี่ที่ป้อนตามภาพที่ 2.3 โดยจะแสดงให้เห็นโครงสร้างของมอเตอร์ไฟฟ้ากระแสสลับที่ตัวโรเตอร์เป็นลวดตัวนำ

ส่วนภาพที่ 2.4 จะแสดงลักษณะของโรเตอร์ที่เป็นแม่เหล็กถาวร



ภาพที่ 2.3 โครงสร้างพื้นฐานของมอเตอร์ไฟฟ้ากระแสสลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

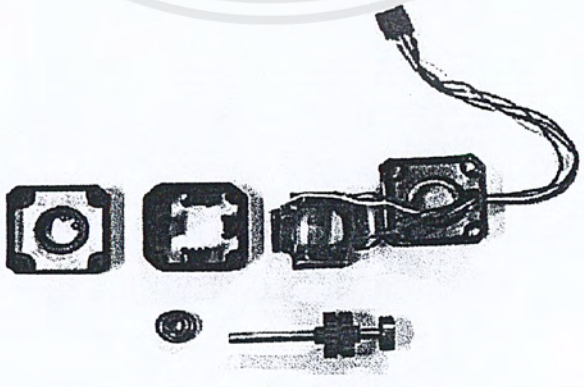


ภาพที่ 2.4 โครงสร้างพื้นฐานของมอเตอร์ไฟฟ้ากระแสสลับที่โรเตอร์เป็นแม่เหล็กถาวร

2.2.1 สเตปปิงมอเตอร์

สเตปปิงมอเตอร์เป็นมอเตอร์ไฟฟ้ากระแสสลับที่มีโรเตอร์เป็นแม่เหล็กถาวร และใช้การกระตุ้นด้วยแม่เหล็กไฟฟ้าจากขดลวดที่สเตเตอร์โดยแบ่งชุดขดลวดที่สเตเตอร์ออกเป็นเฟสต่าง ๆ

การขับสเตปปิงมอเตอร์ต้องใช้วงจรขับควบคุมการจ่ายกระแสไฟฟ้าให้กับขนลวดตัวนำในแต่ละเฟสเพื่อสร้างสนามแม่เหล็กไฟฟ้าให้สอดคล้องกับการหมุนของโรเตอร์ที่เป็นแม่เหล็กชนิดถาวร



ภาพที่ 2.5 โครงสร้างของสเตปปิงมอเตอร์

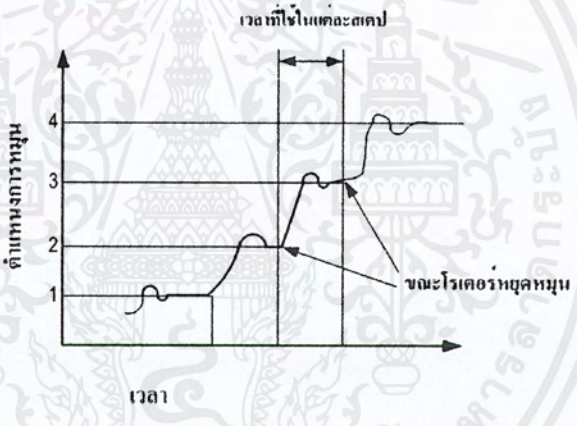
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 หลักการทำงานของสเตปปิงมอเตอร์

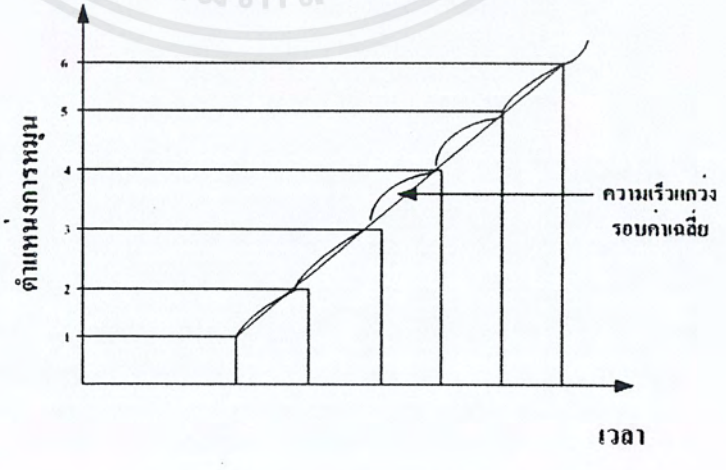
สเตปปิงมอเตอร์เป็นมอเตอร์ไฟฟ้ากระแสสลับชนิดที่เปลี่ยนสัญญาณดิจิตอลเป็นการเคลื่อนที่ทางกล จึงเหมาะสำหรับการเชื่อมต่อกับอุปกรณ์ทางดิจิตอลหรือคอมพิวเตอร์ โดยการทำงานของสเตปปิงมอเตอร์โดยส่วนใหญ่จะขึ้นอยู่กับสัญญาณพัลส์ (PULSE) กระตุ้นที่ป้อนให้กับขดลวดเฟสของมอเตอร์ในลำดับที่ถูกต้องด้วยวงจรจัดลำดับลอจิกและกระแสที่พอเพียงเพื่อจะป้อนให้กับวงจรขับ

การทำงานของสเตปปิงมอเตอร์ใช้อัตราเร็วของแต่ละสเตปนั้น สามารถจะแบ่งออกเป็นลักษณะของ 2โหมดการทำงานคือ

1. โหมดของการหมุนเป็นสเตป (Discrete Mode)
2. โหมดของการหมุนแบบต่อเนื่อง (Slewing Mode)



ภาพที่ 2.6 แสดงการหมุนเป็นสเตป



ภาพที่ 2.7 แสดงการหมุนแบบต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะการหมุนแบบสเตปจะมีช่วงเวลาหยุดนิ่งของโรเตอร์ก่อนที่จะเปลี่ยนสเตปถัดไป ดังนั้นหากมีการเพิ่มอัตราเร็วในแต่ละสเตปให้เร็วขึ้นและเป็นไปอย่างต่อเนื่อง การหมุนของมอเตอร์ก็จะเปลี่ยนเป็นการหมุนแบบต่อเนื่อง

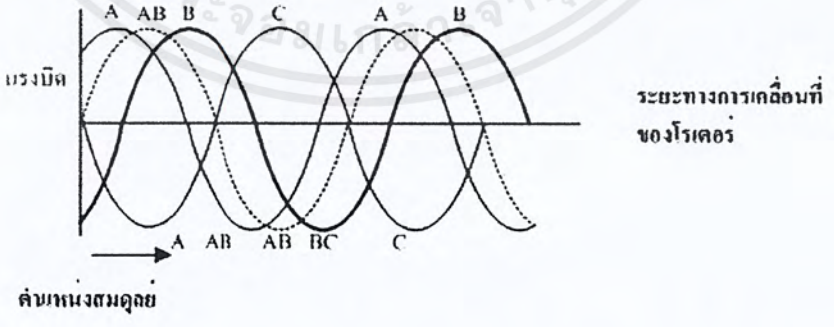
การที่นำสเตปปิงมอเตอร์ไปใช้งานไม่ว่าจะเป็นโหมดการทำงานแบบใดหรือเป็นมอเตอร์แบบไหนควรจะมีการศึกษาคุณลักษณะต่างๆ เช่น แรงบิด การตอบสนองต่อความถี่สูงสุดและความสัมพันธ์ระหว่างการเปลี่ยนแปลงของแรงบิดกับค่าของกระแส ทั้งนี้เพื่อให้สามารถควบคุมการทำงานของสเตปปิงมอเตอร์ได้อย่าง มีประสิทธิภาพมากที่สุด ซึ่งคุณลักษณะเหล่านี้สามารถแบ่งออกได้เป็นสองลักษณะคือ

1. คุณลักษณะทางสถิตย์ (Static Characteristics)
2. คุณลักษณะทางผลวัต (Dynamic Characteristics)

2.2.2.1 คุณลักษณะทางสถิตย์ (Static Characteristics)

คุณลักษณะทางสถิตย์ หมายถึง คุณลักษณะในสภาวะที่มอเตอร์หยุดนิ่งไม่มีการเคลื่อนไหว ซึ่งจะเป็นตำแหน่งที่มีความสมดุลย์ของการหมุนมากดังที่สุดกล่าวคือ เป็นตำแหน่งที่ซี่ฟันของโรเตอร์และซี่ฟันของสเตเตอร์มีความสัมพันธ์กันอยู่ในแนวเดียวกัน และยังคงอยู่ในตำแหน่งนี้จนกว่าจะมีการกระตุ้นอีกครั้ง โดยที่ตำแหน่งสมดุลย์ของโรเตอร์จะมีค่าของแรงบิดเท่ากับศูนย์ ซึ่งถ้าโรเตอร์มีการเคลื่อนที่ไปในทิศทางใดทิศทางหนึ่ง มอเตอร์ก็จะทำการสร้างสมดุลย์ให้กับโหลดโดยการสร้างแรงบิดขึ้นมาในทิศทางตรงกันข้ามกับทิศทางการเคลื่อนที่

ผลของแรงบิดกับการเคลื่อนที่ของโรเตอร์จะมีลักษณะคล้ายกับรูปคลื่นซายน์



ภาพที่ 2.8 กราฟคุณลักษณะทางสถิตย์ของสเตปปิงขนาดสามเฟส

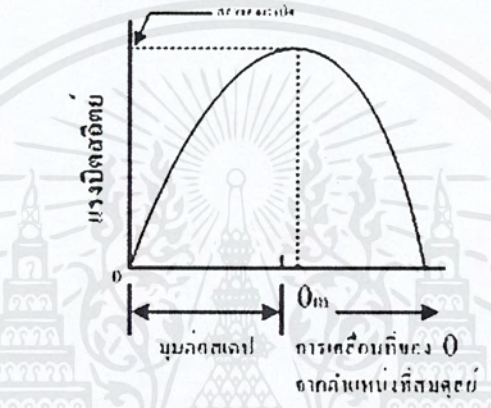
แรงบิดสูงสุดที่เกิดขึ้นจะขึ้นอยู่กับกระแสที่จ่ายให้เฟสนั้น ๆ โดยที่แรงบิดนี้เรียกว่าสภาพคงแรงบิด(HoldingTorque)หรือแรงบิดสถิตย์(StaticTorque)โดยลักษณะความสัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

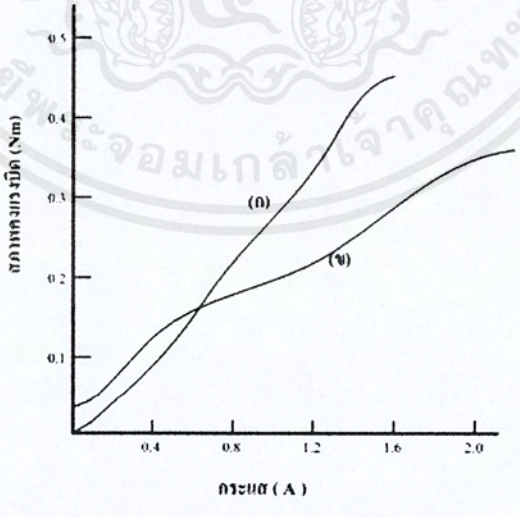
ระหว่างค่าของแรงบิดกับกระแสที่เปลี่ยนแปลงไปจะเป็นลักษณะเชิงเส้น แต่ก็จะมีขีดจำกัดอยู่ที่จุดอิ่มตัวแม่เหล็กของทั้งโรเตอร์และสเตเตอร์ซึ่งมีโครงสร้างเป็นแม่เหล็ก โดยที่จุดอิ่มตัวคือจุดที่ไม่มีการเพิ่มขึ้นของแรงบิดแม้ว่าจะมีการเพิ่มขึ้นของกระแสเฟสก็ตาม

คุณลักษณะทางสถิตย์จะแสดงความสัมพันธ์ดังต่อไปนี้

1. คุณลักษณะของแรงบิดกับมุมสเตป
2. คุณลักษณะของแรงบิดกับกระแส



ภาพที่ 2.9 กราฟคุณลักษณะของแรงบิดต่อมุมสเตป



ภาพที่ 2.10 กราฟคุณลักษณะของแรงบิดต่อกระแส

เส้น (ก) แทนค่าของสเตปปิงแบบแปรค่าความต้านทาน

เส้น (ข) แทนค่าของสเตปปิงแบบไฮบริดจ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2.2 คุณลักษณะทางพลวัต

ในการพิจารณาคุณลักษณะสถิตย์ของแรงบิดของสเตปปิงมอเตอร์ จะพิจารณาก็ต่อเมื่อไม่มีการเคลื่อนที่ของโรเตอร์ แต่การพิจารณาคุณลักษณะทางพลวัตของสเตปปิงมอเตอร์นั้นจะทำให้เห็นมองถึง ความสามารถในการสร้างแรงบิดที่สัมพันธ์กับความเร็ว โดยสามารถศึกษาได้จากคู่มือ (Data Sheet) ของบริษัทผู้ผลิต ดังตัวอย่างตามภาพที่ 2.11

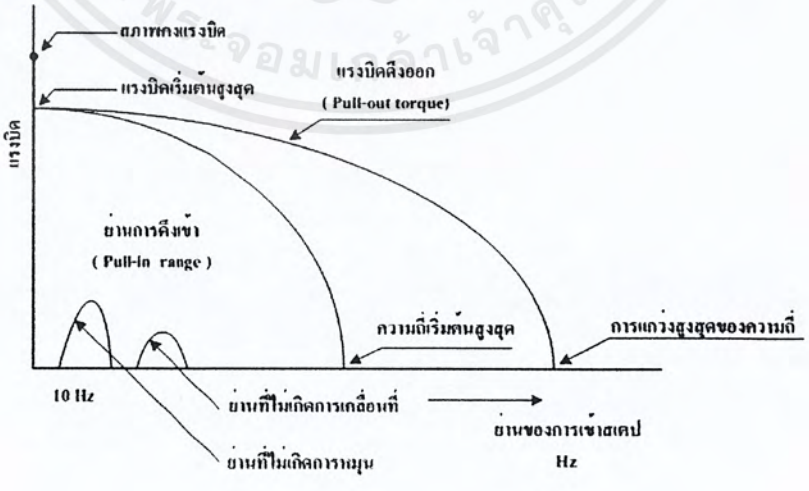
ก. คุณลักษณะของแรงบิดดึงเข้า (Pull-in Torque)

เป็นคุณลักษณะเริ่มต้นและจะอ้างอิงกับย่านของแรงบิดที่เกิดจากแรงเสียดทานของโหลดที่ซึ่งมอเตอร์สามารถเคลื่อนที่และหยุดได้โดยปราศจากการสูญเสียในสเตป สำหรับย่านของความถี่ต่าง ๆ ที่ใช้นั้นจำนวนของพัลส์ที่ใช้ในการทดสอบจะอยู่ในช่วง 100เฮิรตซ์ (H_Z)

ข. คุณลักษณะของแรงบิดดึงออก (Pull-out Torque)

สำหรับพิจารณาคุณลักษณะในขณะที่มอเตอร์เคลื่อนที่หลังจากที่ได้ทำการทดสอบมอเตอร์ให้เกิดการเคลื่อนที่โดยใช้วงจรขับเคลื่อนลักษณะพิเศษ ในกรณีของการกระตุ้นในย่านของการเคลื่อนที่ด้วยตนเอง (Self Starting Range) ความถี่ของพัลส์จะค่อย ๆ เพิ่มขึ้นมากจนกระทั่งมอเตอร์เกิดการเคลื่อนที่โดยปราศจากการเข้าจังหวะ

ความสัมพันธ์ระหว่างแรงบิดที่เกิดจากแรงเสียดทานของโหลดกับความถี่พัลส์สูงสุด จะทำให้มอเตอร์สามารถเคลื่อนที่ได้อย่างมีการเข้าจังหวะ ซึ่งจะเรียกว่าคุณลักษณะดึงออกดังภาพที่ 2.11 โดยที่ขนาดของเส้นโค้งดึงออกจะขึ้นอยู่กับวงจรขับ การต่อเชื่อม (Coupling) เครื่องมือที่ใช้ในการวัด (Measuring Instruments)



รูปที่ 2.11 กราฟคุณลักษณะทางพลวัตของสเตปปิงมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค. ความถี่สูงสุดที่ใช้ในการเคลื่อนที่ (Maximum Starting Frequency)

เป็นการกำหนดการควบคุมความถี่สูงสุดให้มอเตอร์เคลื่อนที่และหยุดได้ในขณะที่ไม่มีโหลดโดยปราศจากการสูญเสียทางสเตปปิงมอเตอร์

ง. ระดับสูงสุดของอัตราการดึงออก (Maximum Pull-out Rate)

เป็นการกำหนดค่าความถี่สูงสุดในขณะที่มอเตอร์สามารถเคลื่อนที่ได้เมื่อไม่มีโหลดโดยปราศจากการสูญเสียทางสเตปปิงมอเตอร์

จ. แรงบิดสูงสุดที่เกิดจากการเคลื่อนที่ (maximum Starting Torque)

เรียกว่า แรงบิดดึงเข้าสูงสุด (Maximum Pull-in Torque) และเป็นการกำหนดค่าของแรงบิดที่เกิดจากแรงเสียดทางสูงสุดของโหลด ซึ่งทำให้มอเตอร์สามารถเคลื่อนที่ได้อย่างเข้าจังหวะกับพัลส์ที่มีความถี่ต่ำประมาณ 10เฮิรตซ์

2.2.3 ประเภทของสเตปปิงมอเตอร์

สเตปปิงมอเตอร์สามารถแบ่งออกเป็นประเภทต่าง ๆ ได้ดังนี้

1. สเตปปิงมอเตอร์แบบแม่เหล็กถาวร
2. สเตปปิงมอเตอร์แบบแปรค่าความต้านทานของสนามแม่เหล็กได้
3. สเตปปิงมอเตอร์แบบไฮบริดจ์

2.2.3.1 สเตปปิงมอเตอร์แบบแม่เหล็กถาวร

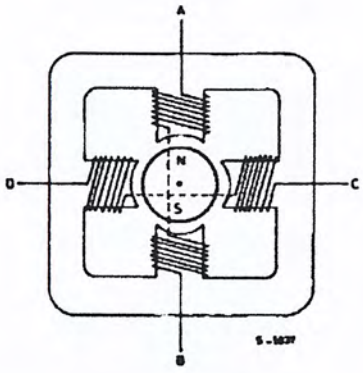
สเตปปิงมอเตอร์ชนิดนี้จะใช้แม่เหล็กถาวรเป็นโรเตอร์โดยที่ตัวของโรเตอร์รูปร่างเป็นทรงกระบอกและสเตเตอร์จะถูกแบ่งออกเป็นซี่ๆ หรือที่เรียกว่าเฟสล้อมรอบตามภาพที่ 2.12 โดยในแต่ละซี่นั้นจะมีขดลวดพันโดยรอบๆ เพื่อทำให้เกิดการกระตุ้นในการสร้างสนามแม่เหล็ก การให้สเตปปิงมอเตอร์แบบแม่เหล็กถาวรนั้นมีขนาดมุมสเตป (step) เล็กลง สามารถทำให้ได้โดยการเพิ่มจำนวนของขั่วแม่เหล็กของโรเตอร์และหรือจำนวนซี่ฟันของสเตเตอร์

แต่การเพิ่มความละเอียดของสเตปก็มีขีดจำกัดในการเพิ่มของจำนวนขั้วแม่เหล็กของโรเตอร์ ทั้งนี้เนื่องจากการสร้างแม่เหล็กถาวรให้มีโครงสร้างแบบมีขั้วแม่เหล็กหลายๆขั้นนั้นทำได้ยากจึงกลายเป็นข้อเสียของสเตปปิงมอเตอร์ ซึ่งในลักษณะนี้มีข้อเสียคือ

ก. มีขนานของมุมสเตปใหญ่จะทำให้มีความละเอียดของสเตปต่อรอบน้อยมาก เนื่องจากว่าโครงสร้างของโรเตอร์เป็นแม่เหล็กถาวร

ข. สเตปปิงมอเตอร์แบบแม่เหล็กถาวรส่วนใหญ่จะมีโครงสร้างขนาดเล็กทำให้ค่าของแรงบิดที่ได้ต่อหน่วยปริมาตรมีค่าต่ำ ถ้าต้องการปรับปรุงประสิทธิภาพในเรื่องของแรงบิดแม่เหล็กถาวรที่ใช้จะต้องทำมาจากสารแม่เหล็กที่มีสภาพความเป็นแม่เหล็กสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.12 โครงสร้างของสเตปปิงมอเตอร์แบบแม่เหล็กถาวรสี่เฟส

2.2.3.2 สเตปปิงมอเตอร์แบบแปรค่าความต้านทานของสนามแม่เหล็กได้ (VariableReluctance Stepping Motor)

สเตปปิงมอเตอร์ชนิดนี้เรียกอีกอย่างหนึ่งว่า วีอาร์สเตปปิงมอเตอร์ โดยโครงสร้างมีโรเตอร์และสเตเตอร์ที่ทำจากโลหะซิลิคอน (Silicon) ซึ่งเป็นสารแม่เหล็กที่มีสภาพความซึมซาบทางแม่เหล็ก (Permeability) สูงมากก็จะสามารถให้เส้นแรงของแม่เหล็กไหลผ่านได้มาก

2.2.3.3 สเตปปิงมอเตอร์แบบไฮบริดจ์ (Hybrid Stepping Motor)

เป็นสเตปปิงมอเตอร์ที่มีโรเตอร์เป็นแม่เหล็กถาวร การใช้ชื่อว่าไฮบริดจ์ได้มาจากการรวมหลักสำคัญของมอเตอร์แบบแม่เหล็กถาวร และแบบแปรค่าความต้านทานของสนามแม่เหล็กได้เข้าด้วยกันทำให้สเตปปิงมอเตอร์ชนิดแบบไฮบริดจ์จะมีโครงสร้างของสเตเตอร์คล้ายกับโครงสร้างของสเตปปิงมอเตอร์แบบแปรค่าต้านต้านทานแม่เหล็กได้ ซึ่งจะมีขดลวดอยู่สองขดขด และขดลวดทั้งองจะพันอยู่ที่ขั้วเดียวกันในลักษณะของสองภาวะขั้น (Bipolar) ซึ่งจะทำให้ขั้วแม่เหล็กมีความต่างกันขณะที่มีการกระตุ้นแต่ละครั้ง

ลักษณะที่สำคัญของสเตปปิงมอเตอร์ชนิดแบบไฮบริดจ์นั้นคือ ตัวโรเตอร์จะเป็นแม่เหล็กชนิดถาวรและรูปร่างทรงกระบอกตามยาวอยู่ในแกนเหล็กของโรเตอร์ เพื่อใช้สร้างสนามแม่เหล็กกั้นเดียวและแต่ละขั้วของแม่เหล็กจะถูกล้อมรอบด้วยฟันเหล็กอ่อน โดยที่ฟันของโรเตอร์กับสเตเตอร์จะอยู่ในตำแหน่งที่เหลื่อมกันอยู่ 90องศา

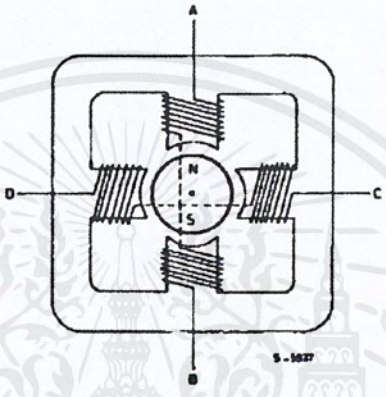
หลักการทำงานของสเตปปิงมอเตอร์แบบไฮบริดจ์ที่แตกต่างจากสเตปปิงมอเตอร์แบบ แปรค่าต้านทานของสนามแม่เหล็กได้ก็คือ แรงบิดที่เกิดขึ้นจากสนามแม่เหล็กจะไม่ขึ้นอยู่กับกระแสที่ไหลผ่านขดลวดเพียงอย่างเดียว แต่จะขึ้นอยู่กับโครงสร้างของซี่ฟันด้วยและข้อดีของสเตปปิงมอเตอร์แบบไฮบริดจ์คือมีนาดของสเตปเล็กทำให้มีความละเอียดของสเตปต่อรอบสูง นอกจากนี้ยังมีค่าของแรงบิดสูงกว่าสเตปปิงมอเตอร์แบบแปรค่าความต้านทานของสนามแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่สเตปปิงมอเตอร์ชนิดที่สามารถแบบแปรค่าต้านต้านทานของสนามแหล็กได้จะมีแรงเฉื่อยทางกลไกน้อยกว่า

2.2.4 การกระตุ้นสเตปปิงมอเตอร์

การกระตุ้นสเตปปิงมอเตอร์สามารถทำได้โดยการป้อนไฟให้กับขดลวดในแต่ละเฟสตามลำดับที่เหมาะสม จากภาพ 2.13 โครงสร้างของสเตปปิงมอเตอร์



ภาพที่ 2.13 โครงสร้างของสเตปปิงมอเตอร์แบบแม่เหล็กถาวร

การกระตุ้นให้สเตปปิงมอเตอร์ทำงานสามารถทำได้ดังนี้

1. การกระตุ้นที่ละเฟส
2. การกระตุ้นแบบย่อยสเตป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 การกระตุ้นที่ละเฟสจะป้อนไฟกระตุ้นให้กับสเตปปิงมอเตอร์ที่ละเฟส

ลำดับที่	เฟสที่1	เฟสที่2	เฟสที่3	เฟสที่4
1	●			
2		●		
3			●	
4				●
5	●			
6		●		
7			●	
8				●

ตารางที่ 2.2 การกระตุ้นแบบย่อยสเตปจะป้อนไฟกระตุ้นให้กับสเตปปิงมอเตอร์

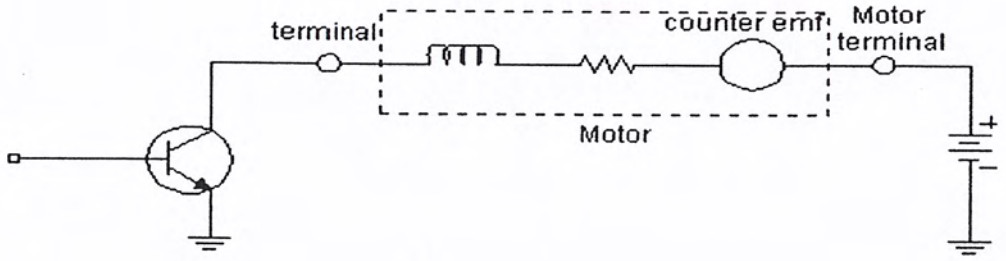
ลำดับที่	เฟสที่1	เฟสที่2	เฟสที่3	เฟสที่4
1	●			
2	●	●		
3		●		
4		●	●	
5			●	
6			●	●
7				●
8	●			●

2.2.5 วงจรขับสำหรับสเตปปิงมอเตอร์

การพิจารณาถึงหลักการในการออกแบบวงจรขับกำลังให้กับขดลวดของสเตปปิงมอเตอร์ควรจะต้องคำนึงถึงองค์ประกอบหลายอย่างด้วยกัน เนื่องจากการหมุนของสเตปปิงมอเตอร์จะทำให้เกิดแรงดันไฟฟ้าย้อนกลับ (Back emf) ซึ่งจะมีทิศทางตรงกันข้ามกับแหล่งจ่ายแรงดันไฟฟ้า

โดยสามารถเขียนรูปวงจรสมมูลย์ (Equivalent circuit) ในหนึ่งเฟสของสเตปปิงมอเตอร์ได้ดังแสดงในภาพที่ 2.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.14 แสดงวงจรสมมูลย์ของสเตปปิงมอเตอร์

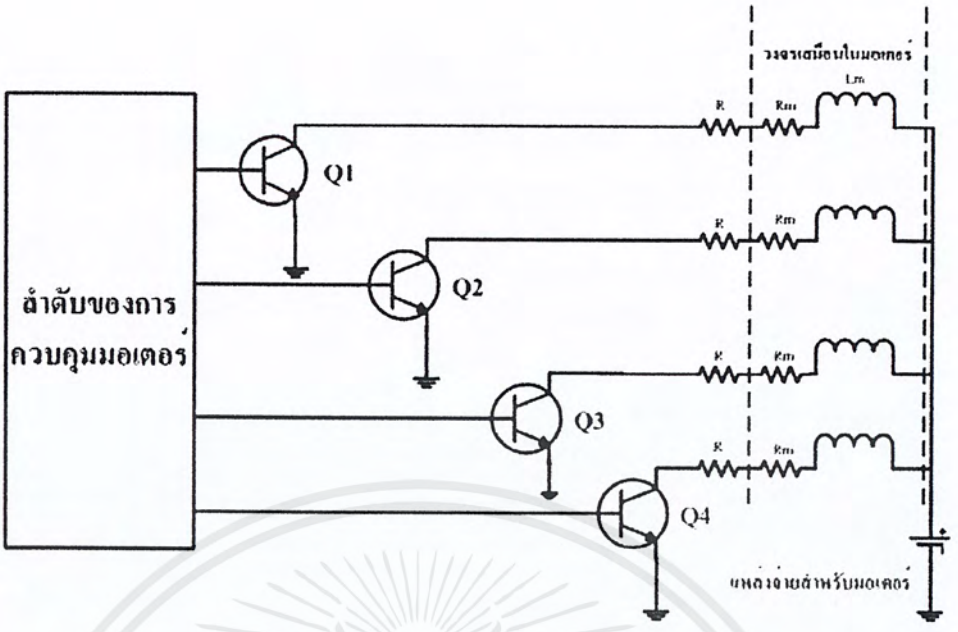
โดยทั่วไป สเตปปิงมอเตอร์ได้ถูกออกแบบให้ทนความร้อนได้สูงถึงประมาณที่ 100 องศา แต่ในการใช้งานจริงจะถูกใช้งานที่เงื่อนไขต่ำกว่าจุดอิ่มตัวที่กำหนดมา โดยทั่วๆไปการขับมอเตอร์นั้นต้องเลือกใช้อุปกรณ์ที่กินกระแสสูง เนื่องจากชุดขดลวดในแต่ละเฟสของสเตปปิงมอเตอร์จะต้องมีการนำและหยุดนำกระแสอยู่ตลอดเวลา ดังนั้นจึงจำอย่างยิ่งเป็นต้องออกแบบเพื่อป้องกันความเสียหายที่จะอาจเกิดขึ้นกับทรานซิสเตอร์กำลังจากแรงดันยอดแหลม(Spike Voltage)ที่เกิดจากการเหนี่ยวนำของกระแสในขดลวด (Inductive Turn Off Spike Voltage) และการเสียสภาพฉับพลันของแรงดัน ซึ่งการออกแบบวงจรขับและวงจรป้องกันสามารถทำได้ดังนี้

1. การใช้ตัวต้านทานต่ออนุกรมกับวงจรขับ
2. การใช้วงจรขับแบบใช้แรงดันไฟฟ้าสองระดับ
3. การใช้วงจรขับแบบชอปเปอร์
4. การใช้วงจรขับแบบบริดจ์

2.2.5.1 การใช้ตัวต้านทานต่ออนุกรมกับวงจรขับ

การต่อความต้านทานอนุกรมกับวงจรขับเป็นที่รูปแบบง่ายๆและราคาถูกสุดทำการเพิ่มประสิทธิภาพการทำงานของสเตปปิง คือการต่อความต้านทานอนุกรมเข้ากับชุดขดลวดแต่ละเฟสของสเตปปิงมอเตอร์เป็นการเพิ่มความต้านทานภายนอก ทำให้ค่าเวลาคงตัวของวงจรลดลงไป ทำให้เวลาของการเพิ่มและลดระดับของกระแสลดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.15 แสดงการใช้ความต้านทานต่ออนุกรมกับขดลวดของสเตปปิงมอเตอร์

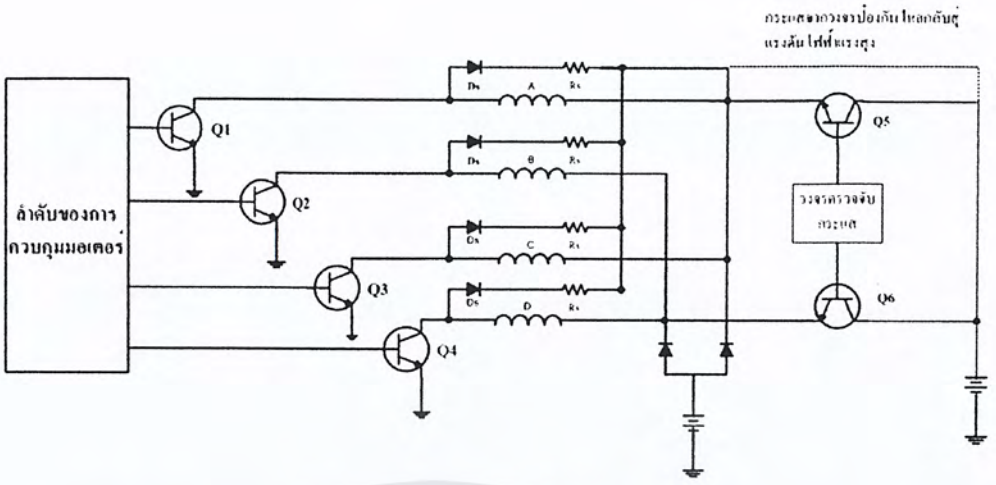
ข้อเสียของการต่อความต้านทานอนุกรมคือ การสูญเสียกำลังงานของต้านทานภายนอกมีมากขึ้น ซึ่งเป็นสาเหตุให้ระบบไม่มีประสิทธิภาพเพียงพอ

2.2.5.2 การใช้วงจรขับแบบใช้แรงดันไฟฟ้าสองระดับ

เป็นวิธีการขับที่มีประสิทธิภาพกว่ารูปแบบของการต่อความต้านทานอนุกรม ซึ่งใช้แรงดันไฟฟ้าระดับสูงสำหรับการยกระดับกระแสและตัดต่อไปยังแรงดันไฟฟ้าระดับต่ำเมื่อถึงอัตรากระแสที่ต้องการหรือถึงเวลาที่กำหนด กระแสที่อยู่ในขดลวดนั้นจะไหลผ่านไอโอดตัว D1 โดยแรงดันไฟฟ้าระดับต่ำจะทำงานแทน วิธีการนี้เรียกว่าการขับแบบแรงดันไฟฟ้าคู่หรือการขับแบบสองระดับ เมื่อพิจารณาการลดลงของเวลาในการเปลี่ยนระดับกระแสของระบบการขับแบบสองระดับเปรียบเทียบกับแบบการต่อความต้านทานอนุกรม แรงดันไฟฟ้าตกคร่อมขดลวดจะยังคงอยู่ทั้งหมดจนกระทั่งกระแสเพิ่มขึ้นจนถึงระดับกระแสที่ต้องการ

แม้ว่าค่าเวลาคงตัวจะไม่ลดลงเหมือนในแบบการต่อความต้านทานอนุกรม ค่าการยกระดับกระแสค่อนข้างจะมีลักษณะเป็นเชิงเส้นหลังจากแรงดันไฟฟ้าระดับสูงจะตัดต่อไปสู่แรงดันไฟฟ้าระดับต่ำ กระแสจะตกลงและเกิดแรงดันย้อนกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 แสดงวงจรขับสเตปปิงมอเตอร์แบบแรงดันไฟฟ้าสองระดับ

ขณะที่ทรานซิสเตอร์ตัวที่ Q5 และตัวที่ Q6 ไม่ทำงาน ไดโอดตัวที่ D1และไดโอดตัวที่ D2 ของแรงดันไฟฟ้าระดับต่ำนำกระแสอยู่ เมื่อแรงดันไฟฟ้าระดับสูงทำงาน ทรานซิสเตอร์ Q5 และ Q6 นำกระแสไดโอด D1 ต้องหยุดนำกระแสทันที และแหล่งจ่ายไฟของแรงดันไฟฟ้าระดับต่ำจะกลายเป็นระบบภายนอก การทำงานจะเป็นลักษณะนี้สลับกันไปตลอดช่วงการทำงาน โดยทั่วไปไดโอดกระแสสูงจะมีค่าเวลาในการคืนตัวช้า ดังนั้นในกรณีนี้จึงมีความจำเป็นต้องใช้ไดโอดชนิดคืนตัวเร็ว (Fast Recovery Diode) เพื่อให้เกิดกระแสยอดแหลม (Current Spike) ในขณะที่มีแหล่งจ่ายไฟอยู่ในระดับที่ปลอดภัย โดยที่แหล่งจ่ายไฟที่มีระดับแรงดันไฟฟ้าต่ำ

ดังนั้นควรน่าออกแบบให้มีระดับกระแสที่เพียงพอในการต้านกระแสย้อนกลับ

2.2.5.3 การใช้วงจรขับแบบชอปเปอร์

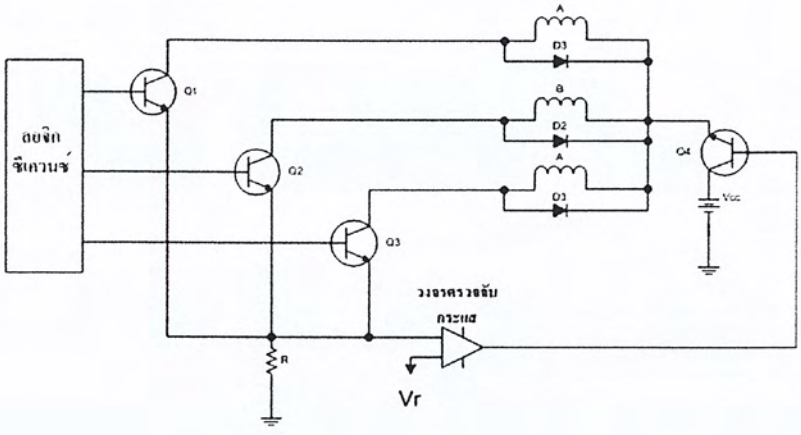
เป็นวงจรขับอีกแบบหนึ่งที่ใช้ขับมอเตอร์ได้อย่างมีประสิทธิภาพสูง นั้นคือใช้การมอดูเลตความกว้างพัลส์ (Pulse Width Modulation) หรือการมอดูเลตความถี่ (Frequency Modulation) ควบคุมกระแสเฉลี่ยที่ให้กับสเตปปิงมอเตอร์วงจรขับแบบชอปเปอร์อย่างง่าย ๆ

จากรูปที่ 2.17 เมื่อเฟสใดเฟสหนึ่งของมอเตอร์ทำงาน ทรานซิสเตอร์ Q4 จะเปิดและปิดไปสู่แรงดันไฟฟ้าระดับสูงตลอดเวลา การใช้แหล่งจ่ายไฟที่มีระดับแรงดันไฟฟ้าสูงเป็นสาเหตุให้กระแสในแต่ละเฟสเพิ่มระดับขึ้นอย่างทันทีทันใด

เช่นเดียวกับในวงจรขับแบบแรงดันไฟฟ้าสองระดับนั้น คือมีวงจรตรวจระดับกระแสเพื่อจะปรับตั้งค่าการหยุดทำงานของทรานซิสเตอร์ตัวที่ Q4 เมื่อกระแสมาถึงระดับที่ต้องการ และเมื่อกระแสลดลงต่ำกว่ากระแสขีดเริ่มเปลี่ยน (Threshold Current) ทรานซิสเตอร์ตัวที่ Q4 จะทำงานอีกครั้งหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



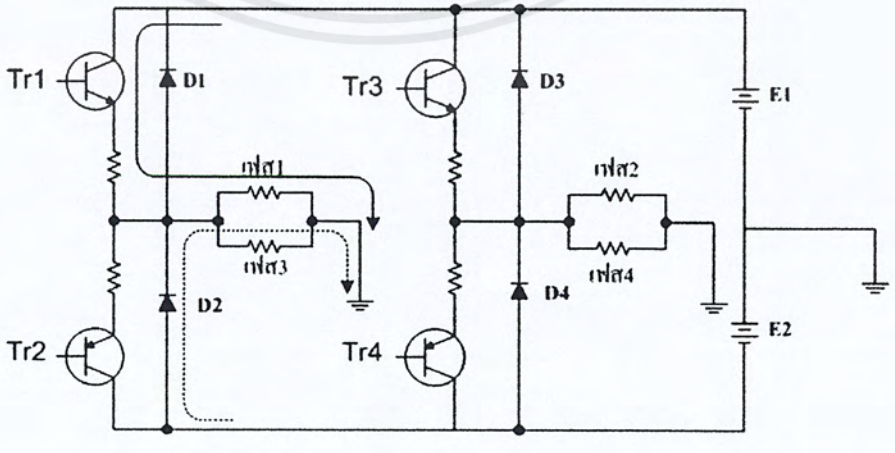
รูปที่ 2.17 แสดงวงจรขับแบบชอปเปอร์

การทำงานของวงจรนี้จะยอมให้กระแสเฉลี่ยคงอยู่ในระดับที่ต้องการ ในกรณีนี้ความถี่ของการตัดช่วงจะอยู่ระหว่างหนึ่งถึงสามสิบกิโลเฮิรตซ์ขึ้นอยู่กับค่าเวลาคงตัวของการพันขดลวดของมอเตอร์และฮีสเตอริส (Hystereis) ของวงจรตรวจระดับกระแส ความถี่ของการตัดช่วงมันจะปรับตั้งอยู่สูงกว่าช่วงที่กำหนดเพื่อหลีกเลี่ยงสัญญาณ รบกวนความถี่ต่ำซึ่งเกิดจากการสั่นของมอเตอร์

2.2.5.4 การใช้วงจรขับแบบบริดจ์

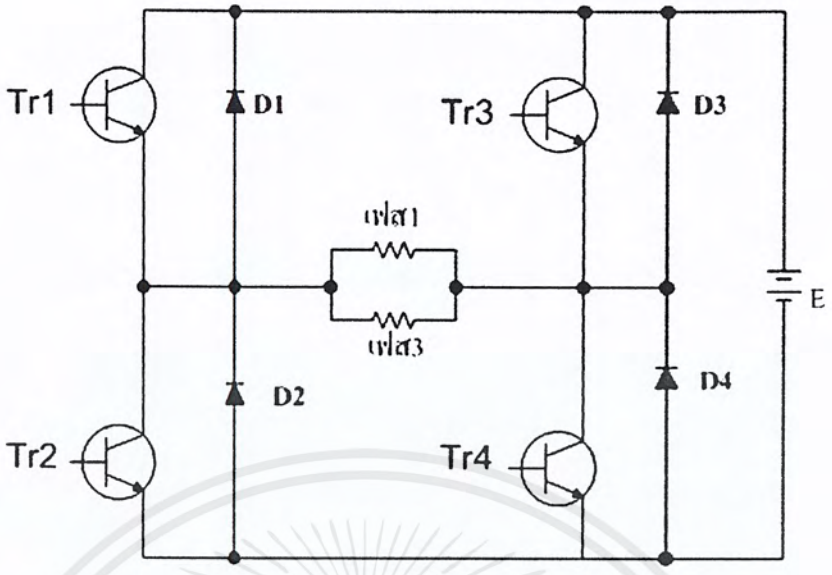
ประสิทธิภาพในการใช้งานของสเตปปิงมอเตอร์จะสูงขึ้น ถ้าควบคุมการไหลของกระแสให้ไหลได้สองทิศทาง

ข้อดีคือให้แรงบิดเพิ่มขึ้นอีก 20–35%ของการใช้วงจรขับแบมีกระแสไหลทางเดียว วงจรขับแบบบริดจ์มีสองชนิด คือแบบที่ใช้สองแหล่งจ่ายและแบบที่ใช้แหล่งจ่ายเดียว



ภาพที่ 2.18 แสดงวงจรขับแบบบริดจ์ ที่ใช้สองแหล่งจ่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.19 แสดงวงจรขับแบบบริดจ์ที่ใช้แหล่งจ่ายเดียว

2.3 ไมโครคอนโทรลเลอร์ MCS-51

โครงสร้างและสถาปัตยกรรมของไมโคคอนโทรลเลอร์MCS-51 แบบเฟลช(flash memory) ที่ใช้จะพูดถึงไมโคคอนโทรเลอร์ตระกูล MCS-51 ซึ่งมีหน่วยความจำแบบแฟลช (flash memory) ผลิตโดย Atmel corporation มีเบอร์ขึ้นต้นด้วย AT89

เหตุผลที่ใช้ไมโครคอนโทรลเลอร์แบบนี้ในการเรียนรู้เพื่อใช้งานไมโครคอนเลอร์MCS-51 มีด้วยกันหลายประการดังนี้

ก. หน่วยความจำโปรแกรมภายในตัวไมโครคอนโทรลเลอร์เป็นแบบแฟลช ซึ่งทำให้สามารถลบและเขียนใหม่ได้นับพันครั้ง จึงสามารถใช้งานในรูปแบบของไมโครคอนโทรลเลอร์ชิปเดี่ยวไม่ต้องใช้หน่วยความจำภายนอก ส่งผลให้สามารถใช้งานพอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ได้มีอย่างเต็มประสิทธิภาพ

ข. ต้นทุนและเวลาในการพัฒนาระบบไมโครคอนโทรลเลอร์ลดลงอย่างมาก เนื่องจากไม่ต้องใช้เครื่องมือพัฒนาจำพวกอีมูเลเตอร์และเครื่องโปรแกรมอีพรอม

ค. บริษัทผู้ผลิตได้ทำการผลิตชิปไมโครลคอนโทรลเลอร์ตระกูลนี้ ออกมาหลายเบอร์มาก แต่จะมีความสามารถแตกต่างกันไป ทำให้มีทางเลือกในการใช้งานสูง

ง. ด้วยการใช้หน่วยความจำภายในตัวไมโครคอนโทรเลอร์ ทำให้สามารถป้องกันการคัดลอกข้อมูลของหน่วยความจำโปรแกรมได้อย่างดี

จ. ในบางเบอร์ของไมโครคอนโทรลเลอร์ที่ผลิตโดย Atmel สามารถทำการโปรแกรมข้อมูลในหน่วยความจำโปรแกรมได้โดยที่ไมต้องถอดตัวไมโครลคอนโทรลเลอร์ออกมาทำการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมใหม่หรือเรียกว่าการโปรแกรมในวงจรหรือในระบบ(In-system programming) ทำให้การพัฒนาและการซ่อมบำรุง ตลอดจนการปรับปรุงหรืออัปเกรดข้อมูลในหน่วยความจำโปรแกรมทำได้อย่างสะดวก ภายใต้งบประมาณที่ไม่สูงมากนัก

ฉ. ชุดคำสั่งและสถาปัตยกรรมพื้นฐานเหมือนกับไมโครคอนโทรล MCS-51 ของผู้ผลิตอื่น ไม่ว่าจะเป็นอินเทล ซีเมนต์ หรือ ดัสลัส

2.3.1 คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 อนุกรม AT89xx

ก. เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต

ข. ภายในหน่วยความจำโปรแกรมเป็นแบบแฟลชสามารถลบและเขียนใหม่ได้พันครั้ง

ค. หน่วยความจำพื้นฐานเป็นหน่วยความจำแบบแรมโดยในบางเบอร์จะมีหน่วยความจำลักษณะแบบอีพรอมเพิ่มเติม

ง. ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานได้ทั้งอินพุตและเอาต์พุต

จ. ไทเมอร์/เคาน์เตอร์ขนาด 16 บิตอย่างน้อย 2 ตัว

ฉ. สามารถรองรับแหล่งกำเนิดอินเตอร์รัปได้ 6 ประเภท

ช. สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์

ในภาพที่ 2.20 เป็นโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89Cxx จะเห็นได้ว่าโครงสร้างของ AT89Cxx จะเหมือนกับคอนโทรลเลอร์ตระกูล MCS-51 หากแตกต่างกันเฉพาะหน่วยความจำโปรแกรมแบบแฟรชที่เพิ่มเติมเข้ามา

หากเป็นตัวไมโครคอนโทรลเลอร์ในอนุกรม 87xx หน่วยความจำโปรแกรมภายในจะเป็นแบบอีพรอม และบางเบอร์โปรแกรมได้เพียงครั้งเดียว

สำหรับในภาพที่ 2.21 แสดงรูปโครงสร้างพื้นฐานของอนุกรม AT89Sxx จะเห็นได้ว่าส่วนประกอบเพิ่มเติมแตกต่างจาก AT89Cxx อยู่หลายส่วนอาทิวงจรของเชื่อมต่ออนุกรมแบบSPI ซึ่งในไมโครคอนโทรเลอร์อนุกรมนี้ใช้ในการเขียนข้อมูลลงในหน่วยความจำโปรแกรมโดยไม่ต้องถอดตัวชิปออกไปจากระบบวงจรหรือเรียกว่าการโปรแกรมภายในวงจร ไทเมอร์/เคาน์เตอร์ขนาด 16บิต ที่เพิ่มเติมเข้ามาอีกหนึ่งตัวเป็นไทเมอร์ 2 และวงจรวอตช์ด็อกใช้ในการตรวจสอบการทำงานผิดพลาดของซีพียู

ในตารางที่ 2.3 แสดงรายละเอียดข้อมูลเฉพาะของไมโครคอนโทรลเลอร์ตระกูล MCS-51 โดยแต่ละเบอร์ที่ Atmel ทำผลิตขึ้นมาและมีใช้งานอยู่ในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51

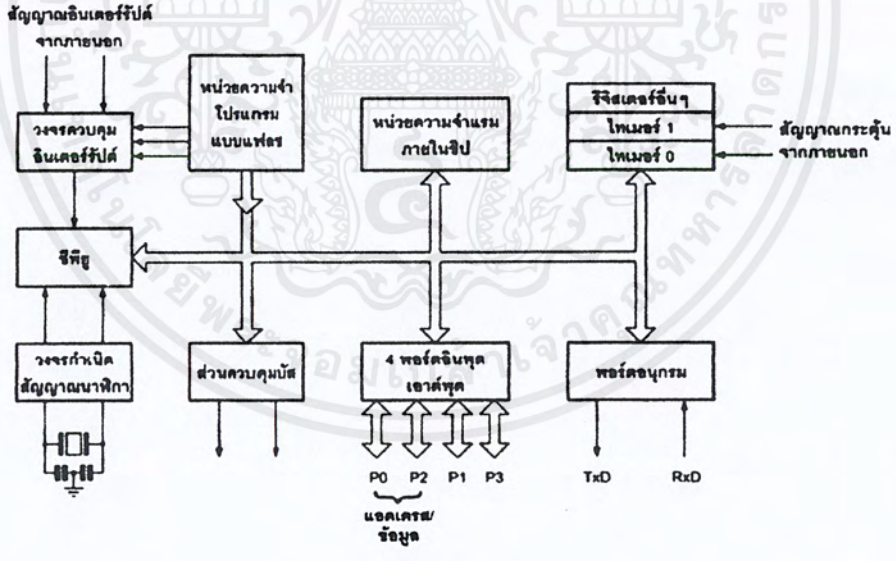
ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทุกเบอร์จะมีสถาปัตยกรรมและขาใช้งานพื้นฐานที่เหมือนกัน ดังแสดงในภาพที่ 2.23 โดยมีรายละเอียดขั้นต้นดังนี้

ก. ขา Vcc ใช้ต่อไฟเลี้ยง +5V

ข. ขา GND เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ

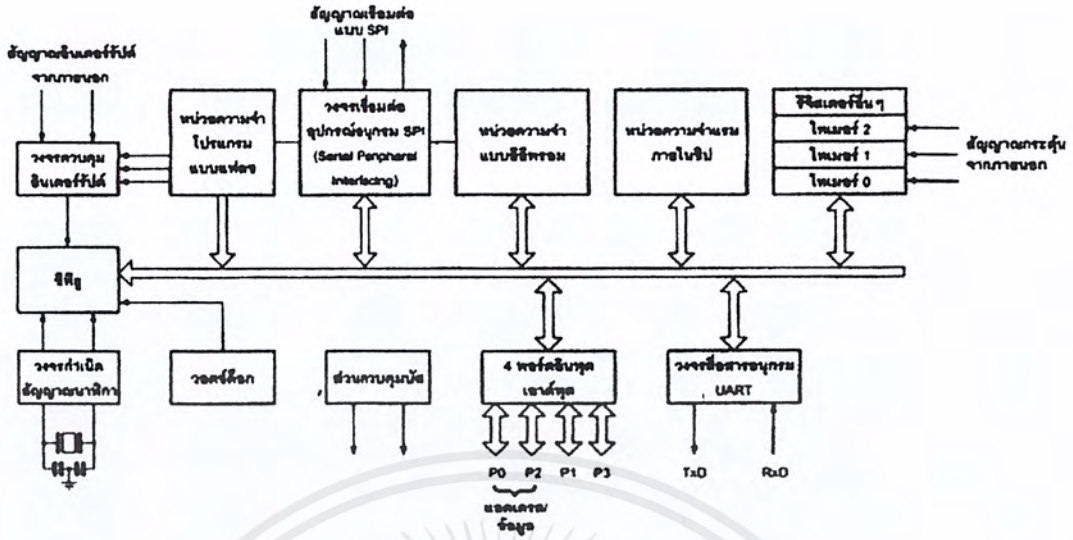
ค. ขาพอร์ต 0 (P0.0-P0.7) มี 8 ขา

แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำการได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลทำให้ขาพอร์ตนั้นมีสถานะปล่อยลอย(float) จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก(A0-A7)และขาข้อมูล(D0-D7)โดยใช้กระบวนการมัลติเพล็กเข้าช่วย เพื่อสลับการทำงานเป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล



รูปที่ 2.25 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89Cxx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.26 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89Sxx

ตารางที่ 2.3 รายละเอียดโดยสรุปบางส่วนของไมโครคอนโทรลเลอร์ MCS-51

เบอร์	หน่วยความจำโปรแกรม	หน่วยความจำข้อมูล	จำนวนไทเมอร์/เคาน์เตอร์ 16 บิต
AT89C1051	แบบแฟลช ขนาด 1 กิโลไบต์	แรม 64 ไบต์	1
AT89C2051	แบบแฟลช ขนาด 2 กิโลไบต์	แรม 128 ไบต์	2
AT89C51	แบบแฟลช ขนาด 4 กิโลไบต์	แรม 128 ไบต์	2
AT89C52	แบบแฟลช ขนาด 8 กิโลไบต์	แรม 256 ไบต์	3
AT89C55	แบบแฟลช ขนาด 20 กิโลไบต์	แรม 256 ไบต์	3
AT89S8252	แบบแฟลช ขนาด 8 กิโลไบต์	แรม 256 ไบต์ อีอีพรอม 2 กิโลไบต์	3
AT89S53	แบบแฟลช ขนาด 12 กิโลไบต์	แรม 256 ไบต์	3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาพอร์ต1 (P1.0-P1.7) มี 8ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำการได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย นอกจากนั้นในอนุกรม AT89Sxx จะใช้ขาP1.0 เป็นขาอินพุตสำหรับนับค่าของไทเมอร์2 และP1.1 เป็นขาอินพุททริกเกอร์ของไทเมอร์2 โดยในขณะที่ขา P1.4-P1.7 เป็นขาเชื่อมต่อแบบ SPI เพื่อทำการโปรแกรมข้อมูลในระบบโดยไม่จำเป็นต้องถอดออก

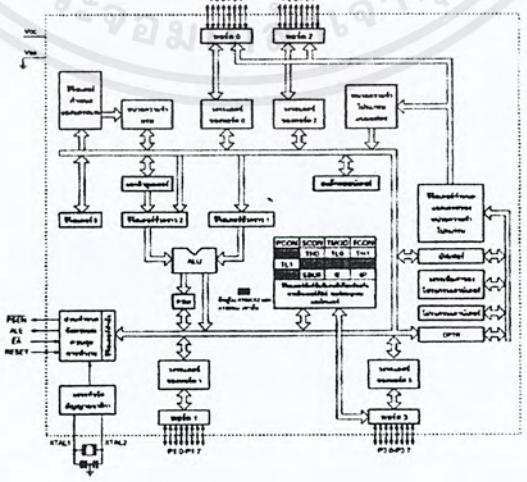
ขาพอร์ต2 (P2.0-P2.7) มี 8ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุตสามารถทำการได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลทำให้ขาพอร์ตนั้นมีสถานะปล่อยลอย(float) จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8-A15)

ขาพอร์ต3 (P3.0-P3.7) มี 8ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุตสามารถทำการได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลทำให้ขาพอร์ตนั้นมีสถานะปล่อยลอย(float) จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ต 3 ยังเป็นขาที่มีหน้าที่ใช้งานพิเศษ ดังมีรายละเอียดดังต่อไปนี้

P3.0 ใช้เป็นขาอินพุตสำหรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD

P3.1 ใช้เป็นขาอินพุตสำหรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD

P3.2 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 0 หรือ ขา INT0



ภาพที่ 2.27 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 2.23 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89C5x

P3.3 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือ ขา INT1

P3.4 ใช้เป็นขาอินพุตรับสัญญาณไทเมอร์จากภายนอกช่อง 0 หรือ ขา T0

P3.5 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือ ขา T1

P3.6 ใช้เป็นขาสัญญาณ WR ในกรณีใช้กับหน่วยความจำภายนอก

P3.7 ใช้เป็นขาสัญญาณ RD ในกรณีใช้กับหน่วยความจำภายนอก

ขารีเซต (Reset) ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณเพื่อรีเซตสถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย 2 แมชชีนไซเกิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างเป็นปกติ

ขา ALE /PROG (Address Latch Enable/Program pulse input) เป็นขาที่สำคัญมากของการควบคุมการแลตซ์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนี้ขานี้ยังใช้เป็นขาสำหรับรับพัลซ์ของการ โปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ MCS-51 ในรุ่นที่มีหน่วยความจำโปรกรมเป็นแบบอีพรอม

ขา PSEN (Program Store Enable) ขานี้ใช้ในการส่งสัญญาณเพื่อขอร้องติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกตัวไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปแกรมภายนอกตัวไมโครคอนโทรลเลอร์จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้งในแต่ละแมชชีนไซเกิล แต่ถ้าหากติดต่อกับหน่วยความจำข้อมูลภายนอก ขานี้จะไม่มีการส่งสัญญาณใดๆออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา EA/Vpp (External Access enable/Programming voltage input) ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าหากขานี้เป็น "0" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอกแต่ถ้าหากขานี้เป็น "1" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายในตัวของไมโครคอนโทรลเลอร์ นอกจากนี้ที่ขานี้ยังใช้เป็นขาอินพุตสำหรับรับแรงดันไฟสูงสำหรับโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชต้องการแรงดันสำหรับการโปรแกรม +12 V

ขา XTAL1 และ XTAL2 เป็นสำหรับต่อคริสตอลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

2.3.3 โครงสร้างและทำงานของพอร์ต

ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีพอร์ตให้ใช้งานทั้งสิ้น 4 พอร์ตคือ พอร์ตที่0 ถึงพอร์ตที่3 แต่ละพอร์ตมีขนาด 8 บิต เป็นพอร์ตแบบ 2 ทิศทาง กล่าวคือ สามารถเป็นได้ทั้งอินพุตสำหรับรับสัญญาณข้อมูลเข้าและเอาท์พุตสำหรับส่งสัญญาณข้อมูลออกทุกพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ใด ดังสรุปได้ในตารางที่ 2.3

ในภาพที่2.24 แสดงวงจรภายในของแต่ละพอร์ตของคอนโทรลเลอร์ MCS-51 แบบแฟลช โดยในภาพที่ 2.24 (ก) เป็นวงจรของพอร์ตที่0 โดยวงจรแลตช์แต่ละบิตและในแต่ละพอร์ตก็คือวงจรดีฟลิฟลอปนั่นเอง การอ่านค่าสถานะของพอร์ตและสถานะของวงจรแลตช์สามารถกระทำได้อย่างอิสระด้วยสัญญาณที่แยกจากกัน นั่นคือสัญญาณอ่านข้อมูลจากขาพอร์ต

และสัญญาณอ่านข้อมูลจากวงจรแลตช์ ส่วนการเขียนข้อมูลมายังพอร์ตต้องส่งสัญญาณมายังขา CLK ของดีฟลิฟลอปในขณะที่ข้อมูลจะผ่านมาทางขาบัสข้อมูลภายในเข้าสู่ขา D ของดีฟลิฟลอปที่พอร์ตนี้มีวงจรมัลติเพล็กซ์สำหรับกำหนดลักษณะการทำงานของพอร์ตว่าต้องการใช้งานเป็นขาพอร์ตอินพุตเอาท์พุตปกติหรือใช้การติดต่อกับหน่วยความจำภายนอกไมโครคอนโทรลเลอร์ เนื่องจากที่ขาพอร์ต0 ไม่มีวงจรพูลอัปภายใน หากมีการนำพอร์ต0 ไปใช้งานเป็นอินพุตจะต้องต่อตัวต้านทานพูลอัปภายนอกเข้าที่ขาพอร์ต 0 ทุกขาด้วย

ในภาพที่ 2.24 (ข) เป็นวงจรของพอร์ต 1 ซึ่งมีลักษณะโดยทั่วไปคล้ายกับพอร์ตที่0 หากแต่ไม่มีวงจรมัลติเพล็กซ์แล้ว เนื่องจากพอร์ตนี้จะไม่ใช้ในการติดต่อกับหน่วยความจำภายนอก แต่จะมีวงจรพูลอัปภายในที่แต่ละบิตของพอร์ตนี้แทน ส่วนรายละเอียดของวงจรพูลอัป

ในภาพที่ 2.24 (ค) เป็นวงจรภายในของพอร์ต2 จะคล้ายกับพอร์ต0 มาก จะต่างเพียงมีวงจรพูลอัปเพิ่มเข้ามาส่วนในรูปที่ 2.24(ง) เป็นวงจรภายในของพอร์ตที่3 จะเห็นว่าคล้ายพอร์ตที่1 มีการเพิ่มเติมวงจรบัฟเฟอร์และวงจรอินพุตเอาท์พุตเมื่อทำงานในฟังก์ชั่นพิเศษเข้ามา

เนื่องจากพอร์ตที่3 สามารถนำไปใช้งานในหน้าที่พิเศษได้ทุกขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3.1 การใช้งานเป็นพอร์ตอินพุต

เนื่องจากพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชสามารถเป็นได้ทั้งอินพุตและเอาต์พุต ดังนั้นจึงมีความจำเป็นอย่างยิ่งต้องความเข้าใจถึงการกำหนดลักษณะการทำงานให้แก่พอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ในการกำหนดให้เป็นพอร์ตอินพุตต้องเริ่มต้นด้วยการเขียนข้อมูล"1"มาที่แต่ละบิตของพอร์ตที่ต้องการใช้งานเป็นอินพุต เพื่อหยุดการทำงานของเฟตที่ใช้ในการขับสัญญาณเอาต์พุตของบิตนั้นๆ ทำให้ขาสัญญาณของพอร์ตที่เชื่อมต่อเข้ากับวงจรพูลอัป ภายในโดยตรงจะส่งผลให้ขาพอร์ตนั้นมีลอจิกเป็น "1" สามารถรับสัญญาณลอจิก "0" จากอุปกรณ์ที่ต้องต่อภายนอกให้ง่าย สัญญาณข้อมูลจากอุปกรณ์ภายนอกจะถูกส่งเข้ามาแล้วเก็บไว้ในวงจรบัฟเฟอร์ภายในพอร์ตแล้วรอให้ซีพียูมาอ่านเข้าไป

เมื่อเป็นเช่นนี้แล้วอุปกรณ์ภายนอกที่เชื่อมต่อกับอินพุตของพอร์ตของไมโคคอนโทรลเลอร์ควรกำหนดให้ทำงานในสภาวะลอจิกเป็น "0" จะดีและสะดวกที่สุด (ซึ่งในปัจจุบันอุปกรณ์อินพุตที่เชื่อมต่อไมโครคอนโทรลเลอร์แทบทั้งหมดทำงานที่ลอจิก "0" แล้ว)

ตารางที่ 2.4 หน้าที่พิเศษของพอร์ต1 ในไมโครคอนโทรลเลอร์ MCS-51

ขา	เบอร์	หน้าที่พิเศษ
P1.0	AT89C52/AT89Sxx	ขา T2 เป็นขาอินพุตนับค่าของไทเมอร์/เคาน์เตอร์ 2 และเป็นขา
P1.1	AT89C52/AT89Sxx	และควบคุมทิศทางของสัญญาณ
P1.4	AT89Sxx	ขา SS (Slave Select) เป็นขาเลือกการติดต่อในกรณีที่เป็นอุปกรณ์สเลฟ ในระบบการติดต่อแบบ SPI
P1.5	AT89Sxx	ขา MOSI (Master data output, Slave data input) ใช้ในการติดต่อกับพอร์ต SPI
P1.6	AT89Sxx	ขา MISO (Master data input, Slave data output) ใช้ในการติดต่อกับพอร์ต SPI
P1.7	AT89Sxx	ขา SCK (Master clock output) เป็นขาสัญญาณนาฬิกาของการติดต่อกับพอร์ต SPI

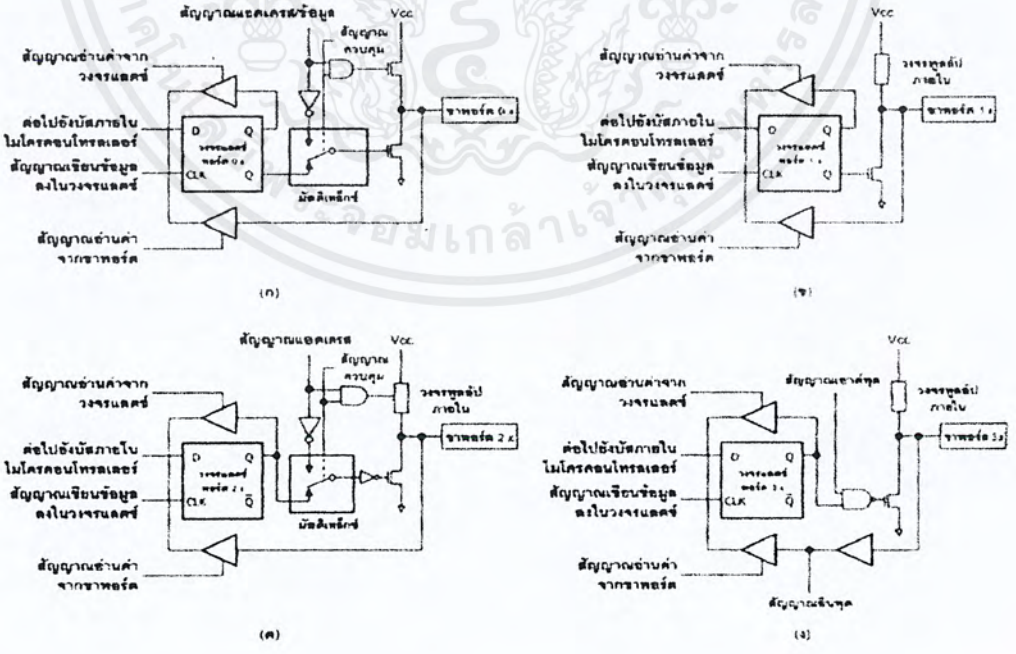
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3.2 การใช้งานเป็นพอร์ตเอาต์พุต

โดยปกติแล้ว ขาพอร์ตจะกำหนดให้มีลักษณะเป็นเอาต์พุตอยู่แล้ว ดังนั้นจึงสามารถส่งข้อมูลออกไปได้อย่างง่ายดายและตรงไปตรงมา กล่าวคือเมื่อต้องการส่งข้อมูล “0” ออกไปทางเอาต์พุตก็ให้เขียนข้อมูล “0” ไปยังวงจรแลตช์ ซึ่งก็จะส่งต่อไปขับเฟต ทำให้เฟตทำงานที่ขาพอร์ตที่กำหนดให้ทำงานก็จะเกิดลอจิก “0” ขึ้น ในทางตรงข้ามหากต้องการส่งข้อมูล “1” ออกไปก็ให้เขียนข้อมูล “1” ไปยังวงจรแลตช์วงจรก็จะหยุดทำงาน ทำให้ที่ขาพอร์ตเชื่อมต่อกับวงจรพูลอัปภายในเกิดเป็นลอจิก “1” ที่ขาพอร์ตนั้น ซึ่งจะคล้ายกับการกำหนดให้เป็นขาอินพุตมาก เพียงแต่แตกต่างกันที่กระบวนการในการเคลื่อนย้ายข้อมูล โดยถ้าเป็นอินพุตจะมีสัญญาณมาอ่านข้อมูลที่บัฟเฟอร์ แต่ถ้าเป็นเอาต์พุตจะไม่มีการอ่านข้อมูลที่บัฟเฟอร์แต่อย่างใด เว้นแต่กรณีที่ต้องการตรวจสอบข้อมูลที่ส่งออกมาทางเอาต์พุต

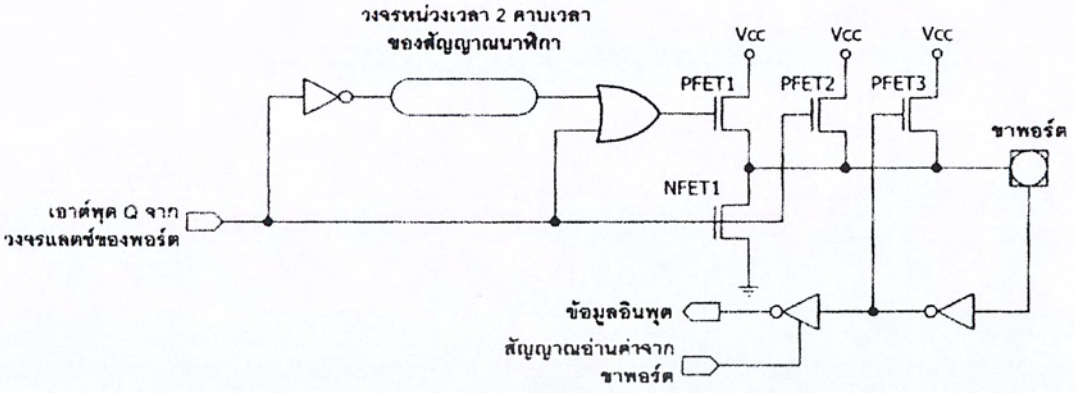
เมื่อใช้งานเป็นพอร์ตเอาต์พุต โดยแต่ละขา (หรือละบิต) ของแต่ละพอร์ตมีความสามารถในการจ่ายกระแสหรือที่เรียกว่า **กระแสซอร์ส (source current)** ได้สูงสุด 10 mA และทุกขารวมกันในแต่ละพอร์ต (ทั้ง 8 บิต) สูงสุด 26 mA สำหรับพอร์ต 0 และ 15 mA สำหรับพอร์ต 1-3 ในกรณีที่ใช้งานทุกพอร์ตเอาต์พุตจะสามารถจ่ายกระแสได้รวมกันสูงสุด 71 mA

ดังนั้นในการใช้งานเป็นพอร์ตเอาต์พุตเพื่อไม่ให้เกิดปัญหาเกี่ยวกับความสามารถในการจ่ายกระแสควรต่อวงจรสบัฟเฟอร์ทางเอาต์พุตเพื่อช่วยในการขับกระแสอีกทางหนึ่ง



รูปที่ 2.24 วงจรภายในของพอร์ตทุกพอร์ตในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจรพูลอัปประกอบด้วยเฟตชนิดพีแชนเนล 3 ตัวคือ PFET1-PFET3 โดย NFET1 จะทำงานเมื่อได้รับลอจิก "1" จากขา Q และหยุดทำงานเมื่อได้รับลอจิก "0" วงจรพูลอัปจะเริ่มต้นทำงานเมื่อ NFET1 ได้รับลอจิก "1" PFET1 จะทำงานนานประมาณ 2 คาบเวลาของสัญญาณนาฬิกาภายใน หลังจากที่เกิดการเปลี่ยนแปลงจากลอจิก "0" เป็นลอจิก "1" ในขณะที่ PFET1 ทำงาน จะทำให้ PFET3 ทำงานตามไปด้วย ทำให้เกิดการพูลอัปขาพอร์ต

รูปที่ 2.25 วงจรพูลอัปภายในพอร์ตไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและพัฒนาชุดอุปกรณ์ที่ต่อร่วมที่ใช้กับมอเตอร์

ระบบขับเคลื่อนสเตปปิงมอเตอร์ที่ใช้การควบคุมโดยการกระตุ้นด้วยเทคนิคแบบมินิสเตป สามารถทำให้การหมุนของ สเตปปิงมอเตอร์เป็นไปอย่างราบเรียบและต่อเนื่องเทียบเคียงได้ราวกับมอเตอร์ไฟฟ้ากระแสตรง ทั้งแบบแยกขดกระตุ้นและมอเตอร์ไฟฟ้ากระแสตรงแบบแม่เหล็กถาวร และสามารถลดสัญญาณรบกวนและการซ่อมบำรุง โดยที่กำลังงานสูญเสียในรูปของความร้อนนั้นก็ลดลงเมื่อเทียบกับสเตปปิงมอเตอร์โดยทั่วๆไป

3.1 หลักการออกแบบชุดขับเคลื่อนสเตปปิงมอเตอร์

หลักการออกแบบโดยทั่วไปๆสามารถแบ่งออกเป็น 3วิธีคือ

1. การออกแบบการป้อนแบบฟูลสเตป (FULL STEP)
2. การออกแบบการป้อนแบบฮาล์ฟสเตป (HALF STEP)
3. การออกแบบการป้อนแบบมินิสเตป (MINI STEP)

3.1.1 การออกแบบการป้อนแบบฟูลสเตป (FULL STEP)

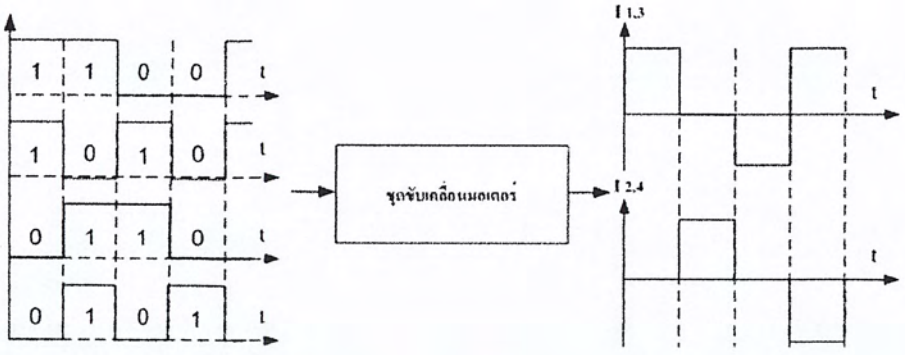
ซึ่งยังสามารถแบ่งลักษณะย่อยออกเป็น 2ลักษณะอีกคือ

1. ลักษณะแบบกระตุ้นเฟสเดียว (One Phase On)
2. ลักษณะแบบกระตุ้นสองเฟส (Two Phase On)

3.1.1.1 ลักษณะแบบกระตุ้นเฟสเดียว (One Phase On)

การออกแบบการป้อนพัลส์แบบกระตุ้นเฟสเดียวคือ การที่ป้อนพัลส์ให้แก่มอเตอร์ที่ละหนึ่งพัลส์ให้แก่สเตปปิงมอเตอร์ ซึ่งเป็นการออกแบบลักษณะแบบกระตุ้นเฟสเดียวนั้นง่ายที่จะใช้งานแต่มี ข้อเสียที่ว่าการหมุนของสเตปปิงมอเตอร์ไม่มีความละเอียดมากนักจึงสามารถใช้งานได้อย่างจำกัด ซึ่งมีเพียง 4สเตปโดยลักษณะการป้อนพัลส์จะเป็นไปตามภาพที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

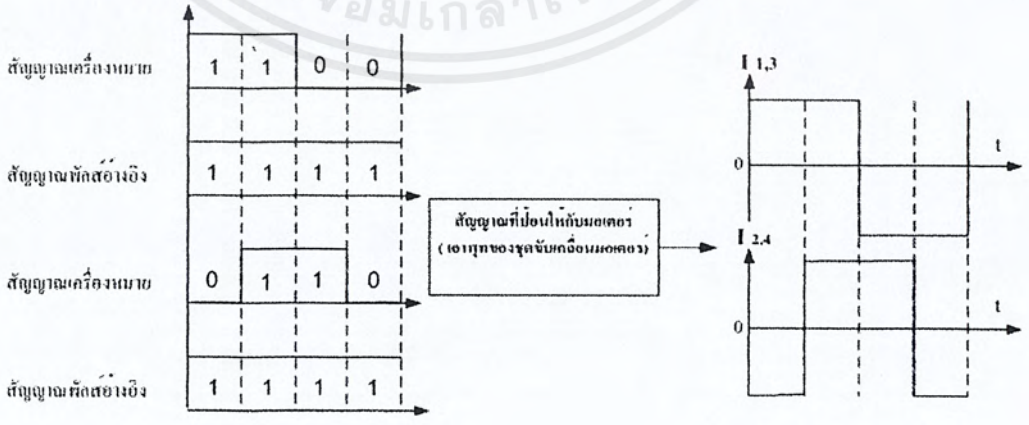


ภาพที่ 3.1 สัญญาณที่ใช้ป้อนชุดขับเคลื่อนมอเตอร์โดยการแบบพัลส์เฟสเดียว

ภาพที่ 3.2 แสดงลักษณะของการหมุนมอเตอร์แบบพัลส์เฟสเดียว

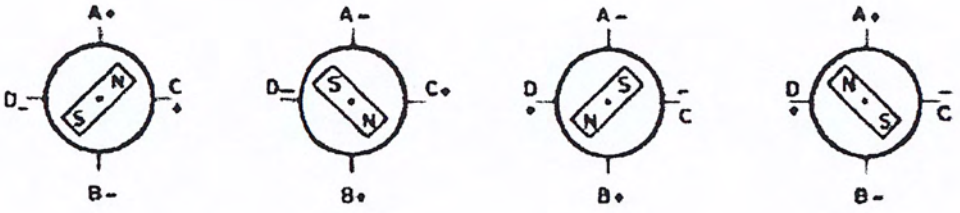
3.1.1.2 แบบกระตุ้นสองเฟส (Two Phase On)

การออกแบบการป้อนพัลส์แบบกระตุ้นสองเฟส จะต้องนำสัญญาณอิงอ้างเข้ามาควบคุมสัญญาณเครื่องเครื่องหมาย โดยลักษณะการป้อนพัลส์แบบกระตุ้นสองเฟสจะมีหลายสเตปกว่าการการป้อนพัลส์แบบกระตุ้นเฟสเดียวโดยลักษณะการป้อนพัลส์จะเป็นไปตามภาพที่ 3.3



ภาพที่ 3.3 สัญญาณที่ใช้ป้อนชุดขับเคลื่อนมอเตอร์โดยการแบบพัลส์สองเฟส

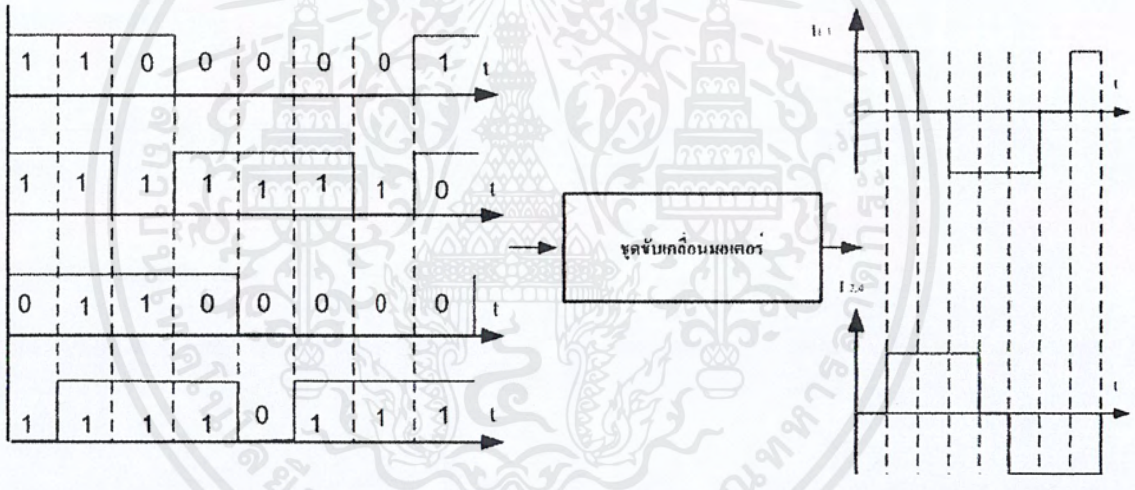
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.4 แสดงลักษณะของการหมุนมอเตอร์แบบพัลส์สองเฟส

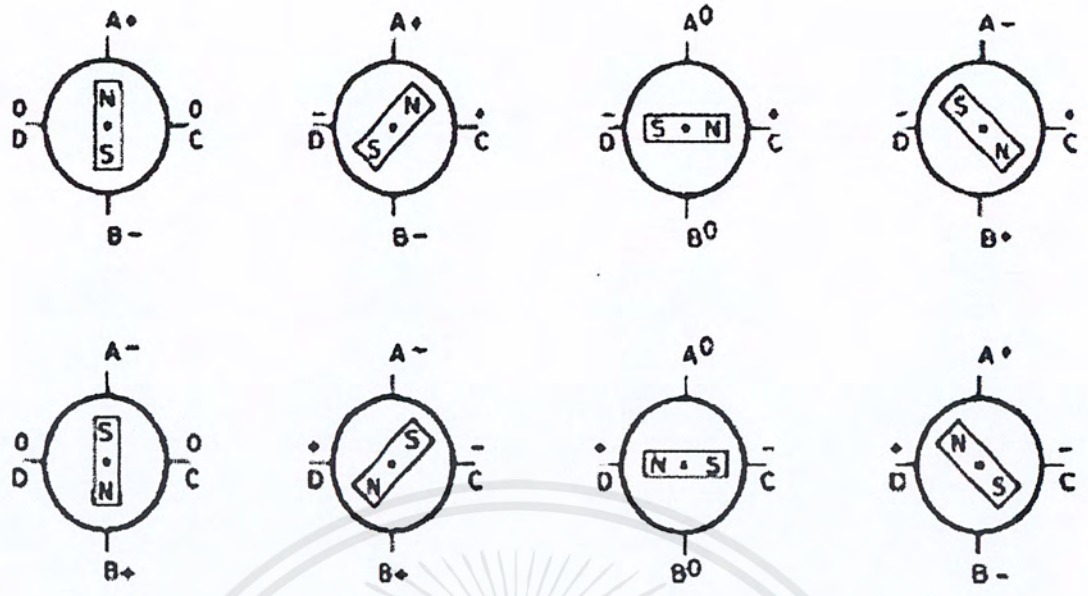
3.1.2 การออกแบบการป้อนฮาล์ฟสเตป (HALF SETEP)

การออกแบบการป้อนแบบฮาล์ฟสเตปนั้น ออกแบบเพื่อที่จะทำให้สเตปปิงมอเตอร์นั้นจะมีลักษณะหมุนที่ต่อเนื่องมากขึ้น ซึ่งจะทำให้งานของสเตปปิงมอเตอร์นั้นสามารถใช้งานได้อย่างกว้าง ลักษณะการป้อนแบบฮาล์ฟสเตปรูปคลื่นจะเป็นไปตามภาพที่ 3.5



รูปที่ 3.5 สัญญาณที่ใช้ป้อนชุดขับเคลื่อนมอเตอร์โดยการฟังชั่นแบบฮาล์ฟสเตป

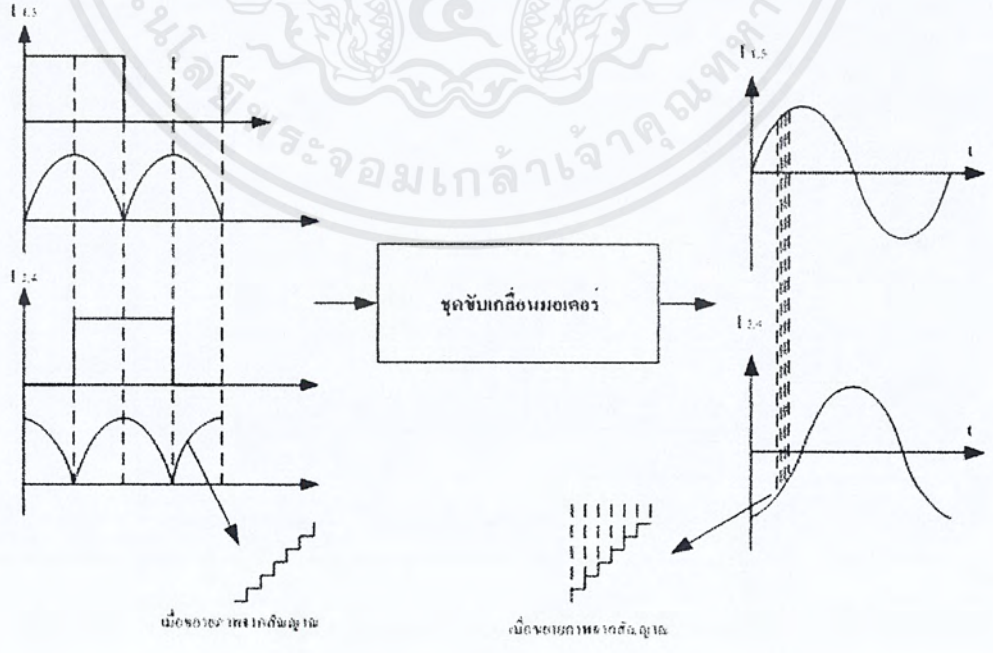
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.6 แสดงลักษณะของการหมุนมอเตอร์ฟังชั่นแบบฮาล์ฟสเตป

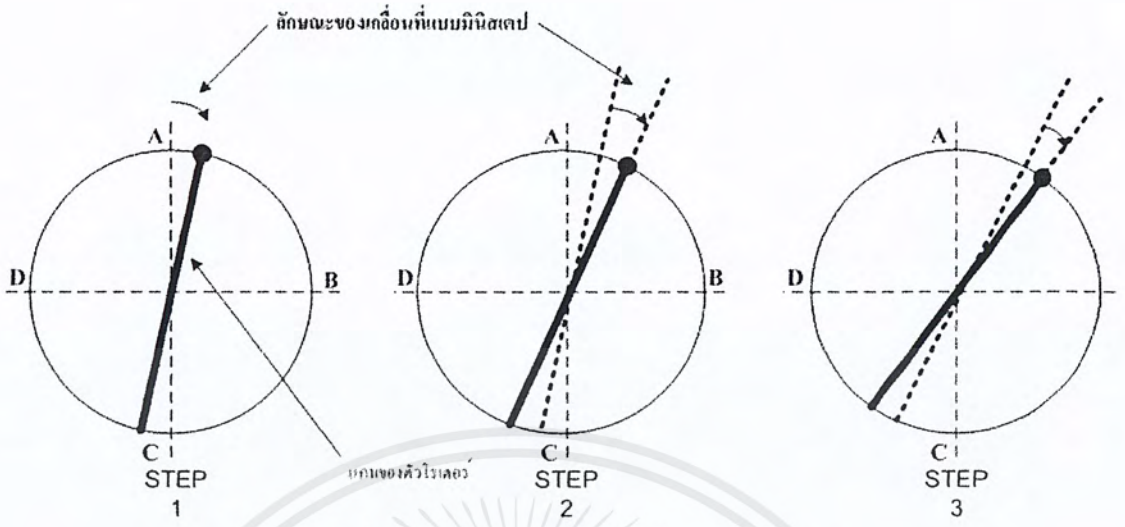
3.2.3 การออกแบบการป้อนแบบมินิสเตป (MINI STEP)

ลักษณะการออกแบบมินิสเตปตามรูปคลื่นชนิดนี้จะมีลักษณะการหมุนต่อเนื่องมากที่สุดจึงทำให้การออกแบบมอเตอร์ที่ต้องการใช้ การหมุนแบบต่อเนื่องสูงมากๆราวกับมอเตอร์กระแสตรง และยังมีความต้องการที่ควบคุมตำแหน่งของมอเตอร์ให้แม่นยำ ดังนั้นจึงมีความจำเป็นอย่างมากที่ต้องใช้ฟังชั่นมินิสเตปมาป็นสัญญาณอ้าง โดยลักษณะรูปคลื่นแบบมินิสเตปเป็นไปตามภาพที่ 3.7



ภาพที่ 3.7 แสดงสัญญาณที่ใช้ป้อนชุดขับเคลื่อนมอเตอร์แบบฟังชั่นมินิเตป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.8 แสดงลักษณะของการหมุนมอเตอร์ฟังชั่นแบบมินิสเตป

3.2 หลักการทำงานชุดของระบบขับเคลื่อน

ระบบขับเคลื่อนที่ออกแบบและพัฒนาขึ้นจากภาพที่3.9ที่ต่อแกนเพลาติดกับตัวโรเตอร์ของมอเตอร์เป็นตัวควบคุมจังหวะการกระตุ้น โดยที่สัญญาณแบบมินิสเตปที่ถูกถอดรหัสออกมาจะใช้ควบคุมการกระตุ้นในแต่ละสเตปการหมุนมอเตอร์นั้น ซึ่งจะถูกควบคุมให้สอดคล้องกับจังหวะการหมุนของโรเตอร์ในขณะนั้น ๆ โดยการใช้วงจรขับเคลื่อนซึ่งจะมีลักษณะแค่ทำการจังหวะเปิด-ปิดการกระตุ้นให้กับสเตปปิงมอเตอร์เท่านั้น

โดยวรจรขับเคลื่อนมอเตอร์ที่ออกแบบจะทำหน้าที่คล้ายกับซี่คอมมิวเตเตอร์และแปรงถ่านในมอเตอร์ไฟฟ้ากระแสตรงนั้นเอง ดังนั้นหากสัญญาณพัลส์ที่ได้จากการตัวไมโครคอนโทรลเลอร์นั้นความละเอียดสูงและสเตปย่อย ของสัญญาณจะทำการหมุนของสเตปปิงมอเตอร์จะราบเรียบมากขึ้นเท่านั้น บล็อกแสดงการทำงานของระบบขับเคลื่อนสามารถแสดงได้ตามภาพที่ 3.9

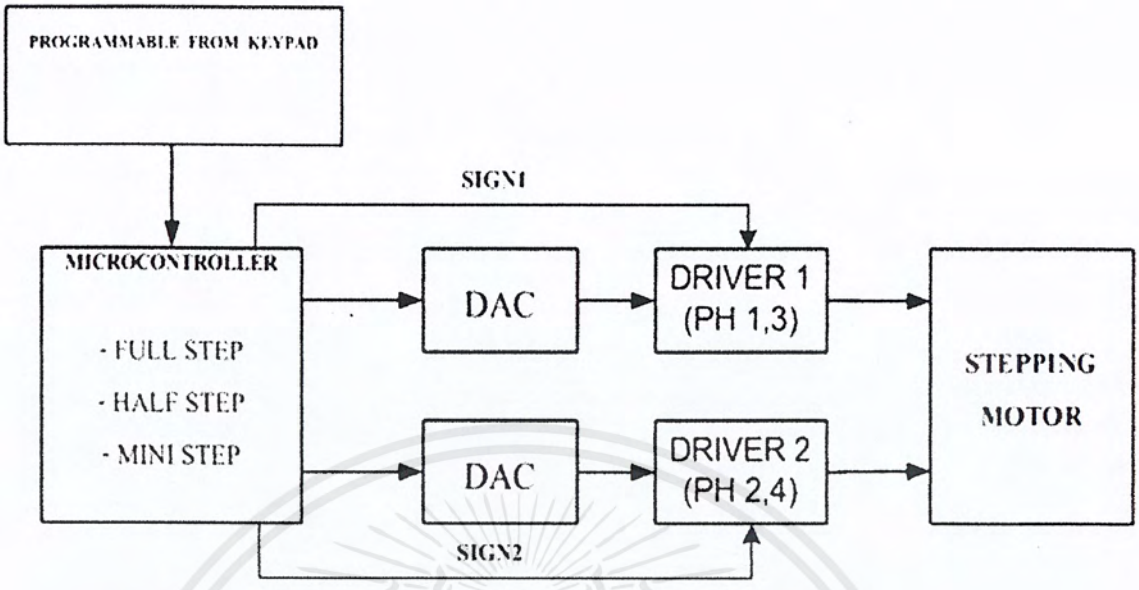
ในส่วนของการขับเคลื่อนจะประกอบไปด้วยส่วนย่อย ๆ ดังต่อไปนี้

1. ส่วนของโปรแกรมและวงจรแปลงดิจตอลเป็นอนาล็อก
2. วงจรขับเคลื่อนมอเตอร์

3.2.1 ส่วนของโปรแกรมและวงจรแปลงดิจตอลเป็นอนาล็อก

หน่วยจัดการข้อมูลและวงจรถอดรหัส ซึ่งเป็นส่วนที่รับสัญญาณจากตัวเข้ารหัสมาวิเคราะห์ทิศทางการหมุนและถอดค่ารหัสในขณะ เวลานั้นๆออกมาเป็น สัญญาณรูปคลื่นพัลส์ความละเอียดของรูปคลื่นพัลส์ที่ได้จะขึ้นอยู่กับความละเอียดของข้อมูลที่ได้จากตัวเข้ารหัสกับวงจรถอดรหัสที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.9 แสดงบล็อกการทำงานของระบบขับเคลื่อน

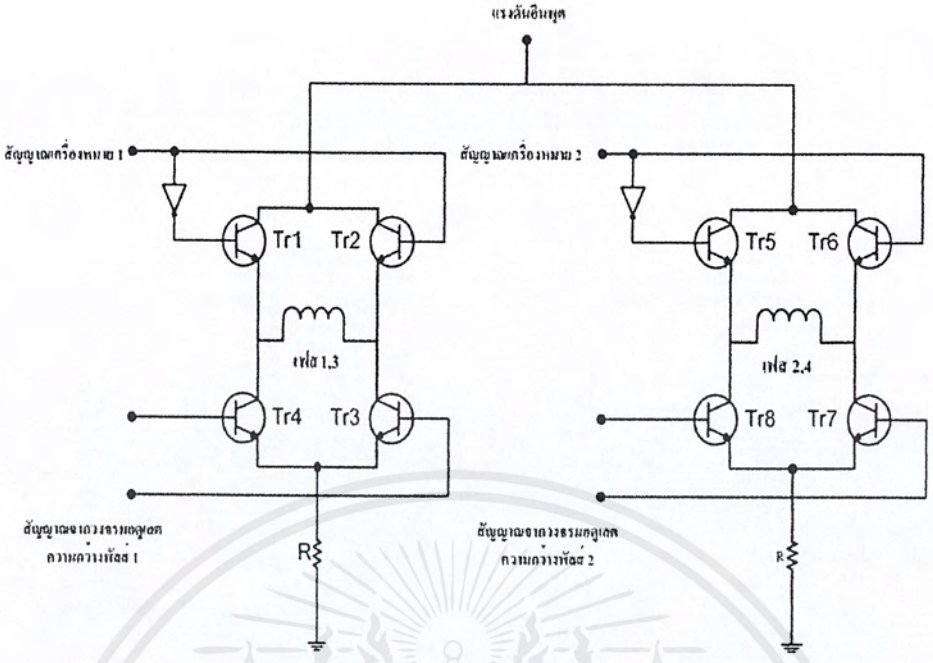
โดยใช้หลักการที่เรียกว่า **Network Resistance** ใช้ออปแอมทำการแปลงสัญญาณดิจิตอล (ซึ่งได้จากไมโครคอนโทรลเลอร์) แปลงเป็นสัญญาณอะนาล็อก เพื่อที่จะนำไปต่อกับชุดขับเคลื่อนมอเตอร์ต่อไป ลักษณะรูปคลื่นที่ได้ออกแบบไว้ในบทที่3 แล้วนำมาเขียนโปรแกรมโดยการทำงานของโปรแกรมสามารถเขียนเป็นได้ดังนี้ ในส่วนของโปรแกรมจะอยู่ในหน้าผนวก

3.2.2 วงจรขับเคลื่อนมอเตอร์

วงจรขับเคลื่อนแบบสองแกนขวางที่ใช้ในการขับมีจำนวน 2ชุดชุดหนึ่งขับขดลวดเฟส1กับเฟส3 ให้สัมพันธ์กับสัญญาณซายน์และอีกชุดหนึ่งกับขดลวดเฟส2กับเฟส4 ให้มีความสัมพันธ์กับสัญญาณโคซายน์

ซึ่งแต่ละชุดควบคุมการไหลของกระแสตามจังหวะการเปิด-ปิด(Switch)ของทรานซิสเตอร์ที่ถูกควบคุมการเปิด-ปิดตามค่าของรอบการทำงาน (Duty Cycle) ตามของสัญญาณพัลส์ที่ได้จากวงจรมอดูเลตตามความกว้าง

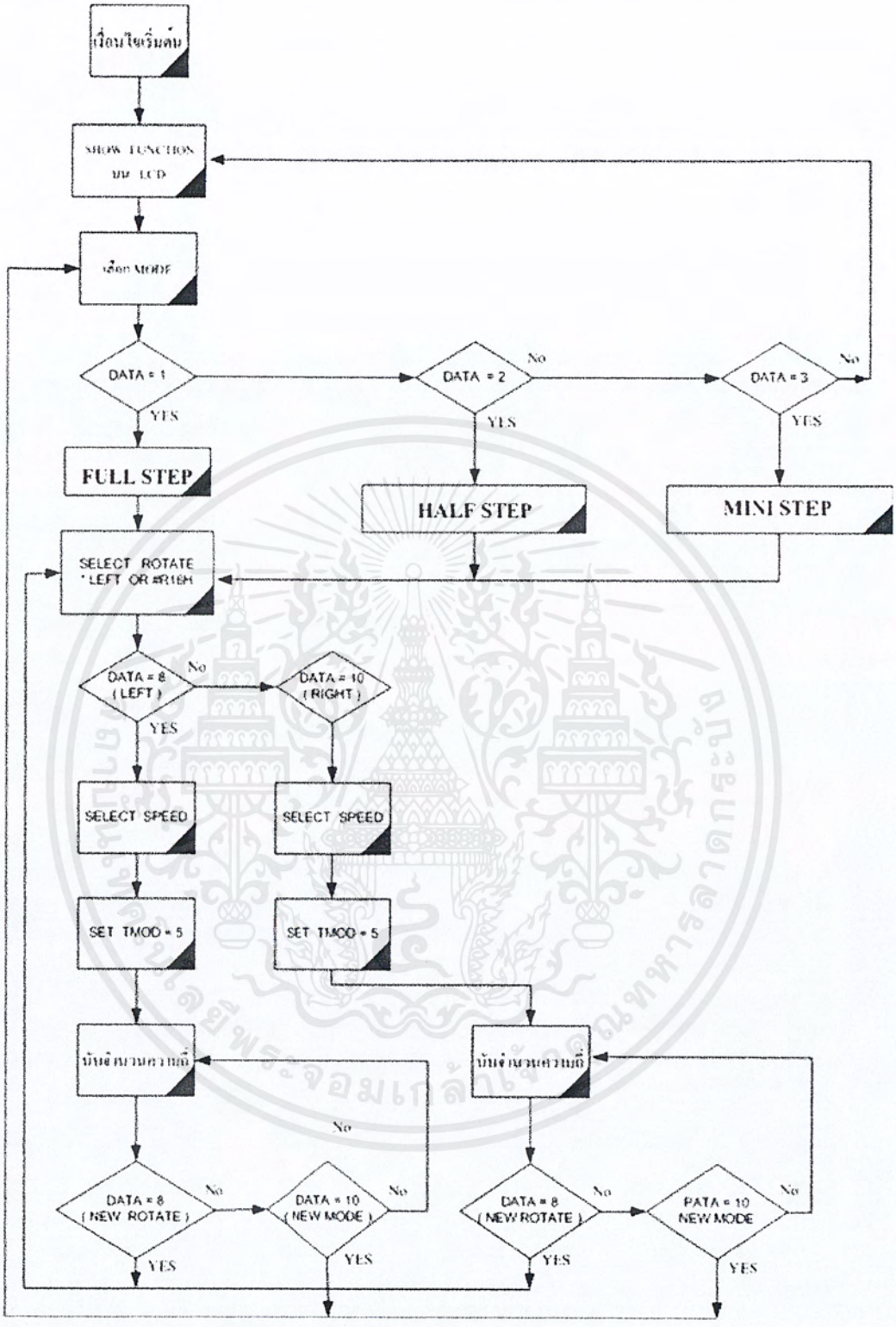
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.10 แสดงวงจรขับเคลื่อนแบบสองแกนขวาง

ภาพที่ 3.11 แสดงตัวอย่างสัญญาณ PMW ที่ใช้ป้อนชุดขับเคลื่อนมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.12 แสดงโฟลชาร์ตการทำงานของโปรแกรมที่นำมาใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

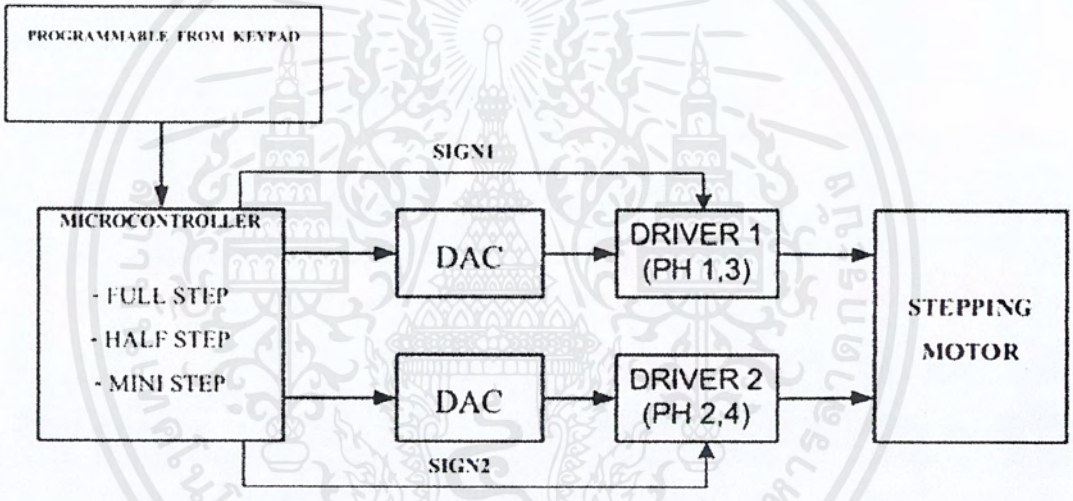
บทที่ 4

ผลการทดลอง

การทดลองระบบขับเคลื่อนที่ได้ออกแบบและพัฒนาในบทที่3 ผลการทดลองแบ่งออกเป็นลักษณะดังนี้

4.1 การทดลองผลการตอบสนองทางความเร็ว

การทดลองการทำงานของระบบขับเคลื่อนสเตปปิงมอเตอร์เพื่อดูผลการตอบสนองทางความเร็วสเตปปิงมอเตอร์ แบ่งออกเป็นดังนี้



ภาพที่ 4.1 แสดงลักษณะของระบบควบคุมโดยรวมที่ได้ออกแบบไว้

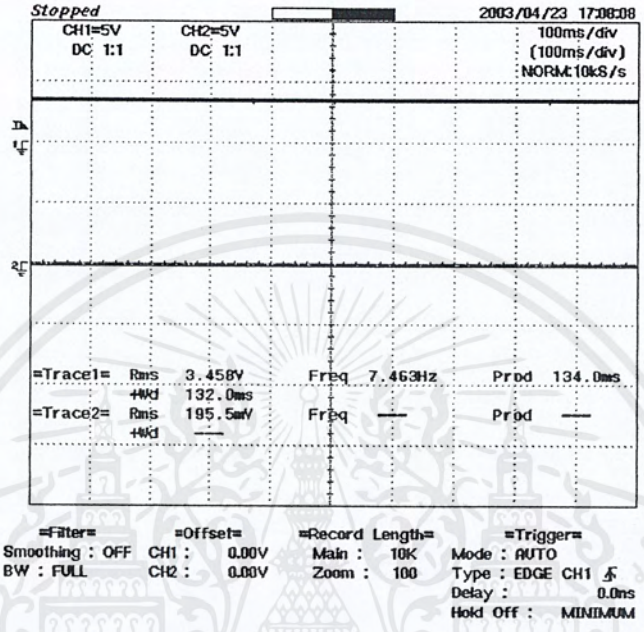
จากหลักการออกแบบชุดขับเคลื่อนสเตปปิงมอเตอร์ในบทที่3นั้น สามารถเขียนโปรแกรมให้ตัวไมโครคอนโทรเลอร์มซึ่งมีลักษณะฟังชั่นดังต่อไปนี้คือ

1. การป้อนแบบฟังชั่นฟูลสเตป (FULL STEP)
2. การป้อนแบบฟังชั่นฮาล์ฟสเตป (HALF STEP)
3. การป้อนแบบฟังชั่นมินิสเตป (MINI STEP)

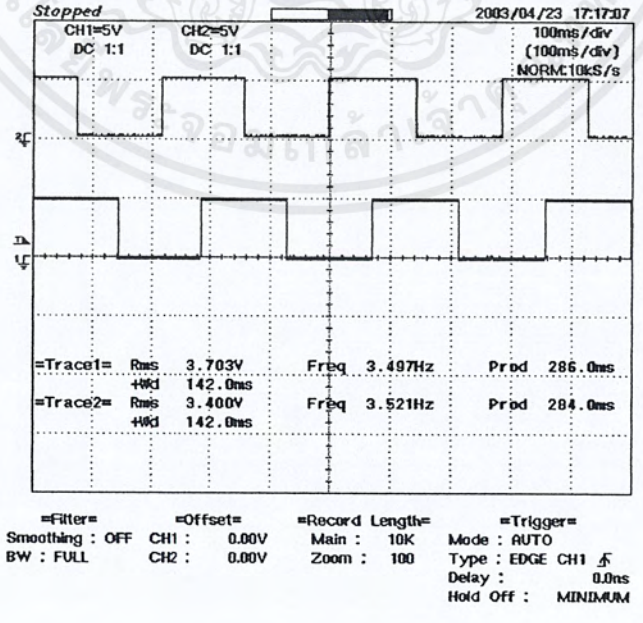
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1 การป้อนแบบฟังชั่นฟูลสเตป (FULL STEP)

จากในบทที่3 จากหลักการออกแบบการแบบฟูลสเตป สามารถวัดรูปคลื่นจากวงจรจริงได้ ดังแสดงใน ภาพที่ 4.2



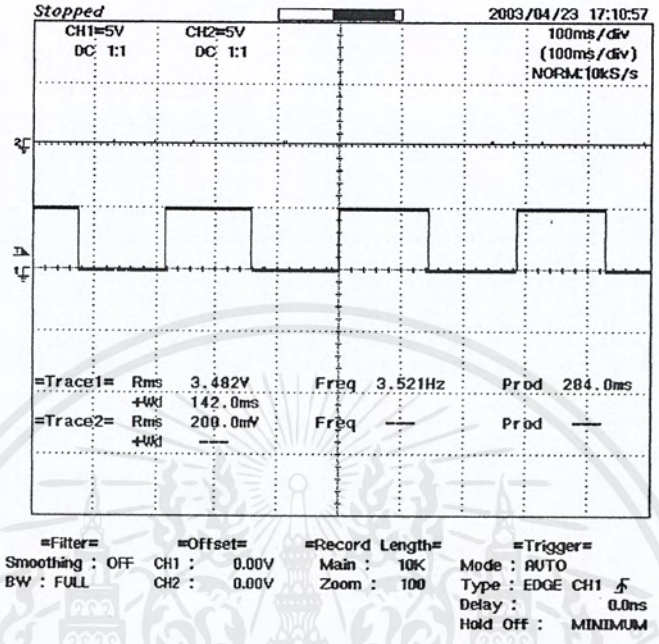
ภาพที่ 4.2 แสดงลักษณะรูปคลื่นสัญญาณอ้างอิงของแบบฟูลสเตป



ภาพที่ 4.3 แสดงลักษณะรูปคลื่นสัญญาณเครื่องหมาย

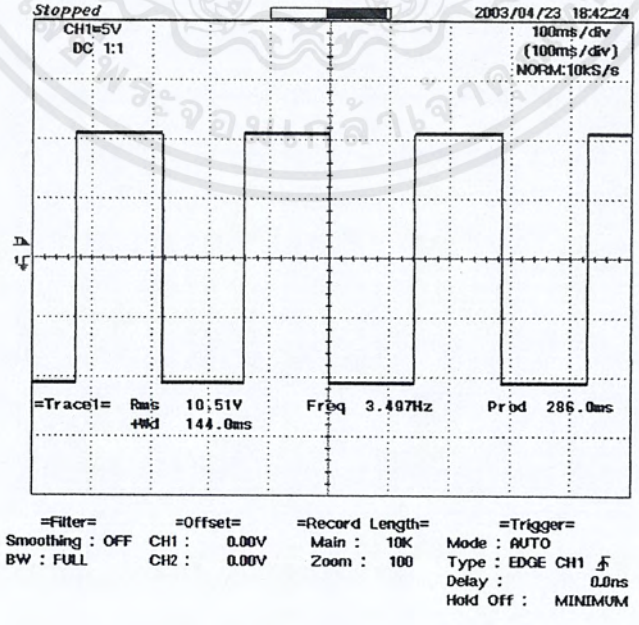
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยภาพที่ 4.5 แสดงถึงคลื่นสัญญาณเครื่องหมายที่1รูปบน และคลื่นสัญญาณเครื่องหมายที่ 2 รูปล่าง โดยที่รูปของสัญญาณมีลักษณะใกล้เคียงกับภาพที่ 3.2 ซึ่งเป็นรูปที่เราได้ออกแบบไว้ก่อน



ภาพที่ 4.4 แสดงรูปคลื่นสัญญาณอ้างอิงกับสัญญาณเครื่องหมาย

จากรูปเป็นการเปรียบเทียบกันระหว่างคลื่นสัญญาณอ้างอิงรูปบนกับสัญญาณเครื่องหมายรูปล่าง ซึ่งจะลักษณะใกล้เคียงกับภาพที่ 3.2 ที่ได้ออกแบบไว้

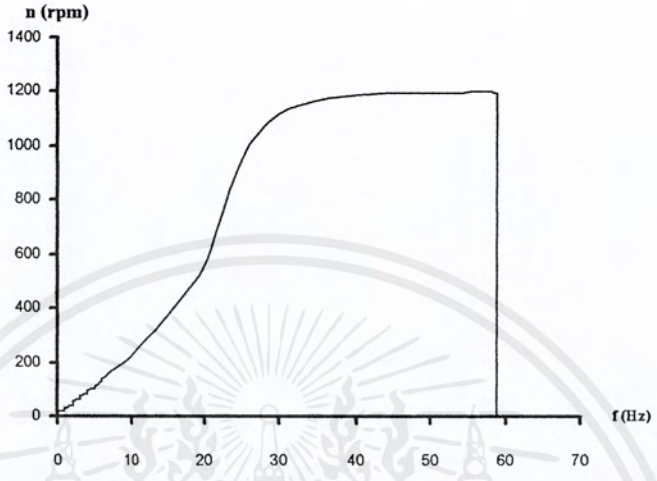


ภาพที่ 4.5 แสดงรูปคลื่นขณะที่จ่ายให้กับมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองโดยการป้อน ณ ความถี่ต่าง

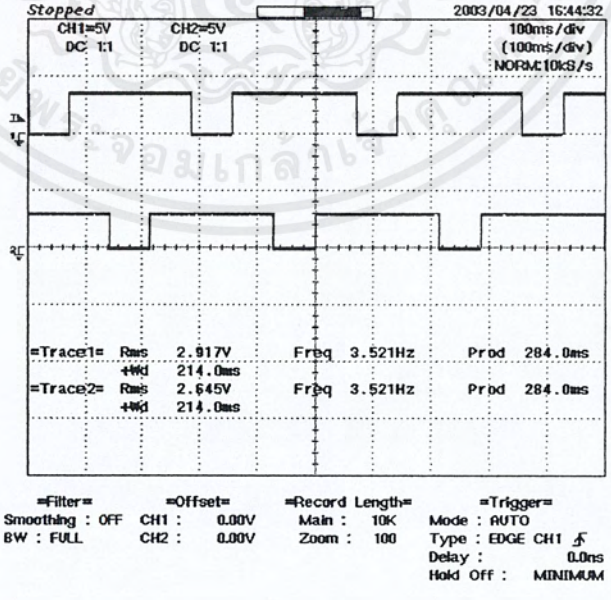
สามารถนำมาเขียนกราฟได้ดังนี้ โดยการป้อนความต่างๆจะมีผลต่อความเร็วรอบของมอเตอร์



ภาพที่ 4.6 แสดงกราฟความสัมพันธ์ความถี่กับความเร็วเมื่อการป้อนแบบฟูลสเตป (Full step)

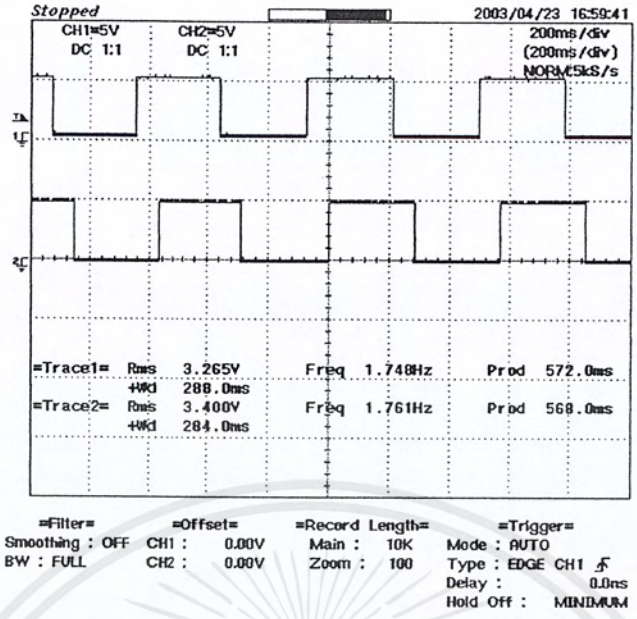
4.1.2 การป้อนแบบฟังชั่นฮาล์ฟสเตป (HALF STEP)

จากการออกแบบการป้อนแบบมินิสเตปซึ่งสามารถดูรูปคลื่นจากภาพที่ 4.7 แสดงรูปคลื่นที่ได้วัดจากวงจรจริง

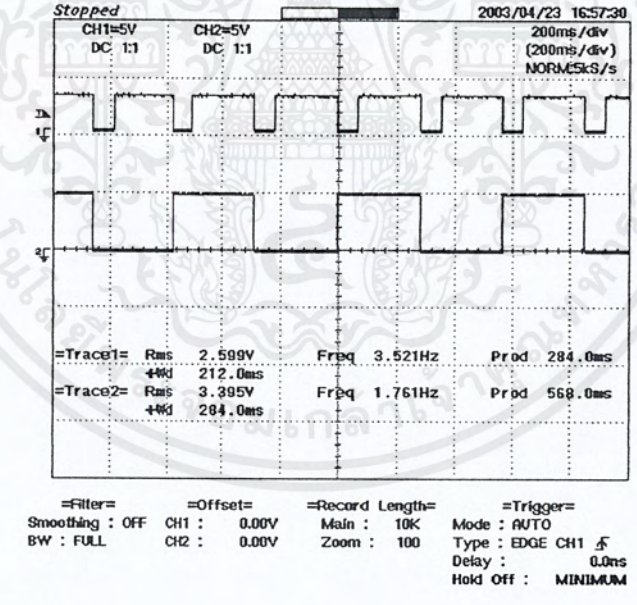


ภาพที่ 4.7 แสดงรูปคลื่นของสัญญาณอ้างอิงของการป้อนแบบฮาล์ฟสเตป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

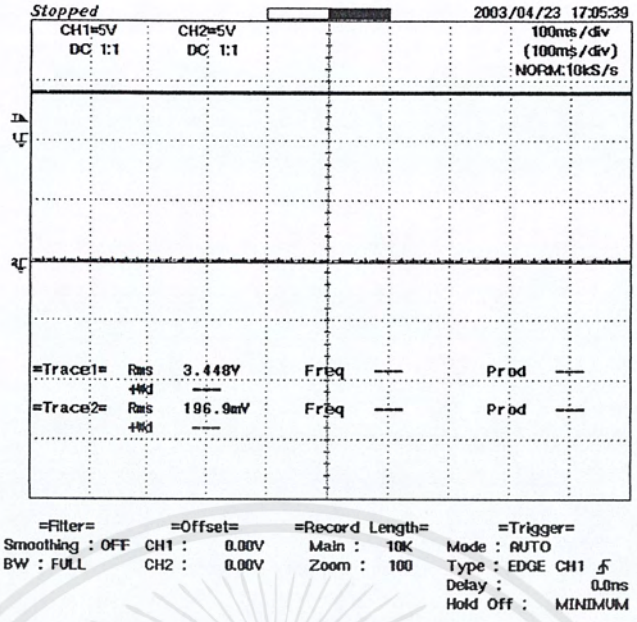


ภาพที่ 4.8 แสดงรูปคลื่นของสัญญาณเครื่องหมายของการป้อนแบบฮาล์ฟสเตป



ภาพที่ 4.9 แสดงรูปคลื่นของแสดงรูปคลื่นสัญญาณอ้างอิงกับสัญญาณเครื่องหมาย ของการป้อนรูปคลื่น แบบฮาล์ฟสเตป

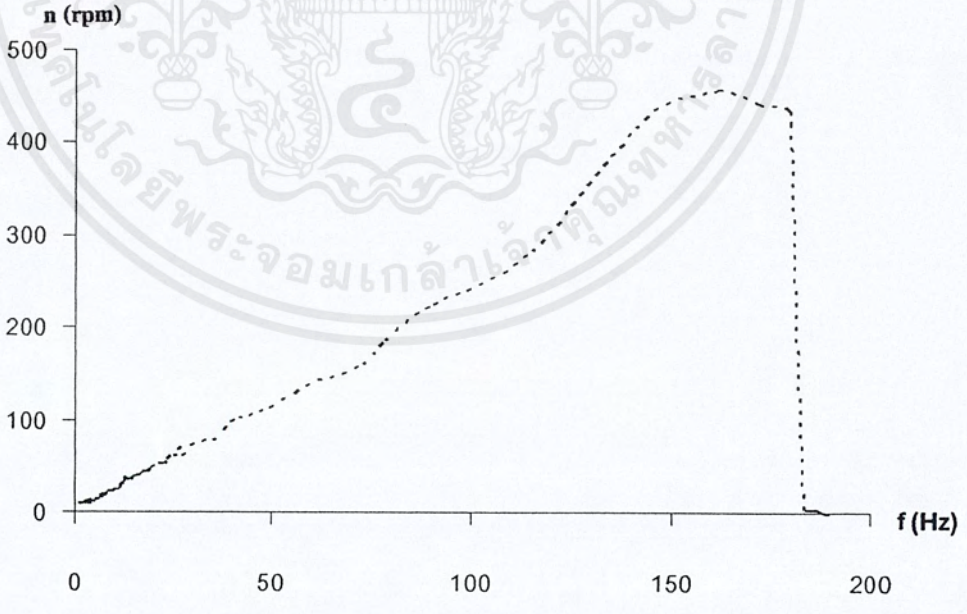
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.10 แสดงรูปคลื่นขณะที่จ่ายให้กับมอเตอร์

ผลการทดลองโดยการป้อน ณ ความถี่ต่าง

สามารถนำมาเขียนกราฟได้ดังนี้ โดยการป้อนความถี่ต่างๆจะมีผลต่อความเร็วรอบของมอเตอร์

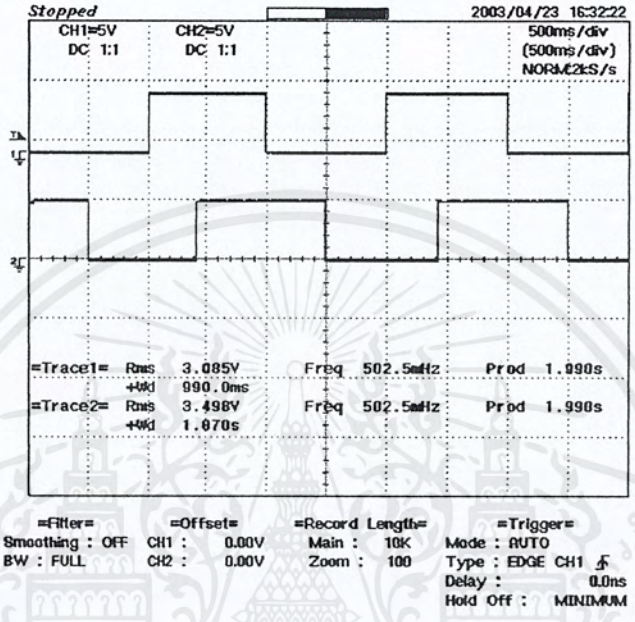


รูปที่ 4.11 แสดงกราฟความสัมพันธ์ความถี่กับความเร็วแบบฮาล์ฟสเตป (Half Step)

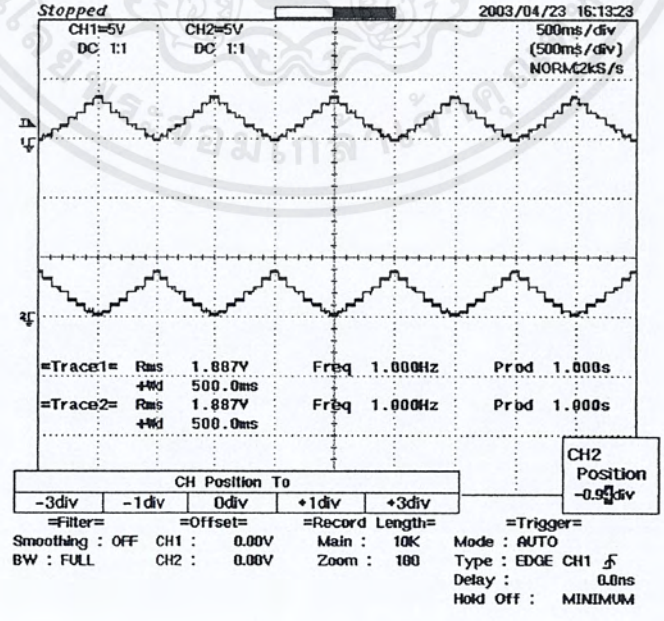
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 การป้อนแบบฟังชั่นมินิสเตป (MINI STEP)

การออกแบบมินิสเตป ออกแบบมาเพื่อที่จะทำให้สเตปปิงมอเตอร์นั้นมีการหมุนที่ต่อเนื่องมากขึ้น ซึ่งการป้อนฟังชั่นแบบมินิสเตปให้แก่มอเตอร์นั้น มีการหมุนที่ต่อเนื่องจะทำให้แก่มอเตอร์สามารถใช้งานที่กว้างขึ้นโดยลักษณะการป้อนฮาล์ฟสเตป จะเป็นไปตามภาพที่ 4.12

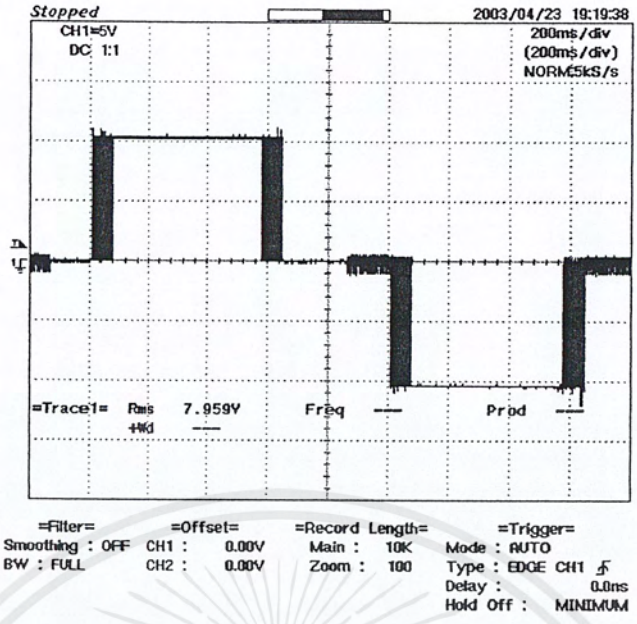


ภาพที่ 4.12 แสดงรูปคลื่นของสัญญาณอ้างอิงของการป้อนแบบมินิเตป



ภาพที่ 4.13 แสดงรูปคลื่นของสัญญาณเครื่องหมายของการป้อนแบบมินิสเตป

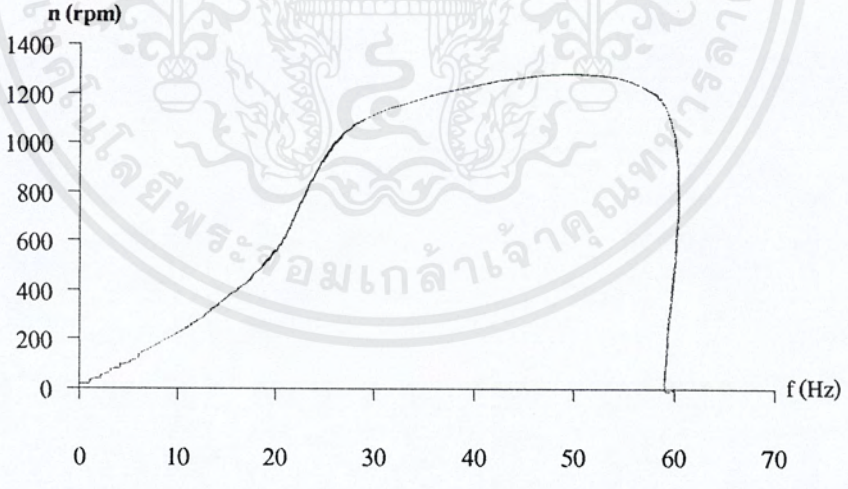
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.14 ลักษณะรูปคลื่นขณะจ่ายให้แก่มอเตอร์

ผลการทดลองโดยการป้อน ณ ความถี่ต่าง

สามารถนำมาเขียนกราฟได้ดังนี้ โดยการป้อนความต่างๆจะมีผลต่อความเร็วรอบของมอเตอร์



รูปที่ 4.15 แสดงกราฟความสัมพันธ์ความถี่กับความเร็วเมื่อการป้อนแบบมินิสเตป (Mini Step)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

การใช้งานสเตปปิงมอเตอร์ โดยทั่วไปนั้นสามารถใช้งานได้ง่าย และทำงานได้ดี แต่ก็ยังมีข้อเสียอีกอย่างหนึ่งคือ การหมุนของสเตปแบบที่ไม่ต่อเนื่องและเกิดแรงดันเหนี่ยวนำที่เกิดจากการหมุนที่เพลาของมอเตอร์

การขับสเตปปิงมอเตอร์แบบมินิสเตป นั้นสามารถทำให้สเตปในการหมุนของมอเตอร์เกิดความราบเรียบและสามารถหมุนได้อย่างต่อเนื่อง ให้ผลตอบสนองทางความเร็วที่ดี

ในโครงงานนี้ได้ทำการออกแบบและสร้างชุดการขับเคลื่อนสเตปปิงมอเตอร์แบบมินิสเตป ที่สามารถโปรแกรมได้ความละเอียดของมินิสเตปได้ตามความต้องการด้วยการกำหนดค่าความของละเอียดในชุดวงจรขับที่เป็น PMW เพื่อศึกษาการทำงานของมอเตอร์และผลตอบสนองในการหมุน

จากผลการทดลองแสดงให้เห็นถึงว่า การขับแบบมินิสเตปนั้นสามารถทำให้การหมุนของสเตปปิงมอเตอร์มีความราบเรียบมากขึ้นและให้ผลตอบสนองทางความเร็วที่สูงมากขึ้นด้วย

5.2 ข้อเสนอแนะ

แต่เมื่อนำมอเตอร์ดังกล่าวไปใช้งานในภาวะที่มีโหลดมาก จะทำให้มอเตอร์มีความเร็วรอบที่ลดลงเนื่องไม่สามารถชนะแรงบิดได้ ดังนั้นหากต้องการพัฒนางานวิจัยชิ้นให้ดีขึ้น ควรใช้ระบบควบคุมแบบลูปปิดโดยใช้พารามิเตอร์ PI เนื่องจากสเตปปิงมอเตอร์มีขนาดเพียงคือหนึ่งออร์เดอร์เราก็จะสามารถควบคุมความเร็วและทอร์คของมอเตอร์ได้แล้ว

5.3 ปัญหาสำหรับงานวิจัย

ก. ต้องศึกษาตัวCONTROLเป็นเวลานาน เนื่องจากผู้ทำงานวิจัยยังไม่มีความรู้มากพอที่จะประยุกต์ใช้งานในระดับสูง จึงต้องศึกษาเป็นเวลานาน

ข. ต้องศึกษาชุดขับเคลื่อนมอเตอร์ ซึ่งมีความซับซ้อนยุ่งยากเป็นอย่างมากทำให้ต้องศึกษาหัวข้อนี้เป็นระยะเวลานาน

ค. อุปกรณ์ที่นำมาต่อใช้งานนั้นเสียหายได้ง่ายและราคาแพง นั้นคือ อุปกรณ์ที่เกี่ยวกับชุด CONTROL และLCD เนื่องจากต้องเสียเวลาไปซื้อ ทำให้การดำเนินงานไม่ต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

(1) ไสว ไม้เหล่,วิศรุต ศรีรัตนะ,ผศ. ประภาษ อุคคกิมาพันธุ์,รศ.กิตติ ตีรเศรษฐ
" การควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรงแบบไม่มีแปลงถ่าน โดยใช้สเตปปิงมอเตอร์-เป็นกรณีศึกษา" วิศวสารลาดกระบัง คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้า-คุณทหารลาดกระบัง ปีที่14 ฉบับที่2 พ.ศ. 2541
(2) วิศรุต ศรีรัตนะ,ไสว ไม้เหล่,ผศ. ประภาษ อุคคกิมาพันธุ์,รศ.กิตติ ตีรเศรษฐ
" มอเตอร์กระแสตรงที่ไม่มีแปลงถ่านโดยใช้สเตปปิงมอเตอร์ " ประชุมวิชาการทวิศวกรรมไฟฟ้า-9สถาบัน ครั้งที่18 มหาวิทยาลัยเทศโนโลยีมหานคร, 1995

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
รายละเอียดโปรแกรมที่ใช้ควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*******************************************************************************
; Program: PROGRAM MABLE MINISTEP DRIVE
; For    : MCS-51 Microcontroller By Mr.CHANNARONG  YOOTTAWARREE
;*******************************************************************************

;-------------------------------------------------------------------------------
; Define Port&Pin Name
;-------------------------------------------------------------------------------

KPAD_ROW0  BIT    P2.0                    ; Keypad Input Row 0
KPAD_ROW1  BIT    P2.1                    ; Keypad Input Row 1
KPAD_ROW2  BIT    P2.2                    ; Keypad Input Row 2
KPAD_ROW3  BIT    P2.3                    ; Keypad Input Row 3
KPAD_COL2  BIT    P2.4                    ; Keypad Output Column 2
KPAD_COL1  BIT    P2.5                    ; Keypad Output Column 1
KPAD_COL0  BIT    P2.6                    ; Keypad Output Column 0
T0_PIN     BIT P3.4                       ; Timer/Counter 0 PIN
LCD_EN     BIT P3.6                ; LCD Module Enable (Active High : Level)
LCD_RS     BIT P3.7                       ; LCD Module Register Select

;-------------------------------------------------------------------------------
; Define User Register
;-------------------------------------------------------------------------------
FLAG   EQU    02FH                        ; User FLAG
OVER   BIT    FLAG.0                      ; Define OVER as bit
KEYPRESS    BIT    FLAG.1
AUXR        EQU    8EH
LCD_ADDR    EQU    030H                   ; For keep LCD Address
LCD_DATA    EQU    031H                   ; For keep LCD Data
KPAD_DATA   EQU    035H                   ; For keep Keypad Data
BUFFER      EQU    036H                   ; Buffer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DATA_SP     EQU   037H          ; Mode Speed
DATA_MOD    EQU   038H          ; DATA MODE
DATA_SIG    EQU   039H          ; DATA SIGNAL
BUF         EQU   040H

;-------------------------------------------------------------------------------
; Main Program.
;-------------------------------------------------------------------------------

ORG   0000H                     ; Reset Vector
AJMP  INITIAL                   ; Jump to initial
ORG   0003H                     ; IE0 Vector
RETI                            ; Retrun interrupt
ORG   000BH                     ; TF0 Vector
SETB  OVER                      ; Set bit OVER Range
RETI                            ; Retrun interrupt
ORG   0013H                     ; IE1 Vector
RETI                            ; Retrun interrupt
ORG   001BH                     ; TF1 Vector
RETI                            ; Return interrupt
ORG   0023H                     ; TI+RI Vector
RETI                            ; Return interrupt
ORG   002BH                     ; TF2+EXF2 Vector
RETI                            ; Return interrupt
INITIAL                         :
MOV   P1,#00000000B             ; Clear Databus
MOV   P3,#00011111B             ; Clear status LCD, P3.2-3.4
MOV   SP,#256-32
MOV   AUXR,#03H
LCALL  DELAY
MAIN: ACALL INIT_LCD            ; Call LCD Initial subroutine
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV    LCD_ADDR,#00H                ; Set Address 00H
ACALL SET_ADDR_LCD;
MOV    DPTR,#TITLE_1                ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                    ; 00H-07H (Increase automatic)
 MOV   LCD_ADDR,#40H                ; Set Address 00H
ACALL SET_ADDR_LCD                  ;
MOV    DPTR,#TITLE_2                ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                    ; 00H-07H (Increase automatic)
ACALL DELAY_1s                      ; Delay
NEW_MODE:   MOV
LCD_ADDR,#00H                       ; Set Address 40H
ACALL SET_ADDR_LCD                  ;
MOV    DPTR,#TITLE_3                ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                    ; 40H-47H (Increase automatic)
MOV    LCD_ADDR,#40H                ; Set Address 00H
ACALL SET_ADDR_LCD                  ;
MOV    DPTR,#TITLE_4                ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                    ; 00H-07H (Increase automatic)
ACALL DELAY_1s                      ; Delay
MOV    LCD_ADDR,#00H                ; Set Address 00H
ACALL SET_ADDR_LCD                  ;
MOV    DPTR,#TITLE_3                ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                    ; 00H-07H (Increase automatic)
MOV    LCD_ADDR,#40H                ; Set Address 00H
ACALL SET_ADDR_LCD                  ;
MOV    DPTR,#TITLE_5                ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                    ; 00H-07H (Increase automatic)
ACALL DELAY_1s                      ; Delay
MOV  LCD_ADDR,#00H                  ; Set Address 40H
ACALL SET_ADDR_LCD                  ;
MOV DPTR,#TITLE_3                   ;Index Pointer ROM to Show LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ACALL WRLINE_LCD                ; 40H-47H (Increase automatic
MOV LCD_ADDR,#40H               ; Set Address 00H
ACALL SET_ADDR_LCD              ;
MOV  DPTR,#TITLE_6              ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                ; 00H-07H (Increase automatic)
ACALL DELAY_1s                  ; Delay
KEY                             :
MOV LCD_ADDR,#00H               ; Set Address 40H
ACALL SET_ADDR_LCD              ;
MOV DPTR,#TITLE_3               ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                ; 40H-47H (Increase automatic
MOV LCD_ADDR,#40H               ; Set Address 00H
ACALL SET_ADDR_LCD              ;
MOV DPTR,#TITLE_31              ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                ; 00H-07H (Increase automatic)
ACALL DELAY_1s                  ; Delay
ACALL DELAY_1s
ACALL WAIT_KEYPRESSED
MOV A,KPAD_DATA
MODE: CJNE    A,#01H,MODE_2
ACALL MODE_FULL
                    RET

MODE_2:
CJNE  A,#02H,MODE_3
ACALL  MODE_HALF
                    RET
MODE_3:
CJNE    A,#03H,KEY
ACALL MODE_MINI
                    RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MODE_FULL:
MOVLCD_ADDR,#00H                ; Set Address 00H
ACALLSET_ADDR_LCD               ;
MOV   DPTR,#TITLE_10            ; Index Pointer ROM to Show LCD
ACALLWRLINE_LCD                 ; 00H-07H (Increase automatic)
MOV   LCD_ADDR,#40H             ; Set Address 00H
ACALLSET_ADDR_LCD               ;
MOV   DPTR,#TITLE_11            ; Index Pointer ROM to Show LCD
ACALLWRLINE_LCD                 ; 00H-07H (Increase automatic)
ACALLDELAY_1s                   ; Delay
ACALLDELAY_1s
ACALLGET_KPAD
CJNE  A,#10,CHECK_1
ACALL FULL_LEFT
                RET

CHECK_1: CJNE   A,#12,MODE_FULL
ACALL FULL_RIGHT
                RET

FULL_LEFT :
MOV LCD_ADDR,#00H               ; Set Address 00H
ACALLSET_ADDR_LCD               ;
1MOV  DPTR,#TITLE_10            ; Index Pointer ROM to Show LCD
ACALLWRLINE_LCD                 ; 00H-07H (Increase automatic)MOV
LCD_ADDR,#40H                   ; Set Address 00H
ACALLSET_ADDR_LCD               ;
MOV   DPTR,#TITLE_13            ; Index Pointer ROM to Show LCD
ACALLWRLINE_LCD                 ; 00H-07H (Increase automatic)
ACALLDELAY_1s                   ; Delay
ACALLDELAY_1s
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ACALL SPEED
MOV   LCD_ADDR,#00H          ; Set Address 00H
ACALL SET_ADDR_LCD           ;
MOV   DPTR,#TITLE_13         ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD             ; 00H-07H (Increase automatic)
MOV   LCD_ADDR,#40H          ; Set Address 46H
ACALL SET_ADDR_LCD           ;
MOV   DPTR,#SCR_SPEED        ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD             ; 40H-47H (Increase automatic)
ACALL DELAY_1s               ; Delay
ACALL DELAY_1s
AGAIN_1:
MOV   LCD_ADDR,#49H          ; Set Address 40H
ACALL SET_ADDR_LCD           ;
ACALL WAIT_KEYPRESSED
MOV   BUFFER,KPAD_DATA
ACALL KPAD2LCD
ACALL WAIT_KEY
ACALL WAIT_KEYPRESSED
ACALL LCD_LSHF
MOV   BUFFER+1,KPAD_DATA
ACALL KPAD2LCD
ACALL WAIT_KEY
ACALL BUFFER2ACC
MOV   R5,#50
MOV   R4,#40
MOV   R3,#30
MOV   R2,#30
MOV   DATA_SP,#1
MOD_1:
ACALL WAIT_KEY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ACALL WAIT_KEYPRESSED_MODE
MOV   P1,#00
MOV   DPTR,#DATA_TABLE_1
MOV   R0,#4
MOV   R1,#00H
MOV   A,#00H
LOOP_1:
MOV   A,R1
MOVC  A,@A+DPTR
MOV   P1,A
CJNE  R5,#00,START_DELAY
CJNE  R4,#00,START_DELAY_0
CJNE  R3,#00,START_DELAY_1
CJNE  R2,#00,START_DELAY_2
ACALL DELAY_START
 INC  R1
DJNZ  R0,LOOP_1
AJMP  MOD_1
START_DELAY:
ACALL DELAY_START_0
 INC  R1
 DEC  R5
DJNZ  R0,LOOP_1
AJMP  MOD_1
START_DELAY_0:
ACALL DELAY_START_1
 INC  R1
DEC   R4
DJNZ  R0,LOOP_1
AJMP  MOD_1
START_DELAY_1:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ACALL DELAY_START_2
 INC     R1
 DEC     R3
DJNZ   R0,LOOP_1
AJMP   MOD_1
START_DELAY_2:
ACALL DELAY_START_3
INC     R1
DEC     R2
DJNZ   R0,LOOP_1
AJMP   MOD_1
FULL_RIGHT:
 MOV   LCD_ADDR,#00H                ; Set Address 00H
ACALL SET_ADDR_LCD                  ;
MOV    DPTR,#TITLE_10               ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                    ; 00H-07H (Increase automatic)
 MOV   LCD_ADDR,#40H                ; Set Address 00H
ACALL  SET_ADDR_LCD                 ;
MOV    DPTR,#TITLE_12               ; Index Pointer ROM to Show LCD
ACALL  WRLINE_LCD                   ; 00H-07H (Increase automatic)
ACALL  DELAY_1s                     ; Delay
ACALL  DELAY_1s
ACALL  SPEED
MOV    LCD_ADDR,#00H                ; Set Address 00H
ACALL  SET_ADDR_LCD                 ;
MOV    DPTR,#TITLE_12               ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                    ; 00H-07H (Increase automatic)
MOV    LCD_ADDR,#40H                ; Set Address 46H
ACALL SET_ADDR_LCD                  ;
MOV    DPTR,#SCR_SPEED              ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                    ; 40H-47H (Increase automatic)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ACALL DELAY_1s                          ; Delay
ACALL DELAY_1s
AGAIN_2:
MOV  LCD_ADDR,#49H                      ; Set Address 40H
ACALL SET_ADDR_LCD                      ;
ACALL WAIT_KEYPRESSED
MOV   BUFFER,KPAD_DATA
ACALL KPAD2LCD
ACALL WAIT_KEY
ACALL WAIT_KEYPRESSED
ACALL LCD_LSHF
MOV   BUFFER+1,KPAD_DATA
ACALL  KPAD2LCD
ACALL WAIT_KEY
ACALL BUFFER2ACC
MOV   R5,#50
 MOV   R4,#40
 MOV   R3,#30
 MOV   R2,#30
 MOV   DATA_SP,#2
MOD_2: ACALL   WAIT_KEY
ACALL WAIT_KEYPRESSED_MODE
MOV   P1,#00
MOV   DPTR,#DATA_TABLE_2
MOV   R0,#4
MOV   R1,#00H
MOV   A,#00H
LOOP_2:
MOV   A,R1
MOVC A,@A+DPTR
MOV   P1,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CJNE   R5,#00,START_DELAY_3
CJNE   R4,#00,START_DELAY_4
CJNE   R3,#00,START_DELAY_5
ACALL DELAY_START
INC    R1
DJNZ   R0,LOOP_2
AJMP   MOD_2
START_DELAY_3:
ACALL DELAY_START_0
INC    R1
DEC    R5
DJNZ   R0,LOOP_2
AJMP   MOD_2
START_DELAY_4:
ACALL DELAY_START_1
INC    R1
DEC    R4
DJNZ   R0,LOOP_2
AJMP   MOD_2
START_DELAY_5:
ACALL DELAY_START_2
 INC    R1
 DEC    R3
 DJNZ   R0,LOOP_2
AJMP   MOD_2
START_DELAY_6:
ACALL DELAY_START_3
 INC    R1
 DEC    R2
DJNZ   R0,LOOP_2
AJMP   MOD_2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MODE_HALF:
MOV   LCD_ADDR,#00H                    ; Set Address 00H
ACALL SET_ADDR_LCD                     ;
MOV   DPTR,#TITLE_10                   ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                       ; 00H-07H (Increase automatic)
MOV   LCD_ADDR,#40H                    ; Set Address 00H
ACALL SET_ADDR_LCD                     ;
MOV   DPTR,#TITLE_11                   ;Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                       ; 00H-07H (Increase automatic)
ACALL DELAY_1s                         ; Delay
ACALL DELAY_1s
ACALLGET_KPAD
 MOV  A,KPAD_DATA
 CJNE A,#10,CHECK_2
ACALL HALF_LEFT
                RET

CHECK_2:
CJNE  A,#12,MODE_HALF
ACALL HALF_RIGHT
                RET

HALF_LEFT:
MOV LCD_ADDR,#00H                      ; Set Address 40H
ACALL SET_ADDR_LCD                     ;
MOV DPTR,#TITLE_10                     ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                       ; 40H-47H (Increase automatic
MOV LCD_ADDR,#40H; Set Address 4   0H
ACALL SET_ADDR_LCD                     ;
MOV  DPTR,#TITLE_15                    ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                       ; 40H-47H (Increase automatic )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ACALL SPEED
MOV   LCD_ADDR,#00H          ; Set Address 40H
ACALL SET_ADDR_LCD           ;
MOV   DPTR,#TITLE_15         ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD             ; 40H-47H (Increase automatic
MOV   LCD_ADDR,#40H          ; Set Address 46H
ACALL SET_ADDR_LCD           ;
MOV   DPTR,#SCR_SPEED        ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD             ; 40H-47H (Increase automatic)
ACALL DELAY_1s               ; Delay
ACALL DELAY_1s
AGAIN_3:
MOV   LCD_ADDR,#49H          ; Set Address 40H
ACALLSET_ADDR_LCD            ;
CALL  WAIT_KEYPRESSED
MOVBUFFER,KPAD_DATA
ACALL KPAD2LCD
ACALL WAIT_KEY
ACALL WAIT_KEYPRESSED
ACALL LCD_LSHF
MOV   BUFFER+1,KPAD_DATA
ACALL KPAD2LCD
ACALL WAIT_KEY
ACALL BUFFER2ACC
MOV   DATA_SP,#3
MOD_3:
ACALL WAIT_KEY
ACALL WAIT_KEYPRESSED_MODE
MOV   P1,#00
MOV   DPTR,#DATA_TABLE_3
MOV   R0,#8
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV   R1,#00H
MOV   A,#00H
LOOP_3:
MOV   A,R1
MOVC  A,@A+DPTR
MOV   P1,A
ACALL DELAY_START
INC   R1
DJNZ  R0,LOOP_3
AJMP  MOD_3
HALF_RIGHT:
MOV   LCD_ADDR,#00H          ; Set Address 00H
ACALL SET_ADDR_LCD           ;
MOV   DPTR,#TITLE_10         ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD             ; 00H-07H (Increase automatic)
MOV   LCD_ADDR,#40H          ; Set Address 00H
ACALL SET_ADDR_LCD           ;
MOV   DPTR,#TITLE_14         ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD             ; 00H-07H (Increase automatic)
ACALL DELAY_1s               ; Delay
ACALL DELAY_1s
ACALL SPEED
MOV   LCD_ADDR,#00H          ; Set Address 00H
ACALL SET_ADDR_LCD           ;
MOV   DPTR,#TITLE_14         ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD             ; 00H-07H (Increase automatic)
MOV   LCD_ADDR,#40H          ; Set Address 46H
ACALL SET_ADDR_LCD           ;
MOV   DPTR,#SCR_SPEED        ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD             ; 40H-47H (Increase automatic)
ACALL DELAY_1s               ; Delay
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ACALL DELAY_1s
AGAIN_4:
MOV  LCD_ADDR,#49H                    ; Set Address 40H
ACALL SET_ADDR_LCD                    ;
ACALL WAIT_KEYPRESSED
MOV  BUFFER,KPAD_DATA
ACALL KPAD2LCD
ACALL WAIT_KEY
ACALL WAIT_KEYPRESSED
ACALL LCD_LSHF
MOV   BUFFER+1,KPAD_DATA
ACALL KPAD2LCD
ACALL WAIT_KEY
ACALLBUFFER2ACC
MOV   DATA_SP,#4
MOD_4:
ACALLWAIT_KEY
ACALLWAIT_KEYPRESSED_MODE
MOV   P1,#00
MOV   DPTR,#DATA_TABLE_4
MOV   R0,#8
MOV   R1,#00H
MOV   A,#00H
LOOP_4:
MOV   A,R1
MOVC A,@A+DPTR
MOV   P1,A
ACALLDELAY_START
INC     R1
DJNZ   R0,LOOP_4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
AJMP   MOD_4
MODE_MINI:
MOV    LCD_ADDR,#00H          ; Set Address 00H
ACALL SET_ADDR_LCD            ;
MOV    DPTR,#TITLE_10         ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD              ; 00H-07H (Increase automatic)
MOV    LCD_ADDR,#40H          ; Set Address 00H
ACALL SET_ADDR_LCD            ;
MOV    DPTR,#TITLE_11         ; Index Pointer ROM to Show LCD
ACALLWRLINE_LCD               ; 00H-07H (Increase automatic)
ACALLDELAY_1s                 ; Delay
ACALL DELAY_1s
ACALLGET_KPAD
 CJNE  A,#10,CHECK_3
 ACALLMINI_LEFT
                RET

CHECK_3:
CJNE   A,#12,MODE_MINI
ACALL   MINI_RIGHT
                RET

MINI_LEFT:
MOVLCD_ADDR,#00H              ; Set Address 00H
ACALL SET_ADDR_LCD            ;
MOV    DPTR,#TITLE_10         ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD              ; 00H-07H (Increase automatic)
MOV    LCD_ADDR,#40H          ; Set Address 00H
ACALL SET_ADDR_LCD            ;
MOV    DPTR,#TITLE_17         ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD              ; 00H-07H (Increase automatic)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ACALL DELAY_1s                    ; Delay
ACALL DELAY_1s
ACALL SPEED
MOV   LCD_ADDR,#00H               ; Set Address 40H
ACALL SET_ADDR_LCD                ;
MOV   DPTR,#TITLE_17              ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                  ; 40H-47H (Increase automatic
MOV   LCD_ADDR,#40H               ; Set Address 46H
ACALL SET_ADDR_LCD                ;
MOV   DPTR,#SCR_SPEED             ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                  ; 40H-47H (Increase automatic)
ACALL DELAY_1s                    ; Delay
ACALL DELAY_1s
AGAIN_5:
MOV   LCD_ADDR,#49H               ; Set Address 40H
ACALL SET_ADDR_LCD                ;
ACALL WAIT_KEYPRESSED
MOV   BUFFER,KPAD_DATA
ACALL KPAD2LCD
ACALL WAIT_KEY
ACALL WAIT_KEYPRESSED
ACALL LCD_LSHF
MOV   BUFFER+1,KPAD_DATA
ACALL  KPAD2LCD
ACALL WAIT_KEY
ACALL  BUFFER2ACC
MOV   DATA_SP,#5
MOD_5:
ACALL WAIT_KEY
ACALL WAIT_KEYPRESSED_MODE
MOV   P1,#00
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV   DPTR,#DATA_TABLE_5
MOV   R0,#28
MOV   R1,#00H
MOV   A,#00H
LOOP_5:
MOV   A,R1
MOVC  A,@A+DPTR
MOV   P1,A
ACALL DELAY_START
INC   R1
DJNZ  R0,LOOP_5
AJMP  MOD_5
MINI_RIGHT:
 MOV  LCD_ADDR,#00H                ; Set Address 00H
ACALL SET_ADDR_LCD                 ;
MOV   DPTR,#TITLE_10               ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                   ; 00H-07H (Increase automatic)
MOV   LCD_ADDR,#40H                ; Set Address 00H
ACALL SET_ADDR_LCD                 ;
MOV   DPTR,#TITLE_16               ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                   ; 00H-07H (Increase automatic)
ACALL DELAY_1s                     ; Delay
ACALL DELAY_1s
ACALL SPEED
MOV   LCD_ADDR,#00H                ; Set Address 40H
ACALL SET_ADDR_LCD                 ;
MOV  DPTR,#TITLE_16                ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                   ; 40H-47H (Increase automatic)
MOV   LCD_ADDR,#40H                ; Set Address 46H
ACALL SET_ADDR_LCD                 ;
MOV   DPTR,#SCR_SPEED              ; Index Pointer ROM to Show LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ACALL WRLINE_LCD                    ; 40H-47H (Increase automatic)
ACALL DELAY_1s                      ; Delay
ACALL DELAY_1s
AGAIN_6:
MOV   LCD_ADDR,#49H                 ; Set Address 40H
ACALLSET_ADDR_LCD                   ;
ACALLWAIT_KEYPRESSED
MOV   BUFFER,KPAD_DATA
ACALLKPAD2LCD
ACALLWAIT_KEY
ACALLWAIT_KEYPRESSED
ACALL LCD_LSHF
MOV   BUFFER+1,KPAD_DATA
ACALL KPAD2LCD
ACALL WAIT_KEY
 ACALLBUFFER2ACC
 MOV   DATA_SP,#6
 MOD_6:
ACALL WAIT_KEY
ACALL WAIT_KEYPRESSED_MODE
MOV   P1,#00
MOV   DPTR,#DATA_TABLE_6
MOV   R0,#28
MOV   R1,#00H
MOV   A,#00H
LOOP_6:
MOV   A,R1
MOVC  A,@A+DPTR
MOV   P1,A
ACALL DELAY_START
INC   R1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DJNZ   R0,LOOP_6
AJMP   MOD_6
SPEED:
MOV    LCD_ADDR,#00H                    ; Set Address 00H
ACALL SET_ADDR_LCD                      ;
MOV    DPTR,#TITLE_7                    ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                        ; 00H-07H (Increase automatic)
MOV    LCD_ADDR,#40H                    ; Set Address 00H
ACALL SET_ADDR_LCD                      ;
MOV    DPTR,#TITLE_8                    ; Index Pointer ROM to Show LCD
ACALL WRLINE_LCD                        ; 00H-07H (Increase automatic)
 ACALLDELAY_1s                          ; Delay
ACALL DELAY_1s
                    RET

;-------------------------------------------------------------------------------
; HEX Code to show LCD
; I/P:          LCD_DATA
;-------------------------------------------------------------------------------

HEX2LCD:
PUSH   ACC                              ; Push ACC.
MOV    A,LCD_DATA                       ; Get Data
MOV    B,#100                           ;
DIV    AB                               ; Divide by 100
ADD    A,#30H                           ; Convert to ASCII
CJNE   A,#30H,HEX2_LCD ; Check x100 = 0 ?  ; 0 => Write Space
HEX2_LCD:
MOV    LCD_DATA,A                       ;
ACALL WRCHAR_LCD                        ; Write x100
MOV    A,B                              ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV   B,#10                    ;
DIV   AB                       ; Divide by 10
ADD   A,#30H                   ; Convert to ASCII
MOV   LCD_DATA,A               ;
ACALL WRCHAR_LCD               ; Write Lower HEX Code
MOV   A,B                      ; Get Remainder x1
ADD   A,#30H                   ; Convert to ASCII
MOV   LCD_DATA,A               ;
ACALL WRCHAR_LCD               ; Write Lower HEX Code
POP   ACC                      ; Pop ACC.
      RET                      ;Return

;-------------------------------------------------------------------------------
; LCD Initialize
;-------------------------------------------------------------------------------

INIT_LCD:
ACALL DELAY_100ms              ; Delay
CLR   LCD_RS                   ; Clear LCD_RS Pin
MOV   P0,#00111000B            ; 8bit Mode
ACALL LCD_CLK                  ; Pulse LCD Clock
ACALL DELAY_10ms               ; Delay
MOV   P0,#00111000B ; 8bit Mode
ACALL LCD_CLK                  ; Pulse LCD Clock
ACALL LCD_OFF                  ; Display Off
ACALL LCD_CLR                  ; Clear Display
MOV   P0,#00000110B ; Entry Mode
ACALL LCD_CLK                  ; Pulse LCD Clock
ACALL LCD_HOME                 ; Return Home Display
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------------------------------------------------------------
; LCD Clear Display
;-------------------------------------------------------------------------------

LCD_CLR:
CLR     LCD_RS                          ; Clear LCD_RS Pin
MOV     P0,#00000001B                   ; Display Clear
ACALL LCD_CLK                           ; Pulse LCD Clock
                        RET

;-------------------------------------------------------------------------------
; LCD Return Home
;-------------------------------------------------------------------------------
LCD_HOME:
CLR   LCD_RS                            ; Clear LCD_RS Pin
MOV     P0,#00000010B                   ; Return Home
ACALL LCD_CLK                           ; Pulse LCD Clock
                        RET

;-------------------------------------------------------------------------------
; LCD Display Off
;-------------------------------------------------------------------------------
LCD_OFF:
CLR     LCD_RS                          ; Clear LCD_RS Pin
MOV     P0,#00001000B ; Display Off
ACALL LCD_CLK                           ; Pulse LCD Clock
                        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------------------------------------------------------------
; LCD Clk
;-------------------------------------------------------------------------------
LCD_CLK:
SETB   LCD_EN                          ; Pulse Clock to LCD_EN
ACALL LCD_DELAY
CLR    LCD_EN
ACALL LCD_DELAY
                    RET

;-------------------------------------------------------------------------------
; LCD Left Shift Display
;-------------------------------------------------------------------------------
LCD_LSHF:
CLR    LCD_RS                          ; Clear LCD_RS Pin
MOV    P0,#00001100B                   ; Display Cursor and Blink
ACALL LCD_CLK                          ; Pulse LCD Clock
                    RET

;-------------------------------------------------------------------------------
; LCD Cursor On
;-------------------------------------------------------------------------------
LCD_BLINK:
CLR    LCD_RS                          ; Clear LCD_RS Pin
MOV    P0,#00001111B                   ; Display Cursor and Blink
ACALL LCD_CLK                          ; Pulse LCD Clock
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------------------------------------------------------------
; LCD Display On
;-------------------------------------------------------------------------------
LCD_ON:
CLR    LCD_RS                          ; Clear LCD_RS Pin
MOV    P0,#00001100B ; Display On
ACALL LCD_CLK
                      RET

;-------------------------------------------------------------------------------
; Set LCD Address
; I/P:          LCD_ADDR
;-------------------------------------------------------------------------------
SET_ADDR_LCD:
CLR    LCD_RS                          ; Clear LCD_RS Pin
MOV    A,LCD_ADDR                      ; Move LCD_ADDR to ACC.
SETB   ACC.7                           ; Set bit ACC.7
MOV           P0,A                     ; Move to DATABUS
ACALL LCD_CLK                          ; Pulse LCD Clock
                      RET

;-------------------------------------------------------------------------------
; Write Character to show LCD
; I/P:          LCD_DATA
;-------------------------------------------------------------------------------
WRCHAR_LCD:
SETB   LCD_RS                          ; Set LCD_RS Pin
MOV    P0,LCD_DATA                     ; Move LCD_DATA to DATABUS
ACALL LCD_CLK                          ; Pulse LCD Clock
ACALL LCD_ON                           ; Display On
                      RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------------------------------------------------------------------------------
; Write Line of 16 Character from ROM
; I/P:DPTR : Locate ROM Address
;-------------------------------------------------------------------------------------------------
WRLINE_LCD:
MOV   R0,#0                          ; Clear loop counter
WRLINE_LCD_1:
MOV   A,#00000010B
SETB  LCD_RS
CLR   A                              ; Clear ACC
MOVC  A,@A+DPTR                      ; Move data from @DPTR to ACC
MOV   P0,A                           ;Move ACC. to DATABUS
ACALL LCD_CLK                        ; Pulse LCD Clock
INC   DPTR                           ; Increase Pointer
INC   R0                             ; Increase loop counter
CJNE  R0,#32,WRLINE_LCD_1            ; Do until 8 times
ACALL LCD_ON                         ; Display On
                    RET

;-------------------------------------------------------------------------------------------------
; Keypad Scan key Subroutine
;-------------------------------------------------------------------------------------------------
GET_KPAD:
MOV   P2,#0FFH                       ; Pull P2 to High
MOV   KPAD_DATA,#0                   ; Clear Keypad Data
CHK_COL0:
CLR   KPAD_COL0                      ; Begin Scan Column 0
MOV   A,P2                           ; Get Port2 Value
ANL   A,#00FH                        ; Get only lower 4 bit
CJNE  A,#00FH,COL0_DETECT            ; Check All rows '1'?
AJMP  CHK_COL1                       ; All rows '1' => check next column
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
COL0_DETECT:
MOV   KPAD_DATA,#01                 ; Initial KPAD_DATA = 1
AJMP  GET_ROW                       ; Jump to get row value
CHK_COL1:
SETB  KPAD_COL0                     ; Stop Scan Column 0
CLR   KPAD_COL1                     ; Begin Scan Column 1
MOV   A,P2                          ; Get Port2 Value
ANL   A,#00FH                       ; Get only lower 4 bit
CJNE  A,#00FH,COL1_DETECT           ; Check All rows '1'?
AJMP  CHK_COL2                      ; All rows '1' => check next column
COL1_DETECT:
MOV   KPAD_DATA,#02                 ; Initial KPAD_DATA = 2
AJMP  GET_ROW                       ; Jump to get row value
CHK_COL2:
SETB  KPAD_COL1                     ; Stop Scan Column 1
CLR   KPAD_COL2                     ; Begin Scan Column 2
MOV   A,P2                          ; Get Port2 Value
ANL   A,#00FH                       ; Get only lower 4 bit
CJNE  A,#00FH,COL2_DETECT           ; Check All rows '1'?
                      RET           ; All rows '1' => return

COL2_DETECT:
MOV   KPAD_DATA,#03                 ; Initial KPAD_DATA = 2
GET_ROW:CLR  KPAD_COL0              ; Enable all Column to find Crosspoint
CLR   KPAD_COL1                     ;
CLR   KPAD_COL2                     ;
JB    KPAD_ROW0,CHK_ROW1            ; Check Row 0 Detect?
                      RET           ; Row 0 Detect => return
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CHK_ROW1:
JB      KPAD_ROW1,CHK_ROW2              ; Check Row 2 Detect?
MOV     A,KPAD_DATA                     ; Add 3 with KPAD_DATA   ADDA,#3
                                        ;
MOV     KPAD_DATA,A                     ;
RET                                     ; Return
CHK_ROW2:
JB      KPAD_ROW2,CHK_ROW3              ; Check Row 2 Detect?
MOV     A,KPAD_DATA                     ; Add 6 with KPAD_DATA
ADD     A,#6                            ;
MOV     KPAD_DATA,A                     ;
RET                                     ; Return
CHK_ROW3:
MOV     A,KPAD_DATA                     ; Add 9 with KPAD_DATA
ADD     A,#9                            ;
MOV     KPAD_DATA,A                     ;

                RET                     ; Return

;-------------------------------------------------------------------------------
; KEYPAD to show LCD
; I/P:          KPAD_DATA
;-------------------------------------------------------------------------------
KPAD2LCD:
MOV     A,KPAD_DATA                     ; Get Keypad Data
ADD     A,#30H                          ; Convert to ASCII
MOV     LCD_DATA,A                      ; Write to lCD
ACALL WRCHAR_LCD                        ;

                RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------------------------------------------------------------
; BCD Code to show LCD
; I/P:          LCD_DATA
;-------------------------------------------------------------------------------
BCD2LCD:
PUSH   ACC                          ; Push ACC. To Stack
PUSH   B                            ; Push B to Stack
MOV    A,LCD_DATA                   ; Get input data value
MOV    B,A                          ; Copy to B
ANL    A,#11110000B  ; Get higher 4 bit
SWAP   A                            ; Swap nibble
ADD    A,#030H                      ; Convert to ASCII
MOV    LCD_DATA,A                   ; Write LCD
ACALL WRCHAR_LCD                    ;
MOV    A,B                          ; Restore value
ANL    A,#00001111B                 ; Get lower 4 bit
ADD    A,#030H                      ; Convert to ASCII
MOV    LCD_DATA,A                   ; Write LCD
ACALL WRCHAR_LCD                    ;
POP    B                            ; Pop B from Stack
POP    ACC                          ; Pop ACC. from Stack
                RET                 ; Return

;-------------------------------------------------------------------------------
; Wait keypad depressed
;-------------------------------------------------------------------------------
WAIT_KEY:
MOV    A,P2                         ; Move P2 to ACC.
ANL    A,#00FH                      ; Get only Lower 4 bit
CJNE   A,#00FH,WAIT_KEY             ; All bit are high ?
                RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------------------------------------------------------------
; Wait keypad pressed 0-9 Only
;-------------------------------------------------------------------------------
WAIT_KEYPRESSED:
ACALL GET_KPAD                          ; Check keypad Data
MOV   A,KPAD_DATA                       ;
CJNE  A,#0,CHK_KEY_NEXT                 ; Check keypad pressed?
AJMP  WAIT_KEYPRESSED                   ; Loop until keypad pressed
CHK_KEY_NEXT:
CJNE  A,#10,CHK_KEY_0                   ; Check key *
AJMP  WAIT_KEYPRESSED                   ; If key * then loop wait
CHK_KEY_0:
CJNE  A,#11,CHK_VALID_KEY               ; Check key 0
MOV   KPAD_DATA,#0                      ; Replace KPAD_DATA = 0
                RET                     ; Return
CHK_VALID_KEY:
JNC   WAIT_KEYPRESSED
                RET                     ; If KPAD_DATA > 11 is Enter #

;-------------------------------------------------------------------------------
; Wait keypad pressed 0-9 Only
;-------------------------------------------------------------------------------
WAIT_KEYPRESSED_MODE:
ACALL GET_KPAD                          ; Check keypad Data
MOV   A,KPAD_DATA                       ;
CJNE  A,#0,CHK_KEY_NEXT_1               ; Check keypad pressed?
RET                                     ; Loop until keypad pressed
CHK_KEY_NEXT_1:
CJNE  A,#10,CHK_KEY_1                   ; Check key *
MOV   A,DATA_SP
CJNE  A,#01,CHK_SP_1; If key * then loop wait
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ACALL FULL_LEFT
RET
CHK_SP_1:
CJNE   A,#02,CHK_SP_2
ACALL FULL_RIGHT
                              RET

CHK_SP_2CJNE  A,#03H,CHK_SP_3
ACALL HALF_LEFT
                              RET
CHK_SP_3:
CJNE   A,#04H,CHK_SP_4
ACALL HALF_RIGHT
                              RET
CHK_SP_4:
CJNE   A,#05H,CHK_SP_5
ACALL MINI_LEFT
                              RET
CHK_SP_5:
CJNE   A,#06H,CHK_KEY_1
ACALL MINI_RIGHT
                              RET
                              RET

CHK_KEY_1:
CJNE   A,#12,CHK_VALID_KEY_1; Check key 0
ACALL KEY                                   ; Replace KPAD_DATA = 0
RET                                         ; If KPAD_DATA > 11 is Enter #
CHK_VALID_KEY_1:
                              RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------------------------------------------
; BUFFER to ACC
; I/P:   BUFFER,BUFFER+1
; O/P:   ACC
;-------------------------------------------------------------
BUFFER2ACC:
MOV   A,BUFFER                    ; Restore BUFFER
ANL   A,#0FH                      ; Get lower 4 bit
MOV   B,#10
MUL   AB
MOV   R1,A                        ; Move to B
MOV   A,BUFFER+1                  ; Restore BUFFER+1
ANL   A,#0FH                      ; Get lower 4 bit
ADD   A,R1                        ; Add BUFFER to make RTC BCD
ADD   A,B
MOV   DATA_SIG,A
                  RET

;-------------------------------------------------------------
; Dummy Delay time LCD_DELAY, 10m, 100m, 250m, 1s
;-------------------------------------------------------------
DELAY:
PUSH   DPH
PUSH   DPL
 PUSH  ACC
MOV    DPTR,#0000H
DEL:
INC    DPTR
MOV    A,DPL
ORL    A,DPH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
JNZ     DEL
POP     ACC
POP     DPL
POP     DPH
RET

LCD_DELAY:
MOV   R7,#002                          ; Do 2 times
LCD_DELAY_1:
MOV   R6,#0E6H                         ; Each loop = 1 ms
LCD_DELAY_2:
NOP
NOP
DJNZ   R6,LCD_DELAY_2
DJNZ   R7,LCD_DELAY_1
                        RET

DELAY_10ms:
MOV   R7,#06                           ; Do 10 times
DELAY_10ms_1:
MOV   R6,#05                           ; Each loop = 1 ms
DELAY_10ms_2:
NOP
NOP
DJNZ   R6,DELAY_10ms_2
DJNZ   R7,DELAY_10ms_1
                              RET

DELAY_100ms:
MOV   R7,#100                          ; Do 100 times
DELAY_100ms_1:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV   R6,#0E6H                          ; Each loop = 1 ms
DELAY_100ms_2:
NOP
NOP
DJNZ  R6,DELAY_100ms_2
DJNZ  R7,DELAY_100ms_1
                        RET

DELAY_250ms:
MOV   R7,#250                           ; Do 250 times
DELAY_250ms_1:
MOV   R6,#0E6H                          ; Each loop = 1 ms
DELAY_250ms_2:
NOP
NOP
DJNZ  R6,DELAY_250ms_2
DJNZ  R7,DELAY_250ms_1
                        RET

DELAY_1s:
MOV   R5,#100                           ; Do 100 times
DELAY_1s_1:
ACALL  DELAY_10ms
DJNZ  R5,DELAY_1s_1
                        RET

DELAY_START:
MOV   R7,DATA_SIG
DELAY_S100ms_1:
MOV   R6,#0A5H                          ; Each loop = 1 ms
DELAY_S100ms_2:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
NOP
NOP
DJNZ   R6,DELAY_S100ms_2
DJNZ   R7,DELAY_S100ms_1
                    RET

DELAY_START_0:
MOV    R7,#0AH
DELAY_S100ms_10:
MOV    R6,#0A6H                    ; Each loop = 1 ms
DELAY_S100ms_20:
NOP
NOP
DJNZ   R6,DELAY_S100ms_20
DJNZ   R7,DELAY_S100ms_10
                    RET

DELAY_START_1:
MOV    R7,#09H
DELAY_S100ms_11:
MOV    R6,#0A6H                    ; Each loop = 1 ms
DELAY_S100ms_21:
NOP
NOP
DJNZ   R6,DELAY_S100ms_21
DJNZ   R7,DELAY_S100ms_11
                    RET

DELAY_START_2:
MOV    R7,#08H
DELAY_S100ms_12:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV    R6,#0A6H                                    ; Each loop = 1 ms
DELAY_S100ms_22:
NOP
DJNZ   R6,DELAY_S100ms_22
DJNZ   R7,DELAY_S100ms_12
                              RET

DELAY_START_3:
MOV    R7,#06H
DELAY_S100ms_13:
MOV    R6,#0A6H                                    ; Each loop = 1 ms
DELAY_S100ms_23:
NOP
DJNZ   R6,DELAY_S100ms_23
DJNZ   R7,DELAY_S100ms_13
                              RET

;-------------------------------------------------------------------------------
; Dummy Delay time SIGNAL
;-------------------------------------------------------------------------------
;-------------------------------------------------------------------------------
;Define Constant < Store in Flash EEPROM Program Memory >
;-------------------------------------------------------------------------------
DATA_TABLE_1:
DB
0EEH,0FEH,0FFH,0EFH
DATA_TABLE_2:
DB
0EFH,0FFH,0FEH,0EEH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DATA_TABLE_3:
DB
0EFH,0E0H,0EEH,0EH,0FEH,0F1H,0FFH,01FH
DATA_TABLE_4:
DB
01FH,0FFH,0F1H,0FEH,0EH,0EEH,0E0H,0EFH
DATA_TABLE_5:
DB
0F1H,0D3H,0B5H,97H,79H,5BH,3DH,1FH,2DH,4BH,69H,87H,0A5H,0C3H,0E1H,0C2H,0A4
H,86H,68H,4AH,2CH,0EH,3CH,5AH,78H,96H,0B4H,0D2H
DATA_TABLE_6:
DB
0D2H,0B4H,96H,78H,5AH,3CH,0EH,2CH,4AH,68H,86H,0A4H,0C2H,0E1H,0C3H,0A5H,87H
69H,4BH,2DH,1FH,3DH,5BH,79H,97H,0B5H,0D3H,0F1H
DATA_TABLE:
DB      00H,01H,02H,03H,04H,05H,06H,07H
DB      8H,09H,0AH,0BH,0CH,0DH,0EH,0FH
DB      10H,11H,12H,13H,14H,15H,16H,17H
DB      18H,19H,1AH,1BH,1CH,1DH,1EH,1FH
DB      20H,21H,22H,23H,24H,25H,26H,27H
DB      28H,29H,2AH,2BH,2CH,2DH,2EH,2FH
DB      30H,31H,32H,33H,34H,35H,36H,37H
DB      38H,39H,3AH,3BH,3CH,3DH,3EH,3FH
DB      40H,41H,42H,43H,44H,45H,46H,47H
DB      48H,49H,4AH,4BH,4CH,4DH,4EH,4FH
DB      50H,51H,52H,53H,54H,55H,56H,57H
DB      58H,59H,5AH,5BH,5CH,5DH,5EH,5FH
DB      60H,61H,62H,63H,64H,65H,66H,67H
DB      68H,69H,6AH,6BH,6CH,6DH,6EH,6FH
DB      70H,71H,72H,73H,74H,75H,76H,77H
DB      78H,79H,7AH,7BH,7CH,7DH,7EH,7FH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB      80H,81H,82H,83H,84H,85H,86H,87H
DB      88H,89H,8AH,8BH,8CH,8DH,8EH,8FH
DB      90H,91H,92H,93H,94H,95H,96H,97H
DB      98H,99H,9AH,9BH,9CH,9DH,9EH,9FH
DB      0A0H,0A1H,0A2H,0A3H,0A4H,0A5H,0A6H,0A7H
DB      0A8H,0A9H,0AAH,0ABH,0ACH,0ADH,0AEH,0AFH
DB      0B0H,0B1H,0B2H,0B3H,0B4H,0B5H,0B6H,0B7H
DB      0B8H,0B9H,0BAH,0BBH,0BCH,0BDH,0BEH,0BFH
DB      0C0H,0C1H,0C2H,0C3H,0C4H,0C5H,0C6H,0C7H
DB      0C8H,0C9H,0CAH,0CBH,0CCH,0CDH,0CEH,0CFH
DB      0E0H,0E1H,0E2H,0E3H,0E4H,0E5H,0E6H,0E7H
DB      0E8H,0E9H,0EAH,0EBH,0ECH,0EDH,0EEH,0EFH
DB      0F0H,0F1H,0F2H,0F3H,0F4H,0F5H,0F6H,0F7H
DB      0F8H,0F9H,0FAH,0FBH,0FCH,0FDH,0FEH,0FFH
DATA_FRE_H:
DB      03CH,09EH,0BDH,0CDH,0D8H,0E0H,0E4H,0E8H
DB      0EAH,0ECH,0EEH,0F0H,0F1H,0F2H,0F3H,0F3H
DB      0F4H,0F5H,0F5H,0F6H,0F6H,0F7H,0F7H,0F7H
DB      0F8H,0F8H,0F8H,0F9H,0F9H,0F9H,0F9H,0F9H
DB      0FAH,0FAH,0FAH,0FAH,0FAH,0FAH,0FAH,0FBH
DB      0FBH,0FBH,0FBH,0FBH,0FBH,0FBH,0FBH,0FBH
DB      0FCH,0FCH,0FCH,0FCH,0FCH,0FCH,0FCH,0FCH
DB      0FCH,0FCH,0FCH,0FCH,0FCH,0FCH,0FCH,0FCH
DB      0FCH,0FDH,0FDH,0FDH,0FDH,0FDH,0FDH,0FDH
DB      0FDH,0FDH,0FDH,0FDH,0FDH,0FDH,0FDH,0FDH
DB      0FDH,0FDH,0FDH,0FDH,0FDH,0FDH,0FDH,0FDH
DB      0FDH,0FDH,0FDH,0FDH,0FDH,0FDH,0FDH,0FEH
DB      0FEH,0FEH,0FEH,0FEH
DATA_FRE_L:
DB      0AFH,057H,097H,037H,0EFH,0BFH,0A7H,08FH
DB      083H,077H,06BH,05FH,027H,053H,01BH,0E3H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DB      047H,073H,0D7H,03BH,0B3H,021H,085H,0D7H
DB      02FH,07CH,0C3H,005H,043H,07DH,0B2H,0E4H
DB      014H,040H,06AH,092H,0B8H,0DBH,0FDH,01DH
DB      03DH,059H,074H,08FH,0A8H,0C0H,0D7H,0EDH
DB      003H,017H,02BH,03FH,054H,061H,073H,085H
DB      093H,0A1H,0B1H,0BFH,0CCH,0D9H,0E6H,0F2H
DB      0FEH,009H,016H,021H,02AH,035H,03FH,049H
DB      052H,05BH,065H,06DH,076H,07EH,086H,08EH
DB      096H,09DH,0A5H,0ACH,0B4H,0BAH,0C0H,0C7H
DB      0CDH,0D4H,0DAH,0E0H,0E6H,0EBH,0F1H,0F6H
DB      0FDH,001H,007H,00BH
DATA_SPE:
DB      0F8H,0F9H,0FAH,0FBH,0FCH,0FDH,0FEH,0FFH;
0123456789ABCDEF
TITLE_1:        DB      ' Programmable '
TITLE_2:        DB      'Mini Step Drive '
TITLE_3:        DB      ' SELECT MODE. '
TITLE_31:       DB      'FU#1/ HA#2/ MI#3'
TITLE_4:        DB      'MODE FULL STEP#1'
TITLE_5:        DB      'MODE HAFL STEP#2'
TITLE_6:        DB      'MODE MINI STEP#3'
TITLE_7:        DB      ' SELECT SPEED '
TITLE_8:        DB      ' (00 - XX)      '
TITLE_9:        DB      '  DUTY CYCLE  '
TITLE_10:       DB      ' SELECT ROTATE. '
TITLE_11:       DB      '# RIGH Or * LEFT'
TITLE_12:       DB      ' MODE FULL RIGH '
TITLE_13:       DB      ' MODE FULL LEFT '
TITLE_14:       DB      ' MODE HAFL RIGH '
TITLE_15:       DB      ' MODE HAFL LEFT '
TITLE_16:       DB      ' MODE MINI RIGH '
```

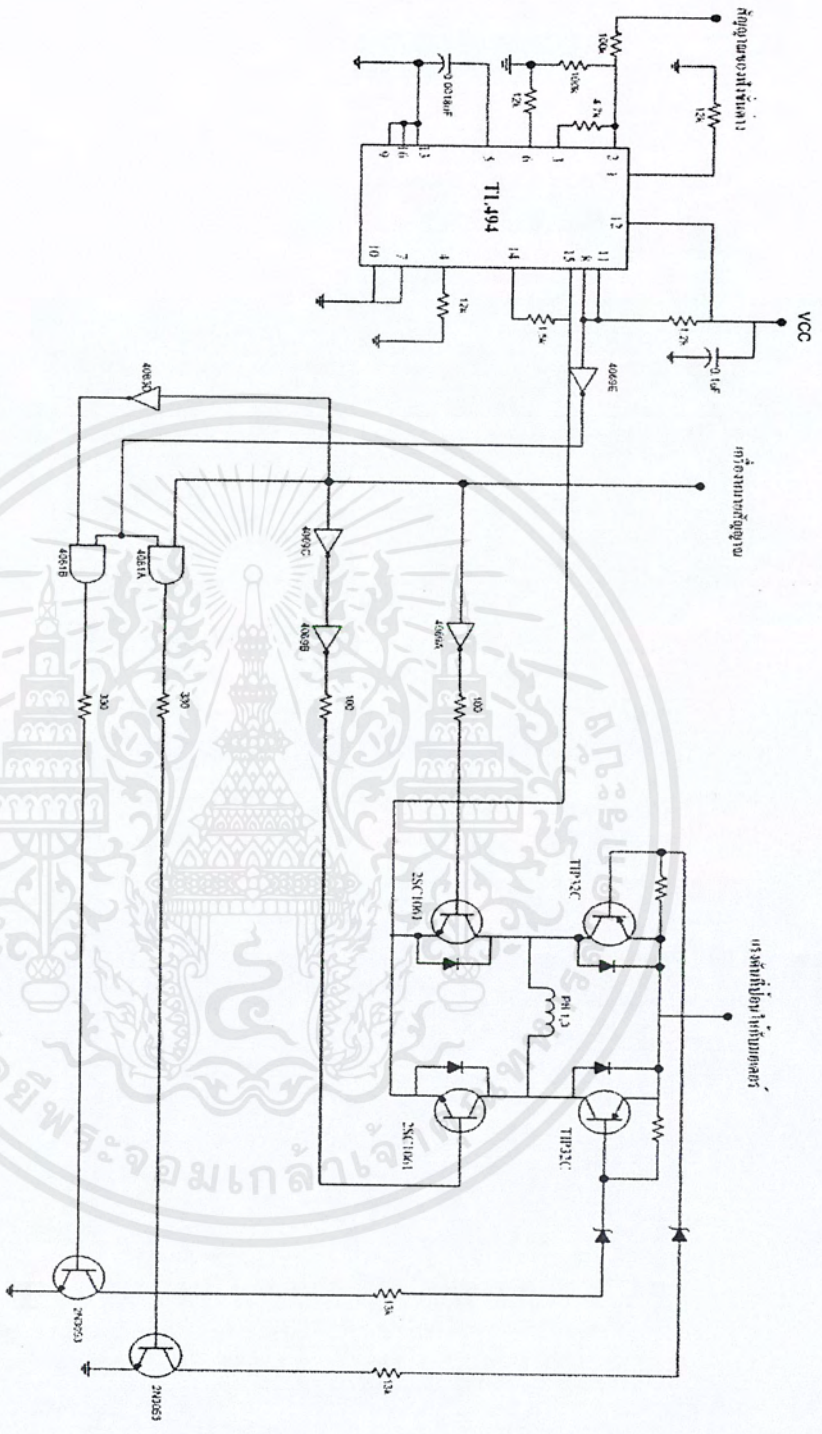
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
TITLE_17:    DB    ' MODE MINI LEFT '
TITLE_18:    DB    ' Freq. Counter  '
SCR_SPEED:   DB    'SPEED:  %'
SCR_RUN:     DB    'SPEED: rps'
SCR_OVER:    DB    'Input over range'
```

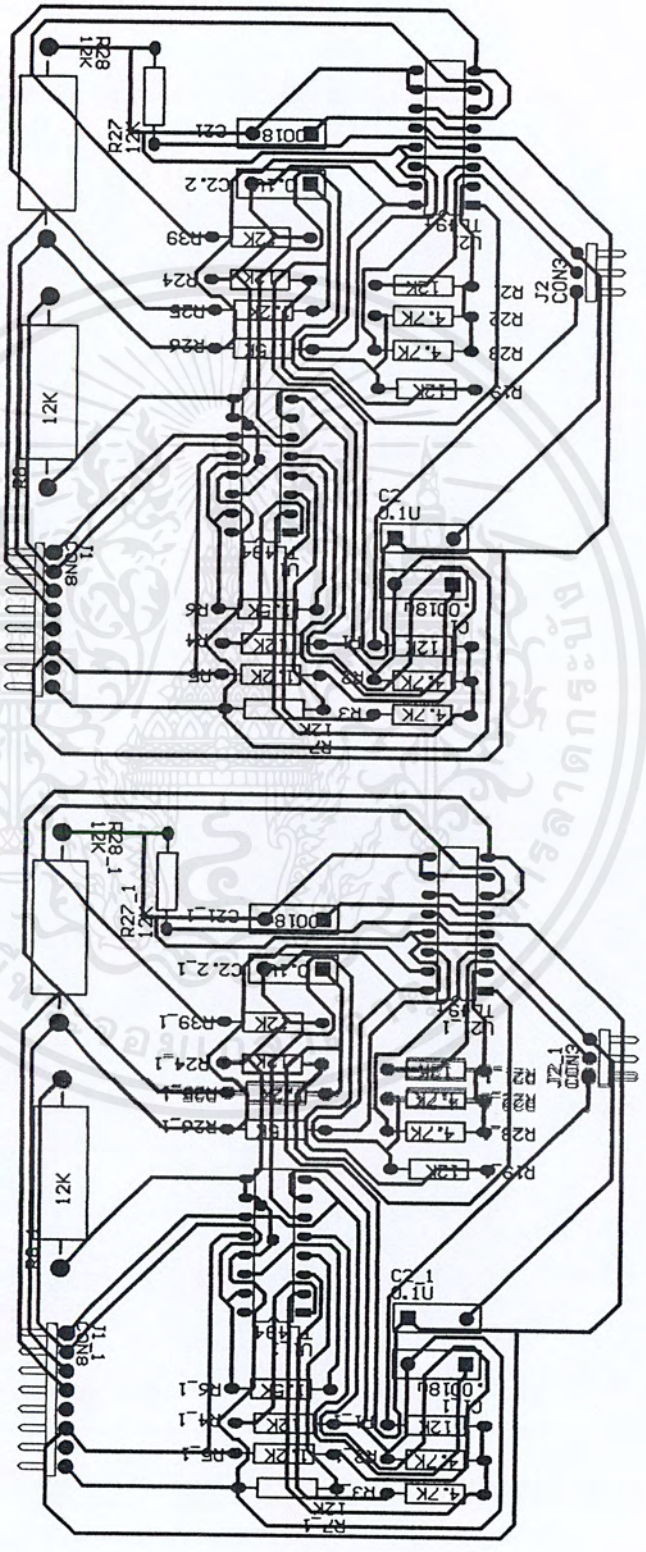

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
รายละเอียดของวงจร
ที่ใช้ในการขับเคลื่อนสเตปปิงมอเตอร์

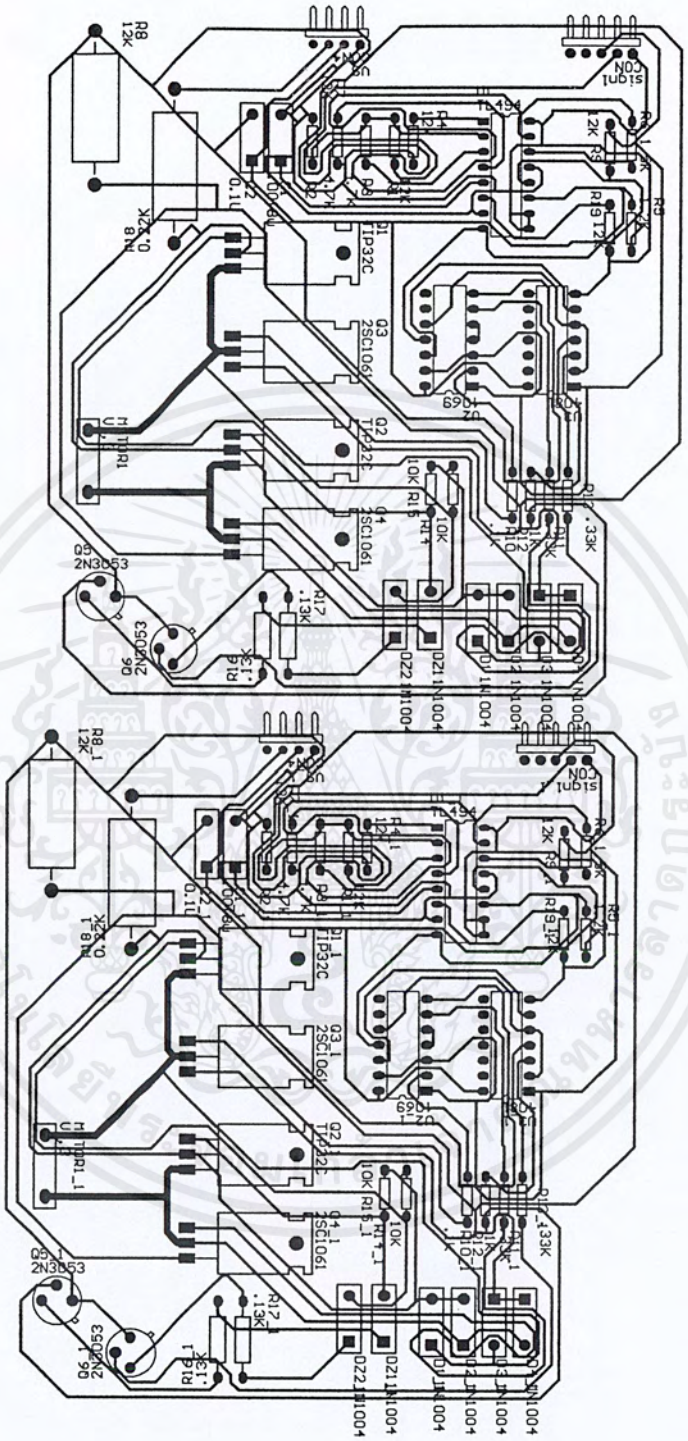
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้