

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องตัดและปอกสายไฟ  
CUTTING AND PEELING WIRE MACHINE

โดย  
นายดำรงศักดิ์ แสงเงิน  
นายรณกร เกียนมา

เลขหมู่..... 1  
เลขทะเบียน..... 62885  
วัน,เดือน,ปี..... 23 ส.ค. 2549

11/22/44

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาอิเล็กทรอนิกส์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องตัดและปอกสายไฟ  
CUTTING AND PEELING WIRE MACHINE



โดย  
นายดำรงศักดิ์ แสงเงิน รหัสประจำตัว 46015176  
นายธรรณกร เกียนมา รหัสประจำตัว 46015198

อาจารย์ที่ปรึกษา  
รศ.พนินฐา แซ่ตั้ง

ปฏิญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาอิเล็กทรอนิกส์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ ปีการศึกษา 2548

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องตัดและลอกสายไฟ

CUTTING AND PEELING WIRE MACHINE

ผู้จัดทำ

นายดำรงศักดิ์ แสงเงิน

นายรณกร เกียนมา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เครื่องตัดและปกสายไฟ

นายดำรงศักดิ์ แสงเงิน  
นาขรณกร เกียนมา  
อาจารย์ที่ปรึกษา รัช.ขนิษฐา แซ่ตั้ง  
ปีการศึกษา 2548

### บทคัดย่อ

เครื่องตัดและปกสายไฟเป็นเครื่องมือที่ควบคุมด้วยไมโครคอนโทรลเลอร์ ซึ่งนำไปควบคุมสแต็ปปีงมอเตอร์จำนวน 3 ตัว โดยสแต็ปปีงมอเตอร์ 2 ตัวแรกจะเป็นจุดที่ใช้สำหรับเลื่อนความยาวของสายไฟตามที่ต้องการ และอีกตัวจะใช้สำหรับควบคุมใบมีดที่จะใช้ตัดและปก สามารถกำหนดค่าความยาวและระยะปก รวมทั้งจำนวนสายไฟได้ โดยใช้คีย์แพดและแสดงข้อมูลต่างๆ ที่จอแสดงผล การควบคุมจะเป็นแบบครึ่งสเตป เพื่อให้ได้ระยะความยาวที่ตัดและปกมีความแม่นยำถึง 0.125 มิลลิเมตร สำหรับส่วนประกอบทางด้านเครื่องกลได้จำลองไว้สำหรับการแสดงการตัดและปกสายไฟที่มีขนาดเส้นผ่านศูนย์กลางสูงสุด 1 มิลลิเมตร เพื่อให้เหมาะสมกับการใช้ในงานอุตสาหกรรมทางด้านแผ่นวงจรพิมพ์ แต่ไม่ได้ออกแบบไว้สำหรับการตัดและปกสายไฟขนาดต่างๆ ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## CUTTING AND PEELING WIRE MACHINE

Mr.Damrongsak Sangngeon

Mr.Ronnakorn Kianma

Assoc.Prof.Khanittha Saetang (Advisor)

Educational Year 2005

### Abstract

Cutting and peeling wire machine is an instrument which control by microcontroller. It control 3 stepping motors that 2 of them are used for sliding length of wire and a last of them is used for control the knife that is cut and peeled. To determine length of wire and length of peeling values, number of wires by use keypad and show data on module LCD. Half-step Controlled in order to the precision from cutting and peeling is 0.125 mm. The mechanism model cutting and peeling demonstrate for 1 mm wires diameter. It is suitable used for in the PCB industry. It doesn't design for normal cutting and peeling wires.

## กิติกรรมประกาศ

โครงการนี้ได้จัดทำขึ้นจนประสบความสำเร็จเนื่องจากได้รับความช่วยเหลือจากหลายๆ ฝ่ายด้วยกันโดยบุคคลแรกที่ขอกู้คือ รศ.ขนิษฐา แซ่ตั้ง อาจารย์ที่ปรึกษาของโครงการนี้ที่ให้ความช่วยเหลือพวกเราทุกเรื่องคอยให้คำแนะนำให้ความรู้ช่วยแก้ปัญหา พวกเราต้องขอขอบพระคุณอาจารย์เป็นอย่างสูง

บุคคลที่มีพระคุณอย่างสูงยิ่งต่อพวกเราคือ บิดา มารดาที่ทำให้พวกเรามีวันนี้ ซึ่งพวกท่านคือกำลังใจที่มีค่ายิ่งเพราะท่านได้ให้การเลี้ยงดูที่ดีให้การอบรมสั่งสอนและเอาใจใส่จนทำให้เราเป็นคนที่ดีในวันนี้ พวกเราจะขอสำนึกในพระคุณอันสูงค่านี้ตลอดไป

และสุดท้ายต้องขอขอบคุณเพื่อนๆ ที่คอยให้ความช่วยเหลือให้คำแนะนำที่เป็นประโยชน์ พวกเราขอขอบคุณมากครับ



นายดำรงศักดิ์ แสงเงิน

นายรณกร เกียนมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
บทคัดย่อ	I
ABSTRACT	II
สารบัญ	III
สารบัญรูปภาพ	V
สารบัญตาราง	VII
บทที่ 1 บทนำ	1
บทที่ 2 อุปกรณ์หลักที่นำมาประยุกต์ใช้	2
2.1 สเตปปี้งมอเตอร์	2
2.1.1 หลักการทำงานเบื้องต้นของสเตปปี้งมอเตอร์	2
2.1.2 ประเภทของสเตปปี้งมอเตอร์	3
2.1.2.1 สเตปปี้งมอเตอร์ประเภทคาร์ริกเตนซ์แปรค่าได้ (Variable Reluctance Stepping Motor)	3
2.1.2.2 สเตปปี้งมอเตอร์ประเภทแม่เหล็กถาวร (Permanent magnet stepping motor หรือ PM stepping motor)	5
2.1.2.3 สเตปปี้งมอเตอร์ประเภทไฮบริด (Hybrid stepping motor)	6
2.1.3 รูปแบบการขับสเตปปี้งมอเตอร์	7
2.1.3.1 การขับแบบกระตุ้นหนึ่งเฟส (ONE-PHASE EXCITATION)	7
2.1.3.2 การขับแบบกระตุ้นสองเฟส (TWO-PHASE EXCITATION)	8
2.1.3.3 การขับแบบกระตุ้นหนึ่งเฟสสลับสองเฟส (ONE-TWO-PHASE EXCITATION)	8
2.2 ไอซีเบอร์ UCN5804B	9
2.2.1 การใช้งาน UCN5804B	10
2.3 ไมโครคอนโทรลเลอร์ MCS-51	12
2.3.1 คุณสมบัติที่สำคัญของ MCS-51	12
2.3.2 โครงสร้างของไมโครคอนโทรลเลอร์ตระกูล MCS-51	12
2.3.3 โครงสร้างหน่วยความจำ	14
2.3.4 หน่วยความจำภายนอก (External Memory)	17
2.3.5 การติดต่อกับหน่วยความจำโปรแกรมภายนอก	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
บทที่ 3 หลักการทำงานของเครื่องปอกสายไฟ	19
3.1 ระบบควบคุม	19
3.2 วงจรขับสเตปป์มอเตอร์	23
3.3 สเตปป์มอเตอร์และอุปกรณ์ทางกล	24
3.4 ภาคจ่ายไฟ	25
3.5 ขั้นตอนการคำนวณและการออกแบบโปรแกรม	25
3.6 ลำดับขั้นตอนการตัดและต่อสายไฟ	26
3.7 Flow Chart การทำงาน	30
บทที่ 4 การทดลอง	35
บทที่ 5 สรุปและวิจารณ์ผล	37
5.1 สรุปผลการทดลอง	37
5.2 ปัญหาและแนวทางแก้ไข	37
5.3 แนวทางในการพัฒนา	38
ภาคผนวก	39
ภาคผนวก ก	40
ภาคผนวก ข	59
กิตติกรรมประกาศ	65
เอกสารอ้างอิง	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 แสดงส่วนประกอบแบบง่าย ๆ ของสเตปป์มอเตอร์มูมโรเตอร์เคลื่อนที่ 60 องศา	3
รูปที่ 2.2 แสดงโครงสร้างอย่างง่ายของสเตปป์มอเตอร์ประเภทคาร์ลัคแดนซ์แปรค่าได้ 3 เฟส	3
รูปที่ 2.3 แสดงตำแหน่งสมดุลมื่อเฟสที่ 1 ถูกกระตุ้น	4
รูปที่ 2.4 แสดงแรงบิดรีสตอริง (Restoring torque) ที่มีทิศทวนเข็มนาฬิกา	4
รูปที่ 2.5 แสดงการเคลื่อนที่ของโรเตอร์ขณะที่เฟสที่ 1 ถูกเปิดออกและเฟสที่ 2 ถูกจ่ายกระแสทันที	5
รูปที่ 2.6 แสดงการเคลื่อนที่ที่สเตปป์มอเตอร์ประกอบในมอเตอร์ประเภทคาร์ลัคแดนซ์แปรค่าได้	5
รูปที่ 2.7 แสดงโครงสร้างเบื้องต้นของสเตปป์มอเตอร์แบบแม่เหล็กถาวร	6
รูปที่ 2.8 แสดงการต่อวงจรขั้วอย่างง่ายสำหรับมอเตอร์สองเฟส	6
รูปที่ 2.9 แสดงการเคลื่อนที่ของโรเตอร์เมื่อถูกกระตุ้นในทิศตามเข็มนาฬิกา	6
รูปที่ 2.10 แสดงโครงสร้างโรเตอร์ของสเตปป์มอเตอร์ประเภทไฮบริด	7
รูปที่ 2.11 แสดงทิศทางการสร้างสนามแม่เหล็กของสเตปป์มอเตอร์ประเภทไฮบริด	7
รูปที่ 2.12 โครงสร้างภายในของ UC5804B	10
รูปที่ 2.13 แสดงรูปโครงสร้างของไมโครคอนโทรลเลอร์ตระกูล MCS-51	12
รูปที่ 2.14 ไดอะแกรมกลุ่มสัญญาณที่ใช้อ่านข้อมูล	17
รูปที่ 3.1 บล็อกไดอะแกรมของเครื่องปลดสายไฟ	19
รูปที่ 3.2 ลักษณะการเชื่อมต่อกันของอุปกรณ์ในระบบควบคุม	20
รูปที่ 3.3 ตำแหน่งปุ่มกดต่างๆ ของ Keypad	21
รูปที่ 3.4 แสดงหน้าจอ LCD ขณะเปิดเครื่อง	22
รูปที่ 3.5 หน้าจอ LCD เมื่อกดปุ่ม 'X หรือ Y'	22
รูปที่ 3.6 หน้าจอ LCD เมื่อกดปุ่ม 'Z' ที่ Keypad	22
รูปที่ 3.7 หน้าจอ LCD เมื่อกดปุ่ม 'START' ที่ Keypad	22
รูปที่ 3.8 วงจรขับสเตปป์มอเตอร์	23
รูปที่ 3.9 แสดงสเตปป์มอเตอร์และอุปกรณ์ทางกล	24
รูปที่ 3.10 ภาค Power Supply	25
รูปที่ 3.11 รูปแบบการวางสายไฟตอนเริ่มต้นที่ถูกต้อง	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
รูปที่ 3.12 ใบมีดส่วนบนเคลื่อนลงมาตัดสายไฟ	26
รูปที่ 3.13 ชุดลูกยางของ M1 เลื่อนสายไฟไปข้างหน้ายาวเท่ากับ X	26
รูปที่ 3.14 ใบมีดส่วนบนเคลื่อนลงมาตัดจนวนของสายไฟ	27
รูปที่ 3.15 ชุดลูกยางของ M1 เลื่อนสายไฟกลับด้วยความยาวเท่ากับ $X + (1 \text{ mm})$	27
รูปที่ 3.16 ชุดลูกยางของ M3 จะหมุนรอบเมื่อสายไฟเลื่อนมาได้ $X + (55 \text{ mm})$	28
รูปที่ 3.17 ชุดลูกยางของ M1 และ M3 จะหยุดหมุนเมื่อสายไฟเลื่อนมาได้ระยะเท่ากับ $2X + Y$	28
รูปที่ 3.18 ใบมีดส่วนบนเคลื่อนลงมาตัดสายไฟให้ขาดออกจากกัน	28
รูปที่ 3.19 ชุดลูกยางของ M1 และ M3 เลื่อนสายไฟกลับตามความยาวที่กำหนด	28
รูปที่ 3.20 ลักษณะของสายไฟที่ปอกเสร็จแล้ว	29
รูปที่ 3.21 Flowchart แสดงการทำงานของเครื่องตัดและปอกสายไฟ	30
รูปที่ 3.22 Flowchart แสดงการรับค่าต่างๆ จาก KEYPAD	31
รูปที่ 3.23 Flowchart แสดงการคำนวณหาค่าพิกัดและระยะต่างๆ	32
รูปที่ 3.24 Flowchart แสดงขั้นตอนการตัดและปอกสายไฟ	33
รูปที่ 3.25 Flowchart แสดงขั้นตอนการตัดและปอกสายไฟ(ต่อ)	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงการถ่ายกระแสแบบกระตุ้นหนึ่งเฟส	8
ตารางที่ 2.2 แสดงการถ่ายกระแสแบบกระตุ้นสองเฟส	8
ตารางที่ 2.3 แสดงการถ่ายกระแสแบบกระตุ้นหนึ่งเฟสลับสองเฟส	9
ตารางที่ 2.4 แสดงลำดับการขับแบบหนึ่งเฟส	10
ตารางที่ 2.5 แสดง ลำดับการขับแบบสองเฟส	11
ตารางที่ 2.6 แสดง ลำดับการขับแบบครึ่งเฟส	11
ตารางผลการทดลองที่ 1 ของการตัดสายที่มีความยาว 10.0 cm	35
ตารางผลการทดลองที่ 2 ของการตัดสายที่มีความยาว 15.0 cm	35
ตารางผลการทดลองที่ 3 ของการตัดสายที่มีความยาว 20.0 cm	36
ตารางผลการทดลองที่ 4 ของการตัดสายที่มีความยาว 25.0 cm	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

ปัจจุบันนี้งานทางด้านอิเล็กทรอนิกส์มีความสำคัญต่อชีวิตประจำวันอย่างมาก ซึ่งมนุษย์ได้นำเอาความรู้พื้นฐานทางอิเล็กทรอนิกส์มาวิเคราะห์ แกะไข คัดแปลงและประดิษฐ์เครื่องมือเครื่องใช้ที่อำนวยความสะดวกต่างๆ มากมาย ในการผลิตงานทางด้านอิเล็กทรอนิกส์นั้นมีส่วนประกอบหลายส่วน ยกตัวอย่างเช่นแผงวงจรอิเล็กทรอนิกส์ ถือได้ว่าเป็นส่วนที่สำคัญอย่างหนึ่ง แต่การจะได้มาซึ่งแผงวงจรที่สำเร็จรูปนั้น จำเป็นต้องมีการออกแบบวงจรและการวางอุปกรณ์ต่างๆ ให้เหมาะสม นอกจากนั้นการเชื่อมโยงอุปกรณ์ต่างๆ เข้าด้วยกันก็มีส่วนสำคัญโดยอาจต้องนำสายไฟมาใช้ในการเชื่อมต่ออุปกรณ์เข้าด้วยกัน เพื่อให้สามารถใช้งานวงจรได้อย่างสมบูรณ์

สายไฟฟ้าเป็นอุปกรณ์อย่างหนึ่งที่ใช้ในการเชื่อมต่ออุปกรณ์ไฟฟ้าชนิดต่างๆ เข้าด้วยกันซึ่งสายไฟที่ใช้มีหลายขนาดแตกต่างกันไปขึ้นอยู่กับชนิดของงานที่จะนำไปใช้ ถ้างานที่ทำเป็นงานที่ต้องใช้สายไฟในปริมาณที่ไม่มากก็สามารถที่จะทำการตัดสายไฟและปอกสายไฟเพื่อนำไปใช้ได้ทันที แต่่างานภายในโรงงานอุตสาหกรรมนั้น ความต้องการในการใช้สายไฟมีปริมาณมากกว่า ซึ่งถ้าการตัดและปอกสายไฟนั้นใช้แรงงานคนจะทำให้งานล่าช้าเสียทั้งเวลาและแรงงานในการปฏิบัติงาน จึงจำเป็นที่จะต้องมียุกรณ์ที่ช่วยในการตัดและปอกสายไฟให้มีความรวดเร็วขึ้น

### 1.1 วัตถุประสงค์ของโครงการ

โครงการวิศวกรรมนี้ได้ทำเครื่องตัดและปอกสายไฟขึ้น โดยมีวัตถุประสงค์เพื่อทำการตัดและปอกสายไฟที่สามารถกำหนดความยาวที่ต้องการตัดและระยะของความยาวที่ต้องการปอกได้

### 1.2 ขอบเขตของโครงการ

1. สามารถตัดและปอกสายไฟได้ขนาดเดียว
2. ออกแบบมาเพื่อให้สามารถปอกสายไฟได้ทั้ง 2 ข้าง
3. สามารถตัดสายไฟได้ที่ความยาวตั้งแต่ 6 เซนติเมตร ถึง 99.9 เซนติเมตร
4. สามารถกำหนดความเร็วในการตัดและปอกได้

### 1.3 ผลที่คาดว่าจะได้รับ

1. นำไปใช้ตัดและปอกสายไฟในขนาดที่กำหนดได้
2. สร้างความสะดวกสบาย รวดเร็ว และลดระยะเวลาในการปอกสายไฟในปริมาณมากๆ
3. นำไปใช้เป็นเครื่องต้นแบบในการพัฒนาเครื่องตัดและปอกสายไฟต่อไป

## บทที่ 2

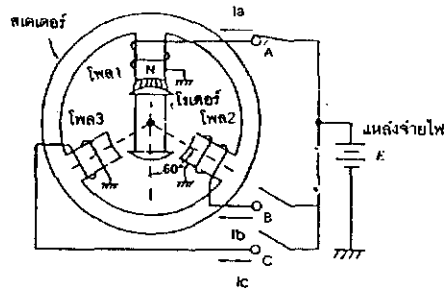
### อุปกรณ์หลักที่นำมาประยุกต์ใช้

#### 2.1 สเตปปีงมอเตอร์

สเตปปีงมอเตอร์เป็นอุปกรณ์ทางไฟฟ้าที่มีลักษณะพิเศษเป็นตัวต้นกำลังแบบดิจิทัลที่สำคัญอย่างหนึ่ง สามารถทำงานโดยเปลี่ยนพลังงานทางไฟฟ้าเป็นพลังงานกลในรูปของการเคลื่อนที่ ซึ่งมีอินพุทเป็นกลุ่มของแรงดันไบนารี และเอาต์พุทเป็นลักษณะของการเคลื่อนที่เชิงมุมเป็นสเตปที่มีค่ามุมคงที่ โดยการจ่ายแรงดันหรือกระแสแก่ขดลวดแต่ละเฟสตามลำดับอย่างถูกต้อง การนำไปใช้งานเมื่อต้องการควบคุมการเคลื่อนที่และตำแหน่งที่ต้องการความถูกต้องแม่นยำ การเคลื่อนที่แต่ละสเตปเกี่ยวข้องกับสัญญาณพัลส์แต่ละอันที่จ่ายแก่ขดลวดสเตเตอร์ทั้งหมด จำนวนสเตปที่เคลื่อนที่จะเท่ากับจำนวนสัญญาณพัลส์ที่จ่ายแก่มอเตอร์ สามารถเคลื่อนที่ทั้งตามเข็มและทวนเข็มนาฬิกาขึ้นอยู่กับการจัดลำดับการจ่ายพัลส์แก่ขดลวด ตัวอย่างการใช้ เช่น เป็นอุปกรณ์ในเครื่องมือทางกล x-y พลอตเตอร์ พรินเตอร์แบบอนุกรม เครื่องพิมพ์ดีด อุปกรณ์ต่อพ่วงทางคอมพิวเตอร์ เป็นต้น

##### 2.1.1 การทำงานเบื้องต้นของสเตปปีงมอเตอร์

จากรูปที่ 2.1 แสดงส่วนประกอบแบบง่าย ๆ ของสเตปปีงมอเตอร์ ประกอบด้วยขั้วโพลสเตเตอร์ 3 ขั้วและโรเตอร์ 2 ขั้ว ที่ทำจากเหล็กอ่อน ขดลวดสเตเตอร์ถูกต่อกับแหล่งจ่ายไฟฟ้าโดยผ่านสวิตช์ 3 ตัว A,B,C เมื่อสวิตช์เปิดวงจรมอเตอร์อยู่ที่ตำแหน่งใดๆก็ได้ แต่เมื่อสวิตช์ A ปิดจะทำให้เกิดสนามแม่เหล็กที่สร้างโดยโพล 1 จะดึงดูดโรเตอร์ให้อยู่ในแนวเส้นตรงเดียวกับขั้วโพลเหมือนแสดงในรูป แต่เมื่อเราเปิดสวิตช์ที่ A และปิดสวิตช์ที่ B โรเตอร์ก็จะอยู่ในแนวเดียวกับขั้วโพล 2 ดังนั้นมอเตอร์จะหมุนในทิศทวนเข็มนาฬิกา 60 องศา เมื่อเปิดสวิตช์ที่ B ปิดสวิตช์ที่ C จะทำให้โรเตอร์อยู่ในแนวเดียวกับขั้วโพล 3 และหมุนทวนเข็มนาฬิกา ดังนั้นเราสามารถทำให้โรเตอร์หมุนในทิศทวนเข็มนาฬิกาได้ 60 องศาต่อสเตป โดยการเปิดและปิดสวิตช์ไปตามลำดับ A,B,C,A,B,C,..... นอกจากนั้นเราสามารถกลับทิศการหมุนได้โดยการกลับลำดับการสวิตช์เป็น A,C,B,A,C,B,..... เพื่อให้ตำแหน่งสุดท้ายของโรเตอร์ยังคงอยู่โดยการยังคงปิดสวิตช์สุดท้ายอยู่ จะทำให้โรเตอร์ยังอยู่ที่ตำแหน่งนี้และเป็นการป้องกันการกระแทกกระเทือนจากแรงภายนอก ในสถานะคงตัวนี้มอเตอร์ยังคงถูกล็อกอยู่โดยแรงภายนอกไม่มากเกินไป เรียกว่าโฮลดิ้งทอร์ก (Holding Torque) ของมอเตอร์ ในการเคลื่อนที่ จากตำแหน่งหนึ่งไปยังตำแหน่งต่อไป การเคลื่อนที่ของโรเตอร์จะถูกกระแทกกระเทือนโดยอินเนอร์เชียล (Inertial) และแรงเสียดทานภายนอกได้



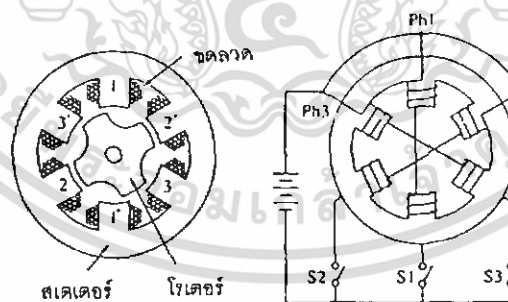
รูปที่ 2.1 แสดงส่วนประกอบแบบง่าย ๆ ของสเตปป์ิงมอเตอร์ 60 องศา

### 2.1.2 ประเภทของสเตปป์ิงมอเตอร์

สเตปป์ิงมอเตอร์แบ่งออกเป็นหลายประเภทที่สอดคล้องกับลักษณะ โครงสร้างทางกลและหลักเกณฑ์การทำงานดังต่อไปนี้

#### 2.1.2.1 สเตปป์ิงมอเตอร์ประเภทอาร์ลัคเตนซ์แปรค่าได้ (Variable Reluctance Stepping Motor)

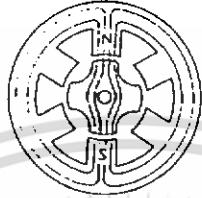
สเตปป์ิงมอเตอร์ประเภทอาร์ลัคเตนซ์แปรค่าได้ เป็นประเภทสเตปป์ิงที่ง่ายที่สุด เพื่อให้ง่ายต่อการเข้าใจพื้นฐานจะใช้มอเตอร์ที่แสดงในรูปที่ 2.2 ซึ่งเป็นมอเตอร์ 3 เฟส มีซี่ฟันสเตเตอร์เท่ากับ 6 ซี่ฟันที่อยู่ตรงข้ามกัน 2 ซี่ทำมุมกัน 180 องศาจะเป็นเฟสเดียวกัน ขดลวดที่พันอยู่บนซี่ฟันแต่ละคู่จะต่อกันแบบอนุกรมหรือขนาน (ในรูปจะแบบอนุกรม) โรเตอร์มีซี่ฟันเท่ากับ 4 ปกติแล้วแกนของสเตเตอร์และโรเตอร์เป็นแผ่นเหล็กบางที่มีส่วนผสมของซิลิคอนวางซ้อนกัน (Laminate silicon steel) แต่โรเตอร์ที่ทำจากแท่งเหล็กซิลิคอนจะเป็นที่นิยมใช้กันมากกว่า วัสดุที่ใช้ทำสเตเตอร์และโรเตอร์ต้องมีค่าซึมซาบทางแม่เหล็ก (Permeability) สูงและให้เส้นแรงเหล็ก (Magnetic flux) ผ่านได้ง่ายเพื่อให้เกิดแรงเคลื่อนแม่เหล็ก (Magnetomotive force) สูงสุด



รูปที่ 2.2 แสดงโครงสร้างอย่างง่ายของสเตปป์ิงมอเตอร์ประเภทอาร์ลัคเตนซ์แปรค่าได้ 3 เฟส

ในรูปที่ 2.2 สมมติให้ซี่ฟันแต่ละคู่ของสเตปป์ิงมอเตอร์มีขั้วที่ตรงกันข้ามคือ 1, 2, 3 เป็นขั้วเหนือ (N) และ 1', 2', 3' เป็นขั้วใต้ (S) เมื่อแต่ละเฟสถูกกระตุ้นให้ทำงาน

กระแสในแต่ละเฟสจะถูกควบคุมโดยการเปิดและปิดสวิตช์ตามลำดับ ถ้ากระแสถูกจ่ายให้เฟส 1 หรือพูดว่าเฟส 1 ถูกกระตุ้นฟลักซ์แม่เหล็กจะเกิดขึ้นดังรูปที่ 2.3 โดยซี่ฟันโรเตอร์คู่หนึ่งจะมีตำแหน่งตรงกับซี่ฟัน 1 และ 1' ของสเตเตอร์ เมื่อโรเตอร์อยู่ในแนวเดียวกับสเตเตอร์ค่าความต้านทานแม่เหล็ก (Magnetic reluctance) จะมีค่าน้อยที่สุดและที่ตำแหน่งนี้มอเตอร์จะหยุดหรือเรียกว่าตำแหน่งสมมูลเมื่อมีแรงภายนอกกระทำกับโรเตอร์ให้ออกจากตำแหน่งสมมูลในทิศตามเข็มนาฬิกา ดังในรูปที่ 2.4

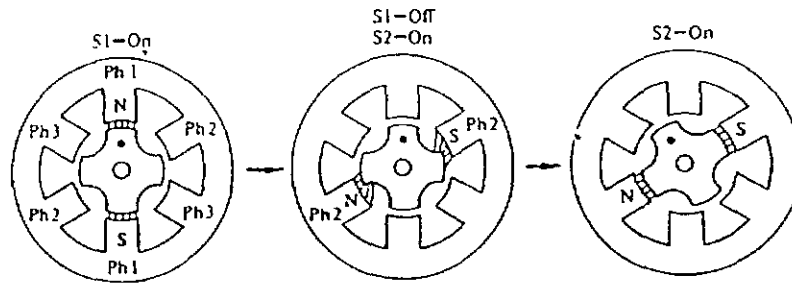


รูปที่ 2.3 แสดงตำแหน่งสมมูลเมื่อเฟสที่ 1 ถูกกระตุ้น

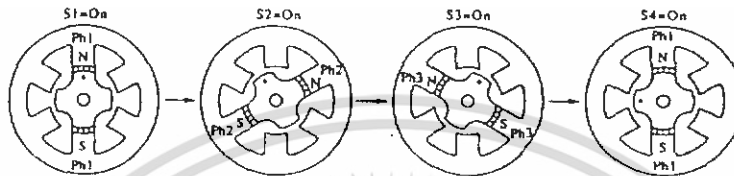


รูปที่ 2.4 แสดงแรงบิดรีตอร์นิ่ง (Restoring torque) ที่มีทิศทวนเข็มนาฬิกา

ในภาพนี้แรงบิดภายนอกมีทิศตามเข็มนาฬิกาทำให้โรเตอร์เคลื่อนที่ไปในทิศทางดังกล่าวด้วยบิดรีตอร์นิ่งจะมีทิศทวนเข็มนาฬิกาตรงกันข้ามกับแรงบิดภายนอกเพื่อรักษาตำแหน่งโรเตอร์ให้ตรงกับตำแหน่งซี่ฟันของสเตเตอร์มากที่สุด ขณะที่ตำแหน่งซี่ฟันของโรเตอร์และสเตเตอร์เบี่ยงเบนไปจากตำแหน่งสมมูลของเฟสที่ถูกกระตุ้นค่าความต้านทานแม่เหล็กจะมีค่ามาก จากรูปที่ 2.5 แสดงลักษณะการเคลื่อนที่ของโรเตอร์ขณะที่เฟสที่ 1 ถูกเปิดออกและเฟสที่ 2 ถูกจ่ายกระแสทันที ค่าความต้านทานแม่เหล็กของมอเตอร์จะเพิ่มขึ้นทันที โรเตอร์จะเคลื่อนที่เป็น สเตปด้วยมุม 30 องศาในทิศทวนเข็มนาฬิกาเพื่อให้ความต้านทานแม่เหล็กมีค่าน้อยที่สุด การเคลื่อนที่ผ่านมุม สเตปเมื่อมีการสวิตช์การกระตุ้นแต่ละครั้งเรียกว่า สเตป หลังจากโรเตอร์เคลื่อนที่ครบ 3 สเตป โรเตอร์จะเคลื่อนที่กลับมายังตำแหน่งจุดเริ่มต้นเดิมดังแสดงในรูปที่ 2.6



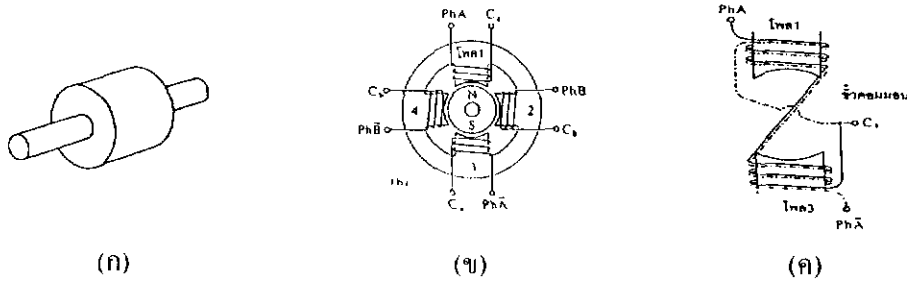
รูปที่ 2.5 แสดงการเคลื่อนที่ของโรเตอร์ขณะที่เฟสที่ 1 ถูกเปิดออกและเฟสที่ 2 ถูกจ่ายกระแสทันที



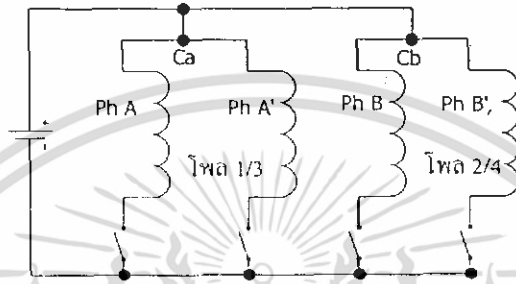
รูปที่ 2.6 แสดงการเคลื่อนที่ที่สเต็ปครบรอบในมอเตอร์ประเภทรีลัคแตนซ์แปรค่าได้

2.1.2.2 สเต็ปป์มอเตอร์ประเภทแม่เหล็กถาวร (Permanent magnet stepping motor หรือ PM stepping motor)

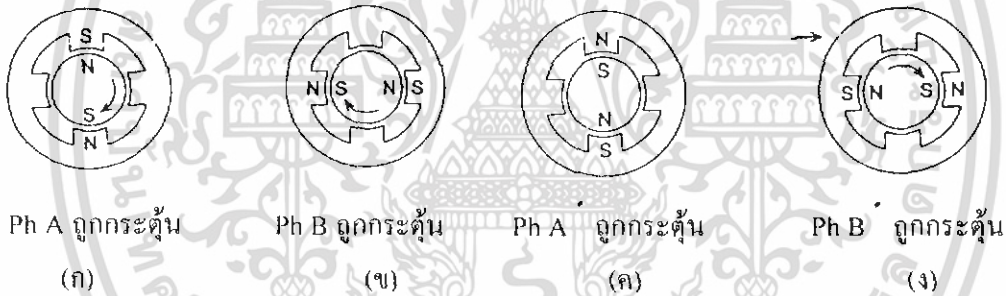
สเต็ปป์มอเตอร์ประเภทที่โรเตอร์ทำจากแม่เหล็กถาวร ถูกเรียกว่า “สเต็ปป์มอเตอร์ประเภทแม่เหล็กถาวร” แสดงโครงสร้างเบื้องต้นดังรูปที่ 2.7 ซึ่งใช้โรเตอร์เป็นแม่เหล็กถาวรแบบทรงกระบอก มีสเตเตอร์ 4 โพลหรือซี่ฟัน แต่ละซี่ฟันจะมีขดลวดพันล้อมรอบอยู่ดังแสดงในรูปที่ 2.7 (ก) และ (ข) ตามลำดับ รูป 2.7 (ค) แสดงการพันขดลวดจริง ซึ่งเรียกว่า “การพันแบบไบโพลาร์ (Bipolar)” โดยขดลวดสองเส้นวางทับกันถูกพันในลักษณะพันแบบเส้นเดียวบนโพล 1 และ โพล 3 โดยขดลวดสองเส้นนี้ถูกแยกออกจากกันเป็นอิสระที่ปลายแต่ละข้างและที่ปลายขั้วอีกข้างหนึ่งที่เขียนว่า Ca, Cb จะต่อเป็นขั้วร่วมต่อเข้ากับขั้วบวกของแหล่งจ่ายไฟซึ่งแสดงดังรูปการสวิทซ์ในรูปที่ 2.8 เมื่อขดลวด Ph A ที่เป็นเส้นที่ข (รูปที่ 2.7(ข)) ถูกกระตุ้นโพล 1 จะเป็นขั้วเหนือ (N) และ โพล 2 จะเป็นขั้วใต้ (S) ถ้า Ph A ถูกกระตุ้นขั้วจะกลับกัน การกระตุ้นขดลวดตามลำดับ Ph A, Ph B, Ph A, Ph B โรเตอร์จะหมุนตามเข็มนาฬิกาเหมือนแสดงดังรูปที่ 2.9 ทำให้มีมุมการสเต็ป 90 องศา ถ้าจำนวนฟันของสเตเตอร์และขั้วแม่เหล็กเพิ่มเป็น 2 เท่า จะทำให้มุมการสเต็ปเป็น 45 องศา มอเตอร์แบบแม่เหล็กถาวรจะมีแรงบิดมากกว่ามอเตอร์แบบค่ารีลัคแตนซ์แปรค่าได้เมื่อมีขนาดเท่ากัน



รูปที่ 2.7 แสดงโครงสร้างเบื้องต้นของสเต็ปปีงมอเตอร์แบบแม่เหล็กถาวร



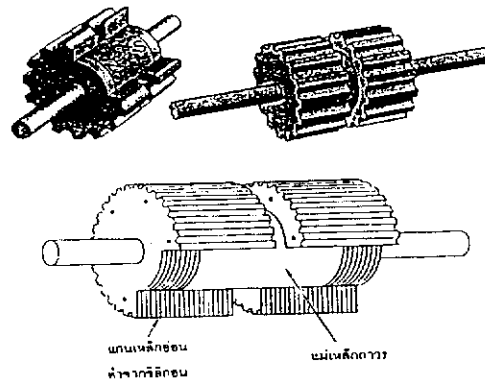
รูปที่ 2.8 แสดงการต่อวงจรขั้วอย่างง่ายสำหรับมอเตอร์สองเฟส



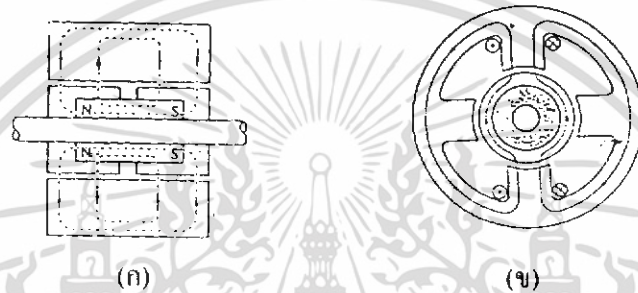
รูปที่ 2.9 แสดงการเคลื่อนที่ของโรเตอร์เมื่อถูกกระตุ้นในทิศตามเข็มนาฬิกา

2.1.2.3 สเต็ปปีงมอเตอร์ประเภทไฮบริด (Hybrid stepping motor)

สเต็ปปีงมอเตอร์ประเภทนี้มีหลักการทำงานผสมผสานกันระหว่างสเต็ปปีงมอเตอร์ประเภทแม่เหล็กถาวรกับประเภทคาร์ลิกต์แดนซ์แปรค่าได้ จึงได้ชื่อว่า “สเต็ปปีงมอเตอร์ประเภทไฮบริด” เพื่อให้ได้มุมสเต็ปที่เล็กและมีค่าแรงบิดสูงจากมอเตอร์ขนาดเล็ก โครงสร้างของสเตเตอร์จะเหมือนกับสเต็ปปีงมอเตอร์ประเภทคาร์ลิกต์แดนซ์แปรค่าได้ ส่วนที่สำคัญของสเต็ปปีงมอเตอร์ประเภทไฮบริดอยู่ที่โครงสร้างของโรเตอร์แสดงดังรูปที่ 2.10 คือมีแม่เหล็กรูปทรงกระบอกที่วางตัวตามแนวแกนโรเตอร์ มีทิศทางการสร้างสนามแม่เหล็กดังรูปที่ 2.11 (ก) โดยแต่ละโพลของแม่เหล็กจะถูกล้อมรอบด้วยซี่เหล็กอ่อนที่ทำจากซิลิคอน ส่วนสนามแม่เหล็กสร้างโดยขดลวดสเตเตอร์เป็นสนามแบบเฮเทอโรโพลาร์ (Heteropolar field) เหมือนดังรูปที่ 2.11 (ข)



รูปที่ 2.10 แสดง โครงสร้าง โรเตอร์ของสเตปิ้งมอเตอร์ประเภทไฮบริด



รูปที่ 2.11 แสดงทิศทางการสร้างสนามแม่เหล็กของสเตปิ้งมอเตอร์ประเภทไฮบริด

### 2.1.3 รูปแบบการขับสเตปิ้งมอเตอร์

รูปแบบการขับหรือการกระตุ้นสเตปิ้งมอเตอร์ (Mode of excitation) คือ การกำจัดลำดับสัญญาณการกระตุ้นวงจรขับเคลื่อนเพื่อควบคุมทิศทางและปรับปรุงประสิทธิภาพของแรงบิดที่อยู่ด้วยกัน 3 แบบดังนี้

#### 2.1.3.1 การขับแบบกระตุ้นหนึ่งเฟส (ONE-PHASE EXCITATION)

เป็นการขับสเตปิ้งมอเตอร์แบบจ่ายกระแสให้กับขดลวดของสเตปิ้งมอเตอร์ครึ่งละหนึ่งเฟสเปลี่ยนไปตามลำดับ จะได้การเคลื่อนที่แบบเต็มสเตป จุดสมดุลของการหมุนจะอยู่ที่เฟสใดเฟสหนึ่ง วิธีนี้แรงบิดจะน้อย การจ่ายกระแสให้แต่ละเฟสแสดงดังตารางที่ 2.1 และสามารถสั่งให้มอเตอร์หมุนได้ทั้งทิศทางตามเข็มนาฬิกาและทิศวนเข็มนาฬิกา

ตารางที่ 2.1 แสดงการจ่ายกระแสแบบกระตุ้นหนึ่งเฟส

Step	เฟสที่ถูกกระตุ้น			
	$\Phi 4$	$\Phi 3$	$\Phi 2$	$\Phi 1$
1	0	0	0	1
2	0	0	1	0
3	0	1	0	0
4	1	0	0	0

### 2.1.3.2 การขับแบบกระตุ้นสองเฟส (TWO-PHASE EXCITATION)

การขับสเตปป์ึงมอเตอร์วิธีนี้ใช้วิธีการจ่ายกระแสให้กับสเตปป์ึงมอเตอร์แบบเต็มสเตป จุดสมดุลของการหมุนจะไม่อยู่ที่เฟสใดเฟสหนึ่ง แต่จะอยู่ตรงจุดกึ่งกลางระหว่างเฟสทั้งสอง แรงบิดของมอเตอร์จะมากกว่าแบบกระตุ้นทีละเฟส มีลักษณะการจ่ายกระแสให้แต่ละเฟสดังตารางที่ 2.2

ตารางที่ 2.2 แสดงการจ่ายกระแสแบบกระตุ้นสองเฟส

Step	เฟสที่ถูกกระตุ้น			
	$\Phi 4$	$\Phi 3$	$\Phi 2$	$\Phi 1$
1	0	0	1	1
2	0	1	1	0
3	1	1	0	0
4	1	0	0	1

### 2.1.3.3 การขับแบบกระตุ้นหนึ่งเฟสสลับสองเฟส (ONE-TWO-PHASE EXCITATION)

เป็นการขับแบบจ่ายกระแสให้เฟสของมอเตอร์แบบหนึ่งเฟสสลับกับการจ่ายแบบสองเฟส วิธีนี้จะเป็นการเคลื่อนที่แบบครึ่งสเตป โดยมีจุดสมดุลของแรงบิดอยู่ที่เฟสใดเฟสหนึ่ง และที่จุดกึ่งกลางระหว่างเฟสทั้งสองวิธีนี้จำนวนสเตปป์ึงจะเพิ่มขึ้นเป็นสองเท่าของสองแบบแรก แต่แรงบิดเฉลี่ยจะน้อยกว่าแบบที่ 2 ซึ่งมีลักษณะการจ่ายกระแสให้เฟสดังตารางที่ 2.3

ตารางที่ 2.3 แสดงการจ่ายกระแสแบบกระตุ้นหนึ่งเฟสสลับสองเฟส

Step	เฟสที่ถูกกระตุ้น			
	$\Phi 4$	$\Phi 3$	$\Phi 2$	$\Phi 1$
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	0	0
6	1	1	0	0
7	1	0	0	0
8	1	0	0	1

## 2.2 ไอซีเบอร์ UCN5804B

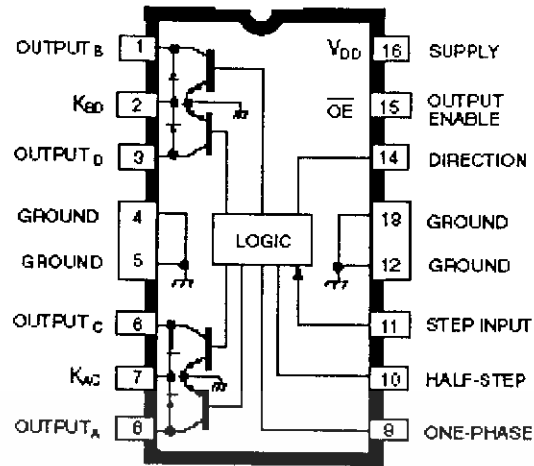
UCN5804B เป็นวงจรรวมที่มีซิมอสลอคจิกเป็นองค์ประกอบภายใน มีเอาต์พุตที่ให้กระแสและแรงดันสูง สามารถใช้ขับเคลื่อนและควบคุมสเตปป์มอเตอร์ให้สามารถหมุนได้แบบเป็นเฟสหรือแบบต่อเนื่อง ซิมอสลอคจิกนี้จะจัดเตรียมขาต่างๆที่ใช้งานไว้อย่างเป็นลำดับ เช่น ขา DIRECTION, OUTPUT ENABLE เป็นต้น และมีการทำงานที่สามารถเลือกทำงานได้ 3 สเต็ปด้วยกัน คือ หนึ่งเฟส (one phase), สองเฟส (two phase), และ ครึ่งสเต็ป (half step)

การขับเคลื่อนสเตปป์มอเตอร์แบบ หนึ่งเฟส จะเรียงลำดับการทำงานของสเตปป์มอเตอร์คือ เป็นเฟสแบบ A-B-C-D หรือแบบ D-C-B-A การทำงานในโหมดนี้จะใช้พลังงานน้อยที่สุดใน การหมุนต่อรอบ และมีความถูกต้องแม่นยำมาก

การขับเคลื่อนสเตปป์มอเตอร์แบบ สองเฟส จะมีการเรียงเฟสในการหมุนแบบ AB-BC-CD-DA การทำงานในโหมดนี้จะทำให้ได้ความเร็วและแรงบิดที่เพิ่มขึ้น

การขับเคลื่อนสเตปป์มอเตอร์แบบ ครึ่งสเต็ป จะเป็นการสลับการทำงานกัน ระหว่างการกระตุ้นแบบหนึ่งเฟสและ สองเฟส และมีการเรียงลำดับการทำงานแบบ A-AB-B-BC-C-CD-D-DA เป็นการเรียงลำดับการทำงานแบบ 8 สเต็ปนั่นเอง และมีคุณสมบัติที่สำคัญดังนี้

1. มีกระแสเอาต์พุตสูงสุด 1.5 A
2. ให้แรงดันเอาต์พุตได้ถึง 35 V



รูปที่ 2.12 โครงสร้างภายในของ UCN5804B

### 2.2.1 การใช้งาน UCN5804B

ภายในไอซี UCN5804B จะมีวงจร POWER-ON RESET (POR) อยู่ภายในเมื่อเราจ่ายแรงดันจากขั้วพลาวย้อนให้กับไอซี ไอซีในสถานะเริ่มต้นนั้นจะมีสถานะ ON หลังจากนั้นจะรีเซ็ตมาเป็นสถานะ OFF และการทำงานในโหมดต่างๆจะเป็นไปตามตารางที่ 2.4 2.5 และ 2.6

ตารางที่ 2.4 แสดงลำดับการขับแบบหนึ่งเฟส

(ขา 10) Half Step = L, (ขา 9) One Phase = H				
Step	A	B	C	D
POR	ON	OFF	OFF	OFF
1	ON	OFF	OFF	OFF
2	OFF	ON	OFF	OFF
3	OFF	OFF	ON	OFF
4	OFF	OFF	OFF	ON

สำหรับการขับมอเตอร์แบบหนึ่งเฟสนั้น จะต้องทำการจ่ายลอจิก “0” ให้กับขา 10 และจ่ายลอจิก “1” ให้กับขา 9 ของ UCN5804B ลักษณะการทำงานนั้นมอเตอร์จะหมุนแบบเรียงตามเฟสไปทีละหนึ่งเฟส คือ เฟส A-B-C-D ซึ่งจะทำงานเรียงตามเฟสเช่นนี้ไปเรื่อยๆ

ตารางที่ 2.5 แสดง ลำดับการขับแบบสองเฟส

(ข10) Half Step = L, (ข19) One Phase = L				
Step	A	B	C	D
POR	ON	OFF	OFF	OFF
1	ON	OFF	OFF	OFF
2	ON	ON	OFF	OFF
3	OFF	ON	ON	OFF
4	OFF	OFF	ON	ON

สำหรับการขับมอเตอร์แบบสองเฟสนั้น จะต้องทำการจ่ายลอจิก “0” ให้กับขา 10 และจ่ายลอจิก “0” ให้กับขา 9 ของ UCN5804B ลักษณะการทำงานนั้นมอเตอร์จะหมุนแบบเรียงตามเฟสไปที่ละสองเฟส คือ เฟส AB-BC-CD-DA ซึ่งจะทำงานเรียงตามเฟสเช่นนี้ไปเรื่อยๆ

ตารางที่ 2.6 แสดง ลำดับการขับแบบครึ่งเฟส

(ข10) Half Step = H, (ข19) One Phase = L				
Step	A	B	C	D
POR	ON	OFF	OFF	OFF
1	ON	OFF	OFF	OFF
2	ON	ON	OFF	OFF
3	OFF	ON	OFF	OFF
4	OFF	ON	ON	OFF
5	OFF	OFF	ON	OFF
6	OFF	OFF	ON	ON
7	OFF	OFF	OFF	ON
8	ON	OFF	OFF	ON

สำหรับการขับมอเตอร์แบบครึ่งเฟสนั้น จะต้องทำการจ่ายลอจิก “1” ให้กับขา 10 และจ่ายลอจิก “0” ให้กับขา 9 ของ UCN5804B ลักษณะการทำงานนั้นมอเตอร์จะหมุนแบบเรียงตามเฟสไปที่ละครึ่งเฟส คือ เฟส A-AB-B-BC-C-CD-D-DA ซึ่งจะทำงานเรียงตามเฟสเช่นนี้ไปเรื่อยๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3 ไมโครคอนโทรลเลอร์ MCS-51

### 2.3.1 คุณสมบัติที่สำคัญของ MCS-51

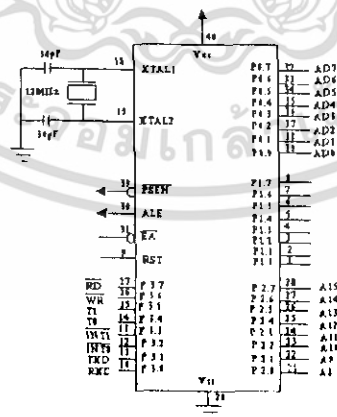
ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีด้วยกันหลายเบอร์ขึ้นกับโครงสร้างภายใน บางเบอร์จะมีหน่วยความจำภายในเป็นแบบ ROM บางเบอร์เป็นแบบ EPROM และปัจจุบันมีแบบ Fetch ROM

คุณสมบัติที่สำคัญของ ชิพไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีดังนี้

1. ต้องการแหล่งจ่ายไฟ 5 โวลต์เพียงชุดเดียว
2. มีหน่วยความจำเก็บโปรแกรมภายในชิพ หน่วยความจำเก็บข้อมูลภายในชิพจำนวน 128 ไบต์ (บาง CPU จะมี 256 ไบต์)
3. มีพอร์ทอินพุต เอาต์พุตจำนวน 4 พอร์ท พอร์ทละ 8 บิตหรือสามารถใช้งานเป็นพอร์ทขนาด 1 บิต แยกจากกันได้รวมทั้งสิ้น 32 บิต
4. สามารถอินเทอร์รัพท์ได้จาก 5 แหล่ง
5. มีพอร์ทอนุกรมที่สามารถรับส่งข้อมูลแบบฟูลดูเพล็กซ์ ด้วยอัตราเร็วในการรับส่งได้ตั้งแต่ 300 ถึง 375 กิโลบิตต่อวินาที
6. สามารถใช้หน่วยความจำโปรแกรมและข้อมูลที่อยู่ภายนอกชิปได้อย่างละ 64 กิโลไบต์
7. คำสั่งส่วนใหญ่ใช้เวลาทำงานเพียง 1 ไมโครวินาที เมื่อใช้คริสตอลความถี่ 12 เมกะเฮิร์ต
8. มีรีจิสเตอร์สำหรับใช้งานเป็น ไทม์เมอร์หรือเคาท์เตอร์เพื่อนับจำนวนสัญญาณนาฬิกาภายในชิพหรือนับการเปลี่ยนแปลงสถานะของสัญญาณภายนอกขนาด 16 บิต จำนวน 2 ตัว
9. หน่วยความจำภายในบางส่วนสามารถเข้าถึงข้อมูลได้ ทั้งระดับไบต์ และระดับบิต เพื่อให้การออกแบบโปรแกรมและการควบคุมระบบทำได้ง่าย

### 2.3.2 โครงสร้างของไมโครคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทุกเบอร์จะมีตำแหน่งขาพื้นฐานที่เหมือนกันดังรูปที่ 2.13



รูปที่ 2.13 แสดงรูปโครงสร้างของไมโครคอนโทรลเลอร์ตระกูล MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ความหมายของขาต่างๆมีดังนี้

พอร์ท 0 (Port 0) ได้แก่ขาที่ 32-39 ของMCS-51 สามารถใช้เป็นอินพุตเอาต์พุต สามารถใช้แบบบิตได้ นอกจากนี้ในการติดต่อกับหน่วยความจำภายนอก ยังใช้เป็นขาบัสดแอสและบัสข้อมูลอีกด้วย

พอร์ท 1 (Port 1) ได้แก่ขาที่ 1-8 เป็นพอร์ทอินพุตเอาต์พุต 8 บิต สามารถใช้แบบบิตได้ คือ P1.0-P1.7

พอร์ท 2 (Port 2) ได้แก่ ขาที่ 21-28 จะใช้งาน 2 หน้าทีคือ ใช้เป็นพอร์ทอินพุตเอาต์พุต สามารถใช้แบบบิตได้และใช้เป็นขาแอสแอส 8 บิต ในการอ้างหน่วยความจำภายนอก

พอร์ท 3 (Port 3) ได้แก่ ขาที่ 10-17 จะใช้งานสองหน้าที่คือ เป็นพอร์ทอินพุตและเอาต์พุต ใช้แบบบิตได้และใช้เป็นขาสัญญาณควบคุมต่างๆซึ่งมีหน้าที่ดังนี้

ขา P3.0 ใ้รับข้อมูลจากภายนอกแบบอนุกรม

ขา P3.1 ใช้ส่งข้อมูลไปภายนอกแบบอนุกรม

ขา P3.2 ใช้เป็นอินพุตเพื่อรับสัญญาณอินเตอร์รัปต์ชนิดที่ 0

ขา P3.3 ใช้เป็นอินพุตเพื่อรับสัญญาณอินเตอร์รัปต์ชนิดที่ 1

ขา P3.4 สัญญาณอินพุตให้เคาท์เตอร์ของไทมเมอร์ 0

ขา P3.5 สัญญาณอินพุตให้เคาท์เตอร์ของไทมเมอร์ 1

ขา P3.6 ใช้เป็นสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับเก็บข้อมูลภายนอกชิป

ขา P3.7 ใช้เป็นสัญญาณควบคุมการอ่านข้อมูลไปยังหน่วยความจำสำหรับเก็บข้อมูลภายนอกชิป

PSEN (Program Store Enable) ขา PSEN เป็นขาที่ส่งสัญญาณออกคือขา 29 ขานี้จะแอกทีฟเมื่อ MCS-51 ต้องการอ่านโปรแกรมจากหน่วยความจำภายนอก โดยปกติถ้าหน่วยความจำภายนอกเป็นอีพรอม ขา PSEN จะต่อกับขาสัญญาณเปิดทางด้านเอาต์พุต (Output Enable : OE) ของอีพรอม

ALE/PROG (Address Latch Enable) เนื่องจากพอร์ท 0 สามารถใช้เป็นขาอ้างตำแหน่งและขาข้อมูล MCS-51 จะมีขา ALE ได้แก่ขา 30 ขานี้ จะใช้มัลติเพล็กซ์ (Multiplex) สัญญาณบิตตำแหน่งของพอร์ท 0 ในการใช้งานระบบ MCS-51 นั้น จะต้องมีอุปกรณ์มาต่อกับพอร์ท 0 ที่ทำหน้าที่คงค่า (Latch) สัญญาณบิตตำแหน่ง เมื่อ MCS-51 ต้องการติดต่อกับหน่วยความจำภายนอก MCS-51 จะส่งสัญญาณบิตตำแหน่งออกมาก่อนทางพอร์ท 0 จากนั้นจะส่งสัญญาณ ALE มาทำหน้าที่ คงค่า (Latch) อุปกรณ์ภายนอก ให้เก็บค่าบิตตำแหน่งของ พอร์ท 0 ไว้เพื่อใช้เป็นบัสข้อมูล

EA/Vpp (External Access) ขา EA ได้แก่ขาที่ 31 ถ้าขานี้เป็นลอจิก “1” จะใช้กับเบอร์ 8051/8052 เพื่อบอกว่าให้อ่านโปรแกรมจากหน่วยความจำภายใน แต่ถ้าเป็นลอจิก “0” จะให้

MCS-51 ทำโปรแกรมโดย อ่านจากหน่วยความจำโปรแกรมภายนอก (ถ้าขา EA เป็น “0” ขา PSEN จะแอกทีฟ) ถ้าเป็นเบอร์ 8031 หรือ 8032 ขา EA จะเป็น “0” เสมอ เพราะไม่มีโปรแกรมหน่วยความจำภายใน แต่ถ้าใช้เบอร์ 8051/8052 ซึ่งมีหน่วยความจำภายในและให้ขา EA เป็น “0” ซึ่งจะหยุดการทำงานของรอมภายในและอ่านโปรแกรมจากอีพรอมภายนอกแทน

**RST (Reset)** ขา RST ได้แก่ขา 9 จะใช้ในการรีเซ็ต MCS-51 โดยให้ขานี้เป็นลอจิก “1” อย่างน้อย 2 คาบเวลาจึงจะรีเซ็ตระบบได้

**XTAL1** ขาที่ 19 ใช้ต่อคริสตัลจากภายนอก โดยเป็นอินพุตเข้าสู่วงจรรอสซิลเลเตอร์

**XTAL2** ขาที่ 18 ใช้ต่อคริสตัลจากภายนอก โดยเป็นเอาต์พุตออกจากวงจรรอสซิลเลเตอร์

### 2.3.3 โครงสร้างหน่วยความจำ

หน่วยความจำสำหรับ MCS-51 จะมี 2 ชนิดคือ หน่วยความจำที่ใช้เก็บโปรแกรมรอมกับหน่วยความจำที่ใช้เก็บข้อมูลในการประมวลผล แรม MCS-51 บางเบอร์เช่น 8051,8052 จะมีหน่วยความจำภายในชิป และ MCS-51 ทุกเบอร์ยังสามารถอ้างหน่วยความจำโปรแกรมภายนอกได้มากที่สุด 64 กิโลไบต์ และอ้างหน่วยความจำภายนอกได้มากที่สุด 64 กิโลไบต์ สำหรับหน่วยความจำแรม ภายในจะประกอบไปด้วยพื้นที่ใช้งานทั่วไป, ชุดรีจิสเตอร์, พื้นที่ใช้งานระดับบิต และรีจิสเตอร์ฟังก์ชันพิเศษ

#### Bit-addressable RAM

ใน MCS-51 จะมีหน่วยความจำที่สามารถอ้างข้อมูลในระดับบิตได้ตั้งแต่ตำแหน่ง 20H ถึง 2FH รวม 16 ไบต์ โดยสามารถเซต, เคลียร์, แอนด์, ออร์ทางลอจิกได้ จำนวนบิตที่ใช้งานได้ทั้งหมดมีจำนวน 128 บิต (8บิตx16ไบต์)

#### ชุดรีจิสเตอร์ (Register Banks)

หน่วยความจำข้อมูลภายในที่ใช้เป็นชุดรีจิสเตอร์มีทั้งหมด 32 ตำแหน่ง โดยจะมี 4 ชุด แต่ละชุดมีรีจิสเตอร์ 8 ตัว คือ R0 ถึง R7 โดยชุดแรกจะอยู่ในตำแหน่ง 00H-07H

#### รีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Register)

ใน MCS-51 รีจิสเตอร์จะใช้หน่วยความจำแรมภายในชิป โดยส่วนหนึ่งเป็นรีจิสเตอร์พิเศษ (Special Function Register : SFR) ซึ่งมีทั้งหมด 21 ตัว โดยรีจิสเตอร์พิเศษต่างๆจะเริ่มที่หน่วยความจำตั้งแต่ 80H ถึง FFH ซึ่งมีทั้งหมด 128 ตำแหน่ง แต่จะเป็นรีจิสเตอร์ฟังก์ชันพิเศษเพียง 21 ตำแหน่ง แต่ถ้าเป็น 8032/ 8051 จะใช้ 26 ตำแหน่งหรือมี SFR 26 ตัว

#### Program Status Word

รีจิสเตอร์ตัวนี้เรียกว่า PSW จะอยู่ที่ตำแหน่ง D0H ซึ่งสามารถเข้าถึงข้อมูลระดับบิตได้ โดยรีจิสเตอร์นี้จะเป็นตัวบอกสถานะต่างๆของไมโครคอนโทรลเลอร์ ความหมายของแต่ละบิตอธิบายได้ดังนี้

1. แฟล็กตัวทศ (Carry Flag : CF) บิตนี้เป็นบิตที่ 7 ของ PSW บิตนี้จะมีค่าสำคัญหากมีการกระทำทางคณิตศาสตร์ โดยบิตนี้จะเซตเมื่อเกิดการทศของบิตที่ 7 ขณะทำการบวกเลขหรือเซตเมื่อเกิดการยืมของบิตที่ 7 เมื่อเกิดการลบเลข

2. แฟล็กตัวช่วยทศ (Auxiliary Carry flag) เมื่อเกิดการบวกแบบ Binary-Code-Decimal (BCD) บิต แฟล็กตัวช่วยทศ (AC) หรือบิตตัวช่วยทศจะถูกเซต เมื่อมีการทศจากบิตที่ 3 ไปบิตที่ 4 หรือถ้าใน Lower Nibble มีค่าระหว่าง 0AH-0FH เนื่องจากรหัส BCD นี้มีค่าได้มากที่สุดแค่ 9 ถ้าหากมีการบวกเลขแบบ BCD จะต้องตามด้วยคำสั่ง DAA (Decimal Adjust Accumulator) เพื่อปรับค่าที่มีค่าเกิน 9 โดยบวกเลข 6 เข้าไป จะทำให้เป็นรหัส BCD ที่แทนเลขฐานสิบได้

3. แฟล็กศูนย์ (Flag 0) เป็นแฟล็กที่ผู้ใช้สามารถใช้งานทั่วไปได้

4. บิตเลือกชุดรีจิสเตอร์ (Register Bank Select Bits) ตามที่ทราบมาแล้วว่าใน MCS-51 จะมีชุดรีจิสเตอร์อยู่ 4 ชุด ถ้าจะเลือกให้ชุดใดแอกทีฟจะกำหนดได้ในบิต RS1 และ RS2 ของ PSW และจะเคลียร์ตัวเองเมื่อระบบถูกรีเซต ถ้าหากต้องการติดต่อกับชุดรีจิสเตอร์ 3 โดยย้าย ข้อมูลจาก R7 ( ตำแหน่ง 1FH) มาเก็บในแอกคิวมิวเลเตอร์

5. แฟล็กค่าเกิน (Overflow Flag) แฟล็ก OV จะถูกเซต หลังจากการกระทำทางคณิตศาสตร์แล้วเกิดค่าเกิน คือจำนวนที่เกิดจากการบวกหรือการลบ มีค่าเกินกว่าที่จำนวนไบต์จะเป็นไปได้คือ มากกว่า +128 หรือน้อยกว่า -125 ตัวอย่างเช่น ถ้าเกิดการบวกเลขสองจำนวนนี้จะเกิดการเซตบิต OV ขึ้นใน PSW

6. บิตพาริตี (Parity Bit) พาริตีบิต (P) เป็นบิตที่บอกค่าพาริตีของรีจิสเตอร์ แอกคิวมิวเลเตอร์ซึ่งอาจเป็นตัวตรวจสอบความถูกต้องของข้อมูลได้ โดยจะเซตหรือเคลียร์ขึ้นกับแอกคิวมิวเลเตอร์ เช่น ถ้าแอกคิวมิวเลเตอร์มีค่าเป็น 10101101 บิต P จะเป็น "1"

### รีจิสเตอร์ B (B register)

รีจิสเตอร์ B จะอยู่ตำแหน่ง F0H ของหน่วยความจำข้อมูลภายใน เป็นรีจิสเตอร์ที่สามารถใช้งานทั่วไปได้ โดยทั่วไปรีจิสเตอร์นี้จะใช้คูณ หรือหารกับรีจิสเตอร์แอกคิวมิวเลเตอร์ เช่น การทำคำสั่ง MUL AB ซึ่งเป็นการคูณแบบ 8 บิต โดยผลลัพธ์ที่ได้จะมีขนาด 16 บิต ซึ่งรีจิสเตอร์ A จะเก็บค่า 8 บิตต่ำ และรีจิสเตอร์ B จะเก็บค่า 8 บิตสูง สำหรับการหารโดยการทำคำสั่ง DIV AB โดยค่าใน A จะถูกหารด้วย B ผลลัพธ์ที่ได้จะเก็บในรีจิสเตอร์ AB โดย B จะเก็บค่า 8 บิตต่ำและ A จะเก็บค่า 8 บิตสูง รีจิสเตอร์ B นี้สามารถเข้าถึงข้อมูลระดับบิตได้ โดยตำแหน่งของบิตคือตำแหน่ง F0H ถึง F7H

### ตัวชี้สแตค (Stack Pointer)

ตัวชี้สแตค (SP) เป็นรีจิสเตอร์ขนาด 8 บิต อยู่ตำแหน่ง 81H การเขียนค่าเข้าไปในตำแหน่งที่ตัวชี้สแตคชี้อยู่นี้ เรียกว่า "Pushing" สำหรับการอ่านค่าที่ SP ชี้อยู่ เรียกว่า "Popping" ค่าของตัวชี้สแตคจะเพิ่มขึ้นหนึ่งก่อนที่จะเขียนข้อมูลลงไป และจะลดลงหนึ่งเมื่ออ่านข้อมูลออกมาแล้ว หาก

โปรแกรมทำคำสั่ง Call จะใช้รีจิสเตอร์สแตคนี้เก็บค่าตำแหน่งเดิมของโปรแกรม (PC) ก่อนที่จะทำโปรแกรมย่อย เมื่อทำโปรแกรมย่อยเสร็จแล้วจะคืนค่าในสแตคให้กับในโปรแกรมตามเดิม โดยปกติค่าโปรแกรมจะกำหนดให้อยู่ในแรมภายใน

#### รีจิสเตอร์ Data Pointer (DPTR)

รีจิสเตอร์ นี้ใช้สำหรับชี้ตำแหน่งรหัสโปรแกรม หรือ ข้อมูลในหน่วยความจำโดยเป็นรีจิสเตอร์ขนาด 16 บิต ซึ่งประกอบด้วยรีจิสเตอร์ 2 ตัว คือ DPL ตำแหน่งที่ 82H โดยจะเก็บเป็น 8 บิตต่ำ และ DPH ตำแหน่งที่ 83H โดยจะเก็บค่า 8 บิตสูง รีจิสเตอร์ทั้งสองตัวนี้จะรวมกันกลายเป็นรีจิสเตอร์ 16 บิต ถ้าหากต้องการเก็บค่า 55H ไปยังตำแหน่งของหน่วยความจำข้อมูลภายนอกตำแหน่งที่ 1000H

#### รีจิสเตอร์พอร์ท (Port Registers)

ใน MCS-51 ค่าของพอร์ทหมายถึงค่าของหน่วยความจำด้วย หากต้องการส่งข้อมูลจากพอร์ท ก็อ่านเพียงค่าจากตำแหน่งที่หน่วยความจำที่พอร์ทนั้นอยู่ใน MCS-51 พอร์ท 0 จะอยู่ที่ตำแหน่ง 80H ,พอร์ท 1 จะอยู่ที่ตำแหน่ง 90H,พอร์ท 2 จะอยู่ที่ตำแหน่ง A0Hและพอร์ท 3 จะอยู่ที่ตำแหน่ง B0H พอร์ท 0, 2, และ 3 โดยทั่วไปจะไม่ใช้ถ้าหากมีการติดต่อกับหน่วยความจำภายนอกหรือใช้เป็นพอร์ทพิเศษ (เช่น อินเทอร์รัพท์, พอร์ทสื่อสารอนุกรม) โดยปกติแล้วจะใช้พอร์ท 1 ในการติดต่อกับอุปกรณ์ภายนอกพอร์ททุกพอร์ทสามารถอ้างถึงข้อมูลในระดับบิตได้

#### รีจิสเตอร์เวลา (Timer register)

ใน MCS-51 เบอร์ 8051 จะมีรีจิสเตอร์ที่ใช้นับและจับเวลาขนาด 16 บิต 2 ตัวคือ Timer 0 อยู่ที่ตำแหน่ง 8AH และ 8CH โดยตำแหน่ง 8AH หมายถึง CL0 ซึ่งจะเป็น 8 ไบต์ต่ำ และ 8CH หมายถึง 8 ไบต์สูง TH0 รีจิสเตอร์อีกตัวคือ ไทม์เมอร์ 1 โดยแบ่งเป็น TL1 อยู่ที่ตำแหน่ง 8BH เป็นไบต์ต่ำและ TH1 อยู่ที่ตำแหน่ง 8DH เป็นไบต์สูง การใช้ไทม์เมอร์จะต้องกำหนดการทำงานในรีจิสเตอร์ TMOD (Timer/Counter Mode Control Register) ซึ่งอยู่ที่ตำแหน่ง 88H เสียก่อน

#### รีจิสเตอร์พอร์ทอนุกรม (Serial Port Register)

MCS-51 จะมีพอร์ทสื่อสารอนุกรม (Serial Port) อยู่ภายในชิพ ซึ่งสามารถจะรับหรือส่งข้อมูลแบบอนุกรมให้เขียนไปที่รีจิสเตอร์ SBUF (Serial Data Buffer) ซึ่งอยู่ที่ตำแหน่ง 99H โดยถ้าต้องการส่งข้อมูลแบบอนุกรมให้เขียนข้อมูลไปที่รีจิสเตอร์นี้ ตัวพอร์ทอนุกรมสามารถโปรแกรมให้ทำงานได้ 4 โหมด โดยโปรแกรมผ่านรีจิสเตอร์ SCON (Serial Port Control Register) ตำแหน่ง 98H

#### รีจิสเตอร์อินเทอร์รัพท์ (Interrupt Port Register)

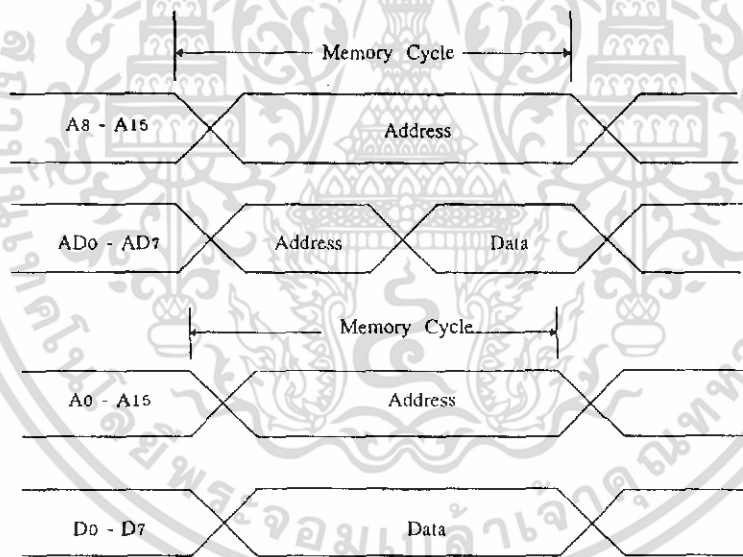
MCS-51 สามารถอินเทอร์รัพท์ได้ 5 ตำแหน่ง โดยมี 2-priorite ตัวอินเทอร์รัพท์จะไม่ทำงานหลังจากระบบถูกรีเซต และทำงานหลังจากที่เขียนข้อมูลไปที่รีจิสเตอร์ IE หรือตำแหน่ง A8H ลำดับความสำคัญสามารถเซตได้ที่รีจิสเตอร์ IP หรือตำแหน่ง B8H

**เพาเวอร์คอนโทรลริจิสเตอร์ (PCON)**

ริจิสเตอร์ PCON อยู่ที่ตำแหน่ง 87H ใช้หยุดการทำงานของ MCS-51 โดยจะหยุดจ่ายสัญญาณนาฬิกาให้ระบบ ทำให้ข้อมูลต่างๆภายใน MCS-51 ไม่มีการเปลี่ยนแปลง นอกจากนี้ยังลดพลังงานไฟฟ้าที่จ่ายให้ MCS-51 ด้วย

**2.3.4 หน่วยความจำภายนอก (External Memory)**

MCS-51 สามารถอ้างหน่วยความจำข้อมูลภายนอกได้ 64 กิโลไบต์ และอ้างหน่วยความจำโปรแกรมภายนอกได้ 64 กิโลไบต์ MCS-51 จะใช้พอร์ท 0 ในการอ้างตำแหน่งหน่วยความจำ 8 บิตล่าง และใช้พอร์ท 0 เป็นพอร์ทข้อมูล (Data) ด้วยโดยใช้ขา ALE มาคงค่าข้อมูลพอร์ท 0 และใช้พอร์ท 2 เป็นขาอ้างตำแหน่ง 8 บิตบน (รวมขาอ้างตำแหน่ง 16 เส้น ซึ่งอ้างได้ 64 กิโลไบต์) นอกจากพอร์ท 0 จะใช้งาน 2 หน้าที่ ในการติดต่อกับหน่วยความจำจะใช้ วิธีมัลติเพล็กซ์ระหว่างตำแหน่งกับข้อมูล พิจารณาจากรูป ถ้าต้องการติดต่อกับหน่วยความจำที่เก็บข้อมูล 8 บิต และเก็บได้ 64 กิโลไบต์ จะต้องใช้สายสัญญาณ 24 เส้น คือเป็นขาตำแหน่ง 16 เส้น และขาข้อมูล 8 เส้น ดังรูป แต่ถ้าใช้วิธีมัลติเพล็กซ์คือ ใช้ขา A0-A7 เป็นขาข้อมูลด้วยคือ D0-D7 จะใช้สายสัญญาณเพียง 16 เส้นเท่านั้น จากรูปที่ 2.14 จะเห็นว่าเมื่อต้องการติดต่อกับหน่วยความจำจะส่งสัญญาณตำแหน่ง A0-A15 ออกมาก่อน 16 เส้น และเวลาต่อมาขา A0-A7 จะถูกเปลี่ยนเป็น D0-D7



รูปที่ 2.14 ไคอะแกรมกลุ่มสัญญาณที่ใช้อ่านข้อมูล

**2.3.5 การติดต่อกับหน่วยความจำโปรแกรมภายนอก**

ในการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก MCS-51 จะส่งค่าตำแหน่งของหน่วยความจำออกไปก่อน ซึ่งค่าตำแหน่งจะเก็บอยู่ใน PC โดยส่งออกไปทางพอร์ท 0 และพอร์ท 2 จากนั้นเวลาต่อมาจะส่งขา ALE ให้เป็นลอจิก “0” เพื่อคงค่าตำแหน่งของ 8 บิตต่ำ คือพอร์ท 0

จากนั้นจะส่งสัญญาณทางขา PSEN ให้เป็นลอจิก “0” เพื่ออ่านข้อมูลซึ่งจะได้อปโสด เข้าไปทางขาบัสข้อมูลคือพอร์ท 0 โค้ดแอมการอ่านข้อมูลจากหน่วยความจำภายนอก

#### การติดต่อกับหน่วยความจำข้อมูลภายนอก

หน่วยความจำข้อมูลภายนอก MCS-51 สามารถอ่านและเขียนได้ ในการติดต่อกับหน่วยความจำข้อมูลภายนอก MCS-51 จะส่งขากำหนดตำแหน่งออกไปทางพอร์ท 0 และพอร์ท 2 จากนั้นจะส่งขา ALE เพื่อไปคงค่าตำแหน่ง 8 บิตค่า โดยการอ่านเขียนข้อมูลนั้นจะใช้ขา RD (P3.7) และขา WR (P3.6) ตามลำดับ

#### กระบวนการรีเซ็ต (Reset Operation)

การรีเซ็ตหรือเริ่มต้นทำงานใหม่ของ MCS-51 จะต้องให้ลอจิก “1” ที่ขา RST เป็นเวลา 2 คาบเวลา (หนึ่งคาบเวลาเท่ากับ 12 สัญญาณนาฬิกา) จากนั้นให้กับเป็นลอจิก “0”

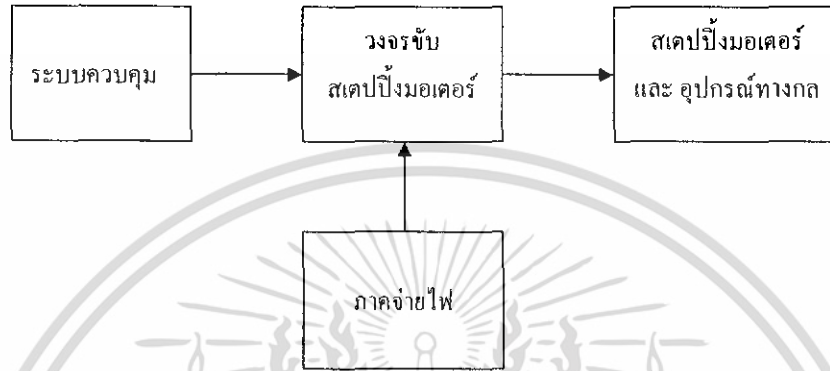


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### หลักการการทำงานของเครื่องตัดและปกสายไฟ

เครื่องตัดและปกสายไฟแบ่งออกเป็นส่วนใหญ่ๆ ได้ 4 ส่วนด้วยกันคือระบบควบคุม วงจรขับ สเตปปีงมอเตอร์ สเตปปีงมอเตอร์และอุปกรณ์ทางกล และภาคจ่ายไฟ ซึ่งในส่วนของ ฮาร์ดแวร์นั้นมีลักษณะโครงสร้างที่เป็นไปตามบล็อกไดอะแกรมในรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมของเครื่องตัดและปกสายไฟ

จากรูปที่ 3.1 สามารถอธิบายหลักการการทำงานของแต่ละส่วนได้ดังนี้

#### 3.1 ระบบควบคุม

ระบบควบคุมนั้นจะเป็นส่วนที่ป้อนคำสั่งในการทำงานให้กับวงจรขับสเตปปีงมอเตอร์ ภาย ในระบบนี้ก็จะมีส่วนประกอบที่สำคัญหลักๆ คือ ไมโครคอนโทรลเลอร์ MCS-51, สวิตซ์แบบ เมตริกซ์ (matrix switch) หรือคีย์แพด (keypad) ขนาด 4x4 จุด จอแสดงผลแบบ โมดูล LCD ขนาด 16x2 และวงจรออสซิลเลทอีก 2 วงจร โดยอุปกรณ์เหล่านี้มีลักษณะการเชื่อมต่อกันดังรูปที่ 3.2

##### 3.1.1 ฟังก์ชันการทำงานของไมโครคอนโทรลเลอร์ MCS-51

จากรูปที่ 3.2 จะเห็นว่าไมโครคอนโทรลเลอร์ MCS-51 ถือเป็นหัวใจหลักของระบบ ควบคุม มีหน้าที่ในการรับค่าจากการกด Keypad ปุ่มต่างๆ แล้วนำไปแสดงผลที่จอ โมดูล LCD หรือ ประมวลผลเพื่อใช้สำหรับขับสเตปปีงมอเตอร์ต่อไป



### 3.1.2 ฟังก์ชันการทำงานปุ่มกดต่างๆ ของ Keypad

ดังที่กล่าวไว้แล้วในตอนต้นว่า Keypad ที่ใช้มีขนาด 4x4 จุด ซึ่งแต่ละปุ่มกดนั้นมีฟังก์ชันการใช้งานแตกต่างกันไป ดังแสดงในรูปที่ 3.3 โดยรายละเอียดของแต่ละปุ่มกดนั้นสามารถอธิบายได้ดังนี้

7	8	9	START
4	5	6	Y
1	2	3	X
0	CLR	Z	↵

รูปที่ 3.3 ตำแหน่งปุ่มกดต่างๆ ของ Keypad

- ปุ่มกด '0-9' เป็นปุ่มกดที่ใช้แสดงค่าต่างๆ ของเลขนั้นๆ
- ปุ่มกด 'CLR' เป็นปุ่มกดที่ใช้ยกเลิกการทำงานทั้งหมด
- ปุ่มกด '↵' หรือ Enter เป็นปุ่มกดที่ใช้ยืนยันค่าต่างๆ ที่กำหนดลงไป
- ปุ่มกด 'X' เป็นปุ่มกดที่ใช้สำหรับกำหนดหรือเปลี่ยนค่าระยะปอกของสายไฟ ซึ่งกำหนดให้เป็นค่าของตัวแปร X ซึ่งจะแสดงเป็นค่าตัวเลข 0.00 cm ที่จอ LCD กำหนดความยาวสูงสุดไม่เกิน 1.00 cm
- ปุ่มกด 'Y' เป็นปุ่มกดที่ใช้สำหรับกำหนดหรือเปลี่ยนค่าระยะตัดของสายไฟ ซึ่งกำหนดให้เป็นค่าของตัวแปร Y ซึ่งจะแสดงเป็นค่าตัวเลข 00.0 cm สามารถกำหนดความยาวได้สูงสุดที่ 99.9 cm
- ปุ่มกด 'Z' เป็นปุ่มกดที่ใช้สำหรับกำหนดจำนวนของสายไฟที่ต้องการจะตัดทั้งหมด ซึ่งจะแสดงเป็นค่าตัวเลข Number=0000 สามารถกำหนดจำนวนได้สูงสุดที่ 9,999 เส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.3 การกำหนดค่าต่าง ๆ

เมื่อเราทำการเปิดเครื่องมาที่หน้าจอ LCD จะแสดงค่าดังนี้

Number= 0000  
00.0 cm 0.00 cm

รูปที่ 3.4 แสดงหน้าจอ LCD ขณะเปิดเครื่อง

จากนั้นจะทำการป้อนค่า Y และ X ตามลำดับที่หน้าจอ LCD จะแสดงค่าดังนี้

Y = 00.0 cm  
X = 0.00 cm

รูปที่ 3.5 หน้าจอ LCD เมื่อกดปุ่ม 'X หรือ Y' ที่ Keypad

เมื่อเราทำการป้อนค่า Y และ X เรียบร้อยแล้วจากนั้นให้เราทำการกดที่ปุ่ม Enter จากนั้นให้เราทำการกดที่ปุ่ม Z ที่หน้าจอ LCD จะแสดงค่าดังนี้

Number = 0000

รูปที่ 3.6 หน้าจอ LCD เมื่อกดปุ่ม 'Z' ที่ Keypad

ให้เราทำการป้อนค่าแล้วกดที่ปุ่ม Enter จากนั้นกดปุ่ม 'START' เป็นปุ่มกดที่ใช้สำหรับการเริ่มต้นทำงานของเครื่องตัดและปอกสายไฟ โดยจอ LCD จะปรากฏดังรูปที่ 3.7 ซึ่งจะมีรูปแบบคล้ายกับตอนที่เราเปิดเครื่องขึ้นมาใหม่

Number= 0005  
10.0cm 0.50cm

รูปที่ 3.7 หน้าจอ LCD เมื่อกดปุ่ม 'START' ที่ Keypad

จากรูปที่ 3.7 เป็นตัวอย่างในการป้อนค่าการใช้งานจริงของเครื่อง ค่าที่แสดง

Number = 0005 คือจำนวนสายที่ต้องการจะปอกทั้งหมดคือ 5 เส้น

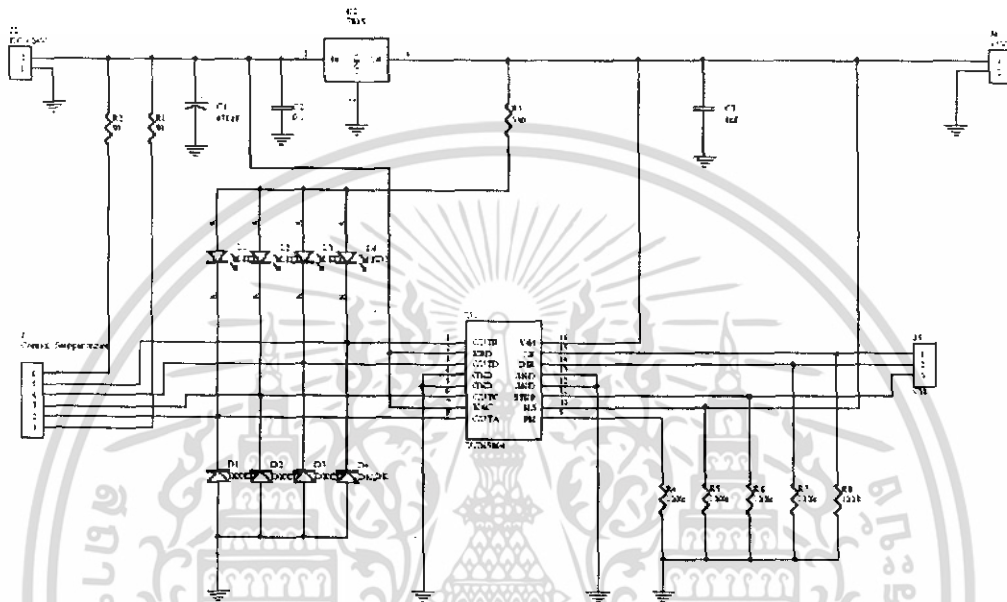
10.0 cm คือระยะความยาวของสายที่ต้องการจะตัดหรือค่า Y

0.50 cm คือขนาดช่วงของปลายสายที่ต้องการจะปอกหรือค่า X

ในกรณีที่ต้องการแก้ไขค่าของ X, Y หรือ Z จะต้องกดปุ่ม 'CLR' เพื่อให้เครื่องหยุดทำงาน ก่อนจากนั้นต้องทำการป้อนค่าใหม่ทั้งหมด เพราะเวลากดปุ่ม 'CLR' ค่าของ X, Y และ Z จะถูก Reset ให้เป็นศูนย์ทั้งหมด

### 3.2 วงจรขับสเตปปีงมอเตอร์

เนื่องจากเครื่องปอกสายไฟเครื่องนี้ใช้สเตปปีงมอเตอร์จำนวน 3 ตัว ดังนั้นจำนวนของ วงจรขับที่ใช้ก็ต้องใช้ถึง 3 วงจรซึ่งมีลักษณะการต่อวงจรเป็นไปดังรูปที่ 3.8

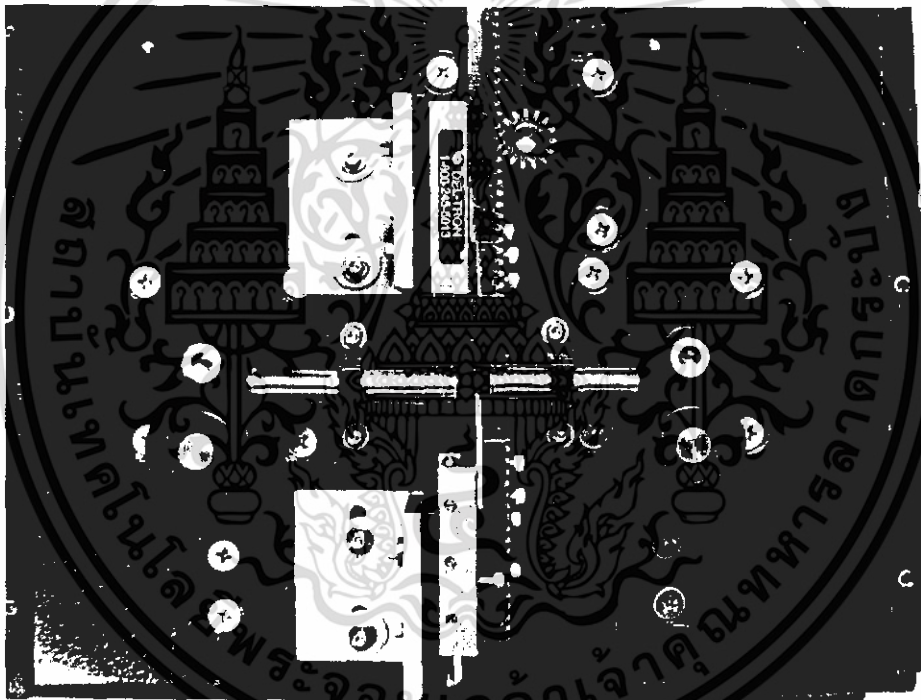


รูปที่ 3.8 วงจรขับสเตปปีงมอเตอร์

จากวงจรในรูปที่ 3.8 จะเห็นว่าควบคุม การทำงานโดยใช้ไอซีเบอร์ UCN5804B ซึ่งมี โหมดการทำงานในการขับสเตปปีงมอเตอร์ถึง 3 แบบ คือ ขับแบบหนึ่งสเตป แบบสองสเตปและ แบบครึ่งสเตป ซึ่งในการทำงานของเครื่องตัดและปอกสายไฟจะใช้การขับแบบครึ่งสเตปเพื่อให้ได้ ความละเอียดและแม่นยำในการทำงาน จะควบคุมจังหวะการหมุนของสเตปปีงมอเตอร์โดยใช้ขา 14 ซึ่งเป็นขา Direction และขา 15 ซึ่งเป็นขา Enable ซึ่งขาทั้งสองมีฟังก์ชันการทำงานที่พิเศษคือ ที่ ขา 14 เมื่อป้อนลอจิก 0 จะทำให้สเตปปีงมอเตอร์หมุนตามเข็มนาฬิกา เมื่อป้อนลอจิก 1 จะทำให้สเตปปีงมอเตอร์หมุนทวนเข็มนาฬิกา สำหรับขา 15 เมื่อป้อนลอจิก 0 จะทำให้สเตปปีงมอเตอร์หมุน เมื่อป้อนลอจิก 1 จะทำให้สเตปปีงมอเตอร์หยุดหมุน

### 3.3 สเตปป์มอเตอร์และอุปกรณ์ทางกล

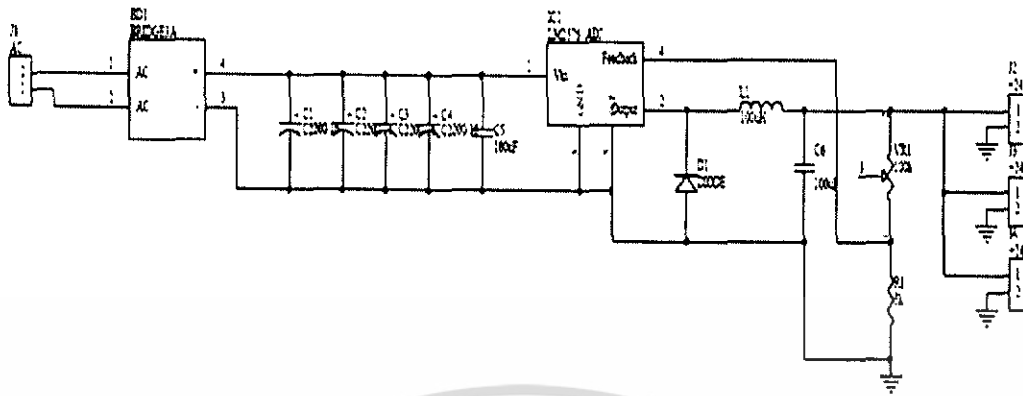
ในส่วนของสเตปป์มอเตอร์และอุปกรณ์ทางกลนั้นจะประกอบไปด้วยส่วนที่ทำหน้าที่เคลื่อนสายไฟซึ่งจะใช้สเตปป์มอเตอร์ 2 ตัวในการทำงาน และสเตปป์มอเตอร์ที่ทำหน้าที่ตัดและปอกสายไฟอีกหนึ่งตัว โดยในส่วนของสเตปป์มอเตอร์ที่ทำหน้าที่เคลื่อนสายไฟนั้นจะอาศัยหลักการง่ายๆ คือเวลาที่สเตปป์มอเตอร์หมุนทวนเข็มนาฬิกาก็จะทำการเคลื่อนสายไฟไปข้างหน้า เมื่อหมุนตามเข็มนาฬิกาสายไฟก็จะถูกเลื่อนกลับมา ในส่วนของชุดปอกสายไฟนั้นก็ประกอบไปด้วยใบมีด 2 เล่ม โดยเล่มแรกจะถูกยึดติดอยู่กับที่และอีกเล่มจะต่อเข้ากับแกนของสเตปป์มอเตอร์ โดยเมื่อเวลาที่สเตปป์มอเตอร์หมุนทวนเข็มนาฬิกาใบมีดจะเลื่อนลงมาปอก (หรือตัดขึ้นอยู่กับลำดับขั้นตอน) และเมื่อสเตปป์มอเตอร์หมุนตามเข็มนาฬิกา ใบมีดก็จะเลื่อนขึ้นกลับมาอยู่ในตำแหน่งเดิม ซึ่งรูปแบบของอุปกรณ์ทางกลดังกล่าวแสดงได้ดังรูปที่ 3.9 สำหรับสเตปป์มอเตอร์ที่ใช้จะเป็นแบบ 1.8 องศาต่อสเตปป์



รูปที่ 3.9 แสดงสเตปป์มอเตอร์และอุปกรณ์ทางกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 ภาคจ่ายไฟ



รูปที่ 3.10 ภาค Power Supply

การทำงานของภาคจ่ายไฟนี้ ในส่วนเริ่มต้นจะใช้หม้อแปลงขนาด 18 โวลต์ 5 แอมป์ทำหน้าที่จ่ายแรงดันไฟเอซีเข้ามาเข้าที่บริดจ์ไดโอดที่ทำหน้าที่แปลงแรงดันไฟเอซีเป็นไฟดีซี หลังจากนั้นแรงดันไฟเอซีผ่านบริดจ์ไดโอดออกมาจะได้ไฟดีซีขนาด 24 โวลต์ออกมา จากนั้นแรงดันไฟดีซีจะเข้ามาเข้าที่ขาอินพุทของไอซีเบอร์ LM2576-ADJ ซึ่งไอซีนี้จะสามารถปรับแรงดันไฟได้ตั้งแต่ 8-40 โวลต์และให้กระแสอยู่ที่ 3 แอมป์ ในส่วนของเครื่องตัดและปกสายไฟนี้จะปรับใช้แรงดันไฟที่ 24 โวลต์ และจะนำแรงดันไฟ 24 โวลต์นี้มาจ่ายให้กับวงจรขับเคลื่อนสเตปป์มอเตอร์ทั้ง 3 วงจรต่อไป

### 3.5 ขั้นตอนการคำนวณและการออกแบบโปรแกรม

ในการควบคุมระยะความยาวของสายไฟที่เลื่อนไปมานั้นจะอาศัยหลักการนับจำนวนสเตปป์ของสเตปป์มอเตอร์ ซึ่งมีหลักในการคำนวณจำนวนของสเตปป์ดังนี้

$$\text{ระยะตัด } Y = Y/0.125 \text{ mm}$$

$$\text{ระยะปอก } X = X/0.125 \text{ mm}$$

โดยค่าของ 0.125 mm ที่นำมาคำนวณนี้เป็นค่าความยาวของระยะครึ่งสเตปป์ ซึ่งมีวิธีการคำนวณดังนี้

$$\text{รัศมีของลูกยาง (r)} = 8 \text{ mm}$$

$$\text{เส้นรอบวงของวงกลมคือ } 2\pi r = 2 * \pi * 8 \text{ mm} = 50.24 \text{ mm}$$

$$\text{สเตปป์มอเตอร์แบบ 1.8 องศา/สเตปป์ 1 รอบจะได้} = 200 \text{ สเตปป์}$$

$$\text{จาก 200 สเตปป์} = 50.24 \text{ mm}$$

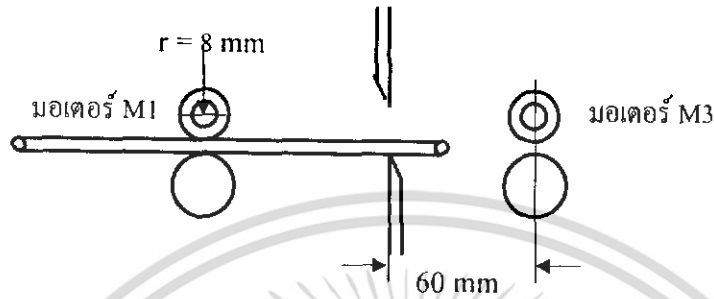
$$\text{จะได้ 1 สเตปป์} = 50.24 \text{ mm}/200 \text{ สเตปป์} = 0.251 \text{ mm}$$

$$\text{ดังนั้น 1/2 สเตปป์} = 0.251 \text{ mm}/2 = 0.125 \text{ mm}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

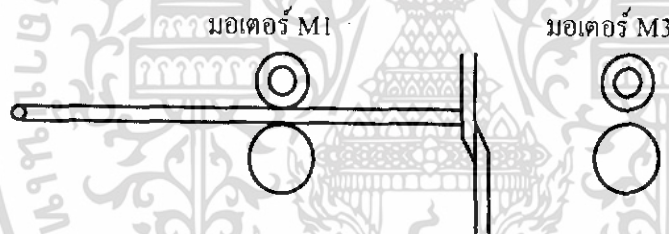
### 3.6 ลำดับขั้นตอนการตัดและปอกสายไฟ

ในขั้นตอนของการตัดและปอกสายไฟนั้น หลังจากที่กำหนดค่าต่างๆ ลงไปแล้ว สิ่งที่สำคัญอีกอย่างคือ ตำแหน่งการวางสายไฟ ซึ่งตำแหน่งการวางที่เหมาะสมนั้นจะต้องวางส่วนปลายของสายไฟให้ยื่นเลยใบมีดออกมา โดยรูปแบบของการวางแสดงได้ดังรูปที่ 3.11 และเมื่อกดปุ่ม 'START' แล้วเครื่องก็จะเริ่มปอกสายไฟโดยมีลำดับขั้นตอนการปอกดังนี้



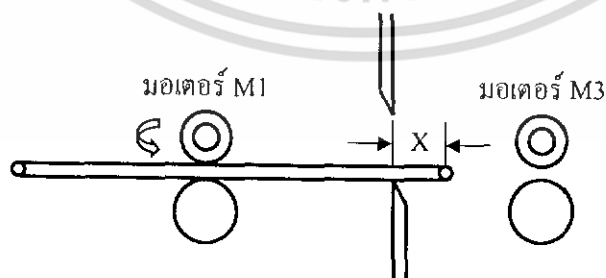
รูปที่ 3.11 รูปแบบการวางสายไฟตอนเริ่มต้นที่ถูกต้อง

1. ใบมีดส่วนบนที่ติดติดอยู่กับ M2 จะเลื่อนลงมาตัดสายไฟส่วนที่ยื่นเลยใบมีดออกมาให้ขาด แสดงได้ดังรูปที่ 3.11 แล้วเลื่อนขึ้นกลับไปยังตำแหน่งเดิม แล้วหยุด



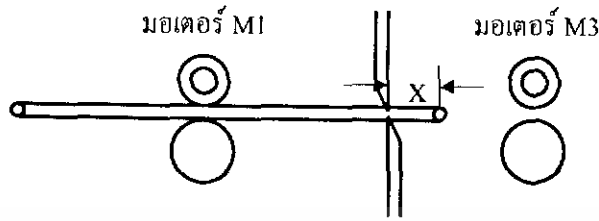
รูปที่ 3.12 ใบมีดส่วนบนเลื่อนลงมาตัดสายไฟที่ยื่นเลยใบมีด

2. ชุดลูกยางที่ยึดติดกับแกนของ M1 จะเลื่อนสายไฟไปข้างหน้าด้วยความยาวเท่ากับ X เมื่อได้ความยาวเท่ากับ X แล้ว M1 ก็จะหยุดเลื่อน แสดงได้ดังรูปที่ 3.13



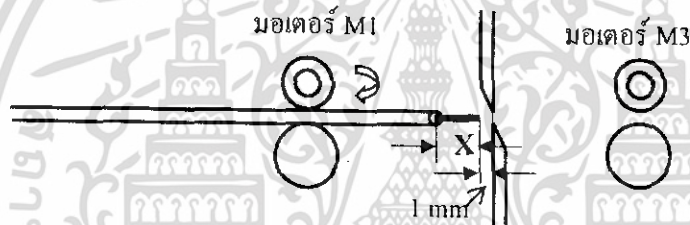
รูปที่ 3.13 ชุดลูกยางของ M1 เลื่อนสายไฟไปข้างหน้ายาวเท่ากับ X

3. ไบมีคส่วนบนจะเลื่อนลงมาตัดที่ถนนสายไฟให้มีความลึกจนถึงผิวของเส้นลวดตัวนำ (จะไม่ตัดเข้าถึงเส้นลวดตัวนำ) ดังแสดงในรูปที่ 3.14 ซึ่งตำแหน่งหรือความลึกที่ตัดนั้น ได้รับและ set ให้คงที่ไว้แล้วว่าจะต้องลึกขนาดไหนโดยการกำหนดจำนวน step ของ M2 ให้คงที่



รูปที่ 3.14 ไบมีคส่วนบนเลื่อนลงมาตัดถนนของสายไฟ

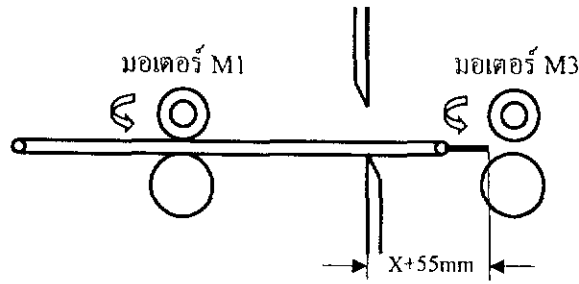
4. ชุดลูกยางที่ยึดติดอยู่กับแกนของ M1 จะเลื่อนสายไฟกลับด้วยความยาวเท่ากับ  $X + (1 \text{ mm})$  เมื่อได้ความยาวที่กำหนดแล้วก็จะหยุด และก็จะเป็นการปกอสายไฟเสร็จไปแล้ว 1 ช่วง แสดงได้ดังรูปที่ 3.15 (เหตุผลที่ต้องบวก 1 mm เข้าไปก็เพื่อที่จะทำให้ถนนและเส้นลวดตัวนำหลุดออกจากกัน)



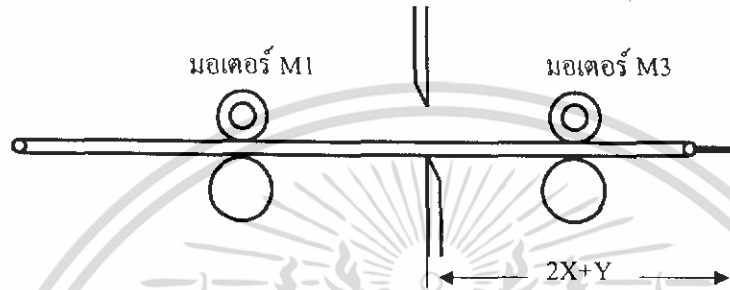
รูปที่ 3.15 ชุดลูกยางของ M1 เลื่อนสายไฟกลับด้วยความยาวเท่ากับ  $X + (1 \text{ mm})$

5. ไบมีคส่วนบนจะเลื่อนกลับขึ้นไปยังตำแหน่งเดิม แล้วหยุด

6. ชุดลูกยางที่ยึดติดกับแกน M1 จะเลื่อนสายไฟไปข้างหน้าด้วยความยาวเท่ากับ  $2X + Y$  โดยในระหว่างนั้นเมื่อเลื่อนสายไฟได้ความยาวเท่ากับ  $X + (55 \text{ mm})$  ชุดลูกยางที่ยึดติดกับแกนของ M3 จะหมุนรอบ แสดงได้ดังรูปที่ 3.16 เหตุผลที่ต้องหมุนรอบก็เพื่อว่า เวลาที่สายไฟเลื่อนมาถึง M3 แล้วจะทำให้สายไฟสามารถเลื่อนต่อไปได้ (เนื่องจากระยะห่างระหว่างชุดไบมีคกับจุดกึ่งกลางชุดลูกยางของ M3 เท่ากับ 60 mm และจากที่กล่าวมาข้างต้น จำเป็นที่จะต้องให้ชุดลูกยางของ M3 หมุนรอบก่อนที่สายไฟจะเลื่อนไปถึง ดังนั้นจึงได้กำหนดให้ชุดลูกยางของ M3 หมุนรอบเมื่อชุดลูกยางของ M1 หมุนมาได้เท่ากับ  $X + (55 \text{ mm})$ ) เมื่อเลื่อนไปถึงความยาวที่กำหนดแล้ว ชุดลูกยางของทั้ง M1 และ M3 จะหยุดหมุน ดังแสดงในรูปที่ 3.17

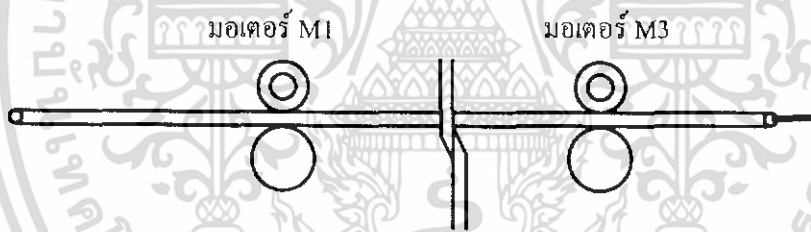


รูปที่ 3.16 ชุดลูกยางของ M3 จะหมุนรอบเมื่อสายไฟเลื่อนมาได้  $X + (55 \text{ mm})$



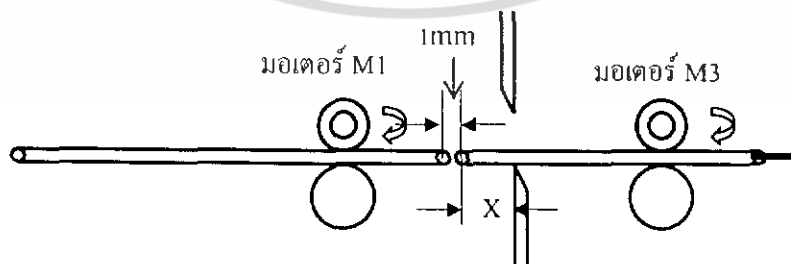
รูปที่ 3.17 ชุดลูกยางของ M1 และ M3 จะหยุดหมุนเมื่อสายไฟเลื่อนมาได้ระยะเท่ากับ  $2X + Y$

7. ไบมีดส่วนบนจะเลื่อนลงมาตัดสายไฟให้ขาดออกจากกัน แสดงได้ดังรูปที่ 3.18 หลังจากนั้นก็เลื่อนขึ้นกลับไปอยู่ยังตำแหน่งเดิม แล้วหยุด



รูปที่ 3.18 ไบมีดส่วนบนเลื่อนลงมาตัดสายไฟให้ขาดออกจากกัน

8. ชุดลูกยางที่ยึดติดกับแกนทั้งของ M1 และ M3 จะเลื่อนสายไฟกลับโดยที่ชุดลูกยางของ M1 จะเลื่อนกลับที่ความยาวเท่ากับ  $X + (1 \text{ mm})$  ในขณะที่ชุดลูกยางของ M3 จะเลื่อนกลับที่ความยาวเท่ากับ  $X$  แสดงได้ดังรูปที่ 3.19

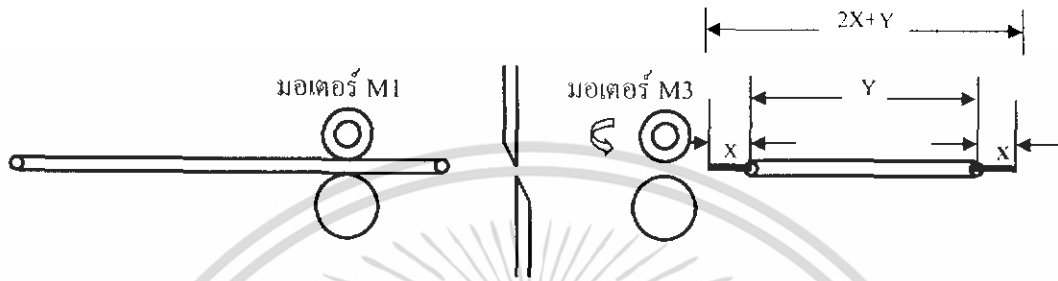


รูปที่ 3.19 ชุดลูกยางของ M1 และ M3 เลื่อนสายไฟกลับตามความยาวที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. ไบมีดส่วนบนจะเลื่อนลงมาตัดจนวนสายไฟให้มีความลึกถึงผิวของเส้นลวดตัวนำ เหมือนกับขั้นตอนในข้อ 3.

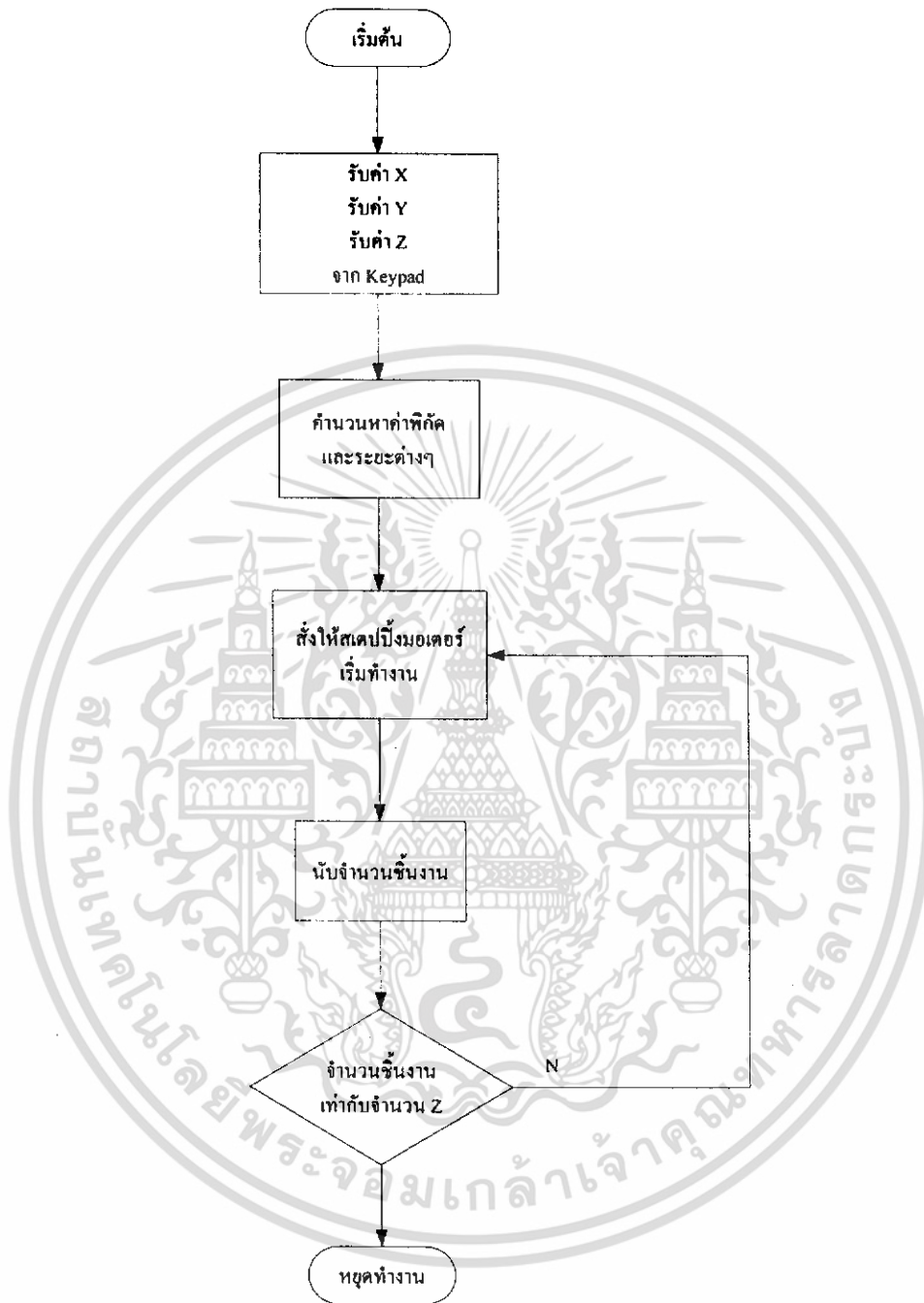
10. ชุดลูกยางของ M3 จะเลื่อนสายไฟไปข้างหน้าด้วยความยาวเท่ากับ  $2X+Y+(10 \text{ mm})$  เหตุผลที่บวก  $10 \text{ mm}$  เข้าไปก็เพื่อต้องการให้ชุดลูกยางของ M3 เลื่อนสายไฟออกมาเอง เมื่อเลื่อนได้ ความยาวตามที่กำหนดแล้วก็จะหยุดเลื่อนและถือเป็นการสิ้นสุดการปอกสายไฟ 1 เส้น แสดงได้ดัง รูปที่ 3.20



รูปที่ 3.20 ลักษณะของสายไฟที่ปอกเสร็จแล้ว

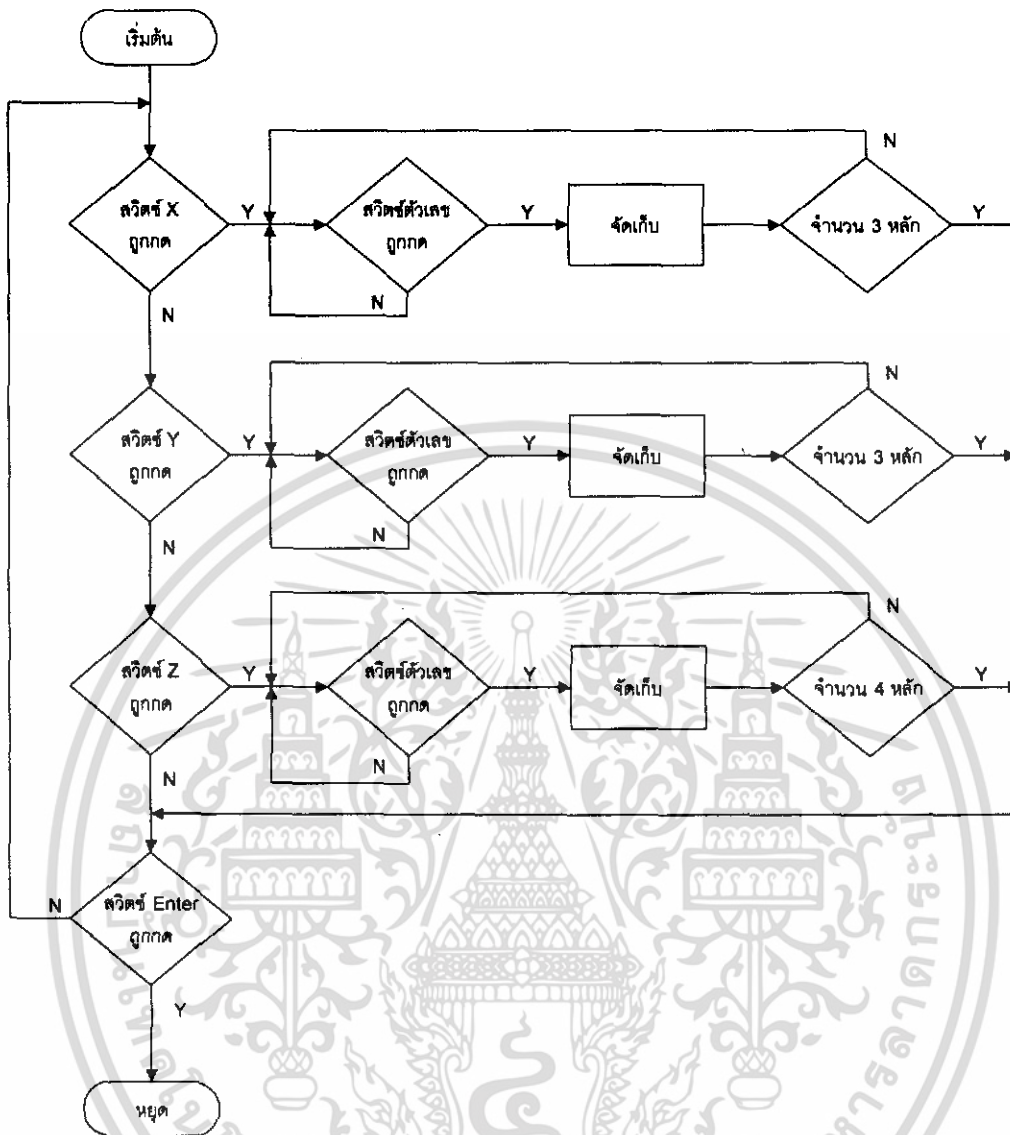
11. ไบมีดส่วนบนจะเลื่อนกลับขึ้นไปอยู่ในตำแหน่งเดิม แล้วหยุด หลังจากนั้นชุดลูกยางของ M1 จะเลื่อนสายไฟไปข้างหน้าด้วยความยาวเท่ากับ  $X$  และจะเริ่มต้นการปอกเส้นต่อๆ ไปตาม ขั้นตอนในข้อ 3. เรื่อยมา จนได้จำนวนสายไฟที่ปอกเสร็จครบตามจำนวนที่เรากำหนดค่าไว้ในตอน เริ่มต้น (ค่าของ  $Z$ ) ก็จะถือเป็นการเสร็จสิ้นการปอกสายไฟ

## 3.7 Flow Chart การทำงาน



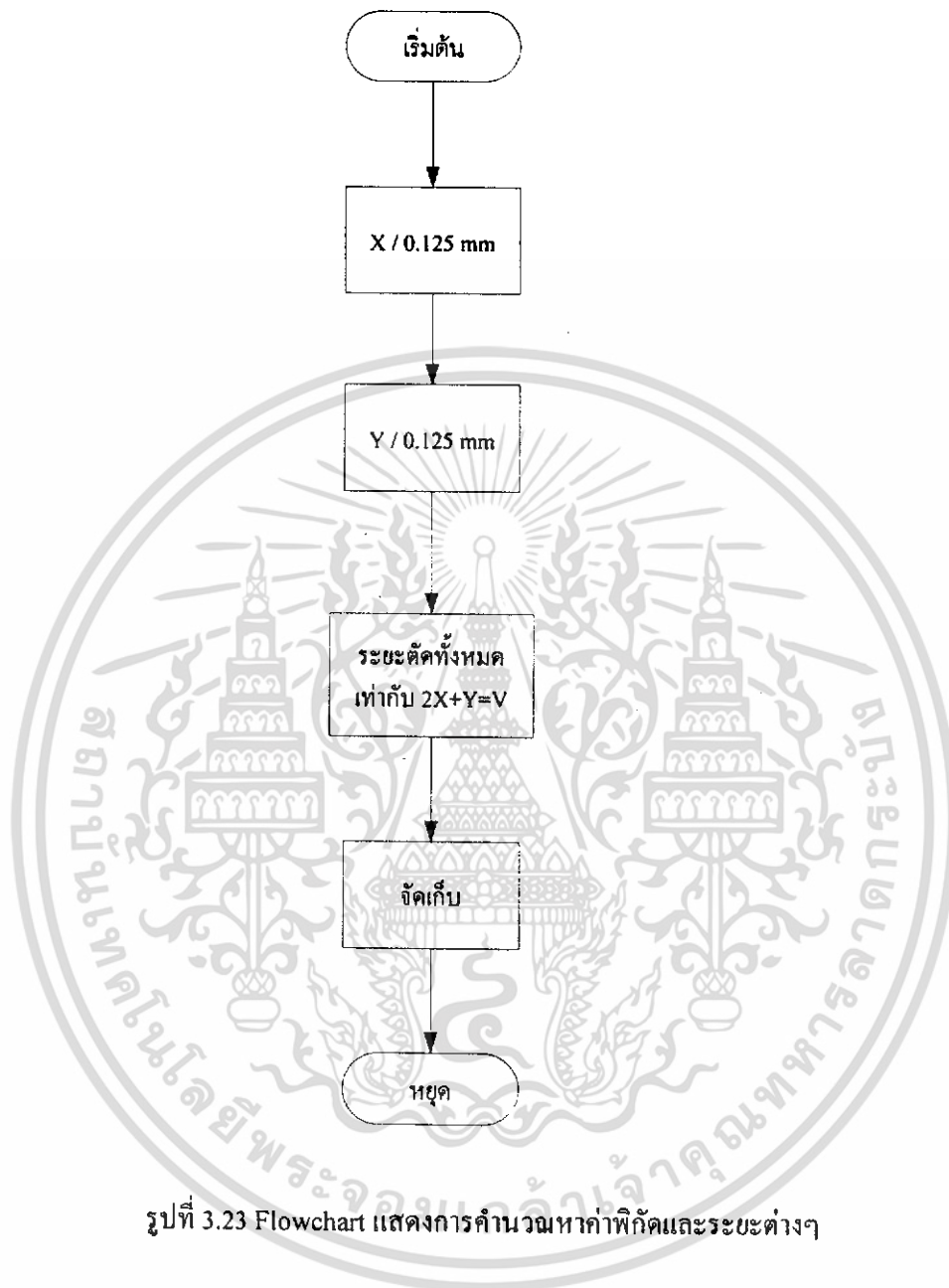
รูปที่ 3.21 Flowchart แสดงการทำงานของเครื่องตัดและปอกสายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



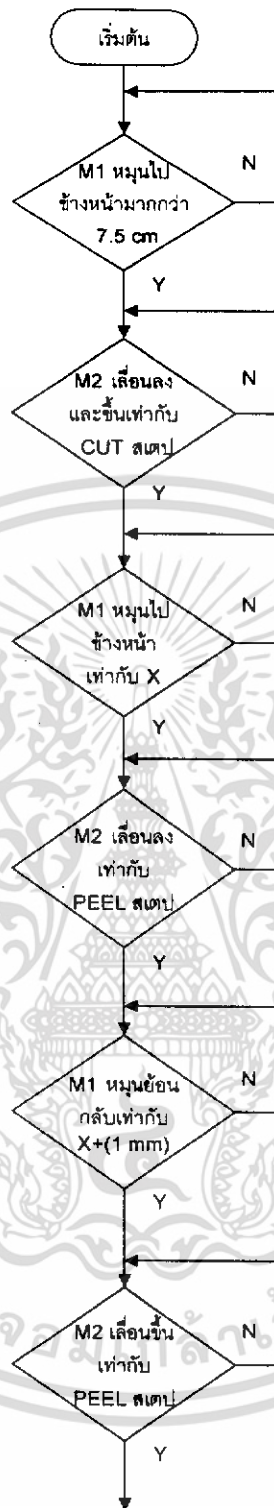
รูปที่ 3.22 Flowchart แสดงการรับค่าต่างๆ จาก Keypad

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



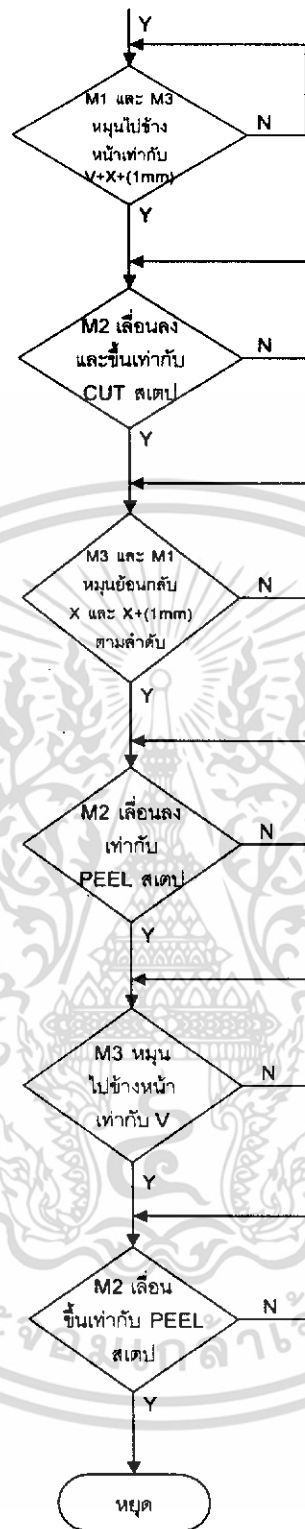
รูปที่ 3.23 Flowchart แสดงการคำนวณหาค่าพิกัดและระบะต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.24 Flowchart แสดงขั้นตอนการตัดและปอกสายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.25 Flowchart แสดงขั้นตอนการตัดและปอกสายไฟ(ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การทดลอง

ตารางผลการทดลองที่ 1 ของการตัดสายที่มีความยาว 10.0 เซนติเมตร

ขนาดของสายที่ตัด (Y)(cm)	ระยะของสายที่ปอก (X)(cm)	จำนวนสาย(เส้น) ที่ตัด (Z)(cm)	ความยาวที่ตัดได้จริง โดยเฉลี่ย (cm)	% ความผิดพลาด (%)
ขนาด 10.0	ขนาด 1.00	5	9.1	9
		10	8.9	11
		15	9.4	6
		20	9.6	4
		25	9.8	2
		30	9.3	7

ตารางผลการทดลองที่ 2 ของการตัดสายที่มีความยาว 15.0 เซนติเมตร

ขนาดของสายที่ตัด (Y)(cm)	ระยะของสายที่ปอก (X)(cm)	จำนวนสาย(เส้น) ที่ตัด (Z)(cm)	ความยาวที่ตัดได้จริง โดยเฉลี่ย (cm)	% ความผิดพลาด (%)
ขนาด 15.0	ขนาด 1.00	5	14.5	3.3
		10	14.6	2.7
		15	14.4	4
		20	14.8	1.33
		25	14.6	2.7
		30	14.7	2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางผลการทดลองที่ 3 ของการตัดสายที่มีความยาว 20.0 เซนติเมตร

ขนาดของสายที่ตัด (Y)(cm)	ระยะของสายที่ปอก (X)(cm)	จำนวนสาย(เส้น) ที่ตัด (Z)(cm)	ความยาวที่ตัดได้จริง (โดยเฉลี่ย) (cm)	% ความผิดพลาด (%)
ขนาด 20.0	ขนาด 1.00	5	19.2	4
		10	19.6	2
		15	19.4	3
		20	19.5	2.5
		25	19.3	3.5
		30	19.8	1

ตารางผลการทดลองที่ 4 ของการตัดสายที่มีความยาว 25.0 เซนติเมตร

ขนาดของสายที่ตัด (Y)(cm)	ระยะของสายที่ปอก (X)(cm)	จำนวนสาย(เส้น) ที่ตัด (Z)(cm)	ความยาวที่ตัดได้จริง (โดยเฉลี่ย) (cm)	% ความผิดพลาด (%)
ขนาด 25.0	ขนาด 1.00	5	24.3	2.8
		10	24.1	3.6
		15	24.9	0.4
		20	24.5	2
		25	24.5	2
		30	24.2	3.2

\*\*\* หมายถึง สายไฟที่ใช้ในการทดลองมีขนาดเส้นผ่านศูนย์กลาง 1 มิลลิเมตร

\*\*\* หมายถึง สายไฟที่ใช้ในการทดลองไม่สามารถทำการปอกได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปและวิจารณ์ผลการทดลอง

#### 5.1 สรุปผลการทดลอง

การทำงานของเครื่องตัดและปอกสายไฟมีกระบวนการทำงานของเครื่องเป็นไปตามที่ได้มีการออกแบบไว้ คือมอเตอร์ M1 และมอเตอร์ M3 จะเป็นตัวเลื่อนสายเข้าและออกตามขั้นตอนที่ได้กำหนดไว้ ส่วนมอเตอร์ M2 ก็จะทำการตัดและปอก ในส่วนของจอแสดงผลและกีย์สวิตซ์มีการทำงานสมบูรณ์ตามที่ได้ออกแบบไว้เช่นกัน และจากตารางผลการทดลองทั้งหมดจะเห็นได้ว่า ค่าความยาวของสายที่ตัดได้นั้นจะไม่ได้ความยาวของสายตามที่กำหนดไว้ โดยจะเกิดความคลาดเคลื่อนของขนาดความยาวของสายไฟ ซึ่งไม่ตรงกับความต้องการจริง ส่วนในด้านการปอกนั้นสายไฟที่ใช้ทดลองไม่สามารถทำการปอกได้จริง แต่กระบวนการทำงานของเครื่องจะทำงานตามรูปแบบการปอกที่วางแผนไว้

#### 5.2 ปัญหาและแนวทางแก้ไข

จากการทดลองปัญหาที่เกิดขึ้นในการตัดมีหลายสาเหตุ ซึ่งพอจะสรุปได้ดังนี้

1. ลักษณะของสายไฟที่จะทำการตัดและปอกนั้นจะมีความโค้งงอไม่สม่ำเสมอ ก็จะมี ความตึงไม่เท่ากันตลอดทั้งเส้น การแก้ไขคือ จะต้องมีการทำสายไฟให้มีความตึงเท่ากันตลอดทั้งเส้นและไม่มี ความโค้งงอของสาย โดยอาจจะใช้เครื่องที่ดึงสายไฟให้ตึงตลอดก่อนที่จะมาป้อนเข้าที่เครื่องตัดและปอกสายไฟ

2. ท่อที่ใช้เป็นทางเดินของสายไฟมีขนาดค่อนข้างใหญ่เมื่อเทียบกับขนาดของสายไฟที่ทำการทดลอง ทำให้เวลาที่สายไฟเลื่อนเข้าไปทิศทางการเคลื่อนที่เกิดการแกว่งไม่เป็นเส้นตรง เมื่อปลายสายไหลพ้นท่อนำสายออกมาแล้วก็จะไปชนกับตัวใบมีดส่วนล่างทำให้สายไฟติดอยู่ที่ใบมีดส่วนล่างไม่สามารถเลื่อนไปอยู่ในร่องของใบมีดส่วนล่างได้ ในขณะที่กระบวนการในการตัดและปอกของชุดมอเตอร์และโปรแกรมยังคงทำงานเหมือนเดิม แต่สายที่จะทำการตัดและปอกนั้นไม่สามารถเป็นไปตามกระบวนการได้ การแก้ไขคือเปลี่ยนขนาดของท่อนำสายให้มีขนาดเล็กลง

3. ลูกยางที่ขบอยู่กับลูกยางที่แกนของมอเตอร์ ที่ใช้ในการเลื่อนสายไฟบางครั้งเมื่อใช้งานเป็นเวลานานๆจะมีความหลวมเกินไป ทำให้มอเตอร์ไม่สามารถดึงสายไฟเข้าไปได้อย่างสม่ำเสมอ การแก้ไขคือ ต้องคอยตรวจสอบลูกยางให้อยู่ในสภาพพร้อมใช้งานอยู่เสมอ

4. ในระหว่างที่ทำการปอกนั้นจะมีเศษฉนวนที่เกิดจากการปอก ค้างอยู่ในท่อเป็นจำนวนมาก การแก้ไขคือ ต้องออกแบบให้ท่อนำสายมีลักษณะที่หมุนได้ เพื่อที่จะสามารถเศษฉนวนออกมาหลังจากเสร็จสิ้นการทำงาน

ส่วนการปกอสายไฟนั้นไม่สามารถทำการปกอจริงได้ สรุปสาเหตุได้ดังนี้

1. ผิวของสายไฟที่นำมาใช้ในการทดลอง มีความลื่นและมัน จึงทำให้ในตอนที่มีมอเตอร์หมุนกระชากกลับเพื่อที่จะปกอนั้น เกิดการลุดของสาย
2. ไบมีดที่ใช้ทำการปกอลงไปไม่ถึงตำแหน่งที่จะทำการปกอ ซึ่งเกิดจากการวางตำแหน่งของไบมีดที่ไม่ตรงจุดพอ
3. ความคมของไบมีดที่จะปกอไม่มากพอ
4. มีการแกว่งของสายขณะทำการปกอ ทำให้บางครั้งสายไฟอยู่ไม่ตรงจุดกึ่งกลางของไบมีด

### 5.3 แนวทางในการพัฒนา

1. ไบมีดที่ใช้ทำการตัดและปกอ ต้องออกแบบให้แข็งแรงและได้มาตรฐาน คือควรจะทำจากเหล็กที่มีความแข็งสูง เพื่อให้สามารถใช้งานได้เป็นเวลานานๆ
2. ปรับปรุงให้เครื่องสามารถตัดและปกอสายไฟให้ได้หลายๆขนาด
3. ปรับปรุงตำแหน่งของชุดมอเตอร์เพื่อให้สามารถตัดสายไฟที่มีความยาวสั้นกว่าเดิมได้ โดยวางโครงสร้างของมอเตอร์ M2 และมอเตอร์ M3 ให้อยู่ใกล้กันมากขึ้น
4. ปรับปรุงทางโครงสร้างของชุดมอเตอร์ใหม่ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เอกสารอ้างอิง

1. สุทธิศักดิ์ พงศ์ธนพานิช, “การใช้งานไมโครคอนโทรลเลอร์ตระกูล8051”, กรุงเทพฯ, ซีเอ็ดยูเคชั่น, 2537
2. Takashi Kenjo, Stepping Motors and their Microprocessor Controls, Oxford, Clarendon Press, 1984
3. [www.phy.davidson.edu/instrumentation/Files/Stepper/5804.pdf](http://www.phy.davidson.edu/instrumentation/Files/Stepper/5804.pdf)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก

## โปรแกรมควบคุม

```

/*-----*/
-----*/
// Program : Main Program
// Description : All systems
// Filename : MainPro
// C compiler : RIDE 51 V6.1
/*-----*/
-----*/
#include <8051.h>
#include <lcd.h>
#include <scankey.h>
#include <startup.h>
/*-----*/
-----*/
sbit at 0xa7 en_m1 ;
sbit at 0xa6 di_m1;
sbit at 0xa5 en_m2;
sbit at 0xa4 di_m2;
sbit at 0xa3 en_m3;
sbit at 0xa2 di_m3;
sbit at 0xa1 ck_hi;
sbit at 0xa0 ck_lo;
sbit at 0xb2 ck_out;
sbit at 0x95 c5; // Bit Column5
sbit at 0x93 r3; // Bit Row3

/*****
*****/
***** Chang Data to Font
*****/
/*****
*****/
unsigned char changdata(unsigned char data1) // Return value key
{ unsigned char data2;
  switch(data1)
  {
    case 0x01 : data2 = '7';
                                break;
    case 0x02 : data2 = '8';
                                break;
    case 0x03 : data2 = '9';
                                break;
    case 0x05 : data2 = '4';
                                break;
    case 0x06 : data2 = '5';
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break;
    case 0x07 : data2 = '6';
        break;
    case 0x09 : data2 = '1';
        break;
    case 0x0a : data2 = '2';
        break;
    case 0x0b : data2 = '3';
        break;
    case 0x0d : data2 = '0';
        break;
    case 0x0e : data2 = 0;
        break;
    case 0x0f : data2 = 0;
        break;
    case 0x04 : data2 = 0;
        break;
    case 0x08 : data2 = 0;
        break;
    case 0x0c : data2 = 0;
        break;
    case 0x00 : data2 = 0;
        break;
    }
    return(data2);
}

/*****
*****/
/***** Chack key Data
*****/
/*****
*****/
void keyinput(unsigned char line, unsigned char point, unsigned char key2) //write
command
{
    if (line==1)
    {
        display_lcd(1,point,key2);
        if (point==10)
        {
            lcd_init2(1,7);
        }
        if (point==7)
        {
            lcd_init2(1,8);
        }
        if (point==8)
        {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        lcd_init2(1,10);
    }
}
if (line==2)
{
    display_lcd(2,point,key2);
    if (point==10)
    {
        lcd_init2(2,7);
    }
    if (point==7)
    {
        lcd_init2(2,9);
    }
    if (point==9)
    {
        lcd_init2(2,10);
    }
}
if (line==3)
{
    display_lcd(1,point,key2);
    if (point==11)
    {
        lcd_init2(1,12);
    }
    if (point==12)
    {
        lcd_init2(1,13);
    }
    if (point==13)
    {
        lcd_init2(1,14);
    }
    if (point==14)
    {
        lcd_init2(1,11);
    }
}
}
}

/*****
*****/
/***** Driving Motor *****/
/*****
*****/
void drivemoter(unsigned char speed, unsigned int value)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{ unsigned int loop4;
if (speed==1)
{
for (loop4=0;loop4<value;loop4++)
{
do
{
ck_out=0;
}
while(ck_lo==0);
do
{
ck_out=1;
}
while(ck_lo==1);
}
}

if (speed==2)
{
for (loop4=0;loop4<value;loop4++)
{
do
{
ck_out=0;
}
while(ck_hi==0);
do
{
ck_out=1;
}
while(ck_hi==1);
}
}

/*****
*****/
/***** Main loop
*****/
/*****
*****/

void main(void)
{ unsigned long int x_value,y_value,z_value,loop2;
  unsigned char key,key2,loop,z5,z6,z7,z8,v1,v2,v3,v4,v5,v6,v7,v8,v9,v10;
  unsigned char x1='0',x2='0',x3='0',y1='0',y2='0',y3='0';
  unsigned char
z1='0',z2='0',z3='0',z4='0',z_value1=0,z_value2=0,z_value3=0,z_value4=0,z_value5=
0;
  delay(2000);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

delay(2000);
delay(2000);
lcd_init();          // Initial LCD
startdisplay_p2(z1,z2,z3,z4,x1,x2,x3,y1,y2,y3);
while(1)            // Jump at hear(infinite)
{
  en_m1==1; //motor1 off
  en_m2==1; //motor2 off
  en_m3==1; //motor3 off
  key = scandatakey();

  if (key==0x08)
  {
    lcd_init();
    startdisplay_p1(x1,x2,x3,y1,y2,y3);
    lcd_init2(1,7);
    loop=0;
    do
    {
      key = scandatakey();
      key2= changdata(key);
      if (key2!=0)
      {
        loop++;
        if (loop == 1)
        {
          x1=key2;
          keyinput(1,7,key2);
        }
        if (loop == 2)
        {
          x2=key2;
          keyinput(1,8,key2);
        }
        if (loop == 3)
        {
          x3=key2;

          loop=0;
          keyinput(1,10,key2);
        }
      }
    } while(key!=0x00);
    lcd_init();
    startdisplay_p1(x1,x2,x3,y1,y2,y3);
  }

  if (key==0x0c)
  {
    lcd_init();
    startdisplay_p1(x1,x2,x3,y1,y2,y3);
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lcd_init2(2,7);
loop=0;
do
{
key = scandatakey();
key2= changdata(key);
if (key2!=0)
{
loop++;
if (loop == 1)
{
y1=key2;
keyinput(2,7,key2);
}
if (loop == 2)
{
y2=key2;
keyinput(2,9,key2);
}
if (loop == 3)
{
y3=key2;
loop=0;
keyinput(2,10,key2);
}
} while(key!=0x00);
lcd_init();
startdisplay_p1(x1,x2,x3,y1,y2,y3);
}

if (key==0x0f)
{
startdisplay_p2(z1,z2,z3,z4,x1,x2,x3,y1,y2,y3);
lcd_init2(1,11);
loop=0;
do
{
key = scandatakey();
key2= changdata(key);
if (key2!=0)
{
loop++;
if (loop == 1)
{
z1=key2;
keyinput(3,11,key2);
}
if (loop == 2)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

        for (loop=0;loop<v3;loop++)
        {
            z_value = z_value+10;
        }
        for (loop=0;loop<v4;loop++)
        {
            z_value++;
        }

        for (loop=0;loop<v5;loop++)
        {
            x_value = x_value+10000;
        }
        for (loop=0;loop<v6;loop++)
        {
            x_value = x_value+1000;
        }
        for (loop=0;loop<v7;loop++)
        {
            x_value = x_value+100;
        }

        for (loop=0;loop<v8;loop++)
        {
            y_value = y_value+1000;
        }
        for (loop=0;loop<v9;loop++)
        {
            y_value = y_value+100;
        }
        for (loop=0;loop<v10;loop++)
        {
            y_value = y_value+10;
        }

        x_value = x_value/12.5;
        y_value = y_value/12.5;
        x_value = x_value-y_value;
        en_m1=0; //motor1 on
        en_m2=0; //motor2 on
        di_m1=1; //motor1 forward
        di_m2=1; //motor2 forward
        drivemoter(2,20+x_value);
        en_m1=1; //motor1 off
        en_m2=1; //motor2 off
        en_m3=0; //motor3 on
        di_m3=1; //motor3 slit down cutting
        drivemoter(2,120);
        di_m3=0; //motor3 slit up
        drivemoter(2,120);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if ((z_value>0)&&(x_value>0)&&(y_value>0))
//
do
{
for (loop2==0;loop2<z_value;loop2++)
{
en_m1=0; //motor1 on
en_m2=0; //motor2 on
en_m3=1; //motor3 off
di_m1=1; //motor1 forward
di_m2=1; //motor2 forward
drivemoter(1,y_value);

en_m1=1; //motor1 off
en_m2=1; //motor2 off
en_m3=0; //motor3 on
di_m3=1; //slit down
drivemoter(1,100);

en_m3=1; //motor3 off
en_m2=0; //motor2 on
di_m2=0; //reverse hi
drivemoter(2,80);

en_m2=1; //motor2 off
en_m3=0; //motor3 on
di_m3=0; //slit up
drivemoter(1,100);

en_m3=1; //motor3 off
en_m1=0; //motor1 on
di_m1=1; //motor1 forward
en_m2=0; //motor2 on
di_m2=1; //motor2 forward
drivemoter(1,80+x_value);

en_m1=1; //motor1 off
en_m2=1; //motor2 off
en_m3=0; //motor3 on
di_m3=1; //motor3 slit down cutting
drivemoter(2,120);

di_m3=0; //motor3 slit up
drivemoter(2,120);

en_m3=1; //motor3 off
en_m1=0; //motor1 on
di_m1=0; //motor1 reverse
drivemoter(1,y_value);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

/*-----*/
-----*/
// Program : LCD display 16 x 2
// Description : Show Message
// Filename : LCD Display
// C compiler : RIDE 51 V6.1
/*-----*/
-----*/
sfr at 0x80 P0;
sbit at 0xb0 e; // Bit control Enable recive data LCD
sbit at 0xb1 rs;

// Bit control Command/Data LCD
/*****
*****/
/***** Function Delay time *****/
/*****
*****/
void delay(int time)
{
    do // do-while loop for delay
    {
        time--; // Decrease counter
    }while(time>0); // If time>0 work in block
}
/*****
*****/
/***** Function send Command to LCD *****/
/*****
*****/
void led_command(unsigned char com)
{
    rs = 0; // Write Command to LCD
    e = 1; // Start generate pulse clock LCD
    P0 = com; // Send data to LCD port
    delay(30); // Delay
    e = 0; // End pulse clock
    delay(30); // Delay
}
/*****
*****/
/***** Function send Data to LCD *****/
/*****
*****/
void led_text(unsigned char text) //write command
{
    rs = 1; // Write Data to LCD
    e = 1; // Start generate pulse clock LCD
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

P0 = text;          // Send data to LCD port
delay(30);          // Delay
c = 0;             // End pulse clock
delay(30);         // Delay
}
/*****
*****/
/***** Function set initial parameter LCD
*****/
/*****
*****/
void lcd_init()
{
    delay(400);          // Delay for initial LCD
    lcd_command(0x38);  // 2 line ,8 bit display ,5*7 dot
    lcd_command(0x0C);  // Display ON , none cursor
    lcd_command(0x01); // Clear screen
}
void lcd_init2(unsigned int line,unsigned char point_lcd)
{
    // delay(400);          // Delay for initial LCD
    lcd_command(0x0F);  // Display ON , have cursor
    if (line==1)
    {
        lcd_command(0x80+point_lcd); // Go to address 0x00
    }
    if (line==2)
    {
        lcd_command(0xc0+point_lcd); // Go to address 0x00
    }
}
/*****
*****/
/***** Line of Diplay
*****/
/*****
*****/
void display_lcd_all (unsigned int line,unsigned char std_lcd,unsigned char text[])
{ unsigned char m;
  if (line==1)
  {
    lcd_command(0x80+std_lcd); // Go to address 0x00
    for(m=0;m<16;m++) // For loop to send data to LCD
    {
        lcd_text(text[m]); // Send data to LCD
    }
  }
  if (line==2)
  {
    lcd_command(0xc0+std_lcd); // Go to address 0x00
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(m=0;m<16;m++) // For loop to send data to LCD
{
    lcd_text(text[m]); // Send data to LCD
}
}
}
/*****
*****/
/***** Pointe of Diplay
*****/
/*****
*****/
//void display_lcd ()
void display_lcd (unsigned int line,unsigned char std_lcd,unsigned char text)
{
    if (line==1)
    {
        lcd_command(0x80+std_lcd); // Go to address 0x00
        lcd_text(text); // Send data to LCD
    }
    if (line==2)
    {
        lcd_command(0xc0+std_lcd); // Go to address 0x00
        lcd_text(text); // Send data to LCD
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*-----*/
-----*/
// Program : Scan keypad 4x4
// Description : Scan keypad 4x4
// Filename : cutting wire machine liberly
// C compiler : RIDE 51 V6.1
/*-----*/
-----*/
sbit at 0x94 c4; // Bit Column4
sbit at 0x95 c5; // Bit Column5
sbit at 0x96 c6; // Bit Column6
sbit at 0x97 c7; // Bit Column7
sbit at 0x90 r0; // Bit Row0
sbit at 0x91 r1; // Bit Row1
sbit at 0x92 r2; // Bit Row2
sbit at 0x93 r3; // Bit Row3

/*****
*****/
/***** Function Scan keypad 4x3 *****/
/*****
*****/
unsigned char scankey(void) // Return value key
{
    unsigned char ret = 0xFF;
    c4 = 0;
    if(r0==0)
    {
        ret = 0x01;
    }
    if(r1==0)
    {
        ret = 0x05;
    }
    if(r2==0)
    {
        ret = 0x09;
    }
    if(r3==0)
    {
        ret = 0x0d;
    }
    c4 = 1;

    c5 = 0;
    if(r0==0)
    {
        ret = 0x02;
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
if(r1==0)
{
    ret = 0x06;
}
if(r2==0)
{
    ret = 0x0a;
}
if(r3==0)
{
    ret = 0x0e;
}
c5 = 1;

c6 = 0;
if(r0==0)
{
    ret = 0x03;
}
if(r1==0)
{
    ret = 0x07;
}
if(r2==0)
{
    ret = 0x0b;
}
if(r3==0)
{
    ret = 0x0f;
}
c6 = 1;

c7 = 0;
if(r0==0)
{
    ret = 0x04;
}
if(r1==0)
{
    ret = 0x08;
}
if(r2==0)
{
    ret = 0x0c;
}
if(r3==0)
{
    ret = 0x00;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    c7 = 1;
    return(ret);           // Return key value
}

unsigned char scandatakey(void)
{ unsigned char key1,key2=0;
  delay(20000);
  do
  {
    key1 = scankey();
  }
  while(key1!=0xff);
  do
  {
    key2 = scankey();
  }
  while(key2!=0xff);
  return(key1);
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*-----*/
-----*/
// Program : Starting Program
// Description : LCD and Scankey
// Filename : MainPro
// C compiler : RIDE 51 V6.1
/*-----*/
-----*/
code unsigned char line1[] = " Y = . cm ";
code unsigned char line2[] = " X = . cm ";
code unsigned char number[] = " Number = ";
code unsigned char number2[] = " . cm . cm";

*****
*****/
/***** Starting LCD
*****/
/*****
*****/
void startdisplay_p1(unsigned char dit1,unsigned char dit2,unsigned char
dit4,unsigned char dit5,unsigned char dit6,
unsigned char dit7)
{
    lcd_init(); // Initial LCD
    display_lcd_all(1,0x0,line1);
    display_lcd_all(2,0x0,line2);
    display_lcd(1,7,dit1);
    display_lcd(1,8,dit2);
    display_lcd(1,10,dit4);
    display_lcd(2,7,dit5);
    display_lcd(2,9,dit6);
    display_lcd(2,10,dit7);
}
void startdisplay_p2(unsigned char dit1,unsigned char dit2,unsigned char
dit3,unsigned char dit4,unsigned char dit5,unsigned char dit6,
unsigned char dit7,unsigned char dit8,unsigned char dit9,unsigned char dit10)
{
    lcd_init(); // Initial LCD
    display_lcd_all(1,0x0,amount);
    display_lcd_all(2,0x0,amount2);
    display_lcd(1,11,dit1);
    display_lcd(1,12,dit2);
    display_lcd(1,13,dit3);
    display_lcd(1,14,dit4);
    display_lcd(2,0,dit5);
    display_lcd(2,1,dit6);
    display_lcd(2,3,dit7);
    display_lcd(2,12,dit10);
    display_lcd(2,11,dit9);
    display_lcd(2,9,dit8);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
void startdisplay_p3(unsigned char dit1,unsigned char dit2,unsigned char
dit3,unsigned char dit4)
{
    lcd_init();        // Initial LCD
    display_lcd_all(1,0x0,amount);
    display_lcd(1,11,dit1);
    display_lcd(1,12,dit2);
    display_lcd(1,13,dit3);
    display_lcd(1,14,dit4);
}
void changdata3(unsigned char data1,unsigned point) // Return value key
{ unsigned char data2;
  switch(data1)
  {
    case 7 : data2 = '7';
                break;
    case 8 : data2 = '8';
                break;
    case 9 : data2 = '9';
                break;
    case 4 : data2 = '4';
                break;
    case 5 : data2 = '5';
                break;
    case 6 : data2 = '6';
                break;
    case 1 : data2 = '1';
                break;
    case 2 : data2 = '2';
                break;
    case 3 : data2 = '3';
                break;
    case 0 : data2 = '0';
                break;
    default: data2 = '0';
                break;
  }
  if (point==1)
  {
    display_lcd(1,11,data2);
  }
  if (point==2)
  {
    display_lcd(1,12,data2);
  }
  if (point==3)
  {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
display_lcd(1,13,data2);  
}  
if (point==4)  
{  
display_lcd(1,14,data2);  
}  
}
```



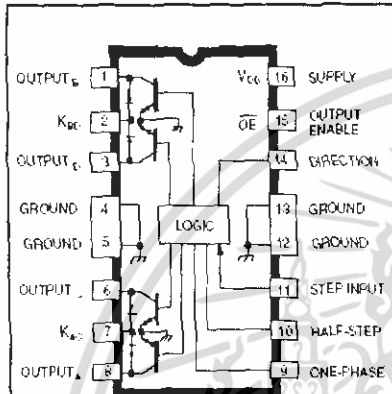
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข  
DATASHEET UCN5804B

# 5804

UCN5804B  
2618412C

## BiMOS II UNIPOLAR STEPPER-MOTOR TRANSLATOR/DRIVER



DA97 W-194

Note that the UCN5804B (dual in-line package) and UCN5804LB (small outline IC package) are electrically identical and share a common terminal number assignment.

### ABSOLUTE MAXIMUM RATINGS

Output Voltage, $V_{CE}$	50 V
Output Sustaining Voltage, $V_{CE(sust)}$	35 V
Output Sink Current, $I_{OUT}$	1.5 A
Logic Supply Voltage, $V_{CC}$	7.0 V
Input Voltage, $V_{IN}$	7.0 V
Package Power Dissipation, $P_D$	See Graph
Operating Temperature Range, $T_A$	-20° C to +85° C
Storage Temperature Range, $T_S$	-55° C to +150° C

Combining low-power CMOS logic with high-current and high-voltage bipolar outputs, the UCN5804B and UCN5804LB BiMOS II translator/drivers provide complete control and drive for a four-phase unipolar stepper motor with continuous output current ratings to 1.25 A per phase (1.5 A startup) and 35 V.

The CMOS logic section provides the sequencing logic, DIRECTION and OUTPUT ENABLE control, and a power-on reset function. Three stepper-motor drive formats, wave-drive (one-phase), two-phase, and half-step are externally selectable. The inputs are compatible with standard CMOS, PMOS, and NMOS circuits. TTL or LSTTL may require the use of appropriate pull-up resistors to ensure a proper input logic high.

The wave-drive format consists of energizing one motor phase at a time in an A-B-C-D (or D-C-B-A) sequence. This excitation mode consumes the least power and assures positional accuracy regardless of any winding imbalance in the motor. Two-phase drive energizes two adjacent phases in each detent position (A-B-BC-CD-DA). This sequence mode offers an improved torque-speed product, greater detent torque, and is less susceptible to motor resonance. Half-step excitation alternates between the one-phase and two-phase modes (A-AB-B-BC-C-CD-D-DA), providing an eight-step sequence.

The bipolar outputs are capable of sinking up to 1.5 A and withstanding 50 V in the off state (sustaining voltages up to 35 V). Ground-clamp and flyback diodes provide protection against inductive transients. Thermal protection circuitry disables the outputs when the chip temperature is excessive.

Both devices are rated for operation over the temperature range of -20° C to +85° C. The UCN5804B is supplied in a 16-pin dual in-line plastic batwing package with a copper lead frame and heat-sinkable tabs for improved power dissipation capabilities; the UCN5804LB is supplied in a 16-lead plastic SOIC batwing package with a copper lead frame and heat-sinkable tabs.

### FEATURES

- 1.5 A Maximum Output Current
- 35 V Output Sustaining Voltage
- Wave-Drive, Two-Phase, and Half-Step Drive Formats
- Internal Clamp Diodes
- Output Enable and Direction Control
- Power-On Reset
- Internal Thermal Shutdown Circuitry

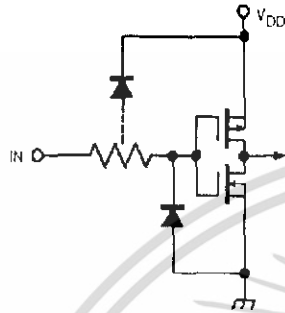
Always order by complete part number, e.g., **UCN5804B**.



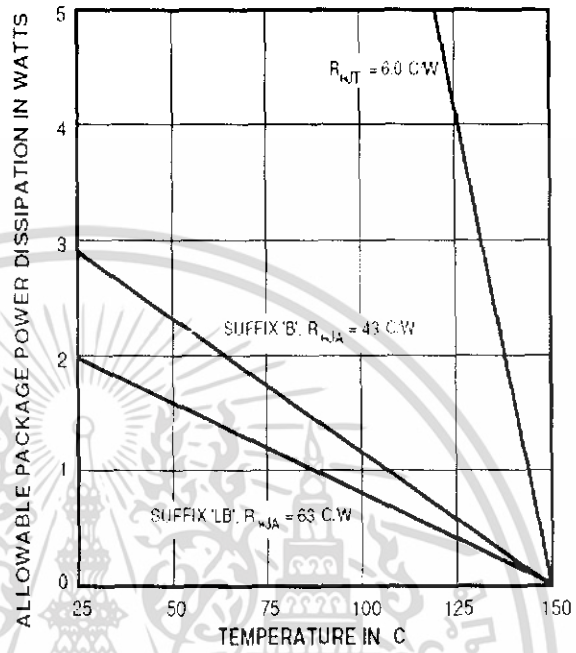
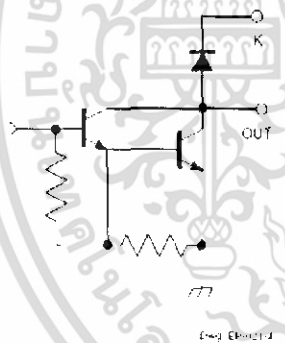
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**5804**  
**BIMOS II UNIPOLAR**  
**STEPPER-MOTOR**  
**TRANSLATOR/DRIVER**

TYPICAL INPUT CIRCUIT



TYPICAL OUTPUT DRIVER



TRUTH TABLE

Drive Format	Pin 9	Pin 10
Two-Phase	L	L
One-Phase	H	L
Half-Step	L	H
Step-Inhibit	H	H



115 Northeast Cutoff, Box 15036  
 Worcester, Massachusetts 01615-0036 (508) 853-5000  
 Copyright © 1987, 2000 Allegro MicroSystems, Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**5804**  
**BiMOS II UNIPOLAR**  
**STEPPER-MOTOR**  
**TRANSLATOR/DRIVER**

ELECTRICAL CHARACTERISTICS at  $T_A = 25^\circ\text{C}$ ,  $T_J \leq 150^\circ\text{C}$ ,  $V_{DD} = 4.5\text{ V to }5.5\text{ V}$   
(unless otherwise noted).

Characteristic	Symbol	Test Conditions	Limits			
			Min.	Typ.	Max.	Units
Output Leakage Current	$I_{CEX}$	$V_{OUT} = 50\text{ V}$	—	10	50	$\mu\text{A}$
Output Sustaining Voltage	$V_{CE(SUS)}$	$I_{OUT} = 1.25\text{ A}$ , $L = 3\text{ mH}$	35	—	—	V
Output Saturation Voltage	$V_{CE(SAT)}$	$I_{OUT} = 700\text{ mA}$	—	1.0	1.2	V
		$I_{OUT} = 1\text{ A}$	—	1.1	1.4	V
		$I_{OUT} = 1.25\text{ A}$	—	1.2	1.5	V
Clamp Diode Leakage Current	$I_R$	$V_R = 50\text{ V}$	—	10	50	$\mu\text{A}$
Clamp Diode Forward Voltage	$V_F$	$I_F = 1.25\text{ A}$	—	1.5	3.0	V
Input Current	$I_{IN(H)}$	$V_{IN} = V_{DD}$	—	0.5	5.0	$\mu\text{A}$
	$I_{IN(L)}$	$V_{IN} = 0.8\text{ V}$	—	-0.5	-5.0	$\mu\text{A}$
Input Voltage	$V_{IN(H)}$	$V_{DD} = 5\text{ V}$	3.5	—	5.3	V
	$V_{IN(L)}$		-0.3	—	0.8	V
Supply Current	$I_{DD}$	2 Outputs ON	—	20	30	mA
Turn-Off Delay	$t_{ON}$	50% Step Inputs to 50% Output	—	—	10	$\mu\text{s}$
Turn-On Delay	$t_{OFF}$	50% Step Inputs to 50% Output	—	—	10	$\mu\text{s}$
Thermal Shutdown Temperature	$T_J$		—	165	—	$^\circ\text{C}$

### TIMING CONDITIONS

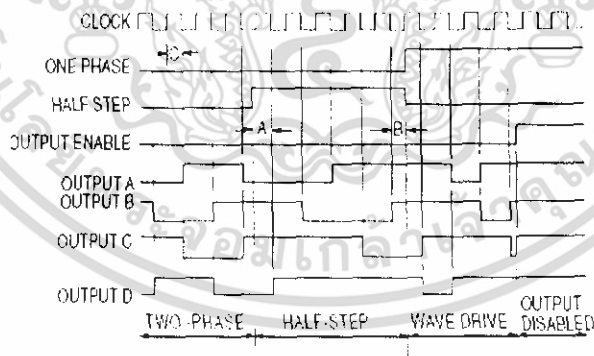


Fig. W-115A

- A. Minimum Data Set Up Time ..... 100 ns  
B. Minimum Data Hold Time ..... 100 ns  
C. Minimum Step Input Pulse Width ..... 3.0  $\mu\text{s}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5804 BiMOS II UNIPOLAR STEPPER-MOTOR TRANSLATOR/DRIVER

### APPLICATIONS INFORMATION

Internal power-on reset (POR) circuitry resets OUTPUT<sub>A</sub> and OUTPUT<sub>B</sub> in the two-phase drive format to the on state with initial application of the logic supply voltage. After reset, the circuit then steps according to the tables.

The outputs will advance one sequence position on the high-to-low transition of the STEP INPUT pulse. Logic levels on the HALF-STEP and ONE-PHASE inputs will determine the drive format (one-phase, two-phase, or half-step). The DIRECTION pin determines the rotation sequence of the outputs. Note that the STEP INPUT must be in the low state when changing the state of ONE-PHASE, HALF-STEP, or DIRECTION to prevent erroneous stepping.

All outputs are disabled (off) when OUTPUT ENABLE is at a logic high. If the function is not required, OUTPUT ENABLE should be tied low. In that condition, all outputs depend only on the state of the step logic.

During normal commutation of a unipolar stepper motor, mutual coupling between the motor windings can force the outputs of the UCN5804B below ground. This condition will cause forward biasing of the collector-to-substrate junction and source current from the output. For many LK applications, this substrate current is high enough to adversely affect the logic circuitry and cause misstepping. External series diodes (Schottky are recommended for increased efficiency at low-voltage operation) will prevent substrate current from being sourced through the outputs. Alternatively, external ground clamp diodes will provide a preferred current path from ground when the outputs are pulled below ground.

Internal thermal protection circuitry disables all outputs when the junction temperature reaches approximately 165°C. The outputs are enabled again when the junction cools down to approximately 145°C.

### WAVE-DRIVE SEQUENCE

Half Step = L, One Phase = H				
Step	A	B	C	D
POR	ON	OFF	OFF	OFF
1	ON	OFF	OFF	OFF
2	OFF	ON	OFF	OFF
3	OFF	OFF	ON	OFF
4	OFF	OFF	OFF	ON

### TWO-PHASE DRIVE SEQUENCE

Half Step = L, One Phase = L				
Step	A	B	C	D
POR	ON	OFF	OFF	ON
1	ON	OFF	OFF	ON
2	ON	ON	OFF	OFF
3	OFF	ON	ON	OFF
4	OFF	OFF	ON	ON

### HALF-STEP DRIVE SEQUENCE

Half Step = H, One Phase = L				
Step	A	B	C	D
POR	ON	OFF	OFF	OFF
1	ON	OFF	OFF	OFF
2	ON	ON	OFF	OFF
3	OFF	ON	OFF	OFF
4	OFF	ON	ON	OFF
5	OFF	OFF	ON	OFF
6	OFF	OFF	ON	ON
7	OFF	OFF	OFF	ON
8	ON	OFF	OFF	ON

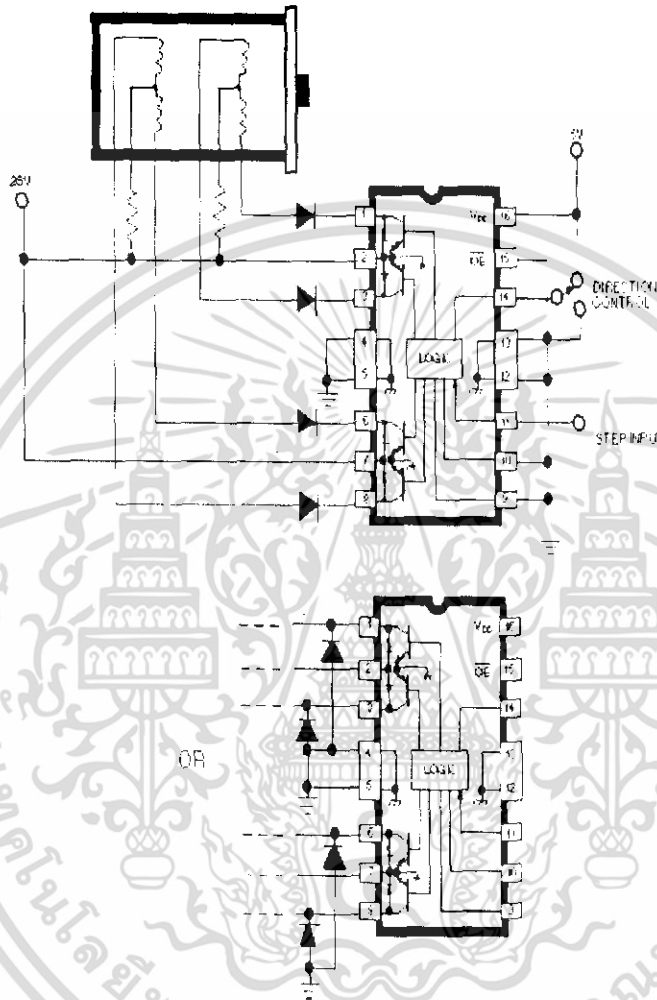


115 Northeast Cutoff, Box 15036  
Worcester, Massachusetts 01615-0036 (508) 853-5000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5804  
BiMOS II UNIPOLAR  
STEPPER-MOTOR  
TRANSLATOR/DRIVER

TYPICAL APPLICATION  
L/R Stepper-Motor Drive



The products described here are manufactured under one or more U.S. patents or U.S. patents pending.

Allegro Microsystems, Inc. reserves the right to make, from time to time, such departures from the detail specifications as may be required to permit improvements in the performance, reliability, or manufacturability of its products. Before placing an order, the user is cautioned to verify that the information being relied upon is current.

Allegro products are not authorized for use as critical components in life-support devices or systems without express written approval.

The information included herein is believed to be accurate and reliable. However, Allegro Microsystems, Inc. assumes no responsibility for its use, nor for any infringement of patents or other rights of third parties which may result from its use.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**5804**  
**BIMOS II UNIPOLAR**  
**STEPPER-MOTOR**  
**TRANSLATOR/DRIVER**

**MOTOR DRIVERS**

Function	Output Ratings*		Part Number†
<b>INTEGRATED CIRCUITS FOR BRUSHLESS DC MOTORS</b>			
3-Phase Power MOSFET Controller		28 V	3933
3-Phase Power MOSFET Controller		50 V	3932
3-Phase Power MOSFET Controller		50 V	7600
2-Phase Hall-Effect Sensor/Driver	400 mA	26 V	3626
Bidirectional 3-Phase Back-EMF Controller/Driver	±600 mA	14 V	8906
2-Phase Hall-Effect Sensor/Driver	900 mA	14 V	3625
3-Phase Back-EMF Controller/Driver	±900 mA	14 V	8902 A
3-Phase Controller/Drivers	±2.0 A	45 V	2936 & 2936-120
<b>INTEGRATED BRIDGE DRIVERS FOR DC AND BIPOLAR STEPPER MOTORS</b>			
Dual Full Bridge with Protection & Diagnostics	±500 mA	30 V	3976
PWM Current-Controlled Dual Full Bridge	±650 mA	30 V	3966
PWM Current-Controlled Dual Full Bridge	±650 mA	30 V	3968
PWM Current-Controlled Dual Full Bridge	±750 mA	45 V	2916
PWM Current-Controlled Dual Full Bridge	±750 mA	45 V	2919
PWM Current-Controlled Dual Full Bridge	±750 mA	45 V	6219
PWM Current-Controlled Dual Full Bridge	±800 mA	33 V	3964
PWM Current-Controlled Full Bridge	±1.3 A	50 V	3953
PWM Current-Controlled Dual Full Bridge	±1.5 A	45 V	2917
PWM Current-Controlled Microstepping Full Bridge	±1.5 A	50 V	3955
PWM Current-Controlled Microstepping Full Bridge	±1.5 A	50 V	3957
PWM Current-Controlled Dual DMOS Full Bridge	±1.5 A	50 V	3972
Dual Full-Bridge Driver	±2.0 A	50 V	2998
PWM Current-Controlled Full Bridge	±2.0 A	50 V	3952
DMOS Full Bridge PWM Driver	±2.0 A	50 V	3958
Dual DMOS Full Bridge	±2.5 A	50 V	3971
<b>UNIPOLAR STEPPER MOTOR &amp; OTHER DRIVERS</b>			
Voice-Coil Motor Driver	±500 mA	6 V	8932 A
Voice-Coil Motor Driver	±800 mA	16 V	8958
Unipolar Stepper-Motor Quad Drivers	1 A	46 V	7024 & 7029
Unipolar Microstepper-Motor Quad Driver	1.2 A	46 V	7042
Unipolar Stepper-Motor Translator/Driver	1.25 A	50 V	5834
Unipolar Stepper-Motor Quad Driver	1.8 A	50 V	2540
Unipolar Stepper-Motor Quad Driver	1.8 A	50 V	2544
Unipolar Stepper-Motor Quad Driver	3 A	46 V	7026
Unipolar Microstepper-Motor Quad Driver	3 A	46 V	7044

\* Current is maximum specified test condition; voltage is maximum rating. See specification for sustaining voltage limits or over-current protection voltage limits. Negative current is defined as coming out of (sourcing) the output.

† Complete part number includes additional characters to indicate operating temperature range and package style.

Also, see 3175, 3177, 3235, and 3275 Hall-effect sensors for use with brushless dc motors.



115 Northeast Cutoff, Box 15036  
 Worcester, Massachusetts 01615-0036 (508) 853-5000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้