



ชุดเชื่อมต่อคอมพิวเตอร์  
COMPUTER INTERFACE UNIT

โดย

นายประครอง ศิริโนนรัง  
นายโกมล ดีปัญญา  
นายประนต แต่อวง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร  
ปริญญาอุตสาหกรรมศาสตรบัณฑิต  
สาขาเทคโนโลยีคอมพิวเตอร์อุตสาหกรรม  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร  
ลาดกระบัง  
ปีการศึกษา 2535

ปีการศึกษา 2535

ชุดเชื่อมต่อคอมพิวเตอร์  
COMPUTER INTERFACE UNIT

โดย

นายประครอง	ศิริโนนรัง	33162117
นายโกมล	ศิโนญา	34162203
นายประนต	แต้วาง	34162213

อาจารย์ที่ปรึกษา

อาจารย์ ภากร หตะสังกาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032723

## คำนำ

ปัจจุบันนี้คอมพิวเตอร์ได้เข้ามามีบทบาทกับตัวเรามากเพื่อให้เกิดความสะดวกสบายและมีประสิทธิภาพในการทำงาน ในการทำงานบางอย่างเราก็ต้องการความแม่นยำในการทำงาน อุปกรณ์ตัวหนึ่งที่ถือได้ว่าเป็นอุปกรณ์ต่อพ่วงภายนอกที่ทำงานได้แม่นยำมาก ที่เรียกว่า stepping motor ซึ่งถือได้ว่าเป็นอุปกรณ์ที่นิยมใช้กันมากในงานที่ต้องการความแม่นยำในการทำงานเช่น ใช้ขับหัวอ่านของ disk drive เป็นต้น ฉะนั้น projectชุดนี้จึงเป็นการศึกษาถึงการทำงานของstepping motor เพื่อที่จะนำมาประยุกต์ใช้งานต่อไป

คณะผู้จัดทำ

15 มีนาคม 2536

หัวข้อปริญญาโท นักศึกษา	ชุดเชื่อมต่อคอมพิวเตอร์ นายประครอง ศิริโนนรัง 33.162117 นายโกมล คีโนญา 34.162203 นายประนต แต้ดวง 34.162213
อาจารย์ที่ปรึกษา ระดับการศึกษา	อาจารย์ ภากร หตะสังกาศ อุตสาหกรรมศาสตร์บัณฑิตทางเทคโนโลยี การวัดคุมทางอุตสาหกรรม
ปีการศึกษา	2535

บทคัดย่อ

ปริญญาโทเป็นส่วนหนึ่งของการศึกษาในระดับปริญญาตรี ปริญญาโทเล่มนี้เป็น การนำเอาคอมพิวเตอร์มาใช้เชื่อมต่อกับ สเตปปีงมอเตอร์ ด้วยชุดเชื่อมต่อ ในส่วนของ การเชื่อมต่อจะประกอบด้วยชุด ดีโคคตำแหน่งของพอร์ต ซึ่งตำแหน่งพอร์ตที่ใช้คือ 6FOH -6F3H และใช้ IC#8255 เป็นพอร์ตการควบคุมการทำงานของ สเตปปีงมอเตอร์ จะทำ โดยการส่งข้อมูลซึ่งจะเป็นการกำหนดทิศทางการหมุน รูปแบบการกระตุ้นเฟสการทำงาน ของสเตปปีงมอเตอร์ ข้อมูลจะถูกส่งผ่าน IC#8255 สู่วงจรขับของสเตปปีงมอเตอร์ ต่อไป รายละเอียดเกี่ยวกับการทดลอง สเตปปีงมอเตอร์ มีในชุดคำถามที่ให้มากับปริญญา- โทฉบับนี้ เพื่อประกอบในการทดลองพร้อมกันด้วย

Thesis Title : COMPUTER INTERFACE UNIT  
Name: Mr.Prakrong Srinonrung 33.162117  
Mr.Komol Deepunya 34.162203  
Mr.Pranot Tae-oung 34.162213  
Thesis Advisor:  
Level of study: Bachelor's Degree of Computer  
Technology Instrument Department  
Academic Year: 1992

### Abstract

Thesis are important for engineering graduate in university. This thesis has been used a personal computer to interface with stepping motor by Interface Unit. Address 6F0H - 6F3H of PC are port's address, and IC#8255 is port to be used for sent control data of stepping motor. Data include of data for selected direction of rotation, and data for selected phase excitation. All data be sent to driver unit and drive stepping motor. Description of the exercise about stepping motor to be used with stepping motor are in exercise paper.

## บทนำ

ปัจจุบันนี้คอมพิวเตอร์ได้มีบทบาทมากในวงการอุตสาหกรรม จะพบได้ว่างานคอนโทรลระบบต่างๆ นิยมใช้คอมพิวเตอร์มาเชื่อมต่อเพื่อความสะดวกในการทำงาน เป็นต้นว่า การควบคุมอุณหภูมิ, ระบบควบคุมด้วยตัวเลข (CNC), ระบบการควบคุมโรงงานอัตโนมัติ, และโดยเฉพาะอย่างยิ่งระบบการควบคุมหุ่นยนต์ ซึ่งระบบการควบคุมเหล่านี้จำเป็นต้องใช้อุปกรณ์ MOTOR ในการหมุนหรือเคลื่อนที่ ในบางครั้งเราจะพบว่ามีความยุ่งยากในการใช้ MOTOR ธรรมดาเข้ามาใช้งานและนำมาต่อเชื่อมกับคอมพิวเตอร์ เนื่องจากวงจรที่ใช้เชื่อมต่อก่อนข้างจะซับซ้อนและทำงานไม่ค่อยเที่ยงตรง ดังนั้นในงานที่ต้องการความละเอียดและเที่ยงตรงจึงนิยมใช้ Stepping Motor มาทำงานดังกล่าว เนื่องจากการเชื่อมต่อกับคอมพิวเตอร์ทำได้สะดวกและผิดพลาดน้อยกว่า

### วัตถุประสงค์ของปริิณยานิพนธ์

- ศึกษาถึงชนิดของ Stepping Motor
- ศึกษาถึงโครงสร้างและการทำงานของ Stepping Motor
- ศึกษาถึงการควบคุม Stepping Motor ด้วยวิธีต่างๆ
- นำข้อมูลต่างๆ เพื่ออ้างอิงในการทดลอง Stepping Motor ต่อไป

### ขอบเขตของปริิณยานิพนธ์

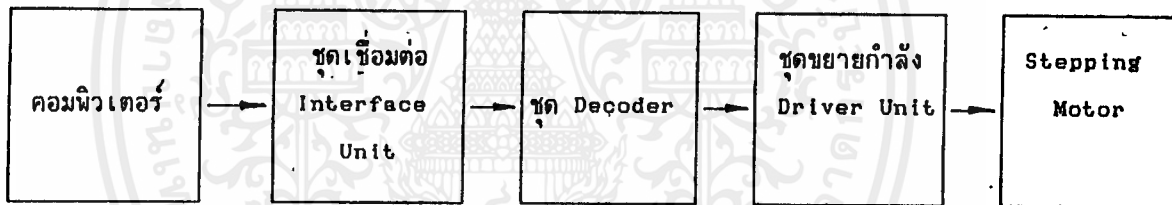
ปริิณยานิพนธ์นี้เป็นส่วนหนึ่งของการทดลองชุดเชื่อมต่อกับคอมพิวเตอร์ (Computer Interface Unit) ซึ่งจะประกอบด้วยกัน 3 กลุ่ม ในส่วนของปริิณยานิพนธ์นี้เป็นการทดลองเกี่ยวกับ Stepping Motor วิชาใช้การเชื่อมต่อจากระบบ BUS ของ Computer ทั่วไปมาเชื่อมต่อกับชุดคอนโทรลเลอร์ และไดรฟ์เวอร์ ของ Stepping Motor วิชาใช้โปรแกรมภาษา C และภาษาปาสคาล เพื่อให้ผู้ใช้ทดลองการกระตุ้นเฟส Stepping Motor ในแบบต่างๆ ต่อไป

## สารบัญ

	หน้า
สเตปป์มอเตอร์	1
ประโยชน์ของสเตปป์มอเตอร์	1
โหมคการทำงาน	2
การกระตุ้นเฟส	5
ชนิดของ STEPPING MOTOR	14
การควบคุม	23
วงจรรีเคเวนเซียลลอจิก	25
วงจรรีบ	26
การแก้ไขการเพิ่มขึ้นของกระแส	38
การทำงานของวงจรรีบทดลอง	40
โปรแกรม	47

## สเตปป์มอเตอร (Stepping motor)

สเตปป์มอเตอร เป็นอุปกรณ์ จำพวกเชิงกลทางไฟฟ้าที่มีอินพุทเป็นกลุ่มของไบนารีโลเตจและเอาต์พุท เป็นลักษณะของการเคลื่อนที่แบบเชิงมุม หรือหมุนไปตามสเตป (แต่ละสเตปอยู่ในช่วง 0.1 ถึง 90 องศา ขึ้นอยู่กับโครงสร้างของสเตปป์มอเตอร) ตามสัญญาณพัลซที่ป้อนให้กับขดสเตเตอร จึงจะเกิดแรงผลักรถโรเตอรทำให้มอเตอรหมุนไป แต่ลักษณะของสเตปป์มอเตอร จะมีขดของสเตเตอรอยู่หลายขดซึ่งเรียกว่า “เฟส” นั้นเมื่อป้อนสัญญาณเชิงเป็นพัลซ ในลักษณะซีควนของเลขไบนารีโดยผ่านวงจรไดรเวอร์ (Driver) จะทำให้โรเตอรหมุนได้อย่างต่อเนื่องกันบล็อคไดอะแกรมรูปที่ 1



รูปที่ 1 บล็อคไดอะแกรมแสดงการควบคุมสเตปป์มอเตอร

### คุณประโยชน์ของสเตปป์มอเตอร

ในระบบควบคุมตำแหน่งที่ใช้สเตปป์มอเตอร นั้น มีข้อดีอยู่หลายประการคือ

1. เป็นลักษณะการควบคุมแบบไม่ต้องการการป้อนกลับไม่ว่าจะเป็นการควบคุมตำแหน่งหรือความเร็ว
2. ความผิดพลาดเกี่ยวกับตำแหน่งแทบไม่มีเลยเนื่องจากการเคลื่อนที่ของสเตปป์มอเตอร นั้นเคลื่อนที่เป็นสเตปด้วยจำนวนองศาที่มีค่าแน่นอน
3. สเตปป์มอเตอร จะถูกนำมาใช้กับเครื่องมือที่ต้องการความละเอียดแม่นยำและใช้อยู่ในเครื่องมือประเภทดิจิทัล เช่น เครื่องวาดรูป เครื่องคอมพิวเตอร์ นิวเมอริคอลลคอนโทรล (Computer Numerical Control) หรือ CNC

4. ไม่จำเป็นต้องใช้วงจรแปลงดิจิทัลเป็นอนาลอกเมื่ออินเตอร์เฟสกับไมโครคอมพิวเตอร์

5. สเตปป์มอเตอร์ เป็นอุปกรณ์ที่เปลี่ยนสัญญาณดิจิทัลไปเป็นการเคลื่อนที่ทางกล ดังนั้นการติดต่อกับอุปกรณ์ดิจิทัลก็เป็นไปได้โดยง่าย และวงจรขยายกำลังจากสัญญาณดิจิทัล (digital power amplifier) ที่ใช้ก็มีราคาถูกกว่าวงจรขยายกำลังเชิงเส้น (linear power amplifier) อีกด้วย

6. การออกแบบวงจรควบคุมสเตปป์มอเตอร์สามารถทำได้ง่ายกว่าวงจรควบคุมมอเตอร์แบบเซอร์โว และยังสามารถออกแบบวงจรในสเตปป์มอเตอร์ทำงานหรือหยุดได้แบบทันทีทันใด

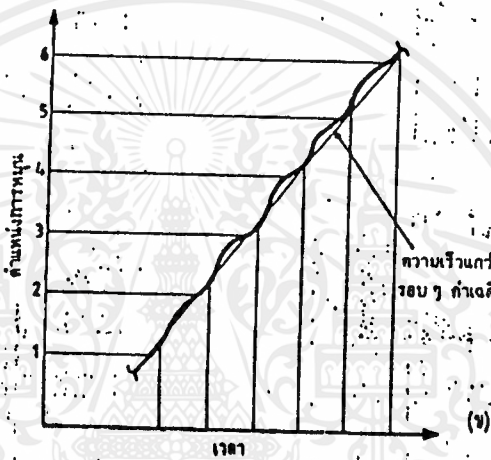
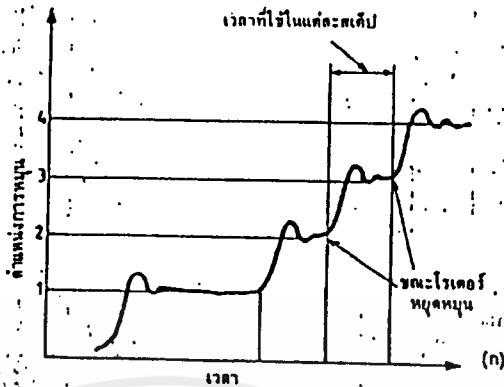
### โหมดการทำงานของ STEPPING MOTORS

ถ้าจะแบ่งโหมดการทำงานของ Stepping Motor ตามอัตราเร็วของ Step จะแบ่งออกได้เป็น 2 โหมด คือ หมุนเป็นสเตป (Discrete Stepping Mode) และหมุนแบบต่อเนื่อง (Slewing mode)

โดยถ้าการหมุนเป็นแบบสเตป และมีเวลาหยุดนิ่งก่อนที่จะเปลี่ยนเป็นสเตปต่อไป ก็จะเรียกการทำงานในโหมดนี้ว่าการหมุนเป็น Step ดังรูป.2.(ก) สำหรับตัวอย่างของเครื่องใช้ที่ทำงานในโหมดนี้คือ เครื่องเจาะบัตร การทำงานต่าง ๆ ก็คือ Stepping Motor จะเป็นตัวส่งแถบกระดาษเข้าไปยังเครื่องปฏุกกระดาษ เพื่อบันทึกข้อมูลลงในแถบกระดาษ ซึ่งการหมุนของ Stepping Motor จะหมุนไปแล้วหยุดชั่วขณะเพื่อปฏุกกระดาษให้เรียบร้อยก่อน แล้วจึงค่อยหมุนต่อไปยังตำแหน่งเจาะใหญ่

ถ้าเพิ่มอัตราเร็วในแต่ละ Step ให้เร็วขึ้น และเป็นไปอย่างต่อเนื่อง ไม่มีการหยุดนิ่ง จะเรียกกำหนดการทำงานนี้ว่า การหมุนแบบต่อเนื่อง ดังแสดงในรูป.2.(ข) ซึ่งสามารถหาความสัมพันธ์ระหว่างความเร็วรอบของมอเตอร์ (n) กับอัตราเร็วของ Step (f) และจำนวนสเตปทั้งหมด (s) ได้ดังสมการ

$$N = 60 \frac{f}{s}$$



รูป 2 กราฟแสดงสแต็ปของการหมุนในโหมดการทำงาน

(ก) หมุนแบบ Step

(ข) หมุนต่อเนื่อง

### กราฟคุณลักษณะของ STEPPING MOTOR

กราฟคุณลักษณะของ Stepping Motor จะเป็นการแสดงความสัมพันธ์ระหว่างอัตราเร็วของ Step และแรงบิดที่แสดงในรูปสำหรับกราฟเส้นประ เรียกว่า Start-step curve หรือ single-step load curve เป็นกราฟที่อยู่ในโหมดการหมุนเป็น Step และเป็นกราฟที่แสดงถึงย่านของแรงบิดที่มอเตอร์สามารถเริ่ม และหยุดหมุนได้ โดยปราศจากความผิดพลาดแม้ที่อัตราเร็วของ Step ต่าง ๆ กัน และกราฟอีกเส้นคือ slew curve ซึ่งทำงานอยู่ในโหมดการหมุน

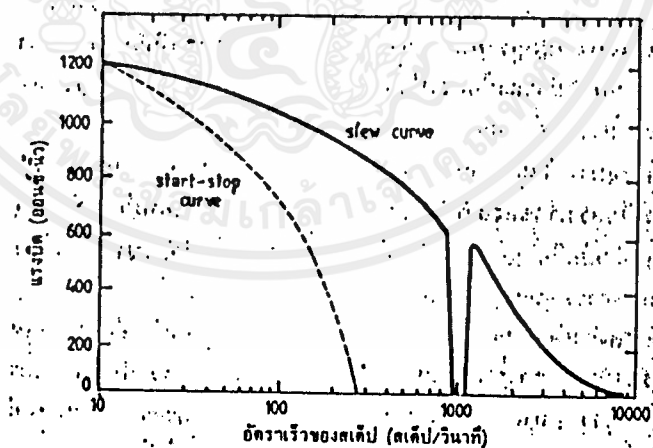
แบบต่อเนื่อง จะเป็นกราฟที่แสดงถึงค่าแรงบิดสูงสุดที่ Stepping Motor สามารถจะกระทำได้ที่อัตราเร็วของ Step ต่าง ๆ กันถ้ามีการใช้งาน Stepping Motor โหมดกราฟนี้ก็อาจทำให้เกิดความผิดพลาดได้ในทางตรงข้าม ถ้าใช้งานอยู่ภายใต้กราฟนี้แม้จะควบคุมแบบระบบเปิดก็มั่นใจได้ว่าทั้งตำแหน่งและความเร็วมีความเที่ยงตรงแน่นอนโดยตำแหน่งของ Motor สามารถคำนวณได้จากสมการ

$$\text{มมที่เปลี่ยนไป} = \text{มมใน 1 Step} \times \text{จำนวนพัลส์ที่ป้อนให้}$$

ส่วนความเร็วคำนวณหาได้จากสมการ

$$n = 60 \frac{f}{s}$$

สำหรับช่วงที่เส้นกราฟขาดหายไปของ slew curve นั้นเป็นย่านเรโซแนนซ์ (Resonant region) ซึ่งเป็นจุดอ่อนสำคัญที่ต้องคำนึงถึงในการพิจารณาใช้งาน เพราะเป็นช่วงที่ไม่เสถียร และควบคุมไม่ได้



รูปที่ 3 กราฟแสดงความสัมพันธ์ระหว่างอัตราเร็วของ Step กับแรงบิดของการทำงานทั้ง 2 โหมดของ Stepping Motor

### โหมดของการกระตุ้นเฟส (MODES OF EXCITATION)

การใช้งาน Stepping Motor อย่างเป็นต่อเนื่อง จำเป็นต้องจ่ายพัลส์เป็นสำคัญอย่างต่อเนื่อง วิธีการกระตุ้นเฟสมีหลายอย่างดังนี้

#### 1. การกระตุ้นแบบเฟสเดียว (Single-phase excitation)

ตาราง...1 แสดงลำดับของการกระตุ้นแบบเฟสเดียวสำหรับ MOTOR แบบ 3 และ 4 เฟสจากตารางเมื่อต้องการให้มอเตอร์หมุนตามเข็มนาฬิกา สามารถทำได้โดยกระตุ้นเฟส ph1-ph2-ph3 ตามลำดับในทางกลับกันถ้าต้องการหมุนเข็มนาฬิกากระตุ้น ph3-ph2-ph1 ตามลำดับเช่นเดียวกับการกระตุ้น ลักษณะนี้เป็นลักษณะ "ONE PHASE ONE DRIVE"

Excitation sequence in the single-phase-on operation.

##### (1) Three-phase motor

	R	1	2	3	4	5	6	7	8
Phase 1	█			█			█		
Phase 2		█			█			█	
Phase 3			█			█			█

##### (2) Four-phase motor

	R	1	2	3	4	5	6	7	8
Phase 1	█				█				█
Phase 2		█				█			
Phase 3			█				█		
Phase 4				█				█	

Pulses 

--	--	--	--	--	--	--	--	--	--

Note: Symbol R indicates 'reset'.

ตารางรูป 1 แสดงลำดับการกระตุ้นแบบเฟสเดียว

2. การกระตุ้นแบบเฟสคู่ (Two-phase excitation)

ตารางที่....1 แสดงลำดับการกระตุ้นแบบเฟสคู่สำหรับ Motor แบบ 3 และ 4 เฟส จากตารางจะพบว่าเมื่อ ph2 off และ ph1 on จะพบว่า ph3 ยังคงถูกกระตุ้นอยู่ ลักษณะของการกระตุ้นแบบนี้เป็นดังรูป 4

Excitation sequence in the two-phase-on operation.

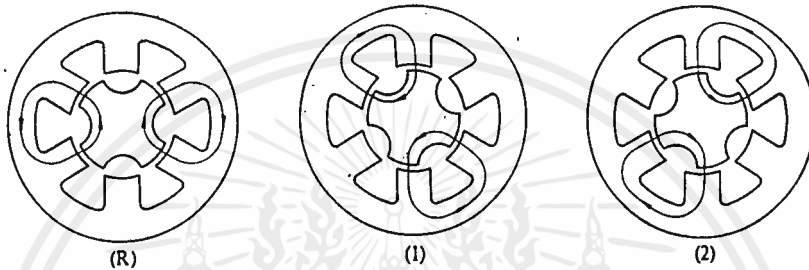
(1) Three-phase motor

Clock state	R	1	2	3	4	5	6	7	8
Phase 1	■	■	→	■	■	■	■	■	■
Phase 2	■	■	→	■	■	■	■	■	■
Phase 3	■	■	■	■	■	■	■	■	■

(2) Four-phase motor

Clock state	R	1	2	3	4	5	6	7	8
Phase 1	■	■	■	■	■	■	■	■	■
Phase 2	■	■	■	■	■	■	■	■	■
Phase 3	■	■	■	■	■	■	■	■	■
Phase 4	■	■	■	■	■	■	■	■	■

ตาราง 2 แสดงลำดับการกระตุ้นแบบเฟสคู่



Change in field pattern as a step proceeds in the two-phase-on mode in a monofilar three-phase VR motor.

#### รูปที่ 4 ลักษณะของการเปลี่ยนเฟสของการกระตุ้นแบบเฟสคู่

#### 3. การกระตุ้นแบบกึ่งสเตป (Half-step mode)

เป็นการรวมรูปแบบการกระตุ้น ระหว่างการกระตุ้นแบบเฟสเดี่ยว และเฟสคู่ ตารางที่...3 เป็นการแสดงลำดับการกระตุ้นแบบกึ่งสเตป การกระตุ้น จะต้องจ่ายสัญญาณนาฬิกา 2 State คือ A ,B โดยที่ A จะจ่ายลักษณะกระตุ้นเฟส เดี่ยว และ B เป็นการจ่าย Clock ในลักษณะกระตุ้นแบบเฟสคู่

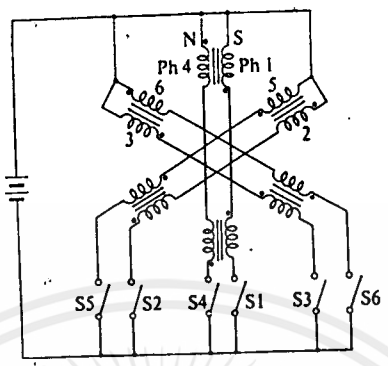
Excitation sequence in the half-step operation (three-phase motor).

Clock state (A)	R	1	2	3	4	5				
Clock state (B)	R	1	2	3	4	5	6	7	8	9
Phase 1	■	■			■	■	■			
Phase 2		■	■	■			■	■	■	
Phase 3				■	■	■			■	■

ตาราง 3 แสดงลำดับการกระตุ้นแบบกึ่ง Step

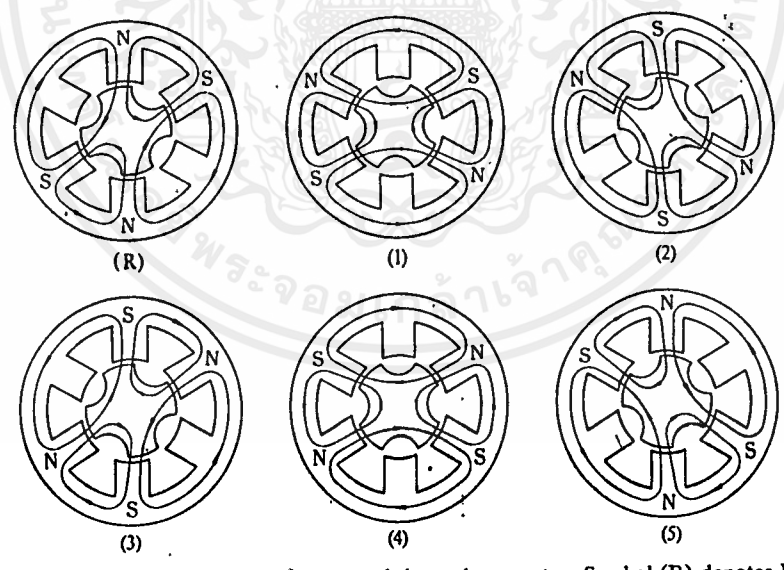
#### 4. Two-phase-on drive of bifilar-wound three phase VR motor

ความต้องการสำคัญในการออกแบบ Motor คือ ต้องการให้ขนาดของเครื่องจักรเล็ก และถ้าเป็นไปได้จะต้องมีความสามารถทางานพิเศษ ดังนั้นจึงทำให้เกิดการกระตุ้นแบบนี้ขึ้น รูป 5 เป็นลักษณะโคอะแกรมของ bifilar wound coil ใน VR Motor 3 เฟสในรูป 6 แสดงลักษณะของฟลักซ์แม่เหล็กเมื่อมีการกระตุ้น ตามตาราง 4 ข้อดีของการกระตุ้นแบบนี้คือ แรงบิดของเครื่องจักรจะดีมาก



Schematic diagram of the bifilar-wound coils in the three-phase VR motor.

รูป 5 ลักษณะโคจรแรงแมของ bifilar wound coil ใน VR Motor 3 เฟส



Field pattern in the bifilar-wound three-phase motor. Symbol (R) denotes Reset in the table in Fig. 5.31 (p. 141).

รูป 6 แสดงลักษณะของฟลักซ์แม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

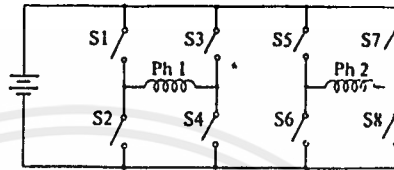
Excitation sequence in the two-phase-on operation for a bifilar-wound three-phase VR motor.

Clock state	R	1	2	3	4	5	6	7	8
Ph1 (S1)	■					■	■		
2 (S2)	■	■					■	■	
3 (S3)		■	■					■	■
4 (S4)			■	■					■
5 (S5)				■	■				
6 (S6)					■	■			

ตาราง 4 แสดงลำดับการกระตุ้น

### 5. การกระตุ้นด้วยวงจรวจรบริจ

วงจรวจรบริจแสดงในรูป 7 เป็นวงจรวจรสำหรับขั้วมอเตอร์ใน bipolar mode ลักษณะการกระตุ้นวงจรวจรตามลำดับในตาราง 5 จากตารางและวงจรวจรที่พบในขดลวดแต่ละขด จะนำกระแสสลับอย่างสม่ำเสมอ ซึ่งลักษณะนี้จะปรับปรุง 25-30%ของแรงบิดต่อหน่วยของ I/P ซึ่งเป็นไปได้เมื่อใช้กับการไครฟ์ bipolar ลักษณะของ Voltage ที่ป้อนให้ วงจรวจร Bipolar drive แสดงในรูป 8



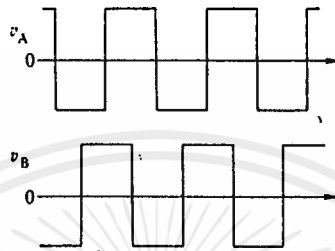
Bridge driver scheme for a two-phase stepping motor.

รูป 7 วงจร Bridge drive สำหรับ Stepping Motor 2 เฟส

Excitation sequence in the bridge operation for a two-phase motor.

Clock state	R	1	2	3	4	5	6	7	8
S1	■	■			■	■			■
S2			■	■			■	■	
S3			■	■			■	■	
S4	■	■			■	■			■
S5		■	■			■	■		
S6	■			■	■			■	■
S7	■			■	■			■	■
S8		■	■			■	■		

ตาราง 5 แสดงลำดับการกระตุ้นสำหรับวงจรบริดจ์



Square-wave voltage for the bipolar drive.

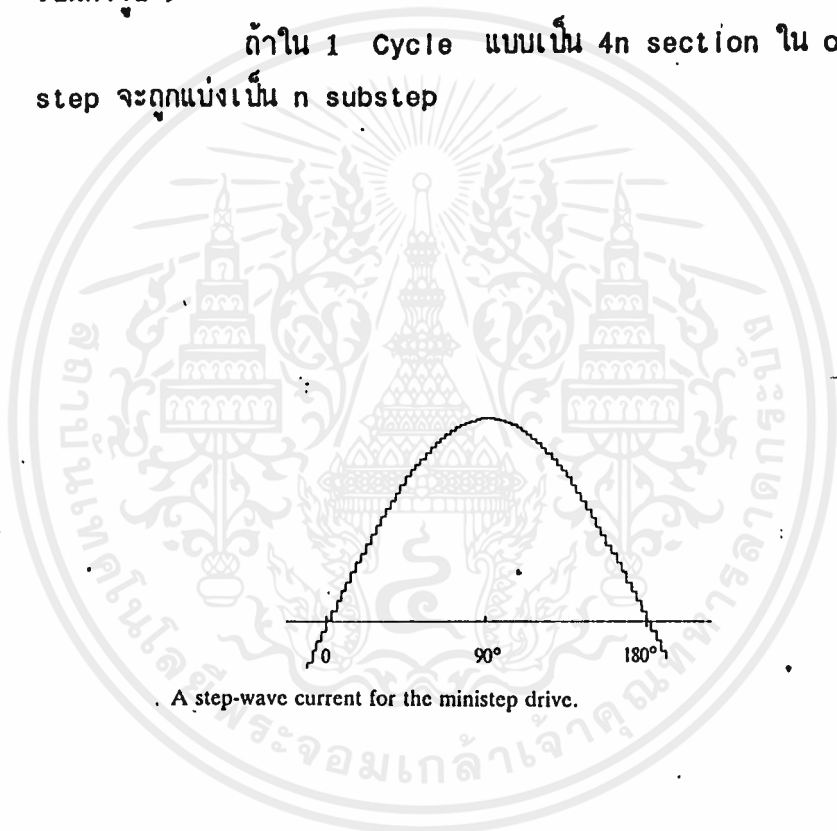
รูป 8 Square-wave Voltage สำหรับวงจรไดรฟ์ Bipolar

### 6. Ministep drive

มันเป็นไปได้ที่จะจัดให้สเต็ปธรรมชาติ 1 สเต็ป ในรูปของสเต็ปเล็ก ๆ หลาย ๆ สเต็ป ในความหมายในทางอิเล็กทรอนิกส์ ซึ่งวิธีการนี้เรียกว่า ministep หรือ Microstep drive ซึ่งจะพบว่าถูกใช้ใน hybrid stepping Motor บ่อย ๆ แนวความคิดนี้มาจาก sinusoidal bipolar drive ของ hybrid motor ซึ่งคล้ายการ Synchronous motor

การจัด Supply current สำหรับ mini step drive จะเป็นดังรูป 9

ถ้าใน 1 Cycle แบบเป็น  $4n$  section ใน one natural step จะถูกแบ่งเป็น  $n$  substep



A step-wave current for the ministep drive.

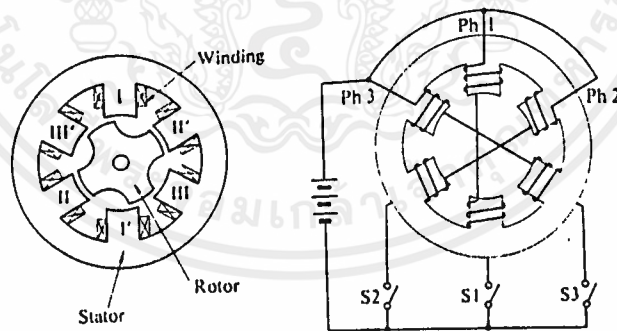
รูป 9 A step wave current สำหรับ ministep drive

### ชนิดของ STEPPING MOTORS

Stepping Motors สามารถแบ่งออกได้หลายอย่างขึ้นอยู่กับโครงสร้างของเครื่องจักรที่นำไปใช้งานและลักษณะการออกแบบใช้งาน

#### 1. VR Motor

Variable-reluctance stepping motor หรือ VR มอเตอร์ เป็น stepping motor ที่มีลักษณะโครงสร้างและการใช้งานที่เป็นพื้นฐานของแบบอื่น ๆ ในรูปที่ 10 แสดงภาพตัดของ VR motor แบบ 3 เฟส โดย stator มี 6 ฟัน ฟันที่อยู่ตรงข้ามกันจะมีความต่างมุม 180 องศา แต่ละฟันจะถูกฟันค้ำยขดลวด และขดตรงข้ามจะถูกค้ำกันแบบอนุกรมหรือขนาน (ในรูปคือแบบอนุกรม) ส่วน Rotor มี 4 ฟัน ทั้ง Stator และ Rotor ทำมาจากเหล็กผสมซิลิกอน ซึ่งเป็นวัสดุที่มีความซึมซับสูง (high permeability) สามารถให้เส้นแรงแม่เหล็กไหลผ่านได้มาก ในการจัดขั้วของแบบตรงข้ามกันจะใช้เป็นฟัน I, II, และ III เป็นขั้วเหนือ และ I', II' และ III' เป็นขั้วใต้



รูปที่ 10 ภาพหน้าตัดและการพันขดลวดของ VR Motor แบบ 3 เฟส

รูปที่ 10 ภาพหน้าตัดและการพันขดลวดของ VR Motor แบบ 3 เฟส

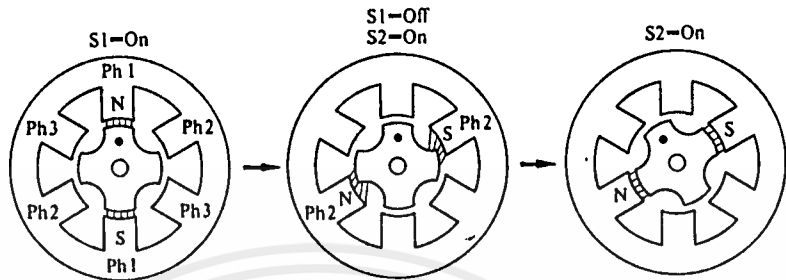
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานจะเริ่มกระตุ้นที่เฟสใดเฟสหนึ่งดังรูปที่ 11 เป็นการกระตุ้นเฟส I ก่อน ( $S_1$  "ON") ถ้าโรเตอร์จะพยายามวางตำแหน่งตัวเลขให้อยู่ในทิศทางที่ทำให้ค่าความต้านทานแม่เหล็กน้อยที่สุด ในขั้น I-I'



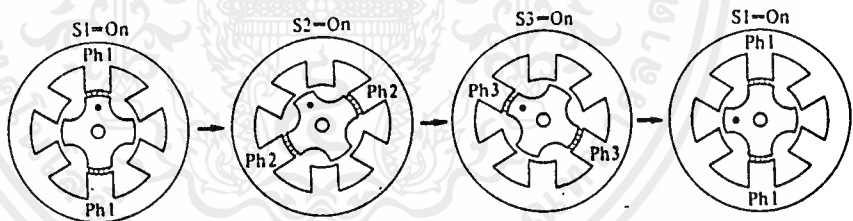
รูปที่ 11 แสดงเส้นแรงแม่เหล็กขณะกระตุ้นเฟส I

ในขณะที่เริ่มต้นที่จะกระตุ้นที่เฟส II ( $S_1$  "OFF",  $S_2$  "ON") ดังรูปที่ 12 เส้นแรงแม่เหล็กจะไม่อยู่ในแนวทางที่เค้นสะดวกจึงทำให้ค่าความต้านทานทางแม่เหล็กมีค่าสูง ตัวโรเตอร์ก็จะพยายามปรับตัวเอง เพื่อให้ค่าความต้านทานน้อยที่สุด ด้วยการใช้การหมุนในทิศทางทวนเข็มนาฬิกา ซึ่งแรงบิดที่ใช้หมุนเกิดจากแรงของเส้นแรงแม่เหล็ก แล้วจะไปหยุดที่ตำแหน่งความต้านทานแม่เหล็กน้อยที่สุด นั่นคือทำให้ได้ 1 step ซึ่งจากรูปแต่ละ step ห่างกัน 30 องศาในการหมุนจนครบสมบูรณ์ทั้ง 3 เฟส แสดงดังรูปที่ 13



How a step motion proceeds when excitation is switched from Ph1 to Ph2.

รูปที่ 12 แสดงขั้นตอนการหมุนเมื่อกระตุ้นเฟส 1 ไปเฟส 2



Step motions as switching sequence proceeds in a three-phase VR motor.

รูปที่ 13 แสดงการกระตุ้นเฟสจนครบสมบูรณ์

ชนิดของ VR MOTOR

1.1 Air-gap คือ แบบที่ช่องว่างระหว่าง stator กับ rotor เป็นอากาศใช้กับงานที่มี step ในการทำงานไม่มากและต้องการแรงหมุนมาก

1.2 Smaller step angles เป็นแบบที่มี step ในการทำงานที่ละเอียดจะพบว่ามีฟันของ stator มากเพื่อทำให้ step แต่ละ step มีมุมเล็กลง ซึ่งจะมีสูตรในการคำนวณดังนี้

$$S = 360/\theta_s$$

$$S = mN_r$$

เมื่อ

S = จำนวนของ step

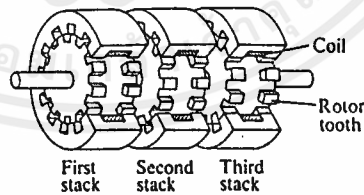
M = จำนวนของเฟส

$N_r$  = จำนวนฟันของ rotor

$\theta_s$  = มุมระหว่าง step

1.3 แบบ Multi-stack

ทั้ง 2 แบบที่กล่าวมาเป็นแบบ single stack แต่ที่จะกล่าวถึงนี้เป็นแบบ Multi stack ซึ่งจะมีลักษณะดังรูปที่ 14



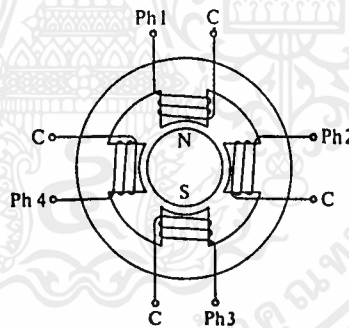
Construction of a multi-stack VR motor.

รูปที่ 14 โครงสร้างของ VR Motor แบบ Multi stack

ลักษณะการนำไปใช้งานเช่น ใน stack 1 เรากระตุ้นเฟสให้หมุนตามเข็มนาฬิกา ซึ่งมุมทั้ง stack 1,2 และ 3 ก็จะหมุนตามกัน แต่เมื่อเรากระตุ้น stack 2 เราใช้การกระตุ้นในแบบวนเข็มนาฬิกา ซึ่งจะทำให้เกิดการหมุนเปลี่ยนไปงานที่นำไปใช้เช่น Numerical Controlled Machines เป็นต้น

## 2. PM STEPPING MOTORS

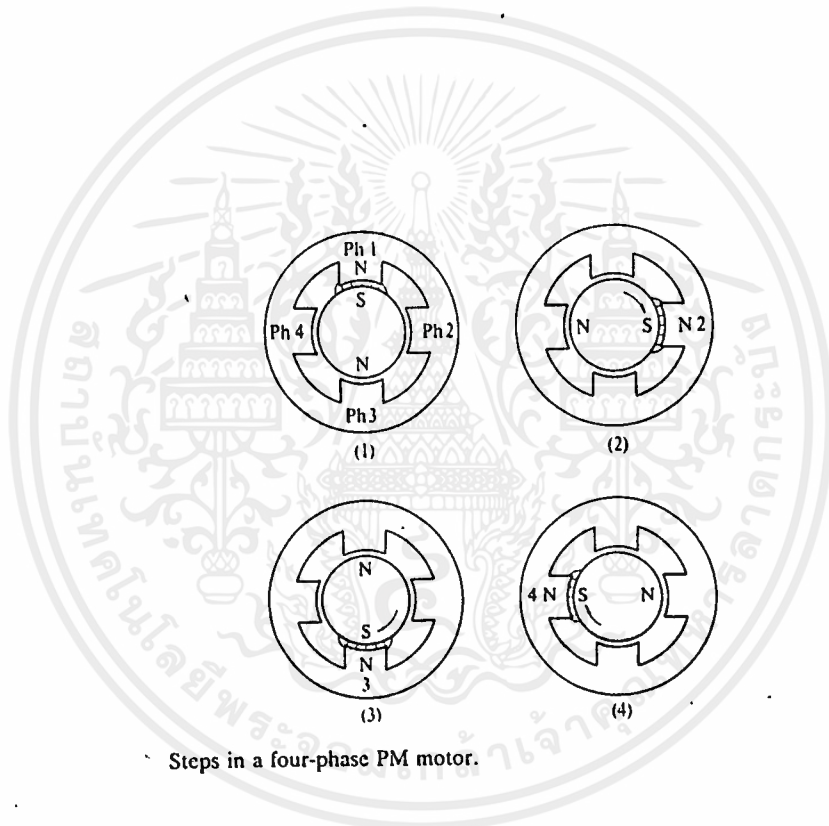
เป็น stepping motors ที่ใช้แม่เหล็กถาวรเป็น rotor (Permanent Magnet Motor) ดังรูปที่ 15 การพันขดลวดที่ stator จะเป็นลักษณะที่แต่ละฟันจะมีขดลวดพันอยู่ ซึ่งถือว่าฟันหนึ่งคือ 1 เฟส



Cross-sectional model of a four-phase PM motor.

รูปที่ 15 ภาพตัดแสดง PM Motor แบบ 4 เฟส

การกระตุ้นเฟสของ PM Motor ทำได้ง่าย ๆ คือ ต่อจุด C (Common) ที่บวกของ supply เมื่อมีการกระตุ้นที่เฟสใด เฟสนั้นก็จะกลายเป็นขั้วเหนือ  
ในรูปที่ 16 แสดงการกระตุ้นเฟส 1-2-3-4 ตามลำดับ และการหมุนของโรเตอร์ในแต่ละ step ซึ่งแต่ละ step มีมุมเท่ากับ 90 องศา ในทิศทางตามเข็มนาฬิกา



Steps in a four-phase PM motor.

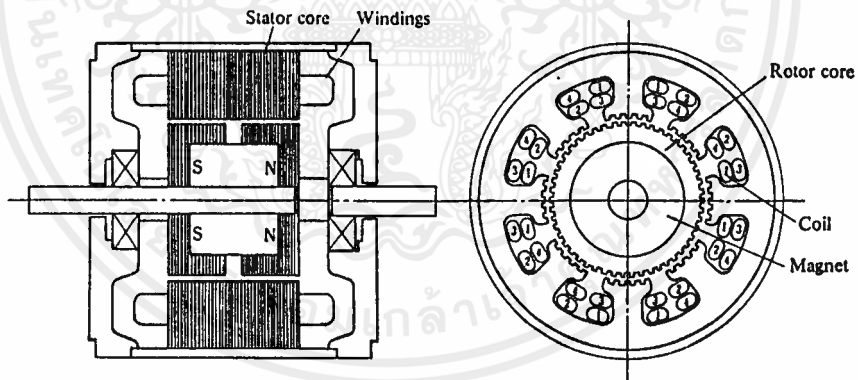
### รูปที่ 16 step ในการหมุนของ PM Motor แบบ 4 เฟส

ถ้าต้องการให้ step ในการหมุนมากขึ้นทำได้โดยการเพิ่มจำนวนฟันของ stator และขั้วบน Rotor ข้อเสียของ PM Motor คือข้อที่ PM Motor มีราคาค่อนข้างแพง ข้อสอง ความหนาแน่นของเส้นแรงแม่เหล็กถูกจำกัดด้วยระดับของเส้นแรงแม่เหล็กภายใน (Magnetic remanence) ของแม่เหล็ก

### 3. HYBRID STEPPING MOTORS

เป็น stepping motor ที่มีแกน rotor เป็นแม่เหล็กถาวร แต่มีลักษณะการทำงานเหมือนกับ VR Motor ในรูป 17 เป็นภาพหน้าตัดโครงสร้างของ hybrid motor

ข้อแตกต่างจาก VR Motor คือใน VR Motor จะมีขั้วลวดบนขั้วของ stator เพียง 1 ขั้วแต่ใน hybrid จะมีขั้วลวด 2 ขั้วในขั้ว stator เดียว

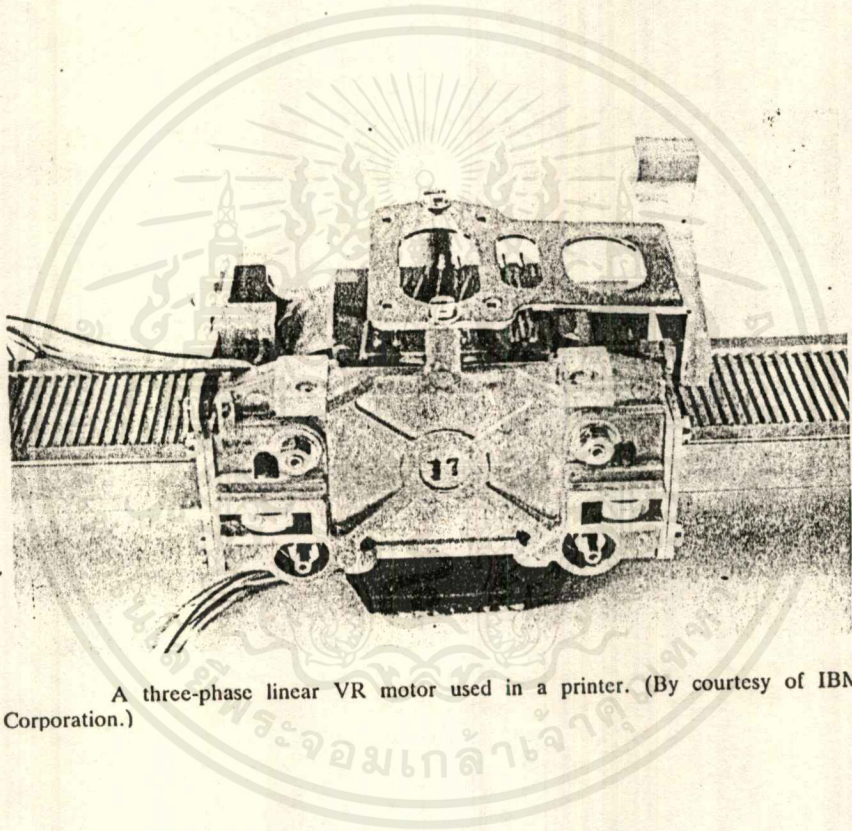


Construction of a hybrid motor.

รูปที่ 17 โครงสร้างของ hybrid motor

#### 4. LINEAR STEPPING MOTOR

เป็นลักษณะของ stepping motor ที่ทำงานอย่างคงที่ ส่วนใหญ่ จะใช้กับลักษณะงาน control เช่น VR Motor ใช้ใน printer เป็นต้น

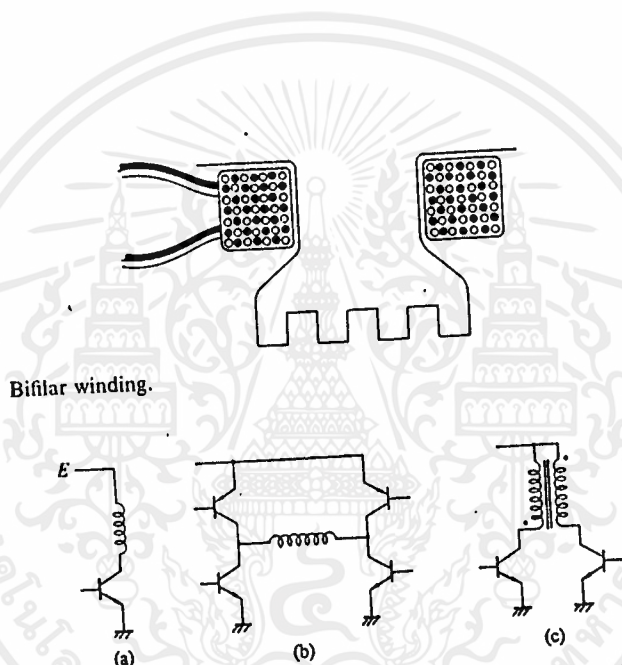


A three-phase linear VR motor used in a printer. (By courtesy of IBM Corporation.)

รูปที่ 18 VR Motor แบบ 3 เฟส ใช้ใน Printer

### 5. MONOFILAR AND BIFILAR WINDING

รูป 19 แสดงการพันขดลวดแบบ bifilar และ monofilar ลักษณะของ Motor แบบนี้จะต่างที่การพันขดลวดเช่นแบบ bifilar เป็นการพันขดลวด 2 เส้นในแกนเดียวกัน

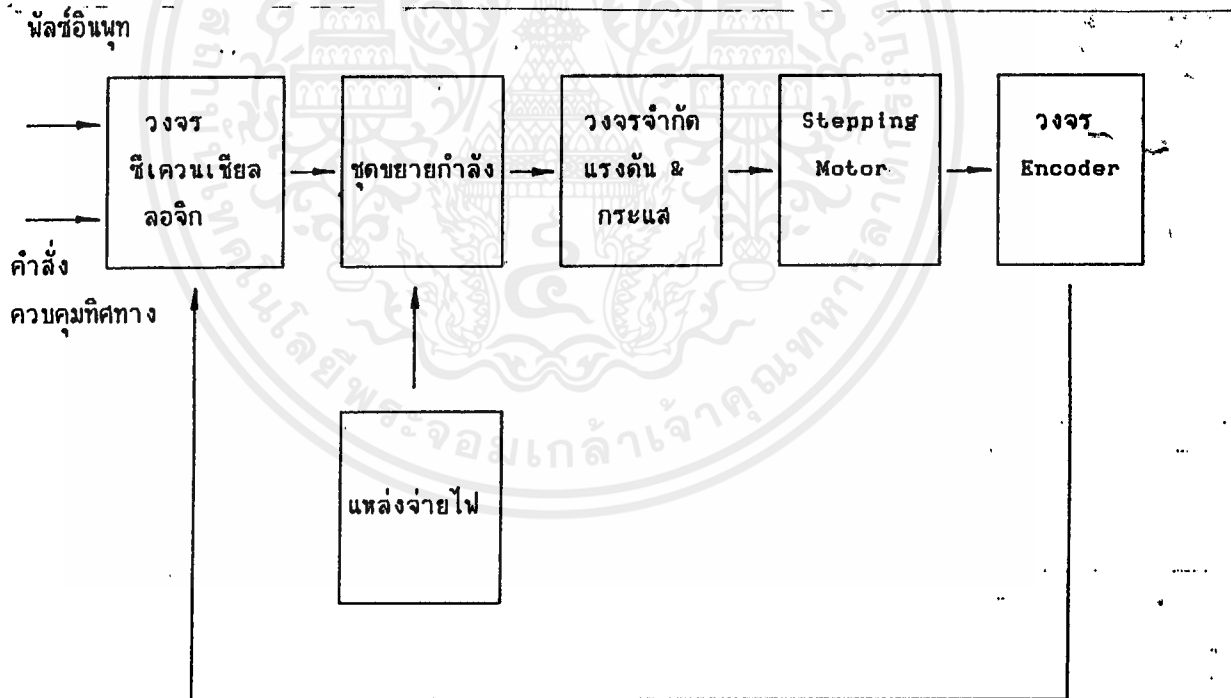


Three fundamental exciting circuits: (a) monofilar winding, unipolar excitation; (b) monofilar winding, bipolar excitation; (c) bifilar winding, bipolar excitation.

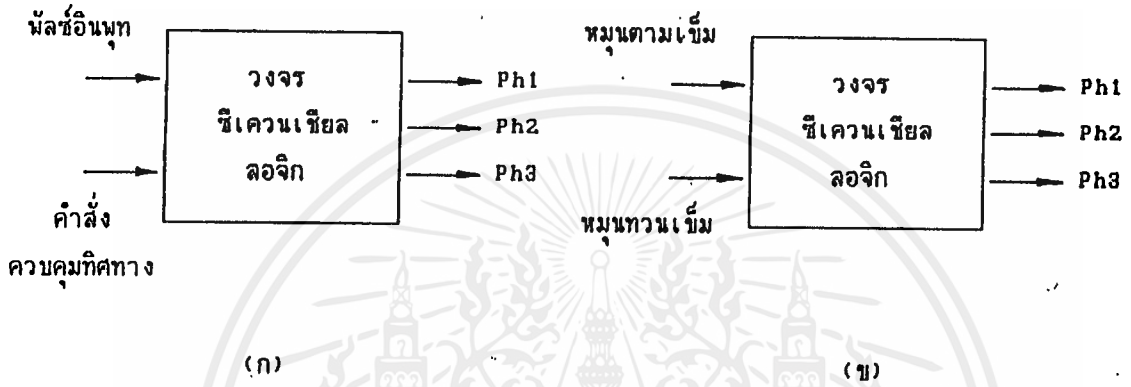
รูปที่ 19 แสดงการพันขดลวดแบบ bifilar และ monofilar

### การควบคุมสเตปป์มอเตอร์

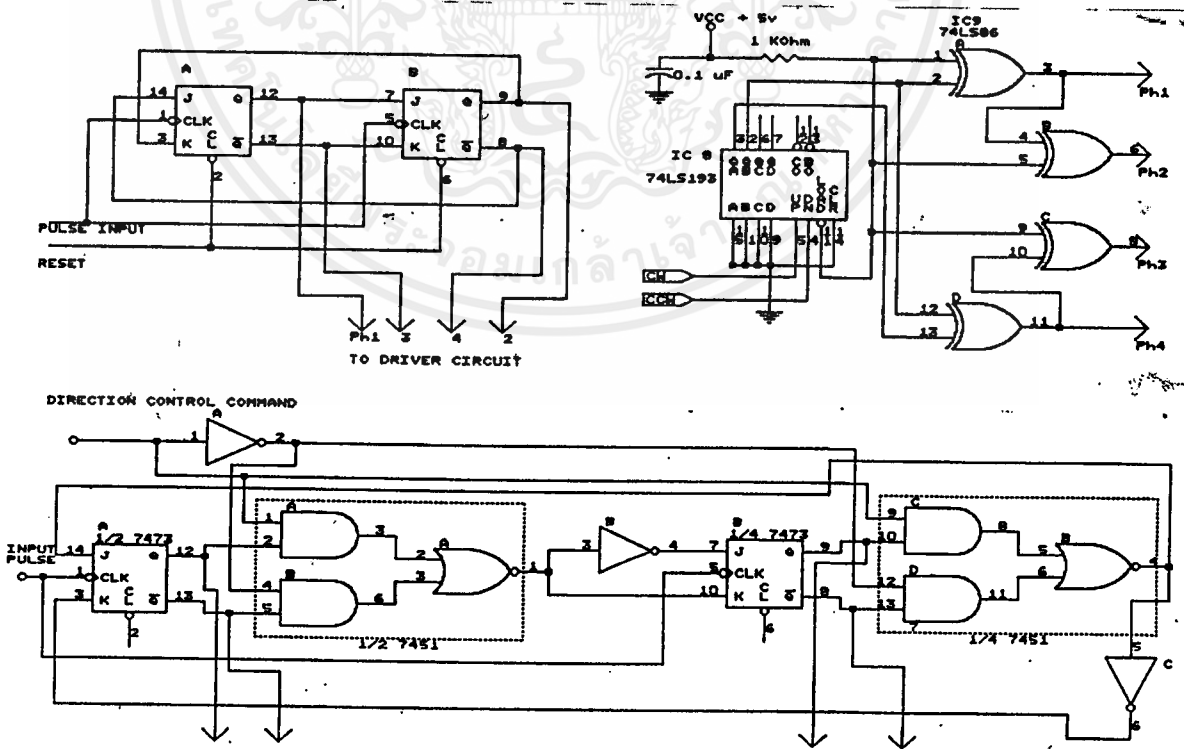
บล็อกไดอะแกรมสำหรับการควบคุมสเตปป์มอเตอร์แสดงดังรูปที่ 20 เป็นวงจรซีเควนเชียลลอจิกจะรับพัลส์อินพุต และคำสั่งควบคุมทิศทางแล้วจึงผลิตพัลส์ที่ใช้ในการกระตุ้นเฟสของมอเตอร์ออกไป แต่จะมีระดับของสัญญาณคำสั่งจึงต้องนำสัญญาณนี้ไปผ่านวงจรขับ เพื่อให้ระดับของสัญญาณสูงขึ้น และจะมีวงจรจำกัดกระแสและแรงดันทำหน้าที่เป็นบัฟเฟอร์ให้กับวงจรขับ เพื่อให้ระดับของสัญญาณสูงขึ้นและมีวงจรจำกัดกระแสและแรงดันทำหน้าที่เป็นบัฟเฟอร์ให้กับวงจรขับและมอเตอร์จากบล็อกไดอะแกรม จะเห็นได้ว่าไม่จำเป็นต้องต่อวงจรเอ็นโค้ดเดอร์ป้อนสัญญาณกลับ เพื่อควบคุมตำแหน่ง และความเร็ว แต่ในบางครั้ง เช่นในกรณีที่มีความสัมพันธ์ของการเคลื่อนที่ระหว่างแกนหมุนของมอเตอร์กับตำแหน่งของโหลดไม่ตรงกัน อันเกิดจากความผิดพลาดของเฟืองเกียร์หรือในกรณีที่โหลดบางอย่างทำให้เกิดการผิดพลาดของสเตปป์ขึ้น ก็จำเป็นต้องมีการใช้สัญญาณย้อนกลับเพื่อควบคุมตำแหน่ง และความเร็วให้ถูกต้อง



รูปที่ 20 บล็อกไดอะแกรมของระบบควบคุมสเตปป์มอเตอร์



รูปที่ 21 แบบของซีเควนเซียลลอจิกสองทิศทางทั้ง 2 แบบ



รูปที่ 22 ตัวอย่างวงจรมอดูเลชันความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเอกสารศึกษาเท่านั้น มิใช่ให้ผู้เห็นลิขสิทธิ์ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

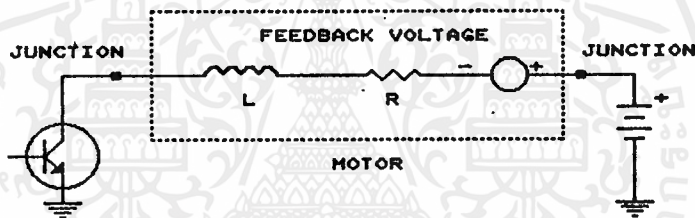
### วงจรรีเคเวนเซียลลอจิก

การออกแบบวงจรรีเคเวนเซียลลอจิกนั้น ความแตกต่างของวงจรรีเคเวนเซียลลอจิกกับว่าผู้ออกแบบจะเลือกใช้อิซิประเภทใดมาใช้ในวงจร วิธีการกระตุ้นเฟลส์จำนวนเฟลส์ของสเตปป์มอเตอร์ และยังรวมถึงจุดประสงค์ของการใช้สเตปป์มอเตอร์ว่าต้องการใช้งานเฉพาะทิศทางเดียวหรือสองทิศทาง ซึ่งในการใช้งานแบบสองทิศทางก็ยังสามารถแบ่งออกได้เป็น 2 แบบ ดังรูปที่ 21 จะเห็นว่าแบบแรกนั้นรับพัลส์อินพุตเข้าทางหนึ่งและรับคำสั่งทิศทางอีกทางหนึ่ง สำหรับในแบบที่สองนั้นก็มีอินพุตอยู่ 2 ทางเช่นกัน แต่จะรับพัลส์พุตทั้งสองทางไม่พร้อมกัน ขึ้นอยู่กับว่าต้องการใช้งานในทิศทางไหน

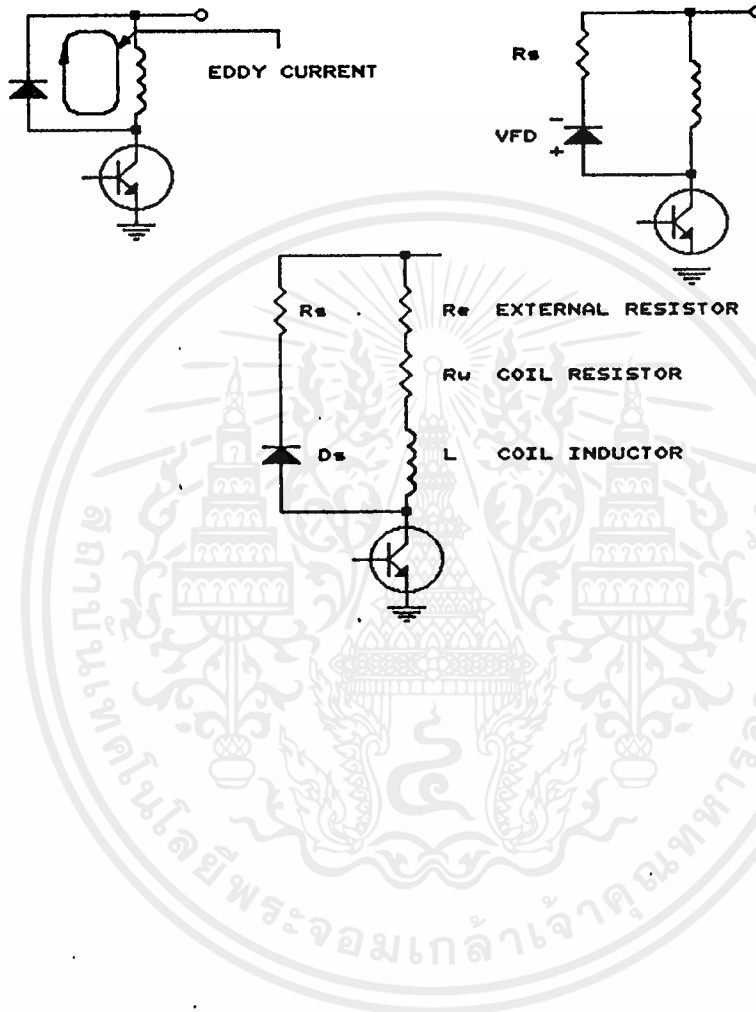
ตัวอย่างสำหรับวงจรรีเคเวนลอจิก แสดงดังในรูปที่ 22 ทั้ง 3 วงจร จะใช้สำหรับสเตปป์มอเตอร์ที่มี 4 เฟลส์ และมีวิธีการกระตุ้นเฟลส์เป็นแบบเฟลส์คู่ สำหรับรูปที่ 22 (ก) ประกอบด้วย J-K ฟลิป-ฟลอป 2 ตัว มีการใช้งานเฉพาะทิศทางเดียวเท่านั้น ส่วนในรูปที่ 22 (ข) มีการใช้ลอจิกเกตเพิ่มเข้ามาด้วย ซึ่งก็เปรียบได้กับบล็อกในรูปที่ 21 (ก) ส่วนในรูปที่ 22 (ค) จะใช้อิซิปรับขึ้นลงแบบเลขฐานสองและลอจิกเกต เพื่อแยกอินพุตในการควบคุมทิศทางซึ่งก็สามารถเปรียบได้กับบล็อกในรูปที่ 21 (ข)

### วงจรรีบ

เมื่อเขียนวงจรสมมูลของขดลวดในสเตปป์มอเตอร์ ก็จะได้ตัวเหนี่ยวนำค้ำอนุกรมกับความต้านทาน และในขณะที่มอเตอร์กำลังหมุนก็จะผลิตแรงดันย้อนกลับขึ้นบนขดลวด วงจรสมมูลจริง ๆ จะเป็นดังรูปที่ 23



รูปที่ 23 วงจรสมมูลของขดลวดขณะสเตปป์มอเตอร์กำลังหมุน



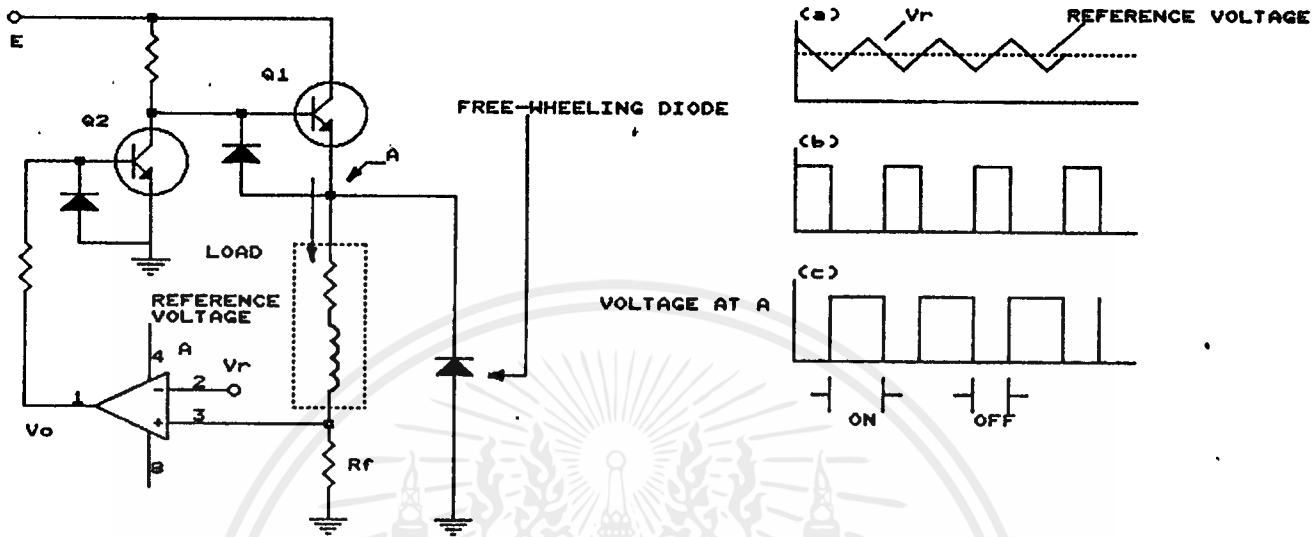
รูปที่ 24 การป้องกันทรานซิสเตอร์ในวงจรขั้วค้ว

(ก) การใช้ไดโอดซีฟเพรสเซอร์

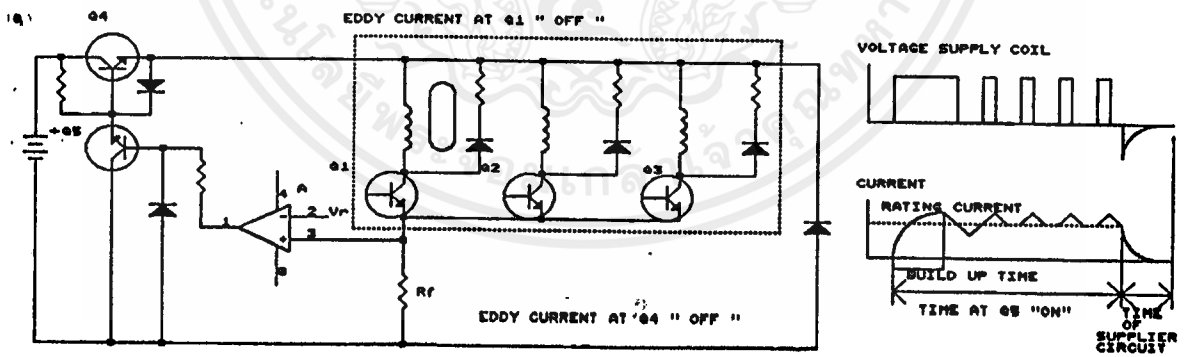
(ข) ไดโอด-รีซิสเตอร์ ซีฟเพรสเซอร์

(ค) การต่อความต้านทาน  $R_s$  ในวงจรขั้วค้วของมอเตอร์ เพื่อลดค่าคงที่เวลา

แรงดันย้อนกลับนี้จะเป็นอันตรายกับทรานซิสเตอร์มาก ในขณะที่ทรานซิสเตอร์กลับมาอยู่ในสภาวะ "OFF" จึงต้องใส่วงจรซัพเพรสเซอร์ (SUPPRESSOR) เข้าไปเพื่อป้องกันทรานซิสเตอร์ไม่ให้เกิดความเสียหายวงจรซัพเพรสเซอร์ง่ายที่สุดคือ ต่อไดโอดคร่อมไปกับขดลวดเลยดังรูปที่ 24 (ก) กระแสย้อนกลับก็จะไหลวนลูปผ่านไดโอดในขณะที่ทรานซิสเตอร์ "OFF" ทำให้เกิดแรงบิดเบรก (BRAKING TORQUE) ขึ้น ดังนั้นจึงต้องเพิ่มความต้านทาน ( $R_s$ ) อนุกรมเข้ากับไดโอดด้วยดังรูปที่ 24 (ข) เพื่อให้เวลาในการสว่างของกระแสไหลวนลดลงเร็วขึ้น แต่ก็มีผลในการเพิ่มขึ้นของแรงดันที่ขาคขดลวดเล็กเตอร์ของทรานซิสเตอร์ด้วยเช่นกัน หนึ่งหากอีกอย่างหนึ่งคือ หลังจากที่ทรานซิสเตอร์ถูกกระตุ้นเฟสให้อยู่ในสภาวะ "ON" จะต้องใช้เวลาระยะหนึ่งก่อนที่จะเข้าสู่สภาวะที่ขั้วด้วยกระแสเต็มพิกัด ซึ่งผลอันนี้เกิดจากค่าคงที่เวลาอันเนื่องมาจากคุณสมบัติของตัวเหนี่ยวนำโดยจะมีผลทำให้แรงบิดลดลงและการตอบสนองช้าลงด้วย วิธีแก้ปัญหอย่างหนึ่งก็คือ การต่อความต้านทาน ( $R_s$ ) อนุกรมเข้ากับขดลวด ดังรูปที่ 24 (ค) เพื่อลดค่าคงที่เวลาให้น้อยลง แต่ในขณะเดียวกันก็ทำให้เกิดความสูญเสียกำลังขึ้นในตัวต้านทานด้วยเช่นกันดังนั้นเพื่อลดปัญหาการสูญเสียจึงมีหันิยมหันมาใช้วงจรพัลส์วิดธ์มอดูเลชัน (PULSE - WIDTH MODULATION : PWM) หรือ วงจรชอปเปอร์ (CHOPPER) แนวนวงจรพื้นฐานของ PWM แสดงในรูปที่ 25 จะใช้ออปแอมป์เป็นตัวเปรียบเทียบแรงดันของสัญญาณอ้างอิง ( $V_r$ ) กับแรงดันตกคร่อม  $R_r$  ที่เกิดจากกระแสไหลลัดไหลผ่าน ถ้าแรงดันตกคร่อม  $R_r$  มีค่ามากกว่าแรงดันไฟตรงอ้างอิงแล้ว  $Q_1$  ก็จะนำกระแสและป้อนไบแอสไปยังขาเบสของ  $Q_1$  ให้ทำงานในโหมด ON/OFF กล่าวคือ ในสภาวะ "ON" จะมีกระแสไหลผ่านโหลดส่วนในสภาวะ "OFF" ก็จะเกิดกระแสลูปผ่านฟรีวีลลิงไดโอด (FREE-WHEELING DIODE) ซึ่งทิศทางของกระแสได้จากรูป



รูปที่ 25 วงจรพื้นฐานของวงจรพัลส์วิดท์มอดูเลชั่น



รูปที่ 26 วงจรพัลส์วิดท์มอดูเลเตอร์สำหรับสแตปปีงมอเตอร์ 3 เฟส และลักษณะสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากลักษณะอันนี้ ทำให้เกิดเป็นสัญญาณสามเหลี่ยมผสมอยู่ในสัญญาณ  
อ้างอิง ( $V_r$ ) มีค่าตั้งแต่ 2-20 กิโลเฮิร์ตซ์ และเพื่อที่จะให้เกิดแรงดันรีปเปิลใน  
กระแสไหลค้ำ ๆ ก็จะต้องพยายามปรับให้แรงดันตกคร่อม  $R_r$  มีค่าเท่ากับแรงดันไฟ  
ตรงอ้างอิง ซึ่งถ้ากระแสไหลค้ำต่ำกว่าค่าที่ต้องการช่วงเวลา "ON" ของ  $Q_1$  จะยาว  
กว่าช่อง "OFF" ทำให้กระแสที่จากแหล่งจ่ายมีค่าสูงขึ้น ซึ่งแรงดันเฉลี่ยที่จ่ายให้มอ  
เตอร์ ( $E_1$ ) จะเป็นไปตามสมการ

$$E_1 = E_x(t_{on}/(t_{on} + t_{off})) \dots (4)$$

วงจร PWM ที่ใช้งานจริงจะเป็นดังรูปที่ 26 หลังจาก  $Q_1$  เริ่มสภาวะ  
"ON" (หลังจากการกระตุ้นเฟส) กระแสไหลค้ำก็จะเพิ่มขึ้น ทำให้เกิดแรงดันตกคร่อม  
 $R_r$  แต่ก็ยังคงต่ำกว่าแรงดันของสัญญาณอ้างอิง ( $V_r$ ) ทำให้  $Q_1$  อยู่ในสภาวะ "ON"  
จึงเกิดแรงดันจากแหล่งจ่ายไฟจ่ายให้กับ  $PH_1$  และกระแสไหลค้ำก็จะเพิ่มขึ้นอย่าง  
รวดเร็วเข้าสู่สภาวะกระแสเต็มพิกัดเนื่องจากผลของการทำงานในโหมด ON/OFF  
ของ  $Q_1$  หลังจากนั้นเมื่อ  $Q_1$  อยู่ในสภาวะ "OFF" (ไม่มีการกระตุ้นเฟส) ก็เกิด  
การลดลงของกระแสอย่างรวดเร็วด้วยวงจรไดโอดรีจิสเตอร์ซัพเพรสเซอร์ รายละเอียด  
ของรูปคลื่นสัญญาณดูได้จากในรูปที่ 26

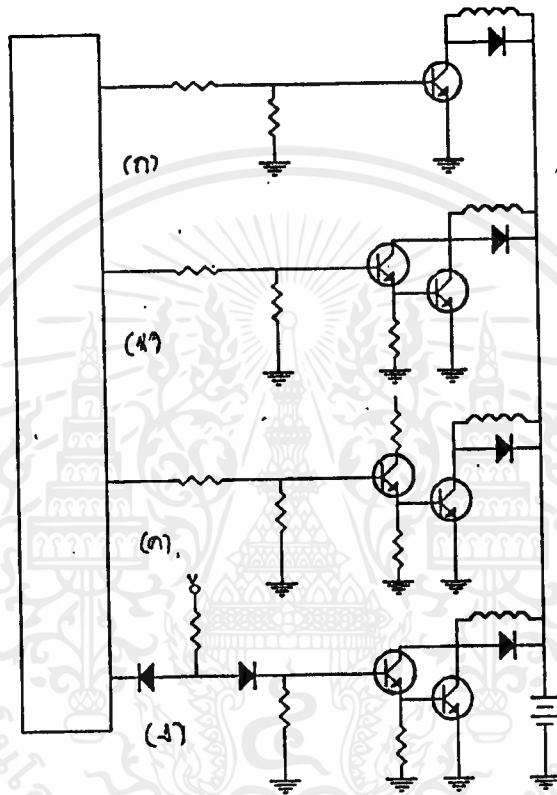
#### วงจรไดเรกเตอร์ของสเตปมอเตอร์

การต่อวงจรซีเควนส์เข้ากับวงจรไดเรกเตอร์

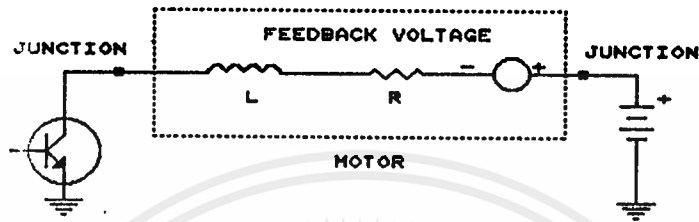
สัญญาณเอาต์พุทของวงจรลอจิกซีเควนส์ จะถูกส่งไปยังอินพุทของวงจร  
เพาเวอร์ไดเรกเตอร์ และด้วยสัญญาณนี้จะไปควบคุมการนำกระแสและหยุดนำกระแส  
ของขดลวดสเตปมอเตอร์วงจรเพาเวอร์ไดเรกเตอร์ อาจเรียกอีกอย่างได้ว่าวงจรขับ  
เคลื่อนมอเตอร์หรือเรียกง่าย ๆ ว่า "วงจรไดเรกเตอร์" วิธีง่ายที่สุดในการต่อวงจร  
ซีเควนส์เข้ากับวงจรไดเรกเตอร์คือ การต่อโดยตรง ดังจะแสดงในรูป 27 (ก) และ  
(ข) แต่ถ้ากระแสเอาต์พุทจากวงจรซีเควนส์ ไม่พอเพียงที่จะไปขับเพาเวอร์ทราน  
ซิสเตอร์แล้ว เราจำเป็นต้องใส่วงจรบัฟเฟอร์สำหรับขยายกระแสระหว่าง 2 ภาค  
ดังแสดงในรูปที่ 27 (ค) และ (ง)

### ปัญหาเกี่ยวกับวงจรโทรเวอร์

ขดลวดของสเตปมอเตอร์เป็นอินดักทีฟ และมีค่าเปรียบเทียบเสมือนผลรวมของอินดักแตนซ์ของขดลวดกับความต้านทานขดลวดของสเตปมอเตอร์ ดังแสดงไว้ในรูปที่ 28 การออกแบบวงจรโทรเวอร์ เราต้องคำนึงถึงแฟลตเตอร์ที่สำคัญ และคุณสมบัติของวงจรถูกกล่าวนี้



รูปที่ 27 ตัวอย่างการต่อระหว่างวงจรลจิกซีควเอนส์ และวงจรโทรเวอร์



รูปที่ 28 วงจรสมมูลย์ของขดลวดของสเตปมอเตอร์

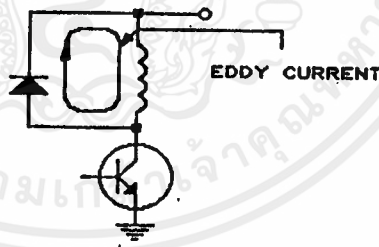
เราจะต้องพิจารณาถึงสภาวะในกรณีวิกฤตของสเตปมอเตอร์ เพาเวอร์ทรานซิสเตอร์และซัพพลายโวลต์เตจ พารามิตเตอร์ของมอเตอร์ จะแปรไปตามข้อพ้อนันท์ให้ของผลิตและสภาวะการทำงานเนื่องจากสเตปมอเตอร์ ได้รับการออกแบบให้ส่งเพาเวอร์ได้สูงที่สุดจากขนาดที่เล็กที่สุด ดังนั้นอนุกรมของโครงร่างสเตปมอเตอร์สามารถมีค่าสูงได้ถึง 100 องศาเซลเซียส ซึ่งความต้านทานของขดลวดจะมีค่าเพิ่มขึ้นได้ 20 ถึง 25 เปอร์เซ็นต์

### ซัพเพรสเซอร์ (SUPPRESSORS)

เมื่อทรานซิสเตอร์ในรูปที่ 28 หยุดนำกระแสจะทำให้เกิดโวลต์เตจค่าสูงขึ้นจำนวนหนึ่งเนื่องจากอัตราการเปลี่ยนของกระแสในอินดักแตนซ์  $L(di/dt)$  และโวลต์เตจนี้อาจจะเป็นอันตรายแก่ทรานซิสเตอร์วิธีการลดโวลต์เตจที่มียอมแหลมและการป้องกันไม่ให้ทรานซิสเตอร์เสียหายกระทำได้หลายวิธี ดังต่อไปนี้

1. ไดโอดซัพเพรสเซอร์ ถ้าเราใส่ไดโอดขนานกับขดลวดของสเตปมอเตอร์ในทิศทางที่แสดงในรูปที่ 29 กระแสหมุนเวียน CIRCULATION CURRENT จะเริ่มไหลหลังจากทรานซิสเตอร์หยุดนำกระแส และกระแสนี้จะลดลงตามเวลา ในรูปแบบนี้กระแสที่ไหลในช่วงการหยุดนำกระแสของทรานซิสเตอร์ จะเปลี่ยนแปลงไม่มาก และศักดาของคอลเล็กเตอร์จะเท่ากับศักดาของแหล่งจ่าย E บวกกับศักดาคกร่อมไดโอด

วิธีการนี้เป็นวิธีที่ง่าย แต่มีข้อเสียที่ว่ากระแสหมุนเวียนจะคงทนอยู่ได้เป็นเวลายาวนานและกระแสจะทำให้เกิดแรงบิดหาล้อ (BRAKING TORQUE)



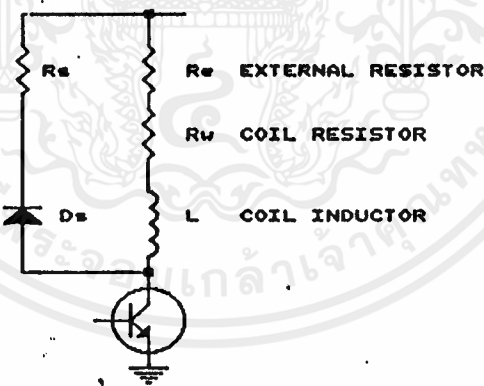
รูปที่ 29 วงจรไดโอดซัพเพรสเซอร์

2. ไดโอดและรีซิสเตอร์ซัพเพรสเซอร์ วิธีการนี้เราต่อต้านความต้านทานอนุกรมกับไดโอด ดังในรูปที่ 30 เพื่อให้กระแสหมุนเวียนลดลงได้อย่างรวดเร็ว โวลต์เต็ม  $V_{Df}$  ที่ป้อนเข้าคอลเล็คเตอร์ในช่วงการหยุดหน้ากระแสของรูปแบบวงจรนี้เท่ากับ

$$V_{Df} = E + IR_s + V_{df}$$

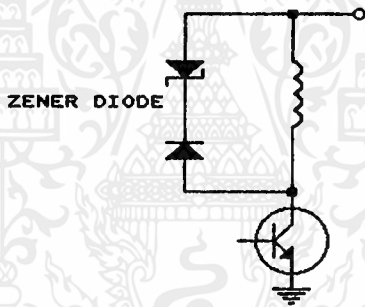
- เมื่อ  $E$  = ศักดาของแหล่งจ่าย
- $I$  = กระแสไหลก่อนที่ทรานซิสเตอร์จะหยุดหน้ากระแส
- $R_s$  = ความต้านทานของตัวซัพเพรสเซอร์รีซิสเตอร์
- $V_{df}$  = ศักดาตกคร่อมไดโอดเมื่อไดโอดนำกระแส

ความต้านทาน  $R_s$  มีค่าสูงจะทำให้กระแสลดลงได้เร็วขึ้นหลังจากทรานซิสเตอร์หยุดหน้ากระแส แต่ศักดาของคอลเล็คเตอร์จะมีค่าสูงขึ้นดังนั้นค่าโวลต์เต็มสูงสุดของทรานซิสเตอร์จำเป็นต้องมีค่าสูงขึ้น เพื่อให้กระแสหมุนเวียนลดลงได้อย่างรวดเร็ว

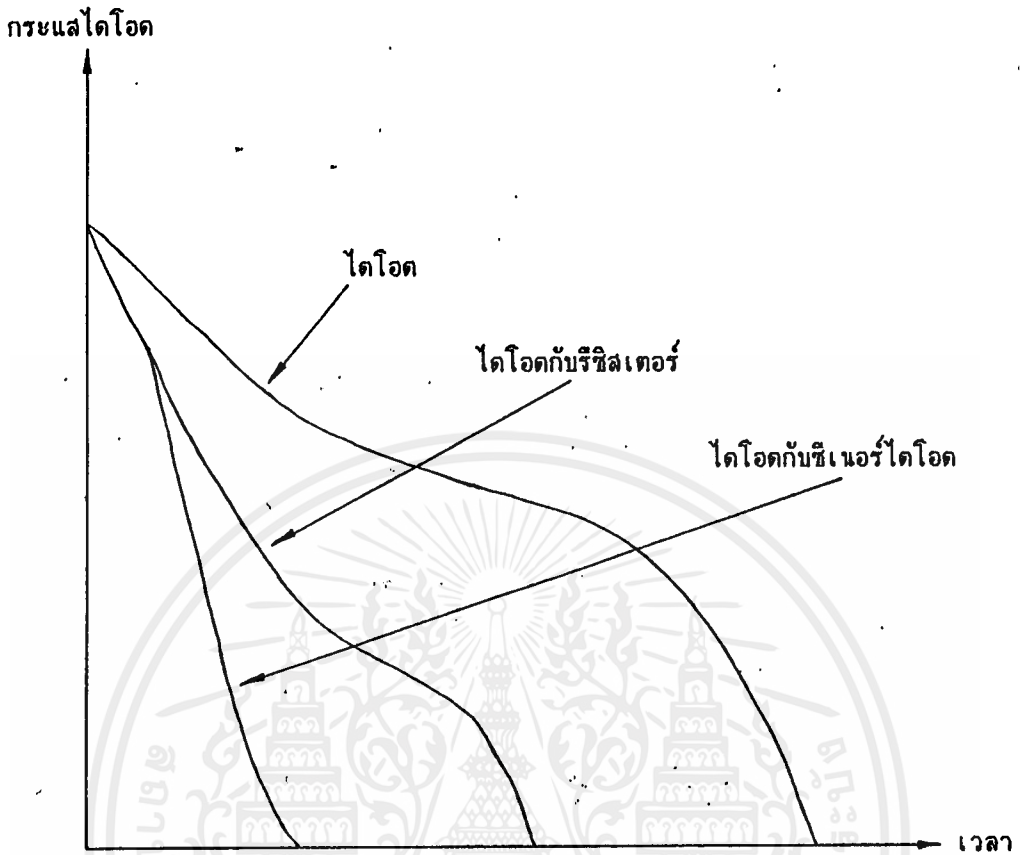


รูปที่ 30 ไดโอดและรีซิสเตอร์ซัพเพรสเซอร์

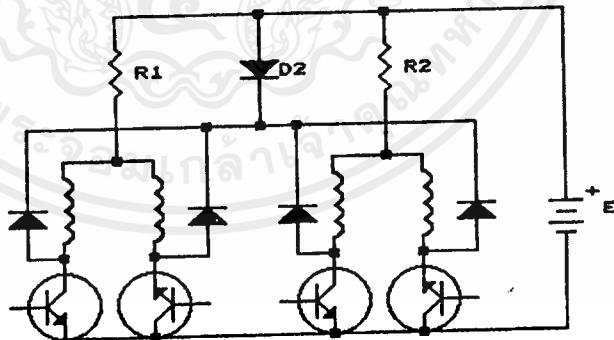
3. ซีเนอร์ไดโอดซีฟเพรสเซอร์ เรามักจะใช้ซีเนอร์ไดโอดก่อนกรมกับไดโอดธรรมดา ดังแสดงในรูปที่ 31 เมื่อเปรียบเทียบรูปวงจรมีที่เห็นแล้ว กระแสหมนเวียนในวงจรมันจะลดลงได้เร็วกว่า หลังจากทรานซิสเตอร์หยุดหน้ากระแสคั้งในรูปที่ 32 นอกจากนั้นวงจรมันยังมีข้อได้เปรียบคือ ศักดาที่คอลเล็คเตอร์จะเท่ากับ ศักดาของแหล่งจ่ายบวกกับศักดาของซีเนอร์ ซึ่งเป็นอิสระต่อกระแสคั้งนั้นการวินจดยค่าศักดาสูงสุดของคอลเล็คเตอร์กระทำได้ง่าย รูปที่ 33 คือ วงจรไดรเวอร์แบบซีเนอร์ไดโอดซีฟเพรสสำหรับสเตมอเตอร์แบบ 4 เฟส ความต้านทาน  $R_1$  และ  $R_2$  จะเป็นตัวช่วยให้กระแสกระตุ้นเพิ่มขึ้นอย่างรวดเร็ว ซึ่งจะได้อธิบายในหัวข้อ”แก้ไขการเพิ่มขึ้นของกระแส”



รูปที่ 31 วงจรซีเนอร์ไดโอดซีฟเพรสเตอร์



รูปที่ 32 แสดงการเปรียบเทียบผลของวงจรซัพเพรสเซอร์หลาย ๆ แบบ



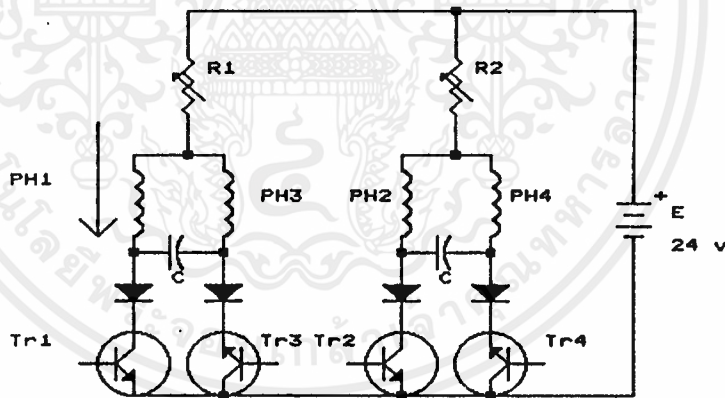
รูปที่ 33 ตัวอย่างวงจรไดโอดเวอร์สำหรับ 4 เฟส แบบมีซีเนอร์ไดโอดซัพเพรสเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. คอนเด็นเซอร์ซีฟเพรสเซอร์ วงจรในรูปแบบนี้ มักจะนิยมใช้กับ สเตปมอเตอร์แบบไบฟิลาไรววด์ (BIFILARWOUND) ดังแสดงในรูปที่ 34 ซึ่งเป็น วงจรไดรเวอร์สำหรับ 4 เฟส เราจะใส่คอนเด็นเซอร์ไว้ระหว่างเฟส 1 กับ เฟส 3 และระหว่างเฟส 2 กับ เฟส 4 คอนเด็นเซอร์ทั้งสองตัวใช้ประโยชน์ได้ 2 ลักษณะคือ

กรณีแรกเมื่อทรานซิสเตอร์หยุดนำกระแส คอนเด็นเซอร์จะต่อกับทรานซิสเตอร์ โดยผ่านทางไดโอดและจะดูดกลืนกระแสที่ค่อย ๆ ลดลงจากขดลวดของมอเตอร์ เพื่อป้องกันทรานซิสเตอร์ไม่ให้ได้รับอันตราย

นอกจากนั้น คอนเด็นเซอร์ยังใช้เป็นตัวแดมเปอร์ไฟฟ้า (ELECTRICAL DAMPER) คือ ช่วยลดความร้อนที่เกิดขึ้นในขดลวดโรเตอร์ของมอเตอร์เนื่องจากการ ออสซิลเลทของโรเตอร์, คอนเด็นเซอร์ซีฟเพรสเซอร์ เหมาะสมกับวงจรไดรเวอร์ที่มี สเต็ปิ่งเรท (STEPTING RATE) ถูกจำกัดอยู่ในย่านแคบ ๆ



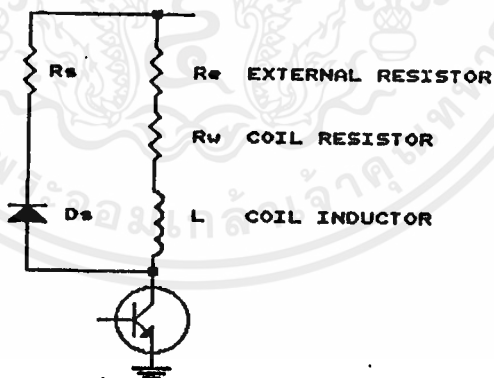
รูปที่ 34 วงจรไดรเวอร์แบบคอนเด็นเซอร์ซีฟเพรสเซอร์สำหรับมอเตอร์ 4 เฟส ความต้านทาน  $R_c$  ใช้สำหรับให้ได้ค่ากระแสตามต้องการ

### แก้ไขการเพิ่มขึ้นของกระแส

เมื่อทรานซิสเตอร์นำกระแสเพื่อกระตุ้นเฟส เพาเวอร์ซัพพลายจะต้องชนะผลของอิมพีแดนซ์ของขดลวดก่อนที่กระแสจะเพิ่มขึ้นได้ถึงค่าที่กำหนด คือ อิมพีแดนซ์มีคุณสมบัติที่จะต้านการเพิ่มขึ้น อัตราส่วนของเวลาการเพิ่มขึ้นของกระแสต่อไซเคิลก็จะมีค่ามากขึ้น และเป็นผลให้แรงบิด (TORQUE) ลดลง และผลตอบสนองลดลง วิธีการลดเวลาการเพิ่มขึ้นของกระแส และปรับปรุงคุณลักษณะของแรงบิดที่ความเร็วสูงให้ดีขึ้น เราสามารถกระทำได้หลายวิธี ดังนี้

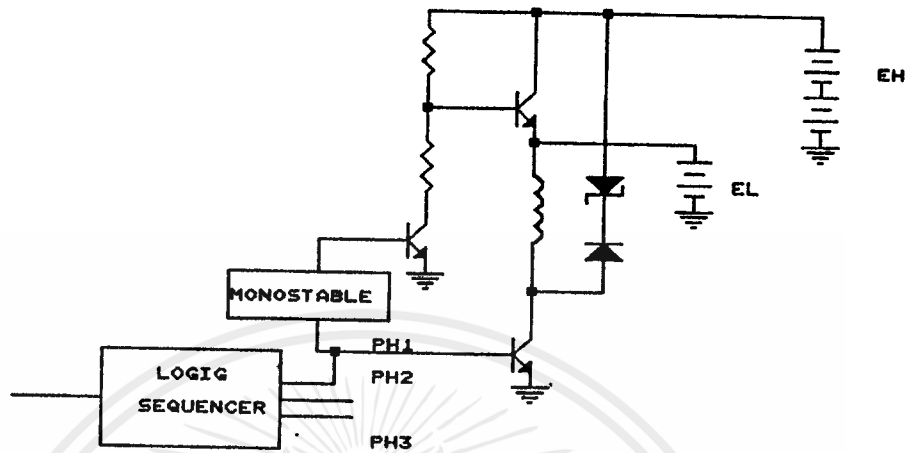
1. ต่อความต้านทานอนุกรม วิธีทางที่ประหยัดที่สุดคือการต่อความต้านทานอนุกรมกับขดลวดของมอเตอร์ ดังแสดงในรูปที่ 35 ศักดาของเพาเวอร์ซัพพลาย E จะต้องได้รับการเลือกค่าที่เหมาะสมเพื่อให้ได้ค่ากระแสที่ต้องการไหลผ่านขดลวดของมอเตอร์ที่สภาวะคงที่ใหม่คอนสแตนท์ของวงจรจะลดลงจาก  $L/R_w$  เป็น  $L/(R_e + R_w)$

2. โวลต์เตจสองค่า เพื่อเป็นการลดการสูญเสียกำลังในวงจรไดรเวอร์เพิ่มประสิทธิภาพของการทำงานของมอเตอร์ เรามักจะใช้วงจรไดรเวอร์ใช้โวลต์สองค่า วงจรสำหรับแสดงดังในรูปที่ 27



รูปที่ 35 แก้ไขการเพิ่มขึ้นของกระแสด้วยการใส่  $R_e$  อนุกรมกับขดลวดของมอเตอร์และเพิ่มค่าศักดาของแหล่งจ่าย E

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 36 วงจรแก้ไขการเพิ่มขึ้นของกระแสด้วยจอร์ไมโครเวอร์แบบใช้โวลต์เดี่ยวสองค่า

STEPPING MOTOR CONTROLLER

Project ชุดทดลอง Stepping motor นี้จะแบ่งออกเป็น 3 ส่วนด้วยกัน คือ

1. ชุด Interface
2. ชุด Decoder
3. ชุด Driver

ส่วนประกอบและการทำงานของวงจร

1. ชุด Interface ในที่นี่จะใช้ไอซีเบอร์ 8255 (U1) ซึ่งเป็น Port Interface ภายในตัวจะประกอบไปด้วยพอร์ท 3 พอร์ทคือ พอร์ท A, พอร์ท B และพอร์ท C ซึ่งตำแหน่งแต่ละพอร์ทดังนี้

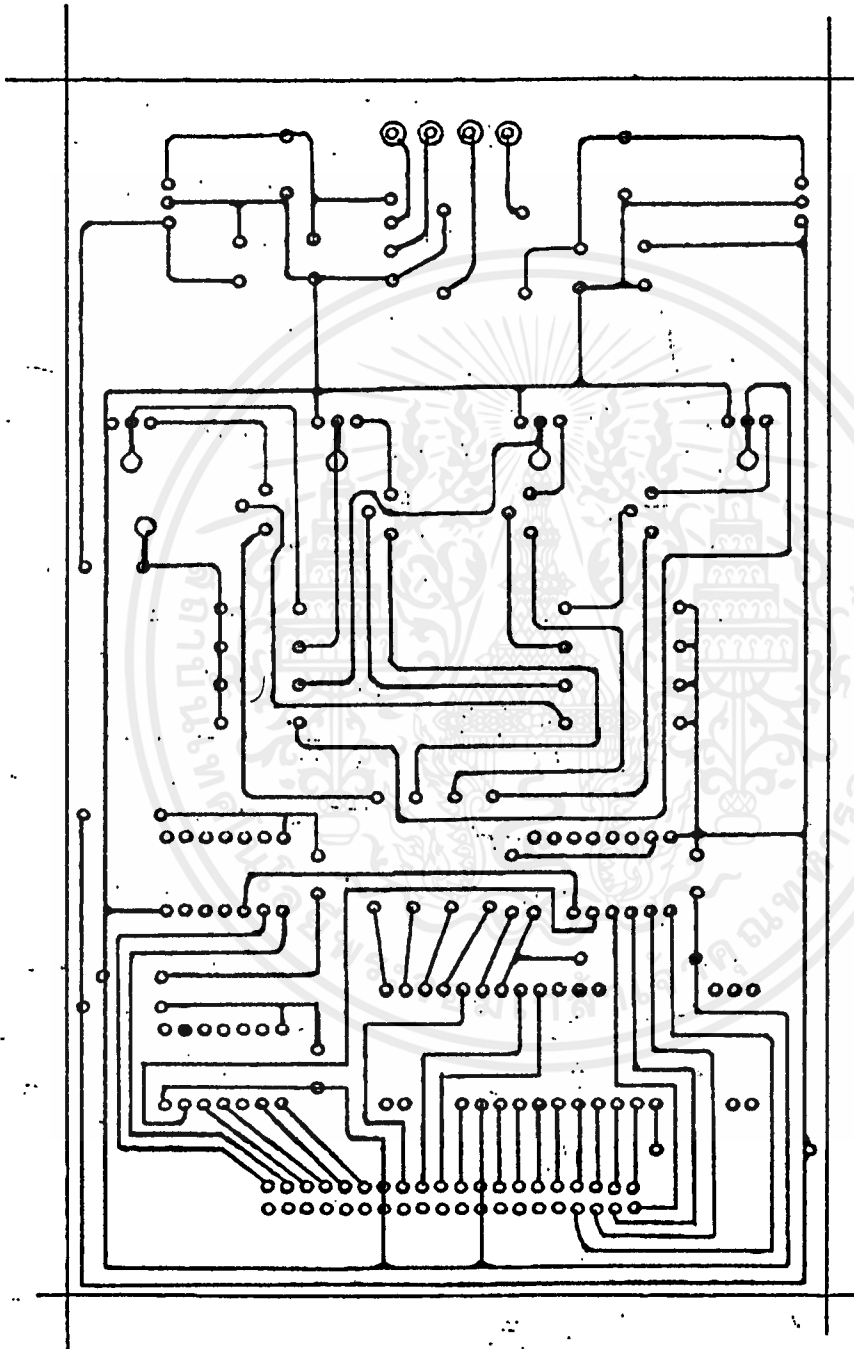
พอร์ท A = 6F0 H

พอร์ท B = 6F1 H

พอร์ท C = 6F2 H

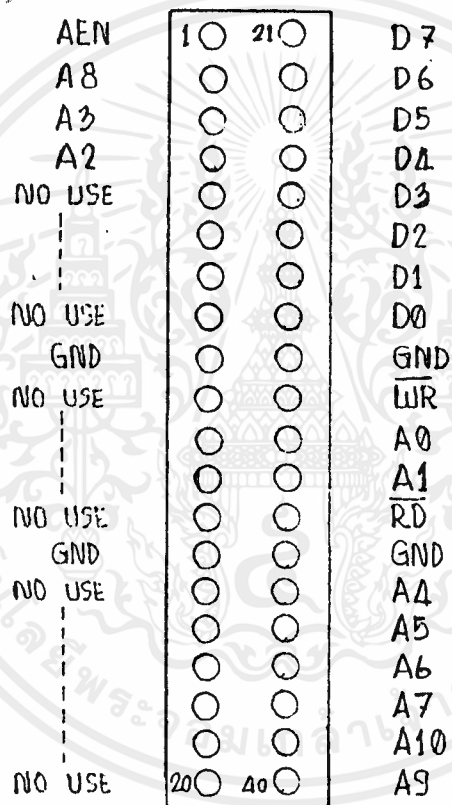
การทำงานโปรแกรมควบคุมจะส่งมาเพื่อเลือก port A เป็น Output port และจะนำสัญญาณ 4 เส้นเพื่อส่งไปยัง Driver Unit และนำไป Control Stepping Motor ต่อไป

2. ชุด Decoder ประกอบด้วย U2, U3A, และ U4A ซึ่งจะนำเอา Address A2-A10 และสัญญาณ AEN จาก Personal Computer มาตีโค้ดเพื่อเลือก Address port 6F0H-6F3H แล้วส่งค่า ไปที่ขา CS (chip enable) ของ 8255 เพื่อเป็นการเปิด Gate ให้ 8255 ทำงาน
3. ชุด Driver ประกอบด้วย Q1-Q8 สัญญาณ Drive phase ของ Stepping Motor จะมาจาก IC#8255 โดย Q1-Q4 จะเป็น Pre-driver ให้กับ Q5-Q8 อีกที ทั้งนี้เพื่อเพิ่มกระแส Drive ส่วน D1-D4 จะต่อคร่อมกับขดของ Stepping Motor เพื่อป้องกัน back EMS อันเกิดจากการยุบตัวของสนามแม่เหล็กของขดลวด



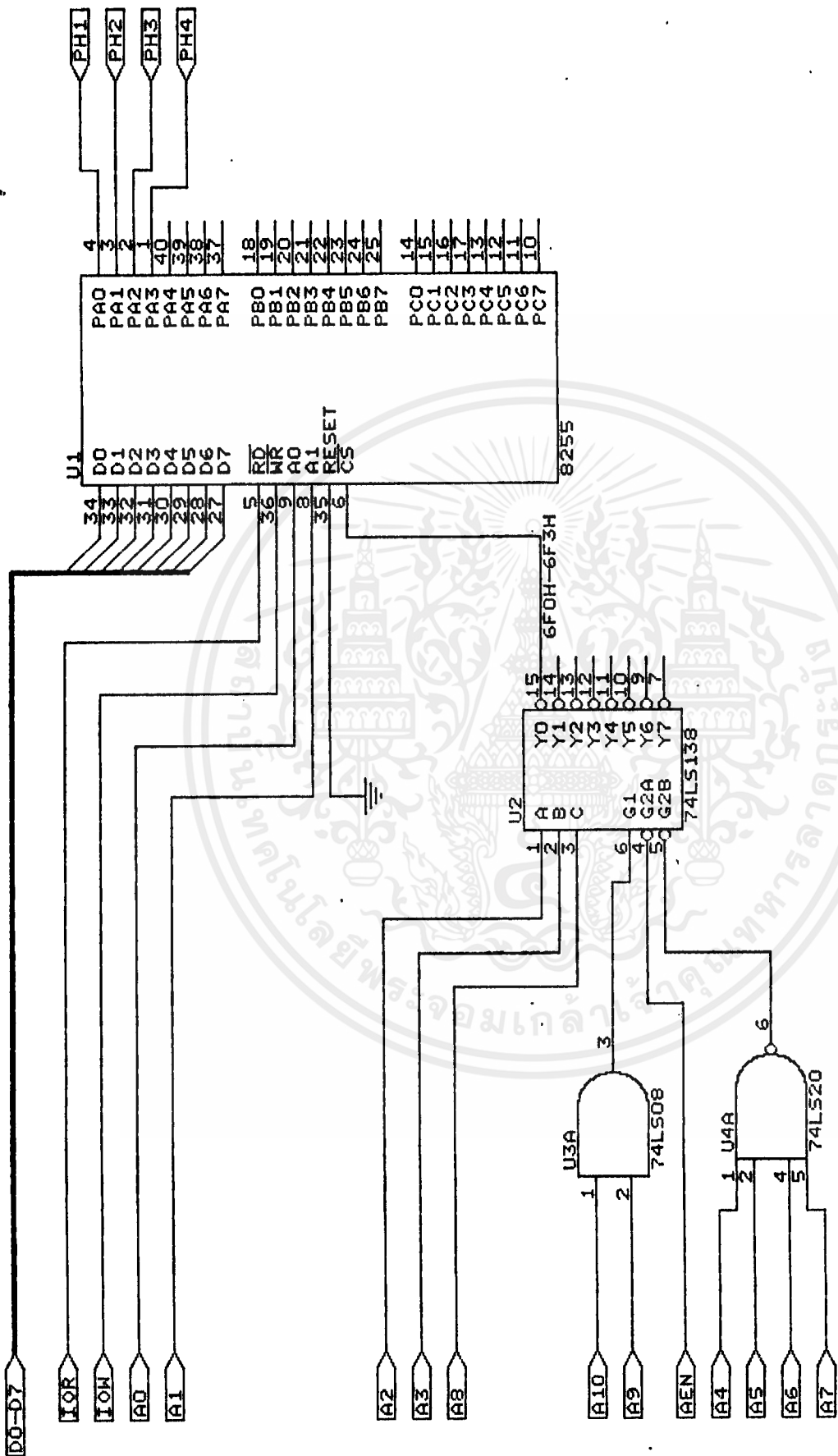
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### CN1



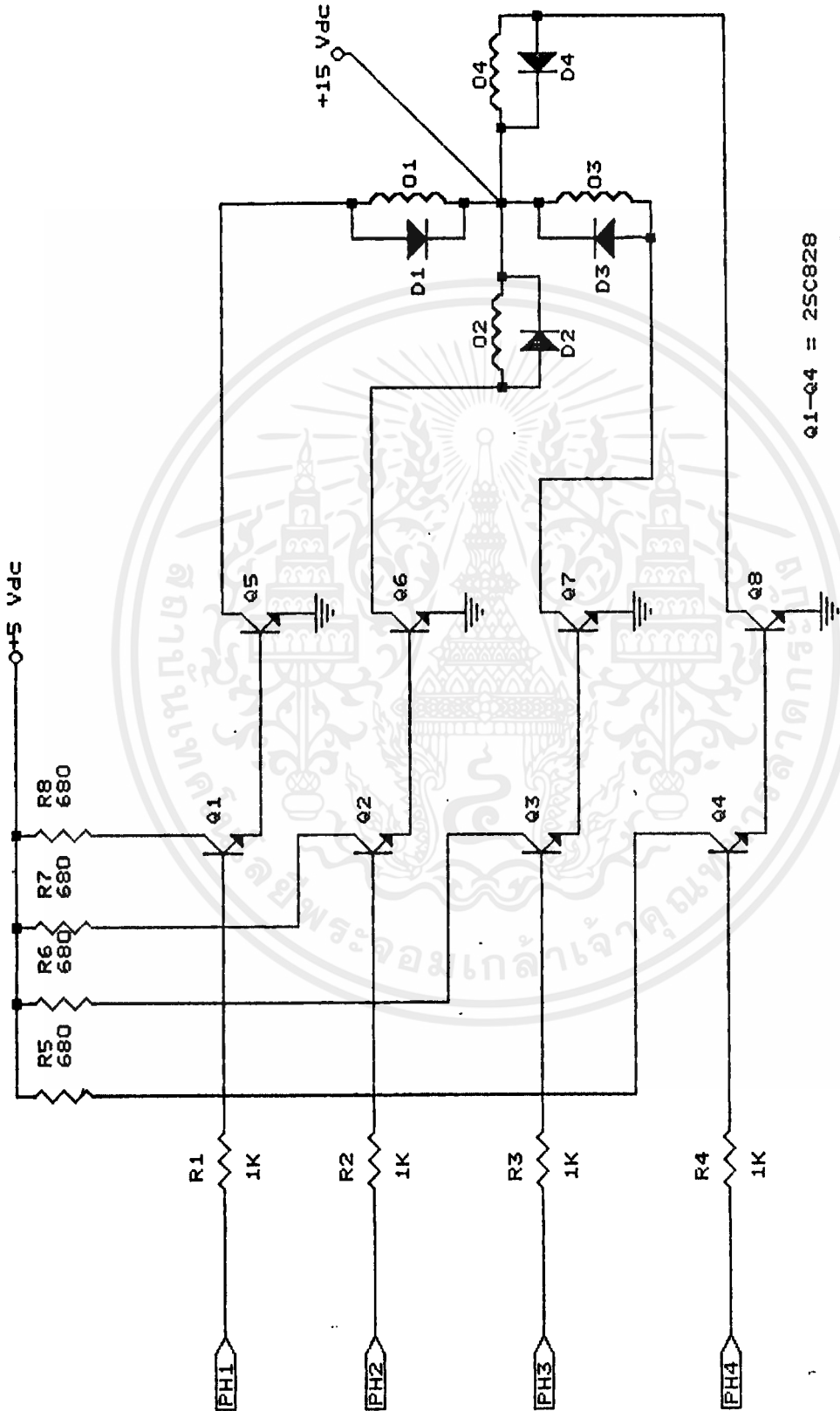
### CONNECTOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



STEPPING MOTOR CONTROLLER	
Size	Document Number
A	INTERFACE
Date:	February 22, 1993
Sheet	1 of 2
REV	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

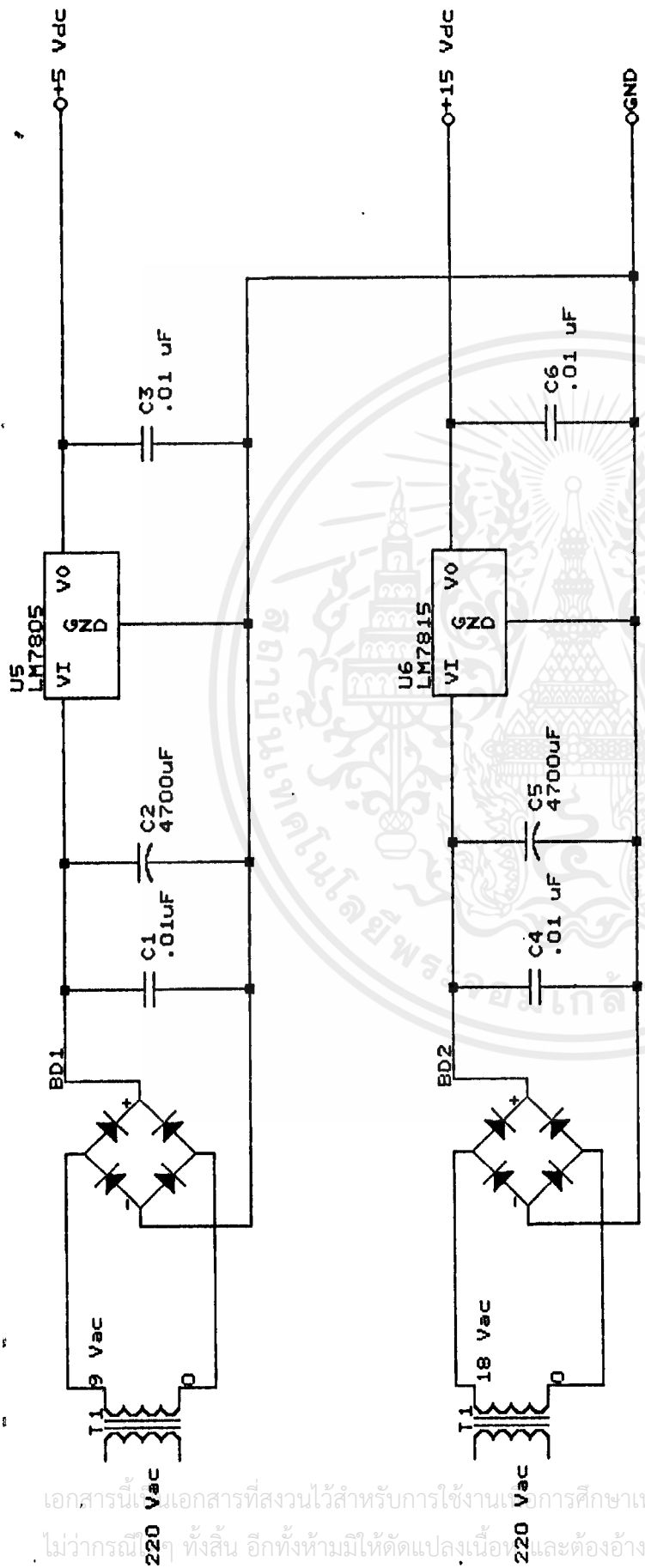


Q1-Q4 = 2SC828  
 Q5-Q8 = 2SC1061  
 D1-D4 = 1N4148

STEPPING MOTOR CONTROLLER

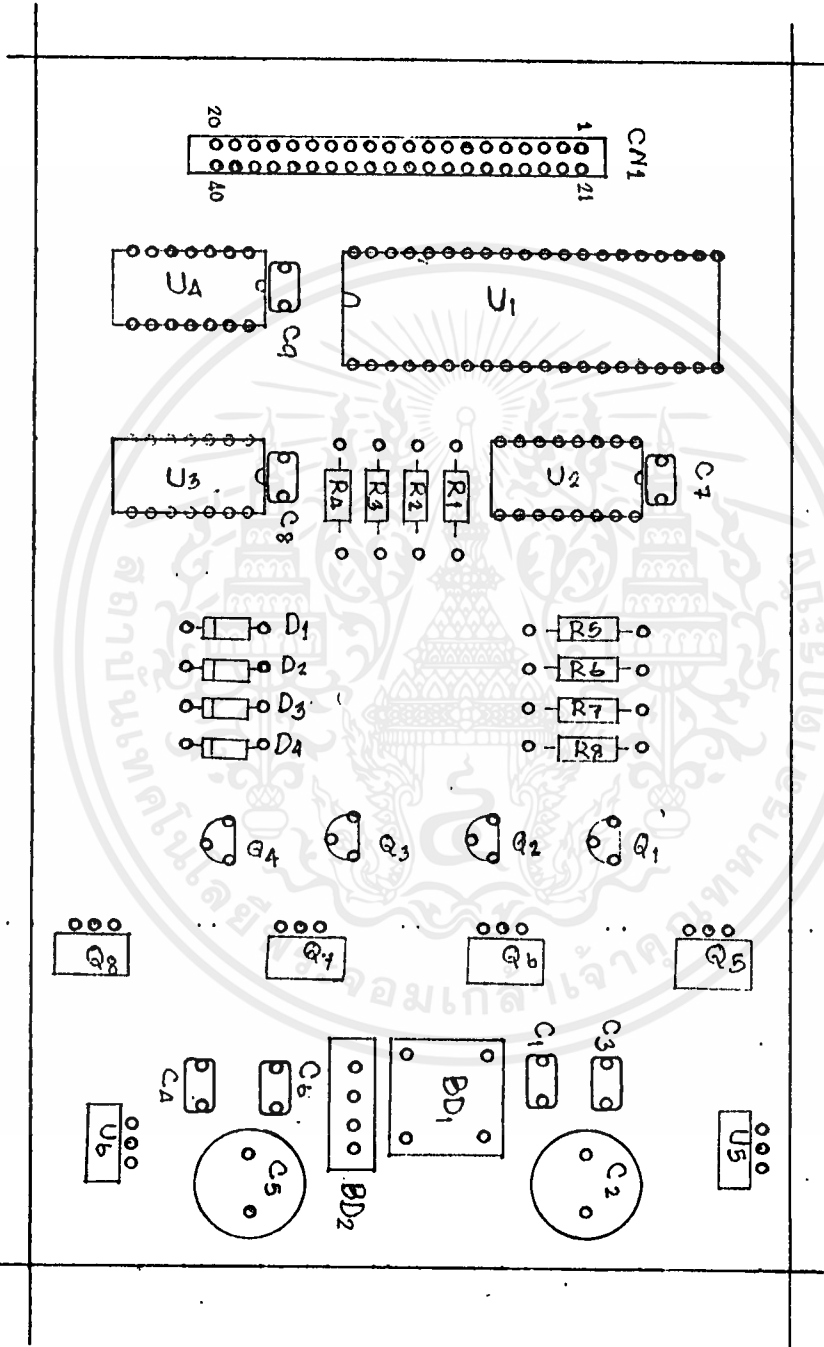
Size Document Number	REV
A	
Date: February 22, 1993	Sheet 2 of 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณี 220 Vac ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

STEPPING MOTOR CONTROLLER	
Size Document Number	REV
A	A
Date: February 24, 1993	Sheet of



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โปรแกรมการทำงานของเครื่องควบคุมสแตปปีงมอเตอร์

```
#include <graphics.h>
#include <math.h>
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <alloc.h>

#define cst 25
#define PortCtrl 0x6f3
#define PortA 0x6f0

void far *CurSer;
int CurrentX,CurrentY;
int Table[10][22];
int Step,Phase;

void getcurser(void)
{
    rectangle(1,1,cst-1,cst-1);
    CurSer = (void far*)farmalloc(imagesize(0,0,cst-1,cst-1));
    getimage(1,1,cst-1,cst-1,CurSer);
    putimage(1,1,CurSer,XOR_PUT);
}

char ReadKey(int X,int Y)
{
    unsigned int i;
    while(!kbhit())
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
    putimage(X,Y,CurSer,XOR_PUT);
    for(i=0;i<65535;i++) ;
    putimage(X,Y,CurSer,XOR_PUT);
    for(i=0;i<65535;i++) ;
}
return getch();
}

void SetTable(int Col,int Row)
{
    if(Table[Row][Col] == 0)
    {
        Table[Row][Col] = 1;
        setfillstyle(1,15);
        bar(CurrentX,CurrentY,CurrentX+cst-2,CurrentY+cst-2);
    }
    else
    {
        Table[Row][Col] = 0;
        setfillstyle(1,0);
        bar(CurrentX,CurrentY,CurrentX+cst-2,CurrentY+cst-2);
    }
}

void OutPort(void)
```

```
{  
    int i,j,c;  
    int Res;  
    char buff[30];  
    c = getcolor();  
    setcolor(15);  
    while(!kbhit())  
    {  
        for(i=0;i<Step;i++)  
        {  
Res = 0;  
for(j=0;j<Phase;j++) Res += Table[j][i] * pow(2,j);  
  
/*sprintf(buff,"Res[%d] : %d",i,Res);  
outtextxy(50,250+(i*10),buff);*/  
  
/*printf("Res[%d] : %d \n",i,Res);  
delay(200);*/  
outp(PortA,Res);  
sleep(1);  
        }  
    }  
    setcolor(c);  
}  
  
main()
```

{

```
int driver=DETECT,mode;
int x1,x2,y1,y2;
int cnt,var,var1,Done=1;
char ch,buff[20];

clrscr();
outp(PortCtrl,0x80); /*Control word of 8255*/
for(x1=0;x1<8;x1++)
  for(x2=0;x2<20;x2++) Table[x1][x2] = 0;
Phase=9;
while(Phase>8)
{
  printf(" Input data of phase motor <Max = 8 Phase> : ");
  scanf("%d",&Phase);
}
printf("\n\n");
Step=21;
while(Step>20)
{
  printf(" Input data of Step motor <Max = 20 Step> : ");
  scanf("%d",&Step);
}
initgraph(&driver,&mode,"");
```

```
settextstyle(SMALL_FONT,HORIZ_DIR,5);
outtextxy(10,323,"ENTER => Selec Phase of Motor");
outtextxy(300,323,"SPACEBAR => Run <Press any key to exit> ");
outtextxy(10,335,"ARROW => Move Cursor ");
outtextxy(300,335,"ESC => Quit ");
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
getcurser();

var = Phase+2;
x1 = y1 = y2 = 50;
x2 = x1 + (cst * (Step + 2));
for(cnt=0;cnt<=var;cnt++)
{
    if(cnt>2)
    {
        sprintf(buff,"%d",cnt-2);
        outtextxy(x1+20,y1-15,buff);
    }

    if(cnt==1)
        outtextxy(((cst*(Step+2))/2),40," STEP OF MOTOR");
    else
        line(x1,y1,x2,y2);
    y1 = y1+cst;
    y2 = y2+cst;
}
```

```
}

var = Step + 2 ;
x1 = x2 = y1 = 50;
y2 = y1 + (cst * (Phase + 2));
for(cnt=0;cnt<=var;cnt++)
{
    if(cnt>2)
    {
        sprintf(buff,"%d",cnt-2);
        outtextxy(x1-15,y1+20,buff);
    }
    if(cnt!=1)
        line(x1,y1,x2,y2);
    else
    {
        settextstyle(DEFAULT_FONT,VERT_DIR,1);
        outtextxy(40,((cst*(Phase+2))/2),"PHASE OF MOTOR");
        settextstyle(TRIPLEX_FONT,HORIZ_DIR,4);
        outtextxy(0,0,"Program Control Steping Motor");
        settextstyle(SMALL_FONT,HORIZ_DIR,5);
        outtextxy(520,18,"By 2Q <KMITL>");

        settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
    }
    x1 = cst+x1;
    x2 = cst+x2;
}
```

```
    }  
    CurrentX = CurrentY = 101;  
    x1 = y1 = 0;  
    while(Done)  
    {  
        ch = ReadKey(CurrentX,CurrentY);  
        if(ch == 0)  
        {  
            ch = getch();  
            switch(ch)  
            {  
                case 77 : if(x1<Step-1)  
                {  
                    CurrentX += cst;  
                    x1++;  
                }; break;  
                case 75 : if(x1>0)  
                {  
                    CurrentX -= cst;  
                    x1--;  
                }; break;  
                case 72 : if(y1>0)  
                {  
                    CurrentY -= cst;  
                    y1--;  
                }; break;  
            }  
        }  
    }  
};
```

```
    case 80 :if(y1<Phase-1)
{
    CurrentY += cst;
    y1++;
}; break;

    default:printf("\b");break;
}
}

    if(ch == 13) SetTable(x1,y1);
    if(ch == 27) Done = 0;
    if(ch == ' ') OutPort();
}
closegraph();
farfree(CurSer);
/*for(x1=0;x1<Phase;x1++)
{
    for(x2=0;x2<Step;x2++)
        printf(" Table[%d][%d] = %d \n",x1,x2,Table[x1][x2]);
    getch();
}*/
}
```

```
program Examination_Stepping_Motor;
```

```
uses crt,dos;
```

```
const maxstp = 8;
```

```
port0 = $6f0;
```

```
port1 = $6f1;
```

```
port2 = $6f2;
```

```
port3 = $6f3;
```

```
on = true;
```

```
off = false;
```

```
bar = $78;
```

```
var phase : array[1..maxstp] of byte;
```

```
index : record
```

```
  x : byte;
```

```
  y : byte;
```

```
end;
```

```
ch : char;
```

```
speed : longint;
```

```
cntset : longint;
```

```
counter : longint;
```

```
quit : boolean;
```

```
flagattr : integer;
```

```
setf : boolean;
```

```
lastattr : integer;
```

```
toggle : byte;
```

```
spmde : boolean;
```

```
cnt : byte;
```

```
procedure init;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
var i : byte;

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
  cntset := 0;
  counter := 0;
  quit := false;
  { textattr := 2;}
  lastattr := textattr;
  clrscr;
  for i := 1 to maxstp do phase[i] := 0;
  index.x := 1 ; index.y := 1;
  counter := 0;
  port[port3] := $80;
  speed := 20;
  flagattr := $72;
end;

procedure dspphase;
var  arg,i : byte;
     flag : char;
begin
  for i := 1 to maxstp do
    begin
      if phase[i] and $01 <> 0 then flag := '√' else flag := ' ';
      gotoxy(19+i*4,10);
      write(flag);

      if phase[i] and $02 <> 0 then flag := '√' else flag := ' ';
      gotoxy(19+i*4,12);
      write(flag);

      if phase[i] and $04 <> 0 then flag := '√' else flag := ' ';
      gotoxy(19+i*4,14);
```

```

write(flag);

if phase[i] and $08 <> 0 then flag := '√' else flag := ' ';
gotoxy(19+i*4,16);
write(flag);
end;
gotoxy(19+index.x*4,8+index.y*2);
end;

```

```

procedure phaseslide(ph,i:byte);

```

```

begin

```

```

if ph and $01 <> 0 then

```

```

begin

```

```

if setf then textattr := flagattr else textattr := lastattr;

```

```

gotoxy(19+i*4,10);

```

```

write(' ');

```

```

end;

```

```

if ph and $02 <> 0 then

```

```

begin

```

```

if setf then textattr := flagattr else textattr := lastattr;

```

```

gotoxy(19+i*4,12);

```

```

write(' ');

```

```

end;

```

```

if ph and $04 <> 0 then

```

```

begin

```

```

if setf then textattr := flagattr else textattr := lastattr;

```

```

gotoxy(19+i*4,14);

```

```

write(' ');

```

```

end;

```

```

if ph and $08 <> 0 then

```

```

begin

```

```

if setf then textattr := flagattr else textattr := lastattr;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามแก้ไขเปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    gotoxy(19+i*4,16);
    write(√');
end;
textattr := lastattr;
end;
```

```
procedure curser(status:boolean);
var   regs : registers;
begin
    regs.cl := 13;
    if status then regs.ch := 13 else regs.ch := $20+14;
    regs.ah := 1;
    intr($10,regs);
end;
```

```
procedure mvup;
begin
    if index.y > 1 then index.y := index.y-1 else index.y := 4;
    gotoxy(19+index.x*4,8+index.y*2);
end;
```

```
procedure mvdn;
begin
    if index.y < 4 then index.y := index.y+1 else index.y := 1;
    gotoxy(19+index.x*4,8+index.y*2);
end;
```

```
procedure mvrt;
begin
    if index.x < maxstp then index.x := index.x+1 else index.x := 1;
    gotoxy(19+index.x*4,8+index.y*2);
```

end;

procedure mvlt;

begin

if index.x > 1 then index.x := index.x-1 else index.x := maxstp;

gotoxy(19+index.x\*4,8+index.y\*2);

end;

procedure mvcurser;

var ch : char;

begin

ch := readkey;

{ gotoxy(50,2); write(ch);}

gotoxy(19+index.x\*4,8+index.y\*2);

case ch of

'H' : mvup;

'K' : mvlt;

'M' : mvrt;

'P' : mvdn;

'-' : quit := true;

else write(#7);

end;

end;

procedure setflag;

begin

if index.y = 1 then

begin

if phase[index.x] and \$1 <> 0 then

phase[index.x] := phase[index.x] - \$1

else

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    phase[index.x] := phase[index.x] + $1;
```

```
end;
```

```
if index.y = 2 then
```

```
    begin
```

```
        if phase[index.x] and $2 <> 0 then
```

```
            phase[index.x] := phase[index.x] - $2
```

```
        else
```

```
            phase[index.x] := phase[index.x] + $2;
```

```
        end;
```

```
if index.y = 3 then
```

```
    begin
```

```
        if phase[index.x] and $4 <> 0 then
```

```
            phase[index.x] := phase[index.x] - $4
```

```
        else
```

```
            phase[index.x] := phase[index.x] + $4;
```

```
        end;
```

```
if index.y = 4 then
```

```
    begin
```

```
        if phase[index.x] and $8 <> 0 then
```

```
            phase[index.x] := phase[index.x] - $8
```

```
        else
```

```
            phase[index.x] := phase[index.x] + $8;
```

```
        end;
```

```
end;
```

```
procedure mkscr;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

begin

gotoxy(12,5);write(' ');

gotoxy(12,6);write(' ');

gotoxy(12,7);write(' ');

gotoxy(12,8);write(' ');

gotoxy(12,9);write(' ');

gotoxy(12,10);write(' ');

gotoxy(12,11);write(' ');

gotoxy(12,12);write(' ');

gotoxy(12,13);write(' ');

gotoxy(12,14);write(' ');

gotoxy(12,15);write(' ');

gotoxy(12,16);write(' ');

gotoxy(12,17);write(' ');

gotoxy(62,16);write('20');

textattr := textattr or bar;

gotoxy(1,22);write('Alt-x:Exit G:Go Space bar:Set/Reset Tab:Toggle Arr

textattr := lastattr;

gotoxy(19+index.x\*4,8+index.y\*2);

end;

procedure setarg(x,y:byte ; var arg:longint);

var ok : boolean;

chin : char;

cnt : byte;

argin : string[10];

code : integer;

oldarg: integer;

first : boolean;

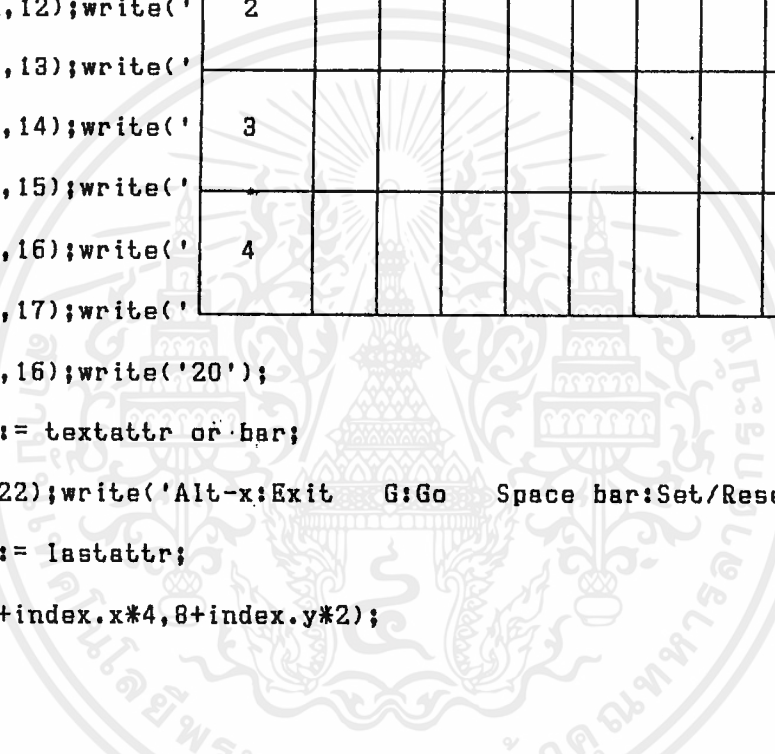
begin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ok := false; argin := ''; cnt := 0; first := true;  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Step Format								
Phase	1	2	3	4	5	6	7	8
1								
2								
3								
4								

Step Count

DELAYS



```
oldarg := arg;
gotoxy(x,y);write(' ');
str(arg,argin);
for cnt := 1 to ((10-length(argin)) div 2) do insert(' ',argin,1);
gotoxy(x,y);write(argin);
gotoxy(x,y);
argin := '';
repeat
  ch := readkey; { write(ord(ch)); }
  if ch in ['0'..'9'] then
    begin
      if first then
        begin
          gotoxy(x,y);write(' ');
          argin := '';
          first := false;
        end;
      argin := argin+ch;
      gotoxy(x,y);write(argin);
      cnt := cnt+1;
    end
  else if ch<>#9 then write(#7);
until (ch=#13) or (cnt > 6) or (ch=#9);
val(argin,arg,code);
if code <> 0 then
  begin
    arg := oldarg;
    str(arg,argin);
  end;
gotoxy(x,y);write(' ');
for cnt := 1 to ((10-length(argin)) div 2) do insert(' ',argin,1);
gotoxy(x,y);write(argin);
```

```
end;
```

```
procedure dspcnt;
```

```
var cntst : string[5];
```

```
    cntsetst : string[5];
```

```
    cnt : byte;
```

```
begin
```

```
    counter := counter+1;
```

```
    str(cntset, cntsetst);
```

```
    gotoxy(58,11);write(' ');
```

```
    for cnt := 1 to ((4-length(cntsetst)) div 2) do insert(' ', cntsetst, 1);
```

```
    gotoxy(58,11);write(cntsetst);
```

```
    gotoxy(62,11);write('//');
```

```
    if counter > 9999 then counter := 0;
```

```
    str(counter, cntst);
```

```
    gotoxy(63,11);write(' ');
```

```
    for cnt := 1 to ((4-length(cntst)) div 2) do insert(' ', cntst, 1);
```

```
    gotoxy(63,11);write(cntst);
```

```
    gotoxy(19+index.x*4, 8+index.y*2);
```

```
end;
```

```
procedure go;
```

```
var ok : boolean;
```

```
    ch : char;
```

```
    last: byte;
```

```
label finish ;
```

```
begin
```

```
    ok := false; curser(off);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
textattr := lastattr;
gotoxy(1,22);clreol;
textattr := bar;
gotoxy(30,22);write('Esc:Exit  Pause:Pause');
repeat
  while not keypressed do
    begin
      for cnt := 1 to 8 do
        begin
          last := cnt-1;
          if last < 1 then last := maxstp;
          setf := false;
          phaseslide(phase[last],last);
          port[port0] := phase[cnt];
          setf := true;
          phaseslide(phase[cnt],cnt);
          dspcnt;
          delay(speed);
        if cntset <> 0 then if cntset = counter then
          begin
            counter := 0;
            goto finish;
          end;
        end;
      end;
    if (readkey = #27) then ok := true;
  until ok;
finish:
  gotoxy(1,22);clreol;
  curser(on);
  textattr := textattr or bar;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
write('Alt-x:Exit G:Go Space bar:Set/Reset Tab:Toggle  
Arrow key:Move Curser');  
textattr := lastattr;
```

```
end;
```

```
procedure clear;
```

```
begin
```

```
counter := -1;
```

```
port[port0] := 0;
```

```
dspcnt;
```

```
end;
```

```
procedure select;
```

```
begin
```

```
setarg(58,11,cntset);
```

```
if ch=#13 then exit
```

```
else
```

```
setarg(58,16,speed);
```

```
end;
```

```
begin
```

```
clrscr;
```

```
init;
```

```
mkscr;
```

```
{ dspcnt;}
```

```
repeat
```

```
ch := readkey;
```

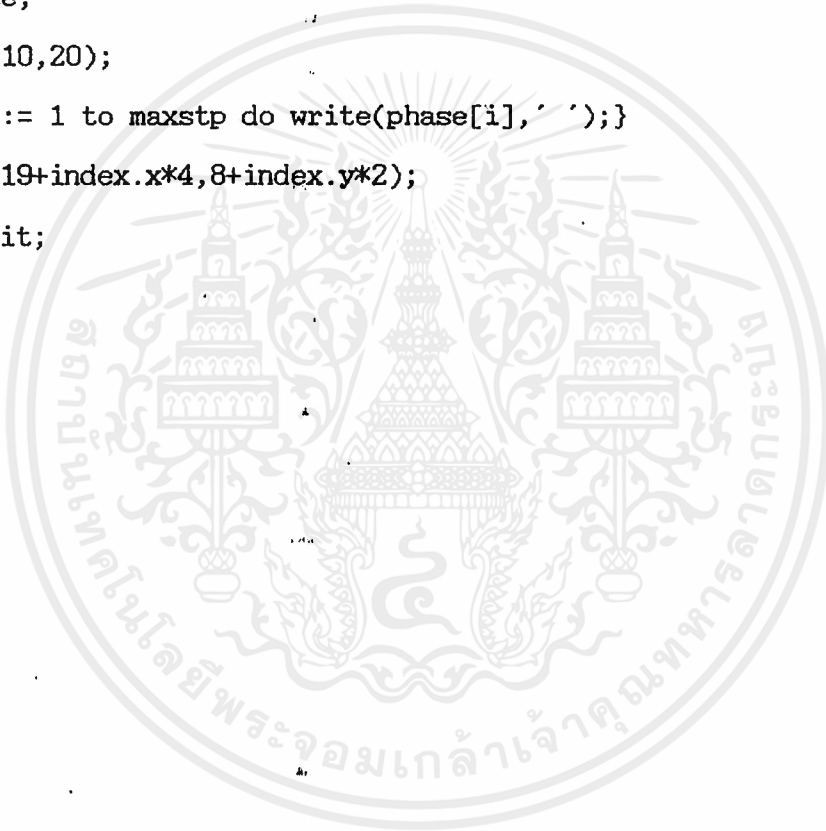
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{ gotoxy(50,3); write(ch,ord(ch));}

  case upcase(ch) of
    #0 : mvcurser;
    ' ' : setflag;
    #9 : select;
    'G' : go;
    'C' : clear;

    end;

  dspphase;
  gotoxy(10,20);
  { for i := 1 to maxstp do write(phase[i],' ');}
  gotoxy(19+index.x*4,8+index.y*2);
  until quit;
  clrscr;
end.
```



## กิตติกรรมประกาศ

### (ACKNOWLEDGMENT)

ความสำเร็จของโครงการนี้ได้ประสบความสำเร็จลุล่วงมาด้วยดีนั้น ก็ได้รับความรู้จาก การเรียนการสอนของอาจารย์ทุกท่าน ซึ่งประสิทธิ์ประสาทวิชาความรู้ให้ จึงขอขอบพระคุณเป็น ที่สุด โครงการนี้จะไม่สำเร็จลุล่วงไปได้ถ้าหากบุคคลดังต่อไปนี้

อาจารย์ภากร	หุตะสังกาศ	อาจารย์ที่ปรึกษา	ให้คำปรึกษาแนะนำ
อาจารย์มานพ	สุดสงวน	อาจารย์ประจำภาค ศอ.	เอื้อเฟื้อสถานที่
คุณศิริธ	เอื้อนุชานก	การบินไทย และ	
คุณสมชาย	เหลื่องพีระชัย	การบินไทย	ให้คำปรึกษาคำนับปรแกรม
คุณฉวีวรรณ	ภูวรัตนกุล	ภาค ศอ. และ	
คุณสุภาณี	สมปาน	บัณฑิตวิทยาลัย	ให้ข้อมูลการพิมพ์ บริษัฏพาณิชย์

และที่ขาดไปไม่ได้เลยคือเพื่อน ๆ ห้อง 2Q ทุกคน ที่ให้กำลังใจถามข่าวคราวความคืบหน้าของ โครงการ จึงขอขอบคุณทุกท่านมา ณ โอกาสนี้ด้วย

# คู่มือการใช้งาน

ชุดทดลอง STEPPING MOTOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# คู่มือการใช้งาน

ชุดทดลอง stepping motor controller ชุดนี้ใช้ program pascal เป็นตัว contpd ซึ่งมีขั้นตอนการใช้งานดังต่อไปนี้

1 เรียง file stp. exe จากไดรฟ์ดังนี้

c:\ STP.EXE

จะขึ้น menu ที่หน้าจอดังนี้

STEP								
PHASE	1	2	3	4	5	6	7	8
1								
2								
3								
4								

STEP COUNT
0

DELAY
200

ALT + X : EXIT G : GO SPACE BAR : SET/RESET

TAP : TOGGLE ARROW KEY : MOVE CURSOR

CURSOR จะอยู่ในตาราง assign phase ซึ่งเราสามารถที่จะ assign phase ของ stepping motor ได้ตามรูปแบบการป้อน phase ของ stepping motor โดยใช้ลูกศรเลื่อนไปตรงช่อง ต่าง ๆ ถ้าต้องการ assign ก็ใช้ SPACE BAR และถ้าจะยกเลิกการ assign ตรงนั้นก็ให้กด SPACE BAR อีกครั้ง

2. ตั้งค่า delay time โดยการกด key TAP cursor จะไปอยู่ในช่อง delay เราสามารถที่จะใส่ค่าได้ตั้งแต่ 0-9999999 แล้วตามด้วย key ENTER cursor จะกลับไปอยู่ในตาราง assign phase
3. กด key G program จะ RUN ทำให้ stepping motor หมุนไปตาม step ที่เรา assign ไว้ และ counter ก็จะเริ่มนับ step ไปเรื่อย ๆ ถ้าต้องการหยุดชั่วคราว ก็กด key PAUSE โปรแกรมจะหยุดชั่วคราว จนกว่าจะมีการกด key อื่น program จึงจะ run ต่อ และเมื่อต้องการออกจากการทำงานก็กด key ESC program จะไปอยู่ที่ main menu

key C จะเป็นการ clear counter  
ความถี่ของ step สามารถคำนวณได้ดังนี้

$$f = 1/\text{delay time} \quad (\text{Hz})$$

การคำนวณ องศา

$$\text{องศา} = 1.8 * \text{step} \quad (\text{degree})$$

(1.8 คือ องศาต่อ step ของ stepping motor ของชุดทดลองนี้)  
เมื่อต้องการออกจาก program (เลิกการทำงาน) ก็กด key

ALT + X โดยการ key ALT ค้างไว้แล้วกด key X ตาม เครื่องจะออกจาก program มาอยู่ใน dos

## ลำดับขั้นการทดลอง

run file STP.EXE บนเครื่อง micro computer แล้ว assign phase ของ stepping motor เป็น step ดังการทดลองต่อไปนี้

### 1. การทดลอง การกระตุ้นแบบเฟสเดียว (SINGLE-PHASE EXCITATION)

assign phase ของ stepping motor ทั้งหมดตามเข็มนาฬิกา (cw) และหมุนทวนเข็มนาฬิกา (ccw) ตามตารางที่ 1 และ 2 ตามลำดับ

		STEP							
PHASE	1	2	3	4	5	6	7	8	
1									
2									
3									
4									

ตารางที่ 1 ทิศทางตามเข็มนาฬิกา (cw)

STEP								
PHASE	1	2	3	4	5	6	7	8
1								
2								
3								
4								

ตารางที่ 2 ทิศทางทวนเข็มนาฬิกา (ccw)

2. การทดลอง การกระตุ้นแบบเฟสคู่ (TWO-PHASE EXCITATION)

assign phase ของ stepping motor ทั้งทิศทางหมุนตามเข็มนาฬิกา (cw) และ ทวนเข็มนาฬิกา (CCW) ตามตารางที่ 3 และ 4 ตามลำดับ

STEP								
PHASE	1	2	3	4	5	6	7	8
1								
2								
3								
4								

ตารางที่ 3 ทิศทางตามเข็มนาฬิกา (cw)

STEP								
PHASE	1	2	3	4	5	6	7	8
1								
2								
3								
4								

ตารางที่ 4 ทิศทางทวนเข็มนาฬิกา (ccw)

3. การทดลอง การกระตุ้นแบบกึ่งสเตป (HALF-STEP MODE)

assign phase ของ stepping motor ทั้งทิศทางหมุนตามเข็มนาฬิกา (cw) และทวนเข็มนาฬิกา (ccw) ตามตารางที่ 5 และ 6 ตามลำดับ

STEP								
PHASE	1	2	3	4	5	6	7	8
1								
2								
3								
4								

ตารางที่ 5 ทิศทางตามเข็มนาฬิกา (cw)

STEP								
PHASE	1	2	3	4	5	6	7	8
1								
2								
3								
4								

ตารางที่ 6 ทิศทางทวนเข็มนาฬิกา (ccw)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## คำถาม

1. stepping motor มีกี่ชนิดอะไรบ้างอธิบายพอสังเขป
2. การป้อน phase ของ stepping motor มีกี่แบบ แต่ละแบบต่างกันอย่างไร และมีผลอะไรกับแรงบิด (torque) หรือไม่
3. stepping motor ที่ใช้ในการทดลองนี้จะหยุดหมุนที่ความถี่เท่าไร และทำไมเมื่อความถี่ของแต่ละ step สูงขึ้นถึงจุดจุดหนึ่ง stepping motor จึงหยุดทำงาน
4. จงยกตัวอย่างการประยุกต์ใช้งาน stepping motor กับงานอุตสาหกรรม



\*\*\*\*\*