



เครื่องกัดแผ่นวงจรพิมพ์อัตโนมัติ

PRINT CIRCUIT BOARD AUTOMATIC CUTTING MACHINE

โดย

นายก่อเกียรติ วรรณนิกุล

นายเจษฎา จินดารัตน์

นายศักดิ์ วันงาม

วัน เดือน ปี..... 11. ๑๑. 2541
เลขทะเบียน..... 0.38.914
เลขเรียกหนังสือ... T.10154 ก๒๕๔๑

- ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

สาขาเทคโนโลยีอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

๓
๒

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใด
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

038914

หัวข้อปริญญานิพนธ์

เครื่องกักแผ่นวงจรพิมพ์อัตโนมัติ

โดย

นายก่อเกียรติ	วรรณนิกุล	เลขประจำตัว	39012002
นายเจษฎา	จินดารัตน์	เลขประจำตัว	39012005
นายภักดี	วันงาม	เลขประจำตัว	39012021

อาจารย์ที่ปรึกษา

อาจารย์ไพศาล

สิทธิโยภาสกุล

อาจารย์মনชนก

ศรีเสื่อขาม

ภาควิชา

เทคนิคอุตสาหกรรม

ปีการศึกษา

2540

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังอนุมัติให้นับ
ปริญญานิพนธ์ฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

คณะกรรมการสอบปริญญานิพนธ์

.....ประธานกรรมการ

(.....)

.....กรรมการ

(.....)

.....กรรมการ

(.....)

.....กรรมการ

(.....)

.....กรรมการ

(.....)

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องกัดแผ่นวงจรพิมพ์อัตโนมัติ

PRINT CIRCUIT BOARD AUTOMATIC CUTTING MACHINE

- โดย 1. นายก่อเกียรติ วรรณนิกุล
2. นายเจษฎา จินคาร์คณ์
3. นายภักดี วันงาม

อาจารย์ที่ปรึกษา อาจารย์ไพศาล สิทธิโยภาสกุล
อาจารย์মনชนก ศรีเสือขาม

บทคัดย่อ

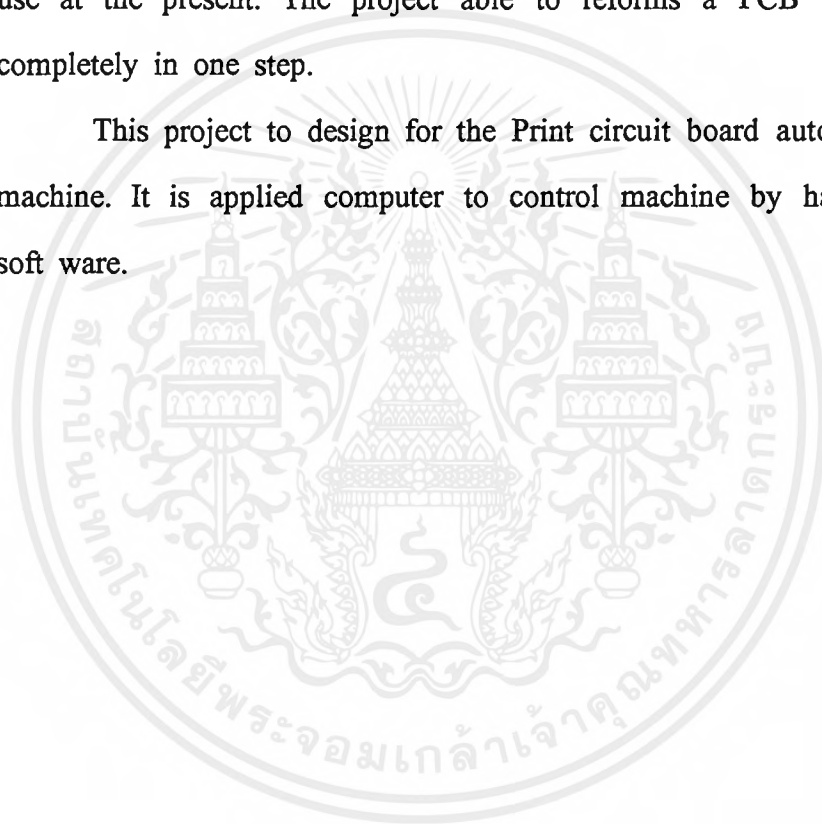
ในปัจจุบันคอมพิวเตอร์ได้เข้ามามีบทบาทในการทำกิจกรรมต่าง ๆ มากมาย โครงการงานชุดนี้จึงได้ก่อกำเนิดขึ้นเพื่อจุดประสงค์ในการพัฒนารูปแบบการทำแผ่นวงจร PCB ให้มีความสะดวกและลดข้อจำกัดบางประการลงโดย ได้มีการพัฒนาและดัดแปลงมาจากเครื่อง พลัสเตอร์ที่ใช้กันอยู่ในปัจจุบัน ซึ่งโครงการงานชุดนี้สามารถทำแผ่นวงจร PCB ให้เสร็จสมบูรณ์พร้อมใช้งานได้ทันที

ปริญญานิพนธ์ฉบับนี้เป็นการออกแบบประยุกต์การใช้เครื่องคอมพิวเตอร์ไปควบคุมกลไก ต่าง ๆ เพื่อสร้างเป็นเครื่องกัดแผ่นวงจรพิมพ์อัตโนมัติขึ้น โดยมีทั้งอุปกรณ์ทาง Software และ Hard ware ทำงานร่วมกัน

Abstract

At the present, computer has been more using or participating in any control system. This project present concern a PCB board pattern development for able to work in facility and reduce the process of production. In which develop and modify from ploter machine that they use at the present. The project able to reforms a PCB board operate completely in one step.

This project to design for the Print circuit board automatic cutting machine. It is applied computer to control machine by hard ware and soft ware.



กิตติกรรมประกาศ

ปริชญานิพนธ์ฉบับนี้ สำเร็จลงด้วยความเรียบร้อย ด้วยการให้การสนับสนุนจากผู้มี
พระคุณ ดังต่อไปนี้

1. อาจารย์ไพศาล ลิทธิโยภาสกุล และ อาจารย์মনชนก ศรีเสือขาม
2. พี่ๆ เพื่อนๆ และน้องๆ ทุกคนที่ได้ให้การช่วยเหลือ

ทางผู้จัดทำขอขอบคุณมา ณ. ที่นี้ด้วย

อนึ่งผู้จัดทำขอขอบคุณงามความดี ทั้งหลายจากปริชญานิพนธ์ฉบับนี้ แก่ บิดา มารดา
ครูบาอาจารย์ และผู้มีพระคุณทุกท่านที่ไม่ได้กล่าวถึง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1	
1.1 บทนำ	1
1.2 เป้าหมายในการวิจัยโครงการ	2
บทที่ 2	
2.1 โครงสร้างและการทำงานสแต็ปป์มอเตอร์	3
2.2 การกระตุ้นเฟสขดลวดสเตเตอร์	5
2.3 การเกิดออสซิลเลตของสแต็ปป์มอเตอร์	7
2.4 กราฟิกพล็อตเตอร์	8
บทที่ 3	
3.1 หลักการทำงานของเครื่องกักแผ่นวงจรพิมพ์อัตโนมัติ	14
3.2 วงจรควบคุมสแต็ปป์มอเตอร์	16
3.3 โครงสร้างทางด้านกลไกของปริ๊นตานิพนธ์	22
3.4 การแปลงข้อมูลจากโปรแกรม Protel	34
บทที่ 4	
4.1 ผลการทดลองและการทดสอบโครงการ	35
4.2 คุณลักษณะเฉพาะ	36
4.3 การใช้งาน	36
บทที่ 5	
5.1 สรุปผลการดำเนินงาน	38
5.2 ปัญหาที่เกิดจากการทดลอง	38
5.3 การแก้ไขปัญหา	39
5.4 การวิเคราะห์และแนวทางการพัฒนาโครงการ	39
บรรณานุกรม	40
ภาคผนวก	41

บทที่ 1

บทนำ

1.1 บทนำ

ตั้งแต่อดีตจนถึงปัจจุบันการทำแผ่นวงจรพิมพ์ (Print Circuit Board ; PCB) ได้มีรูปแบบและขั้นตอนการทำหลายรูปแบบ ซึ่งแตกต่างกันไปเพื่อให้มีความสะดวกและรวดเร็วมากยิ่งขึ้น แต่ในแต่ละรูปแบบก็ยังติดข้อจำกัดบางประการอยู่

โครงการชุดนี้จึงได้ก่อกำเนิดขึ้นเพื่อจุดประสงค์ในการพัฒนารูปแบบการทำแผ่นวงจร PCB ให้มีความสะดวกและลดข้อจำกัดบางประการลง โดยโครงการชุดนี้ได้พัฒนาและคิดแปลงมาจากเครื่องพล็อตเตอร์ที่ใช้กันอยู่ในปัจจุบัน ซึ่งโครงการชุดนี้สามารถทำแผ่นวงจร PCB ให้เสร็จสมบูรณ์พร้อมใช้งานได้ทันที

ปริญญานิพนธ์ฉบับนี้ จึงเป็นการรวบรวมเอาแนวคิดหลักการทำงานและทฤษฎีต่างๆ ที่เกี่ยวข้องกับโครงการเอาไว้เพื่อเป็นแนวทางสำหรับการพัฒนาให้มีความสมบูรณ์และสะดวกมากยิ่งขึ้น

1.2 เป้าหมายในการวิจัยโครงการ

สำหรับการวิจัยโครงการนี้ได้กำหนดเป้าหมายไว้ว่า ทำการออกแบบและสร้างเครื่องที่เป็นต้นแบบขึ้น ซึ่งถูกควบคุมการทำงานโดยใช้เครื่องคอมพิวเตอร์ โดยการ Interface ควบคุมส่วนเคลื่อนที่ต่าง ๆ อันได้แก่ ส่วนควบคุมตำแหน่งแกน x และแกน y ส่วนควบคุมการขึ้นลงของ หัวกัศและเจาะ รวมถึงการหมุนของหัวกัศด้วย

ในส่วนควบคุมตำแหน่งแกน x และแกน y ใช้ Stepping Motor จำนวน 2 ตัวควบคุมการเคลื่อนที่ไปในแนวแกน x และแกน y ทำให้สามารถควบคุมตำแหน่งได้คงที่และมีความผิดพลาดน้อยมาก แต่ในส่วนของหัวกัศจะใช้เป็น DC Motor ที่มีความเร็วรอบสูงขับเคลื่อนหัวกัศ

การควบคุมการใช้ร่วมกับโปรแกรม Protel โดยการออกแบบแผนวงจรพิมพ์และผ่านโปรแกรมควบคุมการทำงานออกมาทาง Port Print



บทที่ 2

ทฤษฎี

2.1 โครงสร้างและการทำงาน สเต็ปป์มอเตอร์

ภายในสเต็ปป์มอเตอร์ประกอบด้วย สเตเตอร์, โรเตอร์ และขดลวดประกอบเข้าด้วยกัน ดังรูปที่ 1 (สมมติเป็นมอเตอร์แบบ 3 เฟส)

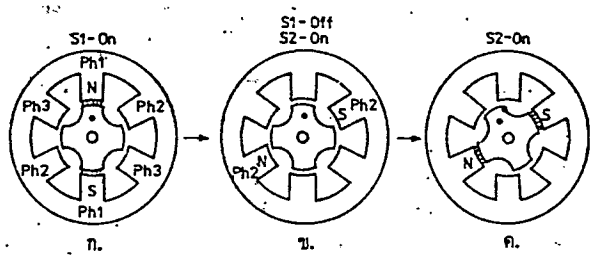


รูปที่ 1 ภาพหน้าตัดของสเต็ปป์มอเตอร์ รูปที่ 2 เส้นแรงแม่เหล็กที่ทำให้เกิดแรงบิดแบบ 3 เฟส

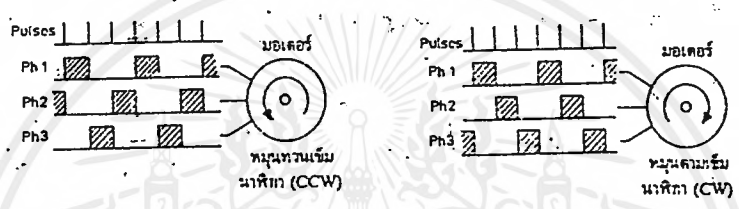
เนื่องจากสเต็ปป์มอเตอร์นี้ โรเตอร์เป็นเหล็กอ่อน ซึ่งมีคุณสมบัติพยายามปรับตัวเองให้อยู่ในแนวที่เส้นแรงแม่เหล็กผ่านมากที่สุดดังรูปที่ 2 เมื่อเกิดเส้นแรงแม่เหล็กขึ้นที่สเตเตอร์ตัดผ่านโรเตอร์ ตัวโรเตอร์ก็จะพยายามปรับตัวเองให้เส้นแรงแม่เหล็กตัดผ่านตัวเองมากที่สุดโดยการหมุนตัวเองทำให้เกิดมุมของการหมุนขึ้น และมอเตอร์จะหยุดหมุนเมื่อเส้นแรงแม่เหล็กที่ตัดผ่านตัวมันถึงจุดมากที่สุด การทำให้สเต็ปป์มอเตอร์ หมุนก็ทำได้โดยอาศัยหลักการนี้ แต่ต้องให้เส้นแรงแม่เหล็กเกิดขึ้น โดยรับช่วงต่อกันไปเรื่อย ๆ ดังรูปที่ 3ก., ข. และ ค. ซึ่งแสดงถึงการหมุนของมอเตอร์ โดยทิศทางขึ้นอยู่กับการขับกระแสเข้าขดลวดว่าจะให้ไปทางไหน และเมื่อต้องการให้มอเตอร์หยุดก็หยุดการขับมอเตอร์ มอเตอร์ก็จะหยุด ณ ตำแหน่งสุดท้ายที่มีการขับที่ สเตเตอร์ ดังนั้นเราจึงสามารถรู้ตำแหน่งของมอเตอร์ได้ โดยการนับจำนวนพัลส์ที่ป้อนให้มอเตอร์โดยใช้สูตร

$$\text{มุมที่เปลี่ยนไป} = \text{ค่ามุมต่อสเต็ป} \times \text{จำนวนพัลส์ที่ป้อนให้}$$

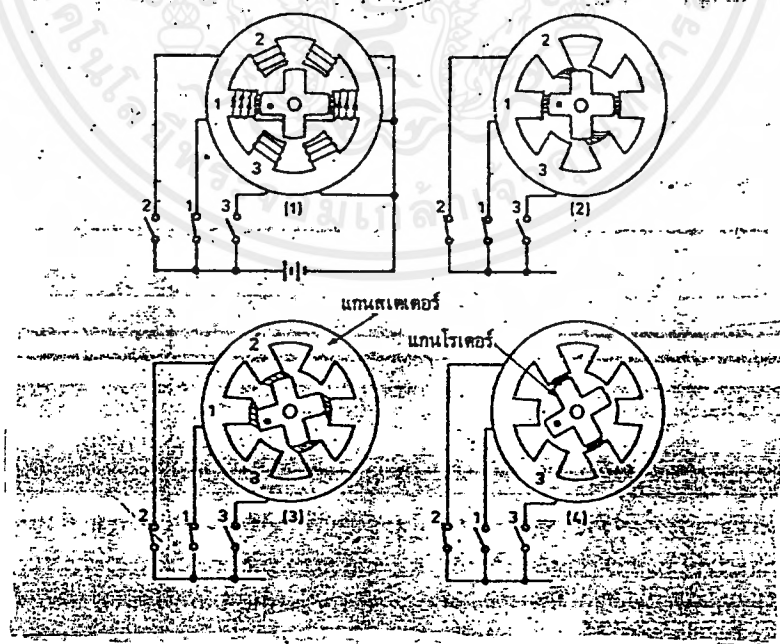
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3 แสดงการเคลื่อนที่ทีละสเต็ป เมื่อกระตุ้นเฟส 1 เฟส 2



รูปที่ 4 แสดงการกระตุ้น แบบเดินหน้าและถอยหลัง



รูปที่ 5 แสดงการทำงานของสเต็ปิ่งมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การกระตุ้นเฟสของลวดสเตเตอร์

ดังที่รู้กันแล้วว่าการทำให้สเต็ปิ้งมอเตอร์หมุนได้นั้น จะต้องกระตุ้นเฟสของลวดสเตเตอร์ให้เรียงกันไปเรื่อยๆทางใดทางหนึ่ง ถ้าต้องการให้หมุนกลับก็กระตุ้นเฟสในทิศทางกลับกัน ซึ่งการกระตุ้นเฟสของสเตเตอร์มีอยู่ 3 แบบ คือ

1. การกระตุ้นเฟสเดียวเรียกว่า single phase excitation

สัญญาณนาฬิกา	R	1	2	3	4	5	6	7	8
เฟสที่ 1	■			■			■		
เฟสที่ 2		■			■			■	
เฟสที่ 3			■			■			■

รูปที่ 6 แสดงการกระตุ้นแบบเฟสเดียว

2. การกระตุ้นสองเฟสเรียกว่า two phase excitation

สัญญาณนาฬิกา	R	1	2	3	4	5	6	7	8
เฟสที่ 1	■			■					
เฟสที่ 2		■			■			■	
เฟสที่ 3			■			■			■

รูปที่ 7 แสดงการกระตุ้นแบบสองเฟส

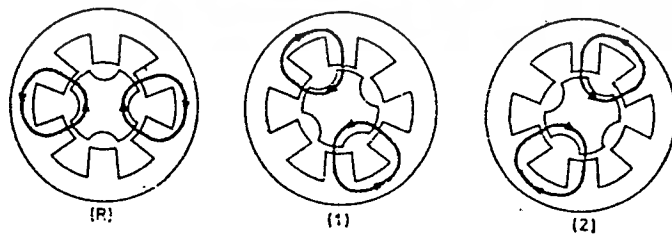
3. การกระตุ้นโดยใช้แบบ 1 และ 2 สลับกันเรียกว่าแบบ one - two phase excitation
หรือแบบ half step operation

สัญญาณเฟสที่	R	1	2	3	4	5	6	7	8
เฟสที่ 1									
เฟสที่ 2									
เฟสที่ 3									

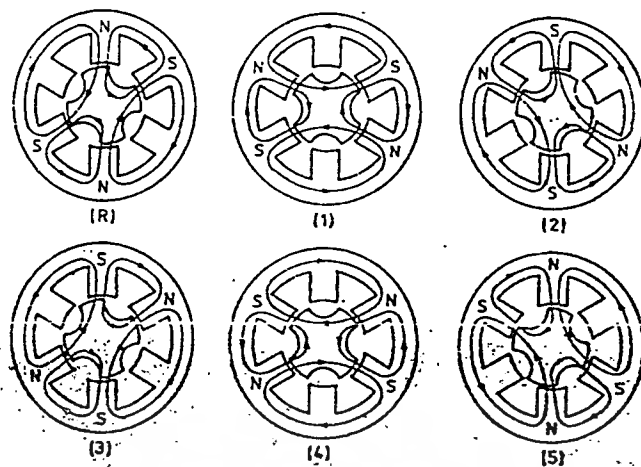
รูปที่ 8 แสดงการกระตุ้นแบบครึ่งสเต็ป

ในการขับแบบกระตุ้นสองเฟส เส้นแรงแม่เหล็กจะไม่ผ่านแกนเหล็กเป็นเส้นตรงเลยที่เดียวเหมือนกระตุ้นเฟสเดียวแต่จะวกกลับมาเข้าสู่แกนทางด้านข้าง ๆ ดังรูปที่ 9 และเส้นแรงแม่เหล็กส่วนหนึ่งจะมาจากแกนตรงข้าม ดังรูปที่ 10

การขับสเต็ปปึงมอเตอร์แบบกระตุ้นสองเฟสนี้จะมีลักษณะเดียวกับ การขับแบบกระตุ้นเฟสเดียว แต่ในการกระตุ้นแต่ละครั้งนั้นจะกระตุ้นที่เดียวพร้อมกัน 2 เฟส จะต่างก็ตรงที่ การขับแบบ 2 เฟส เข้าตำแหน่งแต่ละสเต็ป ได้เร็วกว่าแบบเฟสเดียว และแรงบิดมากกว่าการขับแบบเฟสเดียวด้วย



รูปที่ 9 เส้นแรงแม่เหล็กแบบขับ 2 เฟส



รูปที่ 10 ลักษณะเส้นแรงแม่เหล็กเมื่อขับแบบ 2 เฟส

2.3 การเกิดออสซิลเลตของสเต็ปิ่งมอเตอร์

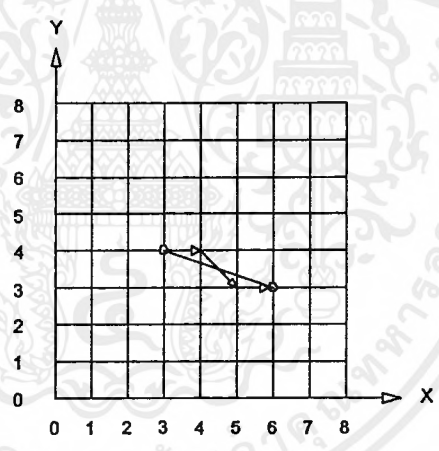
ความเร็วของการหมุนของสเต็ปิ่งมอเตอร์เป็น ติเนียร์กับความถี่ที่ป้อนให้สเต็ปิ่งมอเตอร์ แต่เมื่อเราป้อนความถี่เพิ่มขึ้นเรื่อย ๆ จนถึงความถี่ค่าหนึ่ง สเต็ปิ่งมอเตอร์ก็จะหยุดหมุน เนื่องจากการที่โรเตอร์หมุนตามฟลักซ์แม่เหล็กไม่ทัน เราเรียกว่า มอเตอร์เกิดการออสซิลเลตซึ่งมอเตอร์แต่ละตัว จะออสซิลเลตที่ความถี่ต่างกันไป โดยทั่วไปจะออสซิลเลตที่ความถี่ประมาณ 500 Hz ในการขับแบบกระตุ้น 1 หรือ 2 เฟส และจะมีความถี่ประมาณ 1 KHz เมื่อขับแบบครึ่งสเต็ป แต่เมื่อลดความถี่ให้ต่ำมอเตอร์จะไม่หมุนทันที และเมื่อความถี่ลดลงจนถึงความถี่หนึ่งมอเตอร์จึงจะเริ่มหมุนอีกครั้ง นั่นก็คือมอเตอร์มีฮิสเตอร์รีซิส ซึ่งมอเตอร์จะเริ่มหมุนที่ความถี่ประมาณ 200 Hz ในการขับแบบ 1 หรือ 2 เฟสและมีความถี่ประมาณ 150 Hz เมื่อขับแบบครึ่งสเต็ป

2.4 กราฟิกพล็อตเตอร์ (GRAPHIC PLOTTER)

2.4.1 หลักการทำงาน

พล็อตเตอร์เป็นเครื่องมือที่ใช้ในการวาดภาพที่ได้จากการเขียนด้วยโปรแกรมคอมพิวเตอร์ ด้านกราฟิก ทำงานด้วยคำสั่งที่ส่งมาจากคอมพิวเตอร์ คำสั่งที่ส่งมาจะเป็นข้อมูลเกี่ยวกับการกำหนดตำแหน่ง (coordinate) คำสั่งวางปากกา ยกปากกา เพื่อลากเส้นจากจุดหนึ่งไปอีจุดหนึ่ง ถ้าเป็นคำสั่งให้เขียนวงกลมหรือส่วนของวงกลมก็จะกำหนดครัมมี กำหนดจุดศูนย์กลางกำหนดรายละเอียดของเส้นรอบวง และกำหนดคองสาของเส้นรอบวง พล็อตเตอร์ จะมีไมโครคอนโทรลเลอร์คอยรับคำสั่งจากคอมพิวเตอร์เพื่อนำมาคำนวณหาตำแหน่งที่จะลากเส้น แล้วส่งสัญญาณให้วงจรมอเตอร์ หมุนมอเตอร์ไปตามจุดต่างๆ ที่คำนวณไว้

2.4.2 การเคลื่อนที่ของพล็อตเตอร์



รูปที่ 11 แสดงลักษณะการเคลื่อนที่ของพล็อตเตอร์

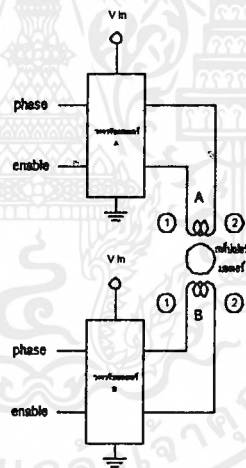
เริ่มแรกต้องกำหนดพื้นที่ของกระดาษให้เป็นจุด ๆ ดังรูปที่ 11 โดยกำหนดเป็นแกน X และ Y เริ่มจากจุด X_0, Y_0 เลื่อนไปทางขวาเป็นแกน X เลื่อนขึ้นบนเป็นแกน Y เส้นแบ่งในแกน X และ Y หมายถึง สเต็ปที่ละเอียดที่สุดของเครื่อง จากรูปที่ 11 สมมติว่ามีคำสั่งให้ลากเส้นจากจุด A ไปจุด B โดยจุด B มีพิกัดที่ X_3, Y_4 และจุด B มีพิกัดที่ X_6, Y_3 เครื่องจะคำนวณหาทางเดินจากจุด A ไปจุด B ทีละสเต็ป เริ่มจากพิกัด X_3, Y_4 ไปยังพิกัด X_4, Y_4 โดยเดินมอเตอร์ไปตามแกน X หนึ่งสเต็ป จากนั้นก็เคลื่อนไปยังจุด X_5, Y_3 โดยเดินมอเตอร์ไปตามแกน X และ Y พร้อมกัน สุดท้ายจะเคลื่อนไปยังจุด X_6, Y_3 โดยเดินมอเตอร์ไปตามแกน X เพียงอย่างเดียวจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เห็นว่าเครื่องไม่สามารถเคลื่อนที่ไปตามเส้นประได้ เนื่องจากต้องเคลื่อนที่เป็นสเต็ปๆ ไป ฉะนั้น ถ้าความละเอียดของสเต็ปยิ่งมากเท่าไร เส้นก็จะดูเป็นเส้นตรงมากขึ้นเท่านั้น

พล็อตเตอร์มาตรฐานทั่วไปความละเอียดจะเท่ากับ 25 ไมครอน (ไมโครเมตร) หรือ 1/1016 นิ้ว พล็อตเตอร์เครื่องนี้ใช้สเต็ปมอเตอร์แบบเคลื่อนที่ 1.8 องศาต่อสเต็ป ในหนึ่งรอบของมอเตอร์มีการเคลื่อนที่ 200 สเต็ป (360 องศา หารด้วย 1.8) และถ้าเคลื่อนที่แบบครึ่งสเต็ปจะเคลื่อนที่ได้ 400 สเต็ปต่อรอบ ฉะนั้นในระยะทาง 1 นิ้ว มอเตอร์จะต้องหมุนถึงสองรอบครึ่ง จึงจะได้จำนวนสเต็ปเท่ามาตรฐาน ซึ่งในทางปฏิบัติภาคกลไกทำได้ยาก ถ้าใช้มอเตอร์ที่มีความละเอียดสูงกว่านี้ ก็จะหามอเตอร์นั้นมีการเคลื่อนที่ที่ละเอียดมากขึ้น โดยการใช้วงจรขับมอเตอร์แบบแบ่งสเต็ปแบบละเอียด (micro step driver)

2.4.3 การขับมอเตอร์



รูปที่ 12 วงจรขับสเต็ปเปอร์มอเตอร์เบื้องต้น

การขับมอเตอร์สเต็ปมอเตอร์ในพล็อตเตอร์นี้ ใช้วงจรขับมอเตอร์แบบแบ่งสเต็ป ละเอียดจากรูปที่ 12 ใช้วงจรขับมอเตอร์ 2 วงจร ที่มีคุณสมบัติเหมือนกันทุกประการ เพื่อขับมอเตอร์หนึ่งตัว ขา PHASE เป็นตัวกำหนดทิศทาง การไหลของกระแสในขดลวด ขา ANABLE ใช้ควบคุมให้เกิดกระแสไหลในขดลวด ซึ่งความสัมพันธ์ของขาทั้งสอง กับกระแสในขดลวด แสดงดังในตารางที่ 1 ในการขับมอเตอร์แบบเต็มสเต็ป (full step driver) นั้น มีหลักการง่าย ๆ คือ ให้มีกระแสไหลในขดลวด 2 ขดสลับกัน ซึ่งสามารถสลับได้ 4 แบบดังแสดงในตารางที่ 2

Enable	Phase	กระแสในขดลวด A และ B
0	X	ไม่มีกระแสไหล
1	0	กระแสไหลจากตำแหน่ง 1 ไป 2
1	1	กระแสไหลจากตำแหน่ง 2 ไป 1

ตารางที่ 1 ความสัมพันธ์ของขา ENABLE และ PHASE กับการไหลของกระแสในขดลวดของสเต็ปเปอร์มอเตอร์

สเต็ป	Enable A	Enable B	Phase A	Phase B	กระแสไหลในขดลวด A	กระแสไหลในขดลวด B
1	1	1	0	0	กระแสไหลจาก 1 ไป 2	กระแสไหลจาก 1 ไป 2
2	1	1	1	0	กระแสไหลจาก 2 ไป 1	กระแสไหลจาก 1 ไป 2
3	1	1	1	0	กระแสไหลจาก 2 ไป 1	กระแสไหลจาก 2 ไป 1
4	1	1	0	1	กระแสไหลจาก 1 ไป 2	กระแสไหลจาก 2 ไป 1

ตารางที่ 2 แสดงการไหลของกระแสในขดลวดของสเต็ปเปอร์มอเตอร์ เมื่อถูกขับให้หมุนแบบเต็มสเต็ป

ถ้าต้องการขับให้มอเตอร์หมุนแบบครึ่งสเต็ป (Half step driver) ก็กำหนดให้มีการหยุดไหลของกระแสในขดลวด A และ B คั่นสลับอยู่ในลักษณะการหมุนแบบเต็มสเต็ป ก็จะได้การหมุนของมอเตอร์แบบครึ่งสเต็ปดังในตารางที่ 3

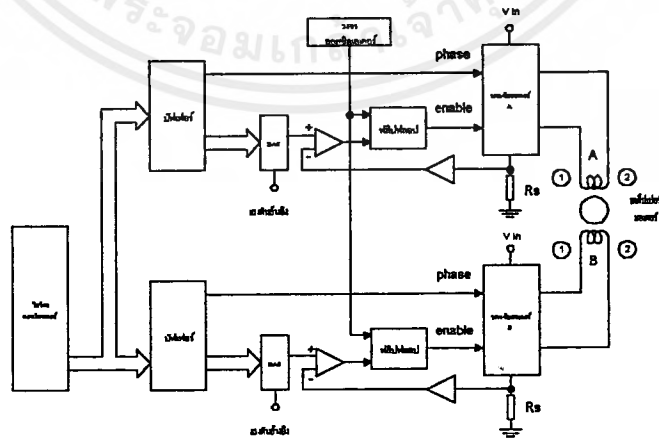
จะเห็นว่าสามารถทำให้มอเตอร์เคลื่อนที่ได้ละเอียดขึ้นอีกเท่าตัวเมื่อควบคุมกระแสในขดลวดได้จากแนวคิดอันนี้จึงนำมาพัฒนาเป็นวงจรขับมอเตอร์แบบแบ่งละเอียด ซึ่งสามารถทำให้

สตีปมอเตอร์ธรรมดาเปลี่ยนเป็นสตีปเปอร์มอเตอร์แบบแบ่งละเอียดสูง โดยสามารถทำให้ละเอียดเท่าไรก็ได้ตามความต้องการ ในโครงการนี้จะกำหนดไว้ 1600 สตีปต่อรอบ

สตีป	Enable A	Enable B	Phase A	Phase B	กระแสไหลในขดลวด A	กระแสไหลในขดลวด B
1	1	1	0	0	กระแสไหลจาก 1 ไป 2	กระแสไหลจาก 1 ไป 2
2	1	0	1	0	กระแสไหลจาก 1 ไป 2	ไม่มีกระแสไหล
3	1	1	1	1	กระแสไหลจาก 2 ไป 1	กระแสไหลจาก 1 ไป 2
4	0	1	0	1	ไม่มีกระแสไหล	กระแสไหลจาก 1 ไป 2
5	1	1	1	1	กระแสไหลจาก 2 ไป 1	กระแสไหลจาก 2 ไป 1
6	1	0	1	1	กระแสไหลจาก 2 ไป 1	ไม่มีกระแสไหล
7	1	1		1	กระแสไหลจาก 1 ไป 2	กระแสไหลจาก 2 ไป 1
8	0	1	0	1	ไม่มีกระแสไหล	กระแสไหลจาก 2 ไป 1

ตารางที่ 3 แสดงการไหลของกระแสในขดลวดของสตีปเปอร์มอเตอร์ เมื่อหมุนแบบครึ่งสตีป

2.4.4 วงจรขับมอเตอร์แบบแบ่งละเอียด

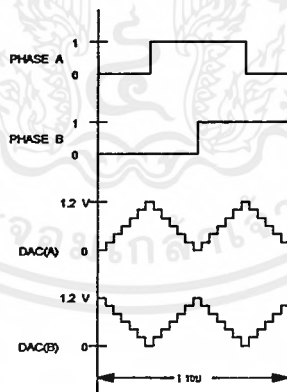


รูปที่ 13 บล็อกไดอะแกรมขับสตีปเปอร์มอเตอร์ในพรีดเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 13 เป็นบล็อกโคโอะแกรมของวงจรขับมอเตอร์แบบแบ่งละเอียดที่ภาคขับมอเตอร์ จะมีตัวต้านทานต่อลงกราวด์อยู่ภาคละหนึ่งตัวให้ชื่อว่า R_s กระแสที่จะไหลผ่านขดลวดต้องไหลผ่าน R_s ก่อนลงกราวด์ ฟลิปฟลอปทำหน้าที่ควบคุมการทำงานของภาคขับมอเตอร์ (ซึ่งก็คือทำหน้าที่ส่งสัญญาณ enable ให้แก่ภาคขับมอเตอร์นั่นเอง) ฟลิป ฟลอปจะทำงานตามสัญญาณนาฬิกาที่ป้อนเข้ามา มีวงจรเปรียบเทียบแรงดันอยู่หน้าฟลิปฟลอป ซึ่งได้รับอินพุต 2 ทาง ทางหนึ่งจากภาคเปลี่ยนสัญญาณดิจิทัลเป็นอนาล็อก (DAC) อีกทางมาจากวงจรขยายแรงดันที่ได้มาจากแรงดันตกคร่อมตัวต้านทาง R_s ภาค DAC ถูกควบคุมด้วยไมโครโพรเซสเซอร์ โดยผ่านทางบัพเฟอร์ส่วนขา PHASE ต่อเข้ากับบัพเฟอร์โดยตรง

จากจุดมุ่งหมายเดิมคือ ต้องควบคุมกระแสในขดลวดให้ได้ ในการควบคุมแบบหยุดกระแสในขดลวด A และ B สลับกันในตำแหน่งที่เหมาะสม แทรกอยู่ในการหมุนแบบเต็มสเต็ป แต่ในวงจรขับมอเตอร์แบบแบ่งละเอียดต้องทำให้การไหลของกระแสในขดลวด A และ B ให้ความลดหล่นกันคือ ไม่ถึงกับหยุดไปเลยแต่ลดหล่นกันเป็นลำดับแทรกอยู่ในช่วงหมุนแบบเต็มสเต็ปเหมือนเดิม



รูปที่ 14 ความสัมพันธ์ของสัญญาณที่ขา phase กับสัญญาณที่ได้จากวงจร DAC

ทั้งสองชุด (A และ B)

วิธีการนี้ได้จากการที่กระแสของขดลวดไหลผ่าน R_s จะเกิดแรงดันตกคร่อม R_s แล้วถูกขยายก่อนที่จะส่งไปยังขดลวดของวงจรเปรียบเทียบแรงดัน แรงดันที่ขดลวดของวงจรเปรียบเทียบแรงดันนี้เป็นเอกสารที่สงวนไว้สำหรับวารสารเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แรงดันต้องเท่ากับแรงดันที่ขาบวกจะมากกว่าไม่ได้ ถ้ามากกว่าวงจรเปรียบเทียบกับแรงดันจะทำให้ฟลิป ฟลอปหยุดทำงาน กระแสในขดลวดหยุดไหล ดังนั้นกระแสในขดลวดจึงถูกควบคุมด้วยแรงดันที่ขาบวกของวงจรเปรียบเทียบกับที่ต่ออยู่กับภาค DAC ซึ่งถูกควบคุมด้วยไมโครคอนโทรเลอร์ในขั้นสุดท้าย

สรุปก็คือ การทำงานทุกอย่างจะถูกควบคุมด้วยซอฟต์แวร์ ซอฟต์แวร์นี้จะทำการควบคุมให้ขา PHASE A และ B ทำงานสลับกัน 4 ครั้ง ในลักษณะของเต็มสเต็ป และส่งข้อมูล 7 บิตให้ภาค DAC ทำการแปลงสัญญาณจะได้สัญญาณเป็นขั้นบันไดรูปสามเหลี่ยม มีเฟสต่างกัน 180 องศาตามรูปที่ 14



บทที่ 3

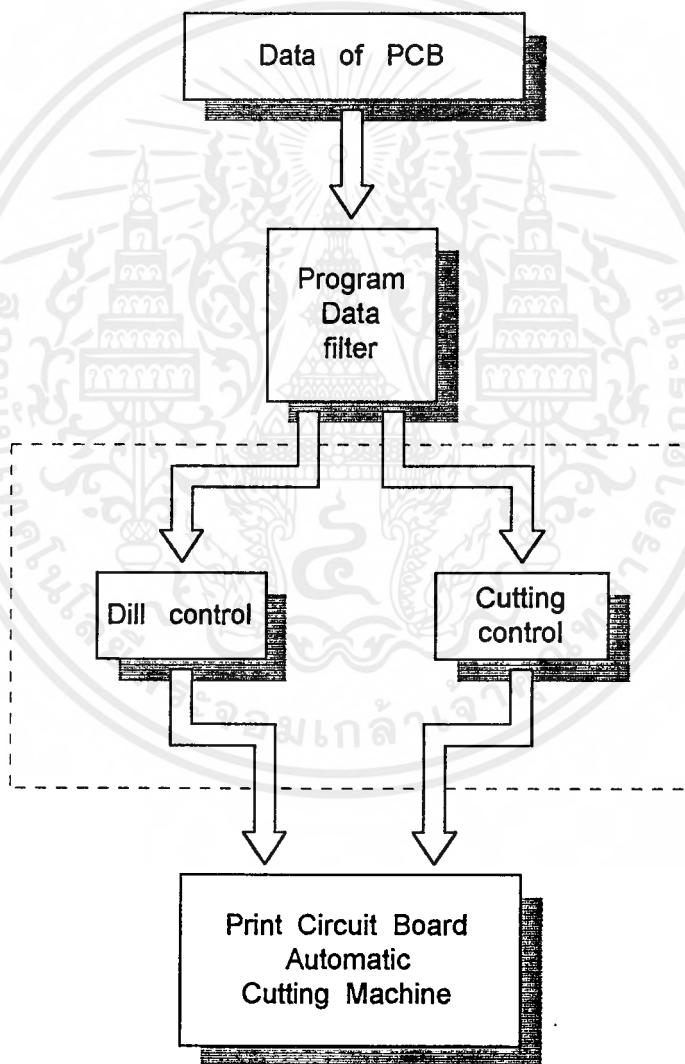
เครื่องกัดแผ่นวงจรพิมพ์อัตโนมัติ

3.1 หลักการทำงานของเครื่องกัดแผ่นวงจรพิมพ์อัตโนมัติ

เครื่องกัดแผ่นวงจรพิมพ์อัตโนมัติเป็นอุปกรณ์ที่ถูกสร้างขึ้นมาเพื่อเพิ่มความสะดวกระบาย ในการทำแผ่นวงจรพิมพ์โดยจะสามารถลดค้ข้อย่างยากต่างๆ ในการทำแผ่นวงจรพิมพ์ โดยอาศัย เครื่องคอมพิวเตอร์เข้ามามีส่วนร่วมในการทำงานของเครื่องนี้

หลักการทำงานจะเป็นการประยุกต์การทำงานของอุปกรณ์ชนิดหนึ่งซึ่งเรียกกันโดยทั่วไปว่าเครื่องพล็อตเตอร์ซึ่งปกติพล็อตเตอร์จะมีหน้าที่ในการวาดรูปจากการสร้างงานบนเครื่องคอมพิวเตอร์ แต่เครื่องมือในโครงการนี้ จะมีการเปลี่ยนแปลงอุปกรณ์บางชิ้นของเครื่องพล็อตเตอร์ โดยจะทำการเปลี่ยนอุปกรณ์ที่ใช้ในการวาดภาพนั้น ให้ไปเป็นชุดอุปกรณ์หัวกัดที่จะสามารถขูดเนื้อทองแดงบนแผ่นวงจรพิมพ์ออกได้

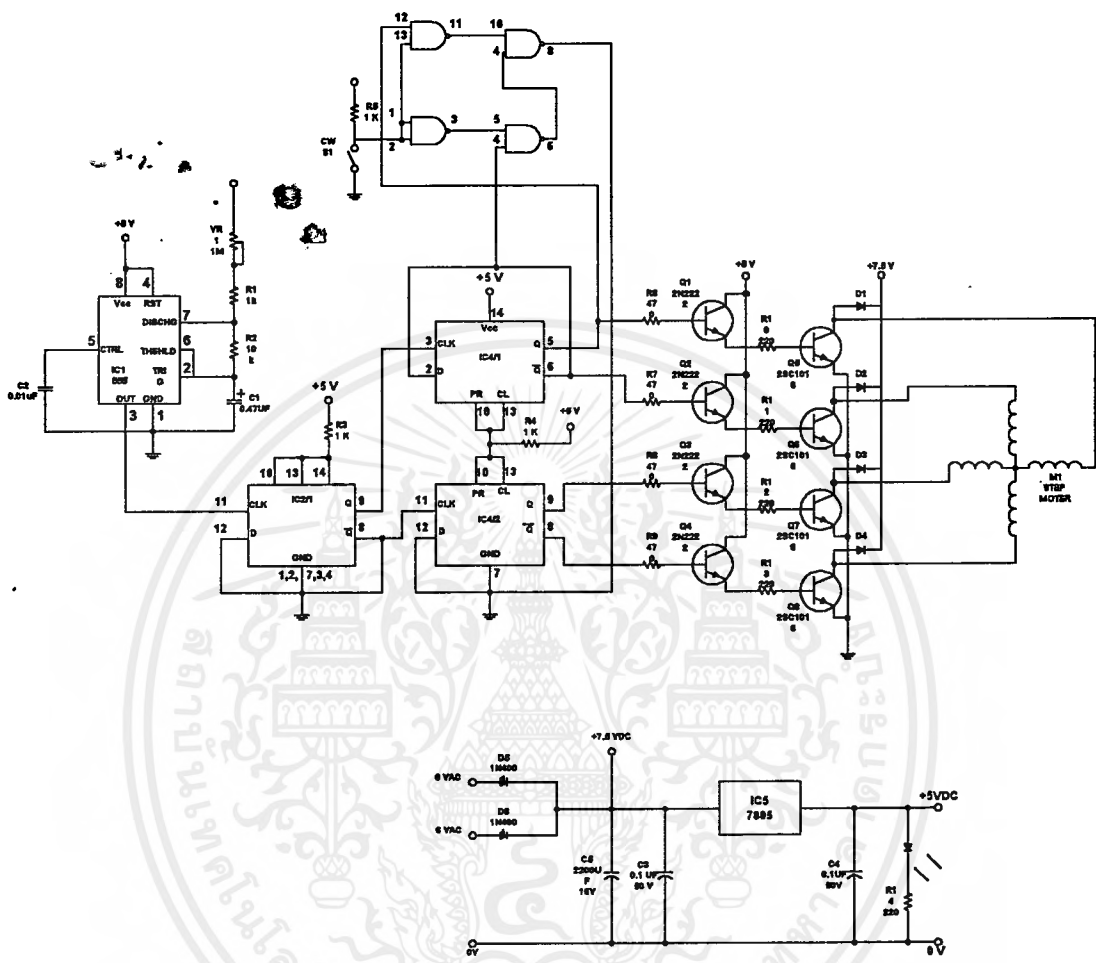
เนื่องจากการทำแผ่นวงจรพิมพ์แบบนี้เป็นการขูดเนื้อทองแดงออกเราจึงไม่สามารถเอาข้อมูล ที่ออกมาจากเครื่องคอมพิวเตอร์ไปใช้ได้เลยทันที จำเป็นจะต้องมีการแปลงข้อมูลเหล่านั้นให้เป็น ข้อมูลที่เราต้องการ (ข้อมูลที่เป็นส่วนของเส้นรอบรูปเท่านั้น) ซึ่งการแปลงข้อมูลนี้ในโปรแกรม Protel for Window จะมีคำสั่งที่ใช้ในการแปลงอยู่ จากนั้นเราก็จะนำเอาข้อมูล ที่ได้ไปผ่านส่วน ที่เป็น Software ที่เขียนขึ้นมาสำหรับเครื่องกัดแผ่นวงจรพิมพ์อัตโนมัติ นำไปควบคุมการเคลื่อน ไหวไปตามข้อมูล พร้อมทั้งทำการขูดเนื้อทองแดงไปด้วย และเครื่องนี้ก็ยังสามารถจะรู้ สำหรับใส่อุปกรณ์ไปพร้อมกันด้วยภายในเครื่องเดียวกัน



BLOCKDIAGRAM ของเครื่องกัดแผ่นวงจรพิมพ์อัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบุคลากรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 วงจรควบคุม สเต็ปป์มอเตอร์ (STEPPING MOTOR)



รูปที่ 15 วงจรสมบูรณ์ของวงจรขับสเต็ปป์มอเตอร์

3.2.1 หลักการทำงานของวงจร

จากรูปวงจรถ่ายการทำงานจะเริ่มที่ไอซีตั้งเวลา เบอร์ 555 ที่ต่อในลักษณะวงจรอะอสเตเบิลด์ มัลติไวเบรเตอร์ ทำหน้าที่กำเนิดสัญญาณสี่เหลี่ยม ความถี่ 3-150 Hz ขึ้นมาโดยสามารถปรับความถี่ได้ด้วย VR1 สัญญาณสี่เหลี่ยมที่ได้จะป้อนให้ IC2 ซึ่งทำหน้าที่จัดคิวไซเคิลของสัญญาณสี่เหลี่ยมให้ได้ 50 เปอร์เซ็นต์ ซึ่งใช้ไอซีเบอร์ 7474 ต่อในลักษณะ T-ฟลิปฟลอป จากนั้นสัญญาณที่มีคิวไซเคิล 50 เปอร์เซ็นต์ ที่ออกที่ขา Q และ Q จะนำไปป้อนให้ IC4/1 และ IC4/2 ซึ่งเป็น D-ฟลิปฟลอป เบอร์ 7474 ที่จะทำงานได้ที่ขอบขาขึ้นของสัญญาณนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นการทำงานของ IC4/1 และ IC4/2 จึงเกิดขึ้นไม่พร้อมกันโดย IC4/1 จะทำงานก่อน IC4/2 ซึ่งในวงจรนี้เราต่อ IC4 เป็น T-ฟลิปฟล็อป IC4/2 จะรับข้อมูล “0” หรือ “1” มาจาก IC3 อีกที่หนึ่งข้อมูลที่รับมานี้จะตรงกับข้อมูลของ IC4/1 หรือตรงกันข้ามกับ IC4/1 ขึ้นอยู่กับสวิตช์ S1 คือ ถ้า S1 อยู่ที่ตำแหน่ง CW เมื่อขา D ของ IC4/1 เป็น “0” IC4/2 ก็ได้รับข้อมูลมาจาก IC3 เป็น “0” หรือถ้า S1 อยู่ที่ตำแหน่ง CCW เมื่อขา D ของ IC4/1 เป็น “0” IC4/2 ก็จะได้รับข้อมูลที่ขา D เป็น “1” นั่นก็คือ ถ้า S1 อยู่ที่ CW IC4/1 และ IC4/2 จะมีข้อมูลเหมือนกัน แต่ถ้า S1 อยู่ที่ CCW IC4/1 และ IC4/2 จะได้รับข้อมูลที่ต่างกัน

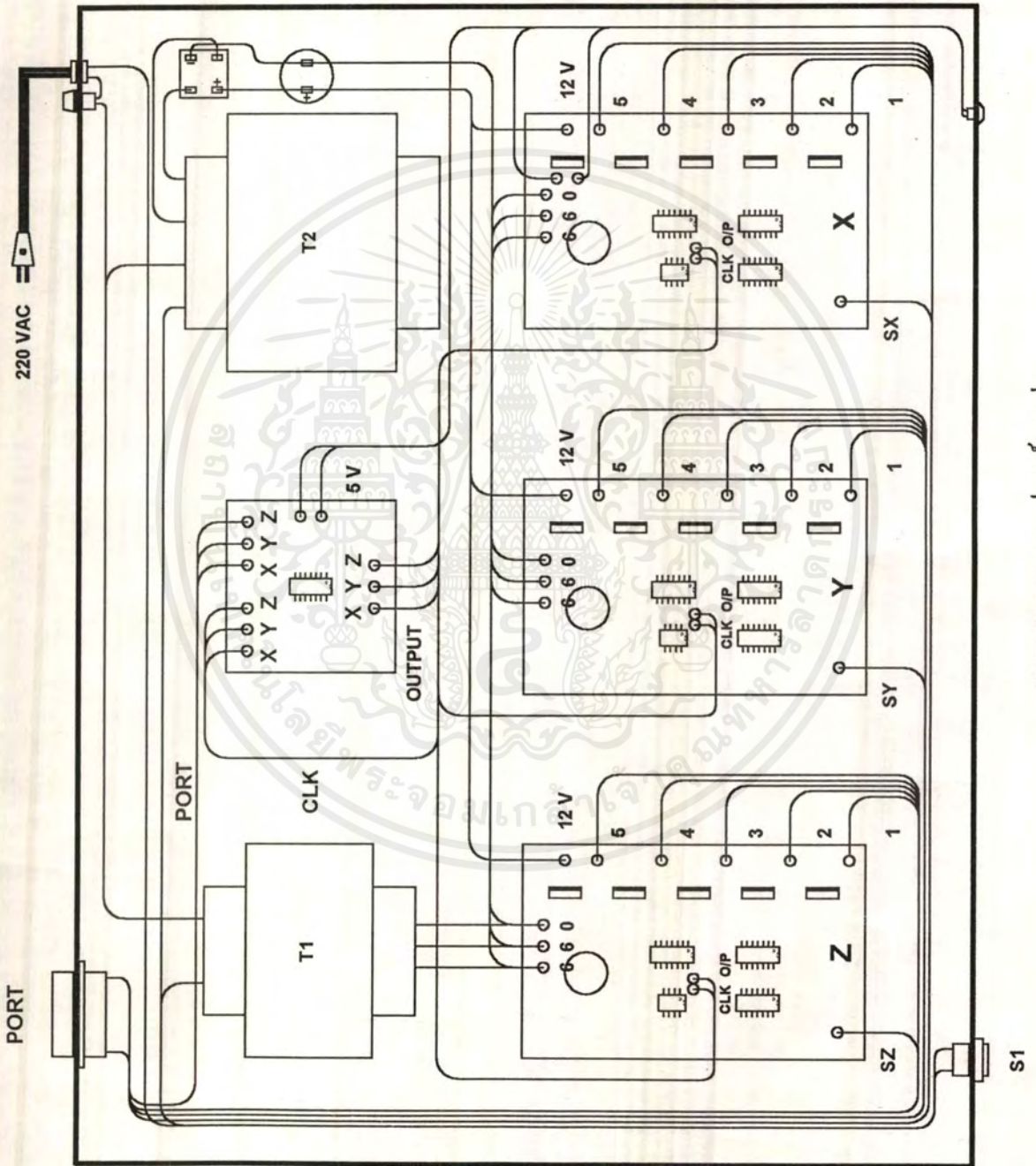
เอาต์พุตที่ได้จาก IC4 คือที่ขา 5, 6, 9 และขา 8 จะนำไปป้อนให้กับ Q1-Q4 ตามลำดับเพื่อใช้ขับ Q5-Q8 โดยสัญญาณที่ใช้ขับ Q5-Q8 มีลักษณะเป็นพัลส์

จะเห็นได้ว่า มีการขับในลักษณะกระตุ้น 2 เฟส ซึ่งมีข้อดีคือ แรงบิดสูงส่วนไดโอด D1-D4 ที่ต่อพร้อมขดลวดของ สเต็ปป์มอเตอร์นั้น ทำหน้าที่ป้องกันแรงดันย้อนกลับซึ่งเกิดขึ้นจากขดลวดของมอเตอร์ ขณะที่ทรานซิสเตอร์หยุดทำงาน เพื่อป้องกันไม่ให้ทรานซิสเตอร์เสียหายสำหรับชุดจ่ายไฟของวงจรในส่วนเลี้ยงมอเตอร์สามารถเปลี่ยนได้ตามขนาดของมอเตอร์ว่าใช้ไฟเลี้ยงกี่โวลต์

รายการอุปกรณ์ชุดควบคุม

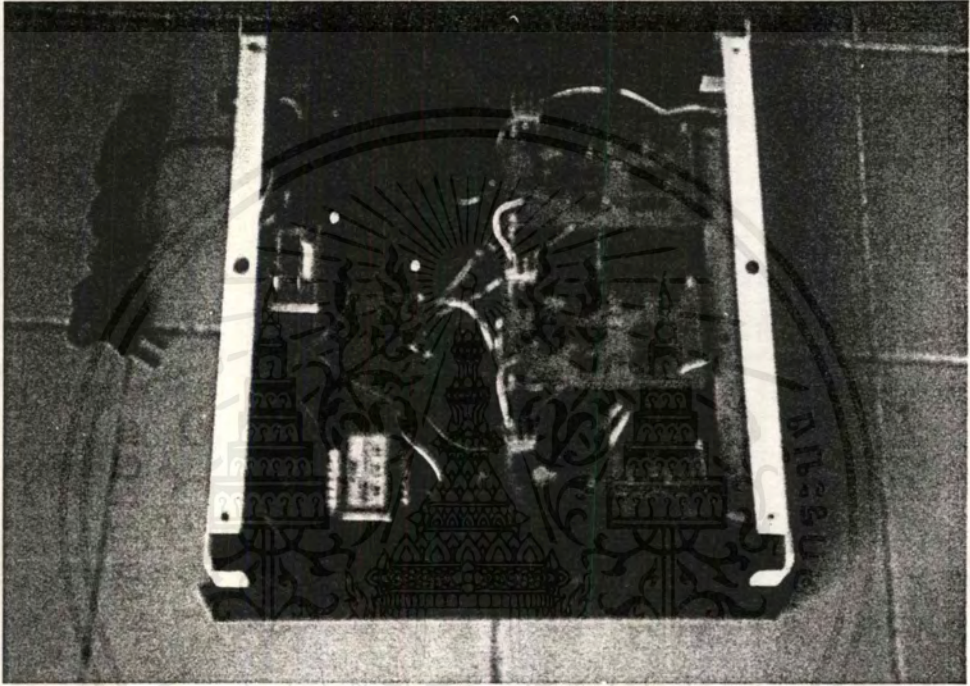
C1	- 0.47 μ F 50 V แทนทาลัม	3 ตัว
C2	- 0.01 μ F 50 V ไมล่ำ	3 ตัว
C3, C4	- 0.1 μ F 50 V ไมล่ำ	6 ตัว
C5	- 2200 μ F 25V อิเล็กโทรไลต์	4 ตัว
D1-D6	- 1N4001	18 ตัว
IC1	- NE555	3 ตัว
IC2,IC4	- 74LS74	3 ตัว
IC3	- 74LS00	3 ตัว
IC6	- 74LS08	1 ตัว
IC5	- 7805	3 ตัว
Q1-Q4	- 2N2222	12 ตัว
Q5-Q8	- 2SC1061	12 ตัว
R1,R3,R5	- 1K Ω /4W \pm 5%	9 ตัว
R2	- 10 K Ω /4W \pm 5%	3 ตัว
R6-R9	- 470 Ω /4W \pm 5%	12 ตัว
R10-R18	- 220 Ω /4W \pm 5%	12 ตัว
VR1	- M Ω B	4 ตัว
T1	- หม้อแปลงเข้า 220V ออก 6-0-6 โวลต์	1 ตัว
T2	- หม้อแปลงเข้า 220V ออก 12-0-12 โวลต์	1 ตัว
S1	- สวิตช์เปิด-ปิด	1 ตัว
RELAY 12V		1 ตัว
มอเตอร์สแต็ปปีง		4 ตัว
แผ่นวงจรพิมพ์, ฟิวส์, หลอดไฟ, สายไฟเอซี, สายเคเบิลภายใน, สายแพ, แทนเครื่อง, ชอกเก็ตไอซี, คอนเน็กเตอร์, แผ่นระบายความร้อน		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพแสดงการลงอุปกรณ์ทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



การติดตั้งชุดควบคุมลงในกล่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 โครงสร้างทางด้านกลไกของปริศยานิทรรศ

3.3.1 ส่วนกลไกเคลื่อนที่ในแนวแกน X และแกน Y

ได้รับการขับเคลื่อนจาก stepping motor โดยใช้เก็ยตัวหนอนขนาด 6 มม. เป็นตัวเคลื่อนไหวทั้งแกน X และแกน Y ชุดรางเลื่อนประกอบด้วยแท่งเหล็กและชุดดลับลูกปืนในแกน X ใช้แท่งเหล็กขนาดเส้นผ่านศูนย์กลาง 12 มม. ในแกน Y ใช้แท่งเหล็กขนาดเส้นผ่านศูนย์กลาง 16 มม. ชุดดลับลูกปืนประกอบด้วยลูกปืนด้านละ 8 ลูก และจะต้องปรับระยะให้เคลื่อนไปบนแท่งเหล็กคู่ได้อย่างคล่องทั้ง 2 แกน ชิ้นส่วนต่างๆ แสดงไว้ในหัวข้อของส่วนประกอบภาคกลไก

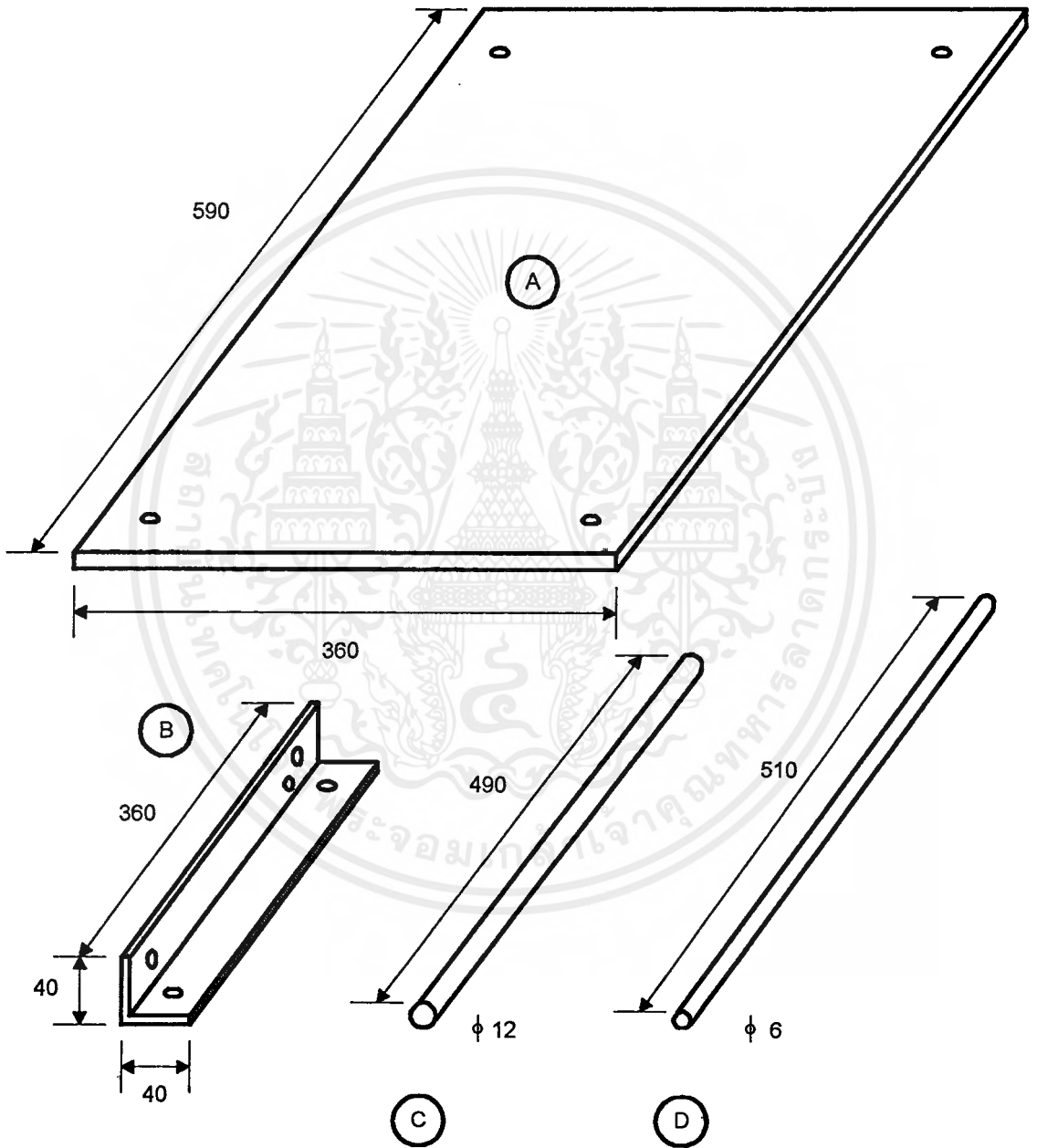
3.3.2 ส่วนกลไกเคลื่อนที่ในแนวแกน Z

ได้รับการขับเคลื่อนจาก stepping motor โดยใช้เก็ยตัวหนอนขนาด 6 มม. เป็นตัวผลักดันให้มอเตอร์ที่ใช้เป็นขับเคลื่อนคอกสวนสำหรับกัด ให้มีการเคลื่อนที่ขึ้น และ ลง ทางด้านแกน Z ชุดรางเลื่อนใช้เป็นรางอลูมิเนียม โดยใช้ จารบี เป็นตัวช่วยให้เกิดการเคลื่อนที่คล่องตัวมากยิ่งขึ้น ชิ้นส่วนต่างๆ แสดงไว้ในหัวข้อของส่วนประกอบภาคกลไก

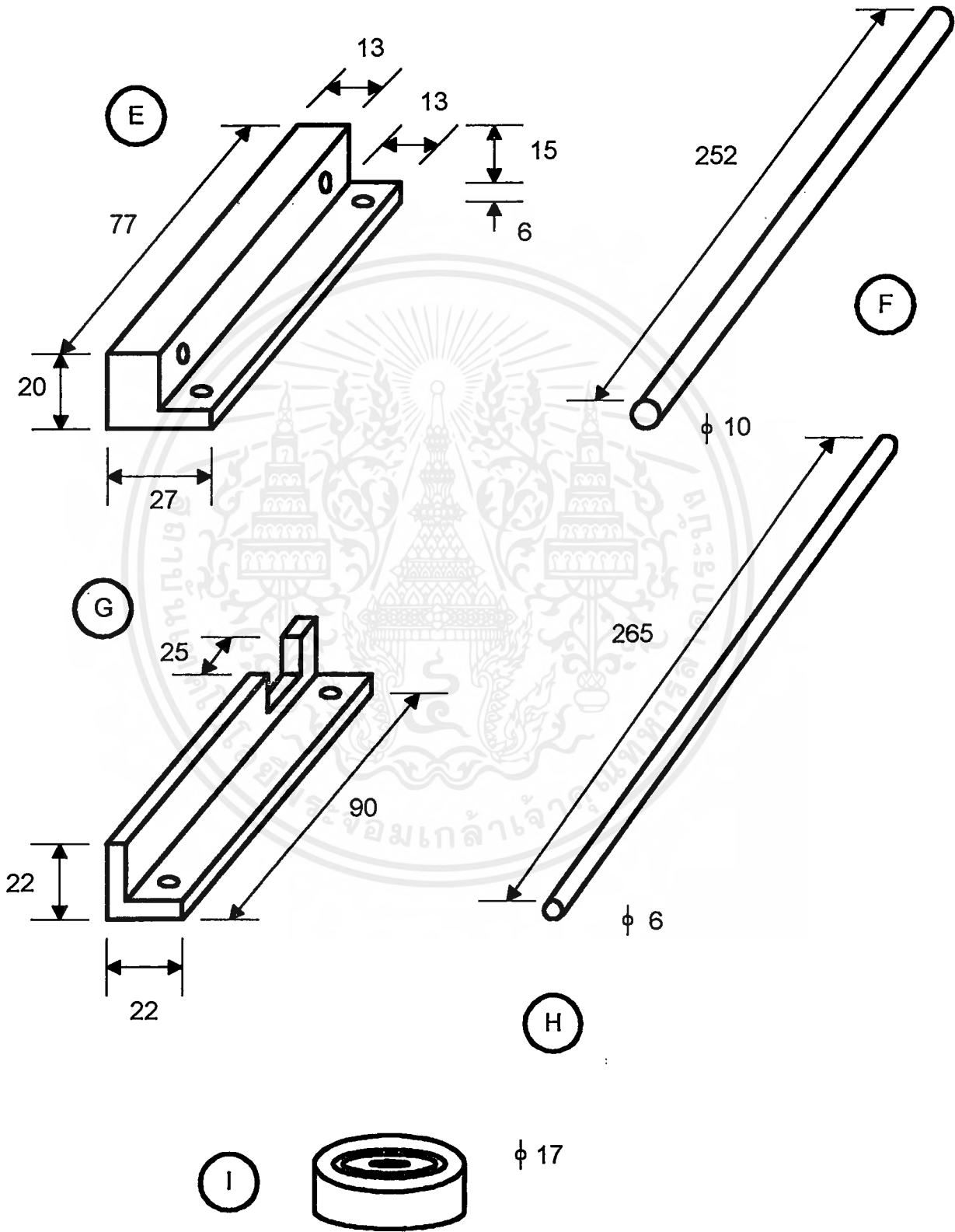
3.3.4 ส่วนกลไกที่ใช้ในการจับยึดมอเตอร์สำหรับดอกกัด

อุปกรณ์ในส่วนนี้ได้รับการออกแบบมาให้มีการทำงานสัมพันธ์กันกับส่วนกลไกที่ใช้ในการเคลื่อนที่ในแนวแกน Z ชิ้นส่วนต่างๆ แสดงไว้ในหัวข้อของส่วนประกอบภาคกลไก

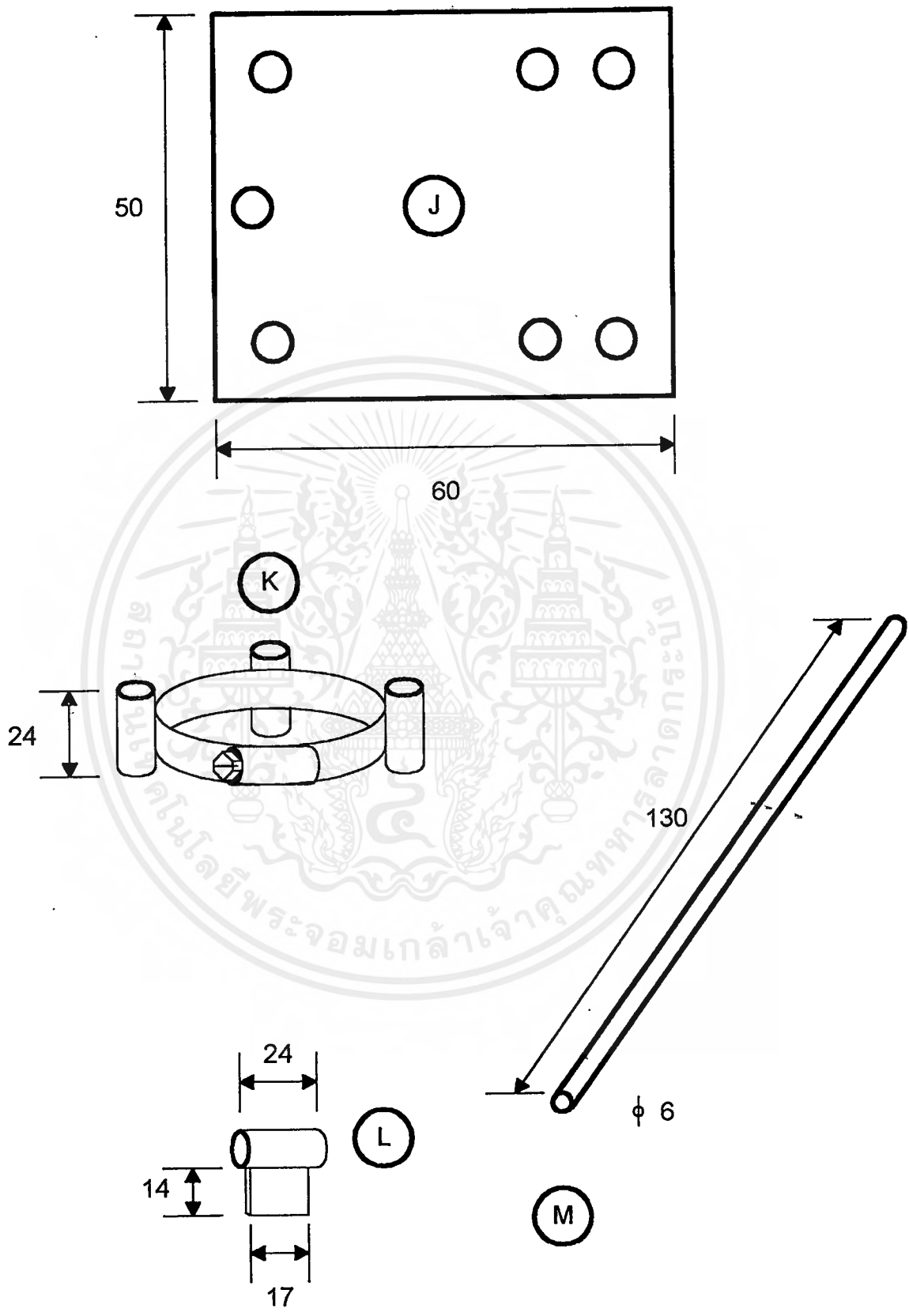
3.3.5 ส่วนประกอบภาคกลไก



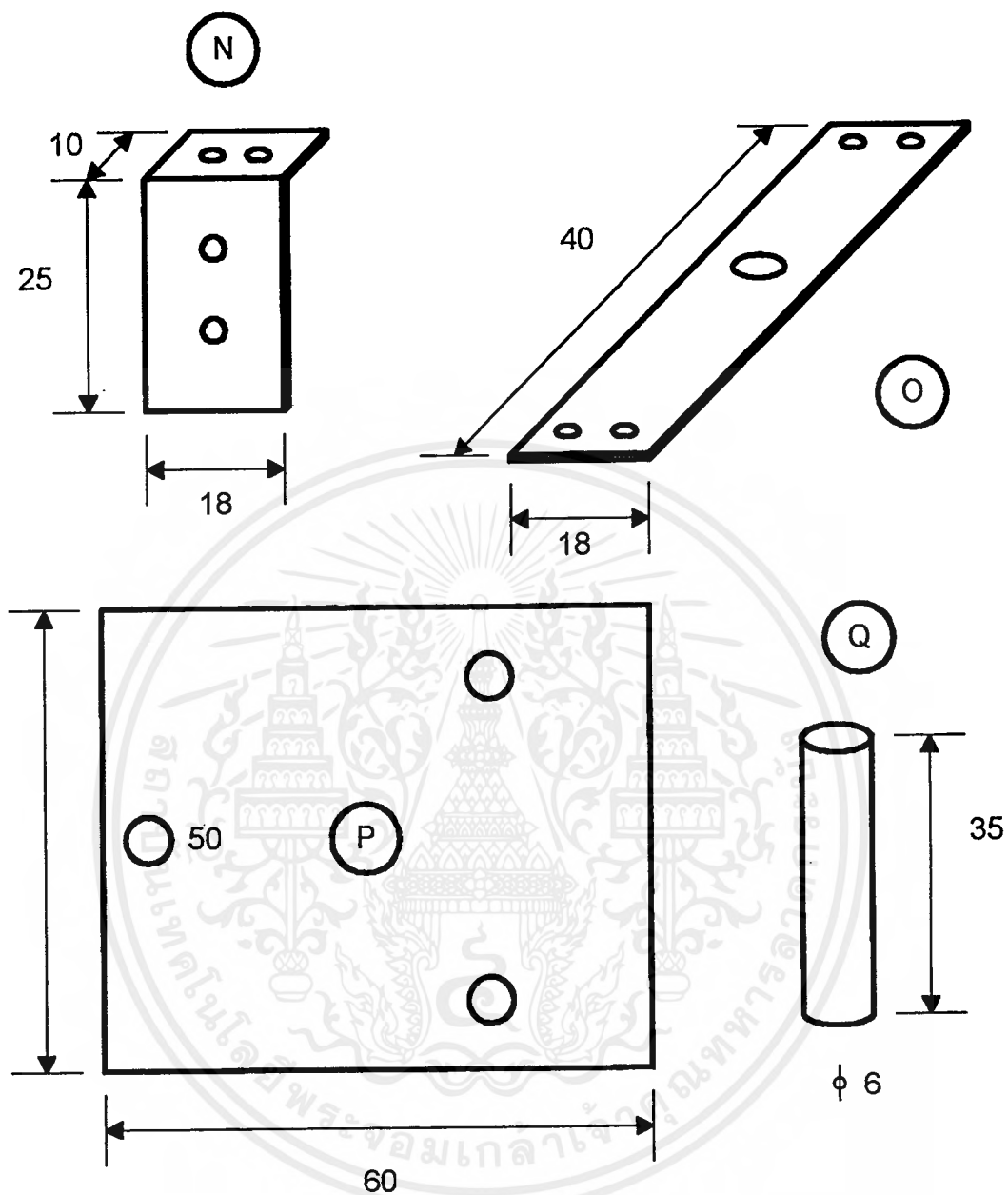
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



หมายเหตุ

A, D, H, J, O, Q

B, C, E, F, G, K, L, N, P

M

I

ชิ้นส่วน D, H, R

ชิ้นส่วน I

มีจำนวนอย่างละ 1 ชิ้น

มีจำนวนอย่างละ 2 ชิ้น

มีจำนวนอย่างละ 3 ชิ้น

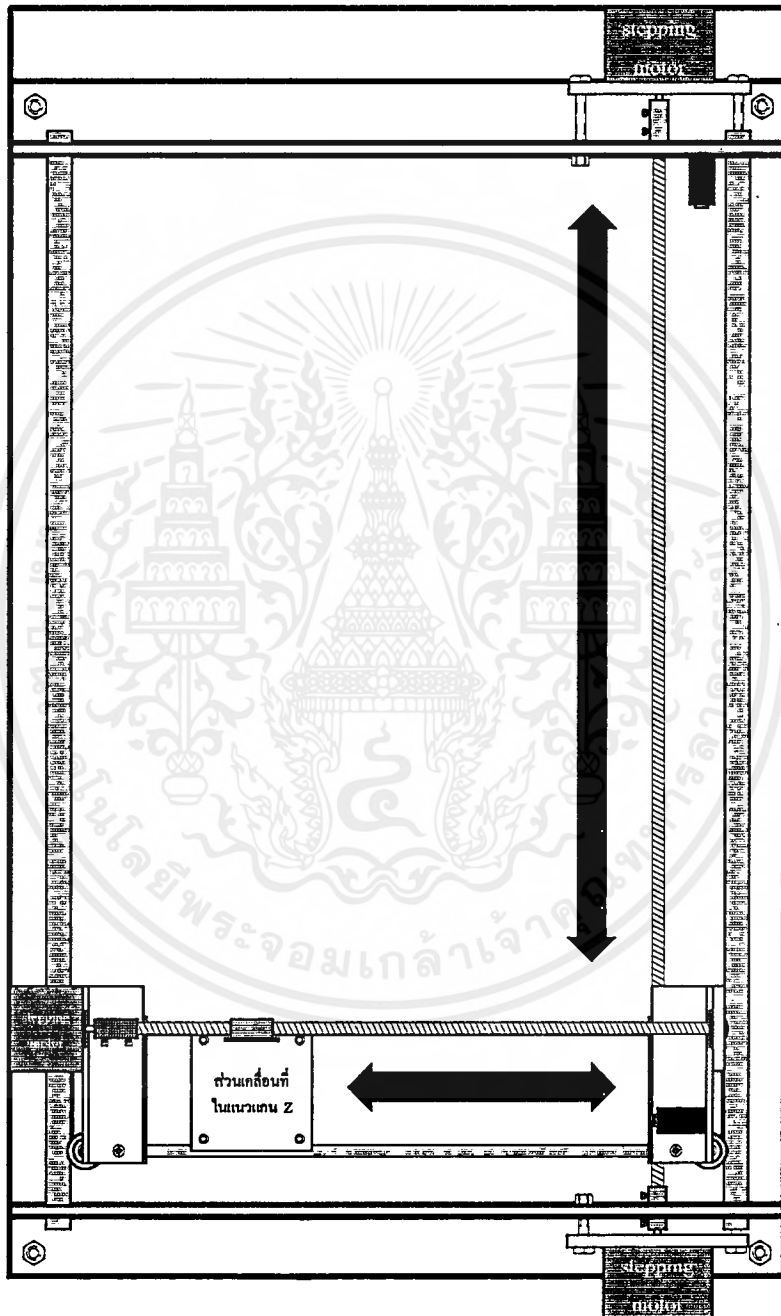
มีจำนวนอย่างละ 16 ชิ้น

เป็นเกลียวตลอดเส้น

เป็นดัดถูกป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

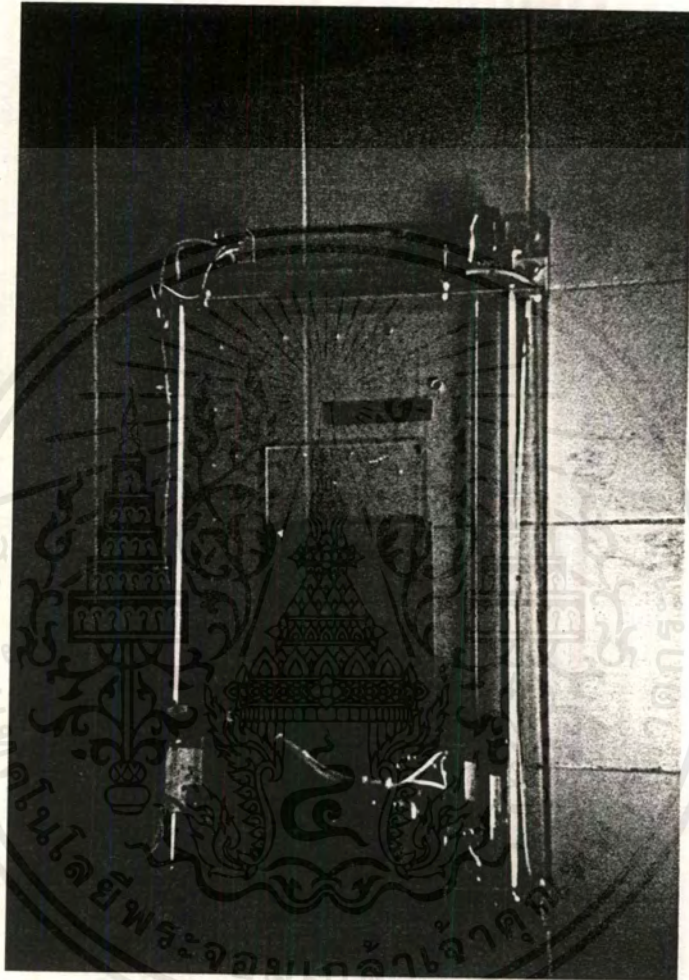
การเคลื่อนที่ในแนวแกน X-Y



จุดไมโครสวิตช์

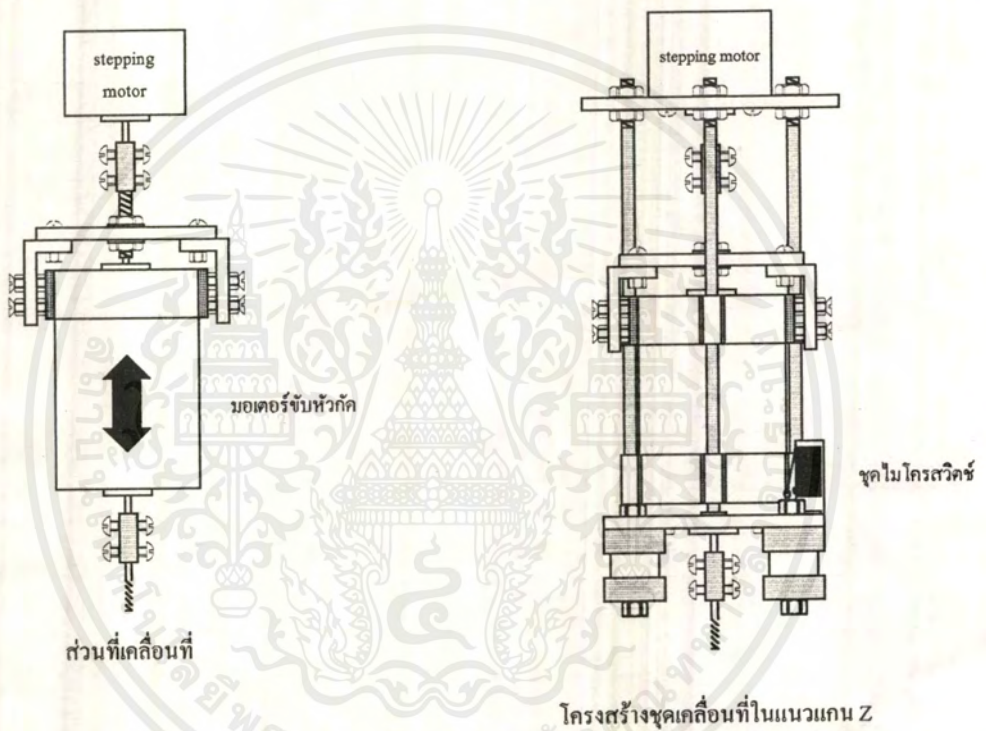
จุดไมโครสวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

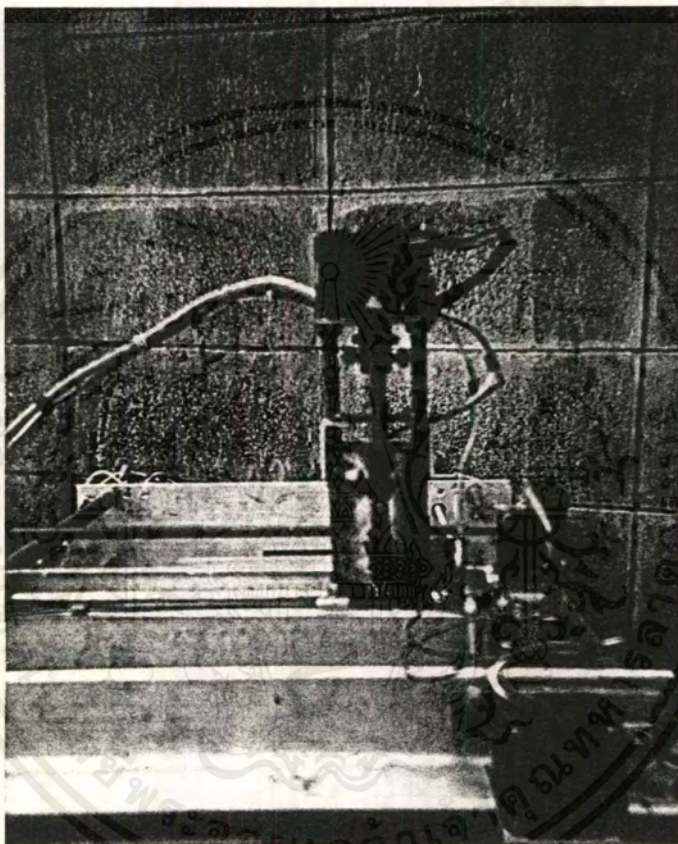


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

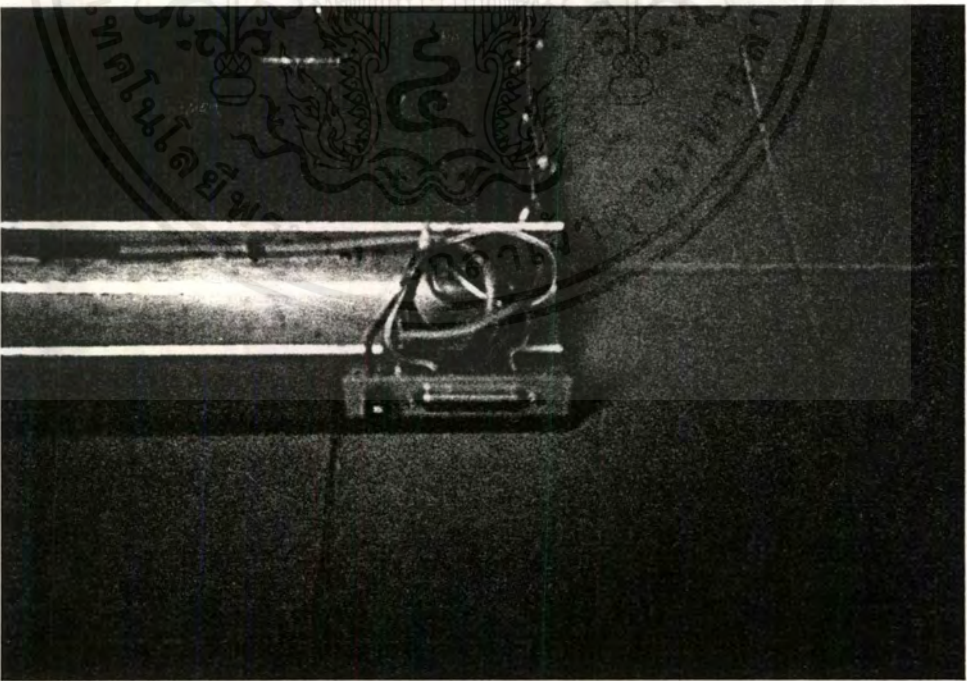
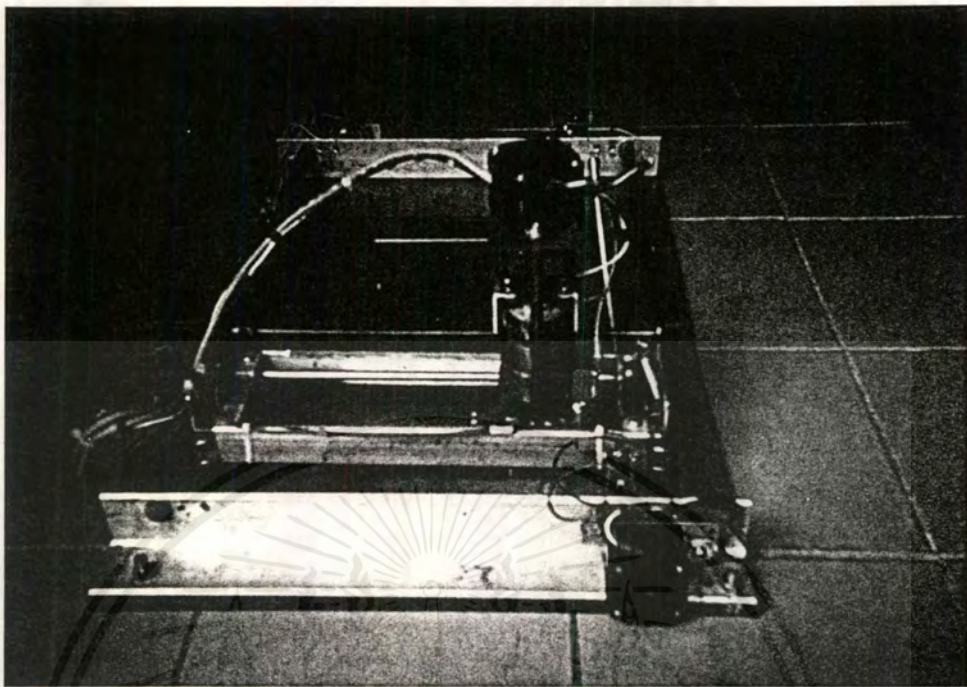
การเคลื่อนที่ในแนวแกน Z



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

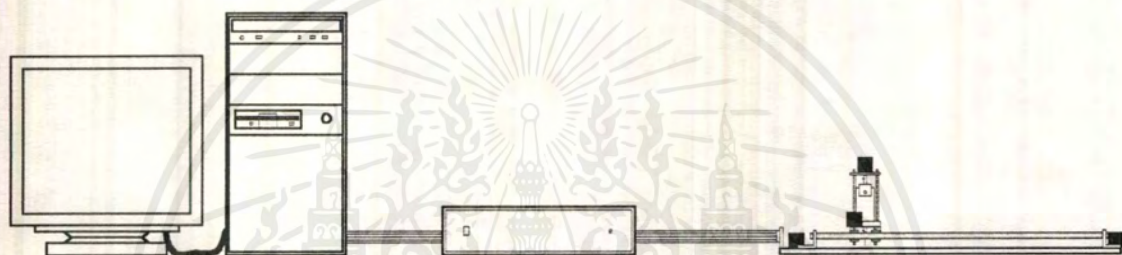


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



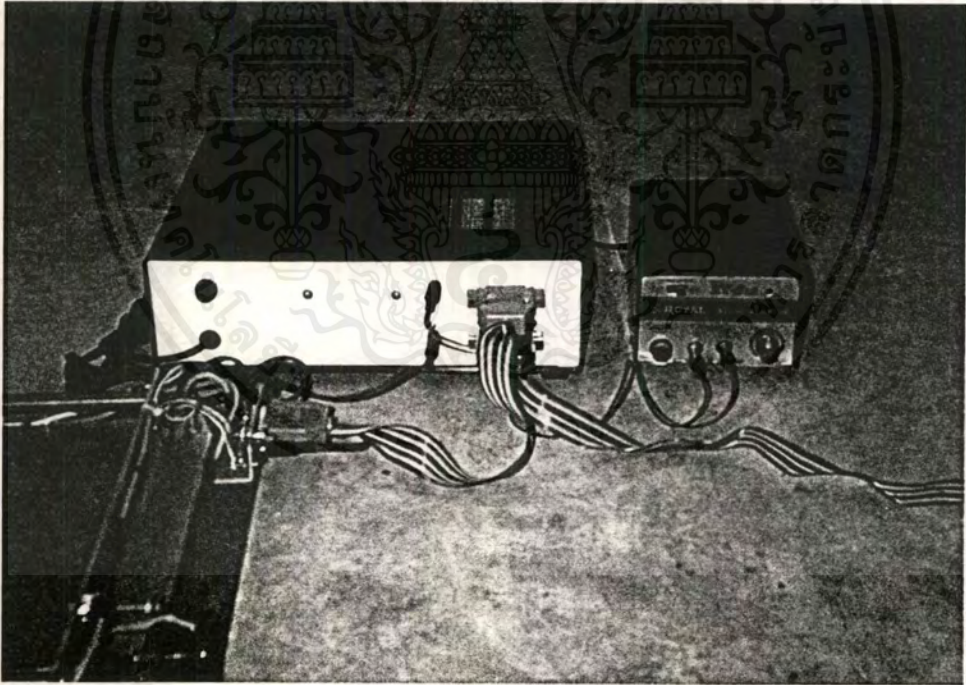
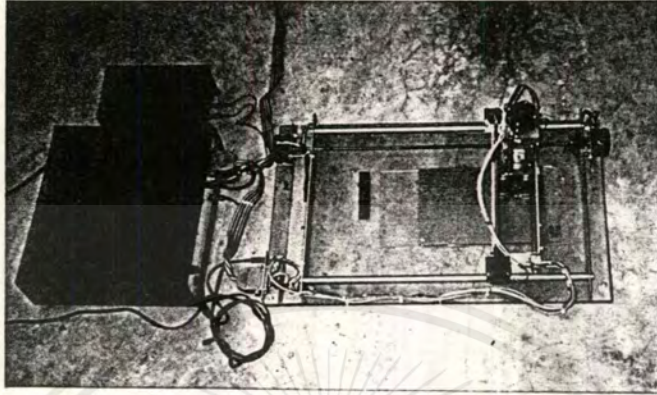
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพแสดงการต่อใช้งานของเครื่องกัดแผ่นวงจรพิมพ์อัตโนมัติ



เครื่องคอมพิวเตอร์ PC เครื่องควบคุมสเต็ปมอเตอร์ เครื่องกัดแผ่นวงจรพิมพ์อัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การแปลงข้อมูลจากโปรแกรม Protel

การแปลงข้อมูลนั้นสามารถทำได้โดยใช้คำสั่งที่มีอยู่ในตัวโปรแกรม Protel โดยเข้าไปที่คำสั่ง File แล้วเลือกไปที่คำสั่ง NC Drill เราจะได้ข้อมูลที่เป็นจุด TXT ซึ่งเราจะเอาข้อมูลในส่วนนี้ไปป้อนให้กับโปรแกรมที่ภาค software เพื่อชี้ตำแหน่งของรูที่ต้องการจะเจาะ



บทที่ 4

การทดลองผลการทดลอง

4.1 ผลการทดลองและการทดสอบโครงงาน

4.1.1 การเจาะ

เครื่องสามารถทำการเจาะแผ่นวงจรพิมพ์ได้ตามที่ต้องการโดยการกำหนดค่าของตำแหน่งที่จะต้องการจะเจาะให้เคลื่อนที่ไปทั้งในแนวแกน X และในแนวแกน Y เมื่อตัวหัวกัดเคลื่อนที่ไปถึงตำแหน่งที่ต้องการ เครื่องจะสั่งงานให้ตัวหัวกัดเคลื่อนที่ลงมาจนถึงตำแหน่งที่ได้ตั้งค่าไว้ ซึ่งก็จะทะลุแผ่นวงจรพิมพ์พอดี แล้วเครื่องก็จะสั่งการให้ตัวหัวกัดเคลื่อนที่ขึ้นมายังจุดเริ่มต้น แล้วเครื่องก็จะทำการสั่งการให้ตัวหัวกัดเคลื่อนที่ไปยังจุดต่อไปที่จะต้องการจะเจาะ เมื่อทำการเจาะเสร็จเรียบร้อยแล้วเครื่องก็จะสั่งการให้ตัวหัวกัดเคลื่อนที่กลับมายังจุดเริ่มต้น เป็นอันเสร็จสิ้นในขั้นตอนของการเจาะ

ค่าของตำแหน่งที่ใช้ในการเจาะจะนำมาจากค่าที่เขียนขึ้นโดยโปรแกรม Protel ซึ่งก่อนที่จะนำค่ามาใช้นั้นจะต้องทำการแปลงค่าข้อมูลก่อน

4.1.2 การกัด

เครื่องจะทำงานคล้ายๆ การเจาะ จะแตกต่างกันตรงที่การปรับระยะของตัวหัวกัดที่จะลงมายังแผ่นวงจรพิมพ์ ให้มีระยะพอดีกับระยะที่เป็นผิวทองแดงที่ติดอยู่บนแผ่นวงจรพิมพ์ กับค่าของข้อมูลที่ป้อนให้กับเครื่อง ซึ่งค่าที่ป้อนให้จะมีจำนวนข้อมูลมากกว่าจำนวนข้อมูลของการเจาะ และค่าจะมีความต่อเนื่อง ซึ่งเป็นลักษณะเหมือนกับเป็นการวาดรูปลายวงจรลงบนแผ่นวงจรพิมพ์ หัวกัดจะทำการขุดเอาเนื้อทองแดงที่อยู่บนแผ่นวงจรพิมพ์ให้หลุดออกไป จึงเกิดเป็นรูปลายวงจรขึ้นบนแผ่นวงจรพิมพ์ เมื่อทำการกัดเสร็จเรียบร้อยแล้วเครื่องก็จะสั่งการให้ตัวหัวกัดเคลื่อนที่กลับมายังจุดเริ่มต้น เป็นอันเสร็จสิ้นในขั้นตอนของการกัด

ค่าของตำแหน่งที่ใช้ในการกัดจะนำมาจากค่าที่เขียนขึ้นโดยโปรแกรม Protel ซึ่งก่อนที่จะนำค่ามาใช้นั้นจะต้องทำการแปลงค่าข้อมูลก่อนเช่นเดียวกันกับการเจาะ

บทที่ 5

สรุปผลการดำเนินงาน

5.1 สรุปผลการดำเนินงาน

5.1.1 งานในเทอมแรก Project I

- ศึกษาการทำงานและหาข้อมูลเกี่ยวกับโครงการ
- ทำการออกแบบในส่วนของกลไก
- ทดสอบและปรับปรุงการทำงานของส่วนกลไกตามที่ออกแบบไว้

5.1.2 งานในเทอมสอง Project II

- ทำในส่วนของคุณควบคุมสเต็ปปีงมอเตอร์
- ทดสอบใช้งานจริงกับชุดกลไก
- ศึกษาและเขียนในส่วนของโปรแกรมควบคุมเครื่อง
- ทดสอบและปรับปรุงโปรแกรมเพื่อให้มีการทำงานที่ถูกต้องและแม่นยำในการใช้งานจริง

5.1.3 ผลของโครงการ

สามารถควบคุมการทำงานของเครื่องผ่านทางเครื่องคอมพิวเตอร์ได้ ในขั้นตอนของการเจาะ และการสั่งงานให้เครื่องทำการกัดได้แต่ยังไม่สมบูรณ์

5.2 ปัญหาที่เกิดจากการทดลอง

ด้านกลไก ชิ้นส่วนและอุปกรณ์บางชิ้นต้องออกแบบเป็นพิเศษ จึงไม่สามารถหาได้ตามท้องตลาด จึงจำเป็นต้องทำการสร้างขึ้นมา และเนื่องจากการขาดเครื่องมือที่จำเป็นบ้างอย่างทำให้เกิดการล่าช้าในการทำงาน

ด้านHard ware อุปกรณ์บางอย่างเช่น ไอซี จากเครื่องค้นแบบเดิมหาได้ยาก จึงทำให้ต้องหาชุดควบคุมใหม่ที่มีอุปกรณ์ขายตามท้องตลาด และสามารถใช้งานได้กับเครื่อง

ด้านSoft ware เนื่องจากมีการเปลี่ยนแปลงภาษาที่ใช้ในการควบคุม ทำให้ต้องทำการปรับเปลี่ยนบางส่วนเพื่อให้เหมาะสมกับการทำงานของเครื่อง

บทที่ 4

การทดลองผลการทดลอง

4.1 ผลการทดลองและการทดสอบโครงงาน

4.1.1 การเจาะ

เครื่องสามารถทำการเจาะแผ่นวงจรพิมพ์ได้ตามที่ต้องการโดยการกำหนดค่าของตำแหน่งที่ต้องการจะเจาะให้เคลื่อนที่ไปทั้งในแนวแกน X และในแนวแกน Y เมื่อตัวหัวกัดเคลื่อนที่ไปถึงตำแหน่งที่ต้องการ เครื่องจะสั่งงานให้ตัวหัวกัดเคลื่อนที่ลงมาจนถึงตำแหน่งที่ได้ตั้งค่าไว้ ซึ่งก็จะทะลุแผ่นวงจรพิมพ์พอดี แล้วเครื่องก็จะสั่งการให้ตัวหัวกัดเคลื่อนที่ขึ้นมายังจุดเริ่มต้น แล้วเครื่องก็จะทำการสั่งการให้ตัวหัวกัดเคลื่อนที่ไปยังจุดต่อไปที่จะต้องเจาะ เมื่อทำการเจาะเสร็จเรียบร้อยแล้วเครื่องก็จะสั่งการให้ตัวหัวกัดเคลื่อนที่กลับมายังจุดเริ่มต้น เป็นอันเสร็จสิ้นในขั้นตอนของการเจาะ

ค่าของตำแหน่งที่ใช้ในการเจาะจะนำมาจากค่าที่เขียนขึ้นโดยโปรแกรม Protel ซึ่งก่อนที่จะนำค่ามาใช้นั้นจะต้องทำการแปลงค่าข้อมูลก่อน

4.1.2 การกัด

เครื่องจะทำงานคล้ายๆ การเจาะ จะแตกต่างกันตรงที่การปรับระยะของตัวหัวกัดที่จะลงมายังแผ่นวงจรพิมพ์ ให้มีระยะพอดีกับระยะที่เป็นผิวทองแดงที่ติดอยู่บนแผ่นวงจรพิมพ์ กับค่าของข้อมูลที่ป้อนให้กับเครื่อง ซึ่งค่าที่ป้อนให้จะมีจำนวนข้อมูลมากกว่าจำนวนข้อมูลของการเจาะ และค่าจะมีความต่อเนื่อง ซึ่งเป็นลักษณะเหมือนกับเป็นการวาดรูปลายวงจรลงบนแผ่นวงจรพิมพ์ หัวกัดจะทำการขูดเอาเนื้อทองแดงที่อยู่บนแผ่นวงจรพิมพ์ให้หลุดออกไป จึงเกิดเป็นรูปลายวงจรขึ้นบนแผ่นวงจรพิมพ์ เมื่อทำการกัดเสร็จเรียบร้อยแล้วเครื่องก็จะสั่งการให้ตัวหัวกัดเคลื่อนที่กลับมายังจุดเริ่มต้น เป็นอันเสร็จสิ้นในขั้นตอนของการกัด

ค่าของตำแหน่งที่ใช้ในการกัดจะนำมาจากค่าที่เขียนขึ้นโดยโปรแกรม Protel ซึ่งก่อนที่จะนำค่ามาใช้นั้นจะต้องทำการแปลงค่าข้อมูลก่อนเช่นเดียวกันกับการเจาะ

4.2 คุณลักษณะเฉพาะ

- ความละเอียด 100 ไมโครเมตร
- พื้นที่ใช้งานทั้งหมด 702 ตารางเซนติเมตร
- ใช้วงจรแผ่นพิมพ์ขนาด A4
- ความเร็วสูงสุด 1 เซนติเมตรต่อวินาที
- ใช้ดอกกัดขนาด 1 มิลลิเมตร, ดอกเจาะใช้ตามขนาดรูที่ต้องการเจาะ
- สื่อสารด้วย parallel port
- ใช้โปรแกรม Turbo C ควบคุม
- สเต็ปมอเตอร์ 1.8 องศาต่อสเต็ป
- วงจรขับมอเตอร์แบบแบ่งละเอียด (Microstep driver)

4.3 การใช้งาน

4.3.1 คำสั่งที่ใช้ในการควบคุม

จากโปรแกรมที่ใช้ควบคุมการทำงานได้กำหนดฟังก์ชันการใช้งานไว้ดังนี้

4	กำหนดให้เคลื่อนที่ไปทางซ้ายของแกน	X
6	กำหนดให้เคลื่อนที่ไปทางขวาของแกน	X
1	กำหนดให้เคลื่อนที่ขึ้นข้างบนตามแกน	Y
3	กำหนดให้เคลื่อนที่ลงข้างล่างตามแกน	Y
•	กำหนดให้ชุดหัวกัดเคลื่อนที่ขึ้นข้างบนตามแกน	Z
0	กำหนดให้ชุดหัวกัดเคลื่อนที่ลงข้างล่างตามแกน	Z
-	กำหนดให้ X เคลื่อนที่ไปทางขวา Y เคลื่อนที่ขึ้น	
+	กำหนดให้ X เคลื่อนที่ไปทางซ้าย Y เคลื่อนที่ลง	
2	กำหนดให้ชุดหัวกัดทำการเจาะ	
5	กำหนดให้ชุดหัวกัดยกขึ้น	
initial	กำหนดให้ชุดหัวกัดเคลื่อนที่ไปยังจุดเริ่มต้น	

4.3.2 การทดลองใช้งานจริง

เริ่มด้วยการต่อชุดควบคุมเข้ากับเครื่อง เปิดโปรแกรมสั่งงานให้ชุดหัวกัดเคลื่อนที่ไปยังจุดเริ่มต้นโดยคำสั่ง initial แล้วป้อนข้อมูลที่เป็น .TXT ให้กับโปรแกรมชุดหัวกัดจะถูกควบคุมโดยโปรแกรม ที่ผ่านชุดควบคุม ซึ่งจะเคลื่อนที่โดยอัตโนมัติไปยังจุดที่ต้องการตามข้อมูลที่ป้อนให้

ในการทดลองได้ทำการทดลองเจาะวงจรม้วนพิมพ์ที่ใช้งานในส่วนควบคุม ซึ่ง ได้ผลเป็นไปตามที่ต้องการ และสามารถใช้งานได้จริง



บทที่ 5

สรุปผลการดำเนินงาน

5.1 สรุปผลการดำเนินงาน

5.1.1 งานในเทอมแรก Project I

- ศึกษาการทำงานและหาข้อมูลเกี่ยวกับโครงการ
- ทำการออกแบบในส่วนของกลไก
- ทดสอบและปรับปรุงการทำงานของส่วนกลไกตามที่ออกแบบไว้

5.1.2 งานในเทอมสอง Project II

- ทำในส่วนของคุณควบคุมสเต็ปปีงมอเตอร์
- ทดสอบใช้งานจริงกับชุดกลไก
- ศึกษาและเขียนในส่วนของโปรแกรมควบคุมเครื่อง
- ทดสอบและปรับปรุงโปรแกรมเพื่อให้มีการทำงานที่ถูกต้องและแม่นยำในการใช้งานจริง

5.1.3 ผลของโครงการ

สามารถควบคุมการทำงานของเครื่องผ่านทางเครื่องคอมพิวเตอร์ได้ ในขั้นตอนของการเจาะ และการสั่งงานให้เครื่องทำการกัดได้แต่ยังไม่สมบูรณ์

5.2 ปัญหาที่เกิดจากการทดลอง

ด้านกลไก ชิ้นส่วนและอุปกรณ์บางชิ้นต้องออกแบบเป็นพิเศษ จึงไม่สามารถหาได้ตามท้องตลาด จึงจำเป็นต้องทำการสร้างขึ้นมา และเนื่องจากการขาดเครื่องมือที่จำเป็นบ้างอย่างทำให้เกิดการล่าช้าในการทำงาน

ด้านHard ware อุปกรณ์บางอย่างเช่น ไอซี จากเครื่องค้นแบบเดิมหาได้ยาก จึงทำให้ต้องหาชุดควบคุมใหม่ที่มีอุปกรณ์ขายตามท้องตลาด และสามารถใช้งานได้กับเครื่อง

ด้านSoft ware เนื่องจากมีการเปลี่ยนแปลงภาษาที่ใช้ในการควบคุม ทำให้ต้องทำการปรับเปลี่ยนบางส่วนเพื่อให้เหมาะสมกับการทำงานของเครื่อง

5.3 การแก้ไข้ปัญหา

เนื่องจากโครงการนี้จะต้องมีการทำงานถึง 3 ส่วนด้วยกัน จำเป็นจะต้องมีอุปกรณ์ และเครื่องมือต่างๆ ที่จำเป็นจะต้องใช้ และมีความชำนาญในการเขียนโปรแกรม

5.4 การวิเคราะห์และแนวทางการพัฒนาของโครงการ

เครื่องกักแค้นวงจรพิมพ์นี้เป็นเพียงเครื่องต้นแบบที่สร้างขึ้นมาเพื่อเป็นการประยุกต์การใช้งานเครื่องนำมาควบคุมเครื่องจักรกลต่างๆ สามารถพัฒนาโครงการนี้ให้มีประสิทธิภาพสูงมากขึ้น เพื่อให้เกิดความแม่นยำในการทำงาน หรือสามารถนำหลักการทำงานในโครงการนี้ไปใช้ควบคุมการทำงานในลักษณะอื่นได้อีก



บรรณานุกรม

1. ปรินญาณินพนธ์เรื่อง เครื่องเจาะ วาด พิมพ์ อัตโนมัติ (DILL PLOT PRINT AUTOMATIC MACHINE) ของนายกิตติ ทาชาวี และ นายขวัญชัย โชมขุนทด หลักสูตรอุตสาหกรรมศาสตรบัณฑิต สาขาเทคโนโลยีอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2537
2. โครงการเรื่อง เครื่องพล็อตเตอร์ ของนักศึกษาระดับประกาศนียบัตรวิชาชีพชั้นสูง ภาคไฟฟ้าอุตสาหกรรม คณะเทคโนโลยีอุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ
3. บทความเรื่อง การประยุกต์ใช้งานไมโครคอมพิวเตอร์ ในการควบคุมเครื่องเจาะ ของ ชำรงค์ศักดิ์ สุกใส และ ดร.โยธิน เปรมปราณีรัชต์ จากวารสารคอมพิวเตอร์ อิเล็กทรอนิกส์ เวลด์ ฉบับที่ 111 และ 112 ปี2530 และบทความเรื่อง ขยายเชิงเก็บบอร์ดเพื่อควบคุมสเตปปีงมอเตอร์ ของ ชำรงค์ศักดิ์ สุกใส จากวารสารคอมพิวเตอร์ อิเล็กทรอนิกส์ เวลด์ ฉบับที่ 116 ปี2531
4. คู่มือการใช้ Protel for Windows อ.บัณฑิต จามรภูติ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <stdio.h>
#include <conio.h>
#include <dos.h>
void init(void);
void stop(void);
void X_left(int lng);
void X_right(int lng);
void Y_up(int lng);
void Y_down(int lng);
void XY_dr(int lng);
void XY_ul(int lng);
void drill();
int x_set=30,y_set=12;
main ()
{
int ch,dly,inpt;

clrscr();
dly = 50;
outport(0x378,0);
while ( ch != 27 )
{
ch = getch();
// inpt = inport(0x378);
gotoxy(x_set,y_set);
printf(" %c",ch);
switch(ch)
{
// X
// case 75: outport(0x378,8);delay(dly);outport(0x378,0);break;
// case 77: outport(0x378,9);delay(dly);outport(0x378,0);break;
// Y
//5750 = 1 mm.
// case 80: outport(0x378,0x20);delay(5750);outport(0x378,0);break; // down
// case 72: outport(0x378,0x22);delay(dly);outport(0x378,0);break; //up

```

```

// Z
case '5': output(0x378,0x10);delay(8000);output(0x378,0); break;
case '2': output(0x378,0x14);delay(8000);output(0x378,0); break;

case '0': output(0x378,0x10);delay(250);output(0x378,0); break;
case '1': output(0x378,0x14);delay(250);output(0x378,0); break;

//XY
// case '7': output(0x378,0x2b);delay(dly);output(0x378,0);break;
// case '9': output(0x378,0x28);delay(dly);output(0x378,0);break;

// X all
case '4': output(0x378,8);break;
case '6': output(0x378,9);break;

// Y all
case '1': output(0x378,0x20);break; // down
case '3': output(0x378,0x22);break; //up

//XY all
case '+': output(0x378,0x2a);break;
case '-': output(0x378,0x29);break;
case 32 : init();break;

// stop
case 13: output(0x378,0);break;

}
}
output(0x378,0);
}
void init(void)
{
int fns;
char xc,yc;
fns = 0;
gotoxy(x_set,y_set);
printf(" initial.");

```

```

xc = inport(0x379);
while ( xc == 0x7f)
{
    xc = inport(0x379);
    XY_dl(1);
}

xc = inport(0x379);
if ((xc == 0x5f) || (xc == 0x4f) ) fns =1;
while( fns != 1)
{
    printf(".");
    xc = inport(0x379);
    if ((xc == 0x5f) || (xc == 0x4f) ) fns =1;
    X_right(1);
}
fns = 0;

xc = inport(0x379);
if ((xc == 0x4f) || (xc == 0x6f) ) fns =1;
while( fns != 1)
{
    printf(".");
    Y_down(1);
    xc = inport(0x379);
    if (xc == 0x4f) fns =1;
}
XY_ul(5);
outport(0x378,0);

}

void stop(void)
{
    outport(0x378,0);
}

```

```

}
void X_left(int lng)
{
  outport(0x378,0x8);delay(lng*1000);
  stop();
}

void X_right(int lng)
{
  outport(0x378,0x9);delay(lng*1000);
  stop();
}

void Y_up(int lng)
{
  outport(0x378,0x20);delay(lng*1000);
  stop();
}

void Y_down(int lng)
{
  outport(0x378,0x22);delay(lng*1000);
  stop();
}

void XY_dr(int lng)
{
  outport(0x378,0x2b);delay(lng*1000);
  stop();
}

void XY_ul(int lng)
{
  outport(0x378,0x28);delay(lng*1000);
  stop();
}

void drill()

```

```
{  
  gotoxy(x_set,y_set);  
  printf(" drill... ");  
  
  output(0x378,0x14);delay(3000);  
  stop();  
  output(0x378,0x10);delay(3000);  
  stop();  
}
```



```

#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <stdlib.h>
#include <string.h>

void init(void);
void stop(void);
void X_left(int lng);
void X_right(int lng);
void Y_up(int lng);
void Y_down(int lng);
void XY_dr(int lng);
void XY_ul(int lng);
void XY_dl(int lng);
void XY_ur(int lng);
void drill();
void plot();
void unplot();
int x_set=30,y_set=12;
main ()
{
    int dly,i=0,lng;
    char ch,x[10],y[10];
    int xo=0,yo=0,xn,yn,xl,yl;
    clrscr();
    stop();
    init();
    clrscr();
    /***** Program *****/
    plot();

```

```

X_right(250);
Y_up(160);
X_left(60);
Y_down(90);
X_left(180);
Y_up(20);
X_right(160);
Y_up(60);
X_left(160);
Y_up(20);
X_right(160);
Y_up(60);
X_left(160);
Y_up(200);
X_right(100);
Y_down(100);
X_right(50);
Y_up(100);
X_right(20);
Y_down(100);
X_right(50);
Y_up(100);
X_right(100);
Y_down(440);

unplot();
/*****End of Program *****/
init();
gotoxy(60,24);
printf( " Any key to Exit ");

```

```
getch();
}

/******* FUNCTION *****/
void init(void)
{
int fns;
char xc,yc;
fns = 0;
gotoxy(x_set,y_set);
printf(" initial.");

xc = inport(0x379);
while ( xc == 0x7f)
{
xc = inport(0x379);
XY_dl(1);
}

xc = inport(0x379);
if ((xc == 0x5f) || (xc == 0x4f) ) fns =1;
while( fns != 1)
{
printf(".");
xc = inport(0x379);
if ((xc == 0x5f) || (xc == 0x4f) ) fns =1;
X_left(1);
}
fns = 0;
```

```

xc = inport(0x379);
if ((xc == 0x4f) || (xc == 0x6f)) fns =1;
while( fns != 1)
{
    printf(".");
    Y_down(1);
    xc = inport(0x379);
    if (xc == 0x4f) fns =1;
}
XY_ur(50);
outport(0x378,0);
}
void stop(void)
{
    outport(0x378,0);
}
void X_left(int lng)
{
    outport(0x378,0x8);delay(lng*100);
    stop();
}
void X_right(int lng)
{
    outport(0x378,0x9);delay(lng*100);
    stop();
}
void Y_up(int lng)
{

```

```

    outport(0x378,0x20);delay(lng*115);
    stop0;
}

void Y_down(int lng)
{
    outport(0x378,0x22);delay(lng*115);
    stop0;
}

void XY_dr(int lng)
{
    outport(0x378,0x2b);delay(lng*100);
    stop0;
}

void XY_ul(int lng)
{
    outport(0x378,0x28);delay(lng*100);
    stop0;
}

void drill()
{
    // gotoxy(x_set,y_set);
    // printf(" drill... ");

    outport(0x378,0x14);delay(8000);
    stop0;
    outport(0x378,0x10);delay(8000);
    stop0;
}

void XY_dl(int lng)

```

```
{
  output(0x378,0x2a);delay(lng*100);
  stop();
}
void XY_ur(int lng)
{
  output(0x378,0x29);delay(lng*100);
  stop();
}
void plot()
{
  output(0x378,0x14);delay(5980);
  stop();
  delay(100);
  output(0x378,0x14);delay(400);
  stop();
}
void unplot()
{
  output(0x378,0x10);delay(6380);
  stop();
}
```

M72

M48

T1F00S00C1.194

%

T01

X14186Y17435

Y17515

X14266Y17470

Y17670

X14186

M30



M72

M48

T1F00S00C0.635

%

T01

X12065Y166116

Y16637

Y166624

Y166878

Y167132

Y167386

Y16764

X121412

Y167386

Y167132

Y166878

Y166624

Y16637

Y166116

M30



```

#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <stdlib.h>
#include <string.h>

void init(void);
void stop(void);
void X_left(int lng);
void X_right(int lng);
void Y_up(int lng);
void Y_down(int lng);
void XY_dr(int lng);
void XY_ul(int lng);
void XY_dl(int lng);
void XY_ur(int lng);
void drill();
void plot();
int x_set=30,y_set=12;
main ()
{
FILE *fp;
int dly,i=0,lng;
char ch,x[10],y[10],fname[15];
int xo=0,yo=0,xn,yn,xl,y;
clrscr();
printf(" Filename : ");
gets(fname);
if ((fp = fopen(fname,"rt")) == NULL)
{
fprintf(stderr, "Cannot open input file.\n");

```

```

    return 1;
}
stop();
init();
clrscr();
ch = fgetc(fp);
while (ch != '%')
{
    // delay(50);
    // printf("%c",ch);
    ch = fgetc(fp);
}

printf("begin \n");
while (ch != EOF)
{
    /******* X FIRST *****/
    if (ch == 'X')
    {
        i=0;
        while ((ch != 'Y') && (ch != '\n'))
        {
            ch = fgetc(fp);
            x[i] = ch;
            i++;
        }
        x[i] = '\0';
        .switch(strlen(x))
        {
            case 5: x[i] = '\0';break;
            case 6: x[i-1] = '\0';break;
        }
    }
}

```

```

    case 7: x[i-2] = '\0';break;
    case 8: x[i-3] = '\0';break;
}

```

```

//***** FIND WAY *****

```

```

    xn = atoi(x);
    xl = (xn - xo);
    if ( xl >= 0 )
    {
        printf("x_right= %d mm. ",xl);
        X_right(xl);
    }
    else {
        xl = 0-xl;
        printf("x_left= %d mm. ",xl);
        X_left(xl);
    }

    xo = xn;
    if (ch == '\n')
    {
        printf("Drill");
        drill();
    }
}

```

```

//***** Y second X *****

```

```

else if (ch == 'Y')
{
    i=0;
    while ( ch != '\n')
    {

```

```

        ch = fgetc(fp);
        y[i] = ch;
        i++;
    }
    y[i] = '\0';
    switch(strlen(y))
    {
        case 5: y[i] = '\0';break;
        case 6: y[i-1] = '\0';break;
        case 7: y[i-2] = '\0';break;
        case 8: y[i-3] = '\0';break;
    }

    yn = atoi(y);
    yl = (yn - yo);
    if (yl >= 0)
    {
        printf(" y_up= %d mm. ",yl);
        Y_up(yl);
    }
    else {
        yl = 0-yl;
        printf("y_down= %d mm. ",yl);
        Y_down(yl);
    }

    yo = yn;

    printf(" Drill ");
    drill();
}

```

```

    }

//***** Y FIRST *****
if (ch == 'Y')
{
    i=0;
    while ( ch != '\n')
    {
        ch = fgetc(fp);
        y[i] = ch;
        i++;
    }
    y[i] = '\0';
    switch(strlen(y))
    {
        case 5: x[i] = '\0';break;
        case 6: y[i-1] = '\0';break;
        case 7: y[i-2] = '\0';break;
        case 8: y[i-3] = '\0';break;
    }

    yn = atoi(y);
    yl = (yn - yo);
    if (yl >= 0)
    {
        printf(" y_up= %d mm. ",yl);
        Y_up(yl);
    }
    else {
        yl = 0-yl;
        printf("y_down= %d mm. ",yl);
    }
}

```

```

    Y_down(yl);
    }

    yo = yn;
    printf(" Drill ");
    drill();
}

```

```

printf("%c" ,ch);
ch = fgetc(fp);

}
fclose(fp);
init();
gotoxy(50,20);
printf(" Any key to Exit ");
getch();

}

```

```

//***** FUNCTION *****
void init(void)
{
    int fns;
    char xc,yc;
    fns = 0;
    gotoxy(x_set,y_set);
    printf(" initial.");
}

```

```

xc = inport(0x379);
while ( xc == 0x7f)
{
    xc = inport(0x379);
    XY_dl(1);
}

```

```

xc = inport(0x379);
if ((xc == 0x5f) || (xc == 0x4f) ) fns =1;
while( fns != 1)
{
    printf(".");
    xc = inport(0x379);
    if ((xc == 0x5f) || (xc == 0x4f) ) fns =1;
    X_left(1);
}
fns = 0;

```

```

xc = inport(0x379);
if ((xc == 0x4f) || (xc == 0x6f) ) fns =1;
while( fns != 1)
{
    printf(".");
    Y_down(1);
    xc = inport(0x379);
    if (xc == 0x4f) fns =1;
}
XY_ur(50);
outport(0x378,0);

```

```
}  
void stop(void)  
{  
    outport(0x378,0);  
}  
void X_left(int lng)  
{  
    outport(0x378,0x8);delay(lng*100);  
    stop();  
}  
  
void X_right(int lng)  
{  
    outport(0x378,0x9);delay(lng*100);  
    stop();  
}  
void Y_up(int lng)  
{  
    outport(0x378,0x20);delay(lng*115);  
    stop();  
}  
  
void Y_down(int lng)  
{  
    outport(0x378,0x22);delay(lng*115);  
    stop();  
}  
void XY_dr(int lng)  
{  
    outport(0x378,0x2b);delay(lng*100);  
    stop();  
}
```

```

}
void XY_ul(int lng)
{
  outport(0x378,0x28);delay(lng*100);
  stop();
}

void drill()
{
  // gotoxy(x_set,y_set);
  // printf(" drill... ");

  outport(0x378,0x14);delay(8000);
  stop();
  outport(0x378,0x10);delay(8000);
  stop();
}

void plot()
{
  // gotoxy(x_set,y_set);
  // printf(" drill... ");

  outport(0x378,0x14);delay(6000);
  stop();
  outport(0x378,0x10);delay(6000);
  stop();
}

void XY_dl(int lng)
{
  outport(0x378,0x2a);delay(lng*100);
  stop();
}

```

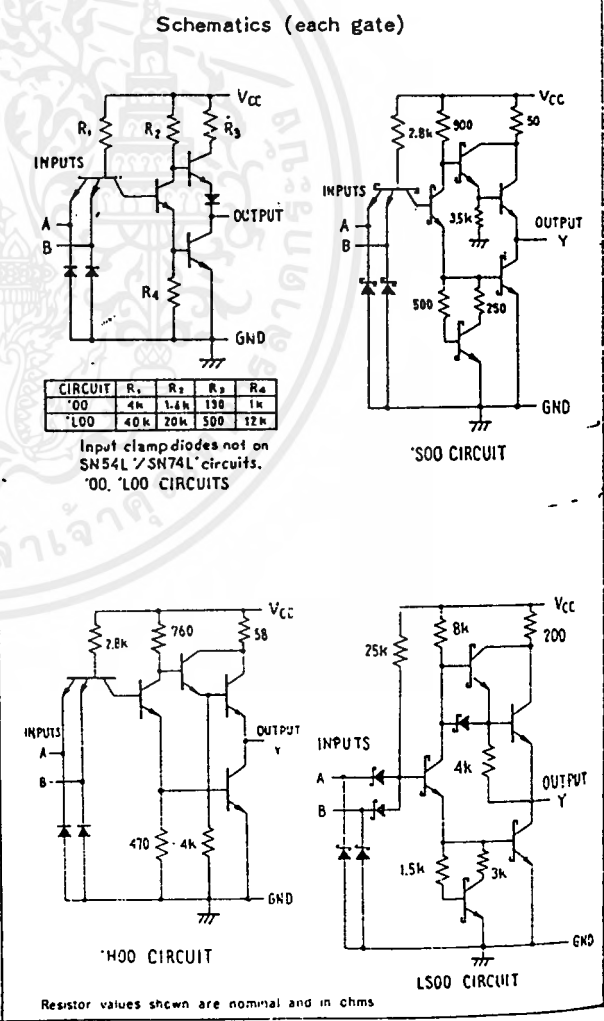
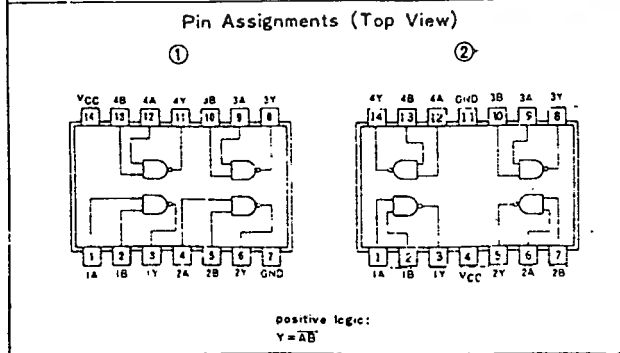
```
}  
void XY_ur(int lng)  
{  
  outport(0x378,0x29),delay(lng*100);  
  stop0;  
}
```



5400/7400 Quaduple 2-Input Positive-NAND Gate

	Schottky TTL				High-Speed TTL				Low-Power Schottky TTL				Standard TTL				Low-Power TTL								
	Device Type		Package		Device Type		Package		Device Type		Package		Device Type		Package		Device Type		Package						
			C	P			M	CF			C	P			M	CF			C	P	M	CF	C	P	M
T.I.	SN54S00	J	N	1	W	SN54H00	J	N	1	W	SN54LS00	J	N	1	W	SN5400	J	N	1	W	SN54L00	J	N	1	W
FAIRCHILD	FM54S00	D	N	1	W	FM54H00	D	N	1	W	FM54LS00	D	N	1	W	FM5400	D	N	1	W	FM54L00	D	N	1	W
MOTOROLA	MC3100	L	N	1	W	MC3100	L	N	1	W	MC3100	L	N	1	W	MC3100	L	N	1	W	MC3100	L	N	1	W
N.S.C.	DM74S00	J	N	1	W	DM54H00	J	N	1	W	DM54LS00	J	N	1	W	DM5400	J	N	1	W	DM54L00	J	N	1	W
PHILIPS	N74S00	J	N	1	W	UJH131-74H00	J	N	1	W	N74LS00	J	N	1	W	FJH131-7400	J	N	1	W	N74L00	J	N	1	W
SIGNETICS	NS4S00	F	N	1	W	S54H00	F	N	1	W	NS4LS00	F	N	1	W	S5400	F	N	1	W	NS4L00	F	N	1	W
SIEMENS	N74S00	F	N	1	W	N74H00	F	N	1	W	N74LS00	F	N	1	W	N7400	F	N	1	W	N74L00	F	N	1	W
FUJITSU						MB601	D	N	1	W	74LS00	M	N	1	W	MB400	I	N	1	W					
HITACHI	HD74S00	D	N	1	W						HD74LS00	D	N	1	W	HD7400	D	N	1	W	HD74L00	D	N	1	W
MITSUBISHI	M55000	P	N	1	W						M74LS00	P	N	1	W	M53200	P	N	1	W					
NEC	μPB2S00	D	N	1	W						74LS00	C	N	1	W	μPB201	D	N	1	W					
TOSHIBA											74LS00	C	N	1	W	TD3400 A	P	N	1	W					

Electrical Characteristics SN54LS00/SN74LS00			
absolute maximum ratings over operating free-air temperature range			
Supply voltage, V _{CC}	7V	Operating free-air temperature range	SN54LS -55°C to 125°C
Input voltage	7V	Storage temperature range	SN74LS 0°C to 70°C
Maximum storage voltage	5.5V	Storage temperature range	-65°C to 150°C
recommended operating conditions			
	SN54LS00		SN74LS00
Supply voltage, V _{CC}	MIN 4.5	MAX 5.5	MIN 4.75
High-level output current, I _{OH}	-400		MAX -400
Low-level output current, I _{OL}	4		MAX 8
Operating free-air temperature, T _A	-55		MAX 125
electrical characteristics over recommended operating free-air temperature range			
PARAMETER	TEST CONDITIONS †	MIN	TYP ± MAX
V _{IH} High-level input voltage		2	
V _{IL} Low-level input voltage		0.8	
V _I Input clamp voltage	V _{CC} = MIN, I _I = -18mA	-1.5	
V _{OH} High-level output voltage	V _{CC} = MIN, V _{IH} = V _{IH} max, I _{OH} = MAX	2.7	3.4
V _{OL} Low-level output voltage	V _{CC} = MIN, V _{IH} = 2V, I _{OL} = 4mA	0.2	0.4
I _I Input current at maximum input voltage	V _{CC} = MAX, V _I = 7V	0.1	
I _{IH} High-level input current	V _{CC} = MAX, V _{IH} = 2.7V	20	
I _{IL} Low-level input current	V _{CC} = MAX, V _{IL} = 0.4V	-0.4	
I _{OS} Short-circuit output current †	V _{CC} = MAX	SN54S Family	-20
		74LS Family	-18
I _{CC} H Supply current	V _{CC} = MAX	Total, outputs high	4
I _{CC} L Supply current	V _{CC} = MAX	Total, outputs low	12
I _{CC} Supply current	V _{CC} = 5V	Average per gate (50% duty cycle)	0.4
t _{PLH} Propagation delay time, low-to-high-level output	V _{CC} = 5V, T _A = 25°C, C _L = 15pF, R _L = 2kΩ	9	
t _{PHL} Propagation delay time, high-to-low-level output		10	



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5408/7408 Quadruple 2-Input Positive-AND Gate

	Schottky TTL				High-Speed TTL				Low-Power Schottky TTL				Standard TTL				Low-Power TTL			
	Device Type		Package		Device Type		Package		Device Type		Package		Device Type		Package		Device Type		Package	
	C	P	M	CF	C	P	M	CF	C	P	M	CF	C	P	M	CF	C	P	M	CF
T.I.	SN54S08	J	J	W1					SN54LS08	J	J	W1	SN5408	J	J	W1				
	SN74S08	J	N1						SN74LS08	J	N1		SN7408	J	N1					
FAIRCHILD	FM54S08 / FM74S08	D	F3		FM54H08 / FM74H08	D	F3		FM54LS08 / FM74LS08	D	F3		FM5408 / FM7408	D	F3					
	FC74S08, FC9508	D	F3		FC74H08, FC94H08	D	F3		FC74LS08, FC94LS08	D	F3		FC7408, FC9408	D	F3					
MOTOROLA					MC3101	L	F1						MC7408	P						
					MC3001	L	F1		SN74LS08	P										
N.S.C.					DM54H08	J	N1		DM54LS08	P			DM5408	J	N1		DM54L08			
					DM74H08	J	N1		DM74LS08	P			DM7408	J	N1		DM74L08			
PHILIPS	N74S08				N74H08				N74LS08				N7408							
					S54H08	F	A1	W1					S5408	F	A1	W1				
SIGNETICS	N74S08	A			N74H08	F	A1		N74LS08	A			N7408	F	A1					
SIEMENS													FLH381	J						
FUJITSU									74LS08	M										
HITACHI									HD74LS08	P			HD7408 / HD2550	P						
MITSUBISHI									1A74LS08	P			M53208	P						
NEC									74LS08	C			μ Pd234	DL						
TOSHIBA													TD3408	P						

Electrical Characteristics SN54LS08/SN74LS08

absolute maximum ratings over operating free-air temperature range

Supply voltage V _{CC}	7V	Operating free-air temperature range	SN54LS	-55°C to 125°C
Input voltage	7V	temperature range	SN74LS	0°C to 70°C
Intermittent voltage	5.5V	Storage temperature range		-65°C to 150°C

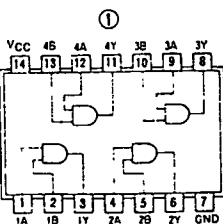
recommended operating conditions

	SN54LS08			SN74LS08			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage V _{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current I _{OH}			-800			-800	μ A
Low-level output current I _{OL}			1			8	mA
Operating free-air temperature T _A	-55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range

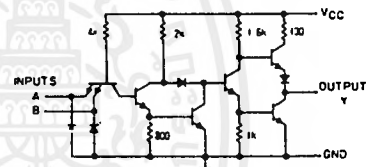
PARAMETER	TEST CONDITIONS †	MIN	TYP ‡	MAX	UNIT
V _{IH}	High-level input voltage		2		V
V _{IL}	Low-level input voltage			0.8	V
V _I	Input clamp voltage	V _{CC} =MIN, I _I =-15mA		-1.5	V
V _{OH}	High-level output voltage	V _{CC} =MIN, V _{IH} =2V, I _{OH} =MAX	2.7	3.4	V
V _{OL}	Low-level output voltage	V _{CC} =MIN, V _{IH} =2V, I _{OL} =4mA	0.25	0.4	V
I _I	Input current at maximum input voltage	V _{CC} =MAX, V _I =7V		0.1	mA
I _{IH}	High-level input current	V _{CC} =MAX, V _{IH} =2.7V		20	μ A
I _{IL}	Low-level input current	V _{CC} =MAX, V _{IL} =0.4V		-0.4	mA
I _{OS}	Short circuit output current †	V _{CC} =MAX		-20 to -100	mA
I _{CCS}	Supply current	V _{CC} =MAX		2.4 to 4.8	mA
I _{CCL}	Supply current	V _{CC} =MAX		4.4 to 8.8	mA
I _{CC}	Quiescent current	V _{CC} =5V, Average per gate (50% duty cycle)		0.85	mA
t _{PLH}	Propagation delay time, low-to-high-level output	V _{CC} =5V, T _A =25°C, C _L =15pF, R _L =2k Ω		8 to 15	ns
t _{PHL}	Propagation delay time, high-to-low-level output			10 to 20	ns

Pin Assignment (Top View)

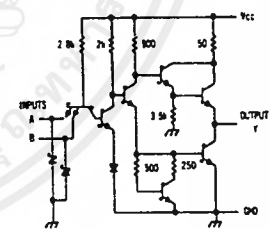


positive logic:
Y=AB

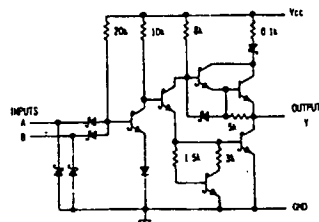
Schematics (each gate)



7408 CIRCUIT



74LS08 CIRCUIT



54LS08 CIRCUIT

Resistor values shown are nominal and in ohms.

† For conditions shown as MIN or MAX, use the appropriate values specified under recommended operating conditions.

‡ All typical values are at V_{CC}=5V, T_A=25°C.

• No1 more than one output should be shorted at a time, and for SN54S/SN74S, duration of output short circuit should not exceed one second.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



5474/7474 Dual D-Type Positive-Edge-Triggered Flip-Flop with Preset and Clear

	Schottky TTL				High-Speed TTL				Low-Power Schottky TTL				Standard TTL				Low-Power TTL						
	Device Type				Package				Device Type				Package				Device Type				Package		
T.I.	SN54S74	J	Q	W	SN54H74	J	Q	W	SN54LS74	J	Q	W	SN5474	J	Q	W	SN54L74	J	Q	W			
FAIRCHILD	FM54S74/FM9S74	DD	ND		FM54H74/FM9H74	DD	ND		FM54LS74/FM9LS74	DD	ND		FM5474/FM974	DD	ND		FM54L74/FM9L74	DD	ND				
MOTOROLA	FC74S74/FC9S74	DD	ND		FC74H74/FC9H74	DD	ND		FC74LS74/FC9LS74	DD	ND		FC7474/FC974	DD	ND		FC74L74/FC9L74	DD	ND				
N.S.C.	DM74S74		ND		DM54H74	J	Q	ND	DM54L74	J	Q	ND	DM5474	J	Q	ND	DM54L74	J	Q	ND			
PHILIPS	N74S74		Q		GJ131/74H74		Q		N74LS74		I		FJ131/7474		Q		S5474		Q				
SIGNETICS	S54S74				S54H74	F	Q	A	Q	S54L74	F	Q	A	Q		Q		S5474	F	Q			
SIEMENS	N74S74				N74H74	F	Q	A	Q	N74LS74		A	Q	N7474	F	Q	A	Q					
FUJITSU									74LS74		MA		MB420		Q	MD							
HITACHI	HD74S74		Q	P					HD74LS74		P	Q	HD7474/HD2510		Q	P							
MITSUBISHI	M74S74		P	Q					M74LS74		F	Q	M5327/M5374		P	Q							
NEC	74S74		C	Q					74LS74		C	Q	μPB214		D	C							
TOSHIBA													TD3474A		P	Q							

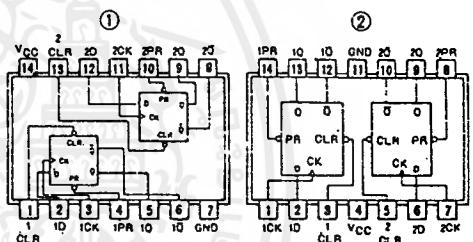
Electrical Characteristics SN54LS74/SN74LS74

absolute maximum ratings over operating free-air temperature range								
Supply voltage, V _{CC}	TV	Operating free-air temperature range	SN54LS					
Input voltage	5 V		SN74LS					
		Storage temperature range	-65°C to 150°C					
recommended operating conditions								
		SN54LS74	SN74LS74					
		MIN	NOM	MAX	MIN	NOM	MAX	UNIT
Supply voltage, V _{CC}		4.5	5	5.5	4.75	5	5.25	V
High-level output current, I _{OH}				-400				μA
Low-level output current, I _{OL}				4			8	mA
Input delay to	Clock high	25		25				ns
	Preset or clear low	25		25				
	High-level data	25		25				
Input setup time, t _{su}	Low-level data	20		20				ns
Input hold time, t _{hd}		5		5				ns
Operating free-air temperature, T _A		-55		125			70	°C

electrical characteristics over recommended operating free-air temperature range

PARAMETER	TEST CONDITIONS †	MIN	TYP ‡	MAX	UNIT
V _{IH}	High-level input voltage		2		V
V _{IL}	Low-level input voltage			0.8	V
V _I	Input clamp voltage	V _{CC} = MIN, I _I = -18 mA		-1.5	V
V _{OH}	High-level output voltage	V _{CC} = MIN, V _{IH} = 2V, V _{IL} = 0.8 V, I _{OH} = MAX	2.7	3.4	V
V _{OL}	Low-level output voltage	V _{CC} = MIN, V _{IH} = 2V, V _{IL} = 0.8 V, I _{OL} = 4 mA	0.25	0.4	V
I _I	Input current at maximum input voltage	D, J, K, Clear; Preset; Clock		0.1	mA
I _{IH}	High-level input current	D, J, K, Clear; Preset; Clock		20	μA
I _{IL}	Low-level input current	D, J, K, Clear; preset; Clock		-0.4	mA
I _{OS}	Short-circuit output current	Series 54LS; Series 74LS		-20	mA
I _{CC}	Supply current (Average per flip-flop)	V _{CC} = MAX, See Note 1		4	mA
f	clock frequency	V _{CC} = 5V, T _A = 25°C, C _L = 15pF, R _L = 2kΩ	25	33	MHz
t _{PLH}	from clear, preset or clock (as appropriate) to Q or Q̄		13	25	ns
t _{PHL}			25	40	

Pin Assignments (Top View)

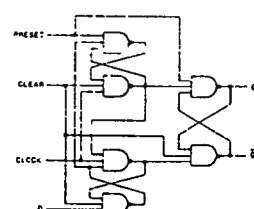


Functional Table

'74, 'H74, 'LS74, 'S74 (See Note 2)

INPUTS		OUTPUTS			
PRESET	CLEAR	CLOCK	D	Q	Q̄
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H*	H*
H	H	↑	H	L	L
H	H	↓	L	L	H
H	H		L	X	O ₀

Functional Block Diagram



'74, 'S74, '74, 'LS74, 'L74 DUAL D FLIP FLOP WITH CLEAR AND PRESET

- NOTES: 1 With all outputs open, I_{CC} is measured with the Q and Q̄ outputs high in turn. At the time of measurement, the clock input is grounded.
 2 H = high level (steady state), L = low level (steady state), X = irrelevant.
 ↑ = transition from low to high level.
 O₀ = the level of Q below the indicated most conditions were established.
 * This configuration is metastable, that is, it will not persist when preset and clear inputs return to their inactive (high) level.

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
 ‡ All typical values are at V_{CC} = 5V, T_A = 25°C.
 * Not more than one output should be shorted at a time.
 † t_{PLH} = propagation delay time, low-to-high-level output;
 † t_{PHL} = propagation delay time, high-to-low-level output;
 ‡ The arrow indicate the edge of the clock pulse used for reference: ↑ for the rising edge.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้