

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

แขนกล

ROBOT ARM



โดย

นางสาวจันทจิรา จำแสง

นายเจษฎา สายใจ

นายรัช สนวนันท์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

๒๒๗
๒๕๔๑

เลขทมิ.....
เลขทะเบียน..... 34049
วัน, เดือน, ปี..... 1 ต.ค. 2541

แขนกล
ROBOT ARM

โดย

นางสาวจันทจิรา คำแสง เลขประจำตัว 38014064
นายเจษฎา สายใจ เลขประจำตัว 38014087
นายรัช สนวนันท์ เลขประจำตัว 38014195

อาจารย์ที่ปรึกษา
อาจารย์โกศล ชวนขยัน

ปฏิญานิพนธ์สำหรับปฏิญาวិชากรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2541

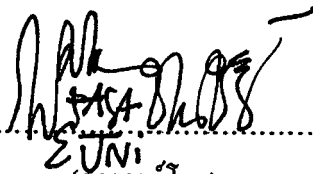
ปริญญาานิพนธ์ปีการศึกษา 2541

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง แขนกล

ผู้จัดทำ

1. นางสาวจันทิฉิรา ฉ่ำแสง เลขประจำตัว 38014064
2. นายเจษฎา สายใจ เลขประจำตัว 38014087
3. นายรัช สนวนันท์ เลขประจำตัว 38014195



.....อาจารย์ที่ปรึกษา

(อาจารย์โกศล ชวนขยัน)

แขนกล

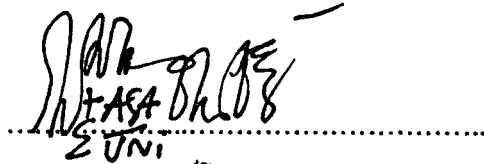
ROBOT ARM

นางสาวจันทจิรา ฉ่ำแสง เลขประจำตัว 38014064

นายเจษฎา สายใจ เลขประจำตัว 38014087

นายรัช สอนนนท์ เลขประจำตัว 38014195

โครงการนี้ได้รับการตรวจสอบแล้ว พร้อมทั้งจะทำการสอบได้



(อาจารย์โกศล ชวนชน)

อาจารย์ที่ปรึกษา

แขนกล

นางสาวจันทจิรา ฉั่วแสง

นายเจษฎา สายใจ

นายรัช สนวนันท์

อาจารย์โกศล ชวนขยัน (อาจารย์ที่ปรึกษา)

ปีการศึกษา 2541

บทคัดย่อ

โครงการนี้นำเสนอแขนกลที่ทำงานโดยใช้สเต็ปมิงมอเตอร์เป็นตัวขับเคลื่อนให้แขนกลเคลื่อนที่ไปยังตำแหน่งที่ต้องการ การควบคุมแขนกลเป็นแบบรูปปิดซึ่งมีความแม่นยำและมีเสถียรภาพ โดยควบคุมจากคอมพิวเตอร์ผ่านวงจรเชื่อมต่อ แขนกลนี้ถูกออกแบบให้มีลักษณะคล้ายกับแขนของมนุษย์ ดังนั้นมันจึงสามารถทำงานได้รอบทิศทางและทำงานได้ในบริเวณกว้าง เนื่องจากเป็นแขนกลที่มีความแม่นยำพอสมควรจึงเหมาะสำหรับนำไปใช้งานที่ต้องการความละเอียด เช่น งานเจาะแผ่นลายวงจร เป็นต้น

ROBOT ARM

Miss Janjira Chamsang

Mr.Jedsada Saijai

Mr.Tawat Suannan

Mr.Kosol Chaunkayan (Advisor)

1998

Abstract

This report is presented the robot arm that moved to the expected point by stepping motor. The controlling of the robot arm is closed loop type providing a high accuracy. We control it by computer via the interface circuit. This robot arm is designed similar to a human arm so it can move around itself and provide large working area. Because this robot arm provide high accuracy so it suitable to used as drilling print circuit board machine.

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
สารบัญ	III
สารบัญรูป	V
บทที่ 1 บทนำ	1
บทที่ 2 แขนกล	2
2.1 คุณลักษณะของแขนมนุษย์	6
2.2 การแบ่งประเภทของหุ่นยนต์อุตสาหกรรม	6
2.3 การเคลื่อนที่ของหุ่นยนต์	10
2.4 end effector	11
2.5 การสั่งงาน	14
บทที่ 3 ทฤษฎีของสเต็ปปีงมอเตอร์	17
3.1 ชนิดของสเต็ปปีงมอเตอร์	19
3.2 รูปแบบการทำงานของสเต็ปปีงมอเตอร์	21
3.3 การควบคุมการทำงานของสเต็ปปีงมอเตอร์	23
3.4 การหาจุดเริ่มต้นและจุดสิ้นสุด (Opto-Interrupter)	26
บทที่ 4 แหล่งจ่ายไฟตรงและหลักการเชื่อมต่อกับคอมพิวเตอร์	30
4.1 แหล่งจ่ายไฟตรง	30
4.2 ระบบเชื่อมต่อกับคอมพิวเตอร์	33
บทที่ 5 การออกแบบแขนกล	44
5.1 แหล่งส่งกำลัง	44
5.2 ระบบส่งกำลัง	44
5.3 การออกแบบโครงสร้างในส่วนต่างๆของแขนกล	47
บทที่ 6 การออกแบบวงจรทางไฟฟ้าและระบบควบคุมแขนกลโดยคอมพิวเตอร์	56
6.1 การออกแบบไฟตรง	56
6.2 การออกแบบวงจรขยายสัญญาณเพื่อขับมอเตอร์	61
6.3 การออกแบบตัวบอกรหัสและวงจรป้อนกลับ	63

	หน้า
6.4 การออกแบบวงจรคัปปลิ่ง	68
6.5 การออกแบบการ์ดอินเตอร์เฟซและ วิธีการควบคุมการหมุนของมอเตอร์	71
6.6 อัลกอริทึมของโปรแกรมภาษาซี	78
6.7 การเคลื่อนที่ของแขนกลและการเขียนโปรแกรมควบคุม	79
บทที่ 7 การทดลองและสรุปผลการทดลอง	105
7.1 ผลการทดลอง	105
7.2 สรุปผลการทดลอง	105
7.3 แนวทางแก้ไข	106
ภาคผนวก	
กิตติกรรมประกาศ	
เอกสารอ้างอิง	

สารบัญรูป

รูปที่		หน้า
รูปที่ 2.1	แสดงลักษณะของแขนกลในอุตสาหกรรม	4
รูปที่ 2.2	แสดงการทำงานของแขนกลในงานอุตสาหกรรม	5
รูปที่ 2.3	แสดงแบบที่มีการเคลื่อนไหวในแบบพิกัดเชิงขั้ว	7
รูปที่ 2.4	แสดงแบบที่มีการเคลื่อนไหวในระบบพิกัดทรงกระบอก	8
รูปที่ 2.5	แสดงแบบการเคลื่อนไหวในลักษณะข้อต่อ	8
รูปที่ 2.6	แสดงแบบของหุ่นยนต์ในระบบพิกัดคาร์ทีเซียน	9
รูปที่ 2.7	แสดงลักษณะการเคลื่อนที่ของแขนแบบต่างๆ	10
รูปที่ 2.8	แสดงการเคลื่อนที่ของมือ	11
รูปที่ 2.9	แสดงลักษณะของ Gripper แบบต่างๆ	12
รูปที่ 2.10	แสดงลักษณะของมือที่เป็นแบบต่างๆ	13
รูปที่ 2.11	แสดงลักษณะการสั่งงานแบบ Lead Through โดยอาศัย Robot simulator	15
รูปที่ 3.1	แสดงลักษณะของสแต็ปปีงมอเตอร์	17
รูปที่ 3.2	แสดงสแต็ปปีงมอเตอร์แบบปรับค่าความต้านทานแม่เหล็กได้	19
รูปที่ 3.3	แสดงสแต็ปปีงมอเตอร์แบบแม่เหล็กถาวร	20
รูปที่ 3.4	แสดงสแต็ปปีงมอเตอร์แบบไฮบริด	21
รูปที่ 3.5	แสดงการทำงานแบบเฟสเดี่ยว	22
รูปที่ 3.6	แสดงการทำงานแบบเฟสคู่	22
รูปที่ 3.7	แสดงการควบคุมแบบลูปเปิด	26
รูปที่ 3.8	แสดงวงจรขับกระแสแบบขั้วเดี่ยว	24
รูปที่ 3.9	แสดงวงจรขับกระแสแบบสองขั้ว	25
รูปที่ 3.10	แสดงการควบคุมแบบลูปเปิด	26
รูปที่ 3.11	แสดงการหาจุดเริ่มต้นและจุดสิ้นสุด	27
รูปที่ 3.12	แสดงตัวอย่างของตัวบอกรหัสแบบสัมบูรณ์	28
รูปที่ 3.13	แสดงตัวอย่างของตัวบอกรหัสแบบเพิ่มค่า	29
รูปที่ 4.1	แสดงส่วนประกอบพื้นฐานของส่วนประกอบไฟตรง	30

รูปที่		หน้า
รูปที่ 4.2	แสดงวงจรเรียงกระแสแบบครึ่งคลื่น	30
รูปที่ 4.3	แสดงวงจรเรียงกระแสแบบเต็มคลื่น โดยใช้หม้อแปลงมีแท่งกลาง	31
รูปที่ 4.4	แสดงวงจรเรียงกระแสแบบเต็มคลื่นแบบบริจัน	31
รูปที่ 4.5	แสดงวงจรกรองกระแสโดยใช้ตัวเก็บประจุ	32
รูปที่ 4.6	แสดงวงจรกรองกระแสโดยใช้ตัวเหนี่ยวนำ	32
รูปที่ 4.7	แสดงวงจรรักษาระดับแรงดันโดยใช้ไอซี	32
รูปที่ 4.8	แสดงการเชื่อมต่อระหว่างคอมพิวเตอร์และแขนกล	34
รูปที่ 4.9	แสดงตำแหน่งของขาสัญญาณต่างบนสล็อตคอมพิวเตอร์	36
รูปที่ 4.10	แสดงตารางการใช้งานพอร์ตที่ตำแหน่งต่างๆ	37
รูปที่ 4.11	แสดงตำแหน่งขาต่างๆของไอซี 8255	40
รูปที่ 4.12	แสดงความหมายแต่ละบิตของรหัสควบคุม	42
รูปที่ 4.13	แสดงรหัสควบคุม 98H	43
รูปที่ 5.1	แสดงลักษณะของเกลียวตัวหนอน	45
รูปที่ 5.2	แสดงลักษณะของเฟืองตรงอัตราทดเฟือง (Velocity Ratio)	46
รูปที่ 5.3	แสดงโครงสร้างของแขนกล	47
รูปที่ 5.4	แสดงลักษณะการวางตัวของแขนกลที่เกิดค่าทอร์กสูงสุด	48
รูปที่ 5.5	แสดงโครงสร้างส่วนฐาน	49
รูปที่ 5.6	แสดงโครงสร้างของส่วนหัวไหล่	53
รูปที่ 5.7	แสดงโครงสร้างส่วนแขนท่อนล่าง	54
รูปที่ 5.8	แสดงโครงสร้างส่วนแขนท่อนบน	55
รูปที่ 6.1	แสดงวงจรรักษาระดับแรงดันโดยใช้ไอซีตระกูล 78XX	59
รูปที่ 6.2	แสดงวงจรรักษาระดับแรงดันที่สามารถปรับค่าแรงดันเอาท์พุทได้	59
รูปที่ 6.3	แสดงวงจรช่วยจ่ายกระแส	60
รูปที่ 6.4	แสดงวงจรขยายกระแสแบบขั้วเดียว	61
รูปที่ 6.5	แสดงวงจรบอกรหัสแบบเพิ่มค่าได้	64
รูปที่ 6.6	แสดงวงจรตรวจจับสัญญาณโดยใช้แสง	64
รูปที่ 6.7	แสดงวงจรปรับแต่งสัญญาณพัลส์	66
รูปที่ 6.8	แสดงวงจรป้อนกลับ	67
รูปที่ 6.9	แสดงวงจรออปโตคัปเปิลอร์แบบพื้นฐาน	68

รูปที่	หน้า
รูปที่ 6.10	69
รูปที่ 6.11	70
รูปที่ 6.12	72
รูปที่ 6.13	73
รูปที่ 6.14	74
รูปที่ 6.15	77
รูปที่ 6.16	79
รูปที่ 6.17	80
รูปที่ 6.18	80
รูปที่ 6.19	82
รูปที่ 7.1	105

บทที่ 1

บทนำ

ในปัจจุบันเทคโนโลยีด้านหุ่นยนต์ (Robot) ได้พัฒนาขึ้นอย่างรวดเร็วอันเป็นผลสืบเนื่องมาจากความต้องการความสะดวกสบายของมนุษย์ ทั้งเพื่อความสะดวกรวดเร็ว และแก้ปัญหาการขาดแคลนแรงงาน อีกทั้งการใช้คนเพื่อผลิตงานในโรงงานอุตสาหกรรม ก็มักก่อให้เกิดปัญหามาก และประสิทธิภาพการทำงานมีก็อยู่จำกัด จึงเริ่มหันมาใช้หุ่นยนต์แทนคนมากขึ้น

มนุษย์ได้พยายามที่จะคิดค้นสร้างสิ่งต่าง ๆ ขึ้นมาใช้ และวิทยาการสมัยใหม่ก็อำนวยความสะดวกการสร้างหุ่นยนต์ไม่ใช่เรื่องยากอีกต่อไป จากเหตุผลดังกล่าวข้างต้น ก็ทำให้พอที่จะมองเห็นได้ว่าในอนาคตอันใกล้หุ่นยนต์จะเข้ามามีบทบาทกับชีวิตมนุษย์อย่างแน่นอน ดังนั้นจึงมีความจำเป็นไม่น้อยที่จะต้องศึกษาและพัฒนารูปแบบของหุ่นยนต์ ให้สามารถใช้งานได้จริงในอุตสาหกรรม เพื่อรองรับความเจริญของการใช้หุ่นยนต์ที่จะเกิดขึ้นในอนาคต

ในโครงการนี้จะเป็นการออกแบบและสร้างแขนกล โดยมีวัตถุประสงค์ของงานคือ

1.1 ศึกษาระบบการอินเตอร์เฟส และการใช้โปรแกรมภาษาซีในการควบคุมแขนกล

1.2 ทำให้ได้ต้นแบบของแขนกล และพัฒนารูปแบบให้ดีขึ้นต่อไป

1.3 เพื่อให้เกิดความรู้ความเข้าใจในส่วนระบบเกียร์ แรงบิด และการเคลื่อนที่ของมอเตอร์

1.4 เพื่อควบคุมให้แขนกลสามารถเคลื่อนที่ไปยังตำแหน่งที่ต้องการได้

ในการออกแบบเลือกระบบขับเคลื่อนโดยใช้ สเต็ปป์มอเตอร์ เริ่มติดตั้งชุดเฟืองส่งกำลังและโครงสร้าง หลังจากนั้นทำการสั่งงานโดยใช้โปรแกรมภาษาซีผ่านวงจรมอเตอร์เฟส เพื่อไปควบคุมการทำงานของสเต็ปป์มอเตอร์

ส่วนรายละเอียดของการควบคุมการจ่ายแรงดันและกระแสมีอยู่ภายในรายงาน ส่วนในระบบเฟืองนั้น เราจะใช้เฟืองที่มีจำนวนซี่ฟันละเอียด เพื่อให้เกิดความคงที่ในขณะที่ทำการเคลื่อนที่

ขอบเขตงานที่ทำ

1. ทำการควบคุมแขนกลให้เคลื่อนที่ได้

2. สามารถเขียนโปรแกรมควบคุมการทำงานให้แขนกลเคลื่อนที่ไปยังตำแหน่งที่ต้องการ

3. สามารถทำการเชื่อมต่อบนคอมพิวเตอร์กับแขนกลได้อย่างสมบูรณ์

ประโยชน์ที่ได้รับ

1. ทำให้ได้ต้นแบบในการพัฒนาแขนง
2. รู้ปัญหาและแนวทางในการพัฒนาแขนง
3. เข้าใจในการเขียนโปรแกรม แก้ไขคำสั่ง ซึ่งอาจมีการกำหนดขอบเขตงานของแขนงมากยิ่งขึ้น
4. ได้ใช้ความคิดและแนวคิดสร้างสรรค์ในการทำงานโปรเจ็คเป็นการฝึกให้มีการทำงานเป็นระบบและมีความรับผิดชอบยิ่งขึ้น

บทที่ 2

แขนกล

เมื่อกล่าวถึงหุ่นยนต์คนส่วนใหญ่มักจะนึกถึงโครงเหล็กขนาดใหญ่ลักษณะคล้ายมนุษย์ เคลื่อนตัวอย่างช้าๆ ดังที่เห็นในภาพยนตร์แนววิทยาศาสตร์ทั่วไป แต่ในปัจจุบันได้ถูกพัฒนาขึ้นมาใช้งานในโรงงานอุตสาหกรรมขนาดใหญ่ มากขึ้น โดยอุตสาหกรรมเหล่านี้ครอบคลุมตั้งแต่โรงงานผลิตรถยนต์เครื่องจักร จนแม้แต่การผลิตไมโครคอมพิวเตอร์ก็ใช้หุ่นยนต์เข้ามาช่วยในการผลิตแต่หุ่นยนต์อุตสาหกรรมเหล่านี้มีลักษณะที่แตกต่างออกไป จากภาพพจน์เดิมที่นิยายวิทยาศาสตร์วาดภาพไว้มาก เพราะส่วนใหญ่จะมีลักษณะเป็นแขนและมือซึ่งใช้ทำงานตามคำสั่งเท่านั้น

สมาคมหุ่นยนต์แห่งประเทศไทย (BRITISH ROBOT ASSOCIATION : BRA) กำหนดนิยามของหุ่นยนต์อุตสาหกรรมไว้ว่า

“คือ อุปกรณ์ที่มีอย่างน้อย 4 องศาแห่งอิสระ (Degree of freedom) ที่สามารถโปรแกรมใหม่ได้ ซึ่งได้รับการออกแบบมาสำหรับการจับย้ายและขนส่ง ชิ้นส่วนเครื่องมือหรือส่วนประกอบพิเศษ โดยการอาศัยการเคลื่อนที่ที่ได้รับการ โปรแกรมไว้สำหรับงานเฉพาะอย่าง และโปรแกรมนี้อาจสามารถเปลี่ยนแปลงได้”

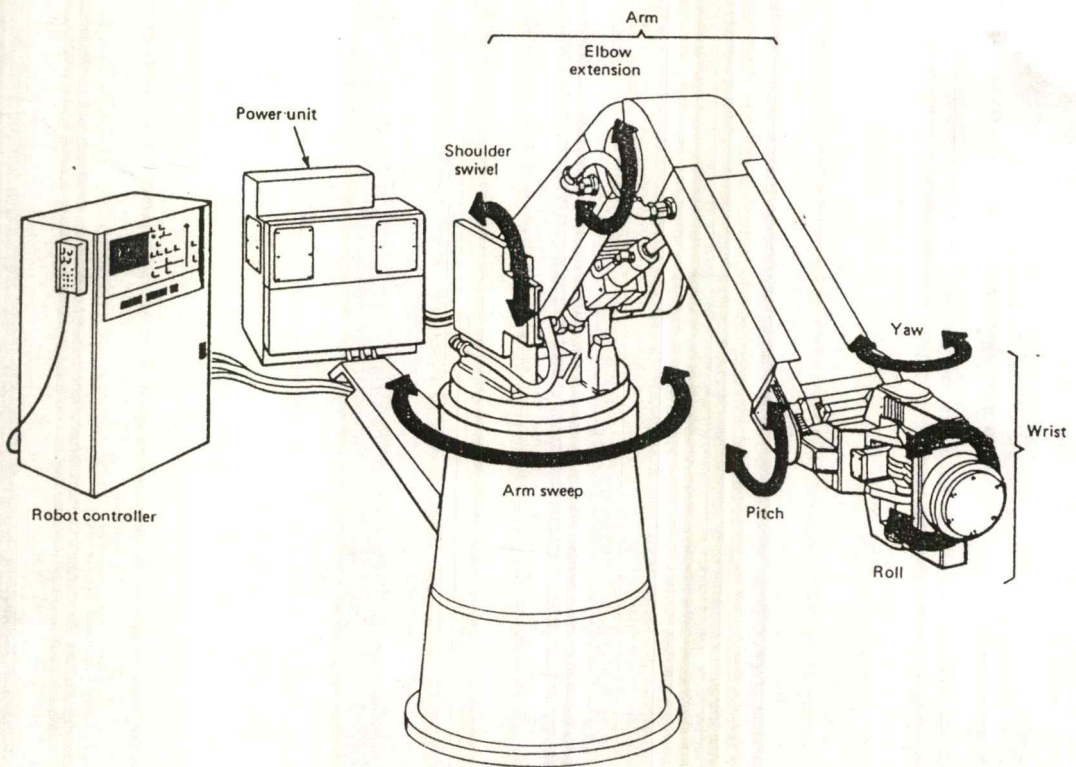
สิ่งหนึ่งที่สำคัญมากในการกำหนดนิยามของ สมาคมหุ่นยนต์แห่งประเทศไทยคือ สิ่งที่เป็นหุ่นยนต์จะต้องมี 4 องศาแห่งอิสระเป็นอย่างน้อย

สถาบันทางหุ่นยนต์ของอเมริกา (ROBOTICS INSTITUTE OF AMERICA:RIA) ให้นิยามว่า

“เป็นมือจับ (manipulator) หลายหน้าที่การทำงาน ที่สามารถโปรแกรมใหม่ได้ ได้รับการออกแบบเพื่อการเคลื่อนย้ายวัสดุ ชิ้นส่วน เครื่องมือหรืออุปกรณ์พิเศษ โดยอาศัยการเคลื่อนที่ที่ได้รับการ โปรแกรมไว้ใช้งานเฉพาะอย่าง”

นิยามของสถาบันทางหุ่นยนต์ของอเมริกานี้ได้รับการยอมรับกันอย่างแพร่หลายในฐานะที่เป็นนิยามของหุ่นยนต์อุตสาหกรรม

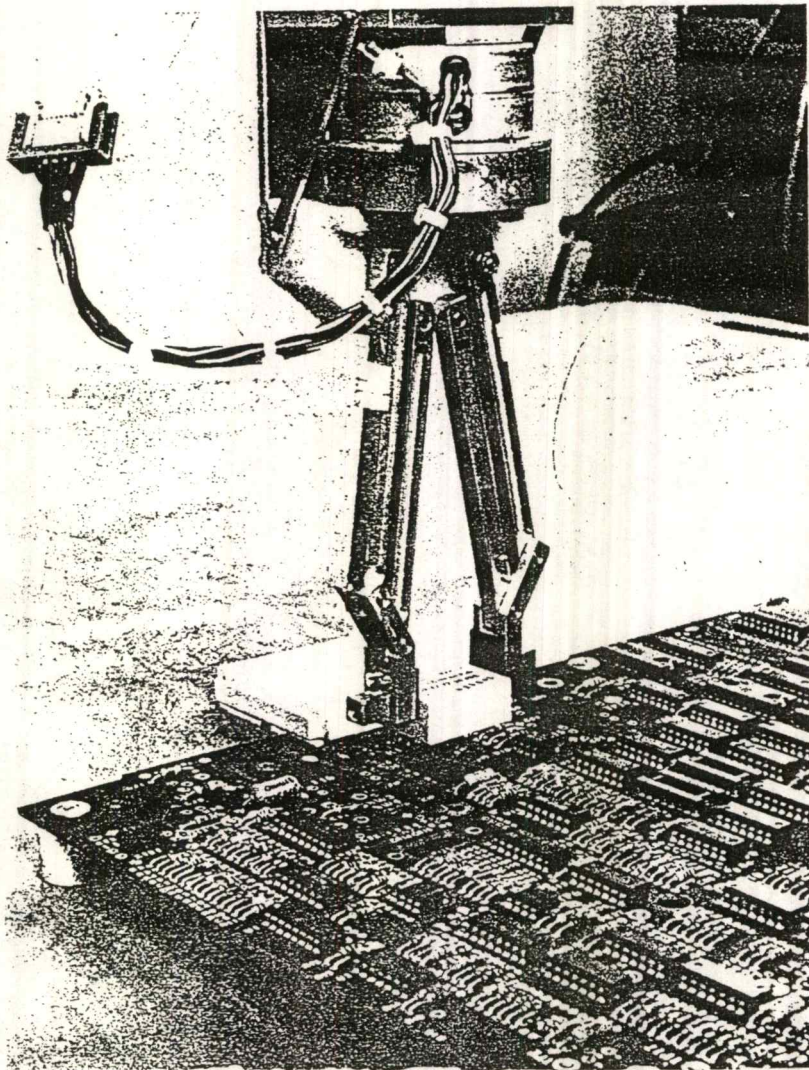
ตัวอย่างของงานที่ใช้หุ่นยนต์ช่วยงานมีมากมายอาทิเช่น การประกอบชิ้นส่วน การพัน
 綵การเชื่อม ฯลฯ โดยตำแหน่งที่หุ่นยนต์เคลื่อนที่ไปปฏิบัติงานนี้จะยังคงมีตลอดเวลาและสามารถ
 ทำงานซ้ำๆ ได้หลายๆ ครั้ง โดยไม่มีการผิดพลาด จนกว่าจะมีการเปลี่ยนแปลงคำสั่ง การทำงานของ
 หุ่นยนต์อุตสาหกรรมนี้ หลักการจะคล้ายกับการทำงานของอุปกรณ์ที่ควบคุมด้วยคำสั่งเชิงตัวเลข
 (numerical control machine) ในลักษณะของการทำงาน การ feedback ฯลฯ โดยมีแนวโน้มว่าจะมี
 การนำไปใช้ในอุตสาหกรรมขนาดเล็กเพิ่มขึ้นเพราะสามารถควบคุมได้ด้วยไมโครคอมพิวเตอร์ทำให้
 ให้การเปลี่ยนแปลงแก้ไขคำสั่งทำได้สะดวก



รูปที่ 2.1 ลักษณะของแขนกลในอุตสาหกรรม

The Intelledex 605 robot can memorize the shapes of 100 different parts. Here it is loading an odd-shaped component into place on a printed circuit board.

Intelledex, Inc.



รูปที่ 2.2 การทำงานของแขนกลในอุตสาหกรรม

2.1 คุณลักษณะของแขนมนุษย์

ในเมื่อเราต้องการที่จะสร้างหุ่นยนต์ให้มีคุณลักษณะเหมือนขนของมนุษย์เราจึงความจำเป็นที่จะต้องศึกษาโครงสร้างและลักษณะการเคลื่อนไหวของแขนมนุษย์ก่อน

แขนของมนุษย์ประกอบด้วยส่วน 2 ส่วนที่เห็นแยกกันอย่างชัดเจน คือ ส่วนที่เป็นข้อมือ ซึ่งมีข้อต่อหลักอยู่ 3 ข้อต่อ และอีกส่วนคือแขน ซึ่งมีข้อต่อหลักอยู่ 2 ข้อต่อ (หัวไหล่และข้อศอก) ก่อนอื่นเรามาคูที่ส่วนแรกก่อน ข้อมือของมนุษย์มีลักษณะดังต่อไปนี้

การหมุนข้อมือ (ROLL (angle) สามารถหมุนได้	$-180 < \theta < +90 = 270$
การพับข้อมือ (PITCH (angle) สามารถพับได้	$-90 < \theta < +50 = 140$
การบิดข้อมือ (YAW (angle) สามารถบิดได้	$-45 < \theta < +15 = 60$

การเคลื่อนที่ของข้อมือทั้ง 3 แบบ มีการเคลื่อนที่ที่เป็นอิสระต่อกัน ดังนั้นเราจึงให้การเคลื่อนที่แต่ละแบบเป็น 1 องศาแห่งอิสระ

ส่วนที่สองคือ แขน ประกอบด้วยข้อต่อหลัก 2 ข้อต่อ แต่มี 3 องศาแห่งอิสระคือ 2 องศาแห่งอิสระที่หัวไหล่ และ อีก 1 องศาแห่งอิสระที่ข้อศอก แต่หัวไหล่ของแขนกลจะมีเพียง 1 องศาแห่งอิสระ ลักษณะหนึ่งที่สำคัญของโครงสร้างของแขน คือ อัตราส่วนระหว่างความยาวของแขนท่อนบนกับแขนท่อนล่าง จะมีค่าราวๆ 1.2 : 1 นั่นคือล่างจะต้องสั้นกว่าแขนท่อนบน ถ้าเราออกแบบให้แขนท่อนล่างยาวกว่าแขนท่อนบน จะทำให้ประสิทธิภาพการทำงานของแขนกลลดลง

2.2 การแบ่งประเภทของหุ่นยนต์อุตสาหกรรม

เราสามารถแบ่งประเภทของแขนกลโดยพิจารณาจากพื้นฐานใหญ่ๆ ดังนี้

1. โครงสร้างภายนอกและการเคลื่อนที่ของหุ่นยนต์
2. ทางเดินของแขนกลซึ่งขึ้นอยู่กับควบคุมการเคลื่อน

2.2.1 การแบ่งประเภทของแขนกลโดยอาศัยการเคลื่อนที่และ โครงสร้างภายนอก

หุ่นยนต์อุตสาหกรรม มีรูปร่างลักษณะแตกต่างกันไปแล้วแต่ลักษณะการใช้งาน โดยขีดความสามารถในการเคลื่อนที่ของมือจะแตกต่างกันไปแต่โดยทั่วไปหากจะจัดแบ่งออกเป็นกลุ่มๆ แล้วก็คงจะแบ่งออกเป็น 4 กลุ่มคือ

2.2.1.1 หุ่นยนต์ที่มีการเคลื่อนไหวได้ใน ระบบพิกัดขั้ว (polar coordinate configuration)

2.2.1.2 หุ่นยนต์ที่มีการเคลื่อนไหวได้ในระบบพิกัดทรงกระบอก (cylindrical coordinate configuration)

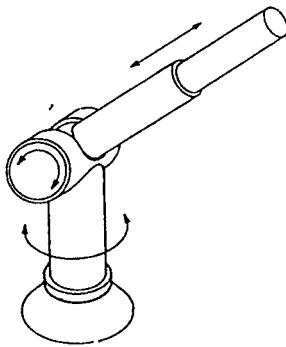
2.2.1.3 หุ่นยนต์ที่มีแขนในลักษณะข้อต่อ (joint arm configuration)

2.2.1.4 หุ่นยนต์ที่มีการเคลื่อนไหวได้ในระบบคาร์ทีเซียน

ลักษณะของหุ่นยนต์ทั้งสี่ชนิดแสดงไว้ในรูปที่ 2.3 - 2.6 โดยรายละเอียดจะเป็นดังนี้

ก) หุ่นยนต์ที่มีการเคลื่อนไหวได้ในระบบพิกัดขั้ว

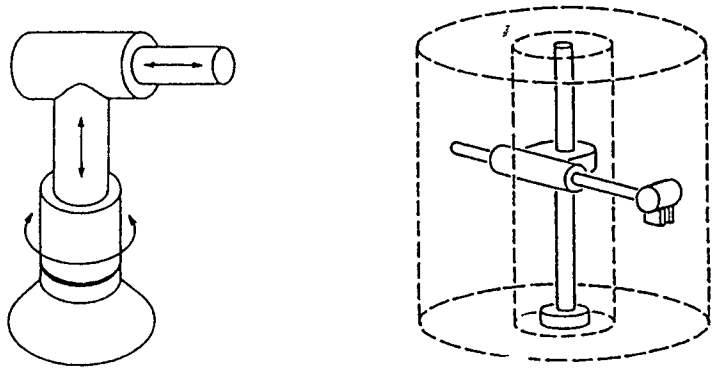
ขอบเขตการเคลื่อนไหวของหุ่นยนต์ที่มีลักษณะเช่นนี้ อาจเรียกอีกอย่างหนึ่งว่าอยู่ในระบบพิกัดทรงกลม (spherical coordinate configuration) ทั้งนี้เพราะลักษณะบริเวณที่แขนหุ่นยนต์จะกวาดไปได้จะมีลักษณะเป็นรูปวงกลม (ดังรูปที่ 2.3)



รูปที่ 2.3 แบบที่มีการเคลื่อนไหวในระบบพิกัดเชิงขั้ว

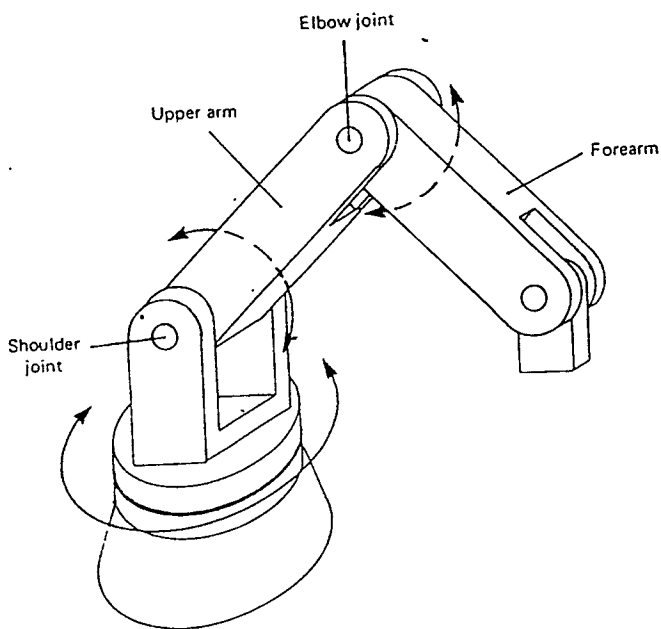
ข) หุ่นยนต์ที่มีเคลื่อนไหวได้ในระบบพิกัดเชิงขั้ว

ขอบเขตของหุ่นยนต์ลักษณะนี้มีลักษณะการกวาดแขนเป็นรูปทรงกระบอกดังรูป



รูปที่ 2.4 แบบที่มีการเคลื่อน ไหวในระบบที่กักทรงกระบอก

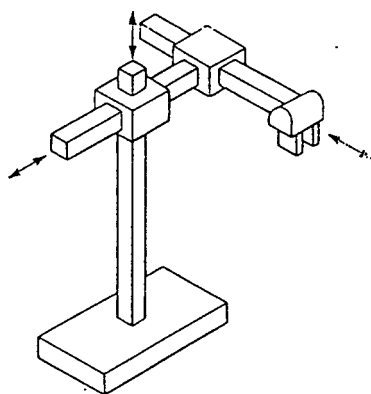
ค) หุ่นยนต์ที่มีลักษณะแขนเป็นข้อต่อหุ่นยนต์ประเภทนี้ จะเลียนแบบลักษณะแขนของมนุษย์โดยส่วนต่างๆ สามารถยึดหดได้เช่นเดียวกับส่วนไหล่ ศอก ข้อศอก และข้อมือ ทำให้การเคลื่อนที่คล่องตัวโดยจะมีรัศมีที่กวาดได้เป็นลักษณะรูปครึ่ง ทรงกลม (ดังรูปที่ 2.5)



รูปที่ 2.5 แบบการเคลื่อนไหวในลักษณะข้อต่อ

ง) หุ่นยนต์ที่มีการเคลื่อนไหวได้ในระบบพิกัดคาร์ทีเซียน

หุ่นยนต์ประเภทนี้จะเคลื่อนที่ได้ใน 3 แกนหลักคือ แกน x, แกน y, และแกน z ของระบบพิกัดคาร์ทีเซียน ดังนั้นขอบเขตการกวาดของมือจะเป็นรูปทรงบล็อกละเอียด (ดังรูปที่ 2.6)



Cartesian coordinate robot

รูปที่ 2.6 แบบของหุ่นยนต์ในระบบพิกัดคาร์ทีเซียน

ทั้งนี้ไม่ว่าจะเป็นหุ่นยนต์ในลักษณะใด งานหลักของหุ่นยนต์ก็คือการจัดการกับเม้าหมาย โดยจะใช้ส่วนที่อยู่ปลายสุดของ แขนเป็นตัวจัดการ ส่วนนี้จะทำหน้าที่คล้ายมือของมนุษย์มีชื่อเต็มเฉพาะว่า (end effector) ดังนั้นในการนำไปใช้งานจะต้องคำนึงความสามารถของส่วนนี้เป็นหลัก โดยปกติแล้วการเคลื่อนที่ของส่วนมือหรือ end effector นี้จะทำได้ใน 6 ระดับขั้นความเสรี (degree of freedom) ซึ่งจะช่วยให้คล้ายการทำงานของมือมนุษย์

2.2.2 การแบ่งประเภทของแขนกลโดยอาศัยทางเดินของแขนกล

การควบคุมทางเดินของแขนกลแบ่งเป็น 2 แบบใหญ่ๆ คือ

1. การควบคุมแบบจุด [Point-to-point (PTP) control]
2. การควบคุมแบบต่อเนื่อง [Continuous path (CP) control]

การควบคุมแบบจุด แขนกลจะได้รับการตั้ง โปรแกรมให้หยุดที่จุดใดจุดหนึ่งและใช้จุดนั้นในการกำหนดจุดที่ต้องการให้หยุดจุดต่อไป แม้ว่าจะไม่สามารถควบคุมการเคลื่อนที่ระหว่างจุดให้อยู่ในระหว่างเส้นทางที่ต้องการ แต่การเคลื่อนที่ไปยังจุดที่กำหนดก็ยังคงต้องตามที่กำหนด ในทางตรงข้ามที่ควบคุมแบบต่อเนื่อง จะสามารถเคลื่อนที่ตามเส้นทางต่างๆ ได้อย่างถูกต้อง

การควบคุมแบบจุดจะมีความแม่นยำสูงมากทั้งในแง่ความเที่ยงตรง และการทำงานซ้ำๆ แม้ว่าการควบคุมแบบต่อเนื่องจะเคลื่อนไหว ได้นูน่นวลมากกว่าในเส้นทางที่ได้รับระบุไว้ แต่ก็มีข้อเสียในด้านความเร็ว ซึ่งเป็นฟังก์ชันของจังหวะการเคลื่อนที่ที่คำนวณ โดยคอมพิวเตอร์ในลักษณะเวลาจริง (real time) ข้อเสียนี้จะทำให้ความเร็วของการเคลื่อนที่แบบต่อเนื่องลดลงถึง 15 -25 % และประสิทธิภาพการทำงานจะลดลงเมื่อเปรียบเทียบกับแขนกลตัวเดียวกันที่ควบคุมแบบจุด

2.3 การเคลื่อนที่ของหุ่นยนต์

การเคลื่อนที่ของหุ่นยนต์สามารถแบ่งการเคลื่อนที่ออกได้ 2 ส่วนคือ

2.3.1 การเคลื่อนที่ของตัวและแขน (body and arm)

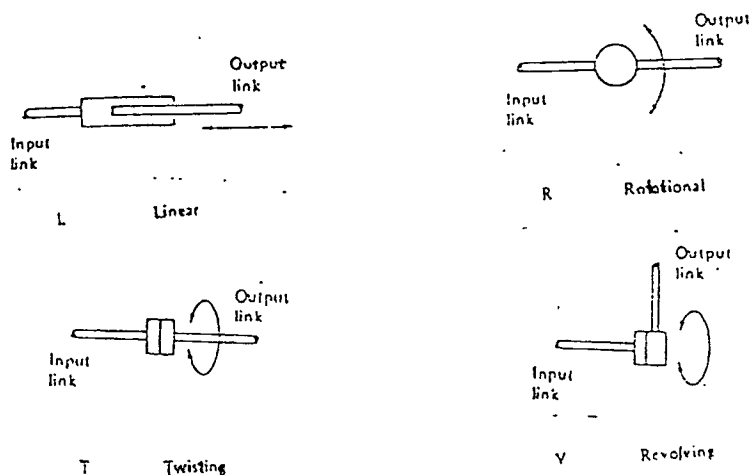
การเคลื่อนที่ของตัวและแขนแบ่งตามลักษณะของข้อหมุน ได้ดังนี้คือ

2.4.1.1 การเคลื่อนที่แบบเส้นตรง (Linear)

2.4.1.2 การเคลื่อนที่แบบหมุนรอบจุดหมุน (Rotational)

2.4.1.3 การเคลื่อนที่แบบบิดรอบจุดหมุน (Twisting)

2.4.1.4 การเคลื่อนที่แบบหมุนตั้งฉาก (Revolving)



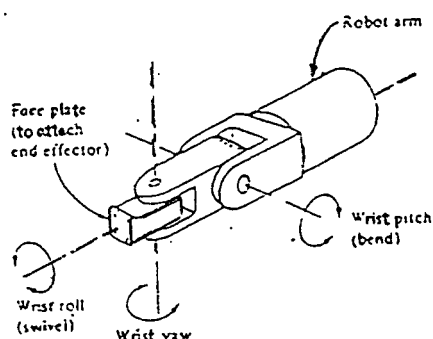
รูปที่ 2.7 ลักษณะการเคลื่อนที่ของแขนแบบต่างๆ

2.3.2 การเคลื่อนที่ของมือ

2.4.2.1 หมุน (Roll) บางครั้งเรียกว่า Swivel ข้อมือหมุนรอบแกนของแขน

2.4.2.2 บิด (Pitch) บางครั้งเรียกว่า Bend ข้อมือยกขึ้นลงในแนวตั้ง

2.4.2.3 ต่าย (Yaw) หมายถึงบิด ไปมาทางซ้ายและขวาของแกนมือ



รูปที่ 2.8 การเคลื่อนที่ของมือ

2.4 End Effector

End Effector หรือ “มือ” ของหุ่นยนต์นี้เป็นอุปกรณ์ที่หุ่นยนต์ใช้ทำงานจริง (ส่วนแขนนั้นใช้เพื่อเคลื่อนตำแหน่ง “มือ” นี้ให้ไปอยู่ในจุดที่ต้องการเท่านั้น) งานที่ส่วนนี้ทำก็มีอาทิเช่น การจับชิ้นส่วน การบัดกรีตามจุด การพ่นสี ฯลฯ ดังนั้นลักษณะของส่วนนี้จึงต่างกันออกไป สุดแท้แต่จินตนาการของผู้ออกแบบ ซึ่งต้องออกแบบให้เหมาะกับงานแต่ถ้าจะมีการแบ่งกันเป็นชนิดแล้ว “End Effector” นี้จะออกได้เป็นสองกลุ่มใหญ่ๆ คือ แบบมือจับ (Gripper) แบบมือที่เป็นอุปกรณ์ (Tool as End Effector)

2.4.1 Gripper

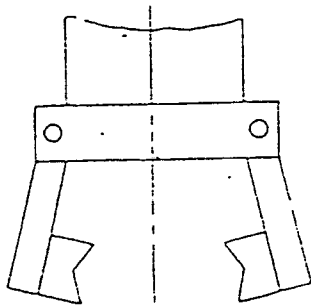
Gripper นี้เป็นอุปกรณ์ที่ใช้จับชิ้นส่วนหรือเครื่องมือ โดยการออกแบบมิได้กลายลักษณะ ทั้งนี้ขึ้นอยู่กับการใช้งาน เช่น

2.4.1.1 แบบมือคีบ (อาศัยแรงเสียดทานในการจับ) ใช้งานได้กับวัสดุทุกประเภท แต่หากเป็นวัสดุอบบางอาจต้องมีอุปกรณ์อื่นช่วยเสริม

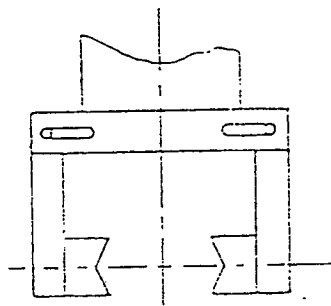
2.4.1.2 แบบ Suction Cup อาศัยคุณสมบัติการทำให้เกิดสภาพสุญญากาศ เป็นตัวดึงชิ้นงาน ใช้ได้กับวัสดุผิวเรียบเท่านั้น

2.4.1.3 แบบ ตะขอ (Hook) สำหรับเกี่ยวของของ เช่น ของที่อยู่บนสายพาน ฯลฯ

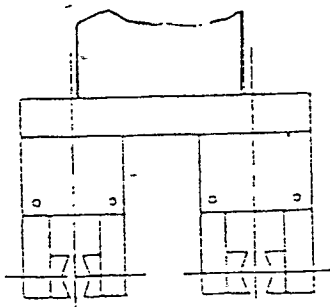
2.4.1.4 แบบ ตัก (Scoop) สำหรับการดำถึงของที่เป็นของเหลวหรือฝุ่นผงตัวอย่าง Gripper แบบต่างๆ แสดงไว้ในรูป



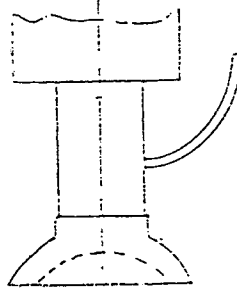
แบบมือคีบ



แบบ Suction Cup



แบบ ตะขอ (Hook)



แบบ ตัก (Scoop)

รูปที่ 2.9 ลักษณะของ Gripper แบบต่าง ๆ

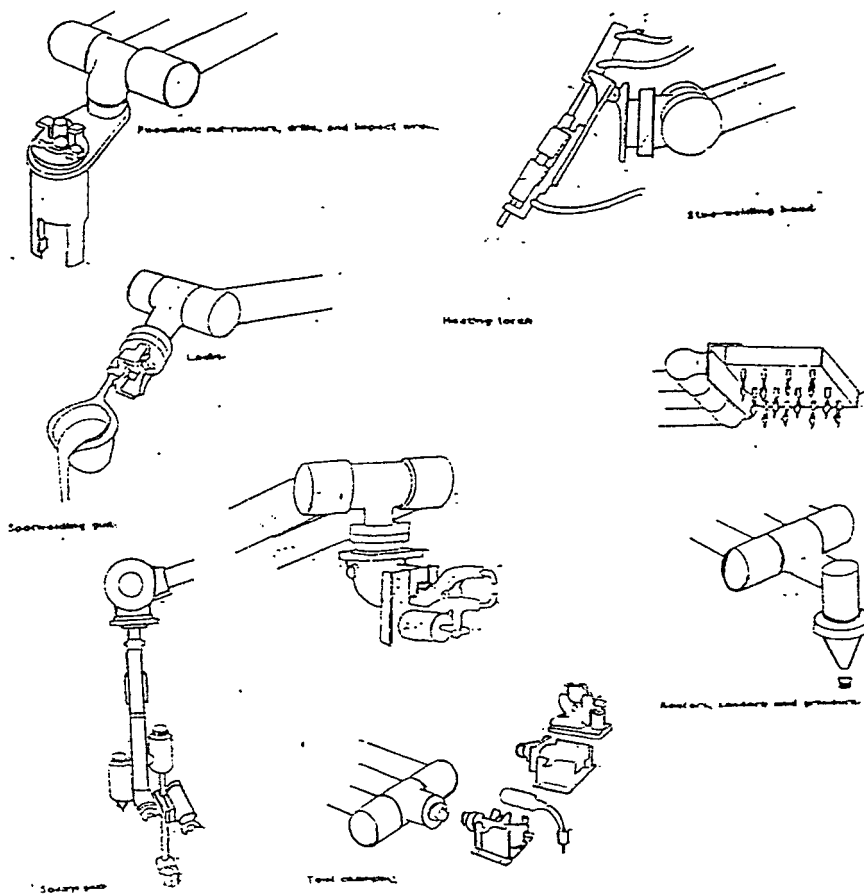
2.4.2 มือที่เป็นอุปกรณ์ (Tool as End Effector)

เนื่องจาก Gripper มีข้อจำกัดใช้ได้เฉพาะกับงานหยิบจับวัตถุเท่านั้น แต่การใช้งานของหุ่นยนต์มีขอบเขตกว้างขวาง จึงได้มีการออกแบบอุปกรณ์ใช้งานติดเข้าไปในส่วนปลายหรือส่วนมือนี้ เพื่อใช้ในการทำงาน โดยอาจจะเป็นอุปกรณ์ที่ติดตายหรืออาจถอดเปลี่ยนได้ ก็แล้วแต่การออกแบบ ตัวอย่างของมือที่เป็นอุปกรณ์นี้มีมากมาย อาทิเช่น

- คัดอุปกรณ์เชื่อมเฉพาะจุด (Spot Welding Gun)
- คัดอุปกรณ์พ่นสี (Spray Paint Gun)
- คัดอุปกรณ์ сверан (Drilling Spindle)

๑๓๑

ตัวอย่างของมือที่เป็นอุปกรณ์นี้แสดงไว้ในรูปที่ 2. 10



รูปที่ 2. 10 ลักษณะของมือที่เป็นอุปกรณ์แบบต่างๆ

2.5 การตั้งงาน

การตั้งงานให้หุ่นยนต์อุตสาหกรรมทำงานตามที่กำหนดนี้ อาจทำได้หลายลักษณะ คือ

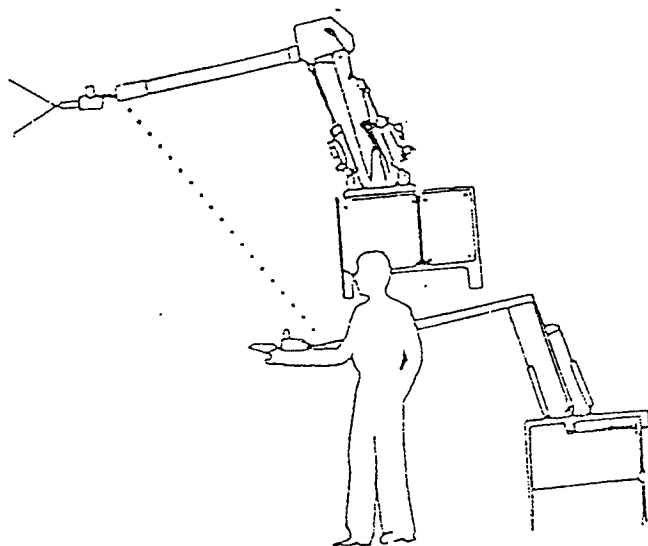
- แบบ Manual
- แบบ Lead Through
- แบบการตั้งงานด้วยโปรแกรม

2.5.1 การตั้งงานแบบ Manual

การตั้งงานแบบนี้เป็นการตั้งงานที่ง่ายที่สุด และใช้กันมากในการตั้งงานให้หุ่นยนต์ ทำงานแบบจากจุดหนึ่งไปจุดหนึ่ง (Point - To - Point) โดยวิธีการจะเป็นการเคลื่อนแกนต่างๆ ไปยังตำแหน่งที่ต้องการโดยการกดปุ่มบังคับ ดังนั้นลักษณะนี้จึงคล้ายกับการ Set Up มากกว่าการโปรแกรม กล่าวคือมีการเคลื่อนไปยังจุดเริ่มต้นแล้วก็กำหนดค่าไว้ จากนั้นก็เคลื่อนไปยังตำแหน่งต่อไป แล้วกำหนดค่าไว้ ฯลฯ โดยค่าที่กำหนดไว้จะเก็บไว้ในหน่วยความจำเรียกออกมาใช้ได้โดยง่าย

2.5.2 การตั้งงานแบบ Lead Through

การตั้งงานแบบนี้คล้าย ๆ กับการจับมือหุ่นยนต์ให้เคลื่อนที่ไปตามแนวที่ต้องการ ในลักษณะที่เป็นการเคลื่อนที่แบบต่อเนื่องแล้วจึงเก็บค่าตำแหน่งต่าง ๆ ที่เคลื่อนที่ไป พร้อมกับความเร็วที่เคลื่อนที่ แต่ในทางปฏิบัติไม่สามารถทำได้ เนื่องจากข้อจำกัดในแง่เทคนิค (อาทิเช่น ระบบเกียร์ ฯลฯ) ดังนั้นจึงมีผู้คิดค้นอุปกรณ์ซึ่งเลียนแบบแขนหุ่นยนต์โดยสามารถเคลื่อนที่ไปยังตำแหน่งต่าง ๆ ได้ (ดังรูป 2.9) อุปกรณ์นี้มีชื่อเรียกว่า Robot Simulator ซึ่งการทำงานก็ใช้หลักการข้างต้นกล่าวคือ ผู้ตั้งงานจะเลื่อนไปยังตำแหน่งที่ต้องการ โดยมีแนวทางการเคลื่อนที่กำหนดได้ โดยตำแหน่งต่าง ๆ จะถูกเก็บไว้ในหน่วยความจำของคอมพิวเตอร์ ซึ่งพบว่าวิธีนี้เป็นการ โปรแกรมโดยตรงวิธีหนึ่ง ซึ่งข้อเสียก็มีบ้างอาทิเช่น ความละเอียดไม่ดี ไม่สามารถกำหนดความเร็วได้ ต้องมีการซื้อ Simulator มาเพิ่มเข้ากับระบบซึ่งเป็นการสิ้นเปลืองและหากการตั้งการเกิดผิดพลาดอาจต้องทำใหม่ทั้งหมด ฯลฯ การตั้งงานชนิดนี้ใช้มากในงานด้านการพ่นสี ชิ้นส่วนรถยนต์ อุปกรณ์ ฯลฯ



รูปที่ 2.11 ลักษณะการทำงานแบบ Lead Through โดยอาศัย Robot Simulator

2.5.3 การทำงานด้วยโปรแกรม

การทำงานให้หุ่นยนต์อุตสาหกรรมการทำงานโดยการเรียกใช้โปรแกรมทำงานนี้ จะเป็นการให้ข้อมูลกับหุ่นยนต์เพื่อให้แขนของหุ่นยนต์เคลื่อนที่ไปยังตำแหน่งที่ต้องการและทำงานตามที่กำหนด โดยในปัจจุบันได้มีผู้เขียนภาษาคอมพิวเตอร์เพื่อสั่งให้หุ่นยนต์ทำงานมากมายหลายภาษา โดยโปรแกรมจะถูกเขียนบนเครื่องคอมพิวเตอร์แล้วส่งไปควบคุมหุ่นยนต์อีกต่อหนึ่ง

2.5.4 การประยุกต์การใช้งาน

ในการใช้หุ่นยนต์ช่วยในงานอุตสาหกรรมนี้เนื่องจากราคาค่อนข้างแพง ดังนั้นการพิจารณาตัดสินใจใช้หุ่นยนต์เหล่านี้จึงต้องพิจารณาเฉพาะงานที่จำเป็น เช่น

- งานที่อันตราย หรืองานที่มนุษย์ที่ไม่สะดวก เช่น บริเวณที่มีรังสี สารพิษ ฯลฯ
- งานที่ต้องทำซ้ำบ่อย ๆ ซึ่งหากมนุษย์ทำอาจผิดพลาดได้ง่าย เพราะเป็นงานที่น่าเบื่อ
- ใช้ในงานที่ต้องการความสามารถเฉพาะ เช่น งานยกของหนัก งานต้องการความละเอียด

๑๓๑

- ใช้ในงานที่ต้องทำต่อเนื่อง

๑๓๑

โดยลักษณะของงานที่ใช้มากก็อาทิเช่น

- งานหยิบจับชิ้นวัสดุ
- งานเชื่อม ไม่ว่าจะเป็นแบบเชื่อมจุด (spot welding) หรือ เชื่อมแบบต่อเนื่อง (arc welding)
- การพันสารซึ่งอาจมีกลิ่น มีเสียงรบกวน เป็นอันตรายต่อสุขภาพ
- การประกอบชิ้นงาน
 - งานตรวจเช็ค (Inspection)

ข้อดีของการใช้หุ่นยนต์ก็มีมากมายอาทิเช่น

- ความปลอดภัย งานที่อันตรายสามารถใช้หุ่นยนต์ทำได้
- ความสม่ำเสมอของงานดี เพราะเมื่อโปรแกรมแล้วก็จะทำตามขั้นตอนด้วยความเร็วที่กำหนด
- ใช้พลังงานน้อย เพราะการใช้หุ่นยนต์อาจไม่ต้องสนใจระบบถ่ายเทอากาศ เครื่องปรับอากาศก็

ไม่จำเป็น

- เพิ่มประสิทธิภาพการทำงาน เช่น งานเสร็จเร็วกว่าใช้แรงงานคน
 - ใช้วัสดุน้อย เช่นงานพันสี จะทำอย่างคงเส้นคงวา ทำให้ประหยัดสี

๑๓๑

แต่ขอบเขตในการประยุกต์ใช้งานจริงแล้วขึ้นอยู่กับจินตนาการของผู้นำไปใช้งานมากกว่า

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

บทที่ 3

ทฤษฎีของสเต็ปปีงมอเตอร์

สเต็ปปีงมอเตอร์เป็นอุปกรณ์ทางแม่เหล็กไฟฟ้าที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานกลในรูปการหมุน โดยมีโครงสร้างทั่วไปประกอบด้วยส่วนที่อยู่กับที่เรียกว่า สเตเตอร์ (Stator) และส่วนที่เคลื่อนที่ได้เรียกว่า โรเตอร์ (Rotor) ในสเต็ปปีงมอเตอร์มุมที่หมุนไปได้นั้นจะมีค่าเท่าๆกันเรียกว่า สเต็ป โดยจำนวนสเต็ปที่หมุนไปได้ขึ้นอยู่กับจำนวนของสัญญาณเข้าที่มีลักษณะเป็นลูกคลื่นพัลส์ค่าๆกัน ส่วนทิศทางในการหมุนของมอเตอร์ขึ้นอยู่กับทิศทางการกระตุ้นจากพัลส์ว่ามีทิศทางทวนเข็มนาฬิกาหรือตามเข็มนาฬิกา และความเร็วในการหมุนจะขึ้นอยู่กับความถี่ในการกระตุ้นของพัลส์

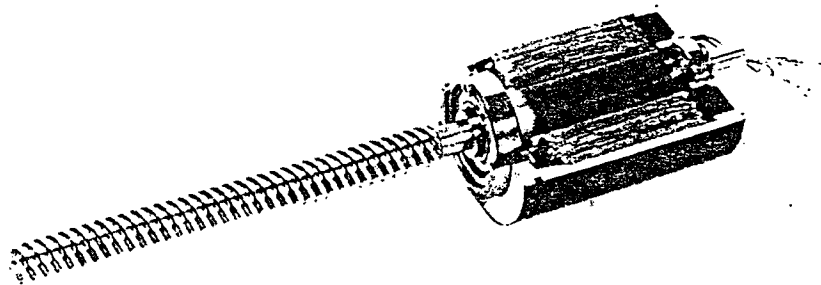


Fig. 2.8. Cutaway view of a single-stack VR motor. (By courtesy of MINEBEA Co., Ltd.)

รูปที่ 3.1 ลักษณะของสเต็ปปีงมอเตอร์

ข้อดีของสเต็ปปีงมอเตอร์

1. มีการทำงานที่เที่ยงตรง และสามารถต่อเชื่อมเข้ากับระบบคอมพิวเตอร์ได้ง่าย
2. โรเตอร์มีโมเมนต์ความเฉื่อย (Moment of inertia) ต่ำ
3. ควบคุมการทำงานได้ง่ายโดยใช้วงจรเชิงเลข (Digital Electronics) , คอมพิวเตอร์ส่วนบุคคล (PC) , ไมโครโพรเซสเซอร์ (Microprocessor) และระบบควบคุมเชิงเลขที่โปรแกรมได้ (Programmable Logic Controller : PLC) เป็นต้น

4. สามารถทำงานได้ทั้งระบบเปิด (Open Loop) และระบบปิด (Close Loop)
5. ไม่สร้างสนามแม่เหล็กที่จะไปรบกวนระบบอื่นๆรอบข้าง

ข้อเสียของสเต็ปป์มอเตอร์

1. ประสิทธิภาพในการทำงานค่อนข้างต่ำเมื่อเทียบกับมอเตอร์ชนิดอื่นๆ
2. ต้องการตัวควบคุมความเร็วของพัลส์ที่จ่ายกระแสให้กับลวดตัวนำเพื่อควบคุมความเร็วของมอเตอร์
3. โมเมนต์ความเฉื่อย (Moment of inertia) ของโพลด์ที่จะขับได้ต้องมีค่าไม่เกิน 4 เท่าของโมเมนต์ความเฉื่อยที่โรเตอร์
4. มีเสียงรบกวนเมื่อใช้งานที่ความเร็วสูงๆ

คุณลักษณะบางประการของสเต็ปป์มอเตอร์

1. แรงบิดคงค่า (Holding Torque) : คือแรงต้านการบิดสูงสุดที่ทำให้มอเตอร์ที่กำลังหมุนหยุดหมุน
2. แรงบิดหน่วง (Detent Torque) : คือแรงบิดสูงสุดที่ทำให้โรเตอร์ของมอเตอร์ที่ยังไม่ทำงานไม่มีการหมุน ค่าแรงบิดหน่วงนี้พบเฉพาะในมอเตอร์ที่มีแกนโรเตอร์ทำด้วยแม่เหล็กถาวรเท่านั้น
3. มุมในการหมุน (Step Angle) : คือมุมที่สเต็ปป์มอเตอร์หมุนได้ตามปกติเมื่อมีการจ่ายกระแสไฟฟ้าให้กับลวดตัวนำในสเตเตอร์ โดยค่าของมุมขึ้นอยู่กับลักษณะโครงสร้างของมอเตอร์นั้นๆ
4. ความเที่ยงตรงของมุมในการหมุน (Step Angle Accuracy) : คือค่าเปอร์เซ็นต์ที่เกินหรือขาดของค่ามุมในการหมุนปกติ
5. ค่าความต้านทานต่อหนึ่งขดลวด (Resistance per winding) : เป็นตัวกำหนดว่าจะมีกระแสไหลผ่านลวดตัวนำ โดยยิ่งค่าความต้านทานมีค่าน้อยกระแสก็จะไหลได้มาก ทำให้มอเตอร์มีค่าแรงบิดมากเพราะแรงบิดมีค่าแปรตามกระแส
6. ค่าความเหนี่ยวนำต่อหนึ่งขดลวด (Inductance per winding) : เป็นตัวบอกความเร็วในการเพิ่มขึ้นของปริมาณกระแสให้ถึงค่าที่ระบุไว้ของลวดตัวนำเมื่อมีการจ่ายกระแสให้ ถ้าค่าความเหนี่ยวนำนี้มีค่ามากการเพิ่มของกระแสก็จะใช้เวลานาน ทำให้การทำงานที่ความเร็วรอบสูงๆ ไม่ดีเท่าที่ควรชนิดของสเต็ปป์มอเตอร์
7. ความเฉื่อยที่โรเตอร์ (Rotor inertia) : เป็นตัวที่ใช้ในการคำนวณความเร็วของมอเตอร์
8. แรงบิดดึงเข้า (Pull in torque) : เป็นค่าแรงบิดที่กระทำต่อมอเตอร์ ที่ยังสามารถทำให้มอเตอร์เริ่มหมุนและหยุดหมุนได้ด้วยตัวเองตามจังหวะกระตุ้นของพัลส์ โดยไม่มีการข้ามจังหวะ
9. แรงบิดดึงออก (Pull out torque) : เป็นค่าแรงบิดสูงสุดที่กระทำต่อมอเตอร์ขณะที่หมุน โดยไม่ทำให้มอเตอร์หยุดหมุนและไม่มีการหมุนข้ามจังหวะ
10. ค่าความถี่ในการหมุนสูงสุด (Maximum starting frequency) : เป็นค่าความถี่สูงสุดของพัลส์ที่ทำให้มอเตอร์หมุนได้โดยไม่ข้ามจังหวะ

3.1 ชนิดของสเต็ปริงมอเตอร์

สามารถแยกตามลักษณะ โครงสร้าง ได้ดังต่อไปนี้คือ

1. แบบปรับค่าความต้านทานแม่เหล็กไฟฟ้าได้ (Variable Reluctance : VR)

ส่วนของโรเตอร์ (Rotor) ทำด้วยโลหะที่มีความซึมซับทางแม่เหล็กสูง เช่น เหล็กอ่อน (Soft Iron) โดยสร้างเป็นลักษณะซี่ของฟันขนาดเท่าๆกันเรียงตัวในแนวเดียวกันกับแกนหมุนของโรเตอร์ ในส่วนของสเตเตอร์ (Stator) จะทำเป็นขั้วโลหะที่มีเส้นลวดตัวนำพันอยู่โดยรอบ เมื่อจ่ายไฟฟ้าให้กับขดลวดเหล่านี้จะทำให้เกิดสนามแม่เหล็กไฟฟ้าขึ้น โดยมีฟลักซ์ (Flux) เกี่ยวข้องกันระหว่างสเตเตอร์และโรเตอร์ ฟลักซ์แม่เหล็กไฟฟ้าที่เกิดขึ้นนี้จะเหนี่ยวนำให้เกิดแรงบิด (Torque) เพื่อให้โรเตอร์หมุนไปยังตำแหน่งที่มีค่าความต้านทานแม่เหล็ก (Reluctance) น้อยที่สุด นั่นคือบริเวณที่มีช่องว่างระหว่างสเตเตอร์กับโรเตอร์น้อยที่สุด

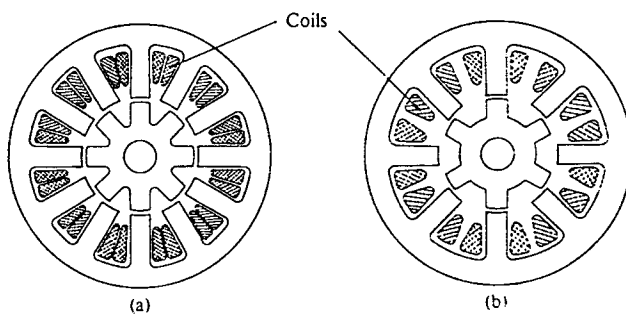


Fig. 2.15. Cross-sectional views of VR motors with 15° step angle. (a) Three-phase motor: number of stator teeth = 12; number of rotor teeth = 8. (b) Four-phase motor: number of stator teeth = 8; number of rotor teeth = 6.

รูปที่ 3.2 สเต็ปริงมอเตอร์แบบปรับค่าความต้านทานแม่เหล็กได้

จากรูปที่ 3.2 จะเห็นว่าในสภาวะเริ่มต้นที่ฟันบนโรเตอร์จะทำมุม 15° กับขั้วของสเตเตอร์ ดังนั้นเมื่อขดลวดตัวนำมีกระแสไหลก็จะทำให้โรเตอร์หมุนไป 15° คือตำแหน่งที่โรเตอร์อยู่ในแนวเดียวกันกับขั้วของสเตเตอร์ การที่จะทำให้สเตปป์มอเตอร์หมุนอย่างต่อเนื่องก็คือการจ่ายกระแสให้กับขดลวดตัวนำเรียงลำดับกันไป โดยถ้าต้องการให้มอเตอร์หมุนตามเข็มนาฬิกาก็ต้องจ่ายกระแสให้กับขดลวด A , B , C , D ตามลำดับ และถ้าต้องการให้มอเตอร์หมุนในทิศทวนเข็มนาฬิกาก็ต้องจ่ายกระแสในทิศทางตรงข้ามคือ D , C , B , A เป็นต้น สเตปป์มอเตอร์ชนิดนี้มีค่าแรงบิดหน่วย (Detain Torque) ค่าทำให้ไม่เป็นที่นิยมนำมาขับโหลดที่มีค่ามาก

2. แบบแม่เหล็กถาวร (Permanent Magnetic : PM)

มีโรเตอร์ที่ทำมาจากแม่เหล็กถาวรมีขั้วแม่เหล็กเหนือ-ใต้ ตรงข้ามกันในลักษณะคั้งได้ฉากกับแกนการหมุนของโรเตอร์ ส่วนของสเตเตอร์มีลักษณะคล้ายกับแบบปรับค่าความต้านทานแม่เหล็กได้ (VR) ทำให้มอเตอร์ชนิดนี้มีค่าแรงบิดหน่วยมากพอที่จะรับโหลดไว้ได้โดยไม่มีการหมุนในขณะที่ไม่มีการจ่ายกระแสไฟฟ้าให้กับขดลวดตัวนำในสเตเตอร์ การทำให้มอเตอร์หมุนใช้หลักการเดียวกันกับมอเตอร์แบบปรับค่าความต้านทานแม่เหล็กได้ (VR) ดังนั้นยังจำนวนขั้วของแม่เหล็กมีค่ามากเท่าใดมุมที่หมุนได้ในแต่ละสเต็ปก็จะมีค่าน้อยเท่านั้น ข้อเสียของมอเตอร์ชนิดนี้ก็คือมีขนาดใหญ่มาก และการทำให้โรเตอร์มีจำนวนขั้วแม่เหล็กมากๆทำได้ยาก ทำให้มุมในการหมุนมีค่ามาก

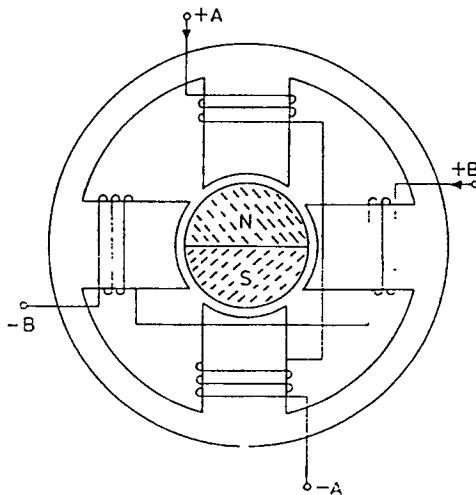


Fig. 1.8 Permanent-magnet stepping motor

รูปที่ 3.3 สเตปป์มอเตอร์แบบแม่เหล็กถาวร

3. แบบไฮบริด (Hybrid Stepping Motor)

เป็นมอเตอร์ที่มีลักษณะการทำงานผสมระหว่างมอเตอร์แบบปรับค่าความต้านทานแม่เหล็กได้ (VR) กับแบบแม่เหล็กถาวร (PM) โดยโรเตอร์ทำมาจากแม่เหล็กถาวร โดยมีขั้วแม่เหล็กเหนือ-ใต้ ในทิศทางเดียวกันกับแกนหมุนของโรเตอร์ ส่วนที่สเตเตอร์จะมีลักษณะขั้วเหมือนกับมอเตอร์ชนิด VR และ PM แต่การเชื่อมต่อและการพันลวดตัวนำจะแตกต่างกัน โดยในมอเตอร์แบบไฮบริดจะนำลวดตัวนำที่มีแปดต่างกันมาพันบนขั้วสเตเตอร์เดียวกันเรียกการพันแบบนี้ว่าแบบ ไบไฟลาร์ (Bifilar) ดังนั้นในการทำงานของแต่ละขั้วจะเกิดจากการทำงานของลวดตัวนำมากกว่าหนึ่งเฟส ลักษณะเช่นนี้ทำให้มอเตอร์แบบไฮบริดมีแรงบิดหน่วยที่มากและมีความละเอียดของมุมในการหมุนมาก

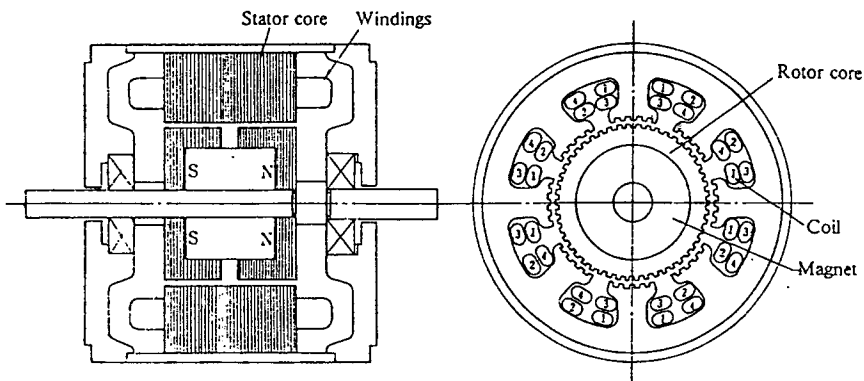


Fig. 2.32. Construction of a hybrid motor.

รูปที่ 3.4 สเต็ปปีงมอเตอร์แบบไฮบริด

3.2 รูปแบบในการทำงานของสเต็ปปีงมอเตอร์

1. แบบเฟสเดียว (Single phase excitation)

ทำการจ่ายกระแสให้กับลวดตัวนำของขั้วสเตเตอร์ทีละขดเรียงตามลำดับกันไปตามรูปที่ 3.5

Step Phase	1	2	3	4	5	6	7	8	9	10
Phase 1	▨				▨				▨	
Phase 2		▨				▨				▨
Phase 3			▨				▨			
Phase 4				▨				▨		

รูปที่ 3.5 แสดงการทำงานแบบเฟสเดี่ยว

2. แบบเฟสคู่ (Two phase excitation)

ทำการจ่ายกระแสให้ถ่วงตัวนำเป็นคู่ๆ เรียงตามลำดับ

Step Phase	1	2	3	4	5	6	7	8	9	10
Phase 1	▨	▨			▨	▨			▨	▨
Phase 2		▨	▨			▨	▨			▨
Phase 3			▨	▨			▨	▨		
Phase 4				▨	▨			▨	▨	

รูปที่ 3.6 แสดงการทำงานแบบเฟสคู่

3.3 การควบคุมการทำงานของสเต็ปิ่งมอเตอร์

สามารถทำการควบคุมได้ 2 วิธีคือ

1. แบบรูปเปิด (Open loop)

เป็นวิธีการควบคุมสเต็ปิ่งมอเตอร์ที่ใช้กันทั่วไป โดยการจ่ายพัลส์ที่มีการจัดเรียงลำดับแล้วตามจำนวนสเต็ปที่ต้องการให้มอเตอร์เคลื่อนที่ไปได้ และไม่จำเป็นที่จะต้องมีการส่งผลย้อนกลับมาจากมอเตอร์เคลื่อนที่ไปตามที่กำหนดจริงหรือไม่ เพราะสเต็ปิ่งมอเตอร์มีความแม่นยำสูงมากทำให้หมุนไปตามที่กำหนดไว้โดยไม่มีการผิดพลาด การควบคุมแบบนี้จะมีความเที่ยงตรงสูงในการนำมอเตอร์ไปขับโหลดที่มีค่าไม่มากนัก ลำดับการควบคุมสามารถเขียนเป็นแผนภาพได้ดังนี้



รูปที่ 3.7 แสดงการควบคุมแบบรูปเปิด

ภาคการทำงานต่างๆ ในการควบคุมมีดังนี้

- ภาคจัดเรียงลำดับสัญญาณ

จะทำหน้าที่ในการเรียงลำดับสัญญาณพัลส์ที่จะนำไปขับสเต็ปิ่งมอเตอร์ให้มีทิศทางตามที่ต้องการ โดยอาจสร้างมาจากวงจรเชิงตรรกะ (Logic electronics) ต่างๆ เช่น ฟลิปฟลอป (Flip-flop) , ลอจิกเกต (Logic gate) เป็นต้น แต่แบบที่นิยมคือใช้หน่วยประมวลผลประเภทไมโครโพรเซสเซอร์ (Microprocessor) , ไมโครคอนโทรลเลอร์ (Microcontroller) หรือคอมพิวเตอร์ส่วนบุคคล (Personal computer) ให้ทำการสร้างลำดับสัญญาณขึ้นมา ซึ่งประหยัดค่าซึ่งจ่ายและมีความยุ่งยากน้อยกว่าการสร้างจากวงจรเชิงตรรกะ

- ภาคขยายสัญญาณ

เนื่องจากสัญญาณที่ออกมาจากวงจรเรียงลำดับพัลส์มีค่าต่ำไม่สามารถจะนำไปขับมอเตอร์ที่มีความต้องการกระแสสูงๆ ได้ จึงต้องใช้วงจรในการขยายกระแสให้มีปริมาณที่มากพอที่จะขับมอเตอร์ได้ วงจรที่นิยมใช้ในการขับมอเตอร์มี 2 ชนิดคือ

1. วงจรขับขั้วเดียว (Unipolar drive circuit)

นิยมใช้กับสเต็ปิ่งมอเตอร์แบบแปรค่าความต้านทานแม่เหล็กได้ (VR) โดยใช้ทรานซิสเตอร์ (Transistor) เป็นตัวเปิดเปิดสัญญาณให้กระแสไหลหรือไม่ไหลไปยังลวดตัว

นำ โดยถ้าทำการจ่ายกระแสที่ขาเบส (Base) ให้มีค่าสูงพอที่จะทำให้ทรานซิสเตอร์อิ่มตัว (Saturate) จะทำให้มีกระแสไหลผ่านขาคอลเลคเตอร์ (Collector) ไปยังขาอิมิตเตอร์ (Emitter) เพื่อผ่านไปยังลวดตัวนำ ในขณะที่ทรานซิสเตอร์อยู่ในสภาวะอื่นตัวนี้แรงดันไฟฟ้าที่ตกคร่อมขดลวดจะมีค่าประมาณแรงดันไฟเลี้ยงวงจร ดังนั้นแหล่งจ่ายไฟเลี้ยงจะต้องมีค่าไม่มากเกินไปที่จะทำให้มีกระแสไฟฟ้าไหลผ่านลวดตัวนำเกินพิกัดที่ลวดตัวนำทนได้ คือ

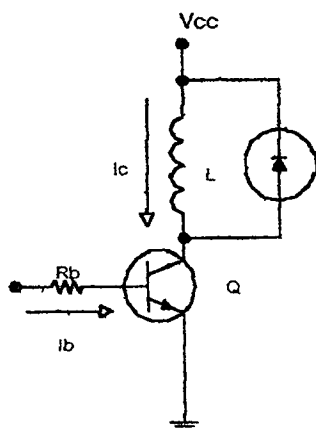
$$V_s = I \cdot (r + R)$$

เมื่อ V_s คือ ค่าแรงดันไฟฟ้าของแหล่งจ่าย (Volt)

I คือ ค่ากระแสสูงสุดที่ขดลวดทนได้ (Ampere)

R_1 คือ ความต้านทานในลวดตัวนำ (Ω)

R_2 คือ ความต้านทานช่วยจำกัดกระแส (Ω)

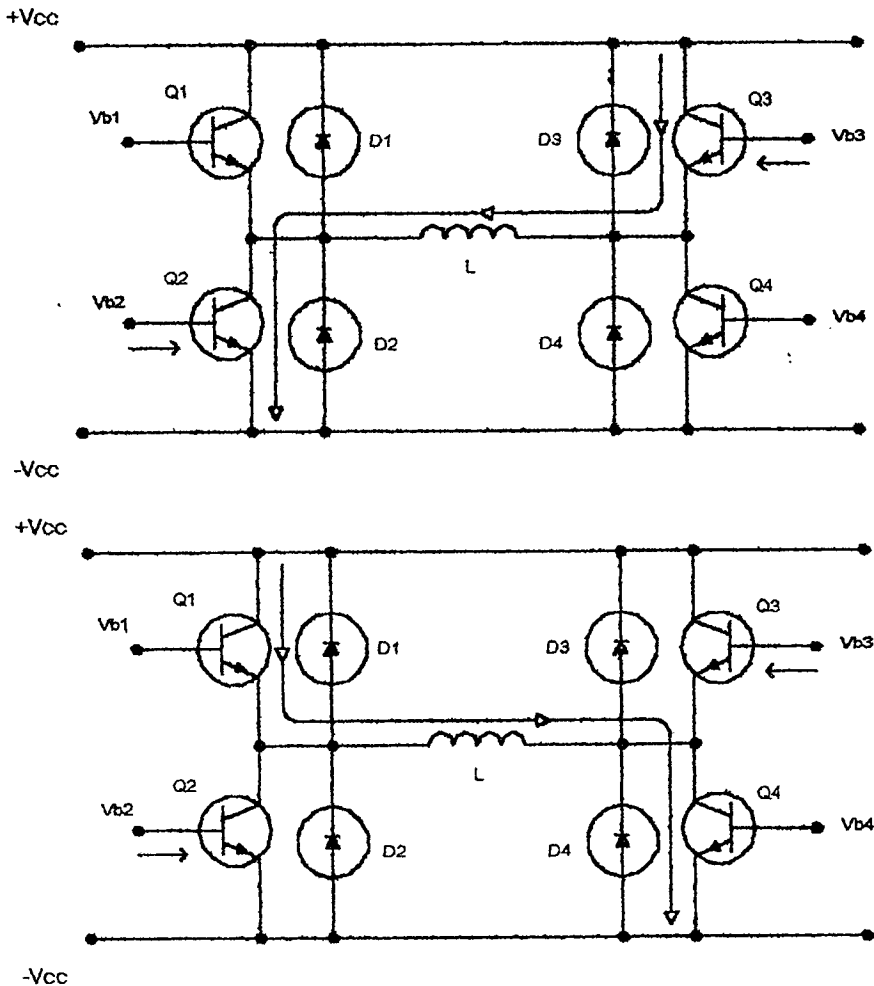


รูปที่ 3.8 วงจรขั้วกระแสแบบขั้วเดียว

เมื่อหยุดจ่ายกระแสให้กับขาเบสก็จะทำให้ทรานซิสเตอร์ไม่นำกระแสแต่กระแสที่ยังไหลผ่านขดลวดไม่สามารถหยุดไหลได้ในทันทีทันใด เนื่องจากคุณสมบัติความเหนี่ยวนำ (Inductance) ในขดลวด ทำให้เกิดการเหนี่ยวนำแรงดันไฟฟ้าตกคร่อมระหว่างขาคอลเลคเตอร์และขาอิมิตเตอร์ของทรานซิสเตอร์ เกิดกระแสไหลเป็นจำนวนมากทำให้ทรานซิสเตอร์พังได้ เราสามารถป้องกันการไหลของกระแสจำนวนมากนี้ผ่านทรานซิสเตอร์ได้โดยการนำกระแสให้ไหลไปในอีกทิศทางหนึ่ง จากการต่อวงจรที่เรียกว่า Free wheeling circuit ซึ่งอาจประกอบด้วยตัวต้านทานและไดโอด (Diode) เพื่อให้กระแสไหลผ่านได้ในทิศทางเดียว

2. วงจรขับสองขั้ว (Bipolar drive circuit)

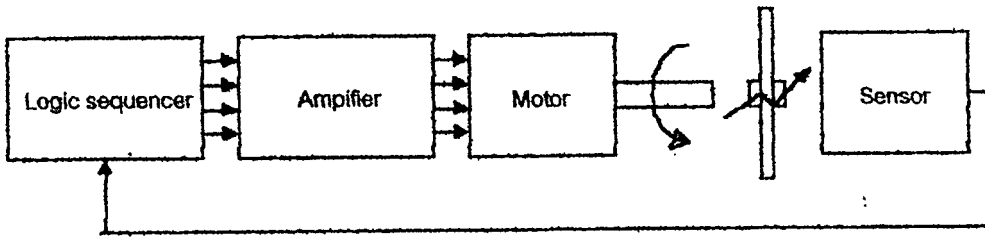
นิยมใช้กับมอเตอร์แบบแม่เหล็กถาวร (PM) และแบบไฮบริด (Hybrid) ใช้ทรานซิสเตอร์ในการควบคุม 4 ตัว โดยในขณะที่ทำงานทรานซิสเตอร์จะนำกระแสหนึ่งคู่ ในขณะที่อีกคู่หนึ่งไม่นำกระแส ทำให้ทิศทางของกระแสที่ไหลผ่านขดลวดมีทิศทางตรงข้ามกัน โดยขณะที่ทรานซิสเตอร์ T1 และ T3 นำกระแสจะทำให้มีกระแสไหลผ่านขดลวดในทิศทางจากซ้ายไปขวา ต่อมาเมื่อทรานซิสเตอร์ T2 และ T3 นำกระแสบ้างกระแสจะไหลจากขวาไปซ้าย แต่ในการขับมอเตอร์แบบนี้จะต้องใช้แหล่งจ่ายไปที่มีขั้วตรงข้ามกัน



รูปที่ 3.9 วงจรขับกระแสแบบสองขั้ว

2. แบบลูปปิด (Close loop)

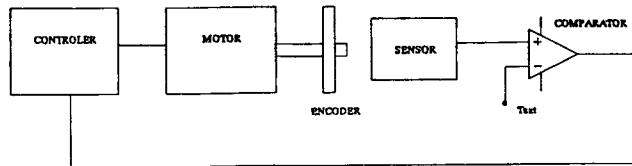
เป็นวิธีการควบคุมสแต็ปปีงมอเตอร์ที่มีการตรวจสอบว่ามอเตอร์เคลื่อนที่ไปยังตำแหน่งที่กำหนดไว้หรือไม่ โดยใช้ตัวตรวจจับ (Sensor) ทำหน้าที่ตรวจจับตำแหน่งของมอเตอร์แล้วส่งผลไปยังวงจรเปรียบเทียบเพื่อทำการเปรียบเทียบกับค่าที่กำหนดไว้ ถ้าค่าที่ส่งกลับมามีค่าน้อยกว่าที่กำหนดไว้ วงจรเปรียบเทียบก็จะทำการกระตุ้นให้วงจรเรียงลำดับสัญญาณ (Logic sequencer) ปล่อยพัลส์ออกมาขับสแต็ปปีงมอเตอร์อีกเพื่อให้มอเตอร์เคลื่อนที่ไปยังตำแหน่งที่กำหนดไว้ การควบคุมมอเตอร์แบบนี้จึงมีความแม่นยำสูงแม้ว่าจะต้องนำไปขับโหลดที่มีค่ามากก็ตาม สามารถเขียนแผนภาพการควบคุมแบบลูปปิดได้ดังนี้



รูปที่ 3.10 แสดงการควบคุมแบบลูปปิด

3.4 การหาจุดเริ่มต้นและจุดสิ้นสุด (Opto-interrupter)

ประกอบด้วยส่วนต่างๆ 3 ส่วนคือ ตัวส่งแสง (Transmitter) เช่นพวกไดโอดเปล่งแสง (LED) , ตัวรับแสง (Receiver) เช่น โฟโตทรานซิสเตอร์ (Photo transistor) , แผ่นกั้นแสง โดยแผ่นกั้นแสงที่มีส่วนสีดำ 1 ชิ้นจะถูกนำมาวางไว้ระหว่างตัวส่งแสงและตัวรับแสง ดังนั้นแสงจะสามารถผ่านจากตัวส่งแสงไปยังตัวรับแสงได้จนกระทั่งมีแถบสีนำมบังไว้ สัญญาณที่ตัวรับแสงจะมีลักษณะเป็นมีกระแสไหลหรือไม่มีกระแสไหลเหมือนการทำงานเป็นสวิทช์ ซึ่งสัญญาณนี้จะถูกนำไปควบคุมให้มอเตอร์หมุนหรือหยุดหมุน

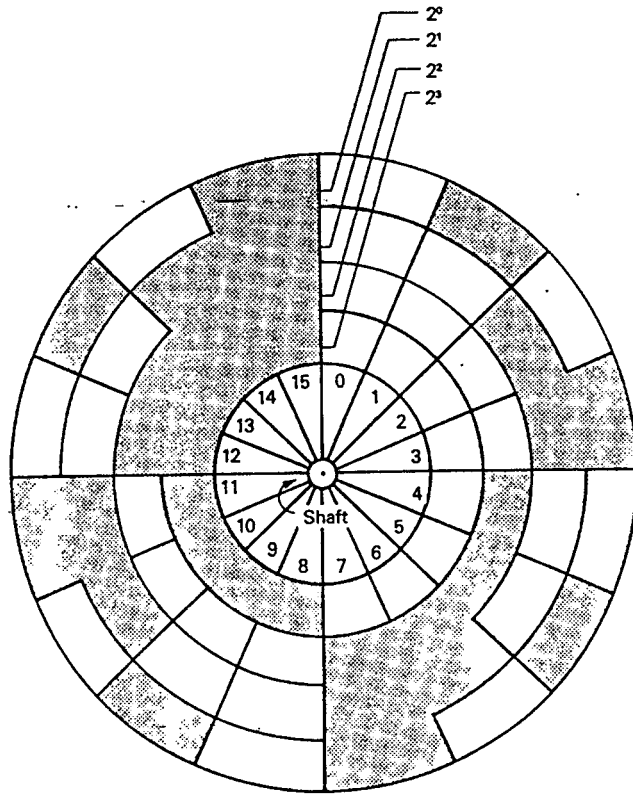


รูปที่ 3.11 แสดงการหาจุดเริ่มต้นและจุดสิ้นสุด

การระบุตำแหน่งของมอเตอร์ จากการใช้ตัวบอกรหัส (Encoder) มี 2 ชนิดคือ

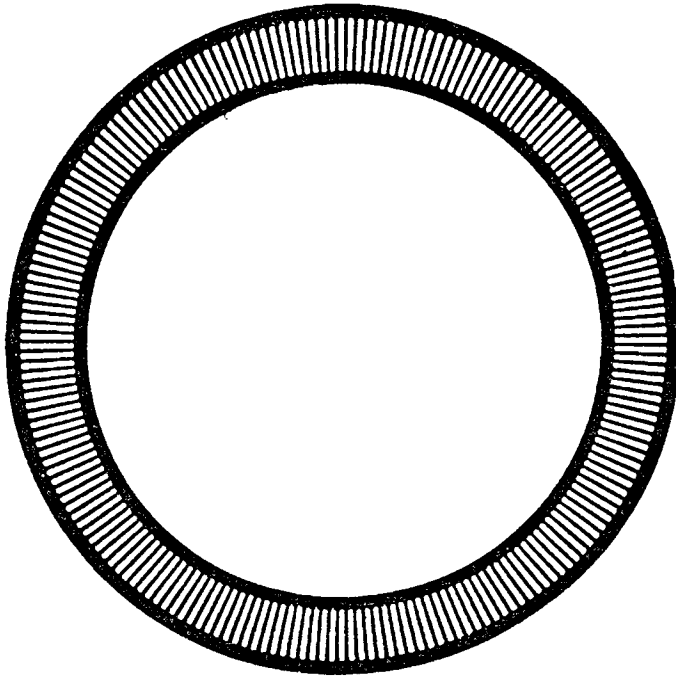
1. แบบสัมบูรณ์ (Absolute Encoder)

เป็นตัวบอกรหัสที่ให้ตำแหน่งที่ถูกต้องของมอเตอร์ทันทีที่มีการจ่ายพลังงานไฟฟ้าให้กับตัวรับและตัวส่งแสง ดังนั้นตำแหน่งของมอเตอร์ที่ได้จากตัวบอกรหัสจะมีความถูกต้องอยู่ตลอดเวลา ข้อดีของตัวบอกรหัสแบบสัมบูรณ์คือถึงแม้ว่าพลังงานไฟฟ้าที่จ่ายให้กับตัวส่งและตัวรับแสงจะหายไปแต่ก็ยังสามารถบอกตำแหน่งของมอเตอร์และส่งข้อมูลไปได้ทันทีที่มีพลังงานไฟฟ้าจ่ายให้ นั่นคือมีคุณสมบัติในการจดจำตำแหน่งได้ ลักษณะของจานบอกรหัสแบบสัมบูรณ์คือมีการแบ่งพื้นที่ของจานตามเส้นรอบวงเป็นช่องๆ เรียกว่า แทรค (Tracks) และตามแนวรัศมีเรียกว่า เซกเตอร์ (Sector) โดยค่ารหัสที่อ่านได้จากตัวรับแสงขึ้นอยู่กับการจัดเรียงของแทรคและเซกเตอร์เหล่านี้ ซึ่งมีการจัดเรียงแบบต่างๆ กันเช่น แบบรหัส BCD (Binary Code Decimal) , แบบรหัสเกรย์ (Gray Code) เป็นต้น การบอกรหัสแบบสัมบูรณ์นี้จะให้ตัวรับและส่งแสงเป็นจำนวนเท่ากับจำนวนของแทรคที่ได้แบ่งไว้ ดังนั้นหากต้องการความละเอียดในการบอกตำแหน่งมากๆ จะต้องทำการแบ่งแทรคและเซกเตอร์ให้มีจำนวนมากๆ ทำให้ตัวบอกรหัสแบบนี้มีราคาแพงและมีขนาดค่อนข้างใหญ่



รูปที่ 3.12 ตัวอย่างของตัวบอกรหัสแบบตัวหมุน

2. แบบเพิ่มค่า (Incremental Encoder) นิยมใช้มากกว่าแบบตัวหมุนเนื่องจากมีขนาดเล็กกว่าและราคาไม่แพงมาก ส่วนมากใช้ในการควบคุมตำแหน่งของข้อต่อต่างๆ ในหุ่นยนต์ ส่วนประกอบของตัวบอกรหัสแบบนี้คล้ายกับแบบตัวหมุนจะต่างกันตรงที่ลักษณะของจาน โดยจะทำเป็นช่องเล็กๆ ตามแนวรัศมีของจานเป็นจำนวนมาก โดยถ้ายังมีจำนวนมากก็จะมี ความละเอียดมาก ข้อดีของตัวบอกรหัสแบบเพิ่มค่าคือสามารถทำให้มีความละเอียดมากๆ ได้ง่าย แต่มีข้อเสียคือถ้าพลังงานที่จ่ายให้กับตัวรับและตัวส่งแสงหยุดทำงานจะต้องมีการปรับแต่งค่าเริ่มต้นใหม่เนื่องจากไม่สามารถจดจำตำแหน่งได้



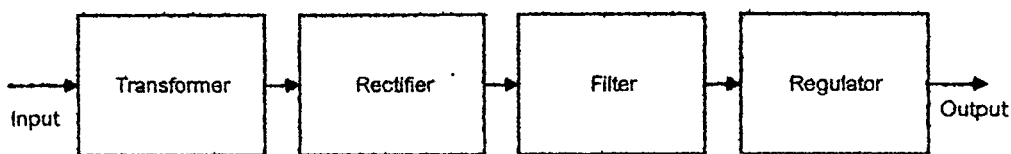
รูปที่ 3.13 ตัวอย่างของตัวบอกรหัสแบบเพิ่มค่า

บทที่ 4

แหล่งจ่ายไฟตรงและหลักการเชื่อมต่อกอนพิวเตอร์

4.1 แหล่งจ่ายไฟตรง

เป็นคำเปลี่ยนรูปพลังงานจากแรงดันไฟฟ้ากระแสสลับ (Alternate Current : AC) เป็นแรงดันไฟฟ้ากระแสตรง (Direct Current : DC) เพื่อจ่ายให้กับวงจรทางด้านอิเล็กทรอนิกส์



รูปที่ 4.1 ส่วนประกอบพื้นฐานของแหล่งจ่ายไฟตรง

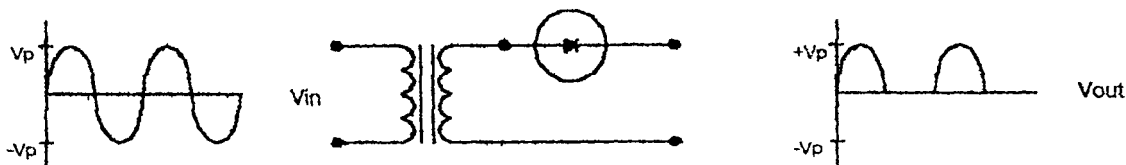
ส่วนประกอบของแหล่งจ่ายไฟตรง

- หม้อแปลงไฟฟ้า (Transformer) ทำหน้าที่ลดระดับแรงดันไฟฟ้าจาก 220 โวลต์ มาเป็นค่าที่ต้องการใช้
- วงจรกรองกระแส (Rectifier) ทำหน้าที่จัดเรียงกระแสไฟฟ้าให้ไหลไปในทิศทางเดียวกัน ส่วนใหญ่ใช้อุปกรณ์สารกึ่งตัวนำพวกไดโอดเรียงกระแส (Rectifier Diode) ลักษณะการจัดเรียงมี 2 ชนิดคือ
 - แบบเรียงกระแสครึ่งคลื่น (Half wave rectifier) ประกอบด้วยไดโอด 1 ตัว ต่ออนุกรมกับแหล่งจ่าย สัญญาณที่ได้จะมีลักษณะครึ่งคาบ ดังรูป แรงดันเฉลี่ยที่ได้มีค่าเท่ากับ

$$V_{dc} = V_p / \pi$$

เมื่อ V_{dc} คือ ค่าแรงดันเฉลี่ย (Volt)

V_p คือ ค่าแรงดันสูงสุด (Volt)



รูปที่ 4.2 วงจรเรียงกระแสแบบครึ่งคลื่น

- แบบเรียงกระแสเต็มคลื่น (Full wave rectifier) มีด้วยกัน 2 แบบคือ

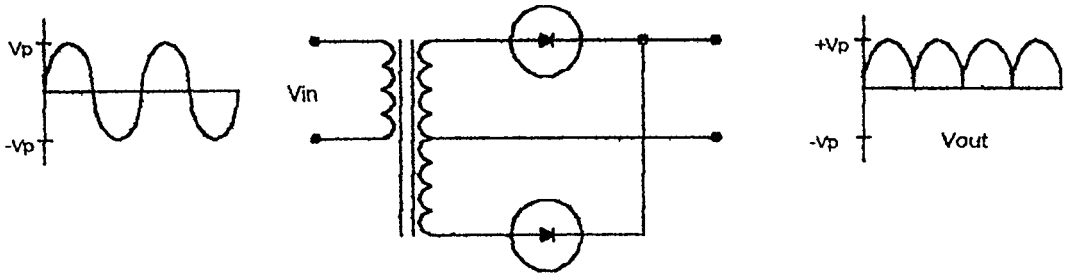
1. แบบใช้กับหม้อแปลงมีแท่งกลาง (Full wave center tap rectifier)

ใช้ไดโอดเรียงกระแส 2 ตัว สัญญาณที่ได้จะมีลักษณะเต็มคาบ โดยค่าแรงดันเฉลี่ย จะมีค่ามากกว่าแบบครึ่งคลื่น คือมีค่าเท่ากับ

$$V_{dc} = 2V_p/\pi$$

เมื่อ V_{dc} คือ ค่าแรงดันเฉลี่ย (Volt)

V_p คือ ค่าแรงดันสูงสุด (Volt)



รูปที่ 4.3 วงจรเรียงกระแสแบบเต็มคลื่น โดยใช้หม้อแปลงมีแท่งกลาง

2. แบบบริจันน์ (Bridge rectifier)

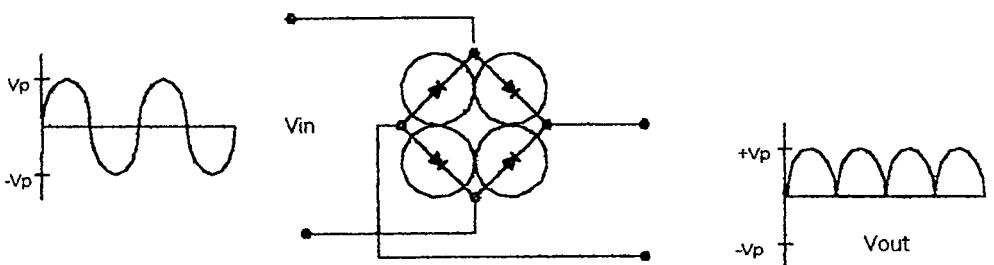
ใช้ไดโอด 4 ตัวต่อกันแบบบริจันน์สัญญาณที่ได้จะมีลักษณะเต็มคาบ โดยค่าแรงดันเฉลี่ยจะมีค่ามากกว่าแบบครึ่งคลื่น คือมีค่าเท่ากับ

$$V_{dc} = 2V_p/\pi$$

เมื่อ V_{dc} คือ ค่าแรงดันเฉลี่ย (Volt)

V_p คือ ค่าแรงดันสูงสุด (Volt)

แต่มีข้อดีกว่าแบบใช้หม้อแปลงมีแท่งกลางคือค่าแรงดันย้อนกลับตกคร่อมไดโอดได้มีค่าน้อยกว่า



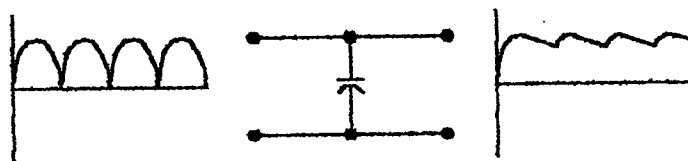
รูปที่ 4.4 วงจรเรียงกระแสแบบเต็มคลื่นแบบบริจันน์

3. วงจรกรองกระแส (Filter)

ทำหน้าที่ลดแรงดันกระเพื่อมที่ออกมาจากวงจรเรียงกระแสทำให้แรงดันเรียบขึ้น ที่นิยมใช้มี

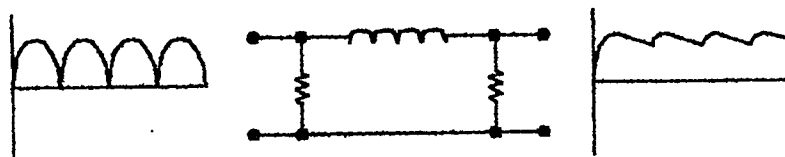
2 ชนิดคือ

- วงจรกรองกระแสโดยใช้ตัวเก็บประจุ (Capacitor filter)



รูปที่ 4.5 วงจรกรองกระแสโดยใช้ตัวเก็บประจุ

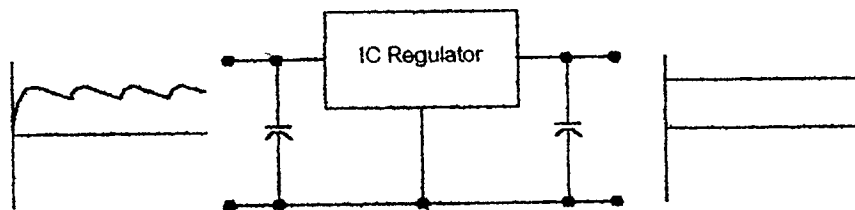
- วงจรกรองกระแสโดยใช้ตัวเหนี่ยวนำ (Inductor filter)



รูปที่ 4.6 วงจรกรองกระแสโดยใช้ตัวเหนี่ยวนำ

4. วงจรรักษาระดับแรงดัน (Regulator)

ทำหน้าที่รักษาระดับแรงดันให้คงที่ที่กระแสโหลดต่างๆกันวงจรที่ใช้ก่อสร้างมาจากวงจรทรานซิสเตอร์แบบต่างๆ หรือใช้ทรานซิสเตอร์ร่วมกับออปแอมป์ (Operational Amplifier : Opamp) แต่ในปัจจุบันมีวงจรสำเร็จรูปในรูปวงจรรวม (IC) ที่ใช้รักษาระดับแรงดันโดยเฉพาะ ซึ่งมีขนาดเล็กและคุณภาพสูง โดยมีทั้งแบบค่าคงที่และปรับค่าได้ เช่น ตระกูล LM78xx , LM79xx , Lm317x เป็นต้น



รูปที่ 4.7 วงจรรักษาระดับแรงดัน โดยใช้ไอซี

5. วงจรช่วยจ่ายกระแส (Current amplifier)

ทำหน้าที่ในการช่วยจ่ายกระแสไปที่โหลดในกรณีที่โหลดต้องการกระแสมากๆ ทั้งนี้เนื่องจาก เอาท์พุทของวงจรรักษาระดับแรงดันโดยทั่วไปมีค่ากระแสที่น้อย ไม่สามารถจ่ายกระแสได้เพียงพอเมื่อนำไปใช้จ่ายกระแสสูงๆได้ โดยอุปกรณ์ที่จะมาช่วยในการจ่ายกระแสอาจเป็นพวกทรานซิสเตอร์แบบต่างๆ ซึ่งอาศัยคุณสมบัติในการขยายกระแส

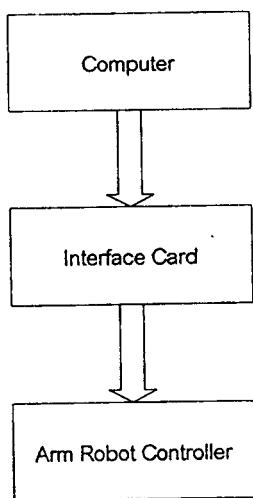
4.2 ระบบเชื่อมต่อกับคอมพิวเตอร์

การควบคุมการเคลื่อนที่ของแขนกลผ่านคอมพิวเตอร์ต้องอาศัยวงจรเชื่อมต่อ(Interface) ระหว่างคอมพิวเตอร์กับแขนกลที่สร้างขึ้น โดยอาศัยคอมพิวเตอร์เป็นตัวแทนควบคุมการเคลื่อนที่ให้เป็นไปตามต้องการ รายละเอียดและขั้นตอนการออกแบบแสดงเป็นหัวข้อหลักๆ ได้ดังนี้

1. หลักการอินเตอร์เฟส
2. ช่องเสียบสัญญาณของคอมพิวเตอร์(slot)
3. การทำงานของ IC เบอร์ 8255
4. การออกแบบการ์ดอินเตอร์เฟส
5. โปรแกรมควบคุมการทำงานของแขนกล

4.2.1 หลักการอินเตอร์เฟส

ปัจจุบันการใช้งานคอมพิวเตอร์เป็นไปอย่างกว้างขวาง เช่น งานด้านเอกสาร งานทางด้านธุรกิจ การแพทย์ รวมถึงทางด้านอุตสาหกรรมที่ใช้คอมพิวเตอร์ควบคุมการทำงานของเครื่องจักร ดังนั้นการนำคอมพิวเตอร์ไปใช้งานในด้านต่างๆนั้นจึงต้องมีหน้าที่เชื่อมต่อกันระหว่างงานนั้นกับคอมพิวเตอร์ โดยรูปแบบการติดต่อที่เป็นมาตรฐานที่นิยมใช้ คือ มาตรฐานการสื่อสารแบบอนุกรม(RS-232) และมาตรฐานการสื่อสารแบบขนาน(GPIB) แต่สำหรับในโปรเจกต์นี้เป็นการใช้งานเฉพาะอย่าง จึงได้ทำการออกแบบวงจรอินเตอร์เฟสขึ้นมาเพื่อเชื่อมต่อระหว่างคอมพิวเตอร์และแขนกล โดยให้สามารถควบคุมการเคลื่อนที่ของแขนกลได้ แสดงดังรูปที่ 4.8



รูปที่ 4.8 แสดงการเชื่อมต่อระหว่างคอมพิวเตอร์และแขนกล

การเชื่อมต่อของสเต็ปป์มอเตอร์ที่ใช้ขับแขนกล (เสมือนเป็นอุปกรณ์อินพุท/เอาต์พุท) เข้ากับระบบบัสของคอมพิวเตอร์โดยตรง นับว่าเป็นการเพิ่มประสิทธิภาพในการทำงานของอุปกรณ์อินพุท/เอาต์พุทที่ต่ออยู่กับระบบให้ดียิ่งขึ้น

การควบคุมอุปกรณ์อินพุท/เอาต์พุท จะกระทำผ่านพอร์ต โดยการอ้างถึงตำแหน่งของพอร์ตที่อุปกรณ์นั้นต่ออยู่โดยตรง ดังนั้นการใช้งานหรือการควบคุมอุปกรณ์อินพุท/เอาต์พุทจึงจำเป็นต้องศึกษาถึงวิธีการอ้างตำแหน่งของพอร์ตให้ถูกต้อง

สำหรับ โปรเจกต์ในเทอมนี้ ได้ทำการอินเทอร์เฟซแขนกลเข้ากับคอมพิวเตอร์ ซึ่งเป็นการนำเอาสัญญาณแอนะล็อกมาทำการดีโค้ดตำแหน่งของพอร์ตอินพุทและพอร์ตเอาต์พุท ซึ่งเป็นทางผ่านของสัญญาณข้อมูลที่เชื่อมระหว่างคอมพิวเตอร์และสเต็ปป์มอเตอร์ โดยคอมพิวเตอร์จะสั่งให้สเต็ปป์มอเตอร์หมุน ไปตาม โปรแกรมภาษาซีที่ตั้งไว้ ส่งผลให้แขนกลหมุนไปในทิศทางที่ต้องการ ข้อมูลที่ส่งออกมาจะผ่านมาทางพอร์ตเอาต์พุทของไอซี 8255 รวมทั้งสามารถรับข้อมูลจากการทดสอบการหมุนเข้ามาทางพอร์ตอินพุทของไอซี 8255 ได้เช่นกัน จะเห็นได้ว่าไอซี 8255 เป็นได้ทั้งพอร์ตอินพุทและพอร์ตเอาต์พุท(รายละเอียดจะเสนอในหัวข้อต่อไป) โดยวงจรทั้งในส่วนของการ์ดดีโค้ดตำแหน่งของพอร์ตและการกำหนดพอร์ตอินพุท/เอาต์พุทรวมกันเป็นวงจรอินเทอร์เฟซ

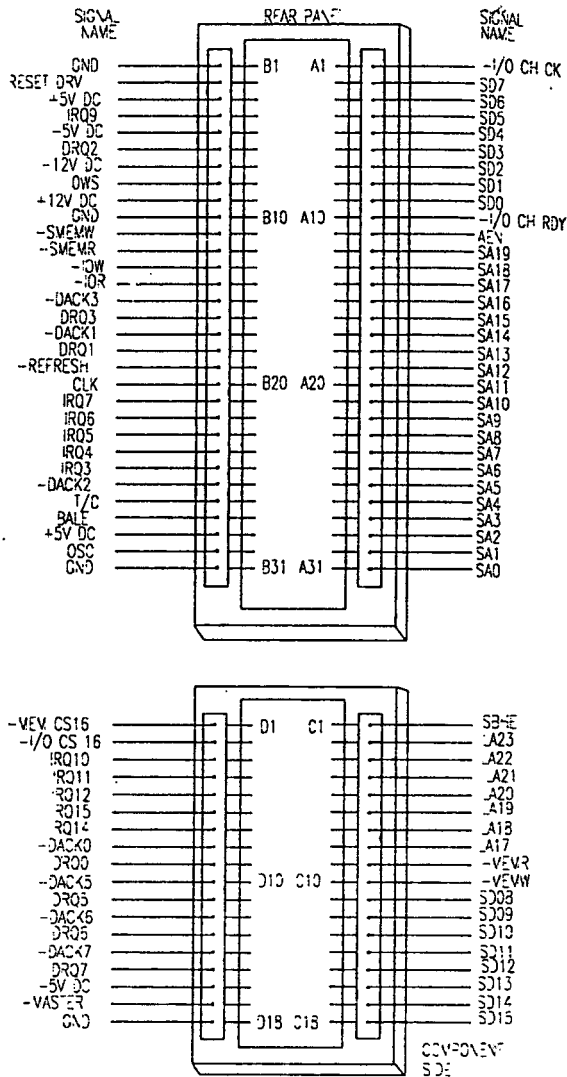
สัญญาณต่างๆที่ใช้ในการอินเทอร์เฟซนี้ นำมาจากสล็อตที่อยู่บนเมนบอร์ดของคอมพิวเตอร์ ซึ่งมีไว้เพื่ออำนวยความสะดวกในกรณีที่มีผู้ใช้ต้องการนำคอมพิวเตอร์ไปใช้งานเฉพาะอย่าง

4.2.2 ช่องเทียบสัญญาณของคอมพิวเตอร์ (Slot)

ภายในคอมพิวเตอร์ที่ใช้ซีพียู เบอร์ 8088 อาจเรียกได้ว่าเป็นรุ่น XT ซึ่งเป็นแบบ 16 บิตเทียม คือมีการประมวลผลภายในเป็นแบบ 16 บิต แต่มีบิตข้อมูลเพียง 8 บิตเท่านั้น และคอมพิวเตอร์ที่ใช้ซีพียูตั้งแต่เบอร์ 80286 ซึ่งเป็นขนาด 16 บิตแท้ คือมีการประมวลผลภายในเป็นแบบ 16 บิต และมีบิตข้อมูล 16 บิต จะเรียกว่าเป็นรุ่น AT

บนเมนบอร์ดของรุ่น XT จะมีขาสัญญาณในแต่ละสล็อตทั้งสิ้น 62 ขา ซึ่งแบ่งเป็น 2 ช่วงๆ ละ 31 ขา ขาสัญญาณที่อยู่ทางด้านขวาของสล็อตเรียก A1-A31 และขาสัญญาณที่อยู่ทางด้านซ้ายของสล็อตเรียก B1-B31 (นับจากด้านหลังของเครื่องคอมพิวเตอร์) แต่ละขาของสล็อตเชื่อมต่อกับเส้นสัญญาณต่างๆบนเมนบอร์ด ทำให้การอินเตอร์เฟสอุปกรณ์อินพุท/เอาต์พุทเข้ากับคอมพิวเตอร์ทำได้สะดวกขึ้น สำหรับรุ่น AT จะมีสล็อตอยู่ 2 ชนิด คือ สล็อตสั้นและสล็อตยาว ในส่วนของสล็อตยาวจะมีจำนวนขาและตำแหน่งของสัญญาณเหมือนกับ XT แต่ส่วนที่เพิ่มมาคือสล็อตสั้น ซึ่งจะมีสัญญาณบิตข้อมูลเพิ่มขึ้นมาอีก 8 บิต (8 บิต บน), สัญญาณบัสแอดเดรสเพิ่มมาอีก 4 บิต และขาสัญญาณควบคุมที่เพิ่มมาขึ้นจากรุ่น XT ซึ่งสัญญาณที่ขาสล็อตนี้จะเป็นสัญญาณอินพุท, สัญญาณเอาต์พุท หรือทั้งสัญญาณอินพุทและเอาต์พุท โดยแบ่งกลุ่มของสัญญาณได้ดังนี้

- เทาเวอร์ซัพพลาย
- สัญญาณบัสแอดเดรส
- สัญญาณบัสข้อมูล
- สัญญาณอินเตอร์รัพต์
- สัญญาณที่ใช้ในกระบวนการ DMA
- สัญญาณควบคุมต่างๆ
- สัญญาณที่ใช้สร้าง wait state
- สัญญาณนาฬิกา



รูปที่ 4.9 แสดงตำแหน่งของขาสัญญาณต่างๆบนสต็อคคอมพิวเตอร์

ก่อนที่จะทำการอินเทอร์เฟซอุปกรณ์อินพุท/เอาต์พุตนั้น จะต้องทราบหน้าที่และตำแหน่งของขาบนสต็อคคอมพิวเตอร์ที่จำเป็น ซึ่งในที่นี้จะอธิบายเฉพาะสัญญาณที่ใช้ในการสร้างวงจรอินเทอร์เฟซสำหรับใช้ควบคุมการหมุนของแขนกล

- รายละเอียดของขาสัญญาณที่ใช้

1. ไฟเลี้ยง +5V :

ขาสัญญาณที่ต่ออยู่กับแหล่งจ่ายไฟตรง เรกูเลเตอร์ +5 โวลต์

2. กราวด์ :

ขาสัญญาที่ต่ออยู่กับกราวด์ของระบบเรกูเลเตอร์

3. A0-A19 (address bus : ขาA31-A12) :

ขาสัญญาทั้ง 20 เส้นนี้เป็นขาเอาต์พุต ใช้กำหนดตำแหน่งของหน่วยความจำหรือตำแหน่งพอร์คของอุปกรณ์อินพุต/เอาต์พุตที่ต้องการติดต่อด้วย โดย A0 มีนัยสำคัญต่ำสุด และ A19 มีนัยสำคัญสูงสุด สำหรับสัญญาณแอดเดรสบัส A0-A19 นี้สามารถอ้างตำแหน่งหน่วยความจำได้สูงสุด 1 เมกะ ไบต์ แต่ในการอ้างตำแหน่งพอร์คของอุปกรณ์อินพุต/เอาต์พุต ออกแบบให้ใช้สายสัญญาณแอดเดรสบัสเพียง 10 เส้น คือ A0-A9 ดังนั้นจึงสามารถอ้างตำแหน่งพอร์คได้ถึง 1024 พอร์ค ซึ่งแบ่งออกเป็น 2 กลุ่ม คือ กลุ่มพอร์คที่มีแอดเดรสอยู่ในช่วง 000H-0FFH จะใช้งานบนเมนบอร์ดสำหรับชิพพอร์คเท่านั้น เช่น 8259(Interrupt Controller),8237(DMA Controller),8253(Timer&Counter) และกลุ่มที่มีแอดเดรสอยู่ในช่วง 100H-3FFH จะใช้งานกับการ์ดขยายต่างๆ ที่เสียบในสล็อต สำหรับโปรโตไทป์การ์ด จะอยู่ที่ตำแหน่ง 300H-31FH

หมายเลขพอร์ค	การใช้งาน
000H-01FH	ตัวควบคุม DMA 1 ,8237A-5
020H - 03FH	ตัวควบคุมอินเตอร์รัพท์ 1 ,8259(มาสเตอร์)
040H - 05FH	ตัวควบคุมไทมเมอร์เคาท์เตอร์ 8254-2
060H - 06FH	ตัวควบคุมพอร์คขนานและคีย์บอร์ด 8255
070H - 07FH	Real Time Clock, NMI ของระบบ
080H - 09FH	DMA page register 74LS162
0A0H - 0BFH	ตัวควบคุมอินเตอร์รัพท์ 2 ,8259(สเลฟ)
0C0H - 0DFH	ตัวควบคุม DMA 2,8237A-5
0F0H	เคสียร์เมทโค โปรเซสเซอร์
0F1H	รีเซ็ทเมทโค โปรเซสเซอร์
0F8H - 0FFH	เมทโค โปรเซสเซอร์
1F0H - 1F8H	ฮาร์ดดิสก์
200H - 207H	เกมอินพุต/เอาต์พุต

หมายเลขพอร์ต	การใช้งาน
278H - 27FH	เครื่องพิมพ์ขนานพอร์ต 2
2F8H - 2FFH	เครื่องพิมพ์อนุกรมพอร์ต 2
300H - 31FH	การ์ด โปรโตไทป์(prototype)
360H - 36FH	สแกนไว้
378H - 37FH	เครื่องพิมพ์ขนานพอร์ต 1
380H - 38FH	SDLC, ไบต์ซิงโครไนซ์ 2
3A0H - 3AFH	ไบต์ซิงโครไนซ์ 1
3B0H - 3BFH	อะแดปเตอร์โมโนโครมและเครื่องพิมพ์
3C0H - 3CFH	สแกนไว้
3D0H - 3DFH	อะแดปเตอร์สีและกราฟฟิก
3F0H - 3F7H	ตัวควบคุมคิสก์โครฟี
3F8H - 3FFH	พอร์ตอนุกรม 1

รูปที่ 4.10 แสดงตารางการใช้งานพอร์ตที่ตำแหน่งต่างๆ

จากตารางพบว่าตำแหน่งของพอร์ตถูกแบ่งออกเป็นช่วงย่อยๆซึ่งกำหนดไว้ใช้กับ อุปกรณ์ อินพุท/เอาต์พุทเฉพาะอย่าง ถ้าในระบบของเราไม่มีการใช้งานอุปกรณ์นั้น เราสามารถนำแอดเดรส ของพอร์ตในช่วงนั้นมาใช้แทนได้ แต่การใช้งานในพอร์ตที่ไม่ได้ถูกกำหนดให้ใช้กับอุปกรณ์อื่นจะใช้ ได้ดีกว่า และในทางที่ดีควรใช้ดิฟสวิทช์ทำเป็นตัวเลือกการดีได้ตำแหน่งของอุปกรณ์อินพุท/เอาต์ พุท

4.D0-D7 (Data bus:ขา A9-A2):

เป็นขาสัญญาณแบบสองทิศทาง(Bidirectional) ซึ่งเชื่อมต่อกับบัสข้อมูลของระบบ ทำหน้าที่ ส่งผ่านข้อมูลระหว่างพอร์ตของอุปกรณ์อินพุท/เอาต์พุทกับคอมพิวเตอร์ โดยบิต D0 มีนัยสำคัญต่ำสุด บิต D7 มีนัยสำคัญสูงสุด สำหรับบัสไซเคิลของการเขียนข้อมูล ข้อมูลจะถูกส่งมาบนบัสข้อมูล ก่อนที่ สัญญาณ IOW จะเปลี่ยนจากลอจิก "0" เป็นลอจิก "1"(ขอบขาขึ้น) ซึ่งขอบขาขึ้นของสัญญาณ IOW นี้ จะสั่งให้พอร์ตเอาต์พุทที่มีแอดเดรสตรงกับค่าแอดเดรสบนบัสแอดเดรสรับข้อมูลนั้นไปเก็บไว้

สำหรับในบัสไซเคิลของการอ่านข้อมูล พอร์ตอินพุทจะส่งข้อมูลออกมาบนบัสข้อมูลก่อนที่ สัญญาณ IOR จะเปลี่ยนจาก ลอจิก "0" เป็น ลอจิก "1"(ขอบขาขึ้น)

5.IOW(I/O Write:ขา B13) :

เป็นสัญญาณเอาต์พุต แอคทีฟที่ลอจิก "0" โดยเป็นขาสัญญาณที่บอกให้อุปกรณ์อินพุต/เอาต์พุตที่ต่ออยู่ ทำการเก็บข้อมูลจากควาต์ตัมเข้าไป

6.IOR(I/O Read:ขา B14) :

เป็นสัญญาณเอาต์พุต แอคทีฟที่ลอจิก "0" โดยเป็นขาสัญญาณที่บอกให้อุปกรณ์อินพุต/เอาต์พุตที่ต่ออยู่ ทำการส่งข้อมูลลงมาที่ควาต์ตัม

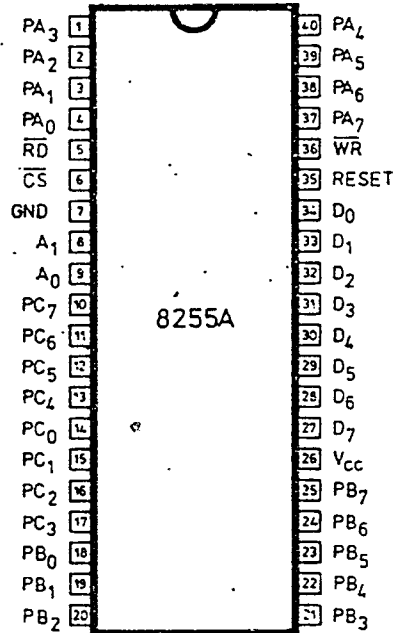
7.AEN(Address Enable:ขา A11) :

เป็นสัญญาณเอาต์พุตที่แสดงให้เห็นว่าบัสไซเคิลที่เกิดขึ้นในช่วงเวลาที่สัญญาณ AEN แอคทีฟ (ลอจิก "1") นั้นเป็นบัสไซเคิลของกระบวนการ DMA แต่บนเมนบอร์ดของคอมพิวเตอร์ใช้สัญญาณนี้ในการคัสเทมเปลพอร์ตอินพุต/เอาต์พุตต่างๆ ซึ่งไม่เกี่ยวข้องกับกระบวนการ DMA เลย

8.RESET(Reset Driver:ขา B2) : เป็นสัญญาณเอาต์พุต แอคทีฟที่ลอจิก "1" สัญญาณนี้ถูกนำไปใช้ในการรีเซ็ตวงจรอินเตอร์เฟสหรืออุปกรณ์อินพุต/เอาต์พุตต่างๆ ในช่วงที่เริ่มจ่ายไฟให้กับระบบ ซึ่งเป็นการทำให้วงจรหรืออุปกรณ์เหล่านั้นถูกปรับให้อยู่ในสถานะที่แน่นอนก่อนที่จะเริ่มต้นการทำงานในระบบ

4.2.3 การทำงานของ IC เบอร์ 8255

8255 เป็นไอซี 40 ขา มีพอร์ตรับส่งข้อมูลทั้งสิ้น 3 พอร์ต คือ พอร์ต A,B และ C และมีพอร์ตที่ทำหน้าที่ควบคุมการทำงานของพอร์ต A,B และ C อีก 1 พอร์ตเรียกว่าพอร์ตควบคุม(Control port) ซึ่งมีลักษณะการจ็เรียงของขาสัญญาณดังนี้



รูปที่ 4.11 ตำแหน่งขาต่างๆของ IC 8255

- หน้าที่ของขา IC 8255

D0-D7 : ใช้สำหรับติดต่อรับส่งข้อมูลกับซีพียู โดยขา D0-D7 นี้ต่อเข้ากับค่าคำสั่งของซีพียู เพื่อให้ซีพียูส่งข้อมูลออกไปยังพอร์คหรือรับข้อมูลจากพอร์คส่งให้แก่ซีพียู

CS(Chip Select) : เป็นขาอินพุทใช้สำหรับรับสัญญาณจากภายนอกเพื่อเลือกว่าจะให้ 8255 ดำเนินงาน ถ้าขานี้ได้รับลอจิก "0" จะทำให้ 8255 ถูกเชื่อมต่อเข้ากับระบบบัสต่างๆของซีพียู ซึ่งพร้อมที่จะรับส่งข้อมูล แต่ถ้าขานี้ได้รับลอจิก "1" ไอซี 8255 จะปลดตัวเองออกจากระบบบัสของซีพียู (จะอยู่ในสถานะ high impedance)

RD(Read Enable) : เป็นขาอินพุทใช้รับสัญญาณจาก ซีพียู ถ้าขานี้ได้รับลอจิก "0" และขา CS เป็นลอจิก "0" ด้วย ไอซี 8255 จะทำการส่งข้อมูลจากพอร์คอินพุทให้แก่ซีพียูทางค่าคำสั่ง

WR(Write Enable) : ขานี้ทำงานตรงข้ามกับขา RD ถ้าขา WR ได้รับลอจิก "0" ด้วย ไอซี 8255 จะรับข้อมูลจากค่าคำสั่งของซีพียู ส่งไปยังพอร์คเอาต์พุทที่ซีพียูต้องการติดต่อ

RESET : ขานี้ทำหน้าที่รีเซ็ต ไอซี 8255 โดยเมื่อ ไอซี 8255 ได้รับสัญญาณนี้ จะทำการเคลียร์สถานะต่างๆของไอซี 8255

A0-A1 : เป็นขาแอดเดรสที่ใช้ในการเลือกพอร์คที่ซีพียูต้องการติดต่อด้วย โดยสามารถเลือกได้ 4 พอร์ค คือ

บิต D3 : เป็นบิตที่กำหนดการทำงานของพอร์ต C บน ถ้าเป็นลอจิก "0" พอร์ต C บนจะเป็นพอร์ตเอาต์พุต ถ้าเป็นลอจิก "1" พอร์ต C บนจะเป็นพอร์ตอินพุต

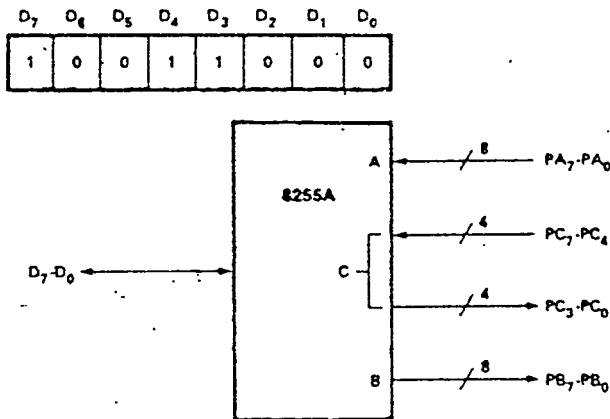
บิต D2 : เป็นบิตที่ใช้สำหรับกำหนดโหมดการทำงานของพอร์ต B ถ้าเป็นลอจิก "0" หมายถึงพอร์ต B ทำงานในโหมด 0 ถ้าเป็นลอจิก "1" จะทำงานในโหมด 1

บิต D1 : เป็นการกำหนดให้พอร์ต B เป็นพอร์ตอินพุตหรือเอาต์พุต ถ้าเป็นลอจิก "0" พอร์ต B จะเป็นพอร์ตเอาต์พุต ถ้าเป็นลอจิก "1" จะเป็นพอร์ตอินพุต

บิต D0 : เป็นบิตที่กำหนดการทำงานของพอร์ต C ล่าง ถ้าเป็นลอจิก "0" พอร์ต C ล่างจะเป็นพอร์ตเอาต์พุต ถ้าเป็นลอจิก "1" จะเป็นพอร์ตอินพุต

- รายละเอียดการทำงานในโหมด 0 ของ IC 8255

การทำงานในโหมด 0 สามารถเลือกให้พอร์ต A,B และ C เป็นพอร์ตอินพุตหรือพอร์ตเอาต์พุตได้ ขึ้นอยู่กับการกำหนดรหัสควบคุม ในที่นี้ขอแสดงเพียงรหัสควบคุมที่ใช้ในการทำโปรเจก



รูปที่ 6 แสดงรหัสควบคุม 98H

ในโปรเจกนี้ใช้งานเฉพาะ โหมด 0 จึงขออธิบายแค่เพียง โหมด 0 อย่างเดียว

บทที่ 5

การออกแบบแขนกล

การออกแบบโครงสร้างทางกล (Mechanical structure) แบ่งออกเป็น 2 ส่วนซึ่งประกอบด้วย

5.1 แหล่งส่งกำลัง (Power Source).

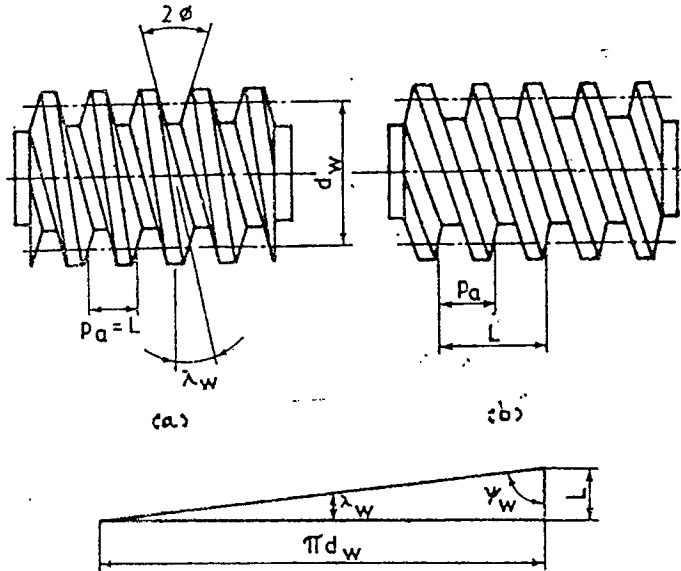
เป็นหน่วยจ่ายกำลังให้กับโครงสร้างทางกลเป็นตัวผลักดันให้โครงสร้างทางกลเคลื่อนที่ แหล่งกำลังอาจจะเป็นมอเตอร์ไฟฟ้า, ไฮดรอลิก และอื่น ๆ โดยแขนกลที่ประกอบขึ้นในการทดลองนี้เป็นแขนกลแบบอาร์คิอูเดค มีคัมกำลังจากสเตปป์มอเตอร์ ซึ่งใช้แบบวาริเอเบิลรีดักแคนซ์ เหมาะสมในการแรงที่จุดต่าง ๆ และคามจุคหนุของแกนจะ ใช้เบริงจะช่วยทำให้สค โหลคที่เกิคขึ้นกับมอเคอร์

5.2 ระบบส่งกำลัง (Power Transmission System)

เป็นตัวส่งกำลังจากแหล่งกำลังไปยัง จุดที่โครงสร้างทางกลเคลื่อนที่ประกอบด้วยชุดส่งกำลัง ดังนี้

5.2.1 ชุดเฟืองหนอน (Worm Gear)

ในโครงสร้างของแขนกลในแต่ละส่วนนั้น ได้ออกแบบโดยใช้สเตปป์มอเตอร์เป็นแหล่งส่งกำลัง เนื่องจากสามารถควบคุมความเร็วและตำแหน่งของมอเตอร์ได้ค่อนข้างแม่นยำ แต่มีจุดด้อยตรงที่สเตปป์มอเตอร์จะทำให้ทอร์คออกมาที่มีค่าค่อนข้างต่ำมาก ทำให้มีค่าไม่เพียงพอที่จะนำไปใช้ขับเคลื่อนโครงสร้างให้เคลื่อนที่ได้ ซึ่งจากคุณสมบัติของชุดเฟืองหนอน ซึ่งให้อัตราทดสูง สามารถที่จะส่งกำลังได้มาก จึงออกแบบให้มีการส่งกำลังจากสเตปป์มอเตอร์ผ่านชุดเฟืองหนอน ลักษณะทั่วไปของชุดเฟืองหนอน เกลียวตัวหนอน มีลักษณะคล้ายคลึงกับสกรูส่งกำลังประกอบด้วยเกลียว (หรือฟัน) ซึ่งมีจำนวนปาก ตั้งแต่หนึ่งถึงหกปาก หรืออาจมากกว่านี้ก็ได้ ลักษณะทั่วไปของเกลียวตัวหนอนแบบหนึ่งปาก และสองปาก ดูได้จากรูป 5.1 (a) และ 5.1 (b)



รูปที่ 5.1 ลักษณะของเกี๋ยวดัวหนอน

- พิตช์ หรือ แอ็กเซียลพิตช์ (axial pitch) P_a เป็นระยะระหว่างเกี๋ยว
 - ทลีด (Lead) L คือระยะที่แกนเกี๋ยวดัวหนอนสามารถเคลื่อนที่ไปได้ต่อจากรูหมุน 1 รอบ
- มุมทลีด (lead angle) λ_w

โดย $\tan \lambda_w = L / \pi(dw)$, dw คือ เส้นผ่าศูนย์กลางกลางของเกี๋ยวดัวหนอน
จะ ให้ความสัมพันธ์และอัตราทดดังนี้

$$T = \omega_w / \omega_g = ds / (dw) \tan \lambda_w = (ds) \tan \lambda_s / dw$$

λ_w = เส้นผ่าศูนย์กลางกลางของเกี๋ยวดัวหนอน

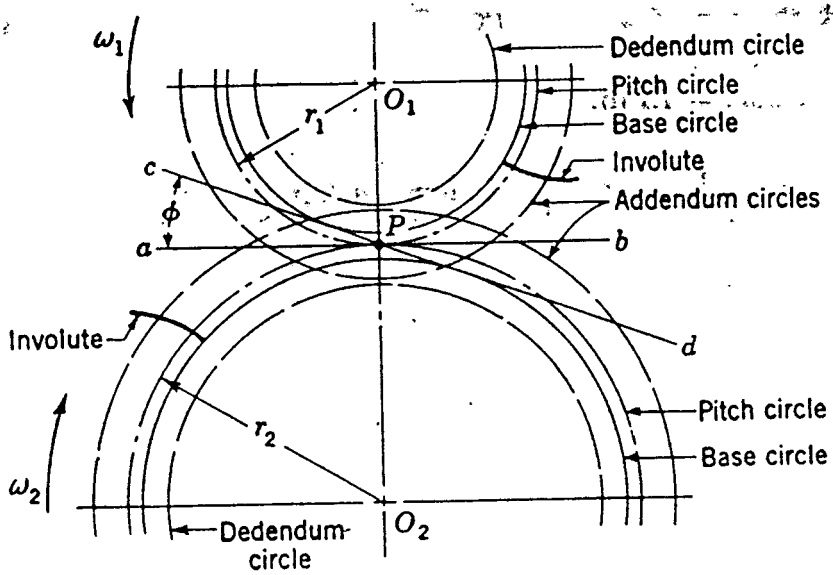
λ_w = มุมของเฟืองหนอน

dw = เส้นผ่าศูนย์กลางกลางของเฟืองหนอน

dw = เส้นผ่าศูนย์กลางกลางของเกี๋ยวดัวหนอน

5.1.2 เฟืองตรง (Spur gear)

หลังจากส่งกำลังผ่านชุดเฟืองหนอน ได้ทอร์กที่ค่าพอสมควรก็นำมาผ่านชุดเฟืองตรง เพื่อคำนวณหาค่าทอร์กตามที่ที่เราต้องการ โดยเฟืองตรงนั้นทำหน้าที่ส่งกำลังและการหมุนจากเพลานึงไปยังอีกเพลานึงที่ขนาดกัน ส่วนมากเฟืองขับ (driving gears) จะมีขนาดเล็กกว่าเฟืองตาม (driven gear) และมีชื่อเรียกว่า พิเนียน (pinion) ดังแสดงในรูป



รูปที่ 5.2 ลักษณะของเฟืองตรงอัตราทดของเฟือง (Velocity ratio)

M_w คืออัตราส่วนระหว่างความเร็วเชิงมุมของเฟืองขับต่อความเร็วเชิงมุมของเฟืองตาม ถ้าให้ "1" และ "2" แทนเฟืองขับและเฟืองตาม จะได้ว่า

$$M_w = w_1 / w_2 = n_1 / n_2 = d_2 / d_1 = N_2 / N_1$$

w = ความเร็วเชิงมุม, rad/S

n = ความเร็วรอบ rpm

d = เส้นผ่านศูนย์กลางพิตช์, mm หรือ in

N = จำนวนฟัน

5.3 การออกแบบโครงสร้างในส่วนต่าง ๆ ของแขนกล

การออกแบบส่วนประกอบต่าง ๆ เพื่อสร้างแขนกล สามารถทำได้ตามขั้นตอนดังนี้

1) กำหนดขนาดของแขนกลต่าง ๆ เช่น ชนิดของแขนกลที่ต้องการสร้าง จำนวนองศาอิสระ ความเร็วของการเคลื่อนที่

2) คำนวณน้ำหนักโดยประมาณของส่วนประกอบต่าง ๆ เมื่อทราบขนาดต่าง ๆ ตามขั้นตอนที่ 1 แล้ว เราก็ทำการร่างแบบของแขนกลขึ้นมาคร่าว ๆ จากนั้นให้นำแบบที่ร่างขึ้นมาคำนวณน้ำหนักโดยประมาณเพื่อจะได้นำไปคำนวณหาขนาดของมอเตอร์และเฟืองต่อไป

3) คำนวณขนาดของมอเตอร์และเฟืองที่ใช้ โดยน้ำหนักจากขั้นตอนที่ 2 มาคำนวณหาแรงบิด (torque) จากนั้นจึงเลือกขนาดของเฟืองและมอเตอร์ที่เหมาะสม

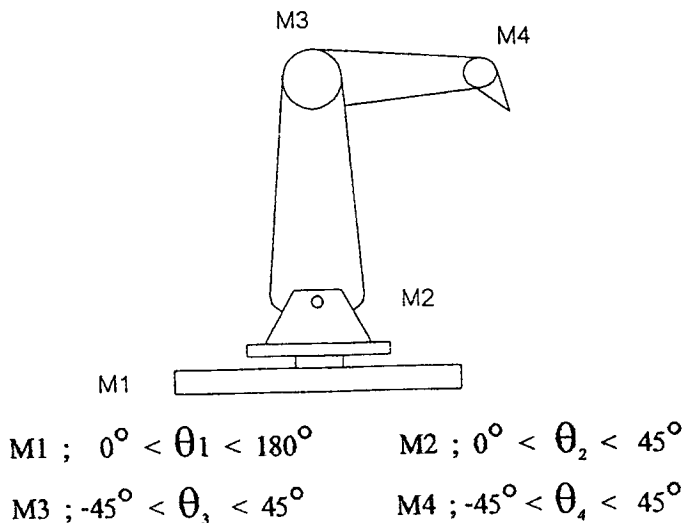
4) สำหรับส่วนที่ต้องการให้มีน้ำหนักเบา สามารถทำได้โดยใช้สายพานเป็นตัวส่งกำลัง ซึ่งขนาดของสายพานต้องคำนึงถึงแรงที่กระทำต่อสายพาน และขนาดที่มีขายในท้องตลาดด้วย

5) ออกแบบเพลลาเพื่อติดตั้งเฟือง

6) ทำการตรวจสอบขนาดและรูปร่างของชิ้นส่วนต่าง ๆ เพื่อคว่าขนาดที่กำหนดขึ้นสามารถนำมาประกอบเข้าด้วยกันได้หรือไม่ หรือมีขายในท้องตลาดหรือไม่ หากขนาดที่ได้ไม่เป็นไปตามที่ต้องการ ก็ให้เริ่มต้นการออกแบบใหม่ตั้งแต่ขั้นตอนที่ 2 ถึงขั้นตอนที่ 5 จนกว่าจะได้ขนาดที่เหมาะสม

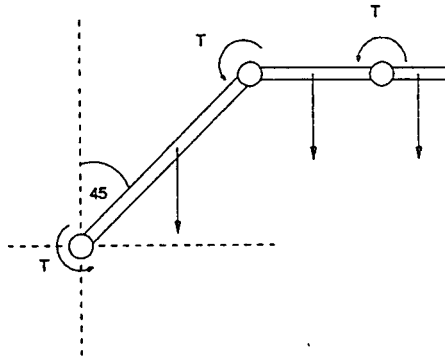
จากที่กล่าวมาทั้งหมดข้างต้นก็จะแจจรายละเอียดของชุดแขนกลที่ได้ออกแบบไว้ดังนี้

อันดับแรกเริ่มกำหนดขอบเขตการเคลื่อนที่ของโครงสร้างแขนกลในแต่ละส่วนซึ่งลักษณะของโครงสร้างของแขนกลแสดงดังรูป



รูปที่ 5.3 รูปแสดงโครงสร้างของแขนกล

ในการออกแบบและคำนวณจะพบว่าลักษณะการวางตัวของแขนซึ่งทำให้เกิดแรงบิด (torque) ค่ามากที่สุด นั้นแสดงคร่าว ๆ ได้ดังรูป



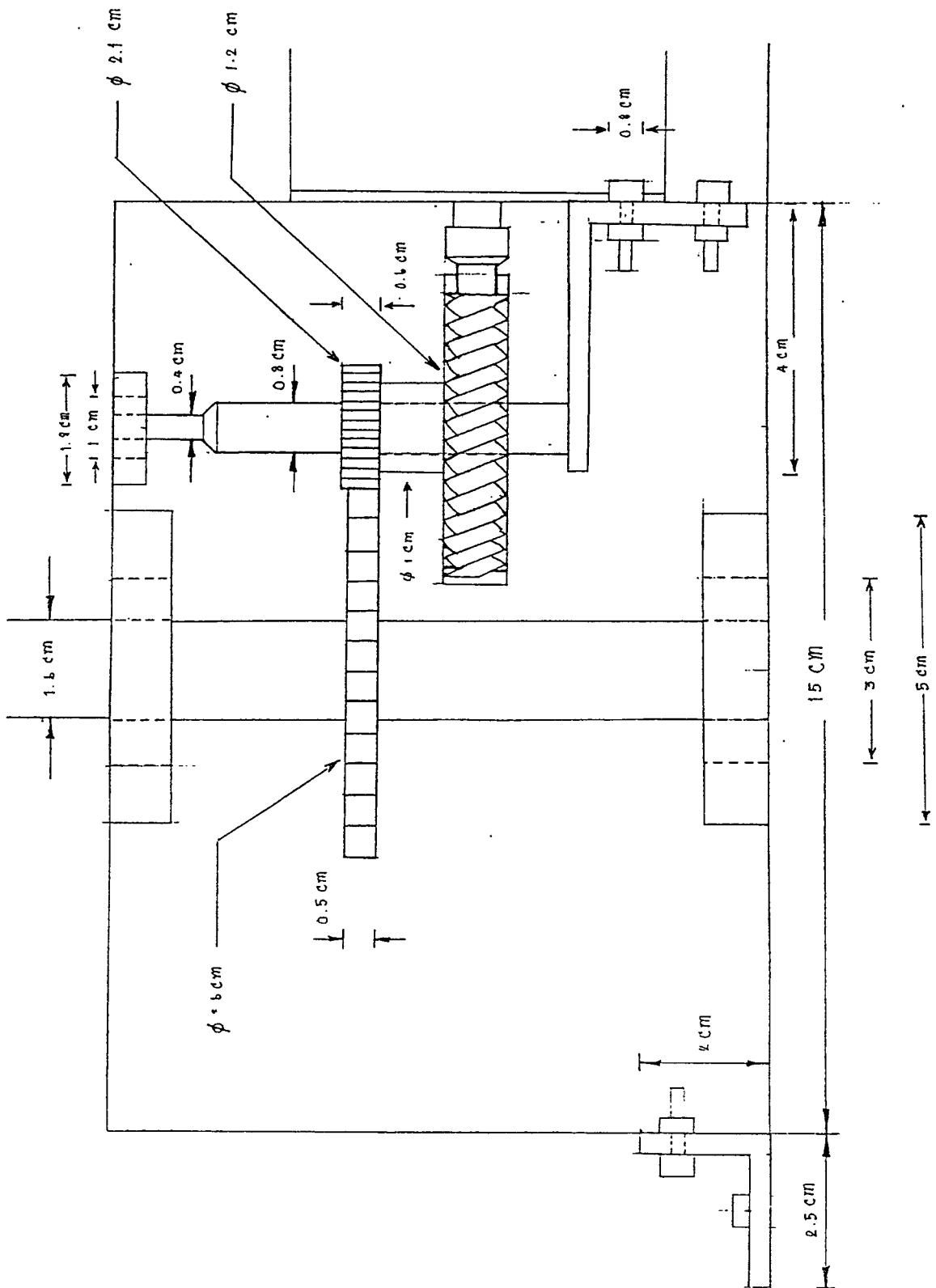
รูปที่ 5.4 แสดงลักษณะการวางตัวของแขนกลที่เกิดค่าทอร์กสูงสุด

5.3.1 ส่วนฐาน (Base Part)

การทำงานของชุดแขนกลทั้งชุดจะประกอบด้วยอุปกรณ์ดังต่อไปนี้

- สเต็ปปีงมอเตอร์ มีค่าทอร์ก 0.055 N.m
- ชุดเฟืองหนอน ที่มีอัตราทด 52 : 1
- ชุดเฟืองตรง ประกอบด้วยเฟืองขับและเฟืองตาม ที่มีอัตราทด 2.9 : 1

โดยเริ่มจากตัวต้นกำลังสเต็ปปีงมอเตอร์ ทำการประกอบชุดเฟืองหนอนเข้ากับอัตราทดของตัวมอเตอร์ โดยตำแหน่งของแกนจะทำมุม 90 องศา กับเกียร์ตัวหนอน ซึ่งมีจำนวนฟัน 48 ฟัน โดยเกียร์ตัวหนอนนี้จะยึดติดอยู่บนแกนเดียวกับเฟืองตรง 21 ฟัน จะหมุนด้วยความเร็วรอบเท่ากับชุดเฟือง 21 ฟันนี้ ก็จะไปขับชุดเฟือง 61 ฟัน ซึ่งเป็นชุดสุดท้ายที่ยึดติดกับแกนหมุนก็สามารถขับเคลื่อนโครงสร้างในส่วนนี้ได้ รูปที่ 5.5 แสดงโครงสร้างของส่วนฐานโดยในส่วนของฐานนี้ประกอบขึ้นโดยมีฐานหมุนเป็นตัวรับน้ำหนัก แขนกล สามารถหมุนอิสระได้ 180 องศา โดยผ่านชุดเฟืองทั้งหมดที่มีอัตราทด 150 : 1 สามารถให้ค่าทอร์กสูงสุด 8.25 N.m ซึ่งเมื่อสเต็ปปีงมอเตอร์ถูกกระตุ้น 1 พัลส์ (1.8 องศา / สเต็ป) ทำให้ส่วนฐานหมุนไป 0.012 องศา



รูปที่ 5.5 แสดงโครงสร้างของตัวฐาน

5.3.2 ส่วนอุปกรณ์หัวเจาะสว่าน

การทำงานของโครงสร้างส่วนนี้ประกอบด้วยอุปกรณ์ดังต่อไปนี้

- มอเตอร์กระแสตรง (DC MOTOR) 8 โวลต์ ขนาดน้ำหนัก 80 กรัม
- ดอกสว่านสำหรับเจาะ
- เฟรมโครงสร้าง (ประมาณน้ำหนักเบา)

คำนวณ ทอร์ก ที่เกิดจากน้ำหนักของส่วนนี้

$$\begin{aligned} T &= F \times R = (mg) \times r \\ &= (0.08 \times 9.81) \times 0.08 \text{ m} = 0.063 \text{ N.m} \end{aligned}$$

การทำงาน ในส่วนนี้เป็นส่วนปลายสุดในการนำอุปกรณ์เจาะลงไปเจาะในตำแหน่งที่ต้องการ บนแผ่นปริ้นซ์ ในการเคลื่อนที่สามารถหมุนได้องศาเท่ากับ $-45^\circ < \theta < 45^\circ$ จากสถานะเริ่มต้น ซึ่งส่วนนี้เคลื่อนที่โดยอาศัยการทำงานของมอเตอร์ผ่านชุดเฟืองที่ประกอบอยู่ในส่วนท่อนแขน ระหว่างข้อต่อกับข้อมือในหัวข้อถัดไป

5.3.3 ส่วนช่วงแขนท่อนกลาง (ส่วนควบคุมการหมุนของข้อมือ)

การทำงานของชุดโครงสร้างนี้ประกอบด้วยอุปกรณ์ดังนี้

- สเต็ปปีงมอเตอร์ 7.28 โวลต์ ขนาดน้ำหนัก 0.2 Kg มีค่าทอร์ก 0.01 N.m
- ชุดเฟืองหนอน มีอัตราทด 31 : 1
- เฟรมโครงสร้าง คำนวณน้ำหนักโดยรวมได้ 0.05 Kg

การทำงาน ของส่วนนี้จะ เป็นชุดอุปกรณ์ที่ใช้ขับเคลื่อนส่วนโครงสร้างที่เจาะแผ่นปริ้นซ์ โดยเริ่มจากตัวต้นกำลังสเต็ปปีงมอเตอร์ ส่งกำลังผ่านชุดเฟืองหนอนที่มีอัตราทด 31 : 1 สามารถส่งค่าทอร์กสูงสุด N.m ไปขับเคลื่อนส่วนปลายได้

คำนวณ ค่าทอร์กสูงสุดที่เกิดจากโครงสร้างส่วนนี้และจากส่วนอุปกรณ์หัวเจาะ

$$\begin{aligned} T &= \{(mg_{\text{สเต็ปปีงมอเตอร์}}) \times (0.1)\} + \{(mg_{\text{เฟรม}}) \times (0.09)\} + \{(mg_{\text{ดรัมมอเตอร์}}) \times (0.1 + 0.17)\} \\ &= (0.2 \times 9.81 \times 0.1) + (0.05 \times 9.81 \times 0.09) + (0.08 \times 9.81 \times 0.27) \\ &= 0.452 \text{ N.m} \end{aligned}$$

เนื่องจากค่าทอร์กที่เกิดขึ้น ดังนั้นการที่จะยกและเคลื่อนที่ส่วนโครงสร้างนี้ต้องอาศัยกำลังที่มีค่าทอร์กสูงกว่า 0.452 N.m ซึ่งอุปกรณ์ชุดส่งกำลังที่จะขับเคลื่อนจะประกอบอยู่ในส่วนของช่วงแขนท่อนบนระหว่างไหล่กับข้อต่อดังแสดงในหัวข้อต่อไป

5.3.4 ส่วนช่วงแขนท่อนบน (ส่วนที่ควบคุมการหมุนของข้อศอก)

การทำงานของชุดโครงสร้างนี้ประกอบด้วยอุปกรณ์ดังต่อไปนี้

- สเต็ปปีงมอเตอร์ 5 โวลท์ ขนาดน้ำหนัก 0.5 Kg มีค่าทอร์ก 0.035 N.m
- ชุดเฟืองตรง มีอัตราทด 2 : 1
- ชุดเฟืองหนอน มีอัตราทด 52 : 4
- เฟรมโครงสร้างพร้อมชุดเฟือง คำนวณน้ำหนักโดยรวมได้ 0.3 Kg

การทำงาน ของอุปกรณ์ในส่วนโครงสร้างนี้ จะเป็นส่วนควบคุมการหมุนของข้อศอก โดยเริ่มจากใช้สเต็ปปีงมอเตอร์เป็นตัวขับเคลื่อนกำลัง ส่งกำลังผ่านชุดเฟืองหนอนและชุดเฟืองตรงซึ่งมีอัตราทดรวมเท่ากับ 104.8 : 1 ทำให้สามารถให้ค่าทอร์กสูงสุด $(0.035 \times 104.8) = 3.668 \text{ N.m}$ ซึ่งจากการคำนวณค่าทอร์กนี้มีค่าเพียงพอที่จะขับเคลื่อนโครงสร้างตั้งแต่ข้อศอกไปได้ โดยเมื่อสเต็ปปีงมอเตอร์ถูกกระตุ้นจำนวน 1 พัลส์ (1.8 องศา / สเต็ป) ทำให้ส่วนข้อศอกหมุนไป 0.01717 องศา

คำนวณ ค่าทอร์กสูงสุดที่เกิดจากโครงสร้างส่วนแขนท่อนบน - ส่วนแขนท่อนกลาง - ส่วนอุปกรณ์เจาะแผ่นปริ้นซ์

$$\begin{aligned}
 T &= (mg) \times r \\
 &= \{ (mg_{\text{สเต็ปปีงมอเตอร์ 5 โวลท์}}) \times (0.08 \cos 45^\circ) \} + \{ (mg_{\text{เฟรม}}) \times (0.1 \cos 45^\circ) \} + \{ (mg_{\text{สเต็ปปีงมอเตอร์ 7.28 โวลท์}}) \} \times \{ (0.1 + 0.2 \cos 45^\circ) \} + \{ (mg_{\text{เฟรม}}) \times (0.09 + 0.2 \cos 45^\circ) \} + \{ (mg_{\text{ชุดเฟืองมอเตอร์}}) \times (0.2 \cos 45^\circ + 0.17 + 0.1) \} \\
 &= (0.5 \times 9.810 \times 0.057) + (0.3 \times 9.81 \times 0.07) + (0.2 \times 9.81 \times 0.24) + (0.05 \times 9.8 \times 10.23) + (0.08 \times 9.81 \times 0.41) \\
 &= 0.279 + 0.2 + 0.47 + 0.113 + 0.32 = 1.383 \text{ N.m}
 \end{aligned}$$

คิดอัตราความปลอดภัย 25 % ดังนั้น จะได้ค่าทอร์ก = 1.729 N.m

ในการทำงานเพื่อให้สามารถขับเคลื่อนโครงสร้างในส่วนแขนท่อนบน - ส่วนแขนท่อนล่าง - ส่วนอุปกรณ์เจาะแผ่นปริ้นซ์จะต้องใช้กำลังส่งที่มีค่าทอร์กมากกว่า 1.729 N.m ซึ่งการทำงานของอุปกรณ์ที่ขับเคลื่อนในส่วนนี้จะประกอบอยู่ในชุดหัวไหล่ดังแสดงในหัวข้อถัดไป

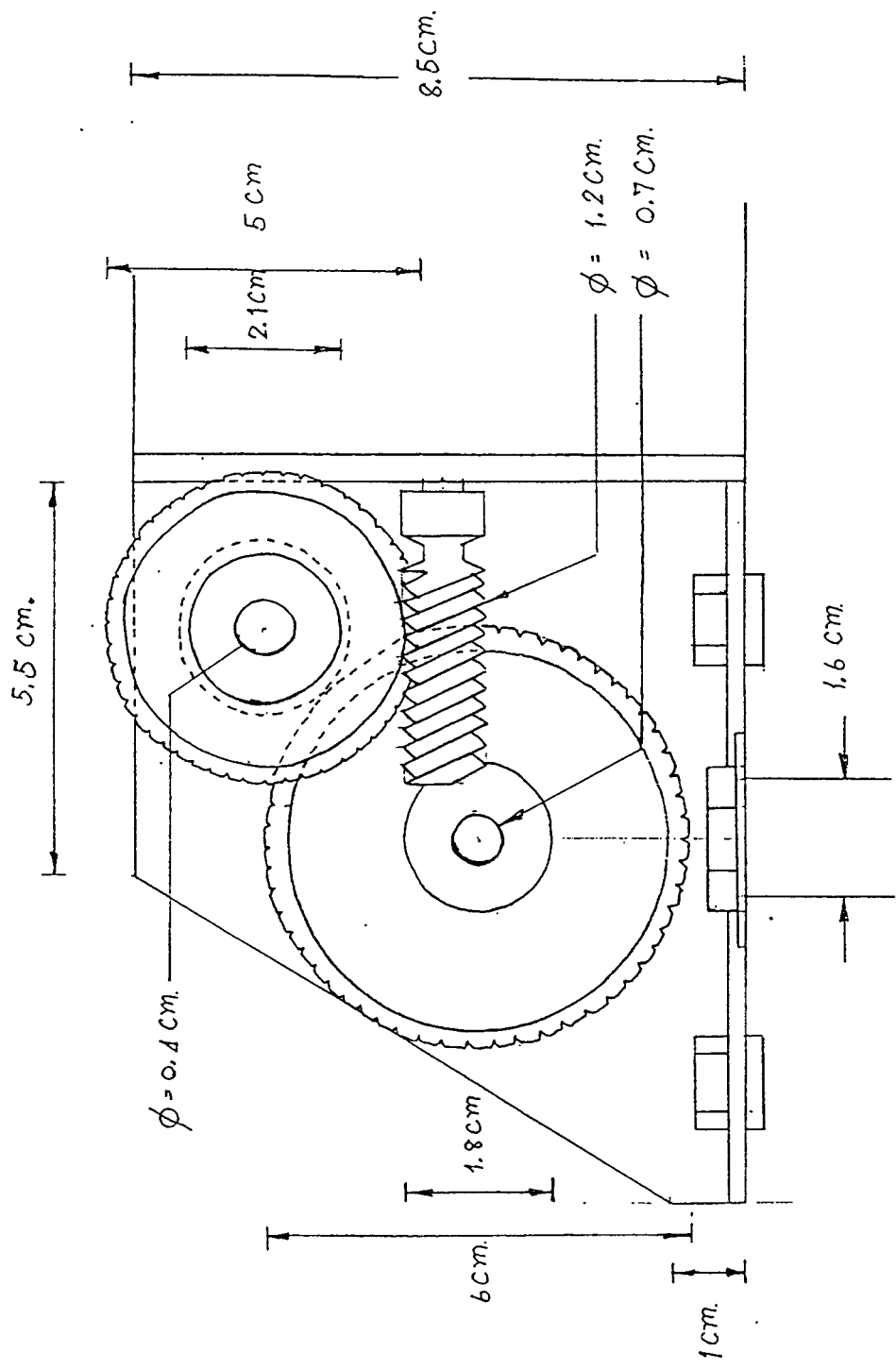
5.3.5 ส่วนหัวไหล่ (Elbow Part)

ในส่วนของหัวไหล่นี้เป็นส่วนที่จะต้องรับกำลังมากที่สุด เนื่องจากจะเป็นโครงสร้างที่จะ

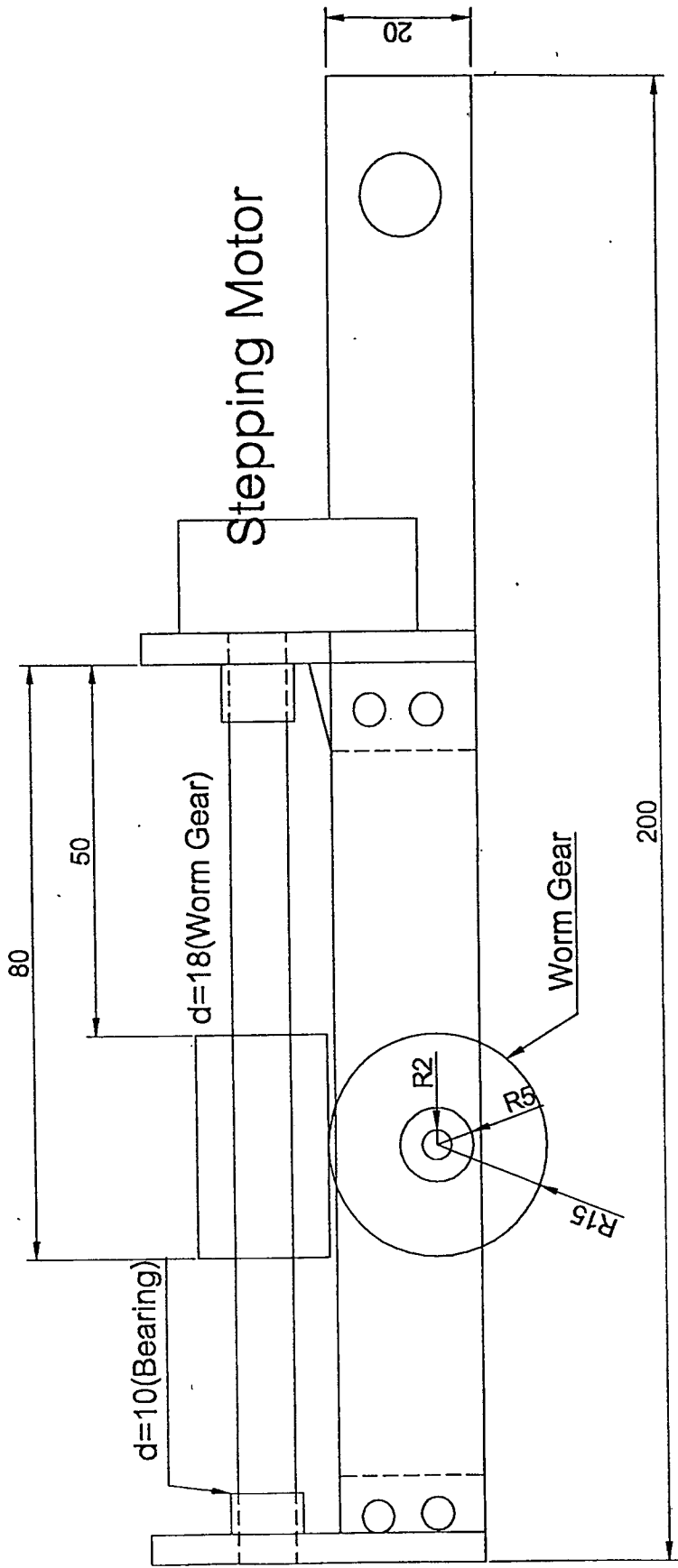
ต้องแบกภาระโครงสร้างของทั้งส่วนแขน ชุดข้อศอก และส่วนอุปกรณ์ต่างๆ ซึ่งในโครงสร้าง ประกอบด้วยอุปกรณ์ดังต่อไปนี้

- สเต็ปป์มอเตอร์ 1 ตัว ที่ให้อัตราทอร์ค 0.07 N.m
- ชุดเฟืองหนอน ที่มีอัตราทด 52.24 : 1
- ชุดเฟืองตรง ที่มีอัตราทด 2.9 : 1

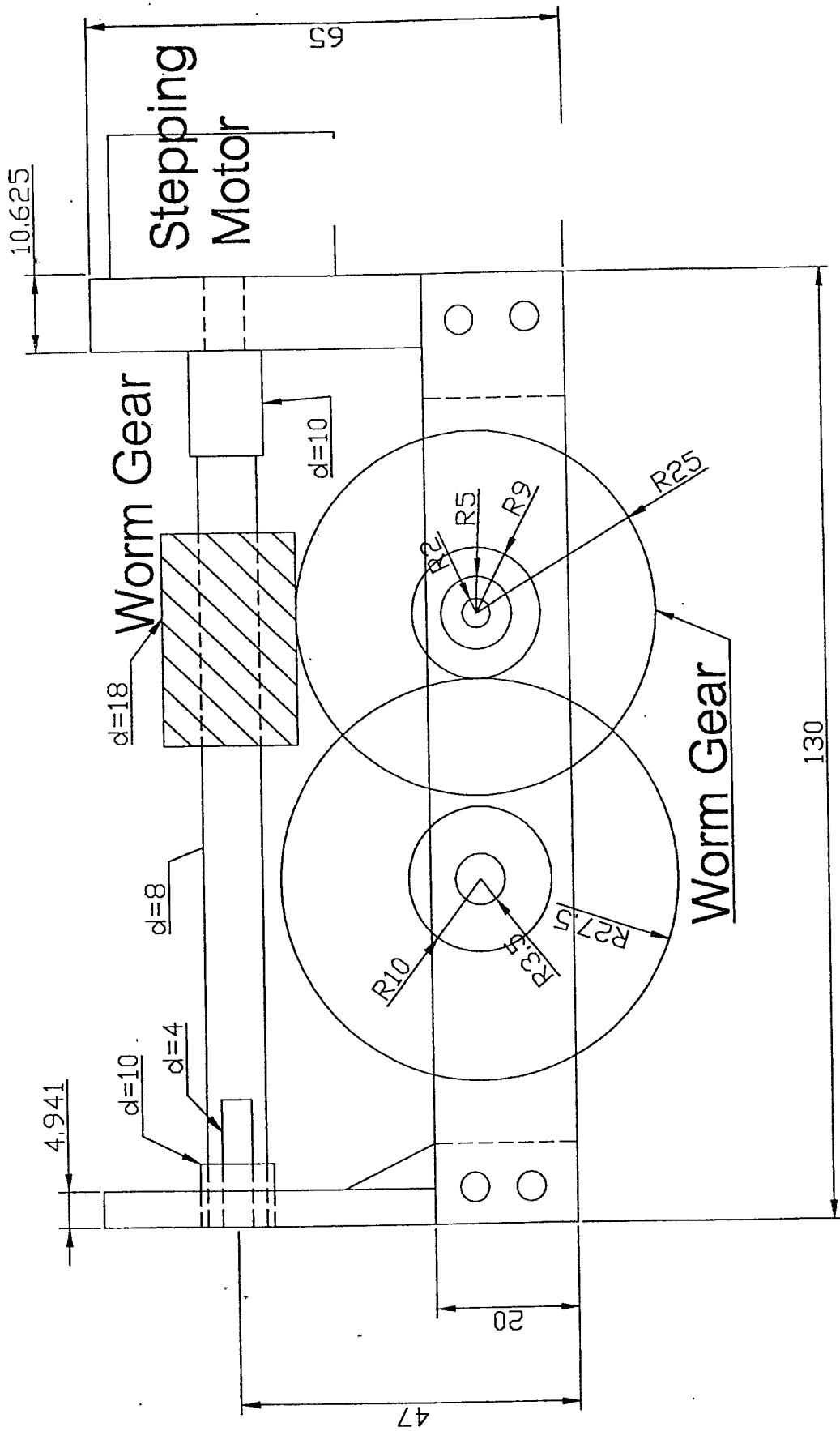
ซึ่งในส่วนหัวไหล่สามารถเคลื่อนที่ได้ 45 องศา ใช้สเต็ปป์มอเตอร์เป็นตัวขับเคลื่อน ส่งผ่านชุดเฟืองที่มีอัตราทดรวม 151 : 1 ดังนั้นสามารถให้ค่าทอร์คสูงสุด $151 \times 2.9 = 10.57$ N.m ซึ่งมีค่าพอที่จะไปขับเคลื่อนโครงสร้างทั้งหมด และเมื่อสเต็ปป์มอเตอร์ถูกกระตุ้น 1 พัลส์ (2 องศา / สเต็ป) จะทำให้มุมที่หัวไหล่เปลี่ยนไป 0.01324 องศา



รูปที่ 5.6 แสดง โครงสร้างของส่วนหัวไหล



รูปที่ 5.7 แสดงโครงสร้างส่วนแขนท่อนล่าง



รูปที่ 5.8 แสดงโครงสร้างของแกนก่อนบน

บทที่ 6

การออกแบบวงจรทางไฟฟ้าและการควบคุมแขนกลโดยคอมพิวเตอร์

6.1 การออกแบบไฟตรง

เลือกใช้วงจรจ่ายไฟที่ใช้ไอซีรักษาระดับแรงดัน ซึ่งสามารถทำการออกแบบในส่วนต่างๆ ได้ดังนี้

6.1.1 การเลือกหม้อแปลง

พิจารณาค่าของแรงดันและกระแสไฟฟ้าที่มอเตอร์และวงจรควบคุมทั้งหมดต้องการได้ดังนี้

-มอเตอร์แต่ละตัวมีคุณสมบัติดังนี้

1. สเต็ปป์มอเตอร์ขนาด 5 โวลท์ (V) , 1.8° ต่อสเต็ป , กระแสสูงสุด 0.891 แอมแปร์ (A)
2. สเต็ปป์มอเตอร์ขนาด 7.28 โวลท์ (V) , 8° ต่อสเต็ป , กระแสสูงสุด 0.395 แอมแปร์ (A)
3. สเต็ปป์มอเตอร์ขนาด 7.4 โวลท์ (V) , 2° ต่อสเต็ป , กระแสสูงสุด 1.123 แอมแปร์ (A)
4. สเต็ปป์มอเตอร์ขนาด 8 โวลท์ (V) , 1.8° ต่อสเต็ป , กระแสสูงสุด 0.658 แอมแปร์ (A)
5. มอเตอร์กระแสตรง (DC) ขนาด 8 โวลท์ (V)

-วงจรควบคุม

ในส่วนของวงจรควบคุมประกอบไปด้วยวงจรถอดจิกเกทแบบทีทีแอลเบอร์ต่างๆดังนี้

- | | |
|-----------------------------------------------|-------------|
| 1. 74LS00 , กระแสไฟเลี้ยงสูงสุด (Icc) 0.4 mA | จำนวน 1 ตัว |
| 2. 74LS08 , กระแสไฟเลี้ยงสูงสุด (Icc) 0.85 mA | จำนวน 1 ตัว |
| 3. 74LS32 , กระแสไฟเลี้ยงสูงสุด (Icc) 1 mA | จำนวน 1 ตัว |
| 4. 74LS125 , กระแสไฟเลี้ยงสูงสุด (Icc) 20 mA | จำนวน 9 ตัว |
| 5. 74LS138 , กระแสไฟเลี้ยงสูงสุด (Icc) 10 mA | จำนวน 1 ตัว |
| 6. 74LS193 , กระแสไฟเลี้ยงสูงสุด (Icc) 34 mA | จำนวน 4 ตัว |

ในขณะที่แขนกลทำงานนั้นมอเตอร์แต่ละตัวจะทำงานสลับกันไปทีละตัวและจะต้องการกระแสสูงสุดเมื่อมอเตอร์กระแสตรงเริ่มทำงาน ฉะนั้นจะพบว่าแรงดันอย่างต่ำที่หม้อแปลงจะต้องจ่ายได้คือ 8 โวลท์ เพื่อจ่ายให้กับสเต็ปป์มอเตอร์ และกระแสที่จะต้องจ่ายได้คือกระแสที่สเต็ปป์มอเตอร์ขนาด 8 โวลท์ , มอเตอร์กระแสตรงและ ไอซีทีทีแอลทั้งหมดต้องการคือ

$$\therefore V_{dc} = 8 \text{ V}$$

$$\therefore I_{dc} = 1.123 + 0.377 = 1.5 \text{ A}$$

ดังนั้นสามารถหาค่าแรงดันและกระแสที่ขดทุติยภูมิ (Secondary) ได้ดังนี้

$$\text{จาก } V_p = V_{dc} + (V_{\text{ripple}}/2)$$

$$\text{ต้องการ } V_{\text{ripple}} = 2.5 V_{p-p}$$

$$\begin{aligned} V_p &= 8 + (2.5/2) \\ &= 9.5 \text{ V} \end{aligned}$$

$$\begin{aligned} \text{จะได้ } V_{\text{rms}} &= V_p/1.414 \\ &= 9.45/1.414 \\ &= 6.89 \text{ Vrms} \end{aligned}$$

เพื่อความปลอดภัย 30 % จะได้ $V_{\text{rms}} = 8.96 \text{ Vrms}$

$$\begin{aligned} \text{และ จาก } I_{\text{rms}} &= 1.2I_{dc} \\ &= (1.2)(1.5) \\ &= 1.8 \text{ Arms} \end{aligned}$$

เพื่อความปลอดภัย 30 % จะได้ $I_{\text{rms}} = 2.34 \text{ Arms}$

∴ เลือกใช้หม้อแปลงที่สามารถจ่ายแรงดันได้ 9 Vrms และจ่ายกระแสได้ 3 Arms ที่ขดทุติยภูมิ

6.1.2 การเลือกวงจรเรียงกระแส

เลือกวงจรเรียงกระแสแบบบริดจ์

สามารถคำนวณอัตราทอนกระแสและแรงดัน ได้ดังนี้

$$\begin{aligned} \text{จาก } V &= V_p - 1.2 \\ &= (9 \times 4.14) - 1.2 \end{aligned}$$

$$\therefore V = 11.53 \text{ V}$$

และไดโอดต้องทนกระแสกระชากในขณะที่มีการประจุที่ตัวเก็บประจุได้ดังนี้

$$\begin{aligned} \text{จาก } \theta_{\text{on}} &= \cos^{-1} \{(V_p - V_{\text{ripple}})/V_p\} \\ t_{\text{on}} &= \theta_{\text{on}}/2\pi f_{\text{ripple}} \\ I_p &= 2I_{dc}/(t_{\text{on}} \times f_{\text{ripple}}) \end{aligned}$$

เมื่อ θ_{on} คือ คาบเวลาที่มีการประจุของตัวเก็บประจุ (เรเดียน)

t_{on} คือ คาบเวลาที่มีการประจุของตัวเก็บประจุ (วินาที)

V_p คือ ค่าแรงดันสูงสุด (โวลท์)

I_p คือ ค่ากระแสสูงสุด (แอมแปร์)

Vripple คือ ค่าแรงดันกระเพื่อม (โวลท์)

Fripple คือ ค่าความถี่ในการกระเพื่อม (เฮิรท์)

$$\begin{aligned}\text{จะได้ } \theta_{on} &= \cos^{-1} \{(9.75-1.75)/9.75\} \\ &= 0.531 \text{ rad}\end{aligned}$$

$$\begin{aligned}t_{on} &= 0.531 / (2\pi \times 100) \\ &= 0.844 \text{ ms}\end{aligned}$$

$$\begin{aligned}\therefore I_p &= (2 \times 1.50) / (0.844 \times 10^{-3} \times 100) \\ I_p &= 35.5 \text{ A}\end{aligned}$$

เพื่อความปลอดภัย 30 % จะได้ $I_p = 46 \text{ A}$

ดังนั้นเลือกใช้บริดจ์ไดโอดที่สามารถทนแรงดันได้ 15 V ทนกระแสได้ 15 A

6.1.3 การเลือกตัวเก็บประจุ

เนื่องจากจำนวนประจุในขณะที่ได้โอดมีการนำกระแสและหยุดนำกระแสต้องมีค่าเท่ากัน
จะได้

$$\begin{aligned}\text{จาก } Q_{in} &= Q_{out} \\ (C) \times (V_{ripple}) &= I_{dc} / F_{ripple}\end{aligned}$$

เมื่อ Q_{in} คือ จำนวนประจุที่ถูกระจุยในตัวเก็บประจุ (คูลอมบ์)

Q_{out} คือ จำนวนประจุที่คายออกจากตัวเก็บประจุ (คูลอมบ์)

I_{dc} คือ ค่ากระแสไฟตรงที่ต้องการ (แอมแปร์)

ต้องการแรงดันกระเพื่อม 3.5 Vp-p จะได้

$$\begin{aligned}C &= I_{dc} / (V_{ripple} \times F_{ripple}) \\ &= 1.5 / (3.5 \times 100) \\ &= 4286 \mu\text{F}\end{aligned}$$

เลือกใช้ตัวเก็บประจุขนาด 4700 μF

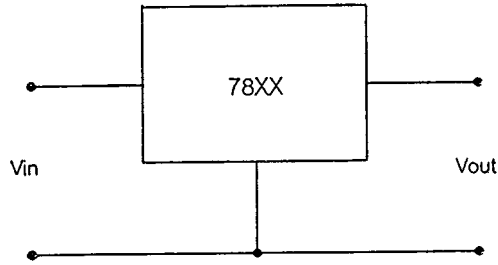
และตัวเก็บประจุต้องทนแรงดันได้อย่างน้อยเท่ากับ $V_p = 12.7 \text{ V}$ เพื่อความปลอดภัย 200 %

\therefore เลือกตัวเก็บประจุขนาด 4700 μF ทนแรงดันได้ 50 V

6.1.4 การออกแบบวงจรรักษาระดับแรงดัน

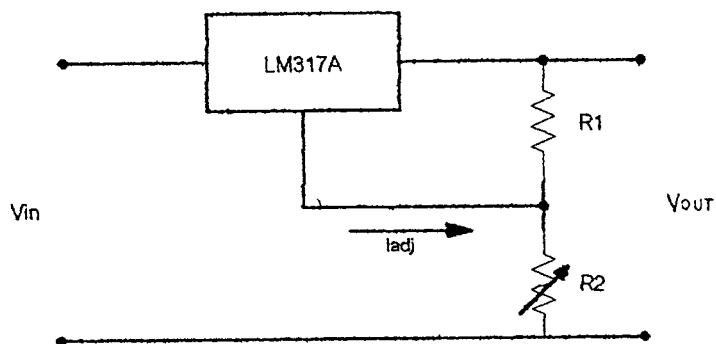
สามารถเลือกใช้ไอซีรักษาระดับแรงดันที่เหมาะสมกับมอเตอร์ต่างๆ ได้ดังนี้

-เลือกใช้ไอซีเบอร์ 78XX ที่ให้ระดับแรงดันเอาต์พุตคงที่ค่าหนึ่ง สำหรับจ่ายให้กับวงจรควบคุมและสแต็ปปีงมอเตอร์ขนาด 5 V และ 8 V และมอเตอร์ไฟฟ้ากระแสตรงขนาด 8 V ซึ่งสามารถออกแบบวงจรได้ดังรูป



รูปที่ 6.1 แสดงวงจรรักษาระดับแรงดันโดยใช้ไอซีตระกูล 78XX

- เลือกใช้ไอซีเบอร์ 317T ที่สามารถปรับค่าแรงดันเอาต์พุตได้ สำหรับจ่ายให้กับสแต็ปปีงมอเตอร์ขนาด 7.28 V และ 7.4 V ซึ่งสามารถออกแบบวงจรได้ดังรูป



รูปที่ 6.2 แสดงวงจรรักษาระดับแรงดันที่สามารถปรับค่าแรงดันเอาต์พุตได้

สามารถคำนวณหาค่าแรงดันเอาต์พุตได้ดังนี้

$$V_{out} = 1.25\{1+(R_2/R_1)\}+(I_{adj})(R_2)$$

เนื่องจากกระแส I_{adj} มีค่าน้อยมาก (ประมาณ $100 \mu A$) สามารถตัดทิ้งได้เมื่อเทียบกับกระแสเอาต์พุต

ดังนั้นสามารถคำนวณหาค่าความต้านทานที่ต้องใช้ในวงจรได้ดังนี้

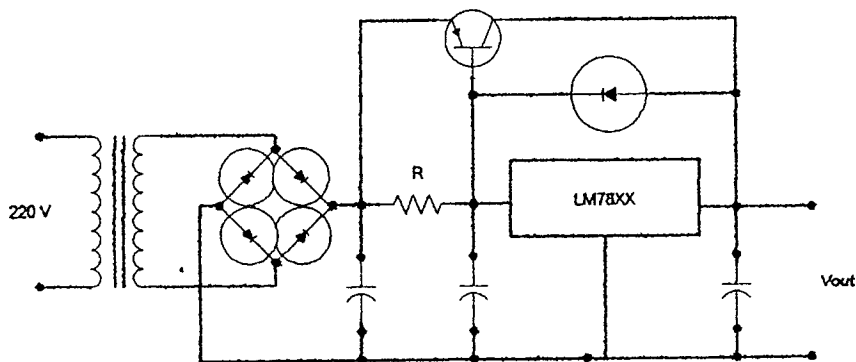
1. ต้องการ $V_{out} = 7.28 V$

$$\begin{aligned} \text{เลือก } R_1 = 1K\Omega \text{ จะได้ } R_2 &= \{(V_{out} \times R_1)/1.25\} - R_1 \\ &= \{(7.28 \times 1000)/1.25\} - 1000 \\ R_2 &= 4824 \Omega \approx 4.8 K\Omega \end{aligned}$$

2. ต้องการ $V_{out} = 7.4 V$

$$\begin{aligned} \text{เลือก } R_1 = 1K\Omega \text{ จะได้ } R_2 &= \{(7.4 \times 1000)/1.25\} - 1000 \\ R_2 &= 4920 \Omega \approx 5 K\Omega \end{aligned}$$

แต่เนื่องจากไอซีรักษาระดับแรงดันทั่วไปไม่สามารถจ่ายกระแสได้ถึงระดับที่มอเตอร์ต้องการ ดังนั้นจึงจำเป็นที่จะต้องมียังวงจรช่วยในการเพิ่มกระแส โดยใช้คุณสมบัติในการขยายกระแสของทรานซิสเตอร์ ซึ่งสามารถคำนวณได้ดังรูป



รูปที่ 6.3 แสดงวงจรช่วยจ่ายกระแส

ต้องการให้กระแสเอาต์พุตสูงสุด 2 A โดยให้ไหลผ่านไอซีรักษาระดับแรงดัน 0.3 A และ กระแสที่เหลือให้ไหลผ่านตัวทรานซิสเตอร์แทน

$$\therefore \text{เลือกใช้ค่า } R = 0.6/0.3 = 2 \Omega$$

พิจารณาแรงดันตกคร่อมทรานซิสเตอร์สูงสุดเท่ากับ

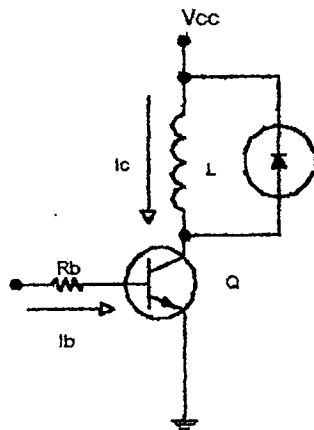
$$V_p - V_{out(\min)} = 12.73 - 5 = 7.73 \text{ V}$$

เกิดพลังงานความร้อนสูงสุด $P(\max) = IV = 2 \times 7.73 = 15.46 \text{ W}$

\therefore เลือกทรานซิสเตอร์เบอร์ MJE2955T ซึ่งมีคุณสมบัติคือ $V_{ce(\max)} = 80\text{V}$, $I_{c(\max)} = 5\text{A}$ และ $P(\max) = 50 \text{ W}$ มาเป็นตัวช่วยจ่ายกระแสให้กับ ไอซี

6.2 การออกแบบวงจรขยายสัญญาณเพื่อขับมอเตอร์

เลือกใช้วงจรขยายแบบขั้วเดี่ยวเนื่องจากสตีปิงมอเตอร์ที่ใช้เป็นแบบปรับค่าความต้านทานแม่เหล็กได้ (VR) โดยวงจรที่ใช้แสดงดังรูป



รูปที่ 6.4 แสดงวงจรขยายกระแสแบบขั้วเดี่ยว

เลือกใช้ทรานซิสเตอร์เบอร์ MJE3055T ซึ่งมีอัตราขยายกระแส (β) เท่ากับ 40 เท่า จากวงจรสามารถเลือกค่าความต้านทาน R_b เพื่อให้ทรานซิสเตอร์ทำงานในแบบอิ่มตัว (Saturate) ได้ดังนี้

1. สำหรับมอเตอร์ขนาด 5 V มีค่าความต้านทานของขดลวดในแต่ละเฟสเท่ากับ 5.5Ω

หาค่ากระแสคอลเลกเตอร์สูงสุด

$$I_{c(\max)} = \{V_{cc} - V_{ce(\text{sat})}\} / R$$

$$= (5 - 0.1) / 5.5 ; V_{ce(\text{sat})} \text{ ของ MJE3055T} = 0.1 \text{ V}$$

$$= 0.891 \text{ A}$$

$$\text{จาก } \beta(V_b - 0.6) / R_b \geq I_c(\text{max})$$

$$\text{จะได้ } R_b \leq \beta(V_b - 0.6) / I_c(\text{max})$$

$$\leq 40(5 - 0.6) / 0.891$$

$$\therefore R_b \leq 197.6 \Omega \text{ เลือก } R_b = 180 \Omega$$

2. สำหรับมอเตอร์ขนาด 7.28 V มีค่าความต้านทานของขดลวดในแต่ละเฟสเท่ากับ 18.2 Ω

หาค่ากระแสคอลเลกเตอร์สูงสุด

$$I_c(\text{max}) = \{V_{cc} - V_{ce}(\text{sat})\} / R$$

$$= (7.28 - 0.1) / 18.2 ; V_{ce}(\text{sat}) \text{ ของ MJE3055T} = 0.1 \text{ V}$$

$$= 0.395 \text{ A}$$

$$\text{จาก } \beta(V_b - 0.6) / R_b \geq I_c(\text{max})$$

$$\text{จะได้ } R_b \leq \beta(V_b - 0.6) / I_c(\text{max})$$

$$\leq 40(7.28 - 0.6) / 0.395$$

$$\therefore R_b \leq 445.6 \Omega \text{ เลือก } R_b = 440 \Omega$$

3. สำหรับมอเตอร์ขนาด 7.4 V มีค่าความต้านทานของขดลวดในแต่ละเฟสเท่ากับ 6.5 Ω

หาค่ากระแสคอลเลกเตอร์สูงสุด

$$I_c(\text{max}) = \{V_{cc} - V_{ce}(\text{sat})\} / R$$

$$= (7.4 - 0.1) / 6.5 ; V_{ce}(\text{sat}) \text{ ของ MJE3055T} = 0.1 \text{ V}$$

$$= 1.123 \text{ A}$$

$$\text{จาก } \beta(V_b - 0.6) / R_b \geq I_c(\text{max})$$

$$\text{จะได้ } R_b \leq \beta(V_b - 0.6) / I_c(\text{max})$$

$$\leq 40(7.4 - 0.6) / 1.123$$

$$\therefore R_b \leq 156.7 \Omega \text{ เลือก } R_b = 150 \Omega$$

4. สำหรับมอเตอร์ขนาด 8 V มีค่าความต้านทานของขดลวดในแต่ละเฟสเท่ากับ 12 Ω

หาค่ากระแสคอลเลกเตอร์สูงสุด

$$I_c(\text{max}) = \{V_{cc} - V_{ce}(\text{sat})\} / R$$

$$= (8-0.1) / 12 ; V_{ce(sat)} \text{ ของ MJE3055T} = 0.1 \text{ V}$$

$$= 0.658 \text{ A}$$

$$\begin{aligned} \text{จาก } \beta(V_b-0.6) / R_b &\geq I_c(\max) \\ \text{จะได้ } R_b &\leq \beta(V_b-0.6) / I_c(\max) \\ &\leq 40(8-0.6)/0.658 \\ \therefore R_b &\leq 267.3 \Omega \text{ เลือก } R_b = 240 \Omega \end{aligned}$$

6.3 การออกแบบตัวบอกรหัสและวงจรถ้อนกลับ

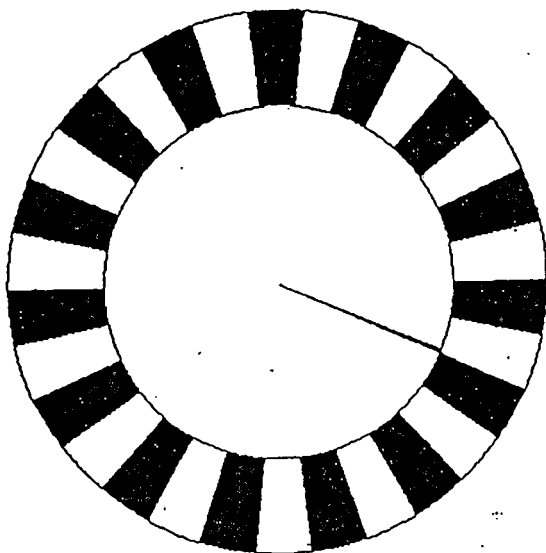
เลือกตัวบอกรหัสแบบเพิ่มค่า (Incremental) เนื่องจากมีราคาถูกและมีขนาดเล็กกว่า ซึ่งในการออกแบบต้องการจำนวนตำแหน่งของมอเตอร์ดังนี้

1. มอเตอร์ขนาด 1.8° ต่อสเต็ป ต้องการจำนวนตำแหน่งในหนึ่งรอบเท่ากับ $360^\circ / 1.8^\circ$ ซึ่งคิดเป็น 200 ตำแหน่ง

2. มอเตอร์ขนาด 2° ต่อสเต็ป ต้องการจำนวนตำแหน่งในหนึ่งรอบเท่ากับ $360^\circ / 2^\circ$ ซึ่งคิดเป็น 180 ตำแหน่ง

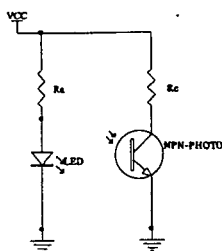
โดยการแบ่งงานตามแนวรัศมีออกเป็นแถบสีขาวและดำสลับกันไป ซึ่งในการเกิดสัญญาณบอกตำแหน่งแต่ละตำแหน่งนั้นจะต้องเกิดจากคู่ของแถบขาวดำหนึ่งคู่ ซึ่งในการสร้างจริงแล้วไม่สามารถสร้างตัวบอกรหัสที่มีขนาดเล็กและมีจำนวนของตำแหน่งครบตามที่ต้องการได้ เพราะต้องสร้างงานบอกรหัสที่มีแถบขาวดำทั้งหมดรวมกันถึง 400 ช่อง และ 360 ช่อง ตามลำดับ ซึ่งในการสร้างจะต้องใช้งานใหญ่มาก

ดังนั้นเราจึงทำการลดตำแหน่งของมอเตอร์ที่อยู่บนตัวบอกรหัสลงแล้วใช้ความสามารถของโปรแกรมเข้าช่วยในการบอกตำแหน่งที่แท้จริงของมอเตอร์ โดยทำการลดตำแหน่งลงจากการเพิ่มมุมในแต่ละตำแหน่งเป็น 24° ดังนั้นจะมีตำแหน่งที่อยู่บนตัวบอกรหัส 15 ตำแหน่งดังรูป แล้วจึงใช้โปรแกรมเข้าช่วยในการบอกตำแหน่งที่ถูกต้องต่อไป ซึ่งในการบอกตำแหน่งโดยใช้โปรแกรมจะนำเสนอในหัวข้อต่อไป



รูปที่ 6.5 แสดงงานบอกรหัสแบบเพิ่มค่าได้

ในส่วนของวงจรป้อนกลับนั้นใช้การตรวจนับโดยใช้แสง (Opto Sensor) ซึ่งประกอบไปด้วยคู่ของตัวส่งและรับแสงจำนวนหนึ่งคู่ โดยตัวส่งแสง (Transmitter) ใช้ไดโอดเปล่งแสงได้ (LED) และตัวรับแสงใช้โฟโตทรานซิสเตอร์ (Photo Transistor) ต่อกันเป็นตัวตรวจนับสัญญาณที่ได้มาจากงานบอกรหัส (Encoder) ซึ่งวงจรเป็นดังรูป



รูปที่ 6.6 แสดงวงจรตรวจนับสัญญาณ โดยใช้แสง

ในส่วนของตัวส่งแสงสามารถหาค่า R_a ซึ่งใช้ในการจำกัดกระแสที่ไหลผ่านตัว LED ได้ดังนี้

$$R_a = (V_{cc} - V_f) / I_a$$

เมื่อ R_a คือ ค่าความต้านทาน

V_{cc} คือ ค่าแรงดันที่ใช้งาน

V_f คือ ค่าแรงดันตกคร่อมไดโอดขณะใช้งาน

I_a คือ ค่ากระแสที่ไหลในวงจร

จากคุณสมบัติของ LED สีแดงจะมีค่า $V_f \approx 1.65 \text{ V}$ และมีค่ากระแสไหลสูงสุดไม่เกิน 100 mA ดังนั้นเลือกค่ากระแสให้ไหลไม่เกิน 20 mA และค่าแรงดันที่ใช้เท่ากับ 5 V จะได้

$$R_a = (5 - 1.65) / (20 \times 10^{-3}) = 167.5 \Omega$$

∴ เลือกใช้ R_a ค่า 200Ω

ในส่วนของตัวรับแสงสามารถหาค่า R_c ซึ่งใช้ในการแปลงกระแสที่ได้จากโฟโตทรานซิสเตอร์ออกมาเป็นแรงดันได้ดังนี้

$$R_c = \{V_{cc} - V_{ce}(\text{sat})\} / I_c$$

เมื่อ R_c คือ ค่าความต้านทานที่ใช้

V_{cc} คือ ค่าแรงดันที่ใช้งาน

$V_{ce}(\text{sat})$ คือ ค่าแรงดันอิมิต์ของโฟโตทรานซิสเตอร์

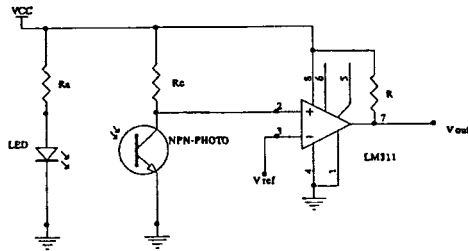
I_c คือ ค่ากระแสที่ไหลผ่าน R_c

โฟโตทรานซิสเตอร์มีค่ากระแส I_c ขณะได้รับแสง (Light current) เป็น 1 mA , มีค่าแรงดันอิมิต์เท่ากับ 0.1 V และค่าแรงดันที่ใช้เท่ากับ 5 V จะได้

$$R_c = (5 - 0.1) / 0.001 = 4900 \Omega$$

∴ เลือกใช้ R_c ค่า $5.6 \text{ K}\Omega$

แต่เนื่องจากสัญญาณที่ได้จากโฟโตทรานซิสเตอร์จะมีลักษณะที่ไม่เป็นพัลส์สี่เหลี่ยม ดังนั้นจึงใช้วงจรปรับแต่งสัญญาณให้มีลักษณะพัลส์สี่เหลี่ยมโดยใช้วงจรเปรียบเทียบแรงดัน (Voltage Comparator) ดังรูป

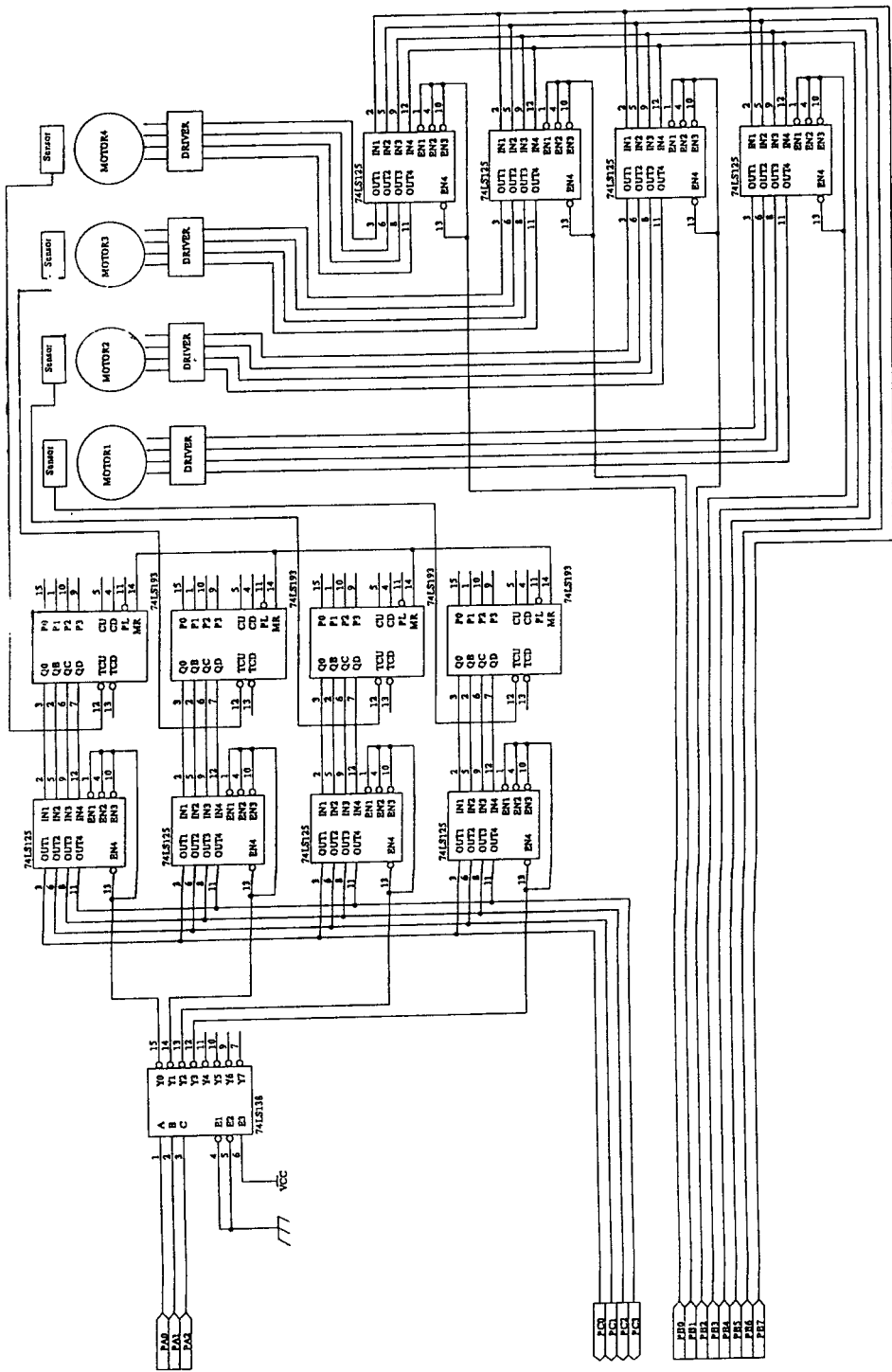


รูปที่ 6.7 แสดงวงจรปรับแต่งสัญญาณพัลส์

สัญญาณที่ได้นี้จะนำไปเปรียบเทียบกับค่าตำแหน่งที่กำหนดให้มอเตอร์หมุนในตอนแรก ถ้ามีความผิดพลาดเกิดขึ้นก็จะมีสัญญาณขับให้มอเตอร์หมุนไปยังค่าแห่งที่ถูกต้อง การป้อนกลับจะนำสัญญาณที่ตรวจจับได้จากเอ็นโค้ดเดอร์มาเปรียบเทียบกับค่าที่กำหนดไว้ในตอนแรก แล้วทำการคำนวณหาความแตกต่างเพื่อปรับแต่งให้มอเตอร์หมุนให้แขนกลเคลื่อนที่ไปยังจุดที่ต้องการ

สัญญาณจากเอ็นโค้ดเดอร์จะผ่านวงจรปรับแต่งสัญญาณพัลส์ให้มีรูปร่างที่คงที่ก่อนเข้าสู่วงจรมับ โดยใช้ไอซีทีทีแอลเบอร์ 74LS193 จำนวน 1 ตัวต่อ 1 ข้อต่อ ทั้งนี้เพราะตำแหน่งของมอเตอร์ถูกแทนด้วยตำแหน่งต่างๆเพียง 15 ตำแหน่ง ซึ่งไอซี 74LS193 หนึ่งตัวสามารถนับได้ 16 ตำแหน่ง (เอาท์พุทมี 4 บิต)

จากนั้นค่าที่นับได้จะถูกส่งไปประมวลผลในคอมพิวเตอร์ แต่จะต้องรับค่าเข้าไปทีละ 1 ข้อต่อเท่านั้นในแต่ละช่วงเวลา ดังนั้นจึงต้องมีวงจรที่ใช้เลือกค่าที่จะอ่าน โดยใช้ไอซีทีทีแอลเบอร์ 74LS125 ซึ่งเป็นบัฟเฟอร์แบบ 3 สถานะ จำนวน 4 บิต สัญญาณจากคอมพิวเตอร์จะผ่านวงจรถีโค้ดเดอร์โดยใช้ไอซีเบอร์ 74LS138 เพื่อทำการเลือกชุดของบัฟเฟอร์ให้ทำงานและยังมีสัญญาณที่จะต้องผ่าน ไปเดลิย์ค่าของ ไอซี 74LS193 ทุกครั้งในการหมุนแต่ละครั้ง ดังวงจร



รูปที่ 6.8 แสดงวงจรป้อนกลับ

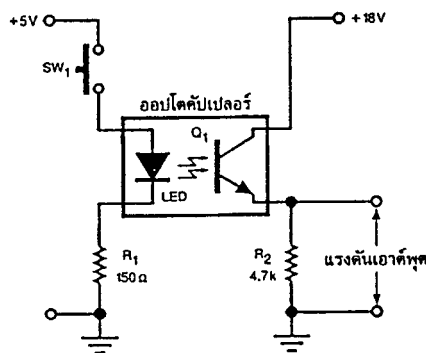
6.4 การออกแบบวงจรคัปปลิ่ง

สำหรับคาร์ดิอินเตอร์เฟสที่ใช้ในการเชื่อมต่อระหว่างคอมพิวเตอร์และแขนกลนี้ได้นำออปโตคัปเปลอร์เข้ามาใช้ เพื่อช่วยป้องกันการลัดวงจรจากภายนอกที่อาจส่งผลกระทบต่อคอมพิวเตอร์ได้ ซึ่งชนิดและหลักการทำงานของออปโตคัปเปลอร์ คือ

6.4.1 ออปโตคัปเปลอร์ (Opto - Coupler)

ออปโตคัปเปลอร์เป็นอุปกรณ์ที่ประกอบด้วย LED ซึ่งปกติเป็นชนิดอินฟราเรด และโฟโตทรานซิสเตอร์หรือโฟโตไดโอดซึ่งถูกผลิตมาคู่กันรวมอยู่ในตัวเดียวกัน

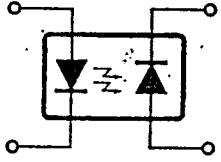
จากรูปเป็นวงจรพื้นฐานของออปโตคัปเปลอร์ โดยด้าน LED จะเป็นอินพุทของวงจรและด้านโฟโตทรานซิสเตอร์เป็นเอาต์พุทของวงจร จากรูปเมื่อสวิตช์ SW1 เปิดวงจร LED และทรานซิสเตอร์ Q1 จะยังไม่ทำงาน แต่เมื่อสวิตช์ SW1 ปิดวงจรจะมีกระแสไหลผ่าน LED และตัวต้านทาน R2 ซึ่งจะทำให้ Q1 ทำงานด้วย และจะได้แรงดันเอาต์พุตที่ตัวต้านทาน R2 จะเห็นว่าเอาต์พุทของวงจรถูกควบคุมโดยส่วนอินพุท แต่ทั้งสองส่วนถูกแยกออกจากกันทางไฟฟ้าอย่างสิ้นเชิง(ซึ่งเป็นหลักการสำคัญของออปโตคัปเปลอร์) วงจรนี้สามารถนำไปประยุกต์ใช้ได้ทั้งสัญญาณดิจิตอลและสัญญาณอะนาลอก



รูปที่ 6.9 แสดงวงจรออปโตคัปเปลอร์พื้นฐาน

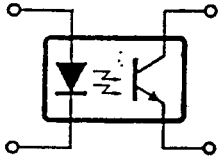
การทำลายกราวด์คู่ระหว่างวงจรสองวงจรอาจทำได้โดยใช้ออปโตคัปเปลอร์ โดย LED จะคัปปลิ่งผ่านแสงไปยังโฟโตทรานซิสเตอร์ ไดโอด หรือทรานซิสเตอร์ซึ่งอยู่ภายในภาชนะเดียวกัน ทำให้กราวด์คู่ถูกตัดออกจากกันโดยสิ้นเชิง เนื่องจากการคัปปลิ่งระหว่างวงจรทั้งสองจะเกิดขึ้นโดยมีแสงเป็นตัวกลางเท่านั้น

ออปโตคัปเปลอร์มีหลายแบบต่างกันตามชนิดเอาต์พุท คือ



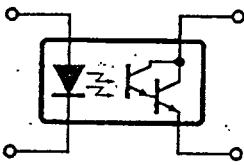
(ก) ชนิดเอาต์พุตเป็นโฟโตไดโอด

เอาต์พุตเป็นโฟโตไดโอด
เหมาะสำหรับงานกำลังไฟฟ้าต่ำ
สวิตซ์ความเร็วต่ำ



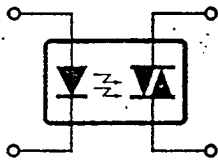
(ข) ชนิดเอาต์พุตเป็นโฟโตทรานซิสเตอร์

เอาต์พุตเป็นโฟโตทรานซิสเตอร์
เหมาะสำหรับงานกำลังไฟฟ้าปาน
กลาง ทำงานเป็นสวิตซ์ซึ่งความเร็วสูง
ซึ่งเป็นแบบที่นิยมใช้งานมากที่สุด



(ค) ชนิดเอาต์พุตเป็นโฟโตคาร์ลิงตันทรานซิสเตอร์

เอาต์พุตเป็น โฟโตคาร์ลิงตัน
ทรานซิสเตอร์ เหมาะสำหรับงานกำลัง
ไฟฟ้าสูง แรงดันต่ำ กระแสสูง งานไฟ
ฟ้าสลับกำลังสูง

(ง) ชนิดเอาต์พุตเป็นโฟโตไดโอดแยก
หรือเรียกว่า ไครเออแคปเปลอร์

เอาต์พุตเป็นโฟโตไดโอดแยก

รูปที่ 6.10 แสดงสัญลักษณ์ของออปโตคัปเปิลอร์แบบต่างๆ

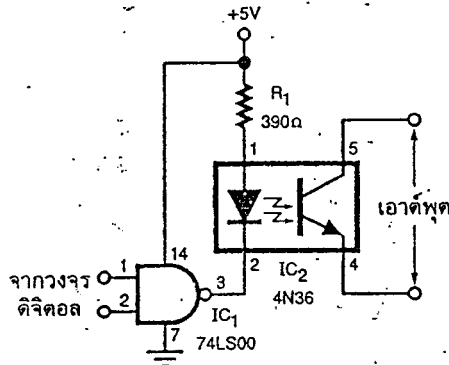
6.4.2 อัตราการถ่ายเท

อัตราการถ่ายเทกระแสไฟฟ้า (Current-Transfer Ratio: CTR) เป็นพารามิเตอร์ที่วัดถึงปริมาณกระแสไฟฟ้าที่ถูกถ่ายเทจากส่วนอินพุทของวงจรซึ่งใช้ LED อินฟราเรด ไปยังส่วนเอาต์พุทซึ่งเป็นอุปกรณ์ใดๆ โดยคิดในหน่วยเปอร์เซ็นต์ของการแยกกันทางไฟฟ้าอย่างสมบูรณ์ ซึ่งสามารถคิดออกมาเป็นอัตราการขยายกระแสหรืออัตราการสูญเสียทางไฟฟ้าก็ได้ เมื่อเทียบแล้วทรานซิสเตอร์ก็คือ การคิดหาอัตราการขยายกระแส หรือ h_{FE} นั่นเอง ซึ่งเปรียบเทียบกับกระแสอินพุทคือ I_B และกระแสเอาต์พุท คือ I_C แต่ I_B ในที่นี้คือกระแสไบอัสตรงของ LED อินฟราเรดหรือ I_F ในรูปเป็นวงจรออปโตคัปเปิลอร์ที่ส่วนเอาต์พุทเป็นโฟโตทรานซิสเตอร์

ค่า CTR ถ้าเป็น 100% หมายความว่ากระแสอินพุทเท่ากับกระแสเอาต์พุทพอดี หากต่ำกว่า 100% หมายความว่าเกิดความสูญเสียขึ้นภายในตัวออปโตคัปเปิลอร์ ถ้าเกิน 100% แสดงว่าออปโตคัปเปิลอร์ตัวนั้นมีความสามารถในการขยายกระแสไฟฟ้า

6.4.3 การใช้ลอจิกเกตในการควบคุมออปโตคัปเปิลอร์

การนำไอซีลอจิกเกตมาเชื่อมกับออปโตคัปเปิลอร์ทำได้โดยง่ายโดยต่อเข้ากับทางภาคอินพุท เพื่อควบคุมการทำงานของไดโอดอินฟราเรดภายในออปโตคัปเปิลอร์ ทั้งนี้เนื่องจากแรงดันไฟฟ้าเพียงไอซีลอจิกเกตและวงจรดิจิทัลโดยส่วนใหญ่จะมีค่า 5 V ซึ่งไม่สูงนัก ทำให้ใช้เป็นไฟเลี้ยงภาคอินพุทของออปโตคัปเปิลอร์ร่วมกันได้ เพียงต่อตัวต้านทานอนุกรมกับไดโอดอินฟราเรดเพื่อจำกัดกระแสเท่านั้น ซึ่งแสดงดังรูป



รูปที่ 6.11 แสดงการใช้ไอซีลอจิกเกตตระกูลทีแอลเพื่อควบคุมออปโตคัปเปิลอร์

ลอจิกเกตที่ใช้เป็นไอซีตระกูลทีทีแอล (TTL) จะมีความสามารถในการจ่ายกระแสสูงพอสมควร แต่ถ้าเป็นลอจิกเกตตระกูล CMOS ลอจิกเกตตระกูลนี้ จะมีความสามารถในการจ่ายกระแสเอาต์พุตต่ำกว่าตระกูล TTL จึงจำเป็นต้องต่อวงจรขับกระแสเอาต์พุตเพิ่มเติม

6.5 การออกแบบการ์ดอินเตอร์เฟซและวิธีควบคุมการหมุนของมอเตอร์

6.5.1 อุปกรณ์ที่ใช้ในวงจร

- วงจรการ์ดอินเตอร์เฟซ

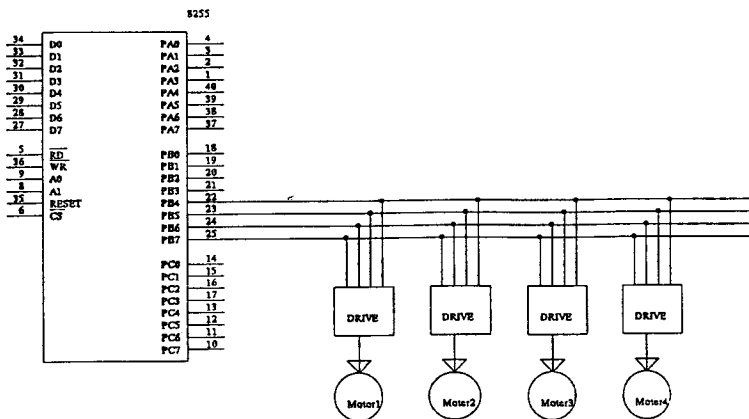
1. ไอซีเบอร์ 8255	1	ตัว
2. ไอซีเบอร์ 74LS08	1	ตัว
3. ไอซีเบอร์ 74LS32	1	ตัว
4. ไอซีเบอร์ 74LS245	2	ตัว
5. ไอซีเบอร์ 74LS688	1	ตัว
6. ไอซีเบอร์ 4N26	15	ตัว
7. คิฟสวิทช์ 8 บิต	1	ตัว
8. ความต้านทานค่า 1 k Ω	20	ตัว
9. ความต้านทานค่า 220 Ω	16	ตัว

- วงจรควบคุมการหมุนมอเตอร์

1. ไอซีเบอร์ 74LS125	9	ตัว
2. ไอซีเบอร์ 74LS193	4	ตัว
3. ไอซีเบอร์ 74LS138	1	ตัว

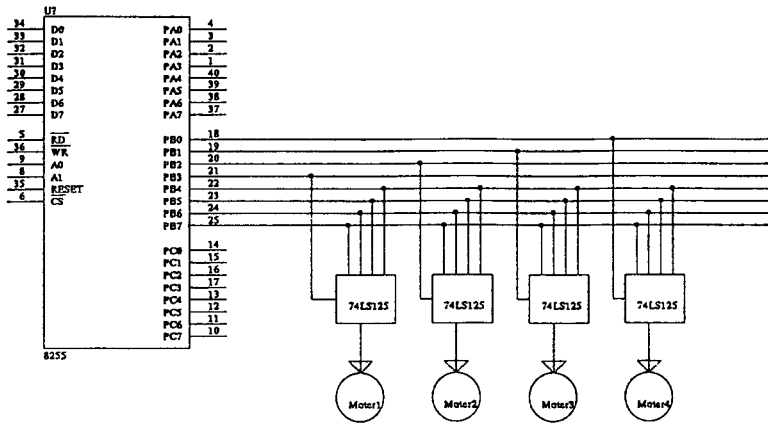
6.5.2 หลักการทำงานของวงจร

สแต็ปปีงมอเตอร์ที่ใช้เป็นแบบปรับค่าความต้านทานแม่เหล็กได้ (VR) ขนาด 4 เฟส ซึ่งจะต้องใช้ค่าคัปของสัญญาณทั้งหมด 4 สัญญาณในการขับให้มอเตอร์แต่ละตัวหมุนดังได้กล่าวมาแล้ว ในการเชื่อมต่อสแต็ปปีงมอเตอร์กับคอมพิวเตอร์ได้นำสัญญาณผ่านพอร์ตซีซึ่งอยู่ในรูปไอซีเบอร์ 8255 โดยใช้สัญญาณจากพอร์ต B จำนวน 8 บิต มาขับมอเตอร์ ในการทำงานจริงๆของมอเตอร์ทั้ง 4 ตัวจะต้องทำงานไปพร้อมๆกัน ดังนั้นจึงต้องต่อสัญญาณที่ได้มาจากพอร์ต B ทั้ง 8 บิต ให้กับมอเตอร์ทุกตัวโดยต่อมอเตอร์ขนานกันดังรูป



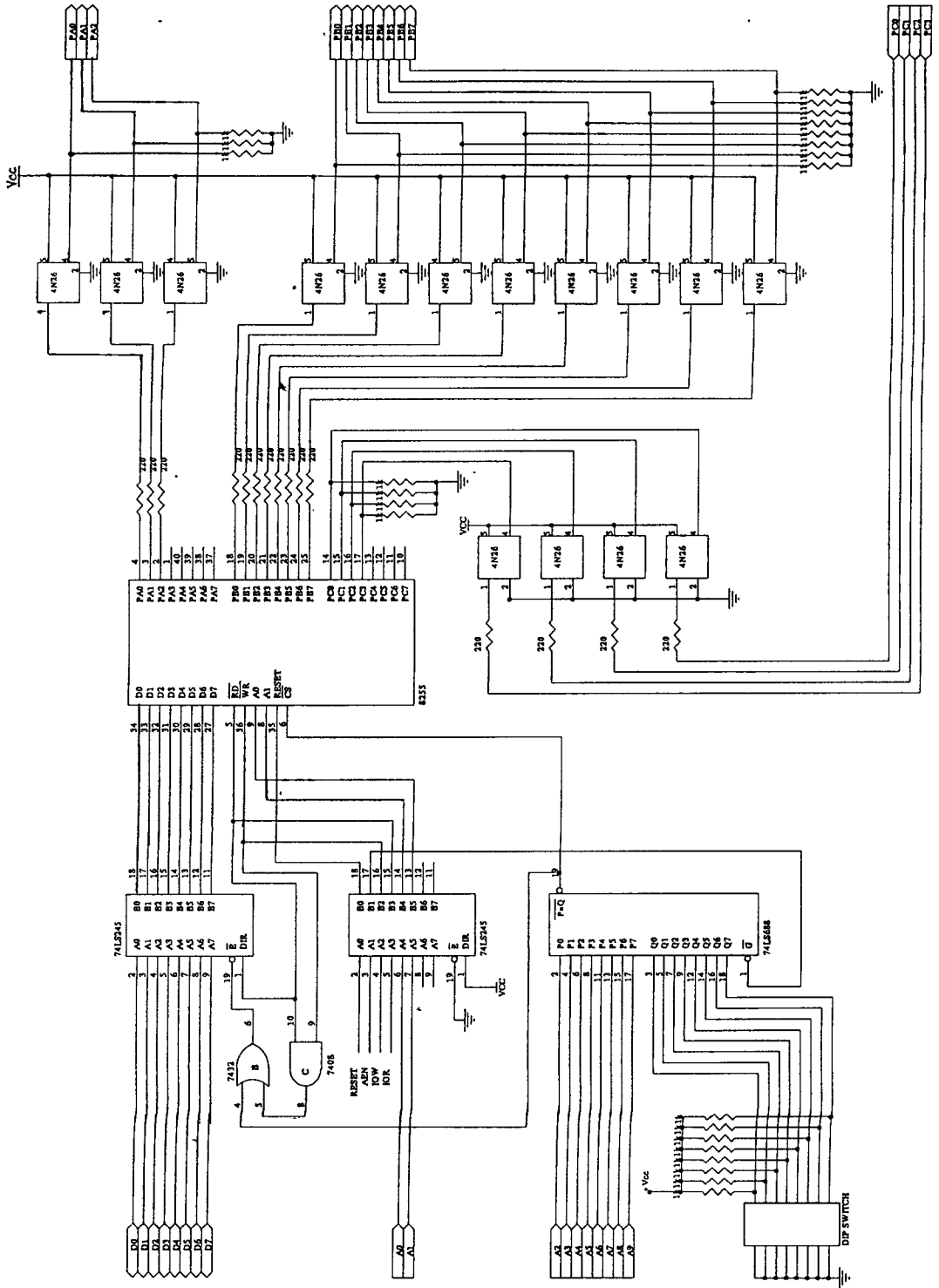
รูปที่ 6.12 แสดงการเชื่อมต่อมอเตอร์กับพอร์ต 8255

แต่ในความเป็นจริงมอเตอร์แต่ละตัวจะไม่หมุนในลักษณะเดียวกัน คืออาจจะหมุนในทิศทางตรงกันข้ามกันและด้วยค่ามุมในการหมุนที่แตกต่างกัน ดังนั้นสัญญาณที่จะนำมาควบคุมก็จะต้องมีความแตกต่างกัน การนำมอเตอร์มาต่อขนานกับพอร์ต B หมคทุกตัวนั้นจึงจำเป็นที่จะต้องมีส่วนควบคุมสัญญาณให้ผ่านไปขับมอเตอร์แต่ละตัวโดยไม่ขึ้นต่อกันผ่านพอร์ต B เพียงหนึ่งพอร์ต ไอซีเบอร์ 74LS125 ซึ่งเป็นวงจรมultiplexer แบบสามสถานะ (Tri-state Buffer) จะต่อขนานกับสายสัญญาณที่มาจากพอร์ต B ของไอซี 8255 แล้วสัญญาณก็จะส่งผ่านไปยังวงจรขับมอเตอร์แต่ละตัว โดยถ้าต้องการให้มอเตอร์ตัวไหนทำงานก็จะส่งสัญญาณไปอนุญาตให้ 74LS125 ผ่านสัญญาณควบคุมไปยังวงจรขับมอเตอร์ต่อไปดังรูป



รูปที่ 6.13 แสดงการควบคุมการหมุนของมอเตอร์ผ่านพอร์ต 8255

การควบคุมแบบนี้ นอกจากจะประหยัดพอร์ตสัญญาณที่จะต้องใช้แล้ว ยังสามารถทำให้มอเตอร์แต่ละตัวทำงานพร้อมกันได้โดยหลักการมัลติเพล็กซ์ทางเวลา (Time Multiplex) หลักการทำงาน คือ การส่งสัญญาณควบคุมของมอเตอร์แต่ละตัวสลับกันไปในช่วงเวลาที่กำหนด ดังนั้นมอเตอร์แต่ละตัวจะหมุนไปได้ค่าหนึ่งในช่วงเวลาที่กำหนดสลับกันไป และเนื่องจากความเร็วในการทำงานของคอมพิวเตอร์สูงมาก ดังนั้นเราจึงมองเห็นเหมือนกับว่ามอเตอร์ทุกตัวหมุนไปพร้อมๆ กัน



รูปที่ 6.14 แสดงวงจรอิเตอร์เฟสและวงจรควบคุมทั้งหมด

จากวงจรอินเทอร์เฟสในรูป สัญญาณข้อมูล(D0-D7),สัญญาณแอดเดรส(A0-A11) และสัญญาณควบคุมต่างๆ ได้แก่ IOR,IOW,AEN และRESET จะถูกส่งผ่านจากสล็อตของคอมพิวเตอร์ โดยสัญญาณข้อมูลจะผ่าน ไอซีเบอร์ 74LS245 ซึ่งเป็นบัฟเฟอร์ขนาด 8 บิต ส่งผ่านข้อมูลได้สองทิศทาง ขึ้นอยู่กับขา DIR สัญญาณ A0-A3,IOR,IOW,AEN และRESET ผ่านไอซีเบอร์ 74LS245 เช่นกัน แต่มีทิศทางการส่งผ่านทิศทางเดียวจากสล็อตคอมพิวเตอร์ไปยังวงจรส่วนอื่น สัญญาณแอดเดรส A4-A11 จะผ่านเข้าไอซีเบอร์ 74LS688 ทำหน้าที่เป็นตัวเปรียบเทียบค่าแอดเดรสระหว่างสัญญาณที่ส่งมาจากสล็อตคอมพิวเตอร์กับสัญญาณที่ดึงจากคิปสวิตช์ซึ่งได้ทำการกำหนดตำแหน่งของพอร์ตที่คอมพิวเตอร์ต้องการติดต่อ โดยถ้าตำแหน่งจากคิปสวิตช์เท่ากับตำแหน่งที่มาจากคอมพิวเตอร์จะทำให้เอาต์พุตขา 19 ของ 74LS688 เป็นลอจิก "0" แต่ไอซี 74LS688 จะทำงานได้ต้องได้รับสัญญาณ AEN จากไอซี 74LS245 ซึ่งจะเป็นลอจิก "0" เมื่อทำคำสั่ง OUT/IN ข้อมูลออกทางพอร์ตที่ได้กำหนดตำแหน่งไว้ เอาต์พุตขา 19 ของ 74LS688 จะเป็นตัวอินเวิลไอซี 8255 ซึ่งสามารถเลือกตำแหน่งของพอร์ตได้ โดยบิต A0-A1 จะเป็นตัวกำหนดพอร์ตของไอซี 8255 ซึ่งเขียนแอสแองคเคอเรสได้ดังนี้ (ในโปรเจกนี้ใช้ตำแหน่งพอร์ตแอสแองคเคอเรสที่ 200H)

A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	หมายเลขพอร์ต
1	0	0	0	0	0	0	0	0	0	200H พอร์ต A
1	0	0	0	0	0	0	0	0	1	201H พอร์ต B
1	0	0	0	0	0	0	0	1	0	202H พอร์ต C
1	0	0	0	0	0	0	0	1	1	203H พอร์ตควบคุม

ในการทำคำสั่ง IN/OUT ข้อมูลแต่ละครั้ง จะมีสัญญาณ IOR หรือ IOW ผ่านแอสแองคเคอเรส ซึ่งทำให้ได้เอาต์พุต ขา 3 เป็น 0 เสมอ ไปทำการออร์กับสัญญาณเอาต์พุตจากขา 19 ของไอซี 74LS688 ซึ่งในขณะนั้นเป็นลอจิก "0" ได้เอาต์พุตจากออร์เกทไปอินเวิล ไอซี 74LS245 ที่ส่งผ่านสัญญาณข้อมูล ทำให้มีข้อมูลอยู่บนสายนำสัญญาณ แต่สัญญาณข้อมูลนั้นจะผ่านไปยังไอซี 8255 หรือ จากไอซี 8255 ไปยังคาต้าบัสได้ ไอซี 8255 ต้องได้รับสัญญาณ RD,WR ที่ส่งมาจากชิพยูก่อน

วงจรอินเทอร์เฟสใช้ไอซี 8255 เพียงตัวเดียว โดยพอร์ต C ถ่าง(C0-C3)ทำหน้าที่เป็นพอร์ตอินพุต และพอร์ต B ทำหน้าที่เป็นพอร์ตเอาต์พุต(4 บิตถ่าง(B0-B3) เป็นบิตควบคุมสัญญาณต่างๆ และ 4 บิตบน(B4-B7) เป็นบิตที่จ่ายพัลส์ไปขับสตีปปีงมอเตอร์ทุกตัว

การขับสแต็ปปีงมอเตอร์ใช้ 4 บิตล่างของพอร์ต B เป็นตัวควบคุมวงจรไครฟ์ของสแต็ปปีงมอเตอร์(ใช้ลอจิก "0")แต่ละตัวและควบคุมการทำงานของคิซิมอเตอร์(ลอจิก "1")ขณะเจาะแผ่นปรีนซ์ ซึ่งสรุปการตั้งค่าของบิต B0-B3 ได้ดังนี้

	B0	B1	B2	B3
- ขับสแต็ปปีงมอเตอร์ M1(8V)	1	1	1	0
- ขับสแต็ปปีงมอเตอร์ M2(7.4V)	1	1	0	1
- ขับสแต็ปปีงมอเตอร์ M3(5V)	0	1	1	1
- ขับสแต็ปปีงมอเตอร์ M4(7.28V)	1	0	1	1
- ขับสแต็ปปีงมอเตอร์ M2 (7.4V) และ M3(5V)และ คิซิมอเตอร์	0	1	0	1
- ขับสแต็ปปีงมอเตอร์ M4(7.28V) และ คิซิมอเตอร์	1	1	1	1

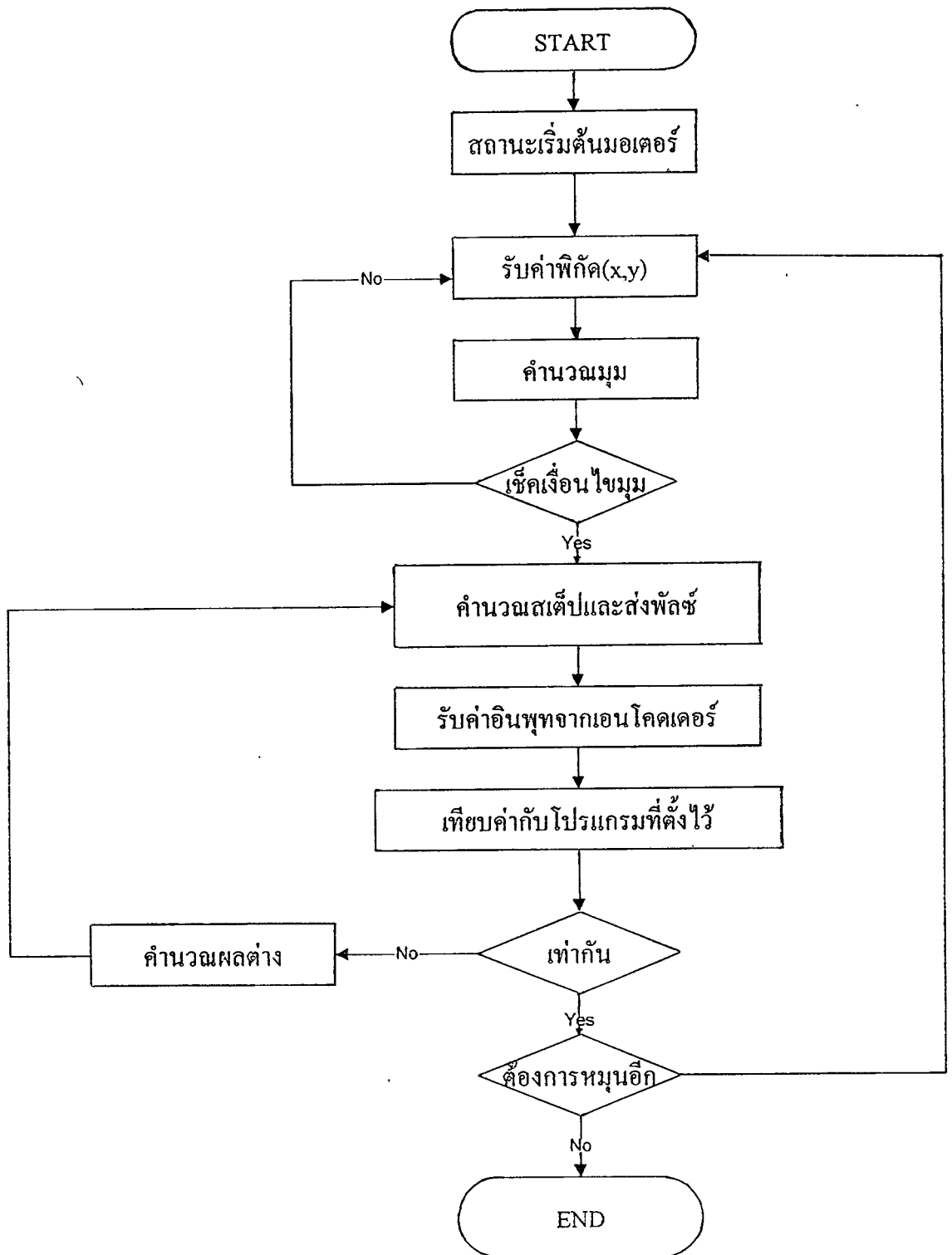
จากค่าของบิตควบคุมการขับสแต็ปปีงมอเตอร์สามารถหาค่าของสมการในการไครฟ์คิซิมอเตอร์ คือ

$$DC = B3 \bullet \overline{B2} \bullet \overline{B1} \bullet \overline{B0} + B3 \bullet \overline{B2} \bullet B1 \bullet B0$$

และได้สมการในการไครฟ์มอเตอร์ M4(7.28) คือ

$$M4 = B0 \bullet B2 \bullet B3$$

สำหรับการตั้งให้แกนกลเคลื่อนที่ ทำได้โดยการเขียนโปรแกรมภาษาซีส่งพัลซ์ไปจ่ายให้กับ สแต็ปปีงมอเตอร์ ครั้งละ 2 เฟส และรับอินพุทจากเอ็นโค้ดเดอร์ เข้ามาเปรียบเทียบการทำงานจริงของสแต็ปปีงมอเตอร์กับ โปรแกรมที่ควบคุมการทำงาน โดยโปรแกรมจะแสดงในหัวข้อต่อไป



รูปที่ 6.15 แสดงแผนภาพการทำงาน

6.6 อัลกอริทึมของโปรแกรมภาษาซี

ขั้นตอนการทำงานของโปรแกรมภาษาซีที่ใช้ในการขับสแต็ปปีงมอเตอร์ซึ่งตั้งให้สถานะของแกนกลอยู่ที่สถานะเริ่มต้น เริ่มจากคอมพิวเตอร์เข้าสู่โปรแกรมหลักเพื่อเลือกเมนูการทำงาน โดยสามารถเลือกไปยังฟังก์ชันที่ต้องการได้โดยการป้อนค่า (x,y) ซึ่งคอมพิวเตอร์จะทำการคำนวณหา มุมที่แกนแต่ละส่วนจะต้องเคลื่อนที่พร้อมกับระบุทิศทางของการเคลื่อนที่ด้วย โดยขอบเขตของมุม ต้องเป็นไปตามที่กำหนด หากไม่อยู่ในฟังก์ชันที่กำหนดให้เริ่มรับค่าฟังก์ชัน (x,y) อีกครั้ง

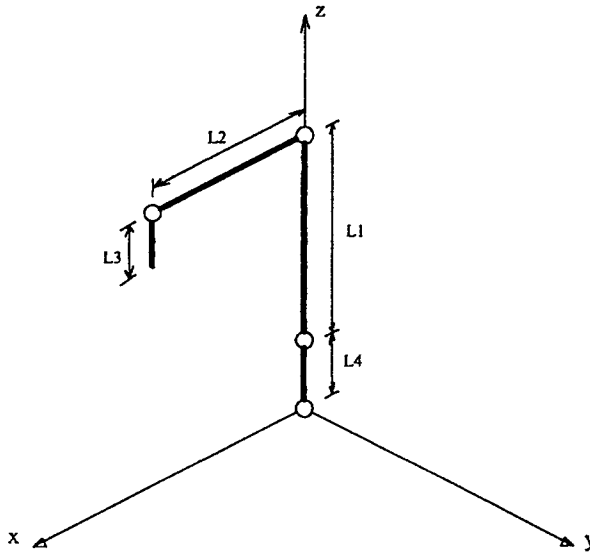
จากนั้นนำค่ามุมไปคำนวณเพื่อหาจำนวนสแต็ปที่ต้องส่งพัลซ์ไปขับมอเตอร์แต่ละตัว โดยเป็นการกระตุ้นครั้งละ 2 เฟส เมื่อส่งพัลซ์ครบตามจำนวนสแต็ปที่คำนวณได้แล้ว คอมพิวเตอร์จะ รอรับค่าเอาต์พุตจากวงจรเอ็นโค้ดเดอร์เข้าพอร์ต C เพื่อทำการเปรียบเทียบจำนวนสแต็ปที่ได้จาก การคำนวณกับจำนวนสแต็ปที่ได้จากการหมุนของสแต็ปปีงมอเตอร์จริงๆ จำนวนผลต่างของ จำนวนสแต็ป ถ้าเป็นศูนย์ให้เข้าสู่เมนูเพื่อเลือกให้แกนกลเคลื่อนที่ไปยังตำแหน่งอื่นหรือจะกลับสู่ สถานะเริ่มต้นอีกครั้งหรือจบการทำงาน

หากพบว่าผลต่างเป็นบวกให้ส่งพัลซ์จ่ายให้กับมอเตอร์ในทิศทางเดียวกับพัลซ์ที่จ่ายให้ มอเตอร์ แต่ถ้าผลต่างเป็นลบให้ส่งพัลซ์ในทิศทางตรงกันข้าม แล้วรับค่าเอาต์พุตจากวงจรเอ็นโค้ด เดอร์เข้ามาเปรียบเทียบอีกครั้ง ซึ่งถ้าเป็นศูนย์ก็จะกลับเข้าสู่เมนูอีก แต่ถ้ายังมีผลต่างเกิดขึ้นให้ส่ง พัลซ์ไปหมุนมอเตอร์แล้วรับค่าเข้ามาเปรียบเทียบจนกว่าจำนวนสแต็ปในการคำนวณจะเท่ากับ จำนวนสแต็ปที่มอเตอร์หมุนไป

6.7 การเคลื่อนที่ของแขนกลและการเขียนโปรแกรมควบคุม

การทำงานของแขนกลนั้นเราสามารถกำหนดพิกัดในระนาบ (x,y) แล้วทำการควบคุมให้แขนกลเคลื่อนที่ไปยังตำแหน่งนั้นๆ ได้ โดยทำการคำนวณหาค่ามุมและทิศทางที่มอเตอร์แต่ละตัวต้องหมุนไป ซึ่งมีวิธีในการคำนวณคือ

1. ณ ตำแหน่งเริ่มต้นลักษณะของแขนกลจะเป็นดังรูป

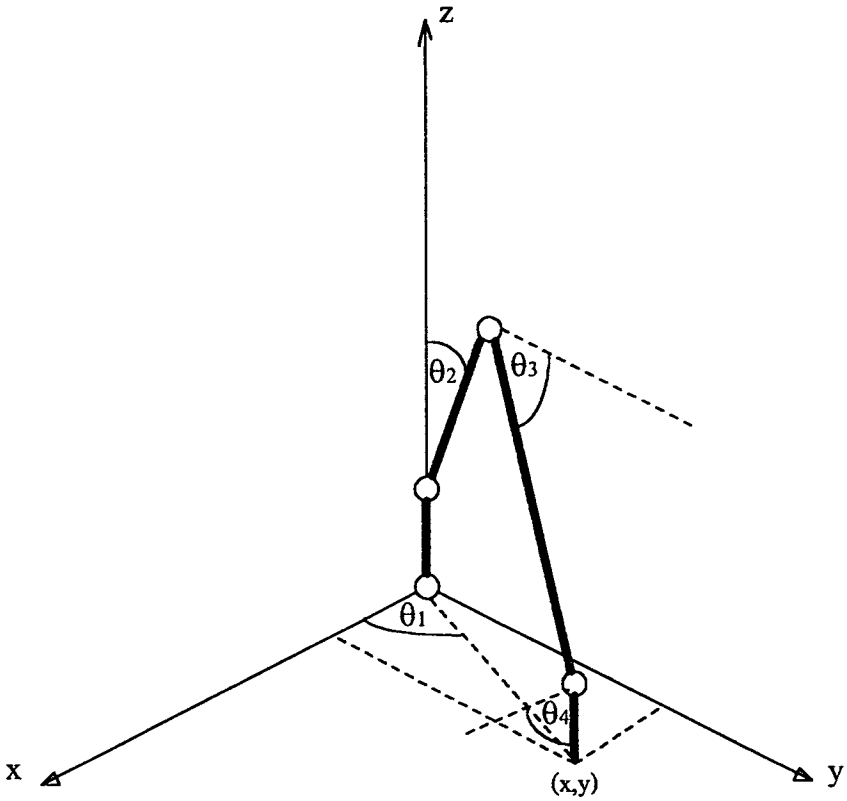


รูปที่ 6.16 แสดงลักษณะของแขนกลในสภาวะเริ่มต้น

เมื่อ	L1 คือ ความยาวของแขนท่อนบน	(เมตร)
	L2 คือ ความยาวของแขนท่อนล่าง	(เมตร)
	L3 คือ ความยาวของส่วนปลายแขน	(เมตร)
	L4 คือ ความสูงของส่วนฐาน	(เมตร)

โดยที่ส่วนฐานจะบังคับให้แขนกลอยู่ในพิกัด $(0,0)$ และส่วนแขนท่อนล่างตั้งท่อนบนจะทำมุมกัน 90° เช่นเดียวกันกับส่วนแขนท่อนบนที่ทำกับส่วนปลาย

2. เมื่อแขนกลเคลื่อนที่ไปยังพิกัด (x,y) ที่กำหนดแล้วจะมีลักษณะการวางตัวดังรูป

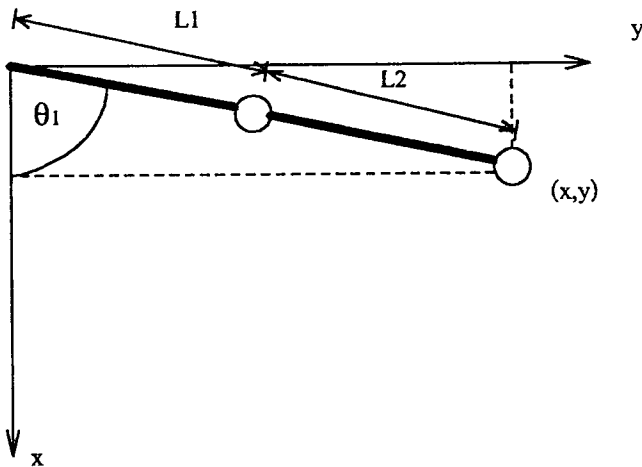


รูปที่ 6.17 แสดงแขนกลในขณะทำงาน

ดังนั้นจะสามารถหาค่ามุมที่มอเตอร์แต่ละตัวจะต้องเคลื่อนที่ไปได้คือ

ก) การหามุมที่มอเตอร์ส่วนฐาน (θ_1)

จากระนาบ (x,y) จะได้ลักษณะของแขนกลดังรูป



รูปที่ 6.18 แสดงแขนกลในขณะทำงานมองจากระนาบ (x,y)

เนื่องจาก $\tan \theta_1 = x/y$
 $\theta_1 = \arctan(x/y)$ เรเดียน

ทำการแปลงมุมจากค่าเรเดียนเป็นองศาโดย

$$\theta_1 = (\theta_1 \times 180^\circ) / \pi \quad \text{องศา}$$

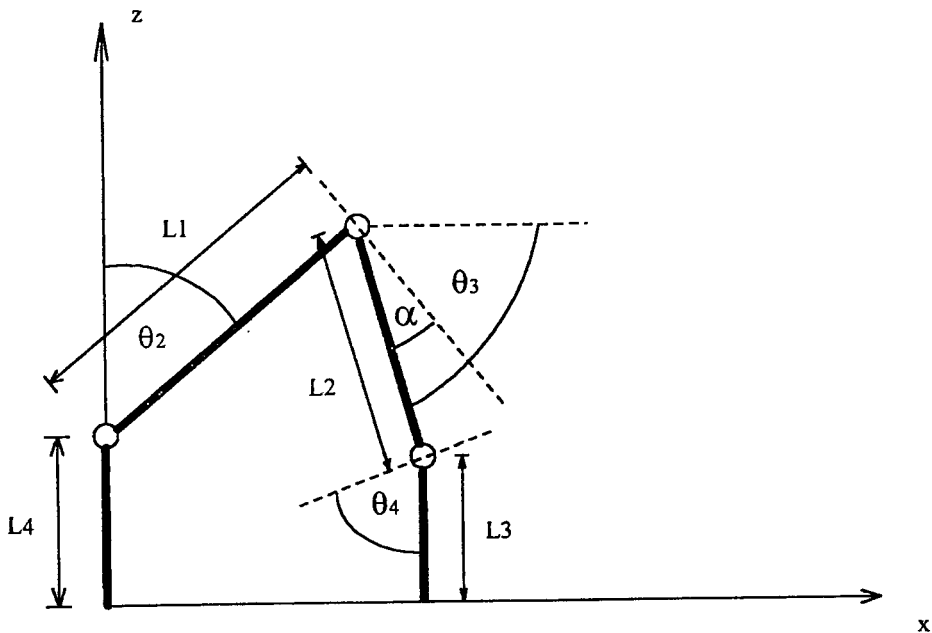
จากนั้นจึงนำค่ามุมที่ได้ไปคำนวณหาจำนวนพัลส์ (n_1) ที่จะต้องจ่ายให้กับมอเตอร์ส่วนฐานซึ่งมีค่ามุมในการหมุน $1.8^\circ / \text{สตีป}$ และมีอัตราทด T_1

จะได้

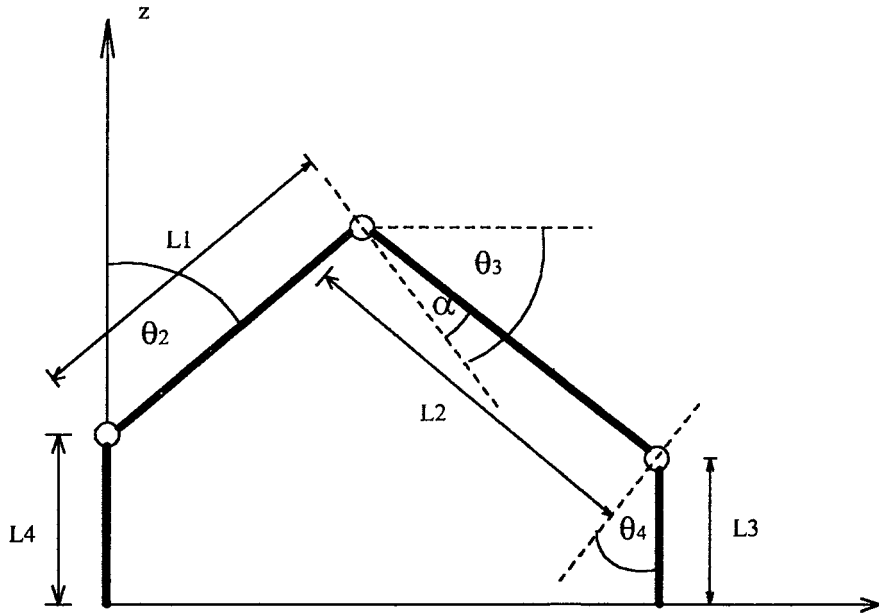
$$n_1 = (\theta_1 / 1.8^\circ) \times T_1$$

ข) การหามุมที่มอเตอร์ส่วนไหล่ (θ_2)

ณ ตำแหน่งพิกัด (x,y) จะสามารถมองเห็นแขนกลจากกระนาบ (x,z) ได้ดังรูป



ก) กรณีที่ $\theta_2 < \theta_3$



ข) กรณีที่ $\theta_2 > \theta_3$

รูปที่ 6.19 แสดงลักษณะของแขนกลในกรณีต่างๆ

จะได้

$$(L1)\sin(\theta_2) + (L2)\cos(\theta_3) = \sqrt{x^2 + y^2} \quad \rightarrow (1)$$

$$(L1)\cos(\theta_2) + L4 = L3 + (L2)\sin(\theta_3) \quad \rightarrow (2)$$

จาก (1) จะได้

$$(L1)\sin(\theta_2) = \{\sqrt{x^2 + y^2} - (L2)\cos(\theta_3)\}$$

ยกกำลังสองทั้งสองข้าง จะได้

$$(L1)^2 \sin^2(\theta_2) = x^2 + y^2 - 2(L2)\cos(\theta_3)(\sqrt{x^2 + y^2}) + (L2)^2 \cos^2(\theta_3) \quad \rightarrow (3)$$

และจาก (2) จะได้

$$(L1)\cos(\theta_2) = L3 + (L2)\sin(\theta_3) - L4$$

เนื่องจากค่า L_3 , L_4 เป็นค่าคงที่ ดังนั้นให้ $L_3 - L_4 = C$

ยกกำลังสองทั้งสองข้าง

$$(L_1)^2 \cos^2(\theta_2) = (C)^2 + 2(L_2)(C)\sin\theta_3 + (L_2)^2 \sin^2(\theta_3) \rightarrow (4)$$

นำ (3) + (4) จะได้

$$(L_1)^2 \{ \sin^2(\theta_2) + \cos^2(\theta_2) \} = x^2 + y^2 - 2(L_2)\cos(\theta_3)(\sqrt{x^2 + y^2}) + (L_2)^2 \cos^2(\theta_3) + (C)^2 + 2(L_2)(C)\sin\theta_3 + (L_2)^2 \sin^2(\theta_3)$$

และเนื่องจาก

$$\begin{aligned} \sin^2(\theta_2) + \cos^2(\theta_2) &= 1 \\ \sin^2(\theta_3) + \cos^2(\theta_3) &= 1 \end{aligned}$$

จะได้

$$(L_1)^2 = x^2 + y^2 - 2(L_2)\cos\theta_3(\sqrt{x^2 + y^2}) + (C)^2 + 2(L_2)(C)\sin\theta_3 + (L_2)^2 \rightarrow (5)$$

เนื่องจาก $\sin^2(\theta_3) + \cos^2(\theta_3) = 1$

จะได้ $\sin^2(\theta_3) = \sqrt{1 - \cos^2(\theta_3)}$

แทนค่าในสมการ (5) แล้วจัดรูป

จะได้

$$\begin{aligned} 2(L_2)(C)(\sqrt{1 - \cos^2(\theta_3)}) &= M + 2(L_2)\cos\theta_3(\sqrt{x^2 + y^2}) \\ \text{เมื่อ } M &= (L_1)^2 - (x^2 + y^2) - (C)^2 - (L_2)^2 \end{aligned}$$

ยกกำลังสองทั้งสองข้าง จะได้

$$\begin{aligned} 4(L_2)(C)\{1 - \cos^2(\theta_3)\} &= M + 4(L_2)\cos(\theta_3)(\sqrt{x^2 + y^2}) + \\ &4(L_2)\cos(\theta_3)(x^2 + y^2) \end{aligned}$$

จากสมการกำลังสองในรูป $aX^2 + bX + C$

จะได้ $X = \{-b \pm \sqrt{b^2 - 4ac}\} / 2a$

แก้สมการหาค่า θ_3

แต่เนื่องจากในการสภาวะปกตินั้นแกนท่อนล่างจะวางตัวตั้งฉากกับแกนท่อนบน ดังนั้นแกนกลจะเคลื่อนที่ได้เป็นมุม α เท่านั้นในแต่ละค่ามุม θ_3 ดังนั้นสามารถคำนวณหามุม α ได้ดังนี้

- กรณีที่ $\theta_3 > \theta_2$ จะได้ α เท่ากับ $\theta_3 - \theta_2$

- กรณีที่ $\theta_3 < \theta_2$ จะได้ α เท่ากับ $\theta_2 - \theta_3$

ทำการแปลงมุมจากค่าเรเดียนเป็นองศาโดย

$$\alpha = (\alpha \times 180^\circ) / \pi \quad \text{องศา}$$

จากนั้นจึงนำค่ามุมที่ได้ไปคำนวณหาจำนวนพัลส์ (n_3) ที่จะต้องจ่ายให้กับมอเตอร์ส่วนข้อศอก ซึ่งมีค่ามุมในการหมุน $1.8^\circ / \text{สเต็ป}$ และมีอัตราทด T_3 จะได้ว่า

$$n_3 = (\alpha / 1.8^\circ) \times T_3$$

ค) การหาค่ามุมที่มอเตอร์ส่วนไหล่ (θ_2)

โดยการแทนค่ามุม θ_3 ลงในสมการที่ (1) หรือ (2)

ทำการแปลงมุมจากค่าเรเดียนเป็นองศาโดย

$$\theta_2 = (\theta_2 \times 180^\circ) / \pi \quad \text{องศา}$$

จากนั้นจึงนำค่ามุมที่ได้ไปคำนวณหาจำนวนพัลส์ (n_2) ที่จะต้องจ่ายให้กับมอเตอร์ส่วนข้อศอก ซึ่งมีค่ามุมในการหมุน $2^\circ / \text{สเต็ป}$ และมีอัตราทด T_2 จะได้ว่า

$$n_2 = (\theta_2 / 2^\circ) \times T_2$$

ง) การหาค่ามุมมอเตอร์ส่วนปลาย (θ_4)

เนื่องจากแขนส่วนปลายจะต้องตั้งฉากกับระนาบที่จะเจาะเสมอ ดังนั้นจะได้

$$\theta_3 = \theta_4$$

ทำการแปลงมุมจากค่าเรเดียนเป็นองศาโดย

$$\theta_4 = (\theta_4 \times 180^\circ) / \pi \quad \text{องศา}$$

จากนั้นจึงนำค่ามุมที่ได้ไปคำนวณหาจำนวนพัลส์ (n_4) ที่จะต้องจ่ายให้กับมอเตอร์ส่วนข้อศอก ซึ่งมีค่ามุมในการหมุน $8.2^\circ / \text{สเต็ป}$ และมีอัตราทด T_4 จะได้ว่า

$$n_4 = (\theta_4 / 8.2^\circ) \times T_4$$

เมื่อได้จำนวนพัลส์ของมอเตอร์ต่างไปแล้วจึงนำไปเขียนโปรแกรมควบคุมการทำงานต่อไป โดยใช้ภาษาซี

```

#include<stdio.h>
#include<dos.h>
#include<math.h>
#define port_A 0x200
#define port_B 0x201
#define port_C 0x202
#define CONP 0x203
int  deg12,deg1,deg2,deg3,deg4,a,b,c,d,step1,step2,step3,step4,
    deg11,deg21,deg31,deg41,o1,o2,o3,o4,state ;
float x,y,l1,l2,l3,l4;
cordinate(float x,float y,float l1, float l2,float l3,float l4);
calstep(int deg1,int deg2,int deg3,int deg4);
initial(int deg11,int deg21,int deg31,int deg41);
step(int step1,int step2,int step3,int step4,int a,int b,int c,int d);
encode(int step1,int step2,int step3,int step4);
main()
{
    float x,y,l1,l2,l3,l4;
    int num;
    clrscr();
    /*initial();*/
start: printf("Please select number(1-3)\n");
    printf(" 1.Goto initial state\n");
    printf(" 2.Goto cordinate(x,y)\n");
    printf(" 3.Goto an angle\n");
    printf(" 4.End program\n");
    printf("select number:");
    scanf("%d",&num);

```

```
switch(num)
```

```
{ case 1:
```

```
    initial(deg11,deg21,deg31,deg41);
```

```
    goto start;
```

```
case 2:
```

```
begin: printf("Please Enter value(x,y):\n");
```

```
    scanf("%f",&x);
```

```
    scanf("%f",&y);
```

```
    l1=19.7;
```

```
    l2=16.7;
```

```
    l3=14;
```

```
    l4=7;
```

```
    cordinate(x,y,l1,l2,l3,l4);
```

```
    if(deg12==0)
```

```
        goto begin;
```

```
    else
```

```
    {
```

```
        /*printf("deg1=%d\t",deg1);*/
```

```
        printf(" deg2=%d\t",deg2);
```

```
        printf(" deg3=%d\t",deg3);
```

```
        printf(" deg4=%d\n",deg4);
```

```
        /*printf("deg11=%d\t",deg11);
```

```
        printf(" deg21=%d\t",deg21);
```

```
        printf(" deg31=%d\t",deg31);
```

```
        printf(" deg41=%d\n",deg41);*/
```

```
        calstep(deg1,deg2,deg3,deg4);
```

```
        step(step1,step2,step3,step4.a,b,c,d);
```

```

        /*encode(step1,step2,step3,step4);
        drill(); */
        goto start;
    }
case 3:
    printf("Enter degree value for motor 1-motor 4\n");
    printf("Enter degree value for motor 1:");scanf("%d",&deg1);
    printf("Enter degree value for motor 2:");scanf("%d",&deg2);
    printf("Enter degree value for motor 3:");scanf("%d",&deg3);
    printf("Enter degree value for motor 4:");scanf("%d",&deg4);

    calstep(deg1,deg2,deg3,deg4);
    /*stepo1=deg1;
    stepo2=deg2;
    stepo3=deg3;
    stepo4=deg4;*/
    step(step1,step2,step3,step4,a,b,c,d);
    goto start;

default:
    exit(0);
}

}

initial( deg11,deg21,deg31,deg41)
{
    int state,s,t,u,v,c1,c2,c3,c4,num,b1,b2,b3,b4,i,j,k,l;
    i=1;
    j=1;

```

```
k=1;
l=1;
input: state=inp(port_C);
if(state!=15)
{
    c1=state%2;
    state=state/2;
    if(state==0)
    {
        c2=0;c3=0;c4=0;}
    else
        c2=state%2;
    state=state/2;
    if(state==0)
    {
        c3=0;c4=0;}
    else
        c3=state%2;
    state=state/2;
    if(state==0)
        c4=0;
    else
        c4=state%2;
    if(deg11<0)
        s=0;
    else
        s=1;
/*
    if(deg21<0)
        t=1;
    else
        t=0;
    if(deg31<0)
```

```

        u=1;
else
        u=0;
if(deg41<0)
        v=1;
else
        v=0;
printf(" s=%d\n",s);
printf(" t=%d\n",t);
printf(" u=%d\n",u);
printf(" v=%d\n",v); */

motor1: while(i&&4 !=1)
{
    if(deg11<=0)
        s=0;
    else
        s=1;
    num = i%4;
    switch(num)
    {
        case 0:
            b1=s?0x37:0x97;break;
        case 1:
            b1=s?0x67:0xc7;break;
        case 2:
            b1=s?0xc7:0x67;break;
        case 3:
            b1=s?0x97:0x37;break;
        default:
            exit(0);
    }
}

```

```

    }
    i++;
    outp(port_B,b1);delay(10);
    goto motor2;
}
motor2: while(j&& c3 !=1)
    {    if(deg21<0)
            t=1;
        else
            t=0;
        num = j%4;
        switch(num)
        {    case 0:
                b2=t?0x3b:0x9b;break;
            case 1:
                b2=t?0x6b:0xcb;break;
            case 2:
                b2=t?0xcb:0x6b;break;
            case 3:
                b2=t?0x9b:0x3b;break;
            default:
                exit(0);
        }
        j++;
        outp(port_B,b2);delay(10);
        goto motor3;
    }
motor3: while(k&& c2 !=1)
    {    if(deg31<0)

```

```

        u=0;
else
        u=1;
num = k%4;
switch(num)
{   case 0:
            b3=u?0x3e:0x9e;break;
        case 1:
            b3=u?0x6e:0xce;break;
        case 2:
            b3=u?0xce:0x6e;break;
        case 3:
            b3=u?0x9e:0x3e;break;
        default:
            exit(0);
    }
    k++;
    outp(port_B,b3); delay(10);
    goto motor4;
}
motor4: while(l&&c1!=1)
{   if(deg41<0)
        v=0;
    else
        v=1;
    num = l%4;
    switch(num)
    {   case 0:
            b4=v?0x3d:0x9d;break;

```

```

        case 1:
            b4=v?0x6d:0xcd;break;
        case 2:
            b4=v?0xcd:0x6d;break;
        case 3:
            b4=v?0x9d:0x3d;break;
        default:
            exit(0);
    }
    l++;
    outp(port_B,b4);delay(50);
    goto input;
}
goto input;
}
else
    printf(" ");
}

```

```

coordinate(float x,float y,float l1,float l2,float l3,float l4)
{ float
a,a1,a2,b,b1,b2,b3,b4,b5,b6,b32,c,deg1a,deg2a1,deg2a2,d1,d2,deg3a,deg3b:
    int deg2a,deg2b,deg3ab,deg4a,deg4b ;
    state=inp(port_C);
    deg12=1;
degr: deg1a=(-1)*(180/3.141592654)*atan(x/y);
    deg1=deg1a;
    if(state==15)

```

```

    deg11=0;
else
    deg11=deg11;
deg1=deg1a-deg11;-
deg11=deg1a;
a1=x*x+y*y;
a2=sqrt(a1);
a=2*I1*a2;
b=2*I1*(I3-I4);
c=(x*x)+(y*y)+((I3-I4)*(I3-I4))+(I1*I1)-(I2*I2);
b1=(4*a*a*c*c)-(4*((a*a)+(b*b))*((c*c)-(b*b)));
if(b1<0)
    deg12=0;
else
{
    b2=sqrt(b1);
    b3=(2*a*c)+b2;
    b32=2*((a*a)+(b*b));
    if(b32==0)
        deg12=0;
    else
    {
        b4=b3/(2*((a*a)+(b*b)));
        b5=(2*a*c)-b2;
        b6=b5/(2*((a*a)+(b*b)));
        deg2a1=(-1)*(180/3.141592654)*asin(b4);
        deg2a2=(-1)*(180/3.141592654)*asin(b6);
        deg2a=deg2a1;
        deg2b=deg2a2;
    }
}

```

```
if((deg2a<=0 && deg2a>=-45)&&(deg2a>=deg2b))
{   deg2=deg2a;
    if(state==15)
        deg21=0;
    else
        deg21=deg21;
    deg2=deg2a-deg21;
    deg21=deg2a;
    d1=(a2-(11*(1)*b4))/l2;
    deg3a=(-1)*(180/3.141592654)*acos(d1);
    deg4a=(-1)*deg3a;
    if(state==15)
        deg41=0;
    else
        deg41=deg41;
    deg4=deg4a-deg41;
    deg41=deg4a;
    deg3ab=deg3a-deg2a;
    if(state==15)
        deg31=0;
    else
        deg31=deg31;
    deg3=deg3ab-deg31;
    deg31=deg3ab;
}
else
{   deg2=deg2b;
    if(state==15)
```

```
calstep(deg1,deg2,deg3,deg4)
{   int num;
    printf(" deg1=%d\n",deg1);
    if(deg1<0)
    {   deg1=(-1)*deg1;
        a=1;
    }
    else
        a=0;
    if(deg2<0)
    {   deg2=(-1)*deg2;
        b=0;
    }
    else
        b=1 ;
    if(deg3<0)
    {   deg3=(-1)*deg3;
        c=1 ;
    }
    else
        c=0;
    if(deg4<0)
    {   deg4=(-1)*deg4;
        d=1 ;
    }
    else
        d=0 ;
    step1=(deg1*138)/1.8;
```

```

step2=(deg2*140)/2;
step3=(deg3*160)/1.8;
step4=(deg4*28)/6;
}
step(int step1,int step2,int step3,int step4,int a,int b,int c,int d)
{
    int a1,b1,c1,d1,i,j,k,l,num;
    i=0;
    j=0;
    k=0;
    l=0;
motor1: while(i!=step1)
    {
        num=i%4;
        switch(num)
        {
            case 0:
                a1=a?0x37:0x97;break;
            case 1:
                a1=a?0x67:0xc7;break;
            case 2:
                a1=a?0xc7:0x67;break;
            case 3:
                a1=a?0x97:0x37;break;
            default:
                exit(0);
        }
        i++;
        outp(port_B,a1 );
        delay(10);
        goto motor2;
    }
}

```

```
motor2: while(j!=step2)
    {   num=j%4;
        switch(num)
        {   case 0:
                b1=b?0x3b:0x9b;break;
            case 1:
                b1=b?0x6b:0xcb;break;
            case 2:
                b1=b?0xcb:0x6b;break;
            case 3:
                b1=b?0x9b:0x3b;break;
            default:
                exit(0);
        }
        j++;
        outp(port_B,b1);
        delay(10);

        goto motor3;
    }

motor3: while(k!=step3)
    {   num=k%4;
        switch(num)
        {   case 0:
                c1=c?0x3e:0x9e;break;
            case 1:
                c1=c?0x6e:0xce;break;
            case 2:
                c1=c?0xce:0x6e;break;
```

```

        case 3:
            c1=c?0x9e:0x3e;break;
        default:
            exit(0);
    }
    k++;
    outp(port_B,c1);
    delay(8);
    goto motor4;
}
motor4: while(l!=step4)
    {    num=l%4;
        switch(num)
        {    case 0:
            d1=d?0x3d:0x9d;break;
            case 1:
            d1=d?0x6d:0xcd;break;
            case 2:
            d1=d?0xcd:0x6d;break;
            case 3:
            d1=d?0x9d:0x3d;break;
            default:
            exit(0);
        }
        l++;
        outp(port_B,d1);
        delay(50);
        goto motor1;
    }

```

```

if((i==step1)&&(j==step2)&&(k==step3)&&(l==step4)==1)
{   outp(port_B,0x00);
    delay(10);
}
else
    goto motor1;
}

```

drill()

```

{   int step,stepj,stepk,stepl,j,k,l,num,b,c,d,b1,c1,d1;
    printf("Enter step for drilling:");
    scanf("%d",&step);
    stepj=step;
    stepk=step;
    stepl=step;
    b=0;
    c=0;
    d=1;
    j=0;
    k=0;
    l=0;
    motor2: while(j!=stepj)
        {   num=j%4;
            switch(num)
            {   case 0:
                    b1=b?0x3d:0x9d;break;
                case 1:

```

```
        b1=b?0x6d:0xcd;break;
    case 2:
        b1=b?0xcd:0x6d;break;
    case 3:
        b1=b?0x9d:0x3d;break;
    default:
        exit(0);
}
j++;
printf("out m2:%x\t",b1);
goto motor3;
}
motor3: while(k!=stepk)
{   num=k%4;
    switch(num)
    {   case 0:
        c1=c?0x3b:0x9b;break;
        case 1:
            c1=c?0x6b:0xcb;break;
        case 2:
            c1=c?0xcb:0x6b;break;
        case 3:
            c1=c?0x9b:0x3b;break;
        default:
            exit(0);
    }
    k++;
    printf("out m3:%x\t",c1);
    goto motor4;
```

```

    }
motor4: while(l!=stepl)
    {    num=l%4;
        switch(num)
        {    case 0:
                d1=d?0x37:0x97;break;
            case 1:
                d1=d?0x67:0xc7;break;
            case 2:
                d1=d?0xc7:0x67;break;
            case 3:
                d1=d?0x97:0x37;break;
            default:
                exit(0);
        }
        l++;
        printf("out m4:%x\t",d1);
        goto motor2;
    }
if((j==stepj)&&(k==stepk)&&(l==stepl)==1)
{    printf("    This is OK.for drilling\n");
    if(b==0)
    {    j=0;
        k=0;
        l=0;
        b=1;
        c=1;
        d=0;
        goto motor2;
    }
}

```

```

        }
        else
            printf(" out drill from PCB\n");
    }
    else
        goto motor2;
}

```

```

encode(int step1,int step2,int step3,int step4 )

```

```

{   int stepi,stepj,stepk,stepl;
    printf(" out 0x02      ");
    printf("step1 =");scanf("%d",&stepi);
    printf(" out 0x03      ");
    printf("step2 =");scanf("%d",&stepj);
    printf(" out 0x04      ");
    printf("step3 =");scanf("%d",&stepk);
    printf(" out 0x05      ");
    printf("step4 =");scanf("%d",&stepl);
    step1=step1-stepi;
    if(step1>0)
        a=0;
    else
    {   a=1;
        step1=(-1)*step1;
    }
    step2=step2-stepj;
    if(step2>0)
        b=0;
    else

```

```
{  b=1;
    step2=(-1)*step2;
}
step3=step3-stepk;
if(step3>0)
    c=0;
else
{  c=1;
    step3=(-1)*step3;
}
step4=step4-stepl;
if(step4>0)
    d=0;
else
{  d=1;
    step4=(-1)*step4;
}
step(step1,step2,step3,step4,a,b,c,d) ;
}
```

□

บทที่ 7

ผลการทดลองและสรุปผลการทดลอง

7.1 ผลการทดลอง

จากการป้อนข้อมูลจำนวน 9 ค่า โดยผ่านฟังก์ชันในการคำนวณ ผลที่ได้ปรากฏดังนี้

ตำแหน่งที่ต้องการ	ตำแหน่งจริงที่ได้ (X,Y)			เฉลี่ย
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	
(X,Y)				(X,Y)
(-10,20)	(-10,20.5)	(-10,19.5)	(-10,19.5)	(-10,19.8)
(-8,24)	(-7.6,24)	(-8,24)	(-8,24)	(-7.9,24)
(-3,22)	(-2.7,22)	(-3,21.5)	(-3,22)	(-2.9,21.8)
(-3,20)	(-3.7,19.5)	(-3.5,20)	(-3.5,19.5)	(-3.6,19.7)
(-1,25)	(-1,24.5)	(-1,25)	(-1,24.5)	(-1,24.7)
(0,24)	(0,23.5)	(0,23.5)	(0,23.5)	(0,23.5)
(5,20)	(4,19)	(4,19)	(4,19)	(4,19)
(5,23)	(3.7,23)	(4.5,23.5)	(4,23)	(4.1,23.2)
(10,25)	(9,26)	(9.5,25.5)	(9.5,25.5)	(9.3,25.7)

รูปที่ 7.1 ตารางผลการทดลอง

7.2 สรุปผลการทดลอง

จากการทดลองพบว่าความแม่นยำในการเจาะยังไม่สมบูรณ์ ทั้งนี้เนื่องจากปัญหาต่างดังนี้

1. ความผิดพลาดทางด้านโครงสร้าง

- ปัญหาที่เกิดขึ้น เช่น ความละเอียดของมุมที่จะต้องหมุนไป มีค่าผิดพลาดจากการที่ระยะห่างของฟันเฟืองอันที่ควรต้องสแกนมีขนาดไม่พอดีกัน จำนวนฟันเฟืองที่หาได้ยังไม่ละเอียดเท่าที่ต้องการ เพราะการที่จะต้องหาซื้อเฟืองที่มีคุณลักษณะที่ต้องการนั้นหาได้ยากมาก นอกจากนี้ความโค้งตัวของโครงสร้างที่ใช้ทำแกนเนื่องจากการรับโหลดมีผลทำให้ได้ความคลาดเคลื่อนเช่นกัน อีกทั้งการตั้งองศาเริ่มแรกของท่อนแกนแต่ละส่วนในสภาวะเริ่มต้นมักมีค่าไม่เที่ยงตรง ส่งผลให้เมื่อโปรแกรมคำนวณค่ามุมแต่ละมุมออกมา ซึ่งเป็นมุมที่แกนแต่ละท่อนจะต้องหมุนไปจาก

สภาวะเริ่มต้น เมื่อสภาวะเริ่มต้นไม่เที่ยงตรง มุมที่หมุนไปก็ไม่ถูกต้อง การเจาะก็จะไม่ได้ตำแหน่งที่ต้องการจริงๆ ซึ่งควรจะมีวิธีที่ทำให้แต่ละมุมของข้อต่อมีค่าเริ่มแรกเท่ากับมุมเริ่มแรกของแต่ละข้อต่อที่กำหนดในโปรแกรมได้อย่างพอดีโดยไม่ต้องเพิ่มการปรับมุมด้วยสายตา

- โครงสร้างยังไม่มีความแข็งแรงพอ ทำให้บางส่วนเกิดการเอียงไปบ้างค่าคำนวณจึงผิดพลาด

2. ความผิดพลาดในส่วนของโปรแกรม

- ในการคำนวณจะต้องส่งค่าความยาวของท่อนแขนแต่ละท่อน (L1, L2, L3, L4) ไปคำนวณเพื่อหาองศาที่แต่ละท่อนแขนนั้นจะต้องหมุนไป ค่าองศาที่ได้ถ้าหากเป็นจุดทศนิยมจะทำการตัดทิ้งทั้งหมด ทำให้มุมคำนวณที่ได้มักเกิดความผิดพลาด

3. ความผิดพลาดในส่วนของวงจรจ่ายแรงดันคงที่

- จากการคำนวณค่ากระแสและแรงดัน ที่จ่ายให้แก่มอเตอร์แต่ละตัว มีค่าเพียงพอที่จ่ายได้ แต่ในการทดลองปรากฏว่า มอเตอร์แต่ละตัวต้องขับโหลดมาก ทำให้มีการหมุนช้า ทำให้โครงสร้างเคลื่อนที่ไปยังตำแหน่งที่เจาะได้ช้า และในบางสภาวะเมื่อมีการยกตัวของโครงสร้างบางลักษณะอาจทำให้จ่ายกระแสได้ไม่พอ ซึ่งส่วนนี้ได้ทำการปรับปรุงไว้แล้ว ในส่วนของน้ำหนักโครงสร้างหากมีการเปลี่ยนอลูมิเนียมเป็นโลหะอื่นที่มีน้ำหนักเบากว่าอาจทำให้การดึงกระแสน้อยลง ส่งผลให้มอเตอร์แต่ละตัวหมุนได้เร็วขึ้น และเคลื่อนที่ไปยังตำแหน่งที่ต้องการได้เร็วขึ้น

7.3 แนวทางแก้ไข

1. ในส่วนของโครงสร้าง การเปลี่ยนเฟืองให้มีจำนวนฟันที่ละเอียดพอ และเฟืองที่ต้องสับกัน ให้มีระยะห่างของฟันใกล้เคียงกันมากขึ้นทำให้ค่าที่หมุนไปละเอียดขึ้น และส่วนโครงสร้างการใช้วัสดุและออกแบบโครงสร้างที่มีน้ำหนักเบาและแข็งแรง จะช่วยให้โครงสร้างมีเสถียรภาพมากขึ้น

2. ในส่วนของโปรแกรม อาจทำการเพิ่มวงจรเซ็นเตอร์ให้แก่มอเตอร์แต่ละตัว เพื่อให้องศาของการหมุนที่แท้จริง มีค่าเท่ากับองศาที่คำนวณได้จริง โดยใช้โปรแกรมรับค่าจากเซ็นเตอร์เข้ามาเปรียบเทียบ

3. ในส่วนของเครื่องจ่ายแรงดันคงที่ เนื่องจากการคำนวณสามารถจ่ายกระแสให้แก่มอเตอร์ได้เพียงพอ แต่เนื่องจากการที่โครงสร้างรับโหลดมากเกินไปในสภาวะยกตัว การปรับปรุงโครงสร้างให้มีขนาดเบาขึ้นและสมดุลในแกนต่างๆ จะช่วยให้ลดภาระในการดึงกระแสจากแหล่งจ่ายให้ดีขึ้น

תוכן

54245/74245 Octal Bus Transceivers with 3-state Outputs

	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			
		C	P	MCF		C	P	MCF		C	P	MCF		C	P	MCF	
T. I.																	
FAIRCHILD																	
MOTOROLA																	
N. S. C.																	
PHILIPS																	
SIGNETICS																	
SIEMENS																	
FUJITSU																	
HITACHI																	
MITSUBISHI																	
NEC																	
TOSHIBA																	

Electrical Characteristics SN54LS245/SN74LS245

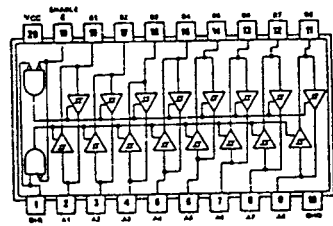
absolute maximum ratings over operating free-air temperature range						
Supply voltage, VCC	7V	Operating free-air temperature range	SN54LS	-55°C to 125°C		
Input voltage	7V	temperature range	SN74LS	0°C to 70°C		
		Storage temperature range		-65°C to 150°C		
recommended operating conditions						
	SN54LS245		SN74LS245		UNIT	
	MIN	NOM	MAX	MIN	NOM	MAX
Supply voltage, VCC	4.5	5	5.5	4.75	5	5.25
High-level output current, IOH			-12			-15
Low-level output current, IOL			12			24
Operating free-air temperature, TA	-55		125	0		70

electrical characteristics over recommended operating free-air temperature range

PARAMETER	TEST CONDITIONS †	SN74LS245		UNIT
		MIN	TYP ‡ MAX	
V _{IH} High-level input voltage		2		V
V _{IL} Low-level input voltage		0.8		V
V _{IK} Input clamp voltage	VCC = MIN, I _I = -18mA	-1.5		V
	Hysteresis (V _{T+} - V _{T-}) A or B input	0.2 0.4		V
V _{OH} High-level output voltage	VCC = MIN, V _{IH} = 2V, V _{IL} = V _{ILmax}	I _{OH} = -3mA	2.4 3.4	V
		I _{OH} = MAX	2	
V _{OL} Low-level output voltage	VCC = MIN, V _{IH} = 2V, V _{IL} = V _{ILmax}	I _{OL} = 12mA	0.4	V
		I _{OL} = 24mA	0.5	
I _{OZH} Off-state output current, high-level voltage applied	VCC = MAX, \bar{O} at 2V	V _O = 2.7 V		10
I _{OZL} Off-state output current, low-level voltage applied		V _O = 0.4V		-200
I _I Input current at maximum input voltage	A or B DIR or \bar{O} VCC = MAX, V _I = 5.5V			0.1
		V _I = 7V		0.1
I _{IH} High-level input current	VCC = MAX, V _{IH} = 2.7V			20
I _{IL} Low-level input current	VCC = MAX, V _{IL} = 0.4V			-0.2
I _{OS} Short-circuit output current	VCC = MAX			-40 -225
I _{CC} Supply current	VCC = MAX, Outputs open	Total, outputs high		48 70
		Total, outputs low		62 90
		Outputs at Hi-Z		64 95

switching characteristics, VCC 5V, TA 25°C

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t _{PLH} Propagation delay time, low-to-high-level output	C _L = 45pF, R _L = 667Ω, See Note 2			8 12	ns
t _{PdL} Propagation delay time, high-to-low-level output				8 12	ns
t _{PZL} Output enable time to low level	C _L = 50F, R _L = 667Ω, See Note 2			27 40	ns
t _{PZL} Output enable time to high level				25 40	ns
t _{PLZ} Output disable time from low level	C _L = 50F, R _L = 667Ω, See Note 2			15 25	ns
t _{PHZ} Output disable time from high level				15 25	ns



SN54LS245 (J) SN74LS245 (J, N)

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
 ‡ All typical values are at VCC = 5V, TA = 25°C.
 * Not more than one output should be shorted at a time, and duration of the short-short should not exceed one second.



Designers' Data Sheet

"SURMETIC"[™] RECTIFIERS

... subminiature size, axial lead mounted rectifiers for general-purpose low-power applications.

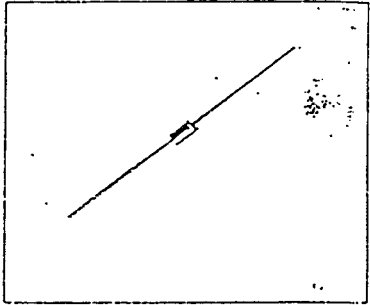
Designers' Data for "Worst Case" Conditions

The Designers' Data Sheets permit the design of most circuits entirely from the information presented. Limit curves — representing boundaries on device characteristics — are given to facilitate "worst case" design.

**1N4001
thru
1N4007**

**LEAD MOUNTED
SILICON RECTIFIERS**

- 50-1000 VOLTS
DIFFUSED JUNCTION



***MAXIMUM RATINGS**

Rating	Symbol	1N4001	1N4002	1N4003	1N4004	1N4005	1N4006	1N4007	Unit
Peak Repetitive Reverse Voltage	VRRM	50	100	200	400	600	800	1000	Volts
Working Peak Reverse Voltage	VPRM								
DC Blocking Voltage	VR								
Non-Repetitive Peak Reverse Voltage (infinite, single pulse, 60 Hz)	VRSM	60	120	240	480	720	1000	1200	Volts
RMS Reverse Voltage	VRRMS	35	70	140	280	420	560	700	Volts
Average Rectified Forward Current (single ohm resistive load, 60 Hz, see Figure 8, T _A = 25°C)	I _O				1.0				mA
Non-Repetitive Peak Surge Current (surge applied at rated load conditions, see Figure 2)	I _{FSM}				30 (for 1 cycle)				mA
Operating and Storage Junction Temperature Range	T _J , T _{stg}				-55 to +175				°C

***ELECTRICAL CHARACTERISTICS**

Characteristic and Conditions	Symbol	Typ	Max	Unit
Maximum Instantaneous Forward Voltage Drop (I _O = 1.0 Amp, T _J = 25°C) Figure 1	v _F	0.93	1.1	Volts
Maximum Full-Cycle Average Forward Voltage Drop (I _O = 1.0 Amp, T _J = 25°C, 1 inch lead)	V _{F(AV)}	-	0.8	Volts
Maximum Average Current (resistive load) (T _J = 25°C, T _J = 100°C)	I _q	3.05 1.0	10 50	mA
Maximum Full-Cycle Average Reverse Current (I _O = 1.0 Amp, T _J = 25°C, 1 inch leads)	I _{R(AV)}	-	30	μA

*Indicates JEDEC Registered Data.

MECHANICAL CHARACTERISTICS

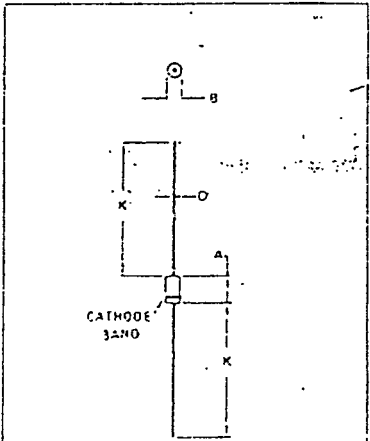
CASE: Transfer Moulded Plastic

MAXIMUM LEAD TEMPERATURE FOR SOLDERING PURPOSES: 350°C, 3/8" from case for 10 seconds at 5 lbs. tension

FINISH: All external surfaces are corrosion-resistant, leads are easily solderable

POLARITY: Cathode indicated by color band

WEIGHT: 0.40 Grams (Approximately)



DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	5.97	5.60	0.235	0.260
B	2.79	3.05	0.110	0.120
D	0.76	0.35	0.030	0.034
K	27.34	-	1.100	-

CASE 59-04
Does Not Conform to DD-1 Outline.

¹Trademark of Motorola Inc.

Supersedes 59-01

DS 5015 A3

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

Photo Detectors
Transistor Output

... designed for application in industrial inspection, processing and control, counters, sorters, switching and logic circuits or any design requiring radiation sensitivity, and stable characteristics.

- Hermetic Package at Economy Prices
- Popular TO-18 Type Package for Easy Handling and Mounting
- Sensitive Throughout Visible and Near Infrared Spectral Range for Wider Application
- Range of Radiation Sensitivities for Design Flexibility
- External Base for Added Control
- Annular Passivated Structure for Stability and Reliability

MAXIMUM RATINGS ($T_A = 25^\circ\text{C}$ unless otherwise noted)

Rating	Symbol	Value	Unit
Collector-Emitter Voltage	V_{CE0}	30	Volts
Emitter-Collector Voltage	V_{ECO}	5	Volts
Collector-Base Voltage	V_{CBO}	± 0	Volts
Total Power Dissipation ($T_A = 25^\circ\text{C}$) Derate above 25°C	P_D	250 2.27	mW mW/°C
Operating Temperature Range	T_A	-55 to +125	°C
Storage Temperature Range	T_{stg}	-55 to +150	°C

STATIC ELECTRICAL CHARACTERISTICS ($T_A = 25^\circ\text{C}$ unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Collector Dark Current ($V_{CC} = 20\text{ V}$, $R_L = 1\text{ Megohm}$, Note 2) $T_A = 25^\circ\text{C}$; $T_A = 35^\circ\text{C}$	I_{CE0}	—	— 5	0.1	μA
Collector-Base Breakdown Voltage ($I_C = 100\ \mu\text{A}$)	$V_{(BR)CBO}$	± 0	—	—	Volts
Collector-Emitter Breakdown Voltage ($I_C = 100\ \mu\text{A}$)	$V_{(BR)CEO}$	30	—	—	Volts
Emitter-Collector Breakdown Voltage ($I_E = 100\ \mu\text{A}$)	$V_{(BR)ECO}$	5	—	—	Volts

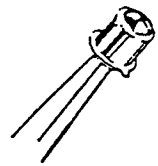
OPTICAL CHARACTERISTICS ($T_A = 25^\circ\text{C}$ unless otherwise noted)

Characteristic	Fig. No.	Symbol	Min	Typ	Max	Unit
Collector-Light Current ($V_{CC} = 20\text{ V}$, $R_L = 100\ \text{Ohms}$, Note 1)	1	I_L	0.1 0.2 0.5 1.5 2	—	—	mA
Photo Current Saturated Rise Time (Note 3)	5	$t_{r(sat)}$	—	1	—	μs
Photo Current Saturated Fall Time (Note 3)	5	$t_{f(sat)}$	—	10	—	μs
Photo Current Rise Time (Note 4)	5	t_r	—	2	—	μs
Photo Current Fall Time (Note 4)	5	t_f	—	3.5	—	μs
Wavelength of Maximum Sensitivity	—	λ_s	—	0.8	—	μm

- NOTES: 1. Radiation flux density (H) equal to 5 mW/cm^2 emitted from a tungsten source at a color temperature of 2970 K.
 2. Measured under dark conditions. $I_H = 0$.
 3. For saturated switching time measurements, radiation is provided by a pulsed xenon arc lamp with a pulse width of approximately 1 microsecond (see Figure 5).
 4. For unsaturated switching time measurements, radiation is provided by a pulsed GaAs (gallium-arsenide) light-emitting diode ($\lambda = 840\text{ nm}$) with a pulse width equal to or greater than 10 microseconds (see Figure 5).

MRD3050
MRD3051
MRD3054
MRD3055
MRD3056

PHOTO DETECTORS
TRANSISTOR OUTPUT
NPN SILICON
30 VOLTS



CASE 32-05
METAL

MRD3050, MRD3051, MRD3054, MRD3055, MRD3056

TYPICAL CHARACTERISTICS

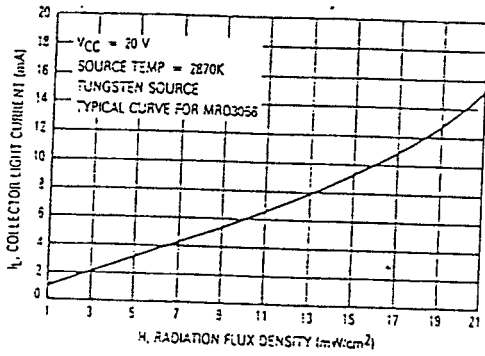


Figure 1. Collector Light Current

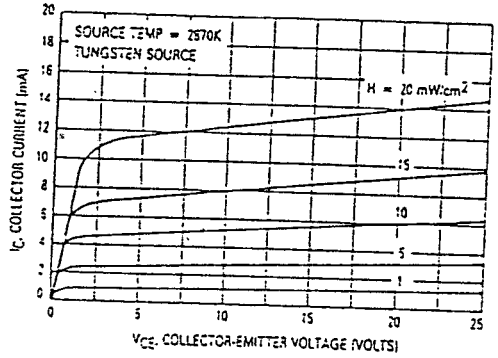


Figure 2. Collector Emitter Characteristics — MRD3056

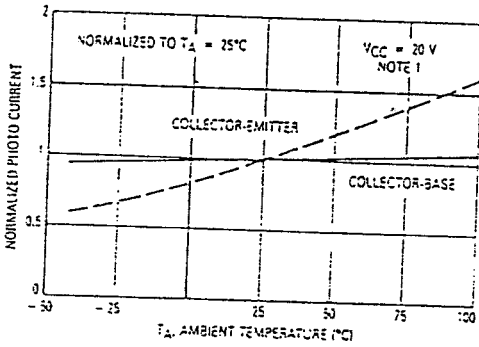


Figure 3. Photo Current versus Temperature

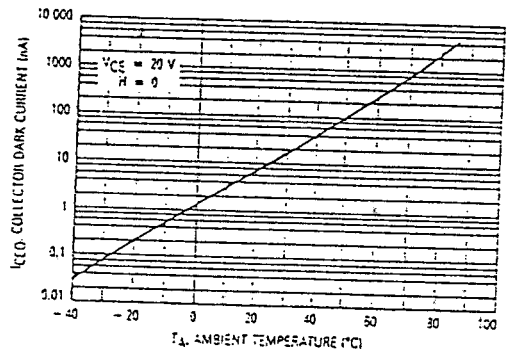


Figure 4. Dark Current versus Temperature

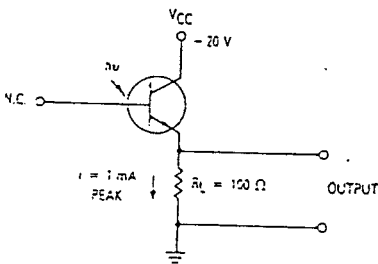


Figure 5. Pulse Response Test Circuit and Waveform

MRD3050, MRD3051, MRD3054, MRD3055, MRD3056

TYPICAL CIRCUIT APPLICATIONS

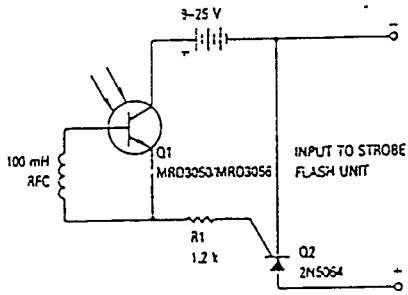


Figure 6. Strobe Flash Slave Adapter

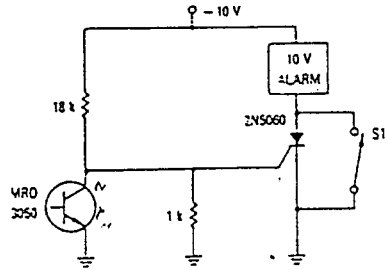


Figure 7. Light Operated SCR Alarm Using Sensitive-Gate SCR

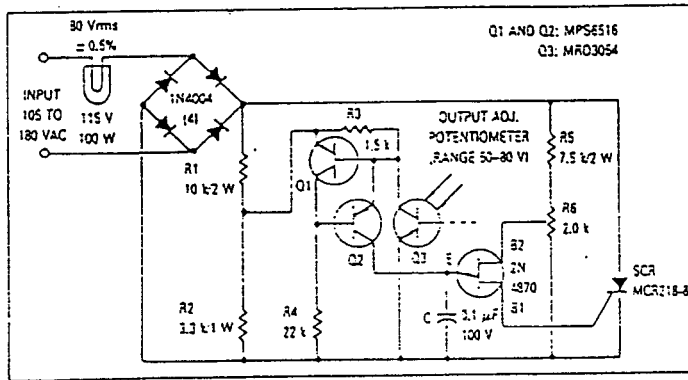
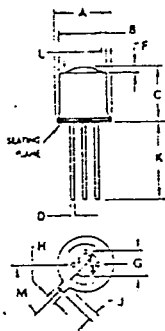


Figure 8. Circuit Diagram of Voltage Regulator for Projection Lamp

*MCR218-B to be used with a heat sink.

OUTLINE DIMENSIONS



- NOTES
 1. LEADS WITHIN 13 mm (0.051) RADIUS OF TRUE POSITION AT SEATING PLANE, AT MAXIMUM MATERIAL CONDITION.
 2. PIN 1 INTERNALLY CONNECTED TO CASE.

STYLE 1
 PIN 1: EMITTER
 2: BASE
 3: COLLECTOR

CASE 82-05
 METAL

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	3.31	3.50	0.130	0.138
B	1.52	1.75	0.060	0.069
C	1.57	1.48	0.062	0.058
D	2.41	2.48	0.095	0.098
E	—	1.14	—	0.045
F	—	1.54 BSC	—	0.061 BSC
G	—	1.17	—	0.046
H	—	1.37	—	0.054
I	—	1.52	—	0.060
J	—	1.27	—	0.050
K	—	1.27	—	0.050
L	—	1.27	—	0.050
M	—	1.27	—	0.050



LM78XX Series Voltage Regulators

General Description

The LM78XX series of three terminal regulators is available with several fixed output voltages making them useful in a wide range of applications. One of these is local on card regulation, eliminating the distribution problems associated with single point regulation. The voltages available allow these regulators to be used in logic systems, instrumentation, HiFi, and other solid state electronic equipment. Although designed primarily as fixed voltage regulators these devices can be used with external components to obtain adjustable voltages and currents.

The LM78XX series is available in an aluminum TO-3 package which will allow over 1.0A load current if adequate heat sinking is provided. Current limiting is included to limit the peak output current to a safe value. Safe area protection for the output transistor is provided to limit internal power dissipation. If internal power dissipation becomes too high for the heat sinking provided, the thermal shutdown circuit takes over preventing the IC from overheating.

Considerable effort was expended to make the LM78XX series of regulators easy to use and minimize the number

of external components. It is not necessary to bypass the output, although this does improve transient response. Input bypassing is needed only if the regulator is located far from the filter capacitor of the power supply.

For output voltage other than 5V, 12V and 15V the LM117 series provides an output voltage range from 1.2V to 57V.

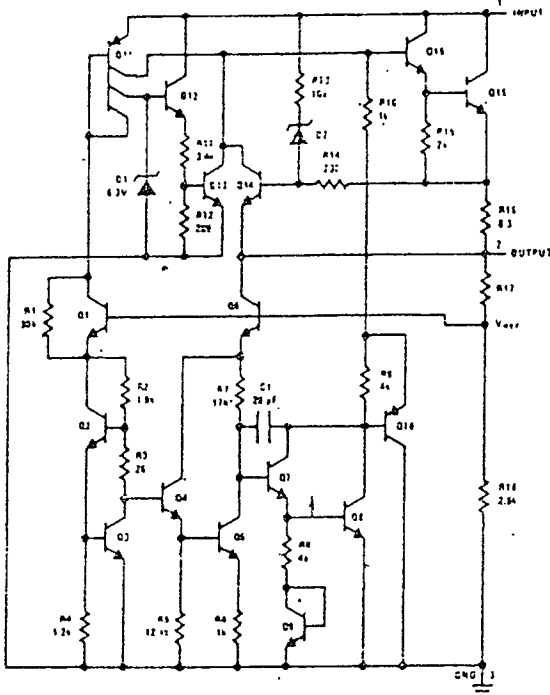
Features

- Output current in excess of 1A
- Internal thermal overload protection
- No external components required
- Output transistor safe area protection
- Internal short circuit current limit
- Available in the aluminum TO-3 package

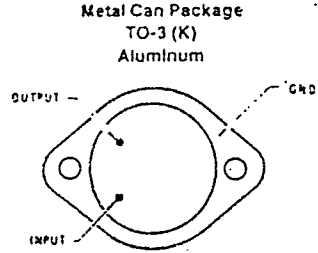
Voltage Range

LM7805C	5V
LM7812C	12V
LM7815C	15V

Schematic and Connection Diagrams

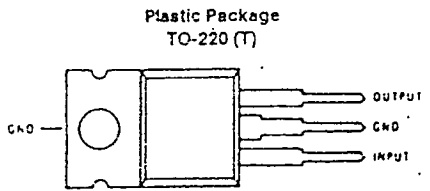


TL/H/7746-1



Bottom View

Order Number LM7805CK, LM7812CK or LM7815CK
See NS Package Number KC02A



Top View

Order Number LM7805CT, LM7812CT or LM7815CT
See NS Package Number T03B

TL/H/7746-2

TL/H/7746-3

LM78XX

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Input Voltage ($V_O = 5V, 12V$ and $15V$) 35V
 Internal Power Dissipation (Note 1) Internally Limited
 Operating Temperature Range (T_A) 0°C to $+70^\circ\text{C}$

Maximum Junction Temperature
 (P Package) 150°C
 (T Package) 150°C
 Storage Temperature Range -65°C to $+150^\circ\text{C}$
 Lead Temperature (Soldering, 10 sec.)
 TO-3 Package K 300°C
 TO-220 Package T 230°C

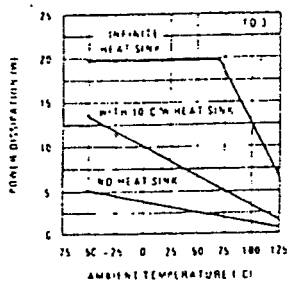
Electrical Characteristics LM78XX (Note 2) $0^\circ\text{C} \leq T_j \leq 125^\circ\text{C}$ unless otherwise noted.

		Output Voltage			5V			12V			15V			Units	
		Input Voltage (unless otherwise noted)			10V			19V			23V				
Symbol	Parameter	Conditions			Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
V_O	Output Voltage	$T_j = 25^\circ\text{C}, 5\text{ mA} \leq I_O \leq 1\text{ A}$			4.8	5	5.2	11.5	12	12.5	14.4	15	15.6	V	
		$P_D \leq 15\text{ W}, 5\text{ mA} \leq I_O \leq 1\text{ A}$ $V_{\text{MIN}} \leq V_{\text{IN}} \leq V_{\text{MAX}}$			4.75		5.25	11.4		12.6	14.25		15.75	V	
ΔV_O	Line Regulation	$I_O = 500\text{ mA}$	$T_j = 25^\circ\text{C}$			3			4			150			mV
			$0^\circ\text{C} \leq T_j \leq +125^\circ\text{C}$			$(7 \leq V_{\text{IN}} \leq 25)$			$(14.5 \leq V_{\text{IN}} \leq 27)$			$(17.5 \leq V_{\text{IN}} \leq 30)$			V
		ΔV_{IN}			50			120			150			mV	
		$0^\circ\text{C} \leq T_j \leq +125^\circ\text{C}$			$(8 \leq V_{\text{IN}} \leq 20)$			$(15 \leq V_{\text{IN}} \leq 27)$			$(18.5 \leq V_{\text{IN}} \leq 30)$			V	
ΔV_O	Load Regulation	$T_j = 25^\circ\text{C}$	$5\text{ mA} \leq I_O \leq 1.5\text{ A}$			10			12			150			mV
			$250\text{ mA} \leq I_O \leq 750\text{ mA}$			25			60			75			mV
		$5\text{ mA} \leq I_O \leq 1\text{ A}, 0^\circ\text{C} \leq T_j \leq +125^\circ\text{C}$			50			120			150			mV	
		$I_O \leq 1\text{ A}$			$T_j = 25^\circ\text{C}$			50			120			150	
ΔV_O	Load Regulation	$0^\circ\text{C} \leq T_j \leq +125^\circ\text{C}$			$(7.5 \leq V_{\text{IN}} \leq 20)$			$(14.6 \leq V_{\text{IN}} \leq 27)$			$(17.7 \leq V_{\text{IN}} \leq 30)$			V	
		ΔV_{IN}			60			75			75			mV	
I_Q	Quiescent Current	$I_O \leq 1\text{ A}$			6			8			150			mV	
		$0^\circ\text{C} \leq T_j \leq +125^\circ\text{C}$			8.5			8.5			8			mA	
ΔI_Q	Quiescent Current Change	$5\text{ mA} \leq I_O \leq 1\text{ A}$			0.5			0.5			0.5			mA	
		$T_j = 25^\circ\text{C}, I_O \leq 1\text{ A}$			1.0			1.0			1.0			mA	
		$V_{\text{MIN}} \leq V_{\text{IN}} \leq V_{\text{MAX}}$			$(7.5 \leq V_{\text{IN}} \leq 20)$			$(14.6 \leq V_{\text{IN}} \leq 27)$			$(17.9 \leq V_{\text{IN}} \leq 30)$			V	
V_N	Output Noise Voltage	$I_O \leq 500\text{ mA}, 0^\circ\text{C} \leq T_j \leq +125^\circ\text{C}$			1.0			1.0			1.0			mA	
		$V_{\text{MIN}} \leq V_{\text{IN}} \leq V_{\text{MAX}}$			$(7 \leq V_{\text{IN}} \leq 25)$			$(14.5 \leq V_{\text{IN}} \leq 30)$			$(17.5 \leq V_{\text{IN}} \leq 30)$			V	
$\frac{\Delta V_{\text{IN}}}{\Delta V_{\text{OUT}}}$	Ripple Rejection	$T_A = 25^\circ\text{C}, 10\text{ Hz} \leq f \leq 100\text{ kHz}$			40			75			90			μV	
		$f = 120\text{ Hz}$			$I_O \leq 1\text{ A}, T_j = 25^\circ\text{C}$ or $I_O \leq 500\text{ mA}$			62 80			55 72			54 70	
R_O	Dropout Voltage	$0^\circ\text{C} \leq T_j \leq +125^\circ\text{C}$			$(8 \leq V_{\text{IN}} \leq 18)$			$(15 \leq V_{\text{IN}} \leq 25)$			$(18.5 \leq V_{\text{IN}} \leq 28.5)$			V	
		$T_j = 25^\circ\text{C}, I_{\text{OUT}} = 1\text{ A}$			2.0			2.0			2.0			V	
		$I = 1\text{ kHz}$			8			18			19			m Ω	
		$T_j = 25^\circ\text{C}$			2.1			1.5			1.2			A	
V_{IN}	Input Voltage Required to Maintain Line Regulation	$T_j = 25^\circ\text{C}, I_O \leq 1\text{ A}$			0.6			1.5			1.8			A	
		$0^\circ\text{C} \leq T_j \leq +125^\circ\text{C}, I_O = 5\text{ mA}$			2.4			2.4			2.4			A	
V_{IN}	Input Voltage Required to Maintain Line Regulation	$0^\circ\text{C} \leq T_j \leq +125^\circ\text{C}, I_O = 5\text{ mA}$			0.6			1.5			1.8			mV/ $^\circ\text{C}$	
		$T_j = 25^\circ\text{C}, I_O \leq 1\text{ A}$			7.5			14.6			17.7			V	

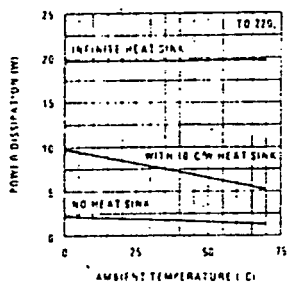
Note 1: Thermal resistance of the TO-3 package (K, KC) is typically $4^\circ\text{C}/\text{W}$ junction to case and $35^\circ\text{C}/\text{W}$ case to ambient. Thermal resistance of the TO-220 package (T) is typically $4^\circ\text{C}/\text{W}$ junction to case and $50^\circ\text{C}/\text{W}$ case to ambient.
 Note 2: All characteristics are measured with capacitor across the input of $0.22\ \mu\text{F}$, and a capacitor across the output of $0.1\ \mu\text{F}$. All characteristics except noise voltage and ripple rejection ratio are measured using pulse techniques ($t_w \leq 10\text{ ms}$, duty cycle $\leq 5\%$). Output voltage changes due to changes in internal temperature must be taken into account separately.

Typical Performance Characteristics

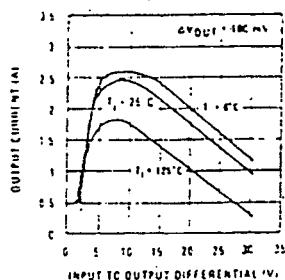
Maximum Average Power Dissipation



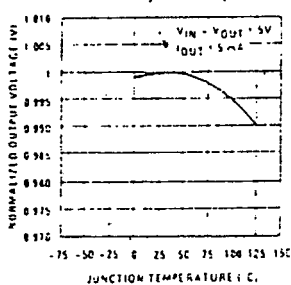
Maximum Average Power Dissipation



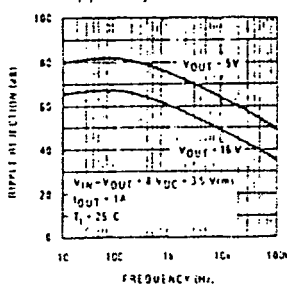
Peak Output Current



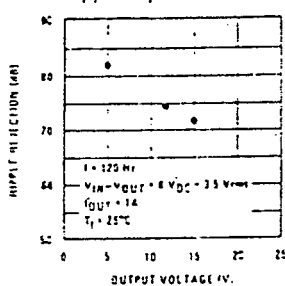
Output Voltage (Normalized to 1V at Tj = 25°C)



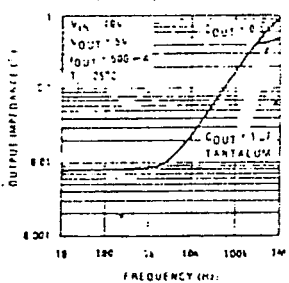
Ripple Rejection



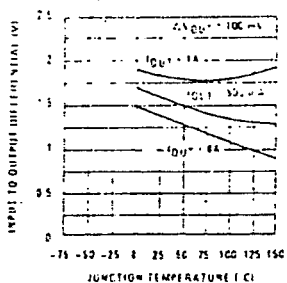
Ripple Rejection



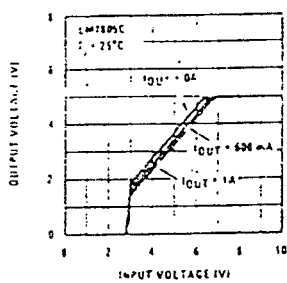
Output Impedance



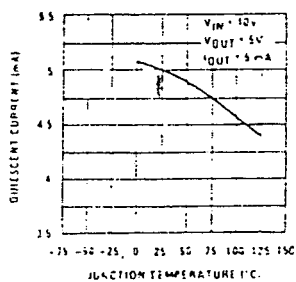
Dropout Voltage



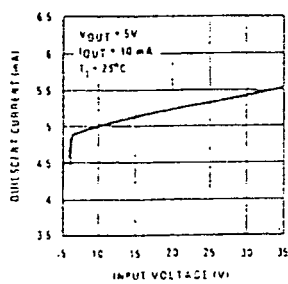
Dropout Characteristics



Quiescent Current



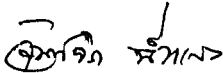
Quiescent Current



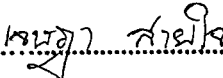
กิตติกรรมประกาศ

ก่อนที่โปรเจกต์นี้จะสำเร็จตามเป้าหมายที่กำหนดได้ ต้องขอบคุณอาจารย์และเจ้าหน้าที่ภาค
วิศวกรรมเครื่องกลที่อำนวยความสะดวกในเรื่องสถานที่และเครื่องมือในการสร้างแขนกล และ
ที่ลืมไม่ได้คืออาจารย์ที่ปรึกษา อาจารย์โกศล ชวนขยัน ที่จัดหาคอมพิวเตอร์มาให้นักศึกษาทำโปร
เจกต์และยังคอยให้คำแนะนำและให้กำลังใจในการทำโปรเจกต์แก่นักศึกษาเสมอ ซึ่งต้องกราบ
ขอบพระคุณอาจารย์มา ณ ที่นี้ด้วย ขอขอบคุณเพื่อน ๆ น้อง ๆ ทุกคนที่เอื้อเฟื้อในการจัดซื้ออุปกรณ์
และให้คำแนะนำที่เป็นประโยชน์

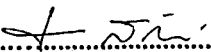
สุดท้ายขอขอบพระคุณทุกท่านที่มีส่วนทำให้โปรเจกต์ครั้งนี้เป็นไปตามเป้าหมาย


.....

(นางสาวจันทจิรา น้าแสง)


.....

(นายเจษฎา สายใจ)


.....

(นายรัช สนวนันท์)

เอกสารอ้างอิง

- Takashi Kenjo , “ Stepping motor and their microprocessor controls ”, Oxford Science Publications.
- Acarnley P.P, Stepping motor : a guide to modern theory and practice”, Institution of Electrical Engineering.
- Joseph Ednard Shigley, Mechanical Engineering Design ”, McGraw Hill.
- Arthur G. Erdman, Machanism Design, Analysis and Systhesis volum I ”, Prentice Hall.
- Ben-Zion Sandler, Robotic Design The Machanisms For Automated Machinery ”, Prentice Hall Internation Editions, 1992.
- Richard D. Klafter, Robotic Engineering, An Integrated Approach ”, Prentice Hall.
- James Harter, Electromachanics, principles, consepts and devices ”, Prentice Hall.
- Theodore F. Bogart, Merritt, “ Electronic Device And Circuits ”.
- Willis J. Tompkins, “ Interfacing Sensor to the IBM PC ”, Prentice Hall.
- Muhammand Ali Mazidi, Janice Gillispie Mazidi, “ The 80x86 IBM PC & Compatible computers Volum II : Design and interfacing of the IBM PC, PS and compatibles , Prentice Hall International, INC., 1995.
- ดร. วริทธิ์ อึ้งภากรณ์, การออกแบบเครื่องจักรกลเล่ม 1-2, บริษัทซีเอ็ด ยูเคชั่น จำกัด.
- มนตรี พจนารถลาวัลย์, การเขียนโปรแกรมคอมพิวเตอร์ด้วยภาษาซี, บริษัทซีเอ็ด ยูเคชั่น จำกัด.
- รศ. ยืน ภู่วรรณ, ทฤษฎีและการประยุกต์ใช้งานไมโครโปรเซสเซอร์ Z-80, บริษัทซีเอ็ด ยูเคชั่น จำกัด, พ.ศ. 2532.