



หุ่นยนต์ส่งลูกเทเบิลเทนนิส
BALL SHOOTING ROBOT

นาย จักร บุญเฉลิมวิภาส
MR. JACQUES BOONCHALERMWIPASS

นาย ภาณุมาศ อรุณเดชาวัฒน์
MR. PANUMAS ARUNDACHAWAT

วัน เดือน ปี.....	-๙.๕๒.๖๔1
เลขทะเบียน.....	038888
เลขเรียกหนังสือ.....	T ๒๐๒๒ ก ๒๑๕๓

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้วยวิธี
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

038888

BALL SHOOTING ROBOT




A THESIS SUBMITTED IN PARTIAL FUFILMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN MECHANICAL ENIGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

1997

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาานิพนธ์

หัวข้อปริญญาานิพนธ์ หุ่นยนต์ส่งลูกเทเบิลเทนนิส
BALL SHOOTING ROBOT
ชื่อนักศึกษา นายจักร บุญเฉลิมวิภาส รหัสประจำตัว 37014046
นายภาณุมาศ อรุณเดชาวัฒน์ รหัสประจำตัว 37014317
หลักสูตร วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมเครื่องกล
อาจารย์ผู้ควบคุมวิทยานิพนธ์



(อ. อัครเดช สิ้นธุภาคร)

หัวข้อปริญญานิพนธ์ หุ่นยนต์ส่งลูกเทเบิลเทนนิส

นักศึกษา นายจักร บุญเฉลิมวิภาส

นาย ภาณุมาศ อรุณเดชาวัฒน์

อาจารย์ผู้ควบคุมปริญญานิพนธ์

อ.อัครเดช สีนฤภัคร

หลักสูตร วิศวกรรมศาสตรบัณฑิต

ภาควิชา วิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยี

พระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ 2540

บทคัดย่อ

ปัจจุบันงานควบคุมทางกลส่วนมากได้มีการใช้คอมพิวเตอร์ มาช่วยควบคุมให้เกิดความแม่นยำมากขึ้น ดังนั้นเพื่อเป็นการทำความเข้าใจเบื้องต้นในงานควบคุมทางกลด้วยคอมพิวเตอร์ จึงทำให้เกิดการศึกษาโครงการนี้ขึ้น โดยเลือกใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มาศึกษาเพราะมีขนาดเล็ก และเหมาะสมกับงานควบคุมที่ไม่ซับซ้อนมากนัก รวมทั้งโครงการนี้ได้พัฒนามาจากเครื่องยิงลูกเทเบิลเทนนิส(ROBOPONG 2000) ซึ่งเป็นระบบควบคุมด้วยมือมาเป็นระบบควบคุมอัตโนมัติ

Thesis Title Robot Ballshooting
Student Mr.Jacques Boonchalemwipass
Mr.Panumas Arundachawat
Thesis Adviser Mr. Akradaech Sindhuphak
Degree Bechelor of Engineering in Mechanical Engineering
Department Mechanical Engineering Faculty of King Mongkut's Institute of
Technology Ladkrabang
Year 1997

Abstract

In the present day the computer is mostly used to control mechanical parts for more accuracy. This project was founded by the purpose to study the basic of mechanical control by computer. Microcontroller MCS-51 was selected. It is compact and suitable for the uncomplicated work. This project was developed from the simple ball shooter (ROBOPONG 2000) with the computer program that control the movement and ball profile automatically.

กิติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ประสบความสำเร็จด้วยดี โดยได้รับความช่วยเหลือ และการสนับสนุน จากบุคคลต่างๆดังรายนามต่อไปนี้

อาจารย์ อัครเดช สินธุภักดิ์ ผู้ให้คำปรึกษา และเงินทุนในการดำเนินงาน รวมทั้งต้นแบบ ของ ROBOPONG 2000 คุณ มณฑา เทียมเมือง ผู้จัดหาเครื่องมือ และอุปกรณ์ในการดำเนินงาน คุณเบ็ล คุณเปา คุณอาร์ม คุณป๊อก คุณเด่น คุณวัฒน์ คุณแอน คุณเอ ผู้ให้คำแนะนำทางด้านโปรแกรมการควบคุม คุณสุทธิโชค ผู้ชื่นชม กับความคิดสร้างสรรค์ของโครงการนี้ มาโดยตลอด คุณปริญญา ผู้สร้างความขบขัน และความสนุกสนานให้เกิดขึ้น คุณเอก คุณมรกต ผู้ให้ความช่วยเหลือทางด้านการพิมพ์ ผู้จัดทำโครงการนี้ขอขอบพระคุณบุคคลต่างๆดังที่ได้กล่าว มาเป็นอย่างสูง

สุดท้ายนี้ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ผู้ให้กำลังใจ และสร้างให้พวกเราได้ลิ้มรสชาติ โลก และ พระผู้เป็นเจ้า ผู้สร้างโลกใบนี้

จักร บุญเฉลิมวิภาส

ภาณุมาศ อรุณเดชาวัฒน์

สารบัญ

หน้า

บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VI
บทที่	
1. บทนำ	1
ขอบเขตของการศึกษาปริญญาโท	1
2. หุ่นยนต์ส่งลูกเทเบิลเทนนิส	2
แผนการทำงานของหุ่นยนต์ส่งลูกเทเบิลเทนนิส	2
ส่วนประกอบของหุ่นยนต์ส่งลูกเทเบิลเทนนิส	3
ภาพแบบของหุ่นยนต์ส่งลูกเทเบิลเทนนิส	7
3. อุปกรณ์ควบคุมการทำงาน	13
บอร์ดแจส 31	13
ชิพเบอร์ 8051	20
4. อุปกรณ์อุปกรณ์ควบคุมการส่งลูกเทเบิลเทนนิส	27
วงจรควบคุมความเร็ว DC Motor	27
เซอร์โวมอเตอร์	28
5. หลักการทำงานของหุ่นยนต์ส่งลูกเทเบิลเทนนิส	31
ลำดับขั้นตอนการทำงาน	31
ขั้นตอนการทำงานของเครื่องยิงลูกเทเบิลเทนนิส	32
6. ข้อสรุปและข้อเสนอแนะ	33
ปัญหาการทำงานของหุ่นยนต์ส่งลูกเทเบิลเทนนิส	33
ข้อเสนอแนะ	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



สารบัญภาพ

	หน้า
1. แสดงดีกรีออฟฟรีดอม	2
2. ชุดตุ๊กตากลูก	3
3. ชุดยิง	5
4. หุ่นยนต์ส่งลูกเทเบิลเทนนิส	5
5. ชนิดแขนกลของหุ่นยนต์ส่งลูกเทเบิลเทนนิส	6
6. ภาพฉายด้านหน้าของชุดยิง	7
7. ภาพตัดด้านข้างของชุดยิง	8
8. ภาพฉายด้านหน้าของชุดตุ๊กตากลูก	9
9. ภาพตัดด้านข้างของชุดตุ๊กตากลูก	10
10. ส่วนประกอบของหุ่นยนต์ส่งลูกเทเบิลเทนนิส	11
11. ส่วนประกอบของชุดตุ๊กตากลูก	12
12. ภาพแสดงตำแหน่งต่างๆบนบอร์ด	15
13. โครงสร้างภายในสำหรับเบอร์ 8051	20
14. การจัดวางขาของ 8051	22
15. แสดงรีจิสเตอร์ของ 8051	22
16. แสดงชุดคำสั่งของไมโครคอนโทรลเลอร์ MCS-51	26
17. รูปวงจร	28
18. แสดงวงจรควบคุมตำแหน่งของเซอร์โวมอเตอร์	29
19. แสดงการทดสอบโดยใช้เฟืองของเซอร์โวมอเตอร์	30
20. แสดงทิศทางการเคลื่อนที่ของเซอร์โวมอเตอร์	30
21. ภาพแสดงลำดับขั้นตอนการทำงาน (FLOW CHART)	31

บทที่ 1

บทนำ

ปัจจุบันกล่าวได้ว่า งานควบคุมอัตโนมัติได้เข้ามามีบทบาทในชีวิตประจำวันมากขึ้นทุกที โดยเฉพาะการควบคุมแขนกลและหุ่นยนต์

หัวใจสำคัญของงานควบคุมอัตโนมัติคือ โปรแกรมควบคุมการทำงาน ซึ่งในปัจจุบันมีให้เลือกใช้หลายโปรแกรมเช่น ภาษาซี ภาษาปาสคาล ภาษาแอสเซมบลี แต่ในปริญญาโทนี้ได้เลือกใช้ภาษาแอสเซมบลี เพราะเลือกใช้ชิพควบคุมตระกูล MCS-51 ซึ่งต้องใช้ภาษาแอสเซมบลี เขียนโปรแกรมในการควบคุมการทำงาน

ขอบเขตของการศึกษาปริญญาโท

ขอบเขตของการศึกษาแบ่งออกเป็น 2 ส่วนคือ

1. ด้านอุปกรณ์ (Hardware)

ด้านอุปกรณ์ได้แบ่งการออกอีกเป็น 2 ส่วน คือ

1.1 ด้านอุปกรณ์ทางกล

อุปกรณ์ทางกล คือ ส่วนที่เกี่ยวข้องกับตัวของหุ่นทั้งหมด ได้แก่

1.1.1 ชุดยิง

1.1.2 ชุดตักลูก

1.2 ด้านอุปกรณ์ควบคุมทางไฟฟ้า

อุปกรณ์ควบคุมทางไฟฟ้า คือ ส่วนอุปกรณ์ที่ควบคุมการทำงานทั้งหมดได้แก่

1.2.1 เซอร์โวมอเตอร์

1.2.2 ดีซีมอเตอร์

1.2.3 วงจรควบคุมควบคุมความเร็วมอเตอร์ล้อยิง

1.2.4 บอร์ดไมโครคอนโทรลเลอร์แอส 31

2. ด้านโปรแกรมควบคุมการทำงาน (Software)

โปรแกรมควบคุมการทำงาน เลือกใช้ภาษาแอสเซมบลี

บทที่ 2

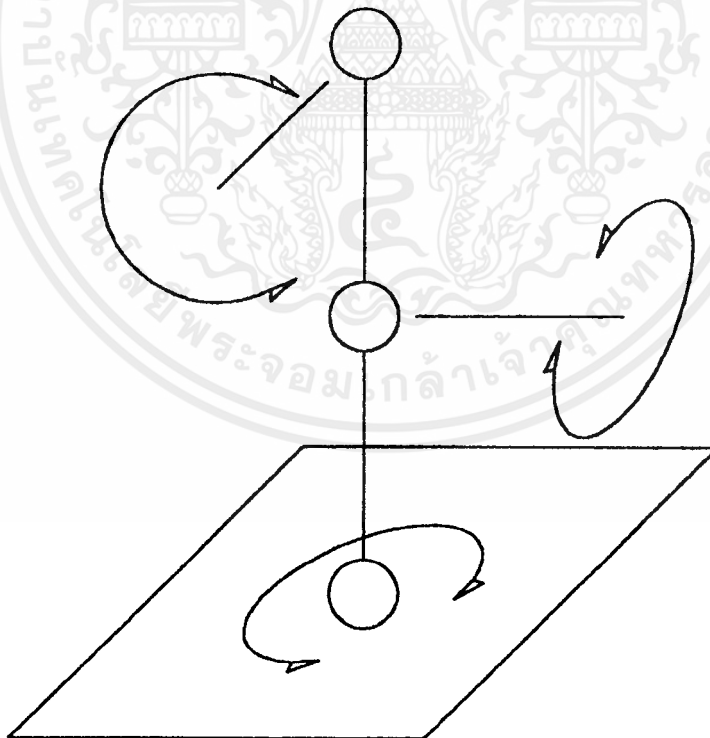
หุ่นยนต์ส่งลูกเทเบิลเทนนิส

1. แกนการทำงานของหุ่นยนต์ส่งลูกเทเบิลเทนนิส

ลักษณะการแกนทำงานของหุ่นยนต์แบบ 3 แกนคือ

1. หมุนรอบแกน x (Pitching)
2. หมุนรอบแกน y (Yawing)
3. หมุนรอบแกน z (Rolling)

การทำงานแบบ 3 แกนนี้จะทำให้สามารถส่งลูกบอลออกมาได้มากมายหลายแบบคล้ายกับการตีโต้ตอบกันในการแข่งขัน



รูปที่ 2.1 แสดงดีกรีoffreedom

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ส่วนประกอบของหุ่นยนต์ส่งลูกเทเบิลเทนนิส

ส่วนประกอบด้วย 2 ส่วนหลัก คือ

2.1 ส่วนชุดเก็บลูก

ชุดเก็บลูกจะมีลักษณะเป็นอ่าง ที่ก้นอ่างจะมีชุดป้อนลูกเข้าสู่ชุดยิง และข้างหลังจะเป็นตาข่ายคอยด์ักลูกบอลให้ตกลงในอ่าง

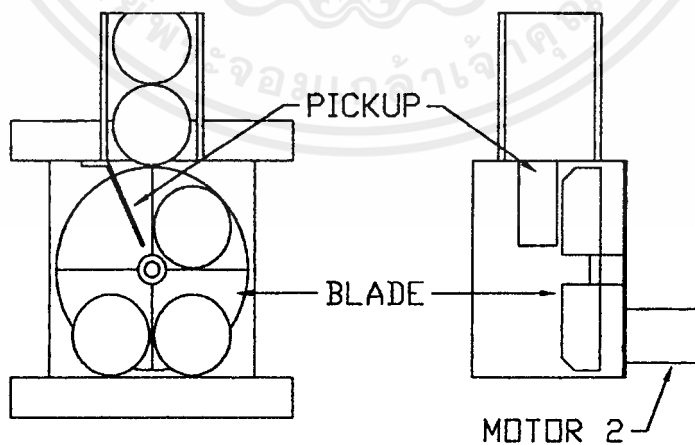
ชุดป้อนลูกจะประกอบด้วย

2.1.1 ใบพัดตักลูก (Blade) จะเป็นลักษณะแผ่นจานมีใบ 4 ใบ แต่ละใบมีองศาต่างกัน 90 องศา

2.1.2 ลิ้นกั้นลูก (Gate) ลิ้นกั้นลูกจะคอยกั้นที่ถูกตักจากใบพัด ให้เข้าไปในรูที่ต่อกับท่อเพื่อส่งให้ชุดยิงต่อไป

2.1.3 มอเตอร์ (Motor) เป็นอุปกรณ์ต้นกำลังจะต่อกับใบพัดทำหน้าที่เป็นต้นกำลังขับเคลื่อนใบพัดตักลูกบอล

2.1.4 ท่อส่งลูก (Pipe) ท่อส่งลูกจะเป็นท่อที่ใช้ส่งลูกบอลจากชุดป้อนลูกเข้าสู่ชุดยิง



รูปที่ 2.2 ชุดตักลูก

2.2 ส่วนยิงลูกบอล

ส่วนยิงลูกบอลจะประกอบด้วย 2 ส่วนย่อย คือ

2.2.1 อุปกรณ์ควบคุมทิศทาง

อุปกรณ์ควบคุมทิศทางจะประกอบด้วย

2.2.1.1 เซอร์โวมอเตอร์ 3 ตัว

2.2.1.2 คันชัก 2 อัน

2.2.1.3 รอกหมุนกระบอกลูก

2.2.1.4 สายพานหมุนกระบอกลูกยิง

เซอร์โวมอเตอร์จะได้รับสัญญาณสั่งมาจากไมโครคอนโทรลเลอร์ ให้หมุนหาตำแหน่งในทั้ง 3 แนวแกนตามที่ต้องการ

2.2.2 อุปกรณ์ยิงลูกบอล

อุปกรณ์ยิงลูกบอลจะประกอบด้วย

2.2.2.1 กระบองยิง ซึ่งจะต่อกับท่อส่งลูก

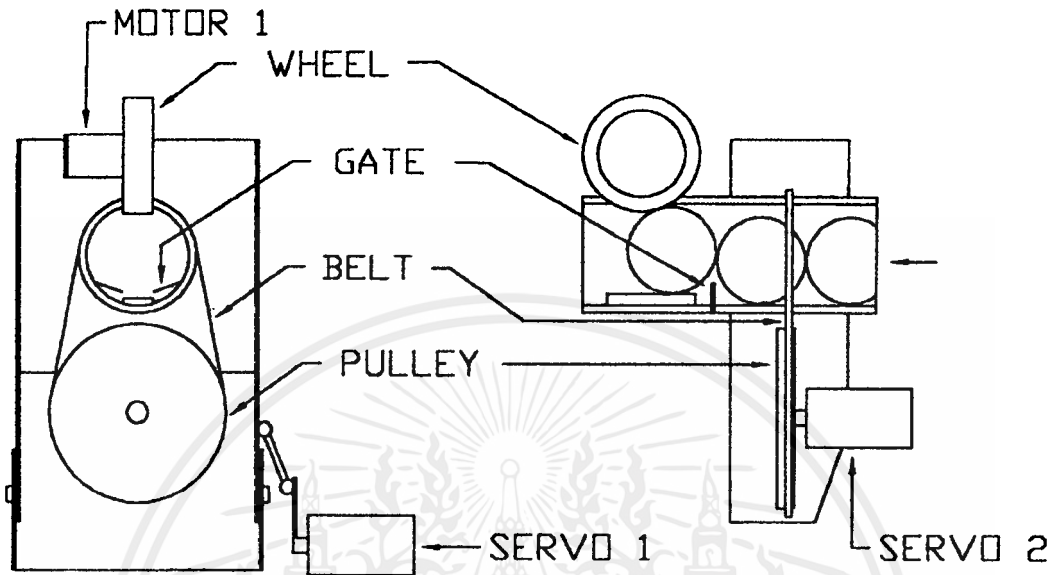
2.2.2.2 มอเตอร์ยิง เป็นอุปกรณ์ต้นกำลังในการยิงลูก

2.2.2.3 วงล้อยิง เป็นอุปกรณ์ที่ต่อกับมอเตอร์ เป็นอุปกรณ์ในการยิงลูกบอล

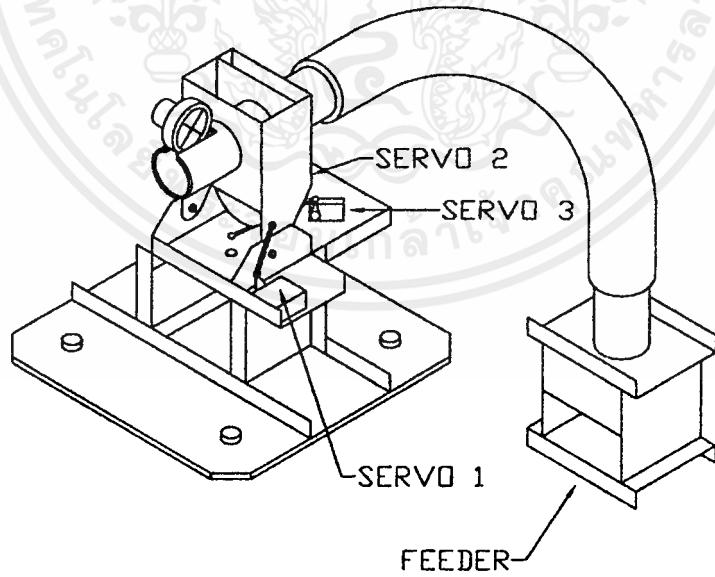
ออกไป

2.2.2.4 ลิ้นกักลูก มีลักษณะเป็นลวดตัดปลายทั้ง 2 ข้าง ทำหน้าที่กักลูกให้ถูกส่งให้ถูกส่งออกยิงได้ที่ละลูก

2.2.2.5 ยางช่วยยิงลูก เป็นยางที่ติดเอาไว้กับตัวกระบอกลูกยิง ทำหน้าที่ยึดลูกไม่ให้ลื่นไถลออกไปที่ละหลายลูก เวลามอเตอร์ตักลูกทำงาน โดยจะยึดให้ลูกออกได้ที่ละลูกในแต่ละจังหวะการทำงาน

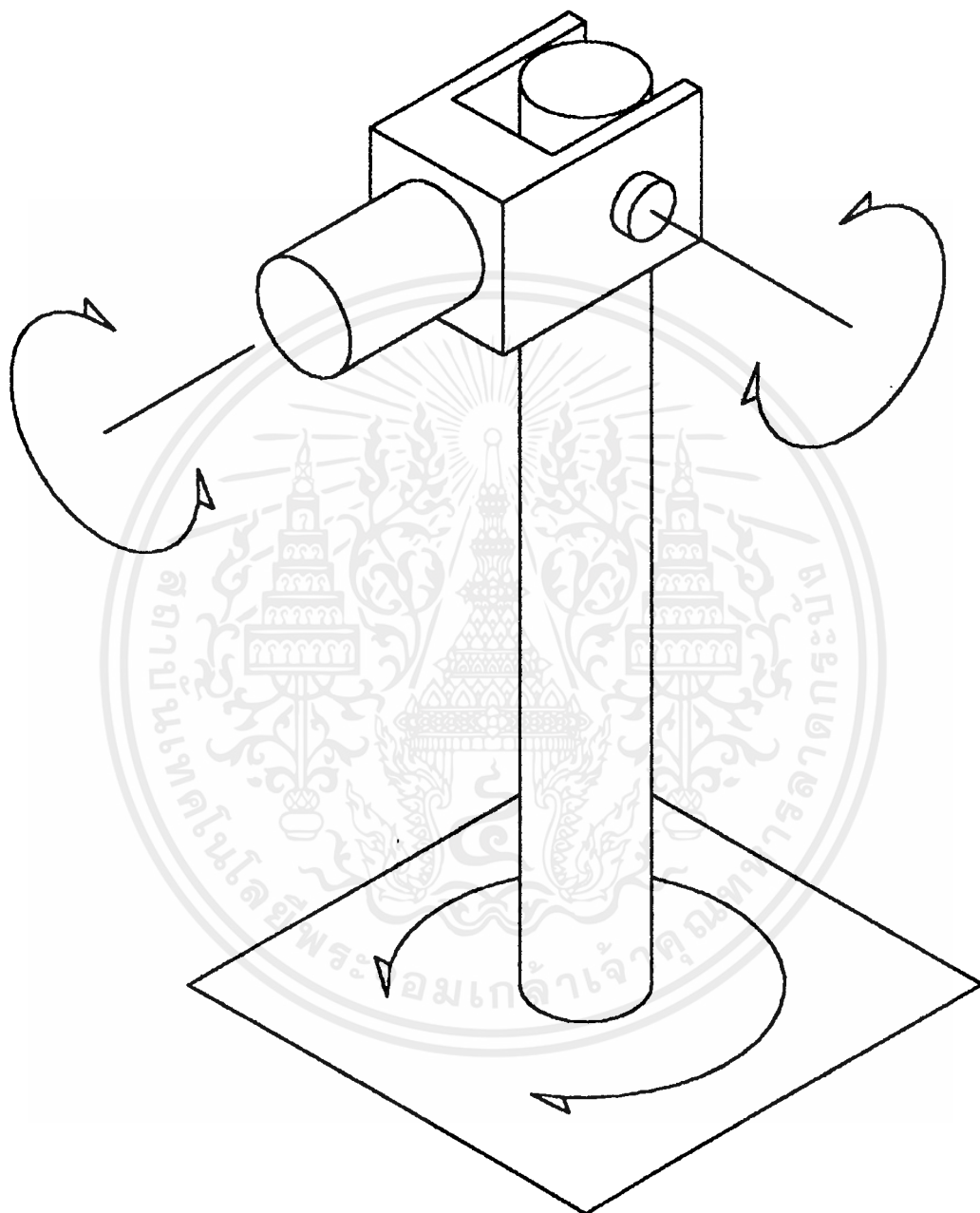


รูปที่ 2.3 ชุดยิง



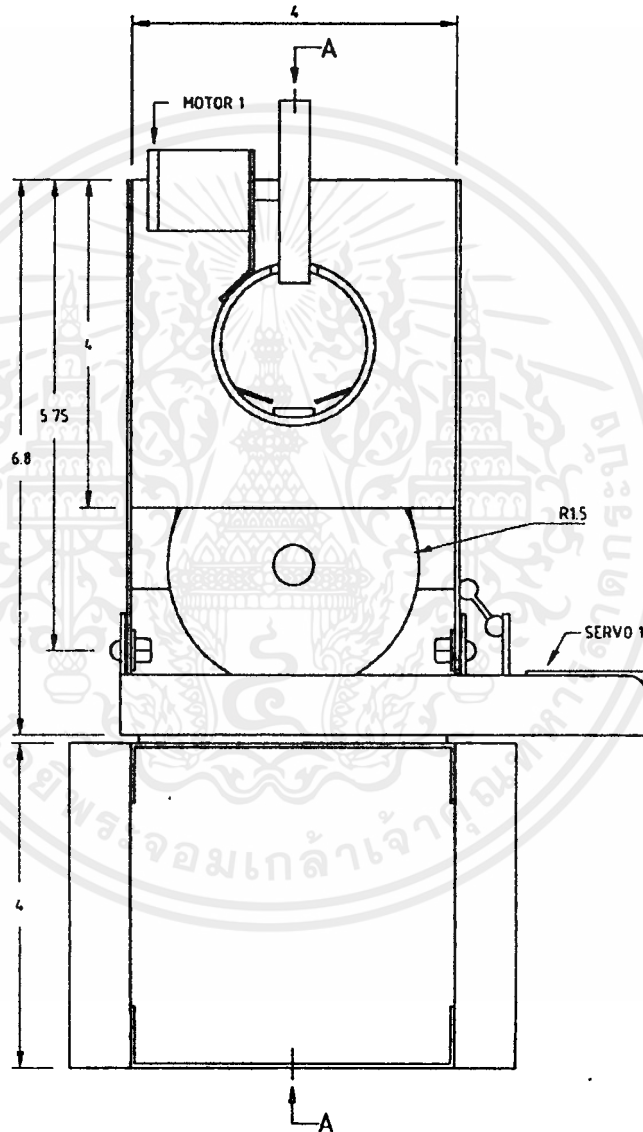
รูปที่ 2.4 หุ่นยนต์ส่งลูกเทเบิลเทนนิส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

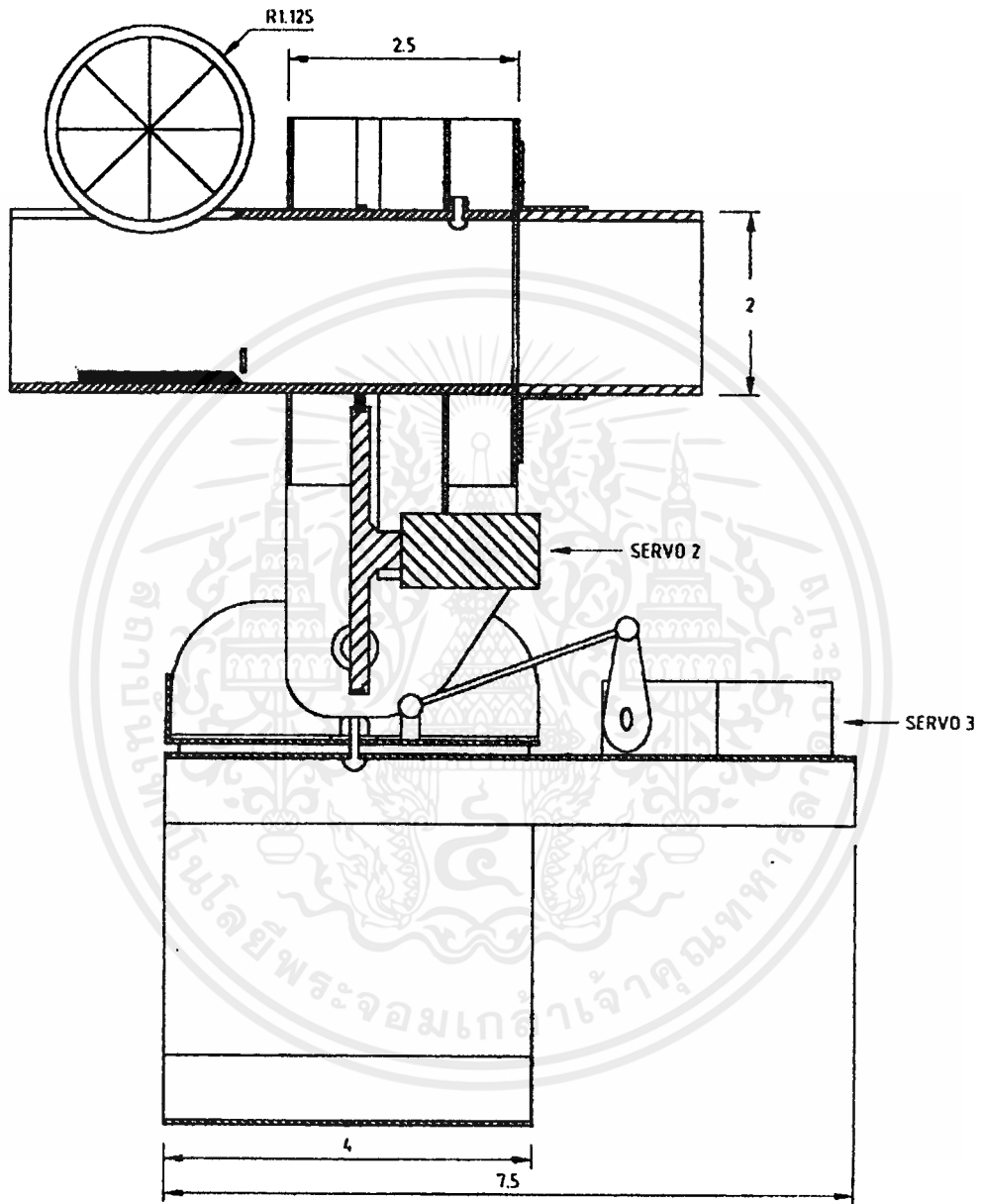


รูปที่ 2.5 ชนิดแกนกลของหุ่นยนต์ส่งลูกเทเบิลเทนนิส

3. ภาพแบบของหุ่นยนต์ส่งลูกเทเบิลเทนนิส



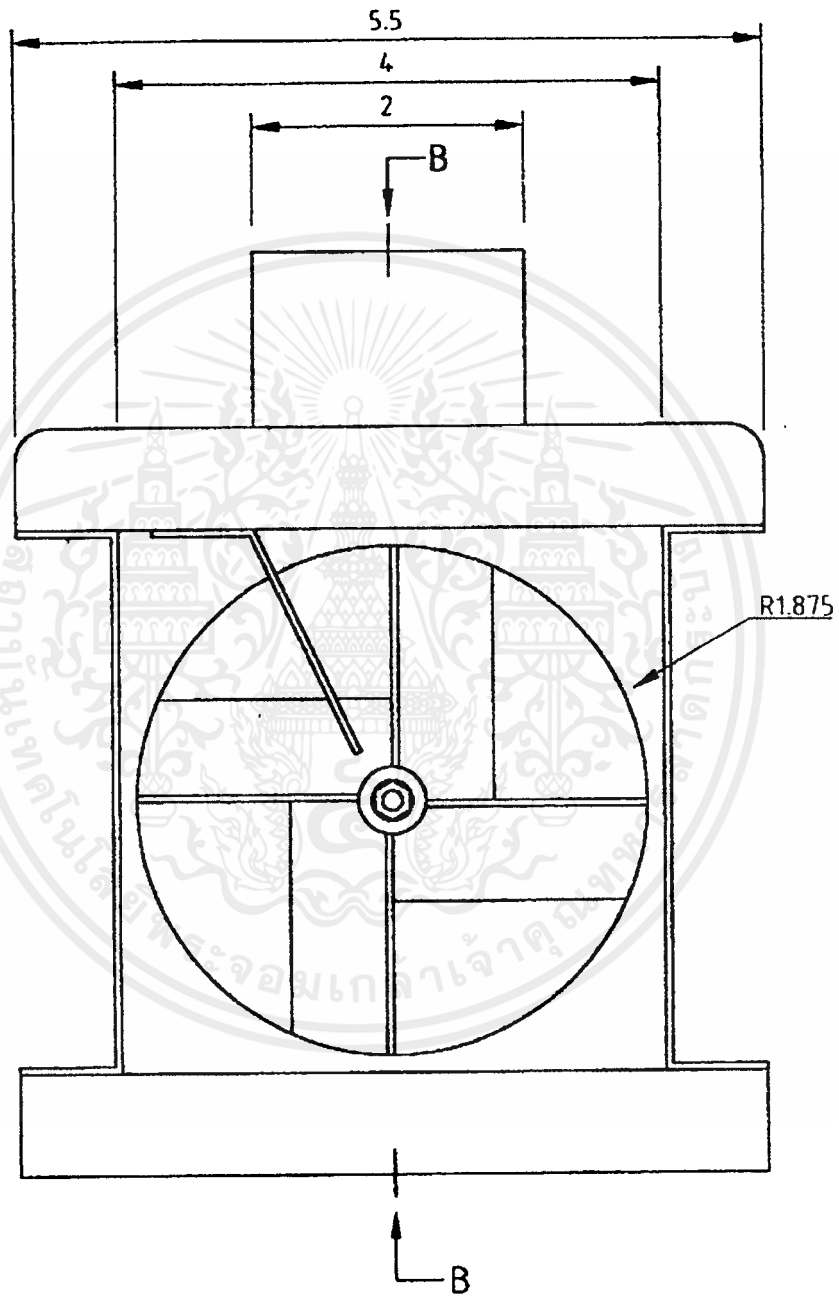
รูปที่ 2.6 ภาพฉายด้านหน้าของชุดยิง



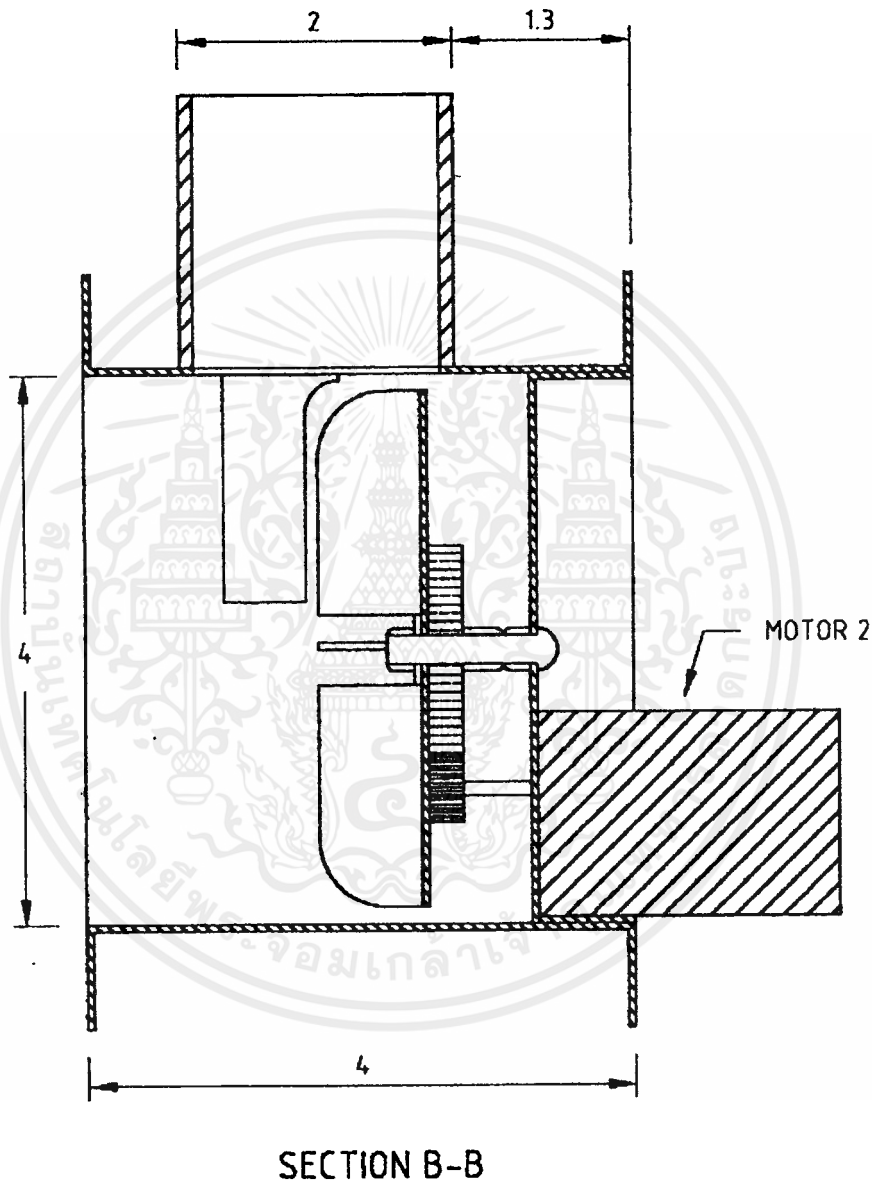
SECTION A-A

รูปที่ 2.7 ภาพตัดด้านข้างของชุดยิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

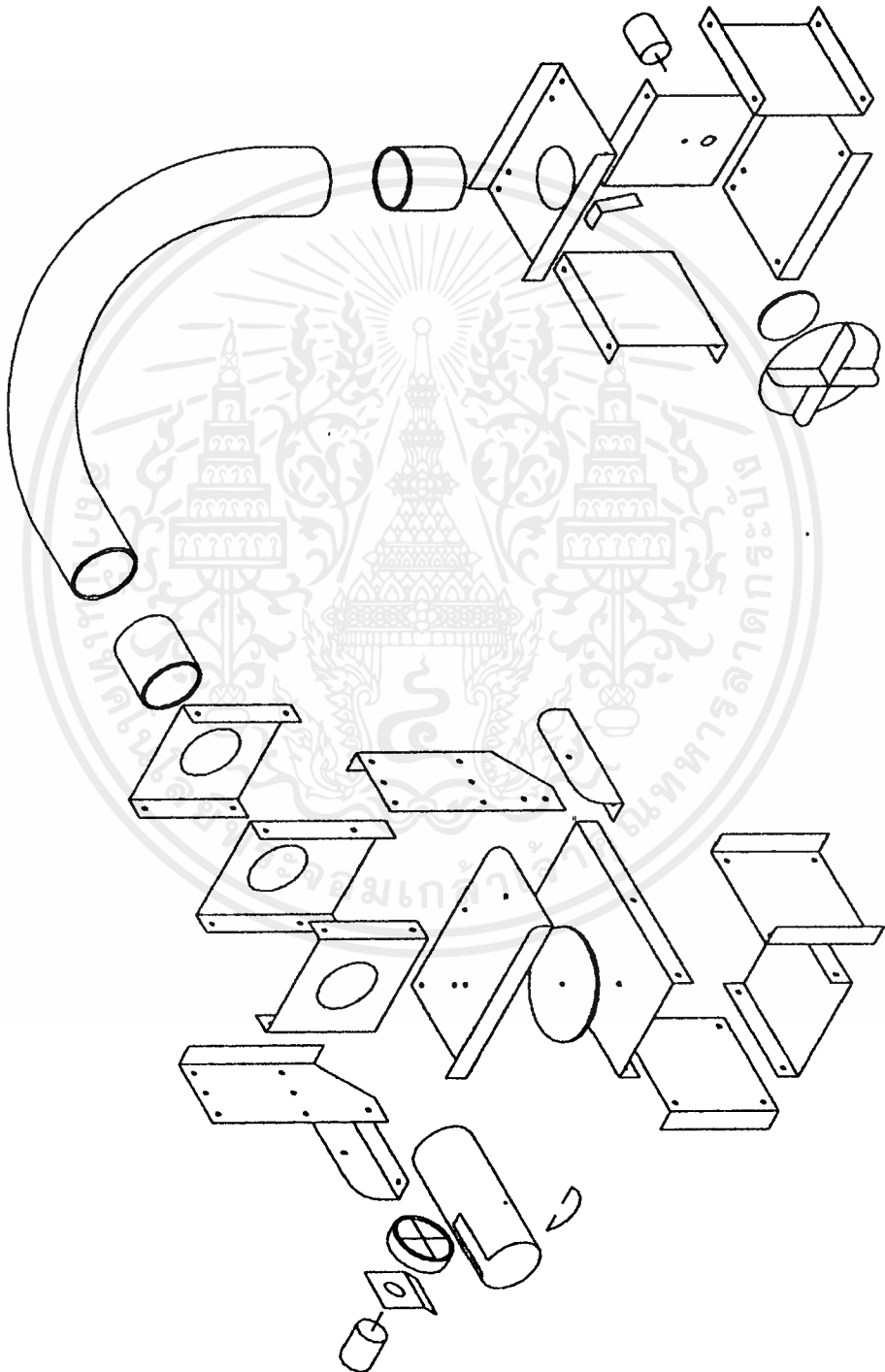


รูปที่ 2.8 ภาพฉายด้านหน้าของชุดตักลูก



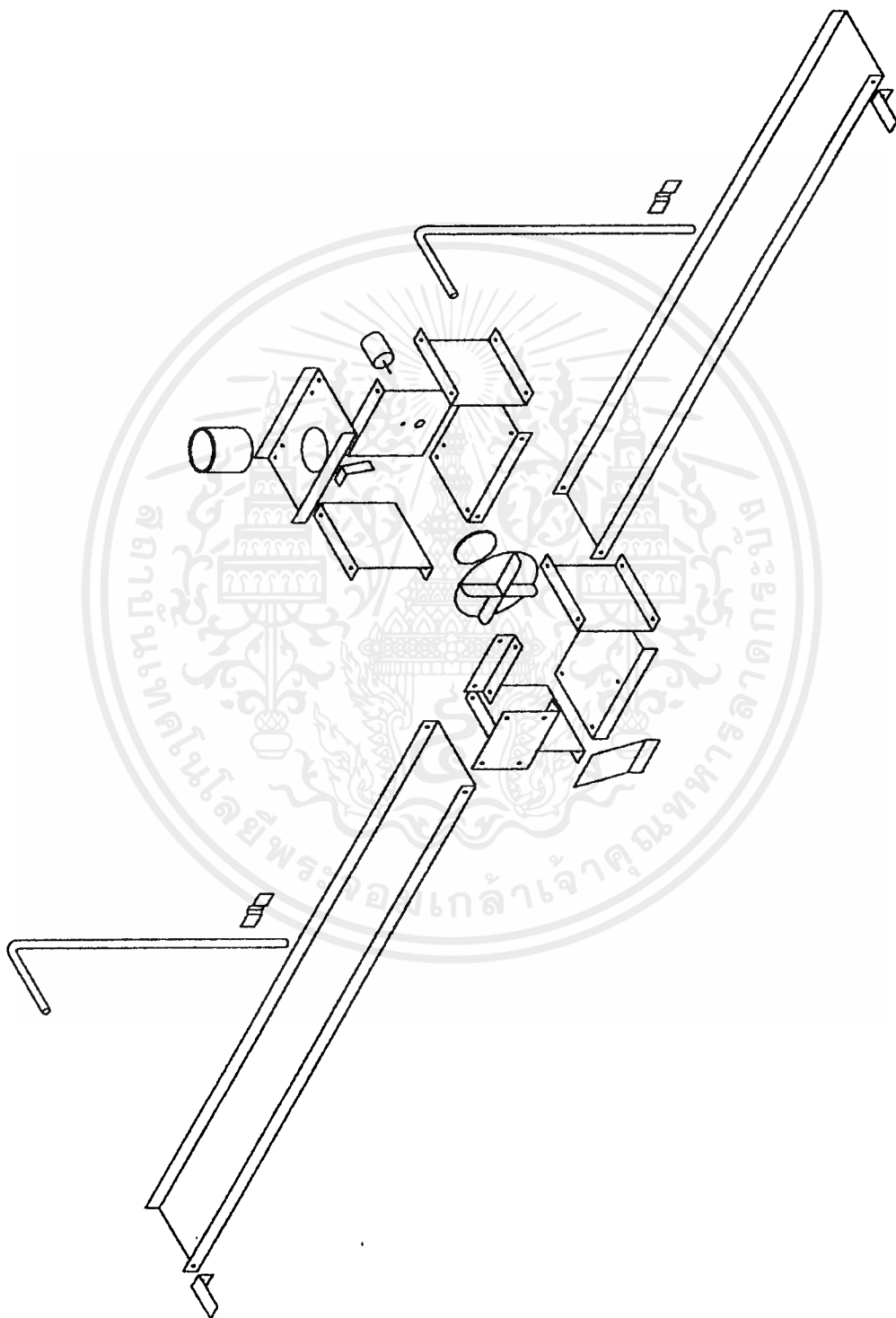
รูปที่ 2.9 ภาพตัดด้านข้างของชุดตักลูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 ส่วนประกอบของหุ่นยนต์ส่งลูกเทเบิลเทนนิส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 ส่วนประกอบของชุดกล้องจุลทรรศน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

อุปกรณ์ควบคุมการทำงาน

1. บอร์ดแอส 31

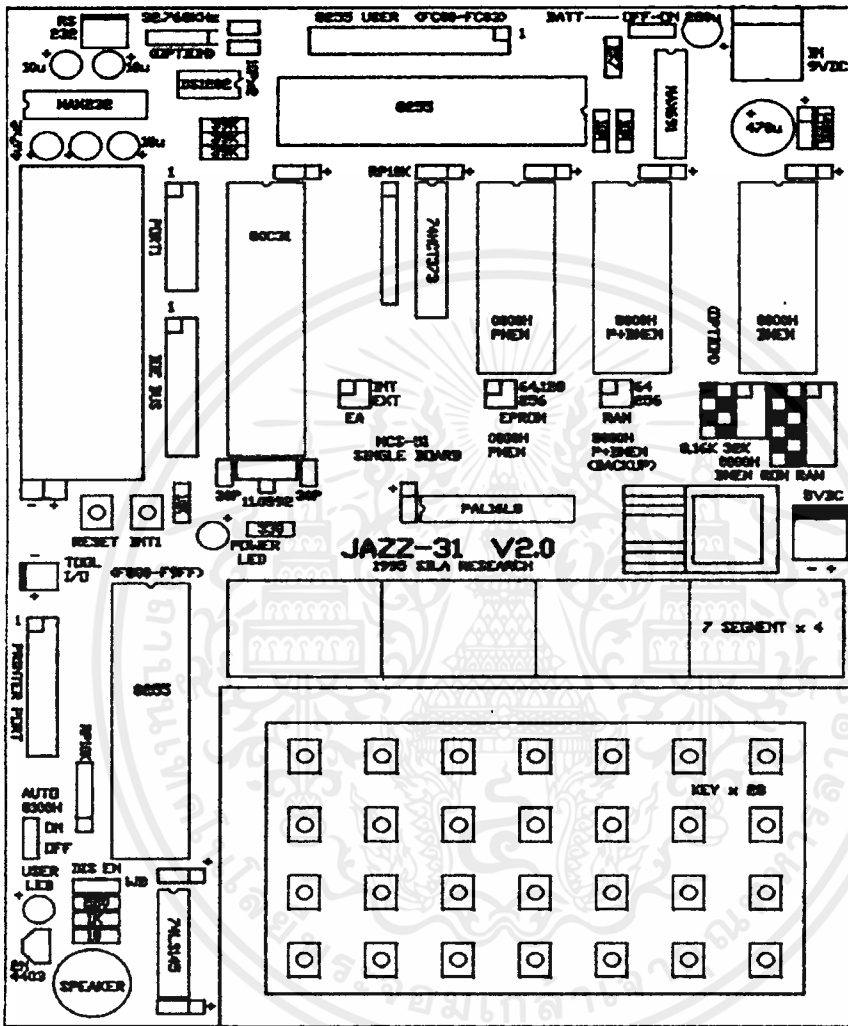
บอร์ดแอส 31 คือ บอร์ดคอมพิวเตอร์ขนาดเล็กที่ใช้เพื่อการพัฒนาทางด้าน อิเล็กทรอนิกส์และการเรียนรู้ทางด้านไมโครคอนโทรลเลอร์ซึ่งเป็นแควงของคอมพิวเตอร์ที่ใช้ในงานด้านอุตสาหกรรม และงานด้านอิเล็กทรอนิกส์ เช่น ระบบอัตโนมัติต่างๆ เครื่องควบคุมในโรงงาน แม้กระทั่งเครื่องใช้ไฟฟ้าต่างๆในปัจจุบัน ที่มีการนำเอาระบบไมโครคอนโทรลเลอร์ ขนาดเล็กมาใช้งาน

หัวใจสำคัญของบอร์ดแอส 31 คือชิพ 80C31 ซึ่งเป็นหนึ่งในตระกูลชิพเดี่ยว MCS-51 ผลิตโดยบริษัทอินเทลมีหลายรุ่นให้เลือกใช้งานจุดเด่นที่เลือกใช้ชิพตระกูล MCS-51 ก็เพราะมีคำสั่งที่สามารถทำงานในระดับบิตซึ่งมีความจำเป็นในการควบคุมโครงงานนี้ ให้เลือกใช้ได้หลายคำสั่ง

1.1 คุณสมบัติของแอส 31 (Hardware)

CPU	80C31 MICROCONTROLLER (CLOCK 11.0592 MHz)
INTERNAL RAM	128 BYTE
PROGRAM MEMORY	32/32 KBYTE 0000-7FFFH (2764/128/256)
DATA MEMORY	0/32 KBYTE 0000-F7FFFH (2764/128/256,6264/256)
PROGRAM + DATA	8/30 KBYTE 8000-F7FFFH (6264/256) WITH BACK-UP
DISPLAY	8 DIGIT 7-SEGMENT LED (RED) 2 LED (POWER/RED,USER/GREEN)
KEY BOARD	28 KEYS (DIRECT=8,HEX=16,OPERATE=4) 2 KEYS HARDWAR (RESET,INT1)
JUMPER	2-WAYS BATTERY (ON,OFF) 2-WAYS EA-PIN (INT,EXT) 2-WAYS AUTO 8100H (ON,OFF) 2-WAYS WATCH-DOG (DISABLE,ENABLE)
SOUND	ON-BOARD SPEAKER (1 BIT)
CONNECTOR	16-PIN PORT1

	20-PIN INPUT/OUTPUT EXPANSION (IOE BUS)
	26-PIN 8255 USER PORT (24 BIT I/O)
	20-PIN PRINTER PORT
	3-PIN SERIAL PORT (RS232)
	2-PIN TOOL INPUT/OUTPUT PORT
	2-PIN 5V DC SUPPLY (IN/OUT)
	2-PIN JACK 9V DC SUPPLY
OPTION	8-PIN RTC SOCKET (DS1202)
	28-PIN MEMORY SOCKET (6264,62256,2764,27128,27256)
PCB SIZE	5.5X7.5 INCH (WORKING AREA 0.8X2.3 INCH)



รูปที่ 3.1 ภาพแสดงตำแหน่งต่างๆบนบอร์ด

1.2 การใช้งานโปรแกรมมอนิเตอร์

โปรแกรมมอนิเตอร์คือ โปรแกรมที่อยู่ในตัว EPROM ซึ่งเป็นโปรแกรมจัดการระบบพื้นฐานทั้งหมดของการใช้งานไม่ว่าจะเป็นเรื่องของการป้อนข้อมูล, แก้ไขข้อมูล, การเข้าสู่ตีบ์กและการใช้งานตามฟังก์ชันต่างๆ ด้วย โปรแกรมมอนิเตอร์จะทำงานตลอดเวลา ที่ผู้ใช้งานใช้งานอยู่ ซึ่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บางที่ดูเหมือนจะเห็นไม่ชัดเจน แต่ในความจริงทุกการกระทำที่เกิดขึ้นมาจากโปรแกรมมอนิเตอร์ ทั้งสิ้นบอร์ดแอส 31 ใช้ โปรแกรมมอนิเตอร์ 17 KBYTE ซึ่งมีความยาวของโปรแกรมรวม 8300 บรรทัด ใช้ตัว EPROM 32 KBYTE ในการเก็บโปรแกรม โปรแกรมมอนิเตอร์เขียนจากภาษา แอสเซมบลี ของ 80C31 ปกติโปรแกรมมอนิเตอร์จะแสดงสถานะเตรียมพร้อมด้วยการแสดงเครื่องหมายขีด ว่างเป็นวงหมุนตามเข็มนาฬิกาอยู่ 2 วง ซึ่งเรียกว่า MONITOR MODE โดย จะรอการกด คีย์ต่างๆ เพื่อจะได้ใช้งานตามฟังก์ชันต่อไป

1.3 โหมดในการใช้งาน

1.3.1 DIRECT FUNCTION คือฟังก์ชันที่กดจากกลุ่มคีย์จำนวน 8 คีย์ทางด้านซ้าย (สีฟ้า) โดยทั่วไป DIRECT FUNCTION จะสามารถกดได้ทันทีที่ต้องการ (มียกเว้นบางกรณีเท่านั้น) โดย เมื่อกดแล้วจะเข้าสู่การใช้งานทันที ซึ่งทั้ง 8 คีย์ ของ DIRECT FUNCTION จะเป็นดังนี้

MON	สำหรับการอยู่ในสถานะเตรียมพร้อม
PREM	สำหรับการอ่านข้อมูลจากหน่วยความจำ ในส่วนที่เป็น PROGRAM MEMORY
DMEM	สำหรับการอ่านและเขียนข้อมูลกับหน่วยความจำ ในส่วนที่เป็น DATA MEMORY
INTR	สำหรับการอ่านและเขียนข้อมูลกับหน่วยความจำ ในส่วน INTERNAL RAM
SFR	สำหรับการอ่านและเขียนข้อมูลกับหน่วยความจำ ในส่วน SFR (SPECIAL FUNCTION REGISTOR) ซึ่งเป็นกลุ่ม REGISTOR พิเศษของ 80C31
DEBUG	สำหรับการทดสอบการทำงานตามโปรแกรมที่ผู้ใช้เขียนขึ้น
FUNC	สำหรับการเข้าสู่การใช้งานแบบ REMOTE MONITOR โดยต่อกับเครื่อง PC

1.3.2 SUB-FUNCTION คือฟังก์ชันเพื่ออำนวยความสะดวกต่างๆ ให้กับผู้ใช้ โดย SUB-FUNCTION จะมีหลากหลายรูปแบบในการใช้งาน ไม่ว่าจะเป็นการกระทำบน DATA MEMORY ซึ่งเป็นประโยชน์ต่อการพัฒนาโปรแกรม เช่น FIND, FILL, COMPARE-BLOCK, COMPARE-HEX รวมไปถึงด้านทูลต่างๆ เช่น STOP-WATCH, TIMER, COUNTER ซึ่งจะมีทั้งหมด 32 ฟังก์ชัน การเรียกใช้งานจะเรียกผ่านคีย์ FUNC (ซึ่งเป็น DIRECT FUNCTION) อีกที และจากนั้นจะต้อง เรียก SUB-FUNCTION ด้วยคีย์ 0-F โดยสนใจที่ข้อความตัวอักษรเล็กๆ (สีเขียว)

1.4 การใช้งาน EDIT KEYS

การใช้งานบนแผงคีย์ของแอส 31 จะมีลำดับจากซ้ายมาขวา โดยเริ่มจาก DIRECT FUNCTION และตามด้วยการป้อนข้อมูลต่างๆ จากคีย์ 0-F ควบคู่ไปกับการใช้งานในส่วน EDIT KEYS จำนวน 4 คีย์ จะใช้บ่อยมาก เปรียบเสมือนกับคีย์ลูกศร หรือคีย์ ENTER บนเครื่อง PC นั้นเอง เพราะฉะนั้น ในส่วนนี้จะได้อธิบายลักษณะของการใช้งานทั้ง 4 คีย์ดังนี้

แบบที่ 1 สำหรับการป้อนข้อมูลแอดเดรส หรือ HEX ต่างๆ ตามฟังก์ชันซึ่งจะมีลักษณะของส่วนแสดงผลคือ XXXX-X หรือ XX-X เช่น เมื่อผู้ใช้กดคีย์ DMEM เครื่องจะแสดงข้อความ 8100-d ซึ่งหมายถึงให้ป้อนเลขที่แอดเดรสก่อนที่จะเข้าไปสู่การใช้งานในโหมดของ DMEM ในระหว่างการป้อนข้อมูลนี้เอง ความหมายของคีย์ทั้ง 4 จะเป็นดังนี้

BACK สำหรับการย้อนกลับไป ก่อนหน้า ซึ่งถ้ามีการป้อนข้อมูลอยู่หลายส่วน การกดคีย์ BACK ก็คือการกลับไปป้อนข้อมูลในส่วนที่ผ่านมาแล้ว หรือเรียกได้ว่าเป็น BACK-FIELD ในกรณีที่ไม่สามารถย้อนได้ก็จะมีผลกระทบอะไร

ENT (ENTER) สำหรับการตกลงตามข้อมูลที่ป้อน และไปสู่การป้อนข้อมูลชุดต่อไป หรือเข้าสู่ทำงานต่างๆ ตามฟังก์ชันที่ต้องการ

UNDO สำหรับการยกเลิกข้อมูลที่กำลังทำการแก้ไข และนำค่าเดิมที่มีอยู่กลับคืนมา โดยทั้งนี้เพื่อกันความผิดพลาดจากการคีย์ที่ไม่ตั้งใจ

BCKD (BACK DIGIT) สำหรับการลบถอยหลัง 1 อักขร ทั้งนี้เพื่อความสะดวกในการแก้ไขตัวเลขที่ป้อนผิดพลาด

แบบที่ 2 สำหรับการใช้งานตามฟังก์ชันที่เกี่ยวข้องกับการอ่านและเขียนข้อมูลในหน่วยความจำ ซึ่งจะเกี่ยวข้องกับ PMEM ,INTR ,DMEM, SFR ซึ่งรายละเอียดเกี่ยวกับความหมายจะได้กล่าวในบทต่อไป ในที่นี้จะแสดงความหมายคร่าวๆ ของคีย์ทั้ง 4 ดังนี้

REFH (REFRESH) สำหรับการ ON/OFF ระบบ DATA-REFRESH

AUTOI (AUTO-INCREMENT) สำหรับการ ON/OFF ระบบ AUTO-INCREMENT

INC (INCREMENT) สำหรับการเพิ่มค่าแอดเดรสที่แสดงอยู่

DEC (DECREMENT) สำหรับการลดค่าแอดเดรสที่แสดงผลอยู่

1.4.1 ตัวสีแดงบนคีย์บอร์ด

อักษรสีแดงตัวเล็กๆ เป็นชื่อของ SFR จำนวน 16 ชื่อ อยู่บนคีย์ 0-F ซึ่งจะเกี่ยวข้องกับการใช้งานของคีย์ SFR และ ดับเบิลโดยเป็นการอ้างอิงรีจิสเตอร์ ส่วน SFR โดยเฉพาะ และคีย์อักษรด้านขวาจะมีอักษรสีแดงอยู่ 4 คีย์เช่นเดียวกัน ซึ่งเป็นการใช้งานของดักเพียงอย่างเดียว ดังนี้

RN สำหรับการแสดงค่ารีจิสเตอร์ R0-R7

RUN สำหรับการสั่งให้โปรแกรมที่ผู้ใช้เขียนขึ้นทำงาน

STEP สำหรับการสั่งให้โปรแกรมที่ผู้ใช้เขียนขึ้นทำงานแบบทีละคำสั่ง

SKIP สำหรับการสั่งให้โปรแกรมที่ผู้ใช้เขียนขึ้นทำงานแบบทีละคำสั่ง จนถึงแอดเดรสที่กำหนด แล้วจึงหยุด

1.4.2 การ REBOOT หรือ POWER UP

แฉล 31 มีความสามารถในการ REBOOT เครื่องได้ การ REBOOT คือการจำลองสภาพเสมือนว่าได้ทำการปิดเครื่องแล้วเปิดใหม่ ซึ่งบางทีในการใช้งานต่างๆ จำเป็นต้องทำการ REBOOT เพื่อการเริ่มต้นตัวแปรใหม่ หรือเพื่อทดสอบความสมบูรณ์ของโปรแกรมโดยเริ่มจากการเปิดเครื่อง ระบบ REBOOT ช่วยให้ความสะดวกแก่ผู้ใช้โดยที่ไม่ต้องดึงเข้าดึงออก JACK DC บ่อยๆ การ REBOOT จะกระทำดังนี้

1. กดคีย์ RESET ค้างเอาไว้
2. จากนั้นให้กดคีย์ INT1 ตามและค้างเอาไว้เช่นเดียวกัน
3. ปลดปล่อยคีย์ RESET ก่อน แล้วจึงปล่อยคีย์ INT1 ตาม
4. เครื่อง จะทำขบวนการทุกอย่างเหมือนกับการเปิดเครื่องครั้งแรก

1.5 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51

- ต้องการแหล่งจ่ายไฟ +5V ชุดเดียว
- มีหน่วยความจำโปรแกรม (Program Memory) ขนาด 4 กิโลไบต์สำหรับเบอร์

8051,8031

และ 8032 ไม่มีหน่วยความจำชุดนี้ ส่วน 8052 มีหน่วยความจำถึง 8 กิโลไบต์

- มีหน่วยความจำเก็บข้อมูล (Data Memory) ขนาด 128 ไบต์ สำหรับ 8052 มีถึง 256

ไบต์



- หน่วยความจำสำหรับโปรแกรมและตาต้า (Program and Data Memory แยกจากกันอย่างละ 64 กิโลไบต์)
- คำสั่งที่ใช้เวลาน้อยที่สุดประมาณ 1 μ s เมื่อทำงานที่ความถี่ 12 MHz
- มี Timer/Counter ขนาด 16 บิต 2 ชุด (สำหรับ 8052 มี 3 ชุด) ทำงานได้ 4 โหมด
- รับอินเตอร์รัพท์ได้ 6 แหล่ง 5 เวกเตอร์
- มีพอร์ทรับส่งข้อมูลอนุกรม (UART) 2 พอร์ท ทั้งรับและส่งในเวลาเดียวกันได้ (Full Duplex)
- เลือกรูปแบบการส่งข้อมูลได้ 4 โหมด
- มีคำสั่งในการทำ AND, OR และ Complement ได้ทั้งแบบ 8 บิตและ 1 บิต

1.6 พอร์ท 8255

8255 USER PORT คือพอร์ทขนาด 28 PIN ที่ต่อจาก 8255 พอร์ทโดยตรงใช้เพื่อการขยายระบบและการทดลองต่างๆ โดย USER PORT นี้จะไม่ถูกใช้งานจากตัวเครื่องเลย ผู้ใช้จะสามารถใช้งานได้อย่างอิสระ โดยมีจำนวน อินพุต/เอาต์พุต ให้ใช้ได้ถึง 28 บิต

ในโครงการนี้ เราใช้ 8255 พอร์ทในการส่งค่าให้กับวงจรปรับความเร็วของมอเตอร์ล้อยิง (โดยใช้พอร์ท A ของ 8255) และใช้เป็นอินพุตในการหยุดการทำงานของเครื่อง (โดยใช้ PORT C บิต 7 ในการตรวจสอบเงื่อนไขการหยุดการทำงาน

1.6.1 โหมดการทำงาน

การทำงานมีอยู่ด้วยกัน 3 โหมด

โหมด 0 มีการทำงานแบบ Basic I/O ไม่มี handshake

โหมด 1 โหมดนี้ใช้พอร์ท A,B ในการรับหรือส่งข้อมูล และใช้พอร์ท C ในการตรวจสอบสัญญาณ

โหมด 2 โหมดนี้ใช้พอร์ท A ในการรับส่งข้อมูล 2 ทิศทาง และ พอร์ท B ในการรับหรือส่งข้อมูลและใช้พอร์ท C บิต 0,1,2 ในการรับส่งข้อมูลบิตและบิต 4,5,6 เป็นสัญญาณ handshake

1.6.2 สัญญาณต่างๆ ของ 8255

D0-D7 บัสข้อมูลเชื่อมโยงกับ CPU

A0-A1 ใช้เลือกพอร์ท A,B,C และ พอร์ทควบคุม

RESET เมื่อขานี้ได้รับสัญญาณกระตุ้นลอจิก 1 จะทำให้ 8255 ถูกรีเซ็ตมีผลทำให้ทุกพอร์ทเป็นอินพุตทันที

PA0-PA7 เป็นพอร์ทขนาน 8 บิต

PB0-PB7 เป็นพอร์ทขนาน 8 บิต

PC0-PC7 เป็นพอร์ทขนาน 8 บิต

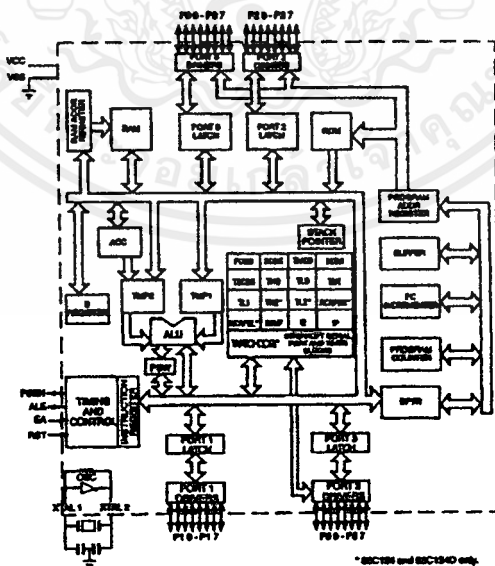
/RD ในการอ่านข้อมูลที่พอร์ทของ 8255 ต้องทำให้ขานี้เป็นลอจิก 0 พร้อมกับ /CS

WR ในการเขียนข้อมูลหรือโปรแกรมลงบน 8255 ต้องทำให้ขานี้เป็นลอจิก 0 พร้อมกับ /CS เมื่อ ขา WR, /RD, A0, A1, /CS ทำงานทั้ง 5 ขา จะมีฟังก์ชันการทำงาน

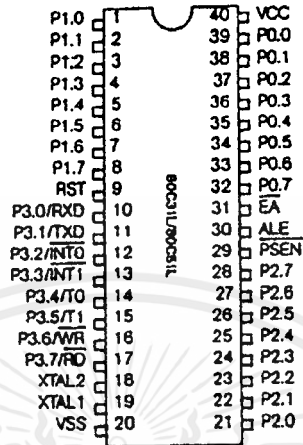
2. ชิพเบอร์ 8051

2.1 โครงสร้างของ 8051

MCS-51 ใช้เทคโนโลยีในการผลิตแบบ NMOS และ CMOS เบอร์ 8032 และ 8052 จะมี ROM BASIC อยู่ภายในจิ้งหรีด สำหรับโปรแกรมด้วยภาษาเบสิก



รูปที่ 3.2 โครงสร้างภายในสำหรับเบอร์ 8051



รูปที่ 3.3 การจัดวางขาของ 8051

2.2 พอร์ตของ 8051

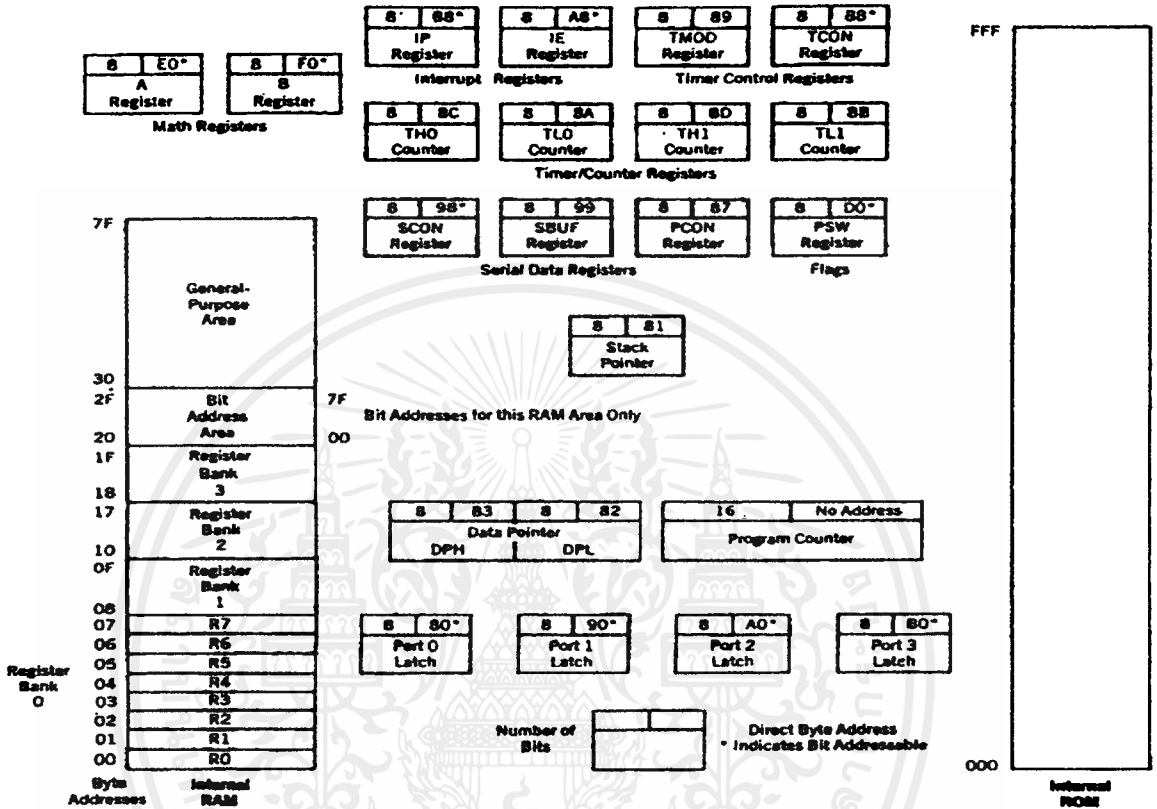
พอร์ต 0,2,3 จะทำหน้าที่เป็น อินพุต / เอาท์พุท พอร์ต 3 ยังมีหน้าที่พิเศษคือควบคุม การส่งข้อมูลอนุกรม ควบคุมการขัดจังหวะภายนอก (External Interrupt) การรับสัญญาณเวลา การอ่าน/เขียนหน่วยความจำข้อมูลภายนอก ส่วนพอร์ต 1 สามารถส่งข้อมูลได้อย่างเดียว

2.3 รีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Register ,SFR)

รีจิสเตอร์ฟังก์ชันพิเศษ เป็นรีจิสเตอร์ที่ใช้งานในหน้าที่ต่าง ๆ กัน ซึ่งบรรจุไว้ในหน่วย ความจำภายใน (Internal Ram) ตำแหน่ง 80H-FFH

แต่รีจิสเตอร์ฟังก์ชันพิเศษที่สำคัญที่สุดในโครงการนี้คือ Timer/Counter (ตำแหน่ง 8CH-8BH) ซึ่งเป็นหัวใจสำคัญของโปรแกรมควบคุมการทำงานของโครงการนี้

ในโครงการนี้เราใช้ Timer เป็นตัวสุ่มค่าขึ้นมาใช้ และใช้เป็นตัวหน่วงเวลาในการสร้าง สัญญาณ pulse



รูปที่ 3.4 แสดงรีจิสเตอร์ของ 8051

2.4 ชุดคำสั่งของ MCS-51

การทำงานของชิพ 80C31 มีทั้งหมด 256 คำสั่ง เป็นภาษาแอสเซมบลี สำหรับชิพเบอร์นี้ วิธีการแปลงภาษาแอสเซมบลีให้เป็นภาษาเครื่องสามารถดำเนินการได้ 2 แบบ คือ การป้อนรหัสภาษาเครื่องด้วยมือหรือใช้ Program Assembler เช่น SXA51, Cross 16, Cross 32 ฯลฯ โดยต้องเขียนโปรแกรมแล้วพิมพ์เก็บไว้ในไฟล์โดยตั้งชื่อให้มีความสมบูรณ์เป็นจุด ASM โปรแกรมที่ใช้พิมพ์อาจจะใช้ SK หรือ Wordstar ก็ได้ (โปรแกรม Editor นั้นเอง) การเขียนโปรแกรมภาษาแอสเซมบลีจะใช้รหัสช่วยจำ (Mnemonic Code) ซึ่งประกอบด้วย Operation Code, Operand และ Comment

2.4.1 ตัวดำเนินการ (Operation Code) ประกอบด้วย 2 ถึง 4 ตัวอักษร เช่น MOV, MOVX, MOVC, ADD ฯลฯ

2.4.2 ตัวถูกดำเนินการ (Operand) เป็นชุดอักษรที่บอกการดำเนินการ อาจจะเป็นชุดเดียวหรือ 2 ชุดก็ได้ถ้าเป็น 2 ชุดจะคั่นด้วยเครื่องหมายคอมม่า (,) เช่น A,#20H เมื่อเขียนให้เต็มรูปแบบของภาษาแอสเซมบลี จะได้ดังนี้ MOV A,#20H หมายความว่า ให้นำค่า 20H เข้ารีจิสเตอร์ A

โดยที่ A หมายถึง ปลายทาง (Destination)

#20H หมายถึง ต้นทาง (Source)

โดยที่ Source จะต้องอยู่ทางขวามือ และ Destination จะอยู่ทางซ้ายมือ

2.4.3 ส่วนขยาย (Comment) เป็นส่วนขยายความหมายของชุดรหัสคำสั่งจะนำหน้าด้วยเครื่องหมาย ";" ในส่วนของ Comment จะมีหรือไม่มีก็ได้

2.5 การแบ่งกลุ่มคำสั่งของ MCS-51

เราสามารถแบ่งกลุ่มคำสั่งได้ 5 กลุ่มใหญ่ๆ ดังนี้

- กลุ่มคำสั่งทางคณิตศาสตร์
- กลุ่มคำสั่งทางตรรกะ (Logic)
- กลุ่มคำสั่งควบคุมลำดับการทำงาน
- กลุ่มคำสั่งการประมวลผลแบบบิต

นอกจากนี้ยังแบ่งตามวิธีการเข้าถึงข้อมูล (Addressing Mode) ได้ดังนี้

- Direct Addressing เช่น MOV A,7FH
- Indirect Addressing เช่น MOV A,@R1
- Immediate Addressing เช่น MOV A,#7FH
- Register Addressing เช่น MOV A,R7

80C31/80C51

INSTRUCTION OPCODES IN HEXADECIMAL ORDER

HEX CODE	NUMB. OF BYTES	MNEM.	OPERANDS	HEX CODE	NUMB. OF BYTES	MNEM.	OPERANDS
00	1	NOP		33	1	RLC	A
01	2	AJMP	code addr	34	2	ADDC	A, #data
02	3	LJMP	code addr	35	2	ADDC	A, data addr
03	1	RR	A	36	1	ADDC	A, @RD
04	1	INC	A	37	1	ADDC	A, @R1
05	2	INC	data addr	38	1	ADDC	A, R0
06	1	INC	@R0	39	1	ADDC	A, R1
07	1	INC	@R1	3A	1	ADDC	A, R2
08	1	INC	R0	3B	1	ADDC	A, R3
09	1	INC	R1	3C	1	ADDC	A, R4
0A	1	INC	R2	3D	1	ADDC	A, R5
0B	1	INC	R3	3E	1	ADDC	A, R6
0C	1	INC	R4	3F	1	ADDC	A, R7
0D	1	INC	R5	40	2	JC	code addr
0E	1	INC	R6	41	2	AJMP	code addr
0F	1	INC	R7	42	2	ORL	data addr A
10	3	JBC	bit addr, code addr	43	3	ORL	data addr, #data
11	2	ACALL	code addr	44	2	ORL	A, #data
12	3	LCALLR	code addr	45	2	ORL	A, data addr
13	1	RC	A	46	1	ORL	A, @R0
14	1	DEC	A	47	1	ORL	A, @R1
15	2	DEC	data addr	48	1	ORL	A, R0
16	1	DEC	@R0	49	1	ORL	A, R1
17	1	DEC	@R1	4A	1	ORL	A, R2
18	1	DEC	R0	4B	1	ORL	A, R3
19	1	DEC	R1	4C	1	ORL	A, R4
1A	1	DEC	R2	4D	1	ORL	A, R5
1B	1	DEC	R3	4E	1	ORL	A, R6
1C	1	DEC	R4	4F	1	ORL	A, R7
1D	1	DEC	R5	50	2	JNC	code addr
1E	1	DEC	R6	51	2	ACALL	code addr
1F	1	DEC	R7	52	2	ANL	data addr, A
20	3	JB	bit addr, code addr	53	3	ANL	data addr, #data
21	2	AJMP	code addr	54	2	ANL	A, #data
22	1	RET		55	2	ANL	A, data addr
23	1	RL	A	56	1	ANL	A, @R0
24	2	ADD	A, data	57	1	ANL	A, @R1
25	2	ADD	A, data addr	58	1	ANL	A, R0
26	1	ADD	A, @R0	59	1	ANL	A, R1
27	1	ADD	A, @R1	5A	1	ANL	A, R2
28	1	ADD	A, R0	5B	1	ANL	A, R3
29	1	ADD	A, R1	5C	1	ANL	A, R4
2A	1	ADD	A, R2	5D	1	ANL	A, R5
2B	1	ADD	A, R3	5E	1	ANL	A, R6
2C	1	ADD	A, R4	5F	1	ANL	A, R7
2D	1	ADD	A, R5	60	2	JZ	code addr
2E	1	ADD	A, R6	61	2	AJMP	code addr
2F	1	ADD	A, R7	62	2	XRL	data addr A
30	3	JNB	bit, addr, code	63	3	XRL	data addr, #data
31	2	ACALL	addr	64	2	XRL	A, #data
32	1	RETI	code addr	65	2	XRL	A, data addr

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

80C31/80C51

HEX CODE	NUMB. OF BYTES	MNEM.	OPERANDS	HEX CODE	NUMB. OF BYTES	MNEM.	OPERANDS
66	1	XRL	A, @R0	99	1	SUBB	A, R1
67	1	XRL	A, @R1	9A	1	SUBB	A, R2
68	1	XRL	A, R0	9B	1	SUBB	A, R3
69	1	XRL	A, R1	9C	1	SUBB	A, R4
6A	1	XRL	A, R2	9D	1	SUBB	A, R5
6B	1	XRL	A, R3	9E	1	SUBB	A, R6
6C	1	XRL	A, R4	9F	1	SUBB	A, R7
6D	1	XRL	A, R5	A0	2	ORL	C, bit addr
6E	1	XRL	A, R6	A1	2	AJMP	code addr
6F	1	XRL	A, R7	A2	2	MOV	C, bit addr
70	2	JNZ	code addr	A3	1	INC	DPTR
71	2	ACALL	code addr	A4	1	MUL	AB
72	2	ORL	C, bit addr	A5		reserved	
73	1	JMP	@A + DPTR	A6	2	MOV	@R0, data addr
74	2	MOV	A, #data	A7	2	MOV	@R1, data addr
75	3	MOV	data addr, #data	A8	2	MOV	R0, data addr
76	2	MOV	@R0, #data	A9	2	MOV	R1, data addr
77	2	MOV	@R1, #data	AA	2	MOV	R2, data addr
78	2	MOV	R0, #data	AB	2	MOV	R3, data addr
79	2	MOV	R1, #data	AC	2	MOV	R4, data addr
7A	2	MOV	R2, #data	AD	2	MOV	R5, data addr
7B	2	MOV	R3, #data	AE	2	MOV	R6, data addr
7C	2	MOV	R4, #data	AF	2	MOV	R7, data addr
7D	2	MOV	R5, #data	B0	2	ANL	C, bit addr
7E	2	MOV	R6, #data	B1	2	ACALL	code addr
7F	2	MOV	R7, #data	B2	2	CPL	bit addr
80	2	SJMP	code addr	B3	1	CPL	C
81	2	AJMP	code addr	B4	3	CJNE	A, #data, code addr
82	2	ANL	C, bit addr	B5	3	CJNE	A, data addr, code addr
83	1	MOVC	A, @A + PC	B6	3	CJNE	@R0, #data, code addr
84	1	DIV	AB	B7	3	CJNE	@R1, #data, code addr
85	3	MOV	data addr, data addr	B8	3	CJNE	R0, #data, code addr
86	2	MOV	data addr, @R0	B9	3	CJNE	R1, #data, code addr
87	2	MOV	data addr, @R1	BA	3	CJNE	R2, #data, code addr
88	2	MOV	data addr, R0	BB	3	CJNE	R3, #data, code addr
89	2	MOV	data addr, R1	BC	3	CJNE	R5, #data, code addr
8A	2	MOV	data addr, R2	BD	3	CJNE	R4, #data, code addr
8B	2	MOV	data addr, R3	BE	3	CJNE	R6, #data, code addr
8C	2	MOV	data addr, R4	BF	3	CJNE	R7, #data, code addr
8D	2	MOV	data addr, R5	C0	2	PUSH	data addr
8E	2	MOV	data addr, R6	C1	2	AJMP	code addr
8F	2	MOV	data addr, R7	C2	2	CLR	bit addr
90	3	MOV	DPTR, #data	C3	1	CLR	C
91	2	ACALL	code addr	C4	1	SWAP	A
92	2	MOV	bit addr, C	C5	2	XCH	A, data addr
93	1	MOVC	A, @A + DPTR	C6	1	XCH	A, @R0
94	2	SUBB	A, #data	C7	1	XCH	A, @R1
95	2	SUBB	A, data addr	C8	1	XCH	A, R0
96	1	SUBB	A, @R0	C9	1	XCH	A, R1
97	1	SUBB	A, @R1	CA	1	XCH	A, R2
98	1	SUBB	A, R0	CB	1	XCH	A, R3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

80C31/80C51

HEX CODE	NUMB. OF BYTES	MNEM.	OPERANDS
CC	1	XCH	A, R4
CD	1	XCH	A, R5
CE	1	XCH	A, R6
CF	1	XCH	A, R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	data addr, code addr
D6	1	XCHD	A, @R0
D7	1	XCHD	A, @R1
D8	2	DJNZ	R0, code addr
D9	2	DJNZ	R1, code addr
DA	2	DJNZ	R2, code addr
DB	2	DJNZ	R3, code addr
DC	2	DJNZ	R4, code addr
DD	2	DJNZ	R5, code addr
DE	2	DJNZ	R6, code addr
DF	2	DJNZ	R7, code addr
E0	1	MOVX	A, @DPTR
E1	2	AJMP	code addr
E2	1	MOVX	A, @R0
E3	1	MOVX	A, @R1
E4	1	CLR	A
E5	2	MOV	A, data addr

HEX CODE	NUMB. OF BYTES	MNEM.	OPERANDS
E6	1	MOV	A, @R0
E7	1	MOV	A, @R1
E8	1	MOV	A, R0
E9	1	MOV	A, R1
EA	1	MOV	A, R2
EB	1	MOV	A, R3
EC	1	MOV	A, R4
ED	1	MOV	A, R5
EE	1	MOV	A, R6
EF	1	MOV	A, R7
F0	1	MOVX	@DPTR, A
F1	2	ACALL	code addr
F2	1	MOVX	@R0, A
F3	1	MOVX	@R1, A
F4	1	CPL	A
F5	2	MOV	data addr, A
F6	1	MOV	@R0, A
F7	1	MOV	@R1, A
F8	1	MOV	R0, A
F9	1	MOV	R1, A
FA	1	MOV	R2, A
FB	1	MOV	R3, A
FC	1	MOV	R4, A
FD	1	MOV	R5, A
FE	1	MOV	R6, A
FF	1	MOV	R7, A

ORDERING INFORMATION

A Q L I M	R P S D J V T	80C31 80C51 80C51F	xxx	- L - S - 1 /B	: D : R
Temperature Range blank : Commercial I : Industrial M : Military L : Ind + BI* Q : Com + BI* A : Automotive	Package Type P : Plastic DIL S : PLCC D : Cardip R : LCC J : J leaded LCC F : Quad Flat Pack V : S Quad Flat Pack (1.4 mm) T : T Quad Flat Pack (1.0 mm)**	Part Number 80C51 Rom 4 K x 8 80C31 External Rom 80C51F Secret ROM version	Customer Rom Code (80C51 only)	Blank : 12 MHz version - 1 : 16 MHz version - S : 20 MHz version - L : Low power. .B : Military Program.	: R Tape and Reel : D Dry Pack
* BI : Burn-In ** : Preliminary					

รูปที่ 3.5 แสดงชุดคำสั่งของไมโครคอนโทรลเลอร์ MCS-51

บทที่ 4

อุปกรณ์ควบคุมการส่งลูกเทเบิลเทนนิส

1. วงจรควบคุมความเร็ว DC Motor

วงจรควบคุมความเร็วมอเตอร์นี้ใช้ควบคุมความเร็วของมอเตอร์ยี่ห้อบลูบอด ซึ่งสามารถปรับระดับความเร็วได้ 16 ระดับ(ใช้ลอจิก 4 บิต)

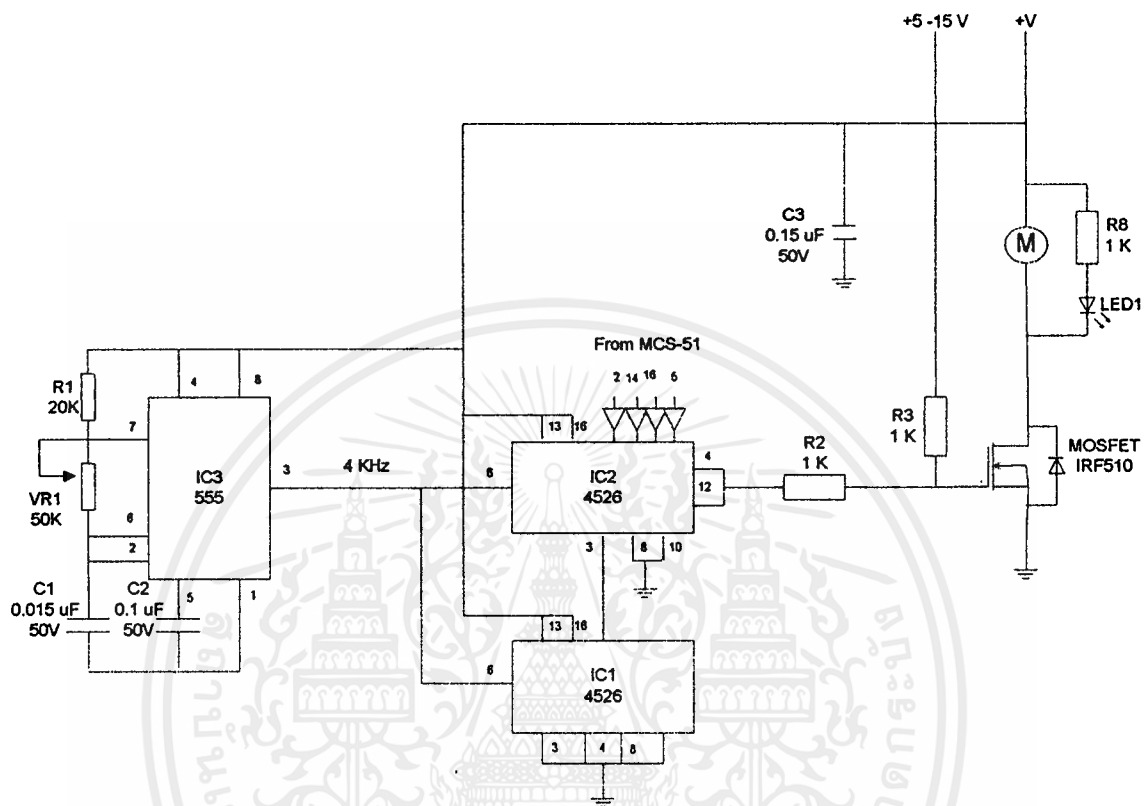
1.1 หลักการทำงาน

เมื่อได้รับสัญญาณขนาด 4 บิตจาก ชิพ ค่าที่ส่งออกมาจะถูกเก็บไว้ใน IC เบอร์ 4526 (IC2) IC เบอร์ 555 จะเป็น IC ที่กำเนิดสัญญาณ pulse ให้กับ IC 4526 ทั้งสองตัว IC1 ทำหน้าที่นับวนจาก 15 ถึง 0 เมื่อนับลงถึง 0 ก็จะส่งสัญญาณให้ IC2 เริ่มโหลดค่าที่ได้รับแล้วนับลง สัญญาณ pulse ที่ออกจากขา IC2 จะทำให้ MOSFET IRF 510 เปิด-ปิด ด้วยช่วงเวลาที่แตกต่างกันทำให้ MOTOR หมุนด้วยความเร็วที่แตกต่างกัน 16 ระดับ

1.2 รายการอุปกรณ์

ตัวต้านทาน	1/4 watt +/- 5 %
VR1	50 k
volumeเกือกม้านอนตัวเล็ก	1 ตัว
R1	1 ตัว
R2 , R3	2 ตัว
ตัวเก็บประจุ	
C1	50 volt 1 ตัว
C2	50 volt 1 ตัว
C3	50 volt 1 ตัว
อุปกรณ์สารกึ่งตัวนำ	
FET 1 - IRF 510	1 ตัว
IC1 , IC2 - 4526	2 ตัว
IC3 - 555	1 ตัว

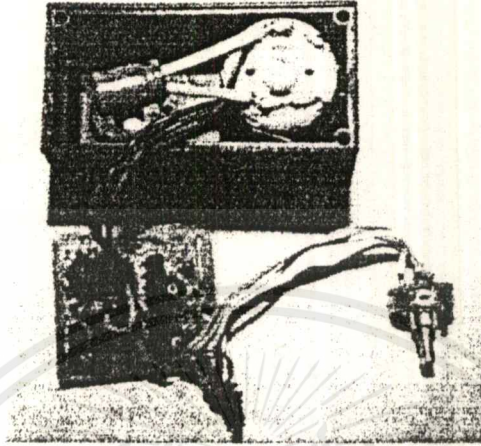
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 รูปวงจร

2. เซอร์โวมอเตอร์

ประกอบเซอร์โวมอเตอร์เป็นอุปกรณ์ที่ใช้ควบคุมตำแหน่งเชิงกลบอลในโครงการนี้ภายในด้วย DC มอเตอร์ 1 ตัว วงจรควบคุมตำแหน่งของมอเตอร์ และชุด Gear Box เพื่อทดรอบลงและเพิ่มแรงบิด



รูปที่ 4.2 แสดงวงจรควบคุมตำแหน่งเซอร์โวมอเตอร์

การทศรอบของเซอร์โวมอเตอร์ สามารถทศรอบลงได้ถึง 143:1 เซอร์โวมอเตอร์ใช้เพียง Vcc 5 Volt 1 แหล่ง และ 1 บิตจาก Port ของ ไมโครคอนโทรลเลอร์ เพื่อป้อน pulse ให้กับเซอร์โวมอเตอร์



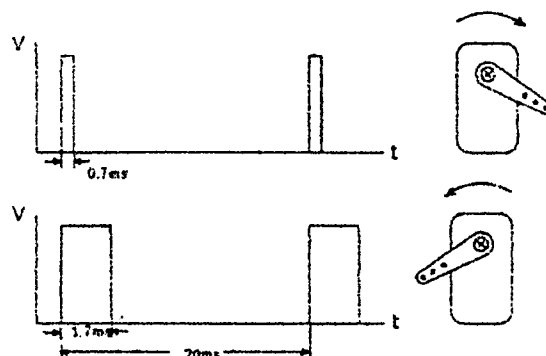
รูปที่ 4.3 แสดงการทศรอบโดยใช้เฟืองของเซอร์โวมอเตอร์

2.1 คุณสมบัติของเซอร์โวมอเตอร์

Control system	+ pulse control 1520 μ s neutral
Operation angleRotary system, one side 15 inch or greater (including trim).
Power supply	4.8 V or 6.0 V (shared with receiver)
Power consumption.....	6.0 V 8mA (at idle)
Output torque	42 oz/in (3kg,cm)
Operating speed.....	0.22 sec/60"
Dimensions.....	1.59x0.77x1.4 inch (40.4x19.8x36m/m)
Weight.....	1.52 oz(45.1g)

2.2 การควบคุมทิศทางของเซอร์โวมอเตอร์

โดยทั่วไปเซอร์โวมอเตอร์รับพัลส์ต่อเนื่องกัน โดยมีคาบเวลาที่เท่ากันคือ 20 ms ทิศทางการของเซอร์โวมอเตอร์ ถูกควบคุมความกว้าง ความแคบของพัลส์ที่มีค่าลอจิกเป็น 1 เซอร์โวมอเตอร์ จะอยู่ตำแหน่งตรงกลางเมื่อลอจิก 1 มีช่วงเวลากว้าง 1.52 ms หันซ้ายสุดเมื่อลอจิก 1 กว้าง 1.02 ms และหันขวาสุดเมื่อลอจิก 1 กว้าง 2.02 ms โดยทั้งหมดจะมีคาบเท่ากันคือ 20 ms



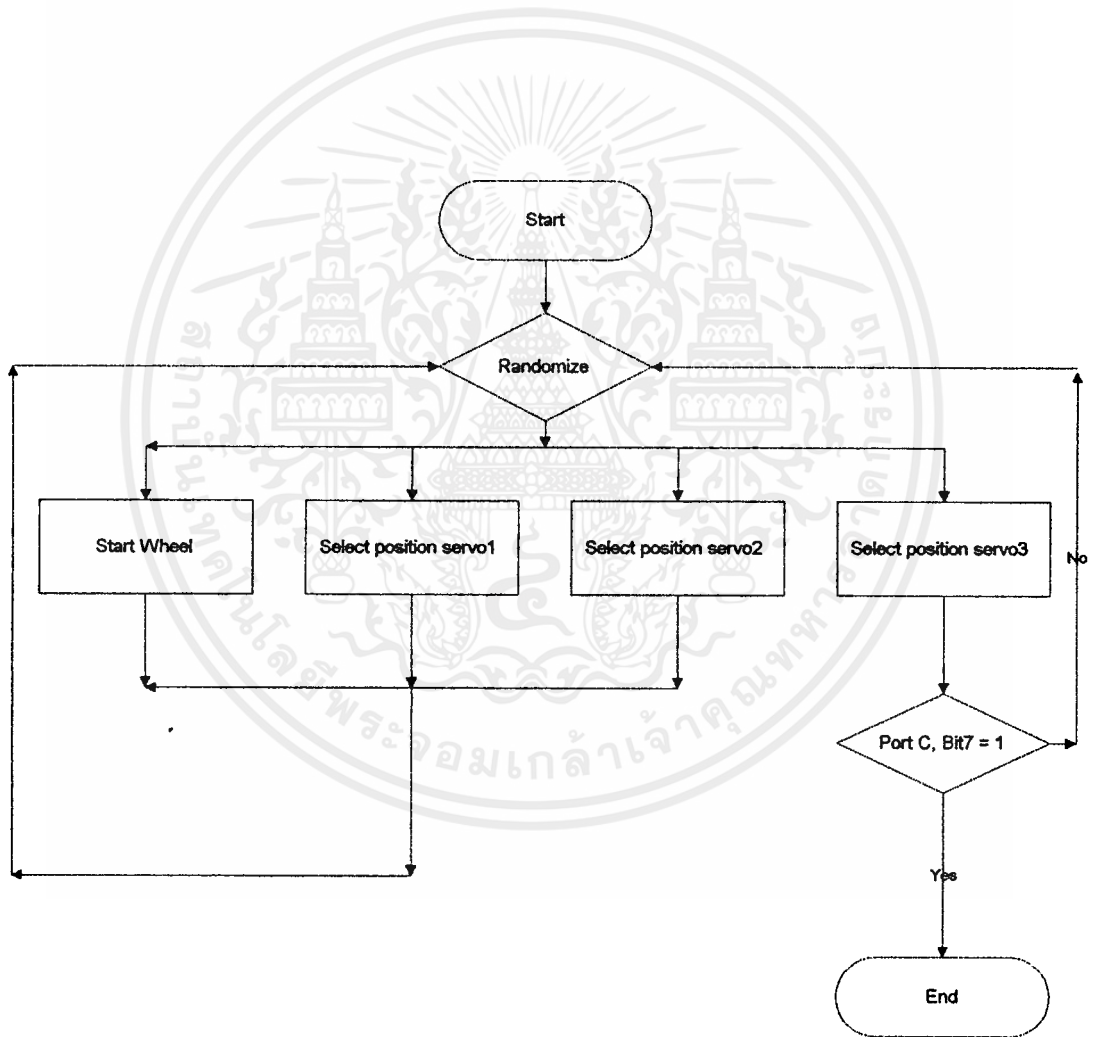
รูปที่ 4.4 แสดงทิศทางการเคลื่อนที่ของเซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

หลักการทำงานของหุ่นยนต์ส่งลูกเทเบิลเทนนิส

1. ลำดับขั้นตอนการทำงาน



รูปที่ 5.1 ภาพแสดงลำดับขั้นตอนการทำงาน (FLOW CHART)

2. ขั้นตอนการทำงานของเครื่องยิงลูกเทเบิลเทนนิส

2.1 ไมโครคอนโทรลเลอร์สั่งให้มอเตอร์ยิงลูกหมุนที่ความเร็วที่ต้องการ

2.2 ไมโครคอนโทรลเลอร์สั่งให้เซอร์โวมอเตอร์ที่ควบคุมการหมุนซ้าย-ขวา (Rolling) ทำงานถึงตำแหน่งที่ต้องการแล้วหยุดการทำงาน

2.3 ไมโครคอนโทรลเลอร์สั่งให้เซอร์โวมอเตอร์ที่ควบคุมการหมุนกระบอกยิง (Yawing) ทำงานถึงตำแหน่งที่ต้องการแล้วหยุดการทำงาน

2.4 ไมโครคอนโทรลเลอร์สั่งให้เซอร์โวมอเตอร์ที่ควบคุมการก้ม-เงย (Pitching) ของกระบอกยิงทำงานถึงตำแหน่งที่ต้องการแล้วหยุดการทำงาน

2.5 ไมโครคอนโทรลเลอร์เริ่มสั่งให้มอเตอร์ป้อนลูกทำงาน เมื่อลูกถูกยิงออกไป 1 ลูก ก็จะหยุดการป้อนลูก

2.6 การทำงานย้อนกลับไปที่ข้อ 1

บทที่ 6

ข้อสรุปและข้อเสนอแนะ

1. ปัญหาของการทำงานของหุ่นยนต์ส่งลูกเทเบิลเทนนิส

1.1 ปัญหาทางกล

ปัญหาทางกลของเครื่องส่งลูกเทเบิลเทนนิสเกิดขึ้นเมื่อ

1.1.1 มีการเคลื่อนที่เปลี่ยนตำแหน่งของเซอร์โวมอเตอร์ โดยเซอร์โวมอเตอร์จะส่งกำลังผ่านคันชัก การเคลื่อนที่แต่ละครั้งจะเกิดความหลวมขึ้น ทำให้ตำแหน่งของการเคลื่อนที่ทั้งสามแนวแกนเกิดการผิดพลาด ซึ่งเกิดจากความยืดหยุ่นของคันชัก

1.1.2 ความเร็วจากการเคลื่อนไหวของเซอร์โวมอเตอร์ ทำให้การเคลื่อนที่ของตัวหุ่นในแต่ละแนวแกนไม่ราบเรียบ

1.1.3 วงล้อยังไม่ได้ศูนย์ ทำให้เกิดการสั่นสะเทือนขณะหมุน

1.2 ปัญหาทางโปรแกรมการควบคุมการทำงาน

ปัญหาทางโปรแกรมควบคุมการทำงานที่สำคัญคือ ตัวบอร์ดแอส 31 ซึ่งเป็นอุปกรณ์ที่ใช้เก็บ และประมวลผลโปรแกรมมีความเชื่อถือได้ในผลิตภัณฑ์ต่ำ ทำให้เกิดปัญหาในการเขียนโปรแกรมควบคุมมาก คือมักเกิดความเสียหายของข้อมูลได้ง่าย เมื่อขาของพอร์ท 1 ขาใดๆ เกิดลัดวงจรถึงกัน

2. ข้อเสนอแนะ

2.1 ควรให้มีการจัดให้เป็นหัวข้อปริญญานิพนธ์ต่อเนื่อง ต่ออีกในปีการศึกษาถัดไป โดยกำหนดขอบเขตการศึกษา ให้หาวิธีที่ทำให้สามารถกำหนดตำแหน่งการตกของลูกเทียงตรง และกำหนดได้ว่าจะให้ลูกตกลงตำแหน่งไหนของโต๊ะ

2.2 ถ้ามีการจัดให้เป็นหัวข้อปริญญานิพนธ์ในปีการศึกษาถัดไป ขอให้เพิ่มจำนวนนักศึกษาที่ทำให้มีสามคนเป็นอย่างน้อย โดยแบ่งขอบเขตการศึกษาเป็นสามส่วนคือ ส่วนทางกล ส่วนทางอิเล็กทรอนิกส์ และส่วนทางโปรแกรม

บรรณานุกรม

1. ผศ. สมยศ จุณณะปิยะ "การใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51" คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
2. Joseph L. Jones "Mobile Robots Inspiration to Implementation" Anita M. Flynn Artificial Intelligence Laboratory Massachusetts Institute of Technology
3. "คู่มือการใช้งาน บอร์ด แจส 31 Version 2.0" ของบริษัท ศิลาริเสิร์ช จำกัด
4. เซมิคอนดักเตอร์ อิเล็กทรอนิกส์ เล่มที่ 172 เรื่อง ดิจิตอลพัลส์วิดิธมอดูเลเตอร์ ปรับความเร็วดีซี มอเตอร์



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



โปรแกรมควบคุมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ORG 8100H

W EQU 01H

X EQU 02H

Y EQU 03H

Z EQU 04H

V EQU 06H

CLR EA

MOV TMOD,#11H

MOV TH0,W

MOV TL0,Z

SETB TR0

MOV R6,#01H

MOV R7,#00H

MOV R3,#01H

MOV R0,#01H

SERVO1: SJMP SBB0

SBB0: CJNE R6,#01H,SBB1

DATA0: MOV W,#0FBH

MOV X,#03DH

MOV Y,#0DBH

MOV Z,#013H

MOV R6,#02H

AJMP S1

SBB1: CJNE R6,#03H,SBB2

DATA1: MOV W,#0FAH

MOV X,#021H

MOV Y,#0DCH

MOV Z,#03EH

MOV R6,#04H

AJMP S1

SBB2: CJNE R6,#05H,SBB3

DATA2: MOV W,#0FAH

MOV X,#080H

MOV Y,#0DBH

MOV Z,#0D0H

MOV R6,#06H

AJMP S1

SBB3: CJNE R6,#07H,SBB4

DATA3: MOV W,#0FAH

MOV X,#0BEH

MOV Y,#0DBH

MOV Z,#072H

MOV R6,#08H

AJMP S1

SBB4: CJNE R6,#02H,SBB5

DATA4: MOV W,#0F9H

MOV X,#065H

MOV Y,#0DCH

MOV Z,#0EBH

MOV R6,#03H

AJMP S1

SBB5: CJNE R6,#04H,SBB6

DATA5: MOV W,#0F9H

MOV X,#06H

MOV Y,#0DDH

MOV Z,#04AH

MOV R6,#05H

AJMP S1

SBB6: CJNE R6,#06H,SBB7

DATA6: MOV W,#0F9H

MOV X,#0C3H

MOV Y,#0DCH

MOV Z,#08DH

MOV R6,#07H

AJMP S1

SBB7: SJMP DATA7
DATA7: MOV W,#0FAH
MOV X,#021H
MOV Y,#0DCH
MOV Z,#03EH
MOV R6,#01H
AJMP S1

SERVO2: SJMP SB0
SB0 : CJNE R7,#01H,SB1
DAT0: MOV W,#0F8H
MOV X,#0C8H
MOV Y,#0DDH
MOV Z,#0A8H
MOV R7,#02H
AJMP S2

SB1: CJNE R7,#03H,SB2
DAT1: MOV W,#0F7H
MOV X,#08DH
MOV Y,#0DEH
MOV Z,#0C3H
MOV R7,#04H
AJMP S2

SB2: CJNE R7,#05H,SB3

DAT2: MOV W,#0F9H

MOV X,#0C3H

MOV Y,#0DCH

MOV Z,#08DH

MOV R7,#06H

AJMP S2

SB3: CJNE R7,#07H,SB4

DAT3: MOV W,#0F6H

MOV X,#0FFH

MOV Y,#0DFH

MOV Z,#051H

MOV R7,#08H

AJMP S2

SB4: CJNE R7,#09H,SB5

DAT4: MOV W,#0F9H

MOV X,#065H

MOV Y,#0DCH

MOV Z,#0EBH

MOV R7,#0AH

AJMP S2

SB5: CJNE R7,#0BH,SB6

DAT5: MOV W,#0F8H

MOV X,#04AH

MOV Y,#0DEH

MOV Z,#06H

MOV R7,#0CH

AJMP S2

SB6: CJNE R7,#0DH,SB7

DAT6: MOV W,#0F7H

MOV X,#0EBH

MOV Y,#0DEH

MOV Z,#065H

MOV R7,#0EH

AJMP S2

SB7: CJNE R7,#0FH,SB8

DAT7: MOV W,#0F9H

MOV X,#06H

MOV Y,#0DDH

MOV Z,#04AH

MOV R7,#00H

AJMP S2

SB8: CJNE R7,#00H,SB9

DAT8: MOV W,#0FAH

MOV X,#021H

MOV Y,#0DCH

MOV Z,#03EH

MOV R7,#01H

AJMP S2

SB9: CJNE R7,#02H,SB10

DAT9: MOV W,#0FCH

MOV X,#058H

MOV Y,#0D9H

MOV Z,#0BCH

MOV R7,#03H

AJMP S2

SB10: CJNE R7,#04H,SB11

DAT10: MOV W,#0FAH

MOV X,#080H

MOV Y,#0DBH

MOV Z,#0D0H

MOV R7,#05H

AJMP S2

SB11: CJNE R7,#06H,SB12

DAT11: MOV W,#0FBH

MOV X,#03DH

MOV Y,#0DBH

MOV Z,#013H

MOV R7,#07H

AJMP S2

SB12: CJNE R7,#08H,SB13

DAT12: MOV W,#0FBH

MOV X,#0F9H

MOV Y,#0DAH

MOV Z,#056H

MOV R7,#09H

AJMP S2

SB13: CJNE R7,#0AH,SB14

DAT13: MOV W,#0FAH

MOV X,#021H

MOV Y,#0DCH

MOV Z,#03EH

MOV R7,#0BH

AJMP S2

SB14: CJNE R7,#0CH,SB15

DAT14: MOV W,#0FBH

MOV X,#098H

MOV Y,#0DAH

MOV Z,#0B5H

MOV R7,#0DH

AJMP S2

SB15: SJMP DAT15

DAT15: MOV W,#0FAH

MOV X,#0BEH

MOV Y,#0DBH

MOV Z,#072H

MOV R7,#0FH

SJMP S2

SERVO3: SJMP SO0

SO0: CJNE R0,#01H,SO1

DA0: MOV W,#0FAH

MOV X,#021H

MOV Y,#0DCH

MOV Z,#03EH

MOV R0,#02H

SJMP S3

SO1: CJNE R0,#02H,S02

DA1: MOV W,#0F9H

MOV X,#0C3H

MOV Y,#0DCH

MOV Z,#08DH

MOV R0,#03H

SJMP S3

SO2: CJNE R0,#03H,S03

DA2: MOV W,#0FAH

MOV X,#021H

MOV Y,#0DCH

MOV Z,#03EH

MOV R0,#04H

SJMP S3

SO3: CJNE R0,#04H,S4

DA3: MOV W,#0F9H

MOV X,#0C3H

MOV Y,#0DCH

MOV Z,#08DH

MOV R0,#05H

SJMP S3

S4: CJNE R0,#05H,S5

DA4: MOV W,#0F9H

MOV X,#065H

MOV Y,#0DCH

MOV Z,#0EBH

MOV R0,#06H

SJMP S3

S5: SJMP DA5

DA5: MOV W,#0F9H

MOV X,#065H

MOV Y,#0DCH

MOV Z,#0EBH

MOV R0,#01H

SJMP S3

S1: MOV R5,#40H

ACALL WHEEL

SJMP L0

S2: MOV R5,#40H

SJMP L1

S3: MOV R5,#40H

SJMP L2

```
L0:      SJMP LOOP0
LOOP0:   CLR P1.1
        MOV TH1,W
        MOV TL1,X
        SETB TR1
JUMP1:   JNB TF1,JUMP1
        CLR TR1
        CLR TF1
        SETB P1.1
        MOV TH1,Y
        MOV TL1,Z
        SETB TR1
JUMP2:   JNB TF1,JUMP2
        CLR TR1
        CLR TF1
        DJNZ R5,LOOP0
        AJMP SERVO2
```

```
L1:      SJMP LOOP1
LOOP1:   CLR P1.2
         MOV TH1,W
         MOV TL1,X
         SETB TR1
JUMP3:   JNB TF1,JUMP3
         CLR TR1
         CLR TF1
         SETB P1.2
         MOV TH1,Y
         MOV TL1,Z
         SETB TR1
JUMP4:   JNB TF1,JUMP4
         CLR TR1
         CLR TF1
         DJNZ R5,LOOP1
         AJMP SERV03
```

L2: SJMP LOOP2

LOOP2: CLR P1.4

LP2: CLR P1.3

MOV TH1,W

MOV TL1,X

SETB TR1

JUMP5: JNB TF1,JUMP5

CLR TR1

CLR TF1

SETB P1.3

MOV TH1,Y

MOV TL1,Z

SETB TR1

JUMP6: JNB TF1,JUMP6

CLR TR1

CLR TF1

DJNZ R5,LP2

SETB P1.4

SETB P1.3

AJMP SERVO1

A8255: MOV A,#80H

MOV DPTR,#0FC03H

MOVX @DPTR,A

RET

```
A8255C:  MOV DPTR,#0FC00H
          MOVX @DPTR,A
          MOV TL0,TH0
          MOV TH0,Z
          RET
```

```
WHEEL:  MOV V,TL0
          ANL V,#0FH
          MOV R2,V
```

```
LIMIT1: CJNE R1,#01H,LIMIT2
          AJMP GO0
```

```
LIMIT2: CJNE R1,#04H,G0
          AJMP GO0
```

```
G0:     CJNE R2,#00H,G1
```

```
ST0:   ACALL A8255
        MOV A,#0DH
        ACALL A8255C
        RET
```

```
G1:     CJNE R2,#01H,G2
```

```
ST1:   ACALL A8255
        MOV A,#0EH
        ACALL A8255C
```

```
RET
```

G2: CJNE R2,#02H,G3

ST2: ACALL A8255

MOV A,#0FH

ACALL A8255C

RET

G3: CJNE R2,#03H,G4

ST3: ACALL A8255

MOV A,#04H

ACALL A8255C

RET

G4: CJNE R2,#04H,G5

ST4: ACALL A8255

MOV A,#05H

ACALL A8255C

RET

G5: CJNE R2,#05H,G6

ST5: ACALL A8255

MOV A,#06H

ACALL A8255C

RET

G6: CJNE R2,#06H,G7

ST6: ACALL A8255

MOV A,#07H

ACALL A8255C

RET

G7: CJNE R2,#07H,G8

ST7: ACALL A8255

MOV A,#08H

ACALL A8255C

RET

G8: CJNE R2,#08H,G9

ST8: ACALL A8255

MOV A,#09H

ACALL A8255C

RET

G9: CJNE R2,#09H,G10

ST9: ACALL A8255

MOV A,#0AH

ACALL A8255C

RET



G10: CJNE R2,#0AH,G11

STA: ACALL A8255

MOV A,#0BH

ACALL A8255C

RET

G11: CJNE R2,#0BH,G12

STB: ACALL A8255

MOV A,#0CH

ACALL A8255C

RET

G12: CJNE R2,#0CH,G13

STC: ACALL A8255

MOV A,#02H

ACALL A8255C

RET

G13: CJNE R2,#0DH,G14

STD: ACALL A8255

MOV A,#03H

ACALL A8255C

RET

G14: CJNE R2,#0EH,G15

STE: ACALL A8255

MOV A,#04H

ACALL A8255C

RET

G15: ACALL A8255

MOV A,#05H

ACALL A8255C

RET

GO0: CJNE R2,#00H,GO1

STO0: ACALL A8255

MOV A,#0CH

ACALL A8255C

RET

GO1: CJNE R2,#01H,GO2

STO1: ACALL A8255

MOV A,#0BH

ACALL A8255C

RET

GQ2: CJNE R2,#02H,G03

STO2: ACALL A8255

MOV A,#07H

ACALL A8255C

RET

G03: CJNE R2,#03H,G04

STO3: ACALL A8255

MOV A,#04H

ACALL A8255C

RET

G04: CJNE R2,#04H,G05

STO4: ACALL A8255

MOV A,#05H

ACALL A8255C

RET

G05: CJNE R2,#05H,G06

STO5: ACALL A8255

MOV A,#06H

ACALL A8255C

RET

GO6: CJNE R2,#06H,GO7

STO6: ACALL A8255

MOV A,#07H

ACALL A8255C

RET

GO7: CJNE R2,#07H,GO8

STO7: ACALL A8255

MOV A,#08H

ACALL A8255C

RET

GO8: CJNE R2,#08H,GO9

STO8: ACALL A8255

MOV A,#09H

ACALL A8255C

RET

GO9: CJNE R2,#09H,GO10

STO9: ACALL A8255

MOV A,#0AH

ACALL A8255C

RET

GO10: CJNE R2,#0AH,GO11

STOA: ACALL A8255

MOV A,#0BH

ACALL A8255C

RET

GO11: CJNE R2,#0BH,GO12

STOB: ACALL A8255

MOV A,#0CH

ACALL A8255C

RET

GO12: CJNE R2,#0CH,GO13

STOC: ACALL A8255

MOV A,#02H

ACALL A8255C

RET

GO13: CJNE R2,#0DH,GO14

STOD: ACALL A8255

MOV A,#03H

ACALL A8255C

RET

GO14: CJNE R2,#0EH,G015

STOE: ACALL A8255

MOV A,#04H

ACALL A8255C

RET

GO15: ACALL A8255

MOV A,#05H

ACALL A8255C

RET

END

