



แขนกล

ROBOT ARM

โดย

นายประสงค์ วงศประสิทธิ์พร 35104250

อาจารย์ที่ปรึกษา

รศ.ดร.มนัส สัจวารศิลป์

วัน เดือน ปี.....	ค.ศ. ๒๕๕๐
เลขทะเบียน.....	๐๒๗๑๐
เลขเรียกหนังสือ.....	T ๐๒๒๐๓. ๒๕๕๗

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

037110

ปริญญาโท ปีการศึกษา 2538

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง แขนก

ผู้จัดทำ

นายประสงค์ วงศประสิทธิ์พร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## แขนกล

นายประสงค์ วงศประสิทธิ์พร  
รศ.ดร.มนัส สัจวรศิลป์ อาจารย์ที่ปรึกษา  
ปีการศึกษา 2538

### บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ กล่าวถึงการออกแบบเพื่อสร้างแขนกลเพื่อใช้งานอุตสาหกรรม โดยเป็นแขนกล 6 แขน ทำงานโดยวงจร INTERFACE ซึ่งทำงานโดย MICROCONTROLLER 8031 โดยรับคำสั่งจาก MICROCOMPUTER

จุดเด่นของแขนกล คือ การกำหนดแรงบีบของ GRIPPER ที่ใช้สัญญาณ FEEDBACK เพื่อไม่ให้วัตถุแตกร้าวได้และมีการตรวจการลื่นไถลของวัตถุที่หีบเพื่อเพิ่มแรงบีบขึ้นในกรณีที่แรงบีบที่กำหนดในครั้งแรกไม่เหมาะสม

# ROBOT ARM

Prasong Wongprasithiporn

Dr.Manus Sangvorasilp Advisor

1995

## ABSTRACT

This Thesis is mentioned about the Robot Arm that designed for industrial use , which is 6 axis robot arm. This Robot Arm work by Stepping Motor and use Microcontroller to control system by receiving the signal from Microcomputer.

The outstanding part of Robot Arm is forced determination of gripper. Use feedback signal to prevent object from fracture and in spect , slipper of object. It can increase force of object in case of the force is unsuitable.

## สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 แขนกล	2
2.1 ชนิดของหุ่นยนต์	2
2.2 มือของหุ่นยนต์	4
2.2.1 เมคคาทรอนิกส์กริปเปอร์	4
2.3 ความแม่นยำของการเคลื่อนที่	6
2.3.1 สเปซเวียนรีโซลูชัน	6
2.3.2 แอคทูเอที	7
2.3.3 รีพีททีบิลิตี	7
2.4 การเคลื่อนที่ของหุ่นยนต์	8
2.4.1 การเคลื่อนที่ของตัวและแขน	8
2.4.2 การเคลื่อนที่ของมือ	9
บทที่ 3 สเตปปีงมอเตอร์	10
3.1 วารีโอเบิลรีดักเตนซ์	12
3.2 เพอร์มานนท์แมกเนต	13
3.3 ไฮบริดสเตปเปอร์มอเตอร์	13
3.4 วิธีการกระตุ้นเฟส	14
บทที่ 4 การประกอบแขนกล	16
4.1 ส่วนฐาน	20
4.2 ส่วนหัวไหล่	21
4.3 ส่วนข้อศอก	22
4.4 ส่วนข้อมือ	23
4.5 ส่วนมือ	25
บทที่ 5 โครงสร้างของโรงงาน	26

5.1 อัลกอริทึมในการควบคุมแขนกล	27
5.2 สมการ Direct kinematic	28
5.2.1 การอ้างอิงมุมของแขนกล	29
5.3 สมการ Inverse kinematic	32
5.4 การประยุกต์ใช้งาน	37
<b>บทที่ 6 การควบคุมและโปรแกรมการควบคุม</b>	39
6.1 วงจรอินเทอร์เฟส	40
6.2 โปรแกรมการควบคุม	43
6.2.1 โปรแกรมภาษาซี	43
6.2.2 โปรแกรมภาษาแอสเซมบลี	45
<b>บทที่ 7 ผลการทดลอง</b>	46
ภาคผนวก	48
หนังสืออ้างอิง	



## สารบัญ

รูปที่	หน้า
<b>บทที่ 2</b>	
2.1 ชนิดของหุ่นยนต์ (a) โพลาร์ (b) ไชเลนดริคัล (c) คาร์ทีเซียน (d) ฮาร์ติคูเลต	3
2.2 มือจับที่ออกแบบให้มีรูปทรงคล้ายกับวัตถุ	5
2.3 มือจับที่อาศัยแรงเสียดทานของนิ้วกับวัตถุ	6
2.4 ความแม่นยำและความสามารถในการทำซ้ำของหุ่นยนต์	8
2.5 ลักษณะการเคลื่อนที่ของแขนแบบต่าง ๆ	9
2.6 การเคลื่อนที่ของมือ	9
<b>บทที่ 3</b>	
3.1 สเตปปีงมอเตอร์	11
3.2 การวางขดลวดในสเตปปีงมอเตอร์ชนิด 3 เฟส	11
3.3 วงจรการควบคุมสเตปปีงมอเตอร์	12
3.4 วงจรขับสเตปเปอร์มอเตอร์แบบต่าง ๆ	13
<b>บทที่ 4</b>	
4.1 แสดงชุดทดลอง	17
4.2 แสดงชุดแขนกล	19
4.3 แสดงส่วนฐาน	20
4.4 แสดงชุดหัวไหล่	21
4.5 แสดงชุดข้อศอก	22
4.6 แสดงส่วนข้อมือ	23
4.7 แสดงชุดมือจับ	24
4.8 แสดงชุดแขนกลและคอมพิวเตอร์	25
<b>บทที่ 5</b>	
5.1 บล็อกไดอะแกรมของแขนกล	26
5.2 ไดอะแกรมการทำงานของแขนกลในรูปแบบ 5 ดีกรีออฟฟรீดอม	28

5.3	บล็อกไดอะแกรมของสมการ direct kinematic	29
5.4	ตำแหน่งโฮมของแขนกล	29
5.5	แสดงตัวอย่างในการอ้างอิงมุม	30
5.6	ตำแหน่งด้านของแขนกลชนิดอาร์ติคูลेटแบบ 5 ดีกรีออฟฟรีดอม	30
5.7	บล็อกไดอะแกรมของสมการ inverse kinematic	33
5.8	แสดงอัลกอริทึมของสมการ inverse kinematic	34
5.9	แสดงการอ้างอิง $q_{234}$	36
5.10	ตัวอย่างการอ้างอิง $q_{234}$	36
5.11	บล็อกไดอะแกรมในการตรวจสอบตำแหน่งด้วยสมการ direct kinematic	38
<b>บทที่ 6</b>		
6.1	แสดงวงจรอินเตอร์เฟส	41
6.2	แสดงวงจรขับเคลื่อนมอเตอร์	42
6.3	แสดง flowchart การทำงานของโปรแกรมควบคุมบนเครื่อง PC	44
6.4	แสดง flowchart การทำงานของโปรแกรมควบคุมบน MCS-51	45

บทที่ 1

## บทนำ

คำว่า “หุ่นยนต์” นี้ค้นกำเนิดจริง ๆ แล้วเกิดขึ้นจากนิยายวิทยาศาสตร์แบบเพื่อฝันหรือที่เราเรียกว่า แฟนตาซี (fantasy) ของชาวเชคโกสโลวาเกีย (CZECHOSLOVAKIA) ในปี 1920 โดยทั่วไปแล้วพอพูดถึงคำว่า “หุ่นยนต์” คนทั่วไปมักจะคิดถึงเครื่องชนิดที่มีรูปร่างเหมือนมนุษย์ สามารถที่จะคิดและทำงานได้ด้วยตัวเอง แต่ที่จริงแล้วคำนิยาม ROBOTICS หมายถึงเครื่องชนิดที่ประกอบกันขึ้นมาเพื่อทำงานบางอย่างแทนมนุษย์ สามารถควบคุมได้โดยใช้ โปรแกรมทางคอมพิวเตอร์ และสามารถที่จะเปลี่ยนการทำงานให้ทำงานอย่างอื่น ๆ ได้ (ตามคำนิยามของ สมาคมหุ่นยนต์โรงงาน (Robotics Industrial Association, RIA) และสถาบันเทคโนโลยีเกี่ยวกับหุ่นยนต์ของอเมริกา (Robotics Institute of America, RIA) )

จากคำนิยามนี้จะเห็นว่าหุ่นยนต์ก็คือเครื่องชนิดอัตโนมัติชนิดหนึ่ง ที่สามารถเปลี่ยนโปรแกรมการทำงานได้ทำนั่นเอง เครื่องชนิดอัตโนมัติ (Automatics) นี้แบ่งออกได้เป็น 3 ชนิด คือ

1. **Fixed Automation** คือเครื่องชนิดอัตโนมัติที่ทำงานอยู่กับที่และทำงานอย่างเดิมตลอด
2. **Programmable Automation** คือเครื่องชนิดอัตโนมัติที่สามารถเปลี่ยนการทำงานได้หลายอย่าง แต่ต้องรองานที่ทำในตอนนั้นให้เสร็จก่อนจึงสามารถเปลี่ยนได้
3. **Flexible Automation** คือเครื่องชนิดอัตโนมัติที่สามารถเปลี่ยนการทำงานได้ และสามารถทำงานหลาย ๆ อย่างได้ในวงรอบการทำงานเดียวกัน

ในรายงานฉบับนี้จะเน้นที่หุ่นยนต์ในโรงงานอุตสาหกรรม (Industrial Robotics) ดังนั้นคำว่าหุ่นยนต์ หรือ Robotics ที่กล่าวถึง จะหมายถึงหุ่นยนต์ในโรงงานอุตสาหกรรมเท่านั้น

**บทที่ 2****แขนกล****2.1 ชนิดของหุ่นยนต์**

หุ่นยนต์แบ่งตามลักษณะการเคลื่อนที่และลักษณะของแขน แบ่งได้เป็น 4 ชนิดใหญ่ๆ คือ

1. Polar Configuration.
2. Cylindrical Configuration.
3. Cartesian Configuration.
4. Articulated Configuration.

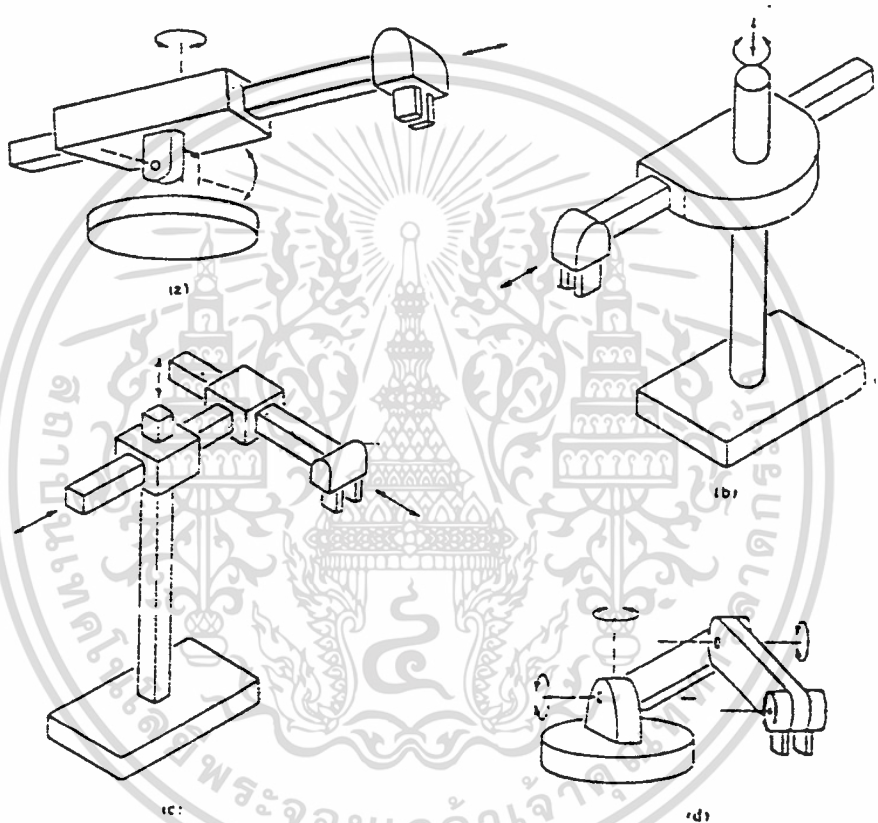
หุ่นยนต์ที่มีลักษณะเป็นแบบ โพลาร์(Polar) นั้นลักษณะการเคลื่อนที่ของแขนจะสามารถยกขึ้นลงได้ในแนวตั้ง โดยยกทำมุมกับฐาน สามารถหมุนได้รอบตัว (รูป 2.1 a) พื้นที่การทำงานจะเป็นแบบทรงกลม ดังนั้นในบางครั้งจึงเรียกว่า "Spherical coordinate"

หุ่นยนต์ที่มีลักษณะเป็นแบบ ไซเลนดริคัล(Cylindrical) (รูป 2.1 b) เคลื่อนที่ขึ้นข้างบนได้ตามแกนตั้งที่เป็นแกนหลัก และสามารถเคลื่อนที่ไปมาได้โดยใช้แกนนอน แกนตั้งสามารถที่จะหมุนได้พื้นที่การทำงานจะเป็นแบบทรงกระบอก

หุ่นยนต์ที่มีลักษณะเป็นแบบ อาร์ติคูลेट(Articulated) (รูป 2.1 c) ลักษณะการเคลื่อนที่จะมีแกน 3 แกน เป็นเหมือนแกน X,Y,Z. ดังนั้นในบางครั้งจึงเรียกว่า หุ่นยนต์เรคติลิเนียร์ (Rectilinear Robot) พื้นที่การทำงานสามารถที่จะทำงานได้ในส่วนที่เป็นด้านของมือเพียงด้านเดียวเท่านั้น เพราะไม่มีการหมุนของฐาน

หุ่นยนต์ที่มีลักษณะเป็น อาร์ติคูลेट หุ่นยนต์แบบนี้จะมีลักษณะใกล้เคียงกับแขนของมนุษย์ มีข้อหมุนต่างๆ เหมือนกัน (รูป 2.1 d) ดังนั้นพื้นที่การทำงานจึงสามารถที่จะทำงานได้ในทุกตำแหน่งในระยะความยาวของแขน

ในโครงการนี้ได้เลือกเอาแบบ อาร์ติคูเลต เป็นต้นแบบในการทำทั้งนี้เพราะว่ามีลักษณะเหมือนแขนของมนุษย์ ซึ่งคนส่วนใหญ่ก็มีความคิดอยู่แล้วว่า หุ่นยนต์ต้องมียลักษณะเหมือนมนุษย์ และอีกอย่างในการพัฒนาหุ่นยนต์ชนิด อาร์ติคูเลตนี้สามารถที่จะพัฒนาไปสู่การทำงานได้ดีกว่าชนิดอื่นๆ



รูปที่ 2.1 ชนิดของหุ่นยนต์ (a) โพลาร์ (b) ไชเลนคริคัล (c) คาร์ทีเซียน (d) อาร์ติคูเลต

ข้อดีข้อเสียของแต่ละชนิดนี้แตกต่างกันออกไป เพราะว่าลักษณะทางกายภาพแตกต่างกัน แต่ถ้ามองในแง่ของการทำงานที่เป็นแบบซ้ำๆ ที่เดิมตลอด ชนิด คาร์ทีเซียน จะสามารถทำงานได้ดีกว่า (สามารถเคลื่อนที่ไปหาเป้าหมายโดยมีความผิดพลาดน้อยที่สุด) แต่ถ้ามองในแง่การเข้าถึงวัตถุชนิดที่เป็นแบบ โพลาร์ และ อาร์ติคูเลต จะสามารถเข้าถึงวัตถุได้ดีกว่าชนิดอื่น และชนิดที่เป็นแบบ ไชเลนคริคัล จะมีข้อดีในแง่ที่สามารถยกวัตถุได้มากกว่า ในงานทั่วไปแล้วใช้แบบ

โพตาร์ท และ ไฮดรอลิก เพราะ 2 ชนิดนี้สามารถที่จะทำงานเป็นแบบ Load และ Unload โดยมีการเคลื่อนที่ของแขนออกไปในด้านข้างได้ดีกว่าชนิดอื่นๆ

## 2.2 มือของหุ่นยนต์ (Robot end effector)

มือจับของหุ่นยนต์คือตัวที่ทำหน้าที่จับสุดท้ายของหุ่นยนต์ มือจับของหุ่นยนต์สามารถแบ่งตามลักษณะของการทำงานได้ดังนี้

1. ทำงานเป็นตัวจับ (Gripper)
2. ทำหน้าที่เป็นเครื่องมือ (Tools)

มือของหุ่นยนต์ชนิดที่เป็นตัวจับมีหน้าที่โดยทั่วไปคือ การจับและการยกวัตถุ สามารถแบ่งออกอีกเป็น 2 ชนิดคือ Single gripper และ Double Gripper. Single gripper หมายถึง มือจับที่มีการเคลื่อนที่ของนิ้วเพียงข้างเดียวอีกข้างหนึ่งอยู่กับที่ และ Double gripper มีการเคลื่อนที่ของนิ้วทั้ง 2 ข้างเข้าหากัน

มือของหุ่นยนต์ชนิดที่เป็นตัวจับ สามารถแบ่งออกเป็นหลายๆ ชนิดตามลักษณะของตัวควบคุมได้ดังนี้

### 2.2.1 เมคคานิกกริปเปอร์ (Mechanical gripper)

เมคคานิกกริปเปอร์ คือมือจับที่มีกลไกทาง เมคคานิก เป็นตัวควบคุมนิ้วบางครั้งเรียกว่า จอว์ส(jaws) หน้าที่พื้นฐานของมือจับแบบเมคคานิก คือการส่งถ่ายกำลังจากมือ ไปสู่วัตถุต้นกำลัง อาจใช้ไฟฟ้า, นิวเมติกส์, พลัง หรือ ไฮดรอลิก ลักษณะการจับของมือจับนี้ สามารถที่จะออกแบบการจับได้ 2 อย่างคือ

1. ใช้ลักษณะทางกายภาพ เช่น ต้องการจับวัตถุเป็นทรงกลมตรงปลายมือก็ทำเป็นส่วนโค้ง ที่มีขนาดพอเหมาะกะกับวัตถุนั้น (รูป 2.2)

2. ใช้ความเสียดทาน (รูป 2.3) นิ้วไม้ได้ออกแบบให้มีลักษณะเหมือนกับรูปทรงของวัตถุ แต่อาศัยแรงเสียดทานของนิ้วกับวัตถุ โดยมีสมการของแรงเสียดทานคือ

$$\mu n_r F_g = w \quad \text{-----}(2.1)$$

เมื่อ  $\mu$  = สัมประสิทธิ์แรงเสียดทานของนิ้วกับวัตถุ

$n_r$  = จำนวนนิ้วของมือจับ

$F_g$  = แรงจับ

$W$  = น้ำหนักของวัตถุ

สมการ 2.1 นี้ใช้เมื่อแรงโน้มถ่วงมีทิศทางขนานกันกับพื้นที่สัมผัส ถ้าทิศทางของแรงโน้มถ่วงเปลี่ยนไปจะใช้สมการ 2.2 คือ

$$\mu n_r F_g = mg \quad \text{-----}(2.2)$$

เมื่อ  $m$  = มวล

$g$  คือ gravity factor และความเร่ง

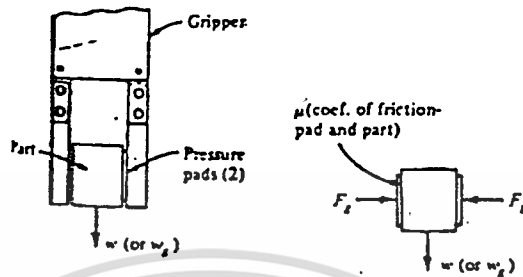
$g=1$  เมื่อ ความเร่งมีทิศทางตรงข้ามกับแรงโน้มถ่วง

$g=2$  เมื่อความเร่งมีทิศทางในแนวราบ (ตั้งฉากกับแรงโน้มถ่วง)

$g=3$  เมื่อความเร่งมีทิศทางเดียวกันกับแรงโน้มถ่วง

### รูปที่ 2.2 มือจับที่ออกแบบนิ้วให้มีรูปทรงคล้ายกับวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 มือจับที่อาศัยแรงเสียดทานของนิ้วกับวัตถุ

### 2.3 ความแม่นยำของการเคลื่อนที่

ความสำคัญอย่างหนึ่งของการสร้างหุ่นยนต์คือต้องการความแม่นยำในการทำงาน ความแม่นยำของหุ่นยนต์แต่ละตัวขึ้นกับ ตัวแปร 3 ประการคือ

1. สเปซเรซอลูชัน (Spatial resolution )
2. แมคคิวรีซี (Accuracy )
3. รีพีททาบิลิตี (Repeatability )

#### 2.3.1 สเปซเรซอลูชัน

คือช่วงการเคลื่อนที่ที่มีระยะทางสั้นที่สุดที่หุ่นยนต์แต่ละตัวสามารถที่จะทำได้ สเปซเรซอลูชัน ขึ้นอยู่กับ องค์ประกอบสำคัญ 2 ประการคือ

2.3.1.1 ระบบการควบคุม (Control system) ระบบการควบคุมนี้จะรวมถึงการวัดสัญญาณป้อนกลับของหุ่นยนต์ด้วย ช่วงของการเคลื่อนที่ที่ระบบควบคุมสามารถที่จะทำได้ขึ้นอยู่กับหน่วยความจำหลักของเครื่องคอมพิวเตอร์ เช่นเครื่องคอมพิวเตอร์ ที่มีหน่วยความจำ 8 bit จะสามารถที่

จะแบ่งช่วงการเคลื่อนที่ออกได้ เป็น 256 ช่วง คือ ช่วงของการเคลื่อนที่ที่คอมพิวเตอร์แบ่งได้ มีค่าเท่ากับ  $2^n$  เมื่อ  $n$  คือหน่วยความจำหลักของเครื่อง

2.3.1.2 ความคลาดเคลื่อนเชิงกล (Mechanical inaccuracy) ความคลาดเคลื่อนเชิงกลของหุ่นยนต์แต่ละตัวขึ้นอยู่กับลักษณะของ ข้อหมุน (joint)-และข้อต่อ (link) และระบบดันทำถ่วงของหุ่นยนต์ตัวนั้นด้วย

### 2.3.2 แอกคูเรซี

แอกคูเรซี คือตัวที่แสดงถึงความสามารถของหุ่นยนต์ในการที่เคลื่อนที่เข้าใกล้จุดเป้าหมายตามที่เราลัง แอกคูเรซี สามารถที่จะกำหนดให้อยู่ในเทอมของ สเปาเซียนริโซลูชัน ได้ทั้งนี้ .เพราะว่าการเคลื่อนที่ให้เข้าใกล้จุดเป้าหมาย ก็ต้องขึ้นอยู่กับช่วงของการเคลื่อนที่ว่า มีความละเอียดมากน้อยเพียงใด ในการทำงานเราต้องวางจุดที่เราต้องการให้หุ่นยนต์ทำงานอยู่ระหว่างกลางของตำแหน่งการเคลื่อนที่ ของหุ่นยนต์ทั้งนี้เพราะว่าความคลาดเคลื่อนเชิงกลมีผลต่อความแม่นยำของหุ่นยนต์

ความแม่นยำของหุ่นยนต์กำหนดให้เท่ากับครึ่งหนึ่งของระยะทางการเคลื่อนที่ที่สั้นที่สุดของหุ่นยนต์ที่สามารถทำได้ความแม่นยำของหุ่นยนต์ขึ้นอยู่กับองค์ประกอบเหล่านี้คือ

1. พื้นที่การทำงานของหุ่นยนต์ถ้าแขนทำงานในพื้นที่การทำงานจะมีความแม่นยำมากกว่าเมื่อแขนออกนอกพื้นที่การทำงาน
2. วงรอบการทำงาน ถ้าวงรอบการทำงานเป็นวงรอบที่แน่นอนความแม่นยำจะมีมากขึ้น
3. น้ำหนักที่ได้รับ ถ้าหุ่นยนต์ทำงานโดยการรับน้ำหนักมากๆ ความแม่นยำจะลดลง

### 2.3.3 วิธีททบีลิตี

วิธีททบีลิตี คือความสามารถของหุ่นยนต์ในการกลับมาทำงานซ้ำที่เดิม จากรูป 2.1 จุดเป้าหมายคือจุด T แต่เนื่องจากข้อจำกัดของหุ่นยนต์ หุ่นยนต์ไม่สามารถที่จะทำงานที่จุด T ได้ดังนั้นจึง

ทำงานที่จุด P ระยะห่างระหว่างจุด P กับจุด T คือตัวแสดงถึงความแม่นยำของหุ่นยนต์ แต่เมื่อสั่งให้หุ่นยนต์กลับมาทำงานที่จุด P

อีกครั้งหนึ่งหุ่นยนต์ไม่สามารถกลับมาทำงานที่จุด P ได้จะทำงานที่จุด R แทนระยะห่างระหว่างจุด P และจุด R คือค่าแสดงถึง วิธีที่ที่บิดเบี้ยว



## รูปที่ 2.4 ความแม่นยำและความสามารถในการทำซ้ำของหุ่นยนต์

### 2.4 การเคลื่อนที่ของหุ่นยนต์

การเคลื่อนที่ของหุ่นยนต์สามารถแบ่งการเคลื่อนที่ที่ออกได้ 2 ส่วนคือ

#### 2.4.1. การเคลื่อนที่ของตัวและแขน (body and arm)

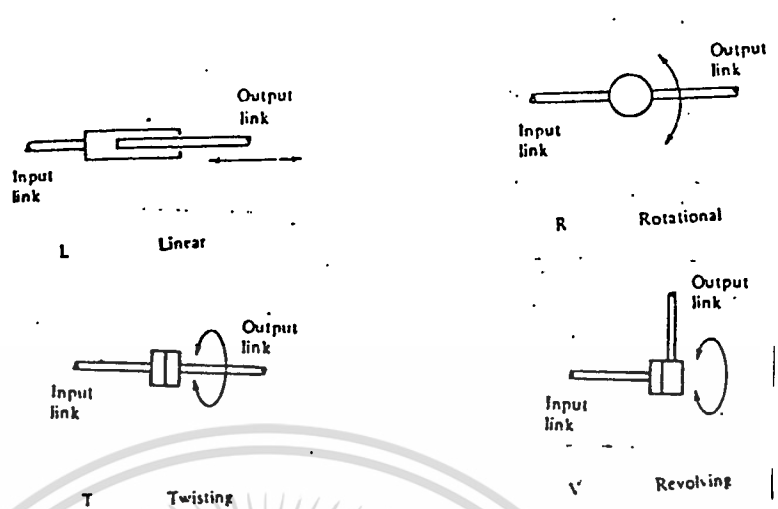
การเคลื่อนที่ของตัวและแขนแบ่งตามลักษณะของข้อหมุนได้ดังนี้คือ

##### 2.4.1.1 การเคลื่อนที่แบบเส้นตรง (Linear)

##### 2.4.1.2 การเคลื่อนที่แบบหมุนรอบจุดหมุน (Rotational)

##### 2.4.1.3 การเคลื่อนที่แบบบิดรอบจุดหมุน (Twisting)

##### 2.4.1.4 การเคลื่อนที่แบบหมุนตั้งฉาก (Revolving)



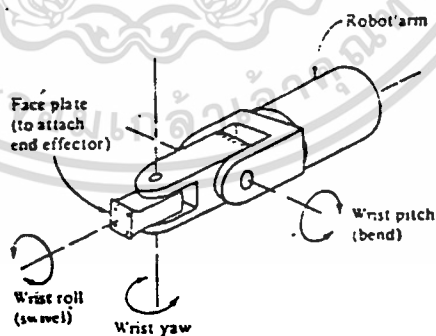
รูปที่ 2.5 ลักษณะการเคลื่อนที่ของแขนแบบต่าง ๆ

2.4.2 การเคลื่อนที่ของมือ

2.4.2.1 หมุน (Roll) บางครั้งเรียกว่า Swival ข้อมือหมุนรอบแกนของแขน

2.4.2.2 บิด (Pitch) บางครั้งเรียกว่า Bend ข้อมือจะยกขึ้นลงในแนวตั้ง

2.4.2.3 ถ่าย (Yaw) หมายถึงการบิดไปมาทางซ้ายและขวาของแกนมือ



รูปที่ 2.3 การเคลื่อนที่ของมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

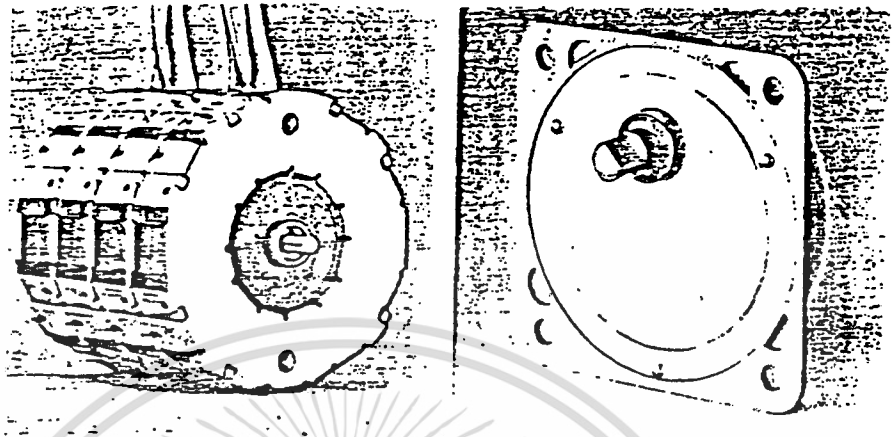
### บทที่ 3

#### สเตปป์ิ่งมอเตอร์ STEPPING MOTOR

สเตปป์ิ่งมอเตอร์ เป็นตัวขับเคลื่อนชนิดหนึ่งที่สามารถนำมาใช้ในการขับเคลื่อนหุ่นยนต์ สเตปป์ิ่งมอเตอร์ นี้ เมื่อเปรียบเทียบกับความสามารถกับมอเตอร์ไฟฟ้ากระแสตรงแล้วจะได้เปรียบกว่ามาก ทั้งนี้เพราะว่า สเตปป์ิ่งมอเตอร์ มีแรงบิดที่สูงกว่ามอเตอร์ไฟฟ้ากระแสตรง และมีรอบการหมุนที่ช้ากว่าทำให้ไม่จำเป็นต้องมีชุดเฟืองทด สามารถที่จะต่อตรงได้เลย แต่มีข้อเสียเปรียบตรงที่การควบคุมค่อนข้างยาก และมีราคาแพงกว่ามอเตอร์ไฟฟ้ากระแสตรงมาก หากมากกว่า รูปที่ 3.1 แสดงลักษณะของ สเตปป์ิ่งมอเตอร์ และรูปที่ 3.2 แสดงการวางตำแหน่งของขดลวดใน สเตปป์ิ่งมอเตอร์ ชนิด 3-เฟส หลักการทำงานอย่างง่าย ๆ ของ สเตปป์ิ่งมอเตอร์ คือเมื่อเราผ่านกระแสไฟฟ้าเข้าไปในขดลวดแอมเจอร์ และขดลวดที่อยู่รอบๆ จะให้เกิดสนามแม่เหล็กเกิดขึ้นขั้วแม่เหล็กที่ตรงข้ามกันจะดึงดูดกันและถ้าต้องการให้หมุนก็เปลี่ยนกระแสไฟฟ้าให้ไปที่ขดลวดรอบนอกขดลวดต่อไป แกนก็จะหมุนไปที่ขั้วของขดลวดนั้น จากรูปแกนของมอเตอร์จะอยู่ที่คู่ 1a,1b เราต้องการให้มีการหมุนก็เปลี่ยนกระแสไฟมาที่ขดลวดคู่ 2a;2b ขดลวดแอมเจอร์ก็จะหมุนมาที่ คู่ 2a,2b คือการหมุนจะหมุนไปที่ละขั้น สเตปป์ิ่งมอเตอร์ ที่แสดงให้ดูเป็นชนิด 3-เฟส หนึ่งรอบของการหมุน มีการหยุด 6 ครั้ง ดังนั้นในการหมุนแต่ละครั้งแกนของมอเตอร์จะเบี่ยง ไปเท่ากับ  $360/6$  คือ 60 องศา

ใน สเตปป์ิ่งมอเตอร์ บางตัวจะมีชุดเฟืองทดอยู่ในตัวเพื่อลดมุมของการหมุนและทำให้มีแรงบิดมากกว่าเดิม สเตปป์ิ่งมอเตอร์ ส่วนมากจะมีมุมของการหมุนเท่ากับ 1.8 องศา

การควบคุม สเตปป์ิ่งมอเตอร์ มีวงจรควบคุมดังแสดงในรูป 3.3 ซึ่งจะมีตัว ดีโคเดอร์ (Decoder) เป็นตัวเลือกว่า จะปล่อยกระแสไฟให้ไหลผ่าน ไปยังขดลวดขั้วใดเพื่อทำให้เกิดการหมุน และมี มอด 3 เคาน์เตอร์ (Mod 3 counter) เป็นตัวกำหนดทิศทางของการหมุนจากรูป ถ้าเลือก CU (Count Up) มอเตอร์ จะหมุนตามเข็มนาฬิกา และถ้าเลือก CD (Count Down) มอเตอร์จะหมุนทวนเข็มนาฬิกา

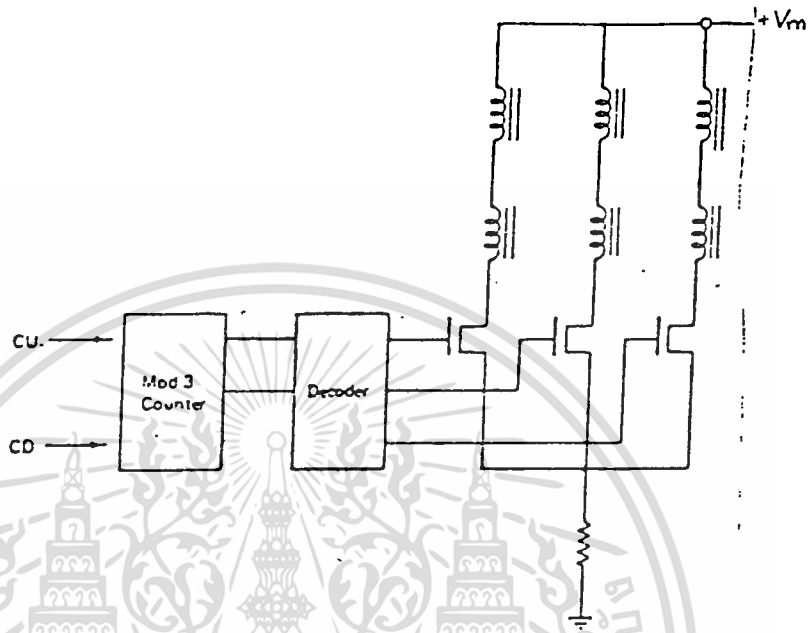


รูปที่ 3.1 สเตปป์มอเตอร์



รูปที่ 3.2 การวางขดลวดใน สเตปป์มอเตอร์ ชนิด 3 เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 วงจรการควบคุม สเตปป์มอเตอร์

สเตปเปอร์มอเตอร์ โดยทั่วไปมีอยู่ 3 ชนิด คือ

- วารีเอเบิล รีลักแตนซ์ [VARIABLE RELUCTANCE (VR)]
- เพอร์มาเนนท์แมกเนต [PERMANENT MAGNET (PM)]
- ไฮบริดสเตปเปอร์มอเตอร์ [HYBRID STEPPER MOTOR (HSM)]

3.1 วารีเอเบิลรีลักแตนซ์ (VR) จะเป็นสเตปเปอร์มอเตอร์ที่ถูกกล่าวถึงและนำไปใช้งานมากที่สุด โดยเฉพาะแบบ 4 เฟส โรเตอร์และสเตเตอร์จะทำมาจากเหล็กผสมซิลิกอน เพลาของมอเตอร์จะหมุนไปเป็นค่ามุมคงที่ หรือเรียกว่า มุมสเตป ( $\theta_s =$  ซีคร้า เอส) มักจะมีมุมที่ต่างกันคือ 0.72, 0.9, 1.8, 2.0, 3.6, 7.5, 15 สำหรับสเตปเปอร์มอเตอร์ชนิดนี้ส่วนใหญ่จะใช้วงจรขับแบบยูนิโพลาร์ (UNIPOLAR) ซึ่งลักษณะการทำงานของวงจรจะต้องป้อนค่า โวลต์เดจเข้าไปในแต่ละ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อตามลักษณะของ พัลส์ ที่ใช้ขับชุด ไดรเวอร์ (DRIVER) ค่า โวลต์แดงที่จ่ายให้มอเตอร์จะใช้อ้างอิงกับกราวด์เสมอ

การหามุมสเตป

$$S = 360 / Os$$

หน่วยองศา

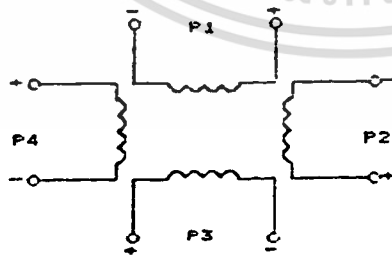
$$Os = mNr$$

m = จำนวนเฟส

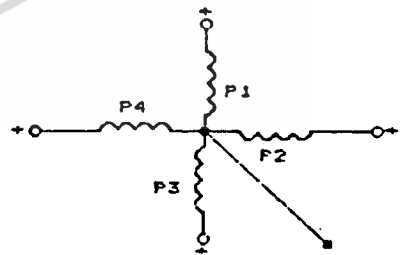
Nr = จำนวนขั้วของโรเตอร์

3.2 เพอร์มาเนนท์แมกเนต (PM) จะเป็นสเตปเปอร์มอเตอร์แบบที่ใช้ตัวโรเตอร์ทำด้วยแม่เหล็กถาวร สเตปเปอร์มอเตอร์แบบนี้ จะใช้ลักษณะการหมุนแบบไบโพลาร์ (BIPOLAR) ซึ่งลักษณะการทำงานจะต้องจ่ายไฟบวกลบเข้าแต่ละเฟสของมอเตอร์โดยตรง

3.3 ไฮบริดสเตปเปอร์มอเตอร์ (HSM) จะเป็นแบบผสมระหว่าง (PM) และ (VR) จำนวนขั้วฟันของโรเตอร์และสเตเตอร์จะไม่เท่ากัน สเตปเปอร์มอเตอร์แบบนี้จะมีลักษณะการทำงานที่ซับซ้อน จึงเป็นแบบที่ไม่ค่อยนิยมใช้งานเท่าใดนัก



BIPOLAR



UNIPOLAR

รูปที่ 3.4 วงจรขับ STEPPER MOTOR แบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทั่วไปวงจรที่ใช้ขับเคลื่อนมอเตอร์ จะนิยมใช้ทั้ง 2 แบบนี้คือ แบบไบโพลาร์ หรือ CHOPPER จะใช้กับมอเตอร์ที่เป็นชนิด PM และแบบยูนิโพลาร์ จะใช้กับชนิด VR ส่วน การขับของมอเตอร์ชนิด HSM จะมีวงจรการขับที่แตกต่างออกไป เนื่องมาจากคุณสมบัติของมอเตอร์ชนิดนี้สามารถทำงานตามกระแสได้ทั้งสองทิศทาง ( สลับขั้วไฟบวกลบที่จ่ายให้กับมอเตอร์ )

3.4 วิธีการกระตุ้นเฟส

- **ซิงเกิลเฟส (SINGLE PHASE EXITATION)** เป็นการกระตุ้นแบบเฟสเดียว ตาม สัญญาณพัลส์ที่ป้อนเข้าชุดขับเคลื่อนมอเตอร์

PHASE	A				B			
	1	0	0	0	1	0	0	0
A	1	0	0	0	1	0	0	0
B	0	1	0	0	0	1	0	0
C	0	0	1	0	0	0	1	0
D	0	0	0	1	0	0	0	1

- **ทูเฟส (TWO PHASE EXITATION)** การกระตุ้นเป็นแบบทีละสองเฟสคู่พร้อมกัน

PHASE	A				B			
	1	0	0	1	1	0	0	1
A	1	0	0	1	1	0	0	1
B	1	1	0	0	1	1	0	0
C	0	1	1	0	0	1	1	0
D	0	0	1	1	0	0	1	1

- **ฮาล์ฟสเตป (HALF STEP EXITATION)** แบบกึ่งสเตป จะเป็นการรวมรูปแบบการหมุนของทั้งสองแบบไว้ในแบบเดียวกัน

PHASE	PULSE								
A	1	1	0	0	0	0	0	0	1
B	0	1	1	1	0	0	0	0	0
C	0	0	0	1	1	1	0	0	0
D	0	0	0	0	0	1	1	1	1

วิธีการกระตุ้นเฟสทั้ง 3 แบบนี้จะมีคุณสมบัติที่แตกต่างกัน ดังนั้นการเลือกวิธีการกระตุ้นจึงจำเป็นที่จะต้องพิจารณาเพื่อให้เหมาะสมกับงานนั้น ๆ การกระตุ้นแบบ ชิงเกิลเฟส จะเป็นแบบที่มีความเที่ยงตรงของตำแหน่งงานสูง แต่จะมีแรงบิด (Torque) น้อย ส่วนการกระตุ้นแบบทวิเฟส มีความเที่ยงตรงของตำแหน่งน้อยกว่าแบบแรก แต่มีแรงบิดสูงกว่า และการกระตุ้นแบบฮาล์ฟสเตป เป็นแบบที่มีความเที่ยงตรงของตำแหน่งน้อยมาก แต่จะมีแรงบิดสูงมากเช่นกัน

#### บทที่ 4

#### การประกอบแขนกล

แขนกลที่ประกอบขึ้นในการทดลองครั้งนี้ เป็นแขนกลแบบอาร์คิคูเลต ซึ่งมีการเคลื่อนที่ 5 ดีกรีออฟฟรీดอม (ไม่รวม กริปเปอร์) มีต้นกำลังมาจากสเตปป์มอเตอร์ 5 ตัว และประกอบกับชุด ดีซีมอเตอร์ 1 ตัว เพื่อควบคุมการบีบของมือ

ในการประกอบใช้วัสดุทั้งเหล็กและพลาสติกสังเคราะห์ (super leane) และอลูมิเนียม ตามตำแหน่ง และความเหมาะสม ในการรับแรงที่จุดต่างๆ และเพื่อเป็นการลดโหลด ที่เกิดกับมอเตอร์ที่จุดหมุนส่วนใหญ่จะใช้บุชแทนลูกปืน เพราะรับรอบการหมุนที่ช้า และบุชสามารถให้ความผิดพลาดได้น้อยกว่า

สายพานจะใช้แบบซิงโครนัส (synchronomous belt) เบอร์ 037 ขนาดความยาวต่างๆกัน ตามขนาดที่เหมาะสม

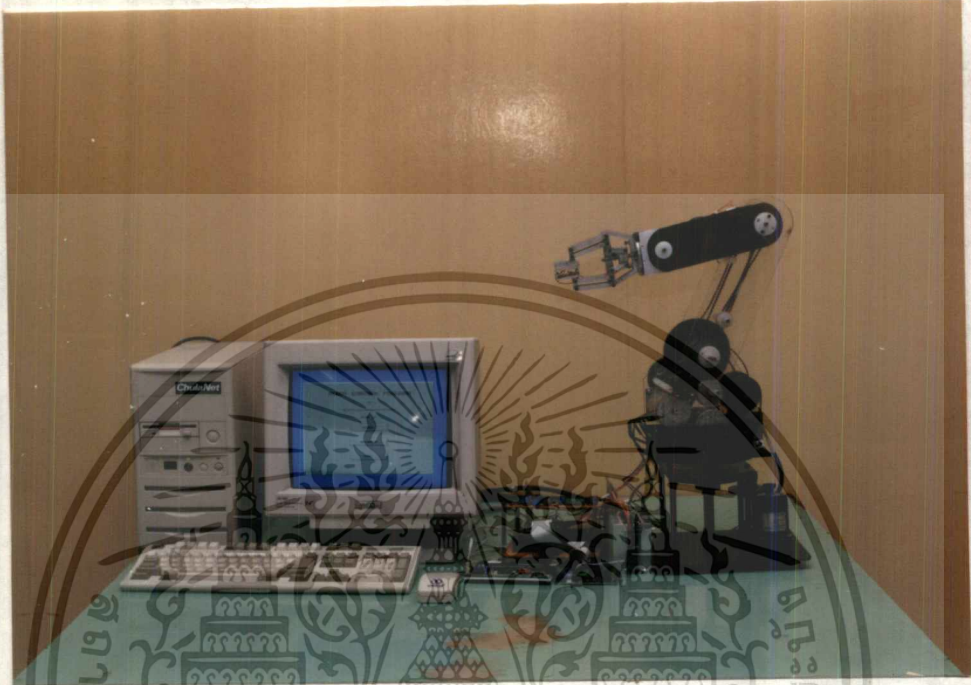
พูลเลย์ใช้เบอร์ 037 ขนาดเส้นผ่านศูนย์กลางต่างๆกันไป

สเตปป์มอเตอร์ ใช้แบบ วารีเอเบิลรีลัคแตนซ์

ฟีนเฟืองใช้แบบที่มีระยะพิท 0.2 มิลลิเมตร เส้นผ่าศูนย์กลางต่างๆกัน

ลิมิตสวิทช์ (limit switch) ใช้ 6 ตัวซึ่งเป็นชนิดที่มีหน้าสัมผัส N.O. (Normal Open)

และ N.C. (Normal Close) อย่างละ 1 ชุด ใช้เพื่อจำกัดขอบเขตการเคลื่อนที่ของแขนกล



รูปที่ 4.1 แสดงชุดทดลอง

จากที่กล่าวมาทั้งหมดข้างต้นก็จะแจกรายละเอียดออกเป็น 5 ส่วนหลักๆของชุดแขนกล โดยทั้ง 5 ส่วนคือ

- 1) ส่วนฐาน (base part)
- 2) ส่วนชุดเฟืองที่หัวไหล่ ( shoulder part)
- 3) ส่วนข้อศอก ( elbow part )
- 4) ส่วนข้อมือ ( hand - connector part )
- 5) ส่วนกริปเปอร์ ( gripper part )

โดยทั้ง 5 ส่วนก็จะต้องทำงานสัมพันธ์กันตลอด เริ่มจากส่วนฐาน การทำงานของชุดแขนกลทั้งชุดจะถูกขับเคลื่อนโดยรอบด้วย สเตปปีงมอเตอร์ ซึ่งต่อกับเฟือง 19 ฟัน ซึ่งเฟืองชุดนี้จะถูกหดรอบให้ช้าลงโดยชุดเฟือง 60 ฟัน เมื่อชุดเฟือง 60 ฟันยึดอยู่บนแกนเดียวกับเฟือง 16 ฟัน ก็จะหมุนด้วยความเร็วรอบเท่ากัน ชุดเฟือง 60 ฟันนี้ ก็จะไปขับชุดเฟือง 110 ฟัน ซึ่งเป็นชุดสุดท้ายที่ต่อกับงานหมุนก็จะขับเคลื่อนชุดแขนกลในแกนนี้ได้

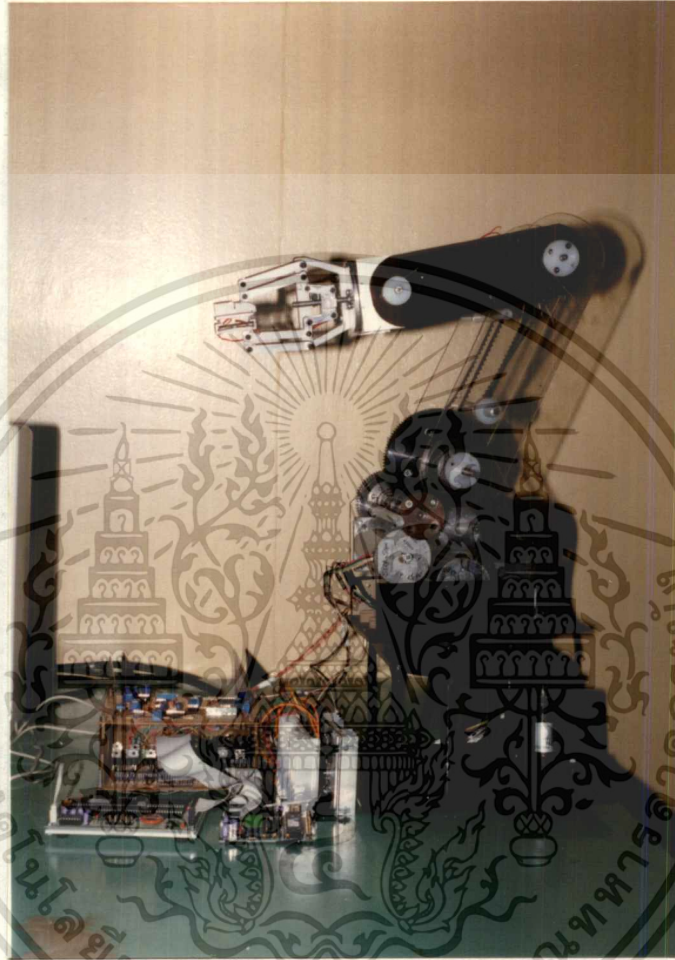
ส่วนต่อไปคือส่วนหัวไหล่ ( elbow part ) ซึ่งชุดหัวไหล่จะประกอบไปด้วย สเตปป์มอเตอร์ 3 ตัว และชุดเฟืองขับพร้อมสายพานส่งกำลัง 3 ชุด เพื่อขับเคลื่อนในส่วนขอส่วนข้อศอก ส่วนข้อมือ และควบคุมการขับเคลื่อนลงของกริปเปอร์ โดยสเตปป์มอเตอร์ ทั้ง 3 ตัวในส่วนหัวไหล่มีการทำงาน ดังนี้คือ สเตปป์มอเตอร์ ตัวใหญ่จะทำหน้าที่ไปขับ ส่วนข้อศอก ให้เคลื่อนที่ ต่อจากนั้นการเคลื่อนที่ในส่วนของข้อมือจะถูกขับด้วย สเตปป์มอเตอร์ ขนาดกลาง และการทำงานของ สเตปป์มอเตอร์ ตัวสุดท้ายก็จะเป็นการควบคุมในการขับเคลื่อน-ลงของชุดกริปเปอร์ จากการทำงานทั้งหมดของมอเตอร์ก็จะต้องอาศัยการส่งผ่านกำลังของชุด เฟืองโดยส่งผ่านกำลังด้วยสายพาน

ส่วนข้อศอกจะทำงานในลักษณะเป็นชุดส่งผ่านกำลังไปยังชุดข้อมืออีกที โดยทำงานในลักษณะเป็นตัวกลางรับแรงในการพาชุดแขนกล ให้ส่วนย่อยอื่นเคลื่อนที่ได้สูงขึ้นได้ ก็จะเป็นการเชื่อมต่อการทำงานของกริปเปอร์ เพราะฉะนั้นในส่วนนี้จึงต้องมีลักษณะขาเพื่อใช้เป็นคล้ายๆกับลำตัว

ในส่วนของข้อมือก็มีลักษณะคล้ายกับส่วนของข้อศอกกล่าวคือ ก็จะช่วยเพิ่มระยะทางให้กับกริปเปอร์ โดยจะได้รับแรงส่งมาจากชุดหัวไหล่ผ่านมาได้ด้วยสายพาน นอกจากนี้ในส่วนของข้อมือจะมีชุดสเตปป์มอเตอร์ เพื่อใช้ในการควบคุมการหมุนของ กริปเปอร์โดยตรงไม่ใช่ชุดเฟืองทดเหมือนในส่วนอื่นๆ

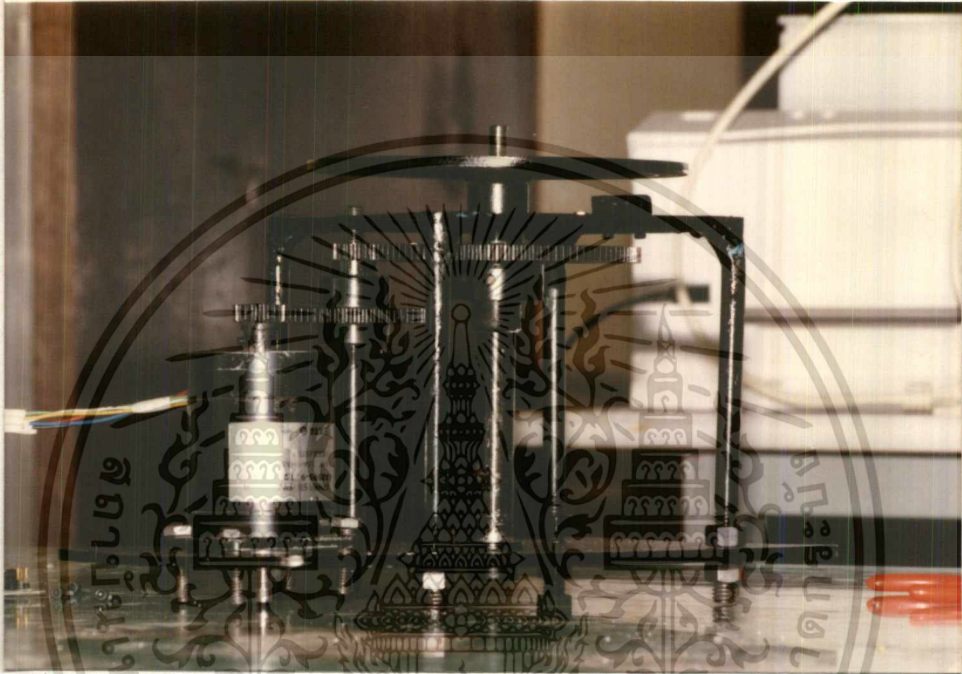
และในส่วนสุดท้ายก็คือส่วนของกริปเปอร์ โดยกริปเปอร์จะทำงานในลักษณะบีบเพื่อจับชิ้นงาน ในการทำงานก็จะเลื่อนกริปเปอร์ให้ขยาย-บีบตัวได้จากการรับแรงในส่วนของดีซีมอเตอร์ ที่ติดอยู่ในชุดเดียวกัน และดีซีมอเตอร์ เมื่อได้รับคำสั่งก็จะขับเคลื่อนกริปเปอร์โดยส่งผ่านกำลังด้วยสายพานแล้วสายพานก็จะหมุนทำให้เกิดลิวหมุนตามไปด้วย เมื่อเกลียวหมุนก็จะไปตั้งฐานของชุดกริปเปอร์ ทำให้ก้านของกริปเปอร์เคลื่อนที่เข้าหากัน

การทำงานของกริปเปอร์ก็จะจับชิ้นงานและการควบคุมน้ำหนักของการบีบไม่ให้หนักเกินไป และผู้ใช้ก็ทราบน้ำหนักได้จากชุดอุป stain gate โดยจะส่งค่าผลตอบสนองกลับมา



รูปที่ 4.2 แสดงชุดแขนกล

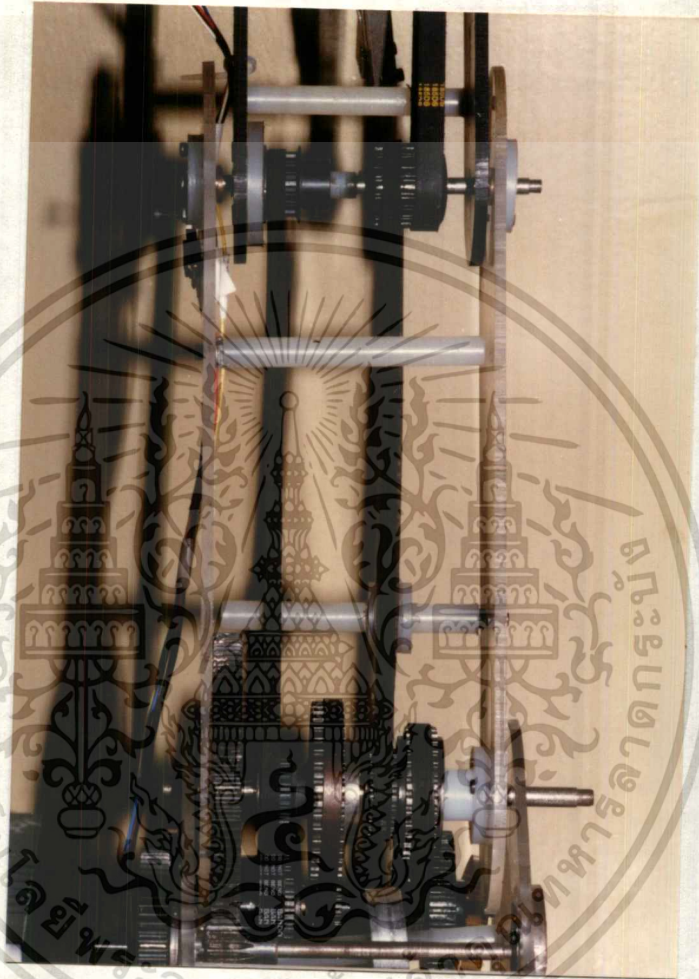
#### 4.1. ส่วนฐาน



รูปที่ 4.3 แสดงส่วนฐาน

ส่วนฐานประกอบขึ้นโดยมีจานหมุนเป็นตัวรับน้ำหนักแกนกล และใช้สเตปปีงมอเตอร์ขับให้หมุนได้ อิศระ 360 องศาโดยผ่านชุดเฟืองที่มีอัตราทด 21.7 : 1 เมื่อสเตปปีงมอเตอร์ ถูกกระตุ้น 1 พัลส์ (1.8องศา/Step) จะทำให้ตัวแกนกลหมุนไป 0.082909 องศา ในกรณีที่มีการยึดออกไปด้านข้างจนสุดจะทำให้ปลายมือขยับไปด้านข้าง 0.8038 มม.

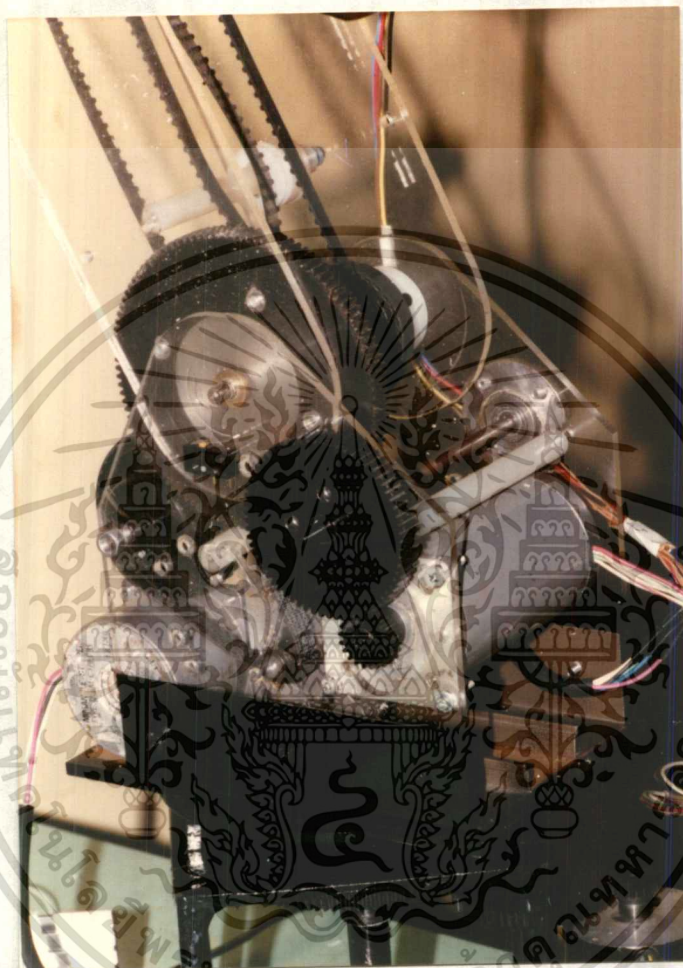
## 4.2 ส่วนหัวไหล่



รูปที่ 4.4 แสดงชุดหัวไหล่

ส่วนหัวไหล่สามารถเคลื่อนที่ได้ 150 องศา โดยใช้สเตปป์มอเตอร์เป็นตัวขับเคลื่อนผ่านชุดเฟืองซึ่งมีอัตราทดรวม  $24.266 : 1$  ซึ่งเมื่อสเตปป์มอเตอร์ ถูกกระตุ้น 1 พัลส์ (1.8 องศา/Step) จะทำให้มุมที่หัวไหล่เปลี่ยนไป 0.07417 องศา

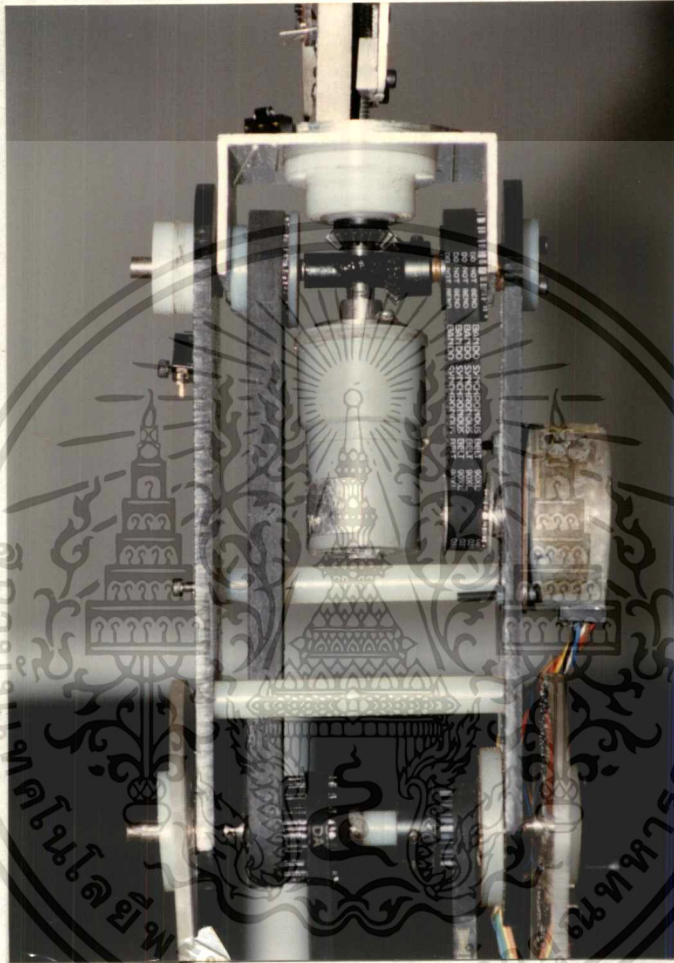
#### 4.3 ส่วนข้อตอก



รูปที่ 4.5 แสดงชุดข้อตอก

ส่วนข้อตอกสามารถเคลื่อนที่ได้ 200 องศา โดยใช้สเตปป์มอเตอร์ ถ่ายกำลังผ่านชุดพูลเลย์ทกรอบซึ่งมีอัตราทด 16 : 1 ส่งกำลังย่านสายพานมาขับเคลื่อนของข้อตอก ซึ่งเมื่อ สเตปป์มอเตอร์ถูกกระตุ้น 1 พัลส์ (0.9องศา / Step) จะทำให้มุมที่ข้อตอกหมุนไป 0.05625 องศา

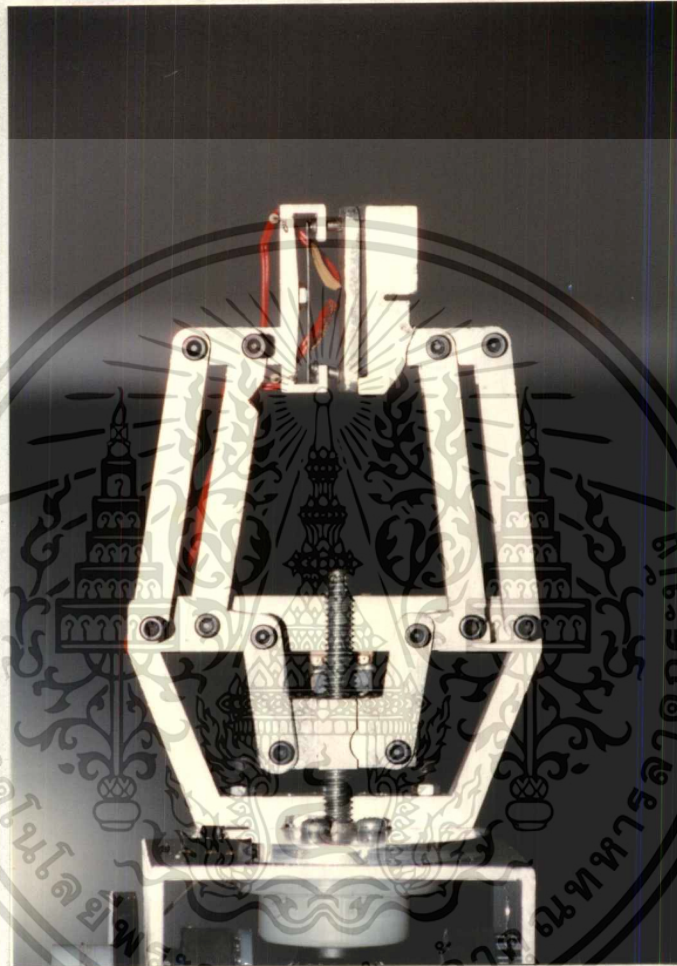
#### 4.4 ส่วนข้อมือ



รูปที่ 4.6 แสดงส่วนข้อมือ

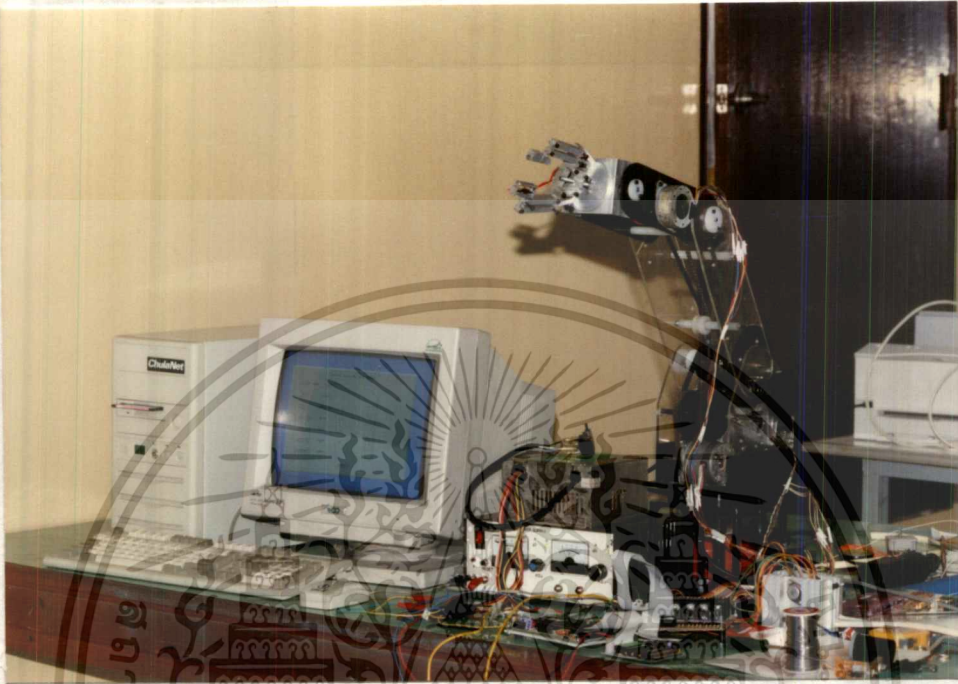
ส่วนข้อมือสามารถเคลื่อนที่ได้ 180 องศา ใช้ สเตปปีงมอเตอร์ ขับโดยส่งกำลังผ่านชุดพูลเลย์ทนต์ และสายพาน ขึ้นมายังชุดข้อมือ ซึ่งมีอัตราทดรวม 7.4 : 1 ซึ่งเมื่อสเตปปีงมอเตอร์ถูกกระตุ้น 1 พัลส์ (1.8 องศา/Step) จะทำให้มุมที่ข้อมือเปลี่ยนไป 0.2432 องศา

#### 4.5 ส่วนมือ



รูปที่ 4.7 แสดงชุดมือจับ

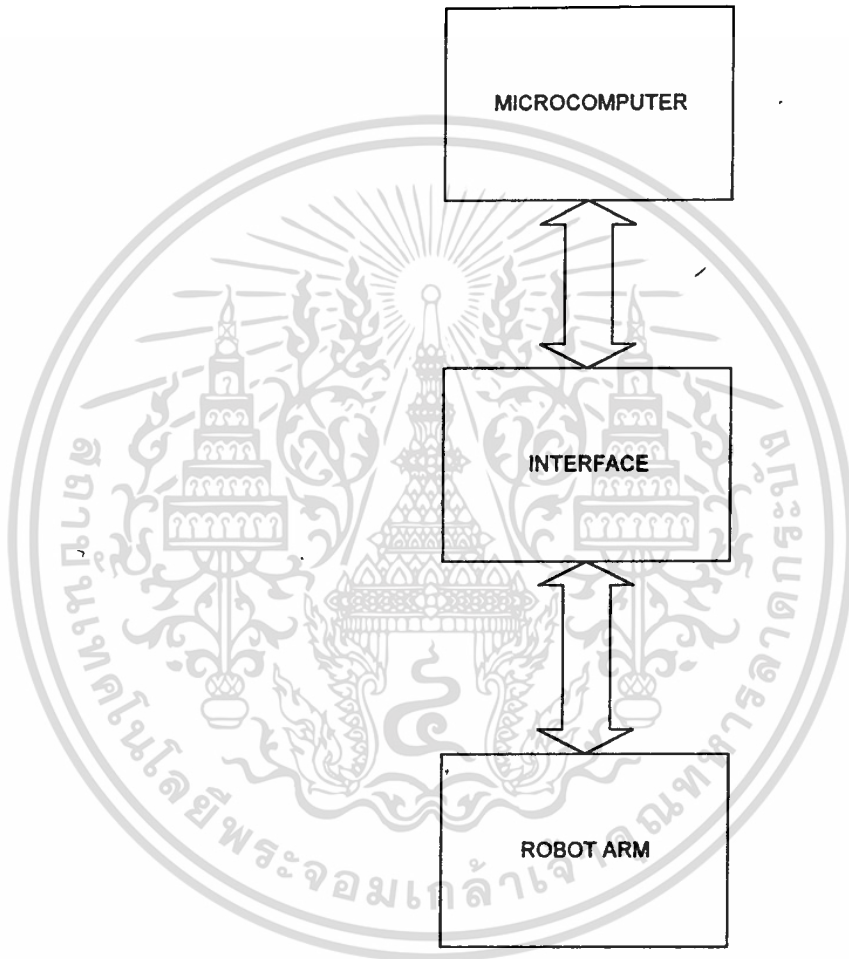
ส่วนมือสามารถ้าออกเพื่อจับวัตถุขนาดใหญ่ไม่เกิน 8 ซม. โดยใช้ดีซีมอเตอร์เป็นตัวต้นกำลัง และหมุนรอบตัวเองได้ 360 องศา โดยใช้พร้อมชุดเฟืองทดเป็นตัวขับเคลื่อน  
 ในส่วนมือของหุ่นยนต์นั้นสามารถกำหนดแรงในการบีบวัตถุได้ โดยจะใช้หลักการของ load cell คือมีการติด strain gate ลงบนแผ่นเหล็กสปริง ในลักษณะที่จะถูกตัดเมื่อรับแรงตัด



รูปที่ 4.8 แสดงชุดแขนกลและคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### โครงสร้างของโครงการ



รูปที่ 5.1 บล็อกไดอะแกรมของแขนกล

ไมโครคอมพิวเตอร์ (Microcomputer) ใช้สำหรับส่งค่าข้อมูลที่ใช้ในการควบคุมแขนกล ไปยังระบบ อินเทอร์เฟซ (Interface) และรับค่า ฟีดแบค (feedback) จากระบบอินเทอร์เฟซ กลับเข้ามา และแสดงผลการทำงานของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเทอร์เฟซ (Interface) รับค่าข้อมูลจากเครื่องคอมพิวเตอร์ แล้วส่งไปยังสเตปป์มอเตอร์ เพื่อขับให้แขนกลเคลื่อนที่ไปยังตำแหน่งที่ต้องการ และรับค่าพีคแบค จากแขนกลกลับมาเพื่อใช้ในการประมวลผลการทำงานของแขนกล อินเทอร์เฟซ จะประกอบด้วย Microcontroller MCS-51 วงจร โดจิก และวงจรขับสเตปป์มอเตอร์

แขนกล (Robot Arm) เคลื่อนที่โดยใช้ สเตปป์มอเตอร์ 5 ตัว ใช้ควบคุมการเคลื่อนไหวของจุดหมุนต่าง ๆ 5 จุด และ สเตปป์มอเตอร์ อีก 1 ตัวใช้สำหรับการทำงานของมือจับ (gripper) ที่มือจับจะมี เซนเซอร์ อีก 2 ชนิดคือชนิดที่ใช้เซ็นเซอร์บีบที่มือจับกระทำต่อวัตถุ และ เซนเซอร์ที่ใช้ตรวจจับการถื่นไถลของวัตถุเมื่อแรงบีบของมือจับ ไม่พอที่จะยกวัตถุขึ้นจากพื้น

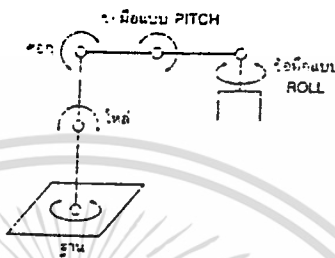
### 5.1 อัลกอริทึมในการควบคุมแขนกล

#### กลวิธีและขั้นตอนในการควบคุมแขนกลทำงานตามจินตนาการ

สิ่งที่บ่งบอกถึงประสิทธิภาพในการทำงานของแขนกล คือความแม่นยำและเที่ยงตรงซึ่งการเคลื่อนไหวขณะการทำงานของแขนกลจะประกอบด้วย การเคลื่อนไหวของ ฐาน, ไหล่, ข้อศอก, ข้อมือ เป็นต้น การเคลื่อนไหวตามส่วนต่างๆ นี้จะส่งผลให้ตำแหน่งของมือเคลื่อนที่ไปตามตำแหน่งต่างๆ ที่ปฏิบัติงาน ฉะนั้น ถ้าส่วนต่างๆ นี้มีการควบคุมที่ไม่ดีพอจะส่งผลถึงความแม่นยำและความเที่ยงตรงในการทำงานนั้น ถคน้อยลง

ดังนั้น มือของแขนกลจะเคลื่อนที่ไปตามตำแหน่งใดๆ ได้นั้น ขึ้นอยู่กับข้อต่อต่างๆ จะเคลื่อนไปในตำแหน่งที่เหมาะสมและสอดคล้องกันด้วย ซึ่งการควบคุมก็ต้องควบคุมตำแหน่งของข้อต่อต่างๆ ให้เหมาะสมกันซึ่งสามารถแสดงได้ด้วยสมการคณิตศาสตร์ สำหรับสมการที่กล่าวถึงในที่นี้มีสองลักษณะด้วยกันคือ สมการ direct kinematic และสมการ inverse kinematic ซึ่งสมการทั้งสองลักษณะใช้แสดงความสัมพันธ์ของข้อต่อต่างๆ และตำแหน่งปลายมือ ก่อนที่จะกล่าวถึงสมการของแขนกล มาทราบถึงศัพท์บางคำที่ใช้กันบ่อย ในแขนกลหรือแขนหุ่นยนต์ คำแรกคือดีกรีออฟฟรียดอม (degree of freedom) หรือ DOF หมายถึงจำนวนข้อต่อหรือจุดต่อที่เคลื่อนที่เป็นอิสระต่อกัน ส่วนอีกคำคือก้าน (link) หมายถึงสิ่งที่เชื่อมต่อระหว่างข้อต่อกับข้อต่อ ตัวอย่างเช่น แขนกลชนิดอาร์ดียูเอคที่แสดงในรูปที่ 5.2 เป็นไดอะแกรมแสดงแขนกลที่มีการเคลื่อนที่

## แบบ 5 คีกรีโออฟฟรีคอม

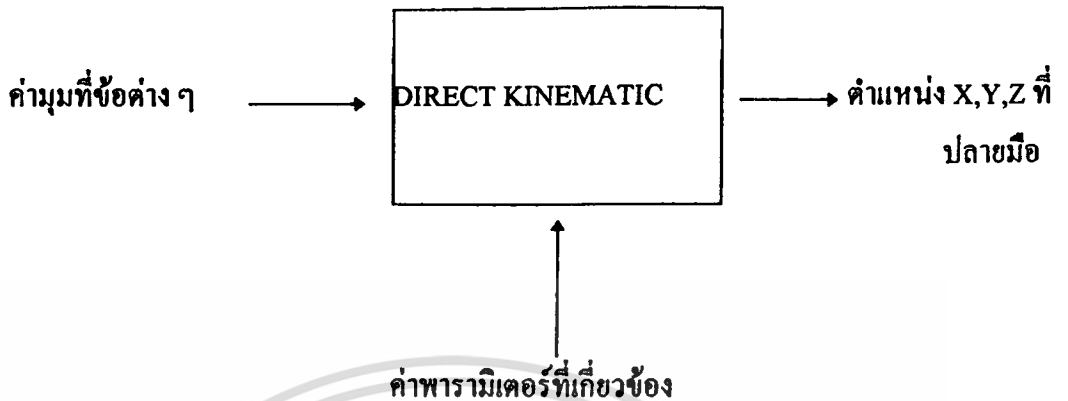


รูปที่ 5.2 โคอะแกรมการทำงานของแขนกลในรูปแบบ 5 คีกรีโออฟฟรีคอม

โดยทั่วไปในการควบคุมแขนกล ให้เคลื่อนที่ไปตามต้องการนั้น จะต้องมีตำแหน่งเริ่มต้นของแขน โดยเป็นตำแหน่งอ้างอิงในการเคลื่อนที่ เรียกว่าตำแหน่ง โฮม (home) ซึ่งก็ขึ้นอยู่กับชนิดของแขนกลและผู้ควบคุมว่าต้องการให้ตำแหน่งนี้อยู่ในลักษณะใดสำหรับบทความในตอนนี้จะอ้างอิงตำแหน่งโฮม ในลักษณะดังรูปที่ 5.2 เป็นหลักรวมทั้งชนิดของแขนกลที่ใช้ในการอ้างอิงสมการหรือสูตรต่าง ๆ เป็นชนิดอาร์ติคูลेट แบบ 5 คีกรีโออฟฟรีคอม

5.2 สมการ Direct kinematic

ความหมายของสมการ Direct kinematic คือสมการของแขนกลที่แสดงความสัมพันธ์ของการเคลื่อนที่ของข้อต่อต่างๆ โดยผลลัพธ์ของสมการคือ ตำแหน่งปลายมือในพื้นที่ 3 มิติ ( ตำแหน่ง  $X, Y, Z$  ) บล็อกโคอะแกรมแสดงการควบคุมแสดงดังรูปที่ 5.3

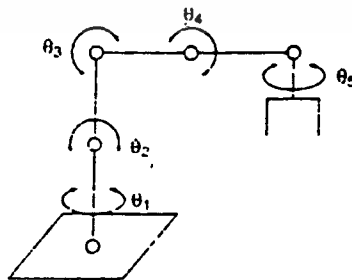


รูปที่ 5.3 บล็อกไดอะแกรมของสมการ direct kinematic

จากบล็อกไดอะแกรมจะพบว่าส่วนที่เป็นอินพุทของสมการ คือ ค่าของมุมที่ข้อต่อต่างๆ ซึ่งจะขึ้นอยู่กับแขนกลว่ามีกี่ข้อต่อ ส่วนค่าพารามิเตอร์ที่เกี่ยวข้อง จะเป็นค่าคงที่ประจำตัวแขนกล เช่น ความยาวของแต่ละก้านจากข้อต่อหนึ่งไปยังอีกข้อต่อหนึ่ง เป็นต้น และเมื่อผ่านสมการนี้ผลลัพธ์ที่ได้ออกมาจะเป็นตำแหน่งของปลายมือที่แขนกลเคลื่อนที่ไปบนพื้นที่ 3 มิติ (X,Y,Z)

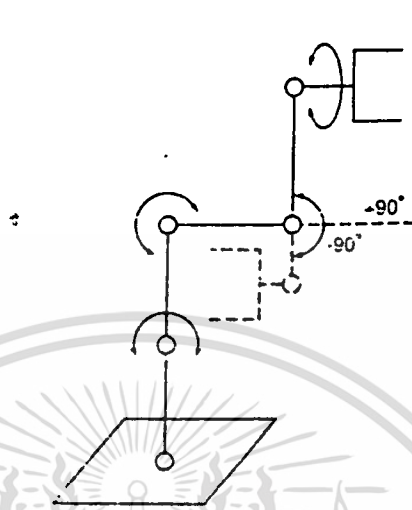
#### 5.2.1 การอ้างอิงมุมของแขนกล

สำหรับค่าของมุมของแต่ละข้อต่อที่ป้อนให้กับสมการ Direct kinematic จะถูกอ้างอิงจากตำแหน่งโฮม ดังนั้น ในการป้อนค่ามุมของแต่ละข้อต่อจำเป็นต้องรู้ขนาดและทิศทางของมุมที่จะป้อนด้วย โดยปกติจะกำหนดให้ที่ตำแหน่งโฮม ของทุกข้อต่อมีมุมเป็น 0 องศา ตัวอย่างการอ้างอิงมุมดังรูปที่ 5.4



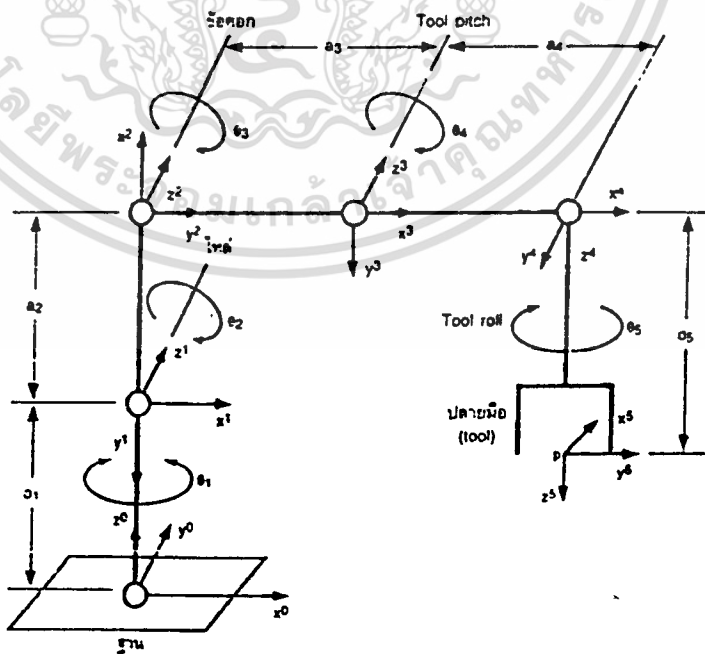
รูปที่ 5.4 ตำแหน่งโฮม ของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 แสดงตัวอย่างในการอ้างอิงมุม

ซึ่งเป็นลักษณะตำแหน่งโฮม อีกรูปแบบหนึ่ง ค่าของมุมของทุกข้อต่อในตำแหน่งนี้จะเป็น 0 องศา  $\theta_1, \theta_2, \theta_3, \theta_4$  และ  $\theta_5$  กำหนดให้เป็น 0 องศา สมมติต้องการเคลื่อนเฉพาะไป +90 องศา รูปแขนกลที่ได้จะเป็นดังรูปที่ 5.5



รูปที่ 5.6 ตำแหน่งด้านของแขนกลชนิดอาร์ตีกูเลต แบบ 5 ดีกรีออฟฟร็ดอม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 5.5 ค่าของมุมอื่นๆ ยังคงเป็น 0 องศาเฉพาะค่า  $\theta_4$  เท่านั้นที่เปลี่ยนแปลง ซึ่งถ้าเป็นค่าลบ มุมก็จะตรงข้ามกับทิศทางบวกโดยเริ่มนับจากตำแหน่งที่กำหนดให้เป็นตำแหน่งอ้างอิง ดังนั้นมุมที่ข้อต่ออื่นๆ ก็จะมีวิธีการอ้างอิงมุมในลักษณะเหมือนกัน

ในรูปที่ 5.6 จะแสดงทิศทางของเวกเตอร์ในแต่ละข้อต่อรวมทั้งแสดงค่าพารามิเตอร์ต่างๆ ที่เกี่ยวข้องกับซึ่งสามารถที่จะอธิบายแยกเป็นส่วนๆ ดังนี้

- $\theta_1$  คือมุมในการกวาดแขนไปมารอบฐาน
- $\theta_2$  คือในการหมุนไหล่ขึ้นลง
- $\theta_3$  คือมุมในการหมุนข้อศอก
- $\theta_4$  คือมุมในการหมุนข้อมือแบบหักขึ้นลง ( toll pitch )
- $\theta_5$  คือมุมในการหมุนข้อมือในลักษณะหน้ามือ ไปหลังมือ ( toll roll )

การหมุนข้อมือ มีการหมุนอยู่ใน 3 ลักษณะคือ YAW , PITCH และ ROLL สำหรับแขนกลที่ใช้อ้างอิงนี้เป็นแบบ 5 ดีกรีออฟฟรีดอม โดยการตัดการเคลื่อนที่ของข้อมือแบบ YAW ไป ดังนั้น ถ้าแขนกลชนิดอาร์ติคูลेट แบบ 6 ดีกรีออฟฟรีดอม ก็จะมีการเคลื่อนที่แบบ YAW นี้ด้วย ซึ่งจะทำให้แขนกลมีลักษณะการเคลื่อนที่ของข้อต่อทุกข้อคล้ายกับแขนของมนุษย์มากที่สุด

สำหรับความจำเป็นของการมีจำนวน ดีกรีออฟฟรีดอม เนื่องจากดีกรีออฟฟรีดอมเป็นการเคลื่อนที่อิสระของแต่ละข้อต่อถ้ามีจำนวนดีกรีออฟฟรีดอมมาก แขนกลก็สามารถเคลื่อนไหวได้จับซ้อนขึ้น แต่การสร้างและการควบคุมก็จะซับซ้อนขึ้นเป็นเงาตามตัวด้วย

สำหรับเวกเตอร์  $X_0, Y_0, Z_0$  จนถึง  $X_5, Y_5, Z_5$  เป็นเวกเตอร์ที่แสดงแนวการเคลื่อนที่ ส่วนค่าพารามิเตอร์ที่เป็นค่าคงที่ของแขนกลมีดังนี้

- $d_1$  คือระยะระหว่างฐาน ไปยังจุดหมุนที่ใหญ่
- $a_2$  คือระยะห่างจากจุดหมุนที่ใหญ่ไปยังจุดหมุนที่ศอก
- $a_3$  คือระยะห่างจากจุดหมุนที่ศอก ไปยังจุดหมุนที่ข้อมือหมุนแบบ PITCH
- $a_4$  คือระยะห่างจากจุดหมุนที่ข้อมือแบบ PITCH ไปยังจุดหมุนข้อมือหมุนแบบ ROLL
- $d_5$  คือระยะห่างจากจุดหมุนที่ข้อมือหมุนแบบ ROLL ไปยังจุดปลายมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการป้อนค่ามุม  $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$  และค่าพารามิเตอร์  $d_1, a_2, a_3, a_4, d_5$  ให้กับสมการ direct kinematic จะได้ตำแหน่ง  $P (X, Y, Z)$  ในพื้นที่ 3 มิติที่มีมือไปวางอยู่เนื่องจากที่มาของสมการ ได้มาจากการวิเคราะห์ทางด้านเวกเตอร์ ฉะนั้นถ้าผู้ใดสนใจ สามารถค้นคว้าเพิ่มเติมได้จากหนังสืออ้างอิง โดยสมการมีดังนี้

$$X = C_1 (a_2 C_2 + a_3 C_2 + a_4 C_{234} - d_5 S_{234})$$

$$Y = S_1 (a_2 C_2 + a_3 C_{23} + a_4 C_{234} - d_5 S_{234})$$

$$Z = d_1 - a_2 S_2 - a_3 S_{23} - a_4 S_{234} - d_5 C_{234}$$

จากสมการมีการใช้สัญลักษณ์เพื่อย่อความหมายให้สมการดูง่ายขึ้นความหมายของตัวย่อเหล่านี้คือ

$S_k$  คือ  $\sin k$

$C_k$  คือ  $\cos k$

$S_{ijk}$  คือ  $\sin (i + j + k)$

$C_{ijk}$  คือ  $\cos (i + j + k)$  โดยค่า  $i, j, k$  คือมุมค่าใด ๆ

ตัวอย่างเช่น  $C_1 = \cos \theta_1$  หรือ  $S_{234} = \sin (\theta_2 + \theta_3 + \theta_4)$  เป็นต้น

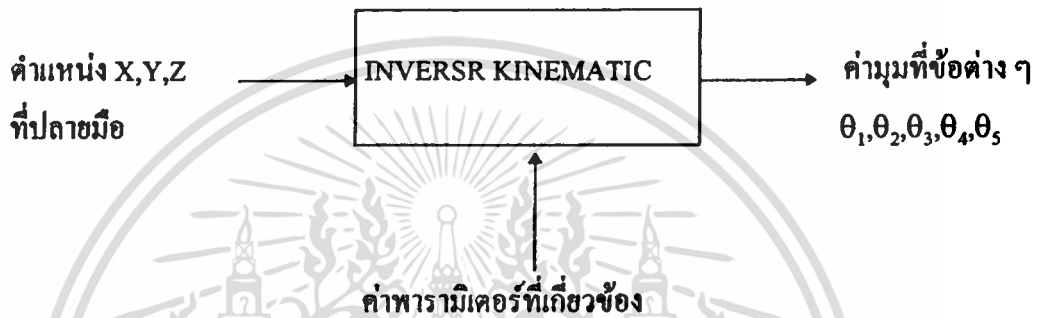
สำหรับแขนกลชนิดอาร์ติคูลเตด อื่นๆ ที่ไม่ใช่แขนกลชนิด แบบ S ดีกรีออฟฟรีดอมก็มีในสมการเช่นกัน

### 5.3 สมการ Inverse kinematic

จากที่ผ่านมามีการทราบความหมายของสมการกันไปแล้วในส่วนต่อมามีได้กล่าวถึงสมการในอีกลักษณะหนึ่ง ซึ่งอาจกล่าวได้ว่าเป็นหัวใจในการควบคุมแขนกลอีกแบบหนึ่งสมการชุดนี้คือสมการ inverse kinematic

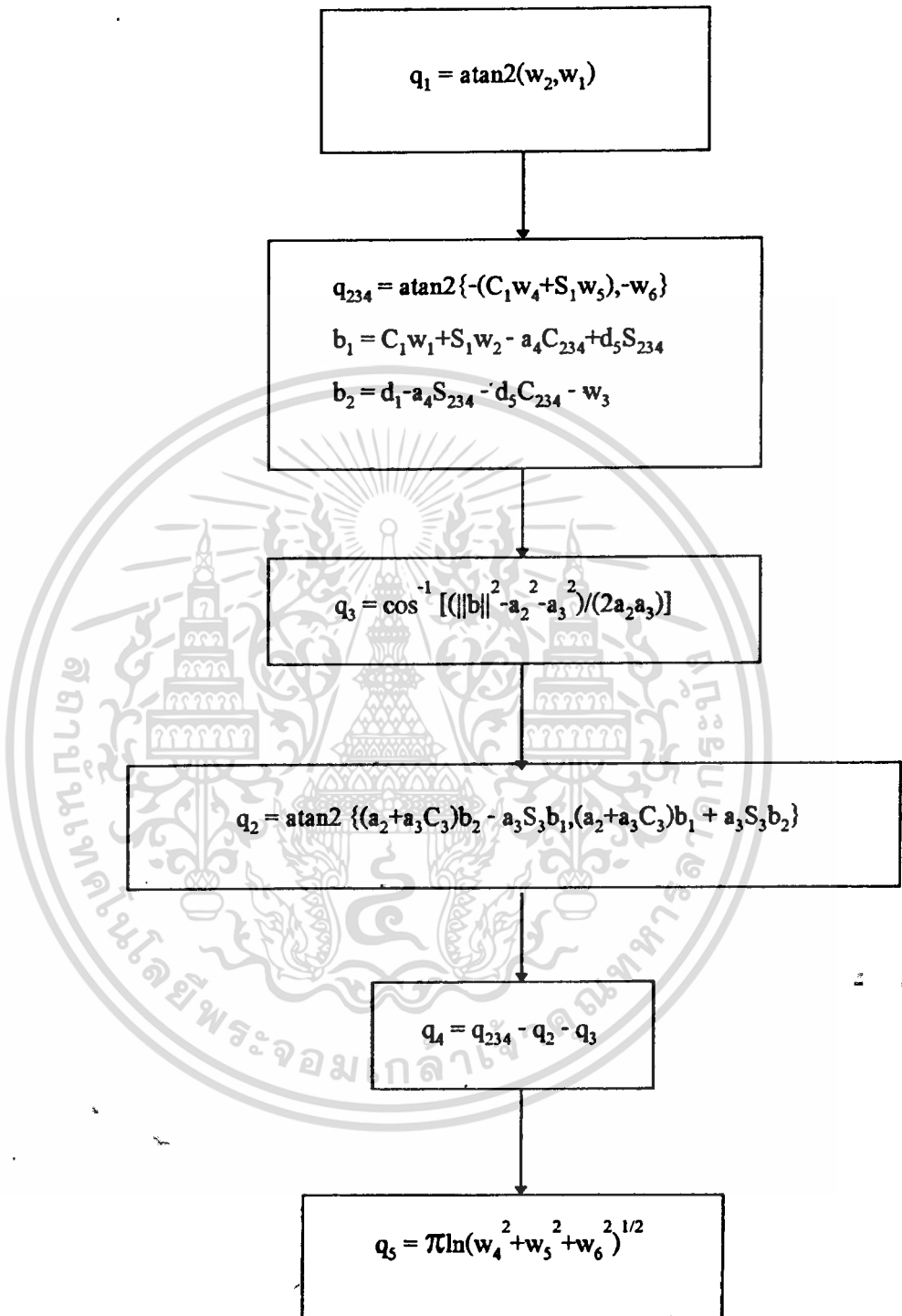
ความหมายของสมการจะมีความหมายที่ตรงข้ามกับสมการคือจากเดิมที่ป้อนค่าของมุมที่ข้อต่อต่างๆเข้าไปในสมการผลลัพธ์ที่ได้จะเป็นตำแหน่ง  $X, Y, Z$  ที่ปลายมือเคลื่อนไป แต่สมการนี้เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเปลี่ยนเป็นการป้อนตำแหน่ง  $X, Y, Z$  ของปลายมือเข้าไปแล้วจะได้ผลลัพธ์เป็นมุมของข้อต่อว่า ต้องเคลื่อนไปที่องศาจากตำแหน่งโฮม ปลายมือจึงจะไปอยู่ตำแหน่งที่ต้องการ บล็อกโคอะแกรม แสดงการทำงานจะแสดงดังในรูปที่ 5.7



รูปที่ 5.7 บล็อกโคอะแกรมของสมการ inverse kinematic

จากรูปที่ 5.7 โดยค่าพารามิเตอร์ต่างๆยังคงใช้ค่าเหมือนเดิมทุกประการจากความหมายของสมการ inverse kinematic เมื่อป้อนตำแหน่ง  $X, Y, Z$  ที่ปลายมือและค่าพารามิเตอร์ที่เกี่ยวข้อง ผลลัพธ์จะได้เป็นมุมของข้อต่อต่างๆคือ  $\theta_1$  ถึง  $\theta_5$  โดยสามารถเขียนเป็นอัลกอริทึมของสมการได้ดังรูปที่ 5.8



### รูปที่ 5.8 แสดงอัลกอริทึมของสมการ inverse kinematic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 5.8 แสดงลำดับการหาค่ามุมตามข้อต่อต่างๆ ซึ่งสามารถอธิบายความหมายของค่าตัวแปรต่างๆ ได้ดังนี้

$q_1$  ซึ่ง  $\theta_1$  เป็นมุมในการหมุนฐาน

$\tan 2$  คืออาร์คแทน 4 ควอดแดนท์

$q_2$  คือ  $\theta_2$  เป็นมุมที่ใหญ่

$q_3$  คือ  $\theta_3$  เป็นมุมที่ศอก

$q_4$  คือ  $\theta_4$  เป็นมุมการหมุนข้อมือคน แบบ PITCH

$q_5$  คือ  $\theta_5$  เป็นมุมการหมุนข้อมือคน แบบ ROLL

$q_{234}$  คือ มุมการหมุนข้อมือแบบ PITCH โดยอ้างอิงจากระนาบพื้น  $(X_0, Y_0, Z_0)$

$w_1$  คือตำแหน่งปลายมือที่ X

$w_2$  คือตำแหน่งปลายมือที่ Y

$w_3$  คือตำแหน่งปลายมือที่ Z

$w_4, w_5, w_6$  คือค่าในการหมุนข้อมือ

$C_1$  คือ  $\cos \theta_1$

$S_1$  คือ  $\sin \theta_1$

$C_3$  คือ  $\cos \theta_3$

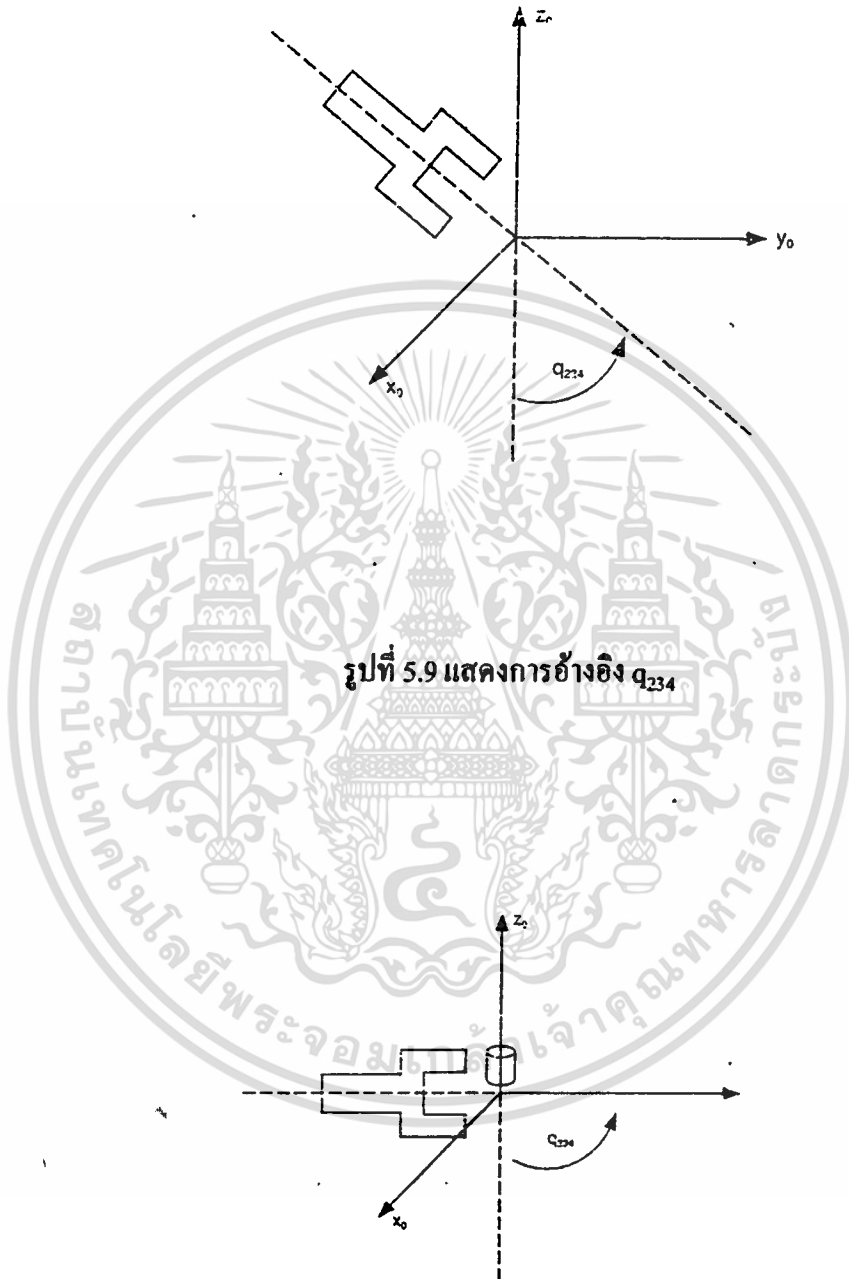
$S_3$  คือ  $\sin \theta_3$

$C_{234}$  คือ  $\cos$  ของมุม  $q_{234}$

$S_{234}$  คือ  $\sin$  ของมุม  $q_{234}$

$$\|b\| = b_1^2 + b_2^2$$

สำหรับ  $q_{234}$  นั้นสามารถอธิบายรายละเอียดโดยรูปที่ 5.9 ประกอบจากความหมายของ  $q_{234}$  เป็นมุมการหมุนข้อมือแบบ PITCH โดยอ้างอิงจากระนาบพื้น  $X_0, Y_0, Z_0$  ในการวัดมุมจะวัดเปรียบเทียบกับแกน  $Z_0$

รูปที่ 5.9 แสดงการอ้างอิง  $q_{234}$ รูปที่ 5.10 ตัวอย่างการอ้างอิง  $q_{234}$ 

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สังเกตได้ว่า  $q_{234}$  เป็นมุมที่มีอิทธิพลกับระนาบพื้น ในที่นี้คือ รูปแบบในการวางมือในการเคลื่อนที่ไปยังตำแหน่งที่ต้องการ ซึ่งที่มาของ  $q_{234}$  มาจากเวกเตอร์  $w_4, w_5, w_6$  ตัวอย่างเช่น ต้องการเคลื่อนมือ ไปจับวัตถุทางด้านข้างแสดงดังรูปที่ 5.10 มุมซึ่งเป็นมุมในการวางมือเมื่อเทียบกับระนาบพื้นของวัตถุจะมีค่าเป็น  $-90$  องศา

โดยทั่วไปค่าอินพุตที่ป้อนให้กับอัลกอริทึมของสมการ inverse kinematic จะเป็นค่า  $w_1, w_2, w_3, q_{234}$  และ  $q_5$  (ตำแหน่งของวัตถุ, มุมในการวางมือ, มุมในการหมุนข้อมือแบบ ROLL) ตามลำดับ ดังนั้น ในการทำงานจริงๆ จะพบว่าอัลกอริทึมของสมการ inverse kinematic ในรูปที่ 5.8 จะมีบางส่วนที่สามารถตัดออกไปได้เพื่อลดความซับซ้อนของสมการ เนื่องจากในการควบคุมแขนกลโดยใช้อัลกอริทึมของสมการ inverse kinematic ค่า  $q_{234}$  ซึ่งเป็นมุมในการวางมือเมื่อเทียบกับระนาบของวัตถุและมุมในการหมุนข้อมือแบบ ROLL ผู้ควบคุมจะเป็นผู้กำหนดลงไปเองโดยไม่ต้องผ่านอัลกอริทึมของสมการ inverse kinematic ซึ่งสมการส่วนที่สามารถตัดออกไปได้คือ

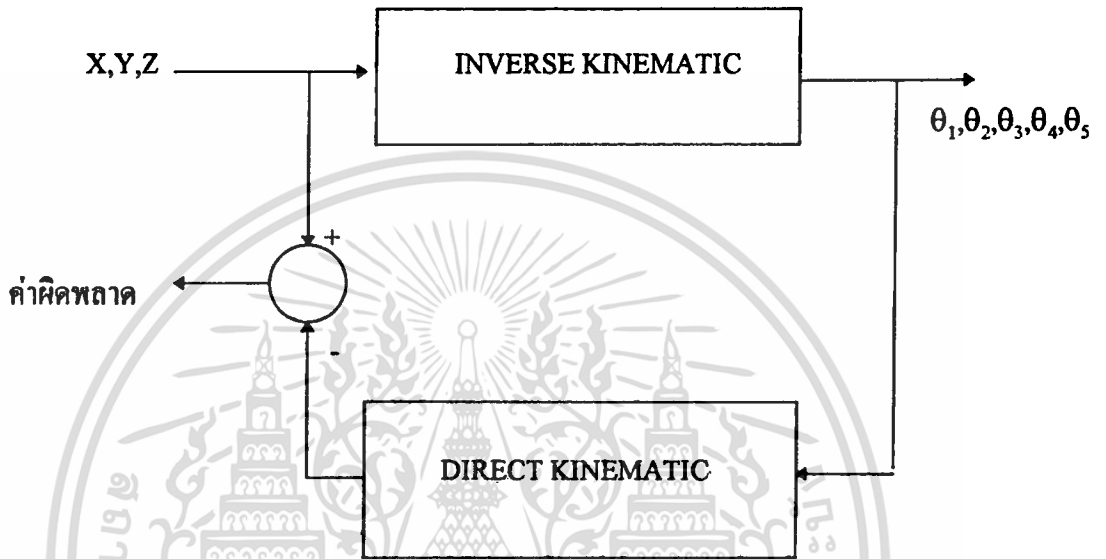
$$q_{234} = \text{atan2} [-(C_1 w_4 - S_1 w_5), -w_6] \text{ และ } q_5 = \pi \ln (w_4^2 + w_5^2 + w_6^2)^{1/2}$$

เอาท์พุตที่ได้จะเหลือเพียงมุม  $\theta_1, \theta_2, \theta_3$  และ  $\theta_4$  นอกจากอัลกอริทึมของสมการ inverse kinematic ของแขนกลชนิดอาร์ติคูลेट แบบ 5 ดีกรีออฟฟรีดอม นี้แล้วยังมีของแขนกลชนิดอื่นๆ อีก รวมทั้งที่มาของสมการนี้ด้วย สามารถค้นคว้าเพิ่มเติมได้จากหนังสืออ้างอิงได้เช่นกัน

#### 5.4 การประยุกต์ใช้งาน

สำหรับการประยุกต์ใช้งานอัลกอริทึมของสมการ inverse kinematic อันดับแรกที่จะเห็นได้ชัดคือ ควบคุมการจับวัตถุที่ตำแหน่ง X, Y, Z ด้านทาง ให้ไปวางที่ตำแหน่ง X, Y, Z ปลายทาง โดยคอมพิวเตอร์จะเป็นตัวคำนวณค่าของมุมที่ข้อต่อต่างๆ เมื่อคำนวณเสร็จ คอมพิวเตอร์จะส่งค่ามุมต่างๆ ไปให้กับชุดขับเคลื่อน ชุดขับเคลื่อนก็จะทำการขับเคลื่อนข้อต่อต่างๆ ไปตามมุมที่ได้มาจากอัลกอริทึมของสมการ inverse kinematic ผลลัพธ์ที่ปลายมือจะอยู่ที่ตำแหน่งที่ต้องการ จากนั้นก็เป็นหน้าที่ของผู้เขียน โปรแกรมควบคุมว่าจะให้ทำอะไรต่อไป

สำหรับสมการ direct kinematic ที่ใช้งานอยู่บางส่วนใช้ในการตรวจสอบตำแหน่งในการเขียนโปรแกรมจำลองหรือจำลอง การทำงานของแขนกล โดยจะใช้สมการ direct kinematic เป็นชุดป้อนกลับ แสดงดังรูปที่ 5.11



รูปที่ 5.11 บล็อกไดอะแกรมในการตรวจสอบตำแหน่งด้วยสมการ direct kinematic

จากบล็อก ไดอะแกรมค่าอาร์ทพูทของอัลกอริทึมของสมการ inverse kinematic จะถูกสมการ direct kinematic นำมาเป็นอินพุทเพื่อป้อนกลับไปหาค่าผิดพลาดในการเคลื่อนที่ ค่าผิดพลาดที่ได้จะถูกส่งไปยังโปรแกรมอีกชุดหนึ่ง ซึ่งทำหน้าที่เป็นตัวควบคุม ถ้าเปรียบเทียบแขนกลจริงๆ กับโปรแกรมจำลองการทำงานนั้น สมการ direct kinematic ก็คือตัวตรวจสอบนั่นเอง

## บทที่ 6

### การควบคุมและโปรแกรมการควบคุม

การใช้คอมพิวเตอร์ควบคุมการทำงานอุปกรณ์ที่เกี่ยวข้องกับการทำงานเชิงอนาล็อก หรือ อุปกรณ์ที่ต้องใช้กำลังไฟสูง ต้องมีวงจรอินเตอร์เฟส เป็นตัวแปลงเสมอ เพราะว่า คอมพิวเตอร์มีการส่งสัญญาณควบคุมเป็นแบบดิจิทัล มีไฟเลี้ยง +5 V จึงไม่สามารถจะควบคุม อุปกรณ์ได้โดยตรง

วงจรอินเตอร์เฟส นี้จะใช้ IC Microcontroller MCS - 51 เป็นตัวประมวลผลข้อมูลที่ส่งมาจากคอมพิวเตอร์ ก่อนที่จะส่งไปยังวงจรขับสเตปปีงมอเตอร์ เพื่อที่จะขับสเตปปีงมอเตอร์ ให้ทำงานจะทำให้แขนกลสามารถเคลื่อนที่หรือทำงานได้ตามที่เราต้องการ

#### โครงสร้างของไมโครคอนโทรลเลอร์ตระกูล MCS - 51

ไมโครคอนโทรลเลอร์ตระกูล MCS - 51 มีสมาชิกในตระกูลหลายเบอร์ด้วยกัน แต่ละเบอร์จะมีคุณสมบัติพิเศษบางอย่างแตกต่างกัน เช่น มี หน่วยความจำภายในสำหรับเก็บ โปรแกรม และข้อมูลภายในชิปเพิ่มขึ้น มีวงจรเปลี่ยนค่าสัญญาณอะนาล็อกเป็นดิจิทัลในตัว สามารถรับสัญญาณอินเตอร์รัปต์ได้หลายชนิด ทำกระบวนการ DMA (Direct Memory Access) ได้ในตัว มีรีจิสเตอร์สำหรับใช้เป็น ไทม์เมอร์หรือเคาน์เตอร์เพิ่มขึ้น

ไมโครคอนโทรลเลอร์เบอร์ที่นับได้ว่าเป็นเบอร์พื้นฐานสำหรับตระกูล MCS - 51 นี้ได้แก่ เบอร์ 8051, 8031, 8751 โดยเบอร์ 8051 จัดเป็นสมาชิกตัวแรกในตระกูล ซึ่งมีหน่วยความจำสำหรับเก็บ โปรแกรมภายในชิปเป็น ROM ขนาด 4 กิโลไบต์ และหน่วยความจำสำหรับเก็บข้อมูลทั่วไปภายใน MCS - 51 (RAM) เองจำนวน 128 ไบต์ มีพอร์ตขนาด 8 บิต 4 พอร์ต มีรีจิสเตอร์สำหรับใช้เป็น ไทม์เมอร์หรือเคาน์เตอร์ขนาด 16 บิตรวม 2 ตัว รับสัญญาณอินเตอร์รัปต์จากภายนอกได้ 2 ชนิด สามารถรับส่งข้อมูลแบบอนุกรมผ่านทางพอร์ตสื่อสารข้อมูลแบบอนุกรม มีวงจร ออสซิลเลเตอร์เพื่อสร้างสัญญาณนาฬิกาควบคุมการทำงานในตัวเอง ส่วนเบอร์ 8751 จะมี

คุณสมบัติเหมือนเบอร์ 8051 ทุกอย่าง ต่างกันเพียงชนิดของหน่วยความจำสำหรับเก็บโปรแกรม ภายในชิปของเบอร์ 8751 จะเป็น EPROM แทนที่จะเป็น ROM ส่วนเบอร์ 8031 จะเหมือนกับเบอร์ 8051 ต่างกันเพียงในเบอร์ 8031 ไม่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิปเท่านั้น

ไมโครคอนโทรลเลอร์ตระกูล MCS - 51 ทุกเบอร์ใช้แรงดันไฟเพียง 5 โวลต์ในการทำงาน ส่วนกระแสไฟฟ้าที่ใช้จะแตกต่างกันไปตามชนิดของเทคโนโลยีที่ใช้ในการผลิตเบอร์ของไมโครคอนโทรลเลอร์ตระกูลนี้ที่มีตัวอักษร C อยู่ตรงกลางเบอร์ เช่น 80C31, 80C51 จะเป็นเบอร์ของชิปที่ผลิตโดยอาศัยเทคโนโลยี CHMOS ซึ่งใช้พลังงานในการทำงานน้อยกว่าและสามารถควบคุมการใช้พลังงานของตัวชิปได้จากโปรแกรมเพื่อการประหยัดพลังงานในระบบ

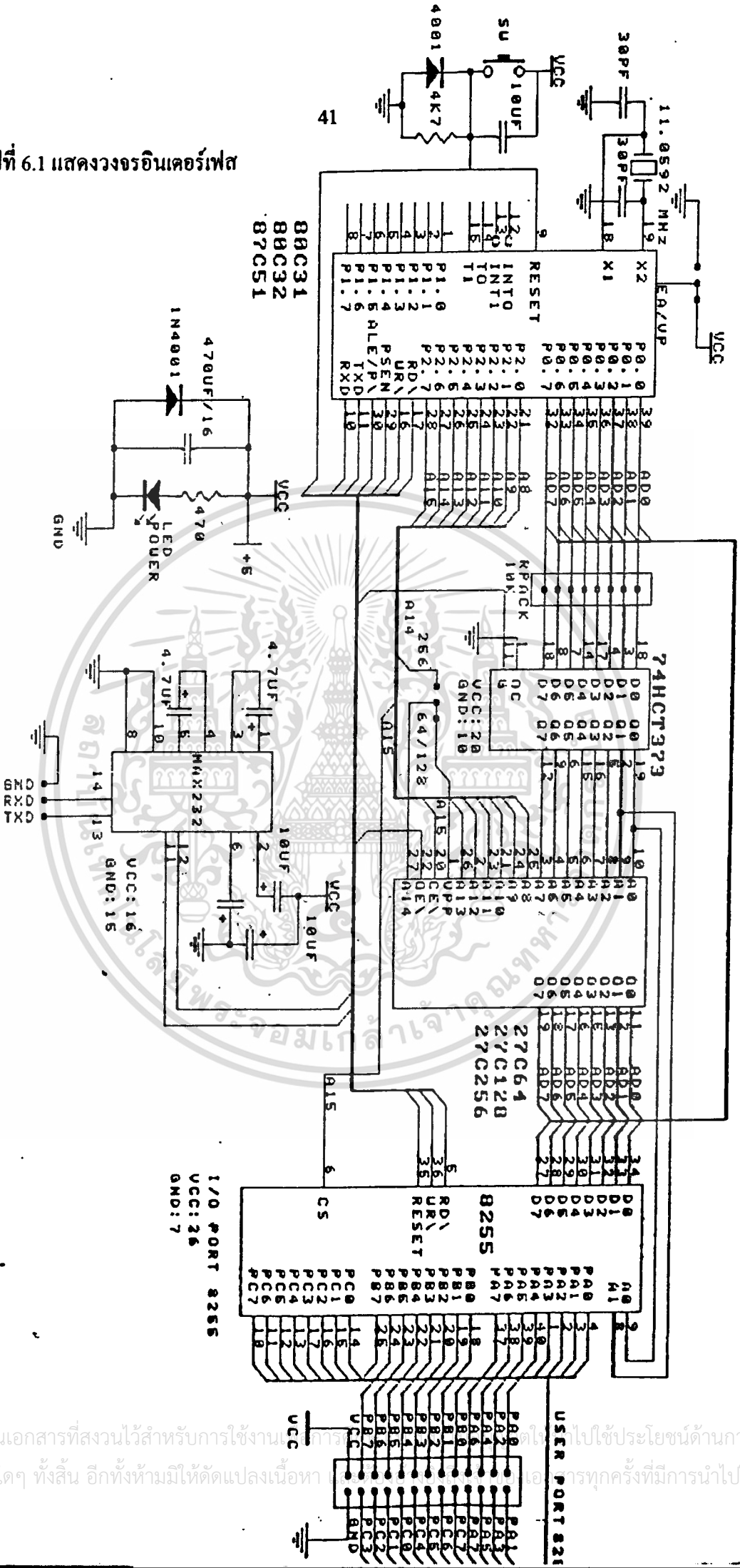
MCS - 51 เป็นตระกูลของไมโครคอนโทรลเลอร์ที่ถูกพัฒนาขึ้นมาจากตระกูล MCS - 48 ดังนั้นจึงมีความสามารถเหนือกว่าหลายอย่าง ซึ่งจะไม่ได้กล่าวรายละเอียดไว้ในที่นี้ แต่จะเปรียบเทียบให้เห็นถึงข้อดีของ MCS - 51 เมื่อเทียบกับ MCS - 48 ให้เห็นเป็นบางช่วง เช่น ความเร็วในการประมวลผลของ MCS - 51 สามารถใช้ความถี่ได้ถึง 12 เมกะเฮิร์ตซ์ หรือสำหรับบางเบอร์ในตระกูลสามารถใช้ได้ถึง 16 เมกะเฮิร์ตซ์ ทำให้ช่วงเวลาในการทำงานแต่ละคำสั่งน้อยมาก เมื่อใช้ความถี่ 12 เมกะเฮิร์ตซ์ คำสั่งที่ใช้เวลาน้อยที่สุดจะใช้เวลาเพียง 1 ไมโครวินาที ส่วนคำสั่งที่ใช้เวลามากที่สุดเพียง 4 ไมโครวินาทีเท่านั้น

### 6.1 วงจรอินเตอร์เฟส (Interface)

วงจรอินเตอร์เฟส ที่ใช้ในโครงงานนี้ จะใช้ไมโครคอนโทรลเลอร์เบอร์ 80C31 คู่ร่วมกับชิปรีพอร์ทเบอร์ 8255 ซึ่งทำหน้าที่เป็นพอร์ตอินพุท/เอาต์พุท EPROM ขนาด 16 กิโลไบต์ เพื่อใช้เก็บโปรแกรมที่ใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์ และชิปเบอร์ MAX232 ซึ่งเป็นพอร์ตการสื่อสารแบบอนุกรมเพื่อให้ไมโครคอนโทรลเลอร์สามารถติดต่อกับคอมพิวเตอร์ได้

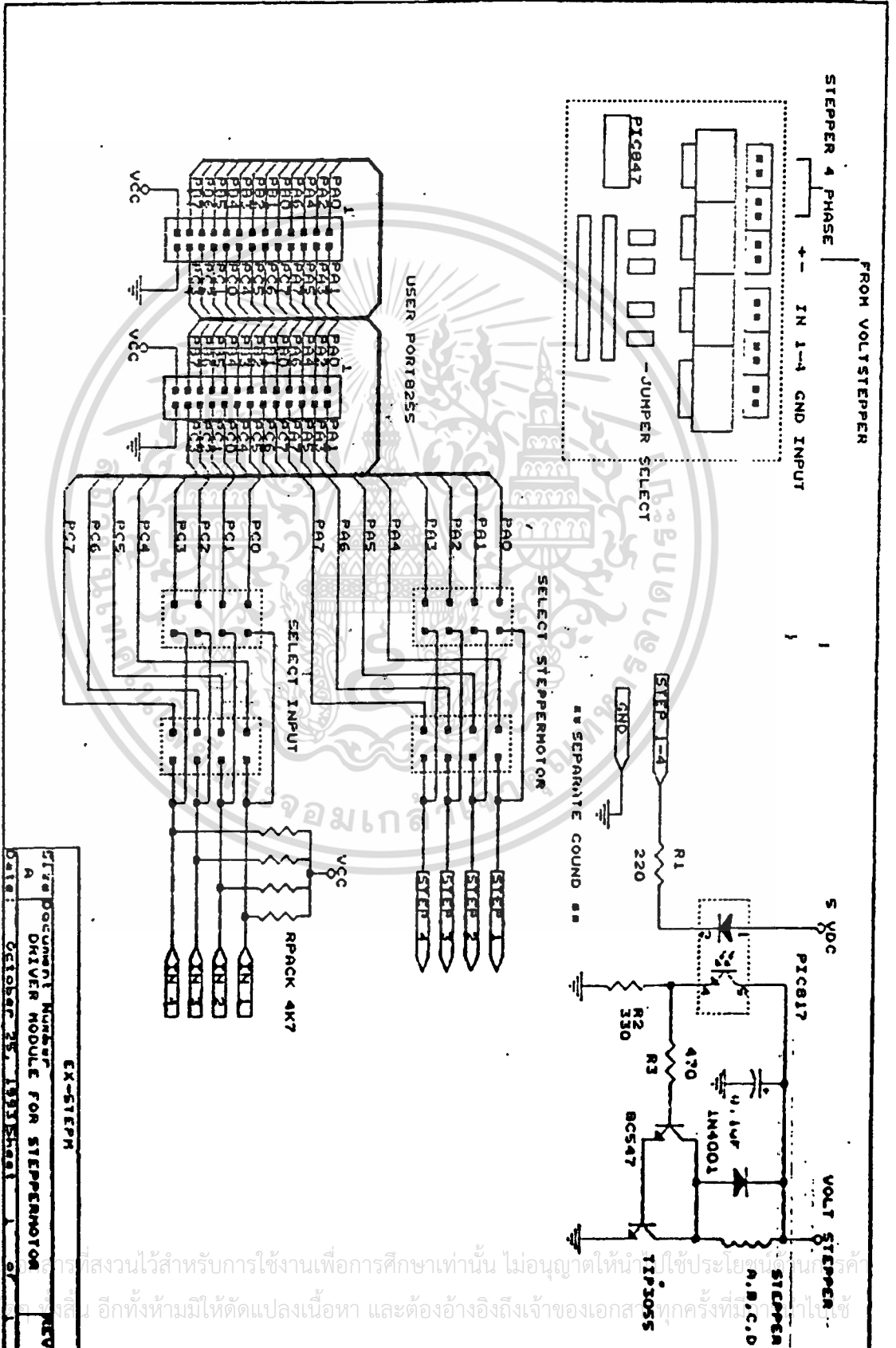
ชิปเบอร์ 8255 ที่ใช้เป็นอินพุท/เอาต์พุทพอร์ต จะใช้ 2 ตัว โดยแต่ละตัวจะมีอินพุท/เอาต์พุท 24 พอร์ต ตัวแรกจะใช้ควบคุมสเตปป์มอเตอร์ 6 ตัว โดยต่อเข้ากับวงจรขับสเตปป์มอเตอร์ 6 ชุด ทำหน้าที่เป็นเอาต์พุทพอร์ตอย่างเดียว ส่วนตัวที่สองทำหน้าที่เป็นอินพุทพอร์ตรับสัญญาณฟีดแบคจากเซ็นเซอร์ต่างๆ

รูปที่ 6.1 แสดงวงจรอินเทอร์เฟส



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในเท่านั้น ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาต  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา

รูปที่ 6.2 แสดงวงจรขับสเตปปีงมอเตอร์



EX-STEPM  
 Step Document Number  
 A DRIVER MODULE FOR STEPMOTOR  
 Date: October 25, 1991

รูปที่ 6.1 แสดงวงจรอินเทอร์เฟซ ประกอบด้วย ไมโครคอนโทรลเลอร์ 80C31 ชิปอินพุท/เอาต์พุท เบอร์ 8255 ชิปสื่อสารแบบอนุกรม เบอร์ MAX232 EPROM สามารถเลือกได้ว่าจะใช้เบอร์อะไรขึ้นกับขนาดหน่วยความจำ เช่นเบอร์ 27C128 มีขนาดหน่วยความจำ 16 กิโลไบต์ และชิปเบอร์ 74HCT373 ทำหน้าที่เป็นบัฟเฟอร์ข้อมูลระหว่างไมโครคอนโทรลเลอร์กับ EPROM

รูปที่ 6.2 แสดงวงจรขับเคลื่อนแป้นพิมพ์มอเตอร์ วงจรจะแสดงเพียง 1 ชุด แต่ในการใช้งานจริงในการขับเคลื่อนหนึ่งตัวจะต้องใช้ทั้งหมด 4 ชุด เพราะในการควบคุมมอเตอร์หนึ่งตัวจะต้องใช้สัญญาณควบคุม 4 บิต

## 6.2 โปรแกรมการควบคุม

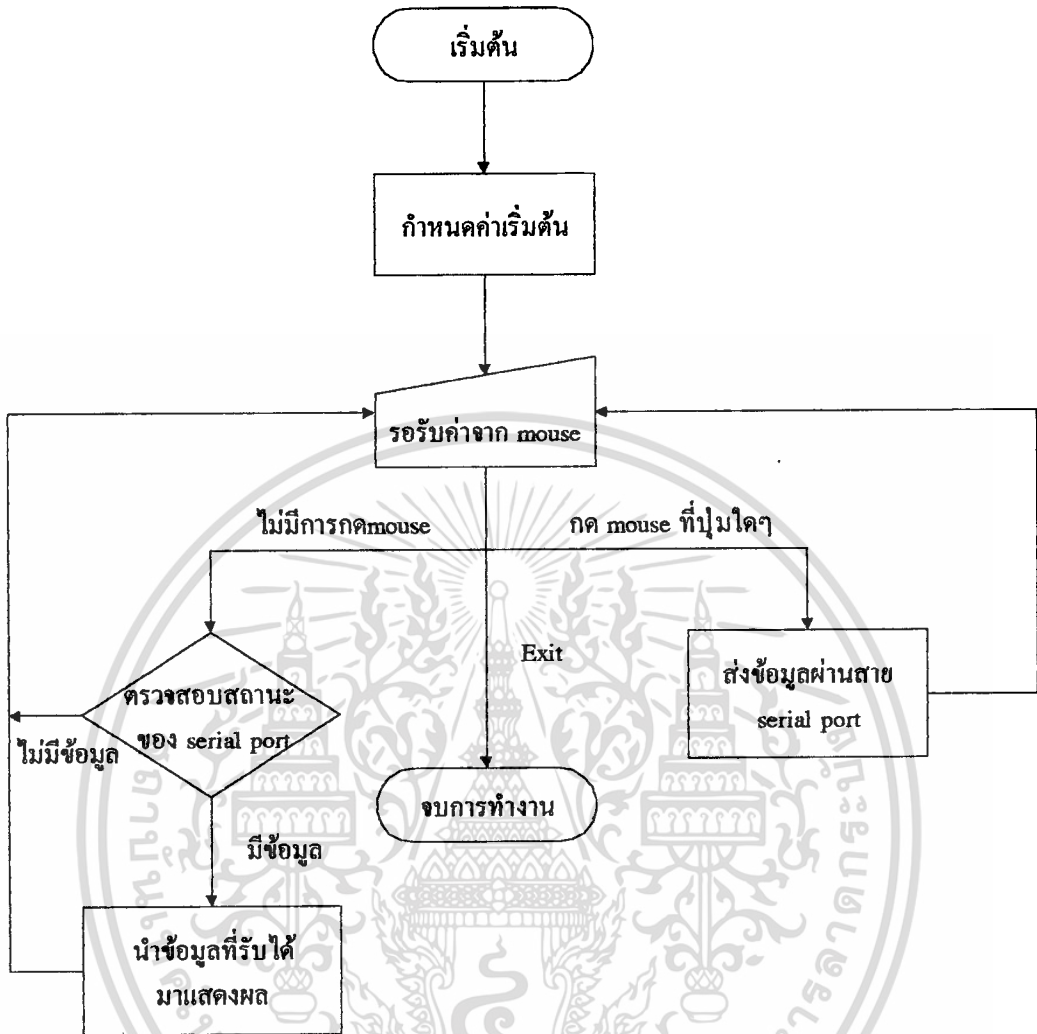
ในการควบคุมการทำงานของแขนกลต้องเขียนโปรแกรมควบคุมที่เครื่องคอมพิวเตอร์และที่อินเทอร์เฟซ โดยที่เครื่องคอมพิวเตอร์ใช้โปรแกรมภาษาซี ( C Language ) ทำการส่งข้อมูลให้อินเทอร์เฟซและรับข้อมูลจากอินเทอร์เฟซกลับมา ส่วนที่อินเทอร์เฟซใช้ภาษาแอสเซมบลีสำหรับ MCS 51 เขียนโปรแกรมเพื่อแปลคำสั่งจากเครื่องคอมพิวเตอร์ไปควบคุมการทำงานของแป้นพิมพ์มอเตอร์ต่าง ๆ

### 6.2.1 โปรแกรมภาษาซี

โปรแกรมภาษาซีมีรายละเอียดดังต่อไปนี้

Project.prj	ใช้เรียกโปรแกรมหลาย ๆ ไฟล์ เพื่อคอมไพล์ (Compile ) รวมกัน
Project.c	เป็นโปรแกรมหลักใช้เรียกฟังก์ชันจากโปรแกรมอื่นและจากตัวมันเอง
Mouse.c	เป็นโปรแกรมที่ใช้เก็บคำสั่งต่าง ๆ ของ Mouse
Graph1.c	เป็นโปรแกรมที่ใช้เก็บคำสั่งที่เกี่ยวกับการจัดจอภาพ
Mouse.h	เป็นโปรแกรมที่ใช้เก็บชื่อคำสั่งและตัวแปรที่ใช้เกี่ยวกับ Mouse

รูปที่ 6.3 flowchart แสดงการทำงานของโปรแกรมควบคุมบนเครื่องพีซี

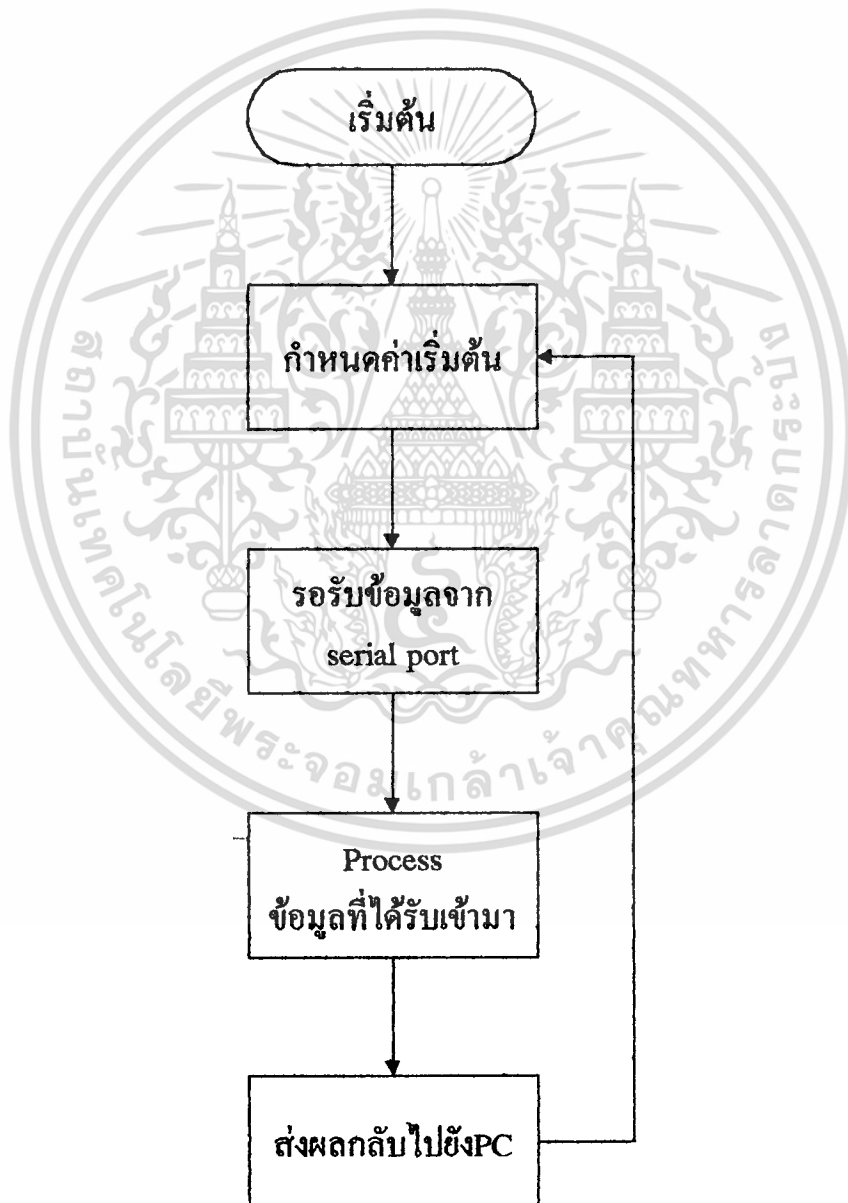


รูปที่ 6.3 flowchart แสดงการทำงานของโปรแกรมควบคุมบนเครื่อง PC

## 6.2.2 โปรแกรมภาษาแอสเซมบลี

โปรแกรมภาษาแอสเซมบลีมีรายละเอียดดังต่อไปนี้

Comm.asm เป็นโปรแกรมที่ใช้ควบคุมการทำงานทั้งหมดของอินเทอร์เฟซ  
รูปที่ 6.4 flowchart แสดงการทำงานของโปรแกรมควบคุมบน MCS-51



รูปที่ 6.4 flowchart แสดงการทำงานของโปรแกรมควบคุมบน MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 7

### ผลการทดลอง

#### ด้านกลไก

ส่วนฐานในการทดลองเลือกเพื่อเปลี่ยนอัตราทด เพื่อควบคุมการหมุนของฐานให้ราบเรียบ และได้ความละเอียดสูง มอเตอร์ที่ใช้ก็สามารถทำงานได้ดี

ส่วนเฟืองที่หัวไหลใช้สเตปป์มอเตอร์ตัวค่อนข้างใหญ่ เพราะต้องรับแรงมากจากโมเมนต์ที่เกิดขึ้น ในการเลือกอัตราทดเฟืองพบว่ามีความเร็วเพียงพอเหมาะสมกับการใช้งาน แต่ค่อนข้างจะมีเสียงดัง เนื่องจากชุดเฟืองที่ทำงานได้ไม่สมบูรณ์เต็มที่ เพราะเกิดจากความผิดพลาดของแกนและนูนต่างๆ

ส่วนข้อศอกใช้सानพานส่งกำลัง ก็สามารถทำงานได้สม่ำเสมอดี แต่มีแรงน้อยเพราะมอเตอร์ตัวเล็กและมีโหลดมาก การทำงานค่อนข้างช้าเพราะทดผ่านพูลเลย์มากเพื่อเพิ่มแรงบิด

ส่วนข้อมือส่งกำลังผ่านसानพานจากชุดพูลเลย์ที่ฐานขึ้นไปทำงานเหมาะสมดีมาก

ส่วนหมุนข้อมือทำงานได้ช้ามากเพราะใช้สเตปป์มอเตอร์ทำงาน 8 ตัว แต่เป็นขนาดเล็กติดที่ด้านบนของแขน

ในส่วนมือจับ ( gripper ) มีแรงในการบีบจับวัตถุได้ดี ระบบกลไกทำงานได้ดี ชุดเฟืองขับฟันกันได้ตรง

ส่วนเซ็นเซอร์นั้นน้ำหนักการบีบที่ปลายมือ ออกแบบโดยใช้ สเตรนเกต (strain gate) ยังไม่สามารถทำงานได้ เนื่องจากยังติดปัญหาด้านวงจรอินเทอร์เฟส

## ปัญหาในการสร้าง

### ด้านฮาร์ดแวร์

ในโครงการชิ้นนี้พบว่ามีปัญหาในการสร้างทั้งในด้านกลไกคือ ตัวสเตปปีงมอเตอร์พบว่ามีความบิดเบี้ยวมากไม่พอเพียงจะนำมาใช้ขับเคลื่อนกลซึ่งมีน้ำหนักมาก และจังหวะการทำงานเป็นสเตปไม่ต่อเนื่อง ทำให้ส่วนที่มีอัตราทดต่ำขาดความราบเรียบในการเคลื่อนที่และการสร้างยังมีความผิดพลาดในตัวเฟือง, พู่เล่ย์ที่จะส่งผลให้เกิดความผิดพลาดในการใช้งานจริง ตัวสเตปมอเตอร์นั้นกินไฟมาก ทำให้วงจรที่ออกแบบมาต้องทนกระแสสูงมาก

### ด้านซอฟต์แวร์

ในการรับส่งข้อมูลหลาย ๆ ไบต์ จะมีปัญหาเรื่องการชิงโครไนซ์ของข้อมูลคือเมื่อเกิดข้อมูลหายไป 1 ไบต์ จะทำให้คำสั่งที่ส่งไปผิดทั้งหมด



ภาคผนวกไฟล์ Project.prj

project.c

graph1.c

mouse.c

ไฟล์ Project.c

```

#include "mouse.h"
#include <stdio.h>
#include <bios.h>
#include <string.h>
#include <dos.h>
#include <process.h>
#include <stdlib.h>
#include "graphics.h"
#include <math.h>

#define DATA_READY 0x100
#define PI 3.1416

```

```
int mode=0;
```

```
int select=0;
```

```
void main()
```

```
{
```

```
int gd = DETECT, gm = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

initgraph(&gd,&gm,"");
background();
initialize();
init_mouse(MOUSE_NEEDED,gd,gm);
for(;;){
    if (button_press(1)) break;
    if (button_press(0)){
        if (mouse_in_box(1,250,220,400,250)){
            bp(250,220,400,250);
            manual1();
            background();
            initialize();
            mouse_on(1);
        }
        if (mouse_in_box(1,250,280,400,310)){
            bp(250,280,400,310);
            bioscom(1,13,1);
        }
        if (mouse_in_box(1,250,340,400,370)){
            bp(250,340,400,370);
            break;
        }
    }
}
mouse_reset();
closegraph();
clrscr();
}

```

```
void button_p()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    struct viewporttype vp;
    getviewsettings(&vp);
    setcolor(DARKGRAY);
    line(2,2,(vp.right-vp.left)-2,2);
    line(2,(vp.bottom-vp.top)-2,2,2);

    setcolor(LIGHTGRAY);
    line((vp.right-vp.left)-2,2,(vp.right-vp.left)-2,(vp.bottom-vp.top)-2);
    line((vp.right-vp.left)-2,(vp.bottom-vp.top)-2,2,(vp.bottom-vp.top)-2);
}

```

```

bp(int q, int w, int e, int r)

```

```

{
    mouse_off(1);
    setviewport(q,w,e,r,1);
    delay(50);
    button_p();
    delay(50);
    button();
    mouse_on(1);
}

```

```

manual1()

```

```

{
    int m1stt,status;
    int abyte,abyte2;
    double c1=0,c2=0,c3=0,c4=0,c5=0;
    background();
    mouse_on(1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

manual();
m1stt=0;
dispstatus();
show();
bioscom(0,0xE3,1);
pos(c1,c2,c3,c4,c5);
for(;;){
/* status=bioscom(3,0,1);
if (status & 256){
a[1]=bioscom(2,0,1);
gotoxy(25,2);
printf("%5d",a[1]);
a[2]=bioscom(2,0,1);
gotoxy(35,2);
printf("%5d",a[2]);
}
if (abyte!=abyte1){
abyte1=abyte;
switch (select) {
case 1 :
x1=abyte1;
pos(x1,x2,x3,x4,x5);
break;
case 2 :
x2=abyte1;
pos(x1,x2,x3,x4,x5);
break;
case 3 :
x3=abyte1;
pos(x1,x2,x3,x4,x5);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break;
    case 4 :
        x4=abyte1;
        pos(x1,x2,x3,x4,x5);
        break;
    case 5 :
        x5=abyte1;
        pos(x1,x2,x3,x4,x5);
        break;
    }
} /*

if (button_press(1)) break;
if (button_press(0)){
if (mouse_in_box(1,200,100,260,120)){
    bp(200,100,260,120);
    if(mode==0) mode=1;
    else if(mode==1) mode=0;
    dispstatus();
    show();
}
if (mouse_in_box(1,100,140,170,160)){
    bp(100,140,170,160);
    select=1;
    bioscom(1,5,1);
    dispstatus();
    show();
}
if (mouse_in_box(1,100,180,170,200)){
    bp(100,180,170,200);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

select=2;
bioscom(1,6,1);
dispstatus();
show();
}
if (mouse_in_box(1,100,220,170,240)){
    bp(100,220,170,240);
    select=3;
    bioscom(1,7,1);
    dispstatus();
    show();
}
if (mouse_in_box(1,100,260,170,280)){
    bp(100,260,170,280);
    select=4;
    bioscom(1,8,1);
    dispstatus();
    show();
}
if (mouse_in_box(1,100,300,170,320)){
    bp(100,300,170,320);
    select=5;
    bioscom(1,9,1);
    dispstatus();
    show();
}
if (mouse_in_box(1,100,340,170,360)){
    bp(100,340,170,360);
    select = 6;
    bioscom(1,10,1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

dispstatus();
show();
}
if (mouse_in_box(1,380,340,390,350)){
    if (select == 6){
        bp(380,340,390,350);
        bioscom(1,14,1);
    }
}
if (mouse_in_box(1,380,350,390,360)){
    if (select == 6){
        bp(380,350,390,360);
        bioscom(1,14,1);
    }
}
if (mouse_in_box(1,400,340,450,360)){
    if (select == 6){
        bp(400,340,450,360);
        bioscom(1,12,1);
    }
}
if (mouse_in_box(1,400,370,450,390)){
    if (select == 6){
        bp(400,370,450,390);
        bioscom(1,11,1);
    }
}
if (mouse_in_box(1,220,380,270,400)){
    bp(220,380,270,400);
    break;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

}

```
if (mouse_in_box(1,400,140,440,220)){
```

```
    bp(400,140,440,220);
```

```
/*    switch (select) {
```

```
        case 1:
```

```
            c1=c1+(4*0.08143);
```

```
            pos(c1,c2,c3,c4,c5);
```

```
            break;
```

```
        case 2:
```

```
            c2=c2+(4*0.07438);
```

```
            c3=c3-(4*0.07438);
```

```
            c4=c4-(4*0.07438);
```

```
            pos(c1,c2,c3,c4,c5);
```

```
            break;
```

```
        case 3:
```

```
            c3=c3+(4*0.05625);
```

```
            c4=c4-(4*0.05625);
```

```
            pos(c1,c2,c3,c4,c5);
```

```
            break;
```

```
        case 4:
```

```
            c4=c4+(4*0.22500);
```

```
            pos(c1,c2,c3,c4,c5);
```

```
            break;
```

```
        case 5:
```

```
            c5=c5+(4*0.41538);
```

```
            pos(c1,c2,c3,c4,c5);
```

```
            break;
```

```
    }*/
```

```
if(mode==0) bioscom(1,3,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if(mode==1)
{
    if(m1stt==0)
    {
        m1stt=2;
        bioscom(1,4,1);
    }
    else if(m1stt==2)
    {
        m1stt=0;
        bioscom(1,255,1);
    }
    else if(m1stt==1)
    {
        m1stt=2;
        bioscom(1,4,1);
    }
}
}

if (mouse_in_box(1,400,240,440,320)){
    bp(400,240,440,320);
/*
    switch (select){
        case 1:
            c1=c1-(4*0.08143);
            pos(c1,c2,c3,c4,c5);
            break;
        case 2:
            c2=c2-(4*0.07438);
            c3=c3+(4*0.07438);
            c4=c4+(4*0.07438);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        pos(c1,c2,c3,c4,c5);
        break;

    case 3:
        c3=c3-(4*0.05625);
        c4=c4+(4*0.05625);
        pos(c1,c2,c3,c4,c5);
        break;

    case 4:
        c4=c4-(4*0.22500);
        pos(c1,c2,c3,c4,c5);
        break;

    case 5:
        c5=c5-(4*0.41538);
        pos(c1,c2,c3,c4,c5);
        break;
    } */
if(mode==0) bioscom(1,1,1);
else if(mode==1)
{
    if(m1stt==0)
    {
        m1stt=1;
        bioscom(1,2,1);
    }
    else if(m1stt==1)
    {
        m1stt=0;
        bioscom(1,255,1);
    }
    else if(m1stt==2)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {
            m1stt=1;
            bioscom(1,2,1);
        }
    }
}
};
};
clrscr();
}

dispstatus()
{
int minX,minY;
minX=0;
minY=0;
mouse_off(1);
setviewport(minX+200,minY+100,minX+260,minY+120,1);
setfillstyle(SOLID_FILL,LIGHTGRAY);
bar(1,1,60,20);
button();
if(mode==0){
    setcolor(BLACK);
    outtextxy(7,7,"Single");
}
else if(mode==1){
    setcolor(BLACK);
    outtextxy(15,7,"Fast");
}

mouse_on(1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

show()
{
int minX,minY;
minX=0;
minY=0;
if(select==0){
    setviewport(minX+200,minY+140,minX+260,minY+160,1);
    bar(0,0,100,20);
    outtextxy(9,7,"STOP");
    setviewport(minX+200,minY+180,minX+260,minY+200,1);
    bar(0,0,100,20);
    outtextxy(9,7,"STOP");
    setviewport(minX+200,minY+220,minX+260,minY+240,1);
    bar(0,0,100,20);
    outtextxy(9,7,"STOP");
    setviewport(minX+200,minY+260,minX+260,minY+280,1);
    bar(0,0,100,20);
    outtextxy(9,7,"STOP");
    setviewport(minX+200,minY+300,minX+260,minY+320,1);
    bar(0,0,100,20);
    outtextxy(9,7,"STOP");
    setviewport(minX+200,minY+340,minX+260,minY+360,1);
    bar(0,0,100,20);
    outtextxy(9,7,"STOP");
}
if(select==1){
    setviewport(minX+200,minY+140,minX+260,minY+160,1);
    bar(0,0,100,20);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

outtextxy(2,7,"WORKING");
setviewport(minX+200,minY+180,minX+260,minY+200,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
setviewport(minX+200,minY+220,minX+260,minY+240,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
setviewport(minX+200,minY+260,minX+260,minY+280,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
setviewport(minX+200,minY+300,minX+260,minY+320,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
setviewport(minX+200,minY+340,minX+260,minY+360,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
    }
if(select==2){
    setviewport(minX+200,minY+140,minX+260,minY+160,1);
    bar(0,0,100,20);
    outtextxy(9,7,"STOP");
    setviewport(minX+200,minY+180,minX+260,minY+200,1);
    bar(0,0,100,20);
    outtextxy(2,7,"WORKING");
    setviewport(minX+200,minY+220,minX+260,minY+240,1);
    bar(0,0,100,20);
    outtextxy(9,7,"STOP");
    setviewport(minX+200,minY+260,minX+260,minY+280,1);
    bar(0,0,100,20);
    outtextxy(9,7,"STOP");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setviewport(minX+200,minY+300,minX+260,minY+320,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
setviewport(minX+200,minY+340,minX+260,minY+360,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
}

if(select==3){
setviewport(minX+200,minY+140,minX+260,minY+160,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
setviewport(minX+200,minY+180,minX+260,minY+200,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
setviewport(minX+200,minY+220,minX+260,minY+240,1);
bar(0,0,100,20);
outtextxy(2,7,"WORKING");
setviewport(minX+200,minY+260,minX+260,minY+280,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
setviewport(minX+200,minY+300,minX+260,minY+320,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
setviewport(minX+200,minY+340,minX+260,minY+360,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
}

if(select==4){
setviewport(minX+200,minY+140,minX+260,minY+160,1);
bar(0,0,100,20);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

outtextxy(9,7,"STOP");
setviewport(minX+200,minY+180,minX+260,minY+200,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
setviewport(minX+200,minY+220,minX+260,minY+240,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
setviewport(minX+200,minY+260,minX+260,minY+280,1);
bar(0,0,100,20);
outtextxy(2,7,"WORKING");
setviewport(minX+200,minY+300,minX+260,minY+320,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
setviewport(minX+200,minY+340,minX+260,minY+360,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
}
if(select==5){
setviewport(minX+200,minY+140,minX+260,minY+160,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
setviewport(minX+200,minY+180,minX+260,minY+200,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
setviewport(minX+200,minY+220,minX+260,minY+240,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
setviewport(minX+200,minY+260,minX+260,minY+280,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setviewport(minX+200,minY+300,minX+260,minY+320,1);
bar(0,0,100,20);
outtextxy(2,7,"WORKING");
setviewport(minX+200,minY+340,minX+260,minY+360,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
}
if(select==6){
setviewport(minX+200,minY+140,minX+260,minY+160,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
setviewport(minX+200,minY+180,minX+260,minY+200,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
setviewport(minX+200,minY+220,minX+260,minY+240,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
setviewport(minX+200,minY+260,minX+260,minY+280,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
setviewport(minX+200,minY+300,minX+260,minY+320,1);
bar(0,0,100,20);
outtextxy(9,7,"STOP");
setviewport(minX+200,minY+340,minX+260,minY+360,1);
bar(0,0,100,20);
outtextxy(2,7,"WORKING");
}
}

```

**pos(double c1,double c2,double c3,double c4,double c5)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
double e1=0,e2=0,e3=0;
char f1[10],f2[10],f3[10];
char a1[10],a2[10],a3[10],a4[10],a5[10];

e1=cos(c1*PI/180)*(225.5*cos(c2*PI/180)+166.5*cos((c2+c3)*PI/180)-153*sin((c2+c3+c4)
*PI/180));
e2=sin(c1*PI/180)*(225.5*cos(c2*PI/180)+166.5*cos((c2+c3)*PI/180)-153*sin((c2+c3+c4)*
PI/180));
e3=163.5-225.5*sin(c2*PI/180)-166.5*sin((c2+c3)*PI/180)-153*cos((c2+c3+c4)*PI/180);

gcvt(e1,5,f1);
gcvt(e2,5,f2);
gcvt(e3,5,f3);

gcvt(c1,5,a1);
gcvt(c2,5,a2);
gcvt(c3,5,a3);
gcvt(c4,5,a4);
gcvt(c5,5,a5);

setviewport(500,140,550,160,1);
setfillstyle(SOLID_FILL,LIGHTGRAY);
bar(3,5,98,18);
setcolor(BLACK);
outtextxy(5,7,"X");
setviewport(480,170,580,190,1);
bar(3,5,98,18);
setcolor(BLACK);
outtextxy(5,7,f1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setviewport(500,220,550,240,1);
bar(3,5,98,18);
setcolor(BLACK);
outtextxy(5,7,"Y");
setviewport(480,250,580,270,1);
bar(3,5,98,18);
setcolor(BLACK);
outtextxy(5,7,f2);
setviewport(500,300,550,320,1);
bar(3,5,98,18);
setcolor(BLACK);
outtextxy(5,7,"Z");
setviewport(480,330,580,350,1);
setfillstyle(SOLID_FILL,LIGHTGRAY);
bar(3,5,98,18);
setcolor(BLACK);
outtextxy(5,7,f3);

setviewport(300,140,400,160,1);
setfillstyle(SOLID_FILL,LIGHTGRAY);
bar(3,5,98,18);
setcolor(BLACK);
outtextxy(5,7,a1);

setviewport(300,180,400,200,1);
setfillstyle(SOLID_FILL,LIGHTGRAY);
bar(3,5,98,18);
setcolor(BLACK);
outtextxy(5,7,a2);

setviewport(300,220,400,240,1);
setfillstyle(SOLID_FILL,LIGHTGRAY);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

bar(3,5,98,18);

setcolor(BLACK);

outtextxy(5,7,a3);

setviewport(300,260,400,280,1);

setfillstyle(SOLID_FILL,LIGHTGRAY);

bar(3,5,98,18);

setcolor(BLACK);

outtextxy(5,7,a4);

setviewport(300,300,400,320,1);

setfillstyle(SOLID_FILL,LIGHTGRAY);

bar(3,5,98,18);

setcolor(BLACK);

outtextxy(5,7,a5);
}

```

### ไฟล์ Mouse.h

```

/* Mouse toolkit header file */

/* Macros to retrieve low and high byte of an integer */

#define lo(f) ((f) & 0xff)

#define hi(f) (lo(f) >> 8)

/* Common scan-ascii codes */

#define CTRLC      0x2e03

#define CTRLH      0x2308

#define CTRLI      0x1709

#define CTRLL      0x260c

#define CTRLK      0x250b

#define CTRLJ      0x240a

```

```

#define CTRLU      0x1615
#define CTRLR      0x1312
#define CRKEY      0x1c0d
#define CTRLCRKEY  0x1c0a
#define UPKEY      0x4800
#define DOWNKEY    0x5000
#define LEFTKEY    0x4b00
#define RIGHTKEY   0x4d00
#define SHFTLEFT   0x4b34
#define SHFTRIGHT  0x4d36
#define DELKEY     0x5300
#define INSKEY     0x5200
#define BSKEY      0x0e08
#define SPACEBAR   0x3920
#define PGUPKEY    0x4900
#define PGDNKEY    0x5100
#define SHFTUPKEY  0x4838
#define SHFTDNKEY  0x5032
#define SHFTPGUPKEY 0x4939
#define SHFTPGDNKEY 0x5133
#define HOMEKEY    0x4700
#define ENDKEY     0x4f00
#define ESCKEY     0x011b
#define ALT_D      0x2000
#define ALT_E      0x1200
#define ALT_I      0x1700
#define ALT_R      0x1300
#define ALT_S      0x1f00
#define ALT_T      0x1400
#define ALT_X      0x2d00

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define F10KEY      0x4400

/* Mouse pseudo "key" codes */

#define LEFT_MOUSE_PRESS  0xff01
#define RIGHT_MOUSE_PRESS 0xff02

#define LEFT_MOUSE_REL    0xff11
#define RIGHT_MOUSE_REL   0xff12

/* Mouse driver function codes */

#define M_RESET          0
#define M_SHOW_CURS     1
#define M_HIDE_CURS     2
#define M_GET_STATUS    3
#define M_SET_CURS      4
#define M_GET_PRESS     5
#define M_GET_REL       6
#define M_SET_X_BOUNDS  7
#define M_SET_Y_BOUNDS  8
#define M_SET_G_CURS    9
#define M_SET_T_CURS   10

/* define other constants */

#define MOUSE_NEEDED     1
#define MOUSE_OPTIONAL  0
#define MOUSE_TEXT_MODE 0

/* Mouse external variable */

extern int mouse_text_x;

extern int mouse_text_y;

extern int mouse_grph_x;

extern int mouse_grph_y;

extern int mouse_initialized;

/* Mouse function prototypes */

extern void mouse(int *m1, int *m2, int *m3, int *m4);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

extern int check_mouse_driver(int need_mouse);
extern int init_mouse(int need_mouse, int gd, int gm);
extern int mouse_reset(void);
extern void move_mouse(int x, int y);
extern void mouse_on(int code);
extern void mouse_off(int code);
extern void mouse_grph_posn(int *x, int *y);
extern void mouse_txt_posn(int *x, int *y);
extern int mouse_in_box(int graphflag, int left, int right, int top, int bottom);
extern int button_release(int b);
extern int button_press(int b);
extern int button_state(void);
extern int mouse_trigger(int button_dir);

```

### ไฟล์ Mouse.c

```

#include <bios.h>
#include <dos.h>
#include <conio.h>
#include <process.h>
#include <stdio.h>
#include "mouse.h"

int mouse_text_x;
int mouse_text_y;
int mouse_grph_x;
int mouse_grph_y;
int mouse_initialized=0;
static int prev_cursor_state=0;

static char far *bios_video_area=(char far *)0x00400049L;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

static int low_resolution=0;
static void set_mouse_posn(int *x, int *y);
static int low_res_mode(int gd, int gm);
void mouse(int *m1, int *m2, int *m3, int *m4)
{
    union REGS inregs, outregs;
    inregs.x.ax = *m1;
    inregs.x.bx = *m2;
    inregs.x.cx = *m3;
    inregs.x.dx = *m4;
    int86(0x33,&inregs,&outregs);
    *m1 = outregs.x.ax;
    *m2 = outregs.x.bx;
    *m3 = outregs.x.cx;
    *m4 = outregs.x.dx;
}
int check_mouse_driver(int need_mouse)
{
    void far *address;
    address = getvect(0x33);
    if ((address == NULL) || (*(unsigned char *)address == 0xcf)){
        if (need_mouse) {
            printf("Mouse driver NOT installed\n");
            exit(1);
        }
        else return 0;
    }
    return 1;
}
int init_mouse(int need_mouse, int gd, int gm)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    int m1;
    mouse_initialized = 0;
    if (check_mouse_driver(need_mouse)){
        if (gd == 7) *bios_video_area = 6;
        if (low_res_mode(gd,gm))
            low_resolution = 1;
        m1 = mouse_reset();
        if (m1){
            mouse_initialized = 1;
            move_mouse(0,0);
            mouse_on(0);
        }
    }
    else{
        if (need_mouse) {
            printf("ERROR activating mouse ... \n");
            exit(1);
        }
    }
}

return mouse_initialized;
}

static int low_res_mode(int gd, int gm)
{
    if ((gd==1 || gd==2 || gd==8) && (gm >=0 && gm.<=3))
        return 1;
    return 0;
}

int mouse_reset(void)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int x1,m1,m2,m3,m4;
mouse_off(1);
m1 = M_RESET;
mouse(&m1,&m2,&m3,&m4);
set_mouse_posn(&m3,&m4);
return m1;
}

void move_mouse(int x, int y)
{
int m1,m2,m3,m4;
if (!mouse_initialized) return;
m1 = M_SET_CURS;
m3 = x*8;
m4 = y*8;
mouse(&m1,&m2,&m3,&m4);
set_mouse_posn(&m3,&m4);
}

void mouse_on(int restoreflag)
{
int m1,m2,m3,m4;
if (mouse_initialized){
if (!restoreflag || prev_cursor_state){
m1 = M_SHOW_CURS;
mouse(&m1,&m2,&m3,&m4);
prev_cursor_state = 1;
}
}
}

void mouse_off(int tempflag)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int m1,m2,m3,m4;
if (mouse_initialized){
    if (prev_cursor_state){
        m1 = M_HIDE_CURS;
        mouse(&m1,&m2,&m3,&m4);
        if (!tempflag) prev_cursor_state = 0;
    }
}
}
void mouse_grph_posn(int *x, int *y)
{
    int m1,m2;
    if (mouse_initialized){
        m1 = M_GET_STATUS;
        mouse(&m1,&m2,x,y);
        set_mouse_posn(x,y);
    }
    else {
        *x = 0; *y = 0;
    }
    return;
}
void mouse_txt_posn(int *x, int *y)
{
    mouse_grph_posn(x,y);
    *x = mouse_text_x;
    *y = mouse_text_y;
    return;
}
int mouse_in_box(int graphflag, int left, int top, int right, int bottom)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    int x,y;
    if (mouse_initialized){
        if (graphflag){
            x = mouse_grph_x;
            y = mouse_grph_y;
        }
        else{
            x = mouse_text_x;
            y = mouse_text_y;
        }
        if ((y >= top) && (y <= bottom) && (x >= left) && (x <= right)) return 1;
    }
    return 0;
}

int button_release(int b)
{
    int m1,m2,m3,m4;
    if (mouse_initialized) {
        m1 = M_GET_REL;
        m2 = b;
        mouse(&m1,&m2,&m3,&m4);
        set_mouse_posn(&m3,&m4);
        if (m2) return 1;
    }
    return 0;
}

int button_press(int b)
{
    int m1,m2,m3,m4;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (mouse_initialized){
    m1 = M_GET_PRESS;
    m2 = b;
    mouse(&m1,&m2,&m3,&m4);
    set_mouse_posn(&m3,&m4);
    if (m2) return 1;
}
return 0;
}

int button_state()
{
    int m1,m2,m3,m4;
    if (mouse_initialized){
        m1 = M_GET_STATUS;
        mouse(&m1,&m2,&m3,&m4);
        set_mouse_posn(&m3,&m4);
        return m2;
    }
    return 0;
}

static void set_mouse_posn(int *x, int *y)
{
    if (low_resolution) *x >>= 1;
    mouse_grph_x = *x;
    mouse_grph_y = *y;
    mouse_text_x = *x/8 + 1;
    mouse_text_y = *y/8 + 1;
}

int mouse_trigger(int button_dir)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int k;
if (bioskey(1)) {
    k = bioskey(0);
}
else {
    k = 0;
    if (button_dir) {
        if (button_press(0)) k = LEFT_MOUSE_PRESS;
        else if (button_press(1)) k = RIGHT_MOUSE_PRESS;
    }
    else {
        if (button_release(0)) k = LEFT_MOUSE_REL;
        else if (button_release(1)) k = RIGHT_MOUSE_REL;
    }
}
return k;
}

```

ไฟล์ Graph1.c

```

#include <graphics.h>
#include <conio.h>

int graphdriver;
int graphmode;
int minX,minY,maxX,maxY;

```

```
background()
```

```
{
```

```
struct viewporttype vp;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

graphdriver = DETECT;
graphmode = VGA;
initgraph(&graphdriver,&graphmode,"");
minX = 0;
minY = 0;
maxX = getmaxx();
maxY = getmaxy();

setfillstyle(SOLID_FILL,LIGHTBLUE);
floodfill(1,1,4);

setfillstyle(SOLID_FILL,LIGHTGRAY);
bar(minX+51,minY+50,maxX-51,maxY-50);
setcolor(WHITE);
line(minX+51,minY+50,minX+51,maxY-50);
line(minX+52,minY+51,minX+52,maxY-51);
line(minX+53,minY+52,minX+53,maxY-52);
line(minX+51,minY+50,maxX-51,minY+50);
line(minX+52,minY+51,maxX-52,minY+51);
line(minX+53,minY+52,maxX-53,minY+52);
setcolor(DARKGRAY);
line(maxX-51,minY+50,maxX-51,maxY-50);
line(maxX-52,minY+51,maxX-52,maxY-51);
line(maxX-53,minY+52,maxX-53,maxY-52);
line(minX+51,maxY-50,maxX-51,maxY-50);
line(minX+52,maxY-51,maxX-52,maxY-51);
line(minX+53,maxY-52,maxX-53,maxY-52);
}

```

`initialize()`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
setcolor(BLACK);
setttextstyle(DEFAULT_FONT,HORIZ_DIR,2);
outtextxy(160,90,"ROBOT CONTROL PROGRAM");
setttextstyle(DEFAULT_FONT,HORIZ_DIR,1);
outtextxy(320,130,"BY");
outtextxy(235,160,"Mr.Prasong Wongprasithiporn");
outtextxy(290,190,"35104250");
setviewport(minX+250,minY+220,minX+400,minY+250,1);
button();
setcolor(BLACK);
setttextstyle(DEFAULT_FONT,HORIZ_DIR,1);
outtextxy(50,12,"MANUAL");
setviewport(minX+250,minY+280,minX+400,minY+310,1);
button();
setcolor(BLACK);
setttextstyle(DEFAULT_FONT,HORIZ_DIR,1);
outtextxy(40,12,"CHECK HOME");
setviewport(minX+250,minY+340,minX+400,minY+370,1);
button();
setcolor(BLACK);
outtextxy(55,12,"EXIT");
}

```

```
button()
```

```

{
struct viewporttype vp;
getviewsettings(&vp);
setcolor(BLACK);
line(1,1,(vp.right-vp.left)-1,1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

line((vp.right-vp.left)-1,1,(vp.right-vp.left)-1,(vp.bottom-vp.top)-1);
line((vp.right-vp.left)-1,(vp.bottom-vp.top)-1,1,(vp.bottom-vp.top)-1);
line(1,(vp.bottom-vp.top)-1,1,1);
setcolor(WHITE);
line(2,2,(vp.right-vp.left)-2,2);
line(2,(vp.bottom-vp.top)-2,2,2);
setcolor(DARKGRAY);
line((vp.right-vp.left)-2,2,(vp.right-vp.left)-2,(vp.bottom-vp.top)-2);
line((vp.right-vp.left)-2,(vp.bottom-vp.top)-2,2,(vp.bottom-vp.top)-2);
}

manual()
{
setcolor(BLACK);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
outtextxy(235,70,"SELECT AXIS OF ARM ROBOT");
outtextxy(100,108,"TOGGLE MODE");
setviewport(minX+200,minY+100,minX+260,minY+120,1);
button();
setviewport(minX+100,minY+140,minX+170,minY+160,1);
button();
setcolor(BLACK);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
outtextxy(12,7,"AXIS 1");
setviewport(minX+100,minY+180,minX+170,minY+200,1);
button();
setcolor (BLACK);
outtextxy(12,7,"AXIS 2");
setviewport(minX+100,minY+220,minX+170,minY+240,1);
button();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setcolor(BLACK);
outtextxy(12,7,"AXIS 3");
setviewport(minX+100,minY+260,minX+170,minY+280,1);
button();
setcolor(BLACK);
outtextxy(12,7,"AXIS 4");
setviewport(minX+100,minY+300,minX+170,minY+320,1);
button();
setcolor(BLACK);
outtextxy(12,7,"AXIS 5");
setviewport(minX+100,minY+340,minX+170,minY+360,1);
button();
setcolor(BLACK);
outtextxy(7,7,"GRIPPER");
setviewport(minX+290,minY+340,minX+335,minY+360,1);
outtextxy(3,7,"FORCE");
setviewport(minX+340,minY+340,minX+380,minY+360,1);
button();
button_p();
setviewport(minX+380,minY+340,minX+390,minY+350,1);
button();
setcolor(BLACK);
line(2,8,5,2);
line(5,2,8,8);
line(8,8,2,8);
setviewport(minX+380,minY+350,minX+390,minY+360,1);
button();
setcolor(BLACK);
line(2,2,8,2);
line(8,2,5,8);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

line(5,8,2,2);
setviewport(minX+400,minY+340,minX+450,minY+360,1);
button();
setcolor(BLACK);
outtextxy(10,7,"CLIP");
setviewport(minX+400,minY+370,minX+450,minY+390,1);
button();
setcolor(BLACK);
outtextxy(6,7,"CLEAR");
setviewport(minX+220,minY+380,minX+270,minY+400,1);
button();
setcolor(BLACK);
outtextxy(10,7,"EXIT");
setviewport(minX+400,minY+140,minX+440,minY+220,1);
button();
arrow(10,70,10,30,7,30,20,10,33,30,30,30,70);
setviewport(minX+400,minY+240,minX+440,minY+320,1);
button();
arrow(10,10,10,50,7,50,20,70,33,50,30,50,30,10);
}

arrow(int x1,int y1,int x2,int y2,int x3,int y3,int x4,int y4,int x5,int y5,
      int x6,int y6,int x7,int y7)
{
setcolor(BLACK);
line(x1,y1,x2,y2);
line(x2,y2,x3,y3);
line(x3,y3,x4,y4);
line(x4,y4,x5,y5);
line(x5,y5,x6,y6);

```

```

line(x6,y6,x7,y7);
line(x7,y7,x1,y1);
}

```

### ไฟล์ Comm.asm

```

ptr1    equ    40h
commnd1 equ    50h

org 0000h
    sjmp    start

org 0023h
    jbc    ri,recv
    reti

recv:  mov    a,sbuf
       mov    commnd1,a
       reti

start: call    delay
       mov    scon,#50h
       mov    tmod,#20h
       mov    th1,#0fdh
       setb   tr1
       setb   ea
       mov    ptr1,#0
       mov    dptr,#3
       mov    a,#80h
       movx   @dptr,a

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov    dptr,#0
mov    a,#0ffh
movx   @dptr,a
mov    dptr,#1
movx   @dptr,a
mov    dptr,#2
movx   @dptr,a
mov    dptr,#0ffh

again: setb  es
       lcall chksta
       sjmp  again

msm1s: mov  a,ptr1
       cjne r0,#0,ss1
       mov  dptr,#1500h
       movc a,@a+dptr
       mov  dptr,#2
       sjmp send

ss1:   cjne r0,#1,ss2
       mov  dptr,#1500h
       movc a,@a+dptr
       mov  dptr,#0
       sjmp send

ss2:   cjne r0,#2,ss3
       mov  dptr,#1200h
       movc a,@a+dptr
       mov  dptr,#1
       sjmp send1

ss3:   cjne r0,#3,ss4

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov    dptr,#1300h
movc   a,@a+dptr
mov    dptr,#1
sjmp  send2
ss4:   cjne r0,#4,send
mov    dptr,#1400h
movc   a,@a+dptr
mov    dptr,#2
sjmp  send3
send:  movx @dptr,a
      lcall delay
      ret
send1: movx @dptr,a
      lcall delay1
      ret
send2: movx @dptr,a
      lcall delay2
      ret
send3: movx @dptr,a
      lcall delay3
      ret

msmf1s: lcall msm1s
        mov  a,ptr1
        inc  a
        cjne a,#8,nover2
        mov  a,#0
nover2: mov  ptr1,a
        ret

```

```

msmr1s: lcall  msmr1s
        mov   a,ptr1
        dec  a
        cjne a,#0ffh,nbelw1
        mov  a,#7
nbelw1: mov   ptr1,a
        ret

```

```

chksta: mov   a,commnd1
        cjne  a,#1,not1
        lcall msmf1s
        mov  commnd1,#0
        ajmp outchk
not1:   cjne  a,#2,not2
        lcall msmf1s
        sjmp outchk
not2:   cjne  a,#3,not3
        lcall msmr1s
        mov  commnd1,#0
        sjmp outchk
not3:   cjne  a,#4,not4
        lcall msmr1s
        sjmp outchk
not4:   cjne  a,#5,not5
        mov  r0,#0
        sjmp outchk
not5:   cjne  a,#6,not6
        mov  r0,#1
        sjmp outchk
not6:   cjne  a,#7,not7

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov    r0,#2
sjmp   outchk
not7:  cjne  a,#8,not8
mov    r0,#3
sjmp   outchk
not8:  cjne  a,#9,not9
mov    r0,#4
sjmp   outchk
not9:  cjne  a,#10,not10
sjmp   outchk
not10: cjne  a,#11,not11
mov    commnd1,#0
clr    p3.2
setb   p1.7
clr    p1.6
st:    jb    p3.2,stop
sjmp   st
stop:  clr    p1.7
setb   p1.6
sjmp   outchk
not11: cjne  a,#12,not12
clr    p1.7
clr    p1.6
sjmp   outchk
not12: cjne  a,#13,not13
sjmp   outchk
not13: cjne  a,#14,not20
mov    commnd1,#0
clr    p1.7
setb   p1.6

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    sjmp  outchk
not20: cjne  a,#255,outchk
    mov   commnd1,#0
outchk: ret

```

```

delay: mov   r7,#20
dl1:  mov   r6,#0
    djnz  r6,$
    djnz  r7,dl1
    ret

```

```

delay1: mov  r7,#20
dl2:  mov  r6,#0
    djnz  r6,$
    djnz  r7,dl2
    ret

```

```

delay2: mov  r7,#20
dl3:  mov  r6,#0
    djnz  r6,$
    djnz  r7,dl3
    ret

```

```

delay3: mov  r7,#20
dl4:  mov  r6,#0
    djnz  r6,$
    djnz  r7,dl4
    ret

```

```
org 1000h
```

```

step1: db  1111110b
    db  11111100b
    db  11111101b

```

```

db 11111001b
db 11111011b
db 11110011b
db 11110111b
db 11110110b

```

org 1100h

```

step2: db 11101111b
db 11001111b
db 11011111b
db 10011111b
db 10111111b
db 00111111b
db 01111111b
db 01101111b

```

org 1200h

```

step3: db 11111100b
db 11111001b
db 11110011b
db 11110110b
db 11111100b
db 11111001b
db 11110011b
db 11110110b

```

org 1300h

```

step4: db 11001111b
db 10011111b
db 00111111b

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

db 01101111b
db 11001111b
db 10011111b
db 00111111b
db 01101111b

org 1400h
step5: db 1111110b
db 1111101b
db 1111011b
db 1111011b
db 1111110b
db 1111101b
db 1111011b
db 1111011b

org 1500h
step6: db 1110111b
db 1101111b
db 1011111b
db 0111111b
db 1110111b
db 1101111b
db 1011111b
db 0111111b

end

```



## หนังสืออ้างอิง

1. ประเมษฐ์ ประณยานันท์,ปิยพงศ์ เผ่าวณิช," คู่มือและการประยุกต์ใช้งาน ไมโครคอนโทรลเลอร์ MCS-51",บริษัทซีเอ็ดยูเคชั่นจำกัด(มหาชน),380 หน้า,2536
2. สุเจตน์ จันทพงษ์,"ไมโครคอนโทรลเลอร์ชิพเดี่ยว 8051",วิทยาลัยมหานคร,178 หน้า,2535
3. Schilling,Robert J.,"Fundamentals of Robotics Analysis and Control",Prentice-Hall,1990
4. K.S. Fu,R.C. Gonzalez,C.S. Lee,"Robotics:Control Vision and Intelligence",McGRAW-HILL ,1987
5. Moshe Shoham,"Robotics Structure",Eshed Robotec(1982),1984



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้