



อาจารย์ที่ปรึกษา

อ.วราคม เนินน้อย

ศจ.ดร.โทชิโอะ อิจิจิม่า

ปีวิเทศาณินพณธ์

ภาควิชา

คณะวิศวกรรมศาสตร

เรื่อง

ผู้จัดทำ

ปีการศึกษา 2533

วิศวกรรมเครื่องกล

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

แขนงกลควบคุมด้วย STEPPING MOTOR

นาย คมสรร แห่งปัจฉิการ 30-1031

นาย วุฒิชัย แก้วไพโรจน์ 30-1271

(อาจารย์ที่ปรึกษา)

(อ.วราคม เนินน้อย)

แผนกควบคุมด้วย STEPPING MOTOR

คณบดี แห่งบัณฑิตวิทยาลัย

วุฒิชัย แก้วไพโรจน์

อจ. วราคม เนินน้อย อาจารย์ที่ปรึกษา

ศจ.ดร.โทชิโอะ ฮิโงชิม่า อาจารย์ที่ปรึกษา

บทคัดย่อ

โครงการวิจัยนี้เป็นการศึกษา และออกแบบแขนกล (ROBOT ARM) ประเภทคล้ายกับแขนมนุษย์ โครงสร้างของแขนกลได้ทำการสร้างและประกอบ ชิ้นส่วนต่างๆที่ได้ออกแบบไว้ โดยมี การเปลี่ยนแปลงบางจุดเพื่อความเหมาะสม พร้อมทั้งสร้างโปรแกรมควบคุมแบบ FORWARD KINEMATIC เพื่อหาค่าตำแหน่งของแขนกลซึ่งเคลื่อนที่ไป และแสดงค่าของตัวแปรที่ป้อนเข้าไปด้วย ให้มีองศาแห่งความเป็นอิสระ (DEGREE OF FREEDOM) เท่ากับ 5 และใช้ STEPPING MOTOR ชนิด HYBRID 4 PHRASE เป็นตัวขับเคลื่อน และจะขับเคลื่อนที่ละ 2 PHRASE โดยใช้ UNIVERSAL CONTROLLER ซึ่งเป็น IC สำเร็จรูปสำหรับขับเคลื่อน MOTOR ที่ต้องการกำลังไฟเพิ่มเติมนอกเพียงเล็กน้อย ของบริษัท SANYO DENKI จำกัด

การควบคุมแขนกลกระทำได้โดยผ่าน MICRO-COMPUTER โดย CPU จะส่งงาน IC-8253 เป็นขั้นแรก และต่อจากนั้น IC-8253 จะส่งงาน PHM 8713 เพื่อขับเคลื่อน STEPPING MOTOR ให้หมุนไปยังตำแหน่งที่ต้องการในอัตราเร็วที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONTROLLING ROBOT-ARM BY STEPPING MOTOR

KOMSON HANGPATCHICAN

WUTTICHAJ KAEWPIROJ

WARAKOM NAEDNOI , ENG.ADVISOR

ASSOCIATE PROFESSOR TOSHIO IIJIMA , DR.ENG.ADVISOR

ABSTRACT

THIS PROJECT INVOLVES STUDYING AND DESIGNING THE ROBOT
-ARM, REVOLUTE TYPE (GENERAL PURPOSE), DEVELOPED FROM LAST TWO
YEARS. THIS ROBOT HAS 5 DEGREES OF FREEDOM WITH 4-PHASES
STEPPING MOTOR . EACH 2 PHASES HAS EXCITED BY THE UNIVERSAL
CONTROLLER PMM 8713 WHICH MANUFACTURED BY SANYO DENKI
CO.,LTD, AND PMM 8253 WILL BE COMMANDED WITH 8713

ROBOTIC STRUCTURE HAS BUILT AND COMPOSED WHICH ACCORDING
TO EXCITING ~~DRAWING~~ AND ~~CHANGED~~ ~~SOME~~ ~~PART~~ ~~OF~~ ~~ROBOTIC~~ . . .
IN ADDITION , THE FORWARD KINEMATIC CONTROLLING PROGRAM HAS
USED FOR THE ANALYSIS.

P

สารบัญ

บทนำ	1
บทที่ 1	พื้นฐาน ROBOT และทฤษฎี การ CONTROL.....	2
บทที่ 2	การสร้าง และประกอบ.....	20
บทที่ 3	การติดต่อกับ COMPUTER เพื่อควบคุม ROBOT ARM.....	24
บทที่ 4	การใช้โปรแกรมควบคุม ROBOT ARM.....	34
บทที่ 5	ผลการทดลอง , สรุปผลและวิจารณ์.....	41
กิตติกรรมประกาศ	46
บรรณานุกรม	47
ภาคผนวก ก.	โปรแกรมควบคุม ROBOT ARM	
ภาคผนวก ข.	รายละเอียด IC และ MOTOR	
ภาคผนวก ค.	DRAWING OF MACHINE PART	

บทนำ

โครงการแขนกล (ROBOT ARM) นี้เป็นโครงการที่ได้มีการศึกษาและออกแบบไว้บางส่วนแล้วในปริทัศน์พนธ์ ปีการศึกษา 2532 ภาควิชาวิศวกรรมระบบควบคุม สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง ซึ่งเนื้อหาของปริทัศน์พนธ์ เป็นส่วนของการออกแบบวงจรควบคุมชุดสเต็ปเปอร์มอเตอร์ (Stepper Motor) ชนิดไฮบริด (Hybrid) 4 เฟส และการออกแบบชิ้นส่วนประกอบต่าง ๆ ของแขนกล (ROBOT ARM) รวมทั้งขนาดและชนิดของตัวขับเคลื่อน มอเตอร์ เฟือง และสายพาน

ดังนั้น จุดประสงค์ของโครงการนี้ จึงเป็นไปในเรื่องของการสร้างชิ้นส่วนต่าง ๆ ที่ได้กำหนดไว้ โดยอาศัยความเป็นไปได้ในการสร้าง การประกอบชิ้นส่วนต่าง ๆ ที่สร้างขึ้นตามแบบ และแก้ไขปัญหาที่เกิดขึ้นที่สามารถกระทำได้ โดยชิ้นส่วนบางชิ้นอาจมีการตัดแปลงต่อเติมเพื่อความเหมาะสมในการประกอบและการทำงานของแขนกล (ROBOT ARM) การออกแบบโปรแกรมการคำนวณ และติดต่อ แปลง และส่งข้อมูลจากคีย์บอร์ดผ่านเครื่องคอมพิวเตอร์ เพื่อส่งค่าไปยังตัวขับเคลื่อน เพื่อให้แขนกลสามารถเคลื่อนที่ไปยังตำแหน่ง หรือมุมที่กำหนดไว้ได้อย่างถูกต้อง พร้อมทั้งแสดงตำแหน่งที่เคลื่อนที่ไปได้ในระนาบ 3 มิติ (space) ศึกษาและทดลองว่าแขนกล (ROBOT ARM) ที่ประกอบขึ้นสามารถเคลื่อนที่โดยใช้โปรแกรมการติดต่อมีความละเอียด แม่นยำเพียงใด พร้อมทั้งวิเคราะห์ถึงจุดที่ควรมีการปรับปรุง ให้เหมาะสมมากยิ่งขึ้น เพื่อสะดวกในการเคลื่อนที่ โดยโปรแกรมการควบคุมนี้ศึกษาแต่ทางด้าน Kinematic Control ส่วนทางด้าน Dynamic Control ยังไม่ได้กล่าวถึง

คุณลักษณะของแขนมนุษย์

ในเมื่อเราต้องการที่จะสร้างแขนหุ่นยนต์ให้มีคุณลักษณะเหมือนแขนของมนุษย์ เราจึงมีความจำเป็นต้องศึกษาโครงสร้างและลักษณะการเคลื่อนไหวของแขนมนุษย์ก่อน

แขนของมนุษย์ประกอบด้วยส่วน 2 ส่วน ที่เห็นแตกต่างกันอย่างชัดเจน คือ ส่วนที่เป็นข้อมือ ซึ่งมีข้อต่อหลักอยู่ 3 ข้อต่อ และอีกส่วนคือ แขน ซึ่งมีข้อต่อหลักอยู่ 3 ข้อต่อ (หัวไหล่, ข้อศอก) ก่อนอื่นเรามาดูส่วนแรกก่อน ข้อมือของมนุษย์มีลักษณะดังต่อไปนี้

การหมุนข้อมือ (ROLL (angle)) สามารถหมุนได้ $-180 <> 0 <> +90 = 270$

การพับข้อมือ (PITCH (angle)) สามารถพับได้ $-90 <> 0 <> +50 = 140$

การบิดข้อมือ (YAW (angle)) สามารถบิดได้ $-45 <> 0 <> +15 = 60$

การเคลื่อนที่ของข้อมือทั้ง 3 แบบ มีการเคลื่อนที่ที่เป็นอิสระต่อกัน ดังนั้นเราจึงให้การเคลื่อนที่แต่ละแบบเป็น 1 องศาแห่งอิสระ

ส่วนที่ 2 คือ แขน ประกอบด้วยข้อต่อหลัก 2 ข้อต่อ แต่มี 3 องศาแห่งอิสระ คือ 2 องศาแห่งอิสระที่หัวไหล่ และอีก 1 องศาแห่งอิสระที่ข้อศอก แต่ที่หัวไหล่ของแขนกลจะมีเพียง

1 องศาแห่งอิสระ ลักษณะหนึ่งที่สำคัญของโครงสร้างของแขน คือ อัตราส่วนระหว่างความยาวของแขนก่อนบนกับแขนก่อนล่าง จะมีค่าราว ๆ 1.2 : 1 นั่นคือแขนก่อนล่างจะต้องสั้นกว่าแขนก่อนบน ถ้าเราออกแบบให้แขนก่อนล่างยาวกว่าแขนก่อนบน จะทำให้ประสิทธิภาพการทำงานของแขนลดลง

การแบ่งประเภทของแขนกล

เราสามารถแบ่งประเภทของแขนกลโดยพิจารณาจากพื้นฐานใหญ่ ๆ ดังนี้

1. โครงสร้างภายนอกและการเคลื่อนที่ของแขนกล ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น จึงจำเป็นต้องมีให้ต่องาน และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
2. ทางเดินของแขนกลซึ่งขึ้นอยู่กับารควบคุมการเคลื่อนที่
3. ประสิทธิภาพในการทำงานของแขนกล

การพิจารณาแบ่งประเภทของแขนกลในขั้นนี้ เราจะอาศัยเฉพาะพื้นฐานข้อ 1 และข้อ 2 เท่านั้น แม้ว่าพื้นฐานข้อ 3 จะมีความสำคัญมากที่สุดแต่จะต้องอาศัยความรู้ทางด้านคินเมติก (Kinematics) และจลนศาสตร์ (Dynamic) ขึ้นสูงซึ่งยากต่อการทำความเข้าใจ การแบ่งประเภทแขนกลในระดับนี้จึงจะไม่นำพื้นฐานข้อ 3 มาพิจารณา

การแบ่งประเภทของแขนกลโดยอาศัยการเคลื่อนที่และโครงสร้างภายนอก

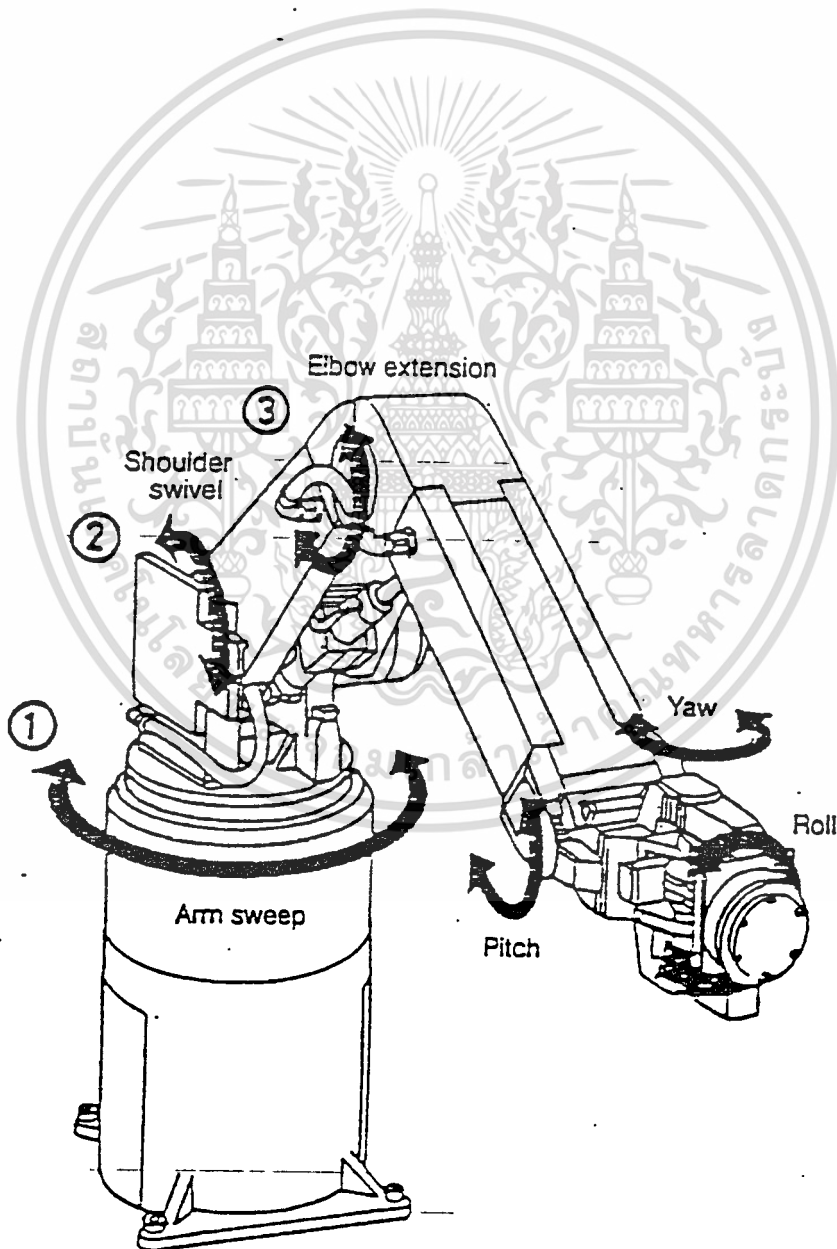
เพื่อให้บรรลุวัตถุประสงค์ในการแบ่งประเภทของแขนกลโดยอาศัยการเคลื่อนที่และโครงสร้างภายนอก เราจะต้องเข้าใจถึงการเคลื่อนที่ของแขนกลเสียก่อน แขนกลมีการเคลื่อนที่อยู่ 3 แบบ แบบแรกคือ การหมุนรอบแกนลองจิจูด (Longitudinal axis) ของตัวเชื่อม (link) ระหว่างข้อต่อ 2 ข้อต่อ (swivel motion) แบบที่ 2 คือ การหมุนข้อต่อรอบแกนอน (Bending motion) แบบที่ 3 คือการเคลื่อนที่แบบเส้นตรงในทิศทางของแกนลองจิจูด อาจยึดออกไปเฉพาะบางส่วนหรือโครงสร้างทั้งหมด (prismatic motion) หากเราแบ่งประเภทของแขนกลตามการเคลื่อนที่ของข้อต่อ จะสามารถแบ่งแขนกลเป็นแบบต่าง ๆ ได้ดังนี้

1. แขนกลที่มีลักษณะคล้ายแขนมนุษย์ (Revolute (jointed arm) robot)
2. แขนกลที่มีการเคลื่อนที่เป็นทรงกลม (Polar (spherical) robot)
3. แขนกลที่มีการเคลื่อนที่เป็นทรงกระบอก (Cylindrical robot)
4. แขนกลแบบคาร์ทีเซียนชนิดเลื่อน (Cartisian (rectangular) robot, sliding-type)
5. แขนกลแบบคาร์ทีเซียนชนิดมีโครง (Cartisian (rectangular) robot, gantry-type)
6. แขนกลแบบผสม (SCARA-type robot)

P

1. แขนกลที่มีลักษณะคล้ายแขนมนุษย์ (Revolute (jointed arm) robot)

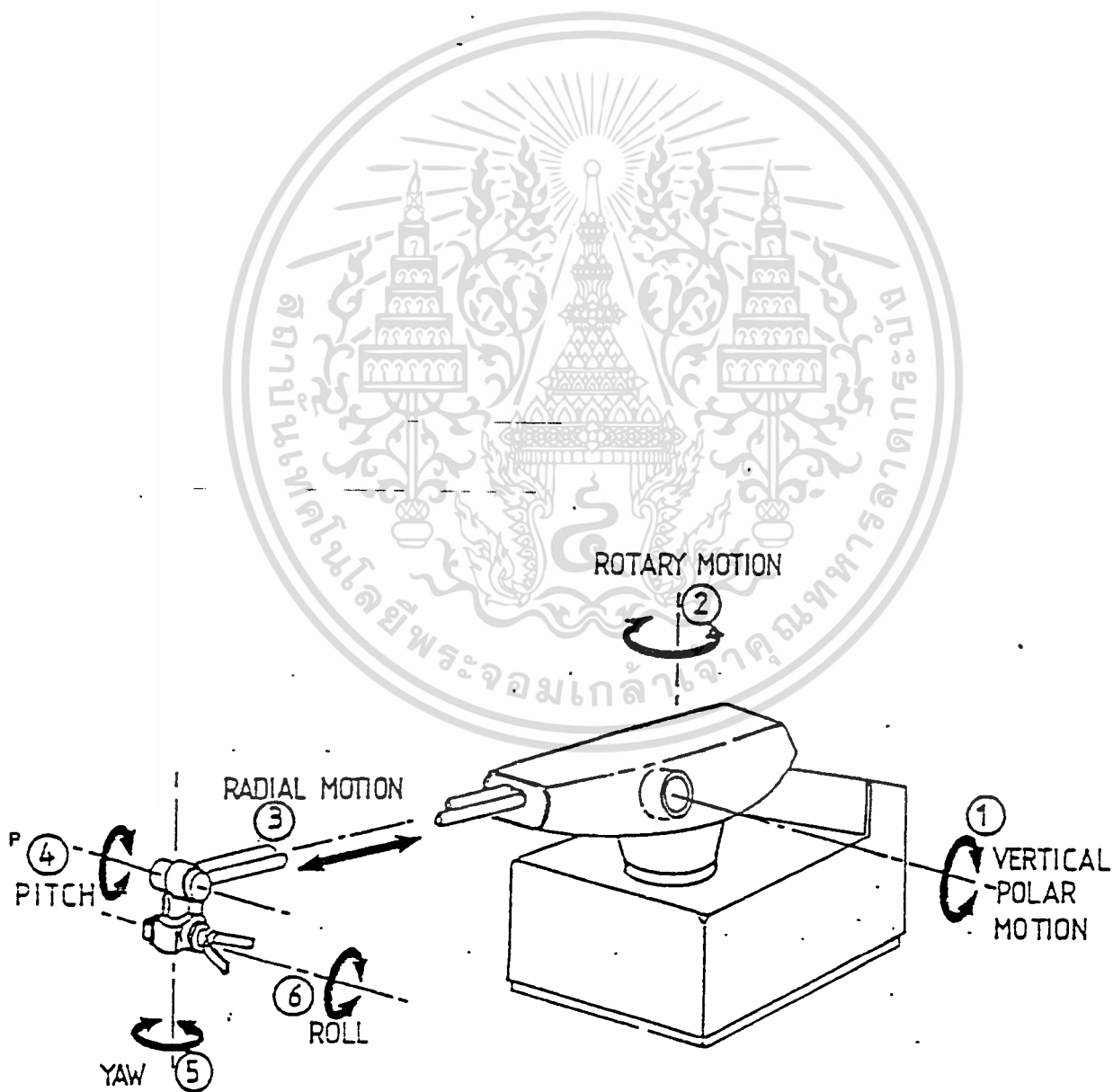
แขนกลชนิดนี้จะมีลักษณะคล้ายกับแขนของมนุษย์มากที่สุด ด้วยเหตุนี้จึงเป็นการง่ายที่จะคิดแปลงนำมาใช้แทนที่มนุษย์มากกว่าแขนกลแบบอื่น ๆ ตัวแขนกลประกอบไปด้วยส่วนเคลื่อนไหว 3 ส่วนใหญ่ ๆ คือ สะเอว (waist), หัวไหล่ซึ่งติดตั้งอยู่บนสะเอว และข้อศอก ซึ่งติดตั้งที่ปลายของหัวไหล่ (ดูรูปที่ 2.1)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
รูปที่ 2.1 แขนกลที่มีลักษณะคล้ายแขนมนุษย์

2. แขนกลที่มีลักษณะการเคลื่อนที่เป็นทรงกลม (Polar (spherical) robot)

แขนกลชนิดนี้จะมีส่วนเคลื่อนที่ 3 ส่วน ส่วนแรกจะหมุนรอบแกนของสะเอวซึ่งอยู่บนฐาน ส่วนที่ 2 คือการหมุนแขนในแนวขึ้น-ลง -อาศัยการหมุนของทั้งสองส่วนนี้ แขนกลจะสามารถหมุนเป็นส่วนหนึ่งของทรงกลม การเคลื่อนที่ของส่วนที่ 3 คือ การขยับเข้าออกของแขน (prismatic motion) รัศมีที่แขนกลสามารถเคลื่อนที่ไปได้จะขึ้นอยู่กับความยาวของแขนที่เลื่อนเข้าออก (prismatic joint) (ดูรูปที่ 2.2)

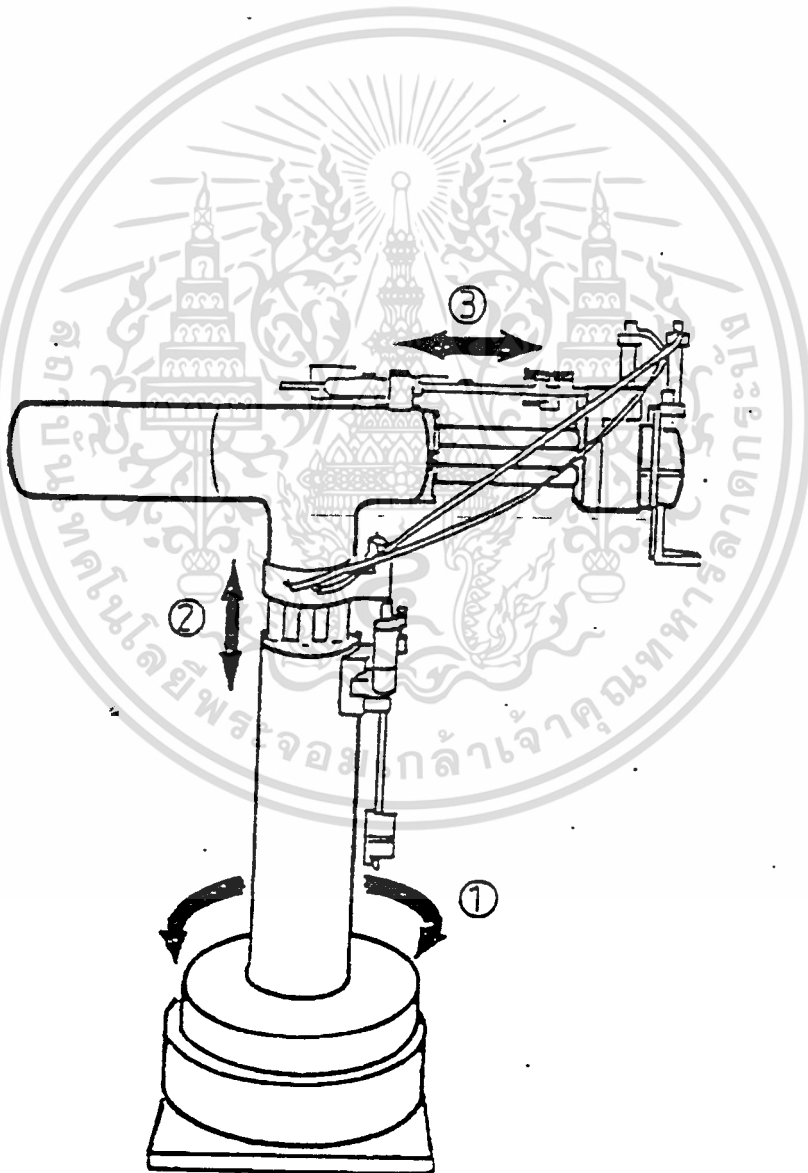


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.2 แขนกลเคลื่อนที่เป็นทรงกลม

3. แขนกลที่มีการเคลื่อนที่เป็นทรงกระบอก (Cylindrical robot)

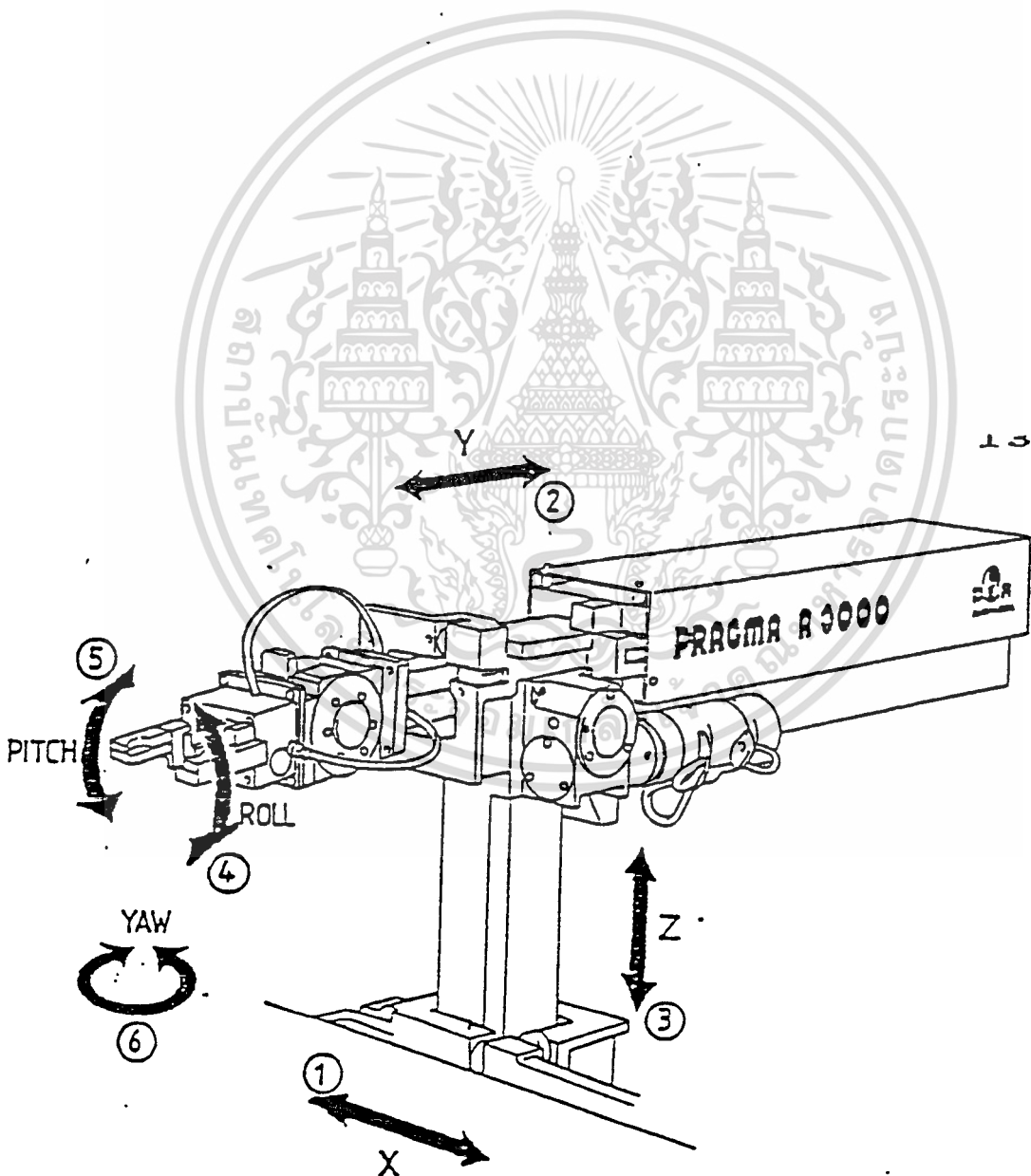
แขนกลชนิดนี้ประกอบด้วยฐาน, แขนในแนวนอน (horizontal arm) และแกนเลื่อนเข้า-ออก (prismatic joint) ซึ่งประกอบอยู่ภายในแขนในแนวนอน ฐานทั้งหมดสามารถขยับขึ้น-ลงได้ แขนในแนวนอนจะหมุนรอบแกนตั้งได้ ส่วนของแกนเลื่อนเข้า-ออกขนานกับฐาน (ดูรูปที่ 2.3)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกรูปที่ 2.3 ในแขนกลที่มีการเคลื่อนที่เป็นทรงกระบอกเอกสารทุกครั้งที่มีการนำไปใช้

4. แขนกลแบบคาร์ทีเซียนชนิดเลื่อน (Cartesian (rectangular) robot, sliding type)

แขนกลชนิดนี้ประกอบด้วยแกน 3 แกนตั้งฉากต่อกันสำหรับเคลื่อนที่ขึ้น-ลง, ซ้าย-ขวา และ หน้า-หลัง โดยอาศัยการเคลื่อนที่ของแกนเลื่อนเข้าออก แม้ว่าแขนกลชนิดนี้จะมีความแม่นยำในการทำงานสูง แต่ไม่นิยมนำมาใช้กันนัก เนื่องจากการคิดแปลงแขนกลชนิดนี้เพื่อใช้ทำงาน แทนมนุษย์ทำได้ยาก (ดูรูปที่ 2.4)

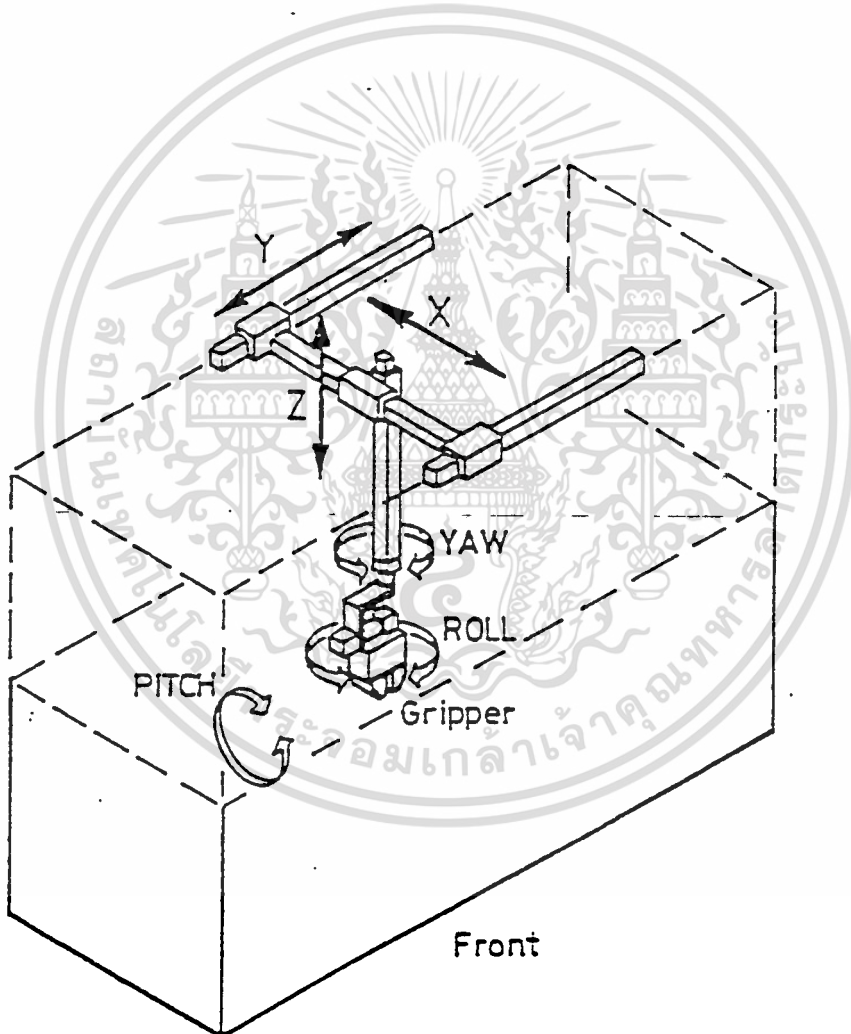


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดูและขอข้อมูลจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 2.4 แขนกลแบบคาร์ทีเซียนชนิดเลื่อน

5. แขนกลแบบคาร์ทีเซียนชนิดมีโครง (Cartesian (rectangular) robot, gantry type)

แขนกลชนิดนี้ มีโครงสร้างคล้ายแบบที่ 4 แต่ต่างกันที่แขนกลแบบนี้จะมีโครงสำหรับให้ตัวแขนกลเคลื่อนที่ไปเท่านั้น (ดูรูปที่ 2.5)

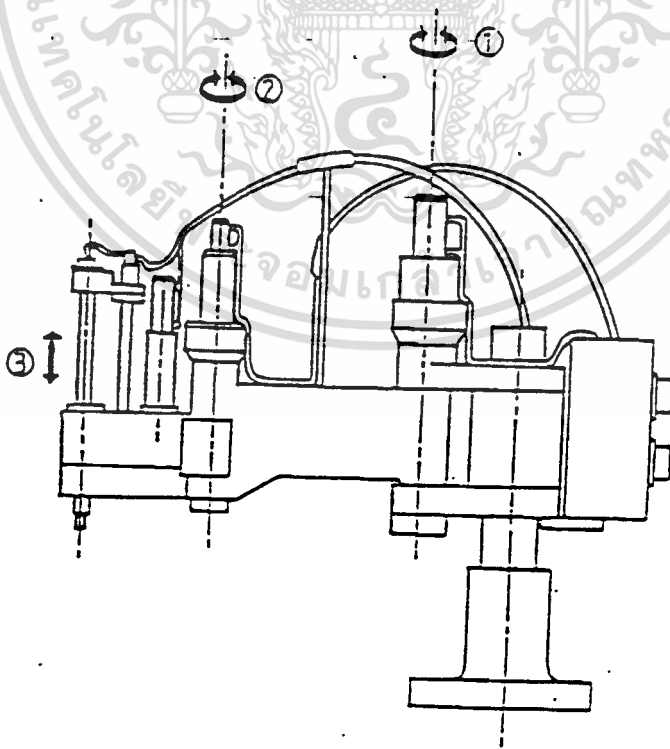


P



6. แขนกลแบบผสม (SCARA-type robot)

แขนกลชนิดนี้เกิดจากการรวมแขนกล 2 แบบเข้าด้วยกัน คือ แขนกลที่มีลักษณะคล้าย
แขนมนุษย์ กับแขนที่มีजरเคลื่อนที่เป็นทรงกระบอก โครงสร้างของแขนกลแบบผสมจะมีแกนหมุนใน
แนวตั้ง ทำให้สามารถเคลื่อนที่ในแนวระนาบได้ ซึ่งมีคุณสมบัติสอดคล้องกับแขนกลทั้งสองแบบ
นอกจากนี้ยังมีแกนหลักในแนวตั้ง ซึ่งทำให้แรงต่าง ๆ ที่กระทำต่อตัวแขนกล เช่น แรงดึงดูดของ
โลก แรงเหวี่ยงจากศูนย์กลาง ไม่มีผลต่อโครงสร้างมากเหมือนกับการที่แกนอยู่ในแนวระนาบ ซึ่ง
ข้อได้เปรียบนี้มีความสำคัญมาก โดยเฉพาะแขนกลที่ต้องการความเร็ว และความแม่นยำในการ
ทำงานสูง (ดูรูปที่ 2.6)



ลักษณะการเคลื่อนที่ของแขนกล

การควบคุมทางเดินของแขนกลแบ่งเป็น 2 แบบใหญ่ ๆ คือ

1. การควบคุมแบบจุด [Point-to-Point (PTP) Control]
2. การควบคุมแบบต่อเนื่อง [Continuous Path (CP) Control]

การควบคุมแขนกลแบบจุด แขนกลจะได้รับการตั้งโปรแกรมให้เคลื่อนที่มาหยุดที่จุดใดจุดหนึ่งที่กำหนดไว้ และใช้จุดนั้นเป็นจุดอ้างอิงในการกำหนดจุดการเคลื่อนที่ต่อไป แต่จะไม่สามารถควบคุมการเคลื่อนที่ระหว่างจุด ให้อยู่ในแนวเส้นที่ต้องการได้ ในทางตรงกันข้าม แขนกลที่ควบคุมแบบต่อเนื่อง จะสามารถเคลื่อนที่ตามเส้นทางอย่างถูกต้อง

การควบคุมแบบจุดจะมีความแม่นยำสูงมาก ทั้งในแง่ของความเที่ยงตรงหรือการทำงานที่ซ้ำ ๆ กัน แม้ว่าการเคลื่อนที่แบบต่อเนื่อง จะมีการเคลื่อนไหวที่นุ่มนวลมากกว่าในเส้นทางที่ได้ระบุไว้แล้ว แต่มีข้อเสียในด้านความเร็ว ซึ่งเป็นฟังก์ชันของจังหวะการเคลื่อนที่ที่คำนวณโดยคอมพิวเตอร์ ในลักษณะเวลาจริง (Real Time) ข้อเสียนี้ทำให้ความเร็วในการเคลื่อนที่แบบต่อเนื่องลดลง 15-25 %

ROBOT ARM KINEMATICS AND DYNAMICS

Kinematics Arm เป็นการวิเคราะห์การเคลื่อนที่และตำแหน่งของแขนกล โดยใช้จุดที่กำหนดเป็นจุดอ้างอิงการเคลื่อนที่ ทั้งนี้การวิเคราะห์หากการเคลื่อนที่แบบ Kinematics จะไม่สนใจถึง Forces และ Moment ที่เกิดขึ้นจากการเคลื่อนที่ การวิเคราะห์แบบ Kinematics นี้จะอธิบายการเคลื่อนที่และตำแหน่ง (The Spatical Displacement) ซึ่งเป็นฟังก์ชันของเวลาของ joint แต่ละ joint โดยบอกถึงตำแหน่งและทิศทางของ End-effector ของแขนกล โดยใช้จุดกำหนดเป็นจุดอ้างอิง

การวิเคราะห์แบบ Kinematics มี 2 ประเภท

1. แบบ Direct หรือ Forward Kinematics
2. แบบ Inverse Kinematics หรือ Arm Solution

จากการเคลื่อนไหวที่เป็นอิสระต่อกันในแต่ละ joint ของแขนกล จึงต้องกำหนด เป้าหมายที่จะเคลื่อนที่ไป โดยอ้างอิงถึงจุดที่กำหนด (The Reference Coordinate Frame) การหาค่าพารามิเตอร์ในแต่ละ joint -- อาจใช้วิธี Matrix แบบ 4x4 Homogeneous Transformation Matrix) ในการอธิบายตำแหน่งความสัมพันธ์ระหว่างจุดหรือ link ข้างเคียงได้

ROBOT ARM DYNAMICS เป็นการนำเอาสมการ Mathematical Formular มา อธิบายการเคลื่อนที่แบบ Dynamic ของแขนกล โดยอาศัยความรู้จาก Physical Law เช่น Law of Newtonian and Lagrangian Mechanic

ทฤษฎีเกี่ยวกับ ROBOT ARM KINEMATICS

A Mechanical Manipulator แบบ open-loop ของในแต่ละ link ซึ่งเชื่อมต่อกัน โดยใช้ revolute หรือ prismatic joint โดยปลาย link ด้านหนึ่งจะยึดติดกับ Supporting Base ในขณะที่ปลายอีกข้างถูกปล่อยอิสระและติดกับ End-effector จากความสัมพันธ์การเคลื่อนที่ของ joint ซึ่งมีผลทำให้เกิดการเคลื่อนที่ของ link โดยจะ อธิบายในรูปของตำแหน่งและทิศทางซึ่งอ้างอิงกับ Fixed Reference Coordinate System

The Theoretical and Practical ใน Kinematics

1. กรณีที่กำหนดตัวแปรการเคลื่อนที่ของแต่ละ joint เช่น มุมการเคลื่อนที่ (joint angle vector) $q(t) = (q_1(t), q_2(t), \dots, q_n(t))^T$ เมื่อ n คือ จำนวน degree of freedom และตัวแปร Geometric ในแต่ละ link มาเพื่อหาค่าตำแหน่ง และทิศทางสุดท้ายของ End-effector ของการเคลื่อนที่โดยอ้างอิงกับ Reference Coordinate System กรณีนี้เรียกว่า The Direct (or foward) Kinematics
2. กรณีที่กำหนดค่าตำแหน่งและทิศทางของ End-effector ของการเคลื่อนที่และ กำหนดตัวแปร Geometric parameters ของแต่ละ link มา เพื่อหาตัวแปรในการเคลื่อนที่ ของแต่ละ joint เพื่อให้ ROBOT ไปยังจุดที่ต้องการและมีตัวแปรที่สุดที่ทำให้ไปยังจุดเดียวกัน ได้ ซึ่งวิธีนี้เรียกว่า The Inverse Kinematics (or Arm Solution)

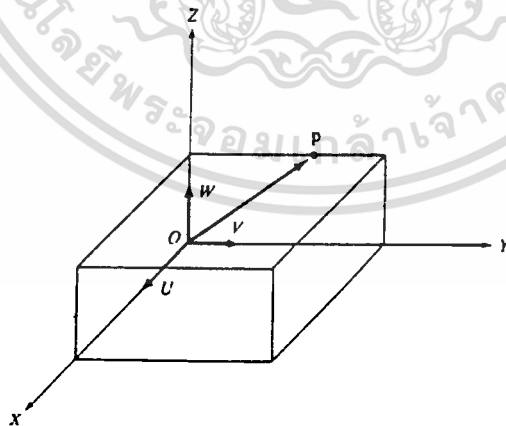
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The Direct Kinematics

Vector และ Matrix Algebra สามารถนำมาประยุกต์ อธิบายตำแหน่งปัจจุบันของ link เมื่ออ้างอิงกับ Fixed Reference Frame กับ Reference Coordinate Frame A 3x3 Matrix สามารถนำมาใช้ในการอธิบายการเคลื่อนที่แบบ Rotation ของ The body-attached frame กับ The reference frame และ Homogeneous Coordinate จะแทน vector ตำแหน่ง 3 มิติของการเคลื่อนที่ A 4x4 Matrix Homogeneous Transformations จะอธิบายการทำงานแบบ Translational และ Rotate

- Rotation Matrix

3x3 Rotation Matrix สามารถใช้แสดงตำแหน่งที่เคลื่อนที่ใน 3 มิติ โดยแสดงการหมุน Coordinate ของ OUVW ซึ่งติดอยู่กับ frame และอ้างอิงกับ Coordinate OXYZ ตามรูป 2.2



จุด P เป็นจุดที่สามารถแสดงตำแหน่งได้ ทั้ง 2 Coordinate เป็น

$$P_{uvw} = (P_u, P_v, P_w)^T$$

และ

$$P_{xyz} = (P_x, P_y, P_z)^T$$

ตัว T หมายถึง Transpose Operation

จากความสัมพันธ์ของ 2 Coordinate ได้ว่า

$$P_{xyz} = R P_{uvw}$$

R คือ Transformation Matrix

- Homogeneous Coordinate and Transformation Matrix

$$P = (WP_x, WP_y, WP_z, W)$$

W เป็น Scale factor สำหรับ ROBOT

$$W = 1$$

A Homogeneous - Coordinate สามารถแทนจุดบน space 3 มิติ ซึ่งเป็น
ประโยชน์ในการหา Transformation Matrix ซึ่งประกอบด้วย rotation ,
translation , scaling และ perspective transformation

$$T = \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ & \\ & \\ & \\ f_{1 \times 3} & 1 \times 1 \end{bmatrix} = \begin{bmatrix} \text{rotation} & \text{position} \\ \text{matrix} & \text{matrix} \\ & - \\ \text{perspective} & \text{scaling} \\ \text{transformation} & \end{bmatrix}$$

Homogeneous Transformation Matrix นี้ใช้แสดงตำแหน่งของจุดบนตัวเฟรม
(OUVW) โดยอ้างอิงกับ OXYZ Coordinate เช่น

$$T_{x..} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_\alpha & -S_\alpha & 0 \\ 0 & S_\alpha & C_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

เป็นการหมุน OUVW ตามมุม α กับแกน X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$T_{y..} = \begin{bmatrix} C_\phi & 0 & S_\phi & 0 \\ 0 & 1 & 0 & 0 \\ -S_\phi & 0 & C_\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

เป็นการหมุน OUVW ทำมุม ϕ กับแกน Y

$$T_{z..} = \begin{bmatrix} C_\alpha & -S_\alpha & 0 & 0 \\ S_\alpha & C_\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

เป็นการหมุน OUVW ทำมุม α กับแกน Z

$$C_\alpha = \cos \alpha, \quad C_\phi = \cos \phi, \quad C_\psi = \cos \psi$$

$$S_\alpha = \sin \alpha, \quad S_\phi = \sin \phi, \quad S_\psi = \sin \psi$$

4X4 Rotation Matrix นี้เรียกว่า Homogeneous Rotation Matrix พื้นฐานในการหมุน

$$T_{tran} = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

เป็นการ Translating OUVW Coordinate ไป (dx, dy, dz) จาก Coordinate อ้างอิง (OXYZ)

4x4 Translation Matrix นี้เรียกว่า Homogeneous Translation Matrix พื้นฐานในการ Translate

โดยให้กับการเคลื่อนที่ 4 แบบ

1. หมุนแกน X_{i-1} เป็นมุม α รอบแกน Z_{i-1}

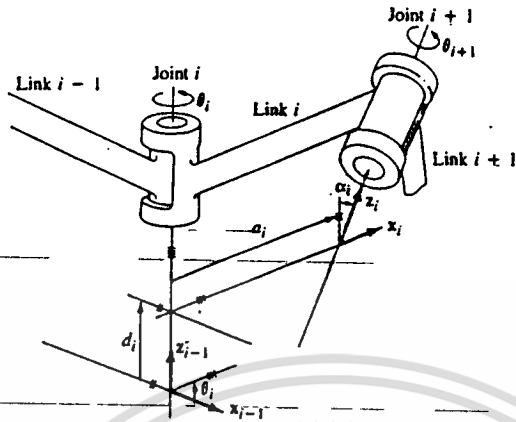
2. เคลื่อนแกน Z_{i-1} เป็นระยะ d_i

3. เคลื่อนแกน X_{i-1} เป็นระยะ a_i

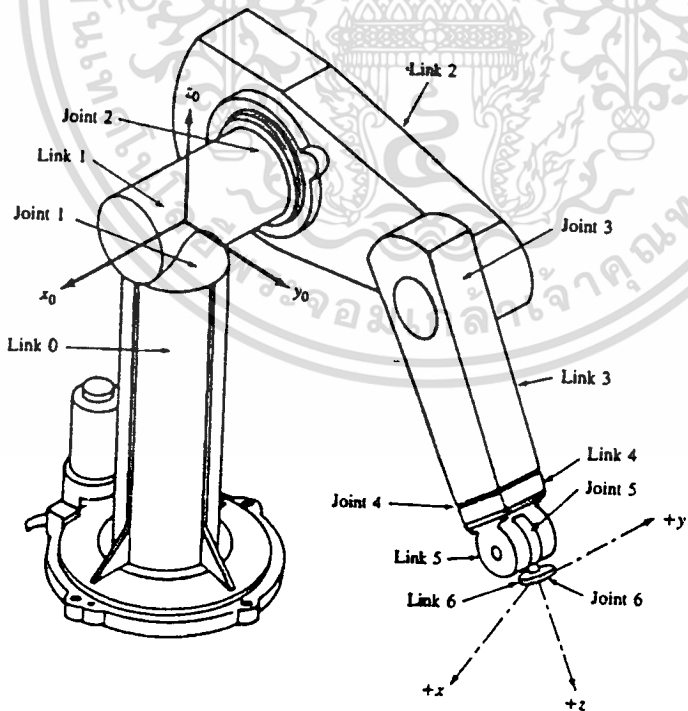
4. หมุนแกน X_i ทำมุม α

เอกสารนี้เป็นเอกสารที่สุพรรณไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลง, 2.8 และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถแสดงให้อยู่ในรูปของ basic homogeneous rotation translation matrix

$${}^{i-1}A_i = T_{z,d} \cdot T_{z,\alpha} \cdot T_{x,a} \cdot T_{x,\alpha}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\alpha_i & -S\alpha_i & 0 & 0 \\ S\alpha_i & C\alpha_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\alpha_i & -C\alpha_i S\alpha_i & S\alpha_i S\alpha_i & a_i C\alpha_i \\ S\alpha_i & C\alpha_i C\alpha_i & -S\alpha_i C\alpha_i & a_i S\alpha_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

P

- Link, Joint and Parameter

การกำหนด Coordinate ของแต่ละ joint

1. แกน Z_{i-1} จะมีทิศทางไปตามการเคลื่อนที่ของ joint ที่ i
2. แกน X_i จะตั้งฉากกับ Z_{i-1} และมีทิศพุ่งออก
3. แกน Y_i จะเป็นไปตามกฎมือขวา (The Right Handed Coordinate System)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดค่า parameter และตัวแปรต่าง ๆ ในแต่ละ joint, link

1. θ_i เป็นมุมจากแกน X_{i-1} กับ X_i โดยวัดรอบแกน Z_{i-1}
2. d_i เป็นระยะระหว่างจุด origin ของ $(i-1)$ coordinate frame ตามแกน Z_{i-1} ถึง X_i
3. a_i เป็นระยะระหว่างแกน Z_{i-1} ถึง X_i โดยวัดระยะตามแกน X_i
4. α_i เป็นมุมจากแกน Z_{i-1} ถึง Z_i โดยวัดรอบแกน X_i

สำหรับ Rotary joint ค่า d_i, a_i, α_i จะเป็น joint parameter ในขณะที่ θ_i จะเป็น joint variable ซึ่งเปลี่ยนไปเมื่อ link i เคลื่อนที่ ตามรูป 2.11

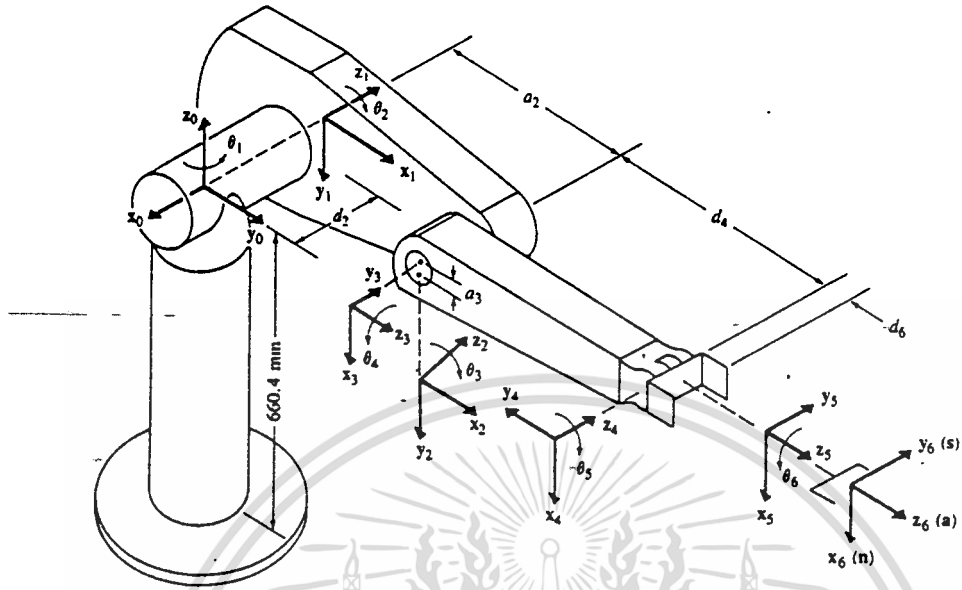
- Kinematics Equation for Manipulator

The homogeneous matrix 0T_i เป็นการกำหนดตำแหน่ง Coordinate ที่ i อ้างอิงกับ base coordinate

$${}^0T_i = {}^0A_1 {}^1A_2 \dots {}^{i-1}A_i$$

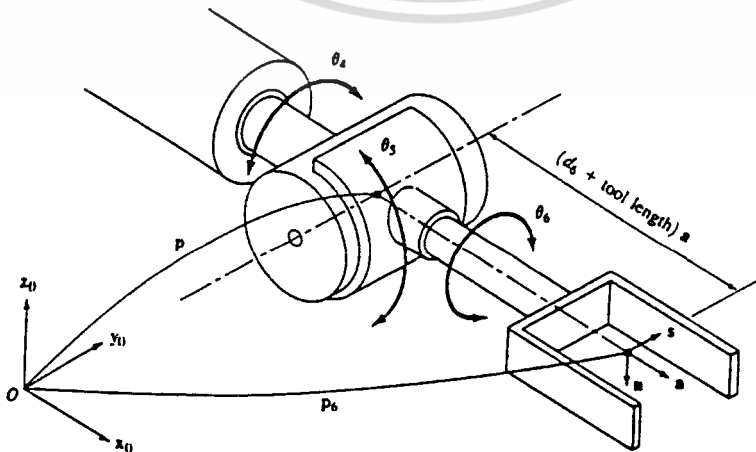
$$= \begin{bmatrix} {}^0R_i & {}^0P_i \\ 0 & 1 \end{bmatrix}$$

P_i เป็น vector จากจุด origin base coordinate ถึงจุดบน i^{th} coordinate ค่าต่าง ๆ ใน matrix ขึ้นอยู่กับตัว parameter ที่กำหนดและค่า joint variable แสดงดังรูปที่ 2.14



PUMA robot arm link coordinate parameters

Joint i	θ_i	α_i	a_i	d_i	Joint range
1	90	-90	0	0	-160 to +160
2	0	0	431.8 mm	149.09 mm	-225 to 45
3	90	90	-20.32 mm	0	-45 to 225
4	0	-90	0	433.07 mm	-110 to 170
5	0	90	0	0	-100 to 100
6	0	0	0	56.25 mm	-266 to 266



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ ROBOT ARM ที่ออกแบบไว้ จะสามารถหาค่า parameter ได้ดังตาราง

joint	θ_i	d_i	a_i	d_i
1	θ_1	-90	100 mm	-40 mm
2	θ_2	0	120 mm	0
3	θ_3	0	130 mm	0
4	θ_4	90	0	0
5	θ_5	0	0	20 mm

$$T = \begin{bmatrix} C_5 i_x - S_5 S_1 & -S_5 i_x - S_1 C_5 & k_x & d_5 k_x + S_x \\ C_5 i_y + S_5 C_1 & -S_5 i_y + C_1 C_5 & k_y & d_5 k_y + S_y \\ C_5 i_z & +S_5 i_z & k_z & d_5 k_z + S_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ค่าต่าง ๆ ได้แสดงไว้ในโปรแกรมแล้ว

บทที่ 2

ขั้นตอนในการสร้างและประกอบ

แขนกล (Robot Arm) ชุดนี้สามารถแยกส่วนประกอบหลัก ๆ ได้เป็น 5 ส่วน คือ

1. ตัวฐาน (Base)
2. ลำตัว (Body)
3. หัวไหล่หรือแขนท่อนบน (Shoulder)
4. ข้อศอกหรือแขนท่อนล่าง (Elbow)
5. ข้อมือและมือจับ (Gripper)

-ตัวฐาน (Base) ประกอบด้วยชุดเฟือง 3 ชุด ซึ่งส่งกำลังขับเคลื่อนจากมอเตอร์ เพื่อขับเคลื่อนให้ส่วนของแขนกลเคลื่อนที่หมุนรอบฐานเป็นมุมต่าง ๆ ตามที่ต้องการ ชุดเฟืองประกอบด้วย เฟืองมอเตอร์ MODULE 1.0 NO. OF TEETH 15 ส่งกำลังไปยังเฟืองทดขนาด NO. OF TEETH 60 MODULE 1.0 จำนวน 2 ชุด ทำให้อัตราการทดทั้งหมด = $60/15 \times 60/15 = 16:1$ แกนเพลลาแต่ละตัวจะสวมอัดกับเฟืองและปลาสตางจะถูกยึดด้วย Bearing เพื่อให้สามารถเคลื่อนไหวได้เหมาะสม ชิ้นส่วนทั้งหมดเป็นเหล็กเพื่อให้มีความแข็งแรงและเหมาะสมกับ load ที่ต้องการให้ขับเคลื่อน

ปัญหาที่เกิดขึ้นในการประกอบ คือ การสวมอัดระหว่างแกนเพลลา กับเฟือง เนื่องจากเครื่องมือที่ใช้ในงานกลึงชิ้นงานมีความละเอียดไม่เพียงพอ จึงทำให้ขนาดของชิ้นส่วนที่ได้มา มีความคลาดเคลื่อน ทำให้การสวมอัดทำได้ยากและต้องใช้แรงในการสวมอัดมากขึ้น ซึ่งเป็นผลทำให้แกนเพลลาเกิดความคดงอ ไม่ได้แนว เมื่อนำชิ้นส่วนทั้งหมดมาประกอบและทดลองขับเคลื่อน จะทำให้เกิดเสียงดัง และทำให้เกิดระยะห่างมากกว่าปกติในการชกกันของฟันเฟือง

ข้อแนะนำในการแก้ไข ควรใช้เครื่องมือในการสร้างชิ้นเพลลาที่มีความละเอียดสูงตามข้อกำหนดในการสวมอัด เพื่อลดความผิดพลาดจากการเสียนวของแกนเพลลา การยึดตัวเพลลา กับตัวฐาน ควรให้มีการปรับเลื่อนได้ เพื่อจัดแนวของเพลลาให้เหมาะสมมากที่สุด เพื่อลดเสียงและระยะห่างระหว่างเฟืองที่มากเกินไป Bearing ที่ใช้ในการรับน้ำหนักของ Body การค้าควรเปลี่ยนเป็น Thrust Bearing เพื่อรับแรงตามแนวแกนน้ำหนักของลำตัวได้ทุกครั้งที่มีการนำไปใช้

ข้อเสนอแนะในการแก้ไข ควรเปลี่ยนมอเตอร์โดยกำหนดให้ Torque ในขณะที่ขับเคลื่อนเพียงพอกับ load ที่เกิดขึ้น และเพิ่มขนาดของเพลานกที่มีสวมอัดและเพิ่มตัวประกบยึดระหว่างเฟืองตามกับเฟืองส่งเพื่อลดการ Slide ที่เกิดขึ้น สำหรับสายพานควรเปลี่ยนขนาดให้เหมาะสมและเพิ่มตัวตั้งความตึงของสายพานซึ่งกำหนดไว้แล้วในบทอ้างอิง

-ข้อศอกหรือแขนท่อนล่าง (Elbow) ประกอบด้วย frame อลูมิเนียมและชุดเฟืองทดส่งกำลังไปยัง Gripper และแขนท่อนล่างให้เคลื่อนที่

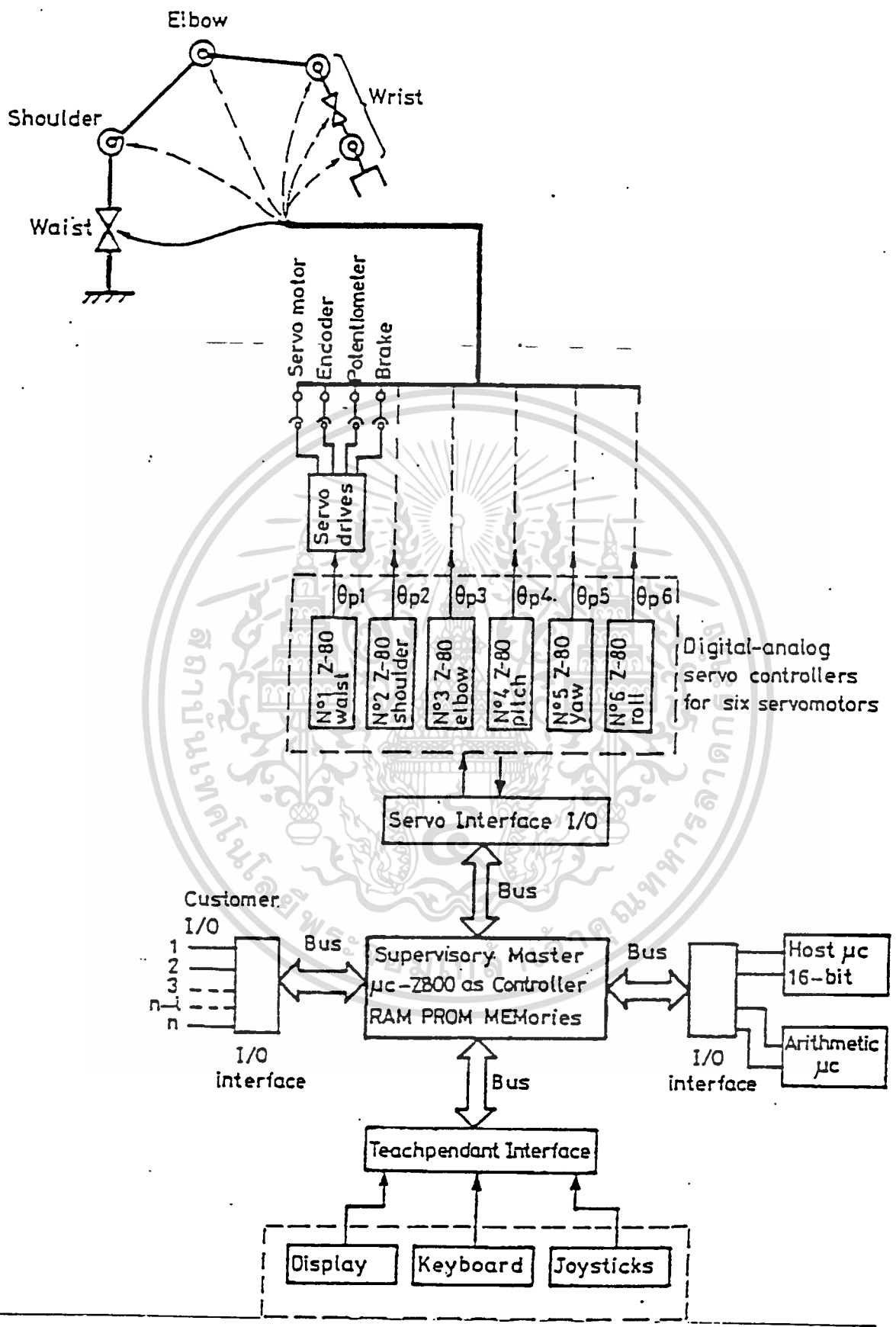
ปัญหาที่เกิดขึ้น การกลิ้งปลอกเพลานกที่มีขนาดเล็ก และต้องสวมอัดกับเฟืองทำได้อาก มีความผิดพลาดสูงเกี่ยวกับขนาด จึงทำให้เวลาประกอบเกิดการหลวมระหว่างปลอกและเพลากลางมากเกินไป ประกับยึดเพลากลางกับ frame ออกแบบการยึด screw ไว้ไม่เหมาะสมยากแก่การยึดให้แน่นเพราะเนื้อที่มีจำกัดเกินไป

ข้อเสนอแนะในการแก้ไข ควรใช้เครื่องส่งในการกลิ้งที่มีความละเอียดสูง เพื่อให้เกิดความแม่นยำในการประกอบ สำหรับตัวประกบยึด frame ควรเปลี่ยนการไขยึดให้ไขจากด้านนอกเข้าไป เพื่อเพิ่มพื้นที่และความสะดวกในการตั้งแนวแกนเพลานก

-ข้อมือและตัวจับ (Gripper) ส่วนข้อมือทำด้วยเหล็กแผ่นบาง ประกอบด้วยชุดเฟืองขับเคลื่อน 2 ชุด ชุดแรกส่งกำลังจากสายพานเพื่อขับเคลื่อนมือให้หมุนขึ้น-ลง และชุดที่ 2 ประกอบด้วยเฟืองดอกจอก เพื่อเปลี่ยนทิศทางการหมุนไปยังตัวมือจับให้หมุนได้ ส่วนตัวมือจับนั้นยังไม่ได้สร้างชิ้น

ปัญหาที่เกิดขึ้น เหมือนกับแขนท่อนล่าง และการตั้งแนวของเฟืองดอกจอกทำได้อาก

ข้อเสนอแนะในการแก้ไข ควรลดขนาดหรือน้ำหนักของ Gripper ลงบ้าง เพื่อลด Torque ในการขับเคลื่อน



เอกสารนี้เป็นเอกสารภาพแสดงการทำงานของระบบกลที่มี 6 แกนนั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การติดต่อกับ COMPUTER เพื่อควบคุม ROBOT ARM

key board เป็น Input Device ที่ User ใช้ติดต่อกับ Computer เพื่อทำการรับ Data ที่ User ส่งเข้าไปควบคุม Robot Arm

Data ที่ส่งผ่านทาง Key Board จะถูกส่งไปที่ Tech Pendant Interface ซึ่งเป็นส่วนที่รับ Data มาจาก Input Device แล้วทำการ Interface ข้อมูลนั้น ต่อไปที่ CPU ของ μ -Computer อีกที

ที่ μ -Computer จะทำการคำนวณโปรแกรม เพื่อหาค่าที่จะใช้ส่งไปควบคุม stepping motor โดยนำค่า Input ที่รับเข้ามาจาก User อันได้แก่ ความเร็ว, ทิศทางการหมุนและขนาดของมุมที่หมุน มาแปลงเป็นค่า โดย ความเร็วก็จะแปลงเป็นความถี่, ขนาดของมุมแปลงเป็น pulse

Data ที่คำนวณและแปลงเป็นค่าที่จะส่งไป drive motor จะถูกส่งไปที่ Servo-Interface ซึ่งเป็น Card ที่ใช้แปลง data ไปเป็นสัญญาณ โดยมี IC เบอร์ 8253 จำนวน 4 ตัวอยู่บน card ทำหน้าที่เป็นตัวนับที่สำคัญ

จากนั้นก็เป็นการ Interface สัญญาณ Data ระหว่าง Card หรือ Servo-Interface กับ controllers ของ Stepping Motor ซึ่งมี 6 port สำหรับต่อกับ Stepping Motor 6 ตัว

Controller ประกอบด้วย 2 ส่วน คือ Sequential Logic และ Driver ซึ่งใช้ IC เบอร์ 8713 เป็นตัวจัดลำดับสัญญาณ

เมื่อ Stepping ได้รับสัญญาณเข้ามา ก็จะทำงานโดย Stepping Motor แต่ละตัวจะมีตำแหน่งอยู่ตาม Joint ต่างๆของ Robot Arm ทำให้ Robot Arm เกิดการเคลื่อนที่เปลี่ยนตำแหน่ง (position) จากเดิมออกไป โดยจะแสดง position ใหม่ที่เปลี่ยนแปลงให้ User ดู

ได้ด้วยการคำนวณจาก Program แล้วแสดงผลออกมาทาง Output Device ซึ่งในที่นี้คือ Monitor ของ μ -computer นั้นเอง

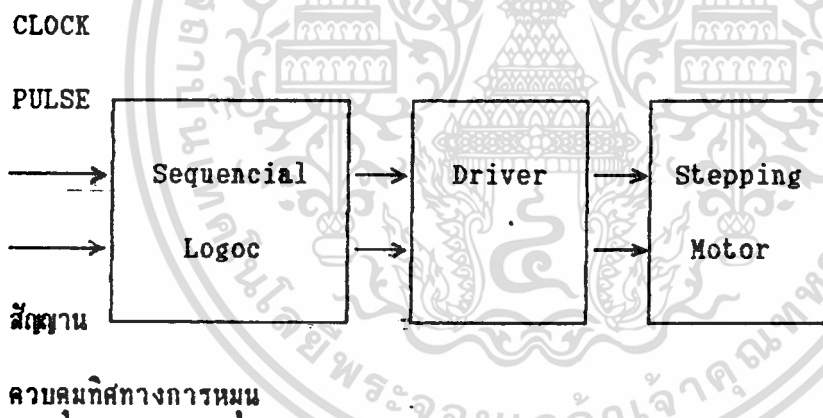
CONTROLLER ของ STEPPING MOTOR

Controller เป็น อุปกรณ์ชิ้นหนึ่งซึ่งมีหน้าที่ส่งสัญญาณไปควบคุม stepping motor

Controllers จะประกอบด้วย 2 ส่วนใหญ่ๆ คือ

- วงจรลอจิกสำหรับจัดลำดับ (Sequential Logic)
- ตัวขับ (Driver)

ดังในรูป



จากในรูป พบว่า มี 2 สัญญาณที่ส่งเข้าไปที่ Controllers คือ

- สัญญาณนาฬิกา (Clock Pulse)
- สัญญาณควบคุมการหมุน

สัญญาณทั้ง 2 นี้มาจากการคำนวณโปรแกรมของ μ -computer แล้วส่งต่อไปให้ servo ไม่-interface ถ้ามีการแปลงเป็นค่าสัญญาณ ก่อนส่งมาให้ controller เอกสารทุกครั้งที่มีการนำไปใช้

DRIVER ของ STEPPING MOTOR

ตัวขับของ Stepping Motor เป็นชุดวงจรที่มี IC PMM 8713 ซึ่งเป็น IC ของบริษัท SANYO DENKI ประเทศญี่ปุ่น เป็น IC ประเภท C-MOS monolithic สำหรับขับ Motor ชนิด 3 เฟส และ 4 เฟส และสามารถขับได้ทั้งแบบกระตุ้น 1 เฟส, กระตุ้น 2 เฟส, และกระตุ้น 1-2 เฟส

PMM 8713 สามารถเลือก Mode ในการขับ Stepping Motor ได้ 6 Mode คือ

- 4 เฟส , กระตุ้น 1 เฟส
- 4 เฟส , กระตุ้น 2 เฟส
- 4 เฟส , กระตุ้น 1-2 เฟส
- 3 เฟส , กระตุ้น 1 เฟส
- 3 เฟส , กระตุ้น 2 เฟส
- 3 เฟส , กระตุ้น 1-2 เฟส

โดยใช้ไฟเลี้ยง 4V+18 ให้กระแส Output อย่างต่ำ 20 m.A. และเลือกสัญญาณขา เข้าได้ 2 ลักษณะคือ

-2 Input Terminal

-1 Inout , 1 ตัวเปลี่ยน Terminal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INTEL 8253

Intel 8253 เป็น IC ของบริษัท Intel แห่งอเมริกา ซึ่งในทันทีทำหน้าที่เป็นตัวนับ ที่สามารถโปรแกรมให้ทำงานได้ โดยสั่งงานผ่าน Control Register

Intel 8253 จะสร้างสัญญาณตัวนับไปควบคุมให้ Stepping Motor ทำงานโดยควบคุม 3 ตัวแปรสำคัญ คือ

- 1) องศาที่หมุนไปของ Stepping Motor ซึ่งจะสัมพันธ์กับ Pulse ที่ป้อนให้กับ Motor
- 2) ความเร็วในการหมุนของ Stepping Motor ควบคุมโดยความถี่ที่ป้อนให้กับ Motor
- 3) ทิศทางในการหมุน

ใน IC 8253 แต่ละตัวจะมีตัวนับสัญญาณอยู่ 3 ตัว ซึ่งจะใช้ตัวนับ 2 ตัวต่อ motor 1 ตัว และมี stepping motor ที่ต้องควบคุม 6 ตัว ดังนั้นจึงต้องใช้ IC 8253 ทั้งหมด 4 ตัว เพื่อให้มีจำนวนตัวนับสัญญาณ 12 ตัว

ตัวนับตัวแรกจะเป็นตัวกำหนดสัญญาณรูปสี่เหลี่ยม (Square Wave Generator) ที่คาบ $T=70$ ns ดัง

$$f = \frac{1}{T}$$

$$= \frac{1}{70 \text{ ns}}$$

$$= 14.31818 \text{ M.Hz}$$

$$= 14.31818 \text{ M.Hz}$$

และทำการปรับเปลี่ยนให้เป็นสัญญาณที่มีความถี่ 447.443 KHz

$$f' = \frac{f}{32}$$

$$32$$

$$= \frac{14.31818 \text{ MHz}}{32}$$

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้ใช้งานเพื่อการศึกษานี้เท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้ง 32 มม. ให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$= 447.0443 \text{ KHz}$$

ตัวนับตัวที่ 2 จะนับจำนวน Pulse ที่ส่งออกไปให้ Stepping Motor จนเมื่อส่งไปครบแล้ว ก็จะมีสัญญาณออกที่ขา Out เพื่อส่งไปให้ตัวนับตัวรถหยุดการหารสัญญาณของ 447.443 KHz

การใช้งานของ 8253 ใน Mode ต่างๆ มีดังนี้

Mode 0 การนับเพื่อสร้างสัญญาณขัดจังหวะ (Interrupt of Terminal Count) คือการนับลงตามความถี่สัญญาณนาฬิกาที่ป้อนเข้ามา เมื่อเป็น 0 ก็จะทำให้ Output เป็น 1 และจะกลับเป็น 0 เมื่อป้อนค่านับใหม่ให้กับ Counter

Mode 1 การสร้าง Pulse ที่โปรแกรมช่วงเวลาได้ (Programmable One-Shot) คือการสร้าง pulse ที่สามารถกำหนดความกว้างได้ โดยเมื่อมีการกระตุ้นที่ขา gate ก็จะทำให้ Output เป็น 0 เท่ากับจำนวนเต็มของ Pulse ของสัญญาณนาฬิกา

Mode 2 ตัวกำเนิดสัญญาณ (Rate Generator)

คือการหารสัญญาณนาฬิกาด้วย N ที่โปรแกรมเข้ามาโดยรูปแบบ Output จะมี 0 อยู่ 1 pulse ต่อ 1 จำนวน N pulse

Mode 3 ตัวกำเนิดสัญญาณรูปสี่เหลี่ยม (Square Wave Generator)

การทำงานคล้ายกับ mode 2 แต่รูปแบบ output จะมีลักษณะที่ 0 และ 1 ที่สลับกัน

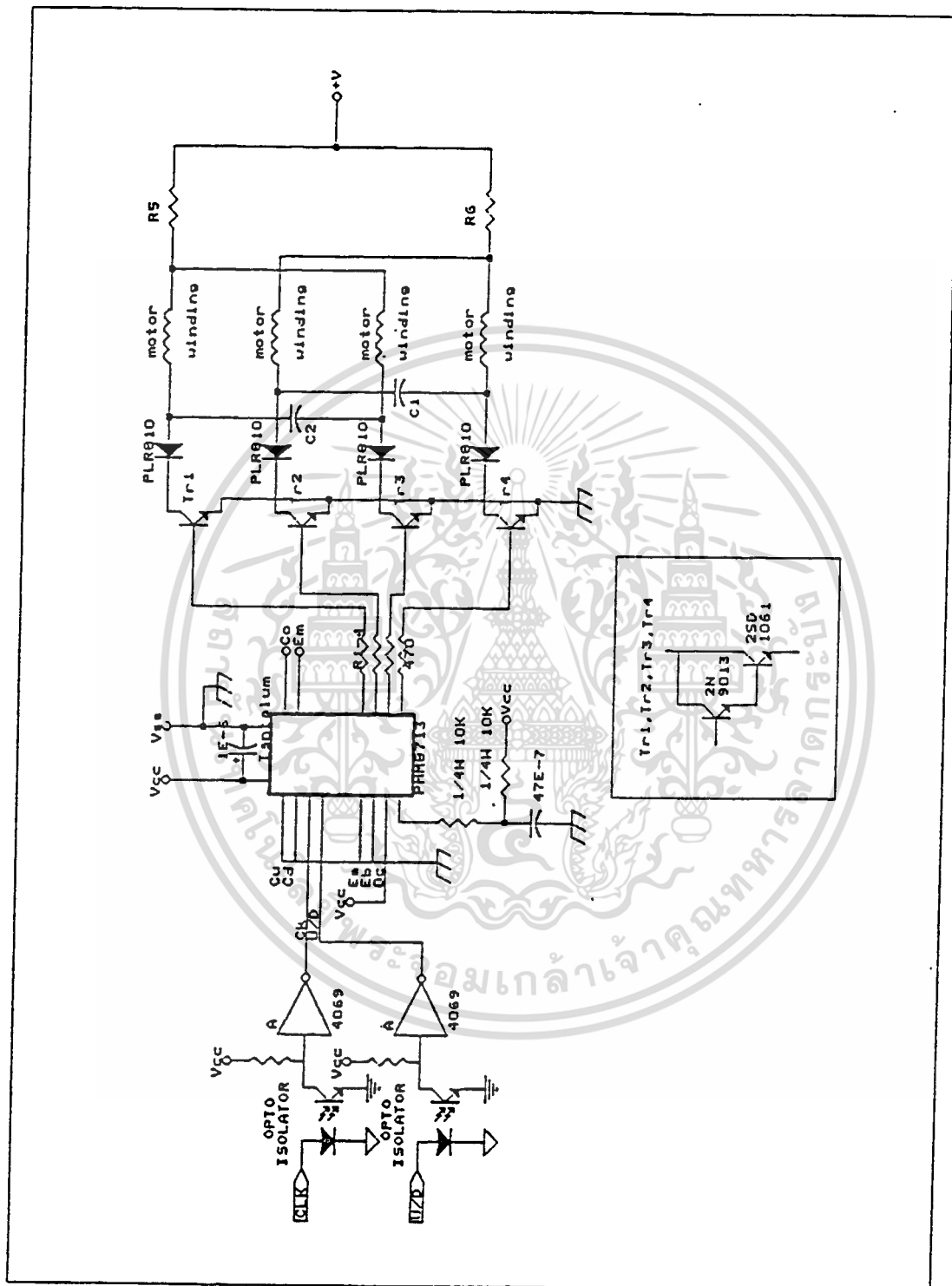
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 Mode 4 การหน่วงเวลาด้วย Software (Software Triggered Generator) ไม่วาระณใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 การทำงานใน mode นี้ ตัวนับจะให้ output จำนวน 1 pulse โดยที่มีการหน่วง

Mode 5 การหน่วงเวลาด้วย HardWare (HardWare Trigged Store)

การทำงานเหมือนกับ mode 4 แต่การกระตุ้นให้เริ่มหน่วงเวลาต้องทำโดย
ป้อน input เข้าที่ขา gate จากนั้นก็จะเริ่มนับ เมื่อนับครบแล้วก็จะให้
output เป็น 0 จำนวน 1 pulse

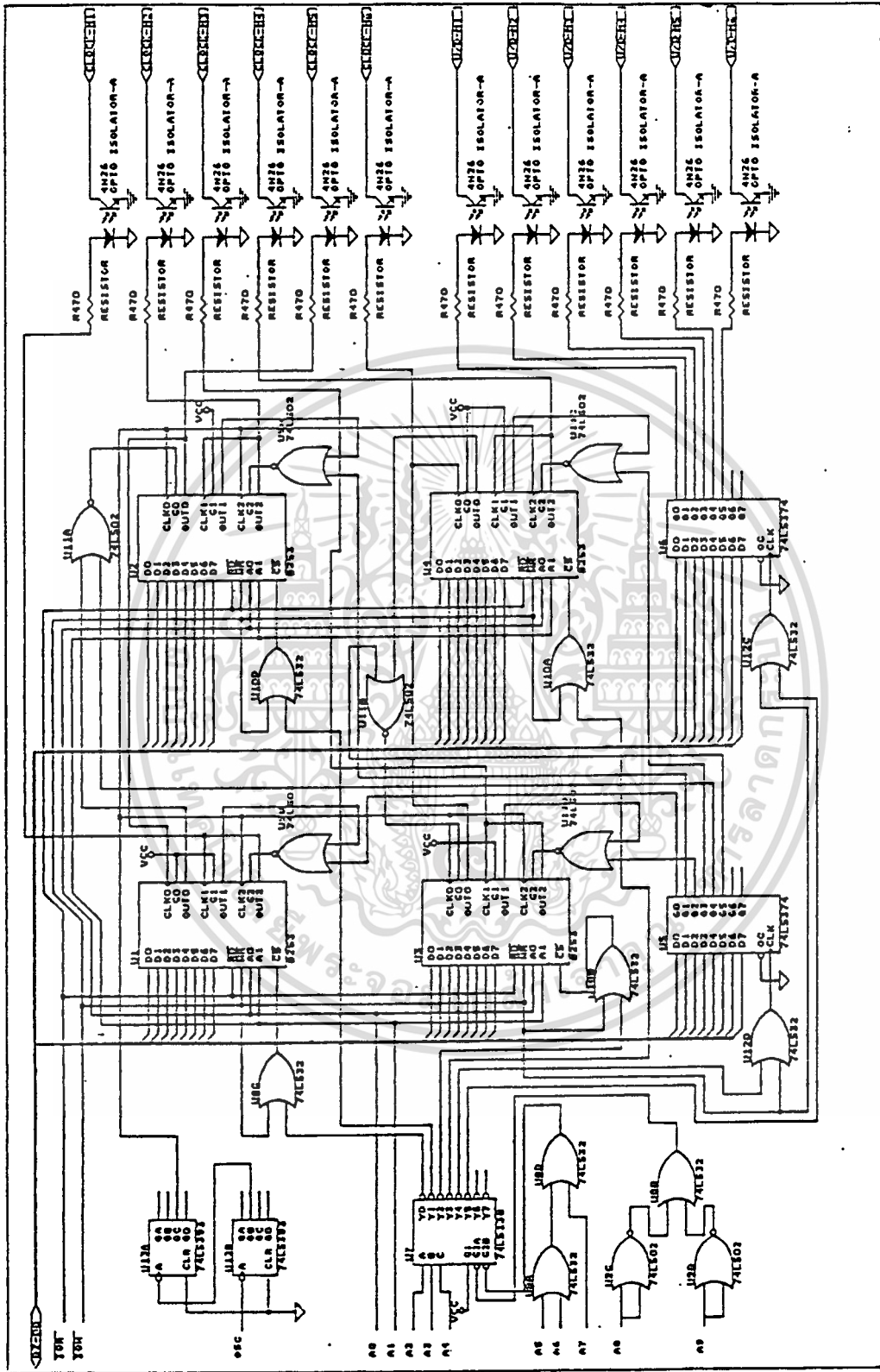


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แสดงวงจรสำหรับ PHM8713 เฉพาะมอเตอร์ 1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แสดงวงจรรวม 8253 ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอร์ท	แอดเดรส
8253 คิวที่ 1	300H-303H
8253 คิวที่ 2	304H-307H
8253 คิวที่ 3	308H-30BH
8253 คิวที่ 4	30CH-30FH
นอร์ททิศทาง	310H-313H
นอร์ทเริ่มนับ	314H-317H

และตารางการใช้งาน 8253 ควบคุมมอเตอร์ดังนี้

มอเตอร์	นอร์ท	หน้าที่
1	302H 301H	ตัวหารเพื่อกำหนดความเร็วในการหมุนมอเตอร์ ตัวนับจำนวนพัลส์ ควบคุมจำนวนองศาของมอเตอร์
2	306H 305H	ตัวหารเพื่อกำหนดความเร็วในการหมุนมอเตอร์ ตัวนับจำนวนพัลส์ ควบคุมจำนวนองศาของมอเตอร์
3	30AH 309H	ตัวหารเพื่อกำหนดความเร็วในการหมุนมอเตอร์ ตัวนับจำนวนพัลส์ ควบคุมจำนวนองศาของมอเตอร์
4	30EH 30DH	ตัวหารเพื่อกำหนดความเร็วในการหมุนมอเตอร์ ตัวนับจำนวนพัลส์ ควบคุมจำนวนองศาของมอเตอร์
5	304H 300H	ตัวหารเพื่อกำหนดความเร็วในการหมุนมอเตอร์ ตัวนับจำนวนพัลส์ ควบคุมจำนวนองศาของมอเตอร์
6	308H 30CH	ตัวหารเพื่อกำหนดความเร็วในการหมุนมอเตอร์ ตัวนับจำนวนพัลส์ ควบคุมจำนวนองศาของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนนอร์ทควบคุมทิศทางมีการใช้งานดังนี้

นอร์ท	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
310H-313H	--	--	M6	M5	M4	M3	M2	M1

ส่วนนอร์ทควบคุมการเริ่มนับมีการใช้งานดังนี้

นอร์ท	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
314H-317H	--	--	M6	M5	M4	M3	M2	M1

* โดยที่ เมื่อบิตใดเป็น 0 คือการเริ่มนับ ถ้าเป็น 1 คือหยุดนับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การใช้โปรแกรมควบคุม ROBOT ARM

ROBOT เป็นโปรแกรมที่ใช้ควบคุม Robot Arm ซึ่งเขียนขึ้นด้วยภาษา C โดยใช้ Turbo C Version 2.0 ของ Borland International , USA

ลักษณะของโปรแกรม

โปรแกรม Robot ประกอบด้วย Windowใหญ่ และในWindowใหญ่ก็ประกอบด้วย Window ส่อๆอีกคือ Window ด้านซ้ายมือบน, Window ด้านซ้ายมือล่าง, และ Windowด้านขวาล่างมีรายละเอียดต่างๆดังนี้คือ

- Window ใหญ่ จะแสดงกรอบหรือขอบเขตใช้สอยของโปรแกรม
- Window ซ้ายบน จะแสดงรายละเอียดของ Part และเบอร์ Port ของแขนกล ที่สามารถควบคุมได้ ซึ่งมีทั้งหมด 5 ส่วน
- Windowซ้ายล่าง เป็นส่วนที่ไว้สำหรับรับ Input และโต้ตอบกับUser และให้แสดงผล Position ที่เปลี่ยนไปของ Robot Arm

นอกจากนี้ ยังมี Details ซึ่งเป็นตัวอักษรแทรกอยู่ระหว่าง Window ต่างๆอีกคือ

- Details ขวาบน ซึ่งแสดงด้วยตัวอักษรสีแดง บอกถึง DATA ที่รับเข้ามาและ Show Data ที่ตำแหน่งเดียวกับ Part ของแขนกลที่จะทำงาน
- Details ซ้ายกลาง เป็นตัวอักษรสี Magenta แสดงถึง Mode

ให้เลือกเพื่อสั่งให้โปรแกรมทำงานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม ROBOT.C ประกอบด้วยไฟล์ทั้งหมด 3 ไฟล์คือ

- 1) ROBOT.C คือโปรแกรมที่เรียกว่า Source File ซึ่งให้ปรับปรุงแก้ไขตัวโปรแกรม หรือเป็นไฟล์สำหรับใช้ EDIT โปรแกรมนั่นเอง
- 2) ROBOT.OBJ เป็นไฟล์ที่มาจากการนำเอาไฟล์ ROBOT.C มาทำการ Compile โดยที่ Turbo C Compiler
- 3) ROBOT.EXE เป็นมาจากการนำเอา Robot.com มาทำการ Link ด้วย Turbo Link และเป็นไฟล์ที่ใช้ทำการ Run เพื่อควบคุม แขนกลซึ่งไฟล์ Robot.exe ต้องอยู่ในไดเรกทอรี ที่มีโปรแกรม EGAVGA.BGI เท่านั้นเพราะเป็นโปรแกรมที่มีการใช้ในแบบ Graphics Mode

ส่วนประกอบที่สำคัญของโปรแกรม

ภาษา C ถือเป็น Structure-Program อย่างหนึ่ง คือมีการเขียนในแบบโครงสร้าง และจะใช้การเขียน Function เพื่อช่วยการทำงานของ Main โปรแกรม ซึ่ง Function นี้เองที่มีส่วนสำคัญต่อโปรแกรม Robot ดังต่อไปนี้

- 1) ส่วนที่ทำให้โปรแกรมทั้งหมดทำงานใน Graphics Mode คือ ฟังก์ชัน `Rassarin()` ซึ่งจะเปลี่ยนจาก Text mode ให้มาเป็น Graphics Mode โดยที่คำสั่ง `Detectgraph` และ `Initgraph`

```
detectgraph(&g_driver,&g_mode)
```

```
initgrapg(&_driver,&g_mode)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ คำสั่ง `detectgraph` จะระบุชนิดของ `driver` และ `mode` เพื่อส่งค่าที่ได้ไปให้กับ

คำสั่ง `initgraph` ทำการเปลี่ยนจาก Text Mode ไปเป็น Graphic Mode

2) ส่วนที่ป้อนค่าให้แก่ IC-8253 ซึ่งจำเป็นต้องป้อนค่าเหล่านี้ให้แก่ IC ก่อนที่จะใช้งานได้โดยใช้ฟังก์ชัน Initialize() ป้อนผ่านเบอร์ Port ด้วยคำสั่ง
 Outputb

```
outputb(0x303,0x30)
```

```
outputb(0x303,0x70)
```

```
outputb(0x303,0x86)
```

เห็นได้ว่าในคำสั่ง outputb จะส่งข้อมูลคือ 0x30 , 0x70 , 0x86 ไปออกทาง port เบอร์ 0x303 ตัวเลขเหล่านี้เป็นเลขฐาน 16 ทั้งสิ้น ซึ่งรายละเอียดของข้อมูลแต่ละตัวมีดังนี้ คือ

0x30 เป็นการเลือก counter 0 และ ส่งข้อมูลออกไปในแบบ binary counter 16 bit

0x70 เป็นการเลือก counter 1 และ ส่งข้อมูลออกไปในแบบ binary counter 16 bit

0x86 ทำงานใน mode 3 ส่งสัญญาณออกไปในรูปของ square wave

3) ส่วนรับข้อมูลและแสดงข้อมูล ซึ่งได้แก่ ฟังก์ชัน choose() และ data() โดย User ต้องใส่จำนวน Part ที่จะควบคุมก่อนแล้วจึงบอกส่วน Part ที่จะควบคุม จากนั้นจึงใส่ระดับความเร็ว, ทิศทาง, และมุมที่จะควบคุมผ่าน part ของ Robot

Arm ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ User สามารถระบุระดับความเร็วได้ 3 ระดับ คือ L, M, H ซึ่งโปรแกรมจะไป

ทำการเปลี่ยนเป็น ความถี่ที่ป้อนให้กับstepping motor อีกทีหนึ่ง

ในส่วนของทิศทางให้ระบุก่อนที่จะป้อนองศาที่หมุน เป็นทิศ + หรือ - ซึ่งโปรแกรมจะ
ไปเรียกฟังก์ชัน Turning(direct) มาใช้งานดังนี้

```
if(direct == '+')
    outportb(0x310,0x00);
else if(direct == '-')
    outportb(0x310,0x3f);
```

การทำงานก็คือ ถ้า direct เป็นบวกก็จะส่งสัญญาณ 0 ออกทุกบิตโดยผ่าน port
เบอร์ 0x310 ในกรณีที่ direct เป็นลบก็จะส่งสัญญาณ 1 ออกทุกบิต โดยผ่าน
port เบอร์ 0x310 เช่นกัน

4) ส่วนคำนวณและแสดงผล Position ของแขนกลที่เคลื่อนที่ไปคือ ฟังก์ชัน Input()
Forward(), และ Output() เมื่อผู้ใช้ป้อน Data ให้กับ micro computer เสร็จ
สิ้นแล้ว จากนั้นเครื่องก็จะทำการนำเอามุมที่ป้อน เข้าไปมาคำนวณหา Position
ใหม่ของ Robot Arm แล้วนำออกแสดงผลทาง Monitor

ในส่วนของโปรแกรมจะแปลงมุมที่ส่งเข้ามาจาก degree เป็น radiant ดังคำสั่งข้างล่าง

$$d[i] = (pi * d[i]) / 180$$

ซึ่งค่า pi ในที่นี้ใช้เป็น 3.14285714

5) ส่วนที่ส่งข้อมูลไปออก port คือ ฟังก์ชัน Port() ซึ่งข้อมูลที่ป้อนผ่านออกไปก็
ได้แก่

Speed-Level, Direction, และมุม

```
pulse[i] = abs(ddeg[i]*ratio)/1.8 ;
```

```
turning(direct);
```

```
outportb(0x308, freq[i]);
```

```
outportb(0x308, freq[i]>> 8);
```

```
outportb(0x30c, pulse[i]);
```

```
outportb(0x30c, pulse[i]>> 8);
```

โปรแกรมจะทำการคำนวณหาค่า pulse เพื่อส่งไป drive stepping motor โดยนำมุมที่ตั้งให้ robot arm เคลื่อนที่ คูณกับอัตราทดของชุดเกียร์ที่แต่ละ part ซึ่งจะมีอัตราทดไม่เท่ากัน จากนั้นก็ส่งค่า data อันได้แก่ frequency และ pulse ไปออก port

การนำไปร่นกรมไปใช้งาน

หลังจากได้เขียน source code เสร็จสิ้น และทำการ compile และ Link แล้ว จะได้ executable file ที่ชื่อ Robot.exe ซึ่งใช้ขับเคลื่อน Robot Arm ให้ทำการ run ทดสอบ

ในตอนแรก User จะพบกรอบและ Window ข้างในกรอบได้แก่ Window ที่ แสดง Part ที่ User จะเลือกทำการควบคุมซึ่งมีทั้งหมด 5 Part คือ

- A: Wrist Part
- B: Showlder Part
- C: Elbow Part
- D: Pitch part
- E: Roll Part

และอีก Window ทางซ้ายด้านล่าง ซึ่ง User ใช้ Input data โดย data ที่ User ต้องใส่เข้าไปได้แก่

-จำนวน Part ของแขนกลที่จะควบคุม

User สามารถควบคุมได้ตั้งแต่ 1-5 part พร้อมกัน

-ระบุ Part ของแขนกลที่จะควบคุมตามจำนวนที่ระบุในตอนแรก

-บอกระดับความเร็ว ซึ่งในที่นี้ ให้ผู้ใช้ได้เลือก 3 ระดับ

1) L: (Low Speed) ช้า

2) M: (Medium Speed) ปานกลาง

3) H: (High Speed) เร็ว

-ระบุทิศทางที่จะให้ แขนกลหมุนมี 2 ทิศทางคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัด 1) ทิศทางเริ่มนาฬิกา "+" ถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ทิศตามเริ่มนาฬิกา "-"

-บอกจำนวนองศาที่หมุนไป

ในขณะที่ User ทำการ Input Data เข้าไป โปรแกรมก็จะแสดง DATA ที่ป้อนโดยแสดงผลทางด้านขวามือให้ตรงกับ Part ที่กำลังควบคุม ด้วยตัวอักษรสีแดง เมื่อป้อน data เข้าไปจนครบแล้ว โปรแกรมก็จะคำนวณและแสดง Position ของ Robot Arm ที่หมุนไปโดยเทียบกับ Reference เดิม ที่ Window ขวามือด้านล่าง ต่อจากนั้น User ก็ต้องเลือก Mode ซึ่งเป็นการ เลือกว่าจะใช้โปรแกรม อย่างไรต่อไป มี 3 Mode คือ

- 1) สั่งให้ Robot Arm มีการหมุนอย่างต่อเนื่อง
- 2) เปลี่ยน Part ของ Robot Arm ที่จะทำการควบคุม
- 3) ออกจากโปรแกรม

ผู้เขียนหวังเป็นอย่างยิ่งว่า โปรแกรม Robot นี้จะช่วยให้นักศึกษาชั้นคณา
เรื่อง " ROBOT ARM " ในภายหลัง ได้สามารถเข้าใจการทำงานของโปรแกรมอย่าง
ถ่องแท้และใช้โปรแกรมอย่างสะดวกและรวดเร็ว

บทที่ 5

ผลการทดลอง

จากการทำการทดลอง drive แขนกลโดยป้อน speed-level,direction และมุมที่ให้แขนกลหมุน ใน mode ที่ต่างๆ กัน คือ

- mode 1 เป็นการสั่งให้ robot arm หมุนแบบต่อเนื่องโดยอ้างอิงจาก reference เดิมตลอด

เช่น ครั้งแรก user สั่งให้robot arm หมุนไปที่ 30 องศา จากนั้น user สั่งให้ robot arm หมุนไปที่ 45 องศาอีกที โปรแกรมก็จะทำการคำนวณหาค่ามุมที่ต้องเพิ่มเพื่อทำให้ robot arm เคลื่อนจาก 30 องศาไปสู่ที่ 45 องศา

$$\begin{aligned}
 \text{มุมที่ต้องเพิ่ม} &= \text{มุมใหม่} - \text{มุมเก่า} \\
 &= 45 - 30 \\
 &= 15 \text{ องศา}
 \end{aligned}$$

จากนั้นโปรแกรมจะหาค่า pulse ที่ robot arm ต้องใช้ในการส่ง robot arm จาก 30 องศา ไปถึง 45 องศา ซึ่งจากผลการทดลอง ทำให้รู้ว่า pulse ที่ใช้คือ 133

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มุม	ความเร็ว	PULSE	POSITION(X, Y, Z)
30	L	266	(303, 175, 60)
45	L	133	(303, 175, 60)
60	L	133	(174, 175, 60)
90	L	266	(0, 349, 60)
180	L	799	(349, 0, 60)
360	L	1599	(349, 0, 60)

ตารางแสดงผลการทดลองโดยใช้ mode 1

*** หมายถึง เมื่อ L แทนค่าของระดับความเร็วโดยใช้ freq. เท่า
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออยู่ในเขตพื้นที่ให้บริการ
ไม่ว่ากรณีใดๆ ห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- mode 2 เป็น mode ที่ user สั่งให้ robot arm เปลี่ยน reference ทุกครั้งเมื่อมีการหมุน

เช่น ครั้งแรก user สั่งให้ robot arm ทำงานใน mode 2 โดสให้ robot arm หมุนไป 30 องศา เมื่อสิ้นสุดการทำงานแล้ว user เลือกทำงานใน mode 2 อีกครั้ง จะทำให้โปรแกรมรับรู้ว่า ขณะนี้ robot arm มีการเปลี่ยน reference ไปใหม่แล้ว คือคิดว่า ที่ 30 องศาของ reference เดิม เป็น 0 องศา หรือเป็น reference ใหม่ซึ่งจะใช้ต่อไป

มุม	ความเร็ว	PULSE	POSITION(X,Y,Z)
30	L	266	(303,175,60)
45	L	399	(303,175,60)
60	L	533	(174,175,60)
90	L	799	(0,349,60)
180	L	1599	(349,0,60)
360	L	3199	(349,0,60)

ตารางแสดงผลการทดลองโดยใช้ mode 2

***หมายเหตุ เมื่อ L แทนค่าของระดับความเร็วโดยให้ freq. เท่า
กับ 4 K.Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลองและวิจารณ์

จากผลการทดลองที่ได้แสดงไปแล้ว ทำให้บอกได้ว่า user สามารถควบคุม robot arm ในแบบ relative coordinate นั่นคือ user ต้องใส่ input ที่เป็นมุมที่สั่งให้ robot arm หมุนในรูปของตำแหน่งที่ robot arm จะไป ณ.ตำแหน่งนั้น

เช่น ถ้า user ป้อนค่ามุมให้กับ robot arm เป็น 90 องศา นั่นก็หมายความว่า robot arm จะหมุนไปที่ตำแหน่ง ที่มุม 90 องศา กับ reference เดิม หรือที่ 0 องศา นั่นเอง

นอกจากนี้ยังพบว่า มีความคลาดเคลื่อน (error) ของตำแหน่งที่ robot arm หมุนไป ซึ่ง จะมากหรือน้อยก็ขึ้นอยู่กับ effect เหล่านี้คือ

- ความเร็ว ยิ่งป้อน speed ให้ robot arm มากเท่าใด ค่าเปอร์เซ็นต์ error ก็จะมีมากขึ้นตามไปด้วย ที่เป็นเช่นนี้เพราะการรอกตัวช่วงแรกของเพลาคที่ใช้ขับเคลื่อน จะไม่สามารถหมุนได้ทันสัญญาณไฟที่จ่ายเข้ามา ในช่วงแรกๆได้ หรือเรียกได้ว่า ระบบทาง mechanics ยังตามระบบทาง electronics ไม่ทัน
- ความผิดของอุปกรณ์ทาง mechanis อันจะทำให้ต้องสูญเสียพลังงานไปบางส่วนเพื่อเอาชนะความผิดนั่นเอง

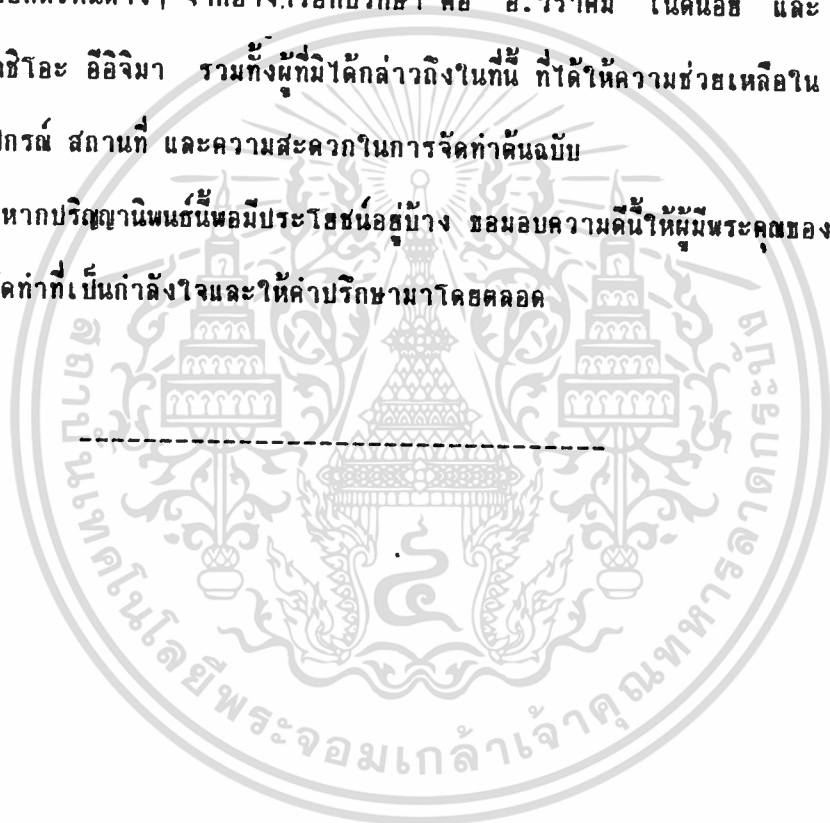
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปยังเว็บไซต์หรือสื่ออื่นใด
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเอียงตัด ซึ่งทำให้การส่งกำลังจากเฟืองขับที่ส่งไปให้เฟืองตัวที่มีครีมนำไปใช้

ผลาคคลาดเคลื่อน

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จ ลุล่วงไปด้วยดี เพราะได้รับการช่วยเหลือ
ชี้แนะข้อคิดเห็นต่างๆ จากอาจารย์ที่ปรึกษา คือ อ.วราคม เนินน้อย และ
ศจ.โทชิโอะ อิจิมา รวมทั้งผู้ที่มิได้กล่าวถึงในที่นี้ ที่ได้ให้ความช่วยเหลือใน
ด้านอุปกรณ์ สถานที่ และความสะดวกในการจัดทำต้นฉบับ

หากปริญญานิพนธ์นี้มีประโยชน์อะไรอย่างหนึ่ง ขอขอบความดีนี้ให้ผู้มีพระคุณของ
คณะผู้จัดทำที่เป็นกำลังใจและให้คำปรึกษามาโดยตลอด



บรรณานุกรม

- 1.) " คู่มือไอซีพีพีพีพอร์ทและหน่วยความจำ", สำนักพิมพ์ ซีเอ็ดยูเคชั่น . พศ 2529
- 2.) บุญเลิศ เอี่ยมทัศนาศิน กุ๊ววรรณ สมนึก ศิริรัต
"โปรแกรมคอมพิวเตอร์ภาษา C ", สำนักพิมพ์ ซีเอ็ดยูเคชั่น , พศ 2521
- 3.) "ภาษาแอสเซมบลีสำหรับเครื่อง IBM PC " สำนักพิมพ์ ซีเอ็ดยูเคชั่น
- 4.) อีน กุ๊ววรรณ ดร.ชัยยงค์ วงศ์สุวรรณ ดร.ไพศาล สงวนหม่ม
"เทคโนโลยีไมโครคอมพิวเตอร์ 16 บิต ", สำนักพิมพ์ ซีเอ็ดยูเคชั่น , พศ 2531
- 5.) "TURBO C REFERENCE GUIDE VERSION 2.0", BORLAND INTERNATIONAL
- 6.) "TURBO C USER'S GUIDE VERSION 2.0", BORLAND INTERNATIONAL
- 7.) "TURBO DEBUGGER USER'S GUIDE VERSION 2.0", BORLAND INTERNATIONAL
- 8.) "TURBO ASSEMBLER REFERENCE GUIDE VERSION 2.0", BORLAND INTERNATIONAL
- 9.) "INSIDE AUTOCAD " , NEW RAIDERS PUBLISHING THOUSAND OAKS
- 10.) "INSIDE AUTOLISP" , NEW RAIDERS PUBLISHING THOUSAND OAKS
- 11.) JOSEPH EDWARD SHIGLEY, "MECHANICAL ENGINEERING DESIGN ",
MCGRAW-HILL
- 12.) K.S.FU , RCGONZALEZ , "ROBOTICS", MCGRAW-HILL, 1987
- 13.) PHILLIPPE COIFFET , "ROBOT TECHNOLOGY " , PRENTICE-HALL INC.

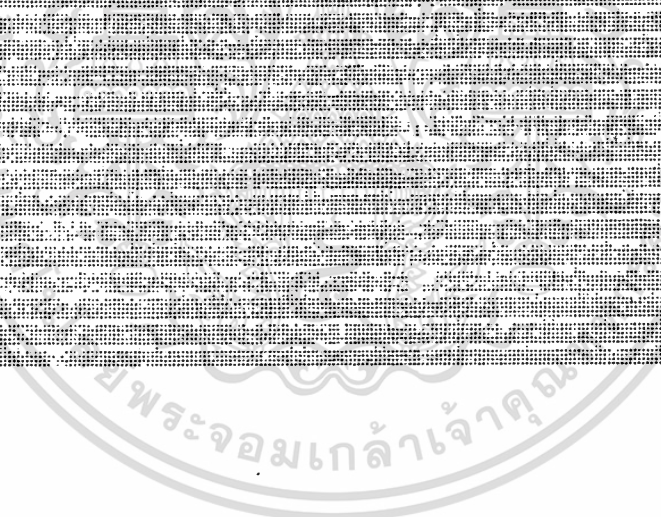
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่อง -

Present By: Ratcharin



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <dos.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <conio.h>
#include <graphics.h>
#include <ctype.h>
#define      PI      3.14285714

double      s (double);
double      c (double);

int         port_data[10], limit, port_add[10] , freq[10] , pulse[10]
deg[10] , direct[10] , mode_1 ,mode_2 ,mode_3, deg_old[10]
delta_deg[10] , ddeg[10] , j ;

char        txt[10] , key ;

float       d[5];

double      nx,ny,nz
sx,sy,sz
ax,ay,az
px,py,pz
ix,iy,iz
kx,ky,kz
g,y,x ;

rossarin()
{
    int      g_driver,g_mode,t,tt ;

    detectgraph(&g_driver,&g_mode);
    initgraph(&g_driver,&g_mode,"");
    setcolor(7);
    setbkcolor(0);
    rectangle(10,10,600,300);
}

initialize()
{
    outportb(0x303,0x30); /* program all 8253 to 16 bits counter */
    outportb(0x303,0x70); /* used only mode 3 and mode 0 */
    outportb(0x303,0xb6);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    outportb(0x307,0x36);
    outportb(0x307,0x70);
    outportb(0x307,0xb6);

    outportb(0x30b,0x36);
    outportb(0x30b,0x70);
    outportb(0x30b,0xb6);

    outportb(0x30f,0x30);
    outportb(0x30f,0x70);
    outportb(0x30f,0xb6);
} /* choose() */

clr_wdw()
{
    int    pos_clr_scr[8] = { 20,200,300,200,300,250,300,200 } ;

    setfillstyle(SOLID_FILL,BLACK);
    bar(20,200,300,250);
} /* clr_wdw() */

turning(direct)
int    direct ;
{
    if( direct == '+' )
        outportb(0x310,0x00);
else    if( direct == '-' )
        outportb(0x310,0x3f);
}/* turning(direct) */

choose()
{
    int    num , i , _x , z , p , j , test , test_1 ;
    char   port_no , port[10] ;

    setfillstyle(SOLID_FILL,BLACK);
    bar(20,125,350,199);
    setfillstyle(SOLID_FILL,BLACK);
    bar(350,120,550,250);
    setcolor(MAGENTA);
    rectangle(20,20,300,120);
    setcolor(LIGHTGREEN);
    outtextxy(10,30,"    DETAILS OF PART LIST :");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

outtextxy(30,50,"A : WRIST      PORT      NO. 0x302 " );
outtextxy(30,60,"B : SHOWLDER PORT      NO. 0x306 " );
outtextxy(30,70,"C : ELBOW      PORT      NO. 0x30a " );
outtextxy(30,80,"D : PITCH      PORT      NO. 0x304 " );
outtextxy(30,90,"E : ROLL      PORT      NO. 0x308 " );

```

```

setcolor(RED);
clr_wdw();
rectangle(20,200,300,250);
outtextxy(30,210," NO. PORTS TO CONTROL ? :<1-5>");
p = 1;

```

```

while( p > 0 )
{
num=getch();

switch(num)      {
case 49:
case 50:
case 51:
case 52:
outtextxy(150,225,&num);
clr_wdw();
rectangle(20,200,300,250);
outtextxy(30,210,"WHICH PORT ? :");
p = -1 ;
break ;

case 53:
port_add[0] = 'A';
port_add[1] = 'B';
port_add[2] = 'C';
port_add[3] = 'D';
port_add[4] = 'E';
outtextxy(150,225,&num);
clr_wdw();
rectangle(20,200,300,250);
outtextxy(30,210,"WHICH PORT ? :");
outtextxy(30,220," FULL CONTROL !!! ");
limit = 5 ;
p = 0 ;
break ;

default:
p = 1 ;
break;
}      /* switch */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

} /* while loop */

while( p != 0 )
{
    limit=(num-48);

    for ( i = 0; i < limit ; i++)
    {
        z=1;

        while( z > 0)
        {
            port_no = getch();
            port_add[i]= toupper(port_no);

            if( (port_add[i] != 'A')&&(port_add[i] != 'B')
                &&(port_add[i] != 'C')&&(port_add[i] != 'D')
                &&(port_add[i] != 'E') )
                z = 2 ;
            else
            {
                switch(limit)
                {
                    case 2:
                    case 3:
                    case 4:

                        for(j=0 ; j < limit ; j++)
                        {
                            if( j != i )
                            {
                                test = ( port_add[i] != port_add[j] ) ? -1 : 1 ;

                                switch(test) {
                                    case 1:
                                        z = 2 ;
                                        j = 10;
                                        break ;

                                    default: /* test != test_1 */
                                        z = -1 ;
                                        break ; }

                                } /* if ( j != i ) */
                                else j = j;
                            }
                        }
                    }
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        } /* for j ) */
        break;

        case 1:
            z = -1 ;
            break;
        default:
            exit;
            break;
    } /* switch */
} /* else */

} /* while ( z > 0 ) */

_x = (i*30) + 30 ;
outtextxy(_x,220,&port_add[i]);

} /* for ( i ) */
p = 0 ;
} /* while ( p < 0 ) */
}

data()
{
    int    i ,ii, speed_data ,deg_data[10]
           index , pos , q , z , qq , s
           last_color,_x , _y , zz;

    char   d_data[10],ss,_freq[10] , _deg[10] , sign ;

    last_color = getcolor();
    setcolor(RED);
    outtextxy(310,30,"SPD./LEV.");
    outtextxy(410,30,"FERQ.");
    outtextxy(475,30,"DEG.");
    outtextxy(550,30,"PULSE.");

    for ( i = 0; i < limit ; i++)
    {
        clr_wdw();
        rectangle(20,200,300,250);
        outtextxy(30,210,"PORT NO. ");

        switch ( port_add[i] )
        {
            case 'A':

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        outtextxy(250,210," : A ");
        _y = 50;
        break;
    case 'B':
        outtextxy(250,210," : B ");
        _y = 60;
        break;
    case 'C':
        outtextxy(250,210," : C ");
        _y = 70;
        break;
    case 'D':
        outtextxy(250,210," : D ");
        _y = 80;
        break;
    case 'E':
        outtextxy(250,210," : E ");
        _y = 90;
        break;
} /* switch */

outtextxy(30,225,"SPEED LEVEL ? < L , M , H > : ");
s = 1 ;
while(s > 0)
{
    q = 1 ;
    while ( q > 0 )
    {
        speed_data=toupper(getch());
        if( (speed_data != 'L')&&(speed_data != 'M')
            &&(speed_data != 'H'))
            q = 1;
        else q = -1;
    } /* while ( q > 0 ) */
    switch ( speed_data )
    {
        case 'L':
            freq[i]=4000;
            break;
        case 'M':
            freq[i]=2400;
            break;
        case 'H':
            freq[i]=1200;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break;
    } /* switch */

    outtextxy(280,225,&speed_data);
    ss = getch();

    if(ss == '\t') {
        setfillstyle(SOLID_FILL,BLACK);
        bar(280,225,290,235);
        s = 1 ;
    } /* if ( ss = '\t' ) */

    else
        s = -1 ;
} /* while (s > 0) */

outtextxy(330,_y,&speed_data);
outtextxy(415,_y,itoa(freq[i],&freq,10) );

outtextxy(30,240,"DEGREE OF ROTATE ? :");
zz = 1;

while( zz > 0) {
    direct[i] = getch();
    if(direct[i] == '+' )
    {
        zz = --1;
        outtextxy(210,240,"+");
        sign = '+';
    }
    else if(direct[i] == '-')
    {
        outtextxy(210,240,"-");
        sign = '-';
        zz = -1;
    } /* if (direct[i] == '-') */
    else
        zz = 1 ;
} /* while (zz > 0) */

    qq = 1 ;
    for( index=0;( qq > 0)&&(index < 9) );index++)
    {
        z = 1;
        while(z > 0)
        {
            deg_data[index]=getch();

            switch(deg_data[index])

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    case '0':
    case '1':
    case '2':
    case '3':
    case '4':
    case '5':
    case '6':
    case '7':
    case '8':
    case '9':
        d_data[index]=deg_data[Index];
        pos=220+(8*Index);
        outtextxy(pos,240,&deg_data[index]);
        z = -1 ;
        qq = 1 ;
        break;
    case '\r':
        z = -1 ;
        qq = -1 ;
        break;
    case '\t':
        setfillstyle(SOLID_FILL,BLACK);
        bar(215,240,299,249);
        for(ii=0;ii<10;ii++)
            d_data[ii] = 0;
        z = -1;
        index = -1 ;
        qq = 1;
        break;
    default:
        z = 1 ;
        qq = 1 ;
        break;
} /* switch */

} /* while (z > 0) */

} /* for loop */

d_data[Index]='\0';
deg[l]=atoi(&d_data);

if( sign == '-' )
    deg[l] = deg[l]*(-1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    outtextxy(475,_y,itoa(deg[i],&_deg[i],10) );
    for(ii=0;ii<10;ii++)
        d_data[ii] = 0;

```

```

} /* FOR LOOP */

```

```

} /* DATA() */

```

```

port()

```

```

{

```

```

    int    i , z ;
    char   _pulse[10] ;

```

```

    clr_wdw();
    settextstyle(2,0,5);
    setcolor(BLUE);
    rectangle(20,200,300,250);
    setcolor(DARKGRAY);
    outtextxy(40,220," RUN PROG. IN COMPLETED ");
    setcolor(RED);
    settextstyle(0,0,1);

```

```

for ( i = 0; i < limit ; i++)

```

```

{

```

```

    switch ( port_add[i] )

```

```

    {

```

```

        case 'A': /* ratio = 16.0 */
            pulse[i] = abs( (ddeg[i]*16.0)/1.8 ) ;
            turning(direct[i]);
            outtextxy(555,50,itoa(pulse[i],&_pulse[i],10) );
            outportb(0x302,freq[i]);
            outportb(0x302,freq[i] >> 8);
            outportb(0x301,pulse[i]);
            outportb(0x301,pulse[i] >> 8 );
            outportb(0x314,0x00);
            break;
        case 'B': /* ratio = 53.29 */
            pulse[i] = abs( (ddeg[i]*53.29)/1.8 );
            /* turning(direct); */
            outtextxy(555,60,itoa(pulse[i],&_pulse[i],10) );
            outportb(0x306,freq[i]);
            outportb(0x306,freq[i] >> 8);
            outportb(0x305,pulse[i]);

```

```

        outportb(0x305,pulse[i] >> 8 );
        outportb(0x314,0x00);
        break;
    case 'C': /* ratio = 17.6 */
        pulse[i] = abs( (ddeg[i]*17.6)/1.8 );
        turning(direct);
        outtextxy(555,70,itoa(pulse[i],&_pulse[i],10) );
        outportb(0x30a,freq[i]);
        outportb(0x30a,freq[i] >> 8);
        outportb(0x309,pulse[i]);
        outportb(0x309,pulse[i] >> 8 );
        outportb(0x314,0x00);
        break;
    case 'D': /* ratio = 12.0 */
        pulse[i] = abs( (ddeg[i]*12.0)/0.18 );
        turning(direct);
        outtextxy(555,80,itoa(pulse[i],&_pulse[i],10) );
        outportb(0x304, freq[i]);
        outportb(0x304,freq[i] >> 8);
        outportb(0x300,pulse[i]);
        outportb(0x300,pulse[i] >> 8 );
        outportb(0x314,0x00);
        break;
    case 'E': /* ratio = 1.00 */
        pulse[i] = abs( (ddeg[i])/0.18 );
        turning(direct);
        outtextxy(555,90,itoa(pulse[i],&_pulse[i],10) );
        outportb(0x308, freq[i]);
        outportb(0x308,freq[i] >> .8);
        outportb(0x30c,pulse[i]);
        outportb(0x30c,pulse[i] >> 8 );
        outportb(0x314,0x00);
        break;
} /* switch */
} /* for loop */
for(i=0;i<5;i++) {
    pulse[i] = 0 ;
    direct[i] = 0 ;
}
} /* port() */

```

```

double c(a)
double a;
{
    double b;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
b=cos((double) a);  
return(b);
```

```
} /* c(a) */
```

```
double s(a)
```

```
double a;
```

```
{  
    double b;
```

```
    b=sin((double) a);  
    return(b);
```

```
} /* s(a) */
```

```
Input()
```

```
{
```

```
    int i;
```

```
    for ( i = 0; i < limit ; i++)
```

```
    {
```

```
        switch ( port_add[i] )
```

```
        {
```

```
            case 'A':  
                d[0] = deg[i];  
                break;
```

```
            case 'B':  
                d[1] = deg[i];  
                break;
```

```
            case 'C':  
                d[2] = deg[i];  
                break;
```

```
            case 'D':  
                d[3] = deg[i];  
                break;
```

```
            case 'E':  
                d[4] = deg[i];  
                break;
```

```
        } /* swiych */
```

```
    }
```

```
    for(i=0;i<5;i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
d[l]=( PI*d[l] ) / 180;
```

```
} /* INPUT */
```

```
forward()
```

```
{
```

```
ix = ( ( c(d[0])*c(d[1]) )*( c(d[2])*c(d[3]) - s(d[2])*s(d[3]) ) )  
      - ( ( c(d[0])*c(d[1]) )*( s(d[2])*c(d[3]) + c(d[2])*s(d[3]) ) ) );
```

```
iy = ( ( s(d[0])*c(d[1]) )*( c(d[2])*c(d[3]) - s(d[2])*s(d[3]) ) )  
      - ( ( s(d[0])*s(d[1]) )*( s(d[2])*c(d[3]) + c(d[2])*s(d[3]) ) ) );
```

```
iz = ( (-s(d[1]) )*( c(d[2])*c(d[3]) - s(d[2])*s(d[3]) ) )  
      - ( ( c(d[1]) )*( s(d[2])*c(d[3]) + c(d[2])*s(d[3]) ) ) );
```

```
kx = ( ( c(d[0])*c(d[1]) )*( c(d[2])*s(d[3]) + s(d[2])*c(d[3]) ) )  
      - ( ( c(d[0])*s(d[1]) )*( s(d[2])*s(d[3]) - c(d[2])*c(d[3]) ) ) );
```

```
ky = ( ( s(d[0])*c(d[1]) )*( c(d[2])*s(d[3]) + s(d[2])*c(d[3]) ) )  
      - ( ( s(d[0])*s(d[1]) )*( s(d[2])*s(d[3]) - c(d[2])*c(d[3]) ) ) );
```

```
kz = ( (-s(d[1]) )*( c(d[2])*s(d[3]) + s(d[2])*c(d[3]) ) )  
      - ( ( c(d[1]) )*( s(d[2])*s(d[3]) - c(d[2])*c(d[3]) ) ) );
```

```
nx = ( c(d[4])*ix ) - ( s(d[4])*s(d[0]) );
```

```
ny = ( c(d[4])*iy ) + ( s(d[4])*c(d[0]) );
```

```
nz = ( c(d[4])*iz );
```

```
sx = (-s(d[4])*ix ) - ( s(d[0])*c(d[4]) );
```

```
sy = (-s(d[4])*iy ) + ( c(d[4])*c(d[0]) );
```

```
sz = ( s(d[4])*iz );
```

```
ax = kx;
```

```
ay = ky;
```

```
az = kz;
```

```
x = c(d[0]);
```

```
px =
```

```
(20*kx)+(c(d[0])*c(d[1])*130*c(d[2]))-(c(d[0])*s(d[1])*130*s(d[2]) ) + (
```

```
120*c(d[0])*c(d[1]) ) + ( 100*c(d[0]) );
```

```
py = (20*ky) + ( s(d[0])*c(d[1])*130*c(d[2])
```

```
)-(s(d[0])*s(d[1])*130*s(d[2]) ) + ( 120*c(d[1])*s(d[0]) ) + ( 100*s(d[0]) );
```

```
pz = (20*kz) + (-s(d[1])*130*c(d[2])
```

```
) - ( c(d[1])*130*s(d[2])
```

```
) - ( 120*s(d[1]) ) + ( 40
```

```
) );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
} /* FORWARD */
```

```
out_put()
```

```
{  
    char    _px[10] , _py[10] , _pz[10] ;  
  
    setfillstyle(SOLID_FILL,BLACK);  
    bar(350,120,550,250);  
    setcolor(YELLOW);  
    rectangle(350,120,550,250);  
    setcolor(GREEN);  
    outtextxy(380,150," SHOW THE POSITIONS ");  
    outtextxy(400,180,"X = ");  
    outtextxy(450,180,ltoa(px,&_px,10) );  
    outtextxy(400,200,"Y = ");  
    outtextxy(450,200,ltoa(py,&_py,10) );  
    outtextxy(400,220,"Z = ");  
    outtextxy(450,220,ltoa(pz,&_pz,10) );  
}
```

```
mode_choose()
```

```
{  
    char    mode ;  
    int     mode_first , i ;  
  
    setcolor(LIGHTMAGENTA);  
    outtextxy(20,130," SELECT ONE MODE !!! ");  
    outtextxy(20,150," (1)...RUN PROG. AGAIN WITH SAME PORT ");  
    outtextxy(20,170," (2)...RUN PROG. AGAIN & CLEAR ALL DATA");  
    outtextxy(20,190," (3)...CAN'T BAER ANY MORE FURTHER");  
    mode_first = 1 ;  
  
    while( mode_first > 0 ) {  
        mode = getch();  
  
        if( mode == '1' )  
        {  
            for(i=0;i<limit;i++)  
            {  
                deg_old[i] = deg[i] ;  
                delta_deg[i] = deg_old[i] ;  
            }  
            setfillstyle(SOLID_FILL,BLACK);  
            bar(305,40,599,115);  
            mode_2 = 1 ;  
        }  
    }  
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        mode_1 = 1 ;
        mode_first = -1 ;
    }

else    if( mode == '2' )
    {
        setfillstyle(SOLID_FILL,BLACK);
        bar(305,40,599,115);
        mode_3 = 1 ;
        mode_2 = -1 ;
        mode_1 = -1 ;
        mode_first = -1 ;
    }
else    if( mode == '3' )
    {
        mode_3 = -1 ;
        mode_2 = -1;
        mode_1 = -1 ;
        mode_first = -1 ;
    }
else
        mode_first = 1 ;
        } /* while (mode_first > 0) */
} /* mode_choose */

/*-----*/
/

main()
{
    int    z , k[10];

    clrscr();
    rossarin();
    initialize();
    mode_3 = 1 ;
    mode_1 = 0 ;

    while( mode_3 > 0 )
    {
        choose();
        mode_2 = 1 ;
        while( mode_2 > 0 )
        {
            data(port_data[10],limit);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Input();
forward();
out_put();

for(j=0;j<5;j++)
    direct[j] = 48 ;

for(j=0;j<5;j++)
{
    k[j] = (deg[j] - delta_deg[j]) ;

    if( k[j] > 0 )
        direct[j] = 43 ;
    else
        direct[j] = 45 ;
} /* for */
if( mode_1 > 0 )
{
    for(j=0;j<5;j++)
        ddeg[j] = deg[j]-delta_deg[j];
} /* if mode_1 > 0 */
if( mode_1 == 0 )
{
    for(j=0;j<5;j++)
        ddeg[j] = deg[j] ;
}

port();
mode_choose();
} /* mode_2 > 0 */
} /* mode_3 > 0 */

getch();
closegraph();
cirscl();

} /* MAIN */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้