ระบบควบคุมอัตโนมัติขนาดเล็ก

MICRO CIM

โดย

1.นายกันตพงษ์ คงหมุน

2.นายนวพล มนัสบุญเพิ่มพูล

3.นายปราโมทย์ อริยาดิเรก

อาจารย์ที่ปรึกษา

อาจารย์เทพจิตร์ เซยโภคา

วัน เดือน ปี 11 ต.ค. 2541

เลขทะเบียน 038877

เลขเรียกหนังสือ T 10121 ก038ป 1

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

038877

ปีการศึกษา 2540

ระบบควบคุมอัตโนมัติขนาดเล็ก

โดย

1.นายกันตพงษ์ คงหมุน รหัสประจำตัว 37014012

2.นายนวพล มนัสบุญเพิ่มพูล รหัสประจำตัว 37014194

3.นายปราโมทย์ อริยาดิเรก รหัสประจำตัว37014243

อาจารย์ที่ปรึกษา

(อาจารย์เทพจิตร์ เชยโภคา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมอัตโนมัติขนาดเล็ก

นายกันตพงษ์ คงหมุน
นายนวพล มนัสบุญเพิ่มพูล
นายปราโมทย์ อริยาดิเรก

อาจารย์ที่ปรึกษา อาจารย์เทพจิตร์ เชยโภคา
ปีการศึกษา 2540

บทคัดย่อ

โครงงานนี้เป็นการสร้างโปรแกรมเพื่อการควบคุมระบบจำลองที่ประกอบด้วย เครื่องกลึง CNC และ แขนกล ให้ทำงานอัตโนมัติ โดยใช้ภาษา Visual BASIC ผู้ควบคุมสามารถใช้โปรแกรมที่เขียนขึ้นอย่างง่ายดาย โดยไม่จำเป็นต้องมีความรู้ทางด้านคำสั่งที่ใช้สั่งงาน CNC มากนัก โปรแกรมจะสั่งให้แขนกลหยิบชิ้นงานที่ต้องการกลึง แล้วนำมาใส่เครื่อง CNC จากนั้น CNC จะทำการกลึงเป็นชิ้นงานตามที่เลือกไว้จากโปรแกรม เมื่อเสร็จงานแขนกลจะนำชิ้นงานที่กลึงเรียบร้อยแล้ว ไปเก็บไว้ ณ จุดเก็บชิ้นงานอย่างเป็นระเบียบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MICRO CIM

Kantapong Kongmoon

Navapol Manusboonpempool

Pramote Ariyadirak

Advisor Teppajit Choeypoca

Academic year 1997

ABSTRACT

This project is designed to creat the program for controlling simulated automatic system which compose of CNC - machine and robot arm by using Visual BASIC computer programming language The operator who lack of knowledge about CNC's Code can use this program easily The robot's gripper grip workpiece to CNC - machine and then lathe to the desired figure Finally robot arm works again to bring the complete workpiece to the store perfectly

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	I - II
สารบัญรูป	III
สารบัญตาราง	IV
บทที่ 1	
บทนำ	1
บทที่ 2	
ทฤษฎีและหลักการ	
1.ทฤษฎีเบื้องต้นเกี่ยวกับเครื่องจักรกลซีเอ็นซี	3
2.หุ่นยนต์	21
3.Visual BASIC for Windows(VB/Win)	36
บทที่ 3	
การ INTERFACE	
PARALLEL INTERFACE	56
SERIAL INTERFACE	62
บทที่ 4	
สรุปและข้อเสนอแนะ	66
กิตติกรรมประกาศ	67
เอกสารอ้างอิง	68
ภาคผนวก	69
ภาคผนวก ก	70
ภาคผนวก ข	98

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ 1.1 เครื่องกลึง NC แบบ 2 แกนและ 3 แกน 6

รูปที่ 1.2 องค์ประกอบของเครื่องจักรกล CNC 6

รูปที่ 1.3 เครื่องกลึง CNC 7

รูปที่ 1.4 เครื่องกลึง CNC แบบ 2 แกน 8

รูปที่ 1.5 การเคลื่อนที่ตัดเฉือนของเครื่องมือตัด 9

รูปที่ 1.6 DIAGRAM ระบบขับป้อน 9

รูปที่ 1.7 ส่วนประกอบของ Motor กระแสตรง 10

รูปที่ 1.8 เปรียบเทียบลักษณะสร้างและขนาดของ Motor กระแสตรง
กับ Motor กระแสสลับแบบ 3 เฟส 11

รูปที่ 1.9 การขับป้อนของโต๊ะงาน 11

รูปที่ 1.10 ลักษณะสร้างภายในของชุดบอลล์สกูร 12

รูปที่ 1.11 การวัดตำแหน่งโดยตรง 13

รูปที่ 1.12 การวัดตำแหน่งทางอ้อม 13

รูปที่ 1.13 การวัดตำแหน่งแบบสมบูรณ์ 14

รูปที่ 1.14 การวัดตำแหน่งแบบต่อเนื่อง 15

รูปที่ 1.15 เพลางานของเครื่องกลึง 15

รูปที่ 1.16 หัวจับงานกลึง 15

รูปที่ 1.17 ชุดยันศูนย์ท้ายแท่น และ แท่นประคองงานกลึง 16

รูปที่ 1.18 องค์ประกอบของเครื่องมือที่ใช้ในงานกลึง 16

รูปที่ 1.19 Insert แบบถอดเปลี่ยนได้รูปทรงต่าง ๆ 16

รูปที่ 1.20 ระบบ NC 17

รูปที่ 1.21 ระบบ CNC 17

รูปที่ 1.22 หน้าที่การทำงานของเครื่องจักร 20

รูปที่ 2.1 หุ่นยนต์อุตสาหกรรม 21

รูปที่ 2.2 ความสัมพันธ์ระหว่างค่าใช้จ่ายกับปริมาณการผลิต 22

รูปที่ 2.3 หุ่นยนต์พิกัดคาร์ทีเซียน 24

รูปที่ 2.4 หุ่นยนต์พิกัดทรงกระบอก 25

รูปที่ 2.5 หุ่นยนต์พิกัดทรงกลม 25

รูปที่ 2.6 หุ่นยนต์ SCARA 26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ 2.7 หุ่นยนต์ข้อต่อหมุน 26

รูปที่ 2.8 ระยะที่เอื้อมถึงและสโตรคของหุ่นยนต์ทรงกระบอก 29

รูปที่ 2.9 Yaw,Pitch และ Row ของ Tool 30

รูปที่ 2.10 ลำดับการหมุน 30

รูปที่ 2.11 ระบบ YPR และ RPY 32

รูปที่ 2.12 ตำแหน่ง Tool ที่อยู่ติดกัน 32

รูปที่ 2.13 ความละเอียดแนวนอนของหุ่นยนต์ทรงกระบอก 33

รูปที่ 2.14 เปรียบเทียบ ACCURACY และ REPEATABILITY 35

รูปที่ 3.1 แสดงไดอะลอกซ์ บอกซ์ Open Picture 40

รูปที่ 3.2 แสดงคอนโทรล Picture Box ในขณะออกแบบ 44

รูปที่ 3.3 แสดงผลของคุณสมบัติ AutoShowChildren ที่มีต่อการโหลด ของฟอร์มลูก MDI 52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง 1.1 เปรียบเทียบการทำงานระหว่างเครื่องจักรกลทั่ว ๆ ไป กับ เครื่องจักรกลเอ็นซี 3
ตาราง 2.1 ชนิดของข้อต่อหุ่นยนต์ 23
ตาราง 2.2 ขอบเขตการทำงานของหุ่นยนต์ที่ขึ้นอยู่กับแกนหลัก 24
ตาราง 2.3 ชนิดของการควบคุมการเคลื่อนที่ของหุ่นยนต์ 27
ตาราง 2.4 ลักษณะเฉพาะของหุ่นยนต์ 27
ตาราง 2.5 แกนของหุ่นยนต์ 28
ตาราง 2.6 การเคลื่อนที่แบบ Yaw, Pitch และ Roll 31
ตาราง 2.7 ความละเอียดในแนวนอนและแนวตั้ง 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในโลกยุคโลกาภิวัตน์ ดังเช่นในปัจจุบัน ได้เกิดการพัฒนาในทุกๆแขนงงานโดยเฉพาะงานที่เกี่ยวข้องกับเทคโนโลยี อาทิ เทคโนโลยีการผลิต I.C,เทคโนโลยีการติดต่อสื่อสาร และที่สำคัญอีกอย่างหนึ่งก็คือ เทคโนโลยีที่ถูกนำมาใช้ในโรงงานอุตสาหกรรม ที่มีการพัฒนาจากการใช้แรงงานคนมาสู่การทำงานโดยเครื่องจักรแบบอัตโนมัติซึ่งจะทำให้เกิดประสิทธิภาพสูงสุดในการทำงานดังนี้

- ลดต้นทุนการผลิต เพราะมีค่าใช้จ่ายในระยะยาวที่ต่ำสามารถทำงานได้ตลอดเวลา ไม่เกิดความเมื่อยล้า ลดความสิ้นเปลืองในการใช้วัตถุดิบ หรือทำให้เกิดผลเสียน้อย

- ช่วยเพิ่มผลผลิต เพราะทำงานได้รวดเร็วกว่ามนุษย์

- ช่วยเพิ่มคุณภาพผลผลิต เพราะมีความแม่นยำในการทำงานสูง

- สะดวกในการปฏิบัติงานเพราะสามารถทำงานภายในห้องควบคุม (Control Room) และเพิ่มความปลอดภัยในงานที่ทำอีกด้วย

ด้วยเหตุนี้จึงมีการนำเอาระบบควบคุมอัตโนมัติมาใช้ในโรงงานอุตสาหกรรมอย่างแพร่หลายในรูปแบบต่างๆกัน ตัวอย่างเช่น

-ใช้ในงานที่เกี่ยวกับการลำเลียงชิ้นงานและการประกอบ เช่น การประกอบรถยนต์,การผลิตมอเตอร์แบบเซอร์โว,การส่งชิ้นงานออกจากเครื่องจักรกล เป็นต้น

-ใช้ในงานผลิตชิ้นงาน เช่น การกลึง,การกัด,การเจาะ

-ใช้ในงานทางด้านส่วนผสมของสารเคมี เช่น การควบคุมอุณหภูมิ,การไหล,ระดับ

สำหรับประเทศที่มีค่าครองชีพสูงย่อมทำให้ค่าจ้างแรงงานสูงตามไปด้วย นำไปสู่ต้นทุนการผลิตที่สูงขึ้น วิธีแก้ปัญหาในทางหนึ่งก็คือ การนำระบบควบคุมอัตโนมัติมาใช้ ดังจะเห็นได้จากโรงงานอุตสาหกรรมในประเทศญี่ปุ่น,สหรัฐ ที่มีการนำหุ่นยนต์มาใช้ในการผลิตเป็นจำนวนมาก สำหรับประเทศไทยได้มีการนำระบบควบคุมอัตโนมัติเข้ามาใช้มากขึ้นเรื่อยๆ เนื่องจากค่าจ้างแรงงานในประเทศสูงขึ้น ทำให้ต้นทุนการผลิตสูงขึ้น ประกอบกับตลาดแรงงานเริ่มไม่เพียงพอ นอกจากนี้ยังมีการกำหนดมาตรฐานสินค้าสากล หรือ ที่เรียกว่ามาตรฐาน ISO ขึ้นทำให้เกิดความตื่นตัวในการพัฒนาสินค้าให้มีคุณภาพตามที่กำหนดเพื่อให้สามารถส่งสินค้าไปแข่งขันกับตลาดต่างประเทศได้ และในขณะเดียวกันที่ประเทศอุตสาหกรรมพัฒนาแล้วกำลังที่จะย้ายฐานการผลิตและออกไปสู่ประเทศที่เริ่มการพัฒนาอุตสาหกรรม เช่น ประเทศไทย, มาเลเซีย,อินโดนีเซีย เป็นต้น การย้ายฐานการผลิตมายังประเทศเหล่านี้ ส่วนมากจะต้องนำเทคโนโลยีการผลิตที่สูงมาด้วยนั่นคือมีระบบอัตโนมัติและหุ่นยนต์อุตสาหกรรมรวมอยู่ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานนี้จึงเป็นการจำลองระบบควบคุมแบบอัตโนมัติที่สามารถนำไปประยุกต์ใช้ในโรงงานอุตสาหกรรมส่วนใหญ่ได้ กล่าวคือ ระบบจะประกอบด้วย เครื่องกลึงแบบ CNC , แขนกล (Robot Arm) โดยมีคอมพิวเตอร์เป็นตัวควบคุมให้ทำงานโดยอัตโนมัติเป็นสื่อกลางการแลกเปลี่ยนข้อมูล ระหว่างเครื่องกลึง CNC และ แขนกล อีกทั้งยังเป็นตัวที่ใช้ติดต่อระหว่างผู้ใช้งานกับเครื่องจักรอีกด้วย

ลักษณะของโครงงานนี้ จะมีภาพรวมวิธีการทำงานที่คล้ายคลึงกับระบบควบคุมที่ใช้จริงในโรงงานอุตสาหกรรมโดยเครื่องจักร เช่น เครื่อง CNC , แขนกล จะติดตั้งอยู่ในส่วนของ Field ส่วนตัวควบคุม เช่น คอมพิวเตอร์ จะอยู่ในส่วนของ Control Room (ห้องควบคุม) และผู้ปฏิบัติงาน (Operator) จะเป็นผู้สั่งงานผ่านทางโปรแกรมที่กำหนดไว้แล้ว โดย Superviser แล้วส่งไปยังชุดเครื่องจักรให้ทำงานตามต้องการได้ ดังนั้นโปรแกรมที่สร้างขึ้นในโครงงานนี้ จึงเป็นลักษณะของการกำหนดชิ้นงานไว้ในโปรแกรมเรียบร้อยแล้ว ผู้ปฏิบัติงานเพียงแค่เลือกรูปชิ้นงานที่ต้องการ จากนั้นเครื่อง CNC และ แขนกล ก็จะทำตามโปรแกรมที่เขียนไว้แล้วโดยอัตโนมัติ และทำให้ได้ชิ้นงานตามที่ต้องการ โดยที่ผู้ปฏิบัติงานไม่จำเป็นต้องรู้วิธีการใช้งานของเครื่อง CNC เช่น ชุดคำสั่ง G - code,M - code ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

1.ทฤษฎีเบื้องต้นเกี่ยวกับเครื่องจักรกลซีเอ็นซี

1.1 พัฒนาการของเครื่องจักรกลเอ็นซีและซีเอ็นซี

1.1.1 ความหมายของเอ็นซีและซีเอ็นซี

เอ็นซี (NC) ย่อมาจากคำว่า Numerical Control หมายถึง การควบคุมเครื่องจักรกลด้วยระบบตัวเลขและตัวอักษร กล่าวคือ การเคลื่อนที่ต่าง ๆ ตลอดจนการทำงานอื่น ๆ ของเครื่องจักรกลจะถูกควบคุมได้โดยรหัสคำสั่งที่ประกอบด้วย ตัวเลข ตัวอักษรและสัญลักษณ์อื่น ๆ ซึ่งจะถูกแปลงเป็นคลื่นสัญญาณ (Pulse) ของกระแสไฟฟ้าหรือสัญญาณออกอื่น ๆ ที่จะไปกระตุ้นมอเตอร์หรืออุปกรอื่น ๆ เพื่อทำให้เครื่องจักรกลทำงานตามขั้นตอนที่ต้องการ

ซีเอ็นซี (CNC) ย่อมาจากคำว่า Computerized Numerical Control ระบบควบคุมเอ็นซีแบบนี้จะมีคอมพิวเตอร์ที่มีความสามารถสูงเพิ่มเข้าไปภายในระบบ ทำให้สามารถจัดการกับข้อมูลที่ป้อนเข้าไปในระบบเอ็นซี และประมวลผลข้อมูลเพื่อนำผลลัพธ์ที่ได้ไปควบคุมการทำงานของเครื่องจักรกล

1.1.2 ความแตกต่างระหว่างเครื่องจักรกลเอ็นซีกับเครื่องจักรกลทั่วไป

ความแตกต่างในการใช้เครื่องจักรกลเอ็นซี เมื่อเปรียบเทียบกับเครื่องจักรกลที่ใช้ทั่วไปก็คือ การตัดสินใจในการกำหนดขั้นตอนการทำงานต่าง ๆ จะกระทำเพียงครั้งเดียว กล่าวคือจะกระทำในขั้นตอนการวางแผนและสร้างโปรแกรมสำหรับควบคุมเครื่องจักรกลเท่านั้น ต่อจากนั้น โปรแกรมก็จะถูกนำไปใช้ในการควบคุมการทำงานของเครื่องจักรกล สำหรับการผลิตชิ้นงานที่ต้องการ โดยสามารถทำการผลิตซ้ำ ๆ กันกี่ครั้งก็ได้ตามต้องการ

ตาราง1.1 เปรียบเทียบการทำงานระหว่างเครื่องจักรกลทั่ว ๆ ไป กับเครื่องจักรกลเอ็นซี

	รายละเอียด	เครื่องจักรกลทั่วไป	เครื่องจักรกลเอ็นซี
1 .	การป้อนโปรแกรม	ไม่มี	มี
2.	การจับยึดชิ้นงาน	มือ	มือ
3.	การจับยึดเครื่องมือตัด	มือ	มือหรือชุดควบคุม
4.	การตั้งจุดอ้างอิง	มือ	มือ
5.	การตั้งความเร็วรอบ	มือ	ระบบควบคุม
6.	การเลื่อนแท่นเลื่อน	มือหมุน	ระบบควบคุม
7.	การเปรียบเทียบระยะ	สายตา	ระบบควบคุม
8.	การตรวจสอบขนาด	เครื่องมือวัด	ใช้เวลาน้อยกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1.3 ความแตกต่างระหว่างระบบเอ็นซีกับระบบซีเอ็นซี

ระบบซีเอ็นซีเป็นระบบที่พัฒนาต่อเนื่องมาจากระบบเอ็นซี ดังนั้น ความแตกต่างระหว่างระบบเอ็นซีกับระบบซีเอ็นซีก็จะอยู่ที่ความสามารถในการทำงานต่าง ๆ จะเพิ่มมากขึ้นเมื่อเปรียบเทียบกับเครื่องจักรกลเอ็นซีดังนี้

1. การแสดงภาพจำลอง (Simulation) การทำงานตามโปรแกรมที่ป้อนเข้าในระบบทางจอภาพ

2. ความจุของหน่วยความจำเพิ่มมากขึ้น สามารถเก็บข้อมูลโปรแกรมได้มาก

3. การแก้ไขและลบโปรแกรมสามารถกระทำได้ที่เครื่องจักรโดยตรง

4. สามารถส่งข้อมูลไปเก็บไว้ในหน่วยความจำภายนอกได้

5. ระบบความปลอดภัยเพิ่มมากขึ้น

6. มีการชดเชยความผิดพลาดที่เกิดจากการวัดและการส่งกำลัง

7. มีโปรแกรมสำเร็จสำหรับการคำนวนค่าต่าง ๆ เช่น ความเร็วรอบ อัตราป้อน เป็นต้น

1.1.4 ข้อดีและข้อเสียของเครื่องจักรกลซีเอ็นซี

ข้อดีของเครื่องจักรกล CNC เมื่อเปรียบเทียบกับเครื่องจักรกลอัตโนมัติประเภทอื่น ๆ พอจะสรุปดังนี้

1. มีความยืดหยุ่นสูง การเปลี่ยนงานใหม่จะแก้ไขหรือเปลี่ยนแปลงเฉพาะโปรแกรมเท่านั้น

2. ความเที่ยงตรง (Accuracy) จะอยู่ระดับเดียวกันตลอดช่วงความเร็วรอบและอัตราป้อนที่ใช้ในการผลิต

3. ใช้เวลาในการผลิตสั้นกว่า

4. สามารถใช้ผลิตชิ้นงานที่มีรูปทรงซับซ้อนได้ง่าย

5. การปรับตั้งเครื่องจักรกระทำได้ง่าย ใช้เวลาน้อยกว่าการผลิตด้วยวิธีอื่น ๆ

6. หลีกเลี่ยงความจำเป็นที่ต้องใช้ช่างควบคุมที่มีทักษะและประสบการณ์สูง

7. ช่างควบคุมเครื่องมีเวลาว่างจากการควบคุมเครื่อง สามารถที่จะจัดเตรียมงานอื่น ๆ ไว้ล่วงหน้าได้

8. การตรวจสอบคุณภาพไม่จำเป็นต้องกระทำทุกขั้นตอนและทุกชิ้น

ส่วนข้อเสียของเครื่องจักรกล CNC มีดังนี้

1. ราคาของเตรื่องจักรค่อนข้างสูง

2. การบำรุงรักษาค่อนข้างซับซ้อน

3.การซ่อมบำรุงต้องใช้ช่างที่มีประสบการณ์สูงและผ่านการฝึกอบรมมาโดยเฉพาะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. จำเป็นต้องใช้ช่างเขียนโปรแกรม (Part Programmer) ที่มีทักษะสูงและฝึกอบรมมาโดยเฉพาะ

5. ราคาของเครื่องมือต่าง ๆ ที่ใช้ในกระบวนการตัดเฉือน เช่น แกนเพลายึดใบมีด มีดกลึงแบบใช้อันเสิร์ต เป็นต้น มีราคาสูง

1.2. <u>เครื่องจักรกลเอ็นซีและซีเอ็นซี</u>

1.2.1 <u>การทำงานของเครื่องจักรกลเอ็นซี</u>

หลักการทำงานของเครื่องจักรกลเอ็นซีหรือซีเอ็นซี จะคล้ายคลึงกับเครื่องจักรกลทั่วไป กล่าวคือ โดยพื้นฐานเบื้องต้นแล้วเครื่องจักรกลเอ็นซีก็จะทำการผลิตชิ้นงานเหมือนกับเครื่องกลึงทั่วไป เพียงแต่ว่าระบบควบคุมเอ็นซีของเครื่องจะทำงานในขั้นตอนต่าง ๆแทนช่างควบคุมเครื่อง อย่างไรก็ตาม ก่อนที่เครื่องจักรกลเอ็นซีจะสามารถทำงานได้นั้น ระบบควบคุมของเครื่องจะต้องได้รับการบอกกกล่าวเสียก่อนว่าจะให้ทำอะไร และจะต้องกล่าวเป็นภาษาที่ระบบควบคุมสามารถเข้าใจได้ นั่นคือ จะต้องป้อนโปรแกรมเข้าไปในระบบควบคุมของเครื่องผ่านแป้นพิมพ์ หรือเทปแม่เหล็ก ก็ได้

เมื่อระบบควบคุมอ่านโปรแกรมที่ป้อนเข้าไปแล้ว ก็จะนำไปควบคุมให้เครื่องจักรกลทำงาน แต่เนื่องจากเครื่องจักรกลเอ็นซีไม่มีมือสำหรับหมุนมือหมุนให้แท่นเลื่อนเคลื่อนที่ได้ดังนั้น แท่นเลื่อนต่าง ๆ จะต้องมีมอเตอร์ป้อน (feed motor) ประกอบอยู่ เช่น เครื่องกลึงซีเอ็นซีจะมีการเคลื่อนที่ 2 แนวแกน ก็จะมีมอเตอร์ป้อน 2 ตัว

เมื่อระบบควบคุมอ่านโปรแกรมแล้ว ก็จะเปลี่ยนรหัสโปรแกรมนั้น ให้เป็นสัญญาณทางไฟฟ้า เพื่อไปควบคุมให้มอเตอร์ทำงาน แต่เนื่องจากสัญญาณที่ออกจากระบบควบคุมนี้มีกำลังน้อย ไม่สามารถไปหมุนขับให้มอเตอร์ทำงานได้ ดังนั้น จึงต้องส่งสัญญาณนี้เข้าไปในภาคขยายสัญญาณของระบบขับ (Drive Amplified) และส่งต่อไปยังมอเตอร์ป้อนของแนวแกนที่ต้องการเคลื่อนที่

ความเร็วและระยะทางการเคลื่อนที่ของแท่นเลื่อน จะต้องกำหนดให้ระบบควบคุมรู้ ช่างควบคุมเครื่องอาศัยสายตามองดูตำแหน่งของคมตัดกับชิ้นงาน ก็จะรู้ว่าจะต้องเลื่อนแท่นเลื่อนไปอีกเป็นระยะทางเท่าใด แต่ระบบควบคุมเอ็นซีมองไม่ได้ ดังนั้น จึงต้องออกแบบอุปกรณ์หรือเครื่องมือที่สามารถจะบอกตำแหน่งของแท่นเลื่อนให้ระบบควบคุมรู้ได้ อุปกรณ์ชุดนี้เรียกว่า **<u>ระบบวัดขนาด (Measuring System)</u>** ซึ่งประกอบด้วยสเกลแนวตรง (Linear Scale)

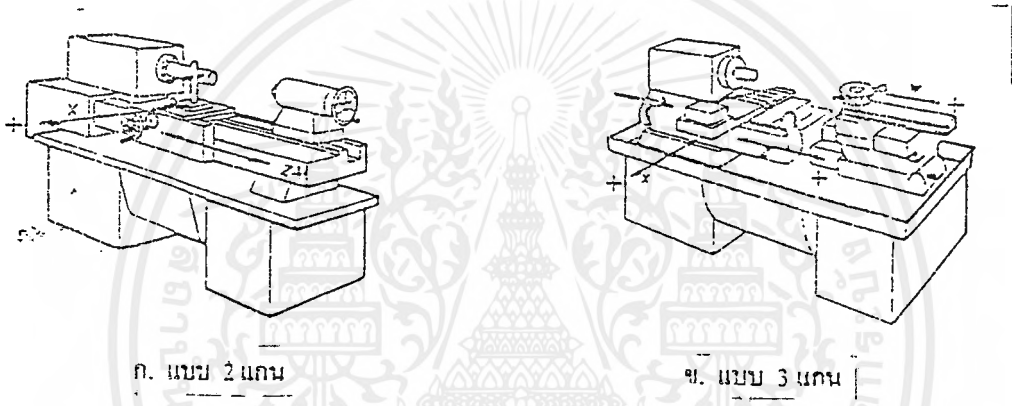
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีจำนวนเท่ากับจำนวนแนวแกนในการเคลื่อนที่ของเครื่องจักรกล ทำหน้าที่ส่งสัญญาณไฟฟ้าที่สัมพันธ์กับระยะทางที่แท่นเลื่อนเคลื่อนที่กลับไปยังระบบควบคุม ทำให้ระบบควบคุมรู้ว่าแท่นเลื่อนเคลื่อนที่ไปเป็นระยะทางเท่าใดแล้ว

จากหลักการควบคุมการทำงานดังกล่าว ทำให้เครื่องจักรกลซีเอ็นซีสามารถผลิตชิ้นงานให้มีรูปทรงและขนาดที่ต้องการได้ จากลักษณะสร้างและการทำงานที่เหนือกว่าเครื่องจักรกลทั่วไป ทำให้เครื่องจักรกลเอ็นซีและซีเอ็นซี เป็นปัจจัยหนึ่งที่มีความสำคัญมากในอุตสาหกรรมอัตโนมัติ และมีปริมาณความต้องการใช้เพิ่มมากขึ้นเรื่อย ๆ

1.2.2 เครื่องกลึงเอ็นซี (NC Latch Machines)

เครื่องกลึงเอ็นซีส่วนใหญ่มีแนวการเคลื่อนที่เพียง 2 แนวแกนเท่านั้น ลักษณะการออกแบบส่วนใหญ่จะขึ้นอยู่กับชนิดของเครื่องกลึง ดังแสดงในรูป 1.2



ก. แบบ 2 แกน ข. แบบ 3 แกน

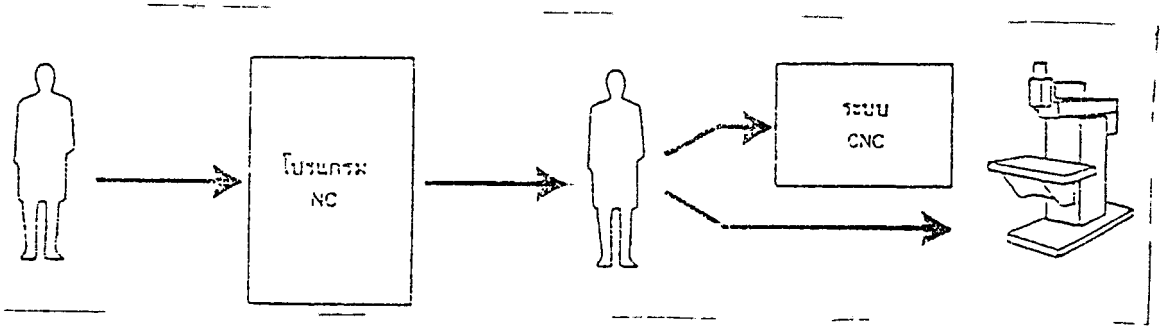
รูปที่ 1.1 เครื่องกลึง NC แบบ 2 แกน และ 3 แกน

1.3. ระบบควบคุมเครื่องจักรกลด้วยตัวเลข

เครื่องจักรกลซีเอ็นซี จะประกอบด้วยองค์ประกอบใหญ่ ๆ อยู่ 2 ส่วน คือ

1. **เครื่องจักรกล** เป็นส่วนที่ทำหน้าที่ ตัดเฉือนชิ้นงานตามขั้นตอนการทำงานที่กำหนดไว้

2. **ระบบซีเอ็นซี** เป็นส่วนที่ทำหน้าที่ควบคุมขั้นตอนการตัดเฉือนทั้งหมด



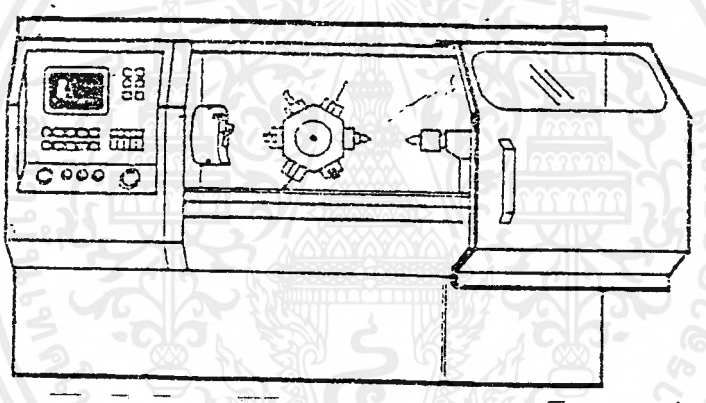
รูปที่ 1.2 องค์ประกอบของเครื่องจักรกล CNC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลเดิมที่อธิบายรายละเอียดของขั้นตอนที่ใช้ในการตัดเฉือนชิ้นงาน จะถูกป้อนเข้าไปในระบบควบคุมของเครื่องจักรกลก่อน ในรูปแบบของโปรแกรมเอ็นซี ซึ่งถูกจัดเตรียมโดยช่างเขียนโปรแกรม ช่างควบคุมเครื่องจะเป็นผู้ป้อนโปรแกรมเข้าไปในระบบควบคุม ซึ่งอาจป้อนด้วยมือผ่านแป้นพิมพ์โดยตรง หรือใช้แถบกระดาษเจาะรูก็ได้ หลังจากนั้น ก็จะเดินเครื่องทดลองโปรแกรม และสังเกตุสภาวะการตัดเฉือนชิ้นงานในแต่ละขั้นตอน บ่อยครั้งที่ช่างควบคุมเครื่องจะต้องจัดเตรียมโปรแกรม หรือเขียนโปรแกรมด้วยตนเอง หรือแก้ไขปรับปรุงโปรแกรมให้มีประสิทธิภาพในการตัดเฉือนสูงสุด ดังนั้น จึงเป็นสิ่งจำเป็นที่ช่างควบคุมเครื่องจะต้องมีความรู้ทั้งระบบควบคุมของเครื่องจักรกลและการเขียนโปรแกรมเอ็นซีด้วย

1.3.1 องค์ประกอบของเครื่องจักรกลที่ควบคุมได้

องค์ประกอบหรือชิ้นส่วนของเครื่องจักรกล ที่ทำหน้าที่เคลื่อนที่เข้าตัดเฉือนชิ้นงาน และองค์ประกอบอื่น ๆ ที่ช่วยเสริมการทำงานตัดเฉือนให้สมบูรณ์ยิ่งขึ้น จะถูกควบคุมโดยโปรแกรมเอ็นซี ด้วยวิธีการควบคุมแบบต่าง ๆ กัน



รูปที่ 1.3 เครื่องกลึง CNC

ช่างชำนาญงานที่ทำหน้าที่ควบคุมการทำงานของเครื่องจักรกลเอ็นซี หรือซีเอ็นซี จะต้องมีความคุ้นเคยกับหน้าที่การทำงานและขีดจำกัดในการทำงานของ เครื่องจักรกลซีเอ็นซีนั้นเป็นอย่างดี ช่างจะใช้วิธีการทำงานแบบง่าย ๆ โดยการจับยึดชิ้นงานเข้ากับโต๊ะงานและคาดว่าจะได้วิธีการตัดเฉือนที่ดีที่สุดไม่ได้ ในทางตรงข้ามช่างจะต้องรู้ว่าองค์ประกอบส่วนใดของเครื่องจักรกลซีเอ็นซีที่สามารถควบคุมได้และมีวิธีการควบคุมอย่างไร องค์ประกอบของเครื่องจักรกลเอ็นซีและซีเอ็นซีที่สามารถควบคุมได้ และจะกล่าวถึงในที่นี้ได้แก่

- แนวแกนป้อน (Feed axes)
- การขับป้อน (Feed drives)
- อุปกรณ์วัดขนาด (Measuring devices)
- เพลางาน (Work Spindle)
- อุปกรณ์จับยึดชิ้นงาน (Workpiece holding devices)

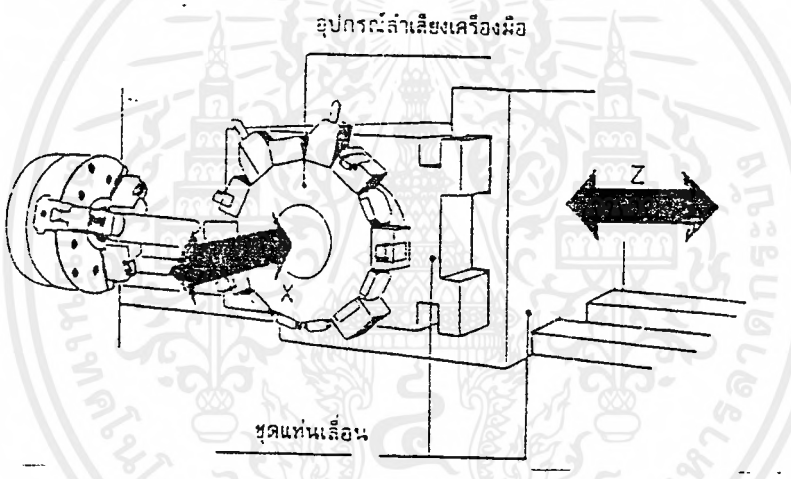
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) แนวการป้อน (Feed axes)

ในการกล่าวถึงเครื่องจักรกลซีเอ็นซี บ่อยครั้งที่จะได้ยินคำว่า **แนวแกน** (axes) ซึ่งหมายถึง แนวการเคลื่อนที่ขององค์ประกอบของเครื่องจักรกล เช่น โต๊ะงาน เพลาหัวเครื่อง อุปกรณ์ลำเลียงเครื่องมือ (Tool carriers) เป็นต้น

สำหรับเครื่องจักรกลทั่วไป การเคลื่อนที่ในแนวแกนต่าง ๆ จะเกิดจากการหมุนมือหมุนหรือโยกคันโยกป้อนอัตโนมัติ (Feed levers)

เครื่องจักรกลซีเอ็นซีมีแนวแกนป้อนรวมกันอยู่หลายแนวแกนทำให้สามารถตัดเฉือนชิ้นงานให้เป็นรูปทรงต่าง ๆ ที่ต้องการได้ การกำหนดแนวแกนต่าง ๆ ของเครื่องจักรกลซีเอ็นซีจะกำหนดตามมาตรฐานสากลภายใต้หัวเรื่อง Coordinate axes and direction of movement for numerically controlled machinery ซึ่งจะกำหนดแนวแกนเหล่านี้โดยใช้ตัวอักษร x,y และ z ดังแสดงในรูป 1.4



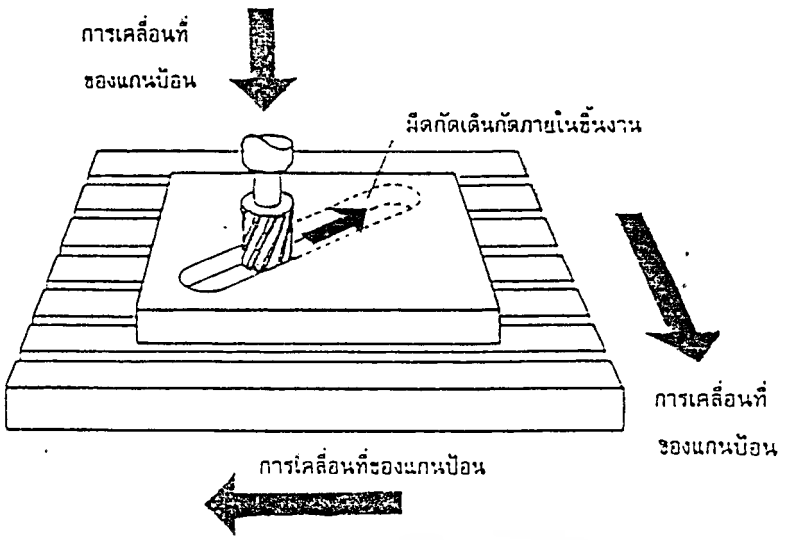
รูปที่ 1.4 เครื่องกลึงซีเอ็นซีแบบ 2 แกน

เครื่องกลึงจะมีแนวการป้อนอยู่ 2 แนวแกน คือ แกน X และ แกน Z ทั้งสองแนวแกนจะอยู่ที่ชุดแท่นเลื่อน ซึ่งมีอุปกรณ์ลำเลียงเครื่องมือ (Tool Carrier) ติดตั้งอยู่ ลักษณะเช่นนี้ทำให้สามารถกลึงชิ้นงานที่มีรูปทรงต่าง ๆ กันได้ตามต้องการ

2) การขับป้อน (Feed drives)

การเคลื่อนที่เรียงลำดับกันหรือพร้อม ๆ กันอย่างต่อเนื่องของแนวแกนป้อน จะทำให้เกิดการตัดเฉือนของเครื่องมือในชิ้นงาน ดังรูป 1.5

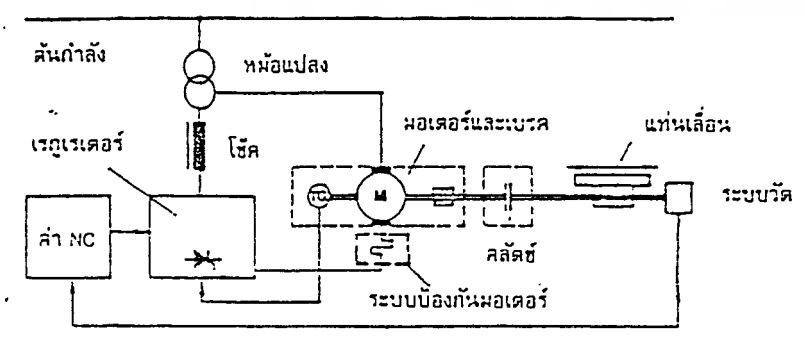
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.5 การเคลื่อนที่ตัดเฉือนของเครื่องมือตัด

การขับป้อนจะทำให้เกิดการเคลื่อนที่ของแท่นเลื่อน ในขณะตัดเฉือนแท่นเลื่อนอาจพาให้ชิ้นงานเคลื่อนที่หรือคมตัดเคลื่อนที่ก็ได้ ระบบขับป้อนโดยทั่วไปจะใช้มอเตอร์กระแสตรงในการหมุนขับและควบคุมการทำงาน ด้วยวงจรอิเล็กทรอนิกส์จากภายนอก มอเตอร์ชนิดนี้จะสามารถหมุนและเบรคให้หยุดได้ทั้งสองทิศทางขณะตัดเฉือนชิ้นงาน การเคลื่อนที่ป้อนจะต้องเป็นไปอย่างสม่ำเสมอและสามารถต้านแรงกระทำจากภายนอกได้ เช่น แรงตัดเฉือน เป็นต้น ด้วยเหตุนี้ระบบขับป้อนจึงต้องได้รับการออกแบบให้มีความแข็งแกร่งสูง การเคลื่อนที่คงที่และสม่ำเสมอสามารถตอบสนองต่อการเปลี่ยนอัตราป้อนได้อย่างรวดเร็ว นอกจากนี้ในขณะทำงานคมตัดอาจที่อหรือการเคลื่อนที่ของแท่นเลื่อนถูกกีดขวาง หรือการเร่งอัตราป้อนให้เคลื่อนที่เร็ว และหยุดโดยทันทีทันใด สาเหตุเหล่านี้! จะทำให้ มอเตอร์รับภาระมากเกินไป (Over loading) ซึ่งอาจทำให้มอเตอร์เสียหายได้ ดังนั้น จึงต้องมีการป้องกัน อุบัติเหตุเหล่านี้

- โดยทั่วไปแล้วจะใช้คลัตช์แบบลูกกลิ้ง (Over running clutch) ร่วมกับวงจรอิเล็กทรอนิกส์ ปัจจัยหนึ่งที่จะทำให้ระบบขับป้อนสามารถทำงานได้อย่างมีประสิทธิภาพก็คือ การเลือกใช้อุปกรณ์ในระบบขับป้อนให้เหมาะสมกับการทำงานของเครื่องจักรและการออกแบบวงจรควบคุมการทำงานที่มีประสิทธิภาพดังแสดงในรูป 1.6



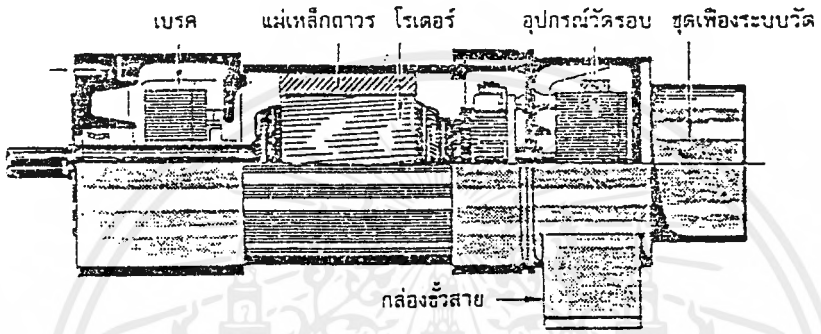
รูปที่ 1.6 Diagram ระบบขับป้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 มอเตอร์

เครื่องจักรกลเอ็นซีสมัยใหม่นั้น จะออกแบบโดยการใช้ ระบบขับป้อนแบบเซอโว (Servodrives)ทำให้สามารถปรับอัตราป้อนและความเร็วรอบได้โดยไม่มีขีดจำกัดของขั้นความเร็วและอัตราป้อน มอเตอร์ที่ใช้ในระบบขับป้อนโดยทั่วไปจะมีอยู่ 3 ชนิดด้วยกันคือ

ก.) มอเตอร์กระแสตรง (DC motors) ลักษณะสร้างของมอเตอร์กระแสตรงจะใช้เป็นแม่เหล็กถาวรที่มี 4,6 หรือ 8 ขั้ว ประกอบด้วยระบบเบรค (brake) แกนมอเตอร์ (Rotor) อุปกรณ์วัดรอบ (Tachogenerator) และอุปกรณ์วัด (Measuring box) ดังแสดงในรูป 1.7



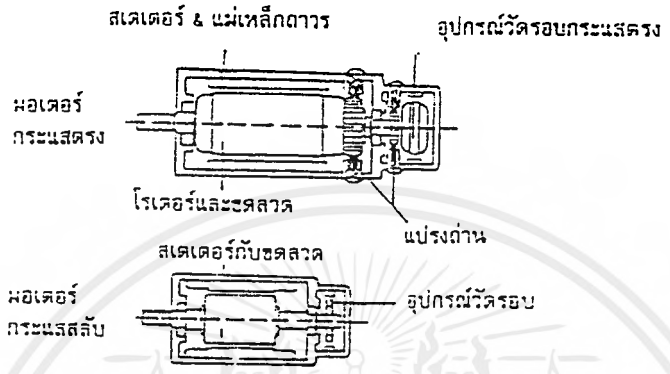
รูปที่ 1.7 ส่วนประกอบของมอเตอร์กระแสตรง

การใช้มอเตอร์กระแสตรง ทำให้สามารถปรับอัตราป้อนได้ละเอียดและมีวงจรควบคุมที่ไม่ซับซ้อน แต่ก็มีข้อเสียตรงที่มอเตอร์ชนิดนี้ต้องใช้แปรงถ่าน ซึ่งจะต้องคอยทำความสะอาดและเปลี่ยนเมื่อแปรงถ่านหมด นอกจากนี้ แปรงถ่านยังทำให้แกนมอเตอร์สึกหรออันเป็นผลทำให้กำลังมอเตอร์ลดลง ข้อเสียอีกประการหนึ่งก็คือ หากต้องการกำลังขับสูง มอเตอร์ก็จะมีขนาดใหญ่ด้วย และเมื่อใช้ความเร็วรอบสูง ๆ จะทำให้แรงบิดลดลง ดังนั้น จึงมักใช้กับเครื่องจักรกลเอ็นซีขนาดเล็กและขนาดกลาง

ข.) มอเตอร์แบบเป็นขั้น (Stepping motors) เป็นมอเตอร์ที่ทำงานแบบต่อเนื่องโดยการแปลงคลื่นสัญญาณที่ป้อนเข้าไปในระบบให้เป็นการเคลื่อนที่เชิงมุม การหมุนในแต่ละมุมหรือขั้นที่เปลี่ยนไป 1 ขั้นจะเท่ากับ 1 คลื่นสัญญาณ ดังนั้น ตำแหน่งของเพลาจะถูกกำหนดโดยจำนวนคลื่นสัญญาณที่ป้อนเข้าไปในระบบ และความเร็วในการหมุนของเพลาจะวัดเป็นจำนวนขั้นต่อวินาที (Steps per second) ซึ่งจะเท่ากับความถี่ของคลื่นสัญญาณที่ป้อนเข้าในระบบที่วัดเป็นจำนวนคลื่นสัญญาณต่อวินาที (pulses per second) ความเที่ยงตรงของระบบจะขึ้นอยู่กับความสามารถของมอเตร์ในการแบ่งขั้นการหมุนตามจำนวนคลื่นสัญญาณที่ป้อนเข้าในระบบ แรงบิดของมอเตอร์ชนิดนี้จะลดลงเมื่อความเร็วในการหมุนแบ่งเพิ่มขึ้น ดังนั้น จึงเหมาะสำหรับเครื่องจักรกลเล็ก ๆ ที่ไม่ต้องใช้กำลังขับมาก เช่น เครื่องพลอตเตอร์ (Plotter machine) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

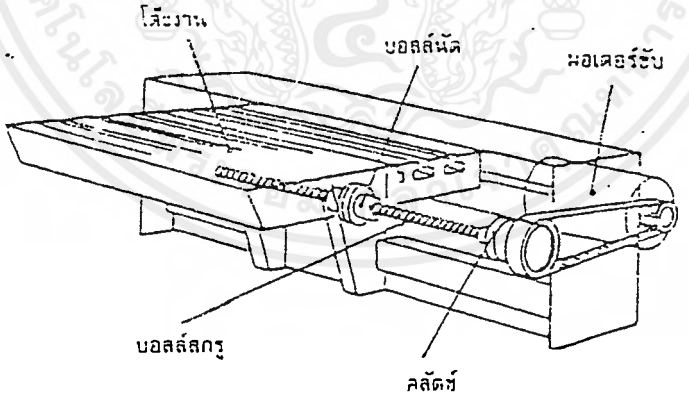
ค.) มอเตอร์กระแสสลับ (Alternate currrent motor) ส่วนมากจะเป็นมอเตอร์แบบซิงโครนัส (Synchronous motor) ข้อดีของมอเตอร์ชนิดนี้คือ ไม่ต้องใช้แปรงถ่าน ทำให้สามารถลดงานบำรุงรักษาได้มาก และมอเตอร์ขนาดเดียวกันเมื่อเปรียบเทียบกับมอเตอร์กระแสตรง จะสามารถให้แรงบิดได้ดีกว่า และมีขนาดเล็กกว่าด้วย ดังแสดงในรูป ส่วนข้อเสียของมอเตอร์แบบนี้คือ วงจรควบคุมจะมีความซับซ้อนมากกว่าวงจรควบคุมมอเตอร์กระแสตรง



รูปที่ 1.8 เปรียบเทียบลักษณะสร้างและขนาดของมอเตอร์กระแสตรงกับมอเตอร์กระแสสลับแบบ 3 เฟส

2.2 บอลล์สกรู (Ball screws)

หัวใจของระบบขับป้อนของเครื่องจักรกลซีเอ็นซีก็คือ การส่งกำลังขับด้วยบอลล์สกรู ซึ่งจะมีลูกบอลล์ไหลหมุนเวียนอยู่ตลอดเวลา บอลล์สกรูจะประกอบด้วยสกรูกับนัตที่มีลักษณะเป็นเกลียวกลม ร่องเกลียวกลมบนสกรูและในนัตจะขุบแข็งและเจียระไนผิวเรียบมันเพื่อลดความฝืด และเพิ่มความเที่ยงตรงในการเคลื่อนที่ ดังแสดงในรป 1.9



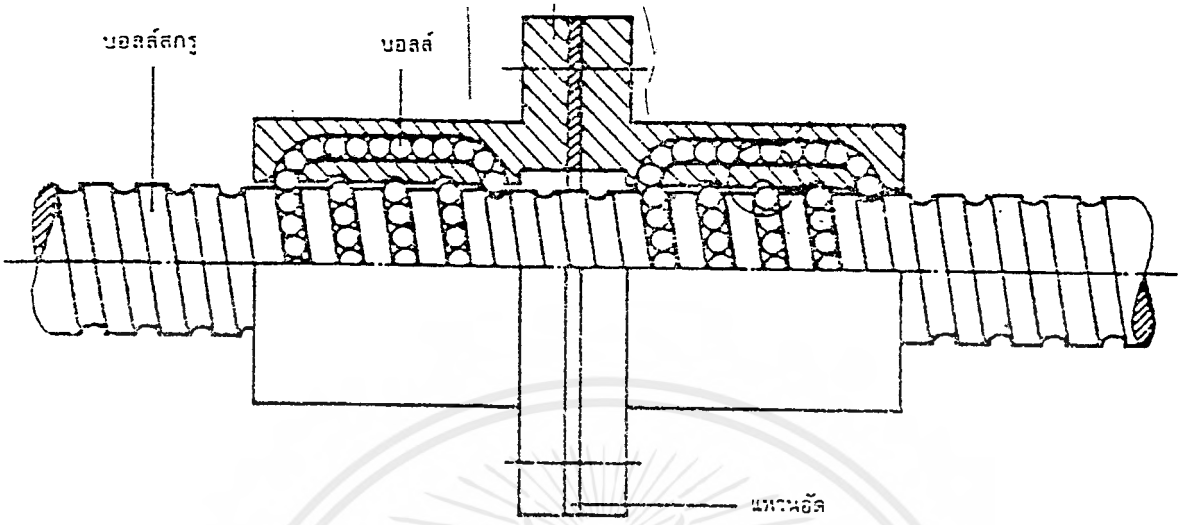
รูปที 1.9 การขับป้อนของโต๊ะงาน

เมื่อมอเตอร์หมุนขับสกรู นัตก็จะเคลื่อนที่ไปตลอดความยาวของสกรู พาให้แท่นเลื่อนและโต๊ะงานเคลื่อนที่ไปตามรางเลื่อน

ภายในของตัวนัตจะประกอบไปด้วยชุดของลูกบอลล์จำนวนมาก ดังแสดงในรูปข้างล่างนี้ ทำให้มั่นใจได้ว่าความเสียดทานในการส่งกำลังขับจากสกรูไปยังแท่นเลื่อนจะมีน้อยมาก นัตจะถูกแบ่งออกเป็นสองซีก และ ขันประกอบยึดเข้าด้วยกัน โดยมีการเตรียมอัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แรงไว้ก่อน ทำให้สามารถลดระยะคลอน (backlash) ให้เหลือน้อยที่สุดจนแทบจะไม่มีเลยได้ ทำให้การเคลื่อนที่ของแท่นเลื่อนมีความเที่ยงตรงสูงสามารถทำงานซ้ำ ๆ กันได้



รูปที่ 1.10 ลักษณะสร้างภายในของชุดบอลล์สกรู

การต่อกำลังระหว่างมอเตอร์กับบอลล์สกรู จะมีชุดคลัตช์ความฝืดเป็นตัวเชื่อม ซึ่งนอกจากจะมีหน้ามี่ต่อกำลังขับแล้ว ยังมีหน้าที่ป้องกันอุบัติเหตุที่เกิดจากแท่นเลื่อนหรือโต๊ะงานชนหรือกระแทกกับสิ่งกีดขวางไม่ให้เครื่องจักรกลซีเอ็นซีเกิดความเสียหายมากเกินไป กล่าวคือ เมื่อมีการชน หรือกระแทกกันขึ้นจนแรงมากถึงค่าหนึ่ง ชุดคลัตช์ก็จะตัดระบบการส่งกำลังขับระหว่างมอเตอร์กับตัวบอลล์สกรูทันที

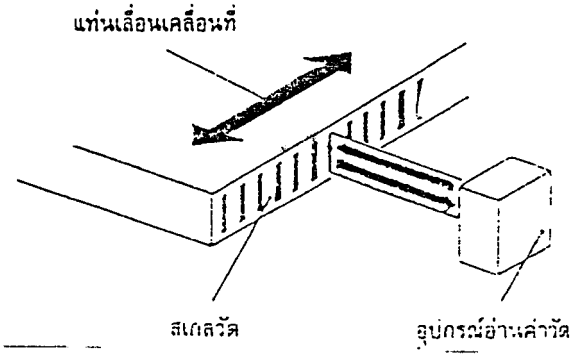
3.) **ระบบวัดขนาด (Measuring System)**

การเคลื่อนที่ไปที่ตำแหน่งต่าง ๆ ในแต่ละแนวแกนของแท่นเลื่อน จะถูกส่งไปยังระบบควบคุมโดยระบบวัดขนาด การวัดตำแหน่งของแท่นเลื่อนสามารถที่จะวัดได้ทั้ง โดยตรง (Direct Measurement) และโดยทางอ้อม (Indirect Measurment)

3.1) **การวัดตำแหน่งโดยตรง**

วิธีนี้จะใช้สเกลวัด (measuring scale) ยึดติดกับแท่นเลื่อนหรือโต๊ะงานโดยตรง ดังแสดงใน รูป 1.11 และข้อดีของวิธีวัดแบบนี้ ก็คือ ความไม่เที่ยงตรงของสกรูนำเลื่อน (Leadscrew) ระบบขับจะไม่มีผลกระทบต่อค่าที่อ่านได้ อุปกรณ์ อ่านค่าวัด (Measuring valve resolver) จะอ่านข้อมูลในการวัดจากขีดสเกลวัด (Measuring scale grid) และแปลงข้อมูลนี้เป็นสัญญาณไฟฟ้าและส่งกลับไปยังระบบควบคุม

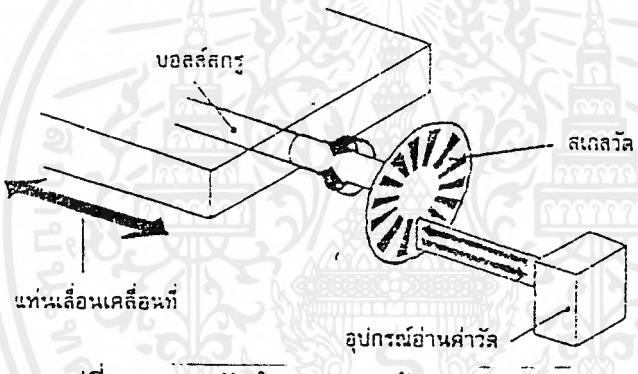
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.11 การวัดตำแหน่งโดยตรง

3.2) การวัดตำแหน่งทางอ้อม

การเคลื่อนที่ของแท่นเลื่อนจะได้รับกำลังขับมาจากการหมุนของบอลล์สกรู อุปกรณ์เปลี่ยนค่าวัด (Resolver) จะบันทึกการเคลื่อนที่หมุนของแผ่นจานสัญญาณ (Pulse disc) ที่ต่อคิดอยู่กับบอลล์สกรู และส่งต่อไปยังระบบควบคุมของเครื่อง ระบบควบคุมก็จะใช้สัญญาณ ได้รับนี้ไปคำนวณหาระยะทางการเคลื่อนที่ของแท่นเลื่อนจากสัญญาณการหมุน (rotation pulses) ของแผ่นจานสัญญาณ ดังแสดงในรูป 1.12



รูปที่ 1.12 การวัดตำแหน่งทางอ้อม

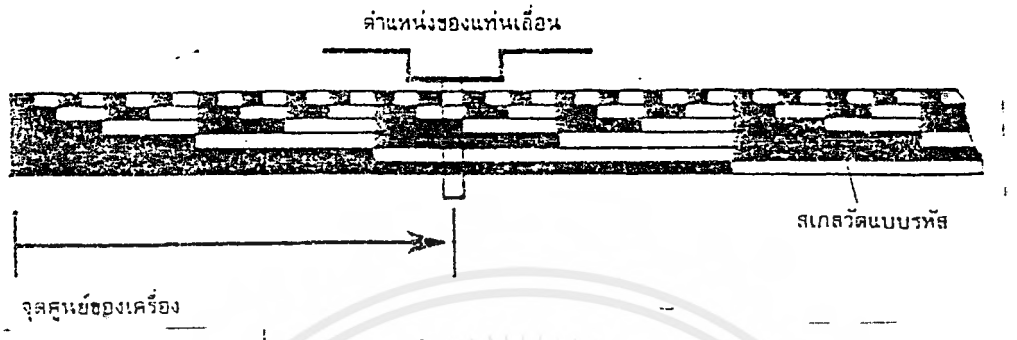
นอกจากการวัดตำแหน่งทางตรงและทางอ้อมแล้ว ในระบบการวัดขนาดของเครื่องจักรกลเอ็นซีที่ต้องการให้การวัดตำแหน่งมีความเที่ยงตรงตลอดแนวแกนป้อน จะต้องต่อระบบขับป้อนเข้ากับอุปกรณ์วัดที่เหมาะสม อุปกรณ์วัดมีอยู่ด้วยกัน 2 ชนิดคือ สเกลวัดแบบรหัส (Coded measuring scale) กับสเกลวัดแบบช่อง (division grid) การใช้สเกลวัดทั้ง 2 ชนิดนี้จะขึ้นอยู่กับวิธีการวัดตำแหน่ง (position measurement) วิธีการวัดตำแหน่งที่นิยมใช้กันทั่วไปมีอยู่ 2 วิธีคือ การวัดตำแหน่งแบบสัมบูรณ์ (ablolute position measurement) กับการวัดตำแหน่งแบบต่อเนื่อง หรือแบบลูกโซ่ (incremental or chain position measurement) ซึ่งมีความแตกต่างกันดังรายละเอียดต่อไปนี้

3.3) การวัดตำแหน่งแบบสัมบูรณ์

คำว่า **สัมบูรณ์** (absolute) ที่ใช้ร่วมกับการวัดตำแหน่งนี้จะหมายความว่า ค่าตำแหน่งต่าง ๆ สามารถวัดได้ตลอดเวลาและเป็นอิสระจากสถานะของเครื่องและระบบควบคุม ทั้งนี้เพราะค่าต่าง ๆ เหล่านี้จะวัดอ้างอิงจากจุดศูนย์อ้างอิง (fixed zero datum) เสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวัดตำแหน่งแบบสัมบูรณ์ ดังแสดงในรูป จะใช้สเกลวัดแบบรหัส (coded measuring scale) ซึ่งจะชี้ตำแหน่งของแท่นเลื่อนที่ถูกต้องตลอดเวลา โดยอ้างอิงจากตำแหน่งจุดศูนย์ของเครื่อง (Mechine zero point) ซึ่งเป็นตำแหน่งศูนย์ที่มีจุดอ้างอิงที่แน่นอนและถาวรของเครื่องจักรกลเอ็นซี



รูปที่ 1.13 การวัดตำแหน่งแบบสัมบูรณ์

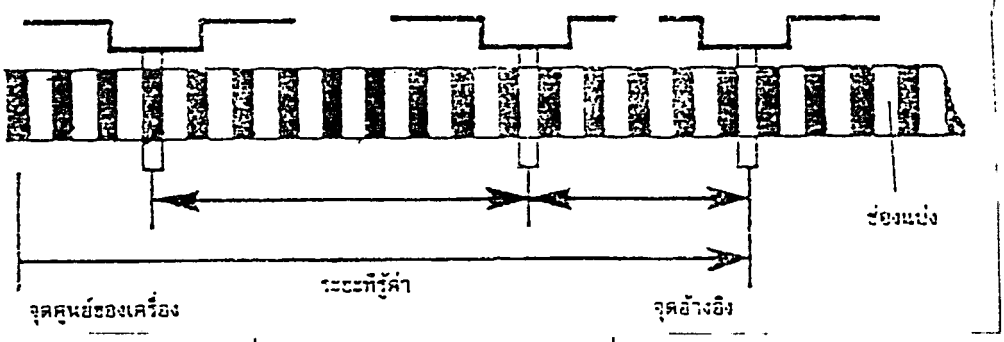
ข้อสำคัญของการใช้วิธีนี้คือ ความยาวของช่วงอ่านค่าวัดของสเกลจะต้องยาวกว่าระยะเลื่อนทำงานของแท่นเลื่อน เพื่อให้ระบบควบคุมของเครื่อง สามารถอ่านค่าวัดได้ทุกตำแหน่งสเกลนี้จะใช้รหัสเป็นระบบตัวเลขฐานสอง (Binary system)

3.4) การวัดแบบต่อเนื่อง

คำว่า **ต่อเนื่อง** (incremental) แปลว่าระยะเลื่อนสั้น ๆ ตามความยาวที่กำหนด ดังนั้นในการวัดตำแหน่งอาจจะเป็นการเพิ่มหรือลดขนาดความยาวในการเคลื่อนที่ที่วัดอยู่ก็ได้ ในระหว่างการเคลื่อนที่ของแท่นเลื่อนระบบควบคุมจะทำการนับจำนวนส่วนแบ่ง (divisions)ที่ตำแหน่งใหม่แตกต่างจากตำแหน่งก่อนหน้านี้เสมอ

วิธีการวัดตำแหน่งแบบต่อเนื่องที่แสดงในรูปถัดไป สเกลวัดจะแบ่งเป็นช่อง (grid) แบบง่าย ๆ โดยที่แต่ละช่องจะมีพื้นที่สว่างกับมืดสลับกันไป เมื่อแท่นเลื่อนเคลื่อนที่ ช่องนี้ก็จะวิ่งผ่านอุปกรณ์เปลี่ยนค่าวัด (resolver) ซึ่งจะทำหน้าที่นับ จำนวนช่องพื้นที่สว่างและมืด จากนั้นก็จะส่งเป็นสัญญาณไฟฟ้าไปยังระบบควบคุมของเครื่อง ระบบควบคุมก็จะนำสัญญาณนี้มาคำนวณหาตำแหน่งสุดท้ายของแท่นเลื่อนที่แตกต่างจากตำแหน่งก่อนหน้านี้ ในทางปฏิบัติที่ต้องการให้วิธีการวัดแบบนี้ทำงานได้อย่างถูกต้อง เมื่อเริ่มเปิดสวิตช์ระบบควบคุมของเครื่องควรจะเลื่อนไปยังจุดที่ทราบค่าระยะห่างจากจุดศูนย์ของเครื่องจุดนี้จะเรียกว่า " จุดอ้างอิง " (Reference Point) หลังจากที่แท่นเลื่อนในแนวแกนต่าง ๆ เลื่อนไปยังจุดอ้างอิงแล้ว อุปกรณ์อ่านค่าวัดก็จะสามารถทำหน้าที่วัดตำแหน่งด้วยช่องสเกลได้

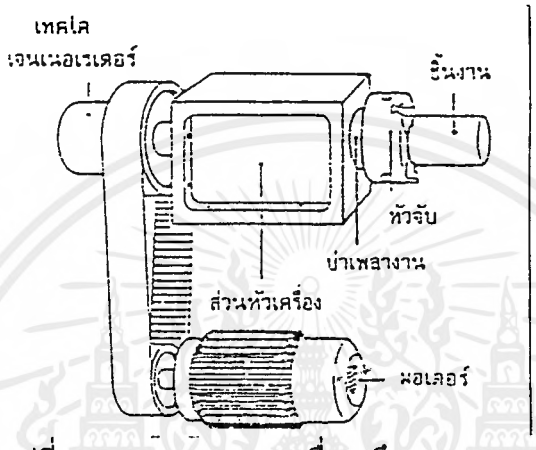
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.14 การวัดตำแหน่งแบบต่อเนื่อง

4.) เพลางาน (Work Spindle)

เพลางานเป็นชิ้นส่วนหรือองค์ประกอบของเครื่องจักรกลที่มีความสำคัญมาก มีหน้าที่หลักในการทำงาน คือจะทำหน้าที่ขับพาให้เครื่องมือ เช่น มีดกลึง ดอกสว่าน เป็นต้น

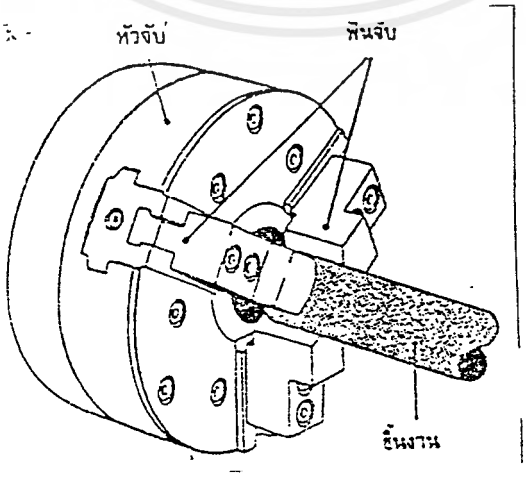


รูปที่ 1.15 เพลางานของเครื่องกลึง

5.) อุปกรณ์จับยึดชิ้นงาน (Workpiece holding devices)

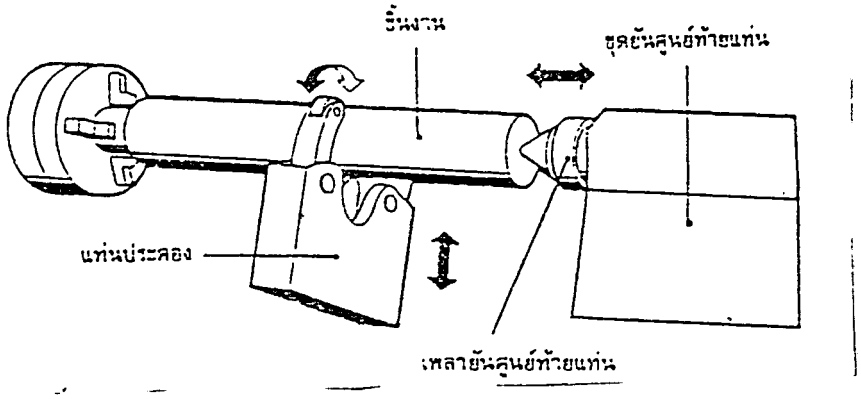
อุปกรณ์จับยึดชิ้นงานจะจัดเตรียมไว้สำหรับยึดชิ้นงานเข้ากับเพลางานในงานกลึง (Latch) สามารถเลือกใช้อุปกรณ์จับยึดชิ้นงานแบบต่าง ๆ กันได้หลายชนิด เช่น

- หัวจับแบบ 2,3 หรือ 4 ฟันจับ
- หน้าจานจับ สำหรับจับยึดชิ้นงานที่มีรูปทรงไม่สมมาตร
- ยันศูนย์สำหรับเพลางานและเพลาของชุดยันศูนย์ท้ายแท่น
- แท่นประคอง



รูปที่ 1.16 หัวจับงานกลึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



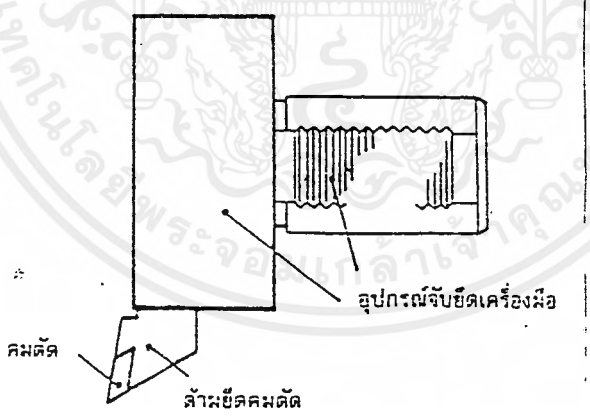
รูปที่ 1.17 ชุดยันศูนย์ท้ายแท่นและแท่นประคองงานกลึง

6.) **เครื่องมือ** (Tool)

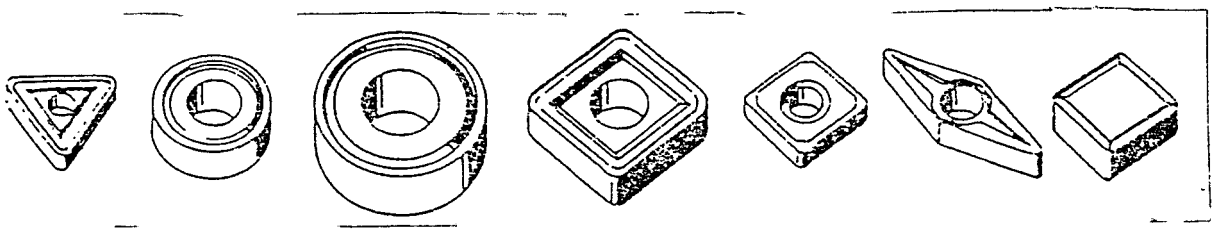
การทำงานของเครื่องจักรกล โดยทั่วไปจะต้องทำงานควบคู่กับเครื่องมือ ซึ่งได้แก่ มีดกลึง ดอกสว่าน เป็นต้น เครื่องมือที่เหมาะสมจึงเป็นสิ่งสำคัญยิ่งและเป็นการช่วยเสริมการทำงานของเครื่องจักรกลซีเอ็นซีให้สามารถทำงานได้ประสิทธิภาพสูงสุด

องค์ประกอบของเครื่องมือที่สมบูรณ์สำหรับเครื่องซีเอ็นซี ซึ่งจะประกอบด้วยส่วนต่าง ๆ ดังนี้

- อุปกรณ์จับยึดเครื่องมือ (Tool Holders)
- ด้ามยึดคมตัด (Tool Tip Carrier)
- คมตัด หรืออินเสิร์ต (Insert)



รูปที่ 1.18 องค์ประกอบของเครื่องมือที่ใช้ในงานกลึง



รูปที่ 1.19 อินเสิร์ตแบบถอดเปลี่ยนได้รูปทรงต่าง ๆ

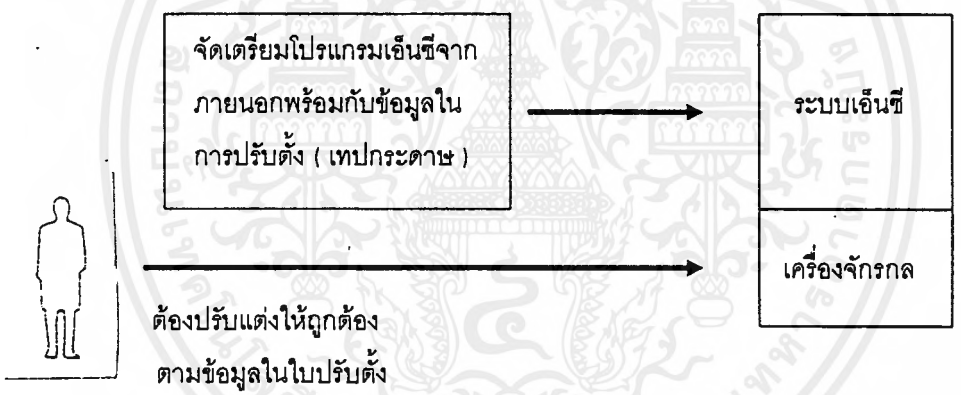
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ระบบควบคุมซีเอ็นซี (CNC Control System)

1.4.1 หน้าที่การทำงานที่โปรแกรมได้ (Programmable Functions)

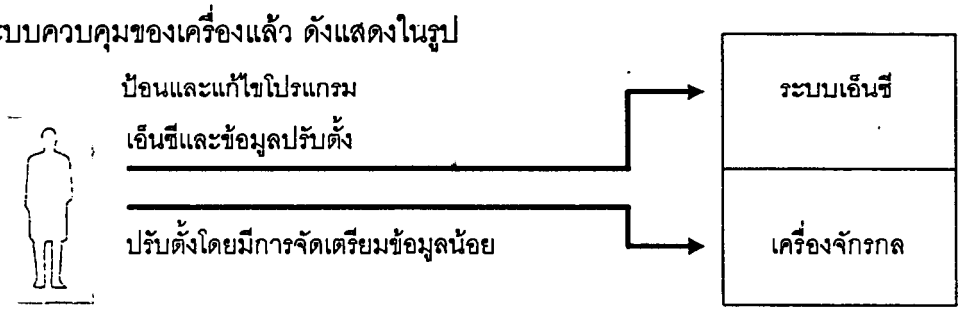
ในปัจจุบัน ระบบควบคุมการทำงานของเครื่องจักรกลสมัยใหม่เกือบทั้งหมด จะควบคุมด้วยระบบซีเอ็นซี แต่เนื่องจากยังคงอ้างอิงถึงโปรแกรมเอ็นซี (NC Program) และเทคโนโลยีเอ็นซี (NC Technology) อยู่ ดังนั้น จึงเป็นเรื่องสำคัญที่จะต้องรู้ถึงความแตกต่างในการทำงานระหว่างระบบเอ็นซีกับระบบซีเอ็นซี

ระบบเอ็นซี (NC System) ดังแสดงในรูปถัดไป จะมีระบบควบคุมประกอบอยู่กับเครื่องจักรกลซึ่งจะต้องจัดเตรียมโปรแกรมเอ็นซีจากภายนอกก่อน แล้วจึงป้อนเข้าไปในระบบควบคุม โดยอาศัยสื่อข้อมูล (data carries) เช่น เทปกระดาษ (Punched tape) เป็นต้น โปรแกรมเอ็นซีที่ป้อนเข้าไปในระบบควบคุมของเครื่องจะถูกเลือกใช้ในขณะเขียนโปรแกรมไว้ก่อนและกำหนดไว้ในใบปรับตั้ง (set-up sheet) ซึ่งช่างควบคุมเครื่องจะต้องจัดเตรียมและประกอบยึดเครื่องมือ ตลอดจนอุปกรณ์จับยึดชิ้นงาน ให้ถูกต้องตามข้อมูลที่กำหนดไว้ใน ใบปรับตั้ง



รูปที่ 1.20 ระบบ NC

ระบบซีเอ็นซี (CNC system) จะมีคอมพิวเตอร์ประกอบอยู่ด้วย ดังนั้น ช่างควบคุมเครื่องไม่เพียงแต่จะสามารถใช้โปรแกรมเอ็นซีสั่งให้เครื่องจักรทำงานได้เท่านั้น แต่จะยังสามารถเขียนและป้อนโปรแกรมด้วยตนเอง ตลอดจนการแก้ไขโปรแกรมได้หลังจากป้อนเข้าไปในระบบควบคุมของเครื่องแล้ว ดังแสดงในรูป



รูปที่ 1.21 ระบบ CNC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขนาดต่าง ๆ ของเครื่องมือตัดและอุปกรณ์จับยึดชิ้นงาน สามารถที่จะเลือกใช้และป้อนเข้าไปในระบบควบคุมซีเอ็นซี ขณะทำการปรับตั้ง (setting-up) และเป็นอิสระจากตัวโปรแกรมเอ็นซี ขนาดต่าง ๆ ของเครื่องมือจะถูกนำไปใช้โดยอัตโนมัติในขณะทำการตัดเฉือน ด้วยเหตุนี้ช่างควบคุมเครื่องจึงไม่จำเป็นต้องมีข้อมูลในการปรับตั้งมากและสามารถที่จะเลือกใช้เครื่องมือและอุปกรณ์จับยึดชิ้นงานได้ด้วยตนเอง

หากพิจารณาถึงภาษาโปรแกรม (Programming language) และเทคโนโลยีทางด้านการตัดเฉือนของเครื่องจักรกลที่ใช้ในระบบเอ็นซีกับซีเอ็นซีแล้วจะไม่แตกต่างกัน

1.4.2 ชนิดของการควบคุม (Control modes)

ลักษณะการควบคุม การเคลื่อนที่ทำงานของแท่นเลื่อนต่าง ๆ ในเครื่องจักรกลเอ็นซีและซีเอ็นซี จะมีการเคลื่อนที่อยู่ 2 ลักษณะ คือ

การเคลื่อนที่ในแนวเส้นตรง (Linear Interpolation หรือ straight lineInterpolation) การเคลื่อนที่ลักษณะนี้ ระบบซีเอ็นซีจะคำนวณหาตำแหน่งของจุดต่าง ๆ ที่ต่อกันเป็นลูกโซ่ในแนวเส้นตรงระหว่างตำแหน่งของเครื่องมือ 2 ตำแหน่งในขณะที่เครื่องมือเคลื่อนที่จากจุดหนึ่งไปยังอีกจุดหนึ่งนั้น ระบบควบคุมซีเอ็นซีจะตรวจสอบและแก้ไขแนวแกนในการเคลื่อนที่ให้ถูกต้องอยู่ตลอดเวลาทำให้การเคลื่อนที่ของเครื่องมือไม่ผิดพลาดหรือคลาดเคลื่อนออกจากจุดต่อของเส้นตรงมากกว่าค่าพิกัดความเผื่อของเครื่องที่กำหนดไว้

การเคลื่อนที่ในแนวเส้นโค้ง (Circular Interpolation) ระบบควบคุมซีเอ็นซี จะคำนวณหาตำแหน่งของจุดต่าง ๆที่ต่อกันเป็นเส้นโค้งตามขนาดรัศมีที่กำหนดระหว่างตำแหน่งของเครื่องมือที่กำหนดไว้ 2 ตำแหน่ง ระบบควบคุมจะอาศัยจุดเหล่านี้ในการตรวจสอบและแก้ไขแนวการเคลื่อนที่ของเครื่องมือให้ถูกต้องและอยู่ภายใน พิกัดความเผื่อของเครื่องจักรกลที่กำหนด

ในระบบควบคุมซีเอ็นซีนั้น จะแบ่งการควบคุมการเคลื่อนที่ทั้งสองลักษณะ ตามลักษณะการเคลื่อนที่ป้อนออกเป็น 3 ชนิด คือ

1) การควบคุมจุดต่อจุด (Point to point control)

การควบคุมแบบนี้ จะมีการควบคุมการเคลื่อนที่ของเครื่องมือระหว่างจุดสองจุดที่โปรแกรมไว้ในลักษณะการเคลื่อนที่เร็ว (Rapid traverse) โดยเครื่องมือจะต้องไม่สัมผัสชิ้นงาน

แนวแกนในการเคลื่อนที่ขึ้นอยู่กับชนิดของระบบควบคุม กล่าวคือมอเตอร์ขับของระบบป้อนอาจจะเริ่มทำงานหลาย ๆ แนวแกนพร้อมกัน หรือ ทำงานทีละแนวแกน จนกว่าจะเคลื่อนที่ถึงตำแหน่งของเครื่องมือที่โปรแกรมไว้ ทำให้ไม่สามารถควบคุมทางเดินของเครื่องมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Tool path) ได้ การควบคุมแบบจุดต่อจุด นี้จะใช้กับเครื่องเจาะ (drilling machine), เครื่องเชื่อมจุด (spot drilling) เป็นต้น

2) การควบคุมการตัดเฉือนแนวเส้นตรง (Straight-cut control)

การควบคุมชนิดนี้ นอกจากจะสามารถควบคุมการเคลื่อนที่ของเครื่องมือแบบเคลื่อนที่เร็วได้แล้ว ยังสามารถควบคุมการเคลื่อนที่ของเครื่องมือในแนวขนานกับแนวแกนของเครื่องจักรกล ตามค่าอัตราป้อนที่ต้องการได้อีกด้วย แต่จะสามารถควบคุมการเคลื่อนที่ได้ครั้งละ 1 แนวแกนเท่านั้น การเคลื่อนที่ของเครื่องมือจะถูกควบคุมด้วยอัตราป้อนและความยาวในการเคลื่อนที่ ระบบการควบคุมการตัดเฉือนแนวเส้นตรงชนิดนี้จะใช้กับเครื่องกัดและเครื่องกลึงแบบง่าย ๆ

3) การควบคุมตามเส้นขอบรูป (Contouring control)

การควบคุมแบบนี้จะสามารถควบคุมการเคลื่อนที่ทำงานได้ดังนี้

- ควบคุมเครื่องมือให้เคลื่อนที่ไปยังตำแหน่งที่ต้องการแบบเคลื่อนที่เร็วได้
- ควบคุมเครื่องมือให้เคลื่อนที่ขนานกับแนวแกน ไปยังตำแหน่งที่ต้องการตามค่าอัตราป้อนได้
- ควบคุมเครื่องมือ ให้เคลื่อนที่ไปยังตำแหน่งใด ๆ บนชิ้นงาน ที่กำหนดโดยแนวเส้นตรงและเส้นโค้งตามค่าอัตราป้อนได้

1.4.3 การควบคุมหน้าที่การทำงานของเครื่องจักรกล (Control of machine function)

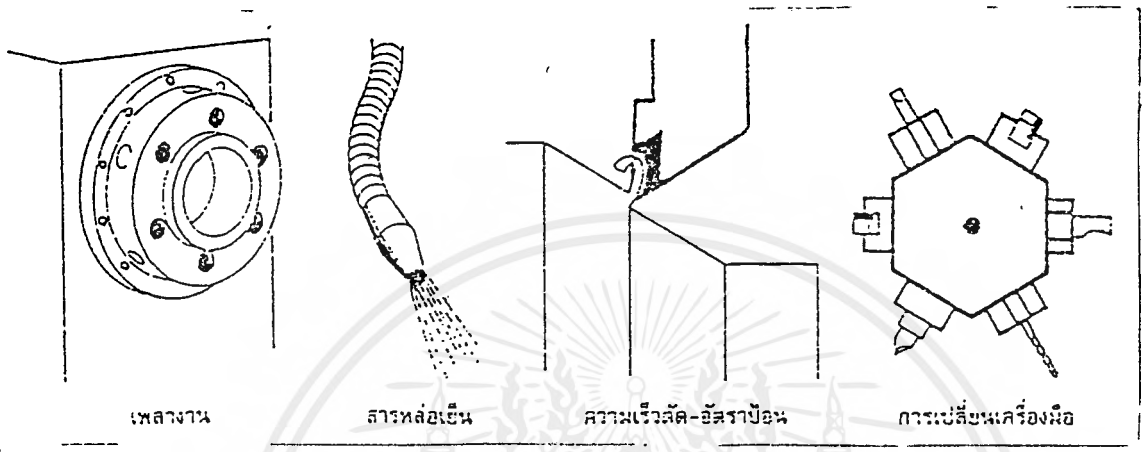
ระบบควบคุมซีเอ็นซีนอกจากจะสามารถควบคุมการเคลื่อนที่ของเครื่องมือ ตามรูปทรงเรขาคณิตของชิ้นงานแล้ว ยังสามารถควบคุมหน้าที่การทำงานอื่น ๆ ที่ช่วยเสริมการทำงานตัดเฉือนของเครื่องจักรกลให้เหมาะสมกับสภาวะการทำงานในขณะนั้นได้อีกด้วยแต่ยังขึ้นอยู่กับชนิดระบบควบคุมอีกด้วย

ตัวอย่างหน้าที่การทำงานต่าง ๆ ที่จำเป็นจะต้องโปรแกรมเพื่อช่วยในการทำงานดังแสดงในรูปข้างล่าง มีดังนี้

- การเริ่มหมุนของเพลางาน ทิศทางการหมุนและการเปลี่ยนความเร็วรอบ
- การกำหนดตำแหน่งของเพลางาน
- การเปิดสารหล่อเย็น และความดันของสารหล่อเย็น
- การรักษาอัตราป้อนให้คงที่
- การเปลี่ยนตำแหน่งของเครื่องมือ
- การรักษาความเร็วตัดให้คงที่
- ชุดยันศูนย์ท้ายแท่น (Tail - stock)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
038877

- ชุดยันศูนย์ท้ายแท่น (Tail - stock)
- อุปกรณ์ใส่และถอดชิ้นงาน (loader and unloader)
- แท่นประคองศูนย์ (Steady rest)
- อุปกรณ์ลำเลียงเศษ (chip conveyer)
- sorter



รูปที่ 1.22 หน้าที่การทำงานของเครื่องจักร

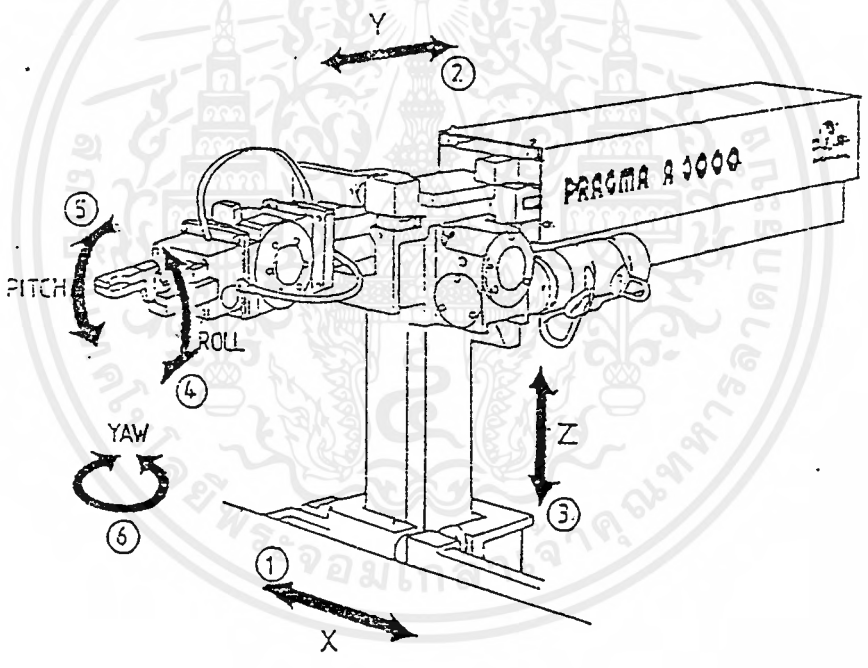
เครื่องจักรกลซีเอ็นซีที่สามารถใช้ระบบควบคุมสั่งการทำงานในหน้าที่ต่าง ๆ ได้ยิ่งมากเท่าใดก็จะเป็นระบบที่มีความเหมาะสมสำหรับการควบคุมแบบอัตโนมัติมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. หุ่นยนต์(Robotics)

2.1การเคลื่อนย้ายโดยหุ่นยนต์ (Robotics Manipulation)

คำว่า หุ่นยนต์ (Robot) สามารถนำไปสู่ความหมายต่างๆกัน ในความคิดของผู้อ่านขึ้นอยู่กับบทความที่เกี่ยวข้อง แต่ในที่นี้หุ่นยนต์ จะหมายถึงหุ่นยนต์ทางอุตสาหกรรม(Industrial robot) ซึ่งถูกเรียกว่า แขนกล (Robotic Manipulator or Robotic Arm) ตัวอย่างของหุ่นยนต์อุตสาหกรรม แสดงดังรูปที่ 2.1ซึ่งเป็นแขนกลข้อต่อหมุน และมีลักษณะคร่าวๆคล้ายคลึงกับแขนมนุษย์ แขนกลเปรียบได้กับ Rigid Rink หลายๆอัน ที่นำมาเชื่อมต่อกันด้วยข้อต่อยืดหยุ่นได้ (Flexible Joints) Link เหล่านี้ทำหน้าที่คล้ายกับอวัยวะบางส่วนของร่างกายมนุษย์ เช่น หน้าอก, ต้นแขน และปลายแขน ในขณะที่ข้อต่อต่างๆเปรียบได้กับ หัวไหล่,ข้อศอกและข้อมือ ที่ปลายแขนกลเป็นส่วนทำงาน(End - Effector) เรียกว่า Tool,Gripper หรือ Hand ปกติ Tool จะเป็นมือจับที่มี 2 นิ้ว หรือมากกว่าใช้ในการเปิดและปิด



รูป 2.1 หุ่นยนต์อุตสาหกรรม

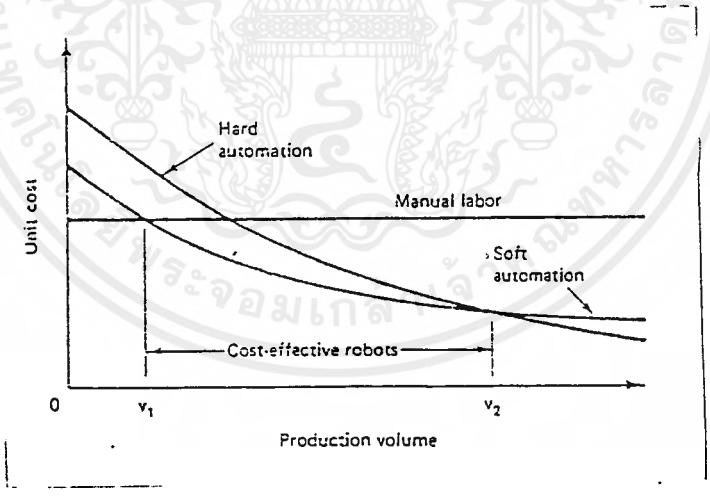
เพื่อที่จะอธิบายลักษณะของหุ่นยนต์อุตสาหกรรมต่อไปอีก เราจะเริ่มต้นด้วยการศึกษาบทบาทของหุ่นยนต์ในการผลิตแบบอัตโนมัติโดยกว้างๆตามด้วยวิธีแบ่งประเภทของหุ่นยนต์โดยใช้เกณฑ์ต่างๆได้แก่ เทคโนโลยีการขับเคลื่อน รูปทรงขอบเขตการทำงานและวิธีควบคุมการเคลื่อนที่ จากนั้นจะเป็นลักษณะเฉพาะอื่นๆของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.เครื่องจักรแบบอัตโนมัติกับหุ่นยนต์ (Automation And Robots)

สายการผลิตแบบ Mass - Production ถูกนำมาใช้ครั้งแรกในต้นศตวรรษที่ 20 (1905) โดยบริษัทฟอร์ดมอเตอร์ ในช่วงหลายสิบปีแรกนั้น เครื่องจักรพิเศษได้ถูกออกแบบและพัฒนาเพื่อใช้ในการผลิตชิ้นส่วนเครื่องกล และไฟฟ้าให้ได้ปริมาณมากๆแต่เมื่อจบรอบการผลิตในแต่ละปีชิ้นส่วนแบบใหม่ก็จะถูกนำมาใช้ ทำให้ต้องมีการปรับเปลี่ยนเครื่องจักร เพื่อใช้ในการผลิตชิ้นส่วนใหม่เหล่านั้น เนื่องจากเป็นการเปลี่ยนแปลงทางด้าน Hareware จึงเรียกเครื่องจักรแบบนี้ว่าเป็น Hard Automation เครื่องจักรและกระบวนการผลิตแบบนี้จะมีประสิทธิภาพมาก แต่จะมีความยืดหยุ่นจำกัด

ในยุคต่อมา อุตสาหกรรมรถยนต์และอุตสาหกรรมอื่นๆได้เริ่มนำเครื่องจักรที่มีความยืดหยุ่นมาใช้ในวัฏจักรการผลิต ปัจจุบันแขนกลที่โปรแกรมได้ (Programmable-Mechanical Manipulators) ถูกนำมาใช้ในงาน เช่น การเชื่อมจุด,การพ่นสี,การลำเลียงวัสดุ และการประกอบชิ้นส่วน เนื่องจากแขนกลที่ควบคุมด้วยคอมพิวเตอร์นี้ สามารถปรับเปลี่ยนผ่านทาง Software เพื่อให้ทำงานได้หลายอย่าง จึงถูกเรียกว่าเป็น Soft Automation การเปรียบเทียบค่าใช้จ่ายระหว่างการใช้แรงงานคน เครื่องจักรแบบ Hard Automation และแบบ Soft Automation เทียบกับปริมาณการผลิตแสดงดังรูปที่ 2.2



รูป 2.2 ความสัมพันธ์ระหว่างค่าใช้จ่ายกับปริมาณการผลิต

จากรูปเห็นได้ชัดว่า ที่ปริมาณการผลิตน้อยๆ ค่าใช้จ่ายจากแรงงานคนจะมีค่าต่ำสุด เมื่อปริมาณการผลิตเพิ่มมากขึ้น จนถึงจุด v_1 การใช้หุ่นยนต์จะมีค่าใช้จ่ายต่ำกว่าการใช้แรงงานคน เมื่อการผลิตเพิ่มมากขึ้นเรื่อยๆ จนถึงจุด v_2 การใช้เครื่องจักรแบบ Hard - Automation จะมีค่าใช้จ่ายต่ำสุด กราฟในรูป 2.2 แสดงถึงแนวโน้มโดยทั่วไป ส่วนตัวเลขข้อมูลที่แน่นอนจะขึ้นอยู่กับลักษณะของสินค้าที่ผลิต ถ้าหุ่นยนต์มีความสามารถมากขึ้นและราคาถูกลงอีก จะทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้ช่วง (v_1,v_2) ซึ่งเป็นช่วงที่การใช้หุ่นยนต์จะส่งผลให้มีค่าใช้จ่ายต่ำกว่าวิธีอื่นขยายกว้างออกไปอีกทั้ง 2 ด้าน

เพื่อที่จะแสดงให้เห็นถึงความแตกต่างอย่างชัดเจน ระหว่างเครื่องจักรอัตโนมัติแบบ Hard และ แบบ Soft จึงขอให้คำนิยามของหุ่นยนต์ในหนังสือเล่มนี้ไว้ดังนี้

คำนิยาม 2.1 หุ่นยนต์(Robot) คือ อุปกรณ์ทางกลที่ควบคุมด้วย Software โดยใช้ Sensor ตรวจจับปลายทำงาน (End Effortor) ของหุ่นยนต์ที่เคลื่อนที่ตามโปรแกรมในพื้นที่การทำงาน (workspace) เพื่อทำการเคลื่อนย้ายวัตถุ

2.การแบ่งประเภทหุ่นยนต์ (Robot Classification)

มีเกณฑ์ที่ใช้แบ่งหลายอย่างดังนี้

2.1 เทคโนโลยีการขับเคลื่อน (Drive Technologies)

พิจารณาจากต้นกำลังต้นกำลังที่ใช้ขับข้อต่อของหุ่นยนต์ ที่นิยมใช้มี 2 แบบ คือ การขับเคลื่อนด้วยไฟฟ้ากับไฮโดรลิค แขนกลส่วนใหญ่ ในปัจจุบันใช้การขับเคลื่อนด้วยไฟฟ้า ในรูปของ DC เซอร์โวมอเตอร์หรือ DC สเตปปิ้งมอเตอร์ ในกรณีการเคลื่อนย้ายวัตถุที่ต้องการความเร็วสูงเช่น เหล็กหลอมละลาย หรือ ชิ้นส่วนรถยนต์ มักนิยมใช้หุ่นยนต์ขับเคลื่อนด้วยไฮโดรลิค แต่ก็มีข้อเสียที่สำคัญคือ ขาดความสะอาด ซึ่งเป็นลักษณะสำคัญในงานประกอบชิ้นส่วนหลายชนิด

หุ่นยนต์ที่ขับเคลื่อนด้วยไฟฟ้าและไฮโดรลิคมักนิยมใช้ Tool แบบนิวเมติก เช่น Gripper ที่ใช้เปิด - ปิด เพื่อหยิบจับสิ่งของ เนื่องจากอากาศเป็นของไหลที่อัดตัวได้ ทำให้ Tool ที่ทำงานโดยใช้อากาศนี้ไม่หยิบจับวัตถุด้วยแรงที่มากดกินไป เหมือนกับ Gripper ที่ใช้กลไกเป็นวัตถุแข็งและไม่มี Sensor

2.2 รูปทรงของขอบเขตทำงาน (Work Envelope Geometries)

Tool ของแขนกลปกติจะติดตั้งอยู่ที่ข้อมือของแขนกล ดังนั้นขอบเขตทำงานสุทธิ (Gross Work Envelope) จึงหมายถึง ขอบเขตในปริภูมิสามมิติ ที่ข้อมือของแขนกลสามารถเคลื่อนที่ไปถึงยังตำแหน่งนั้นได้ เราจะเรียกแกนของ 3 ข้อต่อแรกของแขนกลว่าเป็นแกนหลัก (Major Axis) ซึ่งใช้ในการพิจารณาตำแหน่งของข้อมือ ส่วนแกนข้อต่อที่เหลือ เรียกว่า แกนรอง (Minor Axis) ใช้กำหนดลักษณะการวางตัวหรือหมุน (Orientation) ของ Tool ดังนั้นรูปทรงของขอบเขตการทำงานจึงพิจารณาได้จาก ลำดับชนิดของข้อต่อที่ใช้ใน 3 แกนแรก ข้อต่อมีหลายแบบ แต่ที่เป็นพื้นฐานและนิยมใช้ในอุตสาหกรรมมี 2 แบบ ดังตาราง 2.1

ตาราง 2.1 ชนิดของข้อต่อหุ่นยนต์

ชนิด	เครื่องหมาย	สัญลักษณ์	อธิบาย
ข้อต่อหมุน (Revolute)	R		การเคลื่อนที่แบบหมุนรอบแกน
ข้อต่อเลื่อน (Prismatic)	P		การเคลื่อนที่เชิงเส้นตามแนวแกน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้ง

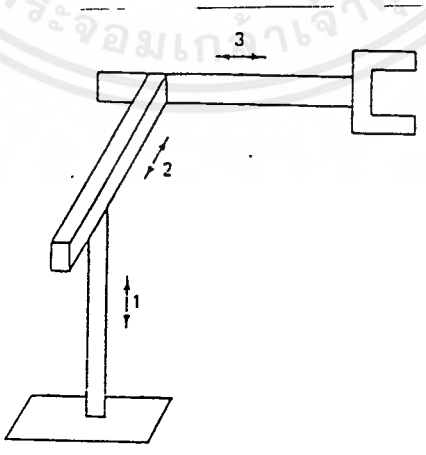
ข้อต่อ Revolute(R) ทำให้เกิดการหมุนรอบแกน ส่วนข้อต่อ Prismatic(P) ทำให้เกิดการเคลื่อนที่ไปตามแนวแกน การรวมกันของข้อต่อ 2 ชนิดนี้ใน 3 แกนหลัก ก่อให้เกิดรูปทรงขอบเขตการทำงานหลายแบบ ดังตัวอย่างในตาราง 2.2 ซึ่งแสดงเฉพาะแบบที่นิยมใช้กันในการวิเคราะห์การเคลื่อนที่ของแขนกล ข้อต่อ Revolute จะวิเคราะห์ได้ยากกว่าแบบ Prismatic ดังนั้นยิ่งมีข้อต่อแบบ Revolute มาก แขนกลก็จะยิ่งซับซ้อน

ตาราง 2.2 ขอบเขตการทำงานของหุ่นยนต์ที่ขึ้นอยู่กับแกนหลัก

หุ่นยนต์	แกน1	แกน2	แกน3	แกน4
คาร์ทีเซียน	P	P	P	0
ทรงกระบอก	R	P	P	1
ทรงกลม	R	R	P	2
SCARA	R	R	P	2
ข้อต่อหมุน	R	R	R	3

P = ข้อต่อเลื่อน R = ข้อต่อหมุน

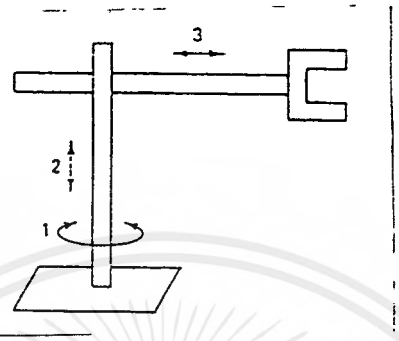
หุ่นยนต์ที่ธรรมดาที่สุดในตาราง 2.2 มีแกนหลักเป็นข้อต่อ Prismatic ทั้งหมด สัญลักษณ์ PPP เรียกว่าเป็นพิกัด คาร์ทีเซียน (Cartesian-Coordinate Robot) หรือ หุ่นยนต์พิกัดฉาก (Rectangular-Coordinate Robot) แสดงดังรูปที่ 2.3 จะเห็นได้ว่า ข้อมือจะเลื่อนขึ้น-ลง เข้า-ออก และเดินหน้า-หลัง ทำให้Work-envelope มีลักษณะเป็นกล่องสี่เหลี่ยมมุมฉาก



รูปที่ 2.3 หุ่นยนต์พิกัดคาร์ทีเซียน

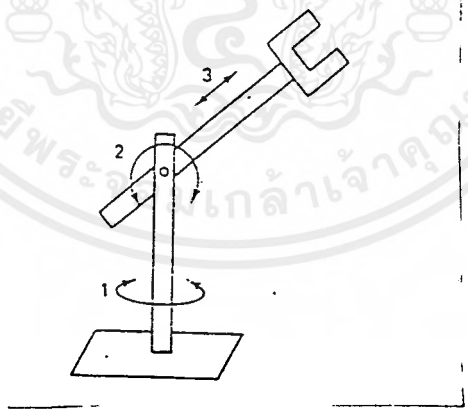
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์พิกัดทรงกระบอก (Cylindrical-Coordinate Robot) สัญลักษณ์ RPP แสดงดังรูปที่ 2.4 ข้อต่อ Revolute ทำให้แขนหมุนรอบแกนตั้งฉากกับฐาน ข้อต่อ Prismatic เลื่อนข้อมือขึ้น-ลงตามแนวตั้ง และเลื่อนเข้า-ออกตามแนวรัศมี ทำให้ Work-Envelope มีลักษณะเป็นทรงกระบอก



รูปที่ 2.4 หุ่นยนต์พิกัดทรงกระบอก

หุ่นยนต์พิกัดทรงกลม(Spherical - Coordinate Robot) สัญลักษณ์ RRP แสดงดังรูปที่ 2.5 ข้อต่อ Revolute ข้อแรกทำให้แขนหมุนรอบแกนแนวตั้งของฐาน ส่วนข้อที่สองทำให้หมุนขึ้น - ลงรอบแกนแนวนอนของไหล่ ส่วนข้อต่อ Prisatic ทำให้ข้อต่อเลื่อนเข้าออกตามแนวรัศมี Work Envelope จึงมีลักษณะเป็นทรงกลม

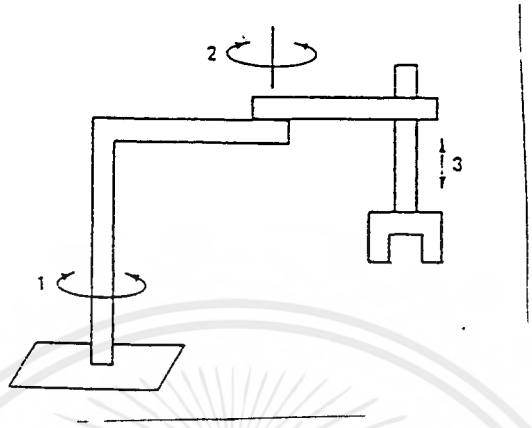


รูป 2.5 หุ่นยนต์พิกัดทรงกลม

หุ่นยนต์ SCARA (Selective Compliance Assembly Robot Arm) มีลักษณะคล้ายหุ่นยนต์พิกัดทรงกลมที่มีข้อต่อ Revolute 2 ข้อ และข้อต่อ Prismatic 1 ข้อ สัญลักษณ์ RRP แต่แกนทั้ง 3 ของ SCARA จะอยู่ในแนวตั้งดังรูปที่ 2.6 Revolute ข้อที่สองทำให้ปลายแขนหมุนรอบแกนตั้งของข้อศอก ซึ่งทั้ง 2 ข้อนี้จะควบคุมการเคลื่อนที่ในระนาบแนวนอน ส่วนการ

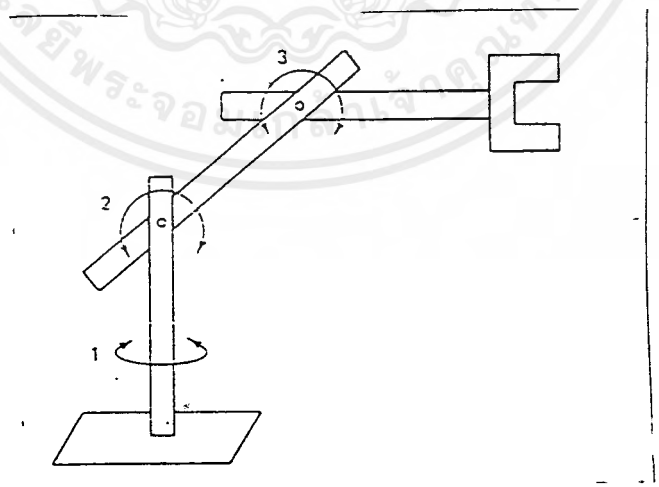
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เคลื่อนที่ในแนวตั้งได้จากข้อต่อ Prismatic ที่เลื่อนข้อมือขึ้นและลง ภาพตัดขวางในแนวนอนของ Work Envelope ค่อนข้างซับซ้อนขึ้นอยู่กับลิมิตในการเคลื่อนที่ของ 2 แกนแรก



รูป 2.6 หุ่นยนต์ SCARA

หุ่นยนต์ข้อต่อหมุน (Articulated - Coordinate Robot or Revolute Robot) สัญลักษณ์ RRR โดย 3 ข้อต่อหลักจะเป็นแบบ Revolute แสดงดังรูปที่ 2.7 หุ่นยนต์แบบนี้มลักษณะคล้ายแขนมนุษย์มากที่สุดโดยข้อต่อแรกทำให้แขนหมุนรอบแกนตั้งของฐาน ข้อต่อที่สองทำให้แขนหมุนขึ้นลงรอบแกนนอนของไหล่ และข้อต่อที่สามทำให้ปลายแขนหมุนขึ้นลง รอบแกนแนวนอนของศอก การเคลื่อนเช่นนี้จึงทำให้ Word Envelope มีรูปร่างซับซ้อน



รูป 2.7 หุ่นยนต์ข้อต่อหมุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 วิธีควบคุมการเคลื่อนที่ (Motion Control Methods)

เกณฑ์ที่ใช้แบ่งประเภทของหุ่นยนต์ อีกแบบหนึ่งก็คือ วิธีที่ใช้ในการควบคุม การเคลื่อนที่ของ Tool การเคลื่อนที่แบบพื้นฐานมี 2 แบบ แสดงในตาราง 2.3 แบบแรก คือ การเคลื่อนที่แบบจุดต่อจุด(Point To Point) โดย Tool จะเคลื่อนที่ไปยังจุดที่ไม่ต่อกันในพื้นที่การทำงานตามลำดับ เส้นทางระหว่างจุดเหล่านี้ผู้ใช้ไม่สามารถควบคุมได้ การเคลื่อนที่แบบจุดต่อจุดใช้ในงานที่ไม่ต้องการเส้นทางที่ต่อเนื่องกัน เช่น การเชื่อมจุด

ตาราง 2.3 ชนิดของการควบคุมการเคลื่อนที่ของหุ่นยนต์

วิธีการควบคุม	การประยุกต์ใช้
จุดต่อจุด (Point to Point)	เชื่อมจุด , หยิบจับ , เคลื่อนย้าย , บรรจุสิ่งของ
เส้นต่อเนื่อง (Continuous Path)	พ่นสี , เชื่อม , ทากาว

อีกแบบหนึ่ง คือ การเคลื่อนที่แบบเส้นทางต่อเนื่อง (Continuous-Path) หรือ เรียกว่าแบบควบคุมเส้นทาง (Controlled -Path) แบบนี้ Tool จะเคลื่อนที่ไปตามเส้นทางในปริภูมิ 3 มิติตามที่กำหนดไว้ ซึ่งความเร็วในการเคลื่อนที่อาจจะมีการเปลี่ยนแปลงระหว่างการเคลื่อนที่ด้วยก็ได้ การเคลื่อนที่แบบนี้นำไปใช้ในงานประเภท พ่นสี เชื่อมรอยต่อและการทากาว

3. ลักษณะเฉพาะของหุ่นยนต์ (Robot Specifications)

นอกจากเกณฑ์ต่างๆที่ใช้ในการแบ่งประเภทของหุ่นยนต์ ดังที่กล่าวมาแล้ว ยังมีลักษณะบางอย่างอีก ที่ช่วยให้ผู้ใช้ในการระบุเลือกหุ่นยนต์ที่ต้องการ ลักษณะเหล่านี้แสดงโดยตารางที่2.4

ตาราง2.4 ลักษณะเฉพาะของหุ่นยนต์

ลักษณะ	หน่วย
จำนวนแกน	-
ความสามารถในการยกน้ำหนัก	kg
ความเร็วสูงสุด , เวลาวัฏจักร	mm/sec
ระยะเอื้อมถึงและสโตรค	mm
การหมุนของเครื่องมือ	deg
การซ้ำตำแหน่งเดิม	mm
ความละเอียดและความแม่นยำ	mm
สภาพแวดล้อมในการทำงาน	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 จำนวนแกน

โดยปกติ 3แกนแรกหรือแกนหลัก ใช้ในการกำหนดตำแหน่งของ Tool ส่วนแกนที่เหลือใช้กำหนดการวางตัว-การหมุนของ Tool ดังแสดงในตาราง 2.5 เนื่องจากแขนกลทำงานในปริภูมิ สามมิติ หุ่นยนต์ 6 แกน จึงเป็นแบบที่เพียงพอแล้วในการกำหนดตำแหน่งและการหมุนของ Tool ในพื้นที่ทำงาน กลไกที่ใช้ในการเปิด-ปิด นิ้วหรือทำงานอย่างอื่นของ Tool ไม่ถือว่าเป็นแกนอิสระ เพราะ มันไม่มีผลต่อตำแหน่งและการหมุนของ Tool หุ่นยนต์ในโรงงานอุตสาหกรรมโดยทั่วไปมีตั้งแต่ 4-6 แกนแต่อาจจะมีมากกว่า 6 แกนก็ได้ แกนที่เพิ่มมานั้น ใช้ในการหลบหลีกสิ่งกีดขวางในพื้นที่ทำงาน

ตาราง 2.5 แกนของหุ่นยนต์

1-3	แกนหลัก	กำหนดตำแหน่งข้อมือ
4-6	แกนรอง	กำหนดการหมุนเครื่องมือ
7-n	แกนเผื่อเหลือ	หลบหลีกสิ่งกีดขวาง

3.2 ความสามารถและความเร็ว (Capacity and Speed)

ความสามารถในการยกน้ำหนัก (Load -Carrying Capacity) จะแตกต่างกันไปตามหุ่นยนต์แต่ละตัว เช่น หุ่นยนต์ในห้องทดลอง อาจยกน้ำหนักได้เพียง 2-3 kg. ในขณะที่หุ่นยนต์อุตสาหกรรมบางชนิดยกน้ำหนักได้ถึง 3-4 ตัน เป็นต้น

ความเร็วปลายอุปกรณ์ (Tool -Tip Speed)ก็มีค่าสูงสุดต่างกันไปตามหุ่นยนต์แต่ละแบบนอกจากนี้ยังมีการวัดความเร็วของหุ่นยนต์อีกแบบ คือ เวลาวัฏจักร (Cycle Time) ซึ่งเป็นเวลาที่ใช้ในการเคลื่อนที่ครบ 1 รอบการทำงาน

เนื่องจากความสามารถในการยกน้ำหนักและความเร็วสูงสุด มีค่าแตกต่างกันไปตามแต่ละชนิด การเลือกใช้หุ่นยนต์ในแต่ละแบบ จึงต้องคำนึงถึง ลักษณะเหล่านี้ อย่างระมัดระวัง งานบางอย่างอาจไม่ต้องการความสามารถในการยกน้ำหนักมากมายนัก บางอย่างอาจต้องการความแม่นยำมากกว่า ความเร็ว เป็นต้น

3.3 ระยะที่เอื้อมถึงและสโตรค (Reach and Stroke)

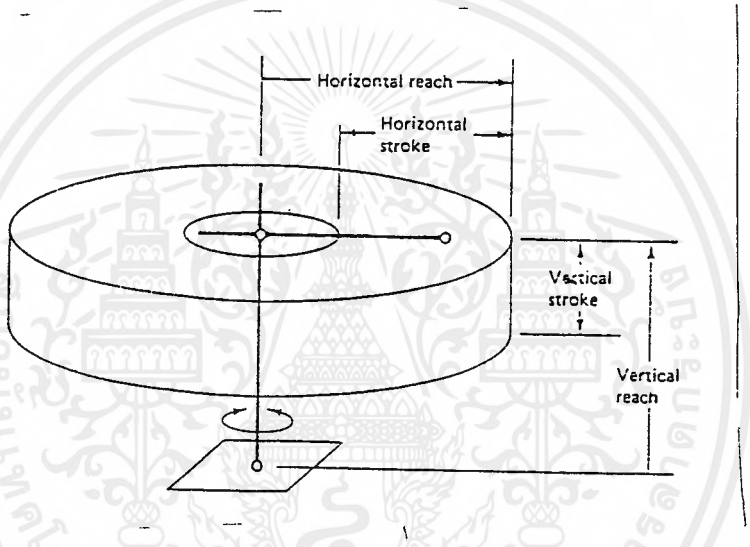
ระยะที่เอื้อมถึงและสโตรค เป็นการวัดขนาดของ Work Envelope อย่างคร่าว ๆ ระยะที่เอื้อมถึงในแนวนอน (Horizontal Reach) คือระยะทางตามแนวรัศมีที่มากที่สุดที่ข้อมือสามารถไปถึง โดยวัดจากแกนแนวตั้งที่ทำให้หุ่นยนต์ สโตรคแนวนอน (Horiontal Stroke) คือระยะทางตามแนวรัศมีทั้งหมด ที่ข้อมือสามารถไปได้ ดังนั้น Horizontal Reach - Horiontal Stroke ก็คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระยะทางตามแนวรัศมีที่น้อยสุดที่ ข้อมือสามารถไปถึงได้โดยวัดจากแกนฐาน ซึ่งมีค่าเป็น บวก เสมอ จึงได้ว่า Stroke <= Reach

ตัวอย่างเช่น Horizontal Reach ของหุ่นยนต์พิกัดทรงกระบอก ก็คือรัศมีของทรงกระบอกอันนอกของ Work Envelope ในขณะที่ Horizontal Stroke คือ ผลต่างระหว่างรัศมีของทรงกระบอกอันในและอันนอก ดังแสดงในรูป 2.8

ในทำนองเดียวกัน Verical Reach ก็คือ ระดับความสูงที่มากที่สุดที่ข้อมือสามารถขึ้นไปถึงโดยวัดจากพื้นทำงาน Vertical Stroke ก็คือ ระยะทางในแนวตั้งทั้งหมดที่ข้อมือสามารถขึ้น-ลงได้



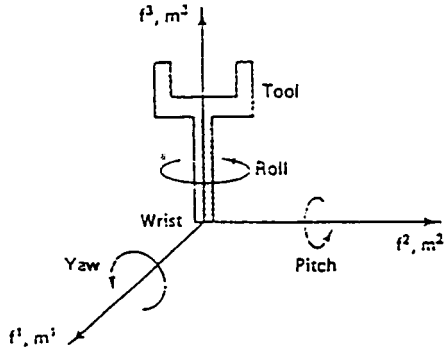
รูป2.8 ระยะที่เอื้อมถึงและสโตรคของหุ่นยนต์ทรงกระบอก

3.4 การหมุนของ Tool (Tool Orientation)

ในขณะที่ 3แกนหลักใช้ในการพิจารณารูปทรง Work Envelope แกนที่เหลือจึงใช้ในการพิจารณาการหมุนของ Tool ถ้ามีแกนของอิสระ 3 แกน ก็ทำให้ Tool หมุนได้ อย่างไม่มีจำกัดในขอบเขต 3 มิติ การระบุลักษณะการหมุนของ Tool จะใช้ระบบ Yaw-Pitch-Roll(YPR) ดังรูปที่ 2.9

ในการระบุการหมุน จะติดตั้งโครงพิกัด Tool M ={m^1,m^2,m^3} ไว้ที่ Tool และโครงนี้จะเคลื่อนที่ไปพร้อมกับ Tool ดังแสดงในรูป 1.8 แกน m^3 อยู่ในแนวเดียวกับแกนของ Tool และชี้ออกจากข้อมือ แกน m^2 ขนานกับทิศทางที่ปลายนิ้วของ Tool เลื่อนปิด-เปิด ส่วนแกน m^1 เป็นไปตามกฎมือขวาของโครงพิกัด M

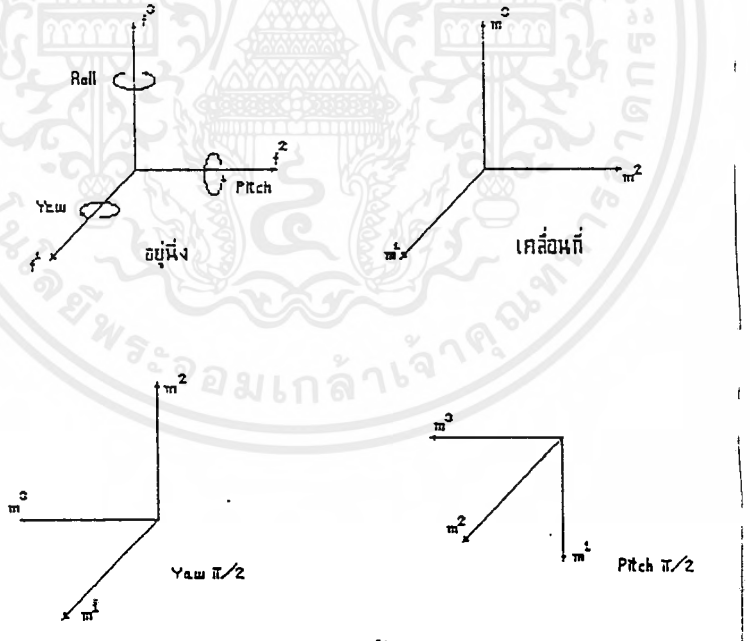
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



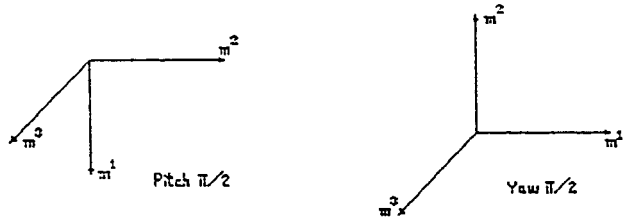
รูป 2.9 Yaw,Pitch และ Row ของ Tool

การเคลื่อนที่แบบ Yaw,Pitch และ Roll จะกระทำตามลำดับกับ แกนที่อยู่นิ่งเมื่อเริ่มต้น โครงพิกัดเคลื่อนที่ของ Tool M จะทับอยู่กับโครงพิกัดอยู่นิ่งของข้อมือ F = {f^1,f^2,f^3} ซึ่งติดอยู่ที่ส่วนปลายของปลายแขน(Forearm) Yaw คือ การหมุน Tool รอบแกนข้อมือ f^1 Pitch คือ การหมุน Tool รอบแกนข้อมือ f^2,Roll คือการหมุน Tool รอบแกนข้อมือ f^3 ในแต่ละกรณี มุมบวกเกิดจากการหมุนทวนเข็มนาฬิกา โดยมองจากปลายแกนกลับเข้าหาจุดกำเนิด

ลำดับการหมุน Yaw,Pitch,Roll มีความสำคัญมาก เพราะมีผลต่อลักษณะการวางตัวสุดท้ายของ Tool ตัวอย่างเช่น หมุน Yaw ไป $\pi/2$ ตามด้วย Pitch $\pi/2$ จะมีลักษณะสุดท้ายต่างกับการหมุน Pitch ไป $\pi/2$ แล้วตามด้วย Yaw $\pi/2$ เป็นต้น



หมุน yaw ตามด้วย pitch



หมุน pitch ตามด้วย yaw

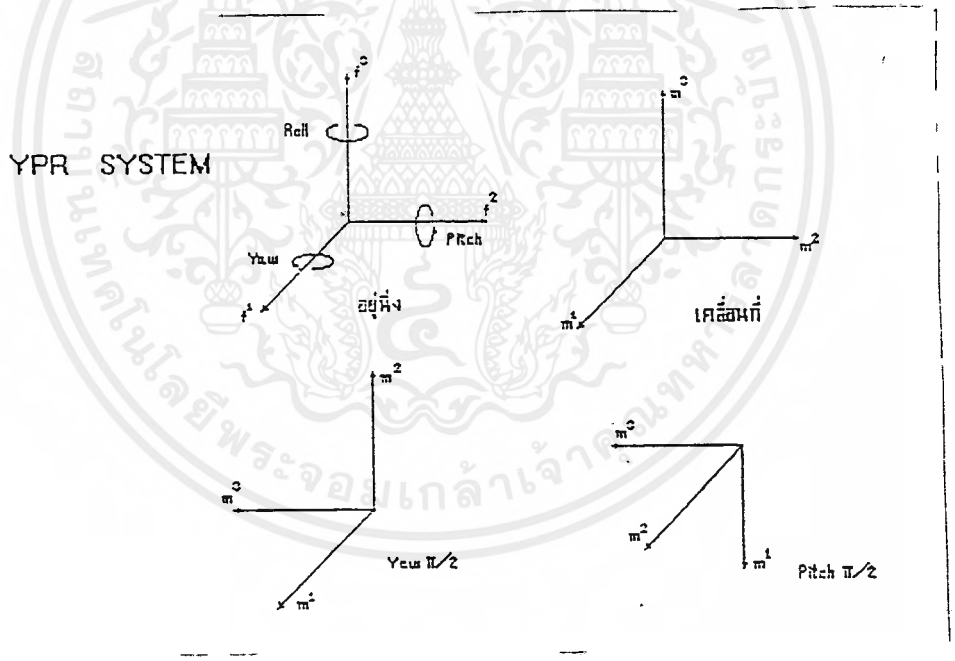
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนด YPR ตามลำดับการหมุน สรุปได้ดังตาราง 2.6

ตาราง 2.6 การเคลื่อนที่แบบ Yaw,PitchและRoll

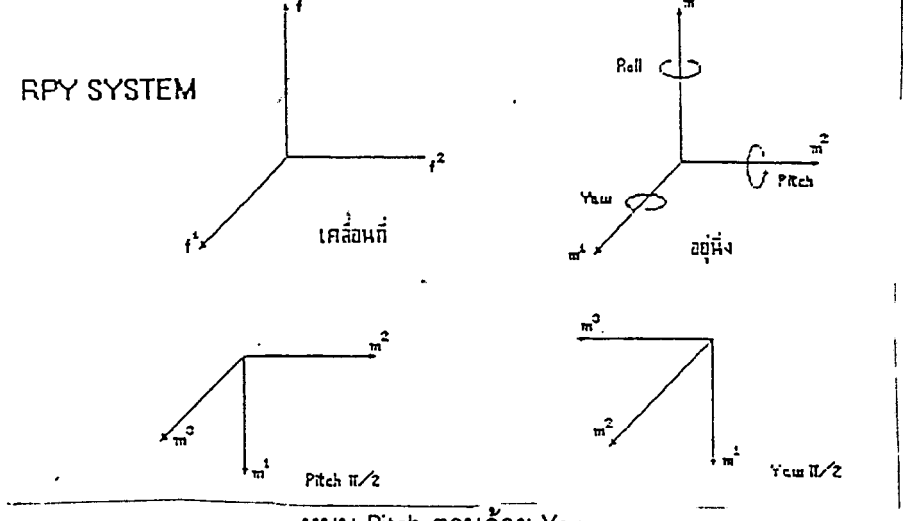
ขั้นตอน	อธิบาย	แกน
1	Yaw	f^1
2	Pitch	f^2
3	Roll	f^3

อีกวิธีหนึ่งที่ใช้ระบุการหมุน คือ ระบุลำดับการหมุนย้อนลำดับกับ YPR และคิดเทียบกับโครงพิกัดเคลื่อนที่ของ Tool M แทนที่โครงพิกัดอยู่นิ่งของข้อมือ F เริ่มแรก Roll คือ การหมุนรอบแกน m^3,Pitch คือ การหมุนรอบแกน m^2 และสุดท้าย Yaw คือ การหมุนรอบแกน m^1 การคิดลักษณะนี้จะให้ผลลัพธ์เหมือนกับแบบ YPR และเรียกว่าเป็นระบบ RPY ซึ่งลำดับการหมุนตามระบบ RPY จะมองเข้าใจได้ง่ายกว่า โดยเฉพาะในกรณีที่มุมไม่เป็นผลคูณของ $\pi/2$



หมุน Yaw ตามด้วย Pitch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



หมุน Pitch ตามด้วย Yaw

รูป 2.11 ระบบ YPR และ RPY

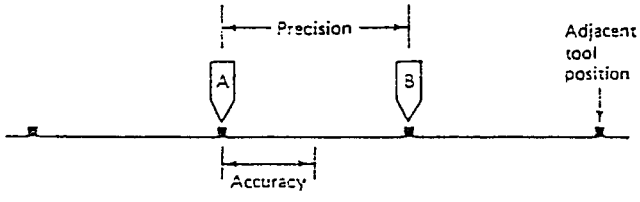
คำนิยาม 2.2 ข้อมือทรงกลม(Spherical Wrist) หุ่นยนต์จะมีข้อมือทรงกลมก็ต่อเมื่อแกนที่ใช้หมุน Tool ตัดที่จุดเดียวกัน

ข้อมือในรูป 2.9 เป็นตัวอย่างของข้อมือทรงกลม เพราะ 3 แกนที่ใช้หมุน$\{m^1, m^2, m^3\}$ ตัดที่จุดเดียวกัน ถึงแม้หุ่นยนต์จะมีแกนน้อยกว่า 3 แกนที่ใช้หมุน Tool ก็ยังถือว่าเป็นข้อมือทรงกลม แต่ก็จะมีข้อจำกัดในการหมุน ไม่ได้เป็นทรงกลมที่สมบูรณ์ หุ่นยนต์-อุตสาหกรรม ส่วนใหญ่จะถูกออกแบบให้มีข้อมือแบบทรงกลมนี้

3.5 ความสามารถในการซ้ำตำแหน่งเดิม,ความละเอียดและความแม่นยำ (Repeatability,Precision and Accuracy)

Repeatability เป็นลักษณะที่สำคัญอีกประการหนึ่ง กล่าวคือเป็นการวัดความสามารถของหุ่นยนต์ ในการวางตำแหน่งของปลายอุปกรณ์ในตำแหน่งเดียวกันซ้ำหลายๆครั้ง เนื่องจากสาเหตุต่างๆเช่น Backlash ในเฟืองและความยืดหยุ่น(Flexibility) ของส่วนต่างๆซึ่งมักทำให้เกิดความผิดพลาดใน Repeatability

Precision ความละเอียด เป็นการวัดระยะห่าง(Resolution) ระหว่างตำแหน่งต่างๆในขอบเขตการทำงานที่สามารถจะกำหนดให้เครื่องมือเคลื่อนที่ไปได้ เช่น ถ้าปลายเครื่องมืออยู่ที่ตำแหน่ง A ในรูป 2.12 ตำแหน่งต่อไปที่ใกล้ที่สุดที่สามารถเคลื่อนที่ไปอยู่ได้ คือ B ดังนั้นความละเอียดคือ ระยะห่างระหว่างจุด A และ B



รูป 2.12 ตำแหน่ง Tool ที่อยู่ติดกัน

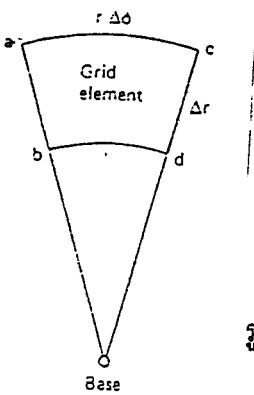
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทั่วไปในระนาบ 3 มิติ ปลายเครื่องมือของแขนกลอาจจะถูกกำหนดให้อยู่ในตำแหน่งใดก็ได้โดยใช้ตารางกริด 3 มิติของจุดที่อยู่ในขอบเขตการทำงาน ตามปกติแล้วกริดเหล่านี้มีรูปแบบไม่คงที่ ความละเอียดรวมของหุ่นยนต์คือ ระยะห่างที่มากที่สุดระหว่างจุดที่อยู่ข้างเคียงกันในกริด ตัวอย่างเช่น จุดกริดของหุ่นยนต์พิกัดคาร์ทีเซียน จะมีจุดข้างเคียงเพียง 8 จุดในระนาบแนวนอนบวกกับ 9 จุดกริดข้างเคียงในระนาบบน และอีก 9 จุดในระนาบข้างใต้ ข้อดีข้อหนึ่งของหุ่นยนต์ชนิดนี้คือ ความละเอียดมีค่าคงที่ตลอดขอบเขตการทำงาน สภาวะของแกนใดๆไม่มีผลต่อความละเอียดของแกนอื่นๆส่วนกรณีหุ่นยนต์ชนิดอื่นแสดงไว้ในตาราง 2.7 โดยแบ่งความละเอียดรวมเป็น 2 ส่วน คือ ในแนวตั้ง กับ แนวนอน

ตาราง 2.7 ความละเอียดในแนวนอนและแนวตั้ง

แกน	ชนิด	หน้าที่
ชนิดหุ่นยนต์	ความละเอียดแนวนอน	ความละเอียดแนวตั้ง
คาร์ทีเซียน	คงที่	คงที่
ทรงกระบอก	ลดลงตามแนวรัศมี	คงที่
ทรงกลม	ลดลงตามแนวรัศมี	ลดลงตามแนวรัศมี
SCARA	เปลี่ยนแปลง	คงที่
ข้อต่อหมุน	เปลี่ยนแปลง	เปลี่ยนแปลง

ในกรณีของ หุ่นยนต์พิกัดทรงกระบอก ความละเอียดแนวนอนมีค่ามากสุด บริเวณพื้นผิวด้านในของขอบเขตการทำงาน และน้อยที่สุดบริเวณพื้นผิวภายนอก เนื่องจากความยาวส่วนโค้งที่เกิดจากหมุนข้อต่อฐานไป จะเป็นสัดส่วนโดนตรงกับรัศมี r ดังแสดงในรูป 2.13 จะเห็นได้ว่า กริดในแนวนอนไม่เป็นสี่เหลี่ยมมุมฉาก และกริดที่อยู่ใกล้ผิวนอกจะมีขนาดใหญ่กว่ากริดที่อยู่ใกล้ผิวในของขอบเขตการทำงาน



รูป 2.13 ความละเอียดแนวนอนของหุ่นยนต์ทรงกระบอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความละเอียดเชิงรัศมี คือ Δr ส่วนความละเอียดเชิงมุมที่หมุนรอบฐานจะแสดงอยู่ในเทอมของความยาวส่วนโค้ง $r\Delta\phi$ โดยมีค่ามากสุดเป็น $R\Delta\phi$ เมื่อ R คือระยะที่เอื้อมถึงในแนวนอนของหุ่นยนต์(Horizontal Reach) ดังนั้น ความละเอียดแนวนอนรวมในกรณีนี้ประมาณได้เป็น

$$\Delta h \approx [(\Delta r)^2 + (R\Delta\phi)^2]^{1/2} \qquad (2.1\text{-}3\text{-}1)$$

ถ้าสมมุติ $\Delta\phi$ และ Δr มีค่าเล็กมาก มุม bac ในรูป 2.13 จะประมาณได้ $\pi/2$ และถ้า Δz แทนความละเอียดแนวตั้งของหุ่นพิกัดทรงกระบอก ดังนั้นความละเอียดรวมทั้งหมดของหุ่นยนต์คือ

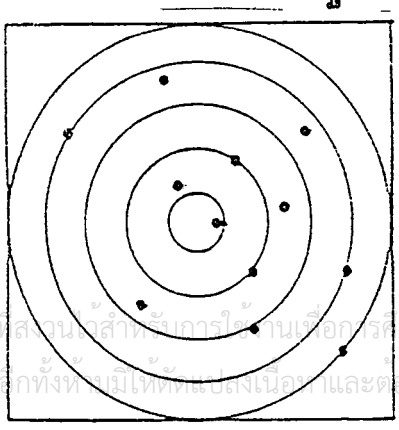
$$\Delta P \approx [(\Delta r)^2 + (R\Delta\phi)^2 + (\Delta z)^2]^{1/2} \qquad (2.1\text{-}3\text{-}2)$$

หุ่นยนต์พิกัดทรงกลม จะมีความละเอียดแนวนอนและแนวตั้งมากที่สุดบริเวณผิวด้านใน และน้อยที่สุดบริเวณผิวด้านนอกของขอบเขตการทำงาน หุ่นยนต์ SCARA ความละเอียดแนวนอนจะมีค่าเปลี่ยนแปลง ความละเอียดแนวตั้งจะคงที่ ส่วนหุ่นยนต์ข้อต่อหมุน(Articulated Robot) ความละเอียดทั้งแนวนอนและแนวตั้งจะมีค่าไม่คงที่

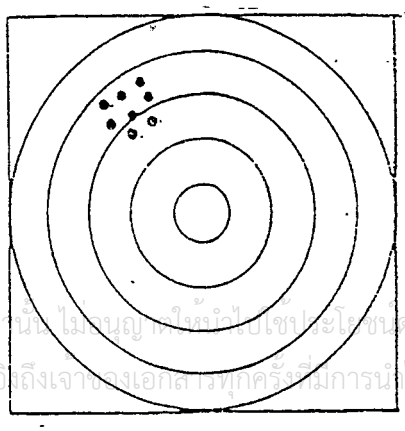
อีกวิธีหนึ่งในการระบุความละเอียดก็คือระบุให้อยู่ในรูปของแต่ละข้อต่อ วิธีนี้จะวิเคราะห์ได้ง่ายกว่า แต่ก็นำไปใช้ประโยชน์ได้น้อยกว่าด้วย ความละเอียดหรือระยะห่าง (Resolution) ของแต่ละข้อต่อขึ้นอยู่กับระบบขับเคลื่อน,เซ็นเซอร์จับตำแหน่งและกลไกส่งกำลัง รวมไปถึงเฟือง,สายพาน ตัวอย่างเช่น สัญญาณตำแหน่งถูกสร้างขึ้นภายในคอมพิวเตอร์ควบคุมแล้วส่งไปยังระบบควบคุมตำแหน่งย้อนกลับแบบอนาลอกภายนอก โดยผ่านตัวแปลงสัญญาณดิจิตอลเป็นอนาลอกขนาด n-bit (DAC) ถ้าเพลาเคลื่อนที่อยู่ในช่วง ϕmin ถึง ϕmax ดังนั้นความละเอียดของเพลา $\Delta\phi_{DAC}$ กรณีนี้เป็น

$$\Delta\phi_{DAC} = (\phi\text{max} - \phi\text{min}) / 2^n \qquad \text{degree per count}$$

ความแม่นยำของแขนกลเป็นการวัดความสามารถในการวางปลายอุปกรณ์ให้ตรงกับตำแหน่งที่ได้กำหนดไว้ภายในขอบเขตการทำงาน ความแม่นยำนี้วิเคราะห์ได้ยากกว่าความละเอียด ขึ้นอยู่กับหลายสาเหตุเช่น Backlash ของเฟือง การเสียรูปร่างอย่างยืดหยุ่น(Elastic Deformation) ของ Link อันเนื่องมาจากการรับโหลด เมื่อหุ่นยนต์ทำงานเป็นระยะเวลานานๆก็จะเกิดความคลาดเคลื่อนสะสมเพิ่มมากขึ้นเรื่อยๆ จึงควรปรับรักษาความแม่นยำเป็นระยะๆ โดยการรีเซตไปที่ตำแหน่งฮาร์ดโฮมมาตรฐานที่ใช้ลิมิตสวิทช์เป็นเซ็นเซอร์

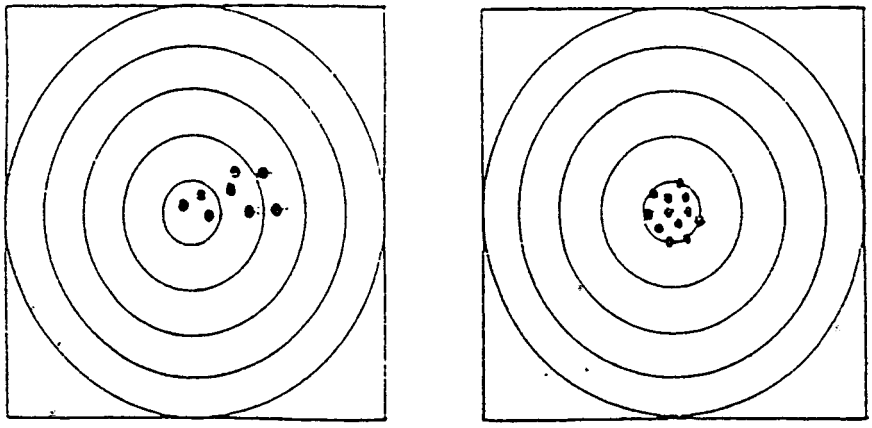


POOR ACCURACY



POOR ACCURACY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.14 เปรียบเทียบ Accuracy และ Repeatability

ถึงแขนกลจะมีความแม่นยำน้อยกว่าเครื่องจักรอัตโนมัติที่ใช้เฉพาะงาน แต่มันก็มีความยืดหยุ่นในการใช้งานมากกว่า ในทางปฏิบัติมักจะมีการใช้เซนเซอร์ติดอยู่ที่ภายนอกเพื่อชดเชยความไม่แน่นอนของสภาพแวดล้อมและหาตำแหน่งที่แท้จริงของแขนกล

3.6 สภาพแวดล้อมในการทำงาน (Operating Environment)

สภาพแวดล้อมในการทำงานขึ้นอยู่กับลักษณะของงานที่ทำ หุ่นยนต์ส่วนใหญ่จะถูกนำไปใช้งานในสถานที่ที่อันตรายหรือเป็นพิษต่อสุขภาพ เช่นการขนถ่ายวัสดุที่มีรังสี การพ่นสี การเชื่อม และบรรจุ และการเทเตาหลอม ในงานเช่นนี้หุ่นยนต์จะถูกออกแบบให้ทำงานที่อุณหภูมิสูงและสกปรก ในกรณีของการพ่นสี แขนกลอาจจะมีที่หุ้มเพื่อป้องกันละอองสีมาจับติดบริเวณข้อต่อ

สภาพแวดล้อมที่เป็นพิเศษอีกกรณีคือ หุ่นยนต์ที่ต้องการความสะอาดมาก เช่นในอุตสาหกรรมผลิตเซมิคอนดักเตอร์ ใช้เคลื่อนย้ายวัสดุพิเศษ เช่น Silicon Wafers และ Photomasks เป็นต้น สภาพแวดล้อมในการทำงานนี้ต้องมีการควบคุมอุณหภูมิ,ความชื้นและการถ่ายเทอากาศอย่างระมัดระวัง หุ่นยนต์ที่ใช้จึงต้องออกแบบให้มีความสะอาดมากๆ เพื่อป้องกันไม่ให้ชิ้นงานสกปรก หุ่นยนต์บางชนิดอาจจะมีตัวดูดอนุภาคที่หลุดมาจากการเสียดสีของผิวสัมผัส เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Visual BASIC for Windows (VB/Win)

3.1 การสร้างกราฟฟิก

เมื่อเราพูดถึงระบบปฏิบัติการวินโดวส์สิ่งหนึ่งที่ลืมเสียมิได้ ก็คือคำว่า **กราฟฟิก** (Graphic) เพราะถ้าหากไม่ใช่ความต้องการสภาพแวดล้อมการทำงานที่เป็นกราฟฟิกแล้วละก็ ระบบปฏิบัติการวินโดวส์ ก็คงจะไม่เป็นที่แพร่หลาย และได้รับความนิยมเป็นอย่างมากในปัจจุบัน ดังนั้นเมื่อศึกษาเรียนรู้ VB/Win ซึ่งเป็นเครื่องมือ สำหรับช่วยในการพัฒนางานที่สามารถรันภายใต้เอนไวรอนเมนท์ของวินโดวส์ได้แล้ว ถ้าไม่กล่าวถึงการสร้างกราฟฟิกก็แสดงว่า เรายังไม่ได้นำความสามารถอันโดดเด่นของ VB/Win มาใช้อย่างจริงจัง ทั้งนี้เนื่องจาก VB/Win ประกอบด้วยเครื่องมือต่าง ๆ มากมาย ที่ทำให้การสร้างสรรค์งานกราฟฟิกสามารถกระทำได้ง่ายและรวดเร็วมากยิ่งขึ้น

สำหรับคอนโทรลกราฟฟิกของ VB/Win ประกอบด้วย picture box,image,line และ shape โดยที่ 2 คอนโทรลแรก จะเป็นคอนโทรลที่สามารถใช้แสดงผลไฟล์กราฟฟิก หรืออิมเมจที่มีนามสกุล .bmp, .wmf, .ico หรือ .dib ได้ และนอกจากนี้ ยังสามารถวาดกราฟฟิกใด ๆ โดยการใช้คำสั่งเกี่ยวกับกราฟฟิกลงในคอนโทรลเหล่านี้ได้ ส่วน 2 คอนโทรลสุดท้าย เป็นคอนโทรลที่มีเฉพาะคุณสมบัติเพื่อกำหนดลักษณะการแสดงผลเท่านั้น โดยจะไม่มีเหตุการณ์สำหรับตอบสนองต่อการกระทำหรือวิธีสำหรับควบคุมการทำงานของคอนโทรลเหล่านี้ได้ โดยปกติเราจะใช้คอนโทรล line และ shape ในการวาดรูปเส้นตรง และรูปทรงเรขาคณิต เช่น สี่เหลี่ยม วงกลม วงรีรูปไข่ หรือ สี่เหลี่ยมปลายมน เป็นต้น ลงในฟอร์มเพื่อเป็นการตบแต่งหน้าตาของฟอร์มให้ดูดีขึ้นเท่านั้น

สำหรับกราฟฟิกที่นำมาใช้ใน Project นี้นั้น ใช้คอนโทรลกราฟิกประเภท picture box และ image ดังนั้นจะขออธิบายถึง 2 หัวข้อนี้เท่านั้น

1. คอนโทรล image

คอนโทรล image เป็นคอนโทรลช่วยในการแสดงผลกราฟฟิกหรือไฟล์กราฟฟิกต่างๆ เช่นเดียวกับคอนโทรล picture box แต่คอนโทรล image จะสามารถทำงานหรือวาดกราฟฟิกภายในคอนโทรลได้รวดเร็วกว่า และที่สำคัญมีการใช้ทรัพยากรของระบบน้อยกว่าอีกด้วย แต่เนื่องจากคอนโทรล image ใช้ทรัพยากรค่อนข้างน้อย ดังนั้นจึงสนับสนุนคุณสมบัติ เหตุการณ์ และ วิธีน้อยกว่า คอนโทรล picture box ในการเลือกใช้คอนโทรลทั้งสอง ต้องคำนึงถึงความต้องการที่จะให้คอนโทรลตอบสนองต่อผู้ใช้ และพิจารณาถึงความสามารถของคอนโทรลเหมาะสมกับงานหรือไม่ ทั้งนี้เพื่อให้สามารถเลือกใช้คอนโทรลได้อย่างเหมาะสม เพราะจะทำให้ประหยัดทรัพยากรของระบบ และช่วยให้แอพพลิเคชันสามารถทำงานได้รวดเร็วขึ้น ซึ่งโดยส่วนใหญ่มักจะนำคอนโทรล image มาใช้ในการสร้าง ไอคอนในทูลบาร์หรือทูลบอกซ์ ทั้งนี้เนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Visual BASIC for Windows (VB/Win)

3.1 การสร้างกราฟฟิก

เมื่อเราพูดถึงระบบปฏิบัติการวินโดวส์สิ่งหนึ่งที่ลืมเสียมิได้ ก็คือคำว่า **กราฟฟิก** (Graphic) เพราะถ้าหากไม่ใช่ความต้องการสภาพแวดล้อมการทำงานที่เป็นกราฟฟิกแล้วละก็ ระบบปฏิบัติการวินโดวส์ ก็คงจะไม่เป็นที่แพร่หลาย และได้รับความนิยมเป็นอย่างมากในปัจจุบัน ดังนั้นเมื่อศึกษาเรียนรู้ VB/Win ซึ่งเป็นเครื่องมือ สำหรับช่วยในการพัฒนางานที่สามารถรันภายใต้เอนไวรอนเมนท์ของวินโดวส์ได้แล้ว ถ้าไม่กล่าวถึงการสร้างกราฟฟิกก็แสดงว่า เรายังไม่ได้นำความสามารถอันโดดเด่นของ VB/Win มาใช้อย่างจริงจัง ทั้งนี้เนื่องจาก VB/Win ประกอบด้วยเครื่องมือต่าง ๆ มากมาย ที่ทำให้การสร้างสรรค์งานกราฟฟิกสามารถกระทำได้ง่ายและรวดเร็วมากยิ่งขึ้น

สำหรับคอนโทรลกราฟฟิกของ VB/Win ประกอบด้วย picture box,image,line และ shape โดยที่ 2 คอนโทรลแรก จะเป็นคอนโทรลที่สามารถใช้แสดงผลไฟล์กราฟฟิก หรืออิมเมจที่มีนามสกุล .bmp, .wmf, .ico หรือ .dib ได้ และนอกจากนี้ ยังสามารถวาดกราฟฟิกใด ๆ โดยการใช้คำสั่งเกี่ยวกับกราฟฟิกลงในคอนโทรลเหล่านี้ได้ ส่วน 2 คอนโทรลสุดท้าย เป็นคอนโทรลที่มีเฉพาะคุณสมบัติเพื่อกำหนดลักษณะการแสดงผลเท่านั้น โดยจะไม่มีเหตุการณ์สำหรับตอบสนองต่อการกระทำหรือวิธีสำหรับควบคุมการทำงานของคอนโทรลเหล่านี้ได้ โดยปกติเราจะใช้คอนโทรล line และ shape ในการวาดรูปเส้นตรง และรูปทรงเรขาคณิต เช่น สี่เหลี่ยม วงกลม วงรีรูปไข่ หรือ สี่เหลี่ยมปลายมน เป็นต้น ลงในฟอร์มเพื่อเป็นการตบแต่งหน้าตาของฟอร์มให้ดูดีขึ้นเท่านั้น

สำหรับกราฟฟิกที่นำมาใช้ใน Project นี้นั้น ใช้คอนโทรลกราฟิกประเภท picture box และ image ดังนั้นจะขออธิบายถึง 2 หัวข้อนี้เท่านั้น

1. คอนโทรล image

คอนโทรล image เป็นคอนโทรลช่วยในการแสดงผลกราฟฟิกหรือไฟล์กราฟฟิกต่างๆ เช่นเดียวกับคอนโทรล picture box แต่คอนโทรล image จะสามารถทำงานหรือวาดกราฟฟิกภายในคอนโทรลได้รวดเร็วกว่า และที่สำคัญมีการใช้ทรัพยากรของระบบน้อยกว่าอีกด้วย แต่เนื่องจากคอนโทรล image ใช้ทรัพยากรค่อนข้างน้อย ดังนั้นจึงสนับสนุนคุณสมบัติ เหตุการณ์ และ วิธีน้อยกว่า คอนโทรล picture box ในการเลือกใช้คอนโทรลทั้งสอง ต้องคำนึงถึงความต้องการที่จะให้คอนโทรลตอบสนองต่อผู้ใช้ และพิจารณาถึงความสามารถของคอนโทรลเหมาะสมกับงานหรือไม่ ทั้งนี้เพื่อให้สามารถเลือกใช้คอนโทรลได้อย่างเหมาะสม เพราะจะทำให้ประหยัดทรัพยากรของระบบ และช่วยให้แอพพลิเคชันสามารถทำงานได้รวดเร็วขึ้น ซึ่งโดยส่วนใหญ่มักจะนำคอนโทรล image มาใช้ในการสร้าง ไอคอนในทูลบาร์หรือทูลบอกซ์ ทั้งนี้เนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากลักษณะการใช้งานไอคอนเหล่านี้ไม่ซับซ้อนและคอนโทรล image ก็มีเหตุการณ์ต่างๆ ที่เพียงพอสำหรับการตอบสนองต่อการกระทำของผู้ใช้ เช่น Click, MouseDown, MouseMove, MouseUp เป็นต้น แต่เนื่องจากคอนโทรล image ไม่มีคุณสมบัติการเป็นตัวบรรจุ (container) ดังนั้นคอนโทรลอื่นๆ ที่วางอยู่ภายใน image จึงไม่มีคุณสมบัติการรวมกลุ่ม (group)

สำหรับคุณสมบัติของ image มีดังต่อไปนี้

BorderStyle

รายงานหรือกำหนดการแสดงผลเส้นขอบคอนโทรล image

รูปแบบการใช้งาน

object.BorderStyle = [value]

value หมายถึง นิพจน์เลขจำนวนเต็ม integer ที่กำหนดการแสดงผลเส้นขอบ ดังนี้

0 ไม่แสดงผลเส้นขอบ (default)

1 แสดงผลเส้นขอบเป็นเส้นทึบ

DragIcon และ DragMode

คุณสมบัติ DragIcon สำหรับกำหนดไอคอนของคอนโทรลเมื่อเกิดเหตุการณ์ลากเม้าส์ (drag) ส่วนคุณสมบัติ DragMode จะใช้ในการกำหนดโหมดการลากเม้าส์

รูปแบบการใช้งาน

object.DragIcon [=icon]

object.DragMode [=mode]

Enabled

รายงานหรือกำหนดให้คอนทรล image สามารถมีหรือไม่มีการตอบสนองต่อเหตุการณ์ ที่ถูกทำให้เกิดขึ้นโดยผู้ใช้ เช่น Click เป็นต้น

รูปแบบการใช้งาน

object.Enabled[=boolean]

Height,Width,Left และ Top

รายงานหรือกำหนดขนาดความกว้าง (Width), ความสูง (Height) และตำแหน่งการวางคอนโทรล (Left, Top) ลงในตัวบรรจุ

รูปแบบการใช้งาน

object.Height[=value]

object.Width[=value]

object.Left[=x]

object.Top[=y]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Index

กำหนดคอนโทรล image ที่มีชื่อเดียวกันภายในฟอร์มเดียวกันของโปรเจ็กต์ ให้เป็นคอนโทรลแบบ อะเรย์ ซึ่งจะมีลักษณะการใช้งาน เหมือนกับอะเรย์โดยทั่วๆไป ของ VB/Win

รูปแบบการใช้งาน

object.[(integer)].Index

MouseIcon

กำหนดไอคอนของเม้าส์สำหรับใช้แสดงผล เมื่อพอยน์เตอร์ของเม้าส์อยู่ภายในพื้นที่ของคอนโทรล image

รูปแบบการใช้งาน

object.MouseIcon = LoadPicture(pathname)

object.MouseIcon = picture

MousePointer

กำหนดไอคอนพอยน์เตอร์ของเม้าส์ในขณะที่เม้าส์อยู่ภายในขอบเขตของคอนโทรล image

รูปแบบการใช้งาน

object.MousePointer[=setting]

Name

กำหนดชื่อของคอนโทรล image เช่น image1 หรือ ToolBarGroup เป็นต้น สำหรับใช้ในการอ้างอิงโดยโค๊ดในการเขียนโปรแกรม ซึ่งคุณสมบัตินี้ผู้อ่านกำหนดได้เฉพาะในขณะออกแบบเท่านั้น และไม่สามารถแก้ไขชื่อของคอนโทรลโดยอาศัยโค้ดในขณะรันแอพพลิเคชัน เพราะจะทำให้เกิดข้อผิดพลาดทันที

Picture

รายงานหรือกำหนดกราฟฟิกหรือไฟล์กราฟฟิกที่จะแสดงผลภายในคอนโทรล image

รูปแบบการใช้งาน

object.Picture[=picture]

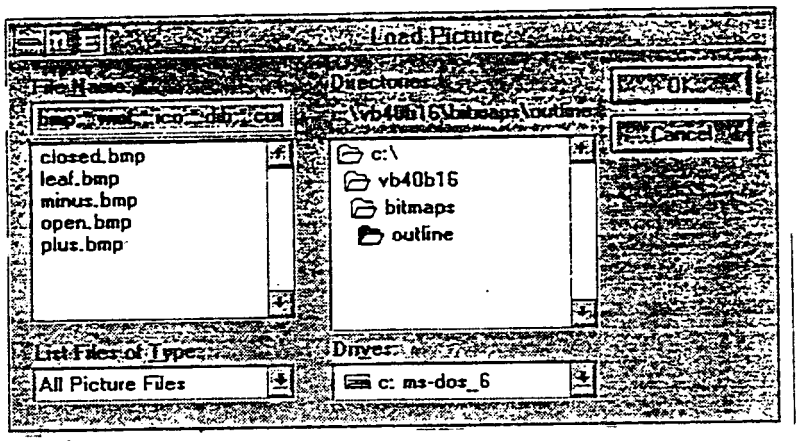
objrct.Picture = LoadPicture(pathname)

picture หมายถึง นิพจน์สตริงที่กำหนดไฟล์กราฟฟิก ซึ่งโดยปกติ picture จะถูกกำหนดให้เท่ากับสตริงว่าง

pathname หมายถึง เส้นทางของไฟล์กราฟฟิก

สำหรับการกำหนดไฟล์กราฟฟิกให้กับคุณสมบัติ Picture ในขณะออกแบบ ให้คลิ๊กที่ปุ่มของคุณสมบัติ Picture ในหน้าต่างคุณสมบัติจะปรากฏไดอะล็อกบ็อกซ์ Open project ดังรูป 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 แสดงไดอะลอกซ์บอกซ์ Open Picture

สำหรับการกำหนดไฟล์กราฟฟิกให้กับคุณสมบัติ Picture จะต้องเป็นไฟล์ ดังนี้

.bmp ไฟล์บิตแม็พ (Bitmap)

.wmf ไฟล์Windows metafile

.dib ไฟล์ Device Independent Bitmap

.ico ไฟล์ icon

.cur ไฟล์ไอคอนของเคอร์เซอร์ สนับสนุนเฉพาะ VB/Win4.0 เท่านั้น

โดยที่ในขณะออกแบบสามารถเปลี่ยนแปลงไฟล์กราฟฟิกที่จะกำหนดให้คุณสมบัติ picture ได้โดยการใช้ฟังชัน LoadPicture ดังโค้ดข้างล่าง

```
Image1.Picture = LoadPicture("c:\windows\256color.bmp")
Image1.Picture = LoadPicture()
```

ซึ่งโค้ดบรรทัดแรกกำหนดให้คุณสมบัติ Picture มีค่าเท่ากับไฟล์ 256color.bmp ในไดเรคทอรีของวินโดว์ในไดรฟ์ C: นั่นก็หมายความว่า ให้คอนโทรล Image1 แสดงผลภาพกราฟฟิกของไฟล์ 256color.bmp ส่วนโค้ดบรรทัดล่างก็เป็นเทคนิคการลบภาพกราฟฟิกออกจากคอนโทรล Image1 สำหรับการกำหนดไฟล์กราฟฟิกให้กับคอนโทรล image ในขณะออกแบบและขณะรันแอพพลิเคชันนั้นมีข้อแตกต่างกันอยู่ประการหนึ่งก็คือ ไฟล์กราฟฟิกที่ถูกกำหนดในขณะออกแบบเมื่อผู้อ่านสร้างไฟล์ .EXE ไฟล์กราฟฟิกนั้นก็จะถูกรวมเข้าไว้ในไฟล์ .EXE ดังนั้นจะต้องมีไฟล์กราฟฟิกเหล่านี้เก็บเอาไว้ต่างหากด้วย และนอกจากคอนโทรล image จะใช้แสดงภาพกราฟฟิกจากไฟล์กราฟฟิกต่างๆ ได้แล้ว ผู้อ่านยังสามารถนำไฟล์ในคลิปบอร์ด (Clipboard) มาแสดงผลภายในคอนโทรล image ได้อีกด้วย โดยสามารถกระทำได้ 2 วิธี ดังนี้

1.<u>ในขณะออกแบบ</u>

ให้คัดลอก(copy) หรือตัด(cut) ภาพกราฟฟิกจากแอพพลิเคชันอื่น ๆ เช่น Paintbrush เพื่อนำภาพกราฟฟิกไปเก็บเอาไว้ในคลิปบอร์ด หลังจากนั้นก็ให้คลิ๊ก ที่คอนโทรล image ภายในฟอร์มแล้ว เลือกคำสั่ง Paste ในเมนู Edit ของ VB/Winเพื่อปะภาพกราฟฟิกจากคลิปบอร์ดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในคอนโทรล image แต่มีข้อสังเกตอยู่ประการหนึ่งตรงที่คุณสมบัติ Picture ของคอนโทรล image ในหน้าต่างคุณสมบัติก็ยังคงเท่ากับ None

2.ในขณะรันแอพพลิเคชัน

เราสามารถจำลองการทำงานของคำสั่ง Copy, Paste ของเมนู Edit ได้โดยการใช้วิธีของ SetData และ GetData ของคลิปบอร์ดตามลำดับ โดยในขณะรันแอพพลิเคชันเราสามารถปะภาพกราฟฟิกจากคลิปบอร์ดลงในคอนโทรล image ได้โดยการใช้โค้ด ดังนี้

```
Image1.Picture = Clipboard.GetData()
```

Stretch

รายงานหรือกำหนดให้กราฟฟิกมีขนาดเปลี่ยนแปลงตามขนาดของคอนโทรล image

รูปแบบการใช้งาน

```
object.Stretch[=boolean]
```

boolean หมายถึง นิพจน์เลขจำนวนเต็ม integer สำหรับกำหนดสถานะของการเปลี่ยนแปลงขนาดกราฟฟิก ดังนี้

True หมายถึง กำหนดให้กราฟฟิกเปลี่ยนขนาดให้พอดีกับขนาดของคอนโทรล image

False หมายถึง กำหนดให้ขนาดของคอนโทรล image เปลี่ยนขนาดให้พอดีกับขนาดของกราฟฟิก

ดังนั้นถ้าหากคุณสมบัติ Stretch มีค่าเท่ากับ False ไม่ว่าผู้อ่านจะเปลี่ยนแปลงคอนโทรล image ให้มีขนาดเท่าไรก็ตาม มันก็จะปรับขนาดของตัวเองให้พอดีกับขนาดของกราฟฟิกทันที

Tag

ใช้จัดเก็บข้อมูลพิเศษ โดยที่ข้อมูลที่ถูกจัดเก็บ ในคุณสมบัตินี้จะไม่มีผลใดๆต่อคอนโทรล image ซึ่งโดยทั่วไปเราจะใช้เป็น ที่จัดเก็บข้อมูลชั่วคราวของคอนโทรลเพื่อใช้แสดงผล หรือทำการบางอย่างโดยโค้ดที่เรากำหนด

รูปแบบการใช้งาน

```
object.Tag [=expression]
```

Visible

รายงานหรือกำหนดให้คอนโทรล image สามารถมองเห็นหรือถูกซ่อนในขณะรันแอพพิเคชัน

รูปแบบการใช้งาน

```
object.Visible [=boolean]
```

ส่วนเหตุการณ์ที่สามารถเกิดขึ้นได้กับคอนโทรล image มีดังต่อไปนี้

Click

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกิดเมื่อผู้ใช้ คลิ๊กเมาส์ภายในคอนโทรล image

รูปแบบของโพรซีเดอร์เหตุการณ์

Sub control_Click([index As Integer])

DblClick

เกิดเมื่อผู้ใช้ดับเบิ้ลคลิ๊กเมาส์ภายในคอนโทรล image

รูปแบบของโพรซีเดอร์เหตุการณ์

sub control_DblClick([index As integer])

DragDrop

เกิดขึ้นเมื่อผู้ใช้มีการลากแล้ววาง (drag-drop)

รูปแบบของโพรซีเดอร์เหตุการณ์

Sub control_DragDrop([index As Integer,] source As control,x As Single,y As Single)

DragOver

เกิดขึ้นเมื่อผู้ใช้มีการลากเม้าส์เนื่องจากขบวนการลากแล้ววางบนคอนโทรลซึ่งภายในโพรซีเดอร์เหตุการณ์นี้จะทำให้ผู้อ่านสามารถทราบถึงสถานะของการลากแล้ววางของคอนโทรลที่เม้าส์ลากผ่านได้ทันที

รูปแบบของโพรซีเดอร์เหตุการณ์

Sub control_DragOver([index As Integer,]source As control,x As Single,y As Single ,state As Integer)

mousedown,mouseup และ mousemove

เหตุการณ์ mousedown เกิดขึ้นเมื่อผู้ใช้มีการกดปุ่มของเม้าส์และเมื่อผู้ใช้ปล่อยปุ่มของเม้าส์ก็จะเกิดเหตุการณ์ mouseup ส่วนเหตุการณ์ mousemove จะเกิดขึ้นเมื่อผู้ใช้มีการเลื่อนพอยน์เตอร์ของเม้าส์ภายในคอนโทรล image

รูปแบบของโพรซีเดอร์เหตุการณ์

Sub control_mousedown([index As Integer,]button As Integer,shift As Integer,x As Single,y As Single)

Sub control_mouseup([index As Integer,]button As Integer,shift As Integer,x As Single,y As Single)

Sub control_mousemove([index As Integer,]button As Integer,shift As Integer,x As Single,y As Single)

สำหรับวิธีที่สามารถใช้ควบคุมการแสดงผลของคอนโทรล image มีดังต่อไปนี้

Drag

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดวิธีการลากเม้าส์ของคอนโทรล ซึ่งวิธีนี้จะถูกใช้เมื่อเรากำหนดให้คุณสมบัติ DragMode ของคอนโทรลมีค่าเท่ากับ 0 หรือการลากอยู่ในโหมด manual แต่อย่างไรก็ดีถ้าหากว่าคุณสมบัติ DragMode ของคอนโทรลมีค่าเท่ากับ 1 หรือการลากอยู่ในโหมด automatic ก็ยังคงสามารถใช้วิธี Drag ควบคุมขั้นตอนการลากเม้าส์ได้เช่นกัน

รูปแบบการใช้วิธี

object.Drag action

move

เคลื่อนย้ายตำแหน่งหรือปรับเปลี่ยนขนาดของคอนโทรล image

รูปแบบการใช้วิธี

object.move left[,top[,width[,height]]]

Refresh

กำหนดให้มีการปรับปรุงการแสดงผลกราฟฟิกของคอนโทรล image ให้ทันสมัยทันที

รูปแบบการใช้วิธี

object.Refresh

ZOrder

กำหนดลำดับการแสดงผลของคอนโทรลในแนวแกน Z(ทิศทางพุ่งออกจากระนาบจอภาพ)

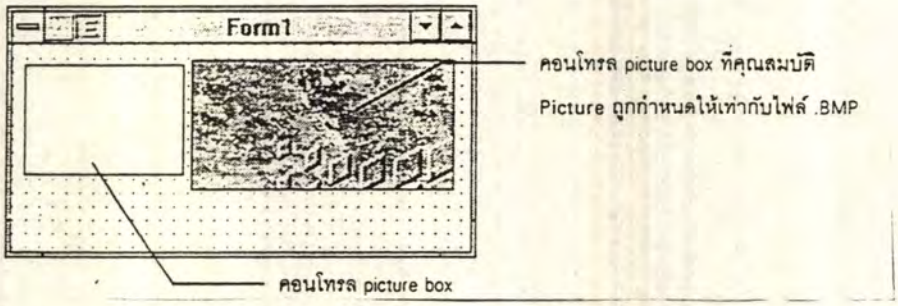
รูปแบบการใช้วิธี

object.ZOder[position]

2. คอนโทรล Picture box

คอนโทรล Picture box เป็นคอนโทรลที่ใช้แสดงผลกราฟฟิกและภาพกราฟฟิกจากไฟล์ได้เช่นเดียวกับคอนโทรล image แต่คอนโทรล Picture box เป็นคอนโทรลที่มีความสามารถเหนือกว่าในหลายๆด้านเช่นการใช้เป็นตัวบรรจุ(container การแสดงผลข้อความแบบกราฟฟิก DDE การวาดกราฟฟิกด้วยฟังก์ชันด้านกราฟฟิกฯลฯ ด้วยความสามารถที่เหนือกว่าหลายๆด้านจึงทำให้คอนโทรล Picture box มีการใช้ทรัพยากรระบบมากกว่า และใช้เวลาในการประมวลผลเพื่อแสดงผลข้อมูลมากกว่าด้วย ดังนั้นการทำคอนโทรล Picture box ไปใช้งานในแอพพลิเคชันก็ควรที่จะเลือกนำไปใช้ในส่วนที่จำเป็นจริงๆ และคอนโทรล image ไม่สามารถรองรับความต้องการนั้นๆทั้งนี้เพื่อให้แอพพลิเคชันที่ได้สามารถทำงานได้รวดเร็วยิ่งขึ้น สำหรับในขณะออกแบบสามารถเลือกคอนโทรลนี้เพื่อมาวางลงบนฟอร์มได้โดยคลิ๊กที่ไอคอน Picture box ในทูลบ็อกซ์ แล้วนำมาลากลงบน ฟอร์มดังตัวอย่างในรูป 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.2 แสดงคอนโทรล picture box ในขณะออกแบบ

สำหรับคุณสมบัติของคอนโทรล Picture box มีดังต่อไปนี้

Align

รายงานหรือกำหนดตำแหน่งและการปรับขนาดอัตโนมัติของ Picture box บนฟอร์ม

รูปแบบการใช้งาน

object.Align[=number]

number หมายถึง นิพจน์เลขจำนวนเต็ม integer ที่กำหนดรายละเอียดของ Picture box ดังนี้

ค่าตัวเลข	รายละเอียด
0	หมายถึงให้ขนาดเป็นไปตามที่ออกแบบหรือกำหนดโดยโค้ดในขณะรันแอพพลิเคชันซึ่งได้ถูกกำหนดให้เป็นค่าปกติในขณะออกแบบ
1	หมายถึง ให้วางชิดขอบบนของฟอร์มและปรับขนาดความกว้างอัตโนมัติตามขนาดความกว้าง(scale width)ของฟอร์ม
2	หมายถึงให้วางชิดขอบล่างของฟอร์ม ปรับขนาดความกว้างอัตโนมัติตามขนาดความกว้าง(scale width)ของฟอร์ม
3	หมายถึงให้วางชิดขอบซ้ายของฟอร์มและปรับขนาดความสูงอัตโนมัติตามขนาดความสูง(scale height)ของฟอร์ม
4	หมายถึงให้วางชิดขอบขวาของฟอร์มและปรับขนาดความสูงอัตโนมัติตามขนาดความสูง(scale height)ของฟอร์ม

ซึ่งโดยปกติในการใช้งาน Picture box สำหรับงานด้านกราฟฟิกต่างๆมักจะกำหนดให้คุณสมบัติ Align มีค่าเท่ากับ 0 และจะปรับเปลี่ยนขนาดของ Picture box โดยโค้ดในขณะรันแอพพลิเคชัน แต่ในบางสถานะการณ์อาจจะประยุกต์ Picture box ไปใช้ในงานด้านการรวมกลุ่มคอนโทรลอื่นๆก็ได้เช่น ใช้สร้างแถบเครื่องมือ(tool bar)หรือแถบบอกสถานะ(status bar) เป็นต้น ซึ่งในกรณีเฉพาะอย่างนี้ถ้าหากกำหนดให้คุณสมบัติ align มีค่าเท่ากับ 1 และ 2 ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับ ก็จะทำให้ Picture box วางชิดขอบบนและขอบล่างของฟอร์มตามลำดับ โดยที่ Picture box จะทำการปรับขนาดความกว้างให้พอดีกับความกว้างของฟอร์มโดยอัตโนมัติ ซึ่งจะทำให้ลดภาระการเขียนโค้ดลงได้มาก

AutoRedraw

รายงานหรือกำหนดการแสดงผลโดยวิธีกราฟฟิกของคอนโทรล Picture box ซึ่งจะมีผลต่อการวาดพื้นที่ของคอนโทรล Picture box ใหม่อีกครั้ง ในกรณีที่คอนโทรล Picture box มีการเปลี่ยนแปลงขนาดหรือคอนโทรล Picture box ถูกซ้อนโดยคอนโทรลหรือฟอร์มอื่นๆ

รูปแบบการใช้งาน

object.AutoRedraw[=boolean]

boolean = true หมายถึง กำหนดให้ VB/Win ทำการวาดกราฟฟิกต่างๆของคอนโทรล Picture box ใหม่อีกครั้ง โดยอัตโนมัติทุกครั้งที่คอนโทรลมีการเปลี่ยนแปลงขนาดหรือถูกซ้อนโดยฟอร์มหรือหน้าต่างอื่นๆ ซึ่งกราฟฟิกและข้อความต่างๆที่วาดลงในคอนโทรล Picture box ก็จะถูกจัดเก็บลงในหน่วยความจำแบบ image ดังนั้นในการวาดคอนโทรล Picture box ใหม่อีกครั้ง VB/Win ก็จะใช้ image ที่จัดเก็บในหน่วยความจำนี้มาใช้

= false หมายถึง กำหนดให้ขบวนการวาดคอนโทรล Picture box ใหม่อีกครั้งโดยอัตโนมัติถูกยกเลิก ดังนั้นในการวาดคอนโทรล Picture box ก็ตองใช้เหตุการณ์ Paint ซึ่งโดยปกติ VB/Win จะกำหนดให้คุณสมบัตินี้มีค่าเป็น false

AutoSize

รายงานหรือกำหนดการปรับขนาดของคอนโทรล Picture box เพื่อให้สามารถแสดงกราฟฟิกได้ทั้งหมด

รูปแบบการใช้งาน

object.AutoSize[=boolean]

boolean = true หมายถึง กำหนดให้คอนโทรลมีการปรับขนาดให้สามารถแสดงผล กราฟ-ฟิกได้หมดโดยอัตโนมัติ

= false หมายถึง กำหนดให้คอนโทรลไม่ต้องปรับขนาดตามกราฟฟิก โดย ถ้าหากขนาดของกราฟฟิกใหญ่กว่าขนาดของคอนโทรล Picture box ส่วนกราฟฟิกที่เกินก็จะถูกตัดทิ้ง (ค่าปกติ)

BackColor และ ForeColor

รายงานหรือกำหนดสีพื้น (BackColor) และ สีตัวอักษร (ForeColor) ของข้อความใน คอนโทรล picture box

รูปแบบการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

object.BackColor[=color]

object.ForeColor[=color]

BorderStyle

รายงานหรือกำหนดการแสดงผลเส้นขอบของคอนโทรล picture box

รูปแบบการใช้งาน

object.BorderStyle = [value]

value หมายถึง นิพจน์เลขจำนวนเต็ม integer ที่กำหนดการแสดงผลเส้นขอบ ดังนี้

0 ไม่แสดงผลเส้นขอบ

1 แสดงผลเส้นขอบเป็นเส้นทึบ ซึ่งเป็นค่าปกติของคอนโทรล picture box

ClipControls

รายงานหรือกำหนดวิธีทางกราฟฟิกที่ใช้ในการเหตุการณ์ Paint โดยจะกำหนดให้ทำการวาดกราฟฟิกใหม่พื้นที่ทั้งหมด หรือเฉพาะบริเวณที่ถูกทับ หรือยื่นออกนอกขอบเขตแสดงผลเท่านั้น ซึ่งสามารถแก้ไขค่าของคุณสมบัตินี้ได้เฉพาะในขณะออกแบบเท่านั้น

รูปแบบการใช้งาน

object.ClipControls

True หมายถึง กำหนดให้วิธีทางกราฟฟิกที่ใช้ในการเหตุการณ์ Paint ทำการวาดกราฟฟิกใหม่พื้นที่ทั้งหมด ซึ่ง VB/Win จะกำหนดให้เป็นค่าปกติ

False หมายถึง กำหนดให้วิธีทางกราฟฟิกที่ใช้ในการเหตุการณ์ Paint ทำการวาดกราฟฟิกใหม่เฉพาะบริเวณที่ถูกทับหรือยื่นออกนอกขอบเขตแสดงผลเท่านั้น

สำหรับ picture box ที่คุณสมบัติ ClipControls มีค่าเท่ากับ False จะสามารถทำการวาดกราฟฟิกได้เร็วกว่า เพราะไม่จำเป็นต้องประมวลผลขบวนการตัดขอบ สำหรับขบวนการตัดขอบ (clipping) นี้ เป็นขบวนการที่วินโดว์ใช้ในการกำหนดของหน้าต่าง ซึ่งวินโดว์จะนำไปใช้ในการวาดหน้าต่างใหม่อีกครั้งเมื่อเหตุการณ์ Paint ถูกเรียก สำหรับ VB/Win เหตุการณ์ Paint ของฟอร์มจะถูกเรียกก็ต่อเมื่อฟอร์มนั้น ๆ ถูกซ้อนทับโดยฟอร์มอื่นๆ หรือฟอร์มได้ถูกแสดงผลหรือฟอร์มได้มีการเลื่อนตำแหน่งแล้วขอบข้างใดข้างหนึ่งของฟอร์มได้ยื่นออกไปนอกขอบจอภาพก็ได้

CurrentX และ CurrentY

รายงานหรือกำหนดตำแหน่งโคออร์ดิเนตในแนวแกน x หรือแนวนอน(CurrentX) และแนวแกน y หรือแนวแกนตั้ง(CurrentY) สำหรับใช้ในการแสดงผลกราฟฟิกหรือตัวอักษรครั้งถัดไป

รูปแบบการใช้งาน

object.CurrentX[=x]

object.CurrentY[=y]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

x , y หมายถึง นิพจน์ชนิดเลขจำนวนจริงที่กำหนดตำแหน่งโคออร์ดิเนตในแนวแกน x และแกน y ตามลำดับ

สำหรับตำแหน่งโคออร์ดิเนตของจุดกำเนิด (origin) จะอยู่ที่มุมบนซ้ายของ picture box โดยที่โคออร์ดิเนตของจุดกำเนิดจะเท่ากับ (0,0) แต่สามารถแก้ไขค่าของโคออร์ดิเนตของพื้นที่ของ picture box ได้โดยการกำหนดค่าให้กับคุณสมบัติ ScaleHeight, ScaleWidth, ScaleLeft และ ScaleTop

DragIcon และ DragMode

คุณสมบัติ DragIcon สำหรับกำหนดไอคอนของคอนโทรล เมื่อเกิดเหตุการณ์ลากเม้าส์(drag) ส่วนคุณสมบัติ DragMode จะใช้ในการกำหนดโหมดการลากเม้าส์

รูปแบบการใช้งาน

object.DragIcon [=icon]

object.DragIcon [=icon]

DrawMode

กำหนดวิธีทางตรรกที่ใช้ในการแสดงผลด้วยวิธีด้านกราฟฟิก เช่น การแสดงผลแบบกลับสี กับการแสดงผลปัจจุบัน (Invert)

รูปแบบการใช้งาน

object.DrawMode[=number]

DrawStyle

กำหนดรูปแบบของเส้นที่ใช้ในการแสดงผลด้วยวิธีด้านกราฟฟิก เช่น เส้นทึบ หรือเส้นประ เป็นต้น

รูปแบบการใช้งาน

object.DrawStyle[=number]

DrawWidth

กำหนดความหนาของเส้นที่ใช้ในการแสดงผลด้วยวิธีด้านกราฟฟิก

รูปแบบการใช้งาน

object.DrawWidth[=number]

Enabled

รายงานหรือกำหนดให้คอนโทรล picture box สามารถมีหรือไม่มีการตอบสนองต่อเหตุการณ์ที่ถูกทำให้เกิดขึ้นโดยผู้ใช้ เช่น Click เป็นต้น

รูปแบบการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

object.Enabled[=boolean]

FillColor

รายงานหรือกำหนดสีที่ใช้ในการระบายด้วยวิธีกราฟฟิก Circle และ Line

รูปแบบการใช้งาน

object.FillColor[=value]

value หมายถึง ค่าสีที่กำหนดซึ่งสามารถกำหนดได้จากถาดสีในขณะออกแบบ หรือโดยการใช้โค้ดของฟังก์ชัน RGB หรือ QBColor หรือโดยการกำหนดด้วยเลขจำนวนเต็ม long

โดยปกติ VB/Win จะกำหนดให้คุณสมบัตินี้มีค่าเท่ากับ 0 (สีดำ) แต่ถ้าหากคุณสมบัติ FillStyle มีค่าเป็น 1 หรือถูกกำหนดให้โปร่งใส (transparent) คุณสมบัติ FillColor ก็จะไม่มีผลต่อวิธีกราฟฟิกข้างต้น

FillStyle

รายงานหรือกำหนดรูปแบบของการระบายพื้นที่ภายในที่สร้างโดยวิธีกราฟฟิก Circle และ Line

รูปแบบการใช้งาน

object.FillStyle[=number]

number = หมายถึง เลขจำนวนเต็มที่กำหนดรูปแบบในการระบายด้วยวิธีกราฟฟิก ดังนี้

0	ระบายทึบ
1	ระบายแบบโปร่งใสหรือไม่มีการระบาย
2	ระบายในรูปแบบเส้นแนวนอน
3	ระบายในรูปแบบเส้นแนวตั้ง
4	ระบายในรูปแบบเส้นแนวทะแยงมุมขึ้น(มุมบนซ้ายมายังมุมล่างขวา)
5	ระบายในรูปแบบเส้นแนวทะแยงมุมลง (มุมล่างซ้ายมายังมุมบนขวา)
6	ระบายในรูปแบบเส้นกากบาท(ตารางกริด)
7	ระบายในรูปแบบทะแยงมุม

Font

คุณสมบัติ Font ใช้ในการกำหนดคุณสมบัติของออปเจ็กต์ฟอนต์ หรือรายงานคุณสมบัติของออปเจ็กต์ Font ของคอนโทรล picture box ซึ่งคุณสมบัตินี้จะมีเฉพาะ VB/Win 4.0 เท่านั้น

รูปแบบการใช้งาน

object.Font

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

hDC

รายงานหมายเลข handle ที่เอนไวรอนเมนท์ของวินโดว์ใช้ในการเข้าถึง Device Context ของออปเจ็กต์ ซึ่งค่านี้จะสามารถใช้ได้เฉพาะเมื่อรันแอพพิเคชันเท่านั้น

รูปแบบการใช้งาน

object.hDC

HelpContextID

กำหนดหมายเลข ID สำหรับหัวข้อที่เกี่ยวข้องในไฟล์ help

LinkItem, LinkMode, LinkTimeOut และ LinkTopic

คุณสมบัติของคอนโทรลสำหรับการติดต่อกับคุณสมบัติอื่นๆ โดยผ่านทางDDE

Parent

รายงานฟอร์มที่บรรจุคอนโทรลเอาไว้ ค่าของคุณสมบัตินี้ไม่สามารถกำหนดได้ในขณะออกแบบ ซึ่งสามารถอ่านค่าของคุณสมบัตินี้ได้เฉพาะ ในขณะรันแอพพิเคชันเท่านั้น

รูปแบบการใช้งาน

object.Parent

TabIndex

กำหนดลำดับการรับโฟกัสของคอนโทรล picture box เมื่อผู้ใช้เลื่อนไปมาระหว่างคอนโทรลโดยใช้คีย์ Tab

รูปแบบการใช้งาน

object.TabIndex[=index]

ส่วนเหตุการณ์ที่สามารถเกิดขึ้น ได้กับคอนโทรล picture box มีดังต่อไปนี้

Change

เกิดขึ้นเมื่อมีการแก้ไขหรือเปลี่ยนแปลงกราฟฟิกหรือข้อมูลที่แสดงผลภายใน picture box หรือโดยการแก้ไขค่าของคุณสมบัติ Picture โดยโค้ดในขณะรันแอพพลิเคชัน

รูปแบบของโพรซีเดอร์เหตุการณ์

Sub object_Change([index as integer])

index หมายถึง ดัชนีบ่งบอกถึงลำดับสมาชิกของคอนโทรลชนิดอะเรย์

GotFocus

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหตุการณ์นี้เกิดขึ้นเมื่อคอนโทรล picture box ได้รับโฟกัส ซึ่งคอนโทรลอาจจะได้รับโฟกัสโดยที่ผู้ใช้กดคีย์ Tab โดยการคลิ้กเม้าส์ก็ได้ โดยการใช้คีย์การเข้าถึง หรือโดยการใช้โค้ดผ่านวิธี SetFocus ก็ได้

รูปแบบของโพรซีเดอร์เหตุการณ์

Sub control_GotFocus([index as integer])

Keydown, KeyUp และ KeyPress

เหตุการณ์ KeyDown เกิดขึ้นเมื่อผู้ใช้มีการกด คีย์ และเมื่อผู้ใช้ปล่อยคีย์ก็จะเกิดเหตุการณ์ KeyUp ส่วนเหตุการณ์ KeyPress เกิดขึ้นเมื่อผู้ใช้มีการกดและ ปล่อยคีย์ซึ่งเหตุการณ์ทั้งสามจะเกิดขึ้นก็ต่อเมื่อคอนโทรล picture box ได้รับโฟกัสเท่านั้น

LostFocus

เหตุการณ์นี้จะเกิดขึ้นก็ต่อเมื่อมีการเปลี่ยนโฟกัสไปยังคอนโทรลอื่นๆ ดังนั้นคอนโทรลที่เคยได้รับโฟกัสมาก่อนก็จะสูญเสียโฟกัสไปนั่นเอง

รูปแบบของโพรซีเดอร์เหตุการณ์

Sub control_LostFocus([index As Integer])

Paint

เกิดขึ้นเมื่อมีการเปลี่ยนแปลงขนาดของ picture box จนมีผลต่อการแสดงผล picture box หรือโดยการเลื่อน picture box ไปมาแล้วมีส่วนหนึ่งส่วนใดของ picture box ถูกซ้อนหรือทับโดยฟอร์มอื่นๆ หรือยื่นออกไปนอกเหนือจากขอบของจอภาพ

รูปแบบของโพรซีเดอร์เหตุการณ์

Sub control_Paint()

Resize

เกิดขึ้นเมื่อมีการเปลี่ยนแปลงขนาดของคอนโทรล picture box

รูปแบบของโพรซีเดอร์เหตุการณ์

Sub object_Resize()

Cls

กำหนดให้มีการลบกราฟฟิกทั้งหมดที่ปรากฏใน picture box

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบการใช้วิธี

object.Cls

Point

รายงานค่าสี RGB ที่แสดงผลตรงตำแหน่งโคออร์ดิเนตที่กำหนดของ picture box

รูปแบบการใช้วิธี

object.Point(x,y)

3.2 ฟอร์มชนิด MDI

จากการใช้ แอพพลิเคชันไมโครซอฟต์เวิร์ด จะเห็นว่าสามารถสร้างหรือเปิดเอกสารได้คราวละหลายๆ ไฟล์ ซึ่งเอกสารที่ถูกเปิดแต่ละไฟล์ก็จะถูกแยกแสดงผลในหน้าต่างที่แยกออกจากกัน โดยมีไมโครซอฟท์เวิร์ดเป็นตัวจัดการเอกสารในแต่ละหน้าต่าง ซึ่งความสามารถในจุดนี้เองที่เรียกว่า MDI form หรือ Multiple-Document Interface Form

ฟอร์มชนิด MDI เป็นฟอร์มชนิดพิเศษของ VB/Win ที่มีคุณสมบัติสามารถใช้เป็นตัวบรรจุฟอร์มอื่นๆเอาไว้ภายในได้ โดยที่ฟอร์มที่จะถูกแสดงผลภายในฟอร์ม MDI จะต้องเป็นฟอร์มที่คุณสมบัติ MDIChild มีค่าเท่ากับ true เท่านั้น และสามารถกำหนดคอนโทรลต่างๆ ให้กับฟอร์มMDI ได้เฉพาะเมนูและคอนโทรล picture box เท่านั้น ทั้งนี้เนื่องจากคอนโทรล picture box เป็นคอนโทรลที่มีคุณสมบัติ Align (จะสังเกตได้ว่าเมื่อเพิ่มคอนโทรล picture box ลงในฟอร์ม MDI คุณสมบัติ Align ก็จะถูกกำหนดให้มีค่าเท่ากับ 1 ทันที และที่สำคัญไม่สามารถแก้ไขให้คุณสมบัตินี้มีค่าเท่ากับ 0 อีกด้วย ดังนั้นจึงสรุปได้ว่าคอนโทรลที่จะเพิ่มให้กับฟอร์ม MDI จะต้องมีคุณสมบัติ Align เท่านั้น) ดังนั้นสำหรับคอนโทรลอื่นๆ เช่น Label หรือ Command button ก็สามารถใส่ให้กับฟอร์ม MDI ได้โดยทางอ้อม โดยการใส่คอนโทรลเหล่านี้ลงในคอนโทรล picture box อีกทีหนึ่ง

VB/Win ได้กำหนดให้ฟอร์ม MDI เป็นฟอร์ม modeless เท่านั้น และในขณะรันแอพพลิเคชันฟอร์มที่คุณสมบัติ MDIChild มีค่าเท่ากับ True จะต้องถูกแสดงผลภายในฟอร์ม MDI เท่านั้น

คุณสมบัติของฟอร์มMDI มีรายละเอียดที่สำคัญดังต่อไปนี้

AutoShowChildren

รายงานหรือกำหนดให้มีการแสดงผลหรือไม่มีการแสดงผลฟอร์มลูก MDI ในขณะโหลดฟอร์มลูก MDI ด้วยประโยค Load

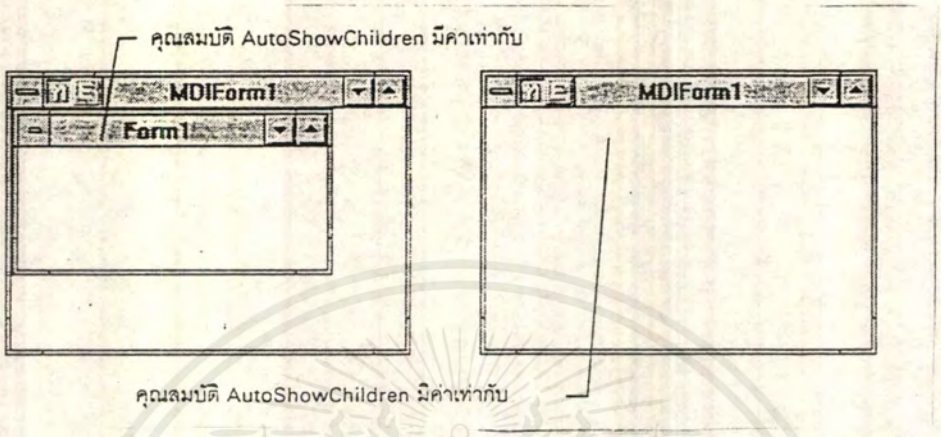
รูปแบบการใช้งาน

[mdiform.]AutoShowChildren[=boolean]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

boolean = True หมายถึง ฟอร์มลูก MDI จะถูกแสดงผลโดยอัตโนมัติในขณะโหลดฟอร์มลูก MDI

= False หมายถึง ฟอร์มลูก MDI จะไม่ถูกแสดงผลโดยอัตโนมัติในขณะโหลดฟอร์มลูก MDI



รูป 3.3 แสดงผลของคุณสมบัติ AutoShowChildren ที่มีต่อการโหลดของฟอร์มลูก MDI

Enabled

กำหนดให้ฟอร์ม MDI มีการตอบสนองต่อเหตุการณ์ที่ถูกทำให้เกิดขึ้นโดยผู้ใช้ เช่น Click เป็นต้น

รูปแบบการใช้งาน

[mdiform.]Enabled [=boolean]

boolean = True หมายถึง กำหนดให้ฟอร์มMDI มีการตอบสนองต่อเหตุการณ์ที่ถูกทำให้เกิดขึ้นโดยผู้ใช้

= False หมายถึง กำหนดให้ฟอร์ม MDI ไม่มีการตอบสนองต่อเหตุการณ์ที่ถูกทำให้เกิดขึ้นโดยผู้ใช้

Icon

กำหนดไอคอนสำหรับฟอร์ม เพื่อใช้แสดงในขณะที่ฟอร์ม MDI แสดงผลในสถานะไอคอน (minimized)

รูปแบบการใช้งาน

[mdiform.]Icon

สำหรับไอคอนที่สามารถกำหนดให้กับฟอร์ม MDI จะเป็นไฟล์ที่มีนามสกุล .ICO และ ในการกำหนดไอคอนให้กับฟอร์มสามารถกระทำได้ทั้งในขณะออกแบบและในขณะรันแอพพลิเคชัน โดยขณะออกแบบก็ให้คลิ๊กที่คุณสมบัตินี้ในหน้าต่างคุณสมบัติ ก็จะปรากฏไดอะล็อกบ็อกซ์เพื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้เลือกไฟล์ .ICO ส่วนในขณะรันแอพพลิเคชัน ก็สามารถเขียนโค้ดเพื่อเลือกไฟล์ .ICO ให้กับคุณสมบัตินี้ได้โดยการใช้ฟังก์ชัน LoadPicture ดังเช่น

MDIForm1.Icon = LoadPicture("c:\dragdrop.ico")

MDIForm1.Icon = LOadPicture()

โดยที่โค้ดบรรทัดบนเป็นการโหลดไฟล์ dragdrop.ico จากไดเรคทอรีรากของไดรฟ์ C: เพื่อใช้เป็นไอคอนให้กับฟอร์ม MDIForm1 ส่วนโค้ดบรรทัดล่างกำหนดไอคอนของฟอร์ม MDIForm1 เป็นไอคอนว่าง

Name

รายงานหรือกำหนดชื่อของฟอร์ม MDI เช่น MDIForm1 หรือ EditorFrame เป็นต้น สำหรับใช้ในการอ้างอิงโดยโค้ดในการเขียนโปรแกรม ซึ่งคุณสมบัตินี้สามารถกำหนดได้ในขณะออกแบบเท่านั้น และไม่สามารถแก้ไขชื่อของฟอร์มได้โดยโค้ดในขณะรันแอพพลิเคชัน เพราะจะทำให้เกิดข้อผิดพลาด

Visible

รายงานหรือกำหนดให้ฟอร์มMDI สามารถแสดงผลหรือถูกซ่อนในขณะรันแอพพลิเคชัน

รูปแบบการใช้งาน

[mdiform.]Visible [=boolean]

boolean = True หมายถึง ให้แสดงผลฟอร์ม MDI

= False หมายถึง ให้ซ่อนฟอร์ม MDI ดังนั้น จึงไม่สามารถมองเห็นได้

ในขณะออกแบบ ถ้าหากกำหนดให้คุณสมบัตินี้มีค่าเป็น False จะยังคงสามารถมองเห็นฟอร์มได้ ทั้งนี้คุณสมบัตินี้จะมีผลเฉพาะในขณะรันแอพพลิเคชันเท่านั้น

WindowState

รายงานหรือกำหนดสถานะการแสดงผลของฟอร์ม MDI ในขณะรันแอพพลิเคชัน

รูปแบบการใช้งาน

[mdiform.]Windowtate[=value]

value หมายถึง เลขจำนวนเต็มที่ใช้กำหนดสถานะของฟอร์ม MDI ดังนี้

0	ขนาดปกติตามที่กำหนดในขณะออกแบบหรือขนาดที่เปลี่ยนแปลงโดยโค้ดหรือผู้ใช้ในขณะรันแอพพลิเคชัน
1	ฟอร์มMDI มีสถานะเป็นไอคอนเช่นเดียวกับเมื่อคลิ๊กที่ปุ่ม Minimize
2	ฟอร์ม MDI มีสถานะขยายขนาดเต็มจอภาพเช่นเดียวกับเมื่อคลิ๊กที่ปุ่ม Maximize

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหตุการณ์ที่เกิดขึ้นกับฟอร์ม MDI ที่สำคัญมีดังนี้

Click

เกิดเมื่อผู้ใช้คลิ๊กเม้าส์ภายในพื้นที่ของฟอร์มMDI

รูปแบบของโพรซีเดอร์เหตุการณ์

Sub MDIForm_Click()

Load และ Unload

เหตุการณ์ Load จะเกิดขึ้นเมื่อภายในแอพพลิเคชันมีการโหลดฟอร์ม MDI มายังหน่วยความจำ ซึ่งเหตุการณ์การโหลดฟอร์มMDI สามารถเกิดขึ้นได้จากหลายๆวิธี ไม่ว่าจะโดยประโยค Load หรือโดยการอ้างอิงถึงคอนโทรลหรือฟอร์มนั้นๆ โดยโค้ดภายในแอพพลิเคชันหรือโดยการกำหนดให้เป็นฟอร์ม startup ก็จะทำให้ฟอร์ม MDI ถูกโหลดอัตโนมัติเมื่อรันแอพพลิเคชัน ส่วนเหตุการณ์ Unload จะเกิดขึ้นเมื่อมีการยกเลิกฟอร์ม MDI ออกจากหน่วยความจำในขณะรันแอพพลิเคชัน

รูปแบบของโพรซีเดอร์เหตุการณ์

Sub MDIForm_Load()

Sub MDIForm_Unload(cancel As Integer)

cancel หมายถึง เลขจำนวนเต็มที่ใช้กำหนดการยกเลิกฟอร์ม โดยถ้าหากมีค่าเท่ากับ 0 ฟอร์ม MDI ก็จะถูกยกเลิกออกจากหน่วยความจำ แต่ถ้าหากมีค่าเป็นจำนวนเต็มที่ไม่เท่ากับ 0 ฟอร์มก็จะไม่ถูกยกเลิกออกจากหน่วยความจำหรือเป็นการยกเลิกเหตุการณ์ Unload

วิธีที่ใช้ในการจัดการกับฟอร์ม MDI ที่สำคัญ มีดังต่อไปนี้

Arrange

กำหนดรูปแบบการจัดเรียงไอคอนหรือฟอร์มลูก MDI ภายในฟอร์ม MDI

รูปแบบการใช้วิธี

[mdiform.]Arrange arrangement

arrangement หมายถึง เลขจำนวนเต็มมีค่าตั้งแต่ 0 ถึง 3 สำหรับกำหนดวิธีการจัดเรียงไอคอนหรือฟอร์มลูก MDI ภายในฟอร์ม MDI ดังนี้

ค่าตัวเลข	รายละเอียด
0	จัดเรียงฟอร์มลูก MDI แบบ cascade
1	จัดเรียงฟอร์มลูก MDI แบบ tile ในแนวนอน
2	จัดเรียงฟอร์มลูก MDI แบบ tile ในแนวตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3	จัดเรียงไอคอนของฟอร์มลูก MDI

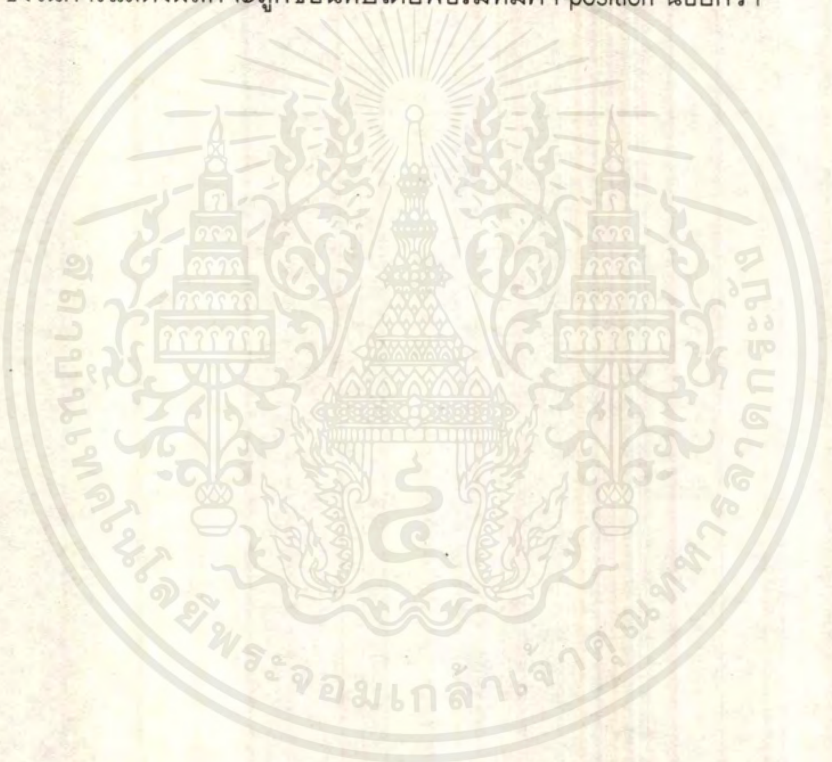
Zorder

กำหนดลำดับการแสดงผลของฟอร์มลูก MDI ในแนวแกน Z (ทิศทางพุ่งออกจากระนาบหน้าจอ)

รูปแบบการใช้วิธี

[mdiform.]ZOrder[position]

position หมายถึง ลำดับการวางฟอร์มลูก MDI ลงบนฟอร์ม MDI โดยที่ถ้าหาก position มีค่าเท่ากับ 0 หรือมีการเรียกใช้วิธีนี้โดยไม่กำหนดค่าของ position ฟอร์มลูก MDI นั้นก็จะถูกวางเอาบนสุด และในทางกลับกันถ้าหากฟอร์มใดที่มีการกำหนดค่าของ position มากที่สุดก็จะถูกวางไว้ล่างสุด ซึ่งในการแสดงผลก็จะถูกซ้อนทับโดยฟอร์มที่มีค่า position น้อยกว่า



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การ INTERFACE

PARALLEL INTERFACE

การส่งข้อมูลทางพอร์ตขนาน เป็นการส่งเป็นชุดข้อมูล ทีละ 8 บิต (1 ไบต์) ทำให้เกิดผลดีในแง่ของความเร็วในการส่งข้อมูล แต่มีข้อเสียในแง่ของระยะทาง เพราะจะส่งได้ในระยะทางไม่เกิน 20 ฟุต การเชื่อมต่อเครื่อง CNC เข้ากับคอมพิวเตอร์ผ่านทางพอร์ตขนานนั้น เป็นการติดต่อเพื่อส่งไบต์ควบคุมจากคอมพิวเตอร์ ไปยังเครื่อง CNC และรับไบต์ควบคุมจากเครื่อง CNC มายังเครื่องคอมพิวเตอร์ ในส่วนของการเชื่อมต่อทางพอร์ตขนานนั้น สามารถแบ่งออกเป็น 2 ส่วนใหญ่ๆได้ดังนี้

1. Hardware
2. Software

1. Hardware

ในส่วนของ Hardware เราแบ่งออกเป็น 2 ส่วน ดังนี้

1.1 Parallel port ของคอมพิวเตอร์

ภายในคอมพิวเตอร์ จะมีพอร์ตขนาน แต่ละพอร์ต จะมีค่าแอดเดรสที่แน่นอน ดังนี้

Port address range	BIOS-configured device code
278 to 27F	LPT1, LPT2, LPT3
378 to 37F	LPT1, LPT2, LPT3
3BC to 2BF	LPT1

LPT : Line Printer Terminal

โดยที่ port address 3BDH, 379H และ 279H จะเก็บค่าของ Status register ของเครื่องพิมพ์ และ port address 3BEH, 37AH และ 27AH จะเก็บค่าของ Control register ของเครื่องพิมพ์ แต่ทั้งสองแอดเดรสนี้ ไม่ได้ถูกนำมาใช้งาน เพราะเราต้องการติดต่อกับเครื่อง CNC (ไม่ใช่เครื่องพิมพ์)

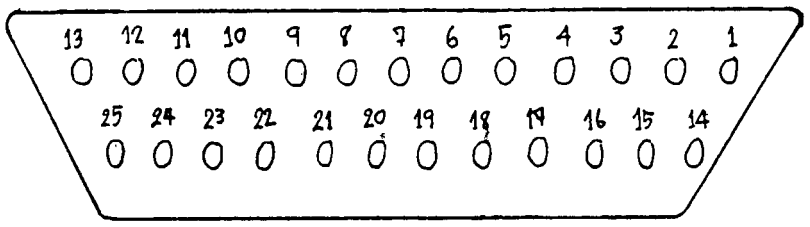
สำหรับ port address ที่เก็บข้อมูลที่ใช้เป็นไบต์ควบคุม คือ 378H ซึ่ง ณ แอดเดรสนี้ สามารถเก็บข้อมูลได้ 8 บิต (D0-D7) เราจึงนำ port address นี้มาใช้งาน เพื่อควบคุมเครื่อง CNC

รูปแบบของ Parallel Interface

1. Centronics interface มี 36 ขา
2. IBM-compatible interface มี 25 ขา

ในที่นี้ เราใช้แบบ IBM-compatible interface ซึ่งมีลักษณะ ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



25-Way Female D -type

จากรูป ประกอบด้วยขาต่างๆ ดังนี้

- ขา 2-9 Output pins ทำหน้าที่เป็น Data port
- ขา 10, 11, 12, 13, 15 Input pins ทำหน้าที่เป็น Status port
- ขา 1, 14, 16, 17 Output pins ทำหน้าที่เป็น Control port
- ขา 18 - 25 ทำหน้าที่เป็น ground

1.2 **Parallel port ของเครื่อง CNC**

ส่วนประกอบทางด้าน Hardware ทั้งหมดของเครื่อง CNC แสดงได้ดังรูปในหน้าถัดไป

ส่วนที่ใช้สำหรับติดต่อกับคอมพิวเตอร์ทางพอร์ตขนานนั้น เราใช้ **DNC- interface** ของ Compact 5- CNC ซึ่งเป็นตัวรับส่งคำสั่งในการทำงานของ CNC - program ดังนั้นเราจึงใช้ DNC-interface ในการรับค่าไบต์ควบคุมจากคอมพิวเตอร์ เพื่อให้สั่งงานทางคอมพิวเตอร์ได้

ตำแหน่งของ DNC-interface คือ ตำแหน่ง **X62** ในรูป โดยแต่ละขาของ DNC ประกอบด้วย

X62 pin 1	Output	Status hand
X62 pin 2	Input	Turret - hsnd operation
X62 pin 3	Input	Instruction G66 + INP
X62 pin 4	Input	Instruction INP + REV
X62 pin 5	-	-
X62 pin 6	Input	Instruction G66 +FWD
X62 pin 7	Output	Status program running
X62 pin 8	Output	Status intermediate stop
X62 pin 9	Input	Instruction switch hand / CNC
X62 pin 10	-	-
X62 pin 11	-	-
X62 pin 12	-	-
X62 pin 13	-	-
X62 pin 14	-	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

X62 pin 15	Output	Output set with M8, M9
X62 pin 16	-	-
X62 pin 17	Input	Instruction start
X62 pin 18	Output	Output set with M22, M23
X62 pin 19	Output	Status main motor ON/OFF
X62 pin 20	Output	Output impulse set with M26
X62 pin 21	Input	Instruction blockage-turret
X62 pin 22	Power	+10 V not controlled
X62 pin 23	Power	GND
X62 pin 24	Power	GND
X62 pin 25	Power	GND
X62 pin 26	Power	+5 V controlled

Output : บิตที่ส่งค่าจากเครื่อง CNC ไปยัง คอมพิวเตอร์

Input : บิตที่รับค่าจาก คอมพิวเตอร์เข้ามา

Power : ไฟเลี้ยง

หลักการทำงานของ DNC-board

1.

- M03 เป็นคำสั่งที่ทำให้ตัวจับชิ้นงานหมุน

  M05 เป็นคำสั่งที่ทำให้ตัวจับชิ้นงานหยุดหมุน

- M30 เป็นคำสั่งจบโปรแกรม ซึ่งจะทำให้ตัวจับชิ้นงานหยุดหมุนโดยอัตโนมัติ โดยไม่ต้องใช้คำสั่ง M05

2.

DNC-interface (X62) ประกอบด้วยขา 26 ขา ซึ่งมีลักษณะการทำงานเป็นดังนี้

A) Output

PIN 1 : Status Hand

บิตแสดงโหมดการทำงาน CNC/Hand

Pin 1 Lo เมื่ออยู่ในโหมดการทำงาน CNC

Pin 1 Hi เมื่ออยู่ในโหมดการทำงาน Hand

PIN 7 : Status Program Running

บิตแสดงสถานะการทำงาน

Pin 7 Lo เมื่อโปรแกรมไม่ได้ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin 7 Hi เมื่อโปรแกรมกำลังทำงาน

PIN 8 : Status Intermediate Stop

บิตแสดงสถานะการหยุดทำงาน

Pin 8 Lo เมื่อไม่หยุดทำงาน

Pin 8 Hi เมื่อหยุดทำงาน

PIN 15 : Output Set M08, M09

บิตแสดงสถานะเมื่อทำงานด้วยคำสั่ง M08, M09

Input of	Produces at switch output/Pin	The condition	Initial condition
M08 M09	X62/15 X62/15	HI LO	LO
M22 M23	X62/18 X62/18	HI LO	LO

PIN 19 : Status main motor ON/OFF

บิตแสดงสถานะเปิด-ปิดของมอเตอร์

Pin 19 Lo เมื่อมอเตอร์ไม่ทำงาน

Pin 19 Hi เมื่อมอเตอร์อยู่

PIN 20 : Pulse output

บิตให้สัญญาณพัลซ์ เมื่อป้อนคำสั่ง M26 โดยมีรูปแบบคำสั่ง ดังนี้

N3/M26/H3 : เมื่อ H คือ จำนวนพัลซ์ มีค่าระหว่าง 0 - 221

B) Inputs

PIN 2 : Turret - Hand operation

PIN 3 : Set the machine to RS-232 operation

(Received as with G66+INP)

PIN 4 : Instruction INP + REV

PIN 6 : Set the machine to RS-232 operation

(Transmit as G66+FWD)

PIN 9 : Instruction switch Hand/CNC

เปลี่ยนโหมดการทำงานจาก Hand เป็น CNC หรือ จาก CNC เป็น Hand

PIN 17 : Start

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มการทำงานของโปรแกรม CNC

PIN 21 : Instruction turret blockage

C) Power

PIN 22-26

ตำแหน่งการเชื่อมต่อ DNC-interface เข้ากับ Parallel port ของคอมพิวเตอร์

เนื่องจากพอร์ตขนานของคอมพิวเตอร์ สามารถรับบิตข้อมูลได้เพียง 8 บิตเท่านั้น ดังนั้นเราจึงต้องเลือกค่าที่ต้องการควบคุมจาก DNC-interface ให้ได้เพียง 8 ค่า โดยที่ยังสามารถควบคุมเครื่อง CNC ให้ทำงานได้ตามที่ต้องการได้ ดังนั้นจึงได้เลือกบิตต่างๆ ได้ดังต่อไปนี้

X62/Pin	Parallel port (378 H)/Pin	หน้าที่	การทำงาน	อธิบาย
3	2	Instruction G66+INP	เปลี่ยนจาก 0 เป็น 1	Load program จาก คอมฯ ไปยัง CNC
6	3	Instruction G66+FWD	เปลี่ยนจาก 0 เป็น 1	Save program จาก CNC ไปยัง คอมฯ
9	4	Switch Hand/CNC	เปลี่ยนจาก 1 เป็น 0	Hand to CNC or CNC to Hand
17	5	Instruction Start	เปลี่ยนจาก 1 เป็น 0	Start program
X62/Pin	Parallel port(379 H)/Pin	หน้าที่	การทำงาน	อธิบาย
1		Status Hand	Hi - Hand	Lo - CNC
7		Status program running	Hi - Running	Lo - Not running
8		Status intermidiate stop	Hi - Stop	Lo - No stop
15		Output set with M8,M9		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Software

โปรแกรมที่ใช้เพื่อควบคุมการทำงานของ CNC เราใช้ Visual Basics 4.0 เป็นโปรแกรมควบคุม ทั้งนี้มีสาเหตุมาจากเราต้องการ software ที่ทำหน้าที่หลักในการรับและส่งค่า, ไม่เน้นการประมวลผลมากนัก และเน้นไปในทางเชื่อมต่อกับผู้ใช้ให้ใช้งานได้สะดวกและง่าย ดังนั้นเราจึงเลือกใช้ Visual Basics

Source code ที่ใช้ควบคุมการรับส่งค่าผ่านทางพอร์ตขนานนั้น จะใช้คำสั่ง INP และOUT โดยมีรูปแบบคำสั่ง ดังนี้

Out Port%,Value

X% = Inp(Port%)

โดยที่

Port% หมายถึง ตำแหน่งของพอร์ต เป็นเลขฐาน 16 เช่น &H378 หมายถึง ตำแหน่งของพอร์ตขนาน

Value หมายถึง ค่าที่ต้องการส่งออกไปยังพอร์ต

เงื่อนไขในการใช้งาน เป็นดังนี

คำสั่ง Inp และ Out เป็นคำสั่งที่อยู่ใน Library ของ VB โดย VB จะเก็บ Library ต่าง ๆ ในรูปของไฟล์ที่มีสกุล .DLL (Dynamic Link Library) โดยไฟล์ที่ทำหน้าที่ในการรับส่งค่าออกจากพอร์ตนั้น มีชื่อว่า **"Inpout.dll"** ซึ่งจะต้องมีการประกาศ Function ไว้ใน Module เพื่อบอกให้ VB รู้ว่า คำสั่ง INP, OUT อยู่ใน Library ใด

ข้อจำกัดของคำสั่ง Inp, Out คือ สามารถรับส่งค่าเป็นจำนวนหรือค่าตัวเลขเท่านั้นไม่สามารถรับส่งค่าที่เป็นตัวอักษรได้ และค่าที่จะป้อนเข้าไปนั้นให้ป้อนเป็นเลขฐานสิบ จากนั้นคอมฯ จะแปลงเป็นเลขฐานสอง แล้วส่งออกไปตามบิตข้อมูล D0-D7 ของพอร์ตขนาน จึงทำให้เราสามารถควบคุมการทำงานของเครื่อง CNC ได้ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SERIAL INTERFACE

การรับส่งโปรแกรมผ่านทาง SERIAL PORT ระหว่าง CNC กับ COMPUTER เราสามารถใช้โปรแกรมสำเร็จรูปที่ชื่อ EMCO ในการรับส่งโปรแกรม และสามารถส่งผ่านทาง DOS ได้อีกทางหนึ่ง แต่ใน PROJECT นี้เป็นการทำงานของ CNC ที่ต้องติดต่อกับอุปกรณ์อื่นอีก เราจึงต้องเขียนโปรแกรมขึ้นมาใหม่ โดยใช้ภาษา VISUAL BASIC

ดังนั้นเราสรุปว่า มี 3 วิธีในการรับส่งโปรแกรมผ่านSERIAL PORT

1.ใช้โปรแกรมชื่อ EMCO

2.ส่งทาง DOS โดยตรง

3.ใช้โปรแกรมภาษา VISUAL BASIC

รายละเอียดทั้ง 3 วิธีมีดังนี้

1.การใช้โปรแกรม EMCO มีขั้นตอนดังนี้

1.1 เขียนโปรแกรมทาง EDITOR ของ DOS โดยมีรูปแบบดังตัวอย่างนี้

```
%!*
    N' G'    X '     Z '   F'  H
    00M03
    01 90
    02 92   5000     500
    03 00   4000      00
    04 88     00 -   500   50  20
    05 00   4000 -   500
    06 84   3000 - 2500   50  20
    07M30
   M
```

1.2 ตั้งชื่อไฟล์โดยใช้สกุล C5N

1.3 เปิดเครื่อง CNC ไปที่ CNC OPERATION ใช้คำสั่ง G66 และกด INPUT

1.4 กด INPUT อีกครั้งหน้าจอ CNC จะขึ้นคำว่า LOAD PROGRAM

1.5 เรียกโปรแกรม EMCO ขึ้นมา และไปที่ SEROUT

1.6 พิมพ์ชื่อไฟล์ที่ต้องการส่ง แล้วกด ENTER

1.7 หน้าจอจะขึ้นว่า Key in text for the punch tape :(Enter = no text) และกด Enter

1.8 โปรแกรมจะถูกส่งไปยังเครื่อง CNC เพื่อนำไปใช้งานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.การส่งผ่านทางDOS โดยตรง มีขั้นตอนดังนี้

2.1 เขียนโปรแกรมทาง EDITOR ของ DOS โดยมีรูปแบบเหมือนการส่งผ่านโปรแกรมทาง EMCO และตั้งสกุล C5N

2.2 เซต COM 2 (BAUD RATE 300,PARITY EVEN,DATABIT 7,STOPBIT 1)

2.3 ใช้คำสั่ง COPY ชื่อไฟล์.C5N COM 2

2.4 โปรแกรมจะถูกส่งไปยัง CNC เพื่อนำไปใช้งานต่อไป

3.การส่งโปรแกรมโดยใช้โปรแกรมภาษา VISUAL BASIC

การรับส่งโปรแกรมโดยใช้โปรแกรมภาษา VISUAL BASIC ชื่อ COMMFUNC มี SOURCE CODE ดังนี้

```
Private Sub ReceiveProgram_Click()
 Comm_Open Commstring$
 If ncid Then Form1.Caption = "Now! Communication Port Open"
  Poll_Comm
 End Sub


Private Sub SendProgram_Click()
   Dim Ret, T, N, Cnt, x As Integer
   Dim Lpstat As COMSTAT
   Dim Textline, Textline1, C As String
   Static Buffer$
   Static Buffer1$
   Static Buffer2$
Comm_Open Commstring$
 If ncid Then Form1.Caption = " Communication Port Open "
Open "c:\vb\26.c5n" For Input As #1
  Do While Not EOF(1)
   Line Input #1, Buffer$
   Textline = Chr$(13)
   Textline1 = Chr$(10)
   Buffer1$ = Buffer$ + Textline + Textline1
   Buffer2$ = Buffer2$ + Buffer1$
  Loop
 T = GetCommError(ncid, Lpstat)
  If Lpstat.CbOutQue < 2000 Then
     Ret = WriteComm(ncid, Buffer2$, Len(Buffer2$))
     If Ret <= 0 Then Ret = -Ret
       Buffer2$ = Mid(Buffer2$, Ret + 1)
  End If
  Close #1
 End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งโปรแกรม

โปรแกรมที่จะทำการส่งไปยัง CNC นั้นเป็นโปรแกรมที่เขียนขึ้นโดยคำสั่ง G code และ M code เพื่อทำให้CNC สามารถทำงานได้ โดยจะเก็บไว้ในFile.C5N ซึ่งเป็น Text fileและสามารถเขียนได้โดยใช้ EDIT ของ DOS ส่วนการส่งโปรแกรมดังกล่าวไปยัง CNC นั้น จะใช้โปรแกรม Visual Basic เป็นตัวส่ง

จากหน้าจอเมื่อต้องการส่งโปรแกรม ให้กดปุ่ม SendProgram ซึ่งการทำงานของปุ่มดังกล่าว มีดังนี้

จากโปรแกรมหลักจะเข้าไปยัง Function ที่อยู่ในไฟล์ Dinamic Link Library (DLL) ซึ่ง Function ดังกล่าวจะทำการเปิดการติดต่อระหว่าง Computer กับ เครื่อง CNC โดยผ่านทาง Serial Port หลังจากนั้นก็จะส่งบิตควบคุมออกไปทางPararal Port เพื่อติดต่อกับ DNC interfaceของCNC ซึ่งจะเป็นการบอกให้ CNC เตรียมรับโปรแกรม

จากนั้นก็กลับมายังโปรแกรมหลักอีกครั้งหนึ่ง แล้วทำการเปิดไฟล์.C5N โดยจะนำข้อมูลในไฟล์ทีละบรรทัดมาเก็บไว้ในหน่วยความจำตัวหนึ่ง หลังจากนั้นก็โหลดข้อมูลบรรทัดต่อไปมาต่อกับข้อมูลเดิมในหน่วยความจำเดิม ดังสมการ

Buffer = Buffer + textline

Buffer = หน่วยความจำที่เป็นตัวเก็บข้อมูลที่จะทำการส่งออก

textline= ข้อมูลในแต่ละบรรทัดที่โหลดมา ซึ่งเมื่อถึง End Of File ก็ทำการหยุดการโหลดไฟล์

(ใน CNC ชนิด Emco นี้ ในการติดต่อกันนั้นตัวที่เป็นการบอกว่าเป็น End Of File นั้นจะเป็นดังนี้

M + ASCII(13) + ASCII(10))

เมื่อหน่วยความจำเก็บข้อมูลของโปรแกรมทั้งหมดที่จะส่งได้จนครบแล้ว ก็จะทำการส่งข้อมูลดังกล่าวออกไปยัง Serial Port ทันที โดยเครื่อง CNC ทำการรอรับอยู่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การ รับ โปรแกรม

เริ่มจาก เมื่อมีการกดปุ่ม Receive Program โปรแกรมการรับซึ่งเป็นโปรแกรมหลัก จะเรียกไปที่ Function ของ ไฟล์ Dynamic Link Library เพื่อทำการเปิดการติดต่อกับอุปกรณ์ภายนอก หลังจากนั้นก็ทำการส่งบิตควบคุมออกทางพอร์ตขนาน ให้กับ CNC เพื่อที่จะให้ส่งโปรแกรมมายัง Computer หลังจากนั้นข้อมูลของ โปรแกรม G code และ M code ทั้งหมด ก็จะถูกส่งมาทางพอร์ตอนุกรม ซึ่งเครื่อง Computer ก็มีการเตรียมตัวที่จะรับโปรแกรมโดยทำการวนลูปรอข้อมูล ซึ่งเมื่อมีข้อมูลเข้ามาก็จะหลุดออกจากลูปมายัง Procedure ที่ทำการเขียนข้อมูลที่ได้ออกทางหน้าจอ

โปรแกรมที่เครื่อง CNC ส่งออกมานั้นจะมีตัว % เป็นหัวของโปรแกรม ซึ่งถือเป็นบิตเริ่มต้นของการส่งโปรแกรมทุก ๆ โปรแกรม และจะมีรหัสแอสกี คือ ascii(13) และ ascii(10) เป็นตัวแสดงถึงการเริ่มบรรทัดใหม่ และมีตัว M+ascii(13)+ascii(10) เป็นตัวบอกว่าสิ้นสุดของการส่งโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
สรุปและข้อเสนอแนะ

1.ข้อสรุป

โครงงานนี้ เป็นการนำอุปกรณ์แต่ละตัวมารวมกันสร้างเป็นระบบ 1 ระบบ และใช้ PC ควบคุมการทำงานของระบบ สิ่งสำคัญของการรวมอุปกรณ์เป็นระบบ คือ รูปแบบ (Format) ของการติดต่อกันของอุปกรณ์แต่ละตัว เช่น รูปแบบคำสั่งที่ PC จะส่งไปควบคุมเครื่อง CNC และ แขนกล ให้ เครื่อง CNC และ แขนกลทำงานตามต้องการ นอกจากนี้จะต้องหาวิธีการ Interface อุปกรณ์แต่ละตัวเข้าด้วยกัน โดยจะต้องคำนึงถึงทรัพยากรที่มีอยู่เป็นหลักก่อน และถ้าไม่เพียงพอจึงต้องหาวิธีขยาย เช่น การ Interface ทั้ง เครื่อง CNC และ แขนกล เข้ากับ Computer ซึ่งต้องติดต่อกันผ่านทางพอร์ตอนุกรม ดังนั้น จึงต้องจัดหาพอร์ตอนุกรมให้เพียงพอต่อการใช้งานได้

ปัญหาอย่างหนึ่งที่เกิดขึ้น คือ แขนกลจะสามารถทำงานได้ตามตำแหน่งที่ต้องการ ก็ต่อเมื่อ เราเซตค่าตำแหน่วตำแหน่งต่างๆไว้เรียบร้อยแล้วก่อนทำงาน ดังนั้นจึงเกิดข้อจำกัดในด้านของตำแหน่งมาก กล่าวคือ อุปกรณ์ทุกอย่างเมื่อเซตค่าตำแหน่งไว้แล้ว จะเคลื่อนย้ายอีกไม่ได้ เพราะจะทำให้ค่าตำแหน่งทุกค่าที่เซตไว้เปลี่ยนแปลงไป ทำให้เกิดข้อผิดพลาดในการทำงานได้ ดังนั้นจะต้องตรวจสอบตำแหน่งของอุปกรณ์ทุกตัวก่อนการทำงานทุกครั้ง

2.ข้อเสนอแนะในการทำวิจัยในอนาคต

งานที่ควรพัฒนาต่อไป คือ การพัฒนาโปรแกรมควบคุมระบบอัตโนมัตินี้ให้มีประสิทธิภาพมากยิ่งขึ้น โดยการใช้ทรัพยากรที่มีอยู่แล้วใน Hardware ของแขนกล และ เครื่อง CNC มาใช้ประโยชน์ในเชิงพัฒนาโปรแกรม เช่น ให้โปรแกรมสามารถตรวจสอบสถานะการทำงานของเครื่อง CNC ได้ เป็นต้น นอกจากนี้ยังมีการพัฒนาอื่นๆอีกที่สามารถทำต่อได้ เช่น พัฒนาโปรแกรมให้มีลักษณะเป็น CAD (Computer Aid Design) กล่าวคือ ผู้ใช้สามารถออกแบบชิ้นงานตามที่ต้องการเองได้ แล้วให้เครื่อง Computer ประมวลผล เป็น Code ที่ส่งให้เครื่อง CNC เข้าใจได้ทันที พัฒนาโปรแกรมให้มีภาพกราฟฟิกที่สวยงามมากยิ่งขึ้น หรือติดตั้งระบบนิวแมติกเพิ่มเติมเพื่อให้ระบบมีการพัฒนาแบบอัตโนมัติโดยสมบูรณ์ และยังมีการพัฒนาอื่นๆอีกมากมายในการพัฒนาอื่นๆอีกมากมาย ในการพัฒนาโปรแกรมและระบบให้มีประสิทธิภาพมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณ อ. เทพจิตร์ เชยโกคา ในการให้คำปรึกษาและช่วยเหลือ ในการทำ Project

ขอขอบคุณ อ. เอกพจน์ ตันตราภิวัฒน์ และ อ.อุดม จันทร์จรัสสุข สำหรับความช่วยเหลือ และคำแนะนำที่เป็นประโยชน์

ขอขอบคุณ ท่านอาจารย์ทุกท่านในการถ่ายทอดความรู้ และ ประสิทธ์ประสาทวิชา

ขอขอบคุณเพื่อน ๆ ทุกคนที่ให้ความช่วยเหลือด้านอุปกรณ์และให้กำลังใจซึ่งกันและกันอย่างไม่ขาดสาย

ขอขอบคุณ คุณพ่อคุณแม่ที่ให้ทุกสิ่งทุกอย่างจนมาถึงทุกวันนี้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. สุทธิศักดิ์ พงศ์ธนาพาณิช, " Visual BASIC 4.0 Professional", บริษัท ซีเอ็ดยูเคชัน, 2539.

2. ผศ. ชาลี ตระการกูล, เทคโนโลยีซีเอ็นซี, สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2538

3. EMCO,EMCO Technics,edition 86-9 ref. no.EN2 024.

4. รศ. กฤษดา วิศวธีรานนท์, "ความรู้พื้นฐานเกี่ยวกับหุ่นยนต์อุตสาหกรรม" วารสารเทคโนโลยี ฉบับหุ่นยนต์อุตสาหกรรม ปีที่ 21, ฉบับที่ 117, 2537

5. Larry Horath, Computer Numerical Control Programing of Machines, 1993

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.
โปรแกรม source code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Option Explicit

Private Sub MDIForm_Load()
    Load Form3
    Load Form4
    Load Form5
    Load Form8
    MDIForm1.Icon = LoadPicture("c:\vb\vbplay1.ico")
    MDIForm1.Arrange 3
End Sub


Private Sub Picture1_Click()

End Sub

    Private Sub Picture2_Click()

    End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Option Explicit
Dim I As Long
 Sub Delay(Delaytime!)
      Dim Looptime!
      Looptime! = Timer + Delaytime!
           Do While Timer < Looptime!
                DoEvents
           Loop
 End Sub

  Private Sub Form_Load()
      Form1.WindowState = 2
      Timer1.Enabled = True
      Timer1.Interval = 10000
      Form1.Visible = True
      Form1.WindowState = 2
      Delay (3)
      Form2.Visible = True
   End Sub


      Private Sub Image1_Click(Index As Integer)

      End Sub


      Private Sub Timer1_Timer()

      End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Option Explicit

Private Sub Auto_Click()
    Form6.Visible = True
    Form2.Visible = False
End Sub


    Private Sub Form_Load()

    End Sub

    Private Sub Image1_Click()

    End Sub


Private Sub Manual_Click()
 MDIForm1.Visible = True
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Option Explicit

  Dim a%, b%, c%, d%, e, f, l As Integer
  Dim LpDCB As DCB
  Dim nCid As Integer, Text As String
  Const Commstring1 = "COM2:9600,n,8,2"
  Const Commstring2 = "COM4:300,e,7,1"
  Dim g, h, j, I As Integer, X As Long
Sub HandCNC1()
c% = Inp(&H378)
Out &H378, c% And 251
d% = Inp(&H378)
End Sub

Sub setinitiall1()
a% = Inp(&H378)
Out &H378, a% Or 4
b% = Inp(&H378)
End Sub

    Private Sub end_Click()

    End Sub

    Private Sub Form_Load()

    End Sub

  Private Sub HandCnc_Click()
    c% = Inp(&H378)
    Out &H378, c% And 251
    d% = Inp(&H378)
  End Sub

    Private Sub Setinitial1_Click()
    a% = Inp(&H378)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Out &H378, a% Or 4
b% = Inp(&H378)
End Sub


Private Sub start_Click()

End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Option Explicit
Dim g, h As Integer
Dim a%, b%, c%, d%, e, f, l As Integer
Dim LpDCB As DCB
Dim nCid As Integer, Text As String
Const Commstring1 = "COM2:9600,n,8,2"
Const Commstring2 = "COM4:300,e,7,1"
Dim j, I As Integer, X As Long

    Sub Close_comm()
    Dim Ret As Integer
    Ret = Closecomm(nCid)
    nCid = 0        'Clear nCid to stop Poll_comm
    End Sub

   Sub Comm_Open(Commstring$)
      Dim T, c As Integer
      Dim Commport As String
      T = Closecomm(2)
      T = Closecomm(3)
      T = Closecomm(4)
      T = Closecomm(1)
      Commport$ = Left$(Commstring$, InStr(Commstring$, ":") - 1)
      nCid = Opencomm(Commport$, 1024, 1024)
      If nCid < 0 Then
        MsgBox "Unable to Open Comm Device:" + Str$(nCid), 16
        If nCid = -2 Then T = Closecomm(Mid(Commstring$, 4, 1))
        End
       End If
       c = BuildCommDCB(Commstring$, LpDCB)
       If c Then
         MsgBox "Unable to Build Comm DCB ", 16
         End
       End If
        LpDCB.Id = Chr$(nCid)
        If (SetcommState(LpDCB)) Then
          MsgBox "Unable to set Comm State", 16
          End
        End If
      End Sub

Public Sub CNC(pol$)
  Dim Ret As Integer, T As Integer, n As Integer, cnt As Integer
  Dim Lpstat As COMSTAT
  Dim textline As String, c As String, textline1 As String
  Static Buffer$, buffer1$, Buffer2$
    Comm_Open Commstring2$                              'to make ncid <> 0
    If nCid = 0 Then Exit Sub
    Open pol$ For Input As #1
    Do While Not EOF(1)
        Line Input #1, Buffer$
        textline = Chr$(13)
        textline1 = Chr$(10)
        buffer1$ = Buffer$ + textline + textline1
        Buffer2$ = Buffer2$ + buffer1$
    Loop
    T = GetCommError(nCid, Lpstat)                      'Get current stats
    If Lpstat.cbOutQue < 2000 Then                     ' Space Left in buffer?
    Ret = WriteComm(nCid, Buffer2$, Len(Buffer2$))     'Place chars in buffer
    If Ret <= 0 Then Ret = -Ret                         'Ret has # of chars output,
    Buffer2$ = Mid(Buffer2$, Ret + 1)
    End If
    Close #1
    End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Sub load1()
   Static pol$
   h = Inp(&H378)
   Out &H378, h Or 1
   pol$ = "c:\vb\pol.c5n"
   CNC (pol$)
   Setinitial_Click
End Sub
 Sub load2()
   Static pol$
   h = Inp(&H378)
   Out &H378, h Or 1
   pol$ = "c:\vb\pol1.c5n"
   CNC (pol$)
   Setinitial_Click
  End Sub

   Sub load3()
   Static pol$
   h = Inp(&H378)
   Out &H378, h Or 1
   pol$ = "c:\vb\pol2.c5n"
   CNC (pol$)
   Setinitial_Click
   End Sub


  Private Sub Setinitial_Click()
   g = Inp(&H378)
   Out &H378, g And 254
   End Sub
```

```
Private Sub Form_Load()

End Sub

      Private Sub Label1_Click()

      End Sub


   Private Sub LoadA1_Click()
      load1
   End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub loadA2_Click()
load2
End Sub


Private Sub loadA3_Click()
load3
End Sub


   Private Sub Picture1_Click()
      loadA2.Enabled = False
      loadA3.Enabled = False
      LoadA1.Enabled = True
   End Sub

    Private Sub Picture2_Click()
      LoadA1.Enabled = False
      loadA3.Enabled = False
      loadA2.Enabled = True
    End Sub


 Private Sub Picture3_Click()
      LoadA1.Enabled = False
      loadA2.Enabled = False
      loadA3.Enabled = True
 End Sub


Private Sub Setinitial2_Click()
g = Inp(&H378)
Out &H378, g And 254
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Option Explicit
Dim I, j As Integer


    Private Sub Form_Load()

    End Sub


Private Sub Image1_Click()

End Sub


Private Sub Setinitial3_Click()
I = Inp(&H378)
Out &H378, I Or 8
End Sub


Private Sub start_Click()
j = Inp(&H378)
Out &H378, j And 247
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Option Explicit
 Dim a%, b%, c%, d%, e, f, l As Integer
 Dim LpDCB As DCB
 Dim nCid As Integer, Text As String
 Const Cqmmstring1 = "COM2:9600,n,8,2"
 Const Commstring2 = "COM4:300,e,7,1"
 Dim g, h, j, I, z, o, p As Integer, X As Long
 Dim Ret As Integer
 Dim M1, M2, M3, M4, M5, M6, M7 As Integer
   Sub back()
      Static Buffer$
      Dim m, n As Integer
      Buffer$ = "6M-2280" + Chr$(13)
      Awrite_comm (Buffer$)
      m = Val(Left(Buffer$, 1))
      n = Val(Mid(Buffer$, 4, 4))
      Text1(m) = Text1(m) - n
      Delay (15)
      Buffer$ = "2M+0530" + Chr$(13)
      Awrite_comm (Buffer$)
      m = Val(Left(Buffer$, 1))
      n = Val(Mid(Buffer$, 4, 4))
      Text1(m) = Text1(m) + n
      Delay (3)
      Buffer$ = "3M-0180" + Chr$(13)
      Awrite_comm (Buffer$)
      m = Val(Left(Buffer$, 1))
      n = Val(Mid(Buffer$, 4, 4))
      Text1(m) = Text1(m) - n
      Delay (3)
      Buffer$ = "8M+2000" + Chr$(13)
      Awrite_comm (Buffer$)
      Delay (15)
      Buffer$ = "6M-0120" + Chr$(13)
      Awrite_comm (Buffer$)
      m = Val(Left(Buffer$, 1))
      n = Val(Mid(Buffer$, 4, 4))
      Text1(m) = Text1(m) - n
      Delay (3)
      Buffer$ = "2M-0530" + Chr$(13)
      Awrite_comm (Buffer$)
      m = Val(Left(Buffer$, 1))
      n = Val(Mid(Buffer$, 4, 4))
      Text1(m) = Text1(m) - n
      Delay (3)
      Buffer$ = "3M+0180" + Chr$(13)
      Awrite_comm (Buffer$)
      m = Val(Left(Buffer$, 1))
      n = Val(Mid(Buffer$, 4, 4))
      Text1(m) = Text1(m) + n
      Delay (3)
      Buffer$ = "6M+2400" + Chr$(13)
      Awrite_comm (Buffer$)
      m = Val(Left(Buffer$, 1))
      n = Val(Mid(Buffer$, 4, 4))
      Text1(m) = Text1(m) + n
      Delay (15)
      Buffer$ = "3M-0256 " + Chr$(13)
      Awrite_comm (Buffer$)
      m = Val(Left(Buffer$, 1))
      n = Val(Mid(Buffer$, 4, 4))
      Text1(m) = Text1(m) - n
      Delay (3)
      Buffer$ = "2M+0372" + Chr$(13)
      Awrite_comm (Buffer$)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
m = Val(Left(Buffer$, 1))
n = Val(Mid(Buffer$, 4, 4))
Text1(m) = Text1(m) + n
Delay (3)
Buffer$ = "8M-2000" + Chr$(13)
Awrite_comm (Buffer$)
Delay (3)
Buffer$ = "3M+0256 " + Chr$(13)
Awrite_comm (Buffer$)
m = Val(Left(Buffer$, 1))
n = Val(Mid(Buffer$, 4, 4))
Text1(m) = Text1(m) + n
Delay (3)
Buffer$ = "2M-0372" + Chr$(13)
Awrite_comm (Buffer$)
m = Val(Left(Buffer$, 1))
n = Val(Mid(Buffer$, 4, 4))
Text1(m) = Text1(m) - n
Delay (3)
End Sub

Sub Checkend()
line1:
  Out &H379, 255     'line 1
  o = Inp(&H379) And 248
  z = o And 16
If z = 0 Then
  GoTo line1
ElseIf z = 16 Then
  closecomm1
  OpenComm1
  Form7.Visible = True
  back
  Form7.Visible = False
End If
End Sub

Private Sub closecomm1()
Form6.Close_comm
End Sub

Sub Close_comm()
    Dim Ret As Integer
    Ret = Closecomm(nCid)
    nCid = 0          'Clear nCid to stop Poll_comm
End Sub
Sub Awrite_comm(Buffer$)
  Dim T, ij As Integer, Lpstat As COMSTAT
  Static buffer1$
  T = GetCommError(nCid, Lpstat)        ' Get current stats
    If Lpstat.cbOutQue < 1024 Then      ' Space left in Buffer ?
        If Ret >= 0 Then
            Ret = WriteComm(nCid, Buffer$, Len(Buffer$))     'Place chars in buffer
            If Ret <= 0 Then Ret = -Ret                 ' Ret is the number of char
    End If
  Buffer$ = Mid$(Buffer$, Ret + 1) ' Remove xmited chars from buffer
    End If
    Exit Sub
  End Sub
      Sub come()
      Static Buffer$
      Dim m, n As Integer
      Buffer$ = "3M-0256 " + Chr$(13)
      Awrite_comm (Buffer$)
      m = Val(Left(Buffer$, 1))
      n = Val(Mid(Buffer$, 4, 4))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Text1(m) = Text1(m) - n
Delay (3)
Buffer$ = "2M+0372" + Chr$(13)
Awrite_comm (Buffer$)
m = Val(Left(Buffer$, 1))
n = Val(Mid(Buffer$, 4, 4))
Text1(m) = Text1(m) + n
Delay (3)
Buffer$ = "8M+2000" + Chr$(13)
Awrite_comm (Buffer$)
Delay (3)
Buffer$ = "2M-0372" + Chr$(13)
Awrite_comm (Buffer$)
m = Val(Left(Buffer$, 1))
n = Val(Mid(Buffer$, 4, 4))
Text1(m) = Text1(m) - n
Delay (3)
Buffer$ = "3M+0256" + Chr$(13)
Awrite_comm (Buffer$)
m = Val(Left(Buffer$, 1))
n = Val(Mid(Buffer$, 4, 4))
Text1(m) = Text1(m) + n
Delay (3)
Buffer$ = "6M-2400" + Chr$(13)
Awrite_comm (Buffer$)
m = Val(Left(Buffer$, 1))
n = Val(Mid(Buffer$, 4, 4))
Text1(m) = Text1(m) - n
Delay (15)
Buffer$ = "2M+0530" + Chr$(13)
Awrite_comm (Buffer$)
m = Val(Left(Buffer$, 1))
n = Val(Mid(Buffer$, 4, 4))
Text1(m) = Text1(m) + n
Delay (3)
Buffer$ = "3M-0180" + Chr$(13)
Awrite_comm (Buffer$)
m = Val(Left(Buffer$, 1))
n = Val(Mid(Buffer$, 4, 4))
Text1(m) = Text1(m) - n
Delay (3)
Buffer$ = "6M+0120" + Chr$(13)
Awrite_comm (Buffer$)
m = Val(Left(Buffer$, 1))
n = Val(Mid(Buffer$, 4, 4))
Text1(m) = Text1(m) + n
Delay (15)
Buffer$ = "8M-2000" + Chr$(13)
Awrite_comm (Buffer$)
Delay (3)
Buffer$ = "2M-0530" + Chr$(13)
Awrite_comm (Buffer$)
m = Val(Left(Buffer$, 1))
n = Val(Mid(Buffer$, 4, 4))
Text1(m) = Text1(m) - n
Delay (3)
Buffer$ = "3M+0180" + Chr$(13)
Awrite_comm (Buffer$)
m = Val(Left(Buffer$, 1))
n = Val(Mid(Buffer$, 4, 4))
Text1(m) = Text1(m) + n
Delay (3)
Buffer$ = "6M+2280" + Chr$(13)
Awrite_comm (Buffer$)
m = Val(Left(Buffer$, 1))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
n = Val(Mid(Buffer$, 4, 4))
Text1(m) = Text1(m) + n
End Sub


Sub Comm_Open(Commstring$)
 Dim T, c As Integer
 Dim Commport As String
 T = Closecomm(2)
 T = Closecomm(3)
 T = Closecomm(4)
 T = Closecomm(1)
 Commport$ = Left$(Commstring$, InStr(Commstring$, ":") - 1)
 nCid = Opencomm(Commport$, 1024, 1024)
 If nCid < 0 Then
   MsgBox "Unable to Open Comm Device:" + Str$(nCid), 16
   If nCid = -2 Then T = Closecomm(Mid(Commstring$, 4, 1))
   End
 End If
 c = BuildCommDCB(Commstring$, LpDCB)
 If c Then
    MsgBox "Unable to Build Comm DCB ", 16
    End
 End If
  LpDCB.Id = Chr$(nCid)
  If (SetcommState(LpDCB)) Then
     MsgBox "Unable to set Comm State", 16
     End
   End If
  End Sub
Sub Delay(Delaytime!)
Dim Looptime!
      Looptime! = Timer + Delaytime!
           Do While Timer < Looptime!
                DoEvents
           Loop
End Sub
Private Sub OpenComm1()

line1

  If nCid Then
     Font.Bold = True
     FontSize = 5
     Else
  Form6.Comm_Open Commstring1$
     GoTo line1
   End If
 End Sub
Sub Setinitiall3()
I = Inp(&H378)
Out &H378, I Or 8
End Sub

Sub Startt()
j = Inp(&H378)
Out &H378, j And 247
End Sub

Private Sub Form_Load()
Dim I As Integer
   Text1(0).Move 30, 270, 40, 18
   Text1(1).Move 90, 270, 40, 18
   Text1(2).Move 150, 270, 40, 18
   Text1(3).Move 210, 270, 40, 18
   Text1(4).Move 270, 270, 40, 18
   Text1(5).Move 330, 270, 40, 18
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Text1(6).Move 390, 270, 40, 18
For I = 1 To 22
  Print ""
  Next
  Font.Bold = True
  FontSize = 9
  Print Tab(40); " Position of Motor "
  Print ""
  Print Spc(8); "M1"; Spc(8); "M2"; Spc(9); "M3"; Spc(9); "M4"; Spc(8); "M5"; Spc(8); "M6"; Spc(8);
M7"
End Sub

Private Sub Label1_Click()

End Sub


Public Sub Picture1_Click()
Dim m, n, I As Integer
Static pol$
Static Buffer$
pol$ = "c:\vb\pol.c5n"
OpenComm1
Form7.Visible = True
come
Delay (15)
Form7.Visible = False
Form3.setinitiall1
Delay (3)
Form3.HandCNC1
Delay (3)
Form4.load1
Delay (15)
Setinitiall3
Delay (5)
Startt
Delay (10)
Checkend
closecomm1
End Sub


Private Sub Picture2_Click()
Dim m, n, I As Integer
Static pol$
Static Buffer$
pol$ = "c:\vb\pol1.c5n"
OpenComm1
Form7.Visible = True
come
Delay (15)
Form7.Visible = False
Form3.setinitiall1
Delay (3)
Form3.HandCNC1
Delay (3)
Form4.load2
Delay (20)
Setinitiall3
Delay (5)
Startt
Delay (10)
Checkend
closecomm1
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Picture3_Click()
Dim m, n, I As Integer
Static pol$
Static Buffer$
pol$ = "c:\vb\pol2.c5n"
OpenComm1
Form7.Visible = True
come
Delay (15)
Form7.Visible = False
Form3,setinitiall1
Delay (3)
Form3.HandCNC1
Delay (3)
Form4.load3
Delay (15)
Setinitiall3
Delay (5)
Startt
Delay (10)
Checkend
closecomm1
End Sub

Private Sub Text1_Change(Index As Integer)

End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Option Explicit

'-----------------------------------------------
'Variables and constants used by the PicClip form:
'-----------------------------------------------

Const PAN_DEFAULT = 2
Dim PicWidth As Integer
Dim PanAmount As Integer
Dim RoverIdx As Integer

' WindowState constant
Const MINIMIZE = 1

' Color Constants
Const DARK_GRAY = &H808080
Const WHITE = &HFFFFFF
Const BLACK = &H0

Private Sub Make3D(pic As Form, ctl As Control)
'-----------------------------------------------
' Wrap a 3D effect around a control on a form.
'-----------------------------------------------
Dim AdjustX As Integer, AdjustY As Integer
AdjustX = Screen.TwipsPerPixelX
AdjustY = Screen.TwipsPerPixelY

' Set the top shading line.
pic.Line (ctl.Left - AdjustX, ctl.Top - AdjustY)-(ctl.Left + ctl.Width, ctl.Top - AdjustY), DARK_GRAY
pic.Line -(ctl.Left + ctl.Width, ctl.Top + ctl.Height), WHITE
pic.Line -(ctl.Left - AdjustX, ctl.Top + ctl.Height), WHITE
pic.Line -(ctl.Left - AdjustX, ctl.Top - AdjustY), DARK_GRAY
End Sub


Private Sub PanLandscape(ByVal PanAmount As Integer)
'-----------------------------------------------
' Pan the landscape by PanAmount.  Positive values
' pan right, negative values pan left.  Check and
' handle if we go off either edge.
'-----------------------------------------------
Dim NewX As Integer
  RoverIdx = (RoverIdx + 1) Mod 7
  picRover = picclip2.GraphicCell(RoverIdx)
End Sub



Private Sub Form3D(ByVal BorderWidth As Integer)
  '-----------------------------------------------
  ' Put a 3D border around the outside of this form.
  '-----------------------------------------------
  Dim AdjustX As Integer, AdjustY As Integer
  Dim I As Integer
  AdjustX = Screen.TwipsPerPixelX
  AdjustY = Screen.TwipsPerPixelY

  ' Set the top shading line.
      Me.Line (0, 0)-(Me.ScaleWidth - AdjustX, 0), BLACK
      Me.Line -(Me.ScaleWidth - AdjustX, Me.ScaleHeight - AdjustY), BLACK
      Me.Line -(0, Me.ScaleHeight - AdjustY), BLACK
      Me.Line -(0, 0), BLACK
```

เอกสารนี้เป็นเอกสาร... ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น... เนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ' Set the top shading line.
      For I = 1 To BorderWidth
      Me.Line (AdjustX * I, AdjustY * I)-(Me.ScaleWidth - (AdjustX * (I + 1)), AdjustY * I), WHITE
      Me.Line -(Me.ScaleWidth - (AdjustX * (I + 1)), Me.ScaleHeight - (AdjustY * (I + 1))), DARK_GRAY
      Me.Line -(AdjustX * I, Me.ScaleHeight - (AdjustY * (I + 1))), DARK_GRAY
      Me.Line -(AdjustX * I, AdjustY * I), WHITE
       Next
       End Sub


Private Sub Form_Load()
'-----------------------------------------------
' Set up the initial pictures.
'-----------------------------------------------
 Me.Move (Screen.Width - Me.Width) \ 2, (Screen.Height - Me.Height) \ 2

'-----------------------------------------------
' Set the initial image for the Rover
'-----------------------------------------------
 RoverIdx = 0
picRover.Picture = picclip2.GraphicCell(RoverIdx)
Timer1.Enabled = True
PanLandscape PanAmount
End Sub



Private Sub Form_Paint()
'-----------------------------------------------
' Repaint our 3D effects.
'-----------------------------------------------
Form3D 2
End Sub



 Private Sub Label1_Click()

 End Sub


 Private Sub Label2_Click()

 End Sub


 Private Sub picRover_Click()

 End Sub


Private Sub Timer1_Timer()

'-----------------------------------------------
' Pan the landscape until the timer is disabled.
'-----------------------------------------------
   PanLandscape PanAmount
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Option Explicit
 Dim a%, b%, c%, d%, e, f, l As Integer
 Dim LpDCB As DCB
 Dim nCid As Integer, Text As String
 Const Commstring1 = "COM2:9600,n,8,2"
 Const Commstring2 = "COM4:300,e,7,1"
 Dim g, h, j, I As Integer, X As Long
 Dim Ret As Integer

 Dim M1, M2, M3, M4, M5, M6, M7 As Integer

    '-----------------------------------------
    ' Variables and constants used by the PicClip form.
    '-----------------------------------------

    Const PAN_DEFAULT = 2

    Dim PicWidth As Integer
    Dim PanAmount As Integer
    Dim RoverIdx As Integer

    ' Used for keyboard operation of panning
    Const KEY_LEFT = &H25
    Const KEY_RIGHT = &H27

    ' WindowState constant
    Const MINIMIZE = 1

    ' Color Constants
    Const DARK_GRAY = &H808080
    Const WHITE = &HFFFFFF
    Const BLACK = &H0

    Private Sub PanLandscape(ByVal PanAmount As Integer)
    '-----------------------------------------
    ' Pan the landscape by PanAmount.  Positive values
    ' pan right, negative values pan left.  Check and
    ' handle if we go off either edge.
    '-----------------------------------------
    Dim NewX As Integer
      RoverIdx = (RoverIdx + 1) Mod 7
      Picture1 = picclip2.GraphicCell(RoverIdx)
    End Sub

    Sub SavePos()
    Dim n As String, Num As Integer ', a, b, c, d, e, f, g As Integer
    n = InputBox("Please Enter Position", "Position")
    Num = Val(n)
    Kru = Kru + 1
    For I = 1 To 8
    Pos(Num, I) = m(I)
    Next
    End Sub

    Sub GotoPos()
    Dim S, T, W, Y As String, Sit, j, I As Integer, Moto1 As Integer
    Dim Motor(8) As Integer, Buffer(8) As String, Speed As String, Moto As String
    S = InputBox("Input Position which you want to go . Oh ! Sorry we can Record Position  Only 20 Positic
", "Go to Position")
    Sit = Val(S)
    If Sit = 0 Then
    MsgBox ("Please Enter Only Number ")
    GotoPos
    End If
    If Sit > 20 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MsgBox ("I allready told you NOT more than 20 . Don't Stupid")
GotoPos
End If
For j = 1 To 8
T = m(j)
Motor(j) = Pos(Sit, j) - m(j)                    ' Find displeacement of Motor
Buffer(j) = "0M-0000"                             ' Set Buffer = 0
Speed = Right(Buffer(j), 4)
If Motor(j) < 0 Then
Moto1 = Motor(j) * -1                            '{
Moto = Str(Moto1)                                '{
ElseIf Motor(j) > 0 Then                                 '       Set displeacement of Motor to be Possitive
Moto = Str(Motor(j))                        '{
ElseIf Motor(j) = 0 Then
Moto = Speed
End If
Y = Moto
W = Buffer(j)
T = Motor(j)
If Len(Moto) = 2 Then
Mid(Buffer(j), 7, 1) = Mid(Moto, 2, 1)
ElseIf Len(Moto) = 3 Then
Mid(Buffer(j), 6, 2) = Mid(Moto, 2, 2)  ' Give Speed of displeacement to Buffer
ElseIf Len(Moto) = 4 Then
Mid(Buffer(j), 5, 3) = Mid(Moto, 2, 3)
ElseIf Len(Moto) = 5 Then
Mid(Buffer(j), 4, 4) = Mid(Moto, 2, 4)
End If
  W = Buffer(j)
    Select Case j
Case 1                                           ' Base part
    If Motor(j) < 0 Then
        Mid(Buffer(j), 3, 1) = "-"
    ElseIf Motor(j) > 0 Then
        Mid(Buffer(j), 3, 1) = "+"
    End If
        Mid(Buffer(j), 1, 1) = j
        m(j) = m(j) + Motor(j)
        Text1(j - 1) = m(j)
Case 2                                           ' Shoulder
    If Motor(j) < 0 Then
        Mid(Buffer(j), 3, 1) = "-"
    ElseIf Motor(j) > 0 Then
        Mid(Buffer(j), 3, 1) = "+"
    End If
        Mid(Buffer(j), 1, 1) = j
        m(j) = m(j) + Motor(j)
        Text1(j - 1) = m(j)
Case 3                                           ' Elbow
    If Motor(j) < 0 Then
        Mid(Buffer(j), 3, 1) = "-"
    ElseIf Motor(j) > 0 Then
        Mid(Buffer(j), 3, 1) = "+"
    End If
        Mid(Buffer(j), 1, 1) = j
        m(j) = m(j) + Motor(j)
        Text1(j - 1) = m(j)
Case 4                                           ' Wrist - Pitch
        Buffer(j + 1) = Buffer(j)
    If Motor(j) < 0 Then
        Mid(Buffer(j), 3, 1) = "-"
        Mid(Buffer(j + 1), 3, 1) = "+"
    ElseIf Motor(j) > 0 Then
        Mid(Buffer(j), 3, 1) = "+"
        Mid(Buffer(j + 1), 3, 1) = "-"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            End If
                Mid(Buffer(j), 1, 1) = j
                Mid(Buffer(j + 1), 1, 1) = j + 1
                m(j) = m(j) + Motor(j)
                Text1(j - 1) = m(j)
                    Buffer(j) = Buffer(j + 1) + Chr(13) + Buffer(j) + Chr(13)
                    W = Buffer(j)
                    Awrite_comm (Buffer(j))
                    GoTo line1
        Case 5                                  ' Wrist - roll
                Buffer(j - 1) = Buffer(j)
            If Motor(j) < 0 Then
                Mid(Buffer(j), 3, 1) = "-"
                Mid(Buffer(j - 1), 3, 1) = "-"
            ElseIf Motor(j) > 0 Then
                Mid(Buffer(j), 3, 1) = "+"
                Mid(Buffer(j - 1), 3, 1) = "+"
            End If
                Mid(Buffer(j), 1, 1) = j
                Mid(Buffer(j - 1), 1, 1) = j - 1
                m(j) = m(j) + Motor(j)
                Text1(j - 1) = m(j)
                    Buffer(j) = Buffer(j) + Chr(13) + Buffer(j - 1) + Chr(13)
                    W = Buffer(j)
                    Awrite_comm (Buffer(j))
                    GoTo line1
        Case 6                                  ' Slide Base
            If Motor(j) < 0 Then
                Mid(Buffer(j), 3, 1) = "-"
            ElseIf Motor(j) > 0 Then
                Mid(Buffer(j), 3, 1) = "+"
            End If
                Mid(Buffer(j), 1, 1) = j
                m(j) = m(j) + Motor(j)
                Text1(j - 1) = m(j)
        Case 7                                  ' Stand Rotate
            If Motor(j) < 0 Then
                Mid(Buffer(j), 3, 1) = "-"
            ElseIf Motor(j) > 0 Then
                Mid(Buffer(j), 3, 1) = "+"
            End If
                Mid(Buffer(j), 1, 1) = j
                m(j) = m(j) + Motor(j)
                Text1(j - 1) = m(j)
        Case 8
            If Motor(j) < 0 Then
                Mid(Buffer(j), 3, 1) = "-"
                Mid(Buffer(j), 4, 4) = "2000"
            ElseIf Motor(j) > 0 Then
                Mid(Buffer(j), 3, 1) = "+"
                Mid(Buffer(j), 4, 4) = "2000"
            End If
                Mid(Buffer(j), 1, 1) = j
                m(j) = Pos(Sit, j)
                W = Buffer(j)
          End Select
        W = Buffer(j)
        Buffer(j) = Buffer(j) + Chr$(13)
        Awrite_comm (Buffer(j))
line1:  Next
        End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Sub Base1(I As Integer)
    Dim T As Integer
    Static Buffer$
```

```
If nCid = 0 Then Exit Sub
    Select Case I
Case 49                                  ' key 1
 Buffer$ = "1M-0012" + Chr$(13)
 M1 = M1 - 12
 Text1(0) = M1
Awrite_comm (Buffer$)
Case 113                                 ' key q
 Buffer$ = "1M+0012" + Chr$(13)
 M1 = M1 + 12
 Text1(0) = M1
 Awrite_comm (Buffer$)
 Case 50                                 ' key 2
 Buffer$ = "2M-0008" + Chr$(13)
 M2 = M2 - 8
 Text1(1) = M2
 Awrite_comm (Buffer$)
 Case 119                                ' key w
 Buffer$ = "2M+0012" + Chr$(13)
 M2 = M2 + 12
 Text1(1) = M2
 Awrite_comm (Buffer$)
 Case 51                                 ' key 3
 Buffer$ = "3M+00010" + Chr$(13)
 M3 = M3 + 10
 Text1(2) = M3
 Awrite_comm (Buffer$)
 Case 101                                ' key e
 Buffer$ = "3M-0013" + Chr$(13)
 M3 = M3 - 13
 Text1(2) = M3
 Awrite_comm (Buffer$)
 Case 52                                 ' key 4
 Buffer$ = "4M+0010" + Chr$(13) + "5M-0010" + Chr$(13)
 M4 = M4 + 10
 Text1(3) = M4
 Awrite_comm (Buffer$)
 Case 114                                ' key r
 Buffer$ = "4M-0010" + Chr$(13) + "5M+0010" + Chr$(13)
 M4 = M4 - 10
 Text1(3) = M4
 Awrite_comm (Buffer$)
 Case 53                                 ' key 5
 Buffer$ = "4M+0010" + Chr$(13) + "5M+0010" + Chr$(13)
 M5 = M5 + 10
 Text1(4) = M5
 Awrite_comm (Buffer$)
 Case 116                                ' key t
 Buffer$ = "4M-0010" + Chr$(13) + "5M-0010" + Chr$(13)
 M5 = M5 - 10
 Text1(4) = M5
 Awrite_comm (Buffer$)
 Case 54                                 ' key 6
 Buffer$ = "6M+0012" + Chr$(13)
 M6 = M6 + 12
 Text1(5) = M6
 Awrite_comm (Buffer$)
 Case 121                                ' key y
 Buffer$ = "6M-0012" + Chr$(13)
 M6 = M6 - 12
 Text1(5) = M6
 Awrite_comm (Buffer$)
 Case 55                                 ' key 7
 Buffer$ = "7M+0010" + Chr$(13)
 M7 = M7 + 10
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Text1(6) = M7
Awrite_comm (Buffer$)
Case 117                              ' key u
Buffer$ = "7M-0010" + Chr$(13)
M7 = M7 - 10
Text1(6) = M7
Awrite_comm (Buffer$)
Case 111                              ' key o
Buffer$ = "8M-2000" + Chr$(13)
Awrite_comm (Buffer$)
Case 99                               ' key c
Buffer$ = "8M+2000" + Chr$(13)
Awrite_comm (Buffer$)

 End Select


 Exit Sub
End Sub


Sub Awrite_comm(Buffer$)
  Dim T, ij As Integer, Lpstat As COMSTAT
  Static buffer1$
  T = GetCommError(nCid, Lpstat)       ' Get current stats
  If Lpstat.cbOutQue < 1024 Then      ' Space left in Buffer ?
  If Ret >= 0 Then
  Ret = WriteComm(nCid, Buffer$, Len(Buffer$))     'Place chars in buffer
  If Ret <= 0 Then Ret = -Ret              ' Ret is the number of char
  End If

  Buffer$ = Mid$(Buffer$, Ret + 1) ' Remove xmited chars from buffer
  End If
  Exit Sub
  End Sub

   Sub Close_comm()
      Dim Ret As Integer
       Ret = Closecomm(nCid)
   nCid = 0         'Clear nCid to stop Poll_comm
   End Sub

       Sub Comm_Open(Commstring$)
        Dim T, c As Integer
        Dim Commport As String
       '-----------------------------------------------
       '--Close any previously open communication ports
       '-----------------------------------------------------------------


        T = Closecomm(2)
        T = Closecomm(3)
        T = Closecomm(4)
        T = Closecomm(1)
       '-----------------------------------------------
       '--Open the communication port,get nCid
       '--nCid < 0 if error,refer to IE_constants
       '-- -2 = port already open,try closing it anyway.
       '-------------------------------------------------------------
        Commport$ = Left$(Commstring$, InStr(Commstring$, ":") - 1)
        nCid = Opencomm(Commport$, 1024, 1024)
        If nCid < 0 Then
          MsgBox "Unable to Open Comm Device:" + Str$(nCid), 16
          If nCid = -2 Then T = Closecomm(Mid(Commstring$, 4, 1))
          End
        End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'-------------------------------------------------
'--Load the CommDCB with parameters from Commstring
 c = BuildCommDCB(Commstring$, LpDCB)
 If c Then
    MsgBox "Unable to Build Comm DCB ", 16
 End
 End If
'-------------------------------------------------
'-- Set the port state
'-------------------------------------------------
LpDCB.Id = Chr$(nCid)
If (SetcommState(LpDCB)) Then
MsgBox "Unable to set Comm State", 16
End
End If
End Sub



Private Sub end_Click()
   Close_comm
   Form8.Caption = "Communication Port Closed"
End Sub


Sub Form_KeyPress(keyascii As Integer)
   RoverIdx = 0
   Picture1.Picture = picclip2.GraphicCell(RoverIdx)
   Timer1.Enabled = True
   PanLandscape PanAmount

   If keyascii = 49 Then
   Base1 (49)
   ElseIf keyascii = 113 Then
   Base1 (113)
   ElseIf keyascii = 50 Then
   Base1 (50)
   ElseIf keyascii = 119 Then
   Base1 (119)
   ElseIf keyascii = 51 Then
   Base1 (51)
   ElseIf keyascii = 101 Then
   Base1 (101)
   ElseIf keyascii = 52 Then
   Base1 (52)
   ElseIf keyascii = 114 Then
   Base1 (114)
   ElseIf keyascii = 53 Then
   Base1 (53)
   ElseIf keyascii = 116 Then
   Base1 (116)
   ElseIf keyascii = 54 Then
   Base1 (54)
   ElseIf keyascii = 121 Then
   Base1 (121)
   ElseIf keyascii = 55 Then
   Base1 (55)
   ElseIf keyascii = 117 Then
   Base1 (117)
   ElseIf keyascii = 111 Then
   Base1 (111)
   ElseIf keyascii = 99 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Base1 (99)
End If
End Sub



Private Sub Form_Load()
 Dim I  As Integer, S As String

 S = " Now , this is a sample of control Robot and this is Teach Position menu. "
 Form8.Font.Bold = True
 FontSize = 10
 Form8.Print Tab(2), S
 FontSize = 7
 Form8.Print
 Form8.Print "  1/Q    Move Base          Right /   Left"
 Form8.Print
 Form8.Print "  2/W   Move Shoulder       Up    /   Down"
 Form8.Print
 Form8.Print "  3/E    Move Elbow         Up    /   Down"
 Form8.Print
 Form8.Print "  4/R    Move Wrist-Pitch   Up    /  Down"
 Form8.Print
 Form8.Print "  5/T    Move Wrist-Roll    Left  /  Right"
 Form8.Print
 Form8.Print "  6/Y    Move Slide Base     +    /  -"
 Form8.Print
 Form8.Print "  O/C    Open / Close Gripper"
 Form8.Print

 Text1(0).Move 30, 200, 40, 18
 Text1(1).Move 90, 200, 40, 18
 Text1(2).Move 150, 200, 40, 18
 Text1(3).Move 210, 200, 40, 18
 Text1(4).Move 270, 200, 40, 18
 Text1(5).Move 330, 200, 40, 18
 Text1(6).Move 390, 200, 40, 18
 Font.Bold = True
 FontSize = 9
 Print Tab(40); " Position of Motor "
 Print ""
 Print Spc(8); "M1"; Spc(8); "M2"; Spc(9); "M3"; Spc(9); "M4"; Spc(8); "M5"; Spc(8); "M6"; Spc(8); "M7"
 End Sub



 Private Sub Picture1_Click()

 End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub start_Click()
Comm_Open Commstring1$
If nCid Then Form8.Caption = "Communication Port Open"
```

```
End Sub

    Private Sub Text1_Change(Index As Integer)

    End Sub


Private Sub Timer1_Timer()
 '-----------------------------------------
 ' Pan the landscape until the timer is disabled.
 '-----------------------------------------
 PanLandscape PanAmount
 End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Option Explicit

Declare Function Inp Lib "InpOut.dll" (ByVal Port%) As Integer
Declare Sub Out Lib "InpOut.dll" (ByVal Port%, ByVal Value%)

'Error Flags
Global nCid, Pos(20, 8), Kru As Integer, Text As String
Global Const Commstring1 = "COM2:9600,n,8,2"
Global Const Commstring2 = "COM4:300,e,7,1"
Global m(8) As Integer
Global Const CE_RXOVER = &H1     'Receive Queue overflow
Global Const CE_OVERRUN = &H2    'Receive  Overrun Error
Global Const CE_RXPARITY = &H4   'Receive  Parity Error
Global Const CE_FRAME = &H8      'Receive  Framing error
Global Const CE_BREAK = &H10     'Break Detected
Global Const CE_CTSTO = &H20     'CTS Timeout
Global Const CE_DSRTO = &H40     'DSR Timeout
Global Const CE_RLSDTO = &H80    'RLSD Timeout
Global Const CE_TXFULL = &H100   ' TX Queue is full
Global Const CE_PTO = &H200      'LPTx   Timeout
Global Const CE_IOE = &H400      'LPTx   I/O Error
Global Const CE_DNS = &H800      'LPTx   Device not selected
Global Const CE_OOP = &H1000     'LPTx   Out-of-Paper
Global Const CE_MODE = &H8000    'Requested mode unsupported

Global Const IE_BADID = (-1)     'Invalid or unsupported ID
Global Const IE_OPEN = (-2)      'Device Already Open
Global Const IE_NOPEN = (-3)     'Device Not Open
Global Const IE_MEMORY = (-4)    'Unable to allocate queues
Global Const IE_DEFAULT = (-5)   'Error in default parameters
Global Const IE_HARDWARE = (-10) 'Hardware Not Present
Global Const IE_BYTESIZE = (-11) 'Illegal Byte Size
Global Const IE_BAUDRATE = (-12) 'Unsupported BaudRate

  Type DCB
  Id As String * 1
  BaudRate As Integer
  ByteSize  As String * 1
  Parity As String * 1
  StopBits As String * 1
  RlsTimeout As Integer
  CtsTimeout As Integer
  DsrTimeout As Integer
  Bit1s As String * 1
  Bit2s As String * 1
  XonChar As String * 1
  XoffChar As String * 1
  XonLim As Integer
  XoffLim As Integer
  PeChar As String * 1
  EofChar As String * 1
  EvtChar As String * 1
  TxDeLay As Integer
End Type

Type COMSTAT
  Bits As String * 1
  cbInQue As Integer
  cbOutQue As Integer
End Type

' COMM declarations
Declare Function Opencomm Lib "User" (ByVal LpComname As String, ByVal wInQueue As Integer, B
yVal wOutQueue As Integer) As Integer
Declare Function SetcommState Lib "User" (LpDCB As DCB) As Integer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

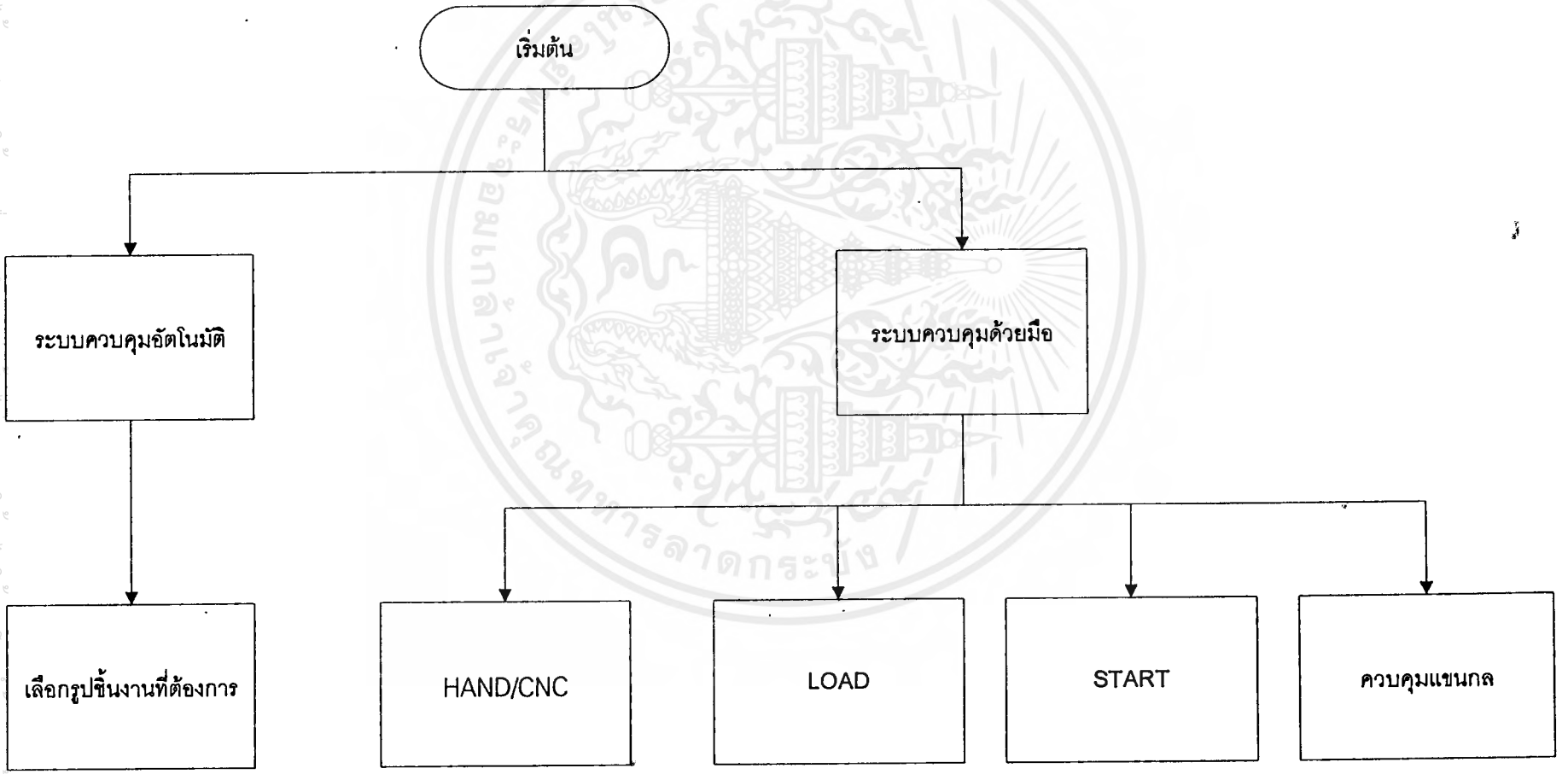
```
Declare Function ReadComm Lib "User" (ByVal nCid As Integer, ByVal Lpbuf As String, ByVal nSize
As Integer) As Integer
Declare Function WriteComm Lib "User" (ByVal nCid As Integer, ByVal Lpbuf As String, ByVal nSize
As Integer) As Integer
Declare Function Closecomm Lib "User" (ByVal nCid As Integer) As Integer
Declare Function BuildCommDCB Lib "User" (ByVal LpDef As String, LpDCB As DCB) As Integer
Declare Function GetCommError Lib "User" (ByVal nCid As Integer, Lpstat As COMSTAT) As Integer

' Functions for Scrolling a Window

Declare Sub ScrollWindow Lib "User" (ByVal hWnd As Integer, ByVal XAmount As Integer, ByVal Ya
mount As Integer, LpRect As Any, lpcliprect As Any)
Declare Sub UpdateWindow Lib "User" (ByVal hWnd As Integer)

' Seting = 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

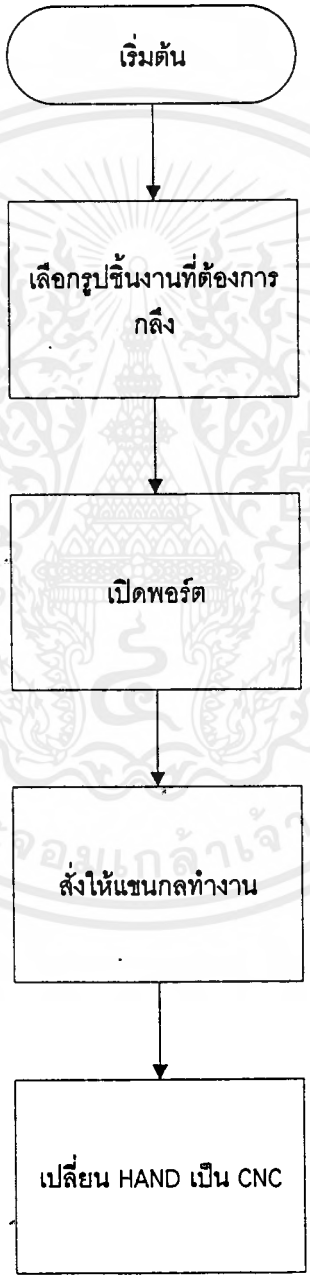
ภาคผนวก ข.
Flowchart การทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart แสดงระบบรวม

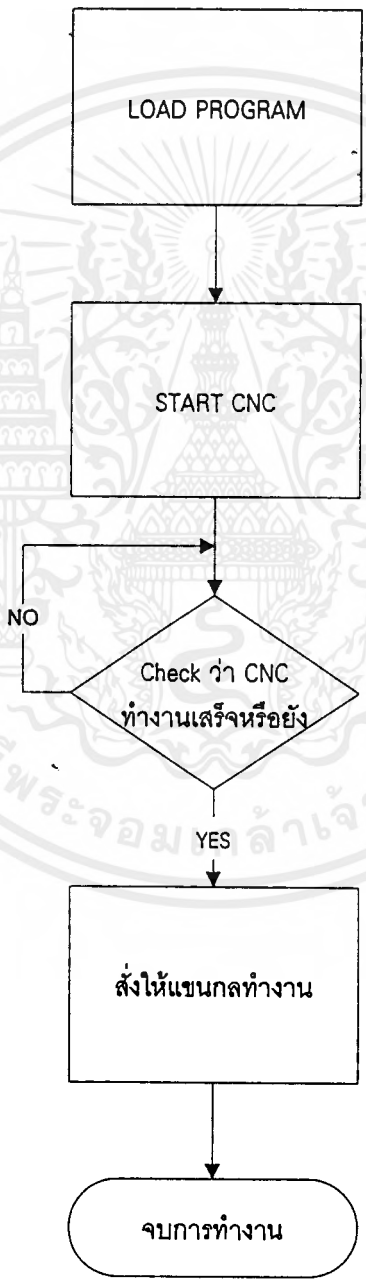


Flowchart แสดงโพรซีเดอร์ระบบควบคุมอัตโนมัติ



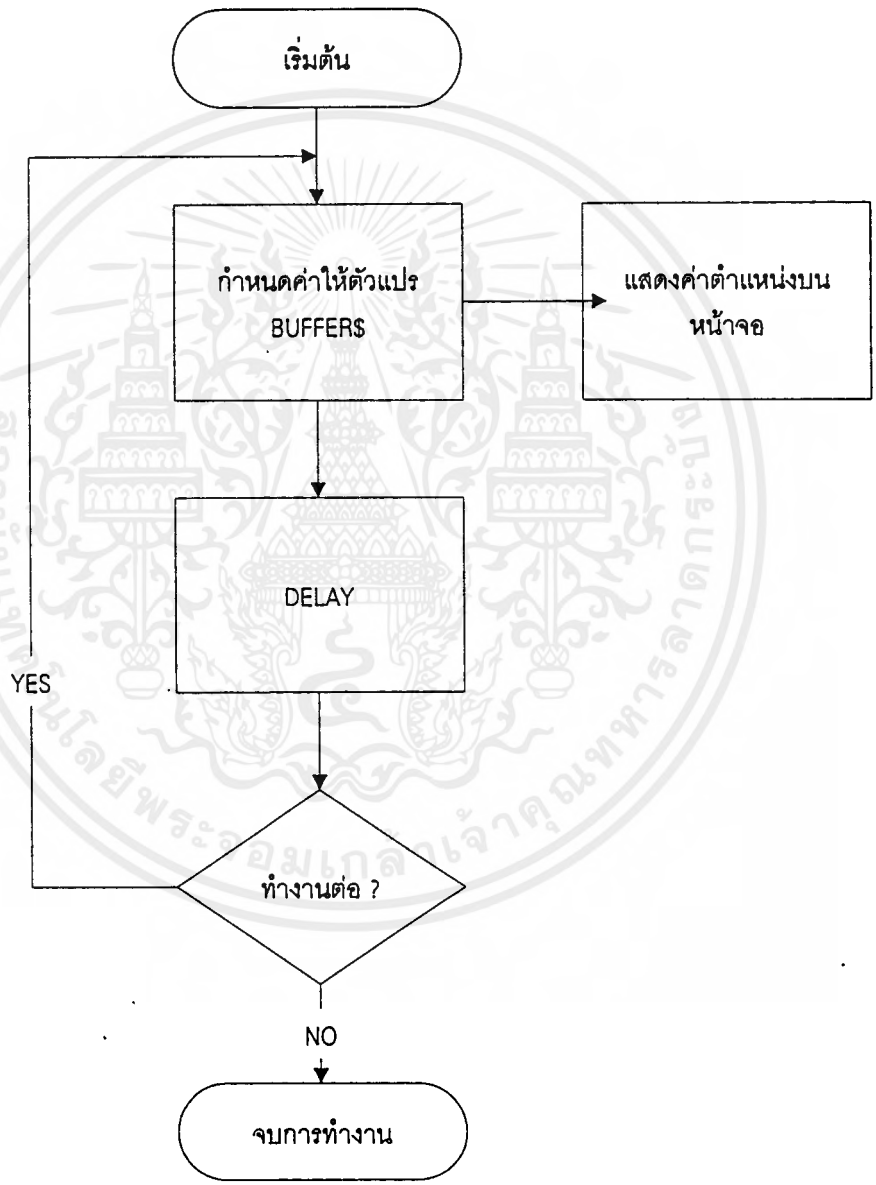
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart แสดงโพรซีเดอร์ ระบบควบคุมอัตโนมัติ (ต่อ)



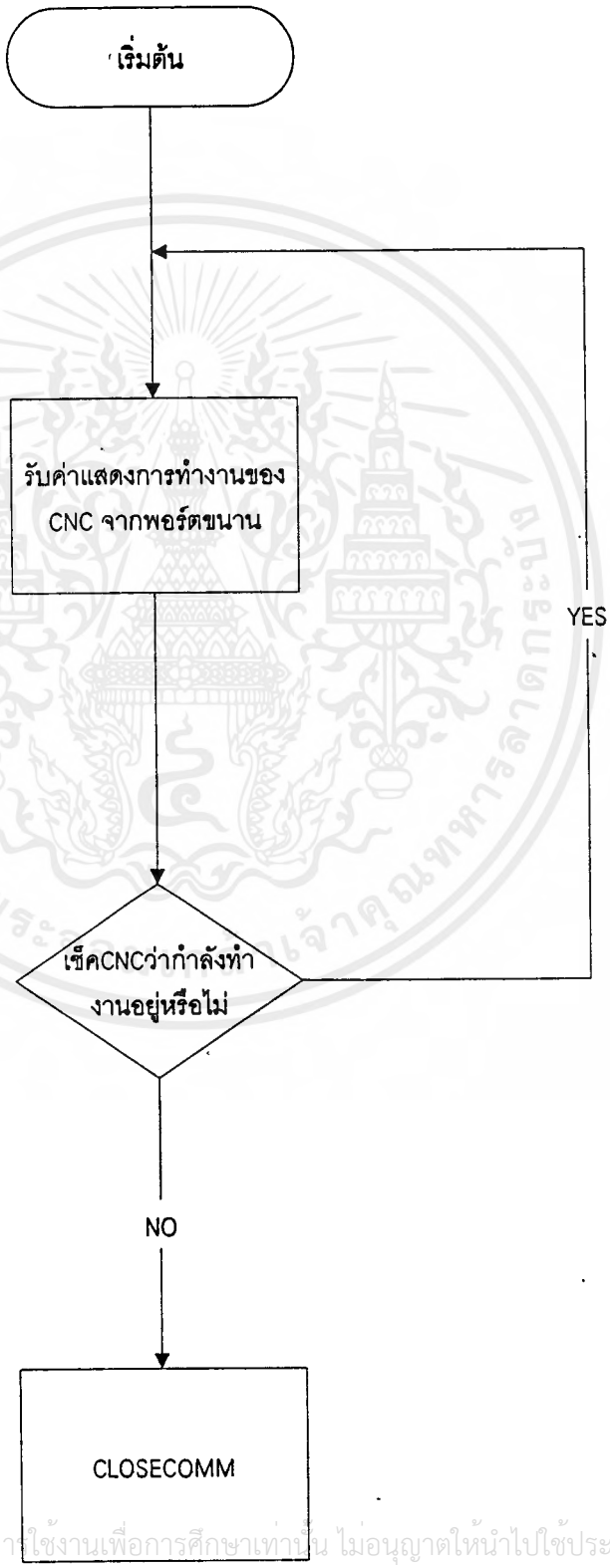
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart แสดงโพรซีเดอร์การทำงานของแขนกล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart แสดงโพรซีเดอร์ checkend

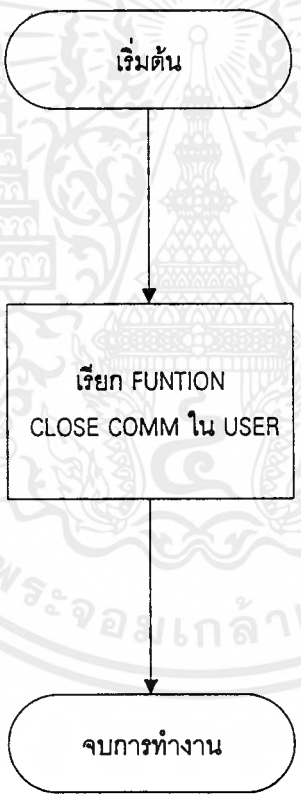


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart แสดงโพรซีเดอร์ checkend (ต่อ)

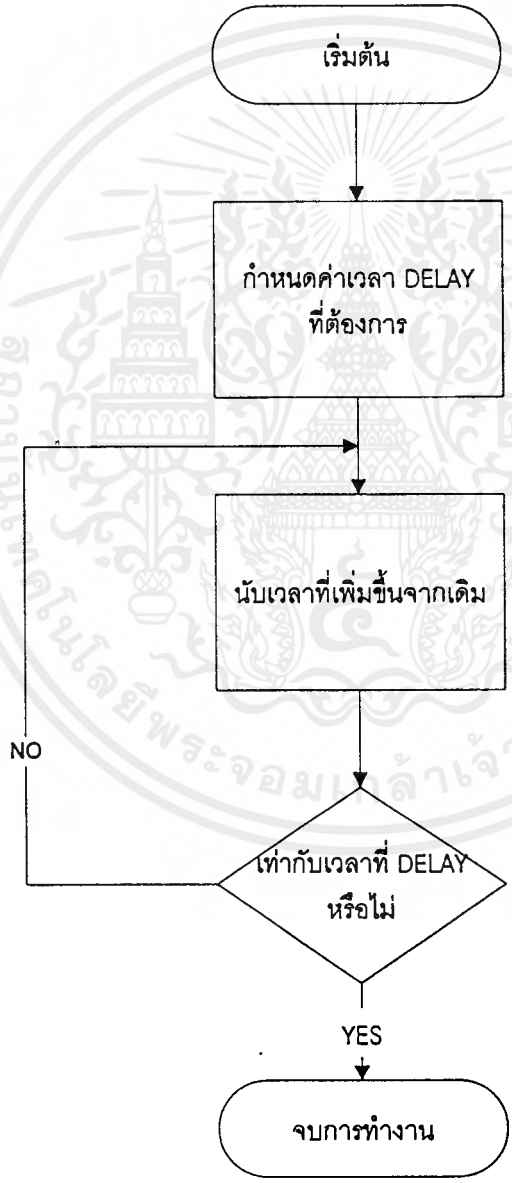
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart แสดงโพรซีเดอร์ closecomm



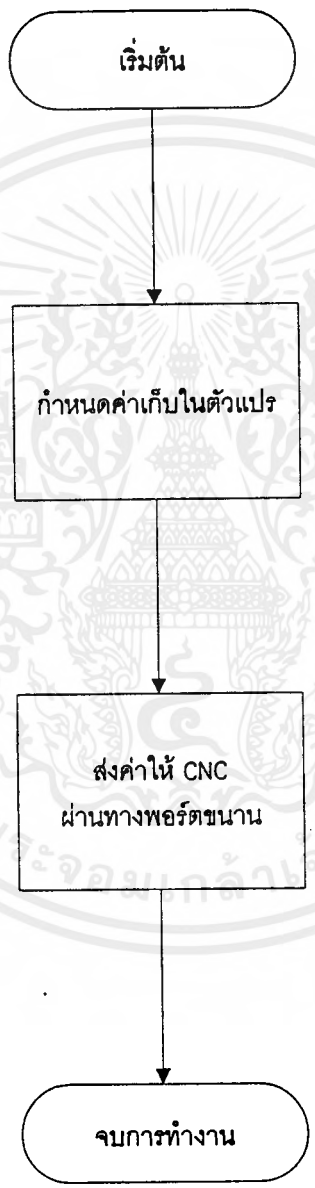
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart แสดงโพรซีเดอร์ delay



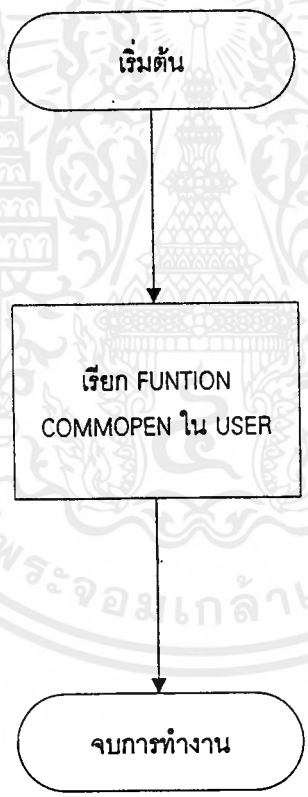
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart แสดงโพรซีเดอร์ hand/cnc



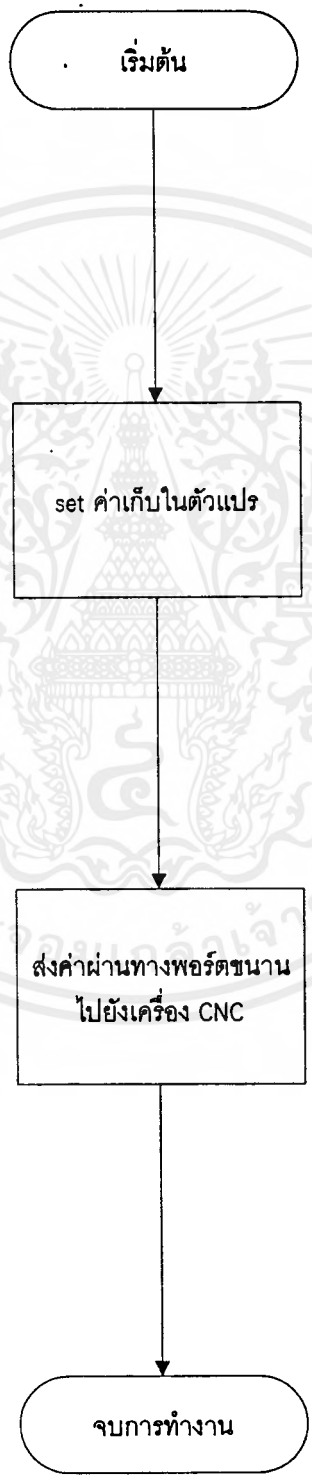
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart แสดงโพรซีเดอร์ open port



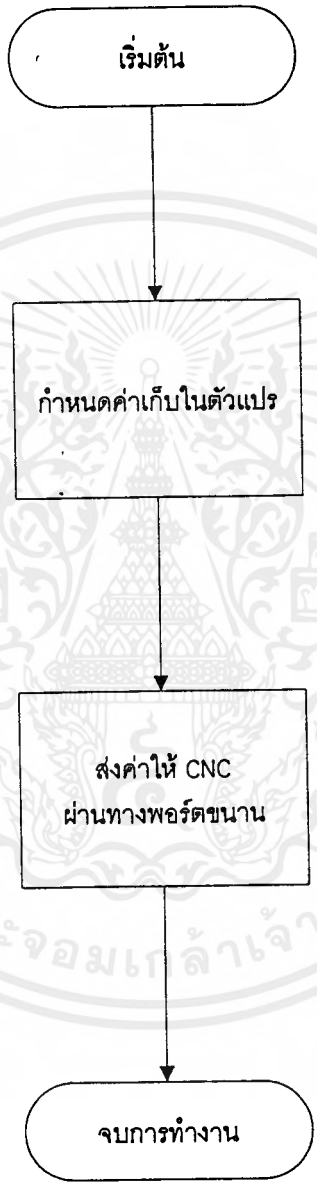
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart แสดงโพรซีเดอร์ setinitial



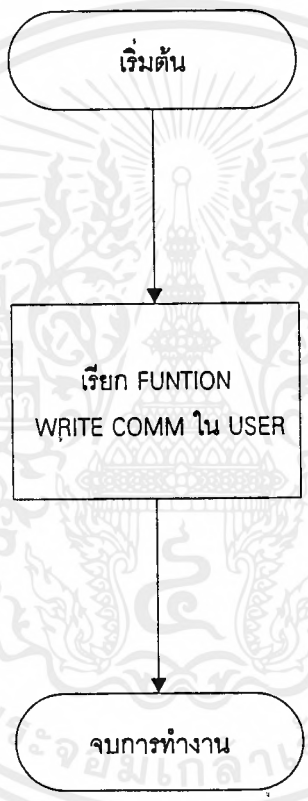
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart แสดงโพรซีเดอร์ start



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

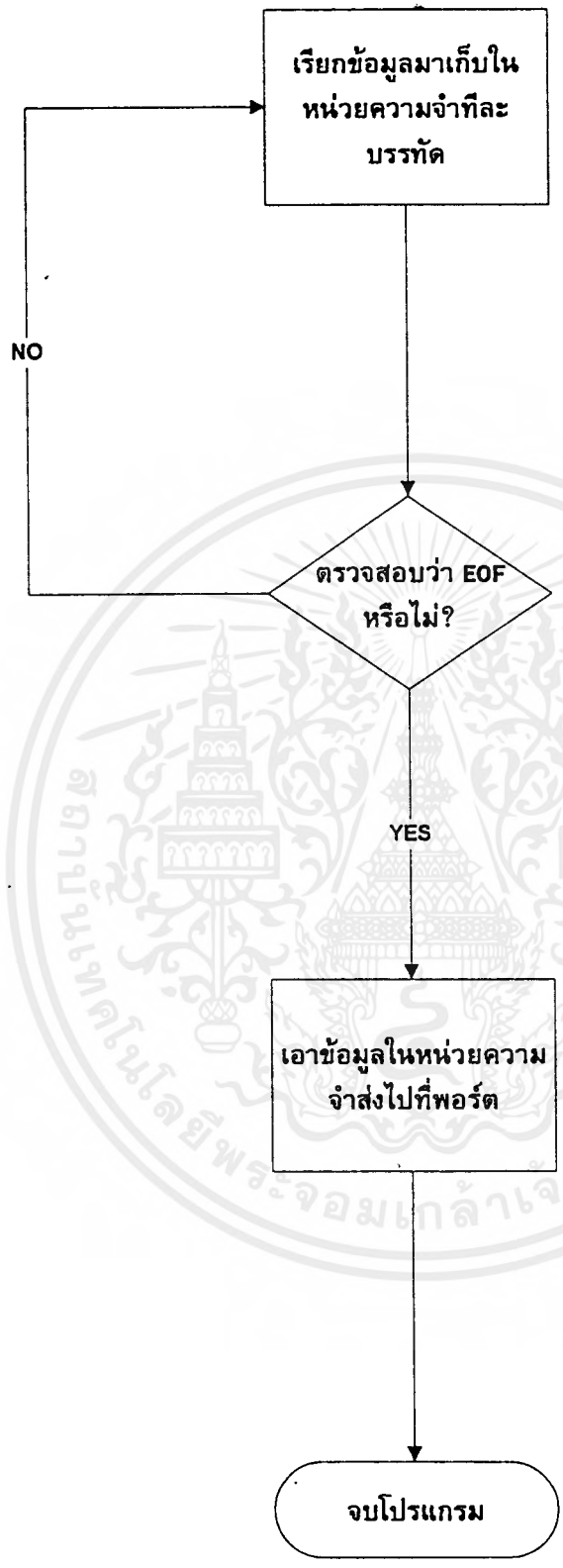
Flowchart แสดงโพรซีเดอร์ writecomm



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FLOWCHART ของโปรแกรมส่งข้อมูลทางพอร์ตอนุกรมและขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้