สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมอัตโนมัติขนาดเล็ก

MICRO CIM

(Micro Computer Integrated Manufacturing)

T050205

โดย

นายชัยชนะ อยู่ยงสินธุ์

นายผดุงศักดิ์ ชัยรุ่งเรืองสิน

อาจารย์ที่ปรึกษา

อาจารย์เทพจิตร์ เชยโภคา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....
เลขทะเบียน.....50205
วัน,เดือน,ปี.27 เม.ย. 2547

.b.....
.i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมอัตโนมัติขนาดเล็ก

MICRO CIM

โดย

นายชัยชนะ อยู่ยงสินธุ์ รหัสประจำตัว 43015301

นายผดุงศักดิ์ ชัยรุ่งเรืองสิน รหัสประจำตัว 43015316

อาจารย์ที่ปรึกษา

.....

(อาจารย์เทพจิตร์ เชยโภคา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมอัตโนมัติขนาดเล็ก

นายชัยชนะ อยู่ยงสินธุ์
นายผดุงศักดิ์ ชัยรุ่งเรืองสิน

อาจารย์ที่ปรึกษา อาจารย์เทพจิตร์ เชยโภคา
ปีการศึกษา 2545

บทคัดย่อ

โครงงานนี้เป็นการสร้างโปรแกรมเพื่อควบคุมการทำงานของ CNC-Compact 5 และ แขนกล ให้ทำงานอัตโนมัติ ซึ่งเรียกว่า Micro Computer Integrated โดยใช้ Visual Basic 6 ผู้ควบคุมสามารถใช้โปรแกรมที่เขียนขึ้นอย่างง่ายดาย โดยผู้สั่งงานสามารถใช้โปรแกรมนี้โดยง่ายดาย โปรแกรมจะสั่งให้แขนกลหยิบชิ้นงานที่ต้องการกลึง แล้วนำมาใส่เครื่อง CNC จากนั้น CNC จะทำการกลึงตามที่ป้อนรหัส G-Code ไว้ เมื่อเสร็จงานแขนกลจะนำชิ้นงานที่กลึงเรียบร้อยแล้วไปเก็บไว้ในจุดที่เก็บชิ้นงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Micro Computer Integrated Manufacturing

Mr.Chaichana Yooyongsin

Mr.Phadungsak Chairungreagsin

Advisor Mr.Teppajit Choeypoca

Academic year 2002

Abstract

This project is designed to create the program for controlling of CNC-Compact 5 and Robot automatic system. Those called Micro Computer Integrated by using Visual Basic 6. The operator can use this program easily. The robot 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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 ทฤษฎีเบื้องต้นเกี่ยวกับเครื่องจักรกลซีเอ็นซี	
1.1 เครื่องจักรกลเอ็นซีและการพัฒนาของซีเอ็นซี	1
1.2 การทำงานของเครื่องจักรกลเอ็นซีและซีเอ็นซี	2
1.3 การขับป้อน (Feed Drives)	3
1.4 บอลส์สกรู (Ball Screws)	5
1.5 ระบบวัดขนาด	6
1.6 เพลางาน (Work Spindle)	9
1.7 เครื่องมือ (Tool)	10
1.8 ชนิดของการควบคุม	11
1.9 ลักษณะเฉพาะของเครื่อง CNC-Compact 5	12
บทที่ 2 หุ่นยนต์ (Robotics)	
2.1 การเคลื่อนย้ายโดยหุ่นยนต์	17
2.2 เครื่องจักรกลแบบ Hard nad Soft Automation	17
2.3 การแบ่งประเภทหุ่นยนต์ (Robot Classification)	18
2.4 ลักษณะเฉพาะของหุ่นยนต์ (Robot Specification)	21
2.5 ความสามารถในการซ้ำตำแหน่งเดิม, ความละเอียดและความแม่นยำ	22
บทที่ 3 การเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอก	
3.1 ลักษณะทางกายภาพของพอร์ตขนาน	23
3.2 การติดต่อผ่านพอร์ตขนาน (Parallel Interfacing)	29
3.3 การติดต่อผ่านพอร์ตอนุกรม (Serial Interfacing)	33
3.4 คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ	35
บทที่ 4 สรุปและข้อเสนอแนะ	42
ภาคผนวก ก.	
ภาคผนวก ข.	
ภาคผนวก ค.	
หนังสืออ้างอิง	
กิตติกรรมประกาศ	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้า

รูปที่ 1.1 เครื่องกลึงซีเอ็นซีแบบ 2 แกน 3
รูปที่ 1.2 การเคลื่อนที่ตัดเฉือนของเครื่องมือตัด 3
รูปที่ 1.3 Diagram ระบบขับป้อน 4
รูปที่ 1.4 ส่วนประกอบของมอเตอร์กระแสตรง 5
รูปที่ 1.5 การขับป้อนของโต๊ะงาน 5
รูปที่ 1.6 ลักษณะโครงสร้างภายในของชุดบอลล์สกรู 6
รูปที่ 1.7 การวัดตำแหน่งโดยตรง 7
รูปที่ 1.8 การวัดตำแหน่งทางอ้อม 7
รูปที่ 1.9 การวัดตำแหน่งโดยสัมบูรณ์ 8
รูปที่ 1.10 การวัดตำแหน่งแบบต่อเนื่อง 9
รูปที่ 1.11 เพลางานของเครื่องกลึง 9
รูปที่ 1.12 หัวจับงานกลึง 10
รูปที่ 1.13 ชุดยันศูนย์ท้ายแท่นและแท่นประคองงานกลึง 10
รูปที่ 1.14 องค์ประกอบของเครื่องมือที่ใช้ในงานกลึง 10
รูปที่ 1.15 อินเสิร์ตแบบถอดเปลี่ยนได้รูปทรงต่างๆ 11
รูปที่ 1.16 แสดงส่วนรายละเอียดของเครื่อง CNC-Compact 5 12
รูปที่ 1.17 Belt pulley drive 14
รูปที่ 2.1 แขนกลข้อต่อหมุน (Scorbot-ER 5 plus) 17
รูปที่ 2.2 หุ่นยนต์พิกัดทรงกระบอก 20
รูปที่ 2.3 หุ่นยนต์พิกัดทรงกลม 20
รูปที่ 2.4 หุ่นยนต์ SCARA 21
รูปที่ 2.5 หุ่นข้อต่อหมุน 21
รูปที่ 2.6 ตำแน่งของ Tool 22
รูปที่ 3.1 แสดงไดอะแกรมเวลาของการส่งข้อมูลไปยังเครื่องพิมพ์ 23
รูปที่ 3.2 แสดงระบบบัสภายในของพอร์ตขนาน 25
รูปที่ 3.3 วงจรภายในของพอร์ต Data 27
รูปที่ 3.4 วงจรภายในของพอร์ต Control 28
รูปที่ 3.5 แสดงวงจรภายในของพอร์ตสถานะ 28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 3.6 แสดงขาของพอร์ตขนาน 30

รูปที่ 3.7 แสดงไทมิ่งไดอะแกรมของการส่งข้อมูลแบบซิงโครนัส 33

รูปที่ 3.8 แสดงไทมิ่งไดอะแกรมของการส่งข้อมูลแบบอะซิงโครนัส 34

รูปที่ 3.9 แสดงตำแหน่งขาของพอร์ตอนุกรม (RS-232) 36

รูปที่ 3.10 แสดงการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอก 37

รูปที่ 3.11 แสดงการเชื่อมต่อขา DB-9 เข้ากับพอร์ตของ CNC-Compact 5 38

รูปที่ 3.12 แสดงแบบฟอร์มการป้อน G-Code และ M-Code 41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 1.1 เปรียบเทียบการทำงานระหว่างเครื่องจักรกลทั่วๆ ไปกับเครื่องจักรกลซีเอ็นซี 1

ตารางที่ 1.2 แสดงค่าขนาดของอินพุตสูงสุดที่สามารถป้อนให้แก่ CNC-Compact 5 16

ตารางที่ 2.1 ชนิดของข้อต่อหุ่นยนต์ 18

ตารางที่ 2.2 แสดงขอบเขตการทำงานของหุ่นยนต์ที่ขึ้นอยู่กับแกนหลัก 19

ตารางที่ 2.3 แสดงลักษณะเฉพาะของหุ่นยนต์ (Scorbot – ER 5 plus) 22

ตารางที่ 3.1 แสดงสัญญาณที่สำคัญของพอร์ตขนานที่ใช้ติดต่อกับเครื่องพิมพ์ 24

ตารางที่ 3.2 แสดงแอดเดรสของพอร์ตขนาน 29

ตารางที่ 3.3 แสดงการเชื่อมต่อระหว่าง Parallel Port กับ DNC – Interface 31

ตารางที่ 3.4 แสดงบิตพาริตี้ของข้อมูล 35

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
ทฤษฎีเบื้องต้นเกี่ยวกับเครื่องจักรกลซีเอ็นซี

1.1 เครื่องจักรกลเอ็นซีและพัฒนาการของซีเอ็นซี

เอ็นซี (NC) ย่อมาจากคำว่า Numerical Control หมายถึง การควบคุมเครื่องจักรกลด้วยตัวเลขและตัวอักษร กล่าวคือการทำงานของเครื่องจักรจะถูกควบคุมโดยรหัสคำสั่งที่ประกอบด้วย ตัวเลข ตัวอักษรและสัญลักษณ์อื่นๆ ซึ่งจะถูกแปลงเป็นสัญญาณไฟฟ้าส่งให้กับมอเตอร์หรืออุปกรณ์อื่นๆ เพื่อทำให้เครื่องจักรกลทำงานตามขั้นตอนที่ต้องการ

ซีเอ็นซี (CNC) ย่อมาจากคำว่า Computerized Numerical Control ระบบควบคุมซีเอ็นซีแบบนี้จะมีคอมพิวเตอร์ที่มีความสามารถสูงเพิ่มเข้าไปภายในระบบ สามารถจัดการกับข้อมูลที่ป้อนเข้าไปในระบบซีเอ็นซีและประมวลผลข้อมูลเพื่อนำผลลัพธ์ที่ได้ไปควบคุมการทำงานของเครื่องจักรกล

ลำดับที่	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

ตาราง 1.1 เปรียบเทียบการทำงานระหว่างเครื่องจักรกลทั่วๆ ไป กับเครื่องจักรกลซีเอ็นซี

ข้อดีและข้อเสียของเครื่องจักรกลซีเอ็นซี

ข้อดีของเครื่องซีเอ็นซี

1. มีความยืดหยุ่นสูง การเปลี่ยนงานใหม่จะแก้ไขหรือเปลี่ยนแปลงเฉพาะโปรแกรมเท่านั้น
2. ความเที่ยงตรง (Accuracy) จะอยู่ระดับเดียวกันตลอดช่วงความเร็วรอบและอัตราป้อนที่ใช้ในการผลิต
3. ใช้เวลาในการผลิตสั้นกว่า
4. สามารถผลิตชิ้นงานที่มีรูปทรงซับซ้อนได้ง่าย
5. การปรับตั้งเครื่องจักรกระทำได้ง่ายกว่า ใช้เวลาน้อยกว่าการผลิตด้วยวิธีอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.หลีกเลี่ยงความจำเป็นที่ต้องใช้ช่างควบคุมที่มีทักษะและประสบการณ์สูง

7.ช่างควบคุมมีเวลาว่างจากการควบคุมเครื่อง สามารถที่จะจัดเตรียมงานอื่นๆไว้ล่วงหน้าได้

8.การตรวจสอบคุณภาพไม่จำเป็นต้องกระทำทุกขั้นตอนและทุกชิ้นงาน

ข้อเสียของเครื่องจักรกลซีเอ็นซี

1.ราคาเครื่องจักรค่อนข้างสูง

2.การบำรุงรักษาค่อนข้างซับซ้อน

3.การซ่อมบำรุงต้องใช้ช่างที่มีประสบการสูงและผ่านการฝึกอบรมมาโดยเฉพาะ

4.จำเป็นต้องใช้ช่างเขียนโปรแกรม (Part Programmer) ที่มีทักษะสูงและฝึกอบรมมาโดยเฉพาะ

5.ราคาของเครื่องมือต่างๆ ที่ใช้ในกระบวนการตัดเฉือน เช่น แกนเพลายึดใบมีด มีดกลึงแบบใช้อินเสิร์ต เป็นต้น มีราคาสูง

1.2 การทำงานของเครื่องจักรกลเอ็นซีและซีเอ็นซี

หลักการทำงานของเครื่องจักรกลเอ็นซีและซีเอ็นซี จะคล้ายคลึงกับเครื่องจักรทั่วไปเพียงแต่ว่าระบบควบคุมเอ็นซีของเครื่องจะทำงานในขั้นตอนต่างๆแทนช่างควบคุมเครื่อง อย่างไรก็ตามก่อนที่เครื่องจักรกลเอ็นซีจะทำงานได้นั้น จะต้องป้อนโปรแกรมเข้าไปในระบบควบคุมของเครื่องผ่านแป้นพิมพ์ หรือเทปแม่เหล็กก็ได้

เมื่อระบบควบคุมอ่านโปรแกรมที่ป้อนเข้าไปแล้ว ก็จะนำไปควบคุมส่วนต่างๆของเครื่องจักรกล เช่น มอเตอร์ป้อน (Feed Motor) ของแท่นเลื่อน แต่เนื่องจากสัญญาณที่ออกจากระบบควบคุมมีกำลังน้อย ไม่สามารถไปหมุนขับมอเตอร์ให้ทำงานได้ ดังนั้นจึงต้องส่งสัญญาณนี้เข้าไปในภาคขยายสัญญาณของระบบขับ (Drive Amplified) และส่งต่อไปยังมอเตอร์ป้อนแนวแกนที่ต้องการเคลื่อนที่

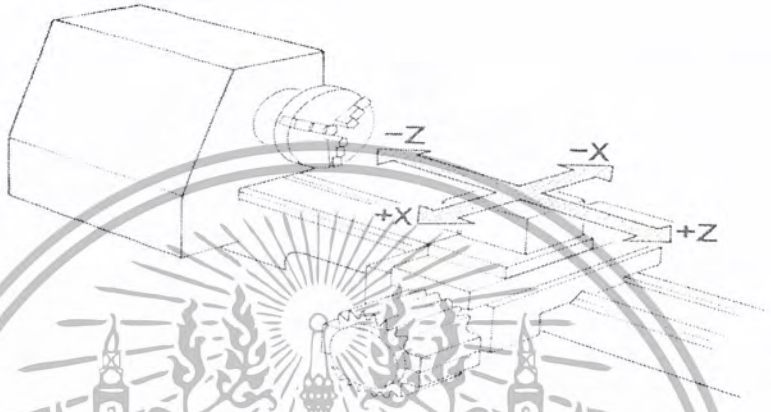
เครื่องจักรกลซีเอ็นซี จะประกอบด้วยองค์ประกอบใหญ่ๆ อยู่ 2 ส่วนคือ

1. **เครื่องจักรกล** เป็นส่วนที่ทำหน้าที่ ตัดเฉือนชิ้นงานตามขั้นตอนการทำงานที่กำหนดไว้
2. **ระบบซีเอ็นซี** เป็นส่วนที่ทำหน้าที่ควบคุมขั้นตอนการตัดเฉือนทั้งหมดองค์ประกอบของเครื่องเอ็นซีและซีเอ็นซีที่สามารถควบคุมได้ ได้แก่
   - แนวแกนป้อน (Feed axes)
   - การขับป้อน (Feed drives)
   - อุปกรณ์วัดขนาด (Measuring drives)
   - เพลางาน (Work spindle)
   - อุปกรณ์จับยึดชิ้นงาน (Work piece holding devices)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.1 แนวแกนป้อน (Feed axes)

แนวแกน (axes) หมายถึง แนวการเคลื่อนที่ของเครื่องจักรกล เช่น โต๊ะงาน อุปกรณ์ลำเลียงเครื่องมือ (Tool carriers) เป็นต้น ซึ่งเครื่องจักรกลซีเอ็นซี มีแนวแกนป้อนรวมกันอยู่หลายแนวแกน ทำให้สามารถตัดเฉือนชิ้นงานให้เป็นรูปทรงต่างๆ ที่ต้องการได้ และจะกำหนดแนวแกนเหล่านี้โดยใช้ตัวอักษร X , Y และ Z ดังแสดงในรูป 1.1

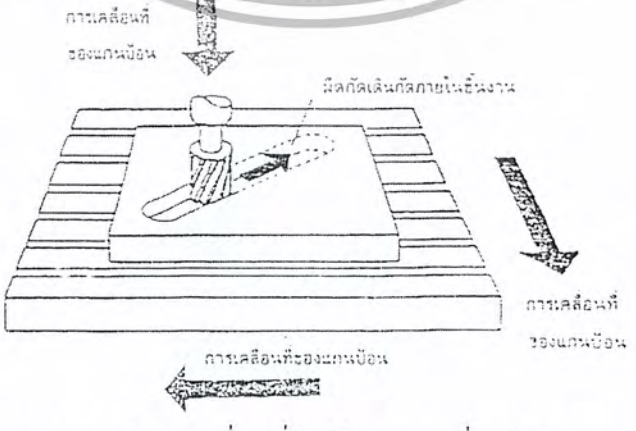


รูปที่ 1.1 เครื่องกลึงซีเอ็นซีแบบ 2 แกน

เครื่องกลึงจะมีแนวการป้อนอยู่ 2 แนวแกน คือ แกน X และ แกน Z ทั้งสองแกนจะอยู่ที่ชุดแท่นเลื่อน ซึ่งมีอุปกรณ์ลำเลียงเครื่องมือ (Tool Carrier) ติดตั้งอยู่ ลักษณะเช่นนี้ทำให้สามารถกลึงชิ้นงานที่มีรูปทรงต่างๆ กันได้ตามต้องการ

1.3 การขับป้อน (Feed drives)

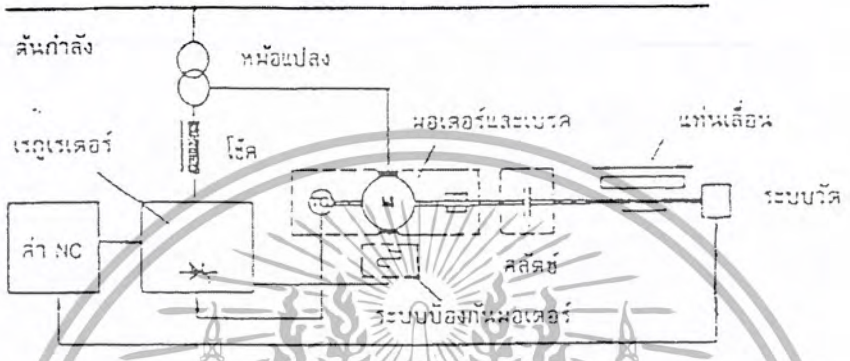
การเคลื่อนที่ที่เรียงลำดับกันหรือพร้อมๆกัน อย่างต่อเนื่องของแนวแกนป้อน จะทำให้เกิดการตัดเฉือนของเครื่องมือในชิ้นงาน ดังรูป 1.2



รูป 1.2 การเคลื่อนที่ตัดเฉือนของเครื่องมือตัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การขับป้อนจะทำให้เกิดการเคลื่อนที่ของแท่นเลื่อน ในขณะตัดเฉือนแท่นเลื่อนอาจพาให้ชิ้นงานเคลื่อนที่หรือคมตัดเคลื่อนที่ก็ได้ ระบบขับป้อนโดยทั่วไปจะใช้มอเตอร์กระแสตรงในการหมุนขับและควบคุมการทำงาน ด้วยวงจรอิเล็กทรอนิกส์จากภายนอก มอเตอร์ชนิดนี้จะสามารถหมุนและเบรกได้ทั้งสองทิศทางขณะตัดเฉือนชิ้นงาน การป้อนชิ้นงานจะต้องเคลื่อนที่ไปอย่างสม่ำเสมอและสามารถต้านแรงกระทำจากภายนอกได้



รูปที่ 1.3 Diagram ระบบขับป้อน

มอเตอร์

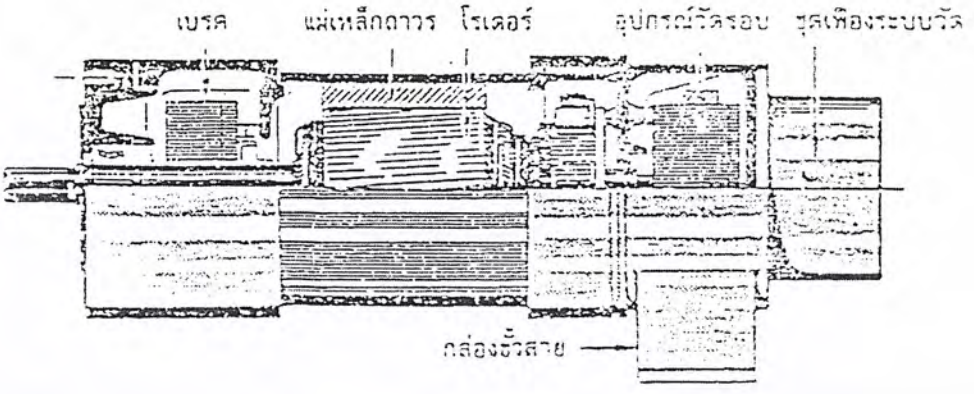
เครื่องจักรกลเอ็นซีสมัยใหม่นั้น จะออกแบบโดยการใช้ระบบขับป้อนแบบเซอโว (Servo drives) ทำให้สามารถปรับอัตราการป้อนและความเร็วรอบได้โดยไม่มีขีดจำกัดของขั้นความเร็วและอัตราการป้อนมอเตอร์ที่ใช้ในระบบขับป้อนโดยทั่วไปจะมีอยู่ 3 ชนิดคือ

- มอเตอร์กระแสตรง (DC motor)
- มอเตอร์แบบเป็นขั้น (Stepping motor)
- มอเตอร์กระแสสลับ (Alternate current motor)

ซึ่งในที่นี้จะกล่าวถึงแต่ มอเตอร์กระแสตรง (DC motor) เนื่องจากมอเตอร์ชนิดนี้ถูกใน EMCO Compact 5 CNC (เครื่อง CNC ที่ใช้ทำการศึกษา)

มอเตอร์กระแสตรง (DC motor) ลักษณะโครงสร้างของมอเตอร์กระแสตรงจะใช้เป็นแม่เหล็กถาวรที่มี 4,6 หรือ 8 ขั้ว ประกอบด้วยระบบเบรก (Brake) แกนมอเตอร์ (Rotor) อุปกรณ์วัดรอบ (Tachogenerator) และอุปกรณ์วัด (Measuring box) ดังแสดงในรูป 1.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

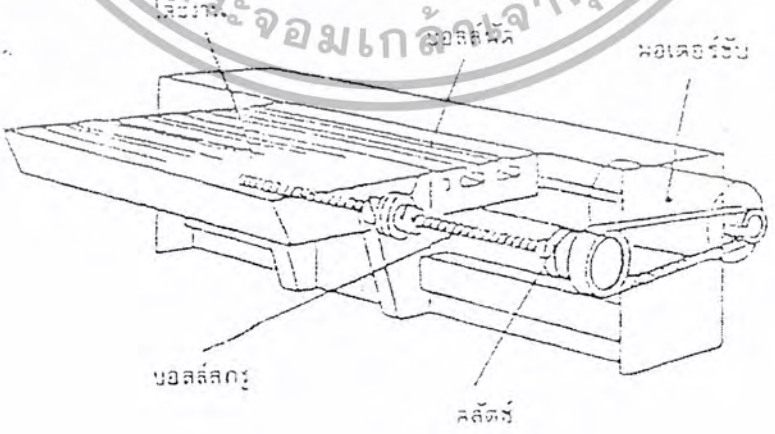


รูปที่ 1.4 ส่วนประกอบของมอเตอร์กระแสตรง

การใช้มอเตอร์กระแสตรง ทำให้สามารถปรับอัตราป้อนได้ละเอียดและมีวงจรควบคุมที่ไม่ซับซ้อน แต่มีข้อเสียที่แปรงถ่านของมอเตอร์ชนิดนี้ ซึ่งจะต้องคอยทำความสะอาดและเปลี่ยนเมื่อแปรงถ่านหมด นอกจากนี้ แปรงถ่านยังทำให้แกนมอเตอร์สึกหรออันเป็นผลทำให้กำลังมอเตอร์ลดลง ข้อเสียอีกประการหนึ่งก็คือ หากต้องการกำลังขับสูง มอเตอร์ก็จะมีขนาดที่ใหญ่ไปด้วย และเมื่อใช้ความเร็วรอบสูงๆ จะทำให้แรงบิดลดลง ดังนั้นจึงมักใช้กับเครื่องจักรกล CNC ขนาดเล็กหรือขนาดกลาง

1.4 บอลล์สกรู (Ball Screws)

หัวใจของระบบขับป้อนของเครื่องจักร ซีเอ็นซีก็คือ การส่งกำลังขับด้วยบอลล์สกรู ซึ่งจะมีลูกบอลล์ไหลหมุนเวียนตลอดเวลา บอลล์สกรูจะประกอบด้วยสกรูกับนัตที่มีลักษณะเป็นเกลียวกลม ร่องเกลียวกลมบนสกรูและในนัตจะชุบแข็งและเจียระไนผิวเรียบมัน เพื่อลดความฝืดและเพิ่มความเที่ยงตรงในการเคลื่อนที่ ดังแสดงในรูป 1.5



รูปที่ 1.5 การขับป้อนของโต๊ะงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายในของนัตจะประกอบไปด้วยชุดของลูกบอลล์จำนวนมาก ทำให้มั่นใจว่าความเสียดทานในการส่งขับจากสกรูไปยังแท่นเลื่อนจะมีน้อยมาก นัตจะถูกแบ่งออกเป็นสองซีกและขันประกบยึดเข้าด้วยกัน โดยมีการเตรียมอัดแรงไว้ก่อน ทำให้สามารถลดระยะคลอน (backlash) ให้เหลือน้อยที่สุดจนแทบไม่มี ทำให้การเคลื่อนที่ของแท่นเลื่อนมีความเที่ยงตรงสูงสามารถทำงานซ้ำๆกันได้

รูปที่ 1.6 ลักษณะโครงสร้างภายในของชุดบอลล์สกรู

การต่อระหว่างมอเตอร์กับบอลล์สกรู จะมีชุดคลัตช์ความฝืดเป็นตัวเชื่อม เพื่อป้องกันอุบัติเหตุที่เกิดจากแท่นเลื่อนหรือโต๊ะงานชน หรือกระแทกกับสิ่งกีดขวางไม่ให้เครื่องจักรกลซีเอ็นซีเกิดความเสียหาย กล่าวคือเมื่อมีการชนหรือกระแทกชุดคลัตช์ก็จะตัดระบบการส่งกำลังขับระหว่างมอเตอร์กับบอลล์สกรูทันที

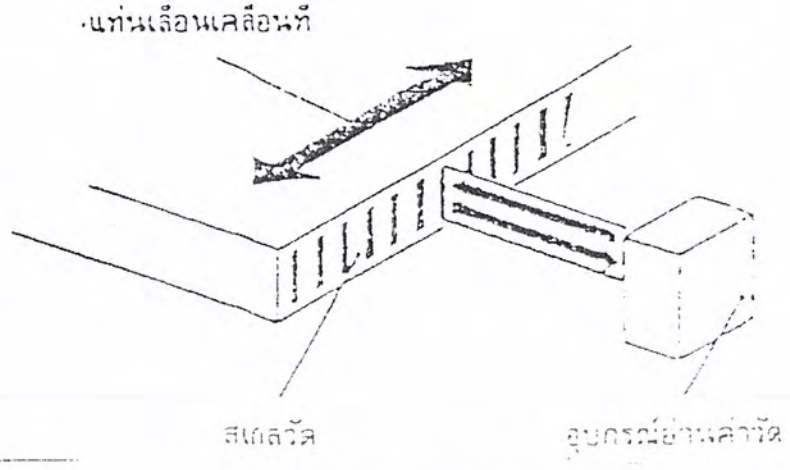
1.5 ระบบวัดขนาด (Measuring System)

การเคลื่อนที่ในตำแหน่งต่างๆของแท่นเลื่อน จะถูกส่งไปยังระบบควบคุมโดยระบบวัดขนาดซึ่งจะวัดได้ทั้งโดยตรง (Direct Measurement) และโดยทางอ้อม (Indirect Measurement)

1.5.1 การวัดตำแหน่งโดยตรง

วิธีนี้จะใช้สเกลวัด (measuring scale) ยึดติดกับแท่นเลื่อนหรือโต๊ะงาน ข้อดีของวิธีนี้ คือความไม่เที่ยงตรงของสกรูนำเลื่อน (Lead screw) ระบบขับจะไม่มีผลกับค่าที่อ่านได้และแปลงข้อมูลที่ได้เป็นสัญญาณไฟฟ้าส่งกลับไปยังระบบควบคุม ดังแสดงในรูป 1.7

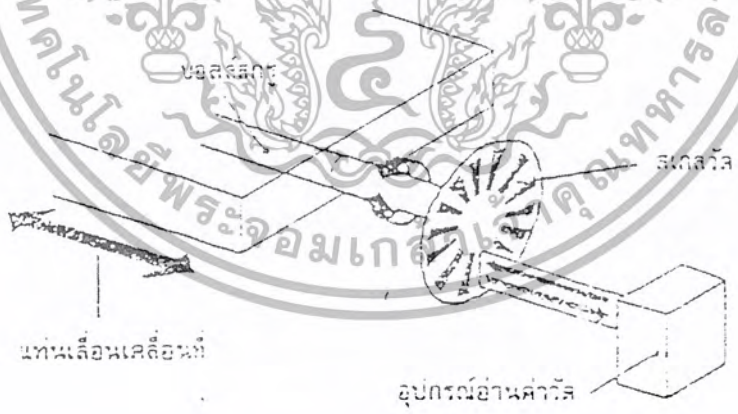
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.7 การวัดตำแหน่งโดยตรง

1.5.2 การวัดตำแหน่งทางอ้อม

การเคลื่อนที่ของแท่นเลื่อนจะได้รับกำลังขับมาจากการหมุนของบอลล์สกรู อุปกรณ์เปลี่ยนค่าวัด (Resolved) จะบันทึกการเคลื่อนที่หมุนของแผ่นจานสัญญาณ (Pulse disc) ที่ติดอยู่กับบอลล์สกรู และส่งต่อไปยังระบบควบคุมของเครื่อง ระบบควบคุมก็จะใช้สัญญาณที่ได้นี้ไปคำนวณระยะทางการเคลื่อนที่ของที่เลื่อนจากสัญญาณการหมุน (Rotation pulses) ของแผ่นจานสัญญาณ ดังแสดงใน รูปที่ 1.8



รูปที่ 1.8 การวัดตำแหน่งทางอ้อม

อุปกรณ์การวัดมีอยู่ด้วยกัน 2 ชนิดคือ สเกลวัดแบบรหัส (Coded measuring scale) กับสเกลแบบแบ่งช่อง (Division grid) การใช้สเกลวัดทั้งสองชนิดจะขึ้นอยู่กับวิธีการวัดตำแหน่ง (Position measurement) และวิธีการวัดตำแหน่งที่นิยมใช้กันทั่วไปมีอยู่ 2 วิธีคือ การวัดตำแหน่งแบบสัมบูรณ์

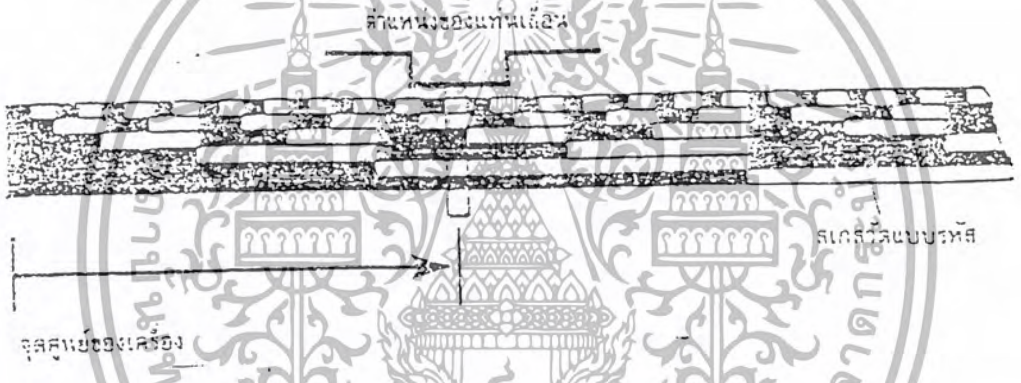
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Absolute position measurement) กับการวัดตำแหน่งแบบต่อเนื่องหรือแบบลูกโซ่ (Incremental or chain position measurement) ซึ่งมีความหมายแตกต่างกันดังรายละเอียดต่อไปนี้

1.5.3 การวัดตำแหน่งแบบสัมบูรณ์

คำว่า สัมบูรณ์ (Absolute) ที่ใช้ร่วมกับตำแหน่งนี้จะหมายความว่า ค่าตำแหน่งต่างๆ ที่สามารถวัดได้ตลอดเวลาและเป็นอิสระจากสถานะของเครื่องและระบบควบคุมทั้งนี้เพราะค่าต่างๆเหล่านี้จะวัดจากจุดอ้างอิง (Fixed zero datum) เสมอ

การวัดตำแหน่งแบบสัมบูรณ์ดังแสดงในรูป1.9 จะใช้สเกลแบบรหัส(Coded measuring scale) ซึ่งจะชี้ตำแหน่งของแท่นเลื่อนที่ถูกต้องตลอดเวลา โดยอ้างอิงจากจุดศูนย์ของเครื่อง (Machine zero point)



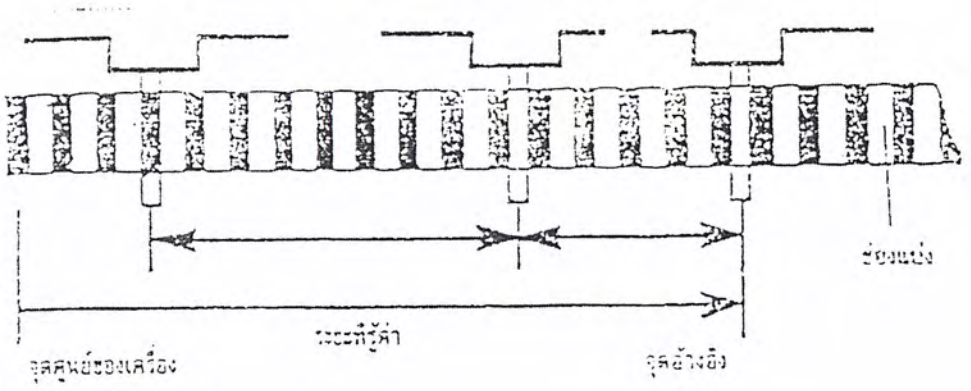
รูปที่ 1.9 การวัดตำแหน่งแบบสัมบูรณ์

ข้อสำคัญของการใช้วิธีนี้คือ ความยาวของช่วงอ่านค่าวัดของสเกลจะต้องยาวกว่าระยะการเคลื่อนที่ของแท่นเลื่อน เพื่อให้ระบบควบคุมของเครื่องสามารถอ่านค่าวัดได้ทุกตำแหน่งสเกลนี้จะใช้รหัสเป็นระบบตัวเลขฐานสอง (Binary system)

1.5.4 การวัดแบบต่อเนื่อง

การวัดตำแหน่งแบบต่อเนื่องที่แสดงในรูป สเกลวัดจะแบ่งเป็นช่อง (grid) แบบง่าย โดยที่แต่ละช่องจะมีพื้นที่สว่างกับมืดสลับกันไป เมื่อแท่นเลื่อนเคลื่อนที่ ช่องนี้ก็จะวิ่งผ่านอุปกรณ์เปลี่ยนค่าวัด (resolved) ซึ่งจะทำหน้าที่นับจำนวนช่องสว่างกับมืด จากนั้นก็จะส่งเป็นสัญญาณไฟฟ้าไปยังระบบควบคุมของเครื่องและระบบก็จะนำสัญญาณไปคำนวณหาตำแหน่งของแท่นเลื่อน ในทางปฏิบัติที่ต้องการให้วิธีนี้ทำงานได้อย่างถูกต้อง เมื่อเริ่มเปิดสวิตช์ระบบควบคุมของเครื่องควรเลื่อนแท่นเลื่อนไปยังจุดที่ทราบค่าระยะห่างจากจุดศูนย์ของเครื่อง จุดนี้เรียกว่า “ จุดอ้างอิง ” (Reference point)

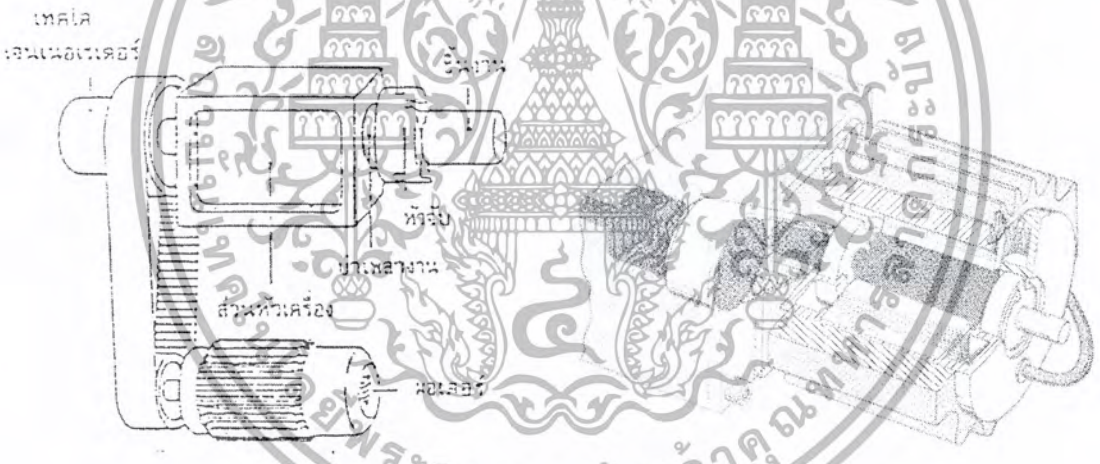
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.10 การวัดตำแหน่งแบต่อเนื่อง

1.6 เพลางาน (Work Spindle)

เพลางานเป็นชิ้นส่วนหรือองค์ประกอบของเครื่องจักรที่มีความสำคัญมาก มีหน้าที่หลักในการทำงานคือ ทำหน้าที่ขับพาให้กับเครื่องมือ เช่น มีดกลึง ดอกสว่าน เป็นต้น



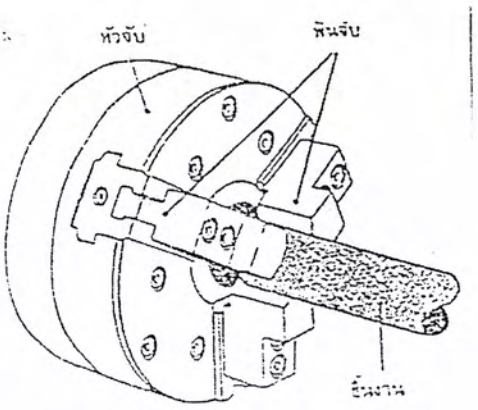
รูปที่ 1.11 เพลางานของเครื่องกลึง

อุปกรณ์จับยึดชิ้นงาน (Work piece holding devices)

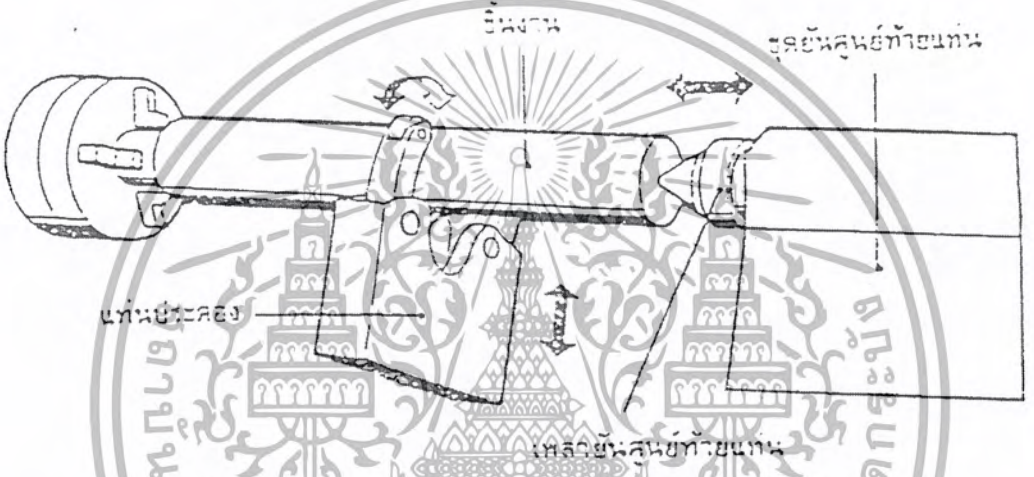
อุปกรณ์จับยึดชิ้นงาน มีหน้าที่ยึดชิ้นงานเข้ากับเพลา ในงานกลึง (latch) สามารถเลือกใช้อุปกรณ์จับยึดชิ้นงานแบบต่างๆ กันได้หลายชนิดเช่น

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.12 หัวจับงานกลึง

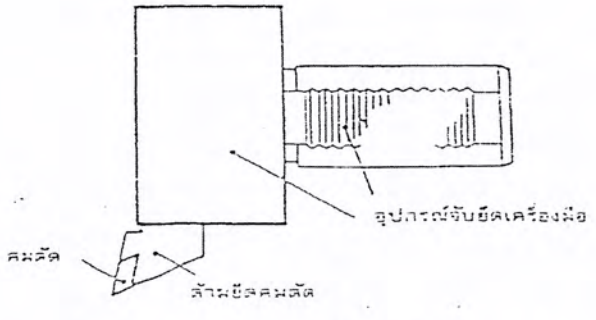


รูปที่ 1.13 ชุดยันศูนย์ท้ายแท่นและแท่นประคองงานกลึง

1.7 เครื่องมือ (Tool)

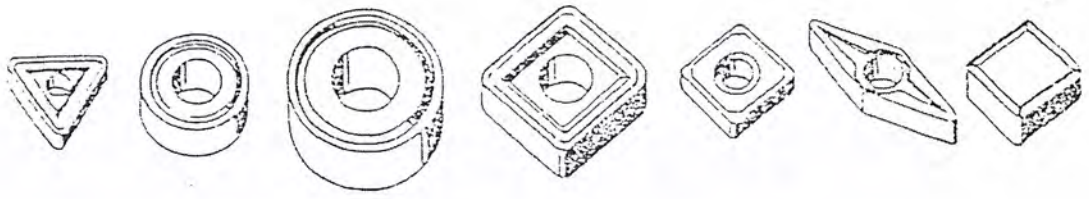
องค์ประกอบของเครื่องมือที่สมบูรณ์สำหรับเครื่องซีเอ็นซี ซึ่งจะประกอบด้วยส่วนต่างๆ ดังนี้

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



รูปที่ 1.14 องค์ประกอบของเครื่องมือที่ใช้ในงานกลึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.15 อินเสิร์ตแบบถอดเปลี่ยนได้รูปทรงต่างๆ

1.8 ชนิดของการควบคุม (Control modes)

ลักษณะการควบคุม การทำงานของแท่นเลื่อนต่างๆ ในเครื่องจักรกล เอ็นซีและซีเอ็นซี จะมีการเคลื่อนที่อยู่ 2 ลักษณะคือ

การเคลื่อนที่ในแนวเส้นตรง (Liner interpolation หรือ straight line interpolation)

การเคลื่อนที่ลักษณะนี้ ระบบซีเอ็นซีจะคำนวณหาตำแหน่งต่างๆ ที่ต่อกันเป็นลูกโดยในแนวเส้นตรงระหว่างตำแหน่งของเครื่องมือ 2 ตำแหน่ง ในขณะที่เครื่องมือเคลื่อนที่จากจุดหนึ่งไปยังอีกจุดหนึ่งนั้น ระบบซีเอ็นซีจะตรวจสอบและแก้ไขแนวแกนในการเคลื่อนที่ให้ถูกต้อง อยู่ตลอดเวลาทำให้การเคลื่อนที่ของเครื่องมือ ไม่ผิดผลาดหรือคลาดเคลื่อนออกจากจุดต่อของเส้นตรงมากกว่าค่าพิกัดความเผื่อของเครื่องที่กำหนดไว้

การเคลื่อนที่ในแนวเส้นโค้ง (Circular Interpolation)

ระบบควบคุมซีเอ็นซีจะคำนวณหา ตำแหน่งของจุดต่างๆ ที่ต่อกันเป็นเส้นโค้งตามขนาดรัศมีที่กำหนดระหว่างตำแหน่งของเครื่องมือที่กำหนดไว้ 2 ตำแหน่ง ระบบควบคุมจะอาศัยจุดเหล่านี้ในการตรวจสอบและแก้ไขแนวการเคลื่อนที่ ของเครื่องมือให้ถูกต้องและอยู่ภายใน พิกัดความเผื่อของเครื่องจักรกลที่กำหนดไว้

ในระบบควบคุมซีเอ็นซีนั้น จะแบ่งการควบคุมการเคลื่อนที่ทั้งสองลักษณะ ตามลักษณะการเคลื่อนที่ป้อนออกเป็น 3 ชนิดคือ

การควบคุมจุดต่อจุด (point to point control)

การควบคุมแบบนี้จะมีการควบคุมการเคลื่อนที่ระหว่างจุดสองจุดที่โปรแกรมลักษณะการเคลื่อนที่เร็ว (Rapid traverse) โดยเครื่องมือจะต้องไม่สัมผัสชิ้นงาน แนวการเคลื่อนที่ขึ้นอยู่กับชนิดของระบบควบคุม กล่าวคือ มอเตอร์ขับของระบบป้อนอาจจะเริ่มทำงานหลายๆแนวแกนพร้อมกัน หรือทำงานทีละแนวแกน จนกว่าจะเคลื่อนที่ถึงตำแหน่งของเครื่องมือที่กำหนดไว้ ทำให้ไม่สามารถควบคุมาทางเดินของเครื่องมือ (Tool path) ได้ การควบคุมจุดต่อจุดมักจะใช้กับเครื่องเจาะ (spot drilling) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมการตัดเฉือนแนวเส้นตรง (Straight-cut control)

การควบคุมชนิดนี้ นอกจากจะสามารถควบคุมการเคลื่อนที่ของเครื่องมือแบบเคลื่อนที่เร็วได้แล้ว ยังสามารถควบคุมการเคลื่อนที่ของเครื่องมือในแนวขนานกับแนวแกนของเครื่องจักรกลตามค่าอัตราป้อนที่ต้องการได้อีกด้วย แต่จะสามารถควบคุมแนวการเคลื่อนที่ได้ทีละหนึ่งแกนเท่านั้น การเคลื่อนที่ของเครื่องมือจะถูกควบคุมด้วยอัตราป้อนและความยาวในการเคลื่อนที่ ระบบควบคุมการตัดเฉือนแนวเส้นตรงชนิดนี้จะใช้เครื่องมือตัดและเครื่องมือกลึงแบบง่ายๆ

การควบคุมตามเส้นขอบรูป (Contouring control)

การควบคุมแบบนี้จะสามารถควบคุมการเคลื่อนที่ทำงานได้ดังนี้

- ควบคุมเครื่องมือให้เคลื่อนที่ไปยังตำแหน่งที่ต้องการแบบเคลื่อนที่เร็วได้
- ควบคุมเครื่องมือให้เคลื่อนที่ขนานกับแนวแกน ไปยังตำแหน่งที่ต้องการ ตามค่าอัตราป้อนได้
- ควบคุมเครื่องมือให้เคลื่อนที่ไปยังตำแหน่งใดๆ บนชิ้นงานที่กำหนดโดยแนวเส้นตรงและเส้นโค้งตามอัตราป้อนได้

1.9 ลักษณะเฉพาะของเครื่อง CNC – Compact 5

เครื่อง CNC-Compact 5 เครื่องนี้เป็นเครื่อง CNC ที่ใช้สำหรับงานกลึงเท่านั้น โดยจะขอกล่าวถึงส่วนต่าง ๆ ของเครื่องดังนี้

รูปที่ 1.16 แสดงส่วนรายละเอียดของเครื่อง CNC-Compact 5

1. สวิตซ์หลัก

เมื่อทำการบิดกุญแจไปทางด้านขวา จะเป็นการ Start การทำงานของเครื่อง CNC

2. Control Lamp – Main switch

เมื่อสวิตซ์หลักเปิดอยู่ Control Lamp ก็จะสว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สวิตซ์สำหรับขับสปินเดิลหลัก

4. ปุ่มสำหรับควบคุมความเร็วของสปินเดิลหลัก

5. ส่วนแสดงความเร็วของสปินเดิล

6. ปุ่มสำหรับปรับอัตราป้อนกลึงมีด

ปรับได้ทั้งทิศทางแกน Z (saddle) และทิศทางแกน X (Cross slide) โดยสามารถปรับค่าได้ตั้งแต่ 10 – 400 mm/min

7. Control Lamp – Hand Operated

เป็นไฟที่แสดงการทำงานในโหมด Hand – Operated

8. คีย์สำหรับการป้อนมีด

สามารถทำได้ทั้งในทิศทาง +X, -X และ +Z, -Z ซึ่งจะมีสัญลักษณ์แสดงอยู่ที่คีย์บอร์ด

9. คีย์สำหรับการเครื่องที่เร็ว

ถ้าเรากดคีย์สำหรับการเคลื่อนที่เร็วพร้อมกับคีย์สำหรับการป้อนมีด จะทำให้มีดกลึงเคลื่อนที่ด้วย ความเร็วขึ้น

10. ส่วนแสดงทางเดินของมีด

ส่วนแสดงผลนี้จะแสดงระยะ X และระยะ Z โดยแสดงเป็นค่าเท่ากับ 100 ต่อ 1 มม. และเครื่องหมายลบจะแสดงให้เห็นเป็นจุดบน Display

11. ปุ่มเปลี่ยนโหมดการทำงาน

ถ้ากดปุ่ม Hand / CNC ไฟจะเปลี่ยนจาก Control Lamp Hand- Operation ไปเป็น Control Lamp CNC-Operation และถ้ากดปุ่มนี้อีกครั้งหนึ่งไฟ Control Lamp ก็จะกลับมาสว่างในตำแหน่งเดิม

12. แอมมิเตอร์แสดงกระแสที่ใช้ขับมอเตอร์ของสปินเดิลหลัก

แอมมิเตอร์จะแสดงให้เห็นถึงการกินกระแสขณะขับมอเตอร์ เพื่อป้องกันมอเตอร์ไม่ให้เกิด การโอเวอร์โหลด กระแสที่จ่ายให้ต้องมีค่าไม่เกิน 2 Amp ขณะทำงานต่อเนื่อง นอกจากนี้เรายังสามารถลดโหลดได้โดยการลดค่าความลึกในการกินชิ้นงานของมีด สำหรับระบบการป้องกันการโอเวอร์โหลดของมอเตอร์ในกรณีที่กินกระแสสูงสุด มอเตอร์จะหยุดการทำงานเมื่อกระแสมีค่า 4 Amp

13. ปุ่มหยุดการทำงานฉุกเฉิน

เมื่อกดปุ่มนี้ จะเกิดการตัดกระแสจากมอเตอร์หลัก (Main Motor), มอเตอร์ที่ใช้ป้อนมีด (Feed Motor) , และส่วนการควบคุมทันที

14. ปุ่ม DEL

ทำหน้าที่ในการ Clear ค่า X และ Z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

15. ปุ่ม Input

จะใช้เมื่อต้องการป้อนข้อมูล

16. Belt pulley drive ดังแสดงในรูปที่ 1.17

รูปที่ 1.17 Belt pulley drive

จากการทดลองการใช้ CNC (Compact 5) สามารถสรุปการทำงานของรหัส G – Code และ M – Code ได้ดังนี้

Code	Function
G 00	Rapid traverse
G 01	Linear interpolation
G 02	Circular interpolation clockwise
G 03	Circular interpolation counter clockwise
G 21	Empty line
G 25	Sub – routine call – up
G66	RS 232 Operation
G 73	Chip breakage cycle
G 81	Drilling cycle
G 82	Drilling cycle with dwell
G 83	Drilling cycle deep hole with withdrawal
G 84	Longitudinal turning

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

G 85	Reaming cycle
G 86	Grooving with division of cut (parameter H)
G 88	Facing with division of cut (parameter H)
G 89	Reaming and drilling with dwell
G 90	Absolute value programming
G 91	Incremental value programming
G 94	Feed in mm/min
G 95	Feed in mm/rev
M 00	Programmed stop
M 03	Main spindle ON
M 05	Main spindle OFF
M 08	Switch exit X62 PIN 15 HIGH
M 09	Switch exit X62 PIN 15 LOW
M 17	Return command to the main program
M 22	Switch exit X62 PIN 18 HIGH
M 23	Switch exit X62 PIN 18 LOW
M 26	Switch exit X62 PIN 20
M 30	End of program
M 99	Circle parameter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1.2 แสดงค่าขนาดของอินพุตสูงสุดที่สามารถป้อนให้แก่ CNC-Compact 5

Address	Metric		Inch	
	Value	Dimension	Value	Dimension
N Block number	00-209	1	00-209	1
G Traverse function (G-Codes)	00-95	1	00-95	1
M Miscellaneous function (M-Codes)	00-99	1	00-99	1
X Coordinate CNC-input	0-±5999	$\frac{1}{100}$ mm	0-±1999	$\frac{1}{1000}$ "
Z Coordinate CNC-input	0-±32760		0-±12900	
X Coordinate hand input	0-±89999		0-±29999	
Z Coordinate hand input	0-±89999		0-±29999	
F Feed	2-499		2-199	
With G94		mm/min		$\frac{1}{10}$ "/min
With G95		$\frac{1}{1000}$ mm/U		$\frac{1}{10000}$ "/U
I Circle point coordinate in X	0-5999	$\frac{1}{100}$ mm	0-1999	$\frac{1}{1000}$ "
K Circle point coordinate in Z	0-[illegible]			
X Dwell (time)	0-±5999	$\frac{1}{100}$ sec	0-±1999	$\frac{1}{100}$ sec
L Jump address	0-221	1	0-221	1
T Tool address	0-499		0-199	
H Parameter division of cut	0-999	$\frac{1}{100}$ mm	0-999	$\frac{1}{1000}$ "
H Parameter width of turning tool (G86)	10-999		10-999	
H Parameter impulse edit	0-999		0-999	
K Thread pitch	2-499		2-199	

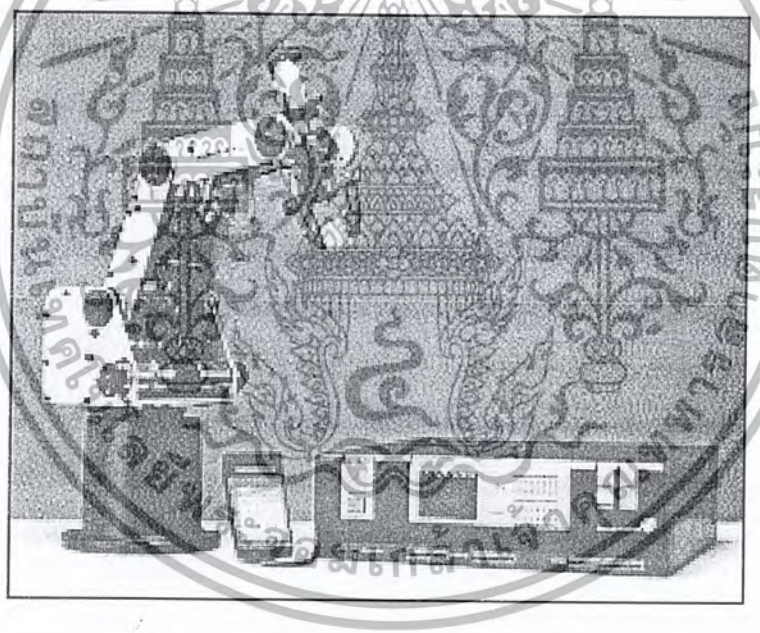
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

บทที่ 2
หุ่นยนต์ (Robotics)

2.1 การเคลื่อนย้ายโดยหุ่นยนต์ (Robotics Manipulation)

หุ่นยนต์ (Robot) ในที่นี้จะหมายถึงหุ่นยนต์ทางอุตสาหกรรม (Industrial robot) ซึ่งถูกเรียกว่า แขนกล (Robotic Manipulator or Robotic Arm) ตัวอย่างของหุ่นยนต์ทางอุตสาหกรรม แสดงดังรูปที่ 2.1 ซึ่งเป็นแขนกลข้อต่อหมุน และมีลักษณะคร่าว ๆ คลายคลึงกับแขนมนุษย์ แขนกลเปรียบได้กับ Rigid Rink หลาย ๆอัน ที่นำมาเชื่อมต่อด้วยข้อต่อยืดหยุ่นได้ (Flexible Joints) Link เหล่านี้ทำหน้าที่คล้ายกับอวัยวะบางส่วนของร่างกายมนุษย์ ที่ปลายแขนกลเป็นส่วนทำงานของ(End–Effector) เรียกว่า Tool , Gripper หรือ Hand ปกติ Tool จะเป็นมือจับที่มี 2 นิ้ว หรือมากกว่าใช้ในการเปิด – ปิด



รูปที่ 2.1 แขนกลข้อต่อหมุน (Scorbot-ER 5 plus)

2.2 เครื่องจักรกลแบบ Hard and Soft Automation

สายการผลิตแบบ Mass – Production ถูกนำมาใช้ครั้งแรกในต้นศตวรรษที่ 20(1905) โดยบริษัทฟอร์ดมอเตอร์ ในช่วงหลายสิบปีแรกนั้นเครื่องจักรแบบพิเศษได้ถูกออกแบบ และพัฒนาเพื่อใช้ในการผลิตชิ้นส่วนเครื่องจักรกลและไฟฟ้าให้ได้ปริมาณมาก ๆ แต่เมื่อจบรอบการผลิตในแต่ละปี ชิ้นส่วนแบบใหม่ก็จะถูกนำมาใช้อีก ทำให้ต้องมีการปรับเปลี่ยนเครื่องจักรเพื่อใช้ในการผลิตชิ้นส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50205

ใหม่เหล่านั้น เนื่องจากการเปลี่ยนแปลงทางด้าน Hardware จึงเรียกเครื่องจักรแบบนี้ว่า Hard Automation เครื่องจักรและกระ บวนการผลิตแบบนี้มีประสิทธิภาพมาก แต่จะมีความยืดหยุ่นจำกัด

ในยุคต่อมา อุตสาหกรรมรถยนต์และอุตสาหกรรมอื่น ๆ ได้เริ่มนำเครื่องจักรที่มีความยืดหยุ่นใช้ในวัฏจักรการผลิต แขนกลที่โปรแกรมได้ (Programmable Mechanical Manipulators) ถูกนำมาใช้ในงาน เช่น การเชื่อมจุด การพ้นสี การลำเลียงวัสดุ และการประกอบชิ้นส่วน เนื่องจากแขนกลที่ควบคุมด้วยคอมพิวเตอร์นี้สามารถที่จะปรับเปลี่ยนผ่านทาง Software เพื่อให้ทำงานได้หลากหลาย จึงถูกเรียกว่า Soft Automation

2.3 การแบ่งประเภทหุ่นยนต์ (Robot Classification)

มีเกณฑ์ที่ใช้แบ่งหลายอย่างดังนี้

2.3.1 เทคโนโลยีการขับเคลื่อน (Driver Technologies)

พิจารณาจากต้นกำลังที่ใช้ขับข้อต่อของหุ่นยนต์ นิยมใช้มี 2 แบบ คือการเคลื่อนด้วยไฟฟ้ากับไฮโดรลิค แขนกลส่วนใหญ่ในปัจจุบันจะใช้การขับเคลื่อนด้วยไฟฟ้า ในรูปของ DC เซอร์โวมอเตอร์ หรือ DC สเตปปิ้งมอเตอร์ ในกรณีของการเคลื่อนย้ายวัตถุที่ต้องการความเร็วสูง เช่น เหล็กหลอมละลาย หรือ ชิ้นส่วนรถยนต์ มักนิยมใช้หุ่นยนต์ขับเคลื่อนด้วยไฮโดรลิค แต่ก็มีข้อเสียที่สำคัญคือขาดความสะอาด ซึ่งเป็นลักษณะสำคัญในงานประกอบชิ้นส่วนหลายชนิด

2.3.2 รูปทรงขอบเขตการทำงาน (Work Geometries)

Tool ของแขนกลปกติจะติดตั้งอยู่ที่ข้อมือของแขนกล ดังนั้นขอบเขตของการทำงานสุทธิ (Gross Work Envelope) จึงหมายถึง ขอบเขตในปริภูมิสามมิติ ที่ข้อมือของแขนกลสามารถเคลื่อนที่ไปถึงตำแหน่งนั้น ๆได้ เราจะเรียกแกนของ 3 ข้อต่อแรกนี้ว่าเป็น แกนหลัก(Major Axis) ซึ่งใช้ในการพิจารณาตำแหน่งของข้อมือส่วนแกนข้อต่อที่เหลือ เรียกว่า แกนรอง (Minor Axis) ใช้กำหนดลักษณะการวางตัวหรือหมุนของ Tool ข้อต่อมีด้วยกันหลายแบบ ที่เป็นพื้นฐานและนิยมใช้ในโรงงานอุตสาหกรรมมี 2 แบบ ดังตาราง 2.1

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

ตารางที่ 2.1 ชนิดของข้อต่อหุ่นยนต์

ข้อต่อ Revolute(R) ทำให้เกิดการหมุนรอบแกน ส่วนข้อต่อ Prismatic (P) ทำให้เกิดการเคลื่อนที่ไปตามแนวแกน การรวมกันของข้อต่อ 2 ชนิดนี้ใน 3 แกนหลัก ก่อให้เกิดรูปทรงขอบเขต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

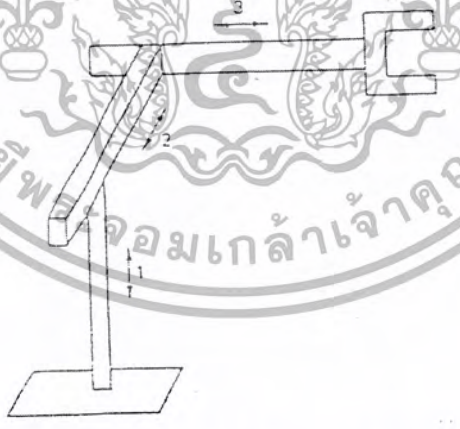
การทำงานหลายแบบ แสดงในตัวอย่างตาราง 2.2 ซึ่งแสดงเฉพาะแบบที่นิยมใช้ในการวิเคราะห์การเคลื่อนที่ของแขนกล

หุ่นยนต์	แกน 1	แกน 2	แกน 3	แกน 4
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 (Prismatic) = ข้อต่อเลื่อน , R (Revolute) = ข้อต่อหมุน

ตาราง 2.2 แสดงขอบเขตการทำงานของหุ่นยนต์ที่ขึ้นอยู่กับแกนหลัก

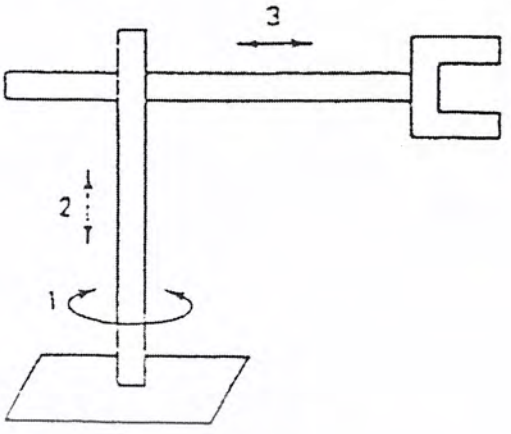
หุ่นยนต์ที่ธรรมดาที่สุดในตารางที่ 2.2 มีแกนหลักเป็นข้อต่อเลื่อน (Prismatic) ทั้งหมด สัญลักษณ์ PPP เรียกว่าพิกัด คาร์ทีเซียน (Cartesian – Coordinate Robot) หรือ หุ่นยนต์พิกัดฉาก (Rectangular – Coordinate Robot) แสดงดังรูปที่ 2.1 จะเห็นได้ว่า ข้อมือจะเลื่อนขึ้น - ลง , เข้า - ออก และเดินหน้าหลัง ทำให้ Work – Envelope มีลักษณะเป็นกล่องสี่เหลี่ยมมุมฉาก



รูปที่ 2.1 หุ่นยนต์พิกัดคาร์ทีเซียน

หุ่นยนต์พิกัดทรงกระบอก (Cylindrical – Coordinate Robot) สัญลักษณ์ RPP แสดงดังรูปที่ 2.2 ข้อต่อ Revolute ทำให้แขนหมุนรอบแกนตั้งฉากกับฐาน ข้อต่อ Prismatic เลื่อนข้อมือขึ้น - ลง ตามแนวตั้ง และเลื่อนเข้า - ออกตามแนวรัศมี ทำให้ Work – Envelope มีลักษณะเป็นทรงกระบอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



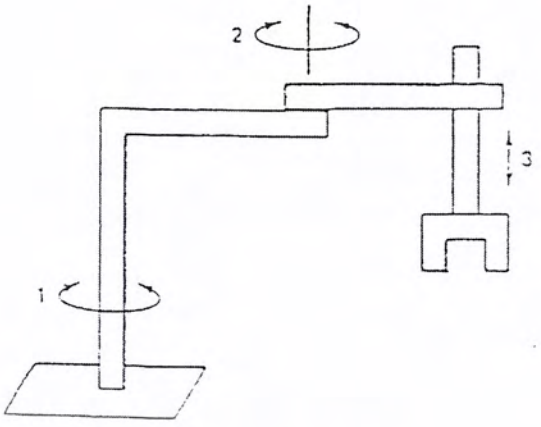
รูปที่ 2.2 หุ่นยนต์พิกัดทรงกระบอก

หุ่นยนต์พิกัดทรงกลม (Spherical – Coordinate Robot) สัญลักษณ์คือ RRP แสดงดังรูปที่ 2.3 ข้อต่อ Revolute ข้อแรกทำให้แขนหมุนรอบแกนแนวตั้งของฐาน ส่วนข้อที่สองทำให้หมุนขึ้น - ลงรอบแกนแนวนอนของไหล่ ส่วนข้อต่อ Prismatic ทำให้ข้อต่อเลื่อนเข้า - ออกตามแนวรัศมี Work Envelope จึงมีลักษณะเป็นทรงกลม

รูปที่ 2.3 หุ่นยนต์พิกัดทรงกลม

หุ่นยนต์ SCARA (Selective Compliance Assembly Robot Arm) มีลักษณะคล้ายกับหุ่นยนต์พิกัดทรงกลมที่มีข้อต่อ Revolute 2 ข้อ และข้อต่อPrismatic 1 ข้อ สัญลักษณ์ RRP แต่แกนทั้งสามของ SCARA จะอยู่ในแนวตั้งดังรูปที่ 2.4 Revolute ข้อที่สองทำให้ปลายแขนหมุนรอบแกนตั้งของข้อศอก ซึ่งทั้ง 2 ข้อนี้ จะควบคุมการเคลื่อนที่ในแนวระนาบแนวนอน ส่วนการเลื่อนที่ในแนวตั้งได้จากข้อต่อPrismaticที่เลื่อนข้อมือขึ้นและลง ภาพตัดขวางในแนวนอนของ WorkEnvelope ค่อนข้างซับซ้อนขึ้นอยู่กับลิมิตในการเคลื่อนที่ของ 2 แกนแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 หุ่นยนต์ SCARA

หุ่นยนต์ข้อต่อหมุน (Articulate – Coordinate Robot or Revolute Robot) ซึ่งมีสัญลักษณ์ RRR โดย 3 ข้อต่อหลักจะเป็นแบบ Revolute แสดงดังรูปที่ 2.5 หุ่นยนต์แบบนี้มีลักษณะคล้ายแขนมนุษย์มากที่สุดโดยมีข้อต่อต่อแรกทำให้แขนหมุนรอบแกนตั้งของฐาน ส่วนข้อต่อที่สองทำให้แขนหมุนขึ้นลงรอบแกนแนวนอนของไหล่ และข้อต่อที่สามนั้นทำให้ปลายแขนหมุนขึ้นลง

รูปที่ 2.5 หุ่นข้อต่อหมุน

2.4 ลักษณะเฉพาะของหุ่นยนต์ (Robot Specifications)

นอกจากเกณฑ์ต่าง ๆ ที่ใช้ในการแบ่งประเภทของหุ่นยนต์ดังที่กล่าวมา ยังลักษณะบางอย่างอีก ที่ช่วยให้ผู้ในการระบุเลือกหุ่นยนต์ที่ต้องการ ลักษณะเหล่านี้แสดงเช่น ตารางที่ 2.3 ซึ่งแสดงลักษณะเฉพาะของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะ	หน่วย
จำนวนแกน (Number of axis)	5 axes
ความสามารถในการยกน้ำหนัก (Payload capacity)	1 kg.
รัศมีของการทำงาน (Maximum operating radius)	610 mm
ความเร็วสูงสุด (Maximum path velocity)	660 mm/sec
การทำงานซ้ำตำแหน่งเดิม (Repeatability)	± 0.15 mm
สภาพแวดล้อมในการทำงาน (Ambient operating)	2°- 40°c

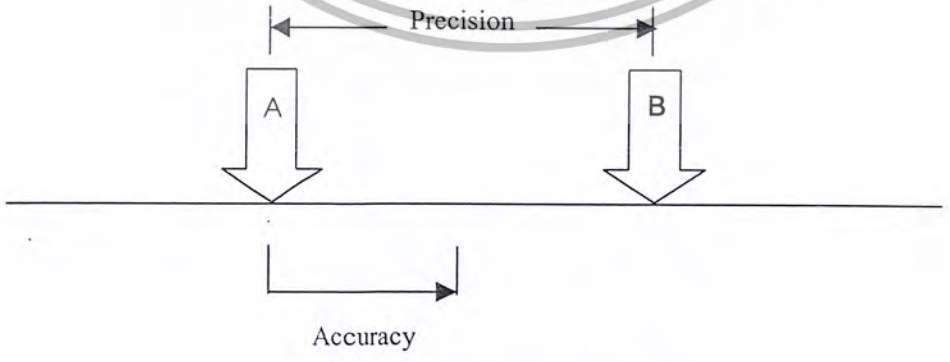
ตาราง 2.3 แสดงลักษณะเฉพาะของหุ่นยนต์ (Scorbot – ER 5 plus)

2.5 ความสามารถในการซ้ำตำแหน่งเดิม , ความละเอียดและความแม่นยำ (Repeatability Precision and Accuracy)

Repeatability เป็นลักษณะที่สำคัญอีกประการหนึ่ง กล่าวคือเป็นการวัดความสามารถของหุ่นยนต์ ในการวางตำแหน่งของปลายอุปกรณ์ในตำแหน่งเดียวกันซ้ำหลาย ๆ ครั้ง เนื่องจากสาเหตุต่าง ๆ เช่น Backlash ในเฟืองและความยืดหยุ่นของส่วนต่าง ๆ ซึ่งมักทำให้เกิดความผิดพลาด

Precision คือ ความละเอียดในการวัดระยะห่างระหว่างตำแหน่งต่าง ๆ ในขอบเขตการทำงานที่สามารถจะกำหนดให้เครื่องมือเคลื่อนที่ไปได้ เช่น ถ้าปลายเครื่องมืออยู่ที่ตำแหน่ง A ในรูป 2.6 ตำแหน่งต่อไปที่ใกล้ที่สุดที่สามารถเคลื่อนที่ไปได้ คือ B ดังนั้นความละเอียดคือ ระยะห่างระหว่างจุด A และ B

Accuracy คือ ความถูกต้องเป็นลักษณะที่สำคัญมากในการทำงาน กล่าวคือ เป็นการวางตำแหน่งของปลายอุปกรณ์ในตำแหน่งที่เราต้องการได้อย่างถูกต้องแม่นยำ



รูปที่ 2.6 ตำแน่งของ Tool

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

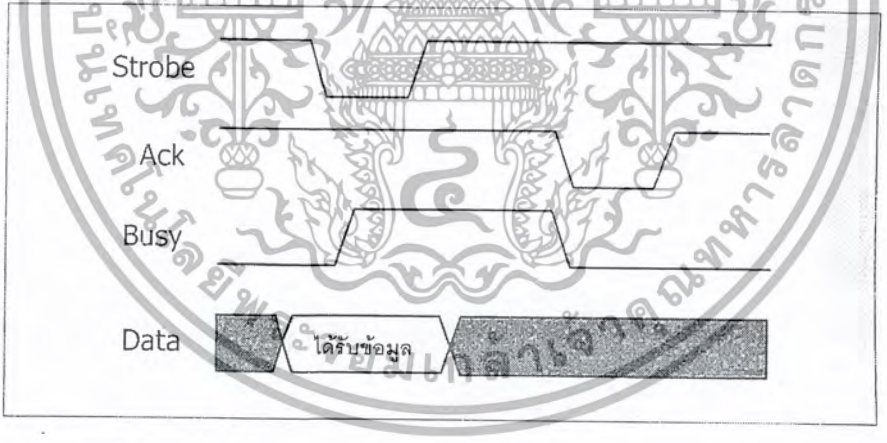
บทที่ 3
การเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอก
(Computer Interfacing)

การเขียนโปรแกรมติดต่อกับเครื่อง CNC – Compact 5 ผู้เขียนจะใช้โปรแกรม Visual Basic 6 ในการเขียนโปรแกรมติดต่อระหว่างคอมพิวเตอร์กับ CNC – Compact 5 ซึ่งมีการติดต่อด้วยกันทั้ง 2 แบบ คือ

1. การติดต่อผ่านพอร์ตขนาน (Parallel Interfacing) ใช้ในการควบคุมการทำงานต่างๆ
2. การติดต่อผ่านพอร์ตอนุกรม (Serial Interfacing) ใช้ในการส่งรหัส G-Code

3.1 ลักษณะทางกายภาพของพอร์ตขนาน

เพื่อให้เข้าใจถึงการนำพอร์ตขนานไปใช้งาน ก่อนอื่นต้องมาทำความเข้าใจก่อนว่า ปกตินั้นการสั่งพิมพ์งานจากคอมพิวเตอร์ไปยังพอร์ตขนานนั้นมีรูปแบบการทำงานอย่างไร ในรูปที่ 3.1 แสดงไดอะแกรมเวลาของการติดต่อระหว่างพอร์ตขนานกับเครื่องพิมพ์ ซึ่งจะเห็นได้ว่ามีสัญญาณที่ใช้งานจริง ๆ มีไม่มาก เริ่มจากสัญญาณพอร์ต Data ถูกส่งออกไปยังเครื่องพิมพ์ พร้อมทั้งส่งสัญญาณ Storbe



รูปที่ 3.1 แสดงไดอะแกรมเวลาของการส่งข้อมูลไปยังเครื่องพิมพ์

ออกไปด้วย เพื่อให้เครื่องพิมพ์รับรู้ว่ามีการส่งข้อมูลใหม่มาที่ขา Data แล้ว จากนั้นคอมพิวเตอร์จะต้องรอการตอบกลับของเครื่องพิมพ์ นั่นคือเครื่องพิพ์จะสร้างสัญญาณ Busy หรือ เพื่อบอกว่าเครื่องพิมพ์ยังไม่พร้อมที่จะรับข้อมูลใหม่ จนกระทั่งเครื่องพิมพ์พร้อม เครื่องพิมพ์จะสร้างสัญญาณ ACK ส่งไปยังคอมพิวเตอร์เพื่อแจ้งว่า พร้อมที่จะรับข้อมูลใหม่แล้ว

สัญญาณข้อมูลขนาด 8 บิต, สัญญาณ Strobe และสัญญาณ ACK (acknowledge) เป็นสัญญาณที่สำคัญในการส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์ นอกจากสัญญาณทั้งสามแล้วส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใหญ่การติดต่อกับเครื่องพิมพ์ยังต้องมีสัญญาณอื่น ๆ ร่วมด้วย เนื่องจากคอมพิวเตอร์ยังต้องทำหน้าที่ถึง 3 อย่างด้วยกันคือ รับข้อมูลจากคอมพิวเตอร์, พิมพ์ข้อมูลที่รับเข้ามา และตอบสนองการใช้งานของผู้ใช้ เช่น การเปลี่ยนฟอนต์ เป็นต้น บางครั้งอาจเกิดเหตุการณ์ที่ไม่ปกติ เช่น บัฟเฟอร์สำหรับรับข้อมูลเต็ม (เนื่องจากเครื่องพิมพ์เป็นอุปกรณ์ที่ทำงานทางกลย่อมทำงานได้ช้ากว่าการส่งข้อมูลของคอมพิวเตอร์) เครื่องพิมพ์จะต้องแจ้งไปยังคอมพิวเตอร์ว่าให้หยุดส่งข้อมูลชั่วคราว เนื่องจากไม่สามารถรับข้อมูลได้มากกว่านี้แล้ว สัญญาณที่ส่งจากเครื่องพิมพ์ไปยังคอมพิวเตอร์คือ สัญญาณ Busy และเมื่อเครื่องพิมพ์เกิดข้อผิดพลาด เช่น กระดาษติด เครื่องพิมพ์จะต้องแจ้งไปยังคอมพิวเตอร์เช่นกัน โดยสัญญาณที่แจ้งไปยังคอมพิวเตอร์เรียกว่าสัญญาณ Error นอกจากนี้เมื่อคอมพิวเตอร์ต้องการรีเซตเครื่องพิมพ์ คอมพิวเตอร์จะต้องส่งสัญญาณ Reset ไปยังเครื่องพิมพ์เพื่อรีเซตเครื่องพิมพ์ด้วย สามารถสรุปขาสัญญาณที่จำเป็นสำหรับการติดต่อดังในตารางที่ 3.1

สัญญาณ	หน้าที่การทำงาน	ทิศทาง
ข้อมูล 8 บิต	ข้อมูลที่ส่งจากคอมพิวเตอร์ไปยังเครื่องพิมพ์	คอมพิวเตอร์
Strobe	แจ้งเครื่องพิมพ์ถึงข้อมูลที่ส่งมา	คอมพิวเตอร์
Acknowledge	เครื่องพิมพ์แจ้งมายังคอมพิวเตอร์ว่าได้รับข้อมูลแล้ว	เครื่องพิมพ์
Busy	แจ้งสถานะว่าเครื่องพิมพ์ไม่ว่างที่จะรับข้อมูลใหม่	เครื่องพิมพ์
Error	แจ้งสถานะว่าเครื่องพิมพ์เกิดข้อผิดพลาด	เครื่องพิมพ์
Reset	รีเซตเครื่องพิมพ์	คอมพิวเตอร์

ตารางที่ 3.1 แสดงสัญญาณที่สำคัญของพอร์ตขนานที่ใช้ติดต่อกับเครื่องพิมพ์

จากตารางที่ 3.1 จะเห็นว่าพอร์ตขนานของคอมพิวเตอร์ยังแยกย่อยออกเป็นอีก 3 พอร์ต ได้แก่ พอร์ตเอาต์พุตที่ทำหน้าที่ส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์ พอร์ตเอาต์พุตสำหรับสัญญาณ Strobe และ Reset พอร์ตอินพุตสำหรับการอ่านค่าสัญญาณ Acknowledge, Busy และสัญญาณ Error จากเครื่องพิมพ์

โดยปกติพอร์ตขนานออกแบบมาให้มีสัญญาณอยู่ทั้งหมด 17 เส้น สายสัญญาณเหล่านั้นจะมีรีจิสเตอร์ 3 ตัวควบคุมการทำงานดังนี้

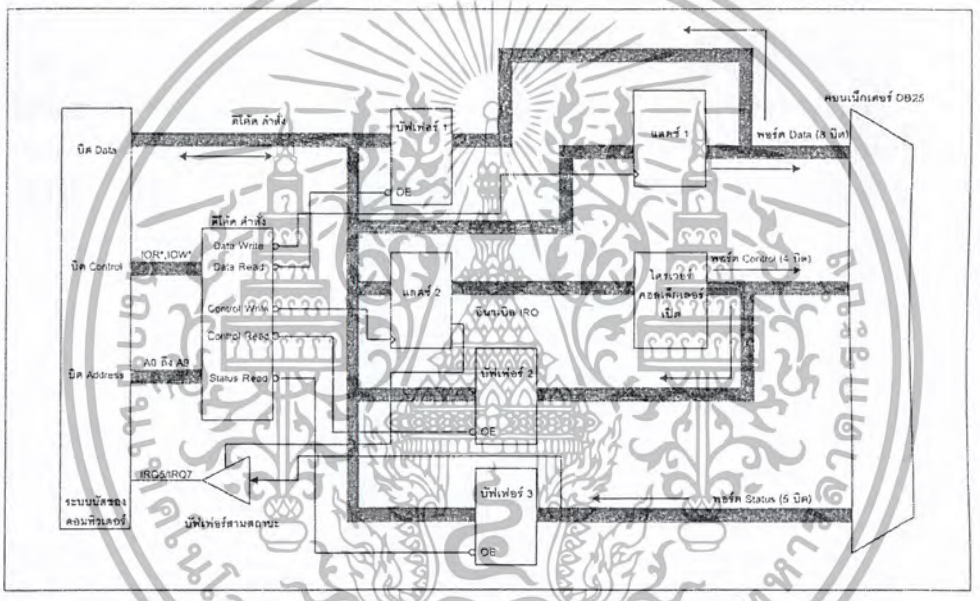
1.พอร์ตเอาต์พุตสำหรับสัญญาณข้อมูล 8 เส้น มีรีจิสเตอร์ Data ควบคุม

2.พอร์ตเอาต์พุตสำหรับการอ่านค่าสถานะต่าง ๆ จากภายนอกมีด้วยกัน 5 เส้น ใช้รีจิสเตอร์ Status ในการควบคุม

3.พอร์ตเอาต์พุตสำหรับส่งสัญญาณควบคุมไปยังอุปกรณ์ภายนอก มีอยู่ด้วยกัน 4 เส้น ใช้รีจิสเตอร์ Control ในการควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บล็อกไดอะแกรมในรูปที่ 3.2 แสดงระบบบัสของคอมพิวเตอร์สำหรับการติดต่อกับพอร์ตขนาน สัญญาณเอาต์พุตจากพอร์ตขนานจะถูกส่งไปยังคอนเน็กเตอร์แบบ DB-25 สำหรับ คอมพิวเตอร์ส่วนใหญ่ในปัจจุบันพอร์ตขนานจะมีมาพร้อมกับเมนบอร์ด ไม่จำเป็นต้องใช้การ์ดเสียบเหมือนในอดีต พร้อมทั้งมีฟังก์ชั่นการทำงานที่ซับซ้อนขึ้น แต่ยังคงสนับสนุนการทำงานของพอร์ตขนานในรูปแบบมาตรฐาน (SPP) อยู่ เมื่อดูจากรูป 3.1 เทียบการทำงานโดยทั่วไปกับการเชื่อมต่อผ่านการ์ดที่เสียบลงในสล็อตของคอมพิวเตอร์แล้ว พอร์ตขนานจะมีลักษณะใกล้เคียงกัน โดยการติดต่อกับพอร์ตขนานจะต้องมีการอ้างแอดเดรส ตำแหน่งแอดเดรสที่ใช้อ้างถึงจะเป็นตำแหน่ง A0 – A9 และใช้ขา IOR และ IOW สำหรับเป็นตัวเลือกว่าต้องการเขียนหรืออ่านรีจิสเตอร์ตัวใด จากการตี้โค้ดแอดเดรส A0 – A9 นี้เองทำให้ได้สัญญาณออกมาเพื่อควบคุมหรือเอ็นเอเบิลวงจรบัฟเฟอร์ต่าง ๆ ดังนี้



รูปที่ 3.2 แสดงระบบบัสภายในของพอร์ตขนาน

DataWrite สัญญาณเอ็นเอเบิลสำหรับนำข้อมูลที่อยู่ในบัส Data ของพอร์ตขนาน

DataRead สัญญาณเอ็นเอเบิลสำหรับอ่านข้อมูลจาก Data ของพอร์ตขนานมาเก็บไว้ในบัส Data

ControlWrite สัญญาณเอ็นเอเบิลสำหรับนำบัสข้อมูลที่อยู่ในบัส Data ไปออกที่ขา Control ของพอร์ตขนาน สำหรับพอร์ตนี้นอกจากจะส่งข้อมูลออกไปยังพอร์ตขนานแล้ว ยังทำหน้าที่อีนาเบิลการอินเตอร์รับของการเปลี่ยนแปลงสัญญาณที่พอร์ต Status อีกด้วย

ControlRead สัญญาณเอ็นเอเบิลสำหรับอ่านค่าข้อมูลจากขา Control มาเก็บไว้ในบัส Data

StatusRead สัญญาณเอ็นเอเบิลสำหรับอ่านค่าข้อมูลจากพอร์ต Status มาเก็บไว้ในบัส Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ตดาต้า (Data Port)

จากรูปที่ 3.3 แสดงให้เห็นว่าพอร์ต Data ประกอบได้ด้วยบัฟเฟอร์ 1 ตัวและไอซีแลตช์อีก 1 ตัว เมื่อคอมพิวเตอร์ต้องการส่งข้อมูลไปยังเครื่องพิมพ์ คอมพิวเตอร์จะเขียนข้อมูลไปยังไอซีแลตช์ 1 ทั้ง 8 บิต เอาต์พุตของไอซีแลตช์ 1 คือ D0-D7 ซึ่งเอาต์พุตนี้จะไปปรากฏอยู่ที่พอร์ตขนานในตำแหน่งขา 2 ถึงขา 9 และที่ขาเอาต์พุตนี้สัญญาณ Data จะส่งกลับไปเป็นอินพุตของบัฟเฟอร์ 1 ด้วย ทำให้คอมพิวเตอร์สามารถอ่านค่าในสถานะปัจจุบันที่เกิดขึ้นกับพอร์ต Data ได้

เมื่อคอมพิวเตอร์ส่งข้อมูล ข้อมูลจะถูกส่งมายังบัสข้อมูลของคอมพิวเตอร์ผ่านไปให้กับไอซี 74LS374 ซึ่งเป็นไอซีแลตช์ข้อมูล และเมื่อต้องการให้ข้อมูลปรากฏที่เอาต์พุต คอมพิวเตอร์จะส่งสัญญาณ DataWrite ออกไปที่ขา CLK ของ 74LS374 เอาต์พุตจาก 74LS374 จะถูกกรองด้วยวงจร RC ซึ่งประกอบด้วยตัวต้านทาน 27 โอห์ม และตัวเก็บประจุ 0.0022 uF เพื่อให้ช่วงเวลาที่เปลี่ยนจากลอจิก "0" เป็นลอจิก "1" หรือจากลอจิก "1" เป็นลอจิก "0" เป็นไปอย่างช้า ๆ เนื่องจากการเปลี่ยนแรงดันที่รวดเร็วทำให้เกิดสัญญาณรบกวนเหนี่ยวนำข้ามไปยังบิตข้อมูลอื่น ๆ ได้ ทำให้ข้อมูลที่ส่งออกไปมีข้อผิดพลาด จากค่าตัวต้านทานและตัวเก็บประจุในวงจรทำให้เกิดการหน่วงเวลาไปประมาณ 60 นาโนวินาที จากวงจรในรูปที่ 3.3 ทำให้เอาต์พุตของพอร์ต Data มีคุณสมบัติดังนี้

- กระแสซิงก์สูงสุด 24 mA
- กระแสซอร์สสูงสุด 2.6 mA
- ระดับแรงดันของลอจิก "1" ต่ำสุดเท่ากับ 2.4 V
- ระดับแรงดันสูงสุดสำหรับลอจิก "0" เท่ากับ 0.5 V

สำหรับบัฟเฟอร์สำหรับการอ่านข้อมูลกลับได้แก่เบอร์ 74LS244 ซึ่งเมื่อต้องการอ่านค่าคอมพิวเตอร์จะส่งสัญญาณ DataRead ออกมาเพื่อเอ็นเอเบิลไอซี 74LS244 สำหรับพอร์ตขนานแบบมาตรฐาน (Standard Parallel Port : SPP) พอร์ต Data จะต้องใช้เพื่อการส่งค่าออกเอาต์พุตเท่านั้น แต่สำหรับพอร์ตขนานที่มีการสื่อสารสองทิศทางสามารถอ่านค่าจากพอร์ต Data ได้ด้วย แต่ก่อนอื่นที่จะอ่านค่าต้องจำเสมอว่าจะต้องป้อนค่าเอาต์พุตให้มีค่าลอจิก "1" ทั้งหมดก่อน

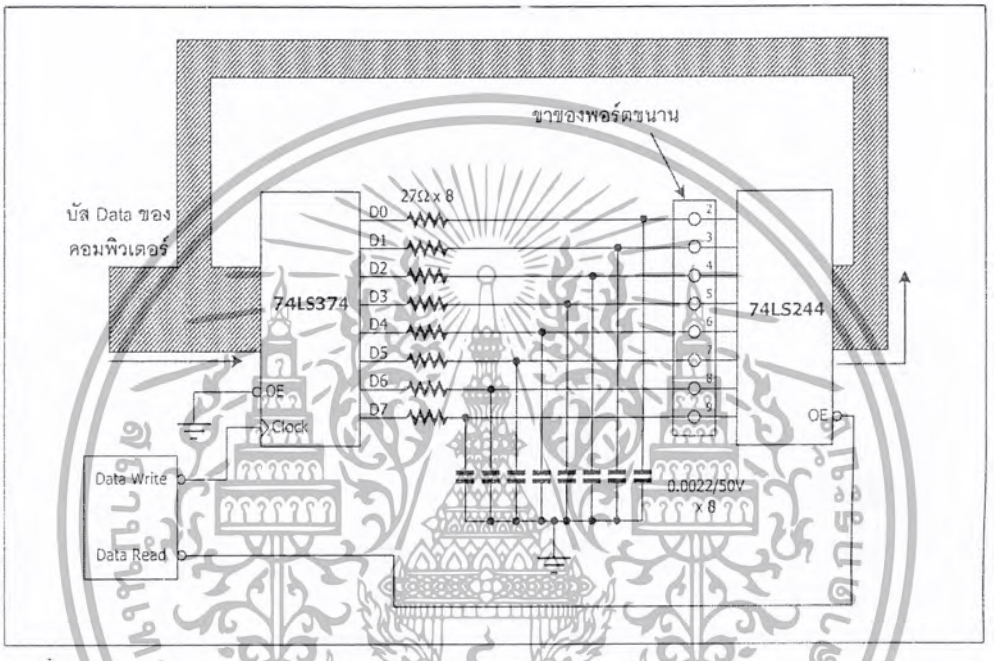
พอร์ต Control

พอร์ต Control ใช้สำหรับคอมพิวเตอร์ควบคุมเครื่องพิมพ์ พอร์ต Control จะประกอบไปด้วยเอาต์พุต 4 บิตที่ต่อออกไปยังเครื่องพิมพ์ ส่วนบิตเอ็นเอเบิลอินเตอร์รัปต์ไม่ได้ถูกต่อออกไป รูปที่ 3.4 แสดงบล็อกไดอะแกรมของพอร์ต Control จะเห็นว่าเอาต์พุตของพอร์ต Control มีอินเวอร์เตอร์แบบคอลเล็กเตอร์ปิดต่อร่วมอยู่ โดยเอาต์พุตเหล่านี้จะถูกพูลอัปไว้ด้วยตัวต้านทานขนาด 4.7 kΩ สำหรับบิต C2 ไม่มีการกลับสถานะลอจิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถานะของพอร์ต Control สามารถอ่านกลับได้และด้วยการใช้บัฟเฟอร์เบอร์ 74LS240 ซึ่งมีเอาต์พุตมีอินเวอร์เตอร์ต่ออยู่ภายใน ทำให้ค่าที่อ่านได้ตรงกับค่าส่งออกไป การควบคุมการอ่านและเขียนข้อมูลกับพอร์ต Control คอมพิวเตอร์จะส่งข้อมูลมาที่ขา Control Write และ Control Read

เนื่องจากเอาต์พุตของพอร์ต Control เป็นแบบคอเล็กเตอร์เปิด ดังนั้นผู้ใช้งานสามารถใช้พอร์ตนี้ในการอ่านค่าสัญญาณอินพุตจากภายนอกได้ โดยก่อนที่จะอ่านค่าจะต้องทำให้ขาพอร์ตที่ต้องการอ่านมีค่าลอจิก "1" เสียก่อน

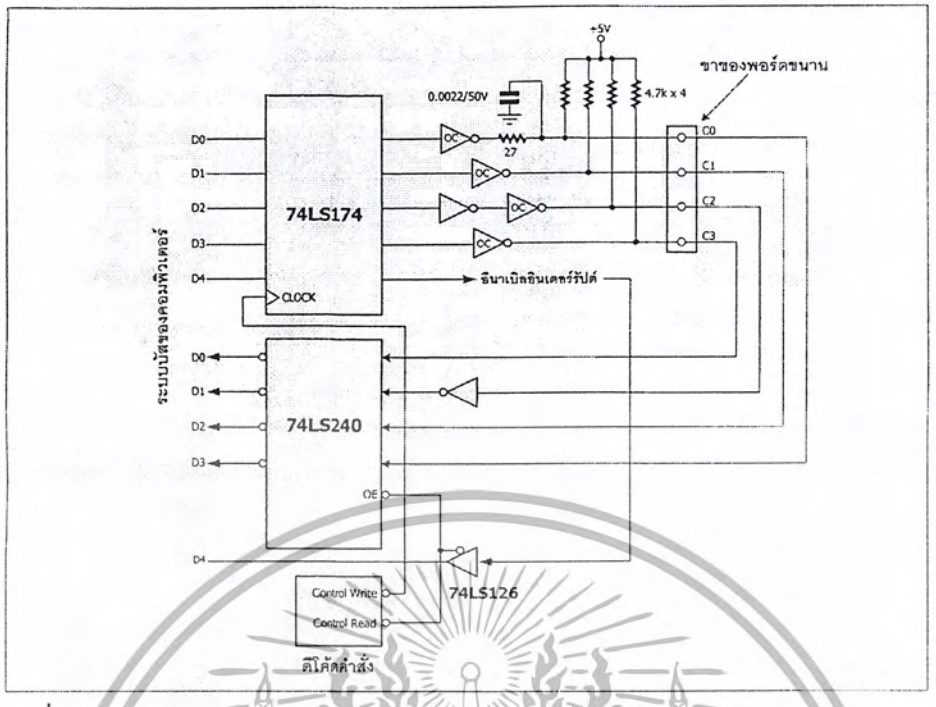


รูปที่ 3.3 วงจรภายในของพอร์ต Data

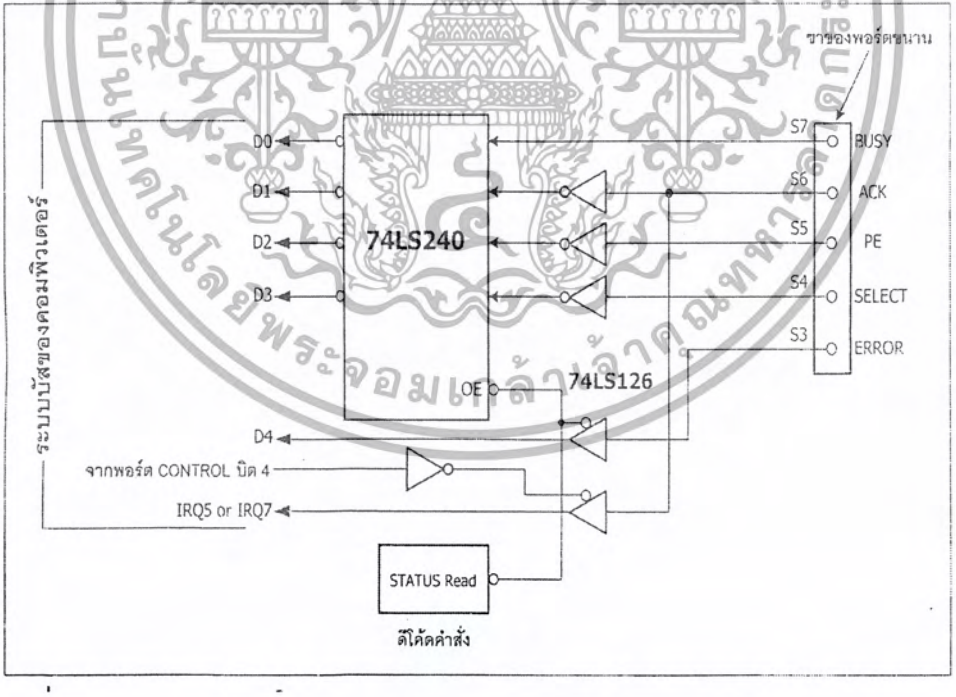
พอร์ตแสดงสถานะหรือพอร์ต Status

พอร์ต Status เป็นพอร์ตที่คอมพิวเตอร์ใช้สำหรับการอ่านค่าสถานะจากเครื่องพิมพ์ รูปที่ 3.5 แสดงลายละเอียดภายในของพอร์ต Status จะสังเกตได้ว่ามีขาสัญญาณอยู่ 5 สัญญาณด้วยกันและจะเรียกชื่อเป็น S3, S4, S5, S6 และ S7 ซึ่งตัวเลขนั้นหมายถึงตำแหน่งบิตของขาเหล่านี้ภายในรีจิสเตอร์ Status นั่นเอง สำหรับบิต S7 จะมีข้อแตกต่างจากบิตอื่น ๆ ที่เมื่อสัญญาณจากภายนอกส่งเข้ามาแล้วจะไม่มีอินเวอร์เตอร์ ในขณะที่ขาอื่น ๆ ผ่านอินเวอร์เตอร์ทั้งหมด ดังนั้นเมื่อข้อมูลผ่านจากขาอินพุตไปยัง 74LS240 ซึ่งเอาต์พุตมีการกลับสถานะทำให้บิต S7 เป็นบิตเดียวที่มีการกลับสถานะ นอกจากนี้วเลาใช้งานถ้าต้องการให้มีการสร้างสัญญาณอินเตอร์รัปต์จากขอบขาขึ้นของขา S6 สามารถกำหนดค่าได้จากพอร์ต Control บิต 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 วงจรภายในของพอร์ต Control



รูปที่ 3.5 แสดงวงจรภายในของพอร์ตสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การติดต่อผ่านพอร์ตขนาน (Parallel Interfacing)

พอร์ตขนานของคอมพิวเตอร์จะมีลักษณะเช่นเดียวกับอุปกรณ์อินพุตเอาต์พุตตัวอื่น ๆ คือเมื่อต้องการติดต่อจะต้องกำหนดแอดเดรสที่ต้องการติดต่อด้วย ดังตารางที่ 3.2 แสดงแอดเดรสของพอร์ตขนาน โดยแบ่งออกเป็น 3 ตำแหน่ง คือ แอดเดรสของรีจิสเตอร์ Data , รีจิสเตอร์ Status , และรีจิสเตอร์ Control

ชื่อพอร์ต	LP1		การนำไปใช้งานกับ
	ฐานสิบ	ฐานสิบหก	CNC – Compact 5
Data	888	378 H	ใช้ควบคุมการทำงานของ CNC
Status	889	379 H	ใช้รับค่าจาก CNC แล้วนำไปแสดง
Control	890	37A H	-----

ตารางที่ 3.2 แสดงแอดเดรสของพอร์ตขนาน

เมื่อต้องการติดต่อกับพอร์ตขนานในตำแหน่งใด ก็ให้ส่งค่าของข้อมูลออกไปที่พอร์ตขนานในตำแหน่งนั้น ๆ ยกตัวอย่างเช่น

Out &H378, &HFF

โดยที่ คำสั่ง Out เป็นการส่งค่าข้อมูลออก

ค่า 378 เป็นแอดเดรสของรีจิสเตอร์ Data

ค่า FF เป็นข้อมูลเลขฐานสิบหก หมายถึงการให้บิตทุกบิตของรีจิสเตอร์ Data มีลอจิกเป็น "1" ทั้งหมด

ส่วนการอ่านค่าจาก CNC – Compact 5 มายังคอมพิวเตอร์ผ่านพอร์ต Status ของพอร์ตขนาน(LP1) สามารถเขียนโปรแกรมดังเช่น

Data = Inp (&H379)

โดยที่ คำสั่ง Inp เป็นคำสั่งการอ่านค่าข้อมูล

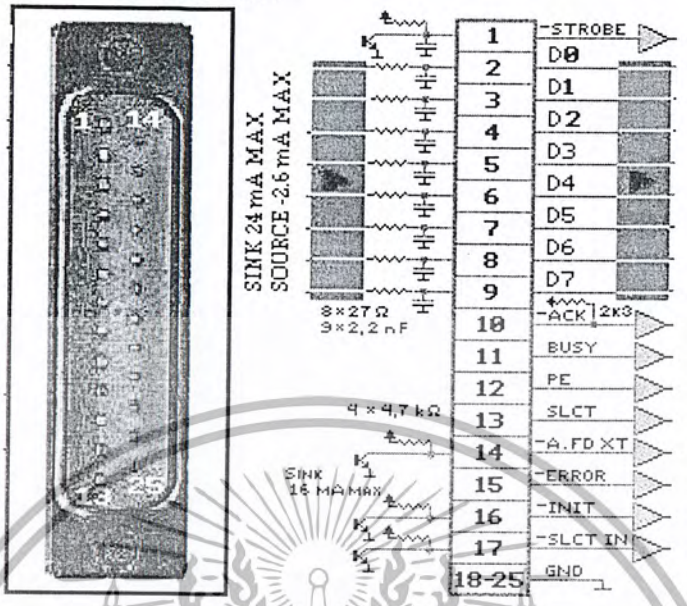
ค่า 379 เป็นตำแหน่งแอดเดรสของรีจิสเตอร์ Status

ค่า Data เป็นตัวแปรที่ใช้เก็บข้อมูลที่อ่านได้จากพอร์ตขนานซึ่งเป็นค่าเลขฐานสิบ

การนำพอร์ตขนานไปใช้งานนั้นจะต้องรู้ตำแหน่งของขาต่าง ๆ ดังรูปที่ 3.6 ซึ่งจะต้องนำขาต่าง ๆ ของพอร์ตขนานต่อกับ DNC – Interface ของ CNC – Compact 5 ซึ่งการต่อนั้นจะแสดงในตารางที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อต่อที่ PC 25 ขา ตัวเมีย



ตำแหน่งขา	ชื่อพอร์ต
2 – 9	Data (D_0 - D_7)
10,11,12,13,15	Status (S_3 - S_7)
1,14,16,17	Control (C_0 - C_3)
18 – 25	GND

รูปที่ 3.6 แสดงขาของพอร์ตขนาน

ขาDNC Interface (X62/PIN)	ตำแหน่งขา Parallel Port	หน้าที่	การทำงาน	วิธีการนำไปใช้งาน
1	15	Status hand	Output	รับค่าสถานะ
2	2	Turret – hand operation	Input	ส่งค่า H01
3	3	Instruction G66 + INP	Input	ส่งค่า H02
4	--	----	--	----
5	--	----	--	----
6	4	Instruction G66 + FWD	Input	----

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7	13	Status program running	Output	รับค่าสถานะ
8	12	Status intermediate stop	Output	รับค่าสถานะ
9	5	Instruction switch hand/	Input	ส่งค่า H08
10	--	----	--	----
11	--	----	--	----
12	--	----	--	----
13	--	----	--	----
14	--	----	--	----
15	--	Output set with M8,M9	Output	----
16	--	----	--	----
17	6	Instruction start	Input	ส่งค่า H10
18	--	Output set with M22,M23	Output	----
19	10	Status main motor	Output	รับค่าสถานะ
20	--	Output impulse set M26	Output	----
21	7	Instruction blockage-turret	Input	ส่งค่า H20
22	--	+ 10 V not controlled	Power	---
23	18	GND	--	----
24	--	GND	--	----
25	--	GND	--	----
26	--	+ 5 V controlled	Power	----

ตารางที่ 3.3 แสดงการเชื่อมต่อระหว่าง Parallel port กับ DNC – Interface

การเขียนโปรแกรมติดต่อกับพอร์ตขนานด้วย Visual Basic 6

การเขียนโปรแกรมด้วย Visual Basic 6 ชุดคำสั่งส่วนใหญ่จะมีรูปแบบใกล้เคียงกับ Qbasic แต่ Visual Basic 6 จะไม่มีคำสั่งสำหรับการติดต่อกับพอร์ตขนานโดยตรงคือ คำสั่ง Inp ( ) และคำสั่ง Out ( ) ดังนั้นเพื่อให้ติดต่อกับพอร์ตขนานได้จึงจำเป็นต้องเพิ่มโปรแกรมเข้าไปโดยโปรแกรมที่เพิ่มเข้ามานี้จะอยู่ในรูปแบบของ DLL (Dynamic Linked Library)

ไฟล์ DLL นี้คือ io.dll สำหรับตำแหน่งที่ใช้เก็บไฟล์ io.dll นั้นจะต้องเก็บไว้ที่ไดเรกตอรี่ SYSTEM ซึ่งอยู่ภายในไดเรกตอรี่ที่เก็บโปรแกรมวินโดวส์ โดยส่วนใหญ่จะมีชื่อเป็น Windows

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยสามารถดาวน์โหลดไฟล์นี้ได้ที่..http://www.geekhideout.com...ซึ่งไฟล์นี้ได้ถูกพัฒนาโดยโปรแกรมเมอร์นิรนามที่ใช้ชื่อว่า **Fred**

การเรียกใช้ไฟล์ io.dll จะต้องมีการประกาศ Function ไว้ใน Module เพื่อบอกให้ Visual Basic รู้ว่าคำสั่ง Inp ( ), Out ( ) อยู่ใน Library ใด ซึ่งมีรูปแบบการกำหนดค่าดังนี้

```
Public Declare Sub Out Lib "io.dll" Alias "PortOut" (ByVal Port As Integer, ByVal Data As Byte)
Public Declare Function Inp Lib "io.dll" Alias "PortIn" (ByVal Port As Integer) As Byte
```

การเขียนโปรแกรมเพื่อสั่งให้ CNC – Compact 5 ทำงานตามปุ่มที่เรากดในโปรแกรม Visual Basic เราสามารถเขียนโค้ดได้ดังนี้

```
Private Sub cmdhand_Click()          'เมื่อคลิกปุ่มนี้จะทำให้เปลี่ยนโหมดระหว่าง hand /CNC
Out &H378, &H8
Call delay
Out &H378, &H0
End Sub
Private Sub cmdinp_Click()           'เมื่อคลิกปุ่มนี้จะทำการเปิดพอร์ตของ CNC เพื่อทำการโหลด G-Code
Out &H378, &H2
Call delay
Out &H378, &H0
End Sub
Private Sub cmdstart_Click()         'เมื่อคลิกปุ่มนี้จะทำการสตาร์ทเครื่อง CNC
Out &H378, &H10
Call delay
Out &H378, &H0
End Sub

Private Sub delay()                  'เป็นโปรมแกรมย่อยเพื่อทำการหน่วงสัญญาณเอาต์พุต
Dim i As Variant
  For i = 1 To 100000
     i = i + 1
  Next i
End Sub
Private Sub Form_Load()              'เมื่อทำการเรียกฟอร์มจะทำให้เอาต์พุตทุกบิตเป็น "0"
```

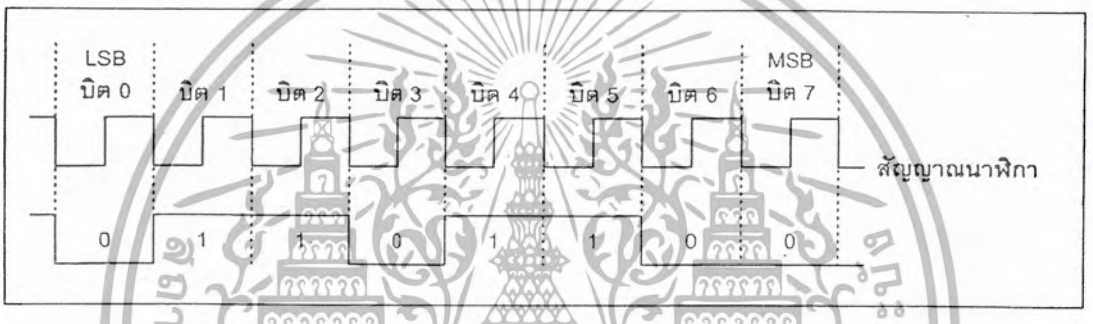
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Out &H378, &H0

End Sub

3.3 การติดต่อผ่านพอร์ตอนุกรม (Serial Interfacing)

การสื่อสารแบบอนุกรมนั้นจะแบ่งออกได้เป็น 2 แบบคือการสื่อสารอนุกรมแบบซิงโครนัส และการสื่อสารอนุกรมแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิการ่วมอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือคีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้นคือ สัญญาณนาฬิกา, ข้อมูลและกราวด์



รูปที่ 3.7 แสดงไทมิ่งไดอะแกรมของการส่งข้อมูลแบบซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัส

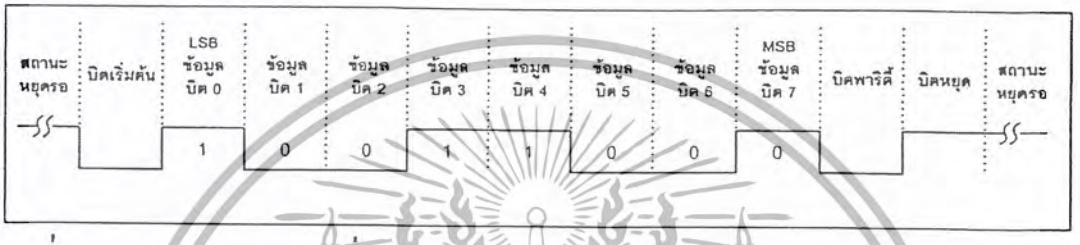
การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับและส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วยเหมือนกับการส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งนี้ว่า **อัตราการถ่ายทอดข้อมูล** หรือ **บอดเรต (Baudrate)** มีหน่วยเป็น บิตต่อวินาที (bit per second : bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนคือ

1. บิตเริ่มต้น (Start Bit) ซึ่งจะมีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรมจะมีขนาด 5,6,7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี้ (Parity Bit) จะมีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้าย (Stop Bit) จะมีขนาด 1,1.5 หรือ 2 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.3 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส ซึ่งเมื่อไม่มีข้อมูลที่จะส่ง ขา DATA จะมีสถานะลอจิก "1" ซึ่งจะเรียกสถานะนี้ว่าสถานะหยุดรอ (Waiting Stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขา DATA มีลอจิก "0" ด้วยช่วงเวลา 1 บิต ซึ่งจะเรียกบิตนี้ว่าบิตเริ่มต้น จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อน ซึ่งข้อมูลในไบต์ที่จะส่งอาจจะมีจำนวนบิต 5, 6, 7 หรือ 8 บิตก็ได้ จากนั้นจะตามด้วยบิตพาริตี้ ซึ่งใช้เพื่อตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่จะส่งคือบิตปิดท้าย ซึ่งจะให้ขาดาต้ามีสถานะลอจิก "1" อีกครั้งด้วยระยะเวลาอย่างน้อย 1 , 1.5 หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว



รูปที่ 3.8 แสดงไทมิ่งไดอะแกรมของการส่งข้อมูลแบบอะซิงโครนัส

อุปกรณ์พิเศษที่ได้รับการออกแบบมาสำหรับการรับและส่งข้อมูลแบบอะซิงโครนัสเรียกว่า Universal Asynchronous Receiver/Transmitter หรือ UART อัตราความเร็วในการรับและส่งข้อมูล (ค่าบอดเรต) จะมีค่ามาตรฐานที่ใช้สำหรับพอร์ตอนุกรม RS-232 ได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200 บิตต่อวินาที และมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ซึ่งการรับส่งแบบอนุกรมโดยไม่ผ่านโมเด็มอาจจะกำหนดค่าบอดเรตได้สูงถึง 115200 บิตต่อวินาที เนื่องจากบอดเรตคือจำนวนบิตของข้อมูลที่สามารถถ่ายทอดได้ภายใน 1 วินาที

ยกตัวอย่างข้อมูลแบบอนุกรมถูกส่งในลักษณะ 8 บิต ไม่มีการตรวจสอบพาริตี้ มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูลที่รับส่งนี้เท่ากับ 10 บิต ถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที และถ้ามีการใช้พาริตี้ความเร็วในการส่งข้อมูลจะเหลือเป็น 872 ไบต์ต่อวินาที

การตรวจสอบพาริตี้สามารถกำหนดให้เป็นแบบคี่ (odd) แบบคู่ (even) หรือไม่มีการตรวจสอบพาริตี้ก็ได้ การตรวจสอบพาริตี้เป็นการตรวจสอบจำนวนรวมของบิตที่เป็นลอจิก "1" ภายในข้อมูลที่ส่งไป 1 ไบต์ว่ามีจำนวนรวมเป็นเลขคู่หรือเลขคี่โดยต้องรวมบิตพาริตี้เข้าไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล	บิตพาริตี้คู่	บิตพาริตี้คี่
00000000	0	1
00000001	1	0
00000010	1	0
00000011	0	1
00000100	1	0
11111110	0	1
11111111	1	0

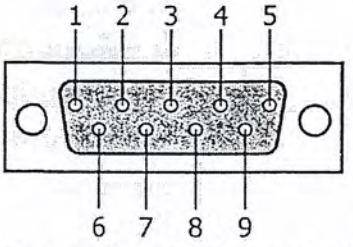
ตารางที่ 3.4 แสดงบิตพาริตี้ของข้อมูล

ยกตัวอย่างเช่นข้อมูลที่จะทำการส่งมีขนาด 8 บิตและมีค่าเท่ากับ 99 ฐานสิบหก หรือ 10011001 ฐานสอง จะเห็นว่าข้อมูลในไบต์นี้มีจำนวนลอจิก "1" จำนวน 4 ตัวซึ่งเป็นเลขคู่ ดังนั้นถ้ากำหนดค่าพาริตี้เป็นคู่ค่าในบิตพาริตี้ จะต้องมีลอจิกเป็น "0" แต่ถ้าพาริตี้เป็นคี่ ค่าที่บิตพาริตี้จะต้องเป็น "1" เพื่อให้ข้อมูล 1 ไบต์รวมทั้งบิตพาริตี้มีจำนวนบิตที่เป็นลอจิก "1" มีจำนวนรวมกันเป็นเลขคี่ ในตารางที่ 3.3 แสดงตัวอย่างของบิตพาริตี้ในการรับส่งข้อมูลอนุกรม บิตพาริตี้ถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART ซึ่งทางภาครับจะต้องทำการกำหนดคุณสมบัติการตรวจสอบพาริตี้ให้ตรงกันว่าจะตรวจสอบพาริตี้คี่หรือพาริตี้คู่ จากนั้นภาครับของ UART จะทำการตรวจสอบค่าพาริตี้ที่เกิดขึ้นว่าเป็นคู่หรือเป็นคี่ โดยการนับจำนวนลอจิก "1" ทั้งหมดรวมทั้งบิตพาริตี้ด้วย นับเป็นการตรวจสอบความผิดพลาดที่เกิดขึ้นในการถ่ายทอดข้อมูลที่ง่ายที่สุด แต่จะเชื่อถือได้เมื่อมีบิตข้อมูลที่ทำการส่งผิดพลาดเพียงบิตเดียวเท่านั้น ถ้าข้อมูลที่ทำการส่งมีบิตที่ผิดพลาดมากกว่า 1 บิต การตรวจสอบด้วยวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตี้บิตเป็น NONE นั้นทั้งภาครับและภาคส่ง จะไม่มีการตรวจสอบพาริตี้

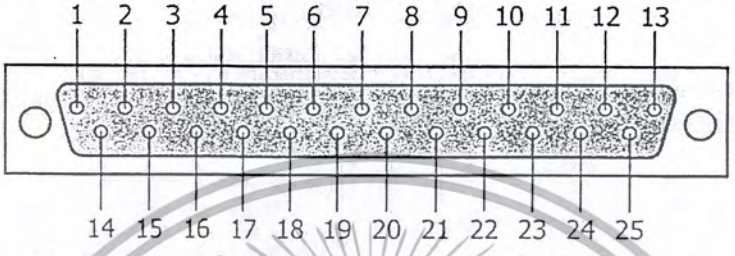
3.4 คอนเน็กเตอร์สำหรับพอร์ต RS – 232 และการเชื่อมต่อ

มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็กเตอร์แบบ DB-25 หรือ DB-9 ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้นเช่นเดียวกับแบบ DB-9 เนื่องจากขาอื่น ๆ ที่เคยใช้งานในอดีต ปัจจุบันมีการใช้งานไม่มากนักจึงถูกยกเลิกไป โดยแสดงดังรูปที่ 3.4 การติดต่อกับอุปกรณ์ภายนอกผ่านทาง RS-232 (Serial Port) ข้อดีของการติดต่อข้อมูลกันผ่านทาง RS-232 ก็คือสามารถใช้ได้ในระยะทางไกลมากกว่าการสื่อสารแบบพอร์ตขนาน และระหว่างอุปกรณ์ที่ติดต่อกันและใช้จำนวนของสายเพียง 2 – 3 เส้นเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) คอนเน็กเตอร์อนุกรม 9 ขาหรือแบบ DB-9 (มองจากด้านหลังคอมพิวเตอร์)



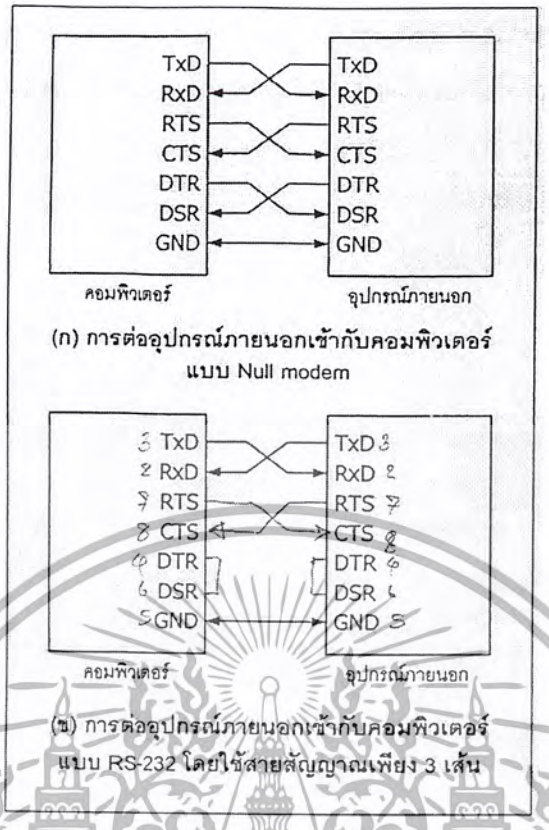
(ข) คอนเน็กเตอร์อนุกรม 25 ขาหรือแบบ DB-25 (มองจากด้านหลังคอมพิวเตอร์)

คอนเน็กเตอร์ DB-9	คอนเน็กเตอร์ DB-25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data Carrier Detect : DCD	อินพุต
2	3	Received Data : RxD	อินพุต
3	2	Transmitted Data : TxD	เอาต์พุต
4	20	Data Terminal Ready : DTR	เอาต์พุต
5	7	Signal Ground : GND	-
6	6	Data Set Ready : DSR	อินพุต
7	4	Request To Send : RTS	เอาต์พุต
8	5	Clear To Send : CTS	อินพุต
9	22	Ring Indicator : RI	อินพุต

รูปที่ 3.9 แสดงตำแหน่งขาของพอร์ตอนุกรม (RS-232)

สำหรับการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกแสดงดังรูปที่ 3.10 ลูกศรในรูปแสดงถึงทิศทางของข้อมูล ในรูปที่ 3.10(ก) เป็นการเชื่อมต่อแบบ Null modem หรือการเชื่อมต่อโดยตรงไม่ต้องผ่านโมเด็ม โดยมีการตรวจสอบหรือแฮนด์เช็กเต็มรูปแบบ ส่วนรูปที่ 3.10(ข) เป็นการเชื่อมต่อแบบ Null modem ในลักษณะที่ใช้สายสัญญาณเพียง 3 เส้น โดยเส้นหนึ่งสำหรับส่งข้อมูล อีกเส้นสำหรับรับข้อมูลและเส้นสุดท้ายเป็นกราวด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แสดงการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอก

หน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS-232 มีดังนี้

- **Data Carrier Detect : DCD** ขานี้จะแอกตีฟเมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการใช้งานปกติขานี้จะไม่ถูกใช้งานมากนัก
- **Receive Data : RD** หรือ **RxD** ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์โดยนำข้อมูลที่อ่านได้เก็บในรีจิสเตอร์บัฟเฟอร์
- **Transmitted Data : TD** หรือ **TxD** ขานี้ใช้เพื่อส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลออกไป
- **Data Terminal Relay : DTR** เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อด้วย โดยขา DTR นี้จะต้องต่อกับขา DSR ของอุปกรณ์ปลายทาง ถ้าต่อแบบ 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกัน
- **Signal Ground : GND** ขากราวด์ของระบบ
- **Data Set Ready : DSR** ขานี้จะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อระหว่างคอม
- คอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Request To Send : RTS** เป็นขาสำหรับส่งสัญญาณร้องขอให้อุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบ Null modem 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อให้การรับและส่งข้อมูลเกิดขึ้นตลอดเวลา
- **Clear To Send : CTS** ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ขา TxD จะถูกส่งออกไป ดังนั้นขานี้จึงใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่

การเชื่อมต่อคอมพิวเตอร์ผ่านทาง RS-232 เข้ากับ CNC-Compact 5

การเชื่อมต่อคอมพิวเตอร์ผ่านทาง RS-232 เข้ากับ CNC – Compact 5 เพื่อที่จะส่งค่า G-Code จากโปรแกรม Visual Basic 6 โหลดเข้า CNC-Compact 5 ซึ่งการเชื่อมต่อนั้นแสดงดังรูปที่ 3.11แสดงการเชื่อมต่อของขา DB-9 เข้ากับพอร์ต CNC-Compact 5 ซึ่งอัตราการถ่ายทอดข้อมูล หรือบอดเรต(baudrate) ของ CNC-Compact 5 จะใช้อยู่ที่ 300 บิตต่อวินาที

รูปที่ 3.11 แสดงการเชื่อมต่อขา DB-9 เข้ากับพอร์ตของ CNC-Compact 5

การตั้งค่าติดต่อกับพอร์ตโดยใช้ Visual Basic 6

สำหรับการใช้งาน Visual Basic 6 จะสามารถใช้คอนโทรล MSComm โดยที่ MSComm จัดเตรียมทางเลือกเอาไว้ 2 ทางเพื่อความสะดวกในการสื่อสารข้อมูล ทางแรกคือ การสื่อสารข้อมูลที่กระตุ้นด้วยเหตุการณ์ (event-driven communication) เป็นรูปแบบการใช้งานที่มีประสิทธิภาพมากสำหรับการตอบสนองแบบทันทีทันใด เช่น เมื่อตัวอักษถูกส่งมาที่พอร์ตอนุกรมหรือการเปลี่ยนแปลงที่ขา Data Carrier Detect (DCD) หรือขา Request To Sand (RTS) เหตุการณ์ Oncomm ของ MSComm จะสามารถตรวจจับสัญญาณนั้นได้ทันที ซึ่งจะกล่าวถึงรายละเอียดในหัวข้อคุณสมบัติ CommEvent ต่อไป ส่วนทางเลือกที่สองเป็นการคอยตรวจสอบค่าเหตุการณ์และความผิดพลาดที่เกิดขึ้นด้วยการดูค่าที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยนแปลงภายในคุณสมบัติ CommEvent หลังจากให้โปรแกรมทำงานในฟังก์ชั่นต่าง ๆ ไปเรียบร้อยแล้ว ซึ่งวิธีนี้ใช้งานได้ดีในกรณีที่โปรแกรมมีขนาดเล็ก

คอนโทรล MSComm 1ตัวสามารถควบคุมการทำงานของพอร์ตอนุกรมได้ 1 พอร์ต ถ้าในโปรแกรมที่ใช้งานต้องการติดต่อกับพอร์ตอนุกรมมากกว่า 1 พอร์ตจะต้องใช้คอนโทรล MSComm มากกว่า 1 ตัวเพื่อควบคุมพอร์ตอนุกรมในแต่ละพอร์ต แอดเดรสของพอร์ตอนุกรมและแอดเดรสของการอินเตอร์รัปต์สามารถเปลี่ยนแปลงได้จากการแก้ไขค่าที่ Control Panel

คอนโทรล MSComm จะมีคุณสมบัติ (property) ดังนี้

1. Property ชื่อ CommPort คือ เลือกคอมพอร์ตที่เราจะต่อใช้งาน ในที่นี้เลือกจะใช้ Com1 อยู่ที่ด้านหลังเครื่องคอมฯ

เช่น **MSComm1.CommPort = 1**

2. Property ชื่อ Settingsคือ การตั้งค่าของการรับส่งข้อมูล ซึ่งจะต้องรู้ด้วยว่าอัตราบอด ของอุปกรณ์ที่จะติดต่อด้วยเป็นเท่าไร โดยมีรายละเอียดการใส่ต่างๆค่าดังนี้

เช่น MSComm1.Settings = "Baud (อัตราการรับส่งข้อมูล) ,Parity(ถ้าไม่ใช้ใส่ N) ,จำนวนบิตข้อมูล, บิตสต๊อบ"

ค่าบอดเรตมาตรฐานที่ใช้กับ MSCOmm มีดังนี้

110 บิตต่อวินาที

300 บิตต่อวินาที (ค่าที่ใช้งานใน CNC-Compact 5)

600 บิตต่อวินาที

1,200 บิตต่อวินาที

2,400 บิตต่อวินาที

9,600 บิตต่อวินาที (ค่าปกติ)

14,400 บิตต่อวินาที

19,200 บิตต่อวินาที

28,800 บิตต่อวินาที

38,400 บิตต่อวินาที

56,000 บิตต่อวินาที

128,000 บิตต่อวินาที

สำหรับค่ามาตรฐานในการกำหนดค่าพาริตี้มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญลักษณ์	รายละเอียด
E	พาริตี้คู่ (Even)
M	ลอจิก " 1 " (MARK)
N	ไม่ใช้ (ค่าปกติ)
O	พาริตี้คี่ (Odd)
S	ลอจิก " 0 " (Space)

ค่าที่ใช้ในการกำหนดจำนวนบิตมี 5 ค่า คือ 4, 5, 6, 7 และ 8 (เป็นค่าปกติ)

ค่าที่ระบุจำนวนบิตปิดท้ายมี 3 ค่า คือ 1 (เป็นค่าปกติ) , 1.5, และ 2

เช่น **MSComm1.Settings = "1200, N, 7, 1"**

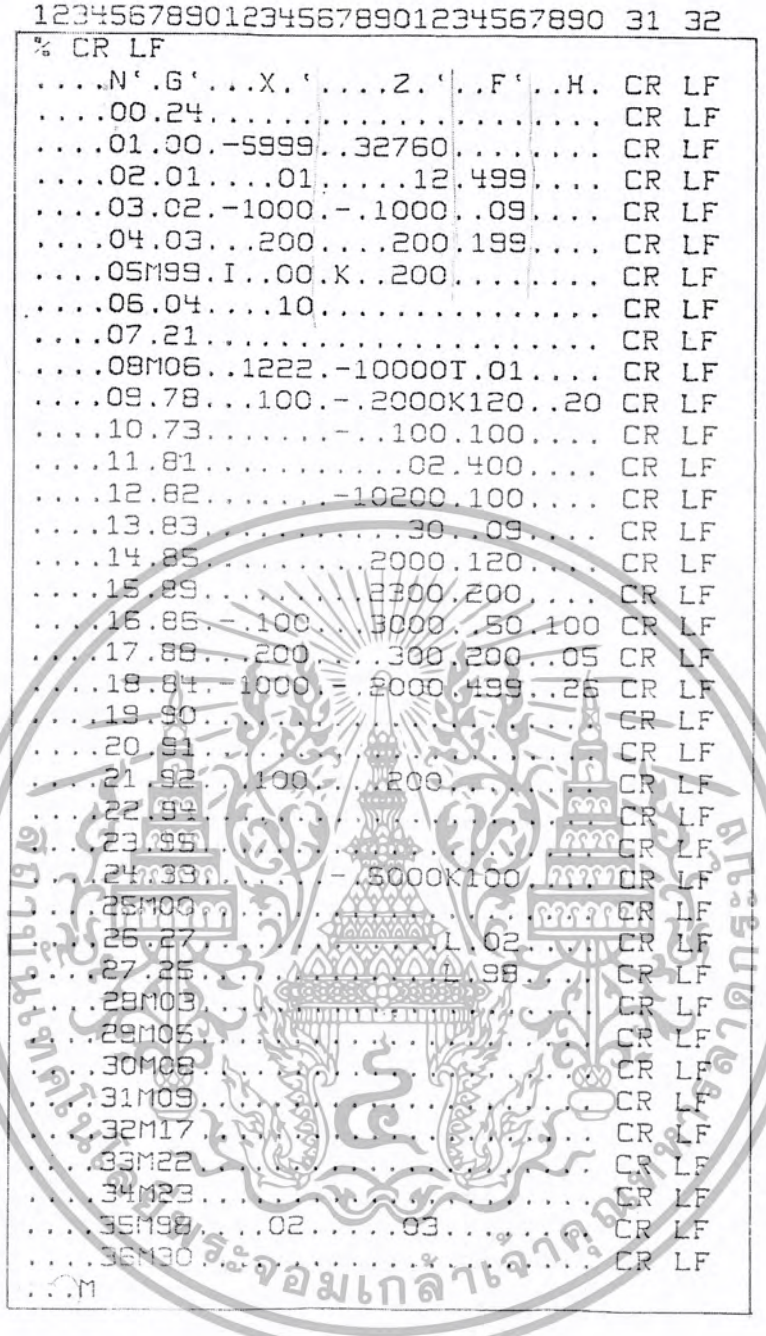
3.Propety ชื่อ PortOpen คือการกำหนดสถานะของพอร์ตอนุกรม เพื่อเปิดและปิดพอร์ตอนุกรม ซึ่งมี 2 สถานะ คือ True , False

เช่น **MSComm1.PortOpen = True**

การตั้งค่าคุณสมบัติ (Property) สามารถตั้งค่า Event อยู่ที่ Form_Load() ซึ่งทำให้ทุกครั้งที่เรียกฟอร์มจะตั้งค่าคุณสมบัติตามนี้

```
Private Sub Form_Load()
    MSComm1.Settings = " 300, N, 7, 1 "
    MSComm1.CommPort = 1
    MSComm1.PortOpen = True
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```
12345678901234567890123456789O 31 32
% CR LF
....N'.G'...X.'....Z.'..F'..H. CR LF
....00.24.....................  CR LF
....01.00.-5999..32760........  CR LF
....02.01....01.....12.499....  CR LF
....03.02.-1000.-.1000..09....  CR LF
....04.03...200....200.199....  CR LF
....05M99.I..00.K..200........  CR LF
....06.04....10...............  CR LF
....07.21.....................  CR LF
....08M06..1222.-10000T.01....  CR LF
....09.78...100.-.2000K120..20 CR LF
....10.73.......-..100.100....  CR LF
....11.81...........02.400....  CR LF
....12.82.......-10200.100....  CR LF
....13.83.............30..09...  CR LF
....14.85.........2000.120....  CR LF
....15.89.........2300.200....  CR LF
....16.86.-.100...3000..50.100 CR LF
....17.88...200....300.200..05 CR LF
....18.84.-1000.-.2000.499..26 CR LF
....19.90.....................  CR LF
....20.91.....................  CR LF
....21.92..100....200.........  CR LF
....22.94.....................  CR LF
....23.95.....................  CR LF
....24.38.......-5000K100.....  CR LF
....25M09.....................  CR LF
....26.27.............L.02....  CR LF
....27.25.............L.98....  CR LF
....28M03.....................  CR LF
....29M05.....................  CR LF
....30M08.....................  CR LF
....31M09.....................  CR LF
....32M17.....................  CR LF
....33M22.....................  CR LF
....34M23.....................  CR LF
....35M98....02.....03........  CR LF
....36M30.....................  CR LF
...M
```

รูปที่ 3.12 แสดงแบบฟอร์มการป้อน G-Code และ M-Code

เมื่อ จุด (.) คือ Space ASCII 32

(‘) ASCII 96

(CR) คือ Carriage Return ASCII 13

(LF) คือ Line Feed ASCII 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
สรุปและข้อเสนอแนะ

ข้อสรุป

โครงงานนี้เป็นการทำงานร่วมกันระหว่างแขนกลกับเครื่องมือกล CNC-Compact 5 โดยใช้คอมพิวเตอร์ควบคุมการทำงานของระบบ การติดต่อระหว่างคอมพิวเตอร์กับ CNC-Compact 5 จะใช้การเชื่อมต่อทั้ง Serial Port และ Parallel Port ซึ่ง Serial Port ทำหน้าที่ส่ง G-Code ให้แก่ CNC-Compact 5 ส่วน Parallel Port ทำหน้าที่ควบคุมการทำงานของ CNC-Compact 5 เช่น เปลี่ยนโหมดการทำงาน (Hand/CNC operation) และสตาร์ทการทำงาน ส่วนแขนกลนั้นจะใช้สัญญาณจาก Parallel Port เพื่อทริกการทำงานของแขนกลให้รับและส่งชิ้นงานให้แก่ CNC

ปัญหาที่เกิดขึ้นคือการจับยึดชิ้นงานเข้ากับหัวจับชิ้นงานของ CNC-Compact 5 นั้น จำเป็นที่จะต้องใช้มือขันหัวจับชิ้นงานก่อนที่จะเริ่มกลึงและหลังกลึงเสร็จแล้วเสมอ ส่วนแขนกลนั้นมีความผิดพลาดที่เกิดขึ้นได้ง่ายจากการซ้ำตำแหน่งเดิม ด้วย

ข้อเสนอแนะ

โครงงานนี้ควรมีการพัฒนาการต่อไปดังต่อไปนี้

1.ควรออกแบบหัวจับยึดชิ้นงานกลึงใหม่เพื่อให้ระบบทำงานแบบระบบอัตโนมัติได้อย่างสมบูรณ์แบบ

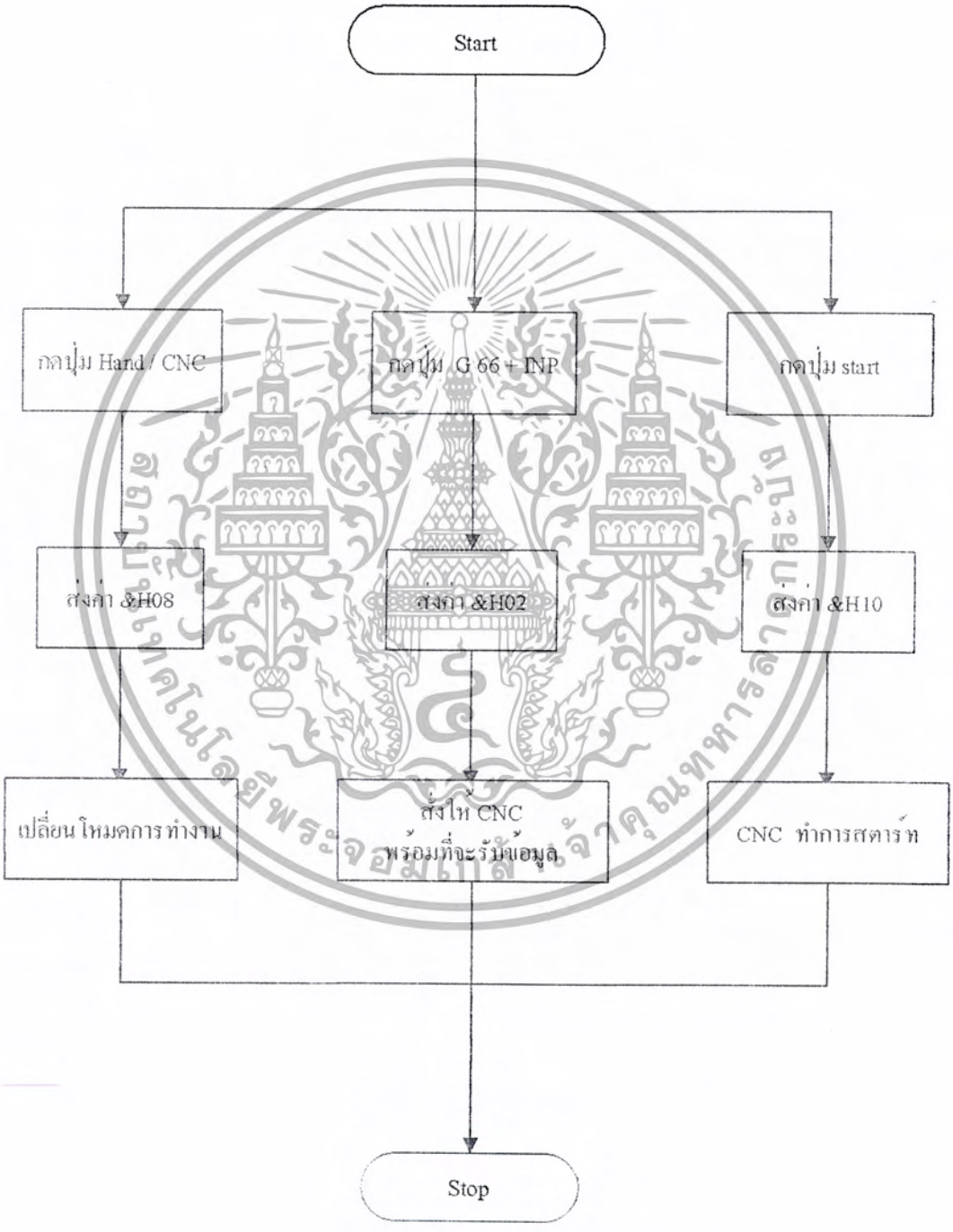
2.ควรสร้างกราฟฟิกแสดงการทำงานของระบบไม่ว่าจะเป็น แขนกลหรือการทำงานของ CNC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

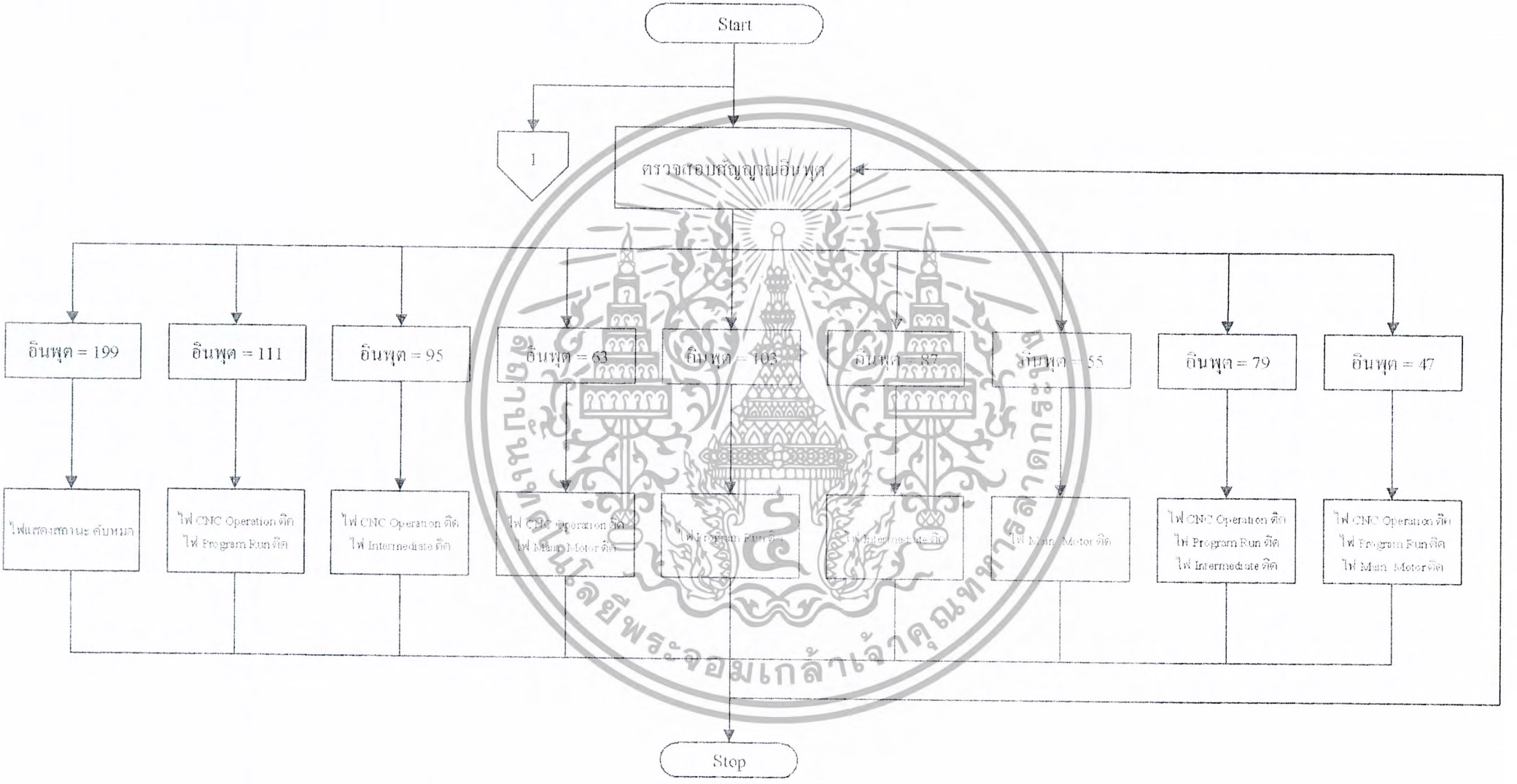
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

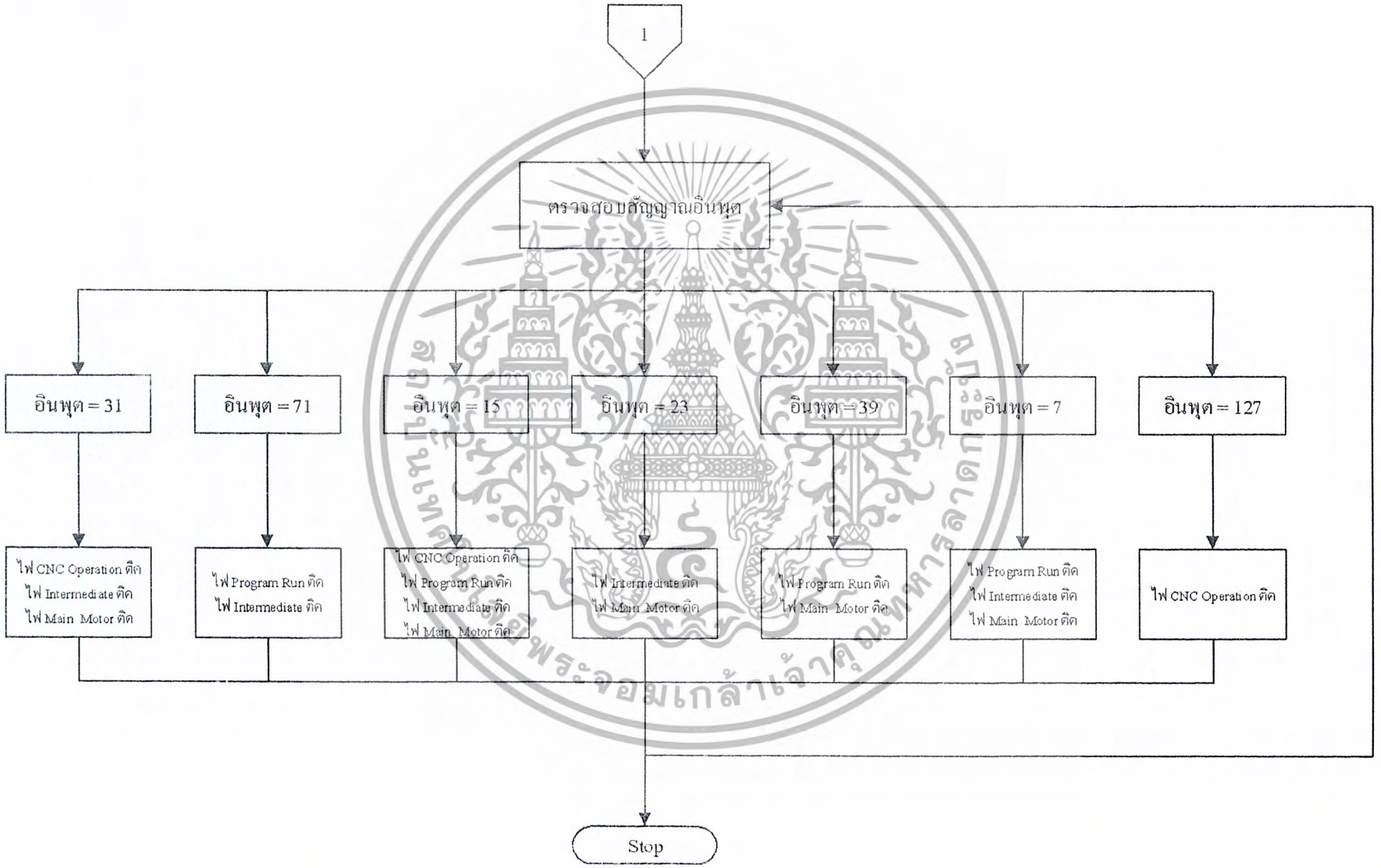
Flowchart แสดงการทำงานของการส่งค่าออกจากพอร์ตขนาน เพื่อควบคุม CNC-Compact 5



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

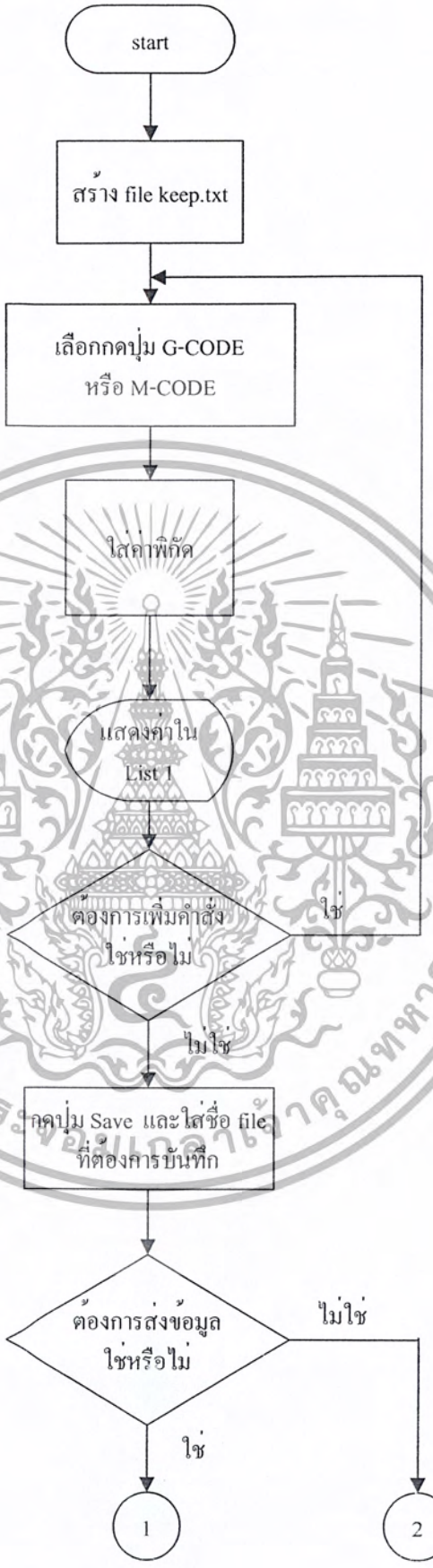
Flowchart แสดงการทำงานของการรับค่าสัญญาณอินพุตจากเครื่อง CNC-Compact 5



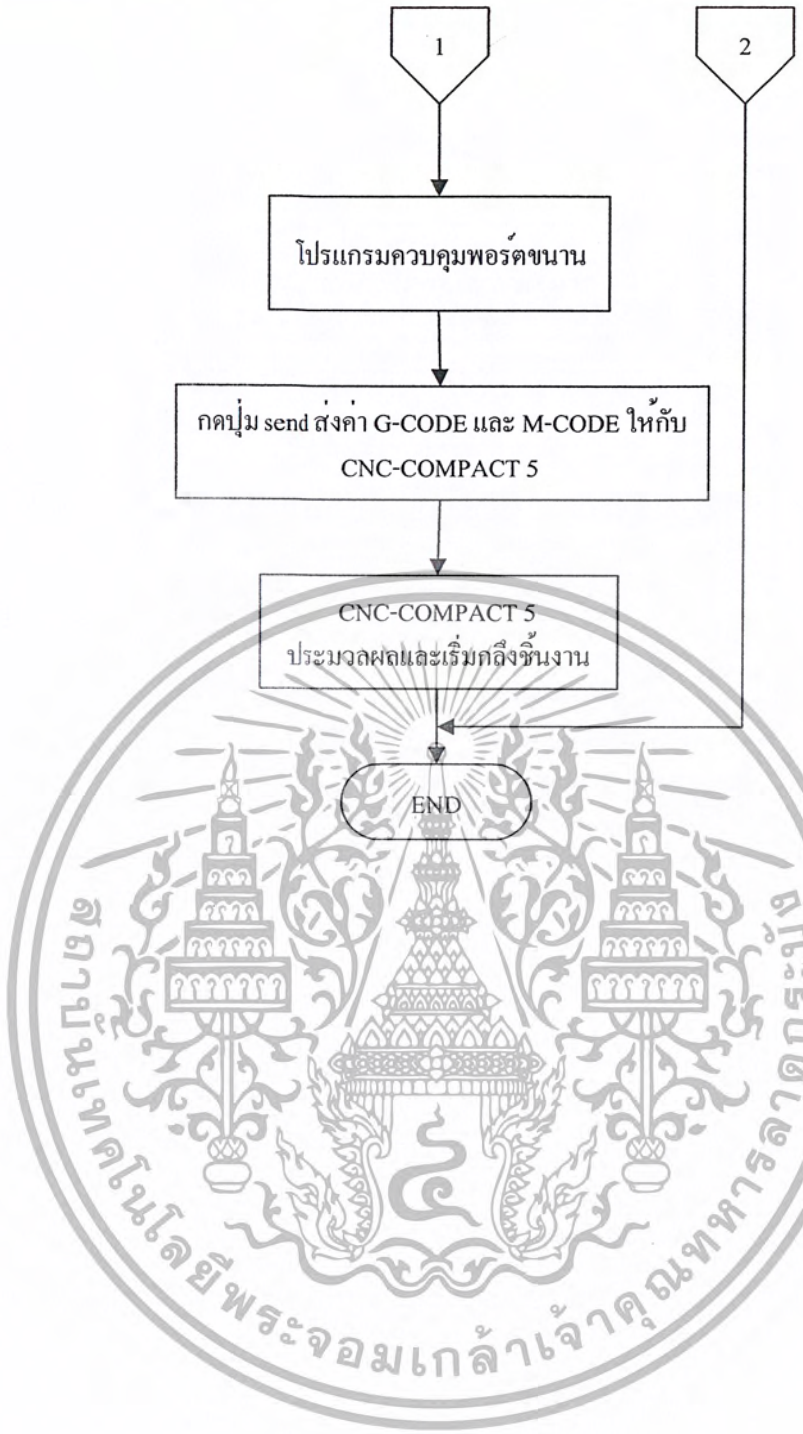

1
ตรวจสอบสัญญาณอินพุต
อินพุต = 31
อินพุต = 71
อินพุต = 15
อินพุต = 23
อินพุต = 39
อินพุต = 7
อินพุต = 127
ไฟ CNC Operation ติด
ไฟ Intermediate ติด
ไฟ Main Motor ติด
ไฟ Program Run ติด
ไฟ Intermediate ติด
ไฟ CNC Operation ติด
ไฟ Program Run ติด
ไฟ Intermediate ติด
ไฟ Main Motor ติด
ไฟ Intermediate ติด
ไฟ Main Motor ติด
ไฟ Program Run ติด
ไฟ Main Motor ติด
ไฟ Program Run ติด
ไฟ Intermediate ติด
ไฟ Main Motor ติด
ไฟ CNC Operation ติด
Stop

Flow chart แสดงกระบวนการส่งค่า G-CODE และ M-CODE

จาก Computer สู่ CNC - COMPACT 5

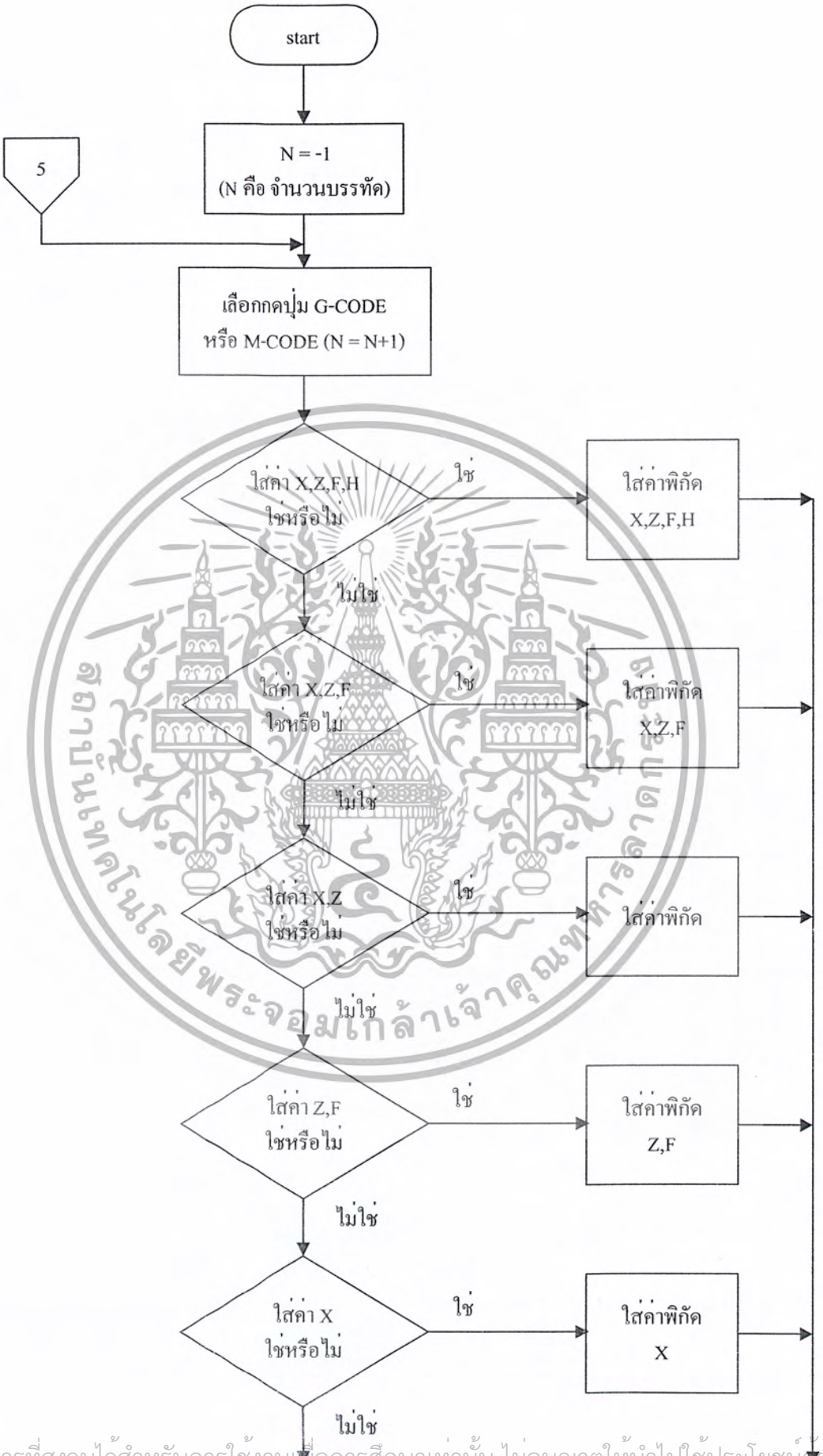


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

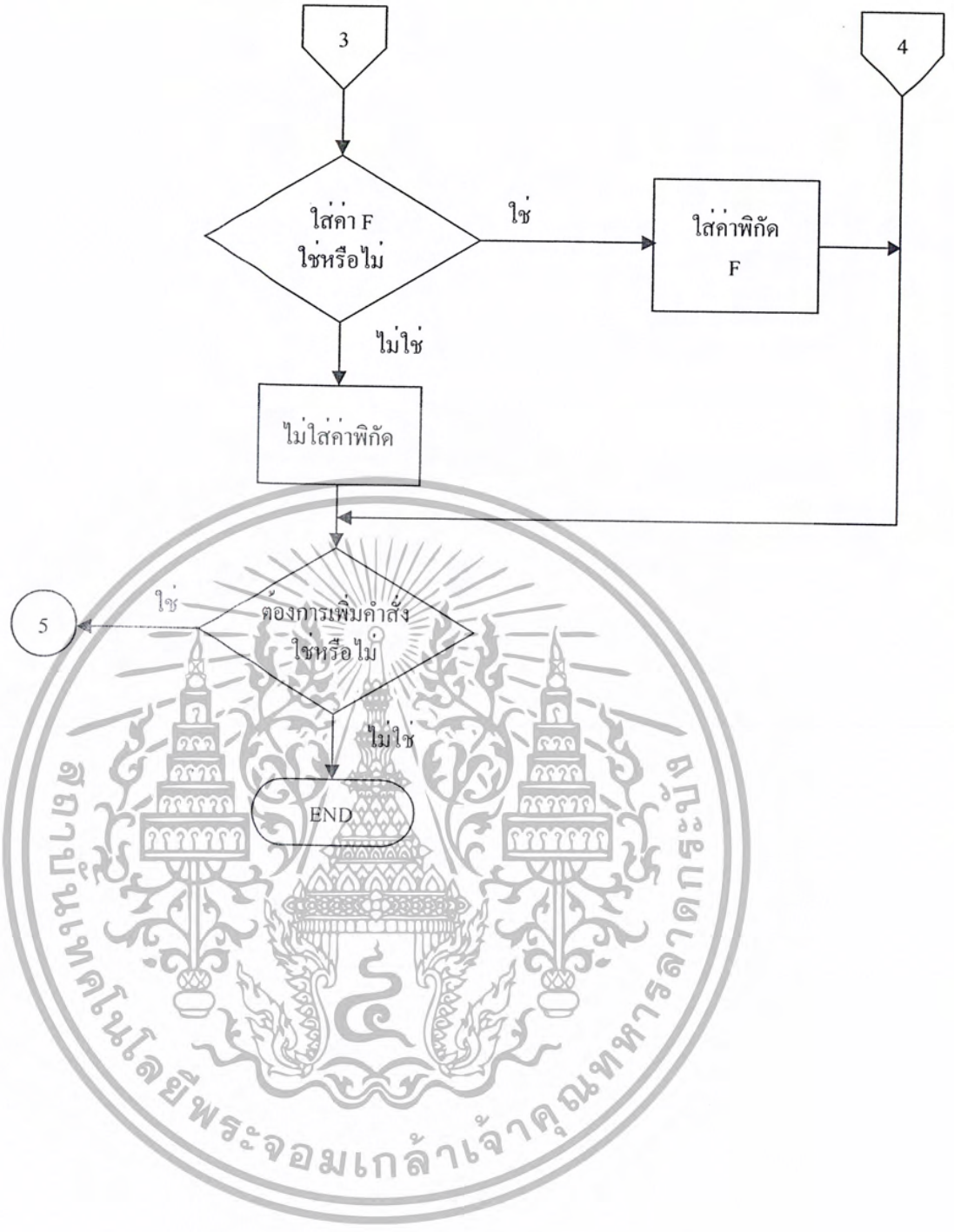


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flow chart แสดงกระบวนการใส่ค่าพิกัด

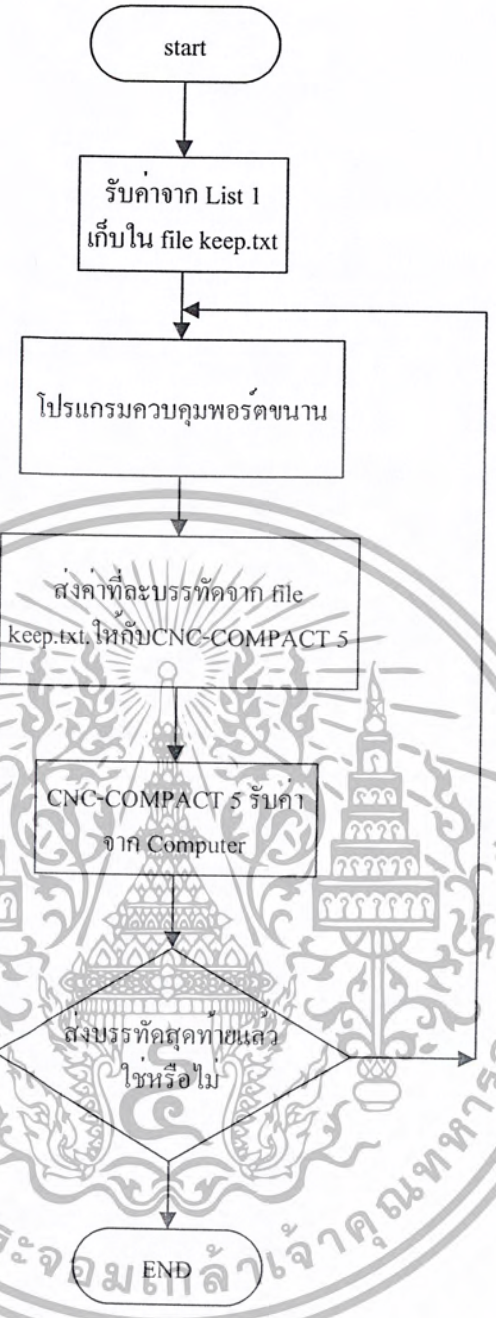


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flow chart แสดงกระบวนการส่งค่า ผ่านพอร์ตอนุกรม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการควบคุม CNC COMPACT 5 ทางพอร์ตอนุกรม

```
Public n, ifilenum As Integer
Public g, Data As Variant
Public senddata, send1, Data2 As Variant
Public longn As Integer
Private Sub Command1_Click()
  n = -1
  Dim fsobject As New FileSystemObject
  Dim del As File
  Set del = fsobject.GetFile("c:\keep.txt")
  del.Delete
  List1.Clear
  Dim myfso As New FileSystemObject
  Dim myfile As TextStream
  Set myfile = myfso.CreateTextFile("c:\keep.txt", True)
  ifilenum = FreeFile
End Sub
Private Sub Command2_Click()
  n = n - 1
  Dim i, j As Integer
  Dim kept, kept1, kept2 As String
    i = List1.ListIndex
  kept = Left(List1.List(i), 6)
  If List1.List(i + 1) = "" Then
    List1.RemoveItem (i)
  ElseIf List1.List(i + 1) <> "" Then
    List1.RemoveItem (i)
    Do
      kept2 = Mid(List1.List(i), 7)
      kept1 = Left(List1.List(i), 6)
      List1.AddItem kept + kept2, i
      List1.RemoveItem i + 1
      i = i + 1
      kept = kept1
    Loop Until List1.List(i) = ""
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    End If
End Sub
Private Sub Command3_Click()
    Dim i, j As Integer
    log1.ShowSave
    file1 = log1.FileName
      If file1 = "" Then
         Exit Sub
      End If
   i = 0
   j = 0
      Do
        If i = 0 Then
          Open file1 For Output As #ifilenum
        Else
          Open file1 For Append As #ifilenum
        End If
        Print #ifilenum, List1.List(i)
        Close #ifilenum
        i = i + 1
        Loop Until List1.List(i) = ""
      Do
        If j = 0 Then
          Open "c:\keep.txt" For Output As #ifilenum
        Else
          Open "c:\keep.txt" For Append As #ifilenum
        End If
        Print #ifilenum, List1.List(j)
        Close #ifilenum
        j = j + 1
      Loop Until List1.List(j) = ""
End Sub
Private Sub exit_Click()
   End
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub G00_Click()
    g = "00"
    Form4.Show
End Sub
Private Sub Form_Load()
    Dim myfso As New FileSystemObject
    n = -1
    Dim myfile As TextStream
    Set myfile = myfso.CreateTextFile("c:\keep.txt", True)
    ifilenum = FreeFile
    MSComm1.Settings = "300,e,7,1"
    MSComm1.CommPort = 1
    MSComm1.PortOpen = True
    MSComm1.InputLen = 0
    MSComm2.Settings = "300,e,7,1"
    MSComm2.CommPort = 2
    MSComm2.PortOpen = True
    MSComm2.InputLen = 0
End Sub
Private Sub G01_Click()
    g = "01"
    Form5.Show
End Sub
Private Sub G02_Click()
    g = "02"
    Form5.Show
End Sub
Private Sub G03_Click()
    g = "03"
    Form5.Show
End Sub
Private Sub G04_Click()
    g = "04"
    Form3.Show
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub G21_Click()
   n = n + 1
   g = 21
   If n < 10 Then
      ndata = "0" & Space(0) & LTrim(Str(n))
   Else: ndata = Str(n)
   End If
   longn = Len(Str(n))
   Data = Space(6 - longn) & ndata & Space(1) & g & _
   Space(21) & Chr(10)
   List1.AddItem Data
   Open "c:\keep.txt" For Append As #ifilenum
   Print #ifilenum, Data
   Close #ifilenum
End Sub
Private Sub G24_Click()
   n = n + 1
   g = 24
   If n < 10 Then
      ndata = "0" & Space(0) & LTrim(Str(n))
   Else: ndata = Str(n)
   End If
   longn = Len(Str(n))
   Data = Space(6 - longn) & ndata & Space(1) & g & _
   Space(21) & Chr(10)
   List1.AddItem Data
   Open "c:\keep.txt" For Append As #ifilenum
   Print #ifilenum, Data
   Close #ifilenum
End Sub
Private Sub G25_Click()
   g = "25"
   Form7.Show
End Sub
Private Sub G27_Click()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    g = "27"
    Form7.Show
End Sub
Private Sub G33_Click()
    g = "33"
    Form6.Show
End Sub
Private Sub G73_Click()
    g = "73"
    Form6.Show
End Sub
Private Sub G78_Click()
    g = "78"
    Form2.Show
End Sub
Private Sub G81_Click()
    g = "81"
    Form6.Show
End Sub
Private Sub G82_Click()
    g = "82"
    Form6.Show
End Sub
Private Sub G83_Click()
    g = "83"
    Form6.Show
End Sub
Private Sub G84_Click()
    g = "84"
    Form2.Show
End Sub
Private Sub G85_Click()
    g = "85"
    Form6.Show
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub G86_Click()
   g = "86"
   Form2.Show
End Sub
Private Sub G88_Click()
   g = "88"
   Form2.Show
End Sub
Private Sub G89_Click()
   g = "89"
   Form6.Show
End Sub
Private Sub G90_Click()
   n = n + 1
   g = "90"
   If n < 10 Then
      ndata = "0" & Space(0) & LTrim(Str(n))
   Else: ndata = Str(n)
   End If
   longn = Len(Str(n))
   Data = Space(6 - longn) & ndata & Space(1) & g & _
   Space(21) & Chr(10)
   List1.AddItem Data
   Open "c:\keep.txt" For Append As #ifilenum
   Print #ifilenum, Data
   Close #ifilenum
End Sub
Private Sub G91_Click()
   n = n + 1
   g = "91"
   If n < 10 Then
      ndata = "0" & Space(0) & LTrim(Str(n))
   Else: ndata = Str(n)
   End If
   longn = Len(Str(n))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Data = Space(6 - longn) & ndata & Space(1) & g & _
    Space(21) & Chr(10)
    List1.AddItem Data
    Open "c:\keep.txt" For Append As #ifilenum
    Print #ifilenum, Data
    Close #ifilenum
End Sub
Private Sub G92_Click()
    g = "92"
    Form4.Show
End Sub
Private Sub G94_Click()
    n = n + 1
    g = "94"
    If n < 10 Then
      ndata = "0" & Space(0) & LTrim(Str(n))
    Else: ndata = Str(n)
    End If
    longn = Len(Str(n))
    Data = Space(6 - longn) & ndata & Space(1) & g & _
    Space(21) & Chr(10)
    List1.AddItem Data
    Open "c:\keep.txt" For Append As #ifilenum
    Print #ifilenum, Data
    Close #ifilenum
End Sub
Private Sub G95_Click()
    n = n + 1
    g = "95"
    If n < 10 Then
      ndata = "0" & Space(0) & LTrim(Str(n))
    Eise: ndata = Str(n)
    End If
    longn = Len(Str(n))
    Data = Space(6 - longn) & ndata & Space(1) & g & _
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Space(21) & Chr(10)
    List1.AddItem Data
    Open "c:\keep.txt" For Append As #ifilenum
    Print #ifilenum, Data
    Close #ifilenum
End Sub
Private Sub M00_Click()
    n = n + 1
    If n < 10 Then
       ndata = "0" & Space(0) & LTrim(Str(n))
    Else: ndata = Str(n)
    End If
    longn = Len(Str(n))
    Data = Space(6 - longn) & ndata & "M00" & Space(21) _
    & Chr(10)
    List1.AddItem Data
    Open "c:\keep.txt" For Append As #ifilenum
    Print #ifilenum, Data
    Close #ifilenum
End Sub
Private Sub M03_Click()
    n = n + 1
    If n < 10 Then
       ndata = "0" & Space(0) & LTrim(Str(n))
    Else: ndata = Str(n)
    End If
    longn = Len(Str(n))
    Data = Space(6 - longn) & ndata & "M03" & Space(21) _
    & Chr(10)
    List1.AddItem Data
    Open "c:\keep.txt" For Append As #ifilenum
    Print #ifilenum, Data
    Close #ifilenum
End Sub
Private Sub M05_Click()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    n = n + 1
    If n < 10 Then
       ndata = "0" & Space(0) & LTrim(Str(n))
    Else: ndata = Str(n)
    End If
    longn = Len(Str(n))
    Data = Space(6 - longn) & ndata & "M05" & Space(21) _
    & Chr(10)
    List1.AddItem Data
    Open "c:\keep.txt" For Append As #ifilenum
    Print #ifilenum, Data
    Close #ifilenum
End Sub
Private Sub M06_Click()
    n = n - 1
    If n < 10 Then
       ndata = "0" & Space(0) & LTrim(Str(n))
    Else: ndata = Str(n)
    End If
    longn = Len(Str(n))
    Data = Space(6 - longn) & ndata & "M06" & Space(21) _
    & Chr(10)
    List1.AddItem Data
    Open "c:\keep.txt" For Append As #ifilenum
    Print #ifilenum, Data
    Close #ifilenum
    Form5.Show
End Sub
Private Sub M08_Click()
    n = n + 1
    If n < 10 Then
       ndata = "0" & Space(0) & LTrim(Str(n))
    Else: ndata = Str(n)
    End If
    longn = Len(Str(n))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Data = Space(6 - longn) & ndata & "M08" & Space(21) _
    & Chr(10)
    List1.AddItem Data
    Open "c:\keep.txt" For Append As #ifilenum
    Print #ifilenum, Data
    Close #ifilenum
End Sub
Private Sub M09_Click()
    n = n + 1
    If n < 10 Then
       ndata = "0" & Space(0) & LTrim(Str(n))
    Else: ndata = Str(n)
    End If
    longn = Len(Str(n))
    Data = Space(6 - longn) & ndata & "M09" & Space(21) _
    & Chr(10)
    List1.AddItem Data
    Open "c:\keep.txt" For Append As #ifilenum
    Print #ifilenum, Data
    Close #ifilenum
End Sub
Private Sub M17_Click()
    n = n + 1
    If n < 10 Then
       ndata = "0" & Space(0) & LTrim(Str(n))
    Else: ndata = Str(n)
    End If
    longn = Len(Str(n))
    Data = Space(6 - longn) & ndata & "M17" & Space(21) _
    & Chr(10)
    List1.AddItem Data
    Open "c:\keep.txt" For Append As #ifilenum
    Print #ifilenum, Data
    Close #ifilenum
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub M22_Click()
   n = n + 1
   If n < 10 Then
      ndata = "0" & Space(0) & LTrim(Str(n))
   Else: ndata = Str(n)
   End If
   longn = Len(Str(n))
   Data = Space(6 - longn) & ndata & "M22" & Space(21) _
   & Chr(10)
   List1.AddItem Data
   Open "c:\keep.txt" For Append As #ifilenum
   Print #ifilenum, Data
   Close #ifilenum
End Sub
Private Sub M23_Click()
   n = n + 1
   If n < 10 Then
      ndata = "0" & Space(0) & LTrim(Str(n))
   Else: ndata = Str(n)
   End If
   longn = Len(Str(n))
   Data = Space(6 - longn) & ndata & "M23" & Space(21) _
   & Chr(10)
   List1.AddItem Data
   Open "c:\keep.txt" For Append As #ifilenum
   Print #ifilenum, Data
   Close #ifilenum
End Sub
Private Sub M30_Click()
   n = n + 1
   If n < 10 Then
      ndata = "0" & Space(0) & LTrim(Str(n))
   Else: ndata = Str(n)
   End If
   longn = Len(Str(n))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Data = Space(6 - longn) & ndata & "M30" & Space(21) _
    & Chr(10)
    List1.AddItem Data
    Open "c:\keep.txt" For Append As #ifilenum
    Print #ifilenum, Data
    Close #ifilenum
End Sub
Private Sub M98_Click()
    g = "M98"
    Form4.Show
End Sub

Private Sub M99_Click()
    g = "M99"
    Form4.Show
End Sub
Private Sub Menuopen_Click()
    Dim i As Integer
    log2.ShowOpen
    file1 = log2.FileName
      If file1 = "" Then
        Exit Sub
      End If
    i = 0
    Open file1 For Input As #ifilenum
        Do
          Line Input #ifilenum, senddata
          List1.AddItem senddata
        Loop Until EOF(ifilenum)
        Close #ifilenum
        Do
          If i = 0 Then
            Open "c:\keep.txt" For Output As #ifilenum
          Else
            Open "c:\keep.txt" For Append As #ifilenum
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        End If
        Print #ifilenum, List1.List(i)
        Close #ifilenum
        i = i + 1
        Loop Until List1.List(i) = ""
End Sub
Private Sub saveas_Click()
  Dim i, j As Integer
  log1.ShowSave
  file1 = log1.FileName
    If file1 = "" Then
      Exit Sub
    End If
  i = 0
  j = 0
    Do
     If i = 0 Then
       Open file1 For Output As #ifilenum
     Else
       Open file1 For Append As #ifilenum
     End If
    Print #ifilenum, List1.List(i)
    Close #ifilenum
    i = i + 1
    Loop Until List1.List(i) = ""
    Do
      If j = 0 Then
        Open "c:\keep.txt" For Output As #ifilenum
      Else
        Open "c:\keep.txt" For Append As #ifilenum
      End If
    Print #ifilenum, List1.List(j)
    Close #ifilenum
    j = j + 1
    Loop Until List1.List(j) = ""
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
Private Sub send_Click()
   Dim i As Integer
   MSComm1.Output = "%" & Chr(13) & Chr(10) & Space(4) & _
   "N" & Chr(96) & Space(1) & "G" & Chr(96) & Space(3) & _
   "X" & Space(1) & Chr(96) & Space(4) & "Z" & Space(1) & _
   Chr(96) & Space(2) & "F" & Chr(96) & Space(2) & "H" & _
   Space(1) & Chr(13) & Chr(10)
   send1 = MSComm2.Input
   List2.AddItem send1
   Open "c:\keep.txt" For Input As #ifilenum
      Do
         Line Input #ifilenum, senddata
         MSComm1.Output = senddata
         send1 = MSComm2.Input
         List2.AddItem send1
         If EOF(ifilenum) Then
            MSComm1.Output = "M" & Chr(13) & Chr(10)
            send1 = MSComm2.Input
            List2.AddItem send1
         End If
      Loop Until EOF(ifilenum)
      Close #ifilenum
End Sub
```

โปรแกรมควบคุม CNC COMPACT 5 ทางพอร์ตขนาน

```
Private Sub fillmotor()
   Shape4.FillStyle = 0
   Shape4.FillColor = RGB(255, 0, 0)
End Sub
Private Sub fillinterstop()
   Shape3.FillStyle = 0
   Shape3.FillColor = RGB(255, 0, 0)
End Sub
Private Sub fillrun()
   Timer2.Enabled = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
Private Sub Timer2_Timer()
    Shape2.FillStyle = 0
    Shape2.FillColor = RGB(255, 0, 0)
End Sub
Private Sub fillcnc()
    Shape1.FillStyle = 0
    Shape1.FillColor = RGB(255, 0, 0)
End Sub
Private Sub cmdhand_Click()
    Out &H378, &H8
    Call delay
    Out &H378, &H0
End Sub
Private Sub cmdinp_Click()
    Out &H378, &H2
    Call delay
    Out &H378, &H0
End Sub
Private Sub cmdstart_Click()
    Out &H378, &H10
    Call delay
    Out &H378, &H0
End Sub
Private Sub delay()
    Dim i As Variant
      For i = 1 To 100000
        i = i + 1
      Next i
End Sub
Private Sub Timer1_Timer()
    Shape1.FillStyle = 1
    Shape2.FillStyle = 1
    Shape3.FillStyle = 1
    Shape4.FillStyle = 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Data2 = Inp(&H379) Or &H7
Text1.Text = Data2
Select Case Data2
  Case Is = 119
    Timer2.Enabled = False
  Case Is = 111
    Call fillcnc
    Call fillrun
  Case Is = 95
    Timer2.Enabled = False
    Call fillcnc
    Call fillinterstop
  Case Is = 63
    Timer2.Enabled = False
    Call fillcnc
    Call fillmotor
  Case Is = 103
    Call fillrun
  Case Is = 87
    Timer2.Enabled = False
    Call fillinterstop
  Case Is = 55
    Timer2.Enabled = False
    Call fillmotor
  Case Is = 79
    Call fillcnc
    Call fillrun
    Call fillinterstop
  Case Is = 47
    Call fillcnc
    Call fillrun
    Call fillmotor
  Case Is = 31
    Timer2.Enabled = False
    Call fillcnc
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Call fillinterstop
            Call fillmotor
        Case Is = 71
            Call fillrun
            Call fillinterstop
        Case Is = 15
            Call fillcnc
            Call fillrun
            Call fillinterstop
            Call fillmotor
        Case Is = 23
            Timer2.Enabled = False
            Call fillinterstop
            Call fillmotor
        Case Is = 39
            Call fillrun
            Call fillmotor
        Case Is = 7
            Call fillrun
            Call fillinterstop
            Call fillmotor
        Case Is = 127
            Timer2.Enabled = False
            Call fillcnc
    End Select
End Sub
```

โปรแกรมการใส่ค่าพิกัด

```
Private Sub Command1_Click()
    Dim x, x1, z, z1, xdata, zdata, fdata, hdata, a, b As String
    Form1.n = Form1.n + 1
      If Form1.n < 10 Then
         Form1.Data = "0" & LTrim(Str(Form1.n))
      Else: Form1.Data = Str(Form1.n)
      End If
    Form1.longn = Len(Str(Form1.n))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
xdata = Len(Text1.Text)
zdata = Len(Text2.Text)
fdata = Len(Text3.Text)
hdata = Len(Text4.Text)
x = Str(xdata - 1)
x1 = Right(Text1.Text, x)
z = Str(zdata - 1)
z1 = Right(Text2.Text, z)
a = Left(Text1.Text, 1)
b = Left(Text2.Text, 1)
If a = "-" And b = "-" Then '-x.-z
   Form1.Data = Space(6 - Form1.longn) & Form1.Data _
   & Space(1) & Form1.g & Space(1) & "-" & Space _
   (4 - x) & x1 & Space(1) & "-" & Space(5 - z) & _
   z1 & Space(4 - fdata) & Text3.Text & Space _
   (4 - hdata) & Text4.Text & Chr(10)
ElseIf a = "-" And b <> "-" Then '-x,+z
   Form1.Data = Space(6 - Form1.longn) & Form1.Data _
   & Space(1) & Form1.g & Space(1) & "-" & Space(4 - x) _
   & x1 & Space(1) & Space(7 - zdata) & Text2.Text & _
   Space(4 - fdata) & Text3.Text & Space(4 - hdata) & _
   Text4.Text & Chr(10)
ElseIf a <> "-" And b = "-" Then '+x,-z
   Form1.Data = Space(6 - Form1.longn) & Form1.Data _
   & Space(1) & Form1.g & Space(6 - xdata) & Text1.Text _
   & Space(1) & "-" & Space(5 - z) & z1 & Space(4 - fdata) _
   & Text3.Text & Space(4 - hdata) & Text4.Text & Chr(10)
Else '+x,+z
   Form1.Data = Space(6 - Form1.longn) & Form1.Data _
   & Space(1) & Form1.g & Space(6 - xdata) & Text1.Text _
   & Space(7 - zdata) & Text2.Text & Space(4 - fdata) & _
   Text3.Text & Space(4 - hdata) & Text4.Text & Chr(10)
End If
Form1.List1.AddItem Form1.Data
Open "c:\keep.txt" For Append As #Form1.ifilenum
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Print #Form1.ifilenum, Form1.Data
    Close #Form1.ifilenum
    Unload Me
End Sub
Private Sub Command1_Click()
    Dim x, x1, xdata, a As String
    Form1.n = Form1.n + 1
      If Form1.n < 10 Then
        Form1.Data = "0" & LTrim(Str(Form1.n))
      Else: Form1.Data = Str(Form1.n)
      End If
    Form1.longn = Len(Str(Form1.n))
    xdata = Len(Text1.Text)
    x = Str(xdata - 1)
    x1 = Right(Text1.Text, x)
    a = Left(Text1.Text, 1)
     If a = "-" Then '-x
       Form1.Data = Space(6 - Form1.longn) & Form1.Data _
       & Space(1) & Form1.g & Space(1) & "-" & Space(4 - x) _
       & x1 & Space(15) & Chr(10)
      Else '+x
        Form1.Data = Space(6 - Form1.longn) & Form1.Data _
        & Space(1) & Form1.g & Space(6 - xdata) & Text1.Text _
        & Space(15) & Chr(10)
      End If
    Form1.List1.AddItem Form1.Data
    Open "c:\keep.txt" For Append As #Form1.ifilenum
    Print #Form1.ifilenum, Form1.Data
    Close #Form1.ifilenum
    Unload Me
End Sub
Private Sub Command1_Click()
    Dim x, x1, z, z1, xdata, zdata, a, b As String
    Form1.n = Form1.n + 1
      If Form1.n < 10 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Form1.Data = "0" & LTrim(Str(Form1.n))
    Else: Form1.Data = Str(Form1.n)
    End If
Form1.longn = Len(Str(Form1.n))
xdata = Len(Text1.Text)
zdata = Len(Text2.Text)
x = Str(xdata - 1)
x1 = Right(Text1.Text, x)
z = Str(zdata - 1)
z1 = Right(Text2.Text, z)
a = Left(Text1.Text, 1)
b = Left(Text2.Text, 1)
 If a = "-" And b = "-" Then '-x,-z
    Form1.Data = Space(6 - Form1.longn) & Form1.Data & _
    Space(1) & Form1.g & Space(1) & "-" & Space(4 - x) _
    & x1 & Space(1) & "-" & Space(5 - z) & z1 & Space(8) _
    & Chr(10)
 ElseIf a = "-" And b <> "-" Then '-x,+z
    Form1.Data = Space(6 - Form1.longn) & Form1.Data & _
    Space(1) & Form1.g & Space(1) & "-" & Space(4 - x) _
    & x1 & Space(1) & Space(7 - zdata) & Text2.Text & _
    Space(8) & Chr(10)
 ElseIf a <> "-" And b = "-" Then '+x,-z
    Form1.Data = Space(6 - Form1.longn) & Form1.Data & _
    Space(1) & Form1.g & Space(6 - xdata) & Text1.Text & _
    Space(1) & "-" & Space(5 - z) & z1 & Space(8) & Chr(10)
 Else '+x,+z
    Form1.Data = Space(6 - Form1.longn) & Form1.Data & _
    Space(1) & Form1.g & Space(6 - xdata) & Text1.Text & _
    Space(7 - zdata) & Text2.Text & Space(8) & Chr(10)
 End If
Form1.List1.AddItem Form1.Data
Open "c:\keep.txt" For Append As #Form1.ifilenum
Print #Form1.ifilenum, Form1.Data
Close #Form1.ifilenum
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Unload Me
End Sub
Private Sub Command1_Click()
    Dim x, x1, z, z1, xdata, zdata, fdata, a, b As String
    Form1.n = Form1.n + 1
      If Form1.n < 10 Then
        Form1.Data = "0" & LTrim(Str(Form1.n))
      Else: Form1.Data = Str(Form1.n)
      End If
    Form1.longn = Len(Str(Form1.n))
    xdata = Len(Text1.Text)
    zdata = Len(Text2.Text)
    fdata = Len(Text3.Text)
    x = Str(xdata - 1)
    x1 = Right(Text1.Text, x)
    z = Str(zdata - 1)
    z1 = Right(Text2.Text, z)
    a = Left(Text1.Text, 1)
    b = Left(Text2.Text, 1)
      If a = "-" And b = "-" Then '-x,-z
        Form1.Data = Space(6 - Form1.longn) & Form1.Data & _
        Space(1) & Form1.g & Space(1) & "-" & Space(4 - x) & _
        x1 & Space(1) & "-" & Space(5 - z) & z1 & Space(4 - fdata) _
        & Text3.Text & Space(4) & Chr(10)
      ElseIf a = "-" And b <> "-" Then '-x,+z
        Form1.Data = Space(6 - Form1.longn) & Form1.Data & _
        Space(1) & Form1.g & Space(1) & "-" & Space(4 - x) & _
        x1 & Space(1) & Space(7 - zdata) & Text2.Text & _
        Space(4 - fdata) & Text3.Text & Space(4) & Chr(10)
      ElseIf a <> "-" And b = "-" Then '+x,-z
        Form1.Data = Space(6 - Form1.longn) & Form1.Data & _
        Space(1) & Form1.g & Space(6 - xdata) & Text1.Text & _
        Space(1) & "-" & Space(5 - z) & z1 & Space(4 - fdata) & _
        Text3.Text & Space(4) & Chr(10)
      Else '+x,+z
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Form1.Data = Space(6 - Form1.longn) & Form1.Data & _
            Space(1) & Form1.g & Space(6 - xdata) & Text1.Text & _
            Space(7 - zdata) & Text2.Text & Space(4 - fdata) & _
            Text3.Text & Space(4) & Chr(10)
        End If
    Form1.List1.AddItem Form1.Data
    Open "c:\keep.txt" For Append As #Form1.ifilenum
    Print #Form1.ifilenum, Form1.Data
    Close #Form1.ifilenum
    Unload Me
End Sub
Private Sub Command1_Click()
    Dim z, z1, zdata, fdata, b As String
    Form1.n = Form1.n + 1
      If Form1.n < 10 Then
        Form1.Data = "0" & LTrim(Str(Form1.n))
      Else: Form1.Data = Str(Form1.n)
      End If
    Form1.longn = Len(Str(Form1.n))
    zdata = Len(Text1.Text)
    fdata = Len(Text2.Text)
    z = Str(zdata - 1)
    z1 = Right(Text1.Text, z)
    b = Left(Text1.Text, 1)
     If b = "-" Then '-z
        Form1.Data = Space(6 - Form1.longn) & Form1.Data & _
        Space(1) & Form1.g & Space(7) & "-" & Space(5 - z) & _
        z1 & Space(1) & Space(4 - fdata) & Text2.Text & Space(4) _
        & Chr(10)
      Else '+z
        Form1.Data = Space(6 - Form1.longn) & Form1.Data & _
        Space(1) & Form1.g & Space(7) & Space(6 - zdata) & _
        Text1.Text & Space(4 - fdata) & Text2.Text & Space(4) & _
        Chr(10)
      End If
```

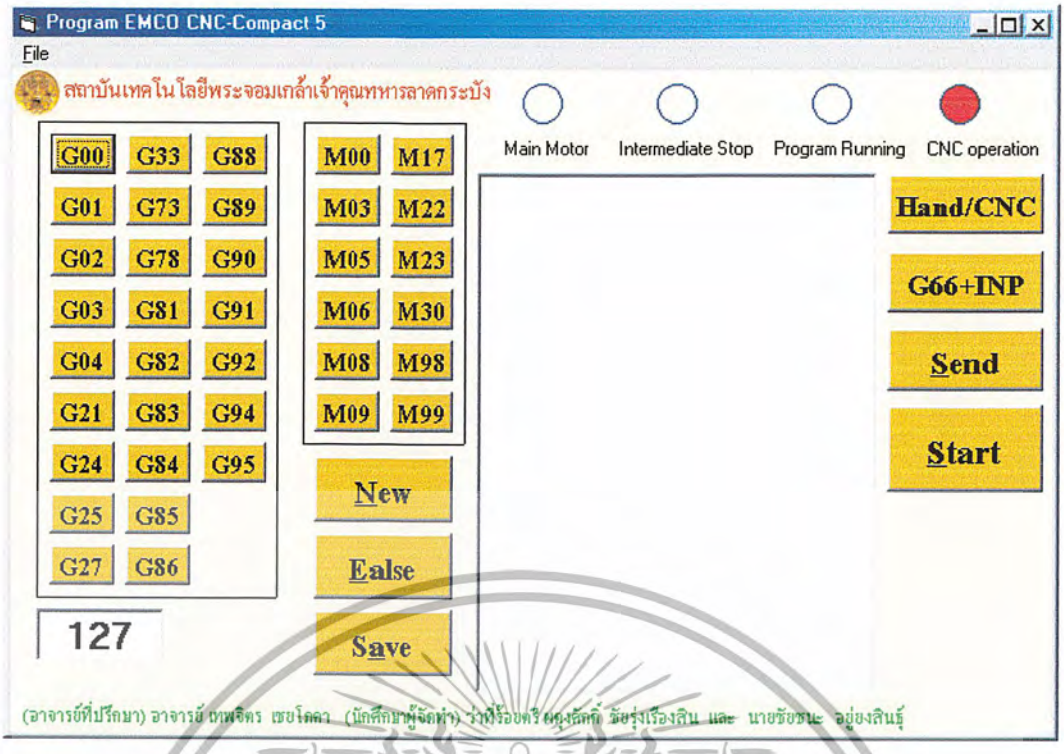
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Form1.List1.AddItem Form1.Data
    Open "c:\keep.txt" For Append As #Form1.ifilenum
    Print #Form1.ifilenum, Form1.Data
    Close #Form1.ifilenum
    Unload Me
End Sub
Private Sub Command1_Click()
    Dim fdata As String
    Form1.n = Form1.n + 1
      If Form1.n < 10 Then
        Form1.Data = "0" & LTrim(Str(Form1.n))
      Else: Form1.Data = Str(Form1.n)
      End If
    Form1.longn = Len(Str(Form1.n))
    fdata = Len(Text1.Text)
    Form1.Data = Space(6 - Form1.longn) & Form1.Data & Space(1) _
    & Form1.g & Space(13) & "L" & Space(3 - fdata) & Text1.Text _
    & Space(4) & Chr(10)
    Form1.List1.AddItem Form1.Data
    Open "c:\keep.txt" For Append As #Form1.ifilenum
    Print #Form1.ifilenum, Form1.Data
    Close #Form1.ifilenum
    Unload Me
End Sub
#If Win32 Then
    Public Declare Sub Out Lib "io.dll" Alias "PortOut" _
    (ByVal Port As Integer, ByVal Data As Byte)
    Public Declare Function Inp Lib "io.dll" Alias "PortIn" _
    (ByVal Port As Integer) As Byte
#Else
    Declare Function Inp Lib "InpOut.DLL" (ByVal Port As Integer) _
    As Byte
    Declare Sub Out Lib "InpOut.DLL" (ByVal Port As Integer, ByVal _
    Value As Byte)
#End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

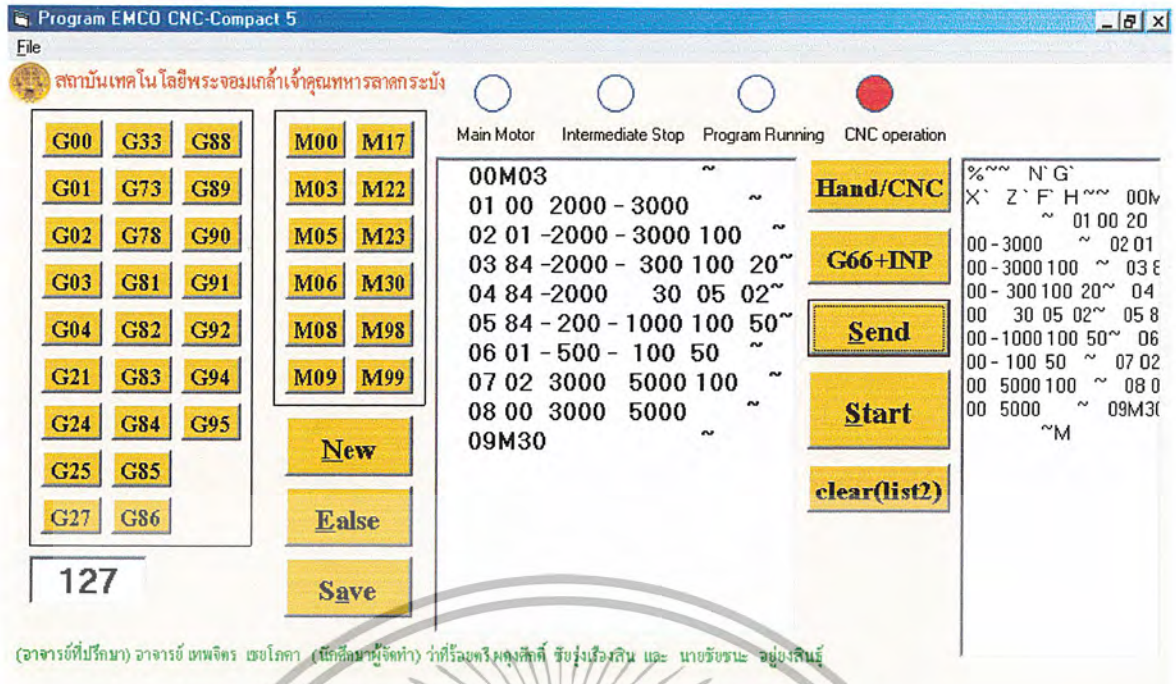
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพแสดงโปรแกรมการติดต่อ CNC-COMPACT 5 และ COMPUTER

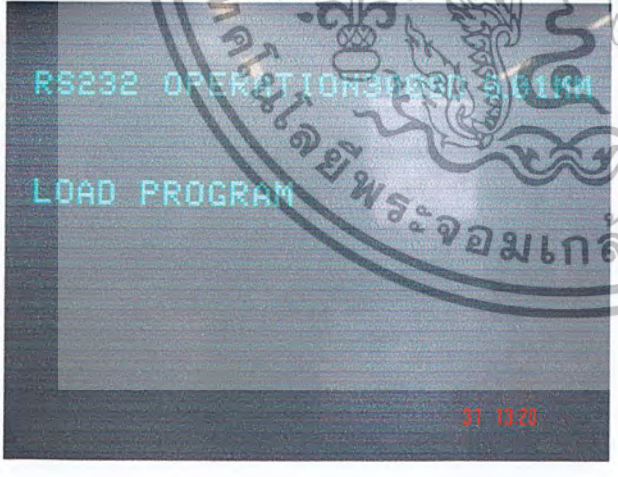
รูปภาพแสดงการเปลี่ยนจาก HAND OPERATION เป็น CNC OPERATION (เมื่อกดปุ่ม Hand/CNC)

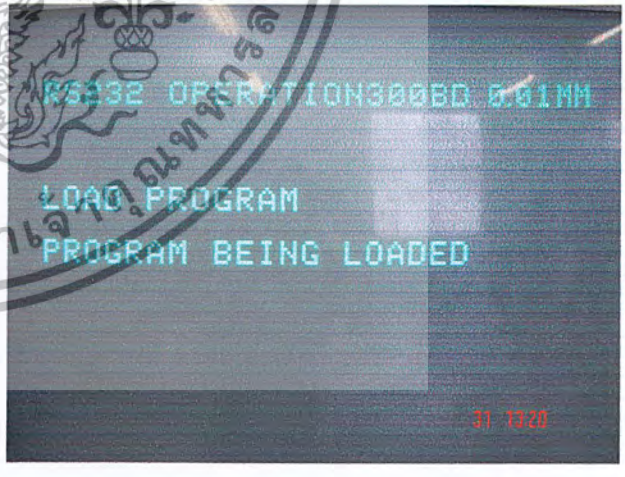
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(อาจารย์ที่ปรึกษา) อาจารย์ เทพจิตร เชยโภคา (นักศึกษาผู้จัดทำ) ว่าที่ร้อยตรี ผดุงศักดิ์ ชัยรุ่งเรืองสิน และ นายชัยชนะ อยู่ยงสินธุ์

รูปภาพแสดงการรับ-ส่ง ข้อมูล ของ COM1 และ COM2





รูปที่ 1 รูปที่ 2

รูปภาพแสดงการสถานะของเครื่อง CNC-COMPACT 5

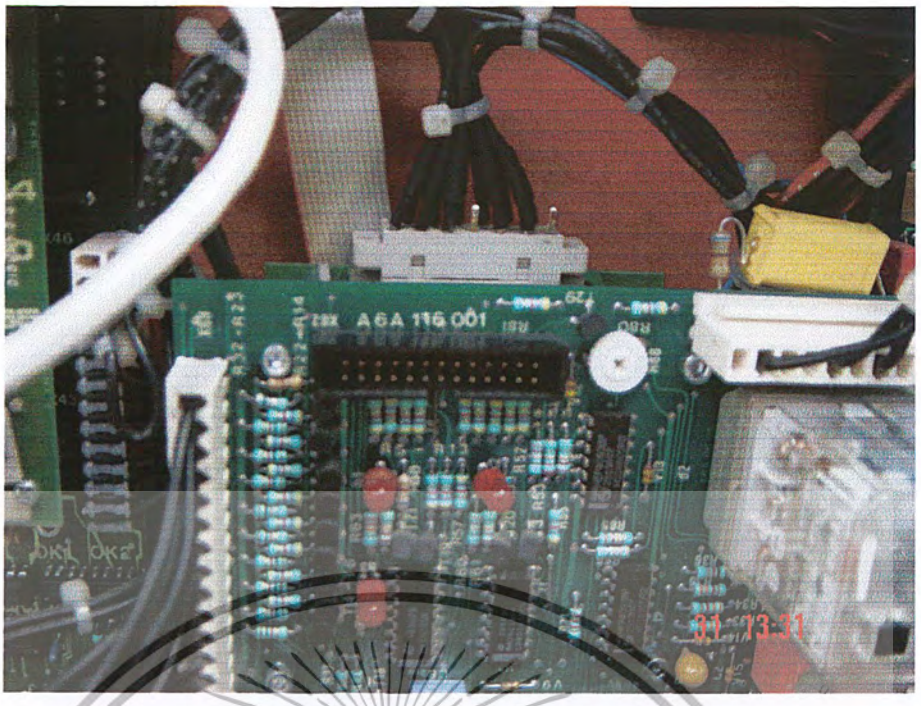
- รูปที่ 1 คือ รูปสถานะการเปิดพอร์ตรับข้อมูลของเครื่อง CNC-COMPACT 5 เมื่อกดปุ่ม G66+INP
- รูปที่ 2 คือ รูปสถานะการเริ่มรับข้อมูลของเครื่อง CNC-COMPACT 5 เมื่อกดปุ่ม Send

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปภาพแสดงการรับข้อมูลจากโปรแกรมเข้าสู่กระบวนการประมวลผล ของ CNC-COMPACT 5

รูปภาพโครงงาน MICRO-CIM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปภาพพอร์ต DNC ที่ใช้ในการติดต่อกับพอร์ตขนาน

รูปภาพพอร์ตอนุกรมที่ใช้ในการส่งค่า G-CODE และ M-CODE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. Eshed Robotec (1982) Ltd., " SCORBOT-3 User's Manual" Eshed Robotec (1982)
2. ฉัททวุฒิ พืชผล. **คู่มือเรียน Visual Basic 6.** พิมพ์ครั้งที่ 5. กรุงเทพ : โปรวิชั่น, 2542
3. กิตติ ภักดีวัฒนะกุล. **Visual Basic 6.**
4. สมศักดิ์ ศรีขจรเกียรติ. **Advanced Visual Basic 6 Teach Yourself.** กรุงเทพ : บิบลิโอไฟล์, 2544. 560 หน้า
5. อรรถพล บุญยะโภคาและชัยวัฒน์ ลิ้มพรจิตรวิไล. **เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตขนาน.** กรุงเทพ : อินโนเวตีฟ เอ็กเพอริเมนต์.
6. กฤษดา ใจเย็น. **เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม.** กรุงเทพ : อินโนเวตีฟ เอ็กเพอริเมนต์.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณ บิดา – มารดา ผู้ที่ซึ่งให้การช่วยเหลือเกื้อกูลในทุกสิ่งทุกอย่าง

ขอขอบพระคุณ อ.เทพจิตร์ เชยโภคา ที่ให้คำปรึกษาและให้ความช่วยเหลือในการทำงานต่างๆ โดยดีมาตลอด

ขอขอบคุณท่านอาจารย์ทุกท่านที่ถ่ายทอดความรู้และสั่งสอนจนมาถึงทุกวันนี้

ขอขอบคุณ ผู้ที่เป็นเพื่อนทุกคนที่ให้การสนับสนุนและให้คำปรึกษาในทุก ๆ ด้าน ตลอดจนผู้ที่เป็นที่รักและผู้ที่รัก

ส่วนดีของปริญญานิพนธ์เล่มนี้

ขอมอบแด่

ผู้มีพระคุณ มีคุณธรรม

และ

ผู้กระทำเพื่อก่อประโยชน์แก่ส่วนรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้