โปรแกรมแปลง G-CODE เป็นระบบพิกัด X-Y-Z

G-CODE TO X-Y-Z COORDINATE TRANSLATION PROGRAM

นายสิทธิชัย มุนเนียม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โปรแกรมแปลง G-CODE เป็นระบบพิกัด X-Y-Z

G-CODE TO X-Y-Z COORDINATE TRANSLATION PROGRAM

นายสิทธิชัย มุนเนียม

T050253

สจ.
ส722ป
2545

b00166385

เลขหมู่.....
เลขทะเบียน 50253
วัน,เดือน,ปี 28 เม.ย. 2547

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b11376065

G-CODE TO X-Y-Z COORDINATE TRANSLATION PROGRAM

SITTICHAI MUNNIUM

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BECHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

DEPARTMENT OF INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์	โปรแกรมแปลง G-Code เป็นระบบพิกัด X-Y-Z
	G-CODE TO X-Y-Z COORDINATE TRANSLATION PROGRAM
นักศึกษาผู้จัดทำ	นายสิทธิชัย มุนเนียม รหัสประจำตัว 43015490
ปริญญา	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมการวัดคุม
ปีการศึกษา	2545

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
ผศ. ประภาษ อุคคกิมาพันธุ์	

วัน/เดือน/ปี ที่สอบ วันอังคารที่ 25 มีนาคม พ.ศ. 2546

สถานที่สอบ ณ ห้องสอบปริญญานิพนธ์ ภาควิชาวิศวกรรมการวัดคุม

ภาควิชารับรองแล้ว

(ผศ. ประสิทธิ์ จุลเสรีวงศ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	โปรแกรมแปลงรหัส G-Code เป็นระบบพิกัด X-Y-Z
	G-CODE TO X-Y-Z COORDINATE TRANSLATION PROGRAM
นักศึกษาผู้จัดทำ	นายสิทธิชัย มุนเนียม
อาจารย์ที่ปรึกษา	ผศ. ประภาษ อุดคกิมาพันธุ์
ปีการศึกษา	2545

บทคัดย่อ

การประยุกต์ใช้ระบบคอมพิวเตอร์ช่วยการผลิต (CAM) เพื่องานอุตสาหกรรมมีบทบาทอย่างมากในงานอุตสาหกรรมผลิตชิ้นส่วนจักรกลและผลิตภัณฑ์ โดยระบบจะถูกออกแบบให้เชื่อมต่อกับเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ โดยมีชุดคำสั่ง G-CODE เป็นสื่อภายในระบบ

ปริญญานิพนธ์นี้เสนอโครงงานออกแบบสร้างโปรแกรมแปลงชุดคำสั่ง G-CODE ให้อยู่ในชุดคำสั่งแบบพิกัด X-Y-Z ที่สามารถนำไปใช้ได้กับเครื่องเขียนภาพแบบแกนประสาน X-Y หรือเครื่องควบคุมการเคลื่อนที่แบบ 3 แกนประสาน X-Y-Z ซึ่งสามารถประยุกต์ใช้ใน งานตัด สลัก หรือกระทั่ง งานเจาะ ได้เทียบเคียงเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	G-Code to X-Y-Z Coordinate Translation Program
Author	Mr.Sittichai Munnium
Thesis Advisor	Asst.Prof. Prapart Ukakimaparn
Year	2003

ABSTRACT

Applications of Computer Aid Manufacturing system (CAM) is a part of mechanical product industrial work that was designed to connected with Computerized Numerical Control machine (CNC) and communicate by G-CODE command set.

This thesis presents the designed and creates of G-CODE to X-Y-Z coordinate translation program that can translate G-CODE command set to X-Y-Z coordinate and send solution into X-Y plotter or 3-axis motion controller for cutting , carving or drilling work similar to work with Computerized Numerical Control Machine.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีเพราะได้รับความเมตตาจาก ผู้ช่วยศาสตราจารย์ประภาษ อุคคกิมาพันธุ์ ที่ได้ให้คำแนะนำแก่ผู้วิจัยตลอดมา อีกทั้งยังเอื้อเฟื้ออุปกรณ์และเครื่องมือต่างๆในการทำปริญญานิพนธ์นี้ ผู้วิจัยรู้สึกซาบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณ คุณกฤษณ์ เสมอพิทักษ์ ที่ให้คำแนะนำอันเป็นประโยชน์ต่อการทำวิจัยและการทำปริญญานิพนธ์ฉบับนี้

และที่ลืมเสียมิได้คือ ขอกราบขอบพระคุณคุณพ่อและคุณแม่ อันเป็นที่รักยิ่ง ที่สนับสนุนและเป็นกำลังใจในการทำปริญญานิพนธ์ฉบับนี้

คุณค่าและประโยชน์อันพึงมีจากปริญญานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VII

สารบัญภาพ.....IX

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและเหตุจูงใจของการวิจัย.....1

1.2 วัตถุประสงค์ของปริญญานิพนธ์.....2

1.3 ขอบเขตของปริญญานิพนธ์.....2

1.4 ขั้นตอนการศึกษา.....2

บทที่ 2 ความรู้เบื้องต้นเกี่ยวกับเครื่องซีเอ็นซี.....4

2.1 กล่าวนำ.....4

2.2 ส่วนประกอบของเครื่องจักรซีเอ็นซีโดยทั่วไป.....5

2.2.1 ส่วนเครื่องจักรที่ใช้ผลิตชิ้นงาน.....6

2.2.2 ส่วนควบคุม.....9

2.2.3 ส่วนโปรแกรมสั่งงานเครื่องจักร.....9

2.3 การทำงานของเครื่องซีเอ็นซี.....11

2.3.1 ระบบควบคุมการเคลื่อนที่ของเครื่องมือ.....11

2.3.2 รูปแบบการทำงานของเครื่องมือ.....13

2.4 ขั้นตอนการสร้างชิ้นงาน.....13

บทที่ 3 หลักการและทฤษฎีที่ใช้ในการออกแบบ.....14

3.1 กล่าวนำ.....14

3.2 ส่วนการรับข้อมูล G-Code.....14

3.2.1 โปรแกรม AutoCAD R14.....14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.2.2 โปรแกรม Master CAM.....16

3.3 ส่วนการแปลงข้อมูล G-Code เป็นระบบพิกัด X-Y-Z.....17

3.3.1 การเขียนโปรแกรมด้วย Visual Basic.....17

3.3.2 ชุดคำสั่ง G-Code และทฤษฎีทางคณิตศาสตร์ที่นำมาประยุกต์ใช้.....28

3.3.3 ทฤษฎีทางคณิตศาสตร์ที่นำมาประยุกต์ใช้.....30

3.4 ส่วนการแสดงผลภาพและการส่งข้อมูลไปยังส่วนควบคุมการเคลื่อนที่.....32

3.4.1 คำสั่งที่ใช้ติดต่อกับ Motion Control.....32

บทที่ 4 การออกแบบและการสร้างโปรแกรมแปลง G-Code.....33

4.1 กล่าวนำ.....33

4.2 ระบบการทำงานโดยรวมของโปรแกรม.....33

4.3 การออกแบบโปรแกรม.....34

บทที่ 5 ผลการทดลอง.....36

5.1 ภาพหน้าต่างของโปรแกรมแปลงรหัส G-Code.....36

5.2 ภาพหน้าต่างของโปรแกรมแปลงรหัส ส่วนการตั้งค่าพารามิเตอร์.....37

5.3 การทดลอง.....38

5.3.1 เขียนแบบชิ้นงานที่ต้องการ ด้วยโปรแกรม AutoCAD.....38

5.3.2 แปลงแบบชิ้นงานให้เป็นข้อมูล G-Code ด้วยโปรแกรม Master CAM.....38

5.3.3 ข้อมูล G-Code ที่ได้จากโปรแกรม Master CAM.....39

5.3.4 เปิดโปรแกรมแปลง G-Code และทำการตั้งค่าพารามิเตอร์.....39

5.3.5 เปิดข้อมูล และทำการแปลงข้อมูล G-Code ให้เป็นระบบพิกัด X-Y-Z.....40

5.3.6 ภาพจำลองก่อนทำการส่งข้อมูลต่อไปยังส่วนควบคุมการเคลื่อนที่.....40

บทที่ 6 สรุปผลและข้อจำกัดของโครงงาน.....41

6.1 สรุปผล.....41

6.2 ข้อจำกัดของโครงงาน.....41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บรรณานุกรม.....42

ภาคผนวก.....43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

3.1 แสดงชื่อคอนโทรลที่สำคัญ.....18

3.2 แสดงชุดคำสั่ง G-Code ที่ใช้สำหรับโครงงานนี้.....28

3.3 แสดงผลของโมชั่นคอนโทรลเมื่อป้อนคำสั่ง 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่		หน้า
1.1	แสดง Project Flow Chart	3
2.1	แสดงเครื่องกัดแนวดิ่งซีเอ็นซี	4
2.2	แสดงเครื่องซีเอ็นซีในรูปแบบต่างๆ	6
2.3	แสดง โต๊ะงาน บอลล์สกรู และ การขับป้อนโต๊ะงาน	7
2.4	แสดงมอเตอร์ไฟฟ้าแบบต่างๆ	8
2.5	แสดงระบบควบคุมเครื่องจักรกลซีเอ็นซี	9
2.6	แสดงผลของคำสั่งในโปรแกรมเอ็นซี	10
2.7	แสดงโปรแกรมบล็อกที่ประกอบด้วยคำ 7 คำ	10
2.8	แสดงการควบคุมแบบจุดต่อจุด	12
2.9	แสดงการควบคุมแบบตัดตรง	12
2.10	การควบคุมการเคลื่อนที่ตามเส้นขอบรูป	13
3.1	แสดงหน้าต่างของโปรแกรม Auto CAD R14	15
3.2	แสดงหน้าต่างของโปรแกรม Master CAM	16
3.3	แสดงคอนโทรล ControlButton	19
3.4	แสดงคอนโทรล Text Box	20
3.5	แสดงคอนโทรล Label	20
3.6	แสดงคอนโทรล Timer	21
3.7	แสดงคอนโทรล Data	25
3.8	แสดงคอนโทรล MsflexGrid	25
3.9	แสดงคอนโทรล MSComm	28
3.10	การเคลื่อนที่ด้วยคำสั่ง G00	29
3.11	การเคลื่อนที่ด้วยคำสั่ง G01	29
3.12	การเคลื่อนที่ด้วยคำสั่ง G02	29
3.13	การเคลื่อนที่ด้วยคำสั่ง G03	30
3.14	แสดงมุม β ซึ่งเกิดจากเส้นของจุดต้นและจุดปลาย	31
3.15	ทิศทางการเคลื่อนที่ ตามเข็มนาฬิกา(Clock Wise; CW)	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
3.16	ทิศทางการเคลื่อนที่ ทวนเข็มนาฬิกา(Counter Clock Wise; CCW).....	32
4.1	แสดงลำดับการทำงานโดยรวม.....	33
4.2	Flow Chart แสดงการทำงานของโปรแกรมแปลง G-Code.....	35
5.1	แสดงหน้าต่างโปรแกรมแปลงรหัส G-Code.....	36
5.2	แสดงหน้าต่างโปรแกรมแปลงรหัส G-Code ส่วนการตั้งค่าพารามิเตอร์.....	37
5.3	แสดงการเขียนแบบชิ้นงานด้วยโปรแกรม AutoCAD.....	38
5.3	แสดงการแปลงแบบชิ้นงานให้เป็นข้อมูล G-Code ด้วยโปรแกรม Master CAM.....	38
5.4	แสดงข้อมูล G-Code ที่ได้จากโปรแกรม Master CAM.....	39
5.5	แสดงการตั้งค่าพารามิเตอร์ของโปรแกรมแปลงรหัส G-Code.....	39
5.5	แสดงการเปิดและแปลงข้อมูล G-Code ให้เป็นระบบพิกัด X-Y-Z.....	40
5.6	แสดงภาพชิ้นงานที่ได้จากการทดลอง.....	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและเหตุจูงใจของการวิจัย

เนื่องด้วยในส่วนของภาคอุตสาหกรรมเป็นจำนวนมากมีความต้องการที่จะปรับปรุงระบบการผลิตให้ดีขึ้น ทั้งด้านคุณภาพ และ ปริมาณของชิ้นงาน ซึ่งแต่เดิมกระบวนการการผลิตจะใช้เครื่องจักรที่มีความจำเป็นต้องใช้ช่างที่มีฝีมือในการทำงาน ทำให้ความสามารถในการสร้างชิ้นงานขึ้นมามีปัจจัยขึ้นอยู่กับหลายๆอย่าง และปัจจัยที่สำคัญก็คือฝีมือของช่างแต่ละบุคคล และด้วยปัจจัยนี้ก็เป็นส่วนสำคัญที่ทำให้ประสิทธิภาพของช่างแต่ละคนไม่เท่ากัน ทำให้ความสามารถในผลิตผลิตภัณฑ์ที่มีลักษณะเหมือนๆกันในปริมาณมากๆน้อยลง เป็นสาเหตุให้คุณภาพมาตราฐานของผลิตภัณฑ์แบบนั้นๆมีค่าไม่เท่ากัน ดังนั้นจึงมีผู้สร้างเครื่องจักรที่ใช้คอมพิวเตอร์เข้ามาควบคุมการทำงาน เรียกว่า เครื่องจักรซีเอ็นซี (Computer Numerical Control; CNC) โดยการทำงานนั้นสามารถโปรแกรมให้เครื่องผลิตชิ้นงานที่ต้องการเป็นจำนวนมากๆได้ และได้คุณภาพของงานตามที่ต้องการทุกชิ้น และยังสามารถเปลี่ยนรูปแบบของชิ้นงานได้โดยง่าย เพียงแต่อาศัยการเปลี่ยนโปรแกรมการสั่งงานเท่านั้น ทำให้ไม่ต้องอาศัยช่างฝีมือที่ชำนาญการมากนักในการทำงาน เพียงแต่ต้องการช่างที่สามารถควบคุมการทำงานของเครื่องจักรให้ได้เท่านั้น ทำให้การทำงานง่ายขึ้น แต่ว่าเครื่องจักรซีเอ็นซีนั้น เป็นเครื่องจักรที่มีราคาสูงมาก ดังนั้นในภาคอุตสาหกรรมขนาดเล็ก จึงไม่สามารถนำเครื่องจักรซีเอ็นซีมาใช้งานได้ เพราะจะเป็นการเพิ่มต้นทุนซึ่งสูงเกินไป จะทำให้ไม่คุ้มค่าในการลงทุน แต่ในอุตสาหกรรมหลายรูปแบบก็มีเครื่องจักรที่ใช้งานเดิมอยู่แล้ว จะพบว่าถ้าเราสามารถปรับปรุงเครื่องจักรเก่าที่ยังใช้งานได้ มาติดตั้งส่วนควบคุม ที่สามารถควบคุมได้ด้วยเครื่องคอมพิวเตอร์ได้ จะทำให้เครื่องจักรตัวนั้นสามารถทำงานได้ในลักษณะเทียบเท่ากับเครื่องซีเอ็นซีราคาแพงได้

ด้วยเหตุดังกล่าวจึงเป็นแรงบันดาลใจให้ออกแบบสร้างโปรแกรมแปลงชุดคำสั่ง G-CODE ให้อยู่ในชุดคำสั่งแบบพิกัด X-Y-Z ที่สามารถนำไปใช้ได้กับเครื่องเขียนภาพแบบแกนประสานX-Y หรือเครื่องควบคุมการเคลื่อนที่แบบ 3 แกนประสาน X-Y-Z ซึ่งสามารถประยุกต์ใช้ใน งานตัด สลัก หรือกระทั่ง งานเจาะ ได้เทียบเคียงเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ที่มีราคาแพงได้

ในการทำงานครั้งนี้ได้นำข้อมูลรหัส G-Code ที่อยู่ในนามสกุล NC (*.NC)ซึ่งเป็นข้อมูลที่ได้มาจากโปรแกรม Master CAM มาแปลงเป็นข้อมูลในระบบพิกัด X-Y-Z โดยใช้โปรแกรมวิชวลเบสิกเป็นเครื่องมือในการแปลงข้อมูล หลังจากได้ทำการทำการทดลองแปลงข้อมูลด้วยโปรแกรมแล้วจะเป็นการนำข้อมูลที่ได้ ส่งไปยังชุดควบคุมการเคลื่อนที่ของเครื่องจักรต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของปริญญานิพนธ์

ปริญญานิพนธ์นี้จะเป็นการศึกษาและออกแบบสร้างโปรแกรมแปลงชุดคำสั่ง G-Code ให้อยู่ในชุดคำสั่งแบบพิกัด X-Y-Z ที่สามารถนำไปใช้ได้กับเครื่องเขียนภาพแบบแกนประสาน X-Y หรือเครื่องควบคุมการเคลื่อนที่แบบ 3 แกนประสาน X-Y-Z ได้

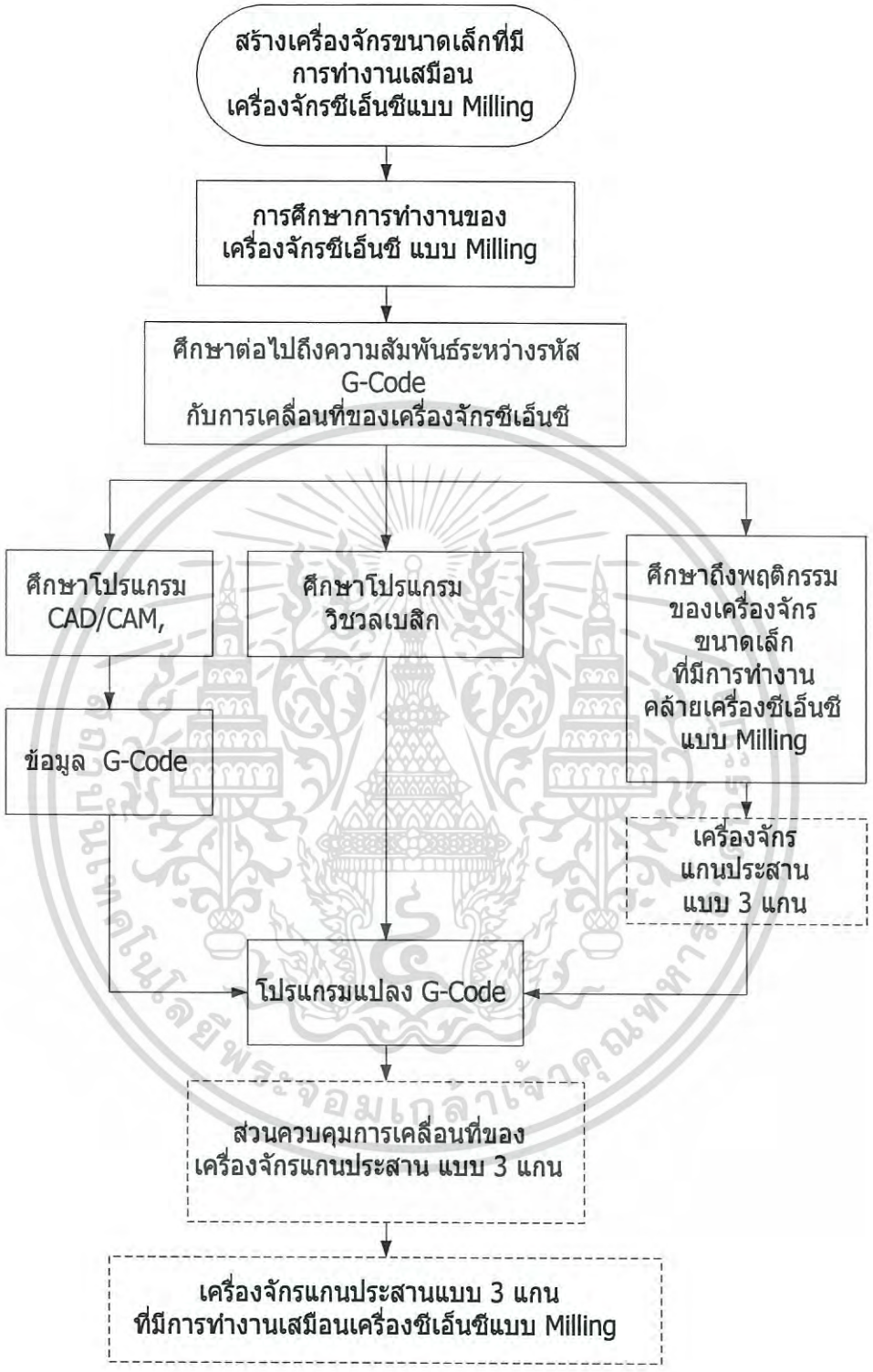
1.3 ขอบเขตของปริญญานิพนธ์

ปริญญานิพนธ์เล่มนี้จะกล่าวถึงการออกแบบสร้างโปรแกรมแปลงชุดคำสั่ง G-CODE ให้อยู่ในชุดคำสั่งแบบพิกัด X-Y-Z ที่สามารถนำไปใช้ได้กับเครื่องเขียนภาพแบบแกนประสาน X-Y หรือเครื่องควบคุมการเคลื่อนที่แบบ 3 แกนประสาน X-Y-Z และสามารถประยุกต์ใช้ใน งานตัด สลัก หรือกระทั่ง งานเจาะ ได้เทียบเคียงเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ได้

1.4 ขั้นตอนการศึกษา

การทำโครงงานวิจัยในปริญญานิพนธ์ฉบับนี้มีขั้นตอนการศึกษาเริ่มจากการศึกษาการทำงานของเครื่องจักรซีเอ็นซี แบบ Milling จากนั้นจึงได้ศึกษาต่อไปถึงความสัมพันธ์ระหว่างรหัส G-Code กับการเคลื่อนที่ของเครื่องจักรซีเอ็นซี เมื่อทราบถึงความสัมพันธ์ทั้งหมดจึงได้มีการศึกษาถึงพฤติกรรมของเครื่องจักรแกนประสาน เพื่อที่จะนำมาใช้ในการเลือนแบบการทำงานให้มีความเสมือนการทำงานของเครื่องจักรซีเอ็นซีแบบ Milling ได้ ตลอดจนได้ศึกษาโปรแกรม CAD/CAM โปรแกรมวิชวลเบสิก ซึ่งเป็นโปรแกรมที่นำมาใช้ในการทำโครงงานครั้งนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



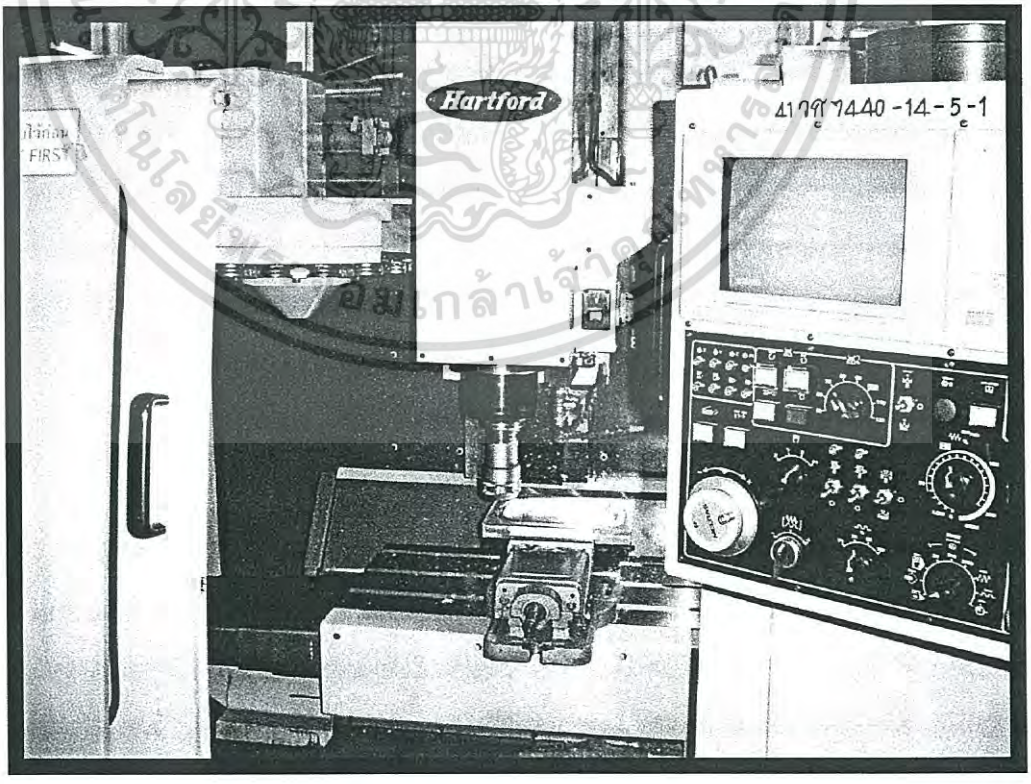
ภาพที่ 1.1 แสดง Project Flow Chart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ความรู้เบื้องต้นเกี่ยวกับเครื่องซีเอ็นซี

2.1 กล่าวนำ

เครื่องจักรซีเอ็นซีเป็นเครื่องจักรที่ใช้คอมพิวเตอร์ในการควบคุมการทำงาน ซึ่งมีหลายรูปแบบแตกต่างกันไปตามลักษณะการใช้งาน เช่น เครื่องกลึง เครื่องกัด เครื่องคว้าน เครื่องตัดเจาะต่างๆ เป็นต้น โดยเครื่องจักรแบบนี้มีลักษณะเด่นอยู่ว่า จะทำงานตามโปรแกรมสั่งงานที่ตั้งไว้ และยังสามารถแก้ไขปรับเปลี่ยนโปรแกรมการทำงาน เพื่อสร้างชิ้นงานในลักษณะต่างๆกันออกมาได้โดยง่าย โดยอาศัยเครื่องจักรตัวเดิม ไม่ต้องออกแบบตัวเครื่องจักรใหม่ เพียงแต่ปรับเปลี่ยนโปรแกรมการทำงานเท่านั้น ทำให้มีความสะดวกในการทำงานเพิ่มขึ้นมาก และยังสามารถทำชิ้นงานที่เหมือนๆกันออกมาได้ตามจำนวนที่ต้องการ ในเวลาที่สามารถกำหนดได้ เป็นการประหยัดทั้งเวลาทำงาน และ แรงงานในการทำงาน และโปรแกรมการทำงานยังเห็นเสมือนการจัดเตรียมการทำงานอย่างเป็นขั้นตอน เป็นการป้องกันความเสียหายที่อาจเกิดได้ และ ยังเป็นการช่วยลดเวลาในการผลิต เช่น ลดเวลาในการตรวจสอบขนาดชิ้นงาน ลดเวลาในการปรับเปลี่ยนความเร็วรอบในการทำงาน ทำให้มีความถูกต้องแม่นยำในการทำงานสูงขึ้นมาก



ภาพที่ 2.1 แสดงเครื่องกัดแนวดิ่งซีเอ็นซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องจักรซีเอ็นซีที่นำมาใช้งานโดยทั่วไปนี้จะมีข้อดี ข้อเสีย เปรียบเทียบกับเครื่องจักรกลโดยทั่วๆไปพอสรุปได้ดังนี้

ข้อดีของเครื่องจักรกลซีเอ็นซี คือ

1. มีความยืดหยุ่นในการทำงานสูง
2. มีความเที่ยงตรงในการทำงานสูง
3. เวลาที่ใช้ในการผลิตจะเร็วกว่า
4. สามารถผลิตชิ้นงานที่มีความซับซ้อนได้ง่ายกว่า
5. การปรับตั้งระบบการทำงานจะง่ายกว่า ใช้เวลาน้อยกว่าการผลิตโดยวิธีอื่นๆ
6. สามารถหลีกเลี่ยงการใช้ช่างควบคุมที่ต้องมีทักษะ และประสบการณ์สูง
7. ช่างควบคุมจะมีเวลาว่างจากการควบคุมเครื่อง ทำให้สามารถจัดเตรียมงานอื่นๆ ต่อไปได้
8. การตรวจสอบคุณภาพจะทำได้ง่าย

ข้อเสียของเครื่องจักรซีเอ็นซี คือ

1. ราคาค่อนข้างสูง
2. การบำรุงรักษาซับซ้อนกว่า
3. จำเป็นต้องใช้ช่างเขียนโปรแกรมที่มีทักษะ และถูกฝึกอบรมมาเฉพาะ
4. ชิ้นส่วน หรือ อะไหล่ในการซ่อมบำรุงมีราคาแพง
5. การซ่อมบำรุงต้องอาศัยช่างที่มีประสบการณ์สูง และผ่านการฝึกอบรมมาอย่างดี
6. ราคาของเครื่องมือ (Tool) ต่างๆที่ใช้ในขบวนการผลิตมีราคาสูง
7. พื้นที่ที่ใช้ในการติดตั้งเครื่องจักรต้องการควบคุมที่ดี เช่น ควบคุมอุณหภูมิ ความชื้น ความร้อน

จากข้อดี และข้อเสียที่เห็นจะพบว่าในระบบอุตสาหกรรมขนาดเล็ก จนถึงขนาดกลางนั้น ในส่วนของราคาจะไม่ค่อยคุ้มค่ากับการลงทุน ดังนั้นถ้าหากสามารถดัดแปลงเฉพาะในส่วนของการควบคุมเข้าไปยังเครื่องจักรเก่าที่มีอยู่จะทำให้ราคาของซีเอ็นซีที่ได้จะถูกลงมาก และในส่วนของซอฟต์แวร์ที่ออกแบบขึ้นไม่ซับซ้อนทำให้ใช้งานได้โดยง่าย

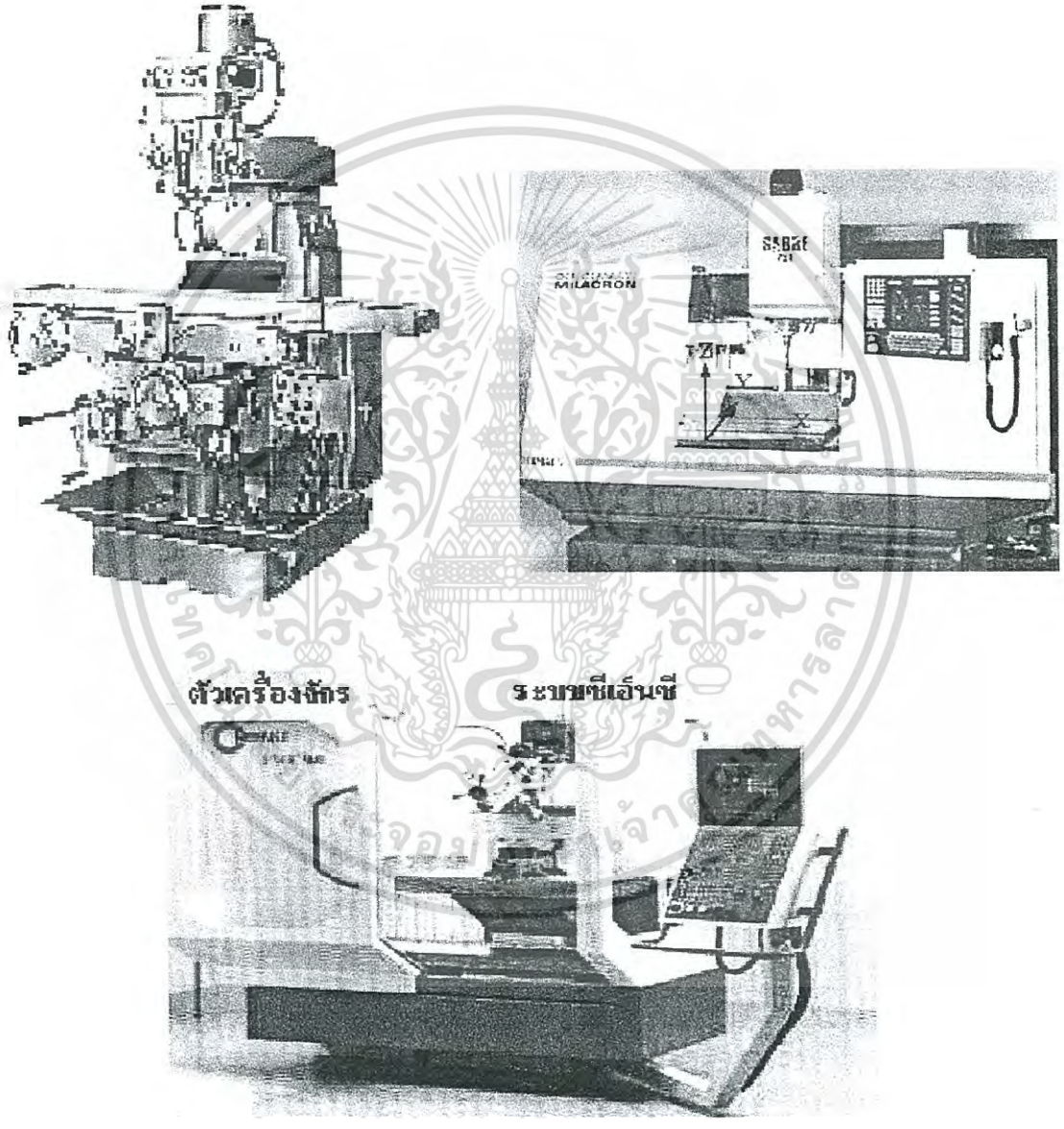
2.2 ส่วนประกอบของเครื่องซีเอ็นซีโดยทั่วไป

เครื่องซีเอ็นซีนี้จะแบ่งเป็น 2 ส่วนด้วยกัน ดังนี้ ส่วนแรกจะเป็นส่วนของเครื่องจักรที่ใช้ควบคุมการทำงาน เป็นส่วนที่รับ บันทึก แก้ไข โปรแกรมการทำงาน ที่ทำหารควบคุมเครื่องจักรให้ผลิตชิ้นงานต่างๆ ออกมาอีกที โดยเครื่องจักรกลแบบนี้การเคลื่อนที่จะอาศัยมอเตอร์เป็นตัวขับให้เคลื่อนที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 ส่วนเครื่องจักรที่ใช้ผลิตชิ้นงาน

เครื่องจักรกลซีเอ็นซีจะอยู่หลายชนิดมากแต่ในที่นี้จะเป็นชนิดเครื่องกัแนวดิ่งซีเอ็นซีซึ่งเครื่องกัดซีเอ็นซีนี้จะเป็นเครื่องจักกลที่มีขอบเขตการทำงานกว้างขวาง คือ นอกจากจะทำงานเป็นเครื่องกัดโดยทั่วไปแล้ว ยังสามารถทำงานในลักษณะอื่นๆ ได้ เช่น ทำรู เกลียว คว้านรูได้อีกด้วย และในเครื่องกัดก็ยังมีแบบที่มีแนวควบคุมตั้งแต่ 3 แกน 4 แกน 5 แกน และ มากกว่านั้น ดังรูป

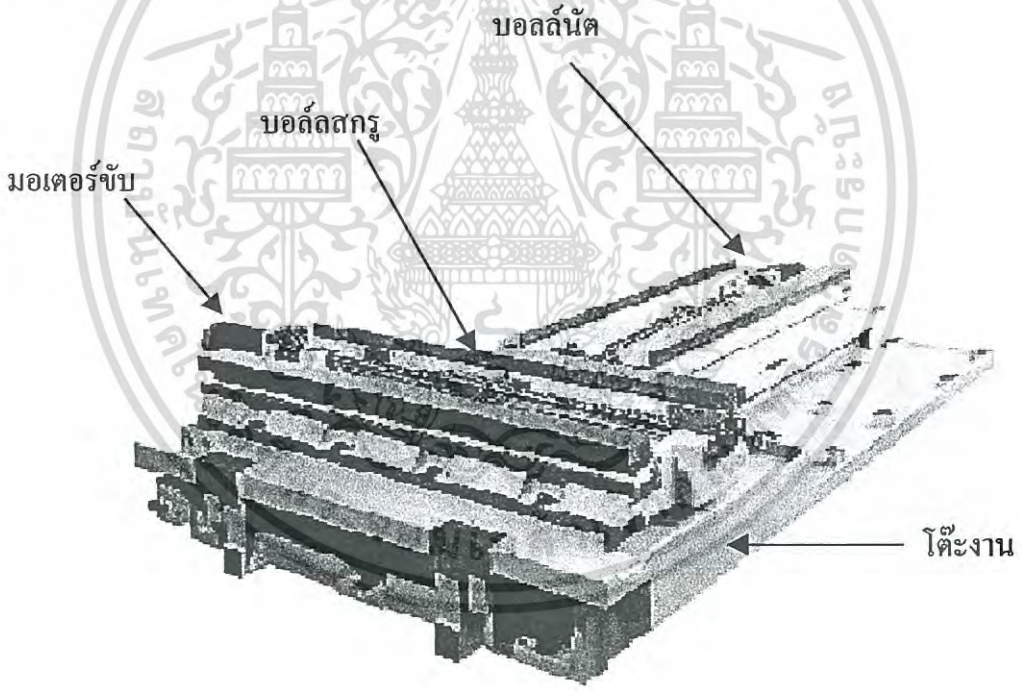


ภาพที่ 2.2 แสดงเครื่องซีเอ็นซีในรูปแบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องกัดนี้จะประกอบด้วยส่วนต่างๆ ดังนี้

โต๊ะสำหรับจับยึดชิ้นงาน เป็นโต๊ะที่จะทำการกัดไปจับยึดไว้บนนั้น โดยการจับยึดชิ้นงานกัด จุดสำคัญจะอยู่ที่การกำหนดตำแหน่งของชิ้นงานบนโต๊ะงานที่ถูกต้อง และรวดเร็ว ควรเป็นอุปกรณ์ที่สามารถถอด และใส่ชิ้นงานได้รวดเร็ว และสามารถบังคับตำแหน่งในการทำงานจองชิ้นงานซ้ำๆ ได้อย่างถูกต้องเที่ยงตรงเสมอ พร้อมทั้งยังสามารถเลื่อนไปมาตามแกนได้อย่างถูกต้องแม่นยำ โดยอาศัยกำลังขับจากมอเตอร์ไฟฟ้าปั่นขับให้โต๊ะเลื่อนส่งผ่านกำลังจากมอเตอร์ไปยังบอลสกรู นัตก็จะเคลื่อนที่ไปตลอดความยาวของสกรู พาแท่นเลื่อนและโต๊ะงานเคลื่อนที่ไปตามรางเลื่อน ในนัตเป็นเหล็กชุบแข็ง เจียรไนผิวเรียบมัน เพื่อเป็นการลดความฝืด และเพิ่มความถูกต้องในการเคลื่อนที่ และยังประกอบด้วยลูกบอลจำนวนมากหมุนและนัตยังออกแบบสำหรับลดระยะคลอนเพื่อให้การเคลื่อนที่ของปท่นเลื่อนมีความเที่ยงตรงสูง ดังนั้นการควบคุมโต๊ะให้เลื่อนไปยังตำแหน่งที่ต้องการก็คือ การควบคุมการหมุนองมอเตอร์ให้หมุนในทิศทาง และความเร็วที่ถูกต้องและแม่นยำ



ภาพที่ 2.3 แสดง โต๊ะงาน บอลล์สกรู และ การขับป้อนโต๊ะงาน

ระบบป้อนขับ โดยทั่วไปใช้มอเตอร์ในการหมุนขับ และควบคุมการทำงานด้วยวงจรอิเล็กทรอนิกส์ เป็นมอเตอร์ไฟฟ้าที่สมารถควบคุมให้หยุด หรือ ให้หมุนในทิศทางที่ต้องการและหมุนในความเร็วใดๆ ที่ต้องการก็ได้ ในระบบซีเอ็นซีสมัยใหม่จะใช้ระบบขับป้อนแบบเซอร์โว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Servo drivers) ทำให้สามารถปรับอัตราป้อน ความเร็วรอบได้อย่างไม่จำกัดตลอดช่วงความเร็วอย่างแม่นยำอย่างรวดเร็ว โดยทั่วๆ ไป มอเตอร์ที่ใช้ในระบบขับป้อนมี 3 ชนิดด้วยกัน คือ

มอเตอร์กระแสตรง (DC Motor)มอเตอร์ชนิดนี้ มีข้อดีคือ เป็นมอเตอร์ที่สามารถปรับอัตราป้อนได้อย่างละเอียด และมีวงจรควบคุมที่ไม่ซับซ้อน แต่มีข้อเสียคือ เมื่อต้องการกำลังขับสูง มอเตอร์จะมีขนาดใหญ่ และเมื่อความเร็วรอบสูงๆ แรงบิดจะลดลง ดังนั้นจะใช้กับเครื่องจักรกลขนาดกลาง และเล็ก

มอเตอร์แบบขั้น (Stepping Motor) เป็นมอเตอร์ที่ทำงานแบบต่อเนื่อง โดยการแปลงคลื่นสัญญาณที่ป้อนให้เป็นการเคลื่อนที่เชิงมุม ความเที่ยงตรงของระบบจะขึ้นอยู่กับความสามารถของมอเตอร์ในการแบ่งขั้นการหมุนตามจำนวนคลื่นของสัญญาณที่ป้อนเข้าระบบ ในมอเตอร์แบบนี้แรงบิดจะลดลงเมื่อความเร็วในการหมุนเพิ่มขึ้น ดังนั้นจะเหมาะสำหรับเครื่องจักรกลเล็กๆ เช่น พร็อตเตอร์

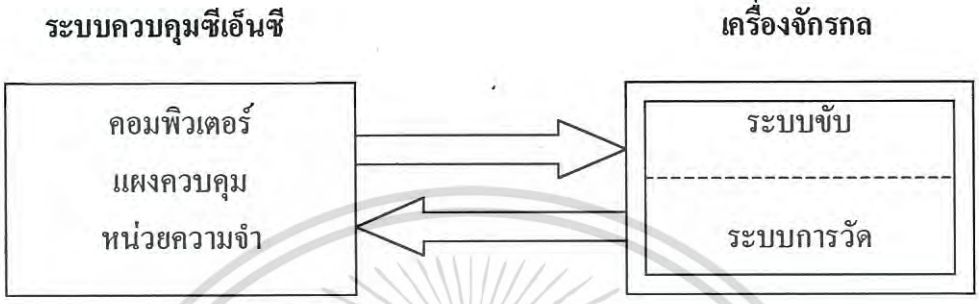
มอเตอร์กระแสสลับ (AC Motor) มอเตอร์ชนิดนี้จะใช้วงจรควบคุมที่ยุ่งยากซับซ้อนกว่า 2 ประเภทข้างต้น แต่ในขนาดที่เท่ากันจะมีแรงบิดสูงกว่า จะใช้งานที่ต้องการแรงบิดที่สูงและขนาดของมอเตอร์จะไม่ใหญ่นักด้วย

ภาพที่ 2.4 แสดงมอเตอร์ไฟฟ้าแบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 ส่วนควบคุม

ในระบบซีเอ็นซีจะมีคอมพิวเตอร์อยู่ในระบบ ซึ่งเป็นตัวเก็บข้อมูลโปรแกรมต่างๆ และยังเป็นตัวประมวลผลควบคุมเครื่องจักรอีกด้วย ทำให้ช่างควบคุมเครื่องสามารถเขียนโปรแกรมได้ด้วยตัวเอง และ ยังสามารถแก้ไปโปรแกรมหลังจากป้อนเข้าไปในระบบควบคุมเครื่องได้



ภาพที่2.5 แสดงระบบควบคุมเครื่องจักรกลซีเอ็นซี

ในส่วนของการควบคุมในเครื่องซีเอ็นซีจะแบ่งเป็น 2 ส่วน คือ การควบคุมการเคลื่อนที่ของแนวแกนในลักษณะต่างๆ ให้เครื่องมือเคลื่อนที่ตามรูปทรงเรขาคณิตของชิ้นงาน และควบคุมหน้าที่การทำงานของเครื่องจักรกล(Control of machine function) เช่นการควบคุมระบบหล่อเย็น การเปิดสัญญาณไฟแสดงการทำงานเครื่อง เป็นต้น

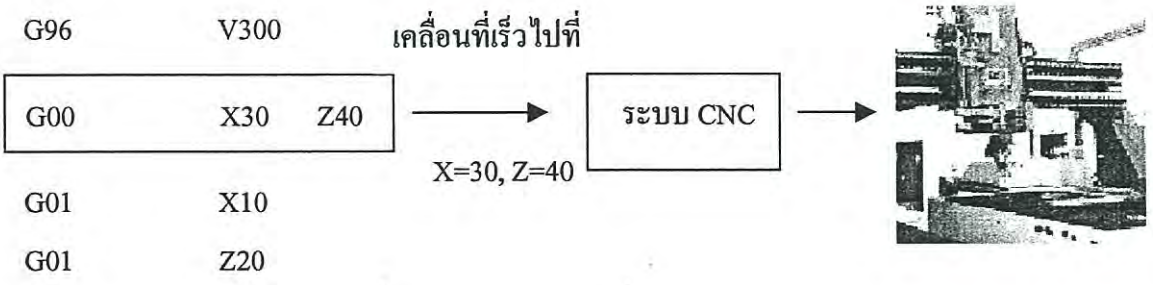
2.2.3 ส่วนโปรแกรมสั่งงานเครื่องจักร

ระบบควบคุมเครื่องจักรกลซีเอ็นซี จะเป็นส่วนที่ทำหน้าที่ควบคุมให้เครื่องจักรกลทำงานตามขั้นตอนต่างๆที่กำหนดไว้ การที่เครื่องจักรเหล่านั้นจะทำงานได้ จะต้องมีโปรแกรม ที่ช่างผู้ควบคุมเครื่องหรือ ช่างเขียนโปรแกรมป้อนเข้าไปในระบบควบคุม และเรียกโปรแกรมที่ป้อนเข้าในระบบว่าโปรแกรมเอ็นซี จากนั้นระบบควบคุมจะอ่านโปรแกรมเอ็นซีและเปลี่ยนข้อมูลเหล่านั้นให้เป็นสัญญาณควบคุมสำหรับเครื่องจักรกล การสร้างโปรแกรมซีเอ็นซีจะมีรูปแบบที่ถูกกำหนดไว้เป็นมาตรฐานไว้แล้ว

โปรแกรมเอ็นซีจะเป็นการกำหนดขั้นตอนการตัดเฉือนทั้งหมด ต่างๆ ของเครื่องจักรไว้พร้อมด้วยเงื่อนไขการทำงานอื่นๆ เช่น อัตราป้อน ความเร็วรอบของเพลางานเป็นต้น หลังจากที่โปรแกรมเอ็นซีนี้ถูกป้อนเข้าไปในระบบควบคุมแล้ว จะสามารถที่จะนำมาใช้ในการตัดเฉือนงานได้หลายๆ ครั้งตามที่ต้องการ

ถ้าเมื่อไหร่ที่มีความต้องการปรับปรุงเปลี่ยนแปลงแก้ไขโปรแกรมให้เหมาะสม ในระบบควบคุมซีเอ็นซีก็จะสามารถทำได้ด้วยช่างควบคุมเครื่องโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.6 แสดงผลของคำสั่งในโปรแกรมเอ็นซี

ในภาพที่ 2.6 คำสั่งโปรแกรมเอ็นซีเขียนว่า เคลื่อนที่เร็วไปยังจุด X=30,Z=30 จะทำให้มอเตอร์ขับในแกน X และ แกน Z เริ่มทำงาน จะทำไปจนกว่าจะเคลื่อนที่ไปยังจุด X=30, Z=40

โปรแกรมเอ็นซีจะประกอบด้วยคำสั่งเป็นบรรทัดต่อๆกัน เรียก แต่ละบรรทัดว่า บล็อก ซึ่งเป็นการเขียนขึ้นตามขั้นตอนการตัดเฉือนที่กำหนดไว้รวมกันนั่นเอง

โปรแกรมบล็อกจะแบ่งออกได้โดยการใช้หมายเลขบล็อก เช่น N10 N20 N30 เป็นต้น

ภาษาโปรแกรมเอ็นซี จะเป็นบล็อกที่ประกอบด้วยคำ(Words) หลายๆ คำรวมกัน คำรวมกันคำเหล่านี้จะประกอบจากตัวอักษร หรือ สัญลักษณ์ กับตัวเลขรวมกัน

ภาพที่ 2.7 แสดงโปรแกรมบล็อกที่ประกอบด้วยคำ 7 คำ

คำที่ใช้ในโปรแกรมบล็อกอาจจะทำหน้าที่เป็นคำสั่ง หรือ เป็นเงื่อนไขเสริมสำหรับการทำงานก็ได้ ขึ้นอยู่กับตัวอักษรและตัวเลขที่กำกับอยู่ ตัวอักษรคำสั่งที่มีความสำคัญมาก คือ G คำสั่ง G (G00-G99) ส่วนมากจะเป็นคำสั่งที่ใช้เกี่ยวกับการควบคุมการเคลื่อนที่ของเครื่องมือ

ส่วนอักษรที่ใช้สำหรับเงื่อนไขเสริมที่สำคัญได้แก่

X-Y-Z : ข้อมูลโคออร์ดิเนท

F : อัตราป้อน

S : ความเร็วรอบของเพลางาน

ในการเขียนภาษาโปรแกรมสำหรับระบบควบคุมซีเอ็นซี ผู้ผลิตจะระบุ

เอกสารนี้ ... นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คำสั่งที่สามารถโปรแกรมได้มีอะไรบ้าง
- การทำงานเสริมด้านใดบ้างที่สามารถใช้ร่วมกับคำสั่งแต่ละคำ
- ตัวอักษร และ ตัวเลขใดบ้างที่ใช้เขียนเป็นคำสั่ง และการทำงานเสริม

เมื่อโปรแกรมควบคุมได้ถูกป้อนเข้าไปในระบบควบคุมแล้ว ระบบควบคุมจะทำการตรวจสอบว่าการเขียนโปรแกรมนั้นเป็นไปตามกฎหรือไม่ ส่วนการป้อนข้อมูลที่เป็นโคออร์ดิเนทที่ไม่ถูกต้องโดยช่างวเขียนโปรแกรมจะตรวจสอบพบได้เมื่อโปรแกรมทำงานแล้ว

2.3 การทำงานของเครื่องซีเอ็นซี

ลักษณะการควบคุมการเคลื่อนที่ของแท่นเลื่อนต่างๆ ในเครื่องจักรซีเอ็นซีมีอยู่ 2 ลักษณะ คือ

การเคลื่อนที่ในแนวเส้นตรง (Linear Interpolation) การเคลื่อนที่ในลักษณะนี้คอมพิวเตอร์ในระบบจะทำการคำนวณหาตำแหน่งของจุดต่างๆ ที่ต่อกันในแนวเส้นตรง ระหว่างตำแหน่งของเครื่องมือ 2 ตำแหน่ง ในขณะที่เครื่องมือเคลื่อนที่จากจุดหนึ่งไปยังอีกจุดหนึ่งนั้น ระบบควบคุมของซีเอ็นซีจะตรวจสอบ และแก้ไขแนวแกนในการเคลื่อนที่ให้ถูกต้องอยู่ตลอดเวลา ทำให้การเคลื่อนที่ของเครื่องมือไม่ผิดพลาด หรือ คลาดเคลื่อนออกจากจุดต่อของเส้นตรงมากกว่าค่าพิกัดความเผื่อของเครื่องที่กำหนดไว้

การเคลื่อนที่แนวเส้นโค้ง (Circular Interpolation) ระบบควบคุมซีเอ็นซีจะคำนวณหาตำแหน่งของจุดต่างๆ ที่ต่อกันเป็นเส้นโค้งตามขนาดรัศมีที่กำหนดระหว่างตำแหน่งของเครื่องมือที่กำหนดไว้ 2 ตำแหน่ง ระบบควบคุมจะอาศัยจุดเหล่านี้ในการตรวจสอบ และแก้ไขแนวการเคลื่อนที่ของเครื่องมือให้ถูกต้อง และอยู่ในพิกัดความเผื่อของเครื่องจักรกลที่กำหนด

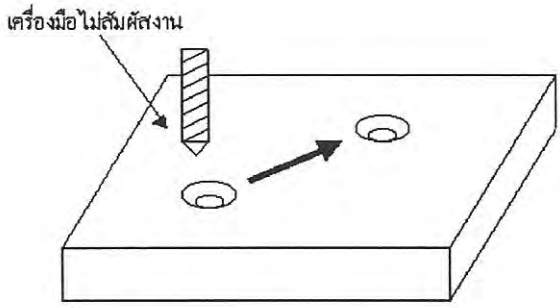
2.3.1 ระบบควบคุมการเคลื่อนที่ของเครื่องมือ

ในระบบควบคุมซีเอ็นซีจะแบ่งการควบคุมการเคลื่อนที่ทั้งสองลักษณะการเคลื่อนที่ป้อนออกเป็น 3 ชนิด คือ

2.3.1.1 การควบคุมการเคลื่อนที่แบบจุดต่อจุด

การควบคุมแบบนี้จะควบคุมการเคลื่อนที่ของเครื่องมือระหว่างจุดสองจุดที่โปรแกรมไว้ในลักษณะการเคล่อนที่เร็ว(Rapid Traverse) โดยที่เครื่องมือจะต้องไม่สัมผัสชิ้นงาน ดังภาพที่ 2.8 แนวแกนในการเคลื่อนที่ขึ้นอยู่กับชนิดของระบบควบคุม คือ มอเตอร์ขับระบบป้อนอาจเริ่มทำงานหลายๆ แนวแกนพร้อมๆ กัน หรือ ทำงานทีละแกน จนกว่าจะเคลื่อนที่ถึงตำแหน่งที่เครื่องมือที่โปรแกรมไว้ การควบคุมแบบจุดต่อจุดมักจะใช้กับเครื่องเจาะ เครื่องเชื่อมจุด เป็นต้น

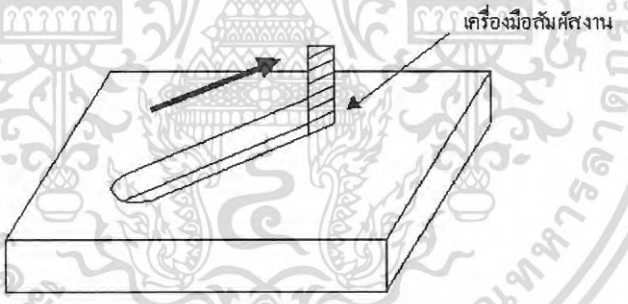
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.8 แสดงการควบคุมแบบจุดต่อจุด

2.3.1.2 การควบคุมการเคลื่อนที่แบบตัดตรง

การควบคุมการเคลื่อนที่ชนิดนี้ สามารถใช้ควบคุมการเคลื่อนที่แบบเคลื่อนที่เร็ว และยังสามารถใช้ควบคุมการเคลื่อนที่ของเครื่องมือในแนวขนานกับแนวแกนของเคครื่องจักรกล ตามค่าอัตราป้อนที่ต้องการ แต่จะสามารถควบคุมการเคลื่อนที่ได้ครั้งละ 1 แนวแกนเท่านั้น การเคลื่อนที่ของเครื่องมือจะถูกควบคุมด้วยอัตราป้อน และความยาวในการเคลื่อนที่ดัง ภาพที่ 2.9 ระบบการควบคุมการตัดเฉือนแนวเส้นตรงชนิดนี้จะใช้กับเครื่องกัด และเครื่องกลึงแบบง่ายๆ



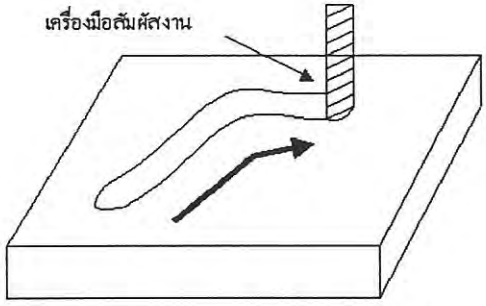
ภาพที่ 2.9 แสดงการควบคุมแบบตัดตรง

2.3.1.3 การควบคุมการเคลื่อนที่ที่เป็นเส้นโค้ง

การควบคุมการเคลื่อนที่แบบนี้สามารถควบคุมการเคลื่อนที่ทำงานได้ดังนี้

- ควบคุมเครื่องมือให้เคลื่อนที่ไปตำแหน่งที่ต้องการแบบเคลื่อนที่เร็วได้
- ควบคุมเครื่องมือให้เคลื่อนที่ขนานกับแนวแกนไปยัง ตำแหน่งที่ต้องการตามค่าอัตราป้อนได้
- ควบคุมเครื่องมือให้เคลื่อนที่ไปยังตำแหน่งใดๆ บนชิ้นงานที่กำหนดในแนวเส้นตรงและเส้นโค้งตามค่าอัตราป้อนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.10 การควบคุมการเคลื่อนที่ตามเส้นขอบรูป

2.3.2 รูปแบบการทำงานของเครื่องมือ

การทำงานของเครื่องมือสามารถ ทำงานแบบ 2 มิติ, แบบ $2^{1/2}$ มิติ และ แบบ 3 มิติ การทำงาน2 มิติ คือ การสร้างชิ้นงานให้อยู่ในระนาบเดียว มีการควบคุมการทำงานเพียง 2 แกน ซึ่งโดยทั่วไปจะเป็นระนาบ X-Y ส่วนแกน Z จะเป็นการกำหนดค่าที่แน่นอน เช่น ความลึก สำหรับการทำงานแบบ $2^{1/2}$ มิติ จะคล้ายกับแบบ 2 มิติ แต่เพิ่มในส่วนแกน Z ที่มีค่าเปลี่ยนแปลงแต่ไม่สามารถเปลี่ยนแปลงพร้อมกันได้ทั้ง 3 แกน ส่วนการทำงานแบบ 3 มิติ จะเป็นการสร้างชิ้นงานโดยที่แกนทั้ง 3 ทำงานพร้อมกัน

2.4 ขั้นตอนการสร้างชิ้นงาน

การใช้งานเครื่องซีเอ็นซี จะต้องคำนึงถึงขั้นตอนการทำงานเพื่อให้ได้งานที่มีคุณภาพ ดังนั้นจึงต้องมีการออกแบบขั้นตอนการทำงาน ดังต่อไปนี้

1.) ขั้นตอนการวางแผน ขั้นตอนนี้ทำหลังจากที่ผู้ออกแบบได้ออกแบบมาแล้ว เป็นการกำหนดว่าต้องทำส่วนไหนก่อนส่วนไหนหลัง เพื่อให้ได้งานตามที่ออกแบบไว้

2.) ขั้นตอนการเขียนโปรแกรมสั่งงาน เป็นการเขียนคำสั่งนำมาเรียงลำดับขั้นตอนการทำงานตามที่ออกแบบไว้

3.) ขั้นตอนการตรวจสอบชุดคำสั่ง ในขั้นตอนนี้ทำได้ 2 วิธี คือ วิธีที่หนึ่ง ทำการทดสอบการทำงานกับเครื่องซีเอ็นซี โดยใช้วัสดุอื่นทดสอบ วิธีที่สอง ใช้โปรแกรมจำลองการทำงาน(Simulation) ในเครื่องคอมพิวเตอร์ โดยทำการทดสอบก่อนการทำงานจริง เพื่อเป็นการป้องกันความผิดพลาดที่อาจเกิดขึ้นได้

4.) ขั้นตอนการผลิต หมายถึงตั้งแต่การเตรียมชิ้นงาน เตรียมเครื่องมืออุปกรณ์ ต่างๆ เพื่อความพร้อมที่จะทำงานได้ ซึ่งหลังจากเดินเครื่องแล้ว เครื่องซีเอ็นซีจะทำงานอัตโนมัติตามขั้นตอนการสั่งงานที่ได้ออกแบบไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
หลักการและทฤษฎีที่ใช้ในการออกแบบ

3.1 กล่าวนำ

ในการออกแบบโปรแกรมแปลง G-Code เป็นระบบพิกัด X-Y-Z ครั้งนี้ได้แยกส่วนประกอบของโปรแกรมออกเป็น 3 ส่วนหลักๆ คือ

1. ส่วนการรับข้อมูล G-Code
2. ส่วนการแปลงข้อมูลG-Code เป็นระบบพิกัด X-Y-Z
3. ส่วนการแสดงผลภาพและการส่งข้อมูลไปยังส่วนควบคุมการเคลื่อนที่

3.2 ส่วนการรับข้อมูล G-Code

การออกแบบโปรแกรมในส่วนของการรับข้อมูล G-Code นั้น จะต้องอาศัยข้อมูลที่เป็นรหัส G-Code ซึ่งจะได้มาจากโปรแกรมประยุกต์ที่ช่วยในงานเขียนแบบ(Computer Aided Design; CAD) ซึ่งในโครงงานวิจัยนี้เลือกใช้โปรแกรม AutoCAD และโปรแกรมประยุกต์ที่ช่วยในการผลิต (Computer Aided Manufacturing; CAM) ซึ่งในโครงงานวิจัยนี้เลือกใช้โปรแกรม Master CAM

3.2.1 โปรแกรม AutoCAD R14

การเขียนแบบและออกแบบโดยใช้คอมพิวเตอร์เป็นเครื่องมือช่วย (Computer Aided Design-CAD) จะเพิ่มขีดความสามารถทำงานได้มากกว่าการเขียนแบบด้วยมือเพราะคอมพิวเตอร์สารถเก็บบันทึกตำแหน่ง พิกัด ขนาด สี และแบบลายเส้นของวัตถุที่เขียนขึ้นโดยจัดเก็บไว้ในระบบฐานข้อมูล เพื่อนำแบบที่เขียนไว้ไปใช้งานต่อ หรือเพื่อการแก้ไขเพิ่มเติมในอนาคต

การใช้คอมพิวเตอร์ช่วยในเขียนแบบทำให้สามารถสร้างเส้นแบบต่างๆ ได้ตามต้องการและแก้ไขดัดแปลงเพิ่มเติมได้เมื่อมีความผิดพลาดโดยไม่ต้องเสียเวลาเขียนแบบใหม่ทั้งหมด แบบที่เขียนด้วยคอมพิวเตอร์จะให้ความละเอียดถูกต้องและนำไปพิมพ์ออกมาได้ตามต้องการ สรุปแล้วข้อได้เปรียบของการใช้คอมพิวเตอร์ช่วยในการเขียนแบบก็คือ ความสะดวกรวดเร็วและความถูกต้องแม่นยำกว่า

แต่ปัญหาของการเรียนรู้เพื่อนำคอมพิวเตอร์มาใช้งานในการเขียนแบบด้วยคอมพิวเตอร์ ก็คือการขาดความมั่นใจที่จะนั่งหน้าเครื่องคอมพิวเตอร์และใช้แป้นพิมพ์หรือเม้าส์ในการเริ่มต้นเขียนแบบ แต่แท้จริงแล้วเครื่องคอมพิวเตอร์ก็คือเครื่องคิดเลข (Calculator) แบบหนึ่งเพียงแต่มีแป้นที่จะให้กดใช้งานได้มากมายและมีขั้นตอนที่แน่นอนในการสั่งงาน จึงไม่มีผู้ใดที่จะสามารถใช้งานเครื่องคอมพิวเตอร์ได้โดยไม่เรียนรู้จากหนังสือคู่มือหรือมีผู้แนะนำ ดังนั้นผู้ที่ต้องการใช้เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

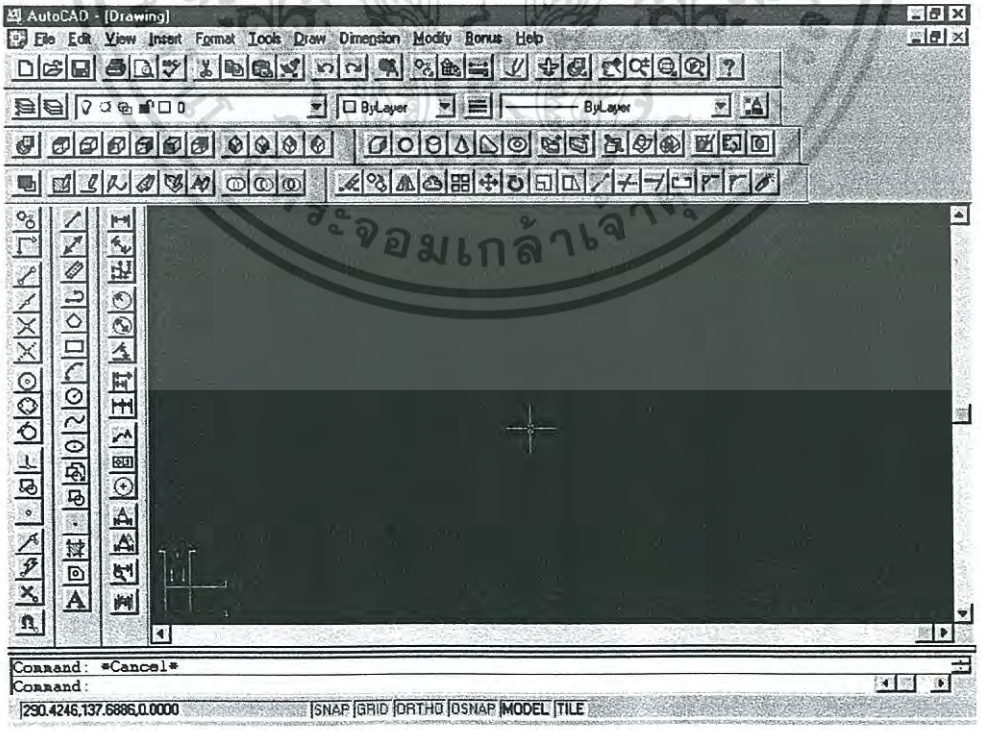
คอมพิวเตอร์ช่วยเขียนแบบจนถึงขั้นที่เกิดความมั่นใจจะต้องมีคู่มือหรือผู้ชี้แนะที่ดีและหัวใจสำคัญก็คือจะต้องมีการฝึกปฏิบัติอยู่เสมอ

การเขียนแบบด้วย AutoCAD นั้นจะต้องเรียนรู้การออกคำสั่งให้โปรแกรมเขียนเส้นตรง เส้นโค้ง วงกลม หรือตัวหนังสือ ผู้ใช้งานจะต้องรู้จักการป้อนข้อมูลและพารามิเตอร์ต่างๆ ให้คอมพิวเตอร์ตีความ โดยใช้แป้นพิมพ์หรือเลือกจากเมนูโดยใช้อุปกรณ์ชี้พารามิเตอร์ที่โปรแกรมคอมพิวเตอร์ต้องการทราบ คือ ตำแหน่ง ขนาด หรือ สี ของเส้นต่างๆ เป็นต้น หลังการออกคำสั่งและป้อนข้อมูลต่างๆ แล้ว คอมพิวเตอร์ก็จะวาดเส้นตามที่ต้องการให้ปรากฏบนจอภาพ

นอกจากการเขียนเส้นต่างๆ แล้ว AutoCAD ยังมีความสามารถในการแก้ไขเส้นที่วาดไว้ เช่น การลบ เคลื่อนย้ายหรือลอกแบบได้รวดเร็วกว่าการเขียนด้วยมือรวมทั้งการแสดงภาพหรือขยายภาพให้สามารถมองได้ชัดเจนขึ้น

AutoCAD เป็นโปรแกรมคอมพิวเตอร์ที่มีความเหมาะสมสำหรับงานเขียนแบบลายเส้น ดังต่อไปนี้

- งานเขียนแบบด้านสถาปัตยกรรม
- งานออกแบบตกแต่งภายในและแบบลายเส้นด้านศิลปกรรม
- การเขียนแผนผัง Flowchart และเขียนกราฟ
- งานเขียนแบบทางวิศวกรรมทั่วไป
- งานเขียนแบบด้านสำรวจภูมิประเทศและภูมิทัศน์



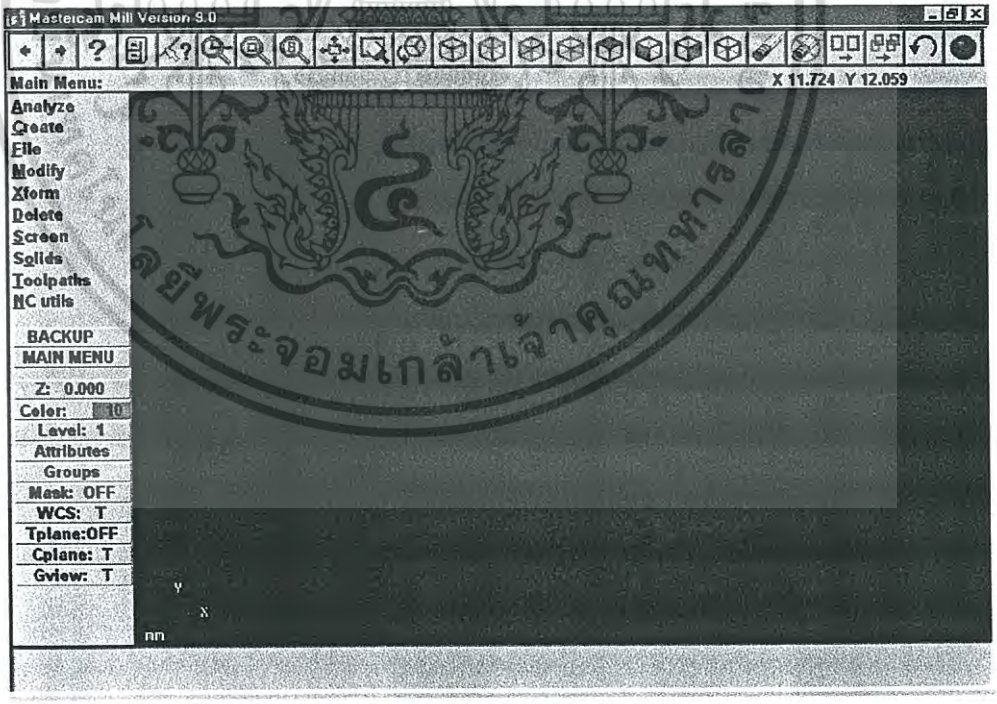
ภาพที่ 3.1 แสดงหน้าต่างของโปรแกรม Auto CAD R14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 โปรแกรม Master CAM

การใช้คอมพิวเตอร์เพื่อช่วยจัดการกับกระบวนการผลิตในโรงงานอุตสาหกรรม โดยอาจควบคุมตั้งแต่การวางแผนจนกระทั่งการจัดการหลังการผลิต ซึ่งกระบวนการของ CAM อาจแบ่งออกเป็น 2 ส่วนหลักๆ คือ

- การใช้คอมพิวเตอร์ช่วยในการผลิตโดยตรง เป็นลักษณะการใช้คอมพิวเตอร์ในงานตรวจสอบ โดยระบบคอมพิวเตอร์ที่ใช้เชื่อมโยงกับกระบวนการผลิตจะทำหน้าที่ตรวจสอบกระบวนการผลิตหรือเก็บข้อมูลจากกระบวนการผลิต การใช้คอมพิวเตอร์เพื่อทำการผลิตโดยตรง โดยการนำข้อมูลระบบ CAD มาช่วยในการควบคุมอุปกรณ์ในการผลิต เช่น เครื่องกัดที่อาศัยคำสั่งเชิงตัวเลข (Numerical Control Machine) หรือ NC Machine Tool
- การใช้คอมพิวเตอร์ช่วยในการผลิตทางอ้อม งานลักษณะนี้จะเป็นงานที่สนับสนุนการผลิต ซึ่งไม่ต่อเชื่อมกับระบบคอมพิวเตอร์โดยตรง แต่อาจจะนำข้อมูลมาประมวลผลสรุป วางแผน เช่น งานเกี่ยวกับการวางแผน การจัดการเกี่ยวกับการจัดซื้อวัตถุดิบ การจัดการในโรงงาน เป็นต้น



ภาพที่ 3.2 แสดงหน้าต่างของโปรแกรม MasterCAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

3.3 ส่วนการแปลงข้อมูล G-Code เป็นระบบพิกัด X-Y-Z

ในส่วนของการแปลงข้อมูล G-Code เป็นระบบพิกัด X-Y-Z นั้น จะใช้โปรแกรม Visual Basic เป็นเครื่องมือในการสร้างโปรแกรมแปลงข้อมูล และอาศัยทฤษฎีทางคณิตศาสตร์มาประยุกต์ใช้ร่วมด้วย

3.3.1 การเขียนโปรแกรมด้วย Visual Basic

3.3.1.1 คอนโทรลภายในของ Visual Basic

Visual Basic เป็นเครื่องมือที่ช่วยพัฒนาโปรแกรมแอปพลิเคชั่นสำหรับวินโดว์ตัวแรก ที่ประสบความสำเร็จเป็นอย่างมากทั้งนี้เนื่องมาจากแนวความคิดที่จะนำเอาความสามารถของคอนโทรลมาใช้ในการออกแบบโปรแกรม เพราะคอนโทรลเป็นเครื่องมือที่ช่วยลดความซับซ้อนในการเขียนโค้ดลงไปได้มากทีเดียวและนอกจากนี้คอนโทรลยังมีส่วนที่แสดงผลเพื่อสื่อความหมายของการทำงานระหว่างคอนโทรลและผู้ใช้ได้อีกด้วย ส่วนการใช้งานก็ไม่มีความซับซ้อนเพียงแต่ ผู้อ่านทำการเชื่อมต่อคอนโทรลเข้ากับสภาพแวดล้อมของ Visual Basic จากนั้นก็สามารถที่จะนำมาเพิ่มลงในฟอร์มได้ทันที สำหรับ Visual Basic ได้มีการแบ่งคอนโทรลออกเป็น 4 กลุ่มหลักๆ ที่สำคัญดังนี้

1. คอนโทรลภายใน (Intrinsic Control) เช่น ComboBox , Command Button หรือ PictureBox เป็นต้น ซึ่งเป็นคอนโทรลที่ถูกสร้างลงในสภาพแวดล้อมของ vb.exe ดังนั้นทุกครั้งที่ผู้อ่านโหลด Visual Basic คอนโทรลเหล่านี้ออกจากแถบกล่องเครื่องมือได้เลย ดังนั้นจึงจัดได้ว่าเป็นคอนโทรลมาตรฐาน(Standard Control) กลุ่มหนึ่งของ Visual Basic
2. คอนโทรลมาตรฐาน (Standard Control) เป็นคอนโทรล ActiveX ที่ถูกสร้างเป็นไฟล์ .ocx ที่แยกออกมาต่างหาก เช่น Dbgrid (Apex data-bound grid) , MSFlexGrid หรือ CommonDialog เป็นต้น ดังนั้นก่อนที่จะสามารถใช้งานคอนโทรลในกลุ่มนี้ได้ เราต้องทำการเชื่อมต่อไฟล์ .ocx เหล่านี้เข้ากับสภาพแวดล้อมของ Visual Basic เสียก่อนโดยใช้คำสั่ง Components ในเมนู Project เช่นเดียวกัน
3. คอนโทรลร่วมวินโดวส์ (Windows Common Control) เป็นคอนโทรล ActiveX ที่ถูกสร้างเป็นไฟล์ .ocx ที่ต้องใช้ร่วมกับไฟล์ .dll ของวินโดวส์ เช่น RichTextBox , Slider หรือ Statusbar เป็นต้น เช่นเดียวกับคอนโทรลมาตรฐาน เพียงแต่คอนโทรลในกลุ่มนี้ได้ถูกจัดเป็นคอนโทรลพื้นฐานของวินโดวส์ 95 โดยที่คอนโทรลร่วมกับวินโดวส์จะถูกจัดเก็บลงในไฟล์ conctl32.ocx และ conct232.ocx
4. คอนโทรล ActiveX รุ่นมืออาชีพ (Professional ActiveX Control) เป็นคอนโทรล ActiveX ที่ถูกสร้างเป็นไฟล์ .ocx เช่นเดียวกับคอนโทรลมาตรฐาน เช่น MSComm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50253

(Communication) , MapiMessages (MAPI message) หรือ MMControl (Multimedia MCI) เป็นต้น แต่คอนโทรลในกลุ่มนี้ได้ถูกสร้างและแจกจ่ายมากับ Visual Basic รุ่น Professional และ Enterprise เท่านั้น

3.3.1.2 คุณสมบัติแสดงค่าของคอนโทรลที่สำคัญ

คอนโทรลทั้งหมดที่มากับ Visual Basic ไม่ว่าจะเป็นคอนโทรลภายในหรือ ActiveX จะมีคุณสมบัติตัวหนึ่งที่ถูกนำมาใช้สำหรับการกำหนดค่า (Value) หรืออ่านจากค่าคอนโทรล และคุณสมบัตินี้ได้ถูกกำหนดให้เป็นคุณสมบัติปกติ (Default) ของคอนโทรล โดยในการเขียนโค้ดเราสามารถใช้เพียงชื่อของคอนโทรล (Control Name) โดยไม่ต้องกำหนดคุณสมบัติปกติของทุกๆ คอนโทรลได้โดยไม่เกิดข้อผิดพลาด เช่น คอนโทรล TextBox ก็จะมีคุณสมบัติ Text เป็นคุณสมบัติปกติของคอนโทรล สำหรับคุณสมบัติปกติของคอนโทรลที่สำคัญมีดังต่อไปนี้

ตารางที่ 3.1 แสดงชื่อคอนโทรลที่สำคัญ

คอนโทรล	คุณสมบัติ
Command Button	Value
DATA	Caption
MSFlexGrid	Text
Label	Caption
TextBox	Text
Timer	Enable

3.3.1.3 การแบ่งกลุ่มของคอนโทรลภายใน

เราจะแยกตามวัตถุประสงค์ของการใช้งานได้ทั้งหมด 4 กลุ่ม

1. คอนโทรลภายในทั่วไป ประกอบด้วยคอนโทรลที่แสดงผลในลักษณะของการเลือกตอบหรือเลือกรายการ เช่น Checkbox , OptionBox หรือ ListBox เป็นต้น
2. คอนโทรลภายในด้านระบบไฟล์ ประกอบไปด้วยคอนโทรลที่ทำหน้าที่ในการติดต่อหรือ แสดงผลระบบไฟล์ (รวมทั้งไดร์ฟ และ ไดเร็คทอรี่ด้วย) ของวินโดวส์ เช่น FileListBox หรือ DirListBox เป็นต้น
3. คอนโทรลภายในด้านกราฟิก ประกอบด้วยคอนโทรลที่ทำหน้าที่ด้านการแสดงผลกราฟิกด้วยวิธีกราฟิกคอนโทรล หรือ ฟังก์ชันวินโดวส์ API หรือไฟล์กราฟิกในรูปแบบต่างๆ เช่น PictureBox , Shape หรือ Image เป็นต้น
4. คอนโทรลภายในด้านเวลา ซึ่งจะมีอยู่คอนโทรลเดียวได้แก่ Timer ซึ่งมีหน้าที่สร้างเหตุการณ์ที่ตอบสนองเป็นครั้งๆตามช่วงเวลาที่ถูกกำหนด

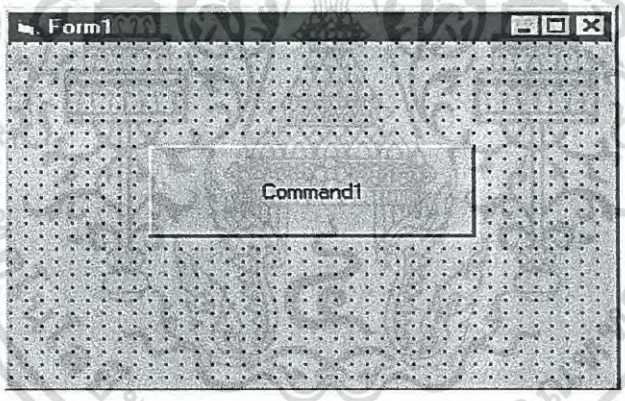
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.4 คอนโทรลภายใน

คอนโทรลภายในก็จะเป็นคอนโทรลพื้นฐานที่ถูกนำไปใช้งานมากที่สุดเพราะจะเป็นกลุ่มคอนโทรลที่ช่วยในการสื่อสารแบบสองทางหรือรับเลือกเงื่อนไขจากผู้ใช้ดังเช่นทุกๆ แอปพลิเคชั่นจะใช้คอนโทรล Command Button สำหรับให้ผู้ใช้เลือกที่ยอมรับ (OK) ยกเลิก (Cancel) หรือ อื่นๆ ตามข้อกำหนดของแต่ละแอปพลิเคชั่นเป็นต้นซึ่งคอนโทรลภายในทั่วไปจะประกอบไปด้วยคอนโทรลต่างๆ ดังต่อไปนี้ คอนโทรล Command Button

3.3.1.5 คอนโทรล Command Button

คอนโทรล Command Button จะเป็นคอนโทรลที่ถูกนำไปใช้งานมากที่สุด เพราะกำหนดให้ผู้ใช้งานเลือก OK หรือ Cancel นั้น เรามักจะใช้คอนโทรล Command Button เป็นส่วนใหญ่ ดังนั้นจึงถือว่าเป็นคอนโทรลที่พื้นฐานที่สุดของ Visual basic เนื่องจากคอนโทรลนี้เป็นปุ่มสำคัญที่ใช้งานในรูปแบบของการคลิกเพื่อยืนยัน ดังนั้นเรียกคอนโทรล Command Button ได้อีกอย่างว่า pushbutton ในขณะออกแบบคอนโทรล Command Button ที่วางลงบนฟอร์มจะมีลักษณะดังตัวอย่างในภาพนี้



ภาพที่ 3.3 แสดงคอนโทรล ControlButton

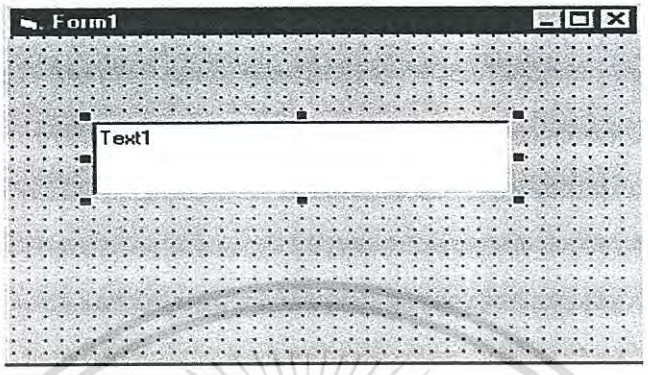
เราสามรถแก้ไขข้อความที่แสดงผลในคอนโทรลนี้โดยการแก้ไขข้อความในคุณสมบัติ Caption ของคอนโทรลในหน้าต่างคุณสมบัติหรือแก้ไขโค้ดในแอปพลิเคชั่นก็ได้

3.3.1.6 คอนโทรล TextBox

คอนโทรล TextBox มักจะถูกนำไปใช้ทุกๆ ฟอร์มที่มีการรับการกรอกข้อความจากผู้ใช้เนื่องจากคอนโทรลนี้ทำหน้าที่แสดงข้อมูล (โดยผ่านทางคุณสมบัติ Text) ในคอนโทรลและยังอนุญาตให้ผู้ใช้สามารถแก้ไขตัวอักษรต่างๆ ของคุณสมบัตินี้ได้ด้วยเช่นกันนอกจากนี้แล้วคอนโทรล TextBox ยังได้รวมเอาความสามารถหลายๆด้านของคอนโทรล Label เช่น สามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

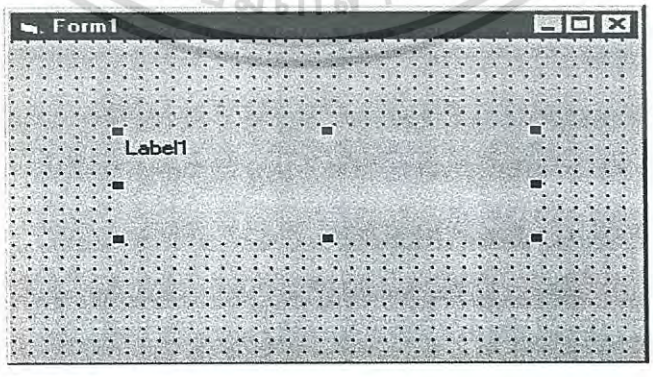
แสดงข้อความมากกว่า 1 บรรทัด ความสามารถด้าน DDE (Dynamic Data Exchange) นอกจากนี้ยังสามารถถูกนำไปใช้ในลักษณะของการกรอกรหัสผ่าน (Password) ได้อีกด้วยดังภาพ



ภาพที่ 3.4 แสดงคอนโทรล Text Box

3.3.1.7 คอนโทรล Label

คอนโทรล Label เป็นคอนโทรลในลักษณะของกราฟิกที่ถูกใช้งานด้านการแสดงผลข้อความบนฟอร์มเหมือนกับผู้อ่านได้นำป้ายข้อความอย่างหนึ่งไปวางไว้บนฟอร์ม เพื่อใช้ในการสื่อข้อความกับผู้ใช้และคอนโทรลนี้ผู้ใช้ไม่สามารถแก้ไขได้โดยตรงด้วยวิธีการคีย์หรือใช้เมาส์ในขณะรันแอพพลิเคชั่น นอกเสียจากภายในแอปพลิเคชั่นจะมีการเขียนโค้ดสำหรับ แก้ไขข้อความในคอนโทรลโดยการแก้ไขค่าคุณสมบัติ Caption เท่านั้น และนอกจากนี้ Label ยังเป็นคอนโทรลที่มีความสามารถด้าน DDE (Dynamic Data Exchange) อีกด้วย ในขณะออกแบบเราสามารถเพิ่มคอนโทรลลงในฟอร์ม หรือตัวบรรจุอื่นๆ ก็จะปรากฏหน้าตาของคอนโทรลดังภาพ



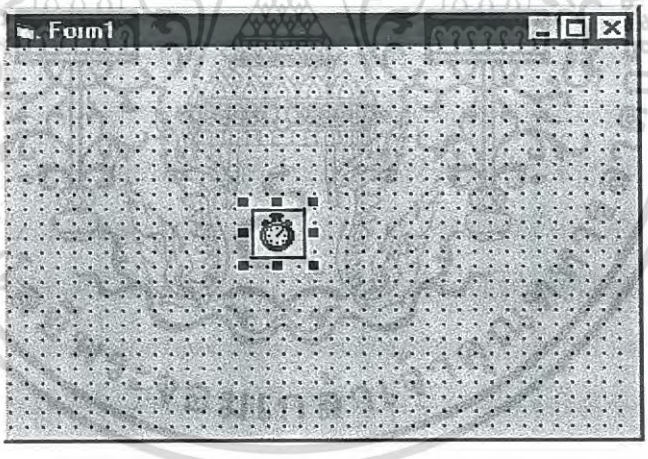
ภาพที่ 3.5 แสดงคอนโทรล Label

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการควบคุมพฤติกรรมของคอนโทรล เราสามารถกระทำได้โดยการกำหนดค่าต่างๆ ให้กับคุณสมบัติของคอนโทรล ซึ่งการแก้ไขของคุณสมบัติเราสามารถกระทำได้ทั้งในขณะออกแบบโดยการแก้ไขค่าในหน้าต่างคุณสมบัติ และรับแอปพลิเคชั่นโดยการเขียนโค้ดแก้ไขค่า คุณสมบัติแต่ก็มีบางคุณสมบัติที่ไม่สามารถแก้ไขได้ในขณะรันแอพพลิเคชั่น เช่น Name เป็นต้น

3.3.1.8 คอนโทรล Timer

คอนโทรล Timer เป็นคอนโทรลที่ใช้ในการควบคุมและจัดการเหตุการณ์ ด้านเวลา ซึ่งเทียบได้กับประโยค ON TIME GOTO ของ QuickBasic โดยเราสามารถเขียนโค้ด เพื่อทำงานใดๆ เมื่อช่วงเวลาผ่านไปตามค่าที่กำหนด เช่น ทำการปรับการแสดงผลของฟอร์มทุกๆ 1 นาที เป็นต้น โดยที่คอนโทรลนี้จะตอบสนองเหตุการณ์เพียงเหตุการณ์เดียวเท่านั้น แต่เราสามารถกำหนดให้แต่ละฟอร์มมีคอนโทรล Timer มากกว่า 1 คอนโทรล เนื่องจากคอนโทรล Timer เป็นคอนโทรลที่ทำงานตามนาฬิกาของระบบดังนั้นมันจึงถูกควบคุมโดยตัวของระบบเอง สำหรับวินโดวส์ 95 และ NT ในทางปฏิบัติจะไม่มีการจำกัดจำนวนของคอนโทรล Timer ในแต่ละฟอร์ม ดังนั้นเราจึงสามารถใช้งานคอนโทรล Timer พร้อมๆ กันครั้งละหลายๆคอนโทรล ได้อย่างไม่จำกัด



ภาพที่ 3.6 แสดงคอนโทรล Timer

ในขณะออกแบบคอนโทรล Timer ที่วางลงบนฟอร์มให้กับฟอร์ม ก็จะมีลักษณะ ดังภาพที่ 3.8 และเมื่อรันแอปพลิเคชั่น คอนโทรลนี้จะไม่ถูกแสดงผลแต่จะมีการทำให้เกิด เหตุการณ์ Timer ทุกครั้งที่ช่วงเวลาครบตามค่าที่ได้กำหนดให้กับคุณสมบัติ Interval ของคอนโทรล Timer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.9 คอนโทรลด้านฐานข้อมูล

แอปพลิเคชั่นที่จะต้องมีการเข้าถึงไฟล์ฐานข้อมูล ตัวแปลภาษาที่เหมาะสมกับการสร้างแอปพลิเคชั่นเหล่านี้ จึงต้องมีเครื่องมือที่สามารถสนับสนุนการจัดการฐานข้อมูลอย่างง่ายและมี ประสิทธิภาพ Visual Basic เป็นตัวแปรภาษาที่มีการสนับสนุนระบบจัดการฐานข้อมูลในรูปแบบ Microsoft Access โดยอาศัย JET Database Engine ซึ่งเป็นเครื่องมือที่ให้โปรแกรมเมอร์สามารถ จัดการกับฐานข้อมูลได้ 2 วิธีดังนี้

1. คอนโทรลด้านฐานข้อมูล (Data Control)

การเข้าถึงฐานข้อมูลด้วยคอนโทรลด้านฐานข้อมูลจะเป็นวิธีที่ง่ายที่สุดในการเขียนโปรแกรมเพราะเราเพียงแต่กำหนดไฟล์ฐานข้อมูลฟิลด์สำหรับแต่ละคอนโทรลและคอนโทรล Data เท่านั้น คอนโทรลเหล่านี้จะจัดการสร้าง การแสดงผล การแก้ไขฟิลด์ต่างๆ ของฐานข้อมูลให้โดยอัตโนมัติหรือตามที่ถูกกำหนดในคุณสมบัติต่างๆ สำหรับความสามารถโดยทั่วไปของคอนโทรลด้านฐานข้อมูลมีรายละเอียดดังต่อไปนี้

- สามารถสร้างตัวแปร Recordset โดยอ้างอิงกับตัวแปร Recordset ที่สนับสนุนโดยคอนโทรลด้านฐานข้อมูล
- แก้ไขโครงสร้างของฐานข้อมูล เช่นการแก้ไขตาราง (Table) ฟิลด์ (Field) หรือ ดัชนี (Index) เป็นต้น
- ค้นหาหรือสืบค้นข้อมูลจากฟิลด์ที่ถูกกำหนดของฐานข้อมูล

คอนโทรลที่สนับสนุนการติดต่อกับฐานข้อมูล (Bound Control) นั้น เราสามรถสังเกตได้จากคอนโทรลที่มีคุณสมบัติ Datafield, Dtatchange หรือ DataSource เป็นต้น ซึ่งคอนโทรล ที่มีคุณสมบัติเหล่านี้ก็จะเป็นคอนโทรลด้านฐานข้อมูลทั้งสิ้น เช่น CheckBox, PictureBox, Data, ProgressBar, RichTextBox หรือ MaskedEdit เป็นต้น

2. Data Access Object (DAO)

DAO เป็นโมเดลของคลาสของออบเจ็กต์ที่สนับสนุนการจัดการในเรื่องฐานข้อมูลในระบบ RelationalDatabase ซึ่งก็จะประกอบด้วยคุณสมบัติ โพรซีเยอร์เหตุการณ์และก็วิธีเพื่อช่วยในการสร้าง แก้ไข จัดเก็บค้นหาและลบฐานข้อมูลโดยในการจัดการกับฐานข้อมูลในทางปฏิบัติ Visual Basic ก็จะอาศัยความสามารถของ JET Database Engine (JET) ซึ่งเป็นเอนจิ้นท์ที่ช่วยในการจัดการฐานข้อมูลด้วยกายภาพสำหรับ Visual Basic ฐานข้อมูลในรูปแบบ Microsoft Access จะถูกกำหนดให้เป็นฐานข้อมูลแม่แบบ (Native Database Format) ที่ถูกจัดการโดย JET ซึ่งกลุ่มชนิดของฐานข้อมูลที่ Visual Basic สนับสนุนโดยผ่านทาง DAO และ JET สามารถแบ่งออกได้เป็น 3 กลุ่ม ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ฐานข้อมูลของ Visual Basic หมายถึง ฐานข้อมูลในรูปแบบของ Microsoft Access (.mdb)ซึ่ง JET สามารถที่จะจัดการได้โดยตรง ดังนั้นจึงเป็นรูปแบบของฐานข้อมูลที่เอ็นจิ้นท์ JET สามารถจัดการได้รวดเร็วที่สุด
- ฐานข้อมูลภายนอก หมายถึง ฐานข้อมูลในรูปแบบ Indexed Sequential Access Method (ISAM) ซึ่งเป็นรูปแบบฐานข้อมูลสำหรับโปรแกรมด้านฐานข้อมูลระดับเดสก์ทอปในปัจจุบันนั่นเอง เช่น dBase IV, FoxPro, Paradox เป็นต้น ซึ่งผู้อ่านสามารถจัดการกับฐานข้อมูลในรูปแบบ ISAM ได้โดยตรงจาก Visual Basic เท่านั้น
- ฐานข้อมูล ODBC หมายถึง ฐานข้อมูลที่ทำการติดติดโดยผ่านทาง Open Database Connectivity (ODBC) ซึ่งเป็น มาตรฐานการจัดการกับฐานข้อมูลในรูปแบบที่แตกต่างกันของไมโครซอฟท์ เช่น ฐานข้อมูลชนิด ไคลเอนต์ เซิร์ฟเวอร์ Microsoft SQL Server เป็นต้น ดังนั้นเราจึงสามารถที่จะสร้าง แอปพลิเคชั่น ชนิด ไคลเอนต์ เซิร์ฟเวอร์ ด้านฐานข้อมูลได้โดยตรงจาก Visual Basic โดยมาตรฐาน ODBC นี้เอง

3.3.1.10 Microsoft JET Database Engine

ในการเขียนโปรแกรมเพื่อจัดการกับฐานข้อมูลของ Visual Basic นั้นเราสามารถใช้อินเตอร์เฟสของ DAO ในการกำหนดวิธีการจัดการกับฐานข้อมูลที่ต้องการแต่ในทางปฏิบัติจริงๆ DAO ก็จะต้องมีการส่งคำสั่งต่อเนื่องไปยัง JET ที่มากับ Visual Basic 5.0 ก็จะเป็นเวอร์ชัน 3.5 ซึ่งจะเป็นเอ็นจิ้นท์ระดับ 32 บิทที่ทำงานภายใต้ระบบปฏิบัติการวินโดวส์ 32 บิทเท่านั้นโดยจะมีความเสถียร ประสิทธิภาพและสมรรถนะการประมวลผลรวดเร็วกว่า JET เวอร์ชันที่ผ่านๆ มาก

- User Interface ผู้อ่านต้องเขียนโค้ดโดยอาศัย DAO หรือ คอนโทรลด้านฐานข้อมูล เพื่อจัดการเก็บฐานข้อมูลที่ต้องการโดยแสดงผลฟิลด์ต่างๆ ของฐานข้อมูลนั้น ผู้อ่านก็ต้องกระทำโดยอาศัยคอนโทรล หรือ ฟอร์มสำหรับการสร้าง แก้ไข ลบทิ้ง จัดเก็บ หรือ ค้นหาข้อมูลในฐาน ข้อมูล Visual Basic ก็กระทำผ่านทางเอ็นจิ้นท์ JET อีกทอดหนึ่ง

- Database Engine JET เป็นไฟล์ไลบรารี (. Dll) ที่ประกอบด้วยฟังก์ชันที่ช่วยในการ จัดเก็บกับไฟล์ฐานข้อมูลในระดับกายภาพ ซึ่งในการทำงาน JET จะแปลงการร้องขอจาก Visual Basic เพื่อที่จะดำเนินการกับฟิลด์ต่างๆ ของไฟล์ฐานข้อมูล และรายงานข้อมูลกลับไปยังโค้ดส่วนที่ร้องขอ (ถ้าหากมี) และนอกจากนี้ JET ยังสนับสนุนการค้นหาหรือประมวลผลฐาน ข้อมูลโดยอาศัยภาษา SQL(Structured Query Language) อีกด้วย

- Data Store ไฟล์ที่จัดเก็บฐานข้อมูลและรายละเอียดทั้งหมดของฐานข้อมูล โดยที่ไฟล์ข้อมูลแม่แบบของ Visual Basic ก็จะจัดเก็บในรูปแบบของ Microsoft Access โดยมีนามสกุล .mdb โดยที่ข้อมูลที่จัดเก็บในไฟล์ฐานข้อมูลจะถูกจัดระเบียบโดยรูปแบบขอวฐานข้อมูลแต่ละประเภท ซึ่งเอ็นจิ้นท์ JET จะทำหน้าที่จัดการกับข้อมูลให้โดยอัตโนมัติ ซึ่งเราสามารถที่จะจัดการกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลหลากหลายชนิดโดยไม่จำเป็นต้องทราบถึงโครงสร้างของไฟล์ฐานข้อมูลชนิดนั้นๆ เลยซึ่งก็เป็นจุดเด่นอีกประการของระบบฐานข้อมูลของ Visual Basic

คอนโทรลด้านฐานข้อมูลเป็นคอนโทรลที่ใช้ในการเข้าถึงฐานข้อมูลคอมพิวเตอร์ ซึ่งหลักๆ สามารถแบ่งออกได้เป็น 3 กลุ่มดังนี้

1. คอนโทรล Data เป็นคอนโทรลหลักที่ใช้ในการควบคุมการติดต่อระหว่างคอนโทรลด้านฐานข้อมูลกับฐานข้อมูลโดยที่คอนโทรล Data จะทำหน้าที่ควบคุมการเข้าถึงฐานข้อมูล เช่น การเคลื่อนที่ไปยังเรคคอร์ด การเปิดปิด การจัดเก็บฐานข้อมูล เป็นต้น

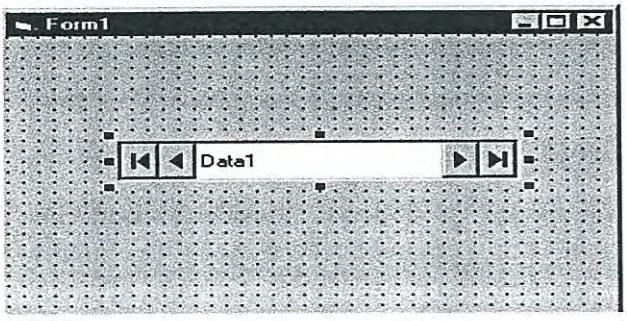
2. คอนโทรลภายใน Data-Aware เป็นคอนโทรลภายในของ visual basic ที่สนับสนุนคุณสมบัติการแสดงผลข้อมูลของฟิลด์หนึ่งๆ ของฐานข้อมูล เช่น CheckBox , PictureBox หรือ TextBox เป็นต้น ซึ่งคอนโทรลภายใน Dtat-Aware จะแตกต่างกับคอนโทรล Data-Bound ที่คอนโทรลภายในData-Awareจะเชื่อมต่อเข้ากับฟิลด์ของฐานข้อมูลคอนโทรลละ 1 ฟิลด์เท่านั้น

3. คอนโทรล Data-Bound เป็นคอนโทรล ActiveX ที่ถูกออกแบบพิเศษเพื่อให้สามารถเชื่อมต่อกับ Record (ประกอบด้วยตั้งแต่ 1 ฟิลด์ขึ้นไป) ของฐานข้อมูล เช่น Dblist, DbCombo, DbGrid และ MSFrexGrid ซึ่งคอนโทรล Dtat-Bound จะสามารถเชื่อมต่อกับหลายๆ เรคคอร์ดได้ในเวลาเดียวกันทั้งนี้เพื่อขยายขีดความสามารถในการจัดการฐานข้อมูลนั่นเอง

3.3.1.11 คอนโทรล Data

คอนโทรล Data เป็นคอนโทรลที่ใช้ในการเข้าถึงฐานข้อมูลคอมพิวเตอร์และทำการ เชื่อมต่อการแสดงผลข้อมูลของแต่ละฟิลด์ในฐานคอมพิวเตอร์ข้อมูลเข้ากับคอนโทรลด้านฐานข้อมูล (Data-Bound or Data-Aware Control) โดยเมื่อมีการเคลื่อนที่ไปยังเรคคอร์ดใดๆ ในฐานข้อมูลด้วยคอนโทรล Data ข้อมูลที่ถูกแก้ไขในคอนโทรลด้านฐานข้อมูลที่เชื่อมต่อกับคอนโทรล Data ซึ่งเป็นเรคคอร์ดปัจจุบันในขณะนั้นก็จะถูกจัดเก็บลงในฐานข้อมูลโดยอัตโนมัติ แล้วจึงเคลื่อนที่ไปยังเรคคอร์ดถัดไปทันที แต่เนื่องจากคอนโทรล Data สามารถที่จะจัดเก็บข้อมูลที่ถูกแก้ไขให้โดยอัตโนมัติ ดังนั้นถ้าหากเราต้องการตรวจสอบความถูกต้องของข้อมูลก่อนที่จะถูกจัดเก็บโดยคอนโทรล Data ก็สามารถกระทำได้โดยการเขียนโค้ดเพื่อตรวจสอบความถูกต้องของข้อมูลในโพรซีเยอร์ของเหตุการณ์ Validate ในขณะออกแบบคอนโทรล Data ที่วางลงบนฟอร์มจะมีลักษณะดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

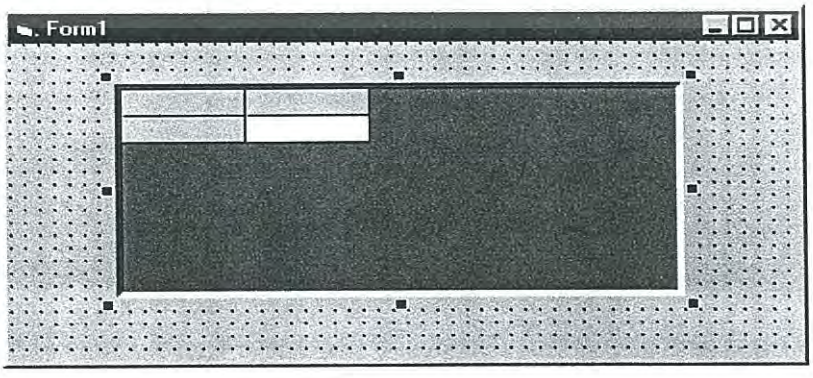


ภาพที่ 3.7 แสดงคอนโทรล Data

3.3.1.12 คอนโทรล MsflexGrid

คอนโทรล MsflexGrid เป็นวัตถุที่แสดงผลค่อนข้างมีประสิทธิภาพ เมื่อเข้าสู่โปรแกรม ถ้ามองไม่เห็นวัตถุควบคุมเอ็มเอสเฟลกกริดในกล่องเครื่องมือ ให้เพิ่มวัตถุควบคุมนี้เข้าไปโดยใช้เมนู Project เป็นวัตถุที่สามารถแสดงผลข้อมูลออกมาเป็นแถวและคอลัมน์ เช่นเดียวกับตาราง การทำงานของเอกซ์เซล ข้อมูลอาจจะเป็นข้อความหรือตัวเลขก็ได้ และยังมีความสามารถที่จะเชื่อมเซลล์เข้าด้วยกัน เรียงลำดับข้อมูลตามแนวคอลัมน์ได้อีกด้วย ความสามารถของคอนโทรล FlexGrid

- ข้อมูลที่แสดงจะสามารถอ่านได้อย่างเดียวไม่สามารถแก้ไขได้
- สามารถย้ายคอลัมน์และแถวได้
- สามารถจัดกลุ่มข้อมูลที่เหมือนกันเข้าเป็นเซลเดียวได้
- แต่ละเซลในคอนโทรลสามารถแสดงได้ทั้งข้อความและรูปภาพ
- ใช้งานร่วมกับ Data Control ในการแสดงข้อมูลฐานข้อมูลได้
- มีฟังก์ชันในการค้นหา และแทนที่ข้อความในคอนโทรล
- สามารถแก้ไขค่าของข้อมูลได้จากคำสั่งในโปรแกรม



ภาพที่ 3.8 แสดงคอนโทรล MsflexGrid

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.13 คอนโทรลคอนโทรล MSComm (Communication)

เนื่องจากในปัจจุบันมีการใช้งานมาตรฐานการเชื่อมต่อแบบ RS 232Cกันอย่างแพร่หลาย ซึ่งเป็นมาตรฐานถูกกำหนดโดย EIA ซึ่งเป็นองค์กรอุตสาหกรรมอิเล็กทรอนิกส์ ของสหรัฐอเมริกา โดยแบ่งการเชื่อมต่อออกเป็น 2 ลักษณะ คือ DTE (Data Terminal Equipment) และ DCE (Data Communication Equipment) ซึ่งโดยปกติ DTE จะต้องต่อเข้ากับ DCE เสมอ เช่น การต่อเครื่องคอมพิวเตอร์ (อุปกรณ์ DTE) เข้ากับอุปกรณ์โมเด็ม (อุปกรณ์ DCE)เป็นต้น

พอร์ทอนุกรม RS 232C จะเป็นพอร์ทของเครื่องคอมพิวเตอร์ที่มีขาต่อ (Connector) ทั้งประเภท 9 และ 25 ขาและเรียกว่า พอร์ท COM1: และ COM2: นั่นเอง ในความจริงพอร์ทอนุกรมไม่ได้ถูกควบคุมโดยตรงจาก CPU บนเมนบอร์ด แต่การสื่อสารทั้งหมดจะถูกเก็บโดยชิป (Universal Asynchronous Receiver/Transmitter; UART) อีกทีหนึ่ง ซึ่งในปัจจุบันเบอร์ที่ใช้กันมากที่สุดก็คือเบอร์ 16550C ซึ่งเป็นเวอร์ชันที่ได้รับการแก้ไขข้อผิดพลาดแล้ว ซึ่งชิป UART นี้จะทำหน้าที่ในการรับและส่งข้อมูลดังต่อไปนี้

การส่งข้อมูล(Data transmission)

- รับตัวอักษรจากเครื่องคอมพิวเตอร์
- แปลงตัวอักษรให้เป็นสายข้อมูลแบบบิต (เราเรียกว่าขบวนการ Serialization)
- สร้างเฟรมข้อมูลโดยการเพิ่มบิตที่จำเป็นสำหรับการสื่อสารและการตรวจสอบเช่น บิต Start , Stop และ Parity เป็นต้น
- ส่งผ่านเฟรมข้อมูลที่สร้างขึ้นมาแล้วจากขั้นตอนที่ผ่านมาด้วยความเร็วของโมเด็มหรือพอร์ทอนุกรม(Baud Rate)
- แสดงสถานะความพร้อมที่จะรับข้อมูลตัวอักษรถัดไปให้กับเครื่องคอมพิวเตอร์ในการรับข้อมูล (Data Receiver)
- รับตัวอักษรจากตัวอินเตอร์เฟส
- ตรวจสอบความถูกต้องของเฟรมข้อมูลตามมาตรฐานเฟรมที่กำหนดโดยถ้าหากเฟรมข้อมูลมีรูปแบบที่มาถูกต้องก็จะมีการแจ้งข้อผิดพลาดทันที
- ตรวจสอบความถูกต้องของพาริตี้
- แปลงสายข้อมูลแบบบิตให้เป็นตัวอักษร
- ส่งตัวอักษรให้กับเครื่องคอมพิวเตอร์
- แสดงสถานะความพร้อมที่รับข้อมูลตัวอักษรถัดไปให้กับอินเตอร์เฟส

คอนโทรล MSComm เป็นคอนโทรลตัวหนึ่งที่ช่วยในการติดต่อกับพอร์ทอนุกรม (Serial Port) ซึ่งผู้อ่านสามารถรับส่งข้อมูลผ่านทางพอร์ทอนุกรมได้ด้วยคอนโทรลนี้ เช่น การติดต่อผ่านทางโมเด็ม หรือ ติดต่อโดยตรงกับบอร์ดอิเล็กทรอนิกส์ เป็นต้น ซึ่งคอนโทรล MSComm ที่มา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนโทรล MSComm เป็นคอนโทรลตัวหนึ่งที่ช่วยในการติดต่อกับพอร์ทอนุกรม (Serial Port) ซึ่งผู้อ่านสามารถรับส่งข้อมูลผ่านทางพอร์ทอนุกรมได้ด้วยคอนโทรลนี้ เช่น การติดต่อผ่านทางโมเด็ม หรือ ติดต่อโดยตรงกับบอร์ดอิเล็กทรอนิกส์ เป็นต้น ซึ่งคอนโทรล MSComm ที่มากับ Visual Basic จะเป็นคอนโทรลที่ทำงานโดยมีการตอบสนองต่อเหตุการณ์แบบ Event-Driven นั่นก็คือ คอนโทรลจะทำหน้าที่ตรวจสอบการเกิดขึ้นหรือร้องขอให้เกิดเหตุการณ์ต่างๆ กับพอร์ทอนุกรมโดยอัตโนมัติและจะมีการแจ้งเตือนให้ผู้อ่านได้รับทราบโดยผ่านโพรซีเยอร์เหตุการณ์ เช่นเดียวกับคอนโทรลทั่วไปของ Visual Basic นั่นเอง ดังนั้นในการเขียนโค้ดเรา จึงไม่จำเป็นต้องสร้างโพซีเยอร์ที่ทำหน้าที่คอยตรวจสอบเหตุการณ์ต่างๆ ของพอร์ทอนุกรมซึ่งจะทำให้ง่ายต่อการทำงานเป็นอย่างมากคอนโทรลMSCommจะมีหน้าที่สำหรับการสื่อสารผ่าน พอร์ทอนุกรม 3 ประการ ดังต่อไปนี้

- หมุนหมายเลขติดต่อกับโทรศัพท์ปลายทางที่กำหนด
- ตรวจสอบการเข้ามาของข้อมูลยังพอร์ทอนุกรมโดยอัตโนมัติ
- ส่งข้อมูลตามที่กำหนดจากโปรแกรมไปยังพอร์ทอนุกรม

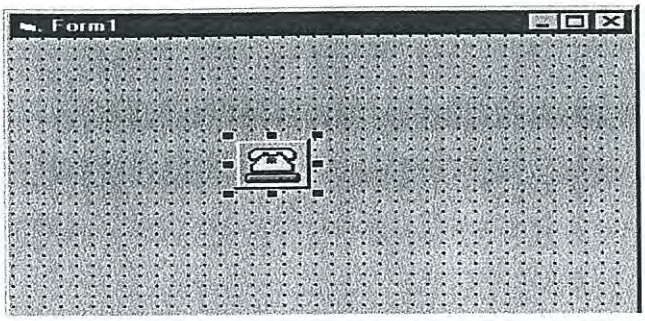
ในความจริงคอนโทรล MSComm ไม่ได้ทำหน้าที่ติดต่อกับพอร์ทอนุกรมโดยตรงแต่มันจะทำหน้าที่เรียกใช้ฟังก์ชันวินโดวส์ API ซึ่งวินโดวส์จะทำการส่งหรือรับข้อมูลผ่านพอร์ทอนุกรมโดยอาศัยไดร์วเวอร์ Comm.drv อีกทอดหนึ่ง ดังนั้นจึงสามารถสรุปสั้นๆ ได้ว่าทุกครั้งที่เรามีการเรียกใช้คอนโทรล MSComm ก็หมายถึงเรียกใช้ฟังก์ชันวินโดวส์ API ซึ่งจะถูกตีความอีกทอดหนึ่งโดยไดร์เวอร์ Comm.drv จากนั้นก็จะส่งผ่านข้อมูลที่ถูกจัดรูปแบบตามมาตรฐานการสื่อสาร (ทั้งนี้ขึ้นกับอุปกรณ์ที่ต่อเข้ากับพอร์ทอนุกรม) ให้กับดีไวซ์ไดรเวอร์อีกทอดหนึ่งนั่นเอง

การกำหนดคุณสมบัติของคอนโทรล MSComm ในขณะออกแบบเราสามารถกระทำได้อย่างสะดวกโดยการคลิ๊กไปที่ปุ่มของรายการ (Custom) ในหน้าต่างคุณสมบัติซึ่งก็จะปรากฏไดอะล็อกบ็อกซ์ Property Pages เพื่อให้เราได้ปรับแต่งค่าของคุณสมบัติของคอนโทรล MSComm สนับสนุน ซึ่งปุ่มต่างๆ ของไดอะล็อกบ็อกซ์ Property Pages มีความหมายดังนี้

- ปุ่มคำสั่ง OK ยอมรับการแก้ไขคุณสมบัติของคอนโทรล MSComm
- ปุ่มคำสั่ง Cancel ยกเลิกการแก้ไขคุณสมบัติของคอนโทรล MSComm
- ปุ่มคำสั่ง Apply อัพเดตคุณสมบัติที่ถูกแก้ไขของคอนโทรล MSComm
- ปุ่มคำสั่ง Help แสดงผล Help ของคอนโทรล MSComm

สำหรับฟอร์มหนึ่งๆ เราสามารถเพิ่มได้หลายๆ คอนโทรลMSComm ทั้งนี้ขึ้นอยู่กับความต้องการของเราในการติดต่อกับพอร์ทอนุกรมใดบ้าง สำหรับวินโดวส์ 95 และ NT 4.0 เราสามารถติดตั้งพอร์ทอนุกรมได้มากกว่า 4 พอร์ท โดยเราสามารถเพิ่มคอนโทรล MSComm ลงในฟอร์ม ก็จะปรากฏดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.9 แสดงคอนโทรล MSComm

3.3.2 ชุดคำสั่ง G-Code และทฤษฎีทางคณิตศาสตร์ที่นำมาประยุกต์ใช้

3.3.2.1 ชุดคำสั่ง G (G Function หรือ Preparatory Function)

ตัวเลขที่ตามชุดคำสั่ง G จะเป็นตัวกำหนดความหมายและการทำงานของเครื่อง CNC ซึ่งแบ่งเป็น 2 ประเภทได้แก่

1. One-shot G-Code G-Code ชนิดนี้จะมีผลให้ CNC ทำงานเฉพาะบรรทัดที่กำหนดเท่านั้น ได้แก่ G00 ซึ่งเป็นการเข้าสู่ตำแหน่งที่กำหนดอย่างรวดเร็ว

2. Model G-Code G-Code ชนิดนี้จะมีผลต่อเนื่องไปยังบรรทัดต่อๆไป จนกว่าจะมี G-Code ในกลุ่มเดียวกันมากำหนด G-Code ใหม่

ตารางที่ 3.2 แสดงชุดคำสั่ง G ที่ใช้สำหรับโครงงานนี้

G-Code	กลุ่ม	ความหมาย
G00	01	เข้าสู่ตำแหน่งที่กำหนดอย่างรวดเร็ว
G01		เคลื่อนที่ตามระยะทางที่เป็นเส้นตรง
G02		เคลื่อนที่ตามแนวเส้นโค้งตามเข็มนาฬิกา CW
G03		เคลื่อนที่ตามแนวเส้นโค้งทวนเข็มนาฬิกา CCW

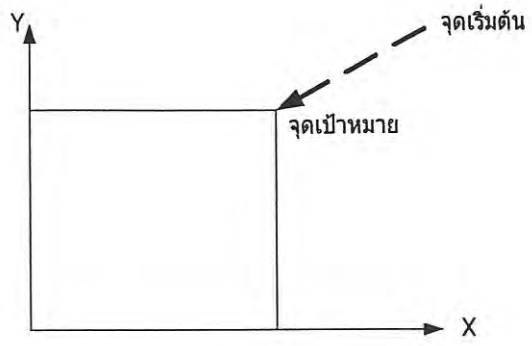
3.3.2.2 รายละเอียดเกี่ยวกับ G-Code

G-Code มีความหมายต่างๆ ดังนี้

1.) G00 (Positioning)

เคลื่อนที่ไปที่จุดที่กำหนดด้วยความเร็วสูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

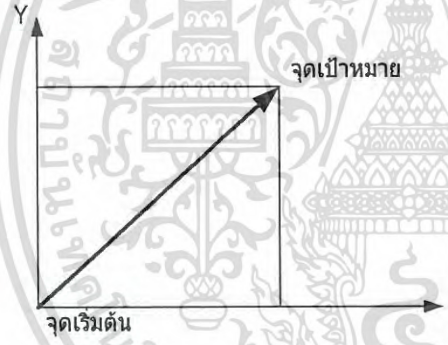


รูปแบบ G00 X__ Y__

ภาพที่ 3.10 การเคลื่อนที่ด้วยคำสั่ง G00

2.) G01 (Linear Interpolation)

เคลื่อนที่ตามระยะทางที่เป็นเส้นตรงไปยังจุด ที่กำหนดด้วยความเร็วตาม Feed Rate (Feed Function) ที่กำหนด

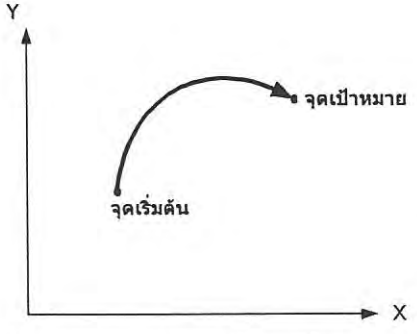


รูปแบบ G01 X_ Y_ F_

ภาพที่ 3.11 การเคลื่อนที่ด้วยคำสั่ง G01

3.) G02 (Circular Interpolation, Clock Wise; CW)

การเคลื่อนที่ตามแนวเส้นโค้งตามเข็มนาฬิกาไปตามรัศมี R (ตามรัศมีที่กำหนด) ไปยังจุดที่กำหนด



รูปแบบ G02 X__Y__Z__ I__ J__K__F__

ภาพที่ 3.12 การเคลื่อนที่ด้วยคำสั่ง G02

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.) G03 (Circular Interpolation, Counter Clock Wise: CCW)

การเคลื่อนที่ตามแนวเส้นโค้งทวนเข็มนาฬิกาไปตามรัศมี R (ตามรัศมีที่กำหนด) ไปยังจุดที่กำหนด

รูปแบบ G03 X__Y__Z__I__J__K__F__

ภาพที่ 3.13 การเคลื่อนที่ด้วยคำสั่ง G03

3.3.3 ทฤษฎีทางคณิตศาสตร์ที่นำมาประยุกต์ใช้

ในการเขียนโปรแกรมนั้นได้มีการนำเอาทฤษฎีทางคณิตศาสตร์มาประยุกต์ใช้เพื่อแก้ปัญหาในหลายๆด้านซึ่งสามารถแบ่งเป็นหัวข้อย่อยได้ดังนี้

3.3.3.1 Linear Interpolation

เมื่อมี Straight cut linear interpolation ส่วนควบคุมจะควบคุมแกนของเครื่องจักร 2 แกนหรือมากกว่านั้นพร้อมๆกันในส่วนของการตัดเชิงมุมส่วนควบคุมจะใช้ข้อมูลที่ได้รับการโปรแกรมมาเก็บไว้แล้วเพื่อคำนวณ องศาหรือความชันของการตัดชิ้นส่วนของเส้นตรง ความยาวที่เปลี่ยนแปลงจากจุดเริ่มต้นจนถึงจุดสุดท้ายจะเป็นตัวกำหนดการแบ่งเส้นและความชันของแต่ละแกน เพื่อทำการควบคุมการเคลื่อนที่ของการตัด ให้เคลื่อนที่เสมือนกับการเคลื่อนที่ไปในทิศทางเดียวกัน ในโครงงานนี้กำหนดให้ linear interpolation เคลื่อนที่ในระนาบเดียวเท่านั้น

การใช้งานอย่างอื่นของ Linear interpolation คือใช้ในการประมาณเส้นโค้ง หรือวงกลม (Circular interpolation) ซึ่งจะกล่าวถึงการประมาณเส้นโค้งในช่วงต่อไป

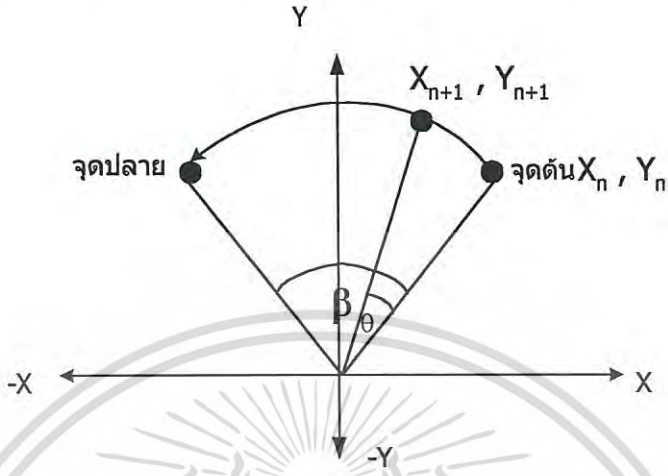
3.3.3.2 Circular Interpolation

เป็นการคำนวณสำหรับการเคลื่อนที่ที่เป็นเส้นโค้งจากจุดหนึ่งไปยังอีกจุดหนึ่ง หรือเป็นวงกลม ข้อมูลที่จำเป็นสำหรับใช้ในการคำนวณเส้นทางเดินของวงกลมได้แก่

- ทิศทางการเคลื่อนที่ ตามเข็มนาฬิกา(Clock Wise;CW) หรือ ทวนเข็มนาฬิกา(Counter Clock Wise;CCW)
- จุดเริ่มต้นของส่วนโค้ง
- จุดสิ้นสุดของส่วนโค้ง
- จุดศูนย์กลางของส่วนโค้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อได้ข้อมูลที่จำเป็นสำหรับส่วนโค้งมาหมดแล้ว จะนำข้อมูลที่มีอยู่มาทำการคำนวณจุดบนส่วนโค้งด้วยการประมาณค่าเส้นตรง หรือกล่าวได้ว่าเป็นลักษณะของการเดินแบบจุดต่อจุด



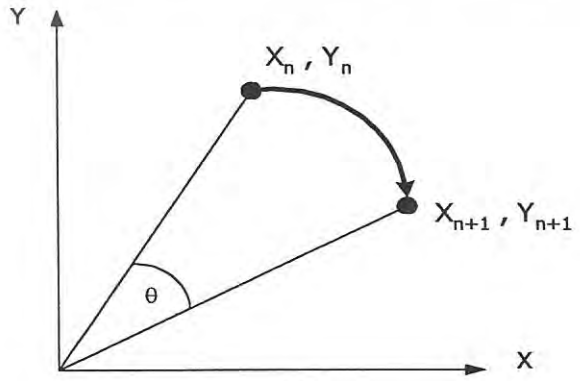
ภาพที่ 3.14 แสดงมุม β ซึ่งเกิดจากเส้นของจุดต้นและจุดปลาย

จากภาพที่ 3.14 จะเห็นได้ว่า มุม β เป็นมุมที่เกิดจากเส้นของจุดต้นและจุดปลาย โดยมุม β นี้จะนำไปใช้ในการสร้างเงื่อนไขในการเขียนโปรแกรมเพื่อกำหนดจำนวนการสร้างเส้นตรงเล็กๆ (Interpolation) ว่าจะต้องมีกี่เส้น ส่วนมุม θ จะเป็นค่ามุมที่เป็นตัวกำหนดความละเอียดในการสร้างเส้นตรงย่อยๆ โดยใช้มุม $\theta = \theta + 0.01$

สมการที่ใช้หาค่า Xn+1 และ Yn+1 ของการเคลื่อนที่ตามเข็มนาฬิกา เป็นดังนี้

$$X_{n+1} = X_n \cos\theta + Y_n \sin\theta$$

$$Y_{n+1} = -X_n \sin\theta + Y_n \cos\theta$$



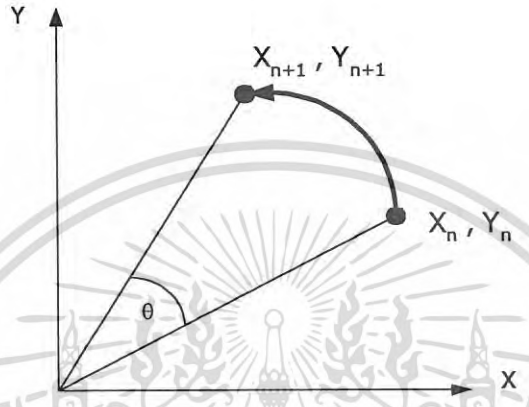
ภาพที่ 3.15 ทิศทางการเคลื่อนที่ ตามเข็มนาฬิกา(Clock Wise; CW)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการที่ใช้หาค่า Xn+1 และ Yn+1 ของการเคลื่อนที่ทวนเข็มนาฬิกา เป็นดังนี้

$$X_{n+1} = X_n \operatorname{Cos}\theta - Y_n \operatorname{Sin}\theta$$

$$Y_{n+1} = X_n \operatorname{Sin}\theta + Y_n \operatorname{Cos}\theta$$



ภาพที่ 3.16 ทิศทางการเคลื่อนที่ ทวนเข็มนาฬิกา(Counter Clock Wise; CCW)

3.4 ส่วนการแสดงผลภาพและการส่งข้อมูลไปยังส่วนควบคุมการเคลื่อนที่

การแสดงผลภาพ คือการนำข้อมูลที่ได้จากการแปลงเป็นระบบพิกัด X-Y-Z มาแสดงภาพจำลองการเดินทางของเครื่องกัดชิ้นงาน(Tool Path) ซึ่งเส้นทางดังกล่าวจะเป็นทางเดินจริงที่เครื่องกัดจะใช้เป็นเส้นทางในการกัดชิ้นงาน ในส่วนของการแสดงผลภาพนี้ได้ใช้ความสามารถในการจัดการของโปรแกรม Visual Basic ซึ่งเป็นคอนโทรล Picture (PictureBox)

ส่วนการส่งข้อมูลไปยังส่วนควบคุมการเคลื่อนที่(Motion Control) นั้นได้อาศัยคุณสมบัติของการเข้ากันได้ระหว่างคอมพิวเตอร์กับบอร์ดควบคุม หรือที่เรียกว่า "โพรโตคอล(Protocol)" ซึ่งมีรายละเอียดดังนี้

3.4.1 คำสั่งที่ใช้ติดต่อกับ Motion Control

ตัวโมชั่นคอนโทรลที่เรานำมาใช้งานนี้เป็นผลงานที่ ผศ. ประภาษ อุคคกิมาพันธุ์ วิจัยสร้างขึ้นมา ใช้ตัวไมโครคอนโทรลเลอร์ของ Philips รุ่น 89C51 RD+ เป็นตัวประมวลผลตัวโมชั่นคอนโทรลนี้ใช้การเชื่อมต่อผ่านพอร์ตอนุกรมโดยใช้รหัสASCIIในการส่งข้อมูลตัวโมชันคอนโทรลสามารถควบคุมการทำงานของมอเตอร์ได้สองแกนและการทำงานจะเป็นแบบเพิ่มพูน (Increment) โดยมีคำสั่งในการใช้งานดังนี้

- คำสั่ง H เป็นคำสั่งที่ป้อนเพื่อดูรูปแบบการใช้งานคำสั่งต่างๆที่ต้องใช้กับตัวโมชั่นคอนโทรล เมื่อป้อนคำสั่งก็จะแสดงผลออกมาดังนี้ ตาม ตารางที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 แสดงผลของโมชั่นคอนโทรลเมื่อป้อนคำสั่ง H

Mechatronic laboratory division. Department of Instrumentation Engineering. Faculty of Engineering King Mongkuit Institute of Technology Ladkrabang. *** MOTION CONTROLLER MENU CHARACTER COMMAND**** This program is a simple of two and half axis Motion controller Operating on the Philips 89C51 RD+ CPU. State of result of the Motion signal on Port 1 (bit 10 to bit 17)		
Command	**Syntax**	**Function**
Read	R	report motion parameter.
Down pen	D	move pen downward.
Up pen	U	move pen upward.
Total para	T y z s a	initial set motion parameter.
Input buff	I cr..percen	input program replay.
Motion	M	replay motion command in buffer.
Go	G	execute motion or start of motion.
Pen	P u1 u2 u3 d1 d2 d3	define pen parameter.
F	F s a	define rapid speed in case of pen up.
Datum	K k1 k2 k3	k1 k2 k3 define work datum.
Work datum	W	Go to works datum.
Remove datum	Q	open datum break point.
Open datum	O	open datum break point.
Fix datum	C	fix datum break point.

- คำสั่ง T เป็นคำสั่งป้อนค่าพิกัดและอัตราความเร็วที่ใช้ในการเคลื่อนที่
  การป้อนคำสั่งมีรูปแบบดังนี้ [T y z f a]
  โดยที่ ค่า y คือการหมุนของมอเตอร์ A
  ค่า z คือการหมุนของมอเตอร์ B
  ค่า f คืออัตราความเร็วในการเคลื่อนที่ มีหน่วยเป็น step / clock ของโมชั่น

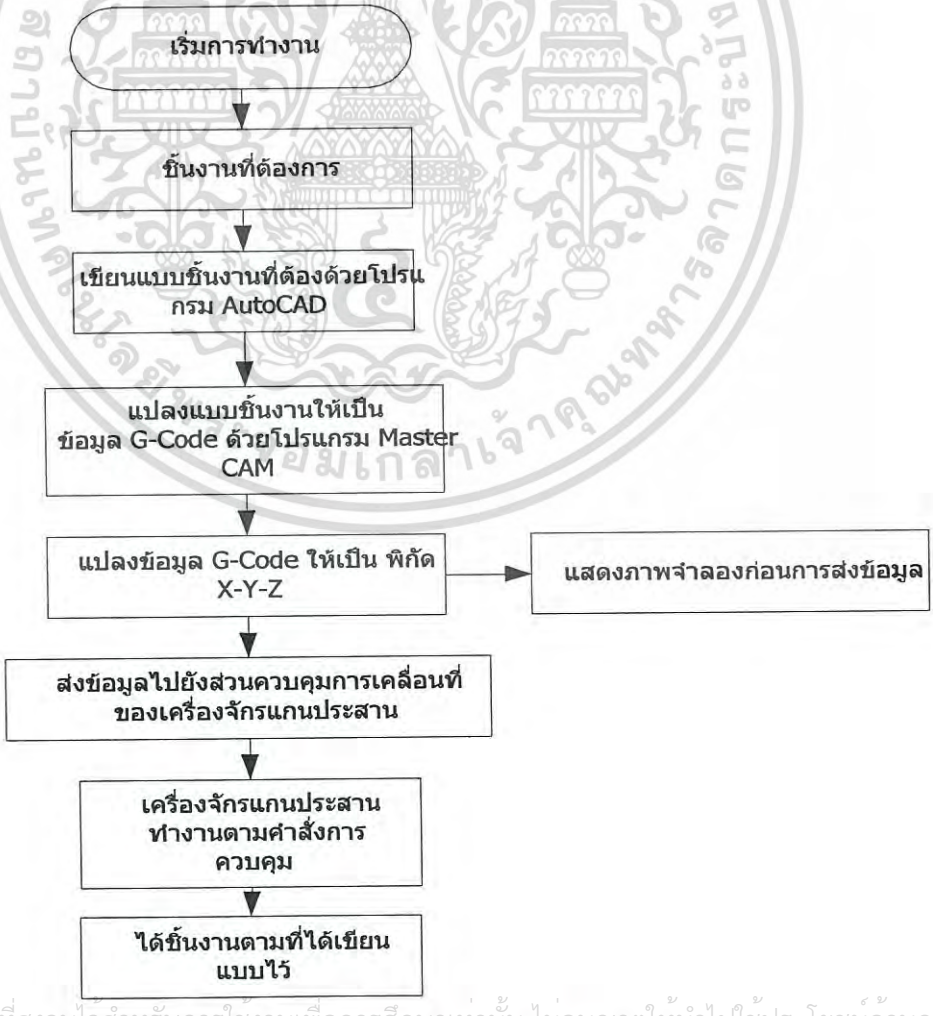
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การออกแบบส่วนของโปรแกรม

4.1 กล่าวนำ

การทำโครงงานวิจัยในปริญญานิพนธ์ฉบับนี้มีขั้นตอนการศึกษาเริ่มจากการศึกษาการทำงานของเครื่องจักรซีเอ็นซี แบบ Milling จากนั้นจึงได้ศึกษาต่อไปถึงความสัมพันธ์ระหว่างรหัส G-Code กับการเคลื่อนที่ของเครื่องจักรซีเอ็นซี เมื่อทราบถึงความสัมพันธ์ทั้งหมดจึงได้มีการศึกษาถึงพฤติกรรมของเครื่องจักรแกนประสาน เพื่อที่จะนำมาใช้ในการเลือนแบบการทำงานให้มีความเสมือนการทำงานของเครื่องจักรซีเอ็นซีแบบ Milling ได้ ตลอดจนได้ศึกษาโปรแกรม CAD/CAM โปรแกรมวิชวลเบสิก ซึ่งเป็นโปรแกรมที่นำมาใช้ในการทำโครงงานครั้งนี้

4.2 ระบบการทำงานโดยรวมของโปรแกรม



ภาพที่ 4.1 แสดงลำดับการทำงานโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

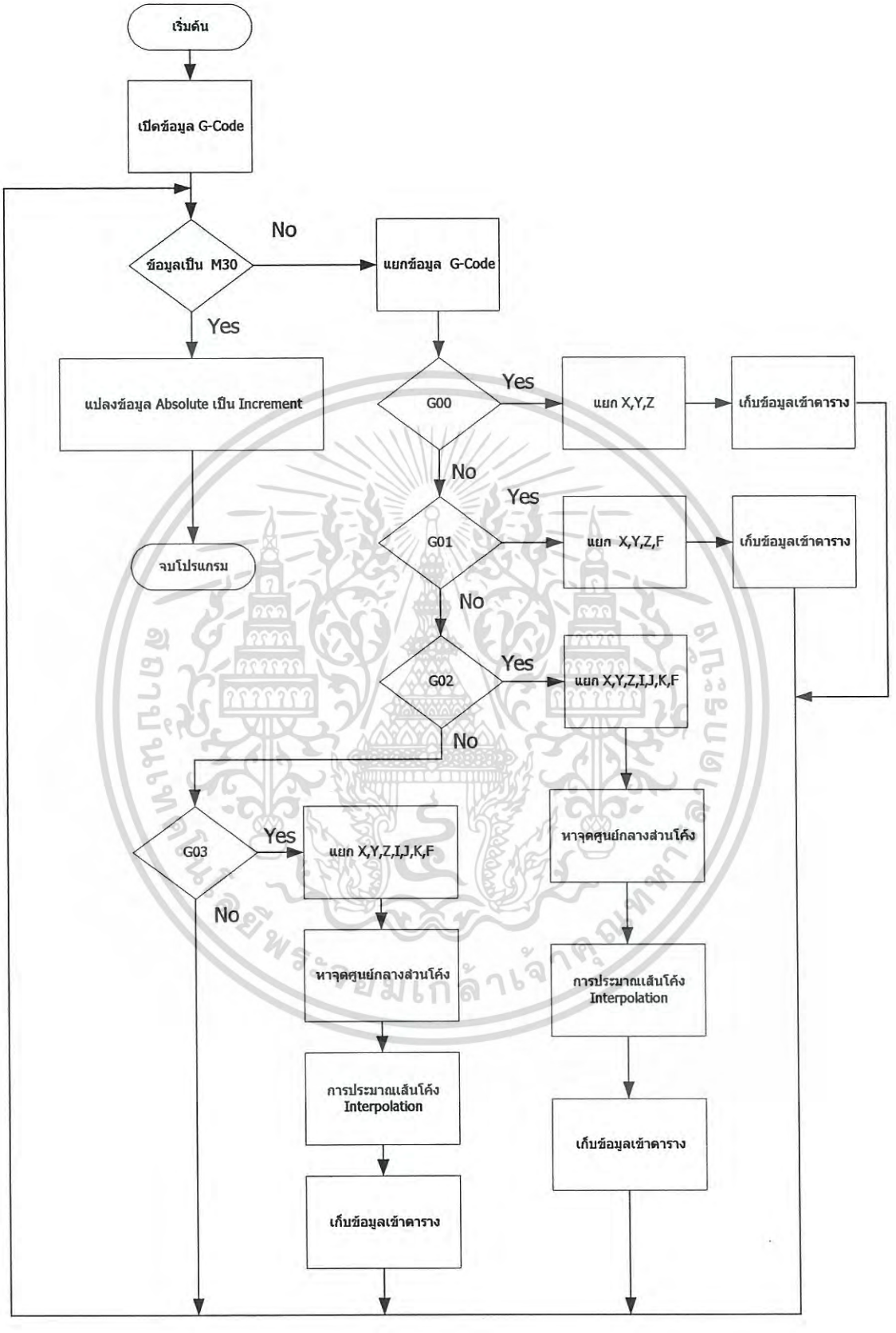
โปรแกรมแปลงรหัส G-Code นี้ จะทำการแปลงข้อมูล G-Code ที่อยู่ในรูปของไฟล์ข้อมูลแบบ NC ไฟล์ (*.NC) ดังนั้นในการทำงานของโปรแกรมจะต้องอาศัยโปรแกรมที่สามารถเขียนแบบชิ้นงานตามที่ต้องได้ โดยในโครงงานนี้เลือกใช้โปรแกรม AutoCAD และโปรแกรมที่สามารถเปลี่ยนแบบของชิ้นงานที่ได้จากโปรแกรม AutoCAD ให้เป็นข้อมูล G-Code ในไฟล์ข้อมูลแบบ NC (*.NC) ในที่นี้ได้เลือกใช้โปรแกรม Master CAM เป็นเครื่องมือช่วยในการทำงาน โดยมีลำดับการทำงานโดยรวมของโปรแกรม ดังนี้

4.3 การออกแบบโปรแกรม

ในการออกแบบโปรแกรมแปลงรหัส G-Code เป็นระบบพิกัด X-Y-Z นั้น ได้ทำการออกแบบโดยใช้โปรแกรมวิชวลเบสิกมาเป็นเครื่องมือในการพัฒนาโปรแกรม ซึ่งตัวโปรแกรมที่ออกแบบจะแบ่งเป็นส่วนสำคัญๆดังนี้

- โปรแกรมส่วนที่ใช้รับข้อมูล G-Code จากโปรแกรมช่วยการผลิต(CAM)ที่เป็นไฟล์ข้อมูลแบบ NC
- โปรแกรมส่วนที่ทำการแปลงข้อมูล G-Code เป็นระบบพิกัด X-Y-Z
- โปรแกรมส่วนที่ใช้แสดงภาพจำลอง และ ส่วนการส่งข้อมูลไปยังเครื่องจักรแกนประสาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



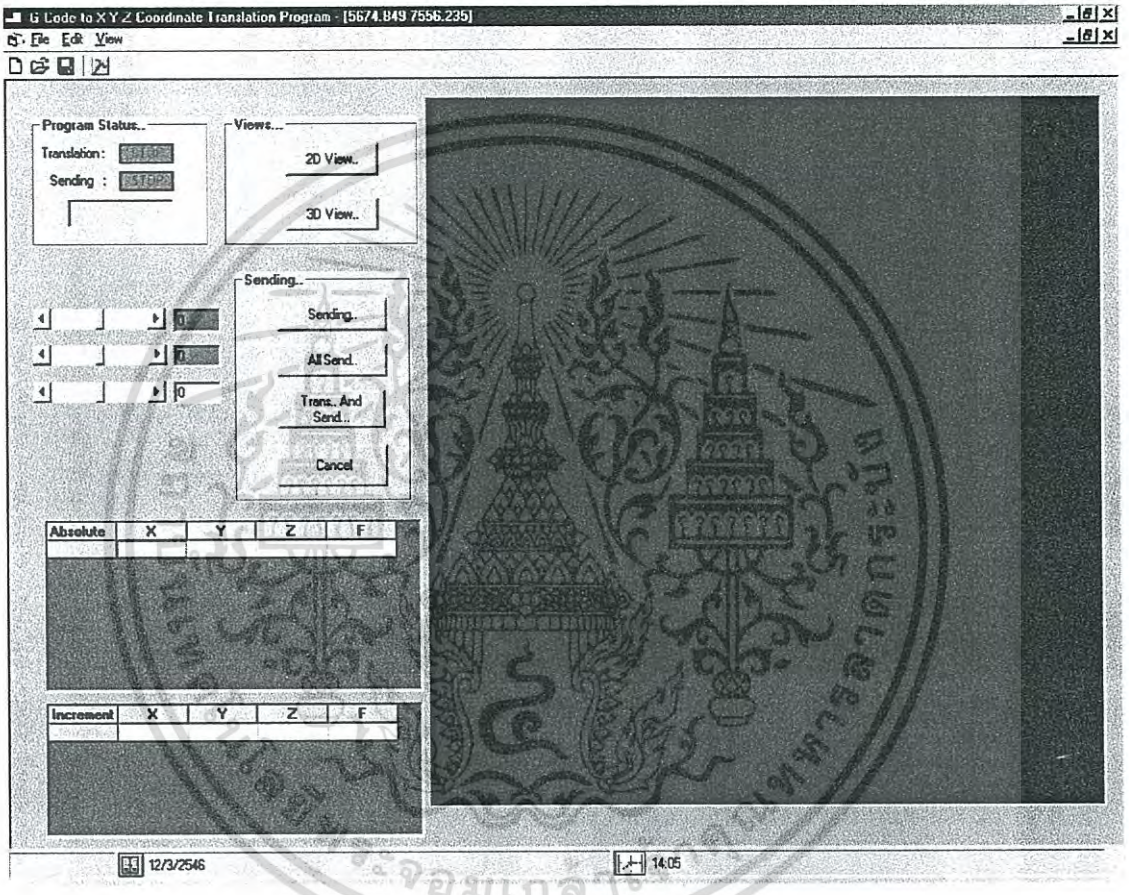
ภาพที่ 4.2 Flow Chart แสดงการทำงานของโปรแกรมแปลง G-Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ให้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ผลการทดลอง

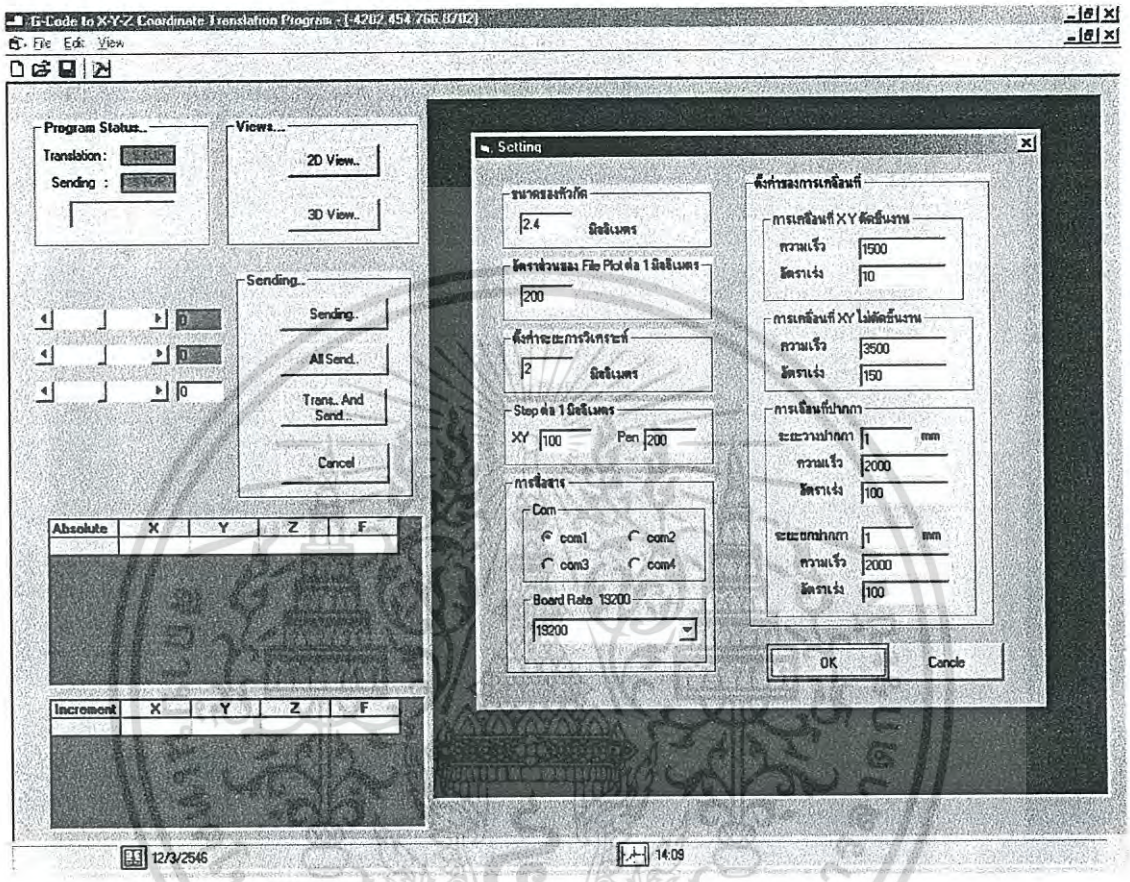
5.1 ภาพหน้าต่างของโปรแกรมแปลงรหัส G-Code



ภาพที่ 5.1 แสดงหน้าต่างโปรแกรมแปลงรหัส G-Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ภาพหน้าต่างของโปรแกรมแปลงรหัส ส่วนการตั้งค่าพารามิเตอร์

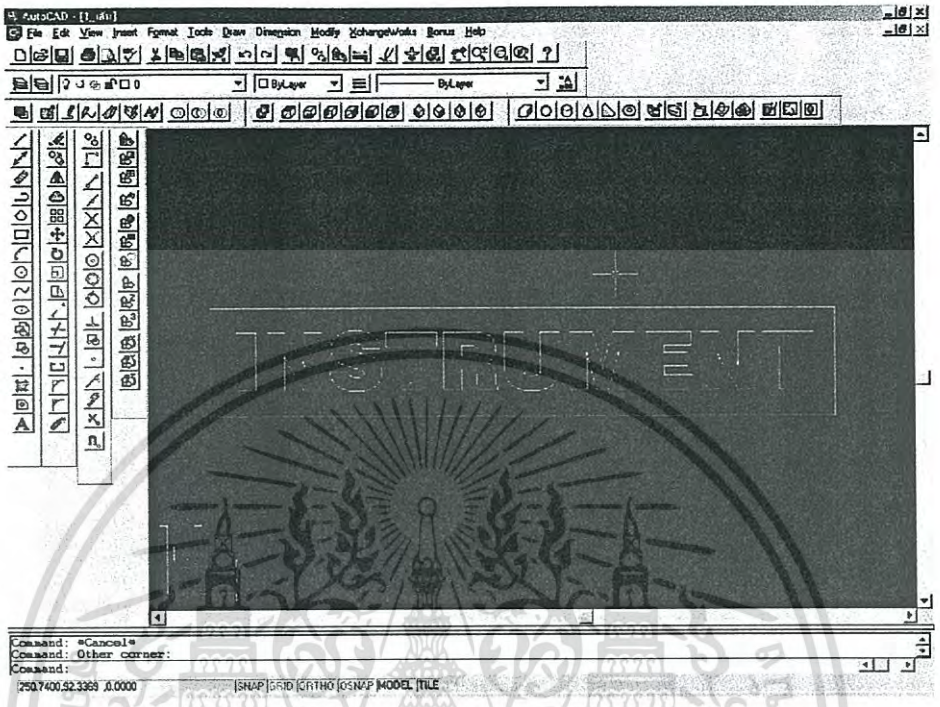


ภาพที่ 5.2 แสดงหน้าต่างโปรแกรมแปลงรหัส G-Code ส่วนการตั้งค่าพารามิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

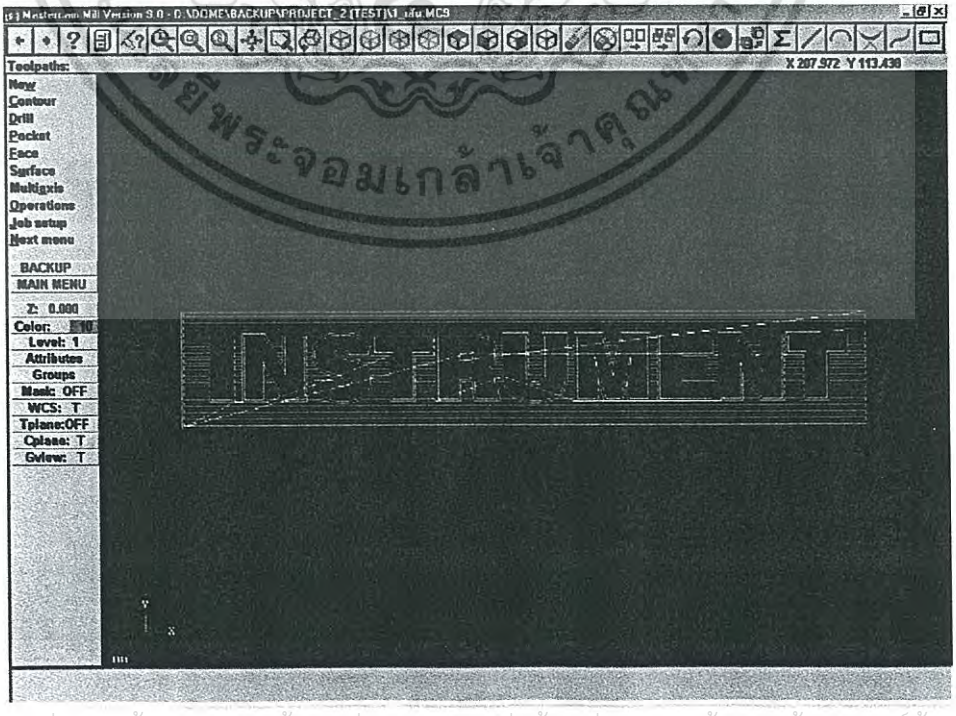
5.3 การทดลอง

5.3.1 เขียนแบบชิ้นงานที่ต้องการ ด้วยโปรแกรม AutoCAD



ภาพที่ 5.3 แสดงการเขียนแบบชิ้นงานด้วยโปรแกรม AutoCAD

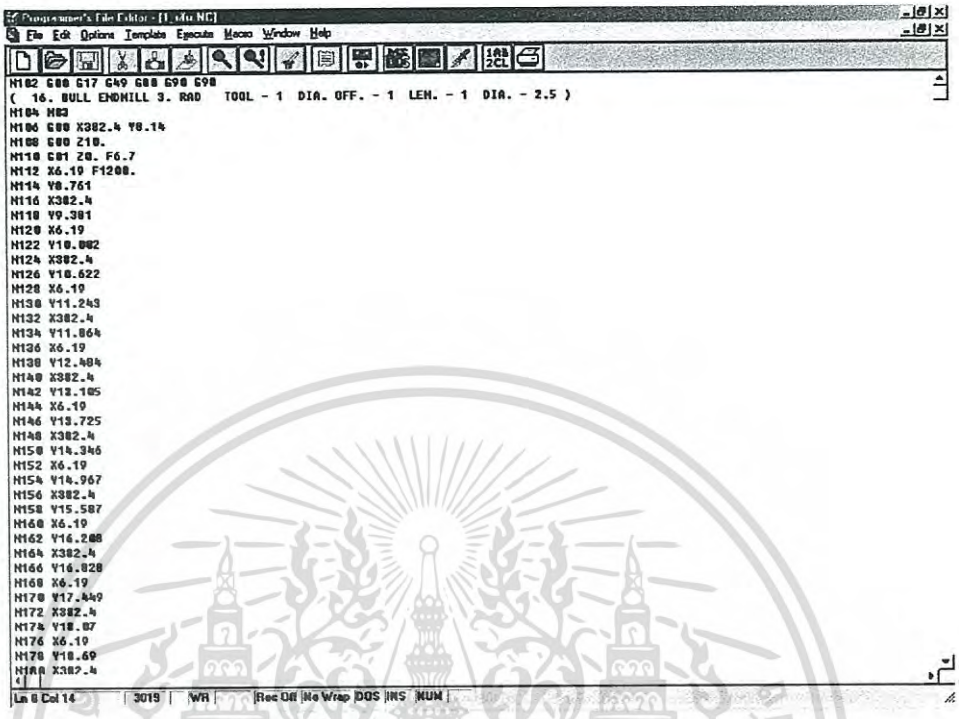
5.3.2 แปลงแบบชิ้นงานให้เป็นข้อมูล G-Code ด้วยโปรแกรม Master CAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

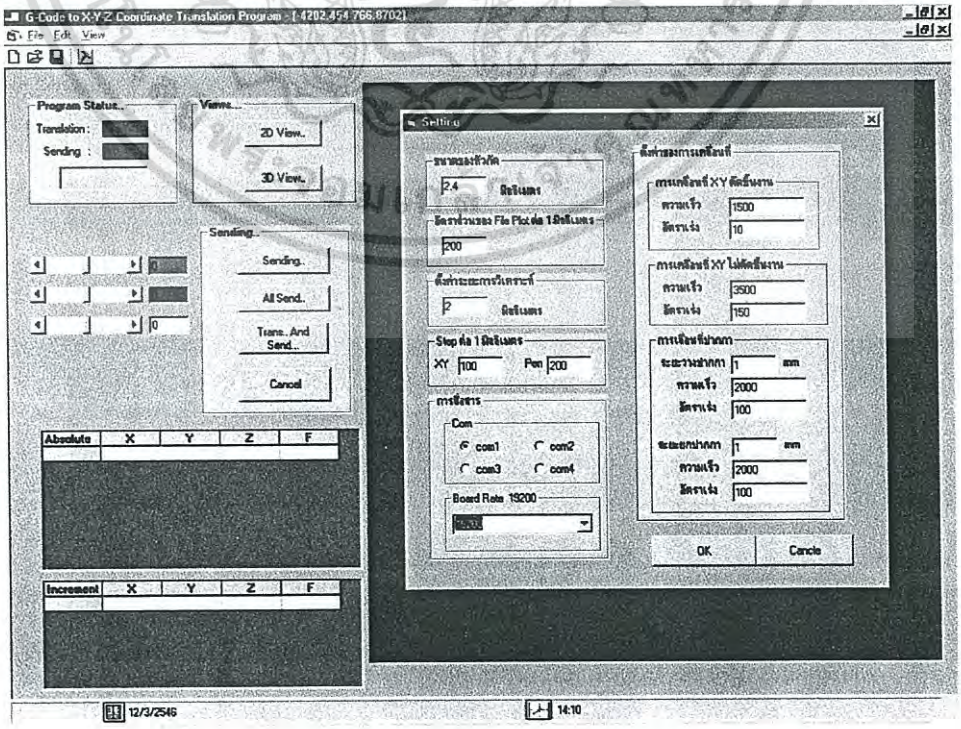
ภาพที่ 5.3 แสดงการแปลงแบบชิ้นงานให้เป็นข้อมูล G-Code ด้วยโปรแกรม Master CAM

5.3.3 ข้อมูล G-Code ที่ได้จากโปรแกรม Master CAM



ภาพที่ 5.4 แสดงข้อมูล G-Code ที่ได้จากโปรแกรม Master CAM

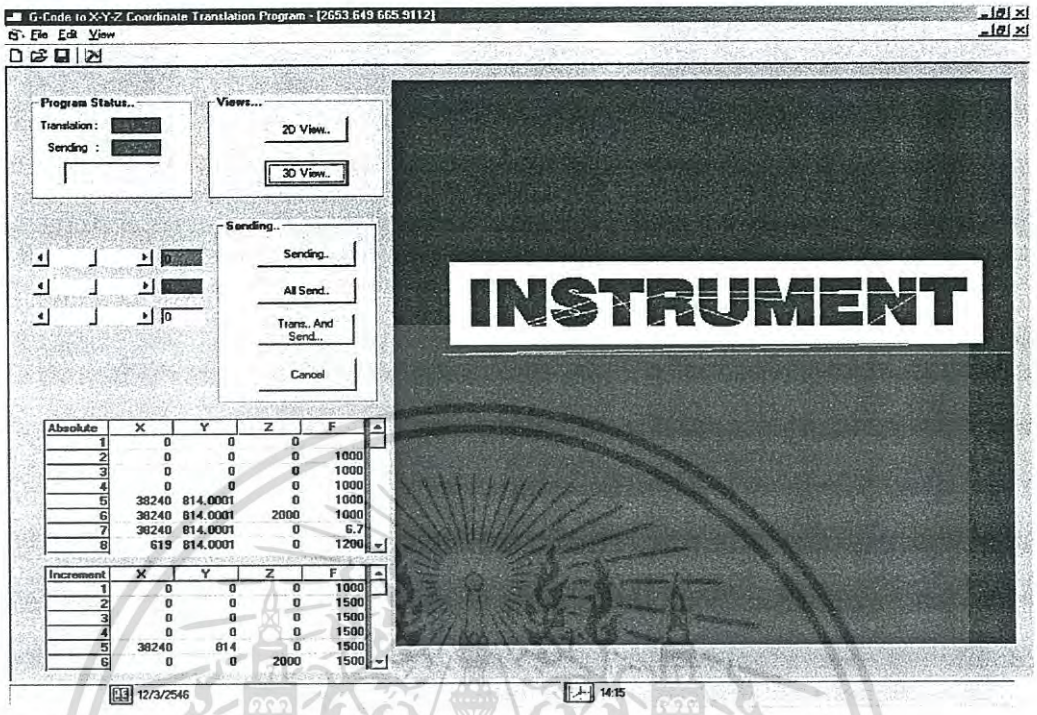
5.3.4 เปิดโปรแกรมแปลง G-Code และทำการตั้งค่าพารามิเตอร์



ภาพที่ 5.5 แสดงการตั้งค่าพารามิเตอร์ของโปรแกรมแปลงรหัส G-Code

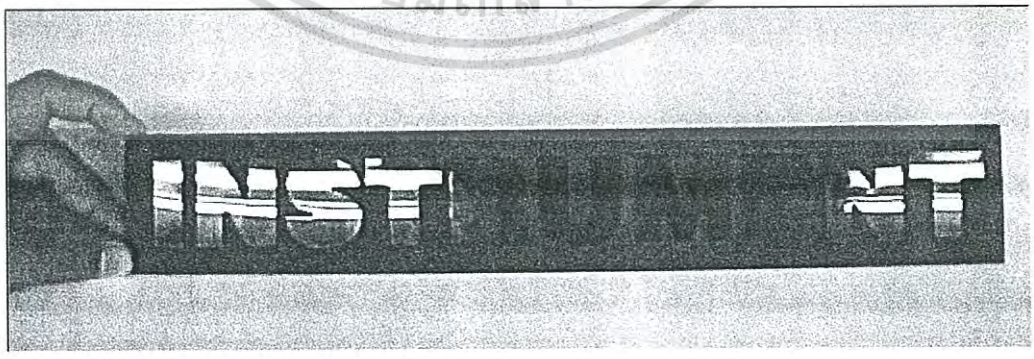
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.5 เปิดข้อมูล และทำการแปลงข้อมูล G-Code ให้เป็นระบบพิกัด X-Y-Z



ภาพที่ 5.5 แสดงการเปิดและแปลงข้อมูล G-Code ให้เป็นระบบพิกัด X-Y-Z

5.3.6 ภาพชิ้นงานที่ได้จากการทดลอง



ภาพที่ 5.6 แสดงภาพชิ้นงานที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปผลและข้อจำกัดของโครงงาน

6.1 สรุปผล

จากการทำโครงงานครั้งนี้ สามารถสรุปขั้นตอนในการทำงานโดยแบ่งออกเป็นส่วนๆ ได้ 4 ส่วน ดังนี้

1. ศึกษาพฤติกรรมของเครื่องจักรซีเอ็นซีแบบ Milling โดยมีเป้าหมายอยู่ที่ความสัมพันธ์ระหว่างข้อมูลที่ใช้ในการควบคุมเครื่องจักรที่เรียกว่า G-Code กับการเคลื่อนที่ของเครื่องจักร
2. ศึกษาและหาเครื่องมือที่จะใช้ในการสร้างการเลียนแบบพฤติกรรมของเครื่องจักรซีเอ็นซีแบบ Milling ซึ่งในโครงงานนี้ได้เลือกใช้โปรแกรมวิชวลเบสิกในการสร้างโปรแกรมแปลงรหัส G-Code และอาศัยเครื่องจักรแกนประสานแบบ 3 แกน เป็นเครื่องจักรแทนเครื่องจักรซีเอ็นซีแบบ Milling
3. สร้างและพัฒนาโปรแกรมแปลงรหัส G-Code ให้มีความสามารถในการแปลงข้อมูลให้สอดคล้องกับส่วนควบคุมการเคลื่อนที่ของเครื่องจักรแกนประสานแบบ 3 แกน
4. แก้ไขและพัฒนาโปรแกรมให้สามารถใช้งานได้ง่าย

จากการทำงานตามขั้นตอนทั้งหมดในข้างต้น จึงเป็นผลให้สามารถสร้างโปรแกรมแปลงรหัส G-Code ให้ทำการแปลงข้อมูล แล้วนำผลลัพธ์ที่ได้ซึ่งอยู่ในระบบพิกัด X-Y-Z ไปส่งต่อให้กับส่วนควบคุมการเคลื่อนที่ของเครื่องจักรแกนประสานแบบ 3 แกน และสามารถควบคุมเครื่องจักรแกนประสานแบบ 3 แกน ให้ทำงานได้เสมือนเครื่องจักรซีเอ็นซีแบบ Milling ได้

6.2 ข้อจำกัดของโครงงาน

- โปรแกรมแปลงรหัสไม่สามารถแปลงฟังก์ชัน G-Code และ ฟังก์ชัน M-Code บางตัวได้
- สำหรับข้อมูล G-Code ที่มีจำนวนบรรทัดมาก จะต้องส่งข้อมูลทันทีที่มีการเปิดไฟล์ ดังนั้นจึงไม่สามารถดูผลลัพธ์ของโปรแกรมและภาพจำลองของชิ้นได้ก่อนการส่งข้อมูลไปยังเครื่องจักรแกนประสานได้
- ไม่สามารถทำให้เครื่องจักรแกนประสานเคลื่อนที่แบบสัมพันธ์กันทั้ง 3 แกนในเวลาเดียวกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. “หลักการโปรแกรม วิเชวลเบสิก” , รศ. เดือน สิธุพัธ์ประทุม ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาลัยบูรพา
2. “Visual Basic 6 ฉบับโปรแกรมเมอร์” , กิตติ ภักดีวัฒนะกุล-จำลอง ครูอุตสาหะ
3. “คู่มือการใช้โปรแกรม Auto CAD 2000:3D Modeling” , ภานุพงษ์ ปัตติสิงห์
4. “Advanced Auto CAD” , กอบเกียรติ สระอุบล กรุงเทพ:ซีเอ็ดยูเคชั่น
5. “คู่มือการใช้โปรแกรม Master CAM V.8”, ณัฐพงษ์ คงประเสริฐ
6. “Master CAM V.6”,Dr. S.C. Jonathan Lin & Dr. F.C. Tony Shiue

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Option Explicit
Public Scal3D As Integer
Dim XY As Boolean, x1 As Single, y1 As Single
Private Sub Command2_Click() 'ดูรูป 2 มิติ
Drawline
'AbsToInc
End Sub
Private Sub Command3_Click()  'ดูรูป 3 มิติ
DrawPart
'Form1.DataSentXYZ.Visible = False
'Form1.DataSentXYZ.Enabled = False
'Form1.DataXYZ.Visible = False
'Form1.DataXYZ.Enabled = False
End Sub
Private Sub Command4_Click()
mnuRunStepTO
End Sub
Private Sub Command5_Click()
SentForward
End Sub
Private Sub Command6_Click()  ' ยกเลิก(ฟังก์ชั่นการแปลงและการส่ง)
SentCancle = False
Open_Cancle = True
Form1.Shape4.BackColor = &HFF&
Form1.Shape5.BackColor = &HFF&
Form1.Label8.Visible = False
Form1.Label9.Visible = True
Form1.Label11.Visible = False
Form1.Label12.Visible = True
End Sub
Private Sub Command8_Click()
RunStepTO
End Sub
Private Sub Form_Load()
Form1.DataXYZ.Clear
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Form1.DataSentXYZ.Clear
Form1.DataXYZ.Rows = 2
Form1.DataSentXYZ.Rows = 2
Form1.DataXYZ.Cols = 5
Form1.DataSentXYZ.Cols = 5
Form1.DataXYZ.TextMatrix(0, 0) = "Absolute"
Form1.DataXYZ.TextMatrix(0, 1) = "      X"
Form1.DataXYZ.TextMatrix(0, 2) = "      Y"
Form1.DataXYZ.TextMatrix(0, 3) = "      Z"
Form1.DataXYZ.TextMatrix(0, 4) = "      F"
Form1.DataSentXYZ.TextMatrix(0, 0) = "Increment"
Form1.DataSentXYZ.TextMatrix(0, 1) = "      X"
Form1.DataSentXYZ.TextMatrix(0, 2) = "      Y"
Form1.DataSentXYZ.TextMatrix(0, 3) = "      Z"
Form1.DataSentXYZ.TextMatrix(0, 4) = "      F"
OpenFileDataMac
End Sub
Private Sub HScroll1_Change()
Text1.Text = HScroll1.Value
DrawPart
End Sub
Private Sub HScroll2_Change()
Text3.Text = HScroll2.Value
DrawPart
End Sub
Private Sub HScroll3_Change()
Text2.Text = HScroll3.Value
DrawPart
End Sub
Private Sub Label4_Click()
End Sub
Private Sub Pic1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
With Form1.Pic1
If Button = 1 Then
    If XY = True Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    x1 = X: y1 = Y
    End If
        Pic1.ScaleLeft = -5000
        Pic1.ScaleTop = -5000
        Pic1.ScaleWidth = 10000
        Pic1.ScaleHeight = 10000
End If
'If XY = False Then
'    If X = X1 And Y = Y1 Then Exit Sub
'    Form1.Pic1.ScaleLeft = X1
'    Form1.Pic1.ScaleTop = Y1
'    Form1.Pic1.ScaleHeight = Abs(X1) + Abs(X)
'    Form1.Pic1.ScaleWidth = Abs(X1) + Abs(X)
'    DrawPart
'End If
XY = True
End With
End Sub
Private Sub Pic1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Form1.Caption = X & " " & Y
XY = True
End Sub
Private Sub Pic1_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
XY = False
With Form1.Pic1
    If Button = 2 Then

        .ScaleLeft = -5000
        .ScaleTop = -5000
        .ScaleWidth = 10000
        .ScaleHeight = 10000
            DrawPart
    Else
        If x1 = X And y1 = Y Then Exit Sub
        Form1.Pic1.ScaleLeft = x1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Form1.Pic1.ScaleTop = y1
        Form1.Pic1.ScaleHeight = Abs(X) + Abs(x1)
        Form1.Pic1.ScaleWidth = Abs(X) + Abs(x1)
        DrawPart
      End If
      End With
End Sub
Option Explicit
Dim ExitForm2 As Boolean, DataText As String * 1
Function SubChckInteger() As Boolean
If InStr("0123456789", DataText) = 0 Then
MsgBox ("ต้องเป็นตัวเลขจำนวนเต็ม ")
SubChckInteger = False
Exit Function
End If
SubChckInteger = True
End Function
Function SubChckIntegerAndPoint() As Boolean
If InStr("0123456789.", Val(DataText)) = 0 Then
MsgBox ("ต้องเป็นตัวเลขจำนวนเต็ม ")
SubChckIntegerAndPoint = False
Exit Function
End If
SubChckIntegerAndPoint = True
End Function
Private Sub CmdCancle_Click()
Frm2Setting.Hide
Unload Frm2Setting
End Sub
Private Sub CmdOKform2_Click()
Dim iFileNum As Integer
DataAnalize = Text3.Text
Datatool = Text1.Text
DataFilePlot = Text2.Text
DataStepXY = Text4.Text
```

เอกสารนี้มีลิขสิทธิ์ สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DataStepPen = Text13.Text
If Option1.Value = True Then DataCom = 1
If Option2.Value = True Then DataCom = 2
If Option3.Value = True Then DataCom = 3
If Option4.Value = True Then DataCom = 4
DataBoardRate = Combo1.Text
DataXySpeed = Text5.Text
DataXYA = Text6.Text
DataPDDistance = Val(Text7.Text) * DataStepPen
DataPDSpeed = Text11.Text
DataPDA = Text10.Text
DataPUDistance = Val(Text8.Text) * DataStepPen
DataPUA = Text12.Text
DataPUSpeed = Text9.Text
DataSpeedNoCut = Text14.Text
DataANoCut = Text15.Text
iFileNum = FreeFile
Open "c:\DataMaG.txt" For Output As #iFileNum
Print #iFileNum, DataAnalize; Chr(13); Datatool; Chr(13); DataFilePlot; Chr(13); DataStepXY; Chr(13);
DataStepPen; Chr(13); DataCom; Chr(13); DataBoardRate _
                ; Chr(13); DataXySpeed; Chr(13); DataXYA; Chr(13); DataPDDistance; Chr(13);
DataPDSpeed; Chr(13); DataPDA; Chr(13); DataPUDistance; Chr(13); DataPUA; Chr(13);
DataPUSpeed; Chr(13); DataSpeedNoCut; Chr(13); DataANoCut; Chr(13)
Close #iFileNum
ExitForm2 = True
'MSComm1.CommPort = DataCom
'MSComm1.Settings = DataBoardRate & ",n,8,1"
Frm2Setting.Hide
Unload Frm2Setting
MDIForm1.MSComm1.CommPort = DataCom
MDIForm1.MSComm1.Settings = DataBoardRate & ",n,8,1"
End Sub
Private Sub Form_Load()
OpenFileDataMac
ExitForm2 = False
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'SetBlockByte
 Text3.Text = DataAnalize
Text1.Text = Datatool
Text2.Text = DataFilePlot
Text4.Text = DataStepXY
Text13.Text = DataStepPen
Frame10.Caption = Frame10.Caption & "  " & DataBoardRate
Combo1.Text = DataBoardRate
If DataCom = 1 Then Option1.Value = True
 If DataCom = 2 Then Option2.Value = True
 If DataCom = 3 Then Option3.Value = True
If DataCom = 4 Then Option4.Value = True
Text5.Text = DataXySpeed
Text6.Text = DataXYA
Text7.Text = DataPDDistance / DataStepPen
Text9.Text = DataPDSpeed
Text10.Text = DataPDA
Text8.Text = DataPUDistance / DataStepPen
Text12.Text = DataPUA
Text11.Text = DataPUSpeed
Text14.Text = DataSpeedNoCut
Text15.Text = DataANoCut
End Sub
Private Sub Text1_Change()
DataText = Right(Text1.Text, 1)
If SubChckIntegerAndPoint = False Then
If Len(Text1.Text) = 0 Then Exit Sub
Text1.Text = Left(Text1.Text, Len(Text1.Text) - 1)
End If
End Sub
Private Sub Text1_LostFocus()
'If Val(Text1.Text) = 0 Then
 '   Text1.Text = ""
  '  MsgBox ("  ไม่สามารถคำนวนข้อมูล 0 ได้    ")
'   End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
Private Sub Text10_Change()
DataText = Right(Text10.Text, 1)
If SubChckInteger = False Then
If Len(Text10.Text) = 0 Then Exit Sub
Text10.Text = Left(Text10.Text, Len(Text10.Text) - 1)
End If
End Sub
Private Sub Text10_LostFocus()
If Val(Text10.Text) = 0 Then
    Text10.Text = ""
    MsgBox ("  ไม่สามารถคำนวนข้อมูล 0 ได้  ")
  End If
End Sub
Private Sub Text11_Change()
DataText = Right(Text11.Text, 1)
If SubChckInteger = False Then
If Len(Text11.Text) = 0 Then Exit Sub
Text11.Text = Left(Text11.Text, Len(Text11.Text) - 1)
End If
End Sub
Private Sub Text11_LostFocus()
If Val(Text11.Text) = 0 Then
    Text11.Text = ""
    MsgBox ("  ไม่สามารถคำนวนข้อมูล 0 ได้  ")
  End If
End Sub
Private Sub Text12_Change()
DataText = Right(Text12.Text, 1)
If SubChckInteger = False Then
If Len(Text12.Text) = 0 Then Exit Sub
Text12.Text = Left(Text12.Text, Len(Text12.Text) - 1)
End If
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Text12_LostFocus()
If Val(Text12.Text) = 0 Then
    Text12.Text = ""
    MsgBox ("  ไม่สามารถคำนวนข้อมูล 0 ได้    ")
  End If
End Sub
Private Sub Text13_Change()
DataText = Right(Text13.Text, 1)
If SubChckInteger = False Then
If Len(Text13.Text) = 0 Then Exit Sub
Text13.Text = Left(Text13.Text, Len(Text13.Text) - 1)
End If
End Sub
Private Sub Text13_LostFocus()
If Val(Text13.Text) = 0 Then
    Text13.Text = ""
    MsgBox ("  ไม่สามารถคำนวนข้อมูล 0 ได้    ")
  End If
End Sub
Private Sub Text14_Change()
DataText = Right(Text14.Text, 1)
If SubChckInteger = False Then
If Len(Text4.Text) = 0 Then Exit Sub
'Text14.Text = Left(Text14.Text, Len(Text14.Text) - 1)
End If
End Sub
Private Sub Text14_LostFocus()
If Val(Text14.Text) = 0 Then
    Text14.Text = ""
    MsgBox ("  ไม่สามารถคำนวนข้อมูล 0 ได้    ")
  End If
End Sub
Private Sub Text15_Change()
DataText = Right(Text15.Text, 1)
If SubChckInteger = False Then
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If Len(Text5.Text) = 0 Then Exit Sub
'Text15.Text = Left(Text15.Text, Len(Text15.Text) - 1)
End If
End Sub
Private Sub Text15_LostFocus()
If Val(Text15.Text) = 0 Then
    Text15.Text = ""
    MsgBox ("  ไม่สามารถคำนวนข้อมูล 0 ได้  ")
  End If
End Sub
Private Sub Text2_Change()
DataText = Right(Text2.Text, 1)
If SubChckInteger = False Then
If Len(Text2.Text) = 0 Then Exit Sub
Text2.Text = Left(Text2.Text, Len(Text2.Text) - 1)
End If
End Sub
Private Sub Text2_LostFocus()
If Val(Text2.Text) = 0 Then
    Text2.Text = ""
    MsgBox ("  ไม่สามารถคำนวนข้อมูล 0 ได้  ")
  End If
End Sub
Private Sub Text3_Change()
DataText = Right(Text3.Text, 1)
If SubChckIntegerAndPoint = False Then
If Len(Text3.Text) = 0 Then Exit Sub
Text3.Text = Left(Text3.Text, Len(Text3.Text) - 1)
End If
End Sub
Private Sub Text3_LostFocus()
If Val(Text3.Text) = 0 Then
    Text3.Text = ""
    MsgBox ("  ไม่สามารถคำนวนข้อมูล 0 ได้  ")
  End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
Private Sub Text4_Change()
DataText = Right(Text4.Text, 1)
If SubChckInteger = False Then
If Len(Text4.Text) = 0 Then Exit Sub
Text4.Text = Left(Text4.Text, Len(Text4.Text) - 1)
End If
End Sub
Private Sub Text4_LostFocus()
If Val(Text4.Text) = 0 Then
    Text4.Text = ""
    MsgBox ("  ไม่สามารถคำนวนข้อมูล 0 ได้  ")
  End If
End Sub
Private Sub Text5_Change()
DataText = Right(Text5.Text, 1)
If SubChckInteger = False Then
If Len(Text5.Text) = 0 Then Exit Sub
Text5.Text = Left(Text5.Text, Len(Text5.Text) - 1)
End If
End Sub
Private Sub Text5_LostFocus()
If Val(Text5.Text) = 0 Then
    Text5.Text = ""
    MsgBox ("  ไม่สามารถคำนวนข้อมูล 0 ได้  ")
  End If
End Sub
Private Sub Text6_Change()
DataText = Right(Text6.Text, 1)
If SubChckInteger = False Then
If Len(Text6.Text) = 0 Then Exit Sub
Text6.Text = Left(Text6.Text, Len(Text6.Text) - 1)
End If
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Text6_LostFocus()
If Val(Text6.Text) = 0 Then
    Text6.Text = ""
    MsgBox ("  ไม่สามารถคำนวนข้อมูล 0 ได้   ")
  End If
End Sub
Private Sub Text7_Change()
DataText = Right(Text7.Text, 1)
If SubChckIntegerAndPoint = False Then
If Len(Text7.Text) = 0 Then Exit Sub
Text7.Text = Left(Text7.Text, Len(Text7.Text) - 1)
End If
End Sub
Private Sub Text7_LostFocus()
If Val(Text7.Text) = 0 Then
    Text7.Text = ""
    MsgBox ("  ไม่สามารถคำนวนข้อมูล 0 ได้   ")
  End If
End Sub
Private Sub Text8_Change()
DataText = Right(Text8.Text, 1)
If SubChckIntegerAndPoint = False Then
If Len(Text8.Text) = 0 Then Exit Sub
Text8.Text = Left(Text8.Text, Len(Text8.Text) - 1)
End If
End Sub
Private Sub Text8_LostFocus()
If Val(Text8.Text) = 0 Then
    Text8.Text = ""
    MsgBox ("  ไม่สามารถคำนวนข้อมูล 0 ได้   ")
  End If
End Sub
Private Sub Text9_Change()
DataText = Right(Text9.Text, 1)
If SubChckInteger = False Then
```

เอกสารนี้มีลิขสิทธิ์ สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If Len(Text9.Text) = 0 Then Exit Sub
Text9.Text = Left(Text9.Text, Len(Text9.Text) - 1)
End If
End Sub
Private Sub Text9_LostFocus()
If Val(Text9.Text) = 0 Then
    Text9.Text = ""
    MsgBox ("  ไม่สามารถคำนวนข้อมูล 0 ได้   ")
  End If
End Sub
Option Explicit
Private Sub mnuExit_Click()
If MsgBox("ต้องการออกจากโปรแกรมหรือไม่ ?", 68, "โปรดยืนยัน") = 6 Then
   End
End If
End Sub
Private Sub mnuFileNew_Click(Index As Integer)
        Form1.Pic1.Cls
        Form1.DataXYZ.Clear
        Form1.DataSentXYZ.Clear
        Form1.DataXYZ.Rows = 2
        Form1.DataSentXYZ.Rows = 2
        Form1.DataXYZ.Cols = 5
        Form1.DataSentXYZ.Cols = 5
        Form1.DataXYZ.TextMatrix(0, 0) = "Absolute"
        Form1.DataXYZ.TextMatrix(0, 1) = "     X"
        Form1.DataXYZ.TextMatrix(0, 2) = "     Y"
        Form1.DataXYZ.TextMatrix(0, 3) = "     Z"
        Form1.DataXYZ.TextMatrix(0, 4) = "     F"
        Form1.DataSentXYZ.TextMatrix(0, 0) = "Increment"
        Form1.DataSentXYZ.TextMatrix(0, 1) = "     X"
        Form1.DataSentXYZ.TextMatrix(0, 2) = "     Y"
        Form1.DataSentXYZ.TextMatrix(0, 3) = "     Z"
        Form1.DataSentXYZ.TextMatrix(0, 4) = "     F"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
Private Sub mnuFileOpen_Click(Index As Integer)
On Error GoTo Error1
With MDIForm1.CommonDialog1
    .DialogTitle = "Open File"
    .Filter = "Text Document(*.NC)|*.nc|All Files(*.*)|*.*"
    .CancelError = True
    .ShowOpen
    'Form1.EditterGcode.Visible = True
OpenText .Filename
End With
Error1:
End Sub
Private Sub mnuView2dimention_Click()
End Sub
Private Sub mnuSetting_Click()
Load Frm2Setting
     Frm2Setting.Show vbModal
    Frm2Setting.ZOrder
End Sub
Private Sub mnuView2dimension_Click()
Drawline
End Sub
Private Sub mnuView3dimension_Click()
DrawPart
Form1.DataSentXYZ.Visible = False
Form1.DataSentXYZ.Enabled = False
Form1.DataXYZ.Visible = False
Form1.DataXYZ.Enabled = False
End Sub
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Key
  Case "New"
      'Form1.EditterGcode = ""
      Form1.Pic1.Cls
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Form1.DataXYZ.Clear
        Form1.DataSentXYZ.Clear
        Form1.DataXYZ.Rows = 2
        Form1.DataSentXYZ.Rows = 2
        Form1.DataXYZ.Cols = 5
        Form1.DataSentXYZ.Cols = 5
        Form1.DataXYZ.TextMatrix(0, 0) = "Absolute"
        Form1.DataXYZ.TextMatrix(0, 1) = "     X"
        Form1.DataXYZ.TextMatrix(0, 2) = "     Y"
        Form1.DataXYZ.TextMatrix(0, 3) = "     Z"
        Form1.DataXYZ.TextMatrix(0, 4) = "     F"
        Form1.DataSentXYZ.TextMatrix(0, 0) = "Increment"
        Form1.DataSentXYZ.TextMatrix(0, 1) = "     X"
        Form1.DataSentXYZ.TextMatrix(0, 2) = "     Y"
        Form1.DataSentXYZ.TextMatrix(0, 3) = "     Z"
        Form1.DataSentXYZ.TextMatrix(0, 4) = "     F"

    Case "Open"
        mnuFileOpen_Click (1)
    Case "Setting"
        Load Frm2Setting
        Frm2Setting.Show vbModal
        Frm2Setting.ZOrder
End Select
End Sub
Option Explicit
Dim Data2 As Variant
Public Sub OpenText(Filename As String)
Dim da1 As String
Dim i As Long, j As Long, data As String, data1 As String, Data2 As String, Data3 As String
Dim Gcode As String, Count As Long, INCorCEN As Integer, INCorCEN_pen As Single
Dim X As Single, Y As Single, z As Single, F As Single, Ig As Single, Jg As Single, Kg As Single
Dim Xole As Single, Yole As Single, Zole As Single
Dim S1 As Boolean, S2 As Integer, FindEnd As String
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
With Form1.DataXYZ
Form1.Label9.Visible = False
Form1.Shape4.BackColor = &HFF00&
Form1.Label8.Visible = True
'.Clear: .Rows = 2: .Cols = 5
'.TextMatrix(0, 0) = "Absolute": .TextMatrix(0, 1) = "X": .TextMatrix(0, 2) = "Y": .TextMatrix(0, 3) = "Z":
.TextMatrix(0, 4) = "F"
Count = 1
.TextMatrix(Count, 0) = Count
.TextMatrix(Count, 1) = 0
.TextMatrix(Count, 2) = 0
.TextMatrix(Count, 3) = 0
Count = Count + 1
.Rows = Count + 1
.TextMatrix(Count, 0) = Count
.Enabled = False
F = DataSpeedNoCut                    'ตั้งค่าความเร็วสูงสุด
INCorCEN = DataStepXY          'ตัวตูณเพื่อเปลี่ยนจาก มิลลิเมตร เป็น สเต็ป
On Error GoTo Error1
Open Filename For Input As #1
'Form1.EditterGcode.Visible = False
'Form1.EditterGcode.Text = ""
Do While Not EOF(1)
    Line Input #1, da1
    Data3 = da1 & " "                 ' เก็บข้อมูล แถวของ จี ไว้ในตัวแปร Data3
     If Open_Cancle = True Then
              Open_Cancle = False
               Exit Sub
     End If
              If InStr(Data3, "G") <> 0 Then
                  data1 = Mid(Data3, (InStr(Data3, "G")), 3)
             End If
             If InStr(Data3, "M30") <> 0 Then GoTo L3
               Select Case data1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Case "G00"

        If InStr(Data3, "X") <> 0 Then X = XGcode(Data3) * INCorCEN
        If InStr(Data3, "Y") <> 0 Then Y = YGcode(Data3) * INCorCEN
        If InStr(Data3, "Z") <> 0 Then z = ZGcode(Data3) * INCorCEN_pen
        If InStr(Data3, "F") <> 0 Then F = FGcode(Data3)
        .TextMatrix(Count, 1) = X
        .TextMatrix(Count, 2) = Y
        .TextMatrix(Count, 3) = z
        .TextMatrix(Count, 4) = 1000
         Count = Count + 1
        .Rows = Count + 1
        .TextMatrix(Count, 0) = Count
        GoTo L3

          Case "G01"
    Gcode = data1

        If InStr(Data3, "X") <> 0 Then X = XGcode(Data3) * INCorCEN
        If InStr(Data3, "Y") <> 0 Then Y = YGcode(Data3) * INCorCEN
        If InStr(Data3, "Z") <> 0 Then z = ZGcode(Data3) * INCorCEN_pen
        If InStr(Data3, "F") <> 0 Then F = FGcode(Data3)
        .TextMatrix(Count, 1) = X
        .TextMatrix(Count, 2) = Y
        .TextMatrix(Count, 3) = z
        .TextMatrix(Count, 4) = F
        Count = Count + 1
        .Rows = Count + 1
        .TextMatrix(Count, 0) = Count


        GoTo L3

Case "G02"
    Gcode = data1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Xole = X: Yole = Y: Zole = z
                If InStr(Data3, "X") <> 0 Then X = XGcode(Data3) * INCorCEN
                If InStr(Data3, "Y") <> 0 Then Y = YGcode(Data3) * INCorCEN
                If InStr(Data3, "Z") <> 0 Then z = ZGcode(Data3) * INCorCEN_pen
                If InStr(Data3, "F") <> 0 Then F = FGcode(Data3)
                If InStr(Data3, "I") <> 0 Then Ig = IGcode(Data3) * INCorCEN
                If InStr(Data3, "K") <> 0 Then Kg = KGcode(Data3) * INCorCEN
                If InStr(Data3, "J") <> 0 Then Jg = JGcode(Data3) * INCorCEN
                .TextMatrix(Count, 1) = X
                .TextMatrix(Count, 2) = Y
                .TextMatrix(Count, 3) = z
                .TextMatrix(Count, 4) = F
                Count = G02(Val(.TextMatrix(Count - 1, 1)), Val(.TextMatrix(Count - 1, 2)),
Val(.TextMatrix(Count - 1, 3)), X, Y, z, Ig, Jg, Kg, Count, 1, Xole, Yole, Zole)
                Count = Count + 1
                .Rows = Count + 1
                .TextMatrix(Count, 0) = Count
                GoTo L3

            Case "G03"

                Xole = X: Yole = Y: Zole = z
                If InStr(Data3, "X") <> 0 Then X = XGcode(Data3) * INCorCEN
                If InStr(Data3, "Y") <> 0 Then Y = YGcode(Data3) * INCorCEN
                If InStr(Data3, "Z") <> 0 Then z = ZGcode(Data3) * INCorCEN_pen
                If InStr(Data3, "F") <> 0 Then F = FGcode(Data3)
                If InStr(Data3, "I") <> 0 Then Ig = IGcode(Data3) * INCorCEN
                If InStr(Data3, "K") <> 0 Then Kg = KGcode(Data3) * INCorCEN
                If InStr(Data3, "J") <> 0 Then Jg = JGcode(Data3) * INCorCEN
                .TextMatrix(Count, 1) = X
                .TextMatrix(Count, 2) = Y
                .TextMatrix(Count, 3) = z
                .TextMatrix(Count, 4) = F
                Count = G03(Val(.TextMatrix(Count - 1, 1)), Val(.TextMatrix(Count - 1, 2)),
Val(.TextMatrix(Count - 1, 3)), X, Y, z, Ig, Jg, Kg, Count, 1, Xole, Yole, Zole)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Count = Count + 1
                .Rows = Count + 1
                .TextMatrix(Count, 0) = Count
            Case "G20"
                INCorCEN = 20 * 100          'G20 หน่วยเป็นนิ้ว 0.001 นิ้ว ต่อ 0.02 มิลลิเมตร
            Case "G21"
                INCorCEN = 100               'G21 หน่วยเป็นมิลลิเมตร
                INCorCEN_pen = DataStepPen
        End Select
L2:
L3:
DoEvents
Loop
Close #1
End With
Close #1
AbsToInc
Form1.Shape4.BackColor = &HFF&
Form1.Label8.Visible = False
Form1.Label9.Visible = True
Exit Sub
Error1:
Resume Next
End Sub

Public Sub AbsToInc()
    Dim i As Long, DataFeed As Single, DataFeedSent As Single
    On Error GoTo Error

    With Form1
            .DataSentXYZ.TextMatrix(1, 0) = "1"
            .DataSentXYZ.TextMatrix(1, 1) = Val(Form1.Text1.Text) * DataStepXY
            .DataSentXYZ.TextMatrix(1, 2) = Val(Form1.Text2.Text) * DataStepXY
            .DataSentXYZ.TextMatrix(1, 3) = Val(Form1.Text3.Text) * DataStepPen
            .DataSentXYZ.TextMatrix(1, 4) = 1000
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  For i = 2 To .DataXYZ.Rows - 2
            .DataSentXYZ.Rows = i + 1
            .DataSentXYZ.TextMatrix(i, 0) = i
            .DataSentXYZ.TextMatrix(i, 1) = Round((Val(.DataXYZ.TextMatrix(i, 1)) - Val
(.DataXYZ.TextMatrix(i - 1, 1))), 0)
            .DataSentXYZ.TextMatrix(i, 2) = Round((Val(.DataXYZ.TextMatrix(i, 2)) - Val
(.DataXYZ.TextMatrix(i - 1, 2))), 0)
            .DataSentXYZ.TextMatrix(i, 3) = Round((Val(.DataXYZ.TextMatrix(i, 3)) - Val
(.DataXYZ.TextMatrix(i - 1, 3))), 0)
            DataFeed = Val(.DataXYZ.TextMatrix(i, 4))
            If DataFeedSent <> DataFeed Then
              If DataFeed < 3000 Then DataFeed = DataXySpeed
              DataFeedSent = DataFeed
            End If
            .DataSentXYZ.TextMatrix(i, 4) = DataFeed
      Next i

      .DataSentXYZ.TextMatrix(i, 0) = i

            .DataSentXYZ.Rows = i + 1
            .DataSentXYZ.TextMatrix(i, 1) = Round(-Val(Form1.Text1.Text) * DataStepXY, 0)
            .DataSentXYZ.TextMatrix(i, 2) = Round(-Val(Form1.Text2.Text) * DataStepXY, 0)
            .DataSentXYZ.TextMatrix(i, 3) = Round(-Val(Form1.Text3.Text) * DataStepPen, 0)
            .DataSentXYZ.TextMatrix(i, 4) = 1000
  End With

Error:

End Sub
Public Function XGcode(data As String) As Single
Dim Index As Integer, Index2 As Integer
data = data & " "
Index = InStr(data, "X")
Index2 = InStr(Index, data, " ")
XGcode = Val(Mid(data, Index + 1, Index2))
```

เอกสารนี้สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Function
Public Function YGcode(data As String) As Single
Dim Index As Integer, Index2 As Integer
Index = InStr(data, "Y")
data = data & " "
Index2 = InStr(Index, data, " ")
YGcode = Val(Mid(data, Index + 1, Index2))
End Function
Public Function ZGcode(data As String) As Single
Dim Index As Integer, Index2 As Integer
data = data & " "
Index = InStr(data, "Z")
Index2 = InStr(Index, data, " ")
ZGcode = Val(Mid(data, Index + 1, Index2))
End Function
Public Function FGcode(data As String) As Single
Dim Index As Integer, Index2 As Integer
data = data & " "
Index = InStr(data, "F")
Index2 = InStr(Index, data, " ")
FGcode = Val(Mid(data, Index + 1, Index2))
End Function
Public Function IGcode(data As String) As Single
Dim Index As Integer, Index2 As Integer
data = data & " "
Index = InStr(data, "I")
Index2 = InStr(Index, data, " ")
IGcode = Val(Mid(data, Index + 1, Index2))
End Function
Public Function KGcode(data As String) As Single
Dim Index As Integer, Index2 As Integer
data = data & " "
Index = InStr(data, "K")
Index2 = InStr(Index, data, " ")
KGcode = Val(Mid(data, Index + 1, Index2))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Function
Public Function JGcode(data As String) As Single
Dim Index As Integer, Index2 As Integer
data = data & " "
Index = InStr(data, "J")
Index2 = InStr(Index, data, " ")
JGcode = Val(Mid(data, Index + 1, Index2))
End Function
Function SetRadias(Ig, Ij, Ik) As Single
SetRadias = (Sqr(Ig ^ 2 + Ij ^ 2 + Ik ^ 2)) / DataStepXY

  If SetRadias > 10 Then
        SetRadias = 15
  Else
        SetRadias = 8
  End If

End Function
Function G02(STARTX As Single, STARTY As Single, STARTZ As Single, ENDX As Single, ENDY As
Single, ENDZ As Single, Ig As Single, Jg As Single, Kg As Single, Count As Long, Step As Integer,
Xole As Single, Yole As Single, Zole As Single) As Long
Dim i As Single, ANSX As Single, ANSY As Single, j As Long, Q As Single
Dim BufferX As Single, BufferY As Single, CountQ As Single, SumQ As Single
Dim Cx As Single, Cy As Single, Cz As Integer, Xn As Single, Yn As Single
Dim W1 As Single, W2 As Single, R As Integer
j = Count - 1
    Cx = STARTX + Ig
    Cy = STARTY + Jg

    R = SetRadias(Ig, Jg, Kg)

    Xn = STARTX - Cx        'จุดเริ่มแกน X เมื่อคิดที่จุดศูนย์กลางที่ 0,0
    Yn = STARTY - Cy        'จุดเริ่มแกน Y เมื่อคิดที่จุดศูนย์กลางที่ 0,0
    BufferX = ENDX - Cx
    BufferY = ENDY - Cy
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    W1 = Abs(Xn - BufferX)
    W2 = Abs(Yn - BufferY)

    Q = Start_End_DegreeG02(Xn, Yn, BufferX, BufferY)    ' คำนวนหาระยะห่างของมุมเริ่มและจบ

BufferX = 0: BufferY = 0: SumQ = 0: CountQ = 0
With Form1.DataXYZ
K1:
  For i = 0.5 To 10 Step 0.5
      CountQ = CountQ + 0.5
      SumQ = SumQ + CountQ
      ANSX = Round(((Xn * Cos((22 / 7) * CountQ / 180) + Yn * Sin((22 / 7) * CountQ / 180))), 6)
      ANSY = Round(((-Xn * Sin((22 / 7) * CountQ / 180) + Yn * Cos((22 / 7) * CountQ / 180))), 6)
      BufferX = Cx + ANSX
      BufferY = Cy + ANSY
      DoEvents

      If SumQ >= Q Then Exit For

      If Abs(Val(.TextMatrix(j, 2)) - BufferY) >= R And Abs(Val(.TextMatrix(j, 1)) - BufferX) >= R Then
         j = j + 1
         .Rows = j + 1
         .TextMatrix(j, 1) = BufferX
        .TextMatrix(j, 2) = BufferY
        .TextMatrix(j, 3) = Zole
        .TextMatrix(j, 0) = j
        '.TextMatrix(j, 4) = Step
        Xn = ANSX
        Yn = ANSY
       CountQ = 0
        GoTo K1
        End If
    Xn = ANSX
    Yn = ANSY
  Next
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            j = j + 1
            .Rows = j + 1
            .TextMatrix(j, 1) = ENDX
            .TextMatrix(j, 2) = ENDY
            .TextMatrix(j, 3) = ENDZ
            .TextMatrix(j, 0) = j
    G02 = j
End With
End Function
Function G03(STARTX As Single, STARTY As Single, STARTZ As Single, ENDX As Single, ENDY As
Single, ENDZ As Single, Ig As Single, Jg As Single, Kg As Single, Count As Long, Step As Integer,
Xole As Single, Yole As Single, Zole As Single) As Long
Dim i As Single, ANSX As Single, ANSY As Single, j As Long, Q As Single
Dim BufferX As Single, BufferY As Single, CountQ As Single, SumQ As Single
Dim Cx As Single, Cy As Single, Cz As Integer, Xn As Single, Yn As Single, R As Integer
On Error GoTo Error1
j = Count - 1
    Cx = STARTX + Ig
    Cy = STARTY + Jg

     R = SetRadias(Ig, Jg, Kg)

    Xn = STARTX - Cx        'จุดเริ่มแกน X เมื่อคิดที่จุดศูนย์กลางที่ 0,0
    Yn = STARTY - Cy        'จุดเริ่มแกน Y เมื่อคิดที่จุดศูนย์กลางที่ 0,0
    BufferX = ENDX - Cx
    BufferY = ENDY - Cy

    Q = Start_End_DegreeG03(Xn, Yn, BufferX, BufferY)  ' คำนวนหาระยะห่างของมุมเริ่มและจบ
BufferX = 0: BufferY = 0: SumQ = 0: CountQ = 0
With Form1.DataXYZ
K1:
  For i = 0.5 To 10 Step 0.5
      CountQ = CountQ + 0.5
      SumQ = SumQ + CountQ
      ANSX = Round(((Xn * Cos((22 / 7) * CountQ / 180) - Yn * Sin((22 / 7) * CountQ / 180))), 6)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ANSY = Round(((Xn * Sin((22 / 7) * CountQ / 180) + Yn * Cos((22 / 7) * CountQ / 180))), 6)
        BufferX = Cx + ANSX
        BufferY = Cy + ANSY
        If SumQ >= Q Then Exit For
        STARTX = Abs(Val(.TextMatrix(j, 1)) - BufferX)
        STARTY = Abs(Val(.TextMatrix(j, 2)) - BufferY)
        If Abs(Val(.TextMatrix(j, 2)) - BufferY) >= R And Abs(Val(.TextMatrix(j, 1)) - BufferX) >= R Then
            STARTX = Abs(Val(.TextMatrix(j - 1, 1)) - BufferX)
            STARTY = Abs(Val(.TextMatrix(j - 1, 2)) - BufferY)
            j = j + 1
            .Rows = j + 1
            .TextMatrix(j, 1) = BufferX
            .TextMatrix(j, 2) = BufferY
            .TextMatrix(j, 3) = Zole
            .TextMatrix(j, 0) = j
            Xn = ANSX
            Yn = ANSY
            CountQ = 0
            GoTo K1
        End If
    Xn = ANSX
    Yn = ANSY
Next
j = j + 1
            .Rows = j + 1
            .TextMatrix(j, 1) = ENDX
            .TextMatrix(j, 2) = ENDY
            .TextMatrix(j, 3) = ENDZ
            .TextMatrix(j, 0) = j
 G03 = j
 Exit Function
Error1:
 G03 = j
 Resume Next
End With
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Function
Function Start_End_DegreeG02(x1 As Single, y1 As Single, x2 As Single, y2 As Single) As Single
Dim u As Single, c As Single, d As Single, i As Integer, d1 As Single
Dim xd As Single, yd As Single
' มุมเริ่ม
If Sgn(x1) = 0 And Sgn(y1) = 1 Then
    d = 90
    GoTo E
End If
If Sgn(x1) = 0 And Sgn(y1) = -1 Then
    d = 270
    GoTo E
End If
If Sgn(x1) = 1 And Sgn(y1) = 0 Then
    d = 180
    GoTo E
End If
If Sgn(x1) = -1 And Sgn(y1) = 0 Then
    d = 0
    GoTo E
End If
If Sgn(x1) = 1 And Sgn(y1) = 1 Then
    d = 180 - ((Atn(y1 / x1) * 180) / (22 / 7))
    'd = d
End If
If Sgn(x1) = -1 And Sgn(y1) = 1 Then
    d = ((Atn(y1 / x1) * 180) / (22 / 7))
    d = Abs(d)
    'd = 180 + d
End If
If Sgn(x1) = -1 And Sgn(y1) = -1 Then
    d = ((Atn(y1 / x1) * 180) / (22 / 7))
    d = 360 - d
    'd = 180 + d
End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If Sgn(x1) = 1 And Sgn(y1) = -1 Then
    d = ((Atn(y1 / x1) * 180) / (22 / 7))
    d = 180 + Abs(d)
    'd = 360 + d
End If
E:
'มุมจบ
If Sgn(x2) = 0 And Sgn(y2) = 1 Then
    d1 = 90
    GoTo F
End If
If Sgn(x2) = 0 And Sgn(y2) = -1 Then
    d1 = 270
    GoTo F
End If
If Sgn(x2) = 1 And Sgn(y2) = 0 Then
    d1 = 180
    GoTo F
End If
If Sgn(x2) = 0 And Sgn(y2) = -1 Then
    d1 = 0
    GoTo F
End If
If Sgn(x2) = 1 And Sgn(y2) = 1 Then
    d1 = ((Atn(y2 / x2) * 180) / (22 / 7))
    d1 = 180 - d1
End If
If Sgn(x2) = -1 And Sgn(y2) = 1 Then
    d1 = ((Atn(y2 / x2) * 180) / (22 / 7))
    d1 = Abs(d1)
End If
If Sgn(x2) = -1 And Sgn(y2) = -1 Then
    d1 = ((Atn(y2 / x2) * 180) / (22 / 7))
    d1 = 360 - d1
End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If Sgn(x2) = 1 And Sgn(y2) = -1 Then
    d1 = ((Atn(y2 / x2) * 180) / (22 / 7))
    d1 = 180 + Abs(d1)
End If
F:
If d1 > d Then
Start_End_DegreeG02 = Abs(d1 - d)
Else
Start_End_DegreeG02 = Abs(d1 + (360 - d))
End If
End Function
Function Start_End_DegreeG03(x1 As Single, y1 As Single, x2 As Single, y2 As Single) As Single
Dim u As Single, c As Single, d As Single, i As Integer, d1 As Single
Dim xd As Single, yd As Single
' มุมเริ่ม
If Sgn(x1) = 0 And Sgn(y1) = 1 Then
    d = 90
    GoTo E
End If
If Sgn(x1) = 0 And Sgn(y1) = -1 Then
    d = 270
    GoTo E
End If
If Sgn(x1) = 1 And Sgn(y1) = 0 Then
    d = 0
    GoTo E
End If
If Sgn(x1) = -1 And Sgn(y1) = 0 Then
    d = 180
    GoTo E
End If
If Sgn(x1) = 1 And Sgn(y1) = 1 Then
    d = ((Atn(y1 / x1) * 180) / (22 / 7))
End If
If Sgn(x1) = -1 And Sgn(y1) = 1 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        d = ((Atn(y1 / x1) * 180) / (22 / 7))
        d = 180 + d
    End If
    If Sgn(x1) = -1 And Sgn(y1) = -1 Then
        d = ((Atn(y1 / x1) * 180) / (22 / 7))
        d = 180 + d
    End If
    If Sgn(x1) = 1 And Sgn(y1) = -1 Then
        d = ((Atn(y1 / x1) * 180) / (22 / 7))
        d = 360 + d
    End If
    E:
    'มุมจบ
    If Sgn(x2) = 0 And Sgn(y2) = 1 Then
        d1 = 90
        GoTo F
    End If
    If Sgn(x2) = 0 And Sgn(y2) = -1 Then
        d1 = 270
        GoTo F
    End If
    If Sgn(y2) = 0 And Sgn(x2) = 1 Then
        d1 = 0
        GoTo F
    End If
    If Sgn(y2) = 0 And Sgn(x2) = -1 Then
        d1 = 180
        GoTo F
    End If
    If Sgn(x2) = 1 And Sgn(y2) = 1 Then
        d1 = ((Atn(y2 / x2) * 180) / (22 / 7))
    End If
    If Sgn(x2) = -1 And Sgn(y2) = 1 Then
        d1 = ((Atn(y2 / x2) * 180) / (22 / 7))
        d1 = 180 + d1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End If
If Sgn(x2) = -1 And Sgn(y2) = -1 Then
    d1 = ((Atn(y2 / x2) * 180) / (22 / 7))
    d1 = 180 + d1
End If
If Sgn(x2) = 1 And Sgn(y2) = -1 Then
    d1 = ((Atn(y2 / x2) * 180) / (22 / 7))
    d1 = 360 + d1
End If
F:
If d1 > d Then
Start_End_DegreeG03 = Abs(d1 - d)
Else
Start_End_DegreeG03 = Abs(d1 + (360 - d))
End If
End Function
Public Sub Drawline()
Dim i As Long, Offset As Single
Dim x1 As Single, y1 As Single, x2 As Single, y2 As Single
With Form1.DataXYZ
    Form1.Pic1.Cls
    Form1.Pic1.DrawWidth = 1
    'Offset = Form1.Pic1.ScaleHeight
    Offset = 0
For i = 2 To .Rows - 2
    x1 = Format(Val(.TextMatrix(i - 1, 1)) / 10, 0)
    y1 = (Offset - Val(.TextMatrix(i - 1, 2)) / 10)
    x2 = Val(.TextMatrix(i, 1)) / 10
    y2 = (Offset - Val(.TextMatrix(i, 2)) / 10)
    'Picture1.PSet (Val(MSFlexGrid4.TextMatrix(I, 1)) / 1, Val(MSFlexGrid4.TextMatrix(I, 2)) / 1)
    Form1.Pic1.Line (x1, y1)-(x2, y2), vbWhite
Next
'Form1.DataGcode.Enabled = True
'Form1.DataGcode.Visible = True
Form1.DataXYZ.Enabled = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Form1.DataXYZ.Visible = True
Form1.DataSentXYZ.Enabled = True
Form1.DataSentXYZ.Visible = True
End With
End Sub
Option Explicit
Public Rot3dX As Long
Public Rot3dY As Long
Public Cx As Long
Public Cy As Long
Public Cz As Long
Public Sub DrawPart()
Dim i As Long
Dim R As Single, X As Single, Y As Single
Dim dx(2) As Long
Dim dy(2) As Long
Dim dz(2) As Long
Dim px() As Long
Dim py() As Long
Dim DataX As Long
Dim DataY As Long
Dim DataZ As Long
R = Form1.HScroll2.Value          ' ระยะตามแนวแกน Z จาก  HScroll_2
Y = Form1.HScroll3.Value          ' ระยะตามแนวแกน Y จาก  HScroll_3
X = Form1.HScroll1.Value          ' ระยะตามแนวแกน X จาก  HScroll_1
Form1.Pic1.Cls
dx(0) = 0: dy(0) = -5000: dz(0) = 0
dx(1) = 5000: dy(1) = 0: dz(1) = 0
dx(2) = 0: dy(2) = 0: dz(2) = 5000
'Convert the points to their new coord system
With Form1.DataXYZ

  ReDim px((.Rows - 2)) As Long
  ReDim py((.Rows - 2)) As Long
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
For i = 1 To .Rows - 2
   DataX = Round(.TextMatrix(i, 1) / 10, 0)
   DataY = Round(.TextMatrix(i, 2) / 10, 0)
   DataZ = Round(Val(.TextMatrix(i, 3) / 10), 0)
   Rot3D DataX, -DataY, DataZ, X, Y, R
   px(i) = Rot3dX
   py(i) = Rot3dY
Next i
End With
'Draw the lines on the PicBox
With Form1
For i = 1 To .DataXYZ.Rows - 3
.Pic1.Line (px(i), py(i))-(px(i + 1), py(i + 1)), RGB(255, 255, 0)
Next i
 ReDim px(2) As Long
 ReDim py(2) As Long
For i = 0 To 2
   Rot3D dx(i), dy(i), dz(i), X, Y, R
   px(i) = Rot3dX
   py(i) = Rot3dY
Next i
.Pic1.Line (px(0), py(0))-(0, 0), &HFF&       ',RGB(255, 0, 0)
.Pic1.Line (px(1), py(1))-(0, 0), &HFF00&    ', RGB(0, 255, 0)
.Pic1.Line (px(2), py(2))-(0, 0), &HFFFF&    ', RGB(0, 0, 255)
End With
End Sub
Private Function DtoR(X)
DtoR = X * (3.1415 / 180)
End Function
Private Function RtoD(X)
RtoD = X * (180 / 3.1415)
End Function
Public Sub Rot3D(X, Y, z, AX, AY, AZ)
Dim AA1 As Variant
Dim AA2 As Variant
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim AA3  As Variant
Dim XROT As Variant
Dim YROT As Variant
Dim ZROT As Variant
AA1 = AZ * (0.0174528)
AA2 = AX * (0.0174528)
AA3 = AY * (0.0174528)
'Calculate the New X,Y Positions
XROT = (X * Cos(AA1)) - (Y * Sin(AA1))
YROT = (X * Sin(AA1)) + (Y * Cos(AA1))
ZROT = (YROT * Sin(AA2)) + (z * Cos(AA2))
'Theses are the variables that are returned
Rot3dX = (XROT * Cos(AA3)) - (ZROT * Sin(AA3))
Rot3dY = (YROT * Cos(AA2)) - (z * Sin(AA2))
'Commented out for speed because not needed (do not have a z-axis on a 2d display!)
'Rot3DZ = (XROT * Sin(A3)) + (ZROT * Cos(A3))
End Sub
Option Explicit
Public SetBlockByte As Integer
Public DataAnalize As Single
Public Datatool As Single
Public DataFilePlot  As Integer
Public DataStepXY As Integer
Public DataStepPen As Integer
Public DataCom As Integer
Public DataCom1 As Integer
Public DataCom2 As Integer
Public DataCom3 As Integer
Public DataCom4 As Integer
Public DataBoardRate As Integer
Public DataXySpeed As Single
Public DataXYA As Integer
Public DataPDDistance As Integer
Public DataPDSpeed As Integer
Public DataPDA As Integer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Public DataPUDistance As Integer
Public DataPUA As Integer
Public DataPUSpeed As Integer
Public DataSpeedNoCut As Single
Public DataANoCut As Single
Public RowGridCount1 As Integer
Public Sub OpenFileDataMac()
Dim data1(20) As String * 10, i As Integer
 Open "c:\DataMaG.txt" For Input As #1      ' เปิด DataMAG เพื่อการทดลอง
i = 0
Do While Not EOF(1)
   Line Input #1, data1(i)
i = i + 1
Loop
   Close #1
DataAnalize = Val(data1(0))
Datatool = Val(data1(1))
DataFilePlot = Val(data1(2))
DataStepXY = Val(data1(3))
DataStepPen = Val(data1(4))
DataCom = Val(data1(5))
DataBoardRate = Val(data1(6))
DataXySpeed = Val(data1(7))
DataXYA = Val(data1(8))
DataPDDistance = Val(data1(9))
DataPUSpeed = Val(data1(10))
DataPDA = Val(data1(11))
DataPUDistance = Val(data1(12))
DataPUA = Val(data1(13))
DataPDSpeed = Val(data1(14))
DataSpeedNoCut = Val(data1(15))
DataANoCut = Val(data1(16))
End Sub
Option Explicit
'Time Delay  Ex  Sloop(1000)  หน่วงเวลา 1 วินาที
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Public Declare Sub Sleep Lib "Kernel32" (ByVal DWMILLISECONDS As Long)
Public SentCancle As Boolean
Public Open_Cancle As Boolean
Dim Positions_X    As String
Dim BoardControl  As Boolean
Sub delay()
Dim times As Variant
 times = Timer
Do
DoEvents
Loop Until Timer >= times + 0.01
End Sub
Sub delay1()
Dim InputAcho As String, i As Integer
K1:
InputAcho = MDIForm1.MSComm1.Input
 DoEvents
 If Len(InputAcho) >= 2 Then GoTo K1
End Sub
Sub Delay2()
Dim times As Variant
  times = Timer
Do
DoEvents
Loop Until Timer >= times + 0.005
End Sub
Sub delay3()
Dim times As Variant
  times = Timer
Do
DoEvents
Loop Until Timer >= times + 2
End Sub
Sub delay4()
 Dim InputAcho As String
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
K1:
delay
InputAcho = MDIForm1.MSComm1.Input
 'Positions_Y.Text = InputAcho
 DoEvents
If InStr(InputAcho, "t") = 0 Then
BoardControl = False
Else: BoardControl = True
End If
End Sub
Public Sub READY()
Dim data As String
data = ""
Do Until InStr(data, "r") <> 0
    Sleep (1)
    data = data & MDIForm1.MSComm1.Input
    'Exit Do
    DoEvents
     Loop
End Sub
Public Sub READY1()
Dim data As String
data = ""
Do Until InStr(data, "ready") <> 0
    Sleep (1)
    data = data & MDIForm1.MSComm1.Input
    'Exit Do
    DoEvents
   Loop
End Sub
Public Sub READY2()
Dim data As String
data = ""
Do Until data = "a"
    delay
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    data = Right(MDIForm1.MSComm1.Input, 3)
    data = Left(data, 1)
    Loop
End Sub
Public Sub READY3()
Dim data As String, i As Integer
Do
data = data & MDIForm1.MSComm1.Input
'Exit Do
DoEvents
Loop Until InStr(data, Chr(13) & Chr(10)) <> 0
    'Sleep (1)
End Sub
Public Function ErrorFunction(CODEerror As Integer)
Select Case CODEerror
        Case 0
            MsgBox ("ไม่มีการกำหนดจุดเริ่มต้น")
End Select
End Function
Public Sub Setmove()   'ทีละเส้น
OpenFileDataMac                    'เปิดข้อมูลที่เก็บค่าการติดตั้ง ที่ไฟล์ DataMac ซึ่งถ้าไม่มีไฟล์นี้ก็จะ
เปิดโปรแกรมไม่ได้
MDIForm1.MSComm1.CommPort = DataCom     'กำหนดการติดต่อสื่อสารที่ port
MDIForm1.MSComm1.Settings = DataBoardRate & ",n,8,1"
With MDIForm1
If Not .MSComm1.PortOpen Then
 .MSComm1.PortOpen = True
End If
.MSComm1.Output = Chr$(13)
delay
.MSComm1.Output = "t 0 0 " & DataXySpeed & " " & DataXYA & Chr$(13)
delay4
If BoardControl = False Then
MsgBox ("   ไม่สามาถส่งข้อมูลเข้า Board Control Motion    ")
GoTo E1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End If
delay
'.MSComm1.Output = "p " & DataPUDistance & " " & DataPUSpeed & " " & DataPUA & " " & -
DataPDDistance & " " & DataPDSpeed & " " & DataPDA & Chr$(13)
.MSComm1.Output = "p " & 0 & " " & DataPUSpeed & " " & DataPUA & " " & 0 & " " & DataPDSpeed & "
" & DataPDA & Chr$(13)
delay1
delay
.MSComm1.Output = "f" & DataSpeedNoCut & " " & DataANoCut & Chr$(13)
delay1
delay
.MSComm1.Output = "d" & Chr$(13)
delay1
delay
End With
MDIForm1.MSComm1.Output = "s" & "1000" & Chr$(13)
READY
E1:
If MDIForm1.MSComm1.PortOpen Then
 MDIForm1.MSComm1.PortOpen = False
End If
End Sub
Public Sub mnuRunStepTO()      'ทีละเส้น
Dim i As Single, j As Single, BlockByte As Integer, NumberBlock As Integer, Xval As Currency, Yval
As Currency
Dim PenSent As String * 1, ByteCount As Integer, DataPen As String * 1, F As Single
Dim DistancePu As Integer, OverRam As Boolean, Positions_X As String, Zval As Currency, Fsent As
Single
OpenFileDataMac                    'เปิดข้อมูลที่เก็บค่าการติดตั้ง ที่ไฟล์ DataMac ซึ่งถ้าไม่มีไฟล์นี้ก็จะ
เปิดโปรแกรมไม่ได้
If Not MDIForm1.MSComm1.PortOpen Then
 MDIForm1.MSComm1.PortOpen = True
End If
'MDIForm1.MSComm1.CommPort = DataCom      'กำหนดการติดต่อสื่อสารที่ port
MDIForm1.MSComm1.Settings = DataBoardRate & ",n,8,1"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
With MDIForm1
.MSComm1.Output = Chr$(13)
delay
.MSComm1.Output = "t 0 0 " & DataXySpeed & " " & DataXYA & Chr$(13)
delay4
If BoardControl = False Then
MsgBox ("  ไม่สามาถส่งข้อมูลเข้า Board Control Motion  ")
GoTo E1
End If
Form1.Label12.Visible = False
Form1.Shape5.BackColor = &HFF00&
Form1.Label11.Visible = True
'.MSComm1.Output = "p " & DataPUDistance & " " & DataPUSpeed & " " & DataPUA & " " & -
DataPDDistance & " " & DataPDSpeed & " " & DataPDA & Chr$(13)
.MSComm1.Output = "p " & 0 & " " & DataPUSpeed & " " & DataPUA & " " & 0 & " " & DataPDSpeed & "
" & DataPDA & Chr$(13)
READY
.MSComm1.Output = "f " & DataSpeedNoCut & " " & DataANoCut & Chr$(13)
READY
.MSComm1.Output = "u" & Chr$(13)
DataPen = "u"
READY
.MSComm1.Output = "o" & Chr$(13)
READY
End With
SentCancle = True
Form1.ProgressBar1.Max = Form1.DataSentXYZ.Rows - 1
Form1.ProgressBar1.Visible = True
For i = 1 To Form1.DataSentXYZ.Rows - 1
If SentCancle = False Then GoTo E1
Xval = (Val(Form1.DataSentXYZ.TextMatrix(i, 1))) '* Abs(Val(PoltScalSET))
Yval = (Val(Form1.DataSentXYZ.TextMatrix(i, 2))) '* Abs(Val(PoltScalSET))
Zval = (Val(Form1.DataSentXYZ.TextMatrix(i, 3)))
F = (Val(Form1.DataSentXYZ.TextMatrix(i, 4)))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Form1.ProgressBar1.Value = i
If Abs(Xval) + Abs(Yval) = 0 Then GoTo L1
 Positions_X = "t " & Xval & " " & Yval
      If F <> Fsent Then
        Fsent = F
          If DataPen = "u" Then
            DataPen = "d"
            MDIForm1.MSComm1.Output = "d" & Chr$(13)
            READY
        End If
        MDIForm1.MSComm1.Output = "s" & Fsent & Chr$(13)
        READY
    End If
MDIForm1.MSComm1.Output = Positions_X & Chr$(13)
'GoTo E1
READY
MDIForm1.MSComm1.Output = "g" & Chr$(13)
READY1
GoTo L2
L1:
If Zval <> 0 Then
  MDIForm1.MSComm1.Output = "k -" & Abs(Zval) & " 500 100" & Chr$(13)
  READY
      If DataPen = "d" Then
        DataPen = "u"
        MDIForm1.MSComm1.Output = "u" & Chr$(13)
        READY
       End If
     If Sgn(Zval) = 1 Then
        MDIForm1.MSComm1.Output = "q" & Chr$(13)
        READY
     Else
        MDIForm1.MSComm1.Output = "w" & Chr$(13)
        READY
     End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DataPen = "d"
        MDIForm1.MSComm1.Output = "d" & Chr$(13)
        READY
End If
L2:
Next i
E1:
Form1.ProgressBar1.Visible = False
If MDIForm1.MSComm1.PortOpen Then
 MDIForm1.MSComm1.PortOpen = False
End If
End Sub
 Sub SentForward()
 Dim i As Long, j As Long, BlockByte As Long, NumberBlock As Integer, Xval As Currency, Yval As
Currency, Zval As Currency
Dim ByteCount As Integer, F As Long, Fsent As Long, DataSent As String, DataPen As String * 1
OpenFileDataMac                    'เปิดข้อมูลที่เก็บค่าการติดตั้ง ที่ไฟล์ DataMac ซึ่งถ้าไม่มีไฟล์นี้ก็จะ
เปิดโปรแกรมไม่ได้
'Form1.ProgressBar1.Max = Form1.DataSentXYZ.Rows
'Form1.ProgressBar1.Value = 0
'Form1.ProgressBar1.Visible = True
SetBlockByte = 12000
If Not MDIForm1.MSComm1.PortOpen Then
 MDIForm1.MSComm1.PortOpen = True
End If
'MDIForm1.MSComm1.CommPort = DataCom      'กำหนดการติดต่อสื่อสารที่ port
MDIForm1.MSComm1.Settings = DataBoardRate & ",n,8,1"
With MDIForm1
.MSComm1.Output = Chr$(13)
delay
.MSComm1.Output = "t 0 0 " & DataXySpeed & " " & DataXYA & Chr$(13)
delay4
If BoardControl = False Then
MsgBox ("   ไม่สามาถส่งข้อมูลเข้า Board Control Motion    ")
 GoTo E1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End If
'.MSComm1.Output = "p " & DataPUDistance & " " & DataPUSpeed & " " & DataPUA & " " & -
DataPDDistance & " " & DataPDSpeed & " " & DataPDA & Chr$(13)
.MSComm1.Output = "p " & 0 & " " & DataPUSpeed & " " & DataPUA & " " & 0 & " " & DataPDSpeed & "
" & DataPDA & Chr$(13)
READY3
.MSComm1.Output = "f " & DataSpeedNoCut & " " & DataANoCut & Chr$(13)
READY3
.MSComm1.Output = "u" & Chr$(13)
DataPen = "u"
READY3
MDIForm1.MSComm1.Output = "i" & Chr$(13)
READY3
.MSComm1.Output = "k -10 1000 100" & Chr$(13)
READY3
.MSComm1.Output = "o" & Chr$(13)
READY3
End With
'MDIForm1.MSComm1.Output = "g" & Chr$(13)
For i = 1 To Form1.DataSentXYZ.Rows - 1
'Form1.ProgressBar1.Value = i
Xval = (Val(Form1.DataSentXYZ.TextMatrix(i, 1)))
Yval = (Val(Form1.DataSentXYZ.TextMatrix(i, 2)))
Zval = Val(Form1.DataSentXYZ.TextMatrix(i, 3))
 F = Val(Form1.DataSentXYZ.TextMatrix(i, 4))
If Abs(Xval) + Abs(Yval) = 0 Then GoTo L1
Positions_X = "t " & Xval & " " & Yval
If F <> Fsent Then
        Fsent = F
        DataPen = "d"
        MDIForm1.MSComm1.Output = "d" & Chr$(13)
        READY3
        MDIForm1.MSComm1.Output = "s " & Fsent & Chr$(13)
        READY3
End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
K1:
MDIForm1.MSComm1.Output = Positions_X & Chr$(13)
READY3
ByteCount = (ByteCount) + Len(Positions_X) + 3
MDIForm1.MSComm1.Output = "g" & Chr$(13)
READY3
L1:
If Zval <> 0 Then
   If DataPen = "d" Then
      DataPen = "u"
      MDIForm1.MSComm1.Output = "u" & Chr$(13)
      READY3
   End If
      MDIForm1.MSComm1.Output = "k -" & Abs(Zval) & " 1000 100" & Chr$(13)
   READY3
      If Sgn(Zval) = 1 Then
       For j = 1 To (Zval \ 10)
       MDIForm1.MSComm1.Output = "q" & Chr$(13)
       READY3
       Next j
      Else
      For j = 1 To Abs(Zval \ 10)
       MDIForm1.MSComm1.Output = "w" & Chr$(13)
       READY3
       Next j
      End If
  End If
      If DataPen = "u" Then
          MDIForm1.MSComm1.Output = "d" & Chr$(13)
          READY3
      End If

      If ByteCount > SetBlockByte Then    'จำนวนที่ส่งมากกว่าที่จะรับได้
        ByteCount = 0
        MDIForm1.MSComm1.Output = "%" & Chr$(13)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        READY3
        Sleep (100)
        MDIForm1.MSComm1.Output = "m" & Chr$(13)
        READY2   'ตรวจสอบว่ามีตัวอักษรเปิดบล็อก "a"
        MDIForm1.MSComm1.Output = "i" & Chr$(13)
        Sleep (10)
      End If
Next
MDIForm1.MSComm1.Output = "%" & Chr$(13)
Sleep (100)
MDIForm1.MSComm1.Output = "m" & Chr$(13)
Sleep (10)
E1:
If MDIForm1.MSComm1.PortOpen Then
 MDIForm1.MSComm1.PortOpen = False
End If
'Form1.ProgressBar1.Visible = False
End Sub
Public Sub RunStepTO()      'ทีละเส้น
Dim i As Single, j As Single, BlockByte As Integer, NumberBlock As Integer, Xval As Currency, Yval
As Currency
Dim PenSent As String * 1, ByteCount As Integer, DataPen As String * 1, F As Single
Dim DistancePu As Integer, OverRam As Boolean, Positions_X As String, Zval As Currency, Fsent As
Single
OpenFileDataMac                         'เปิดข้อมูลที่เก็บค่าการติดตั้ง ที่ไฟล์ DataMac ซึ่งถ้าไม่มีไฟล์นี้ก็จะ
เปิดโปรแกรมไม่ได้
With MDIForm1.CommonDialog1
      .DialogTitle = "Open File"
      .Filter = "Text Document(*.NC)|*.nc|All Files(*.*)|*.*"
      .CancelError = True
      .ShowOpen

End With
If Not MDIForm1.MSComm1.PortOpen Then
 MDIForm1.MSComm1.PortOpen = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End If
MDIForm1.MSComm1.Settings = DataBoardRate & ",n,8,1"
With MDIForm1
.MSComm1.Output = Chr$(13)
delay
.MSComm1.Output = "t 0 0 " & DataXySpeed & " " & DataXYA & Chr$(13)
delay4
If BoardControl = False Then
MsgBox ("   ไม่สามาถส่งข้อมูลเข้า Board Control Motion   ")
GoTo E1
End If
Form1.Label12.Visible = False
Form1.Label11.Visible = True
Form1.Shape5.BackColor = &HFF00&
.MSComm1.Output = "p " & 0 & " " & DataPUSpeed & " " & DataPUA & " " & 0 & " " & DataPDSpeed & "
" & DataPDA & Chr$(13)
READY
.MSComm1.Output = "f " & DataSpeedNoCut & " " & DataANoCut & Chr$(13)
READY
.MSComm1.Output = "u" & Chr$(13)
DataPen = "u"
READY
.MSComm1.Output = "o" & Chr$(13)
READY
End With
SentCancle = True
Form1.ProgressBar1.Max = Form1.DataSentXYZ.Rows - 1
Form1.ProgressBar1.Visible = True
L3:
OpenText_sent_one
GoTo E1
E1:
Form1.ProgressBar1.Visible = False
Form1.Shape4.BackColor = &HFF&
Form1.Shape5.BackColor = &HFF&
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Form1.Label8.Visible = False
Form1.Label9.Visible = True
Form1.Label11.Visible = False
Form1.Label12.Visible = True
If MDIForm1.MSComm1.PortOpen Then
 MDIForm1.MSComm1.PortOpen = False
End If
End Sub
Public Sub OpenText_sent_one()
Dim da1 As String
Dim i As Long, j As Long, data As String, data1 As String, Data2 As String, Data3 As String
Dim Gcode As String, Count As Long, INCorCEN As Integer, INCorCEN_pen As Single
Dim X As Single, Y As Single, z As Single, F As Single, Ig As Single, Jg As Single, Kg As Single
Dim Xole As Single, Yole As Single, Zole As Single
Dim S1 As Boolean, S2 As Integer
Dim X_old As Long, Y_old As Long, Z_old As Long
Dim i_ As Long, DataFeed As Single, DataFeedSent As Single
With Form1.DataXYZ
Form1.Label9.Visible = False
Form1.Label8.Visible = True
Form1.Shape4.BackColor = &HFF00&
.Clear: .Rows = 2: .Cols = 5
.TextMatrix(0, 0) = "Absolute": .TextMatrix(0, 1) = "X": .TextMatrix(0, 2) = "Y": .TextMatrix(0, 3) = "Z":
.TextMatrix(0, 4) = "F"
Count = 1
.TextMatrix(Count, 0) = Count
.TextMatrix(Count, 1) = 0
.TextMatrix(Count, 2) = 0
.TextMatrix(Count, 3) = 0
Count = Count + 1
.Rows = Count + 1
.TextMatrix(Count, 0) = Count
.Enabled = False
F = DataSpeedNoCut              'ตั้งค่าความเร็วสูงสุด
INCorCEN = DataStepXY           'ตัวตูณเพื่อเปลี่ยนจาก มิลลิเมตร เป็น สเต็ป
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Open MDIForm1.CommonDialog1.Filename For Input As #1
Do While Not EOF(1)
   Line Input #1, da1
   If Open_Cancle = True Then GoTo L1
    Data3 = da1 & " "                ' เก็บข้อมูล แถวของ จี ไว้ในตัวแปร Data3
          If InStr(Data3, "G") <> 0 Then
              data1 = Mid(Data3, (InStr(Data3, "G")), 3)
          End If
            Select Case data1
              Case "G00"
                        If InStr(Data3, "X") <> 0 Then X = XGcode(Data3) * INCorCEN
                        If InStr(Data3, "Y") <> 0 Then Y = YGcode(Data3) * INCorCEN
                        If InStr(Data3, "Z") <> 0 Then z = ZGcode(Data3) * INCorCEN_pen
                        If InStr(Data3, "F") <> 0 Then F = FGcode(Data3)
                        .TextMatrix(Count, 1) = X
                        .TextMatrix(Count, 2) = Y
                        .TextMatrix(Count, 3) = z
                        .TextMatrix(Count, 4) = 1000
                        Count = Count + 1
                        .Rows = Count + 1
                        .TextMatrix(Count, 0) = Count
                        GoTo L3
               Case "G01"
                   Gcode = data1
                        If InStr(Data3, "X") <> 0 Then X = XGcode(Data3) * INCorCEN
                        If InStr(Data3, "Y") <> 0 Then Y = YGcode(Data3) * INCorCEN
                        If InStr(Data3, "Z") <> 0 Then z = ZGcode(Data3) * INCorCEN_pen
                        If InStr(Data3, "F") <> 0 Then F = FGcode(Data3)
                        .TextMatrix(Count, 1) = X
                        .TextMatrix(Count, 2) = Y
                        .TextMatrix(Count, 3) = z
                        .TextMatrix(Count, 4) = F
                        Count = Count + 1
                        .Rows = Count + 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                .TextMatrix(Count, 0) = Count
                GoTo L3
        Case "G02"
            Gcode = data1
                Xole = X: Yole = Y: Zole = z
                If InStr(Data3, "X") <> 0 Then X = XGcode(Data3) * INCorCEN
                If InStr(Data3, "Y") <> 0 Then Y = YGcode(Data3) * INCorCEN
                If InStr(Data3, "Z") <> 0 Then z = ZGcode(Data3) * INCorCEN_pen
                If InStr(Data3, "F") <> 0 Then F = FGcode(Data3)
                If InStr(Data3, "I") <> 0 Then Ig = IGcode(Data3) * INCorCEN
                If InStr(Data3, "K") <> 0 Then Kg = KGcode(Data3) * INCorCEN
                If InStr(Data3, "J") <> 0 Then Jg = JGcode(Data3) * INCorCEN
                .TextMatrix(Count, 1) = X
                .TextMatrix(Count, 2) = Y
                .TextMatrix(Count, 3) = z
                .TextMatrix(Count, 4) = F
                Count = G02(Val(.TextMatrix(Count - 1, 1)), Val(.TextMatrix(Count - 1, 2)),
Val(.TextMatrix(Count - 1, 3)), X, Y, z, Ig, Jg, Kg, Count, 1, Xole, Yole, Zole)
                Count = Count + 1
                .Rows = Count + 1
                .TextMatrix(Count, 0) = Count
                GoTo L3
        Case "G03"
                Xole = X: Yole = Y: Zole = z
                If InStr(Data3, "X") <> 0 Then X = XGcode(Data3) * INCorCEN
                If InStr(Data3, "Y") <> 0 Then Y = YGcode(Data3) * INCorCEN
                If InStr(Data3, "Z") <> 0 Then z = ZGcode(Data3) * INCorCEN_pen
                If InStr(Data3, "F") <> 0 Then F = FGcode(Data3)
                If InStr(Data3, "I") <> 0 Then Ig = IGcode(Data3) * INCorCEN
                If InStr(Data3, "K") <> 0 Then Kg = KGcode(Data3) * INCorCEN
                If InStr(Data3, "J") <> 0 Then Jg = JGcode(Data3) * INCorCEN
                .TextMatrix(Count, 1) = X
                .TextMatrix(Count, 2) = Y
                .TextMatrix(Count, 3) = z
                .TextMatrix(Count, 4) = F
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    Count = G03(Val(.TextMatrix(Count - 1, 1)), Val(.TextMatrix(Count - 1, 2)),
Val(.TextMatrix(Count - 1, 3)), X, Y, z, Ig, Jg, Kg, Count, 1, Xole, Yole, Zole)
                    Count = Count + 1
                    .Rows = Count + 1
                    .TextMatrix(Count, 0) = Count
                Case "G20"
                  INCorCEN = 20 * 100          'G20 หน่วยเป็นนิ้ว 0.001 นิ้ว ต่อ 0.02 มิลลิเมตร
                Case "G21"
                  INCorCEN = 100               'G21 หน่วยเป็นมิลลิเมตร
                  INCorCEN_pen = DataStepPen
              End Select
L2:
L3:
DoEvents
  With Form1
    For i_ = 1 To .DataXYZ.Rows - 3
              .DataSentXYZ.Rows = i_ + 1
              .DataSentXYZ.TextMatrix(i_, 0) = i_
              .DataSentXYZ.TextMatrix(i_, 1) = Round((Val(.DataXYZ.TextMatrix(i_ + 1, 1)) - Val
(.DataXYZ.TextMatrix(i_, 1))), 0)
              .DataSentXYZ.TextMatrix(i_, 2) = Round((Val(.DataXYZ.TextMatrix(i_ + 1, 2)) - Val
(.DataXYZ.TextMatrix(i_, 2))), 0)
              .DataSentXYZ.TextMatrix(i_, 3) = Round((Val(.DataXYZ.TextMatrix(i_ + 1, 3)) - Val
(.DataXYZ.TextMatrix(i_, 3))), 0)
              DataFeed = Val(.DataXYZ.TextMatrix(i, 4))
              If DataFeedSent <> DataFeed Then
                If DataFeed < 3000 Then DataFeed = DataXySpeed
                DataFeedSent = DataFeed
              End If
              .DataSentXYZ.TextMatrix(i_, 4) = DataFeed
    Next i_
    End With
    Form1.DataXYZ.Rows = 2
    Form1.DataXYZ.TextMatrix(1, 1) = X
      Form1.DataXYZ.TextMatrix(1, 2) = Y
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Form1.DataXYZ.TextMatrix(1, 3) = z
            Count = 2
        Form1.DataXYZ.Rows = 3
RunStep
Loop
L1:
Close #1
  End With
        Form1.DataXYZ.Rows = 1
End Sub
Public Sub AbsToInc_1()
   Dim i As Long, DataFeed As Single, DataFeedSent As Single
   On Error GoTo Error
   With Form1
      For i = 1 To .DataXYZ.Rows - 2
               .DataSentXYZ.Rows = i + 1
               .DataSentXYZ.TextMatrix(i, 0) = i
               .DataSentXYZ.TextMatrix(i, 1) = Round((Val(.DataXYZ.TextMatrix(i, 1)) - Val
(.DataXYZ.TextMatrix(i - 1, 1))), 0)
               .DataSentXYZ.TextMatrix(i, 2) = Round((Val(.DataXYZ.TextMatrix(i, 2)) - Val
(.DataXYZ.TextMatrix(i - 1, 2))), 0)
               .DataSentXYZ.TextMatrix(i, 3) = Round((Val(.DataXYZ.TextMatrix(i, 3)) - Val
(.DataXYZ.TextMatrix(i - 1, 3))), 0)
               DataFeed = Val(.DataXYZ.TextMatrix(i, 4))
               If DataFeedSent <> DataFeed Then
                 If DataFeed < 3000 Then DataFeed = DataXySpeed
                 DataFeedSent = DataFeed
               End If
               .DataSentXYZ.TextMatrix(i, 4) = DataFeed
      Next i
   End With
Error:
End Sub
Public Sub RunStep()      'ทีละเส้น
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim i As Single, j As Single, BlockByte As Integer, NumberBlock As Integer, Xval As Currency, Yval
As Currency
Dim PenSent As String * 1, ByteCount As Integer, DataPen As String * 1, F As Single
Dim DistancePu As Integer, OverRam As Boolean, Positions_X As String, Zval As Currency, Fsent As
Single
For i = 1 To Form1.DataSentXYZ.Rows - 1
Xval = (Val(Form1.DataSentXYZ.TextMatrix(i, 1))) '* Abs(Val(PoltScalSET))
Yval = (Val(Form1.DataSentXYZ.TextMatrix(i, 2))) '* Abs(Val(PoltScalSET))
Zval = (Val(Form1.DataSentXYZ.TextMatrix(i, 3)))
F = (Val(Form1.DataSentXYZ.TextMatrix(i, 4)))
'Form1.ProgressBar1.Value = i
If Abs(Xval) + Abs(Yval) = 0 Then GoTo L1
 Positions_X = "t " & Xval & " " & Yval
    If F <> Fsent Then
        Fsent = F
          If DataPen = "u" Then
            DataPen = "d"
            MDIForm1.MSComm1.Output = "d" & Chr$(13)
            READY
        End If
        MDIForm1.MSComm1.Output = "s" & Fsent & Chr$(13)
        READY
    End If
MDIForm1.MSComm1.Output = Positions_X & Chr$(13)
READY
MDIForm1.MSComm1.Output = "g" & Chr$(13)
READY1
'GoTo L2
L1:
If Zval <> 0 Then
  MDIForm1.MSComm1.Output = "k -" & Abs(Zval) & " 500 100" & Chr$(13)
  READY
        MDIForm1.MSComm1.Output = "u" & Chr$(13)
        READY
    If Sgn(Zval) = 1 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MDIForm1.MSComm1.Output = "q" & Chr$(13)
            READY
        Else
            MDIForm1.MSComm1.Output = "w" & Chr$(13)
            READY
        End If
    End If
    L2:
    Next i
    End Sub
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้