สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การออกแบบและการวางแผนการผลิตด้วยคอมพิวเตอร์

DESIGNING AND PRODUCTION PLANNING BY COMPUTER

โดย

นางสาว อัญชนา	แซ่โอ้ว
นาย คณิน	ศรีธันดร
นาย ธนวัฒน์	บุญปัญญาวุฒิ
นาย นราดล	วัฒนะชัยยันต์
นางสาว ปิยนุช	สิงหะคเชนทร์
นางสาว วนา	หงษ์คำดี
นาย วิทยา	ทองสุข
นาย สิริศักดิ์	สว่างจิตร

ส4ว.
ด734ก
2549

เลขหมู่.....
เลขทะเบียน 72140
วัน,เดือน,ปี 11 มิ.ย. 2550

.b 11763917
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การออกแบบและการวางแผนการผลิตด้วยคอมพิวเตอร์

DESIGNING AND PRODUCTION PLANNING BY COMPUTER

ผู้จัดทำ

นางสาว อัญชนา	แซ่โอ้ว	45010959
นาย คณิน	ศรีธันคร	46010080
นาย ธนวัฒน์	บุญปัญญาวุฒิ	46010284
นาย นราดล	วัฒนะชัยยันต์	46010341
นางสาว ปิยนุช	สิงหะคเชนทร์	46010455
นางสาว วนา	หงษ์คำดี	46010647
นาย วิทยา	ทองสุข	46010706
นาย สิริศักดิ์	สว่างจิตร	46010838

....อาจารย์ที่ปรึกษา

(อาจารย์เทพจิตร์ เชยโภคา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบและการวางแผนการผลิตด้วยคอมพิวเตอร์

โดย

นางสาว อัญชนา	แซ่โอ้ว	45010959
นาย คณิน	ศรีธันดร	46010080
นาย ธนวัฒน์	บุญปัญญาวุฒิ	46010284
นาย นราดล	วัฒนะชัยยันต์	46010341
นางสาว ปิยนุช	สิงหะคเชนทร์	46010455
นางสาว วนา	หงษ์คำดี	46010647
นาย วิทยา	ทองสุข	46010706
นาย สิริศักดิ์	สว่างจิตร	46010838

อาจารย์ที่ปรึกษา

อาจารย์เทพจิตร์ เชยโภคา

ปีการศึกษา 2549

บทคัดย่อ

ปัจจุบันมีการนำเอาเครื่องจักร CNC เข้ามาใช้ในอุสาหกรรมรมอย่างกว้างขวางเนื่องจากสามารถทำงานได้มาตรฐาน รวดเร็ว และไม่จำเป็นต้องใช้ช่างที่มีฝีมือมากนัก และเนื่องจากเครื่องจักรประเภท CNC มีราคาสูงมาก ดังนั้นเพื่อเป็นการลดต้นทุนการผลิตโครงงานนี้จึงนำเสนอการปรับปรุงเครื่องจักรในระบบให้แรงงานคนควบคุมมาเป็นเครื่องจักรที่สามารถทำงานเทียบเท่ากับเครื่องจักรชนิด CNC โดยการสร้างโปรแกรมสำหรับเครื่องกัด และทำการปรับปรุงเครื่องจักเก่าให้สามารถทำงานในระบบ CNC ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DESIGNING AND PRODUCTION PLANING BY COMPUTER

By

Anchana Sae-aow

Kanin Sritundorn

Thanawat Boonpanyavut

Naradon Watanachaiyan

Piyanoot Singhakachen

Wana Hongkumdee

Withaya Thongsook

Sirisak Sawangjit

Advisor

Thepajit Cheypoca

Academic Year 2006

ABSTRACT

Nowadays CNC machine has been widely used throughout industrial business due to its ability to make a standard production with speedily and needs no craftsman but a few technicians. However, a price of the CNC machine is quiet high. Thus in order to decrease a production cost, this project has proposed to improve an old-manual machine for the likewise CNC machine; by doing that, the machine structure has been adjusted and an application software has been developed to automate the machine.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำปริญญานิพนธ์ฉบับนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความช่วยเหลือเป็นอย่างดี จาก อ.เทพจิตร์ เชยโภคา ที่ได้กรุณาให้คำปรึกษาแนะนำที่ดีมาโดยตลอดตั้งแต่ต้น รวมทั้งเอื้อเฟื้ออุปกรณ์ที่จำเป็น และความช่วยเหลืออื่นๆที่เป็นประโยชน์ต่อโครงงาน ผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูง

ขอบคุณเพื่อนๆ ทุกคนที่ให้กำลังใจ สนับสนุนอุปกรณ์ที่ขาดเหลือ กระตุ้นเตือน รวมทั้งคอยถามไถ่ความคืบหน้าของโครงงานอยู่เสมอ

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่คอยเป็นกำลังใจที่ดีตลอดมา รวมถึงการสนับสนุนในเรื่องของงบประมาณที่ขาดเหลือ ตลอดจนเป็นแรงบันดาลใจที่ดีที่สุดที่ทำให้โครงงานนี้สำเร็จสมบูรณ์ลงได้

ผู้จัดทำ

นางสาว อัญชนา แซ่โอ้ว

นาย คณิน ศรีธันดร

นาย ธนวัฒน์ บุญปัญญาวุฒิ

นาย นราดล วัฒนะชัยยันต์

นางสาว ปิยนุช สิงหะคเชนทร์

นางสาว วนา หงษ์คำดี

นาย วิทยา ทองสุข

นาย สิริศักดิ์ สว่างจิตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VII
สารบัญตาราง	X
บทที่ 1 บทนำ	1
1.1 กล่าวนำ	1
1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์	2
1.3 ขั้นตอนการศึกษาและการจัดทำโครงงาน	2
1.4 รายละเอียดของปริญญานิพนธ์	4
บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง	5
2.1 ความรู้เบื้องต้นของการเขียนโปรแกรมแปลงไฟล์ DXF เป็นไฟล์ G-CODE	5
2.1.1 ความรู้เบื้องต้นเกี่ยวกับไฟล์ DXF	5
2.1.2 ความรู้เบื้องต้นเกี่ยวกับ G-CODE	8
2.2 การเขียนโปรแกรมติดต่อกับพอร์ตขนานด้วย Visual Basic 6	9
2.2.1 การติดตั้งไฟล์ inpout32.dll	9
2.2.2 การประกาศฟังก์ชันเพื่อเรียกใช้ไฟล์ inpout32.dll ใน visual basic6	9
2.2.3 การเขียนคำสั่งเพื่อรับ-ส่งข้อมูลใน Visual Basic 6	10
2.3 พอร์ตขนาน (PARALLEL PORT)	10
2.3.1 DATA PORT	10
2.3.2 STATUS PORT	11
2.3.3 CONTROL PORT	11
2.4 ความรู้เบื้องต้นสำหรับการพัฒนาเครื่องจักร	15
2.4.1 มอเตอร์แบบกระตุ้นแยกอิสระ (Separately exited)	15
2.4.1.1 การปรับความเร็วของมอเตอร์กระแสตรง	16
บทที่ 3 หลักการออกแบบ	22
3.1 ส่วนโปรแกรม	22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

3.1.1 โปรแกรมส่วนการแปลงไฟล์ (DXF to G-Code Converter) 22

3.1.1.1 Tools Window (1) 23

3.1.1.2 G-CODE Window (2) 24

3.1.1.3 Work Space Window (3) 30

3.1.2 โปรแกรมส่วนการควบคุมเครื่องจักร (MANUAL FORM) 33

3.1.2.1 โปรแกรมส่วนการควบคุมเครื่องจักร (MANUAL FORM) 33

3.1.2.2 MANUAL DATA INPUT 37

3.1.2.3 POSITION 0 40

3.1.2.4 MANUAL BY MOUSE DOWN 41

3.1.2.5 POSITION STATUS 43

3.1.2.6 AXIS STATUS 44

3.1.2.7 SETTINGS 44

3.2 การรับค่าจากไฟล์ G-CODE เพื่อประมวลคำสั่ง 45

3.2.1 หลักการพิจารณาค่าG-Code ที่อ่านเข้ามา 45

3.2.2 ขั้นตอนการนำค่าที่อ่านได้ไปใช้ 47

3.3 การคำนวณหาจำนวนพัลซ์ที่จะทำให้แกน X และแกน Y เคลื่อนที่ 51

3.3.1 การคำนวณหาจำนวนพัลซ์ที่จะทำให้แกน Z เคลื่อนที่ได้ระยะ 1 มิลลิเมตร 52

3.4 ส่วนการปรับปรุงเครื่องจักร 53

3.5 ส่วนวงจรควบคุมเครื่องเจาะ 56

3.5.1 Emergency switch 56

3.5.2 Speed control Dc motor circuit 57

3.5.3 วงจรตัดการทางาน Dc Motor 58

3.5.4 Limit Switch 58

บทที่ 4 การทดลองและผลการทดลอง 60

4.1 การทดลองสั่งการเครื่องจักร โดยตรงผ่านหน้าต่าง MANUAL FORM 60

4.2 การทดลองกัดรูปเส้นตรง 63

4.3 การทดลองกัดรูปวงกลม 66

บทที่ 5 บทวิจารณ์และบทสรุป 67

5.1 สรุปผลการทดลอง 67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
5.2 วิจารณ์ผลการทดลอง	67
5.3 ปัญหาที่พบในการทำโครงงานนี้	67
5.4 แนวทางในการพัฒนา	67
ภาคผนวก ก โปรแกรม MANUAL FORM	69
ภาคผนวก ข โปรแกรม DXF to G-Code Converter	105
ภาคผนวก ค รูปการออกแบบเครื่องกัด	139
ภาคผนวก ง เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์	145
ง.1 เอกสารคู่มือการใช้งาน ULN2001A-ULN2002A	145
ง.2 เอกสารคู่มือการใช้งาน LM124	153
เอกสารอ้างอิง	173

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่ **หน้า**

1.1 แสดง Project Flow 3

2.1 การใช้งานขาพอร์ตต่างๆ 13

2.2 แสดงการนำค่าที่ส่งกลับมาใช้ในการตัดการทำงานของปุ่มต่างๆ 13

2.3 แสดงการบังคับแต่ละแกน 14

2.4 แสดงวงจรสมมูลของมอเตอร์แบบกระตุ้นอิสระ(Separately exited) 15

2.5 แสดงกราฟลักษณะสมบัติภายนอก(External Characteristics) 15

2.6 แสดงการทำงานของมอเตอร์แบบกระตุ้นอิสระในสภาวะอยู่ตัวใดๆ 16

2.7 แสดงโครงสร้าง DC Motor 17

2.8 D.C. Motor Circuit 19

2.9 แผนภาพบล็อกของ armature – controlled motor 20

2.10 แผนภาพบล็อก ของ Field – controlled motor 21

3.1 แสดงหน้าต่างของโปรแกรมส่วนการแปลงไฟล์ 22

3.2 แสดงหน้าต่าง Tools 23

3.3 แสดงหน้าต่างการนำไฟล์ DXF เข้ามาแปลงเป็นไฟล์ NC 24

3.4 แสดงหน้าต่าง G-CDOE 24

3.5 แสดงไฟล์ G-code ที่ได้จากการแปลงไฟล์ DXF พร้อมทั้งรูปภาพ 29

3.6 แสดงหน้าต่าง Work Space 30

3.7 แสดงหน้าต่างโปรแกรมควบคุมเครื่องจักร 33

3.8 ตำแหน่งของJOGและOption7.value 34

3.9 แสดง ตำแหน่งของTextต่างๆ 35

3.10 แสดงMANUAL DATA INPUT 37

3.11 แสดงค่าตัวแปรต่างๆในสมการเส้นตรง 37

3.12 แสดงส่วนควบคุม POSITION0 40

3.13 แสดง MANUAL BY MOUSEDOWN และ DATA FROM H379 PORT 42

3.14 แสดง POSITION STATUS 43

3.15 AXIS STATUS 44

3.16 แสดง SETTINGS 44

3.17 แสดงการอ่านค่าไฟล์ G-CODE เพื่อนำไปประมวลคำสั่ง 45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

รูปที่	หน้า
3.18 แสดงการอ่านค่าไฟล์ G-CODE ที่เป็นจำนวนของข้อมูล	46
3.19 แสดงการอ่านค่าตัวแปรที่แตกต่างกัน	46
3.20 แสดงตัวแปรที่มีค่าไม่เหมือนกัน	46
3.21 แสดงลักษณะข้อมูลของไฟล์ G-CODE	47
3.22 แสดง การนับข้อมูล	47
3.23 แสดงการนับช่องว่าง	48
3.24 แสดงการนับตัวแปร	49
3.25 แสดงข้อมูลของสเตปมอเตอร์	51
3.26 แสดงระยะพิทของสกรู	52
3.27 แสดงการทำงานของเฟือง	52
3.28 ภาพประกอบโดยสมบูรณ์ของเครื่องกัดก่อนมีการแก้ไขปรับปรุง	53
3.29 มอเตอร์ตัวเก่าที่ทำให้การเคลื่อนที่ของเครื่องกัดมีอัตราทด 12.5:1	54
3.30 มอเตอร์ตัวใหม่ที่ทำให้การเคลื่อนที่ของเครื่องกัดมีอัตราทด 1:1	55
3.31 ตำแหน่งของลิมิตสวิตซ์ที่ติดไว้เพื่อหยุดการทำงานของมอเตอร์	56
3.32 แสดงวงจร Emergency switch	57
3.33 Speed control dc motor circuit	57
3.34 วงจรตัดการทางาน Dc Motor	58
3.35 แสดงรูป Limit switch	59
4.1 แสดงเครื่องกัดที่ติดตั้งเสร็จสมบูรณ์	60
4.2 แสดงการตั้งค่าจุดเริ่มต้นในหน้าต่าง MANUAL FORM ที่ใช้สั่งการเครื่องกัดโดยตรง	61
4.3 แสดงการตั้งค่าจุดสุดท้ายในหน้าต่าง MANUAL FORM ที่ใช้สั่งการเครื่องกัดโดยตรง	61
4.4 แสดงตำแหน่งของเครื่องกัดที่จุดเริ่มต้น	62
4.5 แสดงตำแหน่งของเครื่องกัดที่จุด X10 Y10	62
4.6 แสดงรูปภาพเส้นตรงที่ออกแบบในโปรแกรม AutoCAD	63
4.7 แสดงหน้าต่าง DXF to G-Code Converter ที่ใช้สั่งการกัดเส้นตรง	64
4.8 แสดงรูปเส้นตรงบนวัสดุที่ได้จากการทำงานของเครื่องกัด	64
4.9 แสดงรูปภาพวงกลมที่ออกแบบในโปรแกรม AutoCAD	65
4.10 แสดงหน้าต่าง DXF to G-Code Converter ที่ใช้สั่งการกัดวงกลม	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

รูปที่ **หน้า**

4.11 แสดงรูปวงกลมบนวัสดุที่ได้จากการทำงานของเครื่องกัด 66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 ข้อมูลที่จำเป็นในการสร้างโปรแกรมแปลงไฟล์จากรูปวงกลม	6
2.2 ข้อมูลที่จำเป็นในการสร้างโปรแกรมแปลงไฟล์จากรูปส่วนหนึ่งของวงกลม	7
2.3 ข้อมูลที่จำเป็นในการสร้างโปรแกรมแปลงไฟล์จากรูปเส้นตรง	7
2.4 แสดงรูปแบบ G-CODE ตามแบบของ Fanuc	8
2.5 แสดงรายละเอียดของ DATA PORT	10
2.6 แสดงรายละเอียดของ STATUS PORT	11
2.7 แสดงรายละเอียดของ CONTROL PORT	12
2.8 สรุปการนำแต่ละพินของพอร์ตขนานมาประยุกต์ใช้กับโปรเจกต์	14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 กล่าวนำ

เนื่องด้วยในส่วนของภาคอุตสาหกรรมจำนวนมากมีความต้องการที่จะปรับปรุงระบบการผลิตให้ดีขึ้น ทั้งด้านคุณภาพ และ ปริมาณของชิ้นงาน ซึ่งแต่เดิมกระบวนการการผลิตจะใช้เครื่องจักรที่มีความจำเป็นต้องใช้ช่างฝีมือในการทำงาน จึงทำให้ความสามารถในการสร้างชิ้นงานขึ้นกับปัจจัยหลายๆอย่าง และปัจจัยหนึ่งที่สำคัญก็คือฝีมือของช่างแต่ละคนไม่เท่ากัน ทำให้ความสามารถในด้านการผลิตผลิตภัณฑ์ที่มีลักษณะเหมือนๆกันในปริมาณมากๆนั้นมีประสิทธิภาพลดลง เป็นสาเหตุให้คุณภาพมาตรฐานการของการผลิตผลิตภัณฑ์ชนิดนั้นๆมีค่าไม่เท่ากัน ดังนั้น จึงมีการสร้างเครื่องจักรที่สามารถใช้คอมพิวเตอร์เข้ามาควบคุมกระบวนการทำงาน เรียกว่า เครื่องจักรซีเอ็นซี (Computer Numerical Control; CNC) โดยการทำงานนั้นสามารถกำหนดโปรแกรมให้เครื่องจักรผลิตชิ้นงานที่ต้องการเป็นจำนวนมากๆได้ โดยที่ได้คุณภาพของงานตามที่ต้องการทุกชิ้น และยังสามารถเปลี่ยนรูปแบบของชิ้นงานได้โดยง่าย เพียงแค่อาศัยการเปลี่ยนโปรแกรมการสั่งงานเท่านั้น ทำให้ไม่จำเป็นต้องอาศัยช่างฝีมือที่ชำนาญการมากนักในการทำงาน เพียงแต่ต้องการช่างที่สามารถควบคุมการทำงานของเครื่องจักรให้ได้เท่านั้น ทำให้การทำงานนั้นสะดวกขึ้น แต่เนื่องจากเครื่องจักรซีเอ็นซีนั้น เป็นเครื่องจักรที่ราคาสูงมาก ดังนั้น ในภาคอุตสาหกรรมขนาดเล็ก จึงไม่สามารถนำเครื่องจักรซีเอ็นซีมาใช้งานได้ เพราะจะเป็นการเพิ่มต้นทุนให้สูงมากเกินไป จะทำให้ไม่คุ้มค่าในการลงทุน เนื่องจากในอุตสาหกรรมหลาย ๆ รูปแบบนั้นก็มีเครื่องจักรเดิมไว้ใช้งานอยู่แล้ว จะพบว่าถ้าเราปรับปรุงเครื่องจักรเก่าที่ยังสามารถใช้งานได้ มาทำการประยุกต์ติดตั้งส่วนของระบบควบคุม ที่สามารถควบคุมได้ด้วยเครื่องคอมพิวเตอร์ จะทำให้เครื่องจักรตัวนั้นสามารถทำงานได้ในลักษณะเทียบเท่ากับเครื่องซีเอ็นซีที่มีราคาแพง

ด้วยเหตุผลดังกล่าวจึงได้มีการออกแบบสร้าง Milling Machine (เครื่องกัด) และสร้างโปรแกรมออกแบบพร้อมทั้งสามารถแปลงชุดคำสั่ง G-CODE ให้อยู่ในรูปแบบของชุดคำสั่งแบบพิกัด X-Y-Z ที่สามารถนำไปใช้ได้กับเครื่อง Milling Machine ที่สร้างขึ้นซึ่งเป็น EMC (Enhance Machine Control) ชนิดหนึ่ง สามารถใช้เป็นต้นแบบในการปรับปรุงเครื่องจักรเก่าที่ใช้งานได้ให้ทำงานในลักษณะเทียบเท่ากับเครื่องซีเอ็นซีที่มีราคาแพง

EMC (Enhance Machine Control) เป็นโปรแกรมที่สร้างมาโดยมีจุดมุ่งหมายเพื่อที่จะควบคุมเครื่อง CNC โดยเฉพาะ มีข้อดีคือ สามารถปรับใช้ได้กับเครื่องสมัยใหม่ โดยไม่ต้องเสียค่าใช้จ่ายทางด้านโปรแกรมในราคาที่สูง สามารถทำงานได้ตามที่ต้องการและยังสามารถปรับปรุงตัวโปรแกรมได้ง่ายต่อการใช้งานในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในปัจจุบัน Software ที่ควบคุม CNC เป็น Software ที่มีราคาแพง และบางตัวต้องใช้งานกับเครื่องจักรที่ออกแบบมาโดยเฉพาะเท่านั้น หรือถ้าหากเป็น Software ที่ทำงานได้กับเครื่องจักรแบบต่างๆ ก็ยังคงต้อง RUN Program ในระบบปฏิบัติการ Linux เท่านั้น เช่น Linux EMC CNC ซึ่งปัจจุบันระบบปฏิบัติการ Windows เป็นที่นิยมมากกว่า Application จำพวก Program cad-cam , AutoCAD , Solid Work หรือ Program cad-cam อื่นๆ นั้นก็จะถูกใช้งานบนระบบปฏิบัติการ Windows มากกว่า ซึ่งการที่จะถ่ายโอนงานจากระบบปฏิบัติการหนึ่งไปอีกระบบหนึ่งนั้นเป็นการยุ่งยาก เราจึงทำการออกแบบ EMC ขึ้นมาเพื่อใช้กับระบบปฏิบัติการ Windows

1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์

1. ศึกษาวิธีการสร้าง และควบคุมเครื่องกัดเพื่อใช้สำหรับงานกัด 2 มิติ
2. ศึกษา และสร้างโปรแกรมควบคุมเครื่องกัดทั้งส่วนการควบคุมโดยตรง และการควบคุมอัตโนมัติ โดยใช้โปรแกรมวิชวลเบสิก 6 (Visual Basic 6)
3. สามารถวิเคราะห์ปรับปรุงตัวเครื่องจักรและโปรแกรมเบื้องต้นได้

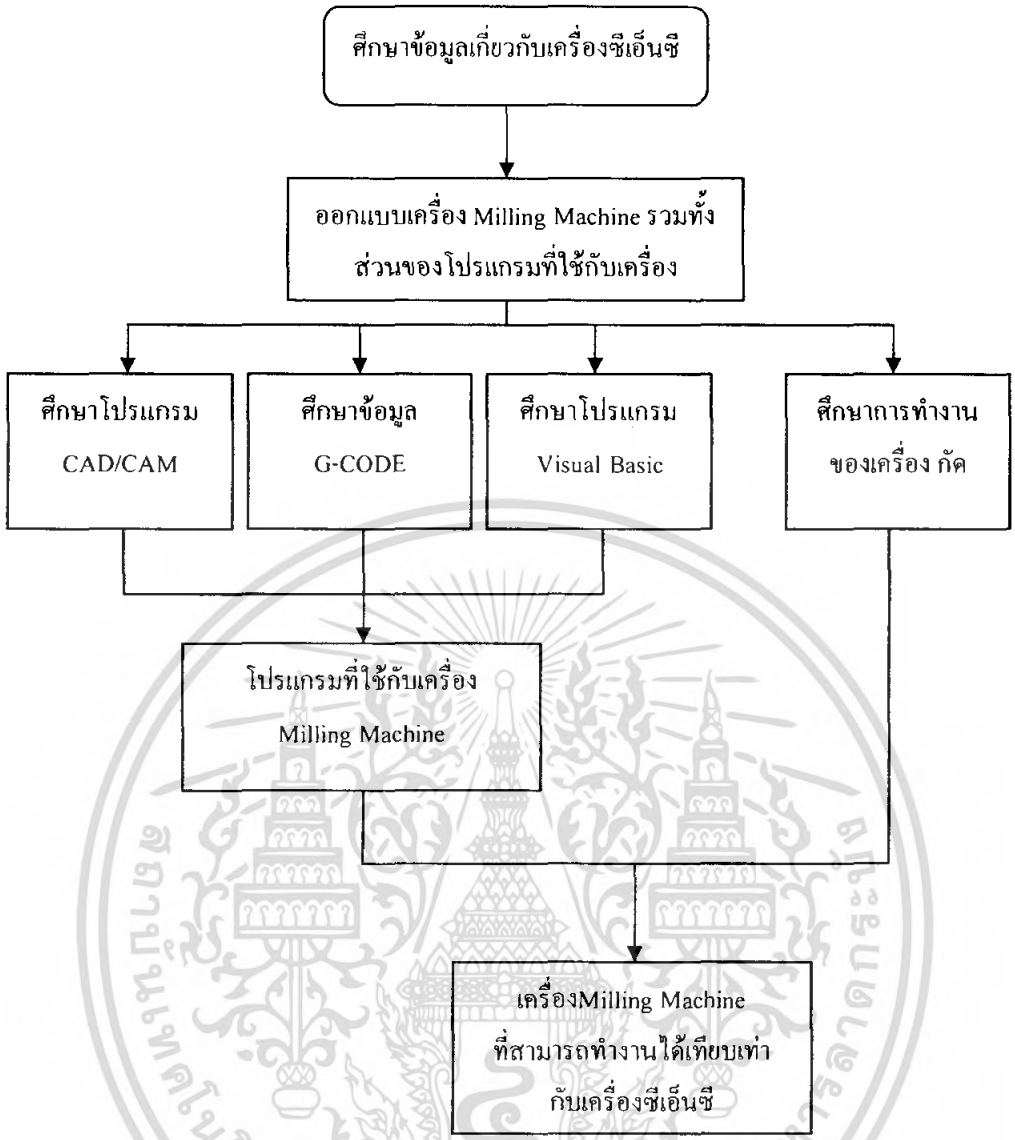
1.3 ขั้นตอนการศึกษาและการจัดทำโครงงาน

การทำโครงงานนี้มีขั้นตอนเริ่มจากการทำการกำหนดขอบเขต และความต้องการของงานที่จะนำเอาเครื่องกัดไปใช้ ขอบเขตของโปรแกรมที่จะสร้างขึ้นเพื่อใช้ในการควบคุมเครื่องกัด วิธีการพัฒนาเครื่องกัดจากเครื่องจักรเดิม รวมไปถึงการศึกษาวิธีการสร้างโปรแกรมที่ใช้สำหรับการสร้างโปรแกรมควบคุมเครื่องกัด

ทำการสร้างโปรแกรมควบคุมเครื่องจักรโดยทำการทดลองส่งข้อมูลผ่าน Port printer ไปยังตู้ควบคุมเครื่องจักร และทำการตรวจสอบเครื่องจักรว่าสมารถทำงานได้ตามต้องการหรือไม่ ทำการสร้างโปรแกรมแปลงไฟล์ DXF เป็นไฟล์ G-CODE และตรวจสอบความถูกต้อง ทั้งการแปลงไฟล์และการประมวลไฟล์ G-CODE ไปเป็นสัญญาณควบคุมเครื่องกัด ซึ่งได้ทำการปรับปรุงเครื่องกัดในส่วนต่างๆไปพร้อมๆกัน

ทดลองและตรวจสอบความถูกต้องของการทำงานของโปรแกรม การทำงานของเครื่องกัดว่าทำงานได้ถูกต้องตามต้องการหรือไม่แล้วทำการปรับปรงแก้ไขให้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 แสดง Project Flow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 รายละเอียดของปริญญานิพนธ์

เนื้อหาที่จะกล่าวในปริญญานิพนธ์ฉบับนี้ประกอบด้วย

บทที่ 1 บทนำ กล่าวนำถึงวัตถุประสงค์ หลักการ ขั้นตอนการศึกษา และการจัดทำโครงงาน พร้อมทั้งรายละเอียดของปริญญานิพนธ์ในแต่ละบท

บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง กล่าวถึงหลักการและทฤษฎีที่เกี่ยวข้องในการสร้างโปรแกรมแปลงไฟล์ DXF เป็น G-CODE หลักการสร้างและออกแบบโปรแกรมควบคุมเครื่องกัดโดยตรง และหลักการปรับปรุงเครื่องกัด

บทที่ 3 หลักการสร้างและออกแบบโปรแกรมแปลงไฟล์ DXF เป็น G-CODE หลักการสร้างและออกแบบโปรแกรมควบคุมเครื่องกัดโดยตรง และหลักการปรับปรุงเครื่องกัด

บทที่ 4 การทดลอง เป็นส่วนการทดลองใช้งานโปรแกรมที่สร้างขึ้นกับเครื่องกัดที่ได้ทำการพัฒนาขึ้นมา

บทที่ 5 บทวิจารณ์และสรุป จะสรุปผลการดำเนินงาน ปัญหาที่เกิดขึ้น และแนวทางการปรับปรุงพัฒนาโครงงานนี้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและความรู้ที่เกี่ยวข้อง

การออกแบบและสร้างเครื่องกัดนั้น จำเป็นต้องศึกษาองค์ประกอบต่างๆที่เกี่ยวข้องต่อการสร้างทั้งส่วนของโปรแกรมและการออกแบบที่สนใจให้เข้าใจเสียก่อน ดังนั้น ในบทนี้จะศึกษาและอธิบายถึงองค์ประกอบต่างๆ ที่จะนำไปใช้งานจริงในการสร้างเครื่องกัด ประกอบด้วยเนื้อหา ดังนี้ โปรแกรมแปลงไฟล์จากไฟล์ DXF เป็นไฟล์ NC (G-CODE) ที่ใช้ในการสั่งการเครื่องกัด โปรแกรมสั่งการเครื่องกัดรวมถึงการปรับปรุงเครื่องกัด

2.1 ความรู้เบื้องต้นของการเขียนโปรแกรมแปลงไฟล์ DXF เป็นไฟล์ G-CODE

ความรู้ที่นำมาใช้ในการเขียนโปรแกรมนี้นั้น นับเป็นส่วนสำคัญมาก ที่จำเป็นจะต้องศึกษาเพื่อให้สามารถเกิดแนวคิดในการสร้างโปรแกรม โดยต้องทำความเข้าใจเกี่ยวกับหลักการของการสร้างระบบภาษา DXF และ G-CODE เบื้องต้น ซึ่งประกอบไปด้วยหัวข้อต่างๆ ดังนี้

2.1.1 ความรู้เบื้องต้นเกี่ยวกับไฟล์ DXF

ไฟล์ DXF เป็นไฟล์ชนิดหนึ่งที่ใช้ในการแลกเปลี่ยนข้อมูลของรูปวาดที่วาดโดยโปรแกรม AutoCAD กับโปรแกรมอื่นๆ โดยสามารถใช้โปรแกรม Text-editor อ่านได้ ซึ่งรูปแบบของไฟล์นั้นมีสี่แบบ คือ ASCII DXF Files, binary DXF, Slide (SLD), และ Slide Library (SLB) รูปแบบที่นำมาใช้กันอย่างกว้างขวาง คือ ASCII DXF Files และ binary DXF ในการทำงานครั้งนี้ได้นำเอารูปแบบ ASCII DXF Files มาใช้

ไฟล์ ASCII DXF Files ประกอบไปด้วยข้อมูลของรูปภาพซึ่งจะแสดงในรูปแบบของ code และ ค่าที่เกี่ยวข้องกับ code นั้นๆ ซึ่งได้จัดแบ่ง code ออกเป็นกลุ่มต่างๆอย่างเป็นระบบในการแสดงข้อมูลของรูปที่สร้างขึ้น ดังนี้

HEADER section. ประกอบด้วยข้อมูลทั่วไปของรูปภาพ ฐานข้อมูลของ AutoCAD รุ่นที่ใช้ และ ระบบของหมายเลขตัวแปลที่ใช้ ซึ่งแต่ละตัวแปรจะประกอบไปด้วยชื่อของตัวแปลและค่าที่เกี่ยวข้อง เช่น $ACADMAINTVER จะมีข้อมูลของหมายเลขรุ่น

CLASSES section. ประกอบด้วยข้อมูลของคำจัดกัดความต่างๆแต่ละส่วนที่จะปรากฏใน BLOCKS, ENTITIES, and OBJECTS sections ของฐานข้อมูล

TABLES section. ประกอบด้วยการจำกัดความของสัญลักษณ์ต่างๆ ดังนี้

APPID (application identification table)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BLOCK_RECORD (block reference table)

DIMSTYLE (dimension style table)

LAYER (layer table)

LTYPE (linetype table)

STYLE (text style table)

UCS (User Coordinate System table)

VIEW (view table)

VPORT (viewpoint configuration table)

BLOCK section. ประกอบด้วยคำจำกัดความของบล็อกต่างๆ และคำอ้างอิงเกี่ยวกับการวาด เช่น 10 หมายถึง Base point DXF: X value; APP: 3D point

OBJECT section. ประกอบด้วยข้อมูลอื่นๆที่ไม่เกี่ยวข้องกับรูปวาดที่ออกแบบ

THUMBNILIMAGE section. ประกอบด้วยโครงร่างของข้อมูลด้านรูปภาพ

ENTITIES section. ข้อมูลรายละเอียดการวาดต่างๆซึ่งจะมีรายละเอียดของข้อมูลที่จำเป็นในการสร้างโปรแกรมแปลงไฟล์ รวมถึงคำอ่างอิงของ block

ตารางที่ 2.1 ข้อมูลที่จำเป็นในการสร้างโปรแกรมแปลงไฟล์จากรูปวงกลม

Circle group codes	**Group codes Description**
100	Subclass marker(AcDbCircle)
39	Thickness(optional; default = 0)
10	Center point (in OCS) DXF:X value; APP : 3D point
20, 30	DXF: Y and Z values of center point (in OCS)
40	Radius
210	Extrusion direction (optional; default = 0, 0, 1) DXF: X value APP: 3D vector
220, 230	DXF: Y and Z values of extrusion direction (optionai)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 ข้อมูลที่จำเป็นในการสร้างโปรแกรมแปลงไฟล์จากรูปส่วนหนึ่งของวงกลม

Arc group codes	**Group codes Description**
100	Subclass marker(AcDbCircle)
39	Thickness (optional; default = 0)
10	Center point (in OCS) DXF: X value; APP: 3D point
20, 30	DXF: Y and Z values of center point (in OCS)
40	Radius
100	Subclass marker(AcDbArc)
50	Start angle
51	End angle
210	Extrusion direction. (option; default = 0, 0, 1) DXF: X value; APP: 3D vector
220, 230	DXF: Y and Z values of extrusion direction (optional)

ตารางที่ 2.3 ข้อมูลที่จำเป็นในการสร้างโปรแกรมแปลงไฟล์จากรูปเส้นตรง

Line group codes	**Group codes Description**
100	Subclass marker(AcDbLine)
39	Thickness(optional; default = 0)
10	Start point (in WCS) DXF: X value; APP: 3D point
20, 30	DXF: Y and Z values of start point (in WCS)
11	End point (in WCS) DXF: X value; APP: 3D point
21, 31	DXF: Y and Z values of end point (in WCS)
210	Extrusion direction (option; default = 0, 0, 1)
220, 230	DXF: Y and Z values of extrusion direction (optional)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 ความรู้เบื้องต้นเกี่ยวกับ G-CODE

G-CODE เป็นชื่อสำหรับเรียกภาษาของโปรแกรมที่ใช้ติดต่อกับเครื่องจักร NC และ CNC ต่างๆ ซึ่งได้รับการพัฒนาโดย EIA ในยุค 1960s และได้รับการปรับปรุงแก้ไขในปี 1980 ที่รู้จักกันในชื่อ RS274D เนื่องจากการพัฒนาของ G-CODE ถูกปรับปรุงและเปลี่ยนแปลงบางส่วนเพื่อใช้งานที่เหมาะกับเครื่องจักรแต่ละประเภท ดังนั้น จึงอาจจะมีความแตกต่างกันเล็กน้อยในรายละเอียดขึ้นอยู่กับเครื่องจักรที่ใช้ แต่ส่วนใหญ่จะรักษามาตรฐานเดียวกันไว้โดยในเริ่มแรกได้ใช้มาตรฐานตามแบบของ Fanuc ตัวอย่าง เช่น G-code จะเริ่มต้นด้วยอักษร G และจะบอกเครื่องจักรว่าให้ทำการอย่างไร ตัวอย่างเช่น ให้เครื่องจักรเคลื่อนที่อย่างรวดเร็ว ให้เครื่องจักรทำการปรับอัตราการเคลื่อนที่ ให้เครื่องจักรทำการจัดระยะ offset ซึ่งสามารถดูจากตารางที่ ได้ ดังนี้

ตารางที่ 2.4 แสดงรูปแบบ G-CODE ตามแบบของ Fanuc

Common Fanuc G-Code	
G00	Fast position
G01	Linear interpolation
G02	CW circular interpolation
G03	CCW circular interpolation
G10/G11	Data writing/Data write cancel
G17	X-Y plane section
G18	X-Z plane section
G19	Y-Z plane section
G31	Skip function (used for probes and tool length measurement systems)
G33	Constant pitch threading
G40	Tool radius compensation off
G41	Tool radius compensation left
G42	Tool radius compensation right
G90	Absolute programming
G91	Incremental programming
G94/G95	Inch per minute/Inch per revolution feed
G96/G97	Constant cutting speed (Constant surface speed)/Constant rotation speed (constant RPM)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจาก G-CODE ตามมาตรฐานของ Fanuc แล้วยังมีการใช้มาตรฐานของ BCL แต่ไม่ได้รับความนิยมอย่างแพร่หลายเท่ากับมาตรฐานของ Fanuc

G-CODE ยังสามารถเขียนเพิ่มขึ้นเองจากผู้ใช้โดยการกำหนดค่าต่างๆที่ต้องการในรูปแบบตัวอักษร G และตามด้วยค่านั้นๆ เท่าที่ต้องการ แต่ข้อมูลที่ได้รับจะมีความเที่ยงตรงน้อยกว่าการใช้โปรแกรมแปลงมาจากโปรแกรมออกแบบ เช่น AutoCad เป็นต้น ซึ่งโปรแกรมเหล่านี้ เรียกว่าโปรแกรม post-processors ตัวอย่างเช่น Alphacam, Artcam, Edgecam, KCam, Plasma cam ซึ่งส่วนมากจะถูกใช้กับเครื่องจักรเฉพาะรุ่นไป โปรแกรม post-processors เหล่านี้ยังสามารถกำหนดค่าที่ต้องการใช้งานหรือต้องการเปลี่ยนแปลงได้ตามลักษณะงานที่แตกต่างกันอีกด้วย ซึ่งโดยทั่วไปแล้ว G-CODE มักถูกแปลกมาจากระบบ CAD ซึ่งเป็นระบบที่ใช้ในการออกแบบอย่างแพร่หลาย

2.2 การเขียนโปรแกรมติดต่อกับพอร์ตขนานด้วย Visual Basic 6

ในการเขียนโปรแกรมควบคุมอุปกรณ์ภายนอกผ่านพอร์ตขนานนั้น Visual Basic 6 ไม่มีฟังก์ชั่นสำหรับติดต่อกับพอร์ตขนานโดยตรง แต่สามารถเขียนโปรแกรมให้ติดต่อกับพอร์ตได้โดยใช้ฟังก์ชั่นที่สำเร็จรูปของ API (application programming interface) โดยสามารถติดต่ออ่านข้อมูลและส่งข้อมูลไปยังพอร์ตที่ต้องการได้

2.2.1 การติดตั้งไฟล์ inpout32.dll

สำหรับการติดตั้งไฟล์ inpout32.dll นั้นให้ผู้ใช้ copy ไฟล์ inpout32.dll ลงในโฟลเดอร์ system ของระบบ windows ซึ่ง windows 95 , 98 ให้ลงไว้ที่โฟลเดอร์ C:\windows\system และ windows Me, 2000 ,XP ให้ลงไว้ที่โฟลเดอร์ C:\windows\system32 หรือ C:\winnt\system32

2.2.2 การประกาศฟังก์ชันเพื่อเรียกใช้ไฟล์ inpout32.dll ใน visual basic6

การเรียกใช้งานไฟล์ inpout32.dll จำเป็นต้องประกาศฟังก์ชันที่หน้าต่างของ code window เสียก่อน ดังนี้

การประกาศฟังก์ชันเพื่อเรียกใช้งานไฟล์ inpout32.dll สำหรับการรับข้อมูล

Private Declare Function Inp Lib “inpout32.dll” Alias “Inp32” (ByVal PortAddress As Integer, ByVal Value As Integer)

การประกาศฟังก์ชันเพื่อเรียกใช้งานไฟล์ inpout32.dll สำหรับการส่งข้อมูล

Private Declare Sub Out Lib “inpout32.dll” Alias “Out32” (ByVal PortAddress As Integer, ByVal Value As Integer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 การเขียนคำสั่งเพื่อรับ-ส่งข้อมูลใน Visual Basic 6

คำสั่งที่ใช้ในการรับข้อมูลมานั้นเราจะใช้คำสั่ง

Inp (พอร์ตแอดเดรส)

ย่อมาจาก Input from a Port เป็นคำสั่งที่ใช้รับข้อมูลเข้ามาจากพอร์ตที่กำหนดส่วนคำสั่งที่ใช้ในการส่งข้อมูลออกจากพอร์ตนั้นเราจะใช้คำสั่ง

Out พอร์ตแอดเดรส, ข้อมูลที่จะส่ง

เป็นการส่งข้อมูลซึ่งต้องมีค่าอยู่ระหว่าง 0-255 โดยส่งออกทางพอร์ตแอดเดรสที่กำหนด

2.3 พอร์ตขนาน (PARALLEL PORT)

พอร์ตขนานสามารถจ่ายข้อมูลสัญญาณเป็นสี่เหลี่ยมหรือ TTL โดยที่สัญญาณลอจิก "1" จะเท่ากับ 5 โวลต์ และลอจิก "0" จะเท่ากับ 0 โวลต์ แบ่งการทำงานหลักเป็น 3 ส่วน คือ

2.3.1 DATA PORT

DATA PORTจะมีอยู่ 8 ขาหรือ 8 pin (ตั้งแต่ขาที่2 - 9) บางครั้งมักจะถูกเรียกว่า DATA REGEISTER ซึ่ง Register ตัวนี้จะส่งค่าได้อย่างเดียวไม่สามารถรับค่าได้

ตารางที่ 2.5 แสดงรายละเอียดของ DATA PORT

Name	Read/write	Bit No.	Signal Name
Data port	Write	bit7	Data7(pin9)
Data port	Write	bit6	Data6(pin8)
Data port	Write	bit5	Data5(pin7)
Data port	Write	bit4	Data4(pin6)
Data port	Write	bit3	Data3(pin5)
Data port	Write	bit2	Data2(pin4)
Data port	Write	bit1	Data1(pin3)
Data port	Write	bit0	Data0(pin2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 STATUS PORT

Status Port เป็นพอร์ตที่อ่านข้อมูลอย่างเดียวไม่สามารถเขียนข้อมูลลงไปได้ พอร์ตนี้จะมีสัญญาณเข้าอยู่ 5 สัญญาณ และสัญญาณ IRQ กับสัญญาณสงวนไว้อีก 2 บิต โดยสัญญาณ Busy จะ Active Low

ตารางที่ 2.6 แสดงรายละเอียดของ STATUS PORT

Name	Read/write	Bit No.	Signal Name
Data port	Read	bit7	Busy
Data port	Read	bit6	nAck
Data port	Read	bit5	PaperEnd
Data port	Read	bit4	Select
Data port	Read	bit3	nError
Data port	Read	bit2	IRQ(not)
Data port	Read	bit1	Reserved
Data port	Read	bit0	Reserved

สำหรับลักษณะการทำงานของแต่ละบิตใน Status Port

- Bit 7 Busy เมื่อ Active หมายถึง พรินเตอร์จะไม่รับข้อมูล
- Bit 6 nAck (Active Low) เมื่อ Active หมายถึง พรินเตอร์พร้อมที่จะทำงาน
- Bit 5 PaperEnd เมื่อ Active หมายถึง พรินเตอร์ไม่มีกระดาษ
- Bit 4 Select เมื่อ Active หมายถึง เลือกพรินเตอร์
- Bit 3 nError เมื่อ Active หมายถึง พรินเตอร์เกิดข้อผิดพลาด (Active Low)
- Bit 2, Bit 1, Bit 0 ไม่ใช้

2.3.3 CONTROL PORT

Control Port เป็นพอร์ตที่ใช้ในการควบคุมพรินเตอร์ สัญญาณในกลุ่มนี้จะ Active Low ยกเว้นสัญญาณหรือ Initialize เท่านั้นที่ไม่ถูก Invert

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.7 แสดงรายละเอียดของ CONTROL PORT

Name	Read/write	Bit No.	Signal Name
Data port	Read	bit3	nEselect (pin17)
Data port	Read	bit2	nlnitialize(pin16)
Data port	Read	bit1	nAutofeed (pin14)
Data port	Read	bit0	nStrobe (pin1)

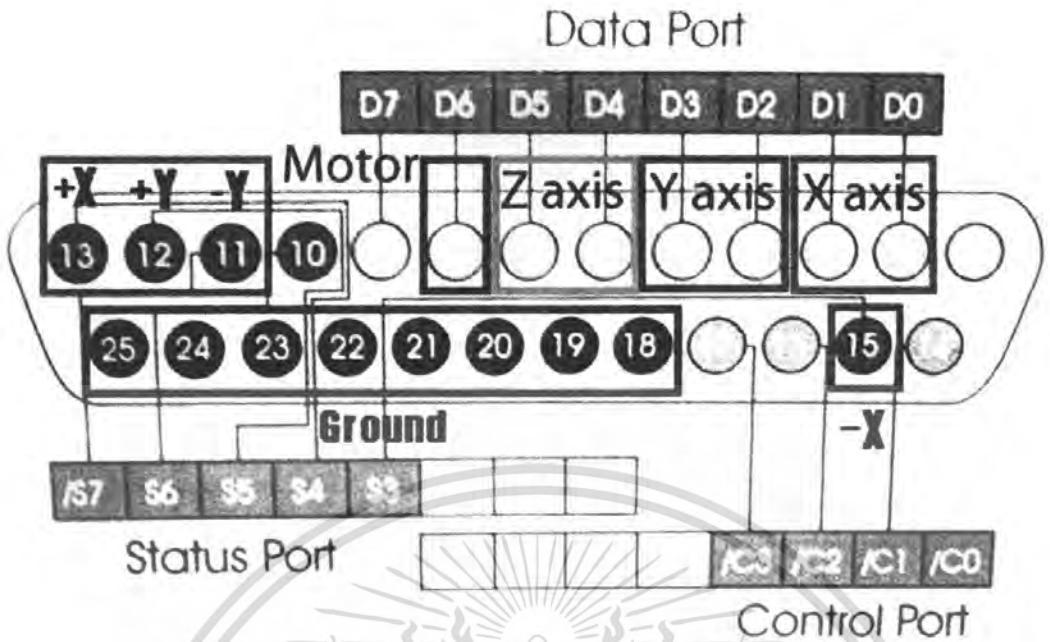
ลักษณะการทำงานของแต่ละบิตใน Control Port

- Bit 3 nESelect Printer เมื่อ Active หมายถึง เลือกพรินเตอร์
- Bit 2 nlnitialize เมื่อ Active หมายถึง รีเซตพรินเตอร์
- Bit 1 nAutoFeed เมื่อ Active หมายถึง พรินเตอร์กรทำ Line Feed
- Bit 0 nStrobe เมื่อ Active หมายถึง การบอกให้พรินเตอร์ทราบว่ามีข้อมูลเข้ามาแล้ว

จากข้อมูลข้างต้นเราสามารถใช้งานขาพอร์ตต่างๆ เพื่อติดต่อกับตัวตู้คอนโทรล ใช้พอร์ตส่วนของ Data port ในการควบคุมทิศทาง และการควบคุมตัวสเต็ปมอเตอร์ โดยเราใช้ 2 ขา ต่อการควบคุม 1 แกน 1 ขา ควบคุมทิศทาง และอีก 1 ขา ควบคุมการเคลื่อนที่ของความเร็ว โดยเราใช้ขา 2 และ 3 กับแกนX, ขา4 และ 5 กับแกนY, ขา 6 และ 7 กับแกน Z , ขา 8 สำหรับการควบคุมการเปิดปิด ของตัวดีซีมอเตอร์ ,ขา 15 สำหรับลิมิตสวิทช์ -X, ขา 13 สำหรับลิมิตสวิทช์ +X, ขา 12 สำหรับลิมิตสวิทช์ +Y, ขา 11 สำหรับลิมิตสวิทช์ -Y และขา 18-25 สำหรับกราวด์

การส่งค่าของสเต็ปมอเตอร์ 2 ขานั้นขาที่เป็นขาบอกทิศทาง จะส่งค่า เป็นลอจิก 1 กับ 0 สลับกันตามทิศที่ต้องการให้เคลื่อนที่ คือ ถ้าต้องการให้หมุนตามเข็ม จะต้องส่งค่าเป็น 1 ค้างไปยังที่ขาของตัวพอร์ตนี้ ถ้าต้องการให้หมุนทวนเข็มก็จะทำการส่งค่าเป็นลอจิก 0 ค้าง นอกจากการกำหนดทิศทางแล้วเราต้องควบคุมความเร็วของการเคลื่อนที่อีกด้วย โดยใช้อีก 1 ขาที่เหลือ ทำหน้าที่ส่งค่าสัญญาณพัลส์ หรือสัญญาณ ลอจิก 0 สลับกับ 1 ไปโดย ตัวตู้คอนโทรลจะตอบสนองกับลอจิก 0 หรือจะทำงานเมื่อ มีลอกจิก 0 เข้ามา (ต้องเป็นสัญญาณพัลส์) ดังนั้นหากต้องการให้มอเตอร์ของแกนนั้นๆหมุนตามเข็ม จะต้องทำการส่งค่าขาทิศทางเป็น 1 และควบคุมการส่งค่าสัญญาณพัลส์ไปยังอีกขาที่เหลือ ถ้าต้องการให้หมุนเร็วก็ทำการส่งค่าสัญญาณพัลส์ไปให้เร็วขึ้น ถ้าต้องการให้หมุนช้าก็ส่งค่าสัญญาณให้ช้าลง

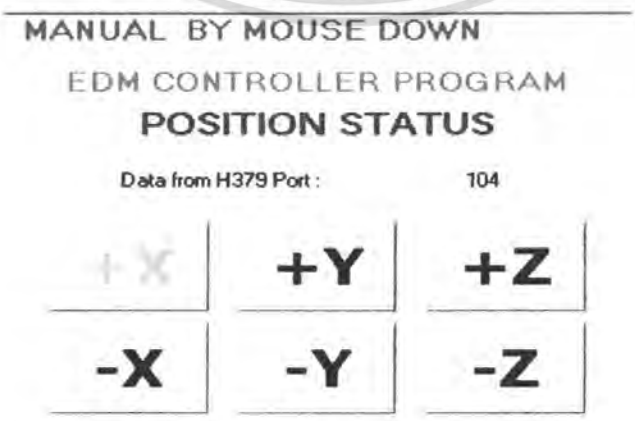
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 การใช้งานขาพอร์ตต่างๆ

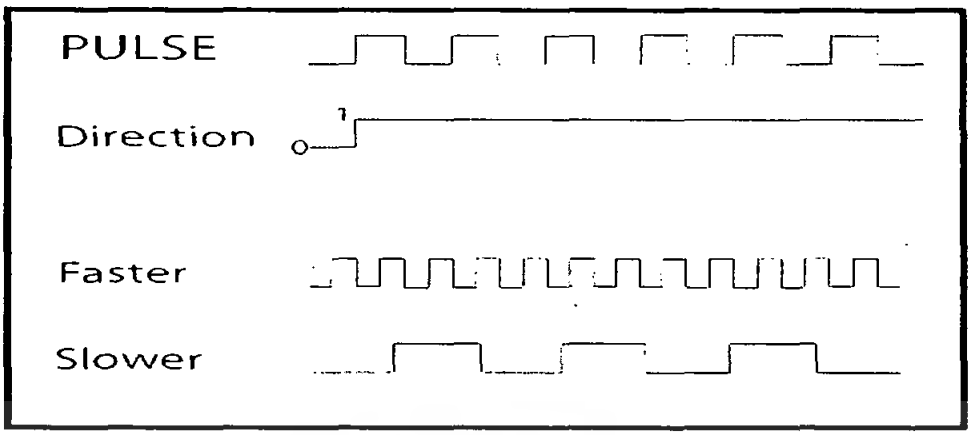
การส่งค่าไปยังวงจรที่ทำการตัดสว่านหรือดีซีมอเตอร์นั้นจะใช้พินขาที่ 8 ในการควบคุมการปิดและเปิดของรีเลย์ โดยถ้าส่งสัญญาณ "1" รีเลย์จะทำงานทำให้สว่านไม่หมุน แต่ถ้าขา 8 มีสัญญาณเป็น "0" รีเลย์จะไม่ทำงานทำให้สว่านทำงานตามปกติ

การส่งค่ากลับมายังโปรแกรม เพื่อให้โปรแกรมรู้ว่าลิมิตสวิทช์ถูกชนเพื่อไปตัดการทำงานของสว่าน และปุ่มบังคับที่ถูกชนทำได้โดยใช้ STATUS PORT ในการรับค่าจากพอร์ตขนานมายังโปรแกรม โดยใช้ ขา 15 ในตำแหน่ง -X , ขา 13 ในตำแหน่ง +X , ขา 12 ในตำแหน่ง+Y และขา 11 ในตำแหน่ง -Y และ ขากราวด์ พิน 18-25 เป็นขากราวด์ของพอร์ตขนาน



รูปที่ 2.2 การนำค่าที่ส่งกลับมาใช้ในการตัดการทำงานของปุ่มต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แสดงการบังคับแต่ละแกน

ตารางที่ 2.8 สรุปการนำแต่ละพินของพอร์ตขนานมาประยุกต์ใช้กับโปรเจกต์

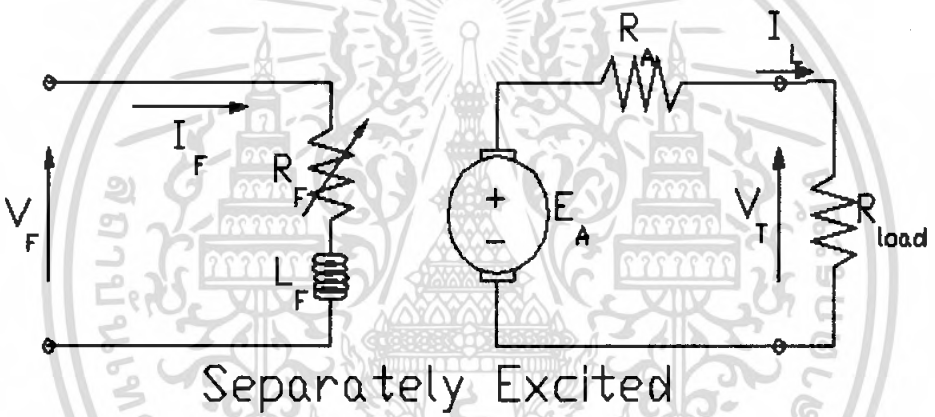
พิน	การประยุกต์ใช้
2	กำหนดความเร็วแกนX
3	กำหนดทิศทางแกนX
4	กำหนดความเร็วแกนY
5	กำหนดทิศทางแกนY
6	กำหนดความเร็วแกนZ
7	กำหนดทิศทางแกนZ
8	ตัดการทำงานของสว่าน(ดีซีมอเตอร์)
11	ขอบเขตการทำงาน-Y
12	ขอบเขตการทำงาน+Y
13	ขอบเขตการทำงาน+X
15	ขอบเขตการทำงาน-X
18	Ground
19	Ground
20	Ground
21	Ground
22	Ground
23	Ground
24	Ground
25	Ground

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

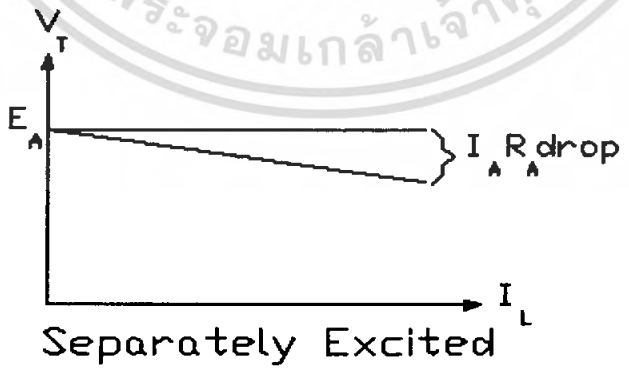
2.4 ความรู้เบื้องต้นสำหรับการพัฒนาเครื่องจักร

ในการปรับปรุงเครื่องกัดนั้นจำเป็นอย่างยิ่งที่จะต้องมีความรู้เกี่ยวกับวิธีการใช้งานมอเตอร์ซึ่งมอเตอร์ที่ใช้ในปัจจุบันมีด้วยกัน 4 แบบ คือ แบบกระตุ้นแยกอิสระ (separately exited) แบบขนาน (shunt) แบบอนุกรม(series) และแบบผสม (compound) ทั้งนี้การปรับปรุงเครื่องเจาะนี้นั้นได้นำมอเตอร์ชนิดกระตุ้นแยกอิสระ (Separately excited) มาใช้ในการทำการพัฒนาเครื่องจักรเดิม เพราะเป็นที่นิยมใช้กันอย่างแพร่หลาย และสามารถควบคุมได้ง่าย

2.4.1 มอเตอร์แบบกระตุ้นแยกอิสระ (Separately exited)

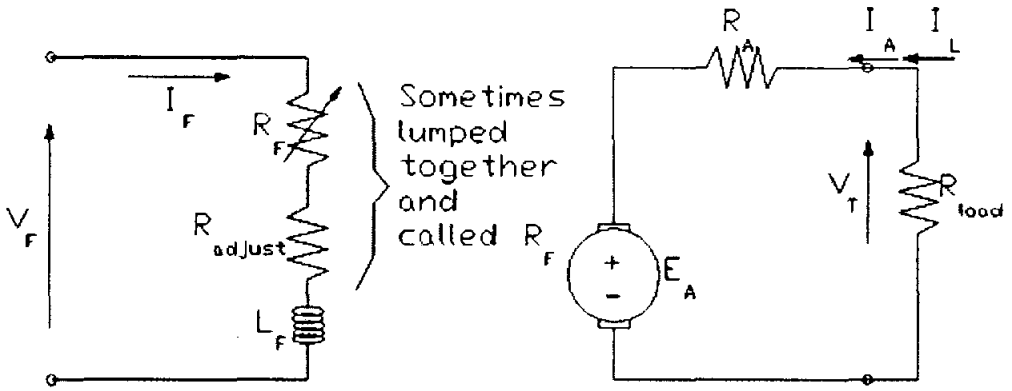


รูปที่ 2.4 แสดงวงจรสมมูลของมอเตอร์แบบกระตุ้นอิสระ(Separately exited)



รูปที่ 2.5 แสดงกราฟลักษณะสมบัติภายนอก(External Characteristics)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 แสดงการทำงานของมอเตอร์แบบกระตุ่นอิสระในสภาวะอยู่ตัวใดๆ

หลักการทำงานของมอเตอร์กระแสตรง พิจารณารูปที่ ซึ่งมีความสัมพันธ์ ดังนี้

$$Field\ \ Circuit: V_F = I_F R_F$$

$$Armature\ \ Circuit: V_T = E_A + I_A R_A$$

$$Mag.Curve: E_A = K \cdot \phi \cdot \omega \ ,\ I_F^{\bullet} = I_F - \frac{\Im_{AR}}{N_F} \ ,\ E_A - I_F^{\bullet} Curve$$

$$Mechanics: \tau_{ind} = K\phi I_A$$

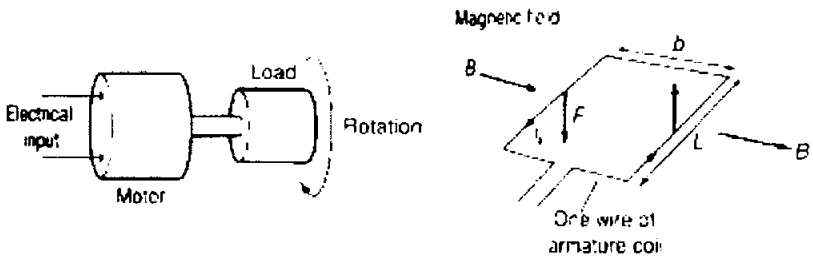
$$Steady\ \ State: \tau_{ind} = \tau_{Load} \ ,\ \omega = \frac{V_T}{K\phi} - \frac{R_A}{[K\phi]^2}\tau_{ind}$$

2.4.1.1 การปรับความเร็วของมอเตอร์กระแสตรง

มอเตอร์ไฟฟ้ากระแสตรง (Direct Current, D.C. Motor) เป็นอุปกรณ์ที่ใช้เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล หรือจะกล่าวว่าเป็นระบบที่มีสัญญาณไฟฟ้าเป็นอินพุท และมีเอาท์พุทเป็นพลังงานกลก็ได้ โดยทั่วไปมอเตอร์จะประกอบด้วยขดลวดที่ส่วนหมุน หรือ armature coil ซึ่งสามารถที่จะหมุนไปได้อย่างอิสระ ขดลวดนี้จะวางอยู่ในสนามแม่เหล็ก ซึ่งอาจจะเป็นแม่เหล็กถาวร หรือส่วนมากจะเป็นแม่เหล็กไฟฟ้าที่สร้างจากกระแสไฟฟ้าตัดผ่าน field coils. และเมื่อมีกระแสไฟฟ้า i_a ไหลผ่าน armature coil ซึ่งวางอยู่ในสนามแม่เหล็ก ก็จะทำให้เกิดแรงผลักทำให้ armature นี้เกิดการหมุน ตามที่แสดงในรูปที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 2.7 แสดงโครงสร้าง DC Motor

ดังนั้น เราสามารถเขียนแบบจำลองทางคณิตศาสตร์ ของมอเตอร์ไฟฟ้ากระแสตรง ได้ 2 แบบ ขึ้นอยู่กับตัวแปรที่ควบคุม คือ

- Armature – Controlled Motor
- Field – Control Motor

แรงกระทำ F ที่กระทำบนขดลวดความยาว L ที่มีกระแส i_a ไหลผ่าน และวางอยู่ในสนามแม่เหล็กที่มีความเข้มสนามแม่เหล็ก B แรง F ที่กระทำเป็นมุมฉากกับ B จะมีค่าเป็น

$$F = Bi_a L \qquad 2.4.1$$

ถ้ามีขดลวดจำนวน N เส้นจะได้

$$F = NBi_a L \qquad 2.4.2$$

แรงกระทำ F นี้จะยังผลให้เกิดแรงบิด T เมื่อ $T = Fb$ โดย b คือความกว้างของขดลวด ดังนั้น

$$T = NBi_a Lb \qquad 2.4.3$$

เราจะเห็นว่า แรงบิด T นี้จะเป็นสัดส่วนกับ Bi_a ส่วนพารามิเตอร์อื่น ๆ จะคงที่สำหรับมอเตอร์หนึ่ง ๆ ดังนั้นเราสามารถเขียนสมการได้เป็น

$$T = k_1 Bi_a \qquad 2.4.4$$

โดยที่

$k_1 = NLb$

72140
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ปรับค่า V_T (Armature Voltage Control)

$$\omega = \frac{V_T}{K\phi} - \frac{T_{ind} \cdot R_A}{[K\phi]^2}$$

ตัวแปรที่ใช้

$V_T = V_a$,Armature Voltage หน่วยเป็น โวลต์

$E_a = V_b$ (back emf) หน่วยเป็น โวลต์

เนื่องจาก armature เป็นขดลวดที่หมุนอยู่ในสนามแม่เหล็ก จึงเกิดการการเหนี่ยวนำและทำให้เกิดการเหนี่ยวนำไฟฟ้าขึ้น ทิศทางของแรงเคลื่อนไฟฟ้าที่เกิดขึ้นนี้ จะตรงกันข้ามกับทิศทางของความต่างศักย์ที่ทำให้เกิดการหมุน ดังนั้นเราจึงเรียกแรงเคลื่อนไฟฟ้าที่เกิดจากการเหนี่ยวนำนี้ว่า back emf โดย back emf นี้ จะเป็นอัตราส่วนกับอัตราการหมุนของ armature และสนามแม่เหล็กที่เกิดขึ้นจากขดลวด ทำให้ความเข้มสนามแม่เหล็ก B มีค่าเป็น

$$V_b = k_2 B\omega \qquad 2.4.5$$

เมื่อ ω คือความเร็วรอบของเพลาและ k_2 เป็นค่าคงที่

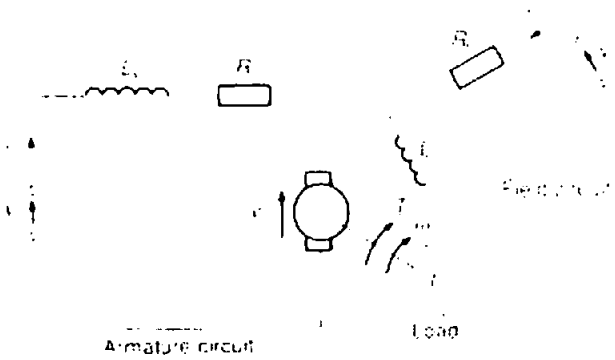
Armature – Controlled Motor เป็นมอเตอร์ที่กำหนดให้กระแสไฟฟ้าให้ Field coil นั้นมีค่าคงที่และความเร็วรอบของมอเตอร์จะปรับโดยปรับเปลี่ยนความต่างศักย์ v_a ที่ให้กับ armature การที่กระแสไฟใน field coil มีค่าคงที่ก็แสดงว่าความเข้มสนามแม่เหล็ก B มีค่าคงที่ ทำให้เราได้

$$V_b = k_3 \omega \qquad 2.4.6$$

เมื่อ k_3 คือค่าคงที่ ในวงจรของ armature นั้นจะประกอบด้วยความต้านทาน R_a และ inductance L_a ตามรูปที่ 2ซึ่งในกรณีนี้จะทำให้เกิด back emf v_b ขึ้น ดังนั้นถ้าหากว่าเราให้ ความต่างศักย์ v_a กับวงจร armature เราจะได้

$$v_a - v_b = La\frac{di_a}{dt} + R_a i_a \qquad 2.4.7$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 D.C. Motor Circuit

เมื่อเราพิจารณาสมการนี้ในรูปของ block diagram ดังที่แสดงในรูปที่ 2 เราจะเห็นว่าค่าที่เราให้กับระบบเป็น v_a แต่จะถูกลดลงด้วย back emf. v_b ซึ่งจะทำให้ได้สัญญาณความผิดพลาด $(v_a - v_b)$ สัญญาณความผิดพลาดที่ได้นี้จะเข้าสู่ขดลวด armature ดังนั้นสมการ (2.4.7) นี้จะอธิบายถึงความสัมพันธ์ระหว่างสัญญาณความผิดพลาด และค่ากระแสที่ได้ใน armature coil i_a แทนค่า v_b ลงในสมการ (2.4.7) จะได้

$$v_a - k_3\omega = L_a \frac{di_a}{dt} + R_a i_a \qquad 2.4.8$$

กระแส i_a ในขดลวด armature จะทำให้เกิดแรงบิด T ซึ่งมีค่าตามสมการ (2.4.4) เป็น

$$T = k_1 B i_a = k_4 i_a \qquad 2.4.9$$

เมื่อ k_4 เป็นค่าคงที่ จากนั้นแรงบิดจะกลายเป็นอินพุทให้กับระบบภาระกรรม ซึ่งในที่นี้คือเอาชนะความเฉื่อยของระบบเวลาและส่วนหมุนทั้งหมด ดังนั้นจากระบบเชิงกล

$$Net\ torque = T - damping\ torque \qquad 2.4.10$$

โดย *damping torque* มีค่าเท่ากับ $c\omega$ เป็นความหน่วง และ c เป็นค่าคงที่ ถ้าหากเราพิจารณาว่าเพลาไม่มีความยืดหยุ่นเชิงมุม หรือเพลาเป็นวัตถุเกร็งเราจะได้

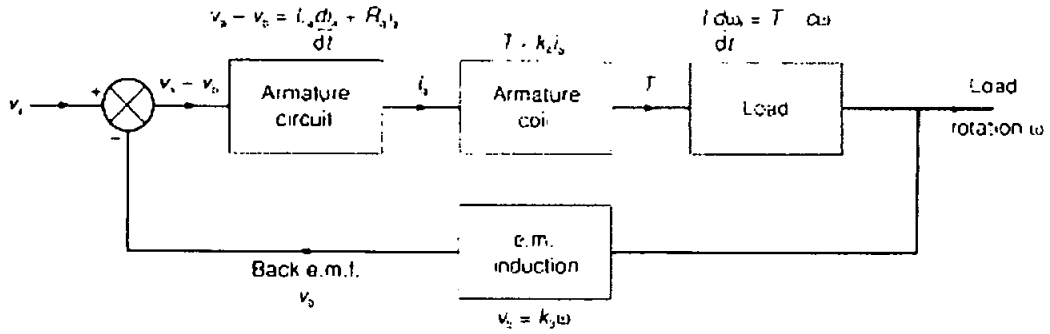
$$Net\ torque = k_4 i_a - c\omega \qquad 2.4.11$$

ซึ่งจากกฎข้อที่ 2 เราจะได้

$$I\frac{d\omega}{dt} = k_4 i_a - c\omega \qquad 2.4.12$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการ 2.4.8 และ 2.4.12 นี้จะอธิบายการทำงานของ armature – controlled motor



รูปที่ 2.9 แผนภาพบล็อกของ armature – controlled motor

- ปรับค่า I_F (Field Current Control)

$$\omega = \frac{V_T}{K_F \cdot I_F} - \frac{T_{ind} \cdot R_A}{\left[K_F \cdot I_F\right]^2}$$

การควบคุมมอเตอร์แบบ field-controlled motor จะทำโดยกำหนดให้กระแสที่ไหลเข้าสู่ armature มีค่าคงที่ แล้วปรับภาระกรรม โดยการเปลี่ยนแปลงค่าความต่างศักย์ของ field circuit ซึ่งวงจร field circuit นี้จะประกอบด้วย inductance ต่ออนุกรมกับ Resistant, R_f ดังนั้นในวงจรนี้

$$v_f = R_f i_f + L_f \frac{di_f}{dt} \qquad 2.4.13$$

เราสามารถพิจารณาการทำงานของ field – controlled motor ในรูปแบบของ block diagram ดังที่แสดงในรูปที่ 4 เราจะมีอินพุทเข้าระบบใน v_f จากนั้นจะทำให้เกิดกระแส i_f ขึ้นใน field – circuit ความสัมพันธ์ของกระแสและความต่างศักย์เป็นไปตามสมการข้างบนนี้ กระแสที่เกิดขึ้นจะทำให้เกิดความเข้มสนามแม่เหล็กและเกิดเป็นแรงบิดที่เหนี่ยวนำ armature แต่เนื่องจากความเข้มสนามแม่เหล็ก B นี้เป็นสัดส่วนกับกระแส i_f ในขณะที่ i_a มีค่าคงที่ ดังนั้นสมการ (2.69) สามารถเขียนได้เป็น

$$T = k_1 B i_a = k_c i_f \qquad 2.4.14$$

เมื่อ k_c เป็นค่าคงที่ จากนั้นแรงบิดจะเปลี่ยนเป็นการเอาชนะภาระกรรมคือระบบเพลาให้เกิดการหมุนด้วยความเร็วเชิงมุม ω ซึ่งจากระบบเชิงกลเราได้

$$Net\ torque = T - damping\ torque \qquad 2.4.15$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

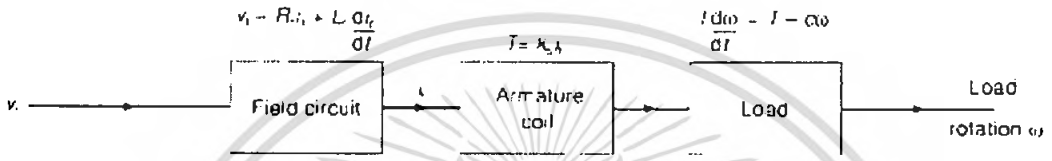
โดย damping torque มีค่าเท่ากับ เมื่อ เป็นค่าคงที่ หากเราไม่พิจารณาความยืดหยุ่นของเพลาเราจะได้

$$Net\ torque = k_{\varsigma} i_f - c\omega \qquad 2.4.16$$

และจากกฎของนิวตันเราจะได้

$$I\frac{d\omega}{dt} = k_{\varsigma} i_f - c\omega \qquad 2.4.17$$

สมการ (2.4.13) และ (2.4.17) จะอธิบายสภาพการทำงานที่เกิดขึ้นกับ field-controlled motor



รูปที่ 2.10 แผนภาพบล็อก ของ Field – controlled motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

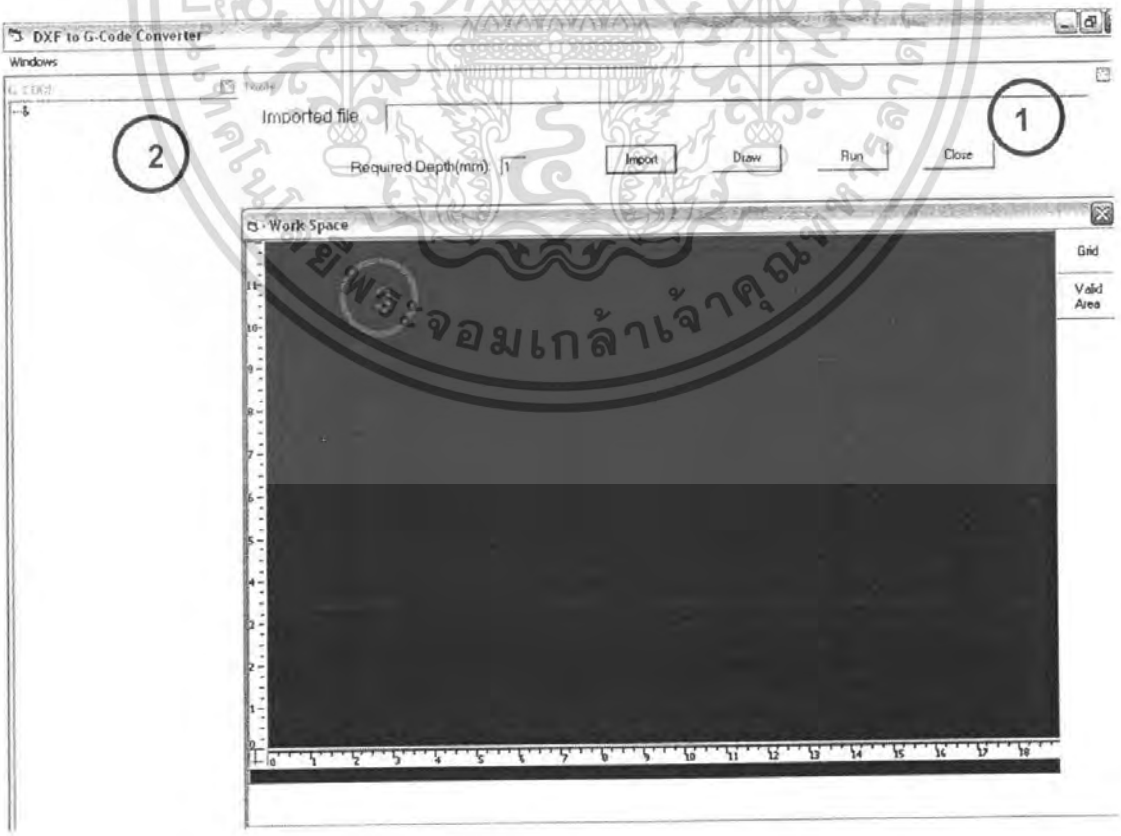
หลักการสร้างและออกแบบ

ในส่วนของการสร้างและออกแบบเครื่องกัดจากการใช้ฝีมือช่างฝีมือมาเป็นระบบอัตโนมัตินั้น ต้องทำงานในสามส่วนต่อไปนี้ คือ ส่วนโปรแกรมแปลงไฟล์ โปรแกรมส่วนติดต่อเครื่องกัด และส่วนการปรับปรุงเครื่องกัด ต้องรู้ว่าแต่ละส่วนมีความสัมพันธ์กันอย่างไรรวมไปถึงขอบเขตความต้องการของการนำเครื่องกัดไปใช้

3.1 ส่วนโปรแกรม

ส่วนโปรแกรมเป็นส่วนที่ใช้สั่งงานเครื่องกัดมีความสามารถทั้งการนำเอาไฟล์รูปภาพจากโปรแกรมออกแบบต่างๆที่บันทึกในรูปแบบไฟล์ DXF มาแปลงเป็นชนิดไฟล์ G-CODE (.nc) แล้วนำไปสั่งให้เครื่องกัดทำงานตามรูปที่ต้องการได้ สามารถแสดงรูปภาพจากข้อมูลไฟล์ DXF ที่นำมาทำการแปลงเป็น G-CODE ทั้งยังมีส่วนให้ผู้ใช้สามารถทำการควบคุมเครื่องกัดได้โดยตรงอีกด้วย

3.1.1 โปรแกรมส่วนการแปลงไฟล์ (DXF to G-Code Converter)

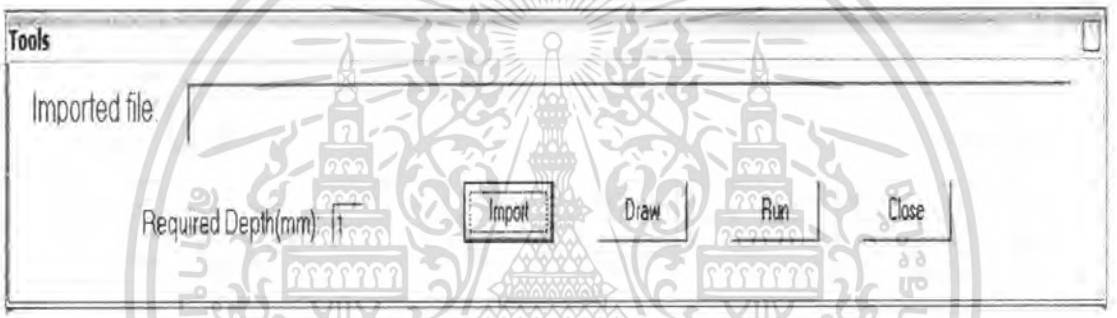


รูปที่ 3.1 แสดงหน้าต่างของโปรแกรมส่วนการแปลงไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DXF to G-Code Converter เป็นหน้าต่างที่ใช้สำหรับนำไฟล์ DXF จากโปรแกรมออกแบบต่างๆในที่นี้คือโปรแกรม AutoCAD เข้ามาเพื่อแปลงเป็นไฟล์ G-CODE (.nc) ซึ่งประกอบไปด้วยสามส่วนคือ ส่วน Tools, ส่วน G-CODE และส่วน Work Space โดยที่แต่ละส่วนจะทำงานสัมพันธ์กัน ยกตัวอย่างเช่น เมื่อมีการเรียกข้อมูลเข้ามาทางหน้าต่างส่วน Tools รายละเอียดของรูปภาพนั้นจะถูกอ่านแล้วทำการแปลงส่งไปแสดงที่ส่วนของ G-CODE และเมื่อส่วนหน้าต่าง G-CODE มีการเปลี่ยนแปลงแล้วทำการกดปุ่ม Draw ที่หน้าต่างของ Tools ก็จะมีผลทำให้หน้าต่างส่วนของหน้าต่าง Work Space มีการแสดงรูปภาพที่แตกต่างจากรูปภาพเดิม เป็นต้น ซึ่งแต่ละส่วนนั้นมีรายละเอียดดังนี้

3.1.1.1 Tools Window (1)



รูปที่ 3.2 แสดงหน้าต่าง Tools

เป็นส่วนที่ใช้ในการควบคุมการทำงานของหน้าต่างอื่นๆประกอบไปด้วย

- Imported file เป็นส่วนที่ใช้แสดงชื่อของไฟล์ DXF ที่ต้องการนำมาแปลงไฟล์
- Required Depth (mm) เป็นส่วนที่ให้ผู้ใช้กำหนดความลึกของการเจาะ โดยจะมีค่าตั้งต้นที่ Imm
- Import เป็นปุ่มที่ใช้สำหรับเรียกไฟล์ DXF ที่ต้องการ โดย แสดง ดังรูปที่ 3.3
- Draw เป็นปุ่มที่ใช้ในการแสดงรูปของ G-CODE ที่มีการเปลี่ยนแปลง
- Run เป็นปุ่มที่สั่งการให้เครื่องกัดทำการกัดตามข้อมูลที่ได้ทำการเลือกข้างต้น
- Close เป็นปุ่มปิดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.3 แสดงหน้าต่างการนำไฟล์ DXF เข้ามาแปลงเป็นไฟล์ NC

3.1.1.2 G-CODE Window (2)

รูปที่ 3.4 แสดงหน้าต่าง G-CDOE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นหน้าต่างสำหรับแสดงไฟล์ G-CODE ที่ได้ทำการแปลงไฟล์มาแล้ว ซึ่งการแปลงไฟล์ DXF มาเป็นไฟล์ G-CODE นั้น เราต้องทราบรายละเอียดของรูปเส้นตรง และวงกลม ซึ่งใช้ข้อมูลจากไฟล์ DXF ดังที่กล่าวในบทที่ 2 จากนั้นเราทำการเขียนฟังก์ชันเพื่อทำการค้นหาข้อมูลที่ต้องการดังนี้

```
Function AcDbCondition()
  'Circle Loop----------------------------------------------------
    If DataVar = "AcDbCircle" Then
      checkAcDb = 1
      Line Input #iFileNum, DataVar2
        If DataVar2 = 10 Then
          Line Input #iFileNum, DataVar2
          Debug.Print "I get X(3) =" & Val(DataVar2)
          x(3) = Val(DataVar2)
        End If
        Line Input #iFileNum, DataVar2
        If DataVar2 = 20 Then
          Line Input #iFileNum, DataVar2
          Debug.Print "I get Y(3) =" & Val(DataVar2)
          Y(3) = Val(DataVar2)
        End If
        Line Input #iFileNum, DataVar2
        Line Input #iFileNum, DataVar2
        Line Input #iFileNum, DataVar2
        If DataVar2 = 40 Then
          Line Input #iFileNum, R
          Debug.Print "I get R =" & R
        End If
        Line Input #iFileNum, DataVar2
        Line Input #iFileNum, DataVar2
        If DataVar2 = "AcDbArc" Then
          Line Input #iFileNum, DataVar2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Line Input #iFileNum, P1
            Line Input #iFileNum, DataVar2
            Line Input #iFileNum, P2

            Debug.Print "I get P1 =" & P1
            Debug.Print "I get P2 =" & P2

            x(1) = Round(x(3) + R * Cos(P1 * 3.141 / 180))
            Y(1) = Round(Y(3) + R * Sin(P1 * 3.141 / 180))

            x(2) = Round(x(3) + R * Cos(P2 * 3.141 / 180))
            Y(2) = Round(Y(3) + R * Sin(P2 * 3.141 / 180))

            Debug.Print x(1) & " , " & Y(1)
            Debug.Print x(2) & " , " & Y(2)

        Else
            Y(2) = Val(R) + Val(Y(3))
            Debug.Print "I get Y(2) =" & Y(2)
              x(2) = x(3)
              x(1) = x(2)
              Y(1) = Y(2)
        End If
'X(1),Y(1)=start
'X(2),Y(2)=end
'X(3),Y(3)=center
'P1=start radius
'P2=end radius
'R=radius
        iFileNum2 = FreeFile
        If check = 1 Then
            Open txtExport For Output As #iFileNum2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ElseIf check > 1 Then
          Open txtExport For Append As #iFileNum2
        End If

        If check = 1 Then
          Print #iFileNum2, "G90"
          Print #iFileNum2, "G00 Z" & txtThick.Text + 2
        End If
        Print #iFileNum2, "G00 X" & x(1); " Y" & Y(1)
        Print #iFileNum2, "G01 Z" & txtThick.Text - txtDepth
        Print #iFileNum2, "G03 X" & x(3); " Y" & Y(3); " I" & x(2); " J" & Y(2)
        Print #iFileNum2, "G00 Z" & txtThick.Text + 2
        Close #iFileNum2
      End If
'End Circle Loop-----------------------------------------

'Line Loop-----------------------------------------
  ' Public X1, X2, Y1, Y2 As Integer
      If (DataVar = "AcDbLine") Or (Form1.KeawClick = 1) Then
      Debug.Print "In lOOp"
        checkAcDb = 2

          Line Input #iFileNum, DataVar2
            If DataVar2 = 10 Then
            Line Input #iFileNum, X1
            Debug.Print "I get X1 =" & Val(X1)
            X1 = Val(X1)
          End If
          Line Input #iFileNum, DataVar2
          If DataVar2 = 20 Then
            Line Input #iFileNum, Y1
            Debug.Print "I get Y1 =" & Val(Y1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Y1 = Val(Y1)
        End If
        Line Input #iFileNum, DataVar2
        Line Input #iFileNum, DataVar2
        Line Input #iFileNum, DataVar2
        If DataVar2 = 11 Then
            Line Input #iFileNum, X2
            Debug.Print "I get X2 =" & X2
            X2 = Val(X2)
        End If
        Line Input #iFileNum, DataVar2
        If DataVar2 = 21 Then
            Line Input #iFileNum, DataVar2
            Debug.Print "I get Y2 =" & DataVar2
            Y2 = Val(DataVar2)
        End If
        '**********
        TmPoint
'************ End Get value ***************
    iFileNum2 = FreeFile
    If check = 1 Then
        Open txtExport For Output As #iFileNum2
    ElseIf check > 1 Then
        Open txtExport For Append As #iFileNum2
    End If
    Print #iFileNum2, "G90"
    Print #iFileNum2, "G00 Z" & txtThick.Text + 2
    Print #iFileNum2, "G00 X" & X1; " Y" & Y1
    Print #iFileNum2, "G01 Z" & txtThick.Text - txtDepth
    Print #iFileNum2, "G01 X" & X2; " Y" & Y2
    Print #iFileNum2, "G00 Z" & txtThick.Text + 2
    Print #iFileNum2, "G00 X0 Y0"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

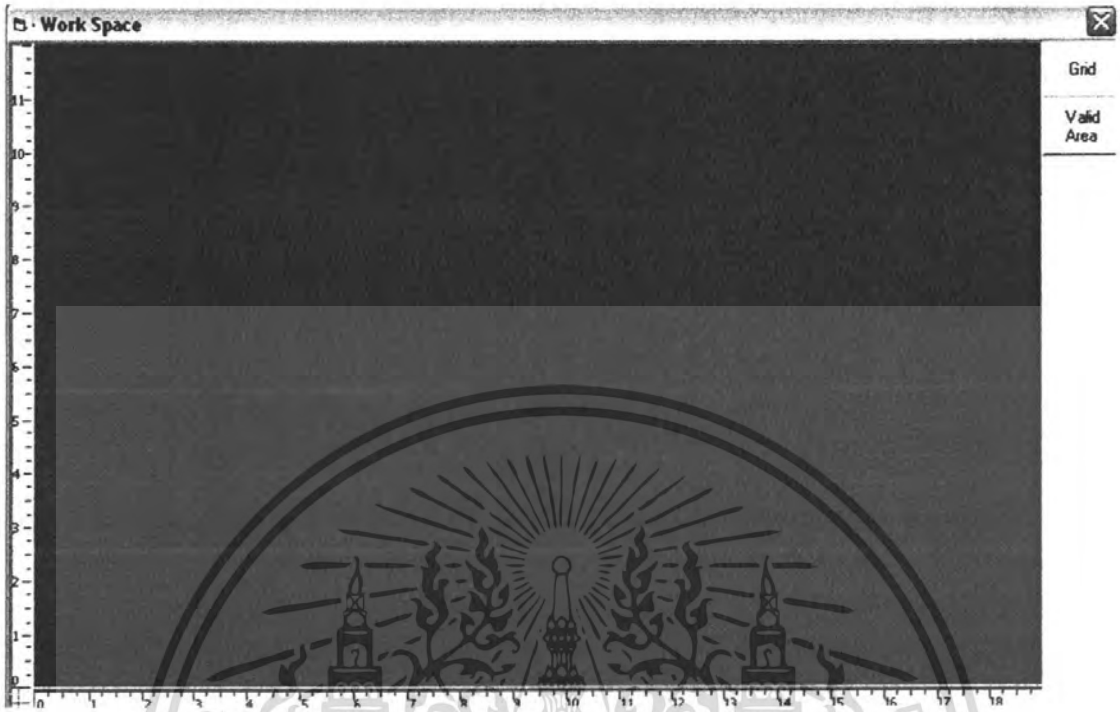
```
    Close #iFileNum2
  End If
'End Line Loop----------------------------------------------
End Function
```

ซึ่งหลังจากการแปลงไฟล์เรียบร้อยแล้วจะแสดงไฟล์ G-CODE ในหน้าต่าง G-CODE พร้อมทั้งแสดงรูปจากไฟล์ DXF ในหน้าต่าง Work Space ดังรูป

รูปที่ 3.5 แสดงไฟล์ G-code ที่ได้จากการแปลงไฟล์ DXF พร้อมทั้งรูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.3 Work Space Window (3)



รูปที่ 3.6 แสดงหน้าต่าง Work Space

เป็นหน้าต่างที่ใช้แสดงรูปภาพที่นำมาจากไฟล์ DXF ซึ่งออกแบบจากผู้ใช้ โดยจะทำการแสดงรูปใหม่ทุกครั้งเมื่อมีการแก้ไขข้อมูลในหน้าต่างของ G-CODE หลังจากกดปุ่ม Draw สามารถเขียนโปรแกรม ดังนี้

```
Sub showPic()
   ii = 0
ii = ii + 1
seq = 1
o2 = myArray(ii)
Do While Not o2 = "---&"
o2 = myArray(ii)
If Mid(o2, 1, 5) = "G00 Z" Then
   PicArrayX(seq) = "HELLO"
   PicArrayY(seq) = "HELLO"
ElseIf o2 = "G00 X0 Y0" And myArray(ii + 1) = "---&" Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับเอกสารใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Timer1.Enabled = True
    'DrawPic
ElseIf Mid(o2, 1, 5) = "G00 X" And Not myArray(ii + 1) = "---&" Then
        FindChar2 (o2)
        PicArrayX(seq) = Mid(o2, SPACE(1) + 2, SPACE(2) - SPACE(1) - 2)
        PicArrayY(seq) = Mid(o2, SPACE(2) + 2, SPACE(3) - SPACE(2) - 2)
ElseIf Mid(o2, 1, 5) = "G01 Z" Then
    seq = seq - 1
ElseIf Mid(o2, 1, 5) = "G01 X" Then
    FindChar2 (o2)
    PicArrayX(seq) = Mid(o2, SPACE(1) + 2, SPACE(2) - SPACE(1) - 2)
    PicArrayY(seq) = Mid(o2, SPACE(2) + 2, SPACE(3) - SPACE(2) - 2)

ElseIf Mid(o2, 1, 5) = "G03 X" Then
    FindChar2 (o2)
    'In case of a full circle
    If PicArrayX(seq - 1) = Mid(o2, SPACE(3) + 2, SPACE(4) - SPACE(3) - 2) And
PicArrayY(seq - 1) = Mid(o2, SPACE(4) + 2, SPACE(5) - SPACE(4) - 2) Then
        PicArrayX(seq - 1) = "HELLO"
        PicArrayY(seq - 1) = "HELLO"
        PicArrayX(seq) = "C" & Mid(o2, SPACE(1) + 2, SPACE(2) - SPACE(1) - 2)
        PicArrayY(seq) = Mid(o2, SPACE(2) + 2, SPACE(3) - SPACE(2) - 2)
        PicArrayR(seq) = Mid(o2, SPACE(3) + 2, SPACE(4) - SPACE(3) - 2) - Mid(o2, SPACE(1)
+ 2, SPACE(2) - SPACE(1) - 2)
        If myArray(ii + 1) = "G01 X" Then
            PicArrayX(seq) = Mid(myArray(ii + 1), SPACE(3) + 2, SPACE(4) - SPACE(3) - 2)
            PicArrayY(seq) = Mid(myArray(ii + 1), SPACE(4) + 2, SPACE(5) - SPACE(4) - 2)
        End If
    End If
End If
    seq = seq + 1
    ii = ii + 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

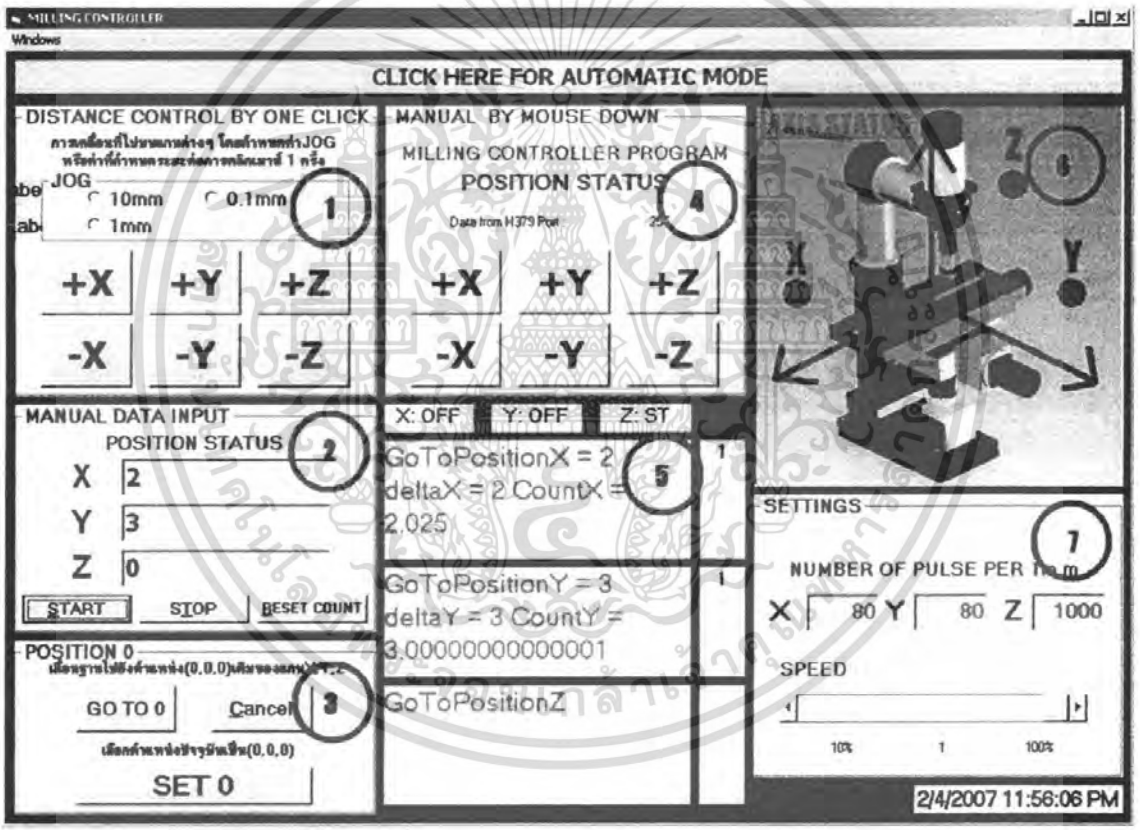
```
Loop
For B = 1 To seq
   Debug.Print PicArrayX(B)
   Debug.Print PicArrayY(B)
   Debug.Print PicArrayR(B)
Next B
   For iii = 2 To seq - 4
      If Not PicArrayY(iii) = "HELLO" Then
      If Val(PicArrayY(iii)) > 60 Then
         PicArrayY(iii) = Trim(Val(PicArrayY(iii)) - (2 * (Val(PicArrayY(iii)) - 60)))
      ElseIf Val(PicArrayY(iii)) < 60 Then
         PicArrayY(iii) = Trim(Val(PicArrayY(iii)) + (2 * (60 - Val(PicArrayY(iii)))))
      End If
      End If
   Next iii
End Sub

Sub DrawPic()
   For iii = 2 To seq - 4
      If Not Mid(PicArrayX(iii), 1, 1) = "C" Then
         If Not PicArrayX(iii + 1) = "HELLO" Then
         If Not PicArrayX(iii) = "HELLO" Then
         frmDrawing.WrkSpace.Line (Val(PicArrayX(iii)), Val(PicArrayY(iii)))-
(Val(PicArrayX(iii + 1)), Val(PicArrayY(iii + 1))), QBColor(14)
         End If
         End If
      Else
         'If Not PicArrayX(iii + 1) = "HELLO" Then
         'If Not PicArrayX(iii) = "HELLO" Then
         Debug.Print Val(Mid(PicArrayX(iii), 2, Len(PicArrayX(iii)))) & "----------
----------"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        frmDrawing.WrkSpace.Circle (Val(Mid(PicArrayX(iii), 2, Len(PicArrayX(iii)))),
Val(PicArrayY(iii))), Val(PicArrayR(iii)), QBColor(14)
        'End If
        'End If
      End If
   Next iii
End Sub
```

3.1.2 โปรแกรมส่วนการควบคุมเครื่องจักร (MANUAL FORM)



รูปที่ 3.7 แสดงหน้าต่างโปรแกรมควบคุมเครื่องจักร

MANUAL FORM เป็นหน้าต่างสำหรับการควบคุมการเคลื่อนที่ตามแกนต่างๆ ซึ่งกำหนดโดยผู้ใช้ ซึ่งประกอบไปด้วยส่วนต่างๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

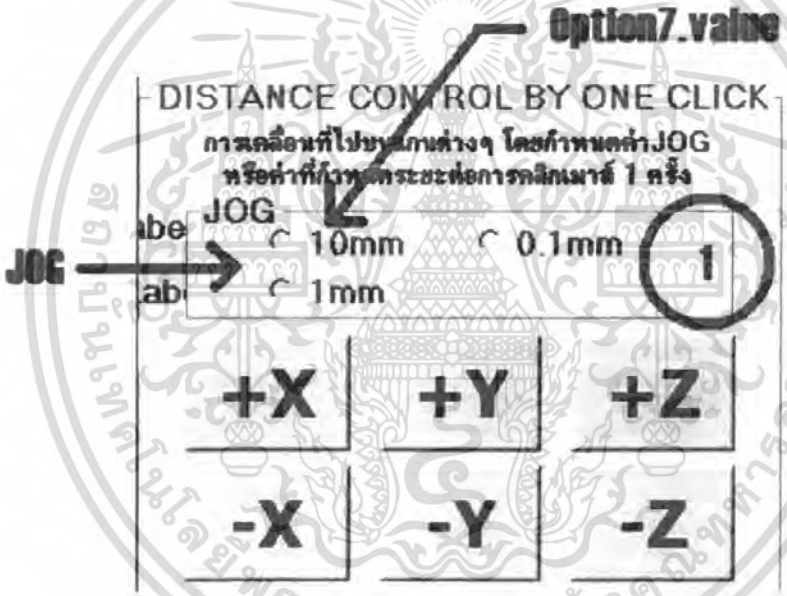
3.1.2.1 DISTANCE CONTROL BY ONE CLICK(1)

เป็นการเคลื่อนที่ไปบนแกนต่างๆ โดยการกำหนดค่าJOG หรือค่าที่กำหนดระยะต่อการคลิกเมาส์ 1 ครั้งซึ่งจะมีการนับค่าตำแหน่งที่เคลื่อนที่ไป และแสดงค่าตำแหน่งที่ POSITION STATUS หลักการเขียนโปรแกรมDISTANCE CONTROL BY ONE CLICK

ก่อนทำการเขียนโปรแกรมส่งพัลส์ผ่านพอร์ตขนาน จะต้องทำการประกาศก่อนโดยพิมพ์ข้อความดังนี้ไว้ที่ส่วนบนสุดของโปรแกรม

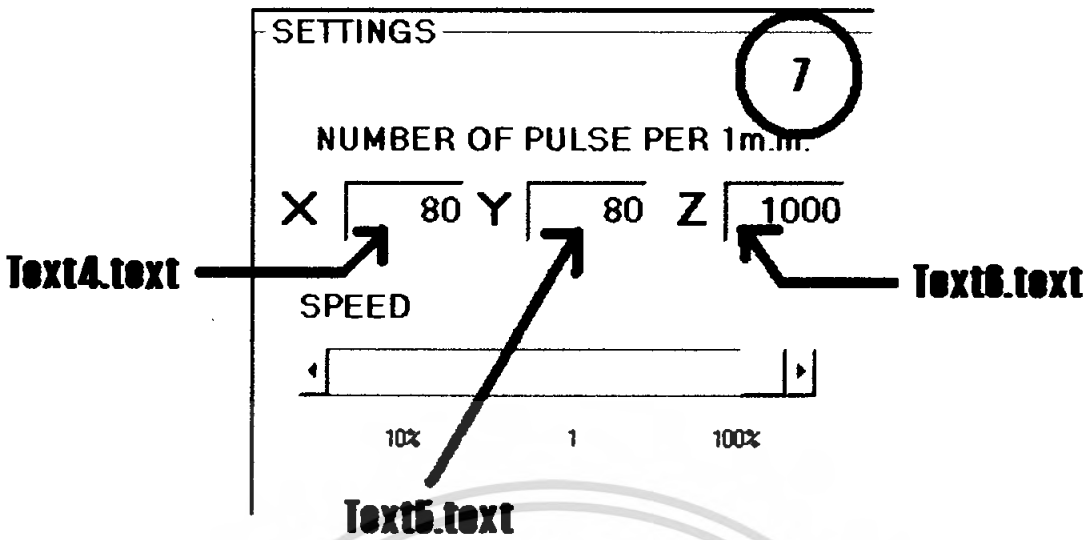
```
Private Declare Sub Out Lib "inpout32.dll" Alias "Out32" (ByVal PortAddress As Integer, ByVal
Value As Integer)
```

โปรแกรมจะนำค่าที่ถูกกำหนดไว้ในส่วน NUMBER OF PULSE PER 1 m.m.(7) มาใช้ในการส่งพัลส์ ซึ่งต่อไปนี้จะเป็นการยกตัวอย่างการเขียนโปรแกรมส่วนการส่งพัลส์บนแกน X



รูปที่ 3.8 ตำแหน่งของJOGและOption7.value

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 แสดง ตำแหน่งของTextต่างๆ

กำหนดให้ Option7.Value คือ ตัวเลือก 10 m.m.

Option8.Value คือ ตัวเลือก 1 m.m.

Option9.Value คือ ตัวเลือก 0.1 m.m.

Text4.Text คือ ค่า NUMBER OF PULSE PER 1 m.m. ของแกนX

Text5.Text คือ ค่า NUMBER OF PULSE PER 1 m.m. ของแกนY

Text6.Text คือ ค่า NUMBER OF PULSE PER 1 m.m. ของแกนZ

- โปรแกรมในส่วนไทเมอร์ของJOG

```
If Option7.Value = True And IsNumeric(Text4.Text) Then
X = Text4.Text * 10
ElseIf Option8.Value = True And IsNumeric(Text4.Text) Then
X = Text4.Text
ElseIf Option9.Value = True And IsNumeric(Text4.Text) Then
X = Text4.Text / 10 End If
```

- โปรแกรมในส่วนของปุ่ม+X

```
A = 0
If Option7.Value = False And Option8.Value = False And Option9.Value = False Then 'check list
Timer17.Enabled = False
MsgBox "You must choose JOG.", vbOKOnly
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ElseIf Option7.Value = True Or Option8.Value = True Or Option9.Value = True Then
Timer17.Interval = HScroll1.Value
Timer17.Enabled = True
End If
X = X + A
```

- โปรแกรมในส่วนไทเมอร์ของ+X

```
  Label10.Caption = "MACHINE ON WORK"
  Shape1.BackColor = vbGreen
  If SelfTimer Then ' Check SelfTimer
    Out pwrite, &H3
    Label19.Caption = "1"
    Label13.Caption = "1"
    Label20.Caption = "GoToPositionX = " & (GoToPositionX) & " deltaX = " & deltaX & "
CountX = " & CountX
    CountX = CountX + (1 / Val(Text4.Text)) ' 1/x is number of pulse per 1mm
    GoToPositionX = GoToPositionX + (1 / Val(Text4.Text))
    A = A + 1 'count
    Label29.Caption = "WORK"
  Else
    Out pwrite, &H2
    Label19.Caption = "0"
    Label13.Caption = "0"
  End If
SelfTimer = Not SelfTimer ' toggle SelfTimer
 Label18.Caption = A
  If Not (X <> A) Then 'X=A
    Timer17.Enabled = False
     Out pwrite, &H0
  Label29.Caption = "> Stop <"
  Label20.Caption = "GoToPositionX = " & (GoToPositionX) & " deltaX = " & deltaX & "
CountX = " & CountX
  Label10.Caption = "X: OFF"
```

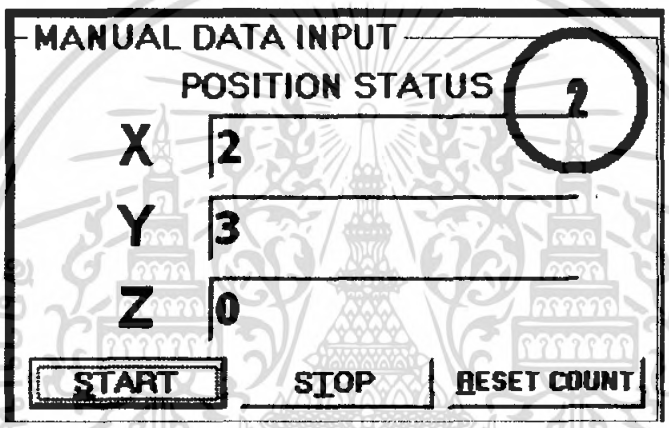
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Shape1.BackColor = vbRed
End If
```

ถ้ามีการใส่ค่าที่ตำแหน่ง Text4.Text แล้วคลิกที่Option7.value ก็จะเป็นกรณีตรงกับเงื่อนไขแรก (If Option7.Value = True And IsNumeric(Text4.Text) Then X = Text4.Text * 10) แล้วคลิกปุ่ม +X โปรแกรมก็จะทำการส่งพัลส์ไปเท่ากับค่าตัวแปร X และมีความเร็วในการส่งพัลส์เท่ากับ HScroll1.Value

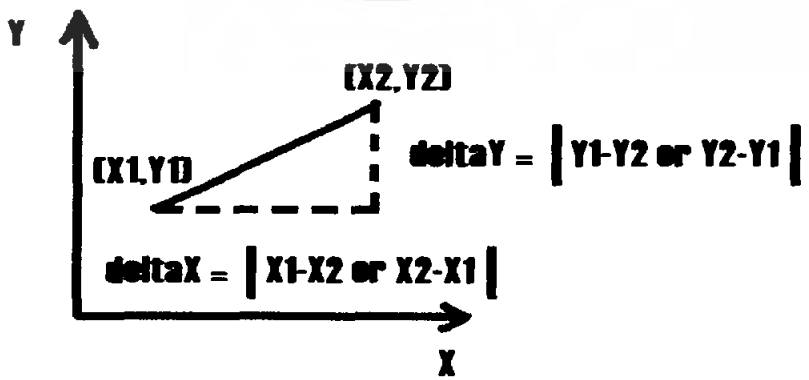
3.1.2.2 MANUAL DATA INPUT

เป็นส่วนของการเคลื่อนที่ไปบนแกนต่างๆโดยการกำหนดค่าตำแหน่งที่ผู้ใช้ต้องการ โดยมีการนับค่าตำแหน่งแสดงที่POSITION STATUS



รูปที่ 3.10 แสดงMANUAL DATA INPUT

เนื่องจากเครื่องกัดเป็นเครื่องกัดแกน X, Y, Z แต่แกน Z นั้นเป็นเพียงความลึกของชิ้นงาน ดังนั้นการเขียนโปรแกรมโดยใช้สมการเส้นตรงจึงใช้เพียงแค่แกน X และ Y กำหนดให้ค่าตัวแปรต่างๆเป็นไปตามรูปด้านล่าง



รูปที่ 3.11 แสดงค่าตัวแปรต่างๆในสมการเส้นตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจุด X1, Y1 และ X2, Y2 ห่างกันเป็นระยะ deltaX และ deltaY ตามแกน X และ Y ดังนั้นการเขียนโปรแกรมส่งพัลส์ จึงเขียนโดยใช้ค่า deltaX และ deltaY มาส่งพัลส์ในแกน X และY ตามลำดับก่อนและหลัง แต่ในกรณีที่ค่า deltaX และ deltaY มีค่าเป็นตัวเลข 2 หลัก การส่งพัลส์อาจจะมีความละเอียดไม่พอ ดังนั้นจึงต้องใช้ฟังก์ชัน Mid และ Cint มาช่วยในการทำตัวเลขให้เป็นหลักเดียวและปัดเศษตามลำดับ ซึ่งจะทำให้การกัดชิ้นงานมีความละเอียดมากขึ้น

- โปรแกรมส่วนของปุ่มSTART

```
fixX = CountX
fixY = CountY
fixZ = CountZ
If Text1.Text = "" Then
Text1.Text = "0"
Timer9.Enabled = False
Timer13.Enabled = False
Timer10.Enabled = False
Timer14.Enabled = False
Timer11.Enabled = False
Timer15.Enabled = False
ElseIf IsNumeric(Text1.Text) Then
SelfTimer = False
deltaX = Text1.Text - GoToPositionX
GoToPositionX = Text1.Text
If deltaX = 0 Then
Timer9.Enabled = False
Timer13.Enabled = False
ElseIf deltaX > 0 Then
checkX = 0
Timer9.Interval = HScroll1.Value
Timer9.Enabled = True
Timer13.Enabled = False
ElseIf deltaX < 0 Then
checkX = 0
Timer13.Interval = HScroll1.Value
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Timer13.Enabled = True
        Timer9.Enabled = False
    End If
Else
MsgBox "Please insert only numeric number in X box.", vbOKOnly
Timer9.Enabled = False
Timer13.Enabled = False
Timer10.Enabled = False
Timer14.Enabled = False
Timer11.Enabled = False
Timer15.Enabled = False
End If
```

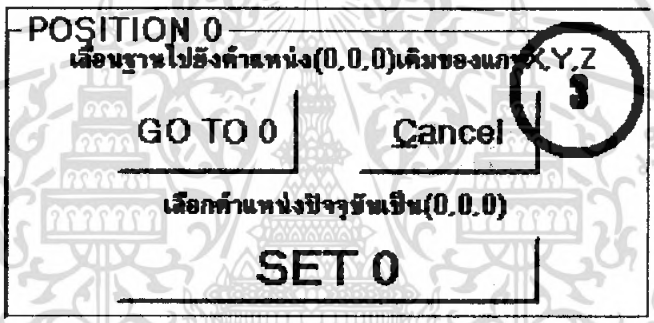
- โปรแกรมส่วนของไทเมอร์ของXทางบวก

```
Function IncreaseX()
If GoToPositionX > CountX Then
Shape1.BackColor = vbGreen
Label10.Caption = "X: +ON"
  For i = 1 To 2 * Mid(CInt(GoToPositionX - fixX), 1, 1)
  If SelfTimer Then ' Check SelfTimer
     Out pwrite, &H3
     CountX = CountX + (1 / Text4.Text) ' 1/x is number of pulse per 1mm
     Label13.Caption = "1"
     Label20.Caption = "GoToPositionX = " & (GoToPositionX) & " deltaX = " & deltaX & "
CountX = " & (CountX)
     Sleep (0.6)
  Else
     Out pwrite, &H2
     Label13.Caption = "0"
  End If
  SelfTimer = Not SelfTimer ' toggle SelfTimer
  Next i
Else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Label10.Caption = "X: OFF"
Label20.Caption = "GoToPositionX = " & (GoToPositionX) & " deltaX = " & deltaX & "
CountX = " & (CountX)
Timer9.Enabled = False
Shape1.BackColor = vbRed
checkX = 1
checkRun
End If
End Function
```

3.1.2.3 POSITION 0



รูปที่ 3.12 แสดงส่วนควบคุม POSITION0

ถูกแบ่งออกเป็น2ส่วนคือ

- **GO TO 0**

เป็นการกลับไปยังตำแหน่ง(0 ,0, 0)เดิม จากตำแหน่งใดๆ โปรแกรมในส่วนของGO TO 0 มีดังนี้

```
Text1.Text = "0"
Text2.Text = "0"
Text3.Text = "0"
Sleep (0.6)
Command7_Click
```

ถ้าคลิกปุ่มGO TO 0 โปรแกรมจะกำหนดให้ค่าทั้ง3ช่องของส่วนMANUAL DATA INPUT เป็น0ทั้งหมด แล้วไปกดปุ่มSTARTของส่วนMANUAL DATA INPUT โดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **SET 0**

เป็นการตั้งค่าตำแหน่งปัจจุบันให้เป็นตำแหน่ง(0, 0, 0) โปรแกรมในส่วนของSET 0 ดังนี้

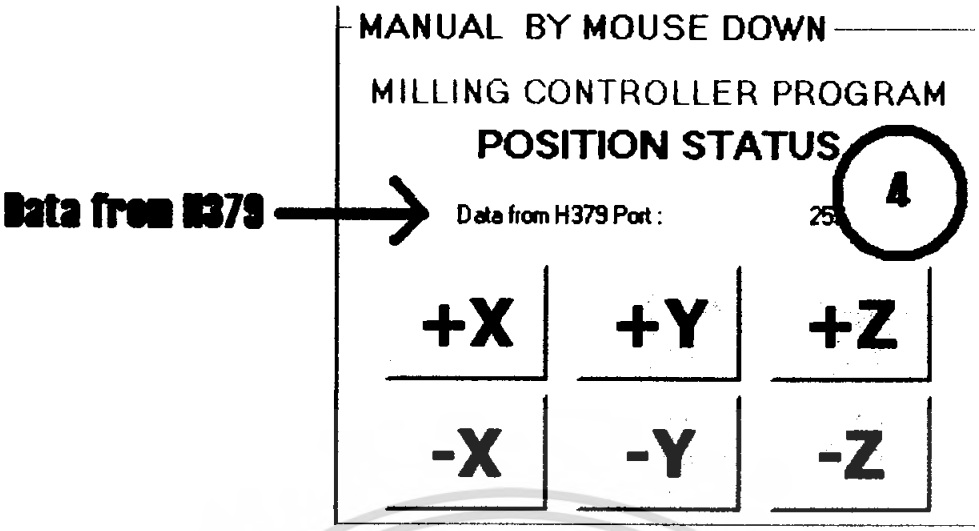
```
Label14.Caption = "BUTTON STATUS"
Command8.Enabled = True
CountX = 0
CountY = 0
CountZ = 0
Text1.Text = "0"
Text2.Text = "0"
Text3.Text = "0"
GoToPositionX = 0
GoToPositionY = 0
GoToPositionZ = 0
deltaX = 0
deltaY = 0
deltaZ = 0
Label10.Caption = "SET0"
Label20.Caption = "GoToPositionX = " & GoToPositionX & " deltaX = " & deltaX & " CountX
= " & CountX
Label21.Caption = "GoToPositionY = " & GoToPositionY & " deltaY = " & deltaY & " CountY
= " & CountY
Label22.Caption = "GoToPositionZ = " & GoToPositionZ & " deltaZ = " & deltaZ & " CountZ =
" & CountZ
```

เป็นการกำหนดค่าต่างๆ ทั้งการนับตำแหน่ง ค่าใน Textbox ของตำแหน่งทั้งหมดให้เป็น 0 เพื่อทำการกำหนดให้ตำแหน่งปัจจุบันของเครื่องเป็นตำแหน่ง (0, 0, 0)

3.1.2.4 MANUAL BY MOUSE DOWN

เป็นการเคลื่อนที่ไปบนตำแหน่งต่างๆ โดยถูกออกแบบขึ้นเพื่อความสะดวกของผู้ใช้ กรณีที่มีการเคลื่อนที่ไปชนลิมิตสวิทช์ โดยผู้ใช้สามารถคลิกที่ปุ่มต่างๆ เพื่อเลื่อนฐานไม่ให้ชนลิมิตสวิทช์ โดยจะไม่มีการนับค่าตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แสดง MANUAL BY MOUSEDOWN และ DATA FROM H379 PORT

DATA FROM H379 PORT

เป็นส่วนของการรับค่าจากพอร์ตขนาน ซึ่งถ้ามีการชนลิมิตสวิทช์ ที่ตำแหน่งใดตำแหน่งหนึ่ง ก็จะมีค่าตัวเลขเปลี่ยนแปลงไป และนำค่าที่เปลี่ยนแปลงไปนี้มาเขียนโปรแกรมเพื่อทำการตัดการทำงานของปุ่มในโปรแกรม ที่ลิมิตสวิทช์ถูกชน

- หลักการเขียนโปรแกรมDATA FROM H379 PORT

ก่อนเขียนโปรแกรมรับค่าจากภายนอก ต้องประกาศฟังก์ชันเพิ่มเติมเสียก่อน ดังนี้

```
Private Declare Function Inp Lib "inpout32.dll" Alias "Inp32" (ByVal PortAddress As Integer)
As Integer
```

- โปรแกรมส่วนไทเมอร์ของH379

เมื่อมีการเปลี่ยนแปลงของตัวเลขต่างๆเนื่องมาจากการที่ลิมิตสวิทช์ที่ตำแหน่งนั้นๆถูกชน เราก็สามารถนำค่าตัวเลขที่เปลี่ยนแปลงนั้นๆ มาเขียนเป็นฟังก์ชันเพิ่มเติมได้ ดังนี้

```
N = Inp(pread)
Label2.Caption = N
Select Case N
Case 126, 120 'No contact on limit switch
lblStatus.Caption = "EDM CONTROLLER PROGRAM"
Command1.Enabled = True
Command2.Enabled = True
Command3.Enabled = True
Command4.Enabled = True
```

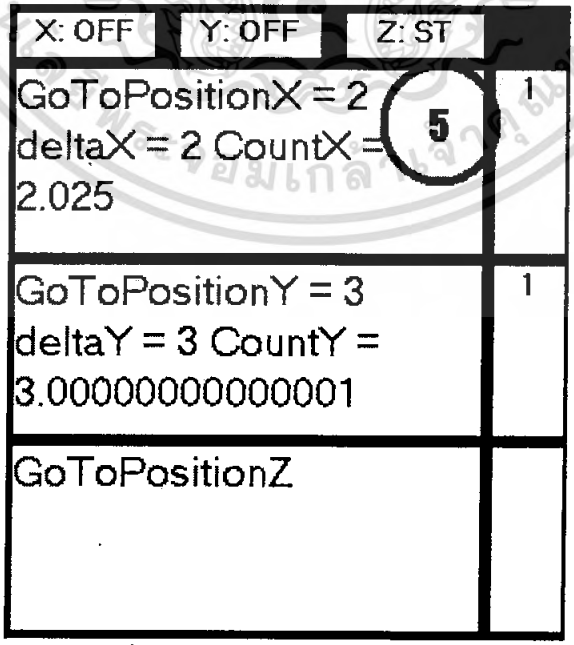
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Command6.Enabled = True
Command12.Enabled = True
Command13.Enabled = True
Command14.Enabled = True
Command15.Enabled = True
Command17.Enabled = True
Case 118, 112 'contact on pin15 +x
Out pwrite, &H40 'cut off electric current drill
Command2.Enabled = False
Command13.Enabled = False
Timer4.Enabled = False
Timer13.Enabled = False
Timer18.Enabled = False
```

เช่น กรณีที่ค่าส่งมาเป็น118 หรือ 112 เราเขียนโปรแกรมให้ส่งสัญญาณ(H40)ไปตัดการทำงานของสว่าน และตัดการทำงานของปุ่มที่ลิมิตสวิทช์เคลื่อนที่ไปชน(Command2.Enabled = False,Command13.Enabled = False)

3.1.2.5 POSITION STATUS

เป็นส่วนที่แสดงค่าตำแหน่งปัจจุบันของแกนต่างๆ ซึ่งถูกเขียนไว้ที่ส่วนอื่นของโปรแกรม

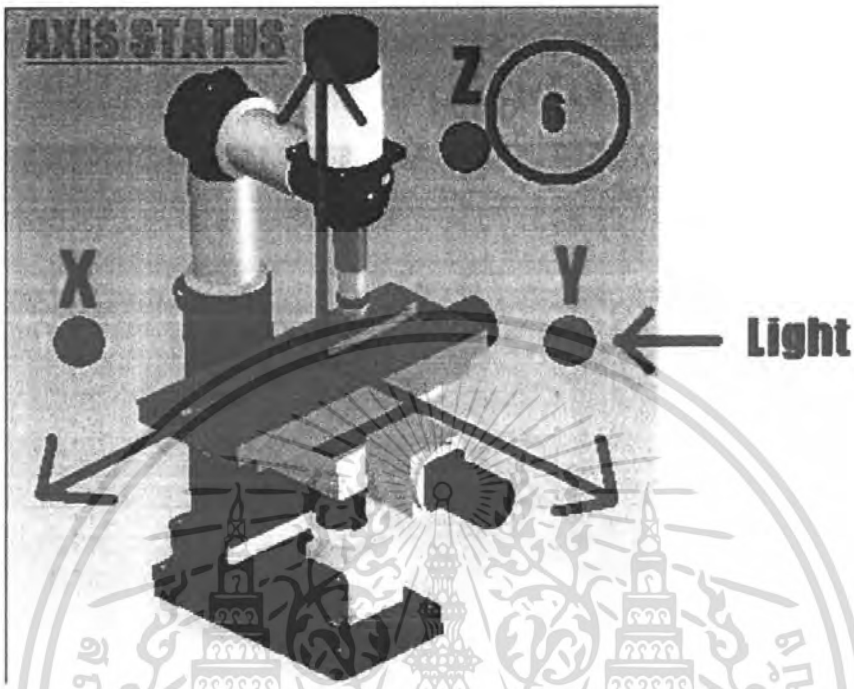


รูปที่ 3.14 แสดง POSITION STATUS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.6 AXIS STATUS

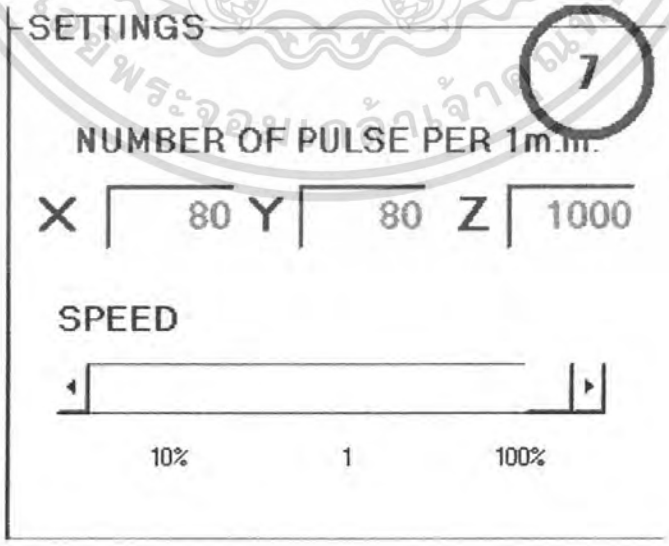
เป็นส่วนของการแสดงว่า ณ เวลานั้น มีการเคลื่อนที่บนแกนใดบ้าง ถ้ามีการเคลื่อนที่บนแกนใด จะมีการเปลี่ยนสีของไฟจากแดงเป็นเขียว



รูปที่ 3.15 AXIS STATUS

3.1.2.7 SETTINGS

ส่วนของการปรับแต่งค่าเริ่มต้นต่างๆของโปรแกรม



รูปที่ 3.16 แสดง SETTINGS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งมี2ค่าที่ผู้ใช้ต้องกำหนดก่อนการเริ่มต้นการใช้โปรแกรม คือ

- **NUMBER OF PULSE PER 1 m.m.**

จำนวนพัลส์ที่ต้องส่งไปยังสเต็ปมอเตอร์ของแต่ละแกนเพื่อที่จะให้ฐานเคลื่อนที่ไปตามแกน 1 มิลลิเมตร ซึ่งจะนำค่าทั้ง3ค่านี้ไปใช้เขียนโปรแกรมในส่วนต่างๆ

- **SPEED**

ส่วนของการปรับแต่งความเร็วในการส่งพัลส์ หรือความเร็วในการเคลื่อนที่ของฐานของเครื่องเจาะซึ่งนำค่านี้ไปเขียนโปรแกรมเพื่อปรับความเร็วของการเคลื่อนที่ของฐานเครื่อง

3.2 การรับค่าจากไฟล์ G-CODE เพื่อประมวลคำสั่ง

3.2.1 หลักการพิจารณาค่าG-Code ที่อ่านเข้ามา

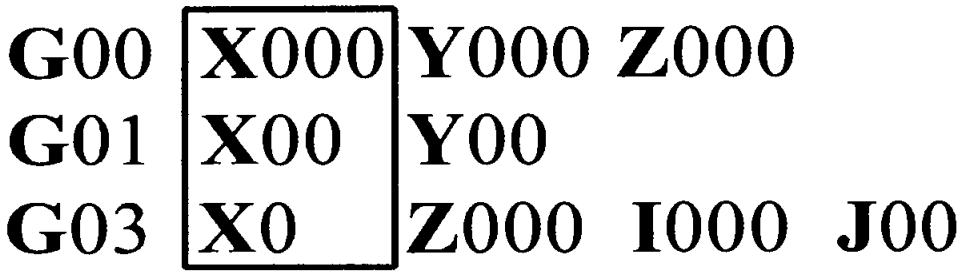
เนื่องจากโปรแกรมส่วน MANUAL FORM นั่นต้องรับข้อมูลจากหน้าต่าง DXF to G-Code Converter ซึ่งทำการส่งข้อมูลการแปลงไฟล์จากไฟล์ DXF เป็นไฟล์ G-CODE ซึ่งวิธีการรับค่าจากไฟล์ G-CODE มาประมวลเป็นคำสั่งเพื่อสั่งการเครื่องกัด ทำได้ดังนี้

```
G00 X000 Y000 Z000
G01 X00  Y00
G03 X0   Z000 I000 J00
```

รูปที่ 3.17 แสดงการอ่านค่าไฟล์ G-CODE เพื่อนำไปประมวลคำสั่ง

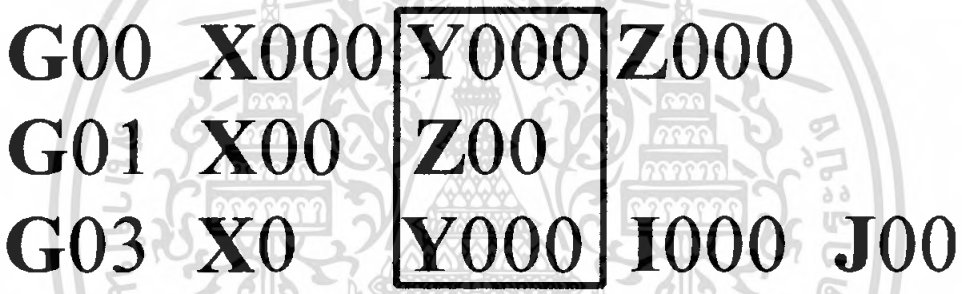
จากรูปที่3.17เราต้องทำการพิจารณาค่าหลักๆที่สามารถเกิดขึ้นได้ในการอ่านไฟล์ G-CODE ซึ่งค่าที่ต้องพิจารณาตัวแรกก่อนก็คือส่วนของ ต้นบรรทัดเป็นส่วนที่จะบอกการทำงานของบรรทัดนั้นๆ ว่าจะต้องดำเนินการต่อไปอย่างไร โดยถ้าค่าที่อ่านได้จากต้นบรรทัดนั้นตรงกับการทำงานของส่วนใด ก็จะทำการส่งค่าไปยังส่วนนั้นเพื่อให้ดำเนินงานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

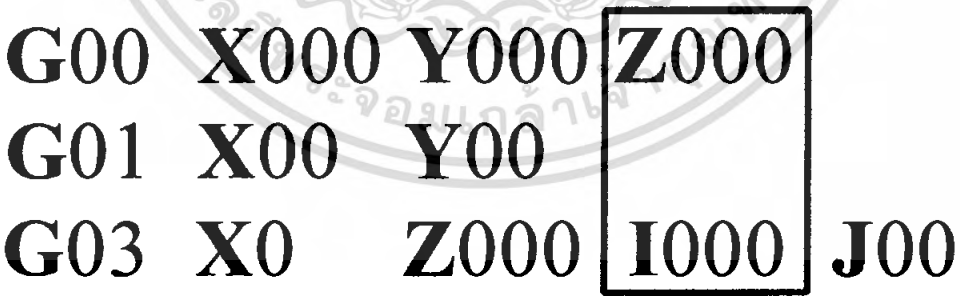


รูปที่ 3.18 แสดงการอ่านค่าไฟล์ G-CODE ที่เป็นจำนวนของข้อมูล

จากรูปที่3.18 ในส่วนนี้ ค่าที่ต้องทำการพิจารณา เป็นค่าของจำนวนที่อยู่หลังตัวแปรต่างๆ ซึ่งมีจำนวนไม่เท่ากัน การนำค่าไปใช้ต้องมั่นใจว่าจะนำเอาค่าที่ถูกส่งไปให้ยังส่วนการทำงานที่เกี่ยวข้องนั้นเป็นค่าที่ถูกต้องดังนั้นการอ่านค่าในส่วนนี้ต้องใช้การคำนวณหาตำแหน่งของค่านั้นเพื่อที่จะส่งค่าได้ไม่ผิดพลาด



รูปที่ 3.19 แสดงการอ่านค่าตัวแปรที่แตกต่างกัน



รูปที่ 3.20 แสดงตัวแปรที่มีค่าไม่เหมือนกัน

จาก รูปที่19 และ 3.20 เราต้องพิจารณาถึงกรณีที่นอกจากค่าในแต่ละตัวแปรอาจไม่เท่ากันแล้ว ตำแหน่งของตัวแปรที่บางตัวไม่มีในคำสั่ง(แถว)นั้น หรือ ตัวแปรอื่นๆที่มีในแต่ล่ะฟังค์ชั่นที่ไม่เหมือนกัน เราต้องสามารถนำค่าไปใช้ได้ถูกต้อง เช่นเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 ขั้นตอนการนำค่าที่อ่านได้ไปใช้

รูปที่ 3.21 แสดงลักษณะข้อมูลของไฟล์ G-CODE

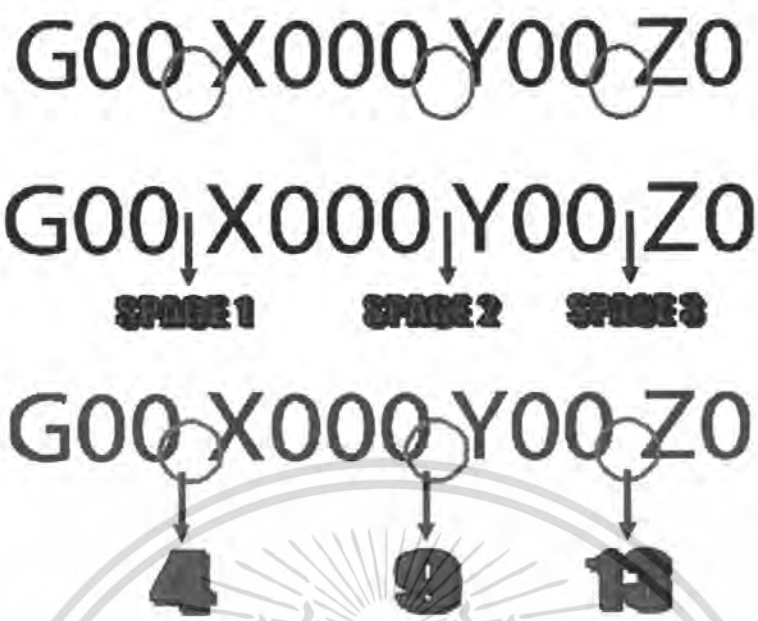
เราทำการเช็คข้อมูลของบรรทัดแรกที่อ่านเข้ามาว่าต้องการให้ทำฟังค์ชั่นใด เพื่อจะได้เลือกส่งค่าให้กับตัวฟังค์ชั่นส่วนนั้นได้ถูกต้อง จากนั้นเมื่อก็เข้าสู่ขั้นตอนการ อ่านค่าข้อมูลในแต่ละบรรทัดเพื่อส่งไปทำงานดังนี้

รูปที่ 3.22 แสดง การนับข้อมูล

```
Function FindChar()        ' ฟังค์ชั่นนี้ทำการเก็บค่าต่างๆ เพื่อเตรียมส่งค่าไปยังส่วนแมนนวล เพื่อ
                           ควบคุม การทำงาน จากค่าตำแหน่งที่อ่านได้'
G = Len(O)                 'ทำการนับตัวแปร O ที่เก็บค่าที่อ่านเข้ามาแต่ละบรรทัดไว้'
```

เราทำการอ่านค่าของไฟล์ G-Code แต่ละบรรทัดก่อนเก็บไว้ที่ตัวแปร O โดยต่อมาเรานับจำนวนตัวอักษรในบรรทัดนั้นๆเก็บค่าไว้ใน G เพื่อที่จะได้ทราบว่าบรรทัดนั้นมีข้อมูลยาวเท่าใดเพื่อนำไปใช้ในขั้นต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

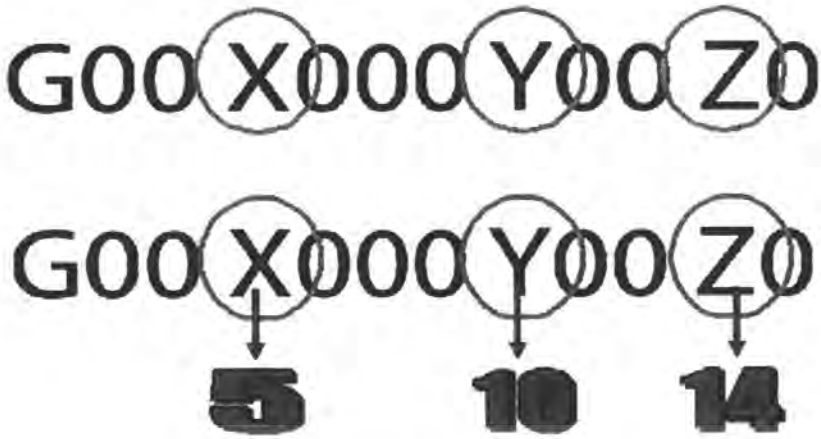


รูปที่ 3.23 แสดงการนับช่องว่าง

```
'Find space for each line input                    'ทำการหาตำแหน่งของตัวช่องว่างหรือspace'
 For N = 1 To G
 If Mid(O, N, 1) = " " Then
 SPACE(j) = N
 j = j + 1
End If
Next N
SPACE(j) = Len(O) + 1                              'ตัวแปรที่เก็บค่าของตำแหน่งของตัวช่องว่าง'
```

หลังจากทราบว่าบรรทัดนั้นมีข้อมูลอยู่กี่ตัวอักษรแล้ว เราต้องทำการนับช่องว่างที่มีว่าอยู่ที่ตำแหน่งใดบ้าง ในส่วนนี้เป็นการเช็คค่าที่ว่ามีข้อมูล ของตัวแปรกี่ตัวเพราะ Space นั้นจะเป็นตัวขั้นของข้อมูลแต่ละชุดในแต่บรรทัดกลังจากขั้นตอนนี้เราก็จะทราบคร่าวๆ ว่ามีข้อมูลกี่ตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.24 แสดงการนับตัวแปร

```
'Find X                          'ทำการหาค่าของตำแหน่ง  X '
For N = 1 To G
If Mid(O, N, 1) = "X" Then
xx = N
End If
Next N

'Find Y                          'ทำการหาค่าของตำแหน่ง Y '
For N = 1 To G
If Mid(O, N, 1) = "Y" Then
yy = N
End If
Next N

'Find Z                          'ทำการหาค่าของตำแหน่ง  Z '
For N = 1 To G
If Mid(O, N, 1) = "Z" Then
zz = N

End If
Next N
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราทำการเก็บค่าตำแหน่งของ ตัวแปรทุกตัวในแต่ละบรรทัด(X, Y, Z, I, J) เก็บค่าไว้เพื่อใช้ในการคำนวณหาว่าจำนวนที่อยู่หลังตัวแปรนั้นมีค่าเท่าใด โดยนำตัวแปรทั้งหมดมาคำนวณดังนี้

SPACE [2] – SPACE [1] - 2 = !!

เราเช็คค่าระหว่างช่องว่างเพื่อหาว่ามีค่าข้อมูลเท่าใดอยู่หลังตัวแปร(ในตัวอย่างจะเท่ากับ 3) เพื่อนำค่าที่มีไปเก็บยังตัวแปรของมันเอง โดยเช็คจาก

SPACE [1] + 1 = !!

เราดูว่าค่าที่อยู่หลังช่องว่างที่1 เป็นตัวอะไร ก็ให้นำค่าไปพักไว้ยังตัวแปรของมันเอง(ในที่นี้คือ x) โดยจะส่งค่าไปไว้ยังในส่วนการทำงานของแมนนวล และทำการเช็คช่องว่างถัดไปเรื่อยๆ ก็สามารถเก็บข้อมูลทั้งหมดและส่งค่าไปยังส่วนการทำงานของฟังค์ชั่นที่ต้องการสั่งให้ทำงานได้ถูกต้องนั่นเอง

ดังนั้นการอ่านค่าตัวแปรที่เป็นตัวอย่างด้านบนสรุปได้ดังนี้

G00 X000 Y00 Z0

1. โปรแกรมก็จะอ่านค่า 3 ตัวแรกก่อนเพื่อกำหนดฟังค์ชั่นที่จะใช้งาน ในที่นี้ คือ การเคลื่อนที่แบบไม่ เปิดการทำงานของมอเตอร์หัวกัด
2. โปรแกรมทำการนับตัวอักษรของบรรทัดนั้นๆ
3. โปรแกรมทำการเก็บตำแหน่งของSpace และค่าตัวแปร(X, Y, Z, I, J)
4. โปรแกรมทำการคำนวณค่าเพื่อหาจำนวนของหลังตัวแปรต่างและส่งค่าไปเพื่อใช้ในการทำงานส่วนต่อๆไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การคำนวณหาจำนวนพัลซ์ที่จะทำให้แกน X และแกน Y เคลื่อนที่

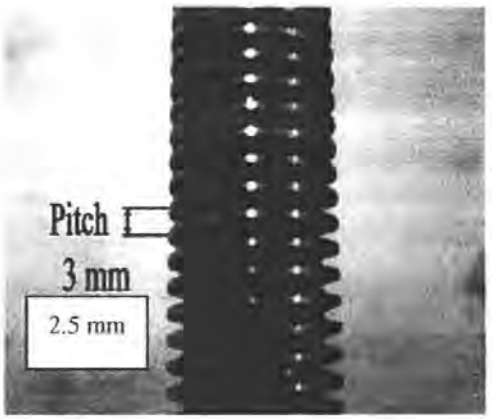
เนื่องจากเครื่อง CNC เป็นเครื่องกัดอัตโนมัติที่มีความละเอียดของชิ้นงานสูง ดังนั้นเราจึงต้องหาจำนวนพัลซ์ที่แน่นอนในการที่จะป้อนเข้าไปยังสเตปมอเตอร์ เพื่อที่จะทำให้ระยะของแกนเคลื่อนที่ได้ 1 มิลลิเมตร

หลักการทำงานของระบบควบคุมการเคลื่อนที่ของแกน X และแกน Y จะมีลักษณะเหมือนกัน คือจะใช้สเตปมอเตอร์ต่อเข้ากับสกรูเพื่อทำหน้าที่หมุนสกรูโดยตรง ซึ่งสาเหตุที่เราเลือกใช้สเตปมอเตอร์เพราะว่าเราจะสามารถควบคุมระยะในการเคลื่อนที่ได้ โดยเราจะมีวิธีการคำนวณจำนวนพัลซ์ที่จะจ่ายให้สเตปมอเตอร์เพื่อให้ระยะของแกนเคลื่อนที่ได้ 1 มิลลิเมตร ดังนี้

รูปที่ 3.25 แสดงข้อมูลของสเตปมอเตอร์

จะเห็นว่ามอเตอร์ของแกน X และแกน Y มีสเตปการเคลื่อนที่เท่ากับ 1.8 Deg/Step (มอเตอร์จะหมุน1.8 องศาต่อการจ่าย 1 พัลซ์) ซึ่งการที่จะให้มอเตอร์หมุนครบหนึ่งรอบเป็นมุม 360 องศานั้นเราจะต้องจ่ายพัลซ์ให้มอเตอร์ทั้งหมด 200 พัลซ์ และถ้าเกิดมอเตอร์หมุนไป 1 รอบ ก็จะทำให้สกรูหมุนไป 1 รอบเช่นกัน โดยจะได้ระยะการเคลื่อนที่เท่ากับระยะลีดสกรู คือ 2.5 มิลลิเมตร เพราะฉะนั้นถ้าเราต้องการหาจำนวนพัลซ์ที่จ่ายเข้าไปแล้วจะทำให้แกน X และแกน Y เคลื่อนที่ได้ระยะ 1 มิลลิเมตรจะต้องทำการคำนวณดังนี้ 200 / 2.5 = 80 พัลซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.26 แสดงระยะพิทของสกรู

3.3.1 การคำนวณหาจำนวนพัลซ์ที่จะทำให้แกน Z เคลื่อนที่ได้ระยะ 1 มิลลิเมตร

มอเตอร์ของแกน Z เป็นสเตปมอเตอร์ที่มีลักษณะเหมือนกับของแกน X และแกน Y แต่จะถูกต่อเข้ากับชุดเฟืองก่อนที่จะนำไปขับสกรูอีกที โดยชุดเฟืองที่นำมาต่อนี้เป็นชุดเฟืองแบบตัวหนอน ซึ่งจะมีหลักการทำงานดังรูป

รูปที่ 3.27 แสดงการทำงานของเฟือง

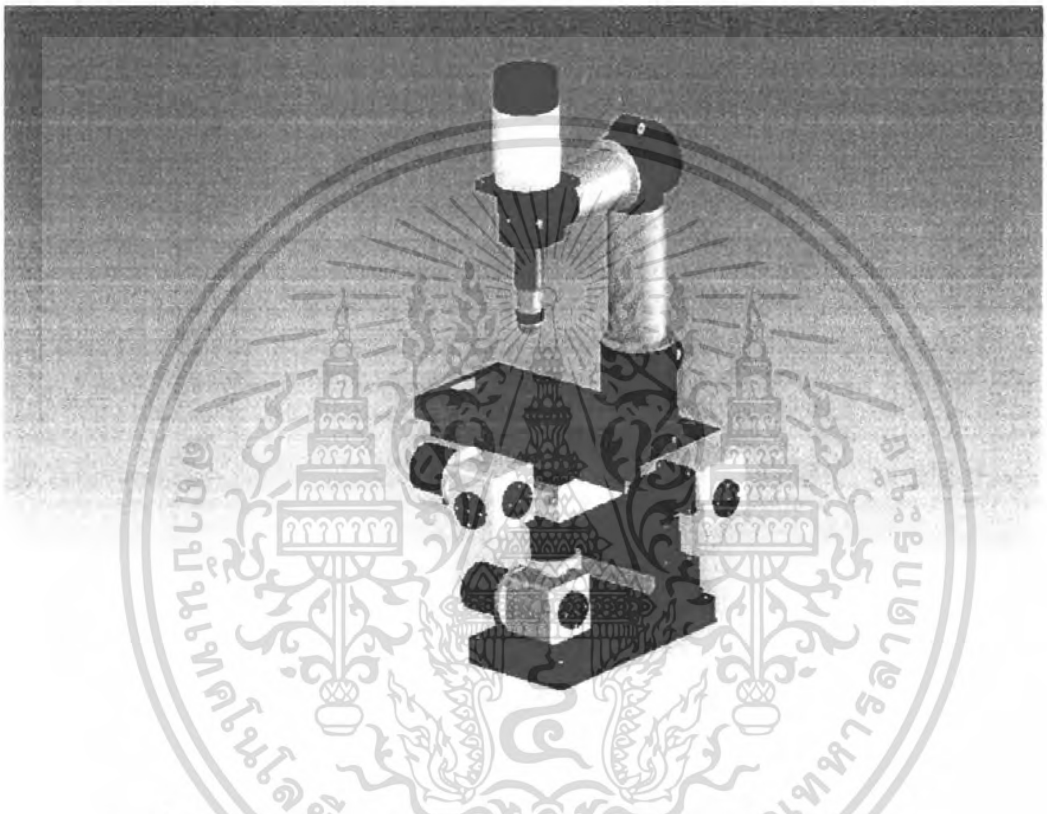
จะเห็นว่าเมื่อมอเตอร์ของแกน Z หมุน 1 รอบ จะทำให้ฟันเฟืองเลื่อนไป 2 ช่อง โดยที่เฟืองตัวนี้จะถูกต่อเข้ากับสกรูที่ใช้เป็นตัวขับเคลื่อนของแกน Z โดยตรง และเนื่องจากเฟืองมีฟันทั้งหมด 25 ซี่ ดังนั้นถ้าต้องการให้เฟืองหมุนครบ 1 รอบ จะต้องจ่ายพัลซ์เพื่อให้มอเตอร์ของแกน Z หมุนไป 25 / 2 = 12.5 รอบ (เฟืองมีอัตราทดเท่ากับ 12.5 : 1)

เพราะฉะนั้นถ้าเราต้องการให้สกรูของแกน Z หมุนไป 1 รอบเราจะต้องจ่ายพัลซ์เข้าไปยังมอเตอร์ของแกน Z เป็นจำนวน 12.5 เท่าของที่จ่ายเข้าไปยังสเตปมอเตอร์ของแกน X และแกน Y ซึ่งจะมีค่าเท่ากับ 12.5 * 80 = 1000 พัลซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ส่วนการปรับปรุงเครื่องจักร

ในการปรับปรุงเครื่องจักรเดิมให้เป็นเครื่องกัดนั้นได้ใช้เส้นทางการเคลื่อนที่ของแกนต่างๆ ตามเดิมเพราะสามารถใช้กับงานกัดได้ แต่เนื่องจากการเคลื่อนของแต่ละแกนนั้นต้องใช้เวลานานมากหากต้องการให้เคลื่อนไปยังตำแหน่งที่ต้องการ ดังนั้นจึงทำการแก้ไขโดยการเปลี่ยนแปลงการทดรอบของเฟืองโดยมีรูปภาพอธิบายดังนี้



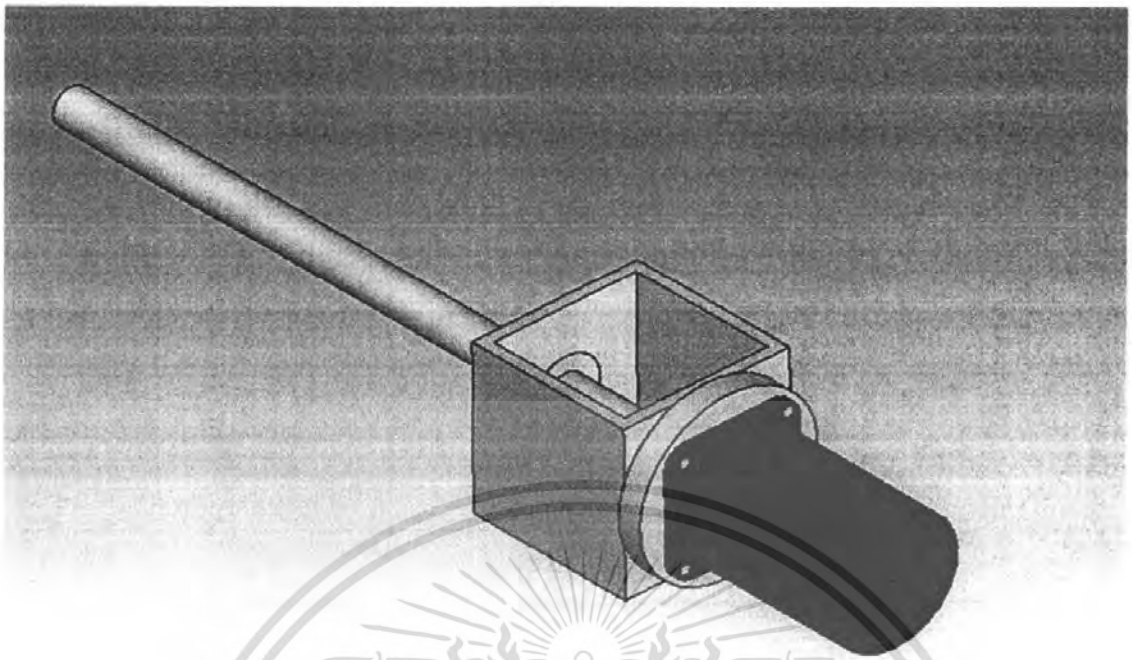
รูปที่ 3.28 ภาพประกอบโดยสมบูรณ์ของเครื่องกัดก่อนมีการแก้ไขปรับปรุง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มอเตอร์ตัวเก่าซึ่งมีการทดรอบของเฟืองอยู่หลายเท่าตัว ทำให้การเคลื่อนที่ของฐานชิ้นงานในแนวแกน X และแนวแกนY ต้องใช้เวลานานมากเพื่อที่จะเคลื่อนที่ไปยังตำแหน่งที่ต้องการ จึงได้ทำการปรับเปลี่ยนตำแหน่งการวางมอเตอร์กับแกนเคลื่อนที่ใหม่ ดังรูปที่ 3.27 และเปลี่ยนมอเตอร์ตัวใหม่เป็นแบบซึ่งไม่มีการทดรอบเลย คือทำการต่อมอเตอร์เข้าโดยตรงกับแกนหมุน เพื่อทำให้แกนหมุนเคลื่อนที่เป็นจำนวนรอบเท่ากับการหมุนของมอเตอร์ เมื่อทดลองทำให้แกนฐานหมุนด้วยมอเตอร์ตัวนี้พบว่าการหมุนเร็วขึ้นและใช้เวลาน้อยลงมากในการเคลื่อนที่

รูปที่ 3.29 มอเตอร์ตัวเก่าที่ทำให้การเคลื่อนที่ของเครื่องกัดมีอัตราทด 12.5:1

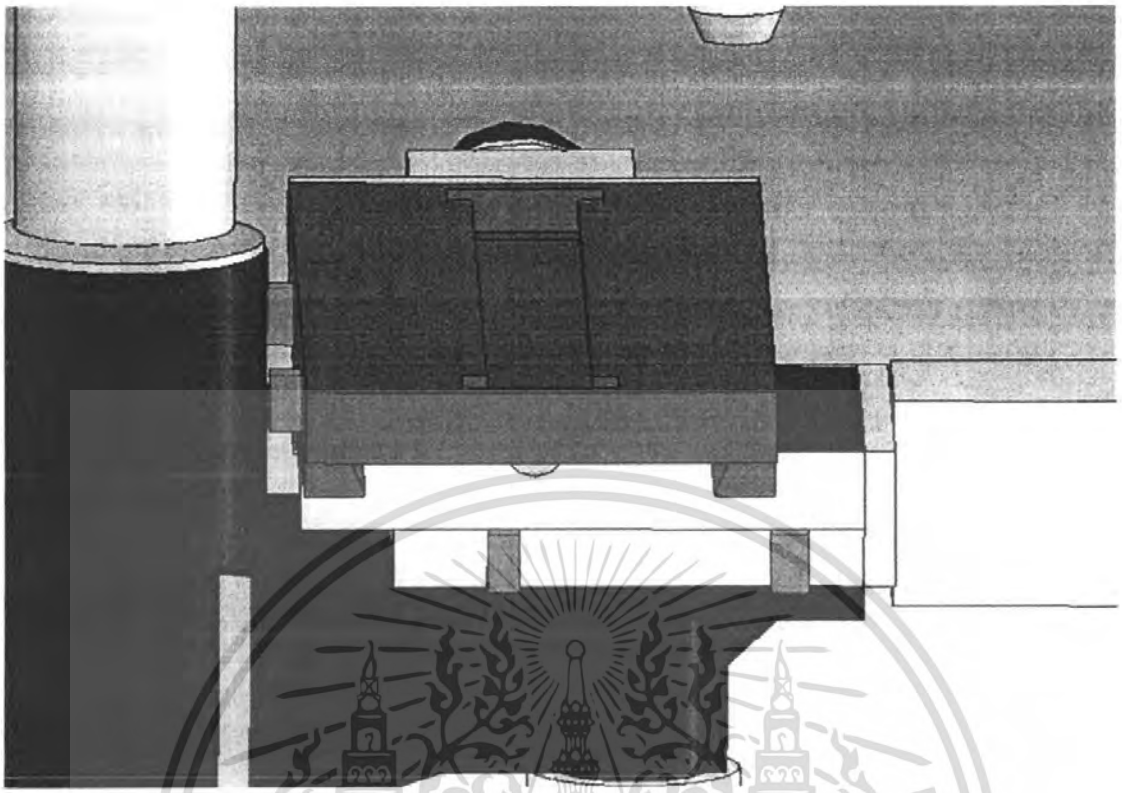
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.30 มอเตอร์ตัวใหม่ที่ทำให้การเคลื่อนที่ของเครื่องกัดมีอัตราทด 1:1

นอกจากนี้ได้ทำการติดตั้งลิมิตสวิตซ์ จำนวน 4 ตัว ในแนวแกน X และ Y แกนละ 2 ตัว เพื่อเป็นการป้องกันความเสียหายที่อาจเกิดขึ้นกับเครื่องกัด ในกรณีที่แกน X หรือ แกน Y เคลื่อนที่ไปในระยะมากเกินไป แล้วหมดระยะเกลียวของแกนมอเตอร์ ทำให้แกนมอเตอร์หยุดทำงาน และอีกสาเหตุหนึ่งเพื่อเป็นการแสดงขอบเขตของชิ้นงานที่ต้องการกัดไปด้วยในตัว แต่สำหรับแกน Z ไม่มีความจำเป็นเนื่องจากการเคลื่อนที่ของแกนมีไม่มากนัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.31 ตำแหน่งของลิมิตสวิตซ์ที่ติดไว้เพื่อหยุดการทำงานของมอเตอร์

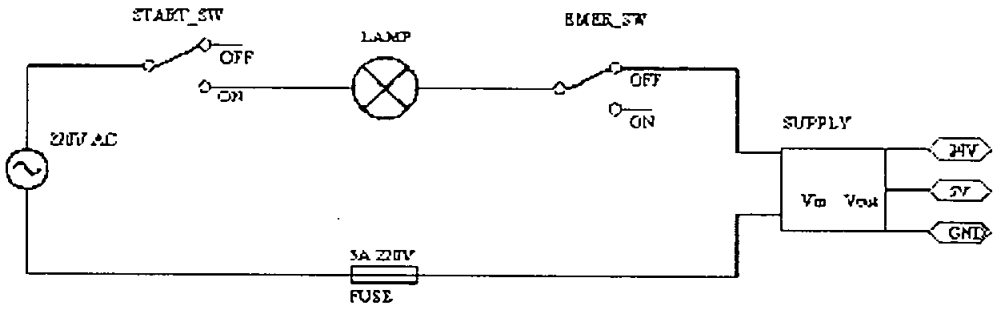
3.5 ส่วนวงจรควบคุมเครื่องเจาะ

วงจรและอุปกรณ์อิเล็กทรอนิกส์ที่นำมาใช้ในการพัฒนาเครื่องกัดประกอบไปด้วยอุปกรณ์ Emergency switch, Speed control Dc motor circuit, วงจรตัดการทางาน และDc motor Limit switch ซึ่งมีรายละเอียด ดังนี้

3.5.1 Emergency switch

Emergency switch ใช้เป็นชนิด switch normally close ใช้ทำการตัดการทำงานของวงจร โดยการตัดไฟเลี้ยงวงจร โดยต่อ switch นี้อนุกรมกับ switch on/off ดังแสดงในรูปที่

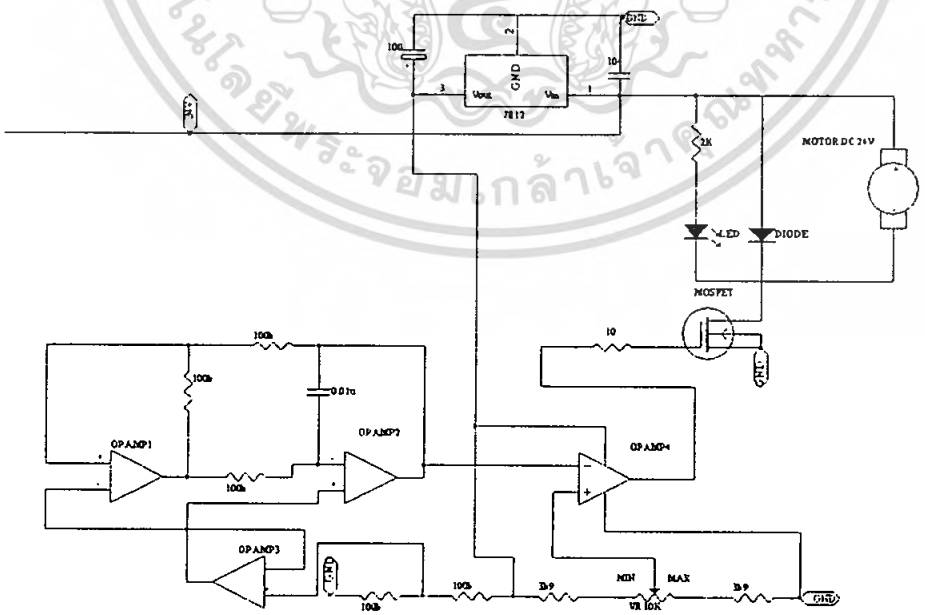
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.32 แสดงวงจร Emergency switch

3.5.2 Speed control Dc motor circuit

ใช้หลักการ(pulse withd modulation) จากรูปที่2. Opamp3 จะทำหน้าที่เป็นตัวจ่ายแรงไฟเปรียบเทียบ 6V ให้กับ Opamp1 และ Opamp2 ซึ่งต่อร่วมกันเป็นวงจรกำเนิดความถี่ โดยที่ Opamp1 จะให้กำเนิดความถี่ ป้อนให้กับวงจร Integrator Opamp3 ได้ความถี่ output ออกไปเป็นรูปฟันเลื่อย ไปที่ขา input ขาหนึ่งของ opamp4 Opamp4 ต่อเป็นวงจร comparator ทำให้ได้ output pulse ที่สามารถปรับเปลี่ยนช่วงความกว้างของ pulse ได้โดยการปรับค่าแรงไฟสำหรับเปรียบเทียบด้วย VR ที่ input อีกขาหนึ่งของตัว comparator สัญญาณ outputที่ได้จากตัว comparator จะป้อนผ่าน ไปยังขาgate ของ MOSFET เพื่อให้ MOSFET อยู่ในสภาวะ on/off เพื่อขับloadส่วนไดโอดที่ต่อขนานกับ load เป็นตัวป้องกันการเกิดแรงไฟย้อนกลับจาก inductorของมอเตอร์

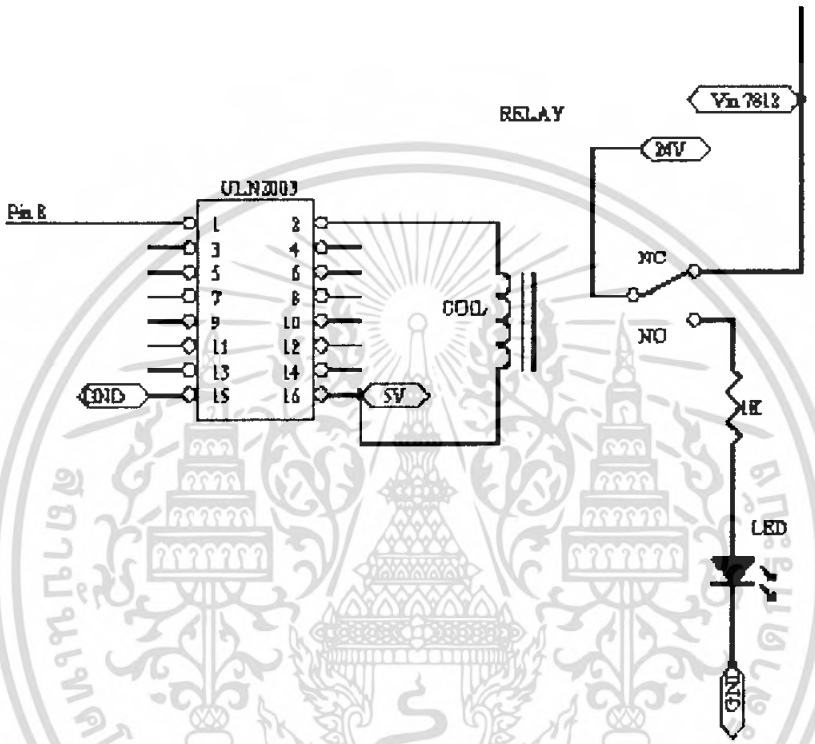


รูปที่ 3.33 Speed control dc motor circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.3 วงจรตัดการทางาน Dc Motor

จากรูปที่3. หลักการทำงานคือเมื่อ Limit switch ถูกชนทำให้วงจรถูกต่อครบวงจร หลังจากนั้นโปรแกรมก็จะประมวลผล และส่งค่าออกมาผ่านULN2003 เพื่อใช้ขับกระแสเข้าขา coil ของ relay เพื่อให้อยู่ในสภาวะ on เพื่อตัดการทำงานของ Dc motor

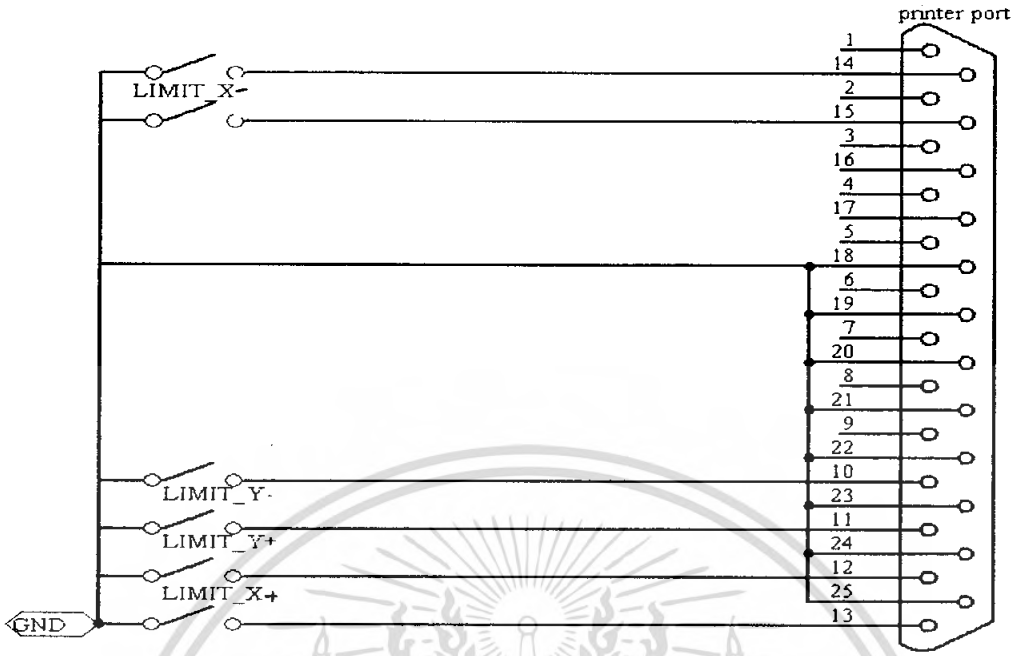


รูปที่ 3.34 วงจรตัดการทางาน Dc Motor

3.5.4 Limit Switch

จากรูปที่4. มีทั้งหมด 6 ตัวเป็น input สำหรับรับค่าเมื่อแต่ละแกนเคลื่อนที่ไปสุด และ limit จะถูกชน เพื่อตัดการทำงาน ของ step motor ของแกนนั้นๆ และ dc motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



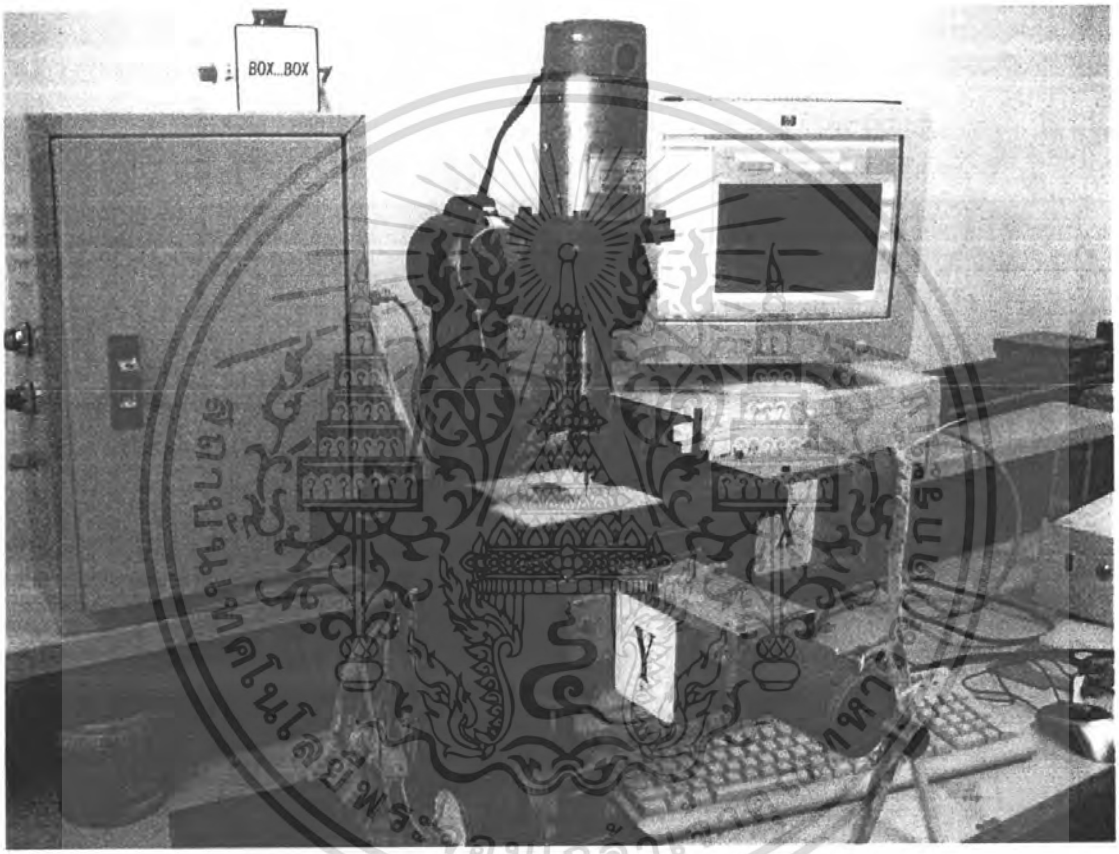
รูปที่ 3.35 แสดงรูป Limit switch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

เมื่อทำการติดตั้งโปรแกรมควบคุมเครื่องกัดเข้ากับตู้ควบคุมสัญญาณ และทำการเชื่อมต่อสัญญาณควบคุมเข้ากับเครื่องกัด แล้วทำการทดลอง ดังนี้

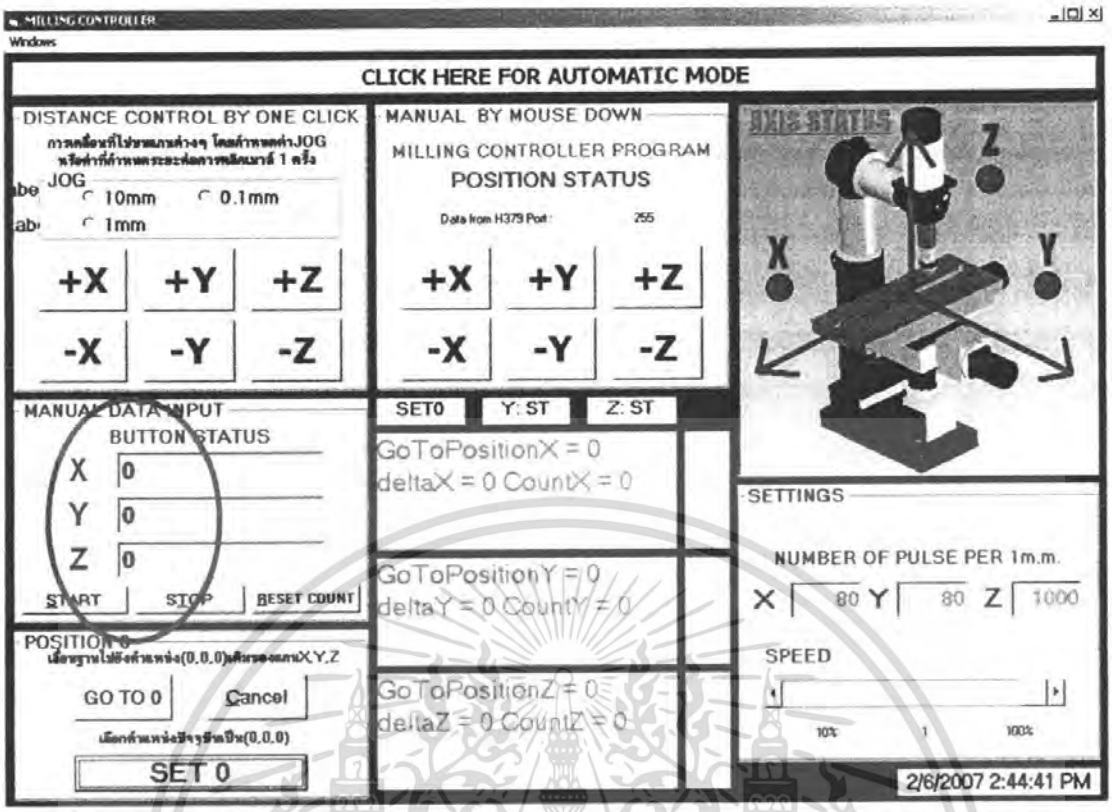


รูปที่ 4.1 แสดงเครื่องกัดที่ติดตั้งเสร็จสมบูรณ์

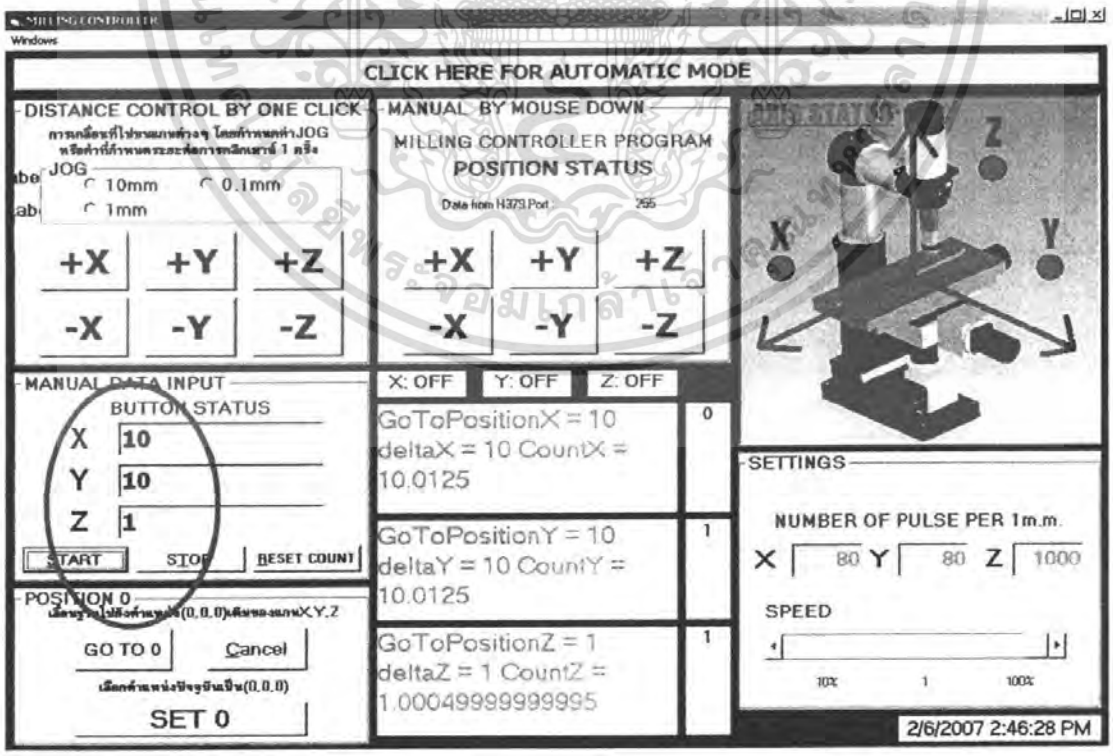
4.1 การทดลองสั่งการเครื่องจักรโดยตรงผ่านหน้าต่าง MANUAL FORM

ทำการทดลองสั่งการมอเตอร์แต่ละแกนของเครื่องกัดไปที่ตำแหน่ง X10 Y10 โดยสั่งการโดยตรงผ่านหน้าต่าง MANUAL FORM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



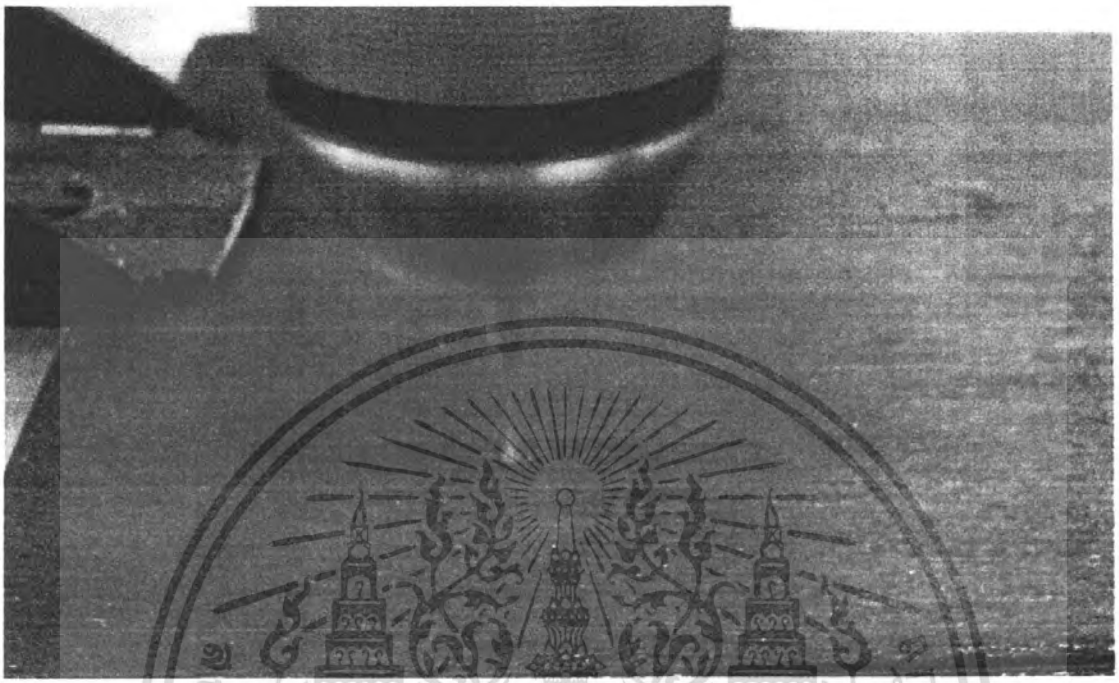
รูปที่ 4.2 แสดงการตั้งค่าจุดเริ่มต้นในหน้าต่าง MANUAL FORM ที่ใช้สั่งการเครื่องกัดโดยตรง



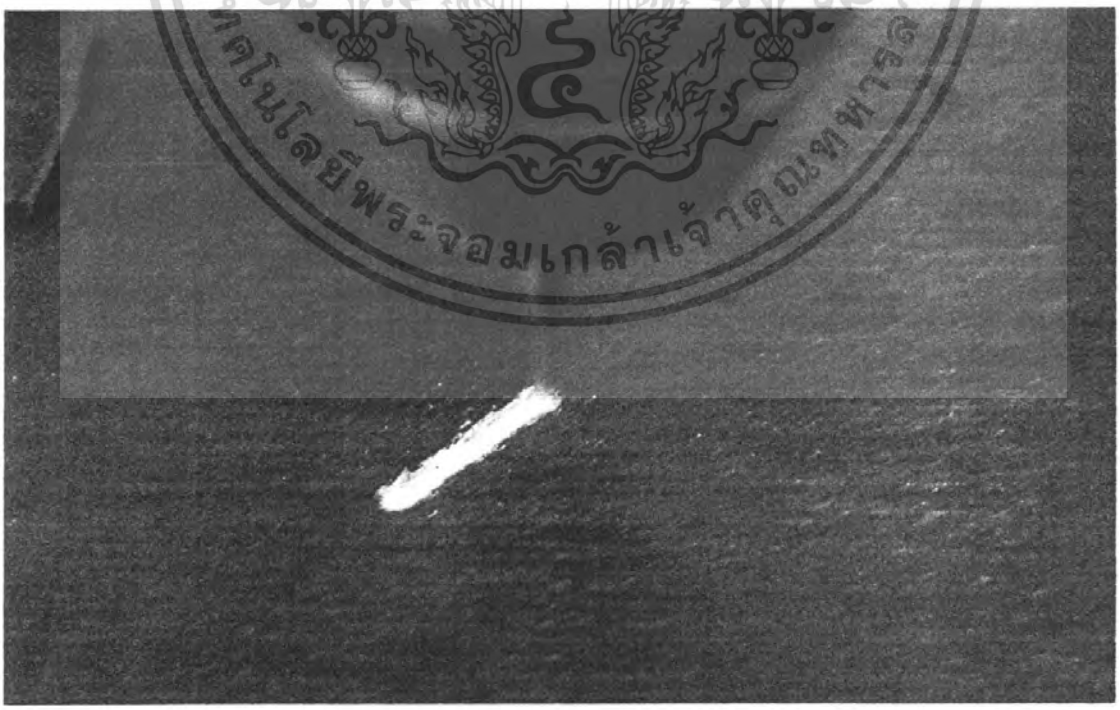
รูปที่ 4.3 แสดงการตั้งค่าจุดสุดท้ายในหน้าต่าง MANUAL FORM ที่ใช้สั่งการเครื่องกัดโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง



รูปที่ 4.4 แสดงตำแหน่งของเครื่องกัดที่จุดเริ่มต้น



รูปที่ 4.5 แสดงตำแหน่งของเครื่องกัดที่จุด X10 Y10

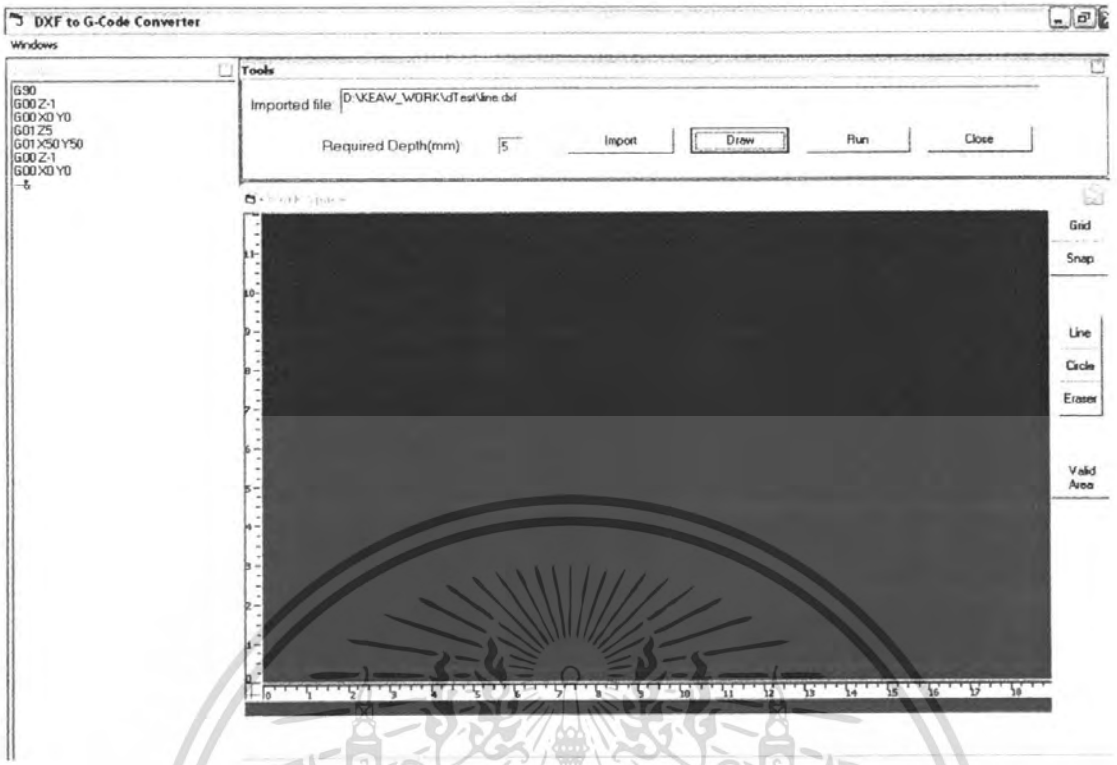
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองกัดรูปเส้นตรง

ทำการทดลองนำไฟล์รูปเส้นตรงจากโปรแกรม AutoCAD แล้วทำการบันทึกเป็นในแบบไฟล์ DXF จากนั้นใช้โปรแกรมที่สร้างขึ้น คือ หน้าต่าง DXF to G-Code Converter นำมาแปลงเป็นไฟล์ G-CODE แล้วสั่งการเครื่องกัดให้ทำการกัดวัสดุเป็นรูปเส้นตรง

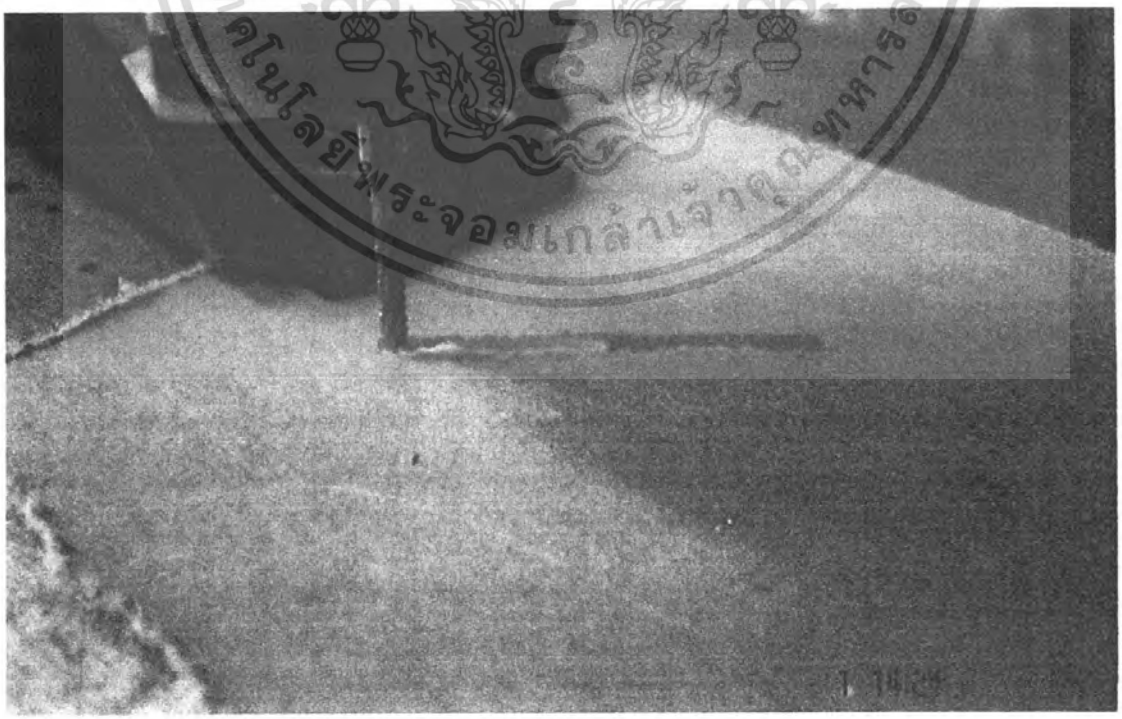
รูปที่ 4.6 แสดงรูปภาพเส้นตรงที่ออกแบบในโปรแกรม AutoCAD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 แสดงหน้าต่าง DXF to G-Code Converter ที่ใช้สั่งการกัดเส้นตรง

ผลการทดลอง



รูปที่ 4.8 แสดงรูปเส้นตรงบนวัสดุที่ได้จากการทำงานของเครื่องกัด

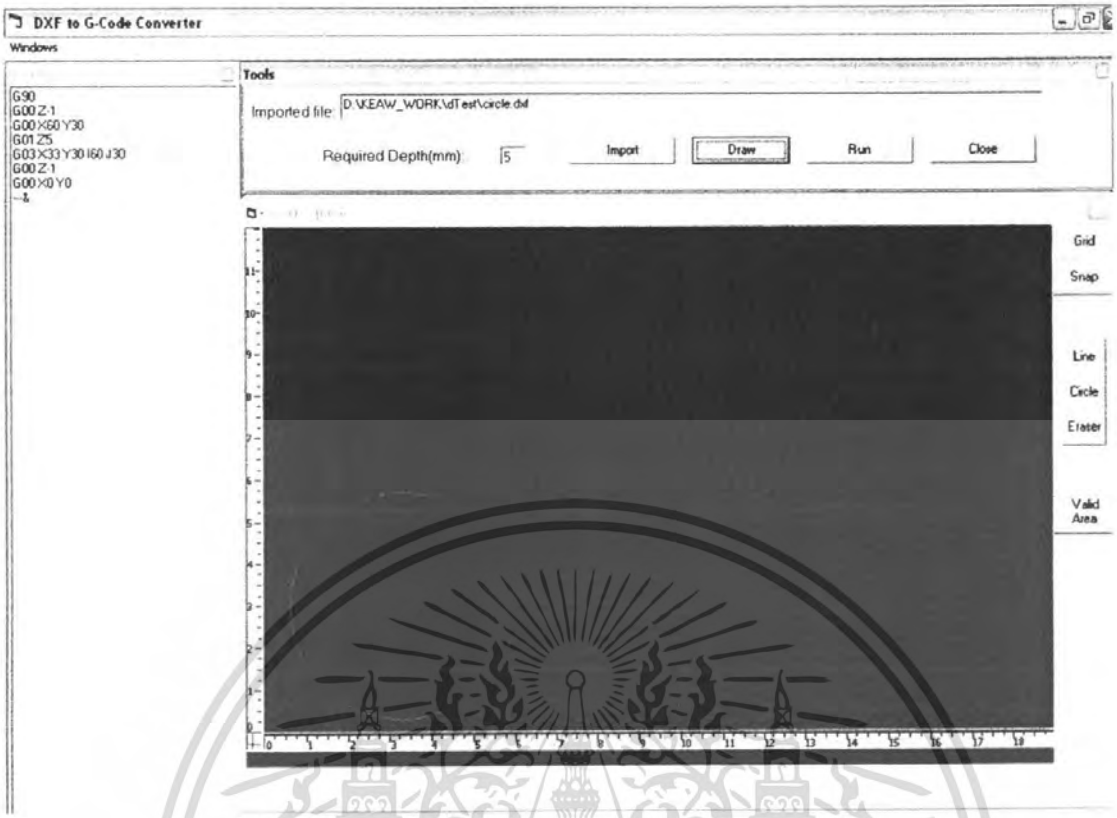
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลองกัดรูปวงกลม

ทำการทดลองนำไฟล์รูปวงกลมจากโปรแกรม AutoCAD แล้วทำการบันทึกเป็นในแบบไฟล์ DXF จากนั้นใช้โปรแกรมที่สร้างขึ้น คือ หน้าต่าง DXF to G-Code Converter นำมาแปลงเป็นไฟล์ G-CODE แล้วสั่งการเครื่องกัดให้ทำการกัดวัสดุเป็นรูปวงกลม

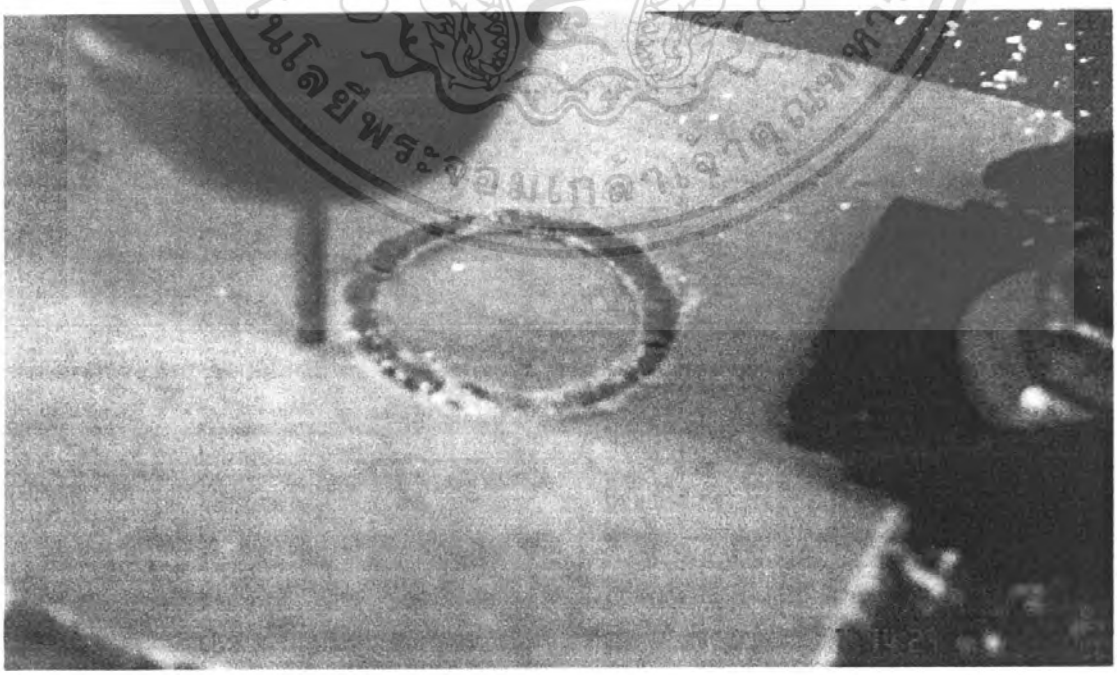
รูปที่ 4.9 แสดงรูปภาพวงกลมที่ออกแบบในโปรแกรม AutoCAD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 แสดงหน้าต่าง DXF to G-Code Converter ที่ใช้สั่งการกัดวงกลม

ผลการทดลอง



รูปที่ 4.11 แสดงรูปวงกลมบนวัสดุที่ได้จากการทำงานของเครื่องกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และบทสรุป

5.1 สรุปผลการทดลอง

จากการทำการทดลองใช้โปรแกรมส่วนควบคุมเครื่องกัดโดยตรง (MANUAL FORM) นั้น ปรากฏว่าเครื่องกัดสามารถทำตามคำสั่งที่ต้องการผ่านทางคอมพิวเตอร์ได้อย่างถูกต้องแม่นยำ และภายในระยะเวลาเหมาะสม

จากการทำการทดลองใช้โปรแกรมส่วนแปลงไฟล์ DXF เป็นไฟล์ G-CODE (DXF to G-Code Converter) เพื่อควบคุมการทำงานของเครื่องกัด ปรากฏว่าโปรแกรมสามารถควบคุมและสั่งการเครื่องกัดได้อย่างถูกต้องตามต้องการ

5.2 วิจารณ์ผลการทดลอง

จากผลการทดลองสั่งการเครื่องกัดทั้งทาง DXF to G-Code Converter และ MANUAL FORM สามารถทำงานได้ถูกต้องตามต้องการ และในระยะเวลาที่กำหนด

5.3 ปัญหาที่พบในการทำโครงงานนี้

ปัญหาที่ประสบจากการทำโครงงานนี้ในช่วงแรก คือ ขาดความรู้และประสบการณ์ในการทำงานทางด้านการใช้โปรแกรมวิชวลเบสิก และขาดความรู้ในทฤษฎีเพราะไม่เคยศึกษาด้านการควบคุมเครื่องจักรและการสร้างโปรแกรมมาก่อน ดังนั้นในการทำงานช่วงแรกจึงต้องใช้เวลาในการศึกษาค้นคว้า

ปัญหาที่ประสบจากการทำโครงงานในช่วงที่สอง คือ ความไม่แน่นอนของอุปกรณ์เนื่องจากการทำงานของมอเตอร์มีข้อบกพร่องเช่น ความเร็วรอบไม่ได้ตามต้องการ นั่นคือ แกน Z เคลื่อนที่ช้าไป

5.4 แนวทางในการพัฒนา

แนวทางในการปรับปรุงพัฒนาโครงงานนี้ในส่วนของโปรแกรมควรจะออกแบบลักษณะโปรแกรมให้ใช้งานง่ายและมีฟังก์ชันในการทำงานครบถ้วน อาจจะใช้ภาษาในการพัฒนาใดก็ได้ตามความเหมาะสมสำหรับการสร้างโปรแกรมเพื่อสามารถพัฒนาโปรแกรมได้ตอบสนองความต้องการของผู้ใช้มากที่สุด

แนวทางการพัฒนาการปรับปรุงเครื่องจักรนั้น ควรจะทำการกำหนดขอบเขตในการทำงานของเครื่องจักรอย่างชัดเจนเพื่อดำเนินการออกแบบและปรับปรุงได้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] ธาริน สิทธิธรรมชารี. ประชา พฤกษ์ประเสริฐ **สร้างโปรแกรมบน windows ด้วย Visual Basic 6.0.** พิมพ์ครั้งที่12. กรุงเทพมหานคร : ซัคเซส มีเดีย. 2548.

[2] อภิชาต ภู่พลับ. **เริ่มต้นเขียนโปรแกรมติดต่อ และควบคุมฮาร์ดแวร์ด้วย Visual Basic.** กรุงเทพมหานคร : ดีอีวี บุค. 2546.

[3] อภิชาต ภู่พลับ. **เขียนโปรแกรม Hardware Interface ด้วย VB6.** กรุงเทพมหานคร : 2548.

[4] กิตติ ภักดีวัฒนะกุล. จำลอง ครูอุตสาหะ **Visual Basic 6 ฉบับโปรแกรมเมอร์.** พิมพ์ครั้งที่13. กรุงเทพนคร : เคทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด. 2549.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรแกรม MANUAL FORM

การประกาศฟังก์ชันและตัวแปรต่างๆ

```
Private Declare Function Inp Lib "inpout32.dll" Alias "Inp32" (ByVal PortAddress As Integer)
As Integer
Private Declare Sub Out Lib "inpout32.dll" Alias "Out32" (ByVal PortAddress As Integer, ByVal
Value As Integer)
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
Public pwrite As Integer
Public pread As Integer
Public N As Integer 'H379
Public X As Integer 'JOG
Public Y As Integer 'JOG
Public Z As Integer 'JOG
Public A As Integer
Public B As Integer
Public C As Integer 'count in jog mode
Public CountX As Double
Public CountY As Double
Public CountZ As Double
Public deltaX As Double
Public deltaY As Double
Public deltaZ As Double
Public GoToPositionX As Double
Public GoToPositionY As Double
Public GoToPositionZ As Double
Public fixX As Double
Public fixY As Double
Public fixZ As Double
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Public SelfTimer As Boolean 'Only true and false

Public checkX, checkY, checkZ, temp As Integer
```

ส่วนคำสั่งที่ใช้ในส่วนปุ่ม+XของDISTANCE CONTROL BY ONE CLICK

```
Private Sub Command12_Click()
A = 0
If Option7.Value = False And Option8.Value = False And Option9.Value = False Then 'check list
Timer17.Enabled = False
MsgBox "You must choose JOG.", vbOKOnly
ElseIf Option7.Value = True Or Option8.Value = True Or Option9.Value = True Then
Timer17.Interval = HScroll1.Value
Timer17.Enabled = True
End If
X = X + A
End Sub
```

ส่วนคำสั่งที่ใช้ในส่วนปุ่ม-XของDISTANCE CONTROL BY ONE CLICK

```
Private Sub Command13_Click()
A = 0
If Option7.Value = False And Option8.Value = False And Option9.Value = False Then 'check list
Timer18.Enabled = False
MsgBox "You must choose JOG.", vbOKOnly
ElseIf Option7.Value = True Or Option8.Value = True Or Option9.Value = True Then
Timer18.Interval = HScroll1.Value
Timer18.Enabled = True
End If
X = X + A
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนคำสั่งที่ใช้ในส่วนปุ่ม+YของDISTANCE CONTROL BY ONE CLICK

```
Private Sub Command14_Click()
B = 0
If Option7.Value = False And Option8.Value = False And Option9.Value = False Then 'check list
Timer19.Enabled = False
MsgBox "You must choose JOG.", vbOKOnly
ElseIf Option7.Value = True Or Option8.Value = True Or Option9.Value = True Then
Timer19.Interval = HScroll1.Value
Timer19.Enabled = True
End If
Y = Y + B
End Sub
```

ส่วนคำสั่งใช้ในส่วนปุ่ม-YของDISTANCE CONTROL BY ONE CLICK

```
Private Sub Command15_Click()
B = 0
If Option7.Value = False And Option8.Value = False And Option9.Value = False Then 'check list
Timer20.Enabled = False
MsgBox "You must choose JOG.", vbOKOnly
ElseIf Option7.Value = True Or Option8.Value = True Or Option9.Value = True Then
Timer20.Interval = HScroll1.Value
Timer20.Enabled = True
End If
Y = Y + B
End Sub
```

ส่วนคำสั่งที่ใช้ในส่วนปุ่ม+ZของDISTANCE CONTROL BY ONE CLICK

```
Private Sub Command16_Click()
C = 0
If Option7.Value = False And Option8.Value = False And Option9.Value = False Then  'check
list
Timer21.Enabled = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MsgBox "You must choose JOG.", vbOKOnly
ElseIf Option7.Value = True Or Option8.Value = True Or Option9.Value = True Then
Timer21.Interval = HScroll1.Value
Timer21.Enabled = True
End If
Z = Z + C
End Sub
```

ส่วนคำสั่งใช้ในส่วนปุ่ม-ZของDISTANCE CONTROL BY ONE CLICK

```
Private Sub Command17_Click()
C = 0
If Option7.Value = False And Option8.Value = False And Option9.Value = False Then 'check list
Timer22.Enabled = False
MsgBox "You must choose JOG.", vbOKOnly
ElseIf Option7.Value = True Or Option8.Value = True Or Option9.Value = True Then
Timer22.Interval = HScroll1.Value
Timer22.Enabled = True
End If
Z = Z + C
End Sub
```

ส่วนคำสั่งที่ใช้กับปุ่มที่เรียกโปรแกรมส่วนAUTOMATIC MODE

```
Private Sub Command19_Click()
   frmMDI.Show
End Sub
```

ส่วนคำสั่งที่ใช้ปิดโปรแกรมโดยMENU EDITOR

```
Private Sub exit_Click() 'to close program by menu editor
Unload Me
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนคำสั่งของปุ่มGO TO 0ที่กลับไปยังตำแหน่ง(0,0,0)เดิม

```
Private Sub Command18_Click()
Text1.Text = "0"
Text2.Text = "0"
Text3.Text = "0"
Sleep (0.6)
Command7_Click
End Sub
```

ส่วนคำสั่งของปุ่มCANCELที่ยกเลิกการกลับไปยังตำแหน่ง(0,0,0)ของPOSITION0

```
Private Sub Command10_Click()
Timer4.Enabled = False
Timer6.Enabled = False
Timer8.Enabled = False
pwrite = &H378
PREADH = &H37A
pread = &H379
Out pwrite, &H0
Out PREADH, &H1 'to command pin1 send bit0
Label1.Caption = "Data from Parallel Port : "
Label2.Caption = "Data"
Label7.Caption = "POSITION STATUS"
Label14.Caption = "POSITION STATUS"
Timer1.Interval = 1
Timer2.Interval = 1
GoToPositionX = 0
GoToPositionY = 0
GoToPositionZ = 0
CountX = 0
CountY = 0
CountZ = 0
Timer9.Enabled = False 'detect timer9,10,11 on when first srcoll
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Timer10.Enabled = False
Timer11.Enabled = False
Timer13.Enabled = False
Timer14.Enabled = False
Timer15.Enabled = False
End Sub
```

ส่วนคำสั่งของปุ่มSET0 ที่ใช้กำหนดตำแหน่ง(0,0,0)

```
Private Sub Command11_Click() 'Set0
Label14.Caption = "BUTTON STATUS"
Command8.Enabled = True
CountX = 0
CountY = 0
CountZ = 0
Text1.Text = "0"
Text2.Text = "0"
Text3.Text = "0"
GoToPositionX = 0
GoToPositionY = 0
GoToPositionZ = 0
deltaX = 0
deltaY = 0
deltaZ = 0
Label10.Caption = "SET0"
Label20.Caption = "GoToPositionX = " & GoToPositionX & " deltaX = " & deltaX & " CountX
= " & CountX
Label21.Caption = "GoToPositionY = " & GoToPositionY & " deltaY = " & deltaY & " CountY
= " & CountY
Label22.Caption = "GoToPositionZ = " & GoToPositionZ & " deltaZ = " & deltaZ & " CountZ =
" & CountZ
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนคำสั่งการคลิกเมาส์ล่องของปุ่ม+XของMANUAL BY MOUSE DOWN

```
Private Sub Command1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
Label7.Caption = "TEST ON +X"
Timer3.Interval = HScroll1.Value ' Set Speed
Timer3.Enabled = True  ' Timer start
Shape1.BackColor = vbGreen
End Sub
```

ส่วนคำสั่งการปล่อยเมาส์ขึ้นของปุ่ม+XของMANUAL BY MOUSE DOWN

```
Private Sub Command1_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As
Single)
Label7.Caption = "BUTTON STATUS"
Timer3.Enabled = False
Out pwrite, &H0
Shape1.BackColor = vbRed
End Sub
```

ส่วนคำสั่งการคลิกเมาส์ล่องของปุ่ม-XของMANUAL BY MOUSE DOWN

```
Private Sub Command2_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
Label7.Caption = "TEST ON -X"
Timer4.Interval = HScroll1.Value ' Set Speed
Timer4.Enabled = True  ' Timer start
Shape1.BackColor = vbGreen
End Sub
```

ส่วนคำสั่งการปล่อยเมาส์ขึ้นของปุ่ม-XของMANUAL BY MOUSE DOWN

```
Private Sub Command2_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As
Single)
Label7.Caption = "BUTTON STATUS"
Timer4.Enabled = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Out pwrite, &H0
Shape1.BackColor = vbRed
End Sub
```

ส่วนคำสั่งการคลิกเมาส์ลงของปุ่ม+YของMANUAL BY MOUSE DOWN

```
Private Sub Command3_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
Label7.Caption = "TEST ON +Y"
Timer5.Interval = HScroll1.Value ' Set Speed
Timer5.Enabled = True  ' Timer start
Shape2.BackColor = vbGreen
End Sub
```

ส่วนคำสั่งปล่อยเมาส์ขึ้นของปุ่ม+YของMANUAL BY MOUSE DOWN

```
Private Sub Command3_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As
Single)
Label7.Caption = "BUTTON STATUS"
Timer5.Enabled = False
Out pwrite, &H0
Shape2.BackColor = vbRed
End Sub
```

ส่วนคำสั่งการคลิกเมาส์ลงของปุ่ม-YของMANUAL BY MOUSE DOWN

```
Private Sub Command4_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
Label7.Caption = "TEST ON -Y"
Timer6.Interval = HScroll1.Value ' Set Speed
Timer6.Enabled = True  ' Timer start
Shape2.BackColor = vbGreen
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนคำสั่งการปล่อยเมาส์ขึ้นของปุ่ม-YของMANUAL BY MOUSE DOWN

```
Private Sub Command4_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As
Single)
Label7.Caption = "BUTTON STATUS"
Timer6.Enabled = False
Out pwrite, &H0
Shape2.BackColor = vbRed
End Sub
```

ส่วนคำสั่งการคลิกเมาส์ลงของปุ่ม+ZของMANUAL BY MOUSE DOWN

```
Private Sub Command5_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
Label7.Caption = "TEST ON +Z"
Timer7.Interval = HScroll1.Value ' Set Speed
Timer7.Enabled = True  ' Timer start
Shape3.BackColor = vbGreen
End Sub
```

ส่วนคำสั่งการปล่อยเมาส์ขึ้นของปุ่ม+ZของMANUAL BY MOUSE DOWN

```
Private Sub Command5_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As
Single)
Label7.Caption = "BUTTON STATUS"
Timer7.Enabled = False
Out pwrite, &H0
Shape3.BackColor = vbRed
End Sub
```

ส่วนคำสั่งการคลิกเมาส์ลงของปุ่ม-ZของMANUAL BY MOUSE DOWN

```
Private Sub Command6_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
Label7.Caption = "TEST ON -Z"
Timer8.Interval = HScroll1.Value ' Set Speed
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Timer8.Enabled = True  ' Timer start
Shape3.BackColor = vbGreen
End Sub
```

ส่วนคำสั่งการปล่อยเมาส์ขึ้นของปุ่ม-ZของMANUAL BY MOUSE DOWN

```
Private Sub Command6_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As
Single)
Label7.Caption = "BUTTON STATUS"
Timer8.Enabled = False
Out pwrite, &H0
Shape3.BackColor = vbRed
End Sub
```

ฟังก์ชันCHECKRUN

```
Function checkRun()
   If checkX = 1 And checkY = 1 Then
      checkRun = True
   End If
   If frmTools.chkStop = 0 Then
      If checkX = 1 And checkY = 1 And checkZ = 1 Then
      frmTools.Command4_Click 'in order to input the next line
      End If
   End If
   temp = temp + 1
End Function
```

ส่วนคำสั่งที่จะปรากฏขึ้นตอนเริ่มแรกของการเปิดโปรแกรม

```
Private Sub Form_Load()
checkX = 1
checkY = 1
checkZ = 1
temp = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Me.Left = 0
    Me.Top = 0
    'Me.Width = 10150
    'Me.Height = 10210

pwrite = &H378
pread = &H379
Out pwrite, &H0
Label7.Caption = "POSITION STATUS"
Label14.Caption = "POSITION STATUS"
Label10.Caption = "X: ST"
Label32.Caption = "Y: ST"
Label33.Caption = "Z: ST"
Label20.Caption = "GoToPositionX"
Label21.Caption = "GoToPositionY"
Label22.Caption = "GoToPositionZ"
Timer1.Interval = 1
Timer23.Interval = 1
Timer2.Interval = 1
HScroll1.Min = 10
HScroll1.Max = 1
HScroll1.Value = 5
GoToPositionX = 0
GoToPositionY = 0
GoToPositionZ = 0
CountX = 0
CountY = 0
CountZ = 0
Timer9.Enabled = False 'detect timer9,10,11 on when first srcoll
Timer10.Enabled = False
Timer11.Enabled = False
Timer13.Enabled = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Timer14.Enabled = False
Timer15.Enabled = False
'MsgBox "PLEASE SET THE VALUES IN SETTING BOX!!!", vbOKOnly
End Sub
```

ส่วนคำสั่งFORM TERMINATE

```
Private Sub Form_Terminate()
   justRun = 0
End Sub
```

ส่วนคำสั่งFORM UNLOAD

```
Private Sub Form_Unload(Cancel As Integer)
   justRun = 0
End Sub
```

ส่วนคำสั่งของScrollที่ใช้เลื่อนปรับความเร็ว

```
Private Sub HScroll1_Change()
Label9.Caption = HScroll1.Value
End Sub
```

ส่วนคำสั่งTIMER1ที่ใช้ในการรับค่าจากภายนอกผ่านพอร์ตขนาน

```
Private Sub Timer1_Timer()
N = Inp(pread)
Label2.Caption = N
Select Case N
Case 126, 120 'No contact on limit switch
lblStatus.Caption = "EDM CONTROLLER PROGRAM"
Command1.Enabled = True
Command2.Enabled = True
Command3.Enabled = True
Command4.Enabled = True
Command6.Enabled = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Command12.Enabled = True
Command13.Enabled = True
Command14.Enabled = True
Command15.Enabled = True
Command17.Enabled = True
Case 118, 112 'contact on pin15 +x
Out pwrite, &H40 'cut off electric current drill
Command2.Enabled = False
Command13.Enabled = False
Timer4.Enabled = False
Timer13.Enabled = False
Timer18.Enabled = False
Case 110, 104 'contact on pin13 -x
Out pwrite, &H40 'cut off electric current drill
Command1.Enabled = False
Command12.Enabled = False
Timer3.Enabled = False
Timer9.Enabled = False
Timer17.Enabled = False
Case 94, 88 'contact on pin12 +y
Out pwrite, &H40 'cut off electric current drill
Command3.Enabled = False
Command14.Enabled = False
Timer5.Enabled = False
Timer10.Enabled = False
Timer19.Enabled = False
Case 254, 248 'contact on pin11 -y
Out pwrite, &H40 'cut off electric current drill
Command4.Enabled = False
Command15.Enabled = False
Timer6.Enabled = False
Timer14.Enabled = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Timer20.Enabled = False
Case 238, 232 'contact pin13,11 -x -y
Out pwrite, &H40 'cut off electric current drill
Command1.Enabled = False
Command12.Enabled = False
Timer3.Enabled = False
Timer9.Enabled = False
Timer17.Enabled = False
Command4.Enabled = False
Command15.Enabled = False
Timer6.Enabled = False
Timer14.Enabled = False
Timer20.Enabled = False
Case 86, 80 'contact pin15,12 +x +y
Out pwrite, &H40 'cut off electric current drill
Command2.Enabled = False
Command13.Enabled = False
Timer4.Enabled = False
Timer13.Enabled = False
Timer18.Enabled = False
Command3.Enabled = False
Command14.Enabled = False
Timer5.Enabled = False
Timer10.Enabled = False
Timer19.Enabled = False
Case 78, 72 'contact pin13,12 -x +y
Out pwrite, &H40 'cut off electric current drill
Command1.Enabled = False
Command12.Enabled = False
Timer3.Enabled = False
Timer9.Enabled = False
Timer17.Enabled = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Command3.Enabled = False
Command14.Enabled = False
Timer5.Enabled = False
Timer10.Enabled = False
Timer19.Enabled = False
Case 246, 240 'contact pin15,11 +x -y
Out pwrite, &H40 'cut off electric current drill
Command2.Enabled = False
Command13.Enabled = False
Timer4.Enabled = False
Timer13.Enabled = False
Timer18.Enabled = False
Command4.Enabled = False
Command15.Enabled = False
Timer6.Enabled = False
Timer14.Enabled = False
Timer20.Enabled = False
'contact on checkpoint
End Select
End Sub
```

ส่วนคำสั่งTIMER +YของMANUAL DATA INPUT

```
Private Sub Timer10_Timer()
   checkY = 0
   IncreaseY
End Sub
Function IncreaseY()
If GoToPositionY > CountY Then
Shape2.BackColor = vbGreen
   Label32.Caption = "Y: +ON"
    For j = 1 To 2 * Mid(CInt(GoToPositionY - fixY), 1, 1)
   If SelfTimer Then ' Check SelfTimer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Out pwrite, &HC
        CountY = CountY + (1 / Text5.Text) ' 1/x is number of pulse per 1mm
        Label15.Caption = "1"
        Label21.Caption = "GoToPositionY = " & GoToPositionY & " deltaY = " & deltaY & "
CountY = " & CountY
        Sleep (0.6)
    Else
        Out pwrite, &H8
        Label15.Caption = "0"
    End If
    SelfTimer = Not SelfTimer ' toggle SelfTimer
    Next j
Else
Label32.Caption = "Y: OFF"
Label21.Caption = "GoToPositionY = " & GoToPositionY & " deltaY = " & deltaY & " CountY
= " & CountY
Timer10.Enabled = False
Shape2.BackColor = vbRed
checkY = 1
checkRun
End If
End Function
```

ส่วนคำสั่งTIMER +XของMANUAL DATA INPUT

```
Private Sub Timer9_Timer()
    checkX = 0
    IncreaseX
End Sub
Function IncreaseX()
If GoToPositionX > CountX Then
Shape1.BackColor = vbGreen
Label10.Caption = "X: +ON"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   For i = 1 To 2 * Mid(CInt(GoToPositionX - fixX), 1, 1)
   If SelfTimer Then ' Check SelfTimer
      Out pwrite, &H3
      CountX = CountX + (1 / Text4.Text) ' 1/x is number of pulse per 1mm
      Label13.Caption = "1"
      Label20.Caption = "GoToPositionX = " & (GoToPositionX) & " deltaX = " & deltaX & "
CountX = " & (CountX)
      Sleep (0.6)
   Else
      Out pwrite, &H2
      Label13.Caption = "0"
   End If
   SelfTimer = Not SelfTimer ' toggle SelfTimer
   Next i
Else
Label10.Caption = "X: OFF"
Label20.Caption = "GoToPositionX = " & (GoToPositionX) & " deltaX = " & deltaX & "
CountX = " & (CountX)
Timer9.Enabled = False
Shape1.BackColor = vbRed
checkX = 1
checkRun
End If
End Function
```

ส่วนคำสั่งTIMER -Xของ MANUAL DATA INPUT

```
Private Sub Timer13_Timer()
   checkX = 0
   DecreaseX
End Sub
Function DecreaseX()
If GoToPositionX < CountX Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Shape1.BackColor = vbGreen
   Label10.Caption = "X: -ON"
   For i = 1 To 2 * -Mid(CInt(GoToPositionX - fixX), 1, 2)
   If SelfTimer Then ' Check SelfTimer
      Out pwrite, &H1
      CountX = CountX - (1 / Val(Text4.Text)) ' 1/x is number of pulse per 1mm
      Label13.Caption = "1"
      Label20.Caption = "GoToPositionX = " & (GoToPositionX) & " deltaX = " & deltaX & "
CountX = " & (CountX)
      Sleep (0.6)
   Else
      Out pwrite, &H0
      Label13.Caption = "0"
   End If
   SelfTimer = Not SelfTimer ' toggle SelfTimer
   Next i
Else
Label10.Caption = "X: OFF"
Label20.Caption = "GoToPositionX = " & (GoToPositionX) & " deltaX = " & deltaX & "
CountX = " & (CountX)
Timer13.Enabled = False
Shape1.BackColor = vbRed
checkX = 1
checkRun
End If
End Function
```

ส่วนคำสั่งTIMER -YของMANUAL DATA INPUT

```
Private Sub Timer14_Timer()
   checkY = 0
   DecreaseY
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Function DecreaseY()
If GoToPositionY < CountY Then
Shape2.BackColor = vbGreen
  Label32.Caption = "Y: -ON"
  For j = 1 To 2 * -Mid(CInt(GoToPositionY - fixY), 1, 2)
   If SelfTimer Then ' Check SelfTimer
      Out pwrite, &H4
      CountY = CountY - (1 / Text5.Text) ' 1/x is number of pulse per 1mm
      Label15.Caption = "1"
      Label21.Caption = "GoToPositionY = " & (GoToPositionY) & " deltaY = " & deltaY & "
CountY = " & (CountY)
      Sleep (0.6)
   Else
      Out pwrite, &H0
      Label15.Caption = "0"
   End If
   SelfTimer = Not SelfTimer ' toggle SelfTimer
   Next j
Else
Label32.Caption = "Y: OFF"
 Label21.Caption = "GoToPositionY = " & (GoToPositionY) & " deltaY = " & deltaY & "
CountY = " & (CountY)
Timer14.Enabled = False
Shape2.BackColor = vbRed
checkY = 1
checkRun
End If
End Function
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนคำสั่งTIMER +ZของMANUAL DATA INPUT

```
Private Sub Timer11_Timer()
   checkZ = 0
   IncreaseZ
End Sub
Function IncreaseZ()
If GoToPositionZ > CountZ Then
Shape3.BackColor = vbGreen
Label33.Caption = "Z: +ON"
   If SelfTimer Then ' Check SelfTimer
      Out pwrite, &H20
      CountZ = CountZ + (1 / 2 * Text6.Text) ' 1/x is number of pulse per 1mm
      Label16.Caption = "1"
      Label22.Caption = "GoToPositionZ = " & GoToPositionZ & " deltaZ = " & deltaZ & "
CountZ = " & CountZ
   Else
      Out pwrite, &H30
      Label16.Caption = "0"
   End If
   SelfTimer = Not SelfTimer ' toggle SelfTimer
Else
Label33.Caption = "Z: OFF"
Label22.Caption = "GoToPositionZ = " & GoToPositionZ & " deltaZ = " & deltaZ & " CountZ =
" & CountZ
Timer11.Enabled = False
Shape3.BackColor = vbRed
checkZ = 1
Debug.Print checkZ
checkRun
End If
End Function
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนคำสั่งTIMER -ZของMANUAL DATA INPUT

```
Private Sub Timer15_Timer()
   checkZ = 0
   DecreaseZ
End Sub
Function DecreaseZ()
If GoToPositionZ < CountZ Then
Shape3.BackColor = vbGreen
 Label33.Caption = "Z: -ON"
   If SelfTimer Then ' Check SelfTimer
      Out pwrite, &H10
      CountZ = CountZ - (1 / 2 * Text6.Text) ' 1/x is number of pulse per 1mm
      Label16.Caption = "1"
      Label22.Caption = "GoToPositionZ = " & (GoToPositionZ) & " deltaZ = " & deltaZ & "And
CountZ = " & (CountZ)
   Else
      Out pwrite, &H0
      Label16.Caption = "0"
   End If
   SelfTimer = Not SelfTimer ' toggle SelfTimer
Else
Label33.Caption = "Z: OFF"
Label22.Caption = "GoToPositionZ = " & (GoToPositionZ) & " deltaZ = " & deltaZ & "And
CountZ = " & (CountZ)
Timer15.Enabled = False
Shape3.BackColor = vbRed
checkZ = 1
checkRun
End If
End Function
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนคำสั่งของปุ่มSTARTของMANUAL DATA INPUT

```
Public Sub Command7_Click() 'START check in any conditions
DoEvents
Debug.Print "cmd7 was clicked"
   fixX = CountX
   fixY = CountY
   fixZ = CountZ
If Text1.Text = "" Then
   Text1.Text = "0"
   Timer9.Enabled = False
   Timer13.Enabled = False
   Timer10.Enabled = False
   Timer14.Enabled = False
   Timer11.Enabled = False
   Timer15.Enabled = False
ElseIf IsNumeric(Text1.Text) Then
   SelfTimer = False
   deltaX = Text1.Text - GoToPositionX
   GoToPositionX = Text1.Text
   If deltaX = 0 Then
      Timer9.Enabled = False
     Timer13.Enabled = False
   ElseIf deltaX > 0 Then
   checkX = 0
     Timer9.Interval = HScroll1.Value
     Timer9.Enabled = True
     Timer13.Enabled = False
   ElseIf deltaX < 0 Then
   checkX = 0
     Timer13.Interval = HScroll1.Value
     Timer13.Enabled = True
     Timer9.Enabled = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    End If
Else
MsgBox "Please insert only numeric number in X box.", vbOKOnly
Timer9.Enabled = False
Timer13.Enabled = False
Timer10.Enabled = False
Timer14.Enabled = False
Timer11.Enabled = False
Timer15.Enabled = False
End If

If Text2.Text = "" Then
Text2.Text = "0"
Timer9.Enabled = False
Timer13.Enabled = False
Timer10.Enabled = False
Timer14.Enabled = False
Timer11.Enabled = False
Timer15.Enabled = False
ElseIf IsNumeric(Text2.Text) Then
SelfTimer = False
deltaY = Text2.Text - GoToPositionY
GoToPositionY = Text2.Text
    If deltaY = 0 Then
    Timer10.Enabled = False
    Timer14.Enabled = False
    ElseIf deltaY > 0 Then
    checkY = 0
    Timer10.Interval = HScroll1.Value
    Timer10.Enabled = True
    Timer14.Enabled = False
    ElseIf deltaY < 0 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    checkY = 0
    Timer14.Interval = HScroll1.Value
    Timer14.Enabled = True
    Timer10.Enabled = False
    End If
Else
MsgBox "Please insert only numeric number in Y box.", vbOKOnly
Timer9.Enabled = False
Timer13.Enabled = False
Timer10.Enabled = False
Timer14.Enabled = False
Timer11.Enabled = False
Timer15.Enabled = False
End If

If Text3.Text = "" Then
Text3.Text = "0"
Timer9.Enabled = False
Timer13.Enabled = False
Timer10.Enabled = False
Timer14.Enabled = False
Timer11.Enabled = False
Timer15.Enabled = False
ElseIf IsNumeric(Text3.Text) Then
SelfTimer = False
deltaZ = Text3.Text - GoToPositionZ
GoToPositionZ = Text3.Text
    If deltaZ = 0 Then
    Timer11.Enabled = False
    Timer15.Enabled = False
    ElseIf deltaZ > 0 Then
    checkZ = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Timer11.Interval = HScroll1.Value
    Timer11.Enabled = True
    Timer15.Enabled = False
    ElseIf deltaZ < 0 Then
    checkZ = 0
    Timer15.Interval = HScroll1.Value
    Timer15.Enabled = True
    Timer11.Enabled = False
    End If
Else
MsgBox "Please insert only numeric number in Z box.", vbOKOnly
Timer9.Enabled = False
Timer13.Enabled = False
Timer10.Enabled = False
Timer14.Enabled = False
Timer11.Enabled = False
Timer15.Enabled = False
End If
End Sub
```

ส่วนคำสั่งของปุ่มSTOPของMANUAL DATA INPUT

```
Private Sub Command8_Click() 'STOP
Label14.Caption = "BUTTON STATUS"
Timer9.Enabled = False
Timer10.Enabled = False
Timer11.Enabled = False
Timer13.Enabled = False
Timer14.Enabled = False
Timer15.Enabled = False
Out pwrite, &H0
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนคำสั่งของปุ่มRESET COUNTของMANUAL DATA INPUT

```
Private Sub Command9_Click()
Label14.Caption = "BUTTON STATUS"
Command8.Enabled = True
CountX = 0
CountY = 0
CountZ = 0
Text1.Text = "0"
Text2.Text = "0"
Text3.Text = "0"
GoToPositionX = 0
GoToPositionY = 0
GoToPositionZ = 0
deltaX = 0
deltaY = 0
deltaZ = 0
Label10.Caption = "RESET COUNT"
Label20.Caption = "GoToPositionX = " & GoToPositionX & " deltaX = " & deltaX & " CountX
= " & CountX
Label21.Caption = "GoToPositionY = " & GoToPositionY & " deltaY = " & deltaY & " CountY
= " & CountY
Label22.Caption = "GoToPositionZ = " & GoToPositionZ & " deltaZ = " & deltaZ & " CountZ =
" & CountZ
End Sub
```

ส่วนคำสั่งTIMER +XของDISTANCE CONTROL BY ONE CLICK

```
Private Sub Timer17_Timer()
   Label10.Caption = "MACHINE ON WORK"
   Shape1.BackColor = vbGreen
   If SelfTimer Then ' Check SelfTimer
      Out pwrite, &H3
      Label19.Caption = "1"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Label13.Caption = "1"
    Label20.Caption = "GoToPositionX = " & (GoToPositionX) & " deltaX = " & deltaX & "
CountX = " & CountX
    CountX = CountX + (1 / Val(Text4.Text)) ' 1/x is number of pulse per 1mm
    GoToPositionX = GoToPositionX + (1 / Val(Text4.Text))
    A = A + 1 'count
    Label29.Caption = "WORK"
  Else
    Out pwrite, &H2
    Label19.Caption = "0"
    Label13.Caption = "0"
  End If
SelfTimer = Not SelfTimer ' toggle SelfTimer

 Label18.Caption = A
  If Not (X <> A) Then 'X=A
    Timer17.Enabled = False
     Out pwrite, &H0
  Label29.Caption = "> Stop <"
  Label20.Caption = "GoToPositionX = " & (GoToPositionX) & " deltaX = " & deltaX & "
CountX = " & CountX
  Label10.Caption = "X: OFF"
  Shape1.BackColor = vbRed
  End If
End Sub
```

ส่วนคำสั่งTIMER -XของDISTANCE CONTROL BY ONE CLICK

```
Private Sub Timer18_Timer()
  Label10.Caption = "MACHINE ON WORK"
  Shape1.BackColor = vbGreen
  If SelfTimer Then ' Check SelfTimer
    Out pwrite, &H1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Label19.Caption = "1"
        Label13.Caption = "1"
        Label20.Caption = "GoToPositionX = " & (GoToPositionX) & " deltaX = " & deltaX & "
CountX = " & CountX
        CountX = CountX - (1 / Val(Text4.Text)) ' 1/x is number of pulse per 1mm
        GoToPositionX = GoToPositionX - (1 / Val(Text4.Text))
        A = A + 1 'count
        Label29.Caption = "WORK"
    Else
        Out pwrite, &H0
        Label19.Caption = "0"
        Label13.Caption = "0"
    End If
SelfTimer = Not SelfTimer ' toggle SelfTimer

Label18.Caption = A
    If Not (X <> A) Then
        Timer18.Enabled = False
        Out pwrite, &H0
    Label29.Caption = "> Stop <"
    Label20.Caption = "GoToPositionX = " & (GoToPositionX) & " deltaX = " & deltaX & "
CountX = " & CountX
    Label10.Caption = "X: OFF"
    Shape1.BackColor = vbRed
    End If
End Sub
```

ส่วนคำสั่งTIMER +YของDISTANCE CONTROL BY ONE CLICK

```
Private Sub Timer19_Timer()
    Label32.Caption = "MACHINE ON WORK"
    Shape2.BackColor = vbGreen
    If SelfTimer Then ' Check SelfTimer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Out pwrite, &HC
        Label19.Caption = "1"
        Label15.Caption = "1"
        Label21.Caption = "GoToPositionY = " & (GoToPositionY) & " deltaY = " & deltaY & "
CountY = " & CountY
        CountY = CountY + (1 / Text5.Text) ' 1/x is number of pulse per 1mm
        GoToPositionY = GoToPositionY + (1 / Val(Text4.Text))
        B = B + 1 'count
        Label29.Caption = "WORK"
    Else
        Out pwrite, &H8
        Label19.Caption = "0"
        Label15.Caption = "0"
    End If
SelfTimer = Not SelfTimer ' toggle SelfTimer

Label18.Caption = B
    If Not (Y <> B) Then
        Timer19.Enabled = False
        Out pwrite, &H0
    Label29.Caption = "> Stop <"
    Label21.Caption = "GoToPositionY = " & (GoToPositionY) & " deltaY = " & deltaY & "
CountY = " & CountY
    Label32.Caption = "Y: OFF"
    Shape2.BackColor = vbRed
    End If
End Sub
```

ส่วนคำสั่งTIMER -Yของ DISTANCE CONTROL BY ONE CLICK

```
Private Sub Timer20_Timer()
    Label32.Caption = "MACHINE ON WORK"
    Shape2.BackColor = vbGreen
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  If SelfTimer Then ' Check SelfTimer
      Out pwrite, &H4
      Label19.Caption = "1"
      Label15.Caption = "1"
      Label21.Caption = "GoToPositionY = " & (GoToPositionY) & " deltaY = " & deltaY & "
CountY = " & CountY
      CountY = CountY - (1 / Text5.Text) ' 1/x is number of pulse per 1mm
      GoToPositionY = GoToPositionY - Val((1 / Text4.Text))
      B = B + 1 'count
      Label29.Caption = "WORK"
    Else
      Out pwrite, &H0
      Label19.Caption = "0"
      Label15.Caption = "0"
    End If
SelfTimer = Not SelfTimer ' toggle SelfTimer

Label18.Caption = B
    If Not (Y <> B) Then
      Timer20.Enabled = False
      Out pwrite, &H0
    Label29.Caption = "> Stop <"
    Label21.Caption = "GoToPositionY = " & (GoToPositionY) & " deltaY = " & deltaY & "
CountY = " & CountY
    Label32.Caption = "Y: OFF"
    Shape2.BackColor = vbRed
    End If
End Sub
```

ส่วนคำสั่งTIMER +ZของDISTANCE CONTROL BY ONE CLICK

```
Private Sub Timer21_Timer()
    Label33.Caption = "MACHINE ON WORK"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Shape3.BackColor = vbGreen
   If SelfTimer Then ' Check SelfTimer
      Out pwrite, &H30
      Label19.Caption = "1"
      Label16.Caption = "1"
      Label22.Caption = "GoToPositionZ = " & (GoToPositionZ) & " deltaZ = " & deltaZ & "
CountZ = " & CountZ
      CountZ = CountZ + (1 / 2 * Text6.Text) ' 1/x is number of pulse per 1mm
      GoToPositionZ = GoToPositionZ + (1 / Val(Text4.Text))
      C = C + 1 'count
      Label29.Caption = "WORK"
   Else
      Out pwrite, &H20
      Label19.Caption = "0"
      Label16.Caption = "0"
   End If
SelfTimer = Not SelfTimer ' toggle SelfTimer

Label18.Caption = C
   If Not (Z <> C) Then
      Timer21.Enabled = False
      Out pwrite, &H0
   Label29.Caption = "> Stop <"
   Label22.Caption = "GoToPositionZ = " & (GoToPositionZ) & " deltaZ = " & deltaZ & "
CountZ = " & CountZ
   Label33.Caption = "Z: OFF"
   Shape3.BackColor = vbRed
   End If
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนคำสั่งTIMER -ZของDISTANCE CONTROL BY ONE CLICK

```
Private Sub Timer22_Timer()
   Label33.Caption = "MACHINE ON WORK"
   Shape3.BackColor = vbGreen
  If SelfTimer Then ' Check SelfTimer
      Out pwrite, &H10
      Label19.Caption = "1"
      Label16.Caption = "1"
      Label22.Caption = "GoToPositionZ = " & (GoToPositionZ) & " deltaZ = " & deltaZ & "
CountZ = " & CountZ
      CountZ = CountZ - (1 / 2 * Text6.Text) ' 1/x is number of pulse per 1mm
      GoToPositionZ = GoToPositionZ - (1 / Val(Text4.Text))
      C = C + 1 'count
      Label29.Caption = "WORK"
   Else
      Out pwrite, &H0
      Label19.Caption = "0"
      Label16.Caption = "0"
   End If
SelfTimer = Not SelfTimer ' toggle SelfTimer

Label18.Caption = C
   If Not (Z <> C) Then
      Timer22.Enabled = False
      Out pwrite, &H0
   Label29.Caption = "> Stop <"
   Label22.Caption = "GoToPositionZ = " & (GoToPositionZ) & " deltaZ = " & deltaZ & "
CountZ = " & CountZ
   Label33.Caption = "Z: OFF"
   Shape3.BackColor = vbRed
   End If
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนคำสั่งTIMER ที่ใช้กำหนดระยะของDISTANCE CONTROL BY ONE CLICK

```
Private Sub Timer23_Timer() 'set pulse value
If Option7.Value = True And IsNumeric(Text4.Text) Then
X = Text4.Text * 10
ElseIf Option8.Value = True And IsNumeric(Text4.Text) Then
X = Text4.Text
ElseIf Option9.Value = True And IsNumeric(Text4.Text) Then
X = Text4.Text / 10
End If

If Option7.Value = True And IsNumeric(Text5.Text) Then
Y = Text5.Text * 10
ElseIf Option8.Value = True And IsNumeric(Text5.Text) Then
Y = Text5.Text
ElseIf Option9.Value = True And IsNumeric(Text5.Text) Then
Y = Text5.Text / 10
End If

If Option7.Value = True And IsNumeric(Text6.Text) Then
Z = Text6.Text * 10
ElseIf Option8.Value = True And IsNumeric(Text6.Text) Then
Z = Text6.Text
ElseIf Option9.Value = True And IsNumeric(Text6.Text) Then
Z = Text6.Text / 10
End If
End Sub
```

ส่วนคำสั่งTIMER +XของMANUAL BY MOUSE DOWN

```
Private Sub Timer3_Timer()
  If SelfTimer Then ' Check SelfTimer
    Out pwrite, &H3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Label7.Caption = "1"
    Else
        Out pwrite, &H2
        Label7.Caption = "0"
    End If
SelfTimer = Not SelfTimer ' toggle SelfTimer
End Sub
```

ส่วนคำสั่งTIMER -Xของ MANUAL BY MOUSE DOWN

```
Private Sub Timer4_Timer()
    If SelfTimer Then ' Check SelfTimer
        Out pwrite, &H1
        Label7.Caption = "1"
    Else
        Out pwrite, &H0
        Label7.Caption = "0"
    End If
SelfTimer = Not SelfTimer ' toggle SelfTimer
End Sub
```

ส่วนคำสั่งTIMER +Yของ MANUAL BY MOUSE DOWN

```
Private Sub Timer5_Timer()
    If SelfTimer Then ' Check SelfTimer
        Out pwrite, &HC
        Label7.Caption = "1"
    Else
        Out pwrite, &H8
        Label7.Caption = "0"
    End If
SelfTimer = Not SelfTimer ' toggle SelfTimer
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนคำสั่งTIMER -YของMANUAL BY MOUSE DOWN

```
Private Sub Timer6_Timer()
 If SelfTimer Then ' Check SelfTimer
    Out pwrite, &H4
    Label7.Caption = "1"
  Else
    Out pwrite, &H0
    Label7.Caption = "0"
  End If
SelfTimer = Not SelfTimer ' toggle SelfTimer
End Sub
```

ส่วนคำสั่งTIMER +ZของMANUAL BY MOUSE DOWN

```
Private Sub Timer7_Timer()
  If SelfTimer Then ' Check SelfTimer
    Out pwrite, &H30
    Label7.Caption = "1"
  Else
    Out pwrite, &H20
    Label7.Caption = "0"
  End If
SelfTimer = Not SelfTimer ' toggle SelfTimer
End Sub
```

ส่วนคำสั่งTIMER -ZของMANUAL BY MOUSE DOWN

```
Private Sub Timer8_Timer()
  If SelfTimer Then ' Check SelfTimer
    Out pwrite, &H10
    Label7.Caption = "1"
  Else
    Out pwrite, &H0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Label7.Caption = "0"
  End If
SelfTimer = Not SelfTimer ' toggle SelfTimer
End Sub
```

ส่วนคำสั่งTIMER ของเวลาปัจจุบัน

```
Private Sub Timer2_Timer() ' time
Label6.Caption = Now
End Sub
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

โปรแกรม DXF to G-Code Converter

โปรแกรมส่วนของ frmMDI (หน้าต่างแม่ของส่วนการแปลงไฟล์รูปเป็นจีโค้ด)

ในส่วนของฟอร์มนี้เป็นการกำหนดค่าพื้นฐานต่างๆของหน้าต่างแม่ และสั่งให้แสดงฟอร์มลูกต่างๆเมื่อมีการเปิดโปรแกรม ดังนี้

- frmTools คือหน้าต่าง Tools
- frmGCD คือหน้าต่าง G-CODE
- frmDrawing คือหน้าต่าง Work Space

```
Private Sub MDIForm_Load()
   Me.Left = 25
   Me.Top = 25
   Me.Width = 15350
   Me.Height = 11000
   frmTools.Show
   frmGCD.Show
   frmDrawing.Show
End Sub

Private Sub SmnuGCD_Click()
   frmGCD.Show
End Sub

Private Sub SmnuMController_Click()
   frmMANUAL.Show
End Sub

Private Sub SmnuTools_Click()
   frmTools.Show
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub

Private Sub SmnuWrkSpace_Click()
  frmDrawing.Show
End Sub
```

โปรแกรมส่วนของ frmTools (หน้าต่างรวบรวมปุ่มการทำงานต่างๆ)

ในส่วนของฟอร์มนี้จะมีปุ่มต่างๆ ไว้ทำหน้าที่ต่างๆกันดังนี้

- ปุ่ม Import ใช้เลือกที่อยู่ไฟล์รูปภาพ (*.dxf) ที่ต้องการแปลงไฟล์เป็นจีโค้ด ทั้งนี้นอกจากการแปลงเป็นจีโค้ดแล้ว ยังสามารถแสดงรูปภาพของไฟล์บนหน้าต่าง Work Space ได้อีกด้วย
- ปุ่ม Draw หลังกดปุ่มนี้ จะมีการแสดงผลของรูปเมื่อมีการแก้ไขคำสั่งจีโค้ดที่หน้าต่าง G-CODE
- ปุ่ม Run ใช้สั่งให้เครื่องกัดทำการกัดวัสดุดังรูปที่ปรากฏที่หน้าต่าง Work Space
- ปุ่ม Close ใช้ปิดหน้าต่าง DXF to G-Code Converter

การประกาศตัวแปร

```
Public iFileNum, iFileNum2, check, checkAcDb As Integer
Public DataVar, DataVar2, txtExport As String
'Public CheckX, CheckY, CheckZ As Integer
Public CHK, chkStop, n As Integer
Public X22, Y22 As Single
Dim X(3), Y(3) As Single
Public X1, X2, X3, Y1, Y2, Y3 As Single
Dim I1, I2, I3, J1, J2, J3, K1, K2, K3 As Single
Public O As String
Public G, iFileNum3, num, j, xx, yy, zz, ii, jj, justRun, GCondition As Integer
Dim SPACE(5) As Integer
Dim a1, a2, a3, a4, a5, a6, R, PP1, PP2 As Double
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
Public chkRunclick As Boolean
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim myArray(100) As String
Dim PicArrayX(100), PicArrayY(100), PicArrayR(100) As Integer
Dim o2 As String
Dim iiv, seq, bb As Integer
```

การประกาศฟังก์ชันต่างๆ

1. Function AcDbCondition() - ฟังก์ชันนำชุดข้อมูลซึ่งแสดงรายละเอียดของรูปภาพในไฟล์ *.dxf ออกมาเก็บไว้ในตัวแปรเพื่อการนำไปใช้ต่อไป ทั้งนี้ยังแบ่งเป็นส่วนย่อยๆ คือ การค้นหาชุดข้อมูลของรูปภาพที่เป็นเส้น และการค้นหาชุดข้อมูลของรูปภาพที่เป็นวงกลม มีรายละเอียดดังนี้

```
Function AcDbCondition()
  'Circle Loop-----------------------------------------------
    If DataVar = "AcDbCircle" Then
      checkAcDb = 1
      Line Input #iFileNum, DataVar2
        If DataVar2 = 10 Then
          Line Input #iFileNum, DataVar2
          X(3) = Val(DataVar2)
        End If
        Line Input #iFileNum, DataVar2
        If DataVar2 = 20 Then
          Line Input #iFileNum, DataVar2
          Y(3) = Val(DataVar2)
        End If
        Line Input #iFileNum, DataVar2
        Line Input #iFileNum, DataVar2
        Line Input #iFileNum, DataVar2
        If DataVar2 = 40 Then
          Line Input #iFileNum, R
        End If
        Line Input #iFileNum, DataVar2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Line Input #iFileNum, DataVar2
        If DataVar2 = "AcDbArc" Then
          Line Input #iFileNum, DataVar2
          Line Input #iFileNum, P1
          Line Input #iFileNum, DataVar2
          Line Input #iFileNum, P2

          X(1) = Round(X(3) + R * Cos(P1 * 3.141 / 180))
          Y(1) = Round(Y(3) + R * Sin(P1 * 3.141 / 180))

          X(2) = Round(X(3) + R * Cos(P2 * 3.141 / 180))
          Y(2) = Round(Y(3) + R * Sin(P2 * 3.141 / 180))

        Else
          X(2) = Val(R) + Val(X(3))
            Y(2) = Y(3)
            X(1) = X(2)
            Y(1) = Y(2)
        End If
'X(1),Y(1)=start
'X(2),Y(2)=end
'X(3),Y(3)=center
'P1=start radius
'P2=end radius
'R=radius
        iFileNum2 = FreeFile
        If check = 1 Then
          Open txtExport For Output As #iFileNum2
        ElseIf check > 1 Then
          Open txtExport For Append As #iFileNum2
        End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        If check = 1 Then
           Print #iFileNum2, "G90"
        End If
        If Not ((X22 = X(1)) And (Y22 = Y(1))) Then
        'Print #iFileNum2, "G00 X0 Y0 Z0"
        Print #iFileNum2, "G00 Z" & -1
        Print #iFileNum2, "G00 X" & CInt(X(1)); " Y" & CInt(Y(1))
        Print #iFileNum2, "G01 Z" & txtDepth
        End If
        Print #iFileNum2, "G03 X" & CInt(X(3)); " Y" & CInt(Y(3)); " I" & CInt(X(2)); " J" &
CInt(Y(2))
        'Print #iFileNum2, "G00 Z" & txtThick.Text + 2
        Close #iFileNum2
        X22 = X(2)
        Y22 = Y(2)
      End If
'End Circle Loop-----------------------------------------


'Line Loop-----------------------------------------------
       If DataVar = "AcDbLine" Then
       Dim X1, X2, Y1, Y2 As Single
        checkAcDb = 2
          Line Input #iFileNum, DataVar2
          If DataVar2 = 10 Then
             Line Input #iFileNum, X1
             X1 = Val(X1)
          End If
          Line Input #iFileNum, DataVar2
          If DataVar2 = 20 Then
             Line Input #iFileNum, Y1
             Y1 = Val(Y1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    End If
    Line Input #iFileNum, DataVar2
    Line Input #iFileNum, DataVar2
    Line Input #iFileNum, DataVar2
    If DataVar2 = 11 Then
        Line Input #iFileNum, X2
        X2 = Val(X2)
    End If
    Line Input #iFileNum, DataVar2
    If DataVar2 = 21 Then
        Line Input #iFileNum, DataVar2
        Y2 = Val(DataVar2)
    End If

iFileNum2 = FreeFile
If check = 1 Then
    Open txtExport For Output As #iFileNum2
    'Print #iFileNum2, "&--"
    Print #iFileNum2, "G90"
ElseIf check > 1 Then
    Open txtExport For Append As #iFileNum2
End If

If Not ((X22 = X1) And (Y22 = Y1)) Then
Print #iFileNum2, "G00 Z" & -1
Print #iFileNum2, "G00 X" & CInt(X1); " Y" & CInt(Y1)
Print #iFileNum2, "G01 Z" & txtDepth
End If
Print #iFileNum2, "G01 X" & CInt(X2); " Y" & CInt(Y2)
'Print #iFileNum2, "G00 Z" & txtThick.Text + 2
'Print #iFileNum2, "G00 X0 Y0"
Close #iFileNum2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        'Store the old co-ordinates in order to compare with the new one
        X22 = X2
        Y22 = Y2
      End If

'End Line Loop----------------------------------------------
End Function
```

2. Function ExportPath() – ฟังก์ชันแปลงชื่อไฟล์รูปภาพ(*.dxf) ให้เป็นไฟล์เก็บจีโค้ด(*.nc) เพื่อนำไปใช้ในการเก็บจีโค้ดที่แปลงไว้หลังจากอิมพอร์ทไฟล์ที่หน้าต่าง Tools มีรายละเอียดดังนี้

```
Function ExportPath()
  Dim checkingChar As String
  Dim charSequence, i As Integer
  For i = 1 To Len(frmTools.txtImport.Text)
    checkingChar = Mid(frmTools.txtImport.Text, i, 1)
    If checkingChar = "." Then
      charSequence = i
      i = Len(frmTools.txtImport.Text)
    End If
  Next i
    txtExport = Left(frmTools.txtImport.Text, charSequence) & "nc"

End Function
```

3. Function FindChar() – ฟังก์ชันค้นหาตำแหน่งเว้นวรรคของจีโค้ดบรรทัดนั้นๆเก็บไว้ในตัวแปร เพื่อนำไปใช้ในการค้นหาค่า X และ Y จากนั้นทำการหาค่า X และ Y แล้วส่งไปยังหน้าต่าง Milling Controller (frmMANUAL) เพื่อทำการกัด โดยมีรายละเอียดดังนี้

```
Function FindChar()
G = Len(O)
j = 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
xx = 99
yy = 99
zz = 99

'Find space for each line input
   For n = 1 To G
      If Mid(O, n, 1) = " " Then
         SPACE(j) = n
         j = j + 1
      End If
   Next n
   SPACE(j) = Len(O) + 1

'Find X
   For n = 1 To G
      If Mid(O, n, 1) = "X" Then
         xx = n
      End If
   Next n

'Find Y
   For n = 1 To G
      If Mid(O, n, 1) = "Y" Then
         yy = n
      End If
   Next n

'Find Z
   For n = 1 To G
      If Mid(O, n, 1) = "Z" Then
         zz = n
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        End If
    Next n


'--Find First Value: either X,Y,Z
    ';Hundred>>3
    If (SPACE(2) - SPACE(1) - 2) = 3 Then
    '"2" is an amount of chars which are a space and "X" or "Y" or "Z"
      If (Mid(O, SPACE(1) + 1, 1) = ("X")) Then
            I2 = Mid(O, xx + 1, 3)
        frmMANUAL.Text1.Text = I2
      ElseIf (Mid(O, SPACE(1) + 1, 1) = ("Y")) Then
            J2 = Mid(O, yy + 1, 3)
        frmMANUAL.Text2.Text = J2
      ElseIf (Mid(O, SPACE(1) + 1, 1) = ("Z")) Then
            K2 = Mid(O, zz + 1, 3)
        frmMANUAL.Text3.Text = K2
      End If
    ';Ten>>2
    ElseIf (SPACE(2) - SPACE(1) - 2) = 2 Then
      If (Mid(O, SPACE(1) + 1, 1) = ("X")) Then
            I2 = Mid(O, xx + 1, 2)
        frmMANUAL.Text1.Text = I2
      ElseIf (Mid(O, SPACE(1) + 1, 1) = ("Y")) Then
            J2 = Mid(O, yy + 1, 2)
        frmMANUAL.Text2.Text = J2
      ElseIf (Mid(O, SPACE(1) + 1, 1) = ("Z")) Then
            K2 = Mid(O, zz + 1, 2)
        frmMANUAL.Text3.Text = K2
      End If
    ';Less than ten>>1
    ElseIf (SPACE(2) - SPACE(1) - 2) = 1 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        If (Mid(O, SPACE(1) + 1, 1) = ("X")) Then
            I2 = Mid(O, xx + 1, 1)
          frmMANUAL.Text1.Text = I2
        ElseIf (Mid(O, SPACE(1) + 1, 1) = ("Y")) Then
            J2 = Mid(O, yy + 1, 1)
          frmMANUAL.Text2.Text = J2

        ElseIf (Mid(O, SPACE(1) + 1, 1) = ("Z")) Then
            K2 = Mid(O, zz + 1, 1)
          frmMANUAL.Text3.Text = K2
        End If
      End If
'--Find Second Value: either Y,Z
      If (SPACE(3) - SPACE(2) - 2) = 3 Then
      '"2" is an amount of chars which are a space and "Y" or "Z"
        If (Mid(O, SPACE(2) + 1, 1) = ("Y")) Then
         J2 = Mid(O, yy + 1, 3)
         frmMANUAL.Text2.Text = J2
        ElseIf (Mid(O, SPACE(2) + 1, 1) = ("Z")) Then
         K2 = Mid(O, zz + 1, 3)
         frmMANUAL.Text3.Text = K2
        End If
      ElseIf (SPACE(3) - SPACE(2) - 2) = 2 Then
        If (Mid(O, SPACE(2) + 1, 1) = ("Y")) Then
         J2 = Mid(O, yy + 1, 2)
         frmMANUAL.Text2.Text = J2
        ElseIf (Mid(O, SPACE(2) + 1, 1) = ("Z")) Then
            K2 = Mid(O, zz + 1, 2)
        frmMANUAL.Text3.Text = K2
        End If
      ElseIf (SPACE(3) - SPACE(2) - 2) = 1 Then
        If (Mid(O, SPACE(2) + 1, 1) = ("Y")) Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        J2 = Mid(O, yy + 1, 1)
        frmMANUAL.Text2.Text = J2
      ElseIf (Mid(O, SPACE(2) + 1, 1) = ("Z")) Then
        K2 = Mid(O, zz + 1, 1)
      frmMANUAL.Text3.Text = K2
      End If
   End If
'--Find First Value: can be only z
   If (G - SPACE(3) - 1) = 3 Then
      K2 = Mid(O, zz + 1, 3)
      frmMANUAL.Text3.Text = K2
   ElseIf (G - SPACE(3) - 1) = 2 Then
        K2 = Mid(O, zz + 1, 2)
      frmMANUAL.Text3.Text = K2
   ElseIf (G - SPACE(3) - 1) = 1 Then
        K2 = Mid(O, zz + 1, 1)
      frmMANUAL.Text3.Text = K2
   End If

        frmMANUAL.Start
End Function
```

4. Function ACos(ByVal number As Double) As Double – ฟังก์ชันนี้รับค่าตัวเลขที่ส่งมาเมื่อมีการเรียกใช้ฟังก์ชัน เพื่อหาค่า arccosine และส่งค่าที่คำนวนได้กลับไป

```
' arc cosine
' error if NUMBER is outside the range [-1,1]

Function ACos(ByVal number As Double) As Double
number = CInt(number)
   If Abs(number) <> 1 Then
      ACos = 1.5707963267949 - Atn(number / Sqr(1 - (number * number)))
   ElseIf number = -1 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ACos = 3.14159265358979
  ElseIf number = 1 Then
    ACos = 0 '(implicit)
  End If
End Function
```

5. Function FindChar_cir() – ฟังก์ชันนี้คล้ายกับ Function FindChar แต่สำหรับการหาตำแหน่งเว้นวรรคของจีโค้ดที่หมายถึงรูปวงกลม ซึ่งมีข้อมูลต่างๆดังนี้

   X – ค่าแกน X ของจุดศูนย์กลางวงกลม

   Y – ค่าแกน Y ของจุดศูนย์กลางวงกลม

   I – ค่าแกน X ของจุดบนวงกลมจุดสุดท้ายของการกัด

   J – ค่าแกน Y ของจุดบนวงกลมจุดสุดท้ายของการกัด

   *หมายเหตุ เมื่อนำค่า I ลบด้วย ค่า X จะได้รัศมีของวงกลม

```
Function FindChar_cir()
G = Len(O)
j = 1

'Find space for each line input
  For n = 1 To G
    If Mid(O, n, 1) = " " Then
      SPACE(j) = n
      j = j + 1
    End If
  Next n


'Find X
  For n = 1 To G
    If Mid(O, n, 1) = "X" Then
      xx = n
    End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Next n


'Find Y
    For n = 1 To G
        If Mid(O, n, 1) = "Y" Then
            yy = n
        End If
    Next n


'Find i
    For n = 1 To G
        If Mid(O, n, 1) = "I" Then
            ii = n
        End If
    Next n


'Find j
    For n = 1 To G
        If Mid(O, n, 1) = "J" Then
            jj = n
        End If
    Next n


'--Find Value: either X,Y,1,J
'--START
'--Find Second Value: X


    ';Hundred>>3
    If (SPACE(2) - SPACE(1) - 2) = 3 Then
    '"2" is an amount of chars which are a space and "X"
        If (Mid(O, SPACE(1) + 1, 1) = ("X")) Then
                I2 = Mid(O, xx + 1, 3)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        a1 = I2
      End If

  ';Ten>>2
  ElseIf (SPACE(2) - SPACE(1) - 2) = 2 Then
      If (Mid(O, SPACE(1) + 1, 1) = ("X")) Then
            I2 = Mid(O, xx + 1, 2)
        a1 = I2
      End If

  ';Less than ten>>1
  ElseIf (SPACE(2) - SPACE(1) - 2) = 1 Then

      If (Mid(O, SPACE(1) + 1, 1) = ("X")) Then
            I2 = Mid(O, xx + 1, 1)
        a1 = I2
      End If

  End If

'--Find Second Value: Y

  If (SPACE(3) - SPACE(2) - 2) = 3 Then
  '"2" is an amount of chars which are a space and "Y"
    If (Mid(O, SPACE(2) + 1, 1) = ("Y")) Then
      J2 = Mid(O, yy + 1, 3)
      a2 = J2
    End If

  ElseIf (SPACE(3) - SPACE(2) - 2) = 2 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    If (Mid(O, SPACE(2) + 1, 1) = ("Y")) Then
     J2 = Mid(O, yy + 1, 2)
     a2 = J2
    End If

  ElseIf (SPACE(3) - SPACE(2) - 2) = 1 Then
    If (Mid(O, SPACE(2) + 1, 1) = ("Y")) Then
      J2 = Mid(O, yy + 1, 1)
     a2 = J2
    End If

  End If

'--Find Second Value: I

  If (SPACE(4) - SPACE(3) - 2) = 3 Then
  '"2" is an amount of chars which are a space and "I"
    If (Mid(O, SPACE(3) + 1, 1) = ("I")) Then
     I2 = Mid(O, ii + 1, 3)
     a3 = I2
    End If

  ElseIf (SPACE(4) - SPACE(3) - 2) = 2 Then
    If (Mid(O, SPACE(3) + 1, 1) = ("I")) Then
     I2 = Mid(O, ii + 1, 2)
     a3 = I2
    End If

  ElseIf (SPACE(4) - SPACE(3) - 2) = 1 Then
    If (Mid(O, SPACE(3) + 1, 1) = ("I")) Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        I2 = Mid(O, ii + 1, 1)
        a3 = I2
      End If

    End If

'--Find First Value: J

    If (G - SPACE(4) - 1) = 3 Then
      If (Mid(O, SPACE(4) + 1, 1) = ("J")) Then
      J2 = Mid(O, jj + 1, 3)
      a4 = J2
     End If

    ElseIf (G - SPACE(4) - 1) = 2 Then
      If (Mid(O, SPACE(4) + 1, 1) = ("J")) Then
        J2 = Mid(O, jj + 1, 2)
        a4 = J2
      End If

    ElseIf (G - SPACE(4) - 1) = 1 Then
      If (Mid(O, SPACE(4) + 1, 1) = ("J")) Then
        J2 = Mid(O, jj + 1, 1)
        a4 = J2
      End If

    End If
    a5 = CInt(frmMANUAL.CountX)
    a6 = CInt(frmMANUAL.CountY)
    circleCo
End Function
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Function circleCo() – ฟังก์ชันนี้ใช้หาจุดพิกัดต่างๆที่อยู่บนวงกลม เพื่อส่งไปยังหน้าต่าง Milling Controller ให้ทำการกัดวัสดุต่อไป

```
Function circleCo()
'a1,a2 is a center
'a3,a4 is an end-pt
'a5,a6 is a st-pt

  R = Sqr(((a1 - a3) * (a1 - a3)) + ((a2 - a4) * (a2 - a4)))

  PP1 = (180 * ACos((a5 - a1) / R)) / 3.14
  PP2 = (180 * ACos((a3 - a1) / R)) / 3.14
  Debug.Print "R = " & R & ", PP1 = " & CInt(PP1) & "," & "PP2 = " & CInt(PP2)

  If CInt(PP1) <> CInt(PP2) Then
    'Start pt.
    If (a1 > a5) And (a2 > a6) Then 'Quadt3
    PP1 = PP1 + 2 * (180 - PP1)
    ElseIf (a1 < a5) And (a2 > a6) Then 'Quadt4
    PP1 = 360 - PP1
    End If

    'End pt.
    If (a1 > a3) And (a2 > a4) Then 'Quadt3
    PP2 = PP2 + 2 * (180 - P2)
    ElseIf (a1 < a3) And (a2 > a4) Then 'Quadt4
    PP2 = 360 - P2
    End If
  End If
    PP1 = CInt(PP1 * 10) / 10
    PP2 = CInt(PP2 * 10) / 10
    Debug.Print "R = " & R & ", PP1 = " & PP1 & "," & "PP2 = " & PP2
    DrawCircle
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Function

Function DrawCircle()
  chkStop = 1
  If PP1 > PP2 Then

   zero = 0

                ' y1=Rsin(P1*pi/180)
                ' y2=Rsin(P2*pi/180)
                ' x1=Rcos(P1*pi/180)
                ' x2=Rcos(P2*pi/180)

    For i = P1 To 360 Step 0.005
      If frmMANUAL.chkStopClick = False Then

      X1 = R * Cos(PP1 * Pi / 180)
      Y1 = R * Sin(PP1 * Pi / 180)

      frmMANUAL.Text1.Text = X1
      frmMANUAL.Text2.Text = Y1
      frmMANUAL.Start
        If frmTools.chkStop = 1 Then
          Do While frmMANUAL.checkX = 0 Or frmMANUAL.checkY = 0

          DoEvents
          Sleep (40)
          Loop

        End If
      End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    End If
  Next i

  For i = zero To PP2 Step 0.005
    If frmMANUAL.chkStopClick = False Then

    X2 = R * Cos(PP2 * Pi / 180)
    Y2 = R * Sin(PP2 * Pi / 180)

    frmMANUAL.Text1.Text = X2
    frmMANUAL.Text2.Text = Y2
    frmMANUAL.Start
      If frmTools.chkStop = 1 Then
        Do While frmMANUAL.checkX = 0 Or frmMANUAL.checkY = 0

        DoEvents
        Sleep (40)
        Loop

      End If
    End If
    End If
  Next i

ElseIf PP2 > PP1 Then

zero = 0

                ' y1=Rsin(P1*pi/180)
                ' y2=Rsin(P2*pi/180)

                ' x1=Rcos(P1*pi/180)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ' x2=Rcos(P2*pi/180)

    For i = PP1 To PP2 Step 0.005
      If frmMANUAL.chkStopClick = False Then

      X1 = R * Cos(PP1 * Pi / 180)
      Y1 = R * Sin(PP1 * Pi / 180)

      frmMANUAL.Text1.Text = X1
      frmMANUAL.Text2.Text = Y1
      frmMANUAL.Start
        If frmTools.chkStop = 1 Then
          Do While frmMANUAL.checkX = 0 Or frmMANUAL.checkY = 0

          DoEvents
          Sleep (40)
          Loop

        End If
      End If
      End If
    Next i

ElseIf PP1 = PP2 Then
'zero = 0

                ' y1=Rsin(P1*pi/180)
                ' y2=Rsin(P2*pi/180)

                ' x1=Rcos(P1*pi/180)
                ' x2=Rcos(P2*pi/180)
    For i = 0 To 360 Step 0.005
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        If frmMANUAL.chkStopClick = False Then
        X1 = (R * Cos(i * 3.14159 / 180)) + a1
        Y1 = (R * Sin(i * 3.14159 / 180)) + a2

        frmMANUAL.Text1.Text = CInt(X1)
        frmMANUAL.Text2.Text = CInt(Y1)

        frmMANUAL.Start
          If frmTools.chkStop = 1 Then
            Do While frmMANUAL.checkX = 0 Or frmMANUAL.checkY = 0

            DoEvents
            Sleep (40)
            Loop

          End If
        End If
      Next i
    End If
    chkStop = 0
    bb = bb + 1
    Run2
End Function
```

7. Function FindGCondition() – ฟังก์ชันนี้ใช้ค้นหาว่าคำสั่งจีโค้ดที่กำลังทำงานอยู่นั้นเป็นคำสั่งประเภทไหน ไม่ว่าจะเป็นคำสั่งเคลื่อนที่ไปที่จุดพิกัด คำสั่งกัดวัสดุโดยเคลื่อนหัวกัดไปที่จุดที่กำหนด และคำสั่งกัดแบบวงกลม

```
Function FindGCondition()
GCondition = 99
  If Mid(O, 1, 3) = "G90" Then

  ElseIf Mid(O, 1, 3) = "G00" Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    GCondition = 0
    'Out pwrite, &H40
    FindChar
    'MoveABS

  ElseIf Mid(O, 1, 3) = "G01" Then
    GCondition = 1
    'Out pwrite, &H0
    FindChar
    'MoveABS

  ElseIf Mid(O, 1, 3) = "G03" Then
    GCondition = 3
    FindChar_cir
    'MoveABS
    'Out pwrite, &H0
    bb = bb - 1
  End If
End Function
```

8. ฟังก์ชันย่อยนี้ จะแสดงหน้าต่าง Import เมื่อกดปุ่ม Import ที่อยู่บนหน้าต่าง Tools

```
Private Sub Command1_Click()
  frmImport.Show
End Sub
```

9. ฟังก์ชันย่อยนี้ จะปิดหน้าต่างทั้งหมดที่ frmMDI ซึ่งเป็นหน้าต่างแม่ควบคุมอยู่

```
Private Sub Command2_Click()
  frmMDI.Hide
End Sub
```

10. Sub Convert() – จะเก็บค่าจีโค้ดที่แปลงได้ใส่ไว้ในไฟล์ ตามชื่อที่ได้จาก Function ExportPath มีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Sub Convert()
   frmDrawing.WrkSpace.Cls
   If frmDrawing.chkGridInt Mod 2 = 1 Then
      frmDrawing.Grid
   End If

   check = 1
   checkAcDb = 0

   X22 = 0
   Y22 = 0
   iFileNum = FreeFile

   Open txtImport.Text For Input As #iFileNum

      ExportPath
'=============================================================
   Do
   Line Input #iFileNum, DataVar
      AcDbCondition
   Loop Until checkAcDb <> 0
      check = check + 1

   Do
   Line Input #iFileNum, DataVar
      AcDbCondition
   Loop Until DataVar = "ENDSEC"
'=============================================================
   Open txtExport For Append As #iFileNum2
   Print #iFileNum2, "G00 Z" & -1
   Print #iFileNum2, "G00 X0 Y0"
   Print #iFileNum2, "---&"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Close #iFileNum2
    Close #iFileNum
    frmGCD.Show

    Open txtExport For Input As #iFileNum
      DataVar = Input(LOF(iFileNum), #iFileNum)
      frmGCD.txtGCD.Text = DataVar
    Close #iFileNum

    readfrmGCD
End Sub
```

11. Sub readfrmGCD() – ในการอ่านค่าจีโค้ดจากหน้าต่าง G-CODE ทั้งนี้เพื่อประโยชน์เมื่อมีการเปลี่ยนแปลงจีโค้ดจากผู้ใช้

```
Sub readfrmGCD()
    Dim LineIn As String
    Dim k As Integer
    LineIn = frmGCD.txtGCD.Text
    k = 1
    num = 1
    For i = 1 To Len(LineIn)
      If Asc(Mid(LineIn, i, 1)) = "13" Then
          myArray(num) = Mid(LineIn, k, i - k)
          num = num + 1
          k = i + 2
      End If
    Next i
    showPic
End Sub
Function FindChar2(O As String)
G = Len(O)
j = 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'Find space for each line input
   For n = 1 To G
      If Mid(O, n, 1) = " " Then
         SPACE(j) = n
         j = j + 1
      End If
   Next n
   SPACE(j) = Len(O) + 1

End Function
```

12. Sub show Pic() – ใช้แปลงค่าจากจีโค้ดเพื่อเป็นข้อมูลที่จะนำไปแสดงรูปที่หน้าต่าง Work Space ต่อไป

```
Sub showPic()

   For C = 1 To 100
      PicArrayX(C) = ""
      PicArrayY(C) = ""
'     PicArrayR(C) = ""
   Next C

iiv = 0
iiv = iiv + 1
seq = 1

o2 = myArray(iiv)
Do While Not o2 = "---&"
o2 = myArray(iiv)
'If Mid(o2, 1, 6) = "G00 Z-" Then

If Mid(o2, 1, 5) = "G00 Z" Then
   PicArrayX(seq) = "HELLO"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    PicArrayY(seq) = "HELLO"
ElseIf (o2 = "G00 X0 Y0" And myArray(iiv + 1) = "---&") Then
    Timer1.Enabled = True
    'DrawPic
ElseIf Mid(o2, 1, 5) = "G00 X" And Not myArray(iiv + 1) = "---&" Then
        FindChar2 (o2)
        PicArrayX(seq) = Mid(o2, SPACE(1) + 2, SPACE(2) - SPACE(1) - 2)
        PicArrayY(seq) = Mid(o2, SPACE(2) + 2, SPACE(3) - SPACE(2) - 2)
ElseIf Mid(o2, 1, 5) = "G01 Z" Then
    seq = seq - 1
ElseIf Mid(o2, 1, 5) = "G01 X" Then
    FindChar2 (o2)
    PicArrayX(seq) = Mid(o2, SPACE(1) + 2, SPACE(2) - SPACE(1) - 2)
    PicArrayY(seq) = Mid(o2, SPACE(2) + 2, SPACE(3) - SPACE(2) - 2)

ElseIf Mid(o2, 1, 5) = "G03 X" Then
    FindChar2 (o2)
    'In case of a full circle
    If PicArrayX(seq - 1) = Mid(o2, SPACE(3) + 2, SPACE(4) - SPACE(3) - 2) And
PicArrayY(seq - 1) = Mid(o2, SPACE(4) + 2, SPACE(5) - SPACE(4) - 2) Then
        PicArrayX(seq - 1) = "HELLO"
        PicArrayY(seq - 1) = "HELLO"
        PicArrayX(seq) = "C" & Mid(o2, SPACE(1) + 2, SPACE(2) - SPACE(1) - 2)
        PicArrayY(seq) = Mid(o2, SPACE(2) + 2, SPACE(3) - SPACE(2) - 2)
        PicArrayR(seq) = Mid(o2, SPACE(3) + 2, SPACE(4) - SPACE(3) - 2) - Mid(o2, SPACE(1)
+ 2, SPACE(2) - SPACE(1) - 2)
        If Mid(myArray(iiv + 1), 1, 5) = "G01 X" Then

            seq = seq + 1
            PicArrayX(seq) = Mid(o2, SPACE(3) + 2, SPACE(4) - SPACE(3) - 2)
            PicArrayY(seq) = Mid(o2, SPACE(4) + 2, SPACE(5) - SPACE(4) - 2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        End If
    End If
End If
    seq = seq + 1
    iiv = iiv + 1
Loop
seq = seq - 1
Debug.Print "This is seq" & seq
For B = 1 To seq
    Debug.Print PicArrayX(B)
    Debug.Print PicArrayY(B)
    Debug.Print PicArrayR(B)
Next B

    For iii = 1 To seq - 1
        If Not PicArrayY(iii) = "HELLO" Then
        If Val(PicArrayY(iii)) > 60 Then
            PicArrayY(iii) = Trim(Val(PicArrayY(iii)) - (2 * (Val(PicArrayY(iii)) - 60)))
        ElseIf Val(PicArrayY(iii)) < 60 Then
            PicArrayY(iii) = Trim(Val(PicArrayY(iii)) + (2 * (60 - Val(PicArrayY(iii)))))
        End If
        End If
    Next iii

End Sub
```

13. Sub DrawPic() – นำข้อมูลที่ได้จาก Sub ShowPic มาวาดรูปแสดงผลที่หน้าต่าง Work Space

```
Sub DrawPic()

    For iii = 2 To seq - 2
        If Not Mid(PicArrayX(iii), 1, 1) = "C" Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        If Not PicArrayX(iii + 1) = "HELLO" Then
        If Not PicArrayX(iii) = "HELLO" Then
        frmDrawing.WrkSpace.Line (Val(PicArrayX(iii)), Val(PicArrayY(iii)))-
(Val(PicArrayX(iii + 1)), Val(PicArrayY(iii + 1))), QBColor(14)
        End If
        End If
      Else
        'If Not PicArrayX(iii + 1) = "HELLO" Then
        'If Not PicArrayX(iii) = "HELLO" Then
        frmDrawing.WrkSpace.Circle (Val(Mid(PicArrayX(iii), 2, Len(PicArrayX(iii)))),
Val(PicArrayY(iii))), Val(PicArrayR(iii)), QBColor(14)
        'End If
        'End If
      End If
    Next iii
End Sub
```

14. ฟังก์ชันนี้ทำงานเมื่อปุ่ม Draw ที่หน้าต่าง Tools ถูกกด โดยจะไปเรียกใช้งานฟังก์ชันอื่นอีกที

```
Private Sub Command3_Click()
    frmDrawing.WrkSpace.Cls
    If frmDrawing.chkGridInt Mod 2 = 1 Then
      frmDrawing.Grid
    End If
    readfrmGCD
End Sub
```

15. Sub Run2() – ฟังก์ชันนี้ช่วยจัดการกับจีโค้ดที่ต้องการกัด

```
Sub Run2()
    DoEvents
    frmMANUAL.Show
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    If justRun = 0 Then
        chkRunclick = True
        justRun = justRun + 1
    End If

        O = myArray(bb)
        If Not Mid(O, 1, 4) = "---&" Then
            FindGCondition
        Else
            justRun = 0
        End If
        bb = bb + 1
End Sub
```

16. ฟังก์ชันนี้ทำงานเมื่อปุ่ม Run ที่หน้าต่าง Tools ถูกกด โดยจะไปเรียกใช้งาน Function Run2 อีกที หลังจากที่แยกข้อมูลทั้งหมดจากหน้าต่าง G-CODE ให้เป็นจีโค้ดทีละบันทัด

```
Private Sub Command4_Click() 'Run button
    frmMANUAL.chkManual = False
    Dim LineIn As String
    Dim k As Integer
    LineIn = frmGCD.txtGCD.Text
    k = 1
    num = 1
    For i = 1 To Len(LineIn)
        If Asc(Mid(LineIn, i, 1)) = "13" Then
            myArray(num) = Mid(LineIn, k, i - k)
            num = num + 1
            k = i + 2
        End If
    Next i
    Run2
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

17. ฟังก์ชันนี้ทำงานเมื่อมีการโหลด frmTools

```
Private Sub Form_Load()
chkRunclick = False
chkStop = 0
justRun = 0
bb = 2
   Me.Left = 3200
   Me.Top = 0
   'Me.Width = 3200
   'Me.Height = 7210
End Sub
```

18. ฟังก์ชันย่อยนี้เป็นไทม์เมอร์ที่ใช้วนเรียกคำสั่งการแสดงรูปที่หน้าต่าง Work Space

```
Private Sub Timer1_Timer()
   DrawPic
End Sub
```

โปรแกรมส่วนของ frmImport (หน้าต่างเพื่อเลือกไฟล์ในการนำเข้าไฟล์)

ในส่วนของฟอร์มนี้ใช้เพื่อเลือกไฟล์ที่ต้องการแปลงเป็นจีโค้ด โดยมีช่องแสดงให้เลือกไดร์ต่างๆ พาร์ทที่มีอยู่ในไดร์นั้นๆ และชื่อไฟล์ในโฟล์เดอร์

```
Function importFile()
   Dim filename As String
   filename = Dir1.Path & "\" & File1.filename
   Debug.Print (filename)
   frmTools.txtImport.Text = filename
   Me.Hide
   If Not Mid(filename, Len(filename) - 2, Len(filename)) = "dxf" Then
      MsgBox "Invalid file type, please select *.dxf file.", vboklnly, "ERROR!"
   Else
      frmTools.Convert
   End If
End Function
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub File1_DblClick()
   importFile

End Sub
Private Sub Command1_Click()
   importFile

End Sub

Private Sub Command2_Click()
   Me.Hide
End Sub

Private Sub Dir1_Change()
   File1.Path = Dir1.Path
End Sub

Private Sub Drive1_Change()
   Dir1.Path = Drive1.Drive
End Sub
```

โปรแกรมส่วนของ frmGCD (หน้าต่างแสดงจีโค้ดที่แปลงแล้ว)

```
Private Sub Form_Load()
   Me.Left = 0
   Me.Top = 0
   Me.Width = 3200
   Me.Height = 10200
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมส่วนของ frmGCD (หน้าต่างแสดงจีโค้ดที่แปลงแล้ว)

การประกาศตัวแปร

```
Private DrawingPen As String
Private XStart, YStart, XOld, YOld As Single
Private chkFirstClick, chkForMouseMove As Integer
Private chkGrid As Boolean
Public chkGridInt As Integer
```

การประกาศฟังก์ชันต่างๆ

1. ฟังก์ชันย่อยนี้ทำงานเมื่อปุ่ม Grid บนหน้าต่าง Work Space ถูกกด

```
Private Sub cmdGrid_Click()
  chkGridInt = chkGridInt + 1
  If chkGridInt Mod 2 = 1 Then
    Grid
    txtCmdLine.Caption = " *Grid On"
  Else
    Timer1.Enabled = False
    WrkSpace.Cls
    txtCmdLine.Caption = " *Grid Off"
  End If
End Sub
```

2. ฟังก์ชันย่อยนี้ทำงานเมื่อปุ่ม Valid Area ถูกกด โดย valid area คือขอบเขตการทำงานที่เครื่องกัดที่นำมาปรับปรุงนี้สามารถทำงานได้

```
Private Sub cmdVArea_Click()
  WrkSpace.Line (0, 120)-(80, 80), QBColor(4), B
End Sub
```

3. ฟังก์ชันย่อยนี้ทำงานเมื่อมีการโหลดหน้าต่าง Work Space โดยมีการกำหนดค่าพื้นฐานต่างๆ ของหน้าต่างนี้ที่นี่

```
Private Sub Form_Load()
  Me.Left = 3200
  Me.Top = 1800
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Me.Width = 11950
    Me.Height = 8350

    chkForMouseMove = 0
    chkGrid = True
    chkGridInt = 0
End Sub
```

4. Sub Grid – ทำหน้าที่แสดงเส้นgrid บนหน้าต่าง Work Space

```
Sub Grid()
WrkSpace.Line (0, 10)-(190, 10), QBColor(1)
WrkSpace.Line (0, 20)-(190, 20), QBColor(1)
WrkSpace.Line (0, 30)-(190, 30), QBColor(1)
WrkSpace.Line (0, 40)-(190, 40), QBColor(1)
WrkSpace.Line (0, 50)-(190, 50), QBColor(1)
WrkSpace.Line (0, 60)-(190, 60), QBColor(1)
WrkSpace.Line (0, 70)-(190, 70), QBColor(1)
WrkSpace.Line (0, 80)-(190, 80), QBColor(1)
WrkSpace.Line (0, 90)-(190, 90), QBColor(1)
WrkSpace.Line (0, 100)-(190, 100), QBColor(1)
WrkSpace.Line (0, 110)-(190, 110), QBColor(1)
WrkSpace.Line (0, 120)-(190, 120), QBColor(1)
'------------------------------
WrkSpace.Line (10, 0)-(10, 120), QBColor(1)
WrkSpace.Line (20, 0)-(20, 120), QBColor(1)
WrkSpace.Line (30, 0)-(30, 120), QBColor(1)
WrkSpace.Line (40, 0)-(40, 120), QBColor(1)
WrkSpace.Line (50, 0)-(50, 120), QBColor(1)
WrkSpace.Line (60, 0)-(60, 120), QBColor(1)
WrkSpace.Line (70, 0)-(70, 120), QBColor(1)
WrkSpace.Line (80, 0)-(80, 120), QBColor(1)
WrkSpace.Line (90, 0)-(90, 120), QBColor(1)
```

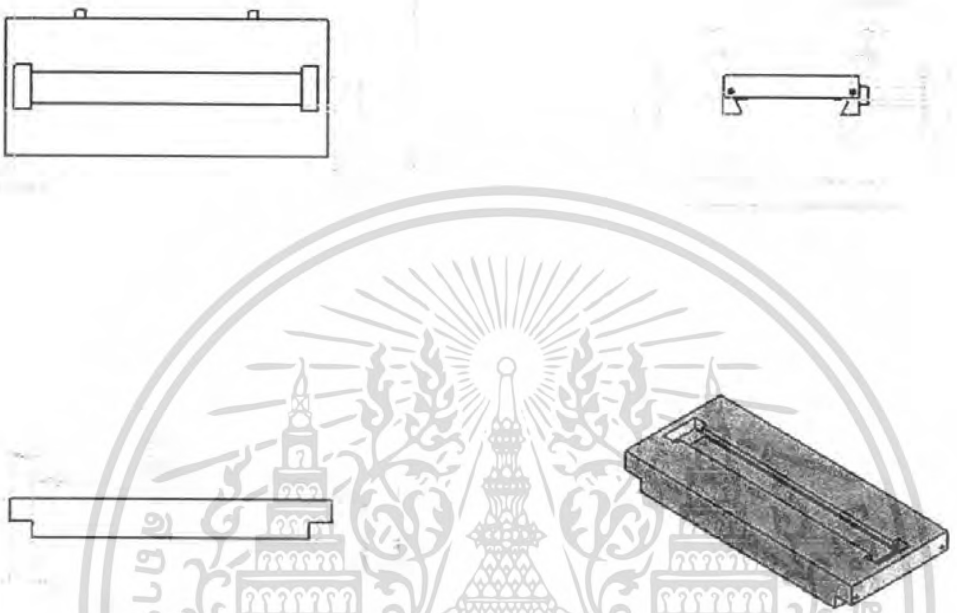
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
WrkSpace.Line (100, 0)-(100, 120), QBColor(1)
WrkSpace.Line (110, 0)-(110, 120), QBColor(1)
WrkSpace.Line (120, 0)-(120, 120), QBColor(1)
WrkSpace.Line (130, 0)-(130, 120), QBColor(1)
WrkSpace.Line (140, 0)-(140, 120), QBColor(1)
WrkSpace.Line (150, 0)-(150, 120), QBColor(1)
WrkSpace.Line (160, 0)-(160, 120), QBColor(1)
WrkSpace.Line (170, 0)-(170, 120), QBColor(1)
WrkSpace.Line (180, 0)-(180, 120), QBColor(1)
WrkSpace.Line (190, 0)-(190, 120), QBColor(1)
End Sub
```

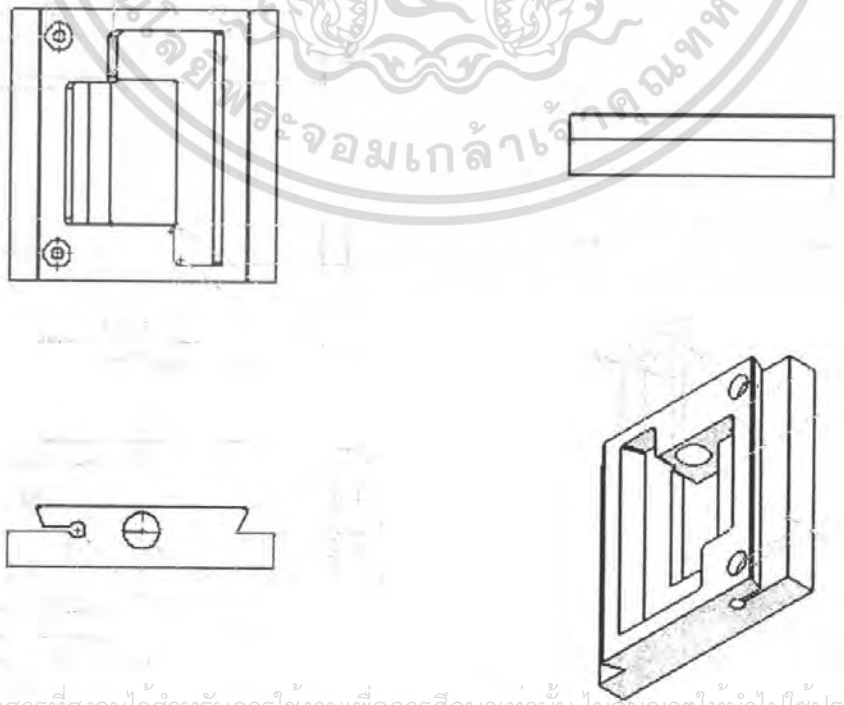
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

รูปการออกแบบเครื่องกัด

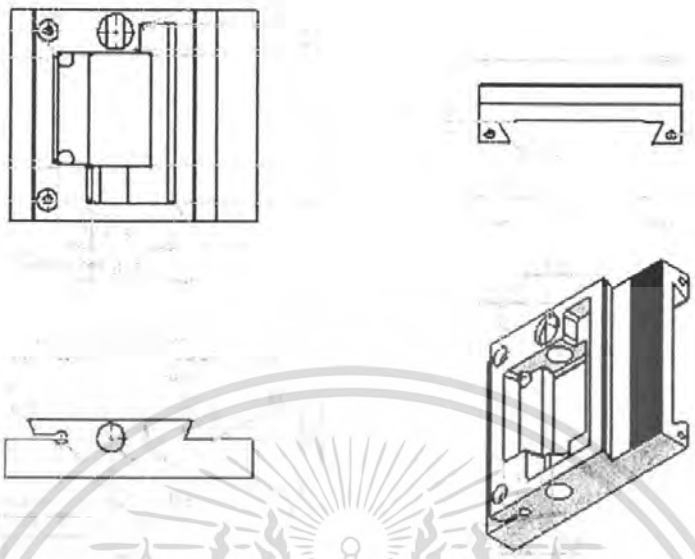


รูปที่ ค.1 ภาพถอดแบบส่วนประกอบของเครื่องกัดก่อนประกอบโดยสมบูรณ์

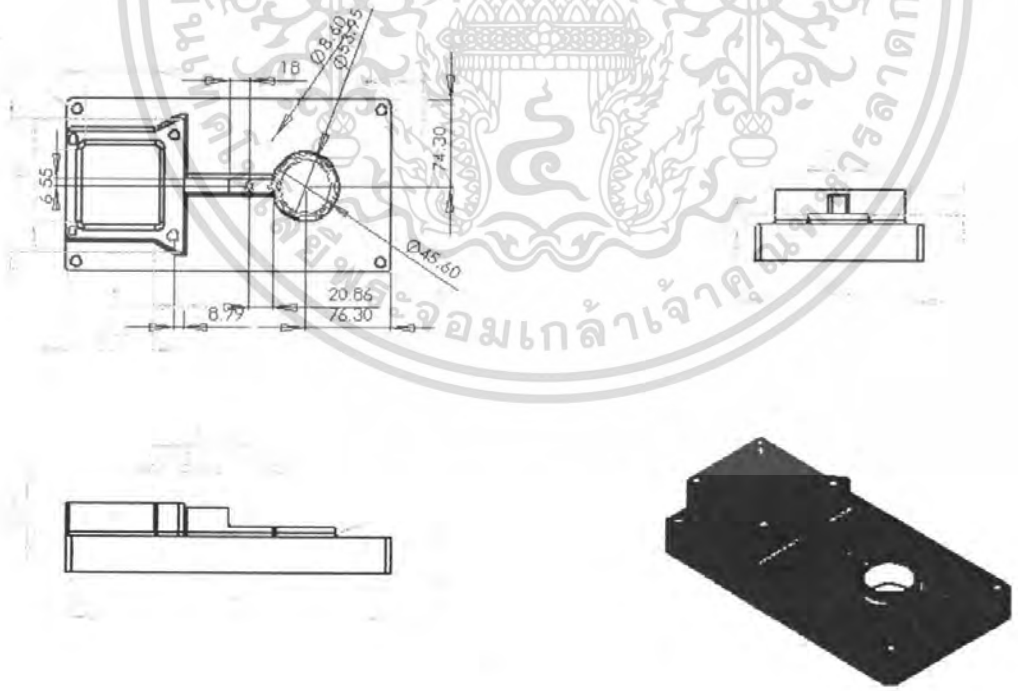


รูปที่ ค.2 ภาพถอดแบบส่วนประกอบของเครื่องกัดก่อนประกอบโดยสมบูรณ์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

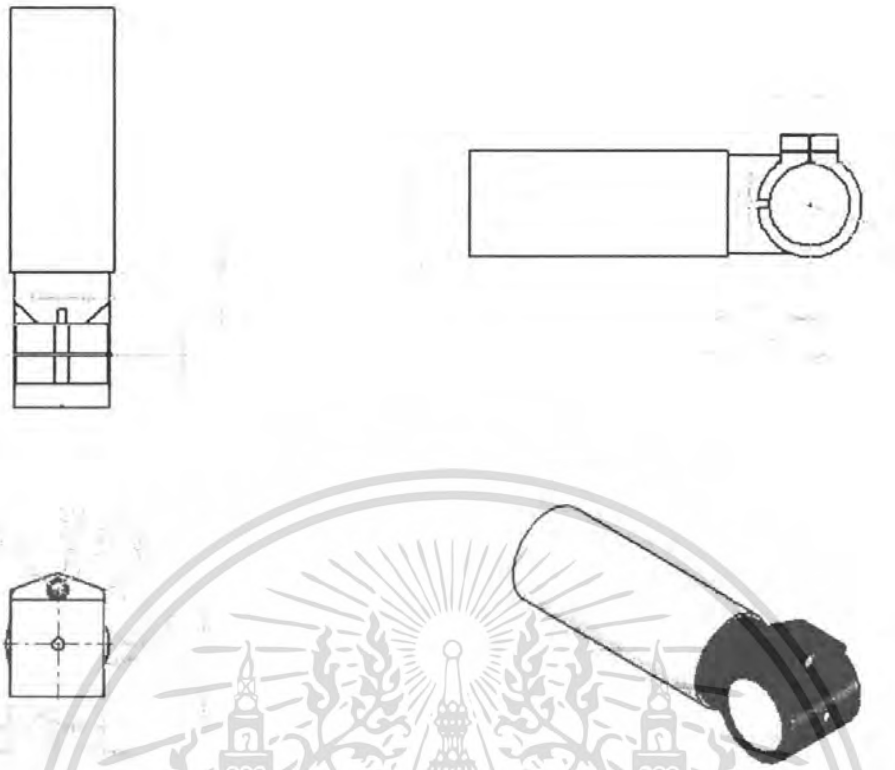


รูปที่ ค.3 ภาพฉายแบบส่วนประกอบของเครื่องกัดก่อนประกอบโดยสมบูรณ์ (ต่อ)

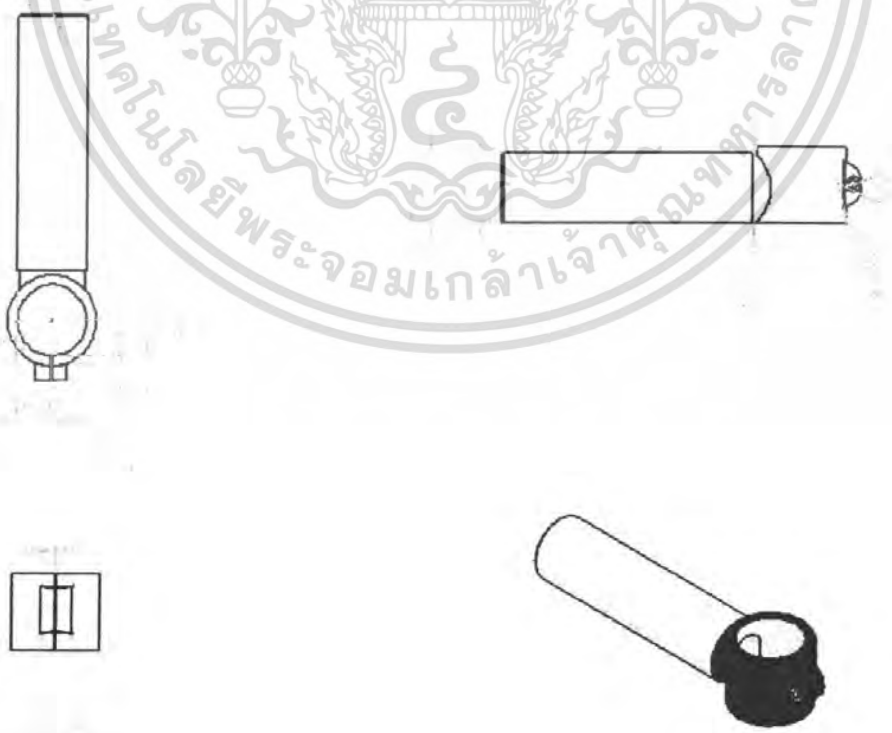


รูปที่ ค.4 ภาพฉายแบบส่วนประกอบของเครื่องกัดก่อนประกอบโดยสมบูรณ์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

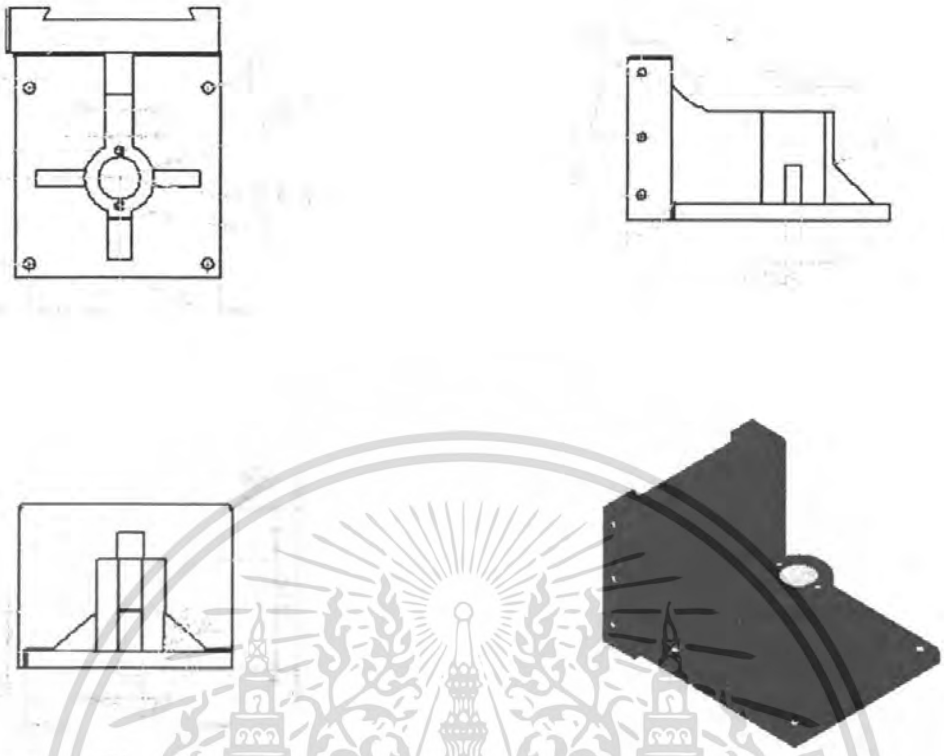


รูปที่ ค.5 ภาพฉายแบบส่วนประกอบของเครื่องกัดก่อนประกอบโดยสมบูรณ์ (ต่อ)

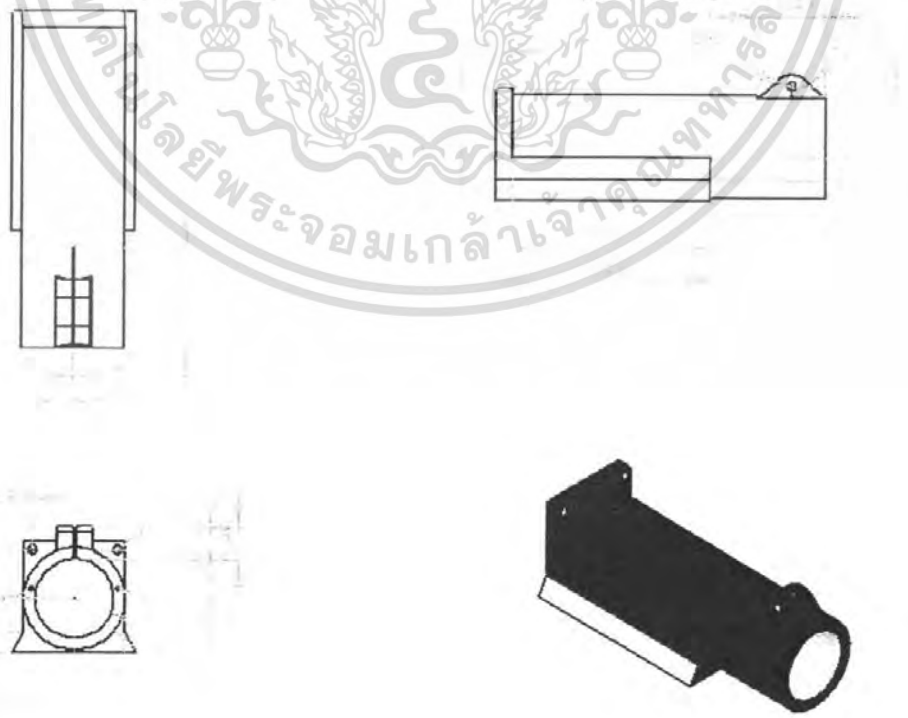


รูปที่ ค.6 ภาพฉายแบบส่วนประกอบของเครื่องกัดก่อนประกอบโดยสมบูรณ์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

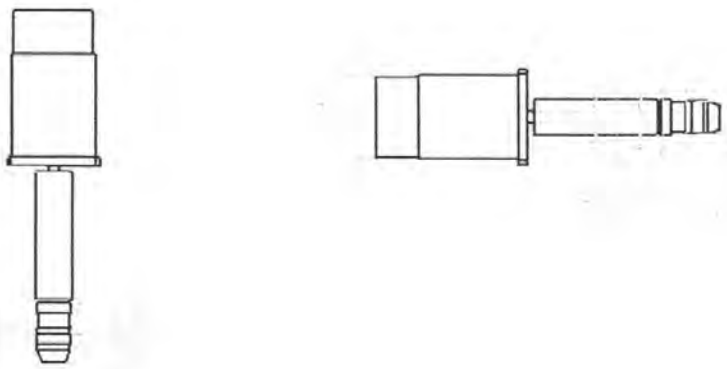


รูปที่ ค.7 ภาพฉายแบบส่วนประกอบของเครื่องกัดก่อนประกอบโดยสมบูรณ์ (ต่อ)

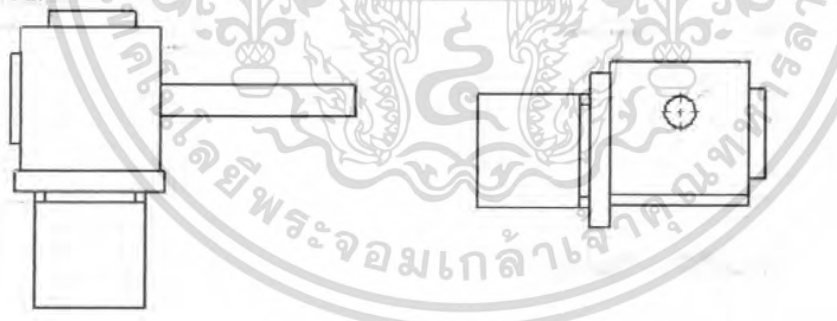


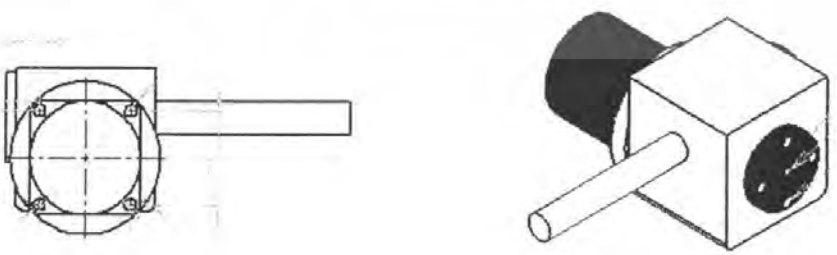
รูปที่ ค.8 ภาพฉายแบบส่วนประกอบของเครื่องกัดก่อนประกอบโดยสมบูรณ์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



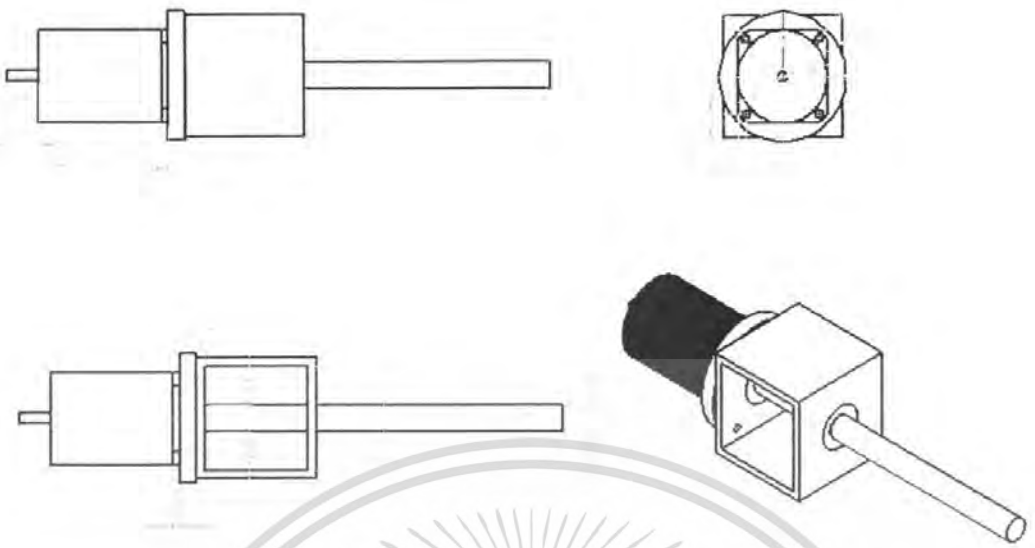
รูปที่ ค.9 ภาพฉายแบบส่วนประกอบของเครื่องกัดก่อนประกอบโดยสมบูรณ์ (ต่อ)





รูปที่ ค.10 ภาพฉายแบบส่วนประกอบของเครื่องกัดก่อนประกอบโดยสมบูรณ์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.11 ภาพถอดแบบส่วนประกอบของเครื่องกัดก่อนประกอบโดยสมบูรณ์ (ต่อ)

รูปที่ ค.12 ภาพประกอบโดยสมบูรณ์ของเครื่องกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

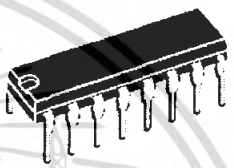
เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์

ง.1 เอกสารคู่มือการใช้งาน ULN2001A-ULN2002A

ULN2001A-ULN2002A
ULN2003A-ULN2004A

SEVEN DARLINGTON ARRAYS

- SEVEN DARLINGTONS PER PACKAGE
- OUTPUT CURRENT 500mA PER DRIVER (600mA PEAK)
- OUTPUT VOLTAGE 50V
- INTEGRATED SUPPRESSION DIODES FOR INDUCTIVE LOADS
- OUTPUTS CAN BE PARALLELED FOR HIGHER CURRENT
- TTL/CMOS/PMOS/DTL COMPATIBLE INPUTS
- INPUTS PINNED OPPOSITE OUTPUTS TO SIMPLIFY LAYOUT



DIP16

ORDERING NUMBERS: ULN2001A/2A/3A/4A

SO16

ORDERING NUMBERS: ULN2001D/2D/3D/4D

DESCRIPTION

The ULN2001A, ULN2002A, ULN2003 and ULN2004A are high voltage, high current darlington arrays each containing seven open collector darlington pairs with common emitters. Each channel rated at 500mA and can withstand peak currents of 600mA. Suppression diodes are included for inductive load driving and the inputs are pinned opposite the outputs to simplify board layout.

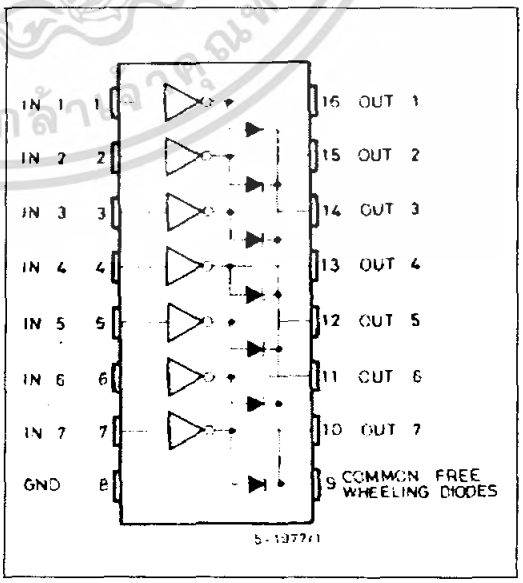
The four versions interface to all common logic families :

ULN2001A	General Purpose, DTL, TTL, PMOS, CMOS
ULN2002A	14-25V PMOS
ULN2003A	5V TTL, CMOS
ULN2004A	6–15V CMOS, PMOS

These versatile devices are useful for driving a wide range of loads including solenoids, relays DC motors, LED displays filament lamps, thermal printheads and high power buffers.

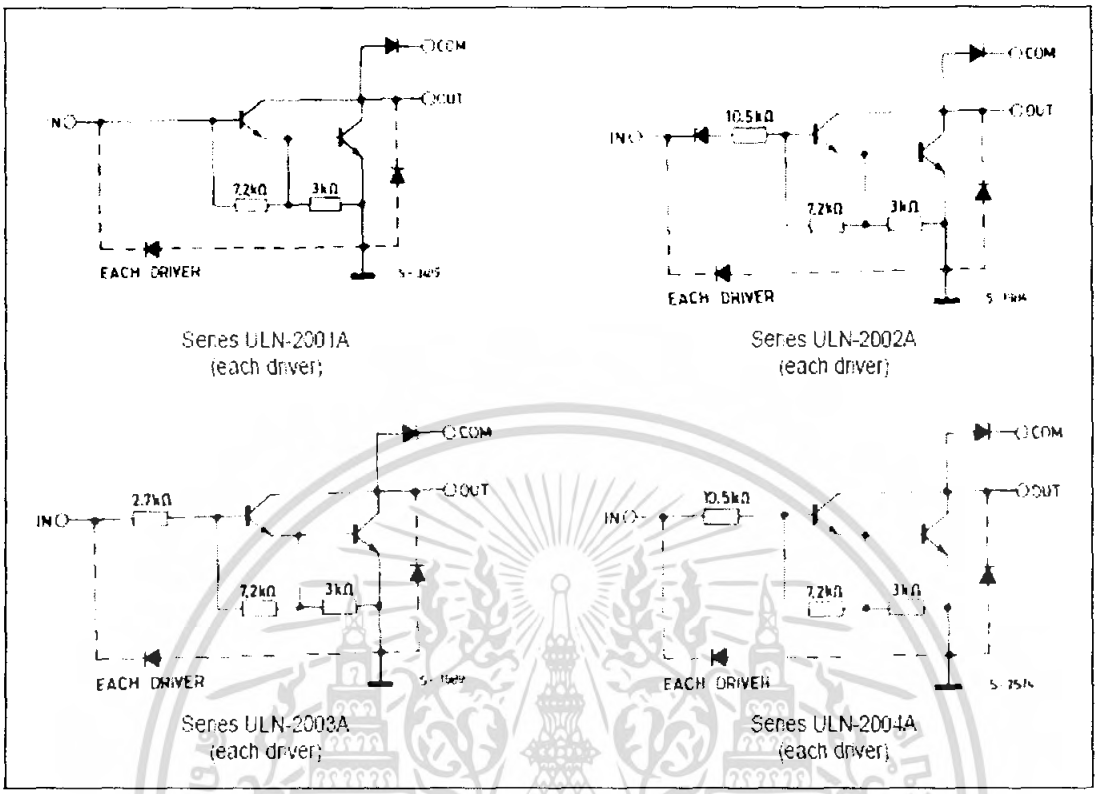
The ULN2001A/2002A/2003A and 2004A are supplied in 16 pin plastic DIP packages with a copper leadframe to reduce thermal resistance. They are available also in small outline package (SO-16) as ULN2001D/2002D/2003D/2004D.

PIN CONNECTION



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SCHEMATIC DIAGRAM



Series ULN-2001A (each driver)

Series ULN-2002A (each driver)

Series ULN-2003A (each driver)

Series ULN-2004A (each driver)

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_o	Output Voltage	50	V
V_{in}	Input Voltage (for ULN2002A/D - 2003A/D - 2004A/D)	30	V
I_c	Continuous Collector Current	500	mA
I_b	Continuous Base Current	25	mA
T_{a-E}	Operating Ambient Temperature Range	– 20 to 85	°C
T_{stg}	Storage Temperature Range	– 55 to 150	°C
T_j	Junction Temperature	150	°C

THERMAL DATA

Symbol	Parameter		DIP16	SO16	Unit
$R_{th\ j\text{-}amb}$	Thermal Resistance Junction-ambient	Max.	70	120	°C/W

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS (T_{amb} = 25°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit	Fig.
I_{CEX}	Output Leakage Current	V_{CE} = 50V T_{amb} = 70 °C, V_{CE} = 50V T_{amb} = 70 °C for ULN2002A V_{CE} = 50V, V_i = 6V for ULN2004A V_{CE} = 50V, V_i = 1V			50 100 500 500	µA µA µA µA	1a 1a 1c 1c
$V_{CE(sat)}$	Collector-emitter Saturation Voltage	I_C = 100mA, I_B = 250µA I_C = 200 mA, I_B = 350uA I_C = 350mA, I_B = 500µA		0.9 1.1 1.3	1.1 1.3 1.6	V V V	2 2 2
$I_{I(on)}$	Input Current	for ULN2002A, V_i = 17V for ULN2003A, V_i = 3.85V for ULN2004A, V_i = 5V V_i = 12V		0.82 0.93 0.35 1	1.25 1.35 0.5 1.45	mA mA mA mA	3 3 3 3
$I_{I(off)}$	Input Current	T_{amb} = 70 °C, I_C = 500µA	50	65		µA	4
$V_{I(on)}$	Input Voltage	V_{CE} = 2V for ULN2002A I_C = 300mA for ULN2003A I_C = 200mA I_C = 250mA I_C = 300mA for ULN2004A I_C = 125mA I_C = 200mA I_C = 275mA I_C = 350mA			 13 2.4 2.7 3 5 6 7 8	V	5
h_{FE}	DC Forward Current Gain	for ULN2001A V_{CE} = 2V, I_C = 350mA	1000				2
C_i	Input Capacitance			15	25	pF	
t_{PLH}	Turn-on Delay Time	0.5 V_i to 0.5 V_o		0.25	1	µs	
t_{PHL}	Turn-off Delay Time	0.5 V_i to 0.5 V_o		0.25	1	µs	
I_R	Clamp Diode Leakage Current	V_R = 50V T_{amb} = 70 °C, V_R = 50V			50 100	µA µA	6 6
V_F	Clamp Diode Forward Voltage	I_F = 350mA		1.7	2	V	7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TEST CIRCUITS

Figure 1a.

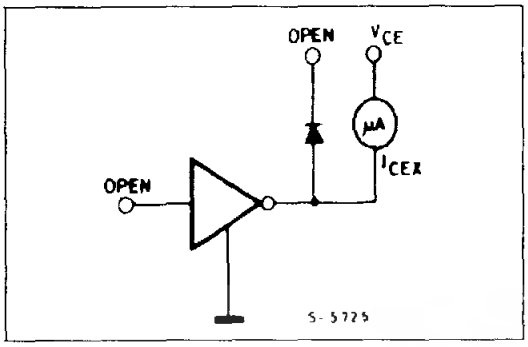


Figure 1b.

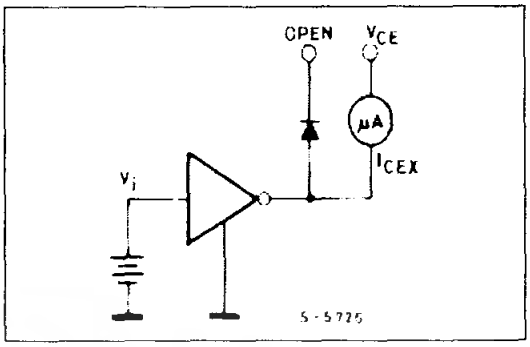


Figure 2.

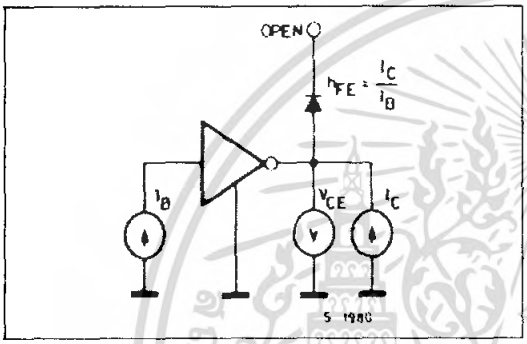


Figure 3.

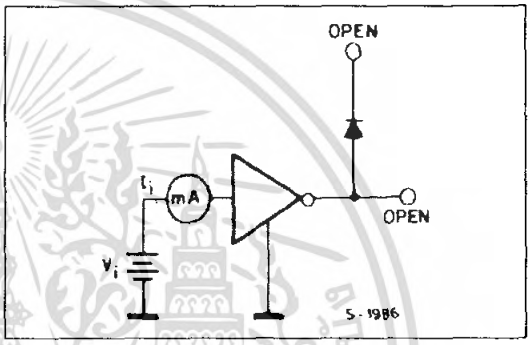


Figure 4.

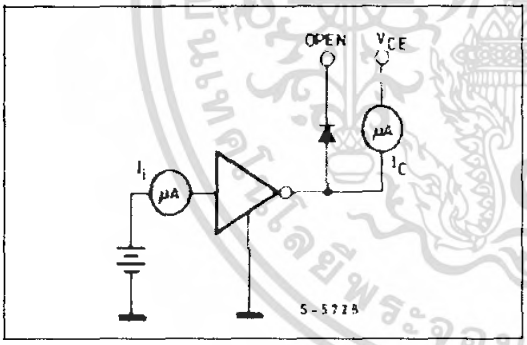


Figure 5.

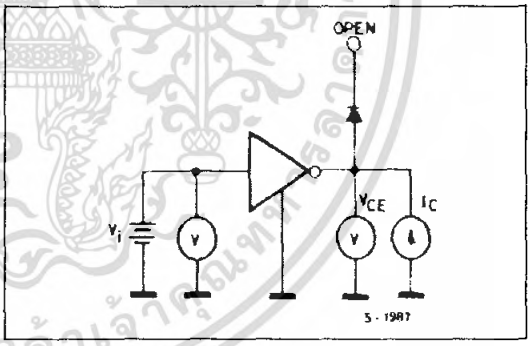


Figure 6.

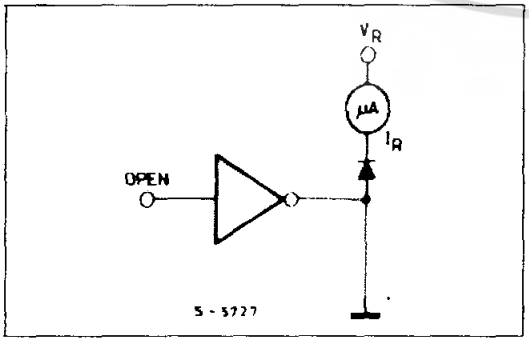
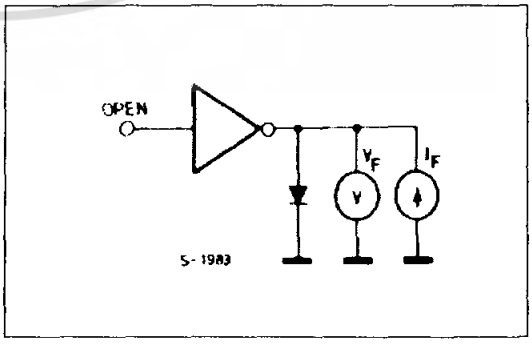


Figure 7.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 8: Collector Current versus Input Current

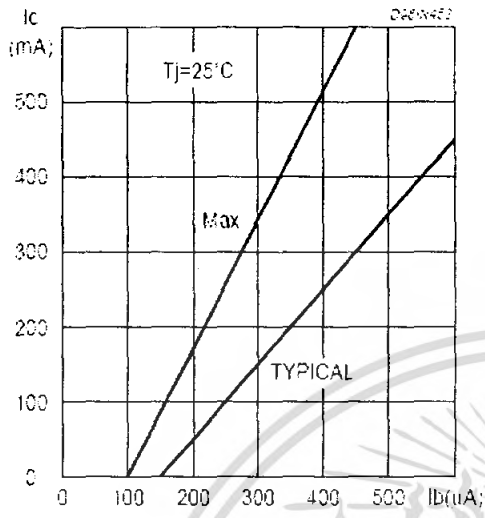


Figure 9: Collector Current versus Saturation Voltage

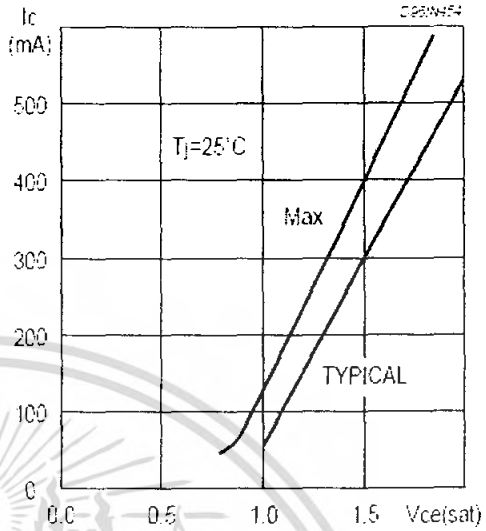


Figure 10: Peak Collector Current versus Duty Cycle

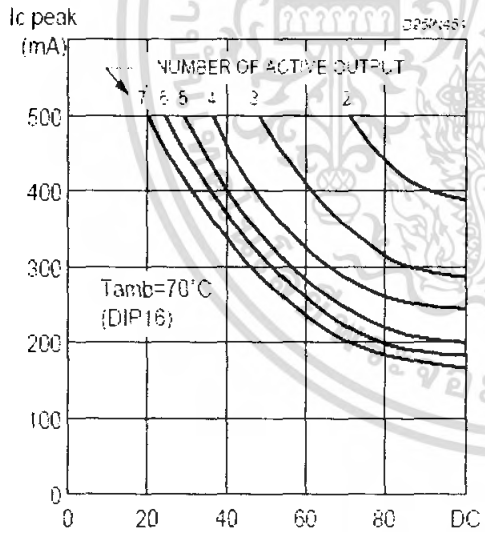
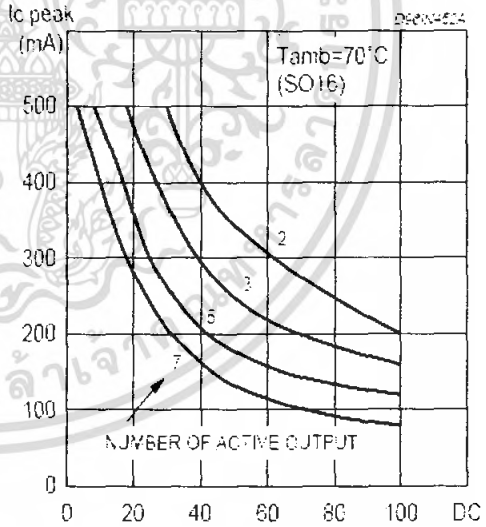


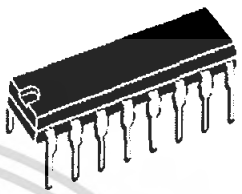
Figure 11: Peak Collector Current versus Duty Cycle



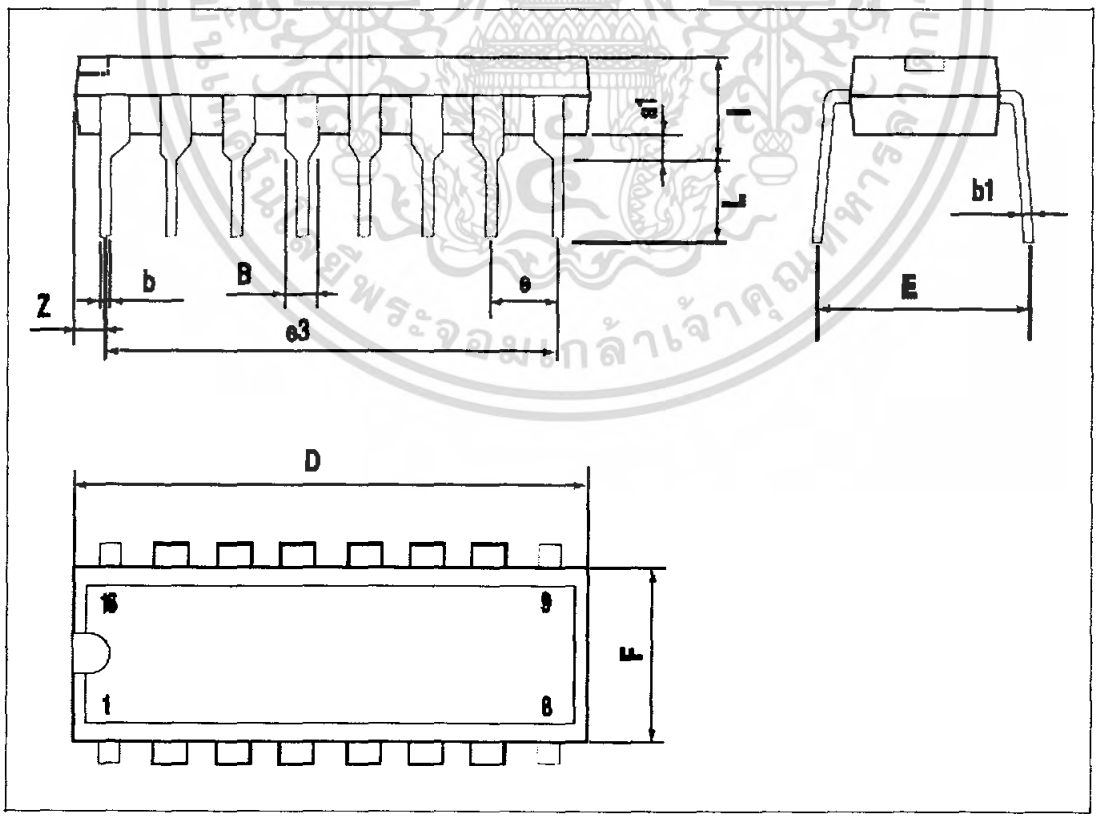
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
a1	0.51			0.020		
B	0.77		1.65	0.030		0.065
b		0.5			0.020	
b1		0.25			0.010	
D			20			0.787
E		8.5			0.335	
e		2.54			0.100	
e3		17.78			0.700	
F			7.1			0.280
I			5.1			0.201
L		3.3			0.130	
Z			1.27			0.050

OUTLINE AND MECHANICAL DATA



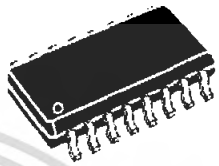
DIP16



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

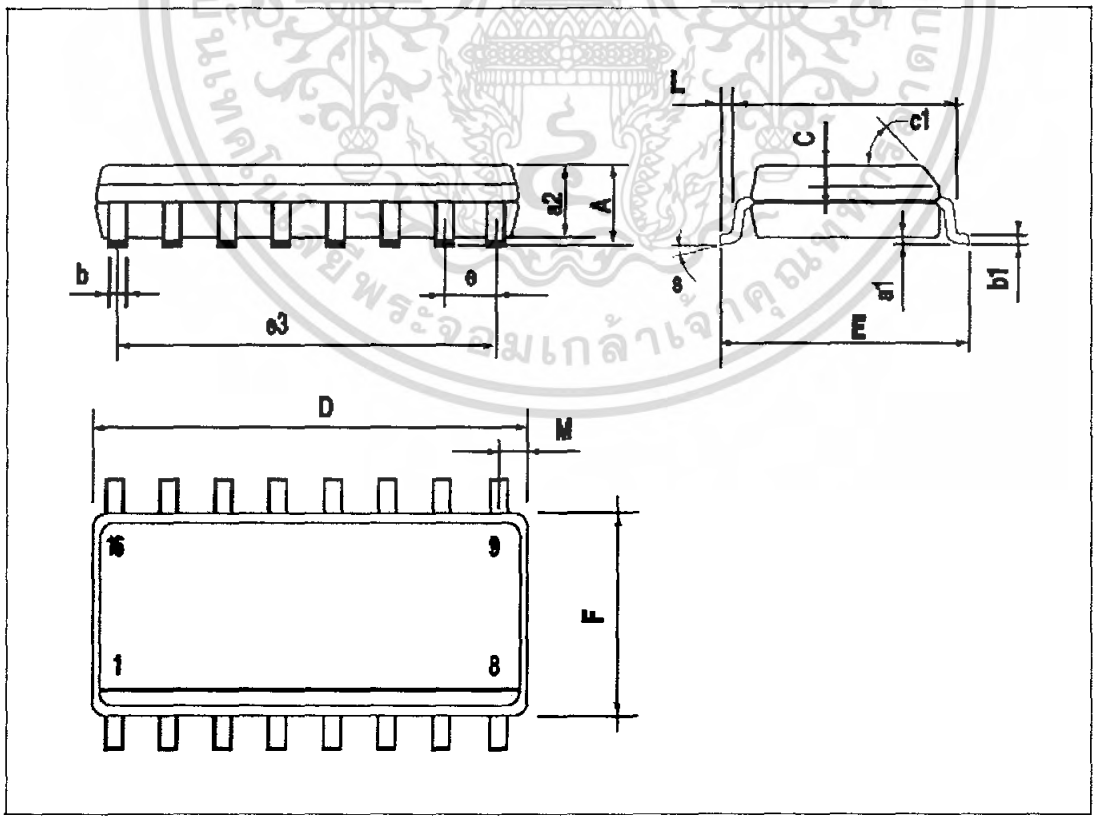
DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			1.75			0.069
a1	0.1		0.25	0.004		0.009
a2			1.6			0.063
b	0.35		0.46	0.014		0.018
b1	0.19		0.25	0.007		0.010
C		0.5			0.020	
c1	45° (typ.)					
D (1)	9.8		10	0.386		0.394
E	5.8		6.2	0.228		0.244
e		1.27			0.050	
e3		8.89			0.350	
F (1)	3.8		4	0.150		0.157
G	4.6		5.3	0.181		0.209
L	0.4		1.27	0.016		0.050
M			0.62			0.024
S	8° (max.)					

OUTLINE AND MECHANICAL DATA



SO16 Narrow

(1) D and F do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm (.006 inch).



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics

© 2002 STMicroelectronics – Printed in Italy – All Rights Reserved

STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - Canada - China - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States.

http://www.st.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ง.2 เอกสารคู่มือการใช้งาน LM124

National Semiconductor

August 2000

LM124/LM224/LM324/LM2902
Low Power Quad Operational Amplifiers

General Description

The LM124 series consists of four independent high gain, internally frequency compensated operational amplifiers which were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage.

Application areas include transducer amplifiers, DC gain blocks and all the conventional op amp circuits which now can be more easily implemented in single power supply systems. For example, the LM124 series can be directly operated off of the standard +5V power supply voltage which is used in digital systems and will easily provide the required interface electronics without requiring the additional ±15V power supplies.

Unique Characteristics

- In the linear mode the input common-mode voltage range includes ground and the output voltage can also swing to ground, even though operated from only a single power supply voltage
- The unity gain cross frequency is temperature compensated
- The input bias current is also temperature compensated

Advantages

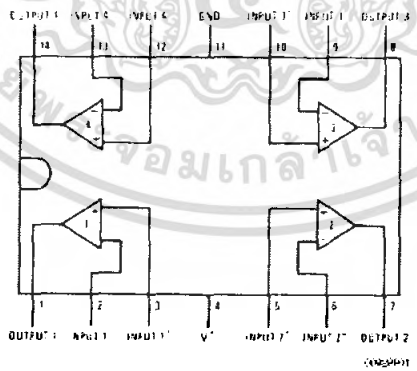
- Eliminates need for dual supplies
- Four internally compensated op amps in a single package
- Allows directly sensing near GND and V_{OUT} also goes to GND
- Compatible with all forms of logic
- Power drain suitable for battery operation

Features

- Internally frequency compensated for unity gain
- Large DC voltage gain 100 dB
- Wide bandwidth (unity gain) 1 MHz (temperature compensated)
- Wide power supply range:
  Single supply 3V to 32V
  or dual supplies ±1.5V to ±16V
- Very low supply current drain (700 µA)—essentially independent of supply voltage
- Low input biasing current 45 nA (temperature compensated)
- Low input offset voltage 2 mV and offset current: 5 nA
- Input common-mode voltage range includes ground
- Differential input voltage range equal to the power supply voltage
- Large output voltage swing 0V to V+ − 1.5V

Connection Diagrams

Dual-In-Line Package



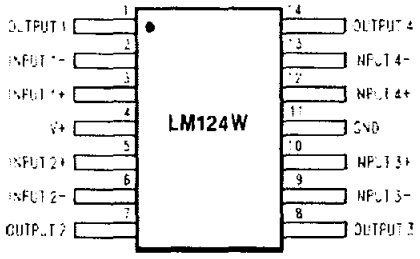
Top View

Order Number LM124J, LM124AJ, LM124J/883 (Note 2), **LM124AJ/883** (Note 1), **LM224J, LM224AJ, LM324J, LM324M, LM324MX, LM324AM, LM324AMX, LM2902M, LM2902MX, LM324N, LM324AN, LM324MT, LM324MTX or LM2902N LM124AJRQML and LM124AJRQMLV**(Note 3)
See NS Package Number J14A, M14A or N14A

© 2004 National Semiconductor Corporation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Connection Diagrams (Continued)



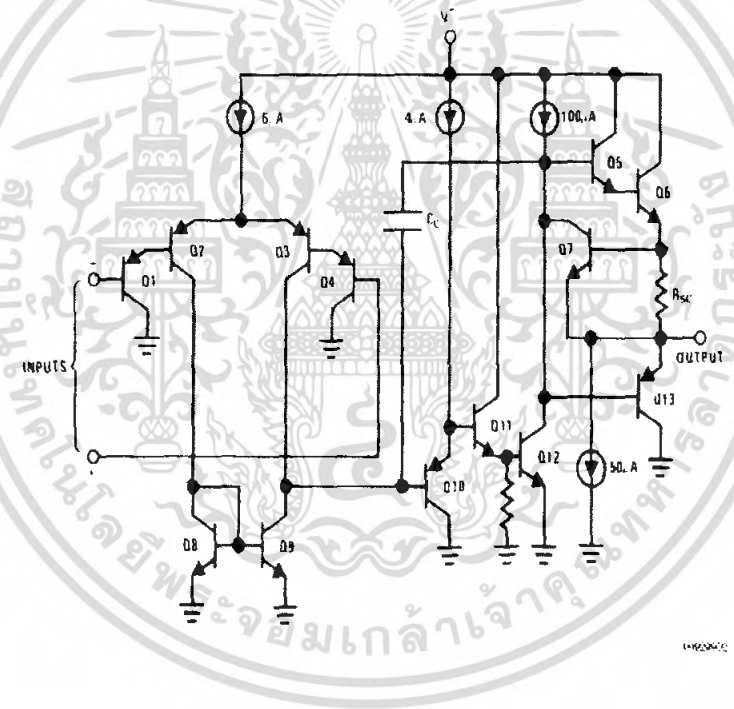
Order Number LM124AW/883, LM124AWG/883, LM124W/883 or LM124WG/883
LM124AWRQML and LM124AWRQMLV(Note 3)
See NS Package Number W14B
LM124AWGRQML and LM124AWGRQMLV(Note 3)
See NS Package Number WG14A

Note 1: LM124A available per JM38510/11006

Note 2: LM124 available per JM38510/11005

Note 3: See STD Mil DWG 5962R99504 for Radiation Tolerant Device

Schematic Diagram (Each Amplifier)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings (Note 12)

Distributors for availability and specifications.

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/

	LM124/LM224/LM324 LM124A/LM224A/LM324A	LM2902
Supply Voltage, V^+	32V	26V
Differential Input Voltage	32V	26V
Input Voltage	−0.3V to +32V	−0.3V to +26V
Input Current		
(V_{IN} < −0.3V) (Note 6)	50 mA	50 mA
Power Dissipation (Note 4)		
Molded DIP	1130 mW	1130 mW
Cavity DIP	1260 mW	1260 mW
Small Outline Package	800 mW	800 mW
Output Short-Circuit to GND		
(One Amplifier) (Note 5)		
$V^+ \leq 15V$ and $T_A = 25°C$	Continuous	Continuous
Operating Temperature Range		−40°C to +85°C
LM324/LM324A	0°C to +70°C	
LM224/LM224A	−25°C to +85°C	
LM124/LM124A	−55°C to +125°C	
Storage Temperature Range	−65°C to +150°C	−65°C to +150°C
Lead Temperature (Soldering, 10 seconds)	260°C	260°C
Soldering Information		
Dual-In-Line Package		
Soldering (10 seconds)	260°C	260°C
Small Outline Package		
Vapor Phase (60 seconds)	215°C	215°C
Infrared (15 seconds)	220°C	220°C

See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.

ESD Tolerance (Note 13)	250V	250V

Electrical Characteristics

$V^+ = +5.0V$, (Note 7), unless otherwise stated

Parameter	Conditions	LM124A			LM224A			LM324A			Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Input Offset Voltage	(Note 8) $T_A = 25°C$		1	2		1	3		2	3	mV
Input Bias Current (Note 9)	$I_{IN(+)}$ or $I_{IN(-)}$, $V_{CM} = 0V$, $T_A = 25°C$		20	50		40	60		45	100	nA
Input Offset Current	$I_{IN(+)}$ or $I_{IN(-)}$, $V_{CM} = 0V$, $T_A = 25°C$		2	10		2	15		5	30	nA
Input Common-Mode Voltage Range (Note 10)	$V^+ = 30V$, (LM2902, $V^+ = 26V$), $T_A = 25°C$	0		$V^+-1.5$	0		$V^+-1.5$	0		$V^+-1.5$	V
Supply Current	Over Full Temperature Range $R_L = \infty$ On All Op Amps										mA
	$V^+ = 30V$ (LM2902 $V^+ = 26V$)		1.5	3		1.5	3		1.5	3	
	$V^+ = 5V$		0.7	1.2		0.7	1.2		0.7	1.2	
Large Signal Voltage Gain	$V^+ = 15V$, $R_L \geq 2k\Omega$ ($V_O = 1V$ to $11V$), $T_A = 25°C$	50	100		50	100		25	100		V/mV
Common-Mode	DC, $V_{CM} = 0V$ to $V^+ - 1.5V$,	70	85		70	65		65	85		dB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Characteristics (Continued)

V^+ = +5.0V, (Note 7), unless otherwise stated

Parameter		Conditions		LM124A			LM224A			LM324A			Units
				Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Rejection Ratio		T_A = 25 C											
Power Supply Rejection Ratio		V^+ = 5V to 30V (LM2902, V^+ = 5V to 26V), T_A = 25 C		65	100		65	100		65	100		dB
Amplifier-to-Amplifier Coupling (Note 11)		f = 1 kHz to 20 kHz, T_A = 25 C (Input Referred)			−120			−120			−120		dB
Output Current	Source	V_{IN}^+ = 1V, V_{IN}^- = 0V, V^+ = 15V, V_O = 2V, T_A = 25 C		20	40		20	40		20	40		mA
	Sink	V_{IN}^- = 1V, V_{IN}^+ = 0V, V^+ = 15V, V_O = 2V, T_A = 25 C		10	20		10	20		10	20		mA
		V_{IN}^- = 1V, V_{IN}^+ = 0V, V^+ = 15V, V_O = 200 mV, T_A = 25 C		12	50		12	50		12	50		µA
Short Circuit to Ground		(Note 5) V^+ = 15V, T_A = 25 C			40	60		40	60		40	60	mA
Input Offset Voltage		(Note 6)				4			4			5	mV
V_{OS} Drift		R_S = 0Ω			7	20		7	20		7	30	µV/ C
Input Offset Current		$I_{IN(+)} - I_{IN(-)}$, V_{CM} = 0V				30			30			75	nA
I_{OS} Drift		R_S = 0Ω			10	200		10	200		10	300	pA/ C
Input Bias Current		$I_{IN(+)}$ or $I_{IN(-)}$			40	100		40	100		40	200	nA
Input Common-Mode Voltage Range (Note 10)		V^+ = +30V (LM2902, V^+ = 26V)		0		V^+−2	0		V^+−2	0		V^+−2	V
Large Signal Voltage Gain		V^+ = +15V (V_O Swing = 1V to 11V), R_L ≥ 2 kΩ		25			25			15			V/mV
Output Voltage Swing	V_{OH}	V^+ = 30V	R_L = 2 kΩ	26			26			26			V
		(LM2902, V^+ = 26V)	R_L = 10 kΩ	27	28		27	28		27	28		
	V_{OL}	V^+ = 5V, R_L = 10 kΩ			5	20		5	20		5	20	mV
Output Current	Source	V_O = 2V	V_{IN}^+ = +1V, V_{IN}^- = 0V, V^+ = 15V	10	20		10	20		10	20		mA
	Sink		V_{IN}^- = +1V, V_{IN}^+ = 0V, V^+ = 15V	10	15		5	8		5	8		

Electrical Characteristics

V^+ = +5.0V, (Note 7), unless otherwise stated

Parameter	Conditions	LM124/LM224			LM324			LM2902			Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Input Offset Voltage	(Note 6) T_A = 25 C		2	5		2	7		2	7	mV
Input Bias Current (Note 9)	$I_{IN(+)}$ or $I_{IN(-)}$, V_{CM} = 0V, T_A = 25 C		45	150		45	250		45	250	nA
Input Offset Current	$I_{IN(+)}$ or $I_{IN(-)}$, V_{CM} = 0V, T_A = 25 C		3	30		5	50		5	50	nA
Input Common-Mode Voltage Range (Note 10)	V^+ = 30V, (LM2902 V^+ = 26V), T_A = 25 C	0		V^+−1.5	0		V^+−1.5	0		V^+−1.5	V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Characteristics (Continued)

V^+ = +5.0V, (Note 7), unless otherwise stated

Parameter		Conditions		LM124/LM224			LM324			LM2902			Units
				Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Supply Current		Over Full Temperature Range $R_L = \infty$ On All Op Amps											mA
		V^+ = 30V (LM2902 V^+ = 26V)			1.5	3		1.5	3		1.5	3	
		V^+ = 5V			0.7	1.2		0.7	1.2		0.7	1.2	
Large Signal Voltage Gain		V^+ = 15V, R_L = 2kΩ (V_O = 1V to 11V), T_A = 25°C		50	100		25	100		25	100		V/mV
Common-Mode Rejection Ratio		DC, V_{CM} = 0V to V^+ − 1.5V, T_A = 25°C		70	85		65	85		50	70		dB
Power Supply Rejection Ratio		V^+ = 5V to 30V (LM2902, V^+ = 5V to 26V), T_A = 25°C		65	100		65	100		50	100		dB
Amplifier-to-Amplifier Coupling (Note 11)		f = 1 kHz to 20 kHz, T_A = 25°C (Input Referred)			−120			−120			−120		dB
Output Current	Source	V_{IN}^+ = 1V, V_{IN}^- = 0V, V^+ = 15V, V_O = 2V, T_A = 25°C		20	40		20	40		20	40		mA
	Sink	V_{IN}^- = 1V, V_{IN}^+ = 0V, V^+ = 15V, V_O = 2V, T_A = 25°C		10	20		10	20		10	20		mA
		V_{IN}^- = 1V, V_{IN}^+ = 0V, V^+ = 15V, V_O = 200 mV, T_A = 25°C		12	50		12	50		12	50		µA
Short Circuit to Ground		(Note 5) V^+ = 15V, T_A = 25°C			40	60		40	60		40	60	mA
Input Offset Voltage		(Note 8)				7			9			10	mV
V_{OS} Drift		R_S = 0Ω			7			7			7		µV/°C
Input Offset Current		$I_{IN(+)} - I_{IN(-)}$, V_{CM} = 0V				100			150		45	200	nA
I_{OS} Drift		R_S = 0Ω			10			10			10		pA/°C
Input Bias Current		$I_{IN(+)}$ or $I_{IN(-)}$			40	300		40	500		40	500	nA
Input Common-Mode Voltage Range (Note 10)		V^+ = +30V (LM2902, V^+ = 26V)		0		V^+−2	0		V^+−2	0		V^+−2	V
Large Signal Voltage Gain		V^+ = +15V (V_O Swing = 1V to 11V) R_L ≥ 2 kΩ		25			15			15			V/mV
Output Voltage Swing	V_{OH}	V^+ = 30V	R_L = 2 kΩ	26			26			22			V
		(LM2902, V^+ = 26V)	R_L = 10 kΩ	27	28		27	28		23	24		
	V_{OL}	V^+ = 5V, R_L = 10 kΩ			5	20		5	20		5	100	mV
Output Current	Source	V_O = 2V	V_{IN}^+ = +1V, V_{IN}^- = 0V, V^+ = 15V	10	20		10	20		10	20		mA
	Sink		V_{IN}^- = +1V, V_{IN}^+ = 0V, V^+ = 15V	5	8		5	8		5	8		

Note 4: For operating at high temperatures, the LM324/LM324A/LM2902 must be derated based on a +125°C maximum junction temperature and a thermal resistance of 88°C/W which applies for the device soldered in a printed circuit board, operating in a still air ambient. The LM224/LM224A and LM124/LM124A can be derated based on a +150°C maximum junction temperature. The dissipation is the total of all four amplifiers — use external resistors, where possible, to allow the amplifier to saturate or to reduce the power which is dissipated in the integrated circuit.

Note 5: Short circuits from the output to V^+ can cause excessive heating and eventual destruction. When considering short circuits to ground, the maximum output current is approximately 40 mA independent of the magnitude of V^+. At values of supply voltage in excess of +15V, continuous short-circuits can exceed the power dissipation ratings and cause eventual destruction. Destructive dissipation can result from simultaneous shorts on all amplifiers.

Note 6: This input current will only exist when the voltage at any of the input leads is driven negative. It is due to the collector-base junction of the input PNP transistors becoming forward biased and thereby acting as input diode clamps. In addition to this diode action, there is also lateral NPN parasitic transistor action.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Characteristics (Continued)

on the IC chip. This transistor action can cause the output voltages of the op amps to go to the V^+ voltage level (or to ground for a large overdrive) for the time duration that an input is driven negative. This is not destructive and normal output states will re-establish when the input voltage, which was negative, again returns to a value greater than −0.3V (at 25°C).

Note 7: These specifications are limited to $-55°C \le T_A \le +125°C$ for the LM124/LM124A. With the LM224/LM224A, all temperature specifications are limited to $-25°C \le T_A \le +85°C$, the LM324/LM324A temperature specifications are limited to $0°C \le T_A \le +70°C$, and the LM2902 specifications are limited to $-40°C \le T_A \le +85°C$.

Note 8: V_O = 1.4V, R_S = 0Ω with V^+ from 5V to 30V; and over the full input common-mode range (0V to V^+ − 1.5V) for LM2902, V^+ from 5V to 26V.

Note 9: The direction of the input current is out of the IC due to the PNP input stage. This current is essentially constant, independent of the state of the output so no loading change exists on the input lines.

Note 10: The input common-mode voltage of either input signal voltage should not be allowed to go negative by more than 0.3V (at 25°C). The upper end of the common-mode voltage range is V^+ − 1.5V (at 25°C), but either or both inputs can go to +32V without damage (+26V for LM2902), independent of the magnitude of V^+.

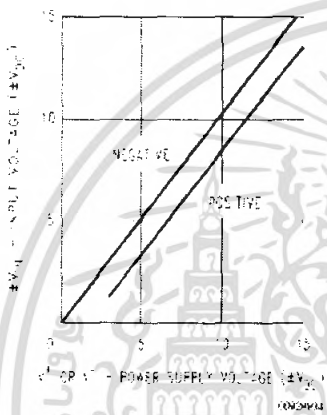
Note 11: Due to proximity of external components, insure that coupling is not originating via stray capacitance between these external parts. This typically can be detected as this type of capacitance increases at higher frequencies.

Note 12: Refer to RETS124AX for LM124A military specifications and refer to RETS124X for LM124 military specifications.

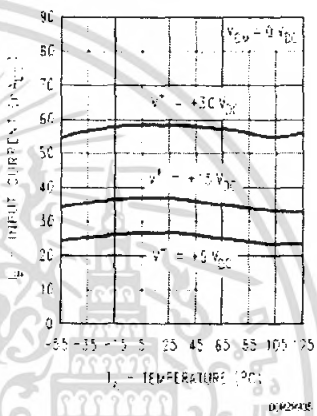
Note 13: Human body model, 1.5 kΩ in series with 100 pF.

Typical Performance Characteristics

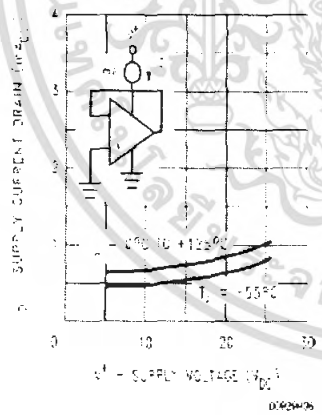
Input Voltage Range



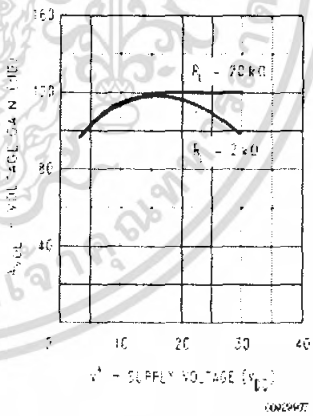
Input Current



Supply Current



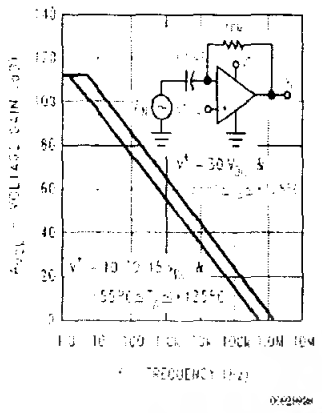
Voltage Gain



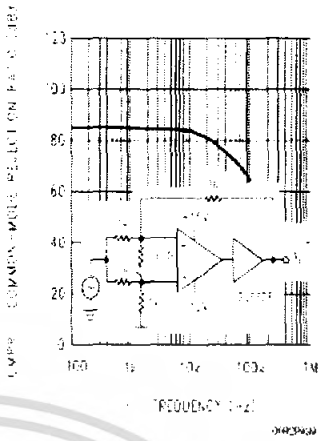
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Performance Characteristics (Continued)

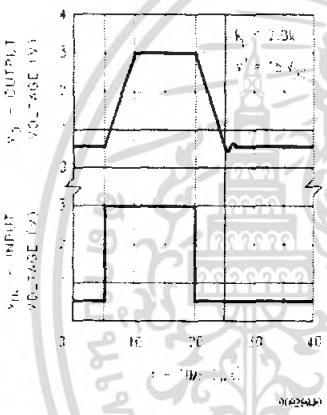
Open Loop Frequency Response



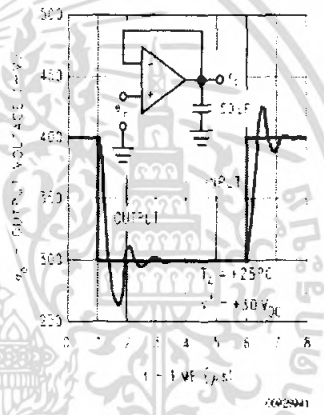
Common Mode Rejection Ratio



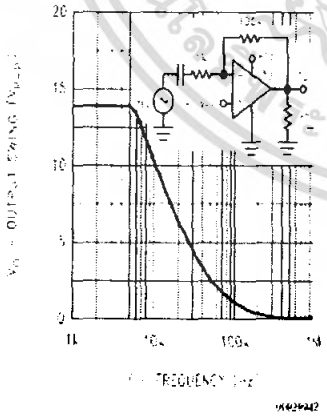
Voltage Follower Pulse Response



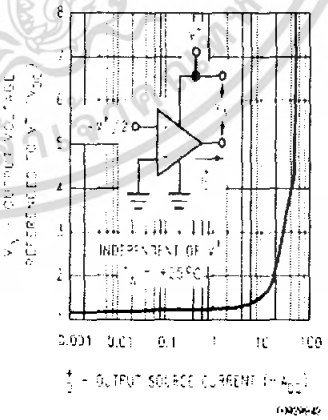
Voltage Follower Pulse Response (Small Signal)



Large Signal Frequency Response



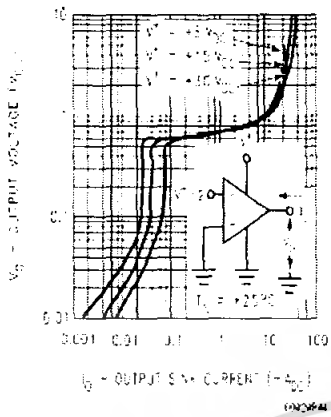
Output Characteristics Current Sourcing



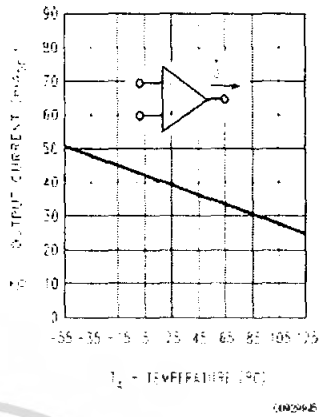
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Performance Characteristics (Continued)

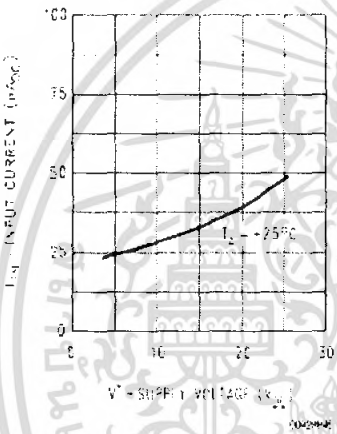
Output Characteristics
Current Sinking



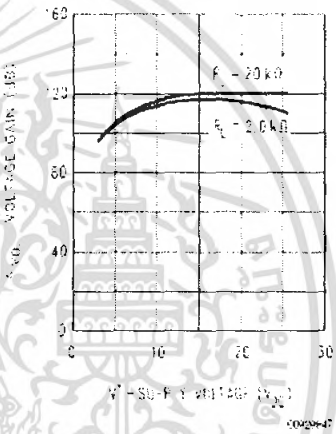
Current Limiting



Input Current (LM2902 only)



Voltage Gain (LM2902 only)



Application Hints

The LM124 series are op amps which operate with only a single power supply voltage, have true-differential inputs, and remain in the linear mode with an input common-mode voltage of 0 V_{DC}. These amplifiers operate over a wide range of power supply voltage with little change in performance characteristics. At 25 C amplifier operation is possible down to a minimum supply voltage of 2.3 V_{DC}.

The pinouts of the package have been designed to simplify PC board layouts. Inverting inputs are adjacent to outputs for all of the amplifiers and the outputs have also been placed at the corners of the package (pins 1, 7, 8, and 14).

Precautions should be taken to insure that the power supply for the integrated circuit never becomes reversed in polarity or that the unit is not inadvertently installed backwards in a test socket as an unlimited current surge through the resulting forward diode within the IC could cause fusing of the internal conductors and result in a destroyed unit.

Large differential input voltages can be easily accommodated and, as input differential voltage protection diodes are not needed, no large input currents result from large differential input voltages. The differential input voltage may be larger than V^+ without damaging the device. Protection should be provided to prevent the input voltages from going negative more than -0.3 V_{DC} (at 25 C). An input clamp diode with a resistor to the IC input terminal can be used.

To reduce the power supply drain, the amplifiers have a class A output stage for small signal levels which converts to class B in a large signal mode. This allows the amplifiers to both source and sink large output currents. Therefore both NPN and PNP external current boost transistors can be used to extend the power capability of the basic amplifiers. The output voltage needs to raise approximately 1 diode drop above ground to bias the on-chip vertical PNP transistor for output current sinking applications.

For ac applications, where the load is capacitively coupled to the output of the amplifier, a resistor should be used, from the output of the amplifier to ground to increase the class A bias current and prevent crossover distortion.

Where the load is directly coupled, as in dc applications, there is no crossover distortion.

Capacitive loads which are applied directly to the output of the amplifier reduce the loop stability margin. Values of 50 pF can be accommodated using the worst-case non-inverting unity gain connection. Large closed loop gains or resistive isolation should be used if larger load capacitance must be driven by the amplifier.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Application Hints (Continued)

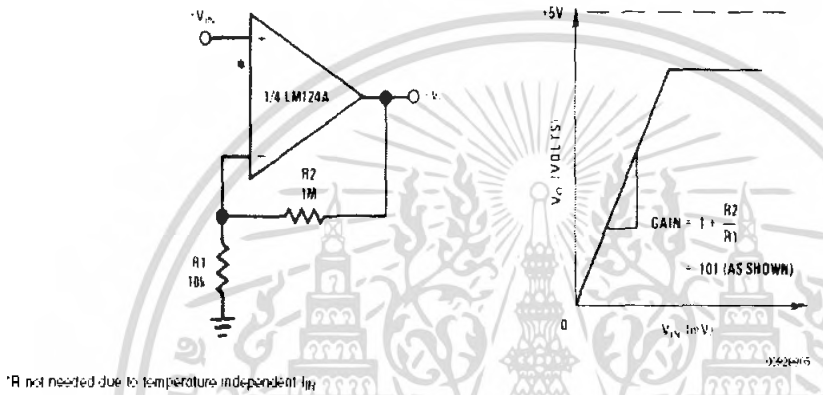
The bias network of the LM124 establishes a drain current which is independent of the magnitude of the power supply voltage over the range of from 3 V_{DC} to 30 V_{DC}.

Output short circuits either to ground or to the positive power supply should be of short time duration. Units can be destroyed, not as a result of the short circuit current causing metal fusing, but rather due to the large increase in IC chip dissipation which will cause eventual failure due to excessive junction temperatures. Putting direct short-circuits on more than one amplifier at a time will increase the total IC power dissipation to destructive levels, if not properly protected with external dissipation limiting resistors in series with the output leads of the amplifiers. The larger value of output source current which is available at 25 C provides a larger output current capability at elevated temperatures (see typical performance characteristics) than a standard IC op amp.

The circuits presented in the section on typical applications emphasize operation on only a single power supply voltage. If complementary power supplies are available, all of the standard op amp circuits can be used. In general, introducing a pseudo-ground (a bias voltage reference of V+/2) will allow operation above and below this value in single power supply systems. Many application circuits are shown which take advantage of the wide input common-mode voltage range which includes ground. In most cases, input biasing is not required and input voltages which range to ground can easily be accommodated.

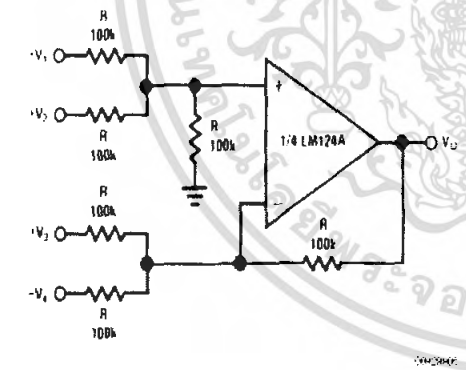
Typical Single-Supply Applications (V^+ = 5.0 V_{DC})

Non-Inverting DC Gain (0V Input = 0V Output)



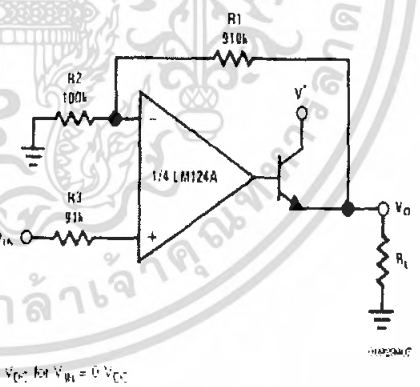
*R not needed due to temperature independent I_{IN}

DC Summing Amplifier
(V_{IN}'S ≥ 0 V_{DC} and V_O ≥ V_{DC})



Where: $V_O = V_1 + V_2 - V_3 - V_4$
($V_1 + V_2$) ≥ ($V_3 + V_4$) to keep V_O > 0 V_{DC}

Power Amplifier

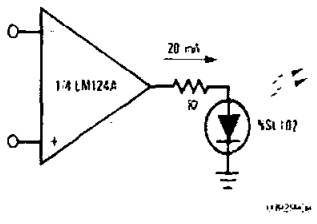


V_O = 0 V_{DC} for V_{IN} = 0 V_{DC}
A_V = 10

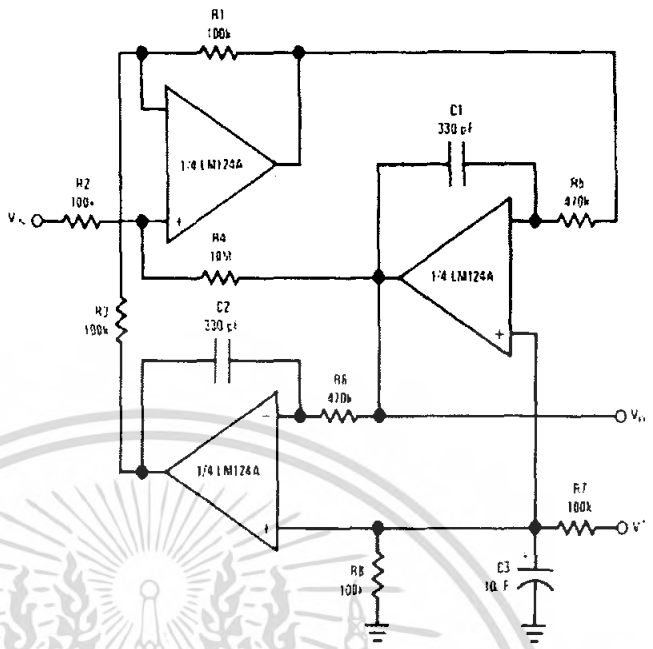
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Single-Supply Applications ($V^+ = 5.0\ V_{DC}$) (Continued)

LED Driver

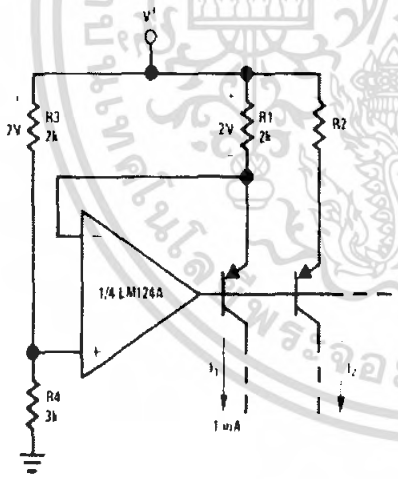


"Bi-QUAD" RC Active Bandpass Filter

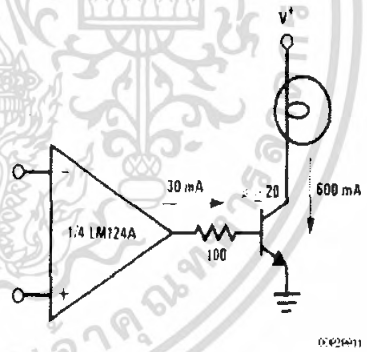


f_o = 1 kHz
Q = 50
A_V = 100 (40 dB)

Fixed Current Sources



Lamp Driver



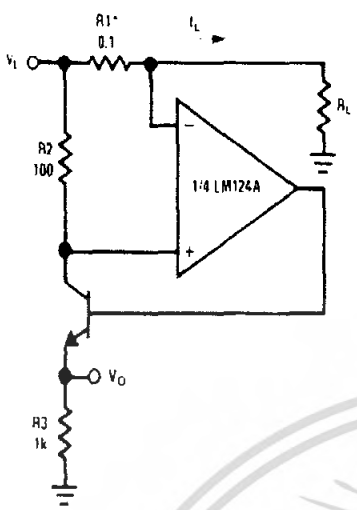
$$I_2 = \left(\frac{R1}{R2}\right) I_1$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Single-Supply Applications

($V^+ = 5.0\ V_{DC}$) (Continued)

Current Monitor

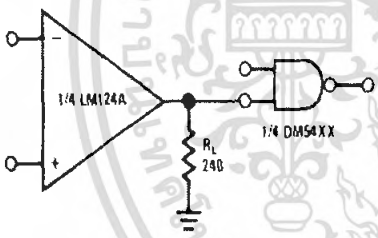


$V_O = \frac{(1\ V/A)(I_L)}{1A}$

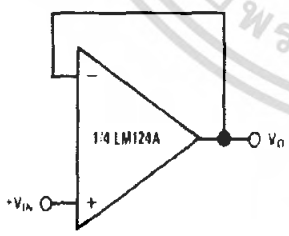
V_L V^+ 2V

*Increase R1 for I_L small

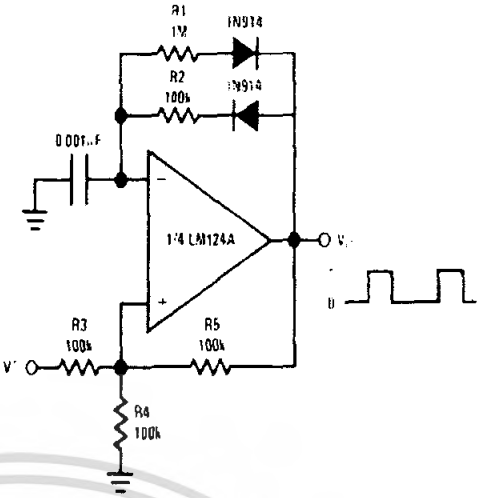
Driving TTL



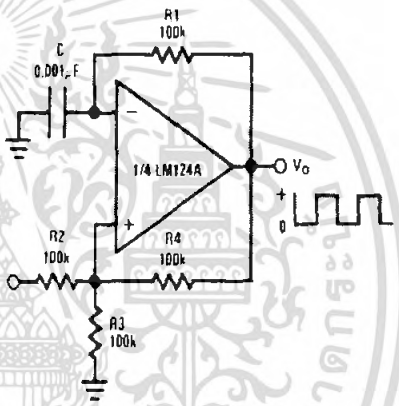
Voltage Follower



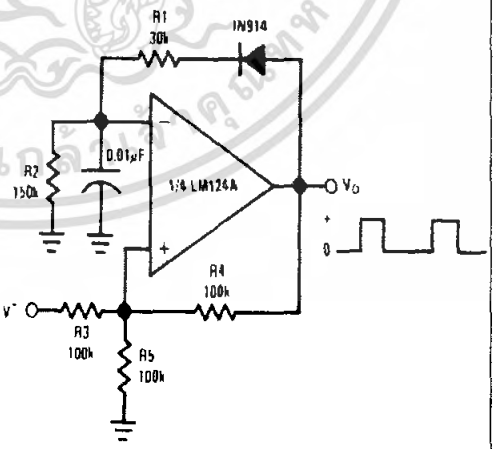
Pulse Generator



Squarewave Oscillator



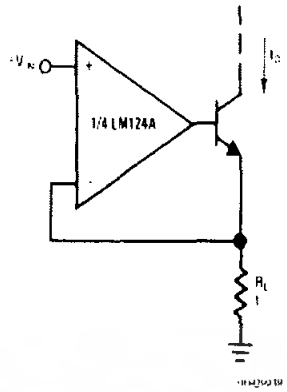
Pulse Generator



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Single-Supply Applications ($V^+ = 5.0\ V_{DC}$) (Continued)

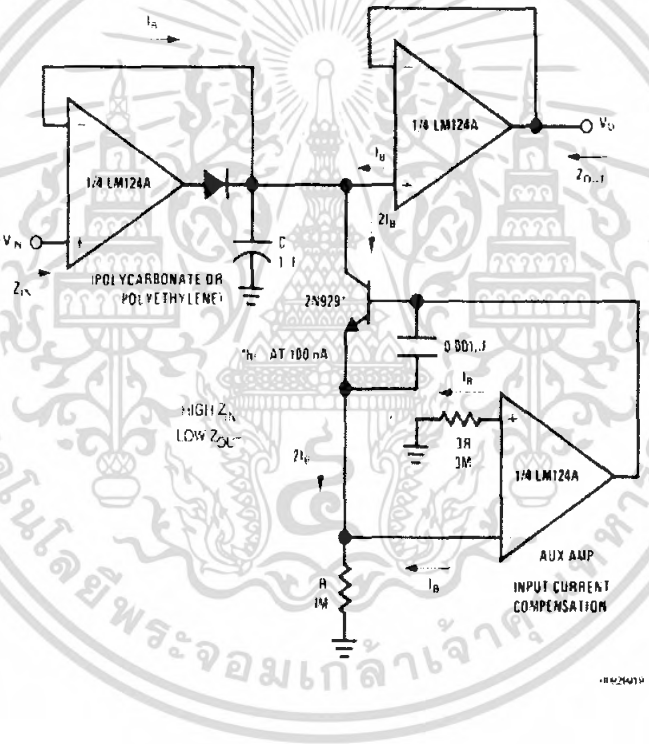
High Compliance Current Sink



$I_O = 1$ amp/volt V_{IN}

Increase R_E for I_O small

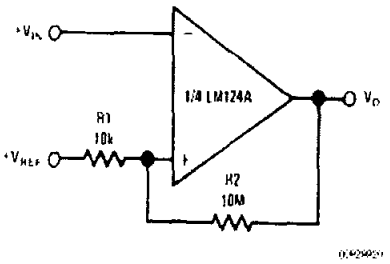
Low Drift Peak Detector



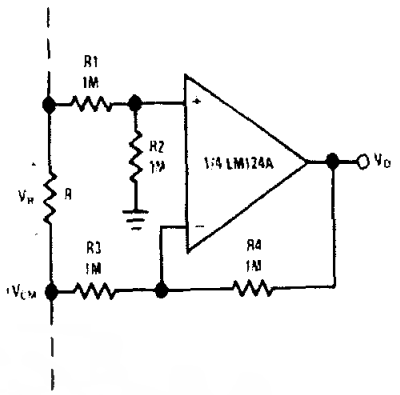
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Single-Supply Applications (V^+ = 5.0 V_{DC}) (Continued)

Comparator with Hysteresis

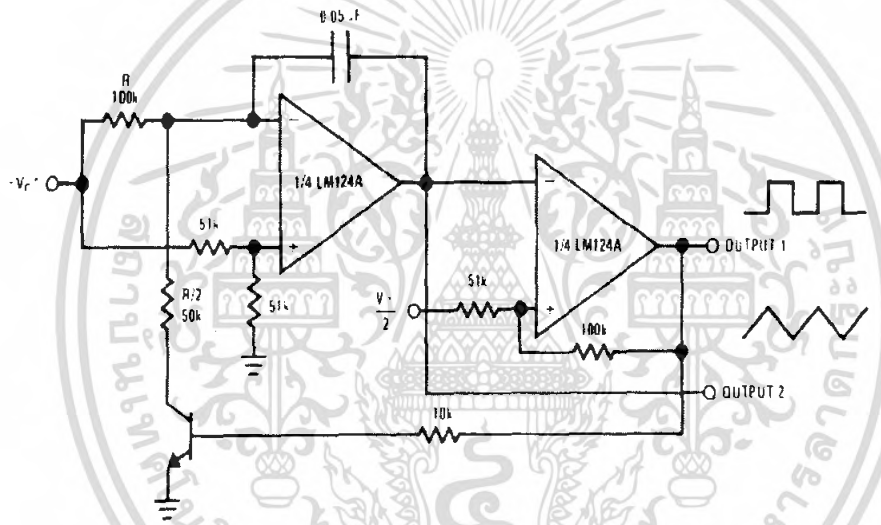


Ground Referencing a Differential Input Signal



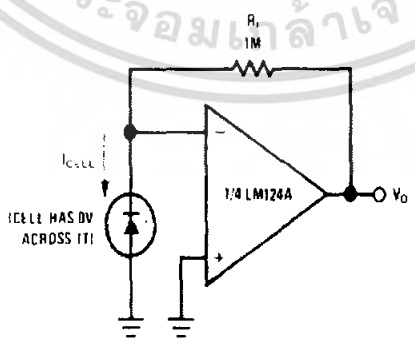
$V_O = V_R$

Voltage Controlled Oscillator Circuit



*Wide control voltage range: 0 $V_{DC} \leq V_C \leq 2$ ($V^+ - 1.5$ V_{DC})

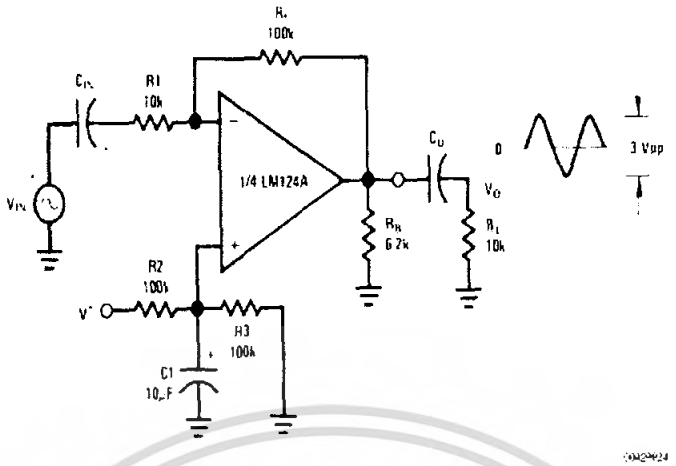
Photo Voltaic-Cell Amplifier



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

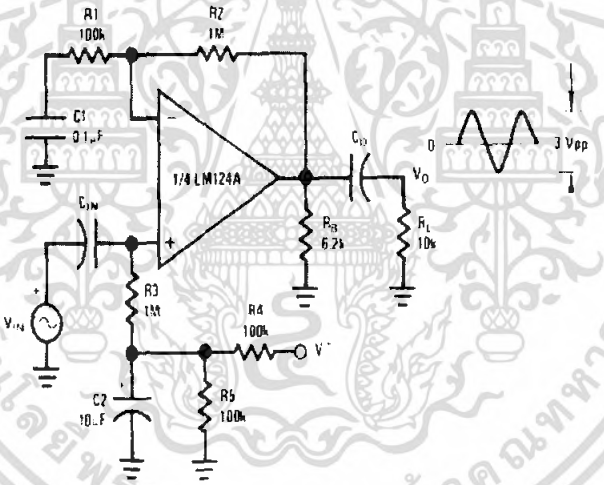
Typical Single-Supply Applications (V^+ = 5.0 V_{DC}) (Continued)

AC Coupled Inverting Amplifier



$$A_V = \frac{R_4}{R_1} \text{ (As shown } A_V = 10)$$

AC Coupled Non-Inverting Amplifier



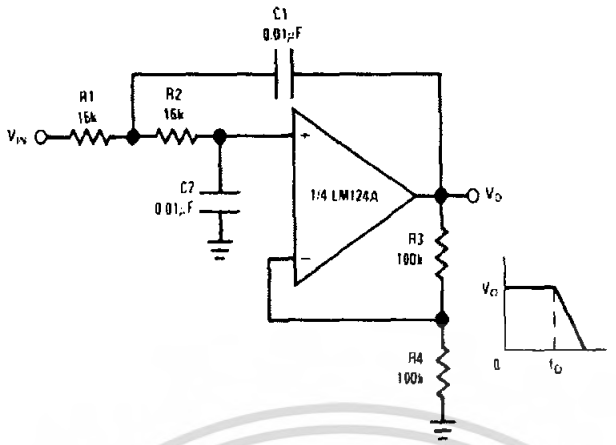
$$A_V = -\frac{R_2}{R_1}$$

A_V = −1 (As shown)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

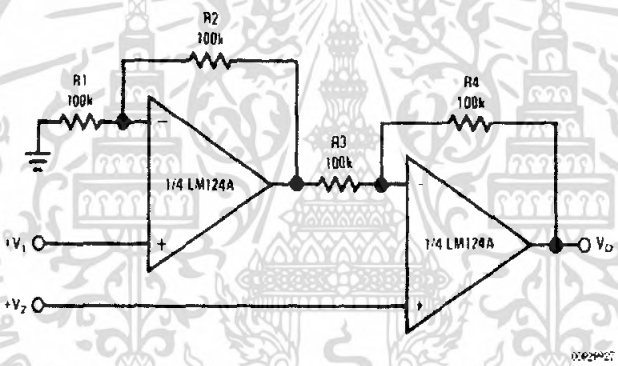
Typical Single-Supply Applications (V^+ = 5.0 V_{DC}) (Continued)

DC Coupled Low-Pass RC Active Filter



f_O = 1 kHz
Q = 1
A_V = 2

High Input Z, DC Differential Amplifier



For $\frac{R1}{R2} = \frac{R4}{R3}$ (CMRR depends on this resistor ratio match)

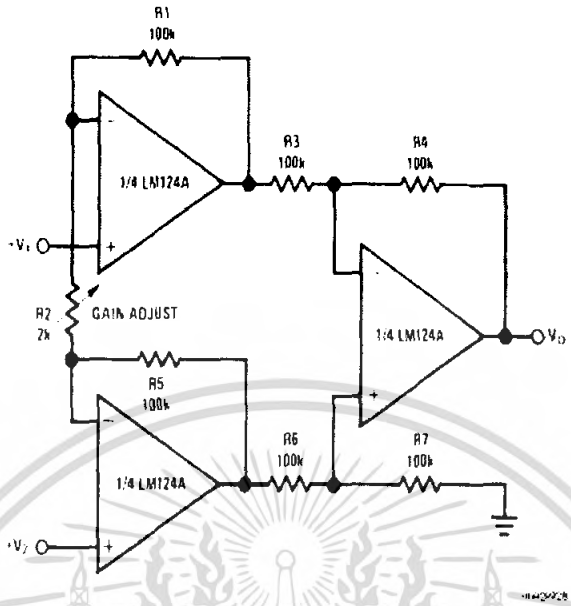
$V_O = 1 + \frac{R4}{R3}(V_2 - V_1)$

As shown $V_O = 2(V_2 - V_1)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Single-Supply Applications (V^+ = 5.0 V_{DC}) (Continued)

High Input Z Adjustable-Gain DC Instrumentation Amplifier

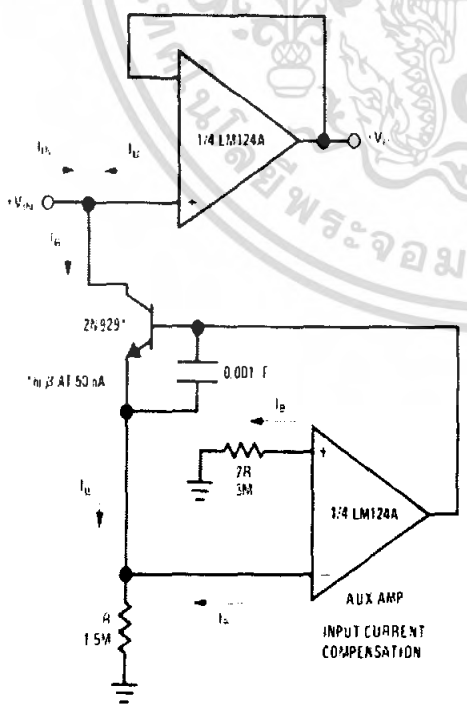


If R1 = R5 & R3 = R4 = R6 = R7 (CMRR depends on match)

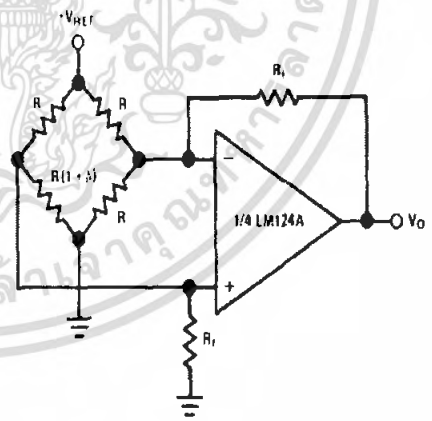
$$V_O = \left(1 + \frac{2R1}{R2}\right)(V_2 - V_1)$$

As shown V_O = 101 (V_2 − V_1)

Using Symmetrical Amplifiers to Reduce Input Current (General Concept)



Bridge Current Amplifier



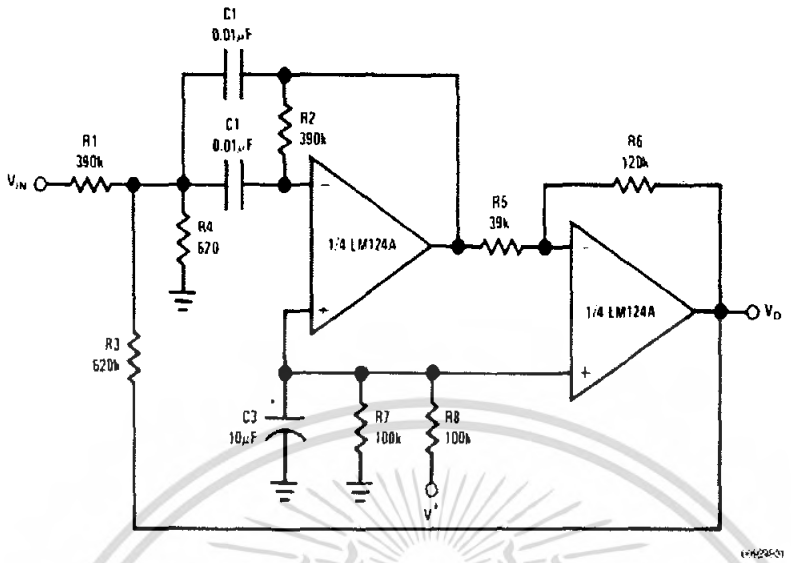
For δ << 1 and RF = R

$$V_O = V_{REF}\left(\frac{\delta}{2}\right)\frac{R_f}{R}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Single-Supply Applications (V^+ = 5.0 V_{DC}) (Continued)

Bandpass Active Filter



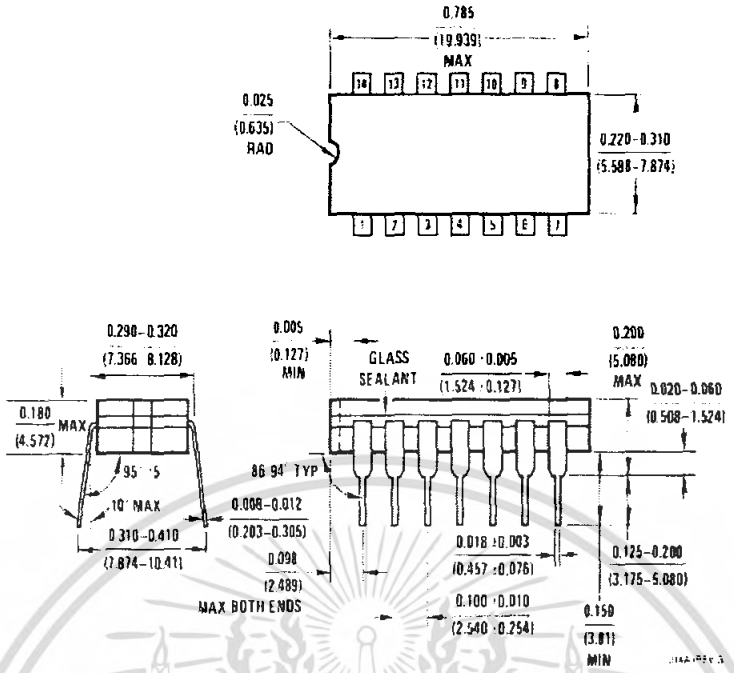
f_O = 1 kHz

Q = 25

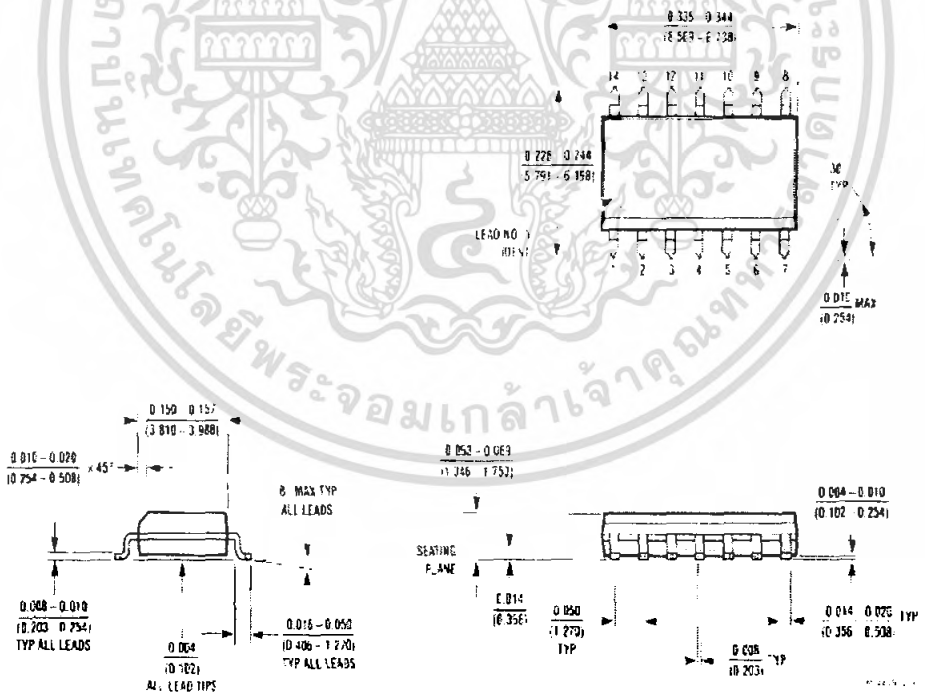
www.national.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Physical Dimensions inches (millimeters) unless otherwise noted



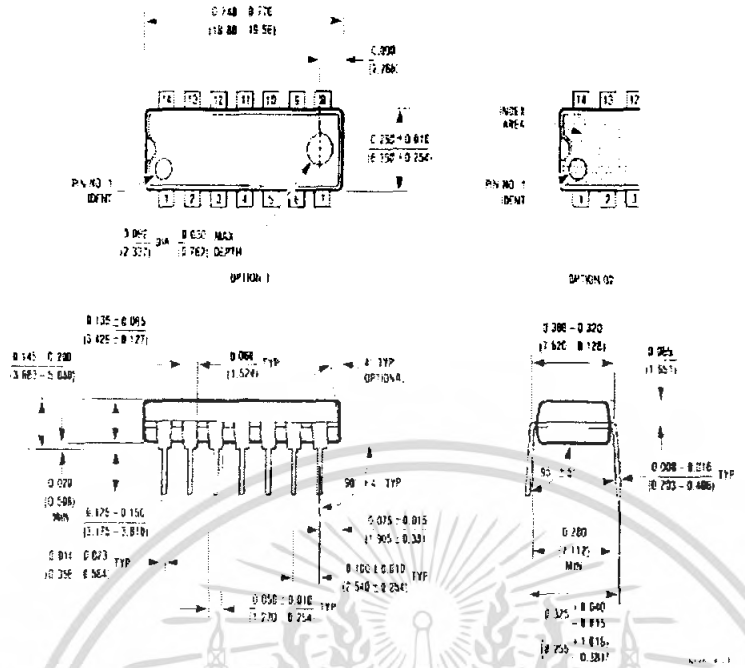
Ceramic Dual-In-Line Package (J)
Order Number JL124ABCA, JL124BCA, JL124ASCA, JL124SCA, LM124J, LM124AJ, LM124AJ/883, LM124J/883, LM224J, LM224AJ or LM324J
NS Package Number J14A



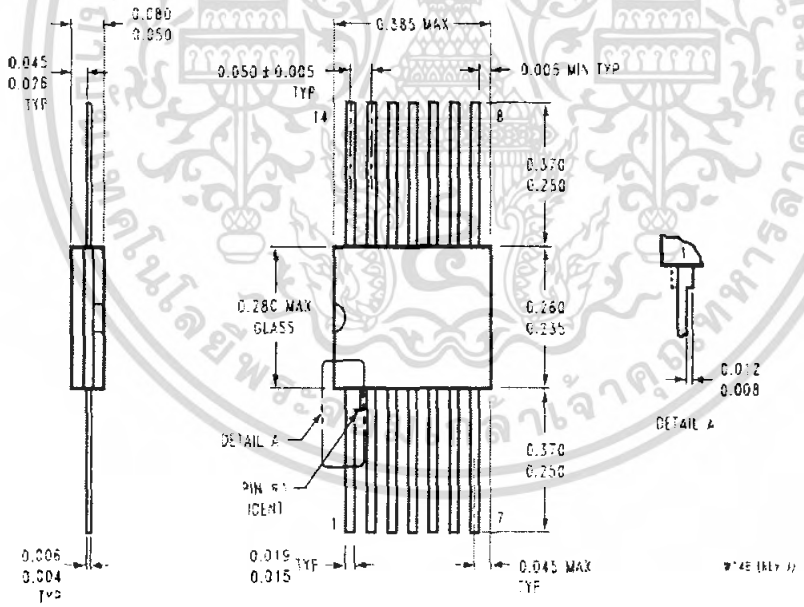
MX S.O. Package (M)
Order Number LM324M, LM324MX, LM324AM, LM324AMX, LM2902M or LM2902MX
NS Package Number M14A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



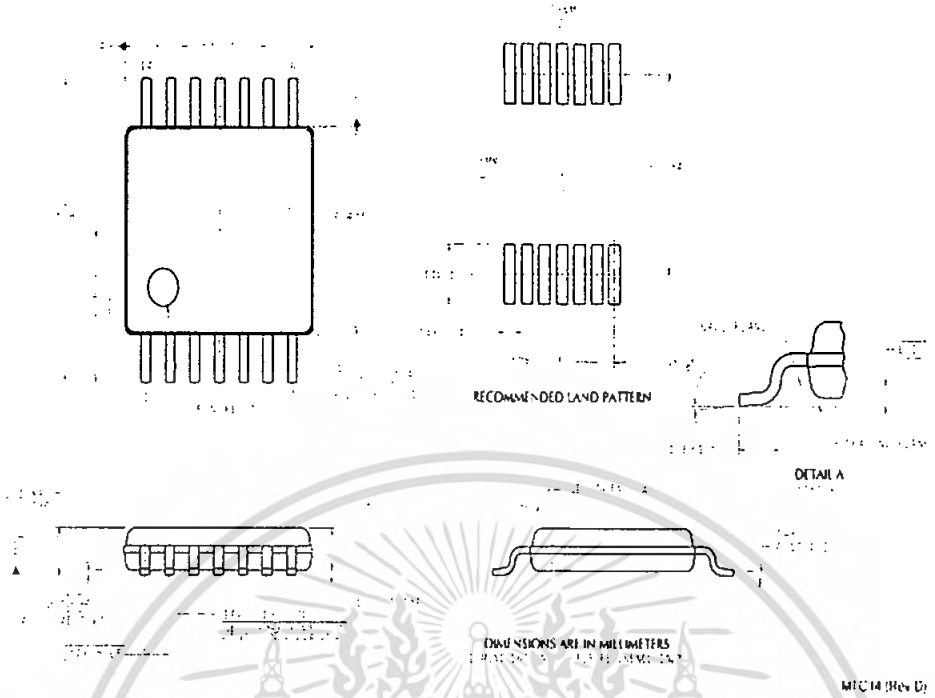
Molded Dual-In-Line Package (N)
Order Number LM324N, LM324AN or LM2902N
NS Package Number N14A



Ceramic Flatpak Package
Order Number JL124ABDA, JL124ABZA, JL124ASDA, JL124BDA, JL124BZA, JL124SDA, LM124AW/883, LM124AWG/883, LM124W/883 or LM124WG/883
NS Package Number W14B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



14-Pin TSSOP
Order Number LM324MT or LM324MTX
NS Package Number MTC14

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

BANNED SUBSTANCE COMPLIANCE

National Semiconductor certifies that the products and packing materials meet the provisions of the Customer Products Stewardship Specification (CSP-9-111C2) and the Banned Substances and Materials of Interest Specification (CSP-9-111S2) and contain no "Banned Substances" as defined in CSP-9-111S2.

National Semiconductor
Americas Customer
Support Center
Email: new.feedback@nsc.com
Tel: 1-800-272-9959

National Semiconductor
Europe Customer Support Center
Fax: +49 (0) 180-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 69 9508 6208
English Tel: +44 (0) 870 24 0 2171
Français Tel: +33 (0) 1 41 91 8790

National Semiconductor
Asia Pacific Customer
Support Center
Email: ap.support@nsc.com

National Semiconductor
Japan Customer Support Center
Fax: 81-3-5639-7507
Email: jpn.feedback@nsc.com
Tel: 81-3-5639-7560

www.national.com

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้