

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โปรแกรมควบคุมเครื่องจักรแบบสามแกน

CAM FOR 3D CNC



โดย
นาย วศิน พงศ์ไพบุลย์ศิริ
นาย สุระสิทธิ์ สันติเกษมสุข
นาย เอกนถุน ทอนหม

เลขหมู่.....
เลขทะเบียน.....103063
วัน,เดือน,ปี.....27 ส.ค. 2552

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมแมคคาทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์การศึกษาปี 2551

ภาควิชาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมควบคุมเครื่องจักรแบบสามแกน

CAM FOR 3D CNC

ผู้จัดทำ นาย วสิน พงศ์ไพบุลย์ศิริ 48010790

นาย สุระสิทธิ์ สันติเกษมสุข 48010829

นาย เอกนถุน ทอนทม 48011141

.....อาจารย์ที่ปรึกษา

(อาจารย์เทพจิตร เขยโกคา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุมเครื่องจักรแบบสามแกน

โดย

นาย วศิน พงศ์ไพบุลย์ศิริ 48010790

นาย สุระสิทธิ์ สันติเกษมสุข 48010829

นาย เอกนถน ทอนทม 48011141

อาจารย์ที่ปรึกษา

อาจารย์ เทพจิตร เชยโศภ

ปีการศึกษา 2551

บทคัดย่อ

การทำโครงการเขียนโปรแกรมเพื่อควบคุมเครื่องจักร CNC เริ่มมาจากที่
นักศึกษาเล็งเห็นความสำคัญของเครื่องจักร CNC ที่ใช้อย่างแพร่หลายในอุตสาหกรรม
ปัจจุบัน และก่อกำเนิดที่ภาควิชาแมคคาทรอนิกส์มีเครื่องจักร CNC ให้นักศึกษาได้ใช้กัน
อย่างเพียงพอ แต่ทั้งนี้ทางภาควิชาต้องซื้อลิขสิทธิ์ของโปรแกรมจากบริษัทข้างนอก
เพื่อให้นักศึกษาได้ทำการศึกษา ผู้จัดทำจึงเล็งเห็นทั้งประโยชน์ในด้านการลดต้นทุนการ
จัดซื้อของทางภาควิชาแมคคาทรอนิกส์ และ เป็นโอกาสในการศึกษาการทำงานของ
เครื่องจักร CNC อย่างลึกซึ้ง

CAM FOR 3D CNC

By

Vasin Pongpaiboonsiri 48010790

Surasit Suntikasemsuk 48010829

Eknarin Thonthom 48011141

Advisor

Mr. Thepajit Cheypoca

Academic year 2008

ABSTRACT

The CNC Programming project is initiated by the importance of the wide spread of CNC Machine using .

Moreover the Mechatronics Major of King Mongkut Institute of Technology

Ladkrabang have to purchase license from business sectors. Therefore this project is

from intention to reduce cost of purchasing license and to study and understand CNC process in dept.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำปริญญานิพนธ์ฉบับนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความช่วยเหลือเป็นอย่างดี จากอาจารย์เทพจิตร เชย โภคา ที่ได้กรุณาให้คำปรึกษาแนะนำที่ดีมาโดยตลอดแต่ต้น รวมทั้งเอื้อเฟื้ออุปกรณ์ที่จำเป็น และความช่วยเหลืออื่นๆ ที่เป็นประโยชน์ต่อโครงการ ผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูง

ขอบคุณเพื่อนๆทุกคนที่ทำให้กำลังใจ สนับสนุนอุปกรณ์ที่ขาดเหลือ คอยกระตุ้น รวมทั้งถามไถ่ความคืบหน้าของโครงการอยู่เสมอ

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณ บิดา มารดา และครอบครัว รวมถึงอาจารย์ที่เคยประสิทธิ์ประสาทวิชา หากทว่าโครงการนี้มีประโยชน์คุณความดีแต่ประการใด ผู้จัดทำขอยกผลแห่งความดีให้แก่บุคคลเหล่านี้

ผู้จัดทำ

นาย วสิน

พงศ์ไพบุลย์ศิริ

นาย สุระสิทธิ์

สันติเกษมสุข

นาย เอกนถุน

ทอนทม

สารบัญ

	หน้า
บทคัดย่อ	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญภาพ	VII
บทที่ 1 บทนำ	
1.1 คำนำ	1
1.2 จุดประสงค์	2
1.3 เป้าหมาย	2
บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง	
2.1 การศึกษาการทำงานของซีเอ็นซี	4
2.2 องค์ประกอบของเครื่องซีเอ็นซีที่ควบคุมได้	5
2.2.1 แนวแกนป้อน	5
2.2.2 การจับป้อน	6
2.2.3 มอเตอร์	6
2.2.4 บอลล์สกรู	9
2.3 สเต็ปมอเตอร์	10
2.3.1 ชนิดและโครงสร้างของสเต็ปมอเตอร์	10
2.3.2 วงจรขับสเต็ปมอเตอร์	17
2.4 เทคโนโลยี CAD CAM	22

สารบัญ(ต่อ)

หน้า

บทที่ 3 โปรแกรม K_EM_CAM

3.1 คอนเซ็ปต์ของโปรแกรม	24
3.2 แนวความคิดการทำงานของโปรแกรม	24
3.3 การเชื่อมต่อผ่าน พอร์ตขนาน	25
3.4 หลักการทำงานของโปรแกรม(โฟลว์ชาร์ท)	26

บทที่ 4 การทดลองและผลการทดลอง

บทที่ 5 บทวิจารณ์และผลสรุป

5.1 สรุปผลการทดลอง	30
5.2 วิจารณ์ผลการทดลอง	30
5.3 ปัญหาที่พบในการทำโครงงานนี้	30
5.4 แนวทางในการพัฒนา	31

ภาคผนวก

เอกสารอ้างอิง

สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 1.1 ตารางเวลาการทำงาน	3
ตารางที่ 2.1 ตารางแสดงมุมของโรเตอร์เทียบกับกระแสไฟฟ้าที่จ่ายแก่เฟสต่าง 8 ตำแหน่ง	13
ตารางที่ 2.2 ตารางแสดงการจ่ายกระแสให้ขั้วแม่เหล็ก	14



สารบัญภาพ

รูปที่	หน้า
รูปที่ 2.1 มอเตอร์กระแสตรง ที่มีขายตามท้องตลาดอย่างแพร่หลาย	7
รูปที่ 2.2 มอเตอร์แบบขั้น (stepping motor)	8
รูปที่ 2.3 มอเตอร์แบบกระแสสลับ	8
รูปที่ 2.4 ชิ้นส่วนบอลล์สกรู ซึ่งเป็นหัวใจของระบบส่งกำลังเครื่องจักรซีเอ็นซี	9
รูปที่ 2.5 สเต็ปมอเตอร์แบบมีสาย 5 เส้น	10
รูปที่ 2.6 มอเตอร์แบบมีสาย 6 เส้น	11
รูปที่ 2.7 สเต็ปมอเตอร์หลายแบบไบโพลาร์	11
รูปที่ 2.8 แสดงภาพถ่ายโครงสร้างสเต็ปมอเตอร์	12
รูปที่ 2.9 (ก) โครงสร้าง (ข) วงจรเทียบเท่า (equivalent circuit) ของมอเตอร์ ชนิด 4 ขด	13
รูปที่ 2.10 วงจรสเต็ปมอเตอร์ ชนิดมีสาย 6 เส้น	15
รูปที่ 2.11 วงจรสเต็ปมอเตอร์ชนิดมีสาย 5 เส้น	16
รูปที่ 2.12 แสดงวิธีขับเอาท์พุทของสเต็ปมอเตอร์	17
รูปที่ 2.13 วงจรสมมูลย์ (equivalent circuit) ของสเต็ปมอเตอร์	18
รูปที่ 2.14 ป้องกันอันตรายต่อทรานซิสเตอร์ด้วยไดโอดชัฟเฟรสเซอร์	18
รูปที่ 2.15 ป้องกันอันตรายต่อทรานซิสเตอร์ด้วยไดโอดและตัวต้านทาน	
ชัฟเฟรสเซอร์	19

สารบัญภาพ(ต่อ)

รูปที่	หน้า
รูปที่ 2.16 ป้องกันอันตรายโดยต่อทรานซิสเตอร์ด้วย	
ซีเนอร์ไดโอดซัพเพรสเซอร์	19
รูปที่ 2.17 ป้องกันอันตรายโดยต่อทรานซิสเตอร์ด้วยตัวเก็บประจุซัพเพรสเซอร์	20
รูปที่ 2.18 วงจรสแตมป์มอเตอร์ต่อกับ Z-80 ซึ่งสามารถดัดแปลงไปใช้กับ	
ไมโครโปรเซสเซอร์เบอร์อื่น ได้	21
รูปที่ 3.1 การทำงานของเครื่อง CNC	24
รูปที่ 3.2 แผนผังพอร์ตขนาน	25
รูปที่ 3.3 พอร์ตขนาน	25
รูปที่ 3.4 Flow Chart ของโปรแกรม	26
รูปที่ 4.1 แสดงหน้าต่างของโปรแกรมคอส	27
รูปที่ 4.2 แสดงหน้าต่างของโปรแกรมคอสขณะผู้ใช้งานเรียกโปรแกรม	
K_EM_cam.exe	27
รูปที่ 4.3 แสดงหน้าต่างของโปรแกรม K_EM_cam.exe	28
รูปที่ 4.4 แสดงหน้าต่างของโปรแกรม K_EM_cam.exe ขณะทำการเรียกไฟล์ G-code	28
รูปที่ 4.5 แสดงหน้าต่างของโปรแกรม K_EM_cam.exe ขณะทำการเรียกไฟล์ G-code	29
รูปที่ 4.6 แสดงหน้าต่างของโปรแกรม K_EM_cam.exe ขณะทำการรันไฟล์ G-code	29

สารบัญภาพ(ต่อ)

รูปที่

หน้า

รูปที่ 4.7 แสดงหน้าต่างของโปรแกรม K_EM_cam.exe ขณะทำการรันไฟล์ G-code

29



บทที่ 1

บทนำ

1.1 คำนำ

โปรเจกต์นี้เป็นโปรเจกต์เกี่ยวกับการสร้างซอฟต์แวร์รับค่าจากไฟล์ของโปรแกรมCAD เพื่อนำไปควบคุมการทำงานของเครื่อง ซีเอ็นซี (เขียนโปรแกรมCAM)

ในปัจจุบันเครื่องซีเอ็นซีได้รับความนิยมอย่างแพร่หลาย เนื่องจากสามารถทำงานได้อย่างแม่นยำ รวดเร็ว และยังช่วยลดปัญหามาตรฐานงานที่อาจบกพร่องจากช่างฝีมือที่เป็นคนได้

นักศึกษาได้เล็งเห็นของประโยชน์ในการสร้างตัวโปรแกรมรับค่าไฟล์CAD เพื่อมาควบคุมซีเอ็นซี อันเนื่องจากตัวโปรแกรมนั้นหากไปซื้อตามท้องตลาดจะมีราคาที่สูงมาก อีกทั้งยังเป็นประโยชน์แก่น้องๆที่สามารถนำตัวโปรแกรมไปใช้งานในปีการศึกษาต่อไปได้

โดยนักศึกษาได้วางแผนในการทำโปรเจกต์ไว้เป็นสองส่วนคือ

- ส่วนการศึกษาซอฟต์แวร์ที่นำมาเขียนโปรแกรม
- การเขียนโปรแกรมแปลงไฟล์ที่ได้จากโปรแกรม CAD เป็นค่าพิกัด X,Y,Z เพื่อนำไปควบคุมเครื่อง CNC โดยก่อนที่จะสามารถเขียนโปรแกรมนี้ได้ นักศึกษาจำเป็นต้องทราบถึงหลักการทำงานของเครื่องซีเอ็นซี และศึกษาการเขียนโปรแกรมพื้นฐาน

ขอบพระคุณ

1.2 จุดประสงค์ของโปรเจก

- นักศึกษาได้มีความรู้ความเข้าใจเกี่ยวกับพื้นฐานการใช้คอมพิวเตอร์มาสั่งการเครื่องจักรกลในระบบควบคุม ซีเอ็นซี โดยโปรแกรม CAD และ CAM
- นักศึกษาทราบวิธีการทำงานของเครื่องซีเอ็นซี
- นักศึกษาทำการศึกษาโปรแกรมเพื่อลงมือเขียนโปรแกรมในการรับไฟล์ G-code มาสั่งการเครื่องซีเอ็นซี (เขียนโปรแกรมCAM)
- นักศึกษาลงมือเขียนโปรแกรมจริง เพื่อต่อสร้างสัญญาณควบคุมเครื่องซีเอ็นซีสามแกน ที่สถาบันฯ

1.3 เป้าหมาย

- โปรแกรมCAM เพื่อใช้ในการสั่งงานเครื่องจักรกล สามแกน ที่สถาบันฯ ใช้ได้จริง
- เป็นประโยชน์แก่น้องในปีการศึกษาถัดๆไป ในการนำไปใช้และนำโปรแกรมไปศึกษา พัฒนา

N o	Subject	July	Aug	Sep	Oct	Nov	Dec	Jan	Feb
1	เลือกโปรเจก	→							
2	วางแผนงาน แบ่งหน้าที่	→							
3	ศึกษา CNC (การทำงานของ ส่วนประกอบ)	→	→						
4	ศึกษา software CAD CAM และ G Code	→	→						
5	หาข้อมูลการทำงานของโปรแกรม CAM	→	→						
6	ศึกษา เครื่องโปรแกรมที่สกรีนได้เขียนโปรแกรมได้			→					
7	ทำรายงานและรายงานผล Project1			→					
8	เลือก เครื่องโปรแกรมที่จะใช้เขียน				→				
9	ศึกษา การใช้โปรแกรม Borland C++				→	→			
10	วางแผนเซตอัพและโครงสร้างของโปรแกรม				→	→			
11	หาข้อมูลและศึกษาลักษณะประกอบของโปรแกรม					→	→		
12	ศึกษาและศึกษาโปรแกรมที่สกรีนได้นำมาใช้งาน ได้จากแหล่งข้อมูลทั้งในและต่างประเทศ					→	→	→	
13	นำส่วนประกอบต่างๆมาศึกษาและทำแบบ ประยุกต์สร้างเป็นโปรแกรม							→	
14	ทดลองแก้ไข และปรับปรุง							→	→
15	ทดลองขับเครื่อง							→	→
16	ทำวิทยานิพนธ์ และนำเสนอ							→	→

ตารางที่ 1.1 ตารางเวลาการทำโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและความรู้ที่เกี่ยวข้อง

2.1 การศึกษาการทำงานของซีเอ็นซี

นักศึกษาได้ศึกษาเกี่ยวกับการทำงานของซีเอ็นซีเพื่อที่จะสามารถสร้างโปรแกรมที่สอดคล้องกับการทำงานของเครื่อง
ซีเอ็นซีเป็นเครื่องจักรที่คอมพิวเตอร์เข้าไปควบคุม โดยสามารถจัดการกับข้อมูล
ที่ผู้ใช้ป้อนเข้าไปในระบบ และประมวลผลข้อมูลที่ได้เพื่อนำไปใช้ควบคุมการทำงานของ
เครื่องจักรกล

เครื่องจักรกลโดยทั่วไป แทนเคลื่อนที่ทำหน้าที่จับชิ้นงาน หรือเครื่องมือตัดให้
เลื่อนไปตามรางเลื่อน โดยการใช้มอเตอร์หมุน หรือโดยการต่อกลไกอัตโนมัติ เช่น ลูก
เบี้ยวในเครื่องกลึงอัตโนมัติ ซึ่งในขณะเดียวกัน ช่วงที่ควบคุมเครื่อง ต้องปฏิบัติงานใน
หน้าที่อื่นๆ ที่จำเป็นต้องใช้ในกระบวนการนั้นๆ เช่น การเปิดปิดสวิตช์ควบคุมการหมุน
ของเพลาหัวเครื่อง เปลี่ยนอัตราการหมุนและความเร็วรอบ เปิดปิดสารหล่อเย็น ในการ
ปฏิบัติหน้าที่ต่างๆ เหล่านี้ ช่วงควบคุมจะต้องใช้ทั้งวิจารณญาณและสติปัญญา ในการ
ตัดสินใจร่วมกัน การกระทำเหล่านี้ช่วงควบคุมต้องกระทำซ้ำๆ ตลอดเวลา แม้ว่าจะเป็น
การผลิตชิ้นงานที่มีรูปทรงเหมือนเดิมทุกประการ นั้นเป็นเหตุให้เกิดข้อผิดพลาดได้

ส่วนเครื่องจักรซีเอ็นซี การเคลื่อนที่ต่างๆ ที่จำเป็นในการผลิตชิ้นงานจะทำงาน
โดยอัตโนมัติด้วยตัวเครื่องเอง ตามรหัสข้อมูลตัวเลขที่ป้อนใช้กับระบบควบคุมใน
รูปแบบของรหัส(จีโค้ด)

เพราะฉะนั้นความแตกต่างของระบบควบคุมแบบเครื่องจักรกล กับ แบบ
เครื่องจักรที่ใช้ระบบซีเอ็นซี คือ การตัดสินใจในของเครื่องที่ควบคุมด้วยระบบซีเอ็นซีจะ
ตัดสินใจในการกำหนดขั้นตอนต่างๆ ในการทำงานงาน เพียงครั้งแรกครั้งเดียว กลางคือ
การทำซ้ำในครั้งต่อไป จะมีวิธีการตัดสินใจที่เหมือนกัน ผลคือ งานที่ได้มีความเป็น
มาตรฐาน และยังช่วยลดเวลาในการตรวจสอบชิ้นงานได้อีกด้วย

การทำงานของเครื่องจักรซีเอ็นซี คือ เมื่อระบบควบคุมอ่านโปรแกรมที่ป้อนเข้า
ไปแล้ว ก็จะนำไปควบคุมเครื่องจักรให้ทำงาน แต่เนื่องจากเครื่องจักรซีเอ็นซีไม่มีมือ

สำหรับหมุนให้แท่นต่างๆเคลื่อนที่ตามแนวแกนได้ ดังนั้นจึงมีมอเตอร์ ประกอบอยู่ทั้งสามแกน

โดยระบบจะอ่านโปรแกรมควบคุมและเปลี่ยนรหัสโปรแกรมนั้นเป็นสัญญาณทางไฟฟ้าเพื่อควบคุมให้มอเตอร์ทำงาน แต่เนื่องจากสัญญาณควบคุมที่ได้อกจากระบบมีกำลังน้อย ไม่สามารถไปหมุนมอเตอร์ได้ ดังนั้น สัญญาณควบคุมจึงถูกส่งไปยังภาคขยายของระบบขับ เพื่อที่จะส่งสัญญาณไปขับมอเตอร์อีกทีหนึ่ง

2.2 องค์ประกอบของเครื่องซีเอ็นซีที่ควบคุมได้

องค์ประกอบหรือชิ้นส่วนของเครื่องจักรที่ทพหน้าที่เคลื่อนที่พาชิ้นงานหรืออุปกรณ์เข้า เลื่อนเนื้อชิ้นงาน จะถูกควบคุมด้วยโปรแกรมซีเอ็นซี

ช่างชำนาญงานที่ทำหน้าที่ควบคุมการทำงานของเครื่องจักรซีเอ็นซี จะต้องมีความคุ้นเคยกับหน้าที่ของการทำงานของเครื่องจักรซีเอ็นซีเป็นอย่างดี ช่างจะใช้วิธีการทำงานแบบง่ายๆ โดยการยึดจับชิ้นงานเข้ากับโต๊ะงานและคาดว่าจะได้วิธีการตัดที่ดีที่สุดไม่ได้ ในทางตรงข้ามช่างจะต้องจัดวางแผนขั้นตอนการทำงานไว้ล่วงหน้าเพื่อที่จะได้ผลผลิตที่ดี ดังนั้น จึงจำเป็นที่ช่างจะต้องรู้ถึงองค์ประกอบส่วนใดของเครื่องจักรที่สามารถควบคุมได้ และมีวิธีการควบคุมอย่างไร องค์ประกอบของเครื่องจักรกลซีเอ็นซีที่สามารถควบคุมได้แก่

- แนวแกนป้อน
- การขับป้อน

2.2.1 แนวแกนป้อน

ในการทำงานของเครื่องจักรซีเอ็นซี บ่อยครั้งที่เรามักได้ยินว่า แนวแกน ซึ่งหมายถึง แนวการเคลื่อนที่ขององค์ประกอบเครื่องจักร เช่น โต๊ะงาน เพลาหัวเครื่อง อุปกรณ์เครื่องมือ

สำหรับเครื่องจักรทั่วไป การเคลื่อนที่ในแนวแกนต่างๆ จะเกิดจากการหมุนมือหมุนหรือโยกคันโยกป้อนอัตโนมัติ

เครื่องจักรซีเอ็นซีมีแนวแกนป้อนร่วมกันอยู่หลายแกน ทำให้สามารถตัดเนื้อรูปทรงชิ้นงานได้อย่างที่ต้องการได้ การกำหนดแนวแกนต่างๆ ของเครื่องจักรกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซีเอ็นซีจะกำหนดตามมาตรฐานสากลภายใต้หัวเรื่อง Coordinate axes and directions of movement for numerically controlled machinery ซึ่งจะกำหนดตามแนวแกน x y z

2.2.2 การขับป้อน

การขับป้อนจะทำให้เกิดการเคลื่อนที่ของแท่นเลื่อน ในขณะที่ตัดหรือเฉือน แท่นเลื่อนอาจนำพาให้ชิ้นงานเลื่อนหรือคมมีดเคลื่อนก็ได้

ระบบขับโดยทั่วไปใช้มอเตอร์กระแสตรงในการควบคุมการทำงาน ด้วยวงจรอิเล็กทรอนิกส์จากภายนอก มอเตอร์ชนิดนี้จะสามารถหมุนและเบรคให้หยุดได้ทั้งสองทิศทางขณะตัดและเลื่อนชิ้นงาน โดยการเคลื่อนที่ป้อนจะต้องเป็นไปอย่างสม่ำเสมอและสามารถต้านแรงต้านจากภายนอกได้ เช่น แรงตัดเฉือนเป็นต้น ด้วยเหตุนี้ระบบขับป้อนจึงต้องออกแบบมาให้มีความแข็งแรงสูง มีการเคลื่อนที่นุ่มนวลสม่ำเสมอ แต่ยังคงสามารถตอบสนองต่อการเปลี่ยนแปลงอัตราการป้อนได้อย่างรวดเร็ว นอกจากนี้ในขณะที่ทำงาน คมตัดอาจทื่อ หรือ การเคลื่อนที่ของแท่นเลื่อนถูกกีดขวาง หรือมีการเร่งอัตราเลื่อนอย่างรวดเร็วหรือ หยุดในทันที สาเหตุเหล่านี้ทำให้มอเตอร์รับภาระหนักเกินไป ซึ่งอาจทำให้มอเตอร์เสียหายได้ ดังนั้นจึงมีการป้องกันอุบัติเหตุเหล่านี้โดยการติดตั้งคัลท์แบบลูกกลิ้ง ร่วมกับวงจรอิเล็กทรอนิกส์ ปัจจัยหนึ่งที่จะทำให้ระบบขับป้อนสามารถทำงานได้อย่างมีประสิทธิภาพคือ การออกแบบวงจรควบคุม

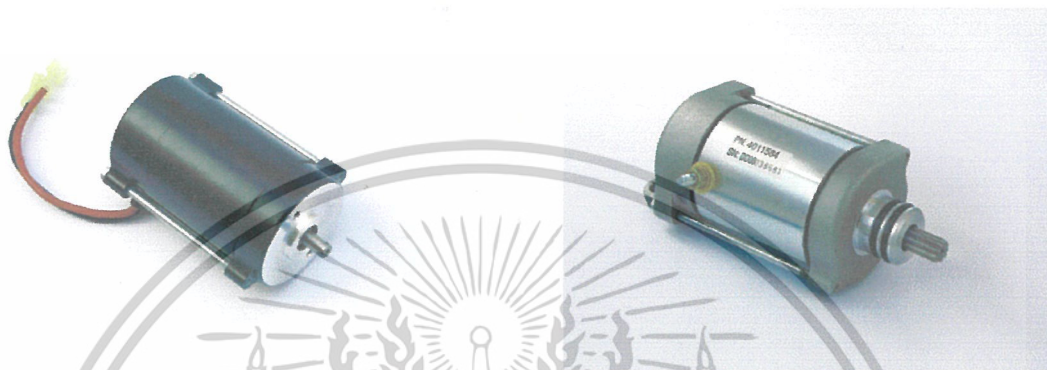
2.2.3 มอเตอร์

เครื่องจักรซีเอ็นซีสมัยใหม่ออกแบบใช้ระบบขับป้อนแบบเซอร์โว ทำให้สามารถป้อนอัตราและความเร็วรอบได้โดยไม่มีข้อจำกัดความเร็วและอัตราป้อน โดยทั่วไปมอเตอร์ที่มใช้ในระบบขับป้อนมี ๓ ชนิด

- มอเตอร์กระแสตรง(dc motor) ลักษณะคือสร้างจากแม่เหล็กถาวรที่มี ๔ หรือ ๘ ขั้ว การใช้มอเตอร์กระแสตรง ทำให้สามารถปรับอัตราการป้อนได้อย่างละเอียดและมีวงจรควบคุมที่ไม่ซับซ้อน แต่ก็มีข้อเสียตรงที่มอเตอร์ประเภทนี้นั่งต้องใช้แปรงถ่าน ซึ่งต้องคอยทำความสะอาดและเปลี่ยนเมื่อแปรงถ่านหมด นอกจากนี้แปรงถ่านยังสามารถสร้างความเสียหายให้กับแกนมอเตอร์อันส่งผลให้กำลัง

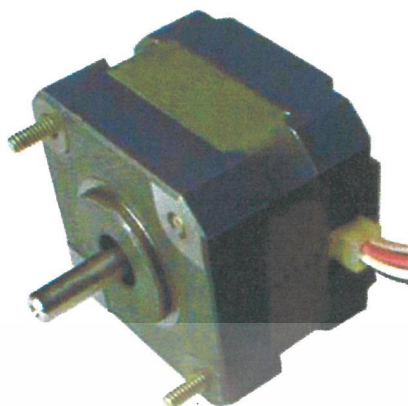
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มอเตอร์ลดลงด้วย ข้อเสียอีกประการหนึ่งคือ หากต้องการกำลังขับที่สูง มอเตอร์ก็จะมีขนาดใหญ่ด้วย และที่ความเร็วรอบสูงๆ จะทำให้แรงบิดน้อยลง ดังนั้นจึงใช้กับเครื่องจักรกลซีเอ็นซีที่มีขนาดเล็กและขนาดกลาง



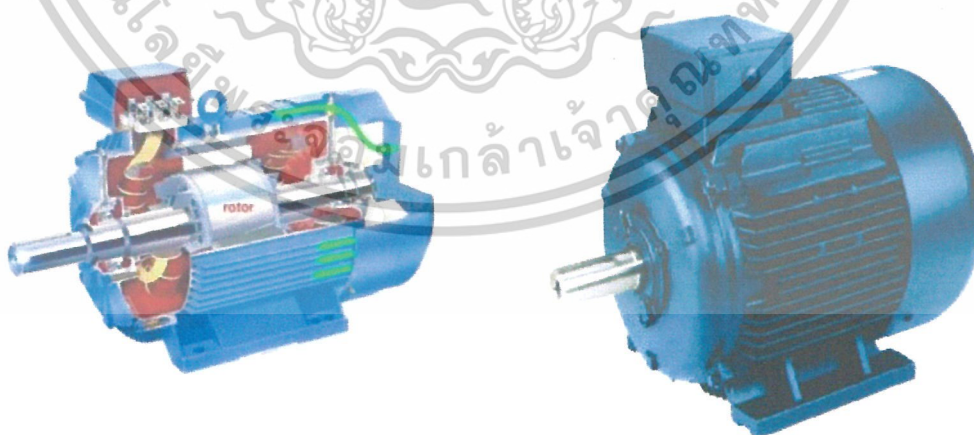
รูปที่ 2.1 มอเตอร์กระแสตรง ที่มีขายตามท้องตลาดอย่างแพร่หลาย

- มอเตอร์แบบขั้น (stepping motor) เป็นมอเตอร์ที่ทำงานแบบต่อเนื่อง โดยการแปลงคลื่นสัญญาณที่หอนเข้าไปในระบบให้เป็นการเคลื่อนที่เชิงมุม การหมุนในแต่ละมุม หรือ แต่ละขั้นที่เปลี่ยนไป ๑ ขั้น จะเท่ากับ ๑ ลูกคลื่นสัญญาณ ดังนั้นตำแหน่งของเพลาจะถูกกำหนดโดยจำนวนลูกคลื่นสัญญาณที่ถูกป้อนเข้าไป และความเร็วของเพลาจะถูกวัดเป็นจำนวนขั้นต่อวินาที (steps per second) ซึ่งจะเท่ากับความเร็วของคลื่นสัญญาณที่หอนเข้าไปในระบบ (pulses per second) ดังนั้นความเที่ยงตรงของระบบจึงขึ้นอยู่กับความสามารถของมอเตอร์ในการแบ่งขั้นการหมุนตามจำนวนสัญญาณที่ป้อนเข้าไปในระบบ แรงบิดของมอเตอร์ชนิดนี้จะลดลงเมื่อความเร็วในการหมุนมากขึ้น ดังนั้นจึงเหมาะสมกับเครื่องจักรกลซีเอ็นซีขนาดเล็กที่ไม่ต้องการแรงขับมาก



รูปที่ 2.2 มอเตอร์แบบขั้น (stepping motor)

- มอเตอร์กระแสสลับ (ac motor) ส่วนมากเป็นมอเตอร์แบบซิงโครไนซ์ ข้อดีของมอเตอร์ชนิดนี้คือ ไม่ต้องใช้แปรงถ่าน ทำให้สามารถลดค่าใช้จ่ายในการบำรุงรักษาได้ และเมื่อวัดขนาดมอเตอร์ก็เทียบกับมอเตอร์กระแสตรง มอเตอร์แบบกระแสสลับจะให้แรงบิดดีกว่า และมีขนาดเล็กกว่าด้วย แต่ข้อเสียคือ วงจรควบคุมค่อนข้างซับซ้อนมากกว่า



รูปที่ 2.3 มอเตอร์แบบกระแสสลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

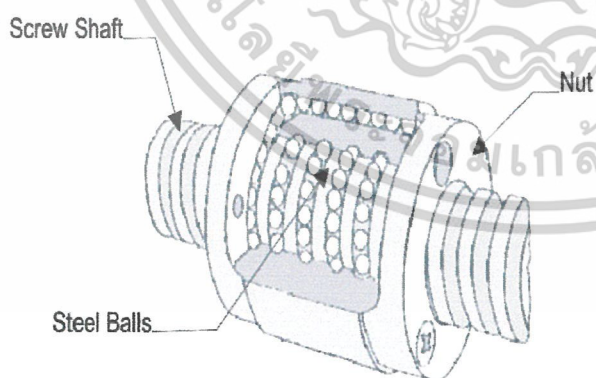
2.2.4 บอลล์สกรู

หัวใจของระบบขับเคลื่อนของซีเอ็นซีก็คือ การส่งกำลังขับเคลื่อนด้วยบอลล์สกรูซึ่งจะมีลูกบอลล์หมุนเวียนอยู่ตลอดเวลา บอลล์สกรูจะประกอบด้วยสกรูกับนัทที่มีลักษณะเป็นเกลียวกลม ร่องเกลียวกลมและสกรูจะขบแข็งและเจียรในผิวให้มันเรียบเพื่อลดความฝืดและเพิ่มความเที่ยงในการเคลื่อนที่

เมื่อมอเตอร์หมุนขับเคลื่อนสกรู นัทก็จะเคลื่อนที่ไปตามความยาวตลอดแนวสกรู พาให้แท่นและโต๊ะเคลื่อนที่ไปตามรางเลื่อน

ภายในตัวของนัทจะประกอบด้วยชุดของลูกบอลล์จำนวนมาก ทำให้มั่นใจได้ว่าความเสียดทานในการส่งกำลังขับเคลื่อนจากสกรูไปยังแท่นเลื่อนจะมีเหลือน้อยมาก นัทจะถูกแบ่งออกเป็นสองซีก และซีกประกอบยึดเข้าด้วยกันโดยมีการเตรียมแรงอัดไว้ก่อน ทำให้สามารถลดระยะคลอน(backlash) เหลือน้อยที่สุดจนแทบจะไม่มีเลยได้ ทำให้การเคลื่อนที่ของแท่นมีความแม่นยำสูง สามารถทำงานซ้ำๆกันได้อย่างมีประสิทธิภาพ

การต่อกำลังระหว่างมอเตอร์กับบอลล์สกรูจะมีชุดคลัทช์ความฝืดเป็นตัวเชื่อม ซึ่งนอกจากมีหน้าที่ต่อกำลังขับเคลื่อนแล้ว ยังมีหน้าที่ป้องกันอุบัติเหตุจากการที่แท่นเลื่อนชนงาน หรือ กระแทกกับสิ่งกีดขวาง กล่าวคือ เมื่อมีการชนหรือกระแทกจนถึงแรงค่าหนึ่ง คลัทช์ก็จะตัดระบบส่งกำลังระหว่างตัวมอเตอร์และบอลล์สกรูทันที



รูปที่ 2.4 ชิ้นส่วนบอลล์สกรู ซึ่งเป็นหัวใจของระบบส่งกำลังเครื่องจักรซีเอ็นซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 สเต็ปมอเตอร์

เนื่องจากเครื่องจักรกลซีเอ็นซีที่สถาบันฯเป็นเครื่องที่ใช้มอเตอร์แบบขั้น นักศึกษาจึงต้องศึกษาข้อมูลเพื่อทว่า การเขียนโปรแกรมจะได้มีความสอดคล้องกับ อุปกรณ์ฮาร์ดแวร์ที่มีอยู่

สเต็ปมอเตอร์ที่ขับเคลื่อนด้วยสัญญาณพัลส์ ลักษณะการขับเคลื่อน จะหมุนรอบ แกนได้ 360 องศา มีลักษณะไม่ต่อเนื่อง แต่มีลักษณะเป็นสเต็ป โดยแต่ละสเต็ปจะ ขับเคลื่อนได้ 1,1.5,1.8 หรือ 2 องศา แล้วแต่ละโครงสร้างของมอเตอร์ ลักษณะงานที่นำ มอเตอร์ไปใช้จะเป็นงานที่ต้องการตำแหน่งแม่นยำ เช่น ระบบขับเคลื่อนหัวแม่พิมพ์ใน เครื่องพิมพ์ (PRINTER)ระบบขับเคลื่อนหัวอ่านในเครื่องอ่านบันทึกเหล็ก ระบบ ขับเคลื่อนตำแหน่งของปากกาใน X-Y PLOTTER เป็นต้น

2.3.1 ชนิดและโครงสร้างของสเต็ปมอเตอร์



รูปที่ 2.5 สเต็ปมอเตอร์แบบมีสาย 5 เส้น

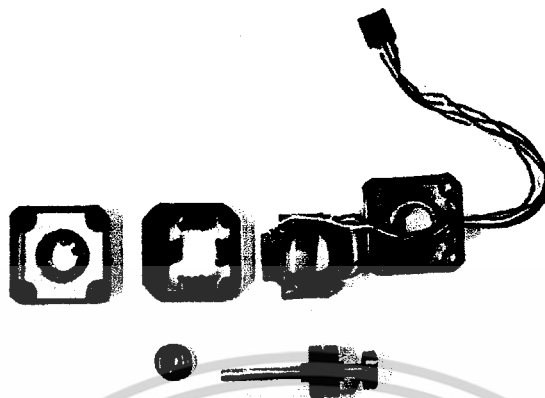


รูปที่ 2.6 มอเตอร์แบบมีสาย 6 เส้น



รูปที่ 2.7 สเต็ปมอเตอร์หลายแบบไบโพลาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แสดงภาพถ่าย โครงสร้างสเต็ปมอเตอร์

สเต็ปมอเตอร์ที่พบในปัจจุบันมี 3 ลักษณะดังนี้

1. แบบแม่เหล็กถาวร(PERMANENT MAGNET_PM)

สเต็ปมอเตอร์แบบ PM จะมีสเตเตอร์ (STATOR) ที่พันขดลวดไว้หลายๆ โพล โดยมีโรเตอร์ (ROTOR) เป็นรูปทรง กระบอกฟันเลื่อย และโรเตอร์ทำด้วยแม่เหล็กถาวร เพื่อป้อนไฟกระแสตรง ให้กับขดลวดสเตเตอร์ จะทำให้เกิดแรงแม่เหล็กไฟฟ้าผลักดันโรเตอร์ ทำให้มอเตอร์หมุนมอเตอร์แบบ PM จะเกิดแรงดูดยึดให้โรเตอร์หยุดอยู่กับที่ แม้จะไม่ได้ป้อนไฟเข้าขดลวด

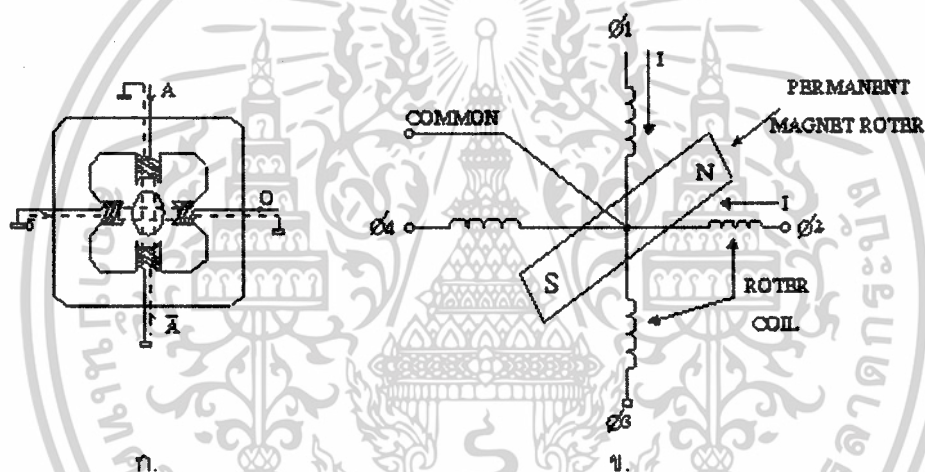
2. แบบแปรค่ารีลักแตนซ์ (VARIABLE RELUCTANCE- VR)

สเต็ปมอเตอร์แบบ VR จะมีการหมุนโรเตอร์ได้อย่างอิสระ แม้จะไม่ได้จ่ายไฟให้โรเตอร์ทำจากสารเฟอร์โรแมกเนติก กำลังอ่อน มีลักษณะเป็นฟันเลื่อยรูปทรงกระบอกโดยจะมีความสัมพันธ์ โดยตรงกับจำนวนโพลในสเตเตอร์ แรงบิดที่เกิดขึ้นจะไปหมุนโรเตอร์ ไปในเส้นทางของอำนาจแม่เหล็กที่มีค่ารีลักแตนซ์ต่ำที่สุด ตำแหน่งที่จะเกิดแน่นอนและมีเสถียรภาพแต่จะเกิดขึ้นได้หลายๆ จุด ดังนั้นเมื่อป้อนไฟเข้าขดลวดต่างๆ ในมอเตอร์แตกต่างกันไป ก็ทำให้

มอเตอร์ หมุนไปตำแหน่งต่างๆ กันโรเตอร์ของ VR จะมีความเฉื่อยของโรเตอร์น้อยจึงมีความเร็วรอบสูงกว่ามอเตอร์แบบ PM

3. แบบผสม(HYBRID-H)

สเต็ปมอเตอร์แบบ H จะเป็นลูกผสมของ VR กับ PM โดยจะมีสเตเตอร์คล้ายกับที่ใช้ใน VR โรเตอร์มีหมวกหุ้ม ปลายซึ่งมีลักษณะของสารแม่เหล็กที่มีกำลังสูง โดยการควบคุมขนาดรูปร่างของหมวกแม่เหล็กอย่างดีทำให้ได้มุม การหมุนและครั้งน้อยและแม่นยำ ข้อดีก็คือ ให้แรงบิดสูงและมีขนาดกระทัดรัด และให้แรงจูงยัดโรเตอร์นิ่งกับที่ตอนไม่จ่ายไฟ



รูปที่ 2.9 (ก) โครงสร้าง (ข) วงจรเทียบเท่า (equivalent circuit) ของมอเตอร์ ชนิด 4 ขด

เฟสที่จ่ายกระแสไฟฟ้า	ϕ_1	$\phi_1 \phi_2$	ϕ_2	$\phi_2 \phi_3$	ϕ_3	$\phi_3 \phi_4$	ϕ_4	$\phi_4 \phi_1$
ตำแหน่งโรเตอร์								

ตารางที่ 2.1 ตารางแสดงมุมของโรเตอร์เทียบกับกระแสไฟฟ้าที่จ่ายแก่เฟสต่าง 8 ตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากลักษณะของมอเตอร์หมุนกับกระแสไฟที่ป้อนแก่เฟสต่างๆจะสามารถสั่งงานให้ STEPPING MOTOR หมุนได้ 3 อย่าง คือ

1.แบบจ่ายกระแสไฟให้เฟสเดียววนเวียนกันไป เรียก ONE-EXCITATION หรือ HALF DRIVE คือ f_1, f_2, f_3, f_4 การ OUT EXCITATION แบบนี้แรงบิดจะน้อย

2. แบบจ่ายกระแสไฟให้พร้อมกันทีละ 2 f เรียก TWO-EXCITATION หรือ FULL STEP คือ $f_1f_2, f_2f_3, f_3f_4, f_4f_1$ หมุนเวียนกันไปแบบนี้แรงบิดจะมาก

3. แบบจ่ายกระแสไฟให้ทีละ 1 เฟส สลับกับ 2 เฟส เรียก ONE-TWO EXCITATION หรือ HALF STEP เหมือนรูปแสดงของมอเตอร์ แต่แบบนี้จำนวน STEP วนเพิ่มจะเป็นตรงกันข้าม

เฟส	ϕ_4	ϕ_3	ϕ_2	ϕ_1
ϕ_1	1	0	0	1
ϕ_2	0	0	1	0
ϕ_3	0	1	0	0
ϕ_4	1	0	0	0

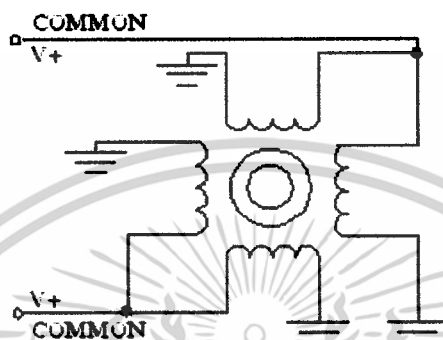
ตารางที่ 2.2 ตารางแสดงการจ่ายกระแสให้ขั้วแม่เหล็ก

การตรวจสอบหาสาย COMMON และสาย GROUND ของ STEPPING แบบ PM (แบบแกนโรเตอร์เป็นแม่เหล็กถาวร) โดยทั่วไป SP MOTOR แบบ PM จะมีอยู่ 2 ชนิด

1. ชนิดที่เป็น COMMON ภายนอก SP MOTOR

แบบนี้มีสายอยู่ 6 เส้น คือ

- สายที่เป็น COMMON 2 เส้น
- สายที่เป็น GROUND 4 เส้น



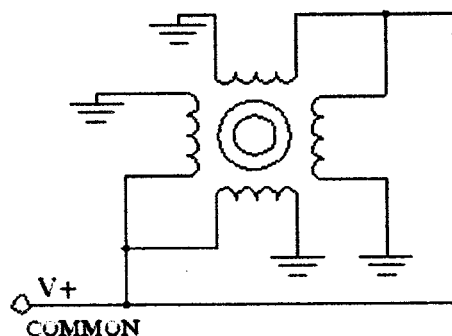
รูปที่ 2.10 วงจรสเต็ปมอเตอร์ ชนิดมีสาย 6 เส้น

สาย COMMON 1 เส้น จะ DRIVE GROUND 2 เส้น ในการเช็คให้ใช้มิเตอร์วัดหาสายที่เป็น COMMON ก่อนโดยการตั้ง RANGE ของมิเตอร์ที่ R*1 จับที่สายทีละคู่ ถ้าหากวัดสาย COMMON เทียบกับสาย GROUND ได้ถูกต้องค่าความต้านทานที่อ่านได้จะน้อย แต่ถ้าวัดผิดสาย คือวัดสาย GROUND เทียบกับ GROUND ค่าความต้านทานที่อ่านได้จะสูงกว่าแต่ถ้าวัดสาย COMMON เทียบกับสาย GROUND ที่ไม่ใช่คู่กันแล้ว เข้มมิเตอร์ก็จะไม่ กระดิก ให้ทดลองวัดเปรียบเทียบกันทีละคู่ ก็จะทราบว่าสายใดเป็นสาย COMMON สายใดเป็นสาย GROUND

2. ชนิดที่เป็น COMMON ภายใน SP MOTOR

แบบนี้มีสายอยู่ 5 เส้น คือ

- สายที่เป็น COMMON 1 เส้น
- สายที่เป็น GROUND 4 เส้น



รูปที่ 2.11 วงจรสเต็ปมอเตอร์ชนิดมีสาย 5 เส้น

ในการวัดให้ทำแบบเดียวกับการวัด SP MOTOR ชนิด COMMON ภายนอกแตกต่างกัน เพียงแบบ COMMON ภายในสาย COMMON 1 เส้น DRIVE สาย GROUND 4 เส้น ดังนั้นหากสายเส้นใดเมื่อวัดเทียบกับสายเส้นอื่นแล้วมีค่าความต้านทานน้อยที่สุดสายเส้นนั้นเป็นสาย COMMON และที่เหลืออีก 4 เส้นจะเป็นสาย GROUND

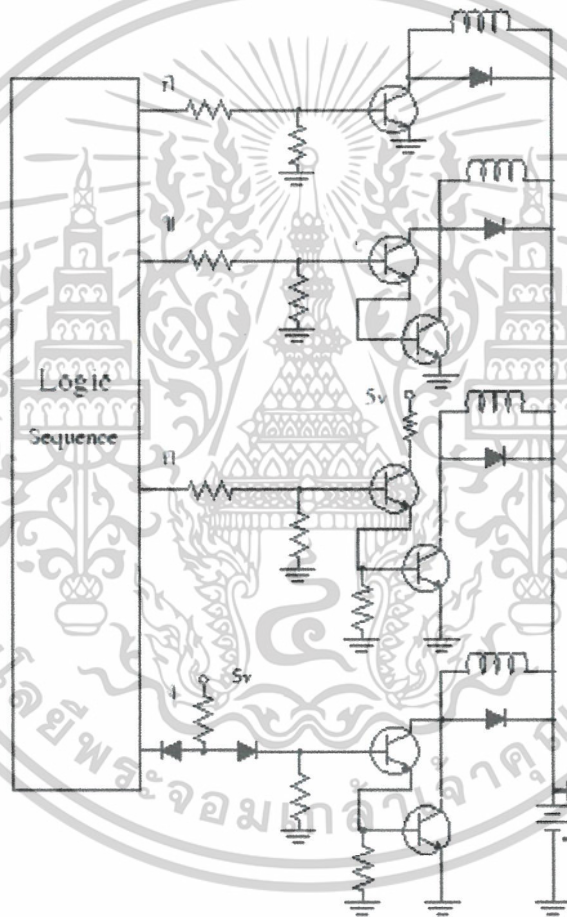
การเรียงเฟสของ STEPPING MOTOR แบบ PM

เมื่อเราทราบว่าสายเส้นใดเป็นสาย COMMON แล้วแต่เรายังไม่ทราบว่าสาย GROUND เส้นใดเป็นเฟสที่ 1 เฟสที่ 2 เฟสที่ 3 และเฟสที่ 4 ในการเรียงเฟสนั้นให้ใช้มิเตอร์วัดโดยนำ V+ เขไปที่สาย COMMON วัดเทียบกับสาย GROUND เส้นใดก็ได้ 1 เส้น จะทำให้แกนโรเตอร์เคลื่อนไปข้างหน้า 1 STEP เมื่อเปลี่ยนสาย GROUND เส้นแรกเป็นเส้นที่ 2 หาก MOTOR ไม่เคลื่อนที่ไปข้างหน้าแสดงว่าการเรียงเฟสไม่ถูกต้องก็ให้วัดเทียบกับสาย GROUND เส้นใหม่ต่อไป หาก MOTOR เคลื่อนที่ไปข้างหน้าตามกัน วัดที่สาย GROUND เส้นต่อไปเรื่อย ๆ ก็จะทำให้ทราบว่าสายเส้นใดเป็นเฟสแรก สายเส้นใดเป็นเฟสที่ 2 เฟสที่ 3 และเฟสที่ 4 การเรียงเฟสของ SP MOTOR แบบ PM ทั้งชนิดที่เป็น COMMON ภายนอกและชนิดที่เป็น COMMON ภายใน ใช้หลักการเดียวกัน

2.3.2 วงจรขับสเต็ปมอเตอร์

วงจรขับ (DRIVE)

เมื่อเรารู้ซีควเอนซ์ของมันแล้วต่อไปเราก็ต้องมีวงจร DRIVE ให้แก่ STEPPING MOTOR วิธีที่ง่ายที่สุดในการ ต่อวงจรซีควเอนส์เข้ากับวงจรขับ คือ การต่อโดยตรง ดังเอาท์พุท ก และ ข แต่ถ้ากระแสเอาท์พุท ของวงจรซีควเอนส์ไม่เพียงพอก็ต้องต่อ บัฟเฟอร์ (BUFFER) เพื่อขยายกระแสดังรูป เอาท์พุท ค และ ง



รูปที่ 2.12 แสดงวิธีขับเอาท์พุทของสเต็ปมอเตอร์

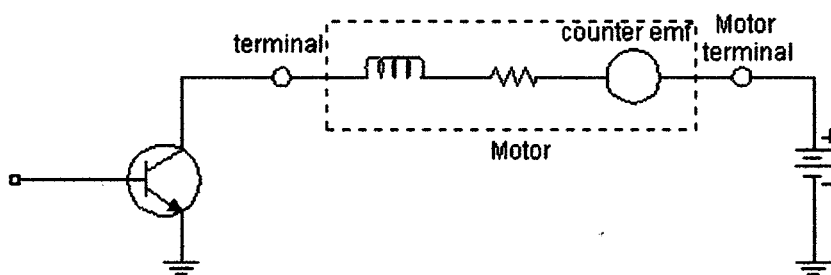
ปัญหาเกี่ยวกับวงจรขับ

ขดลวดของสเต็ปมอเตอร์เป็น โหลดชนิดตัวเหนี่ยวนำ และมีค่า เปรียบเสมือนผลรวมของความเหนี่ยวนำ (Inductance) อนุกรมกับความต้านทาน

ผังรูปที่ 2.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

103063

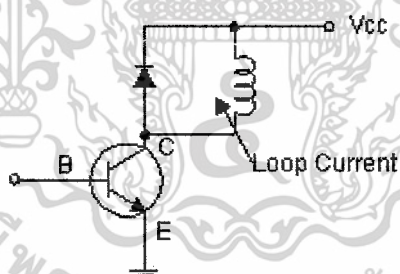


รูปที่ 2.13 วงจรสมมูล (equivalent circuit) ของสเต็ปมอเตอร์

ซัพเพรสเซอร์ (SUPPRESSOR)

เมื่อทรานซิสเตอร์รูป A หยุดนำกระแสจะทำให้เกิดแรงเคลื่อนไฟฟ้าค่าสูงจำนวนหนึ่งเนื่องจากผลของการเปลี่ยนแปลงของกระแสในตัวเหนี่ยวนำ และแรงเคลื่อนไฟฟ้านี้จะเป็นอันตรายต่อทรานซิสเตอร์ได้ป้องกันได้ด้วยวิธีการต่อไปนี้

1. ใช้ไดโอดซัพเพรสเซอร์

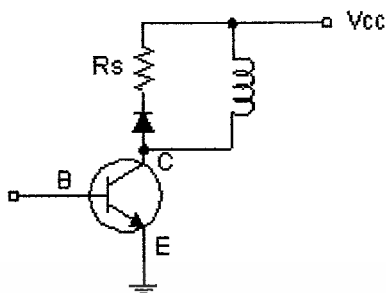


รูปที่ 2.14 ป้องกันอันตรายต่อทรานซิสเตอร์ด้วยไดโอดซัพเพรสเซอร์

กระแสหมุนเวียน CIRCULATING CURRENT จะเริ่มไหลหลังจากทรานซิสเตอร์ หยุดนำกระแสและสวิตช์คาบคอลลเลคเตอร์ จะเท่ากับสวิตช์ของแหล่งจ่ายไฟฟ้า ข้อเสียคือ กระแสจะหมุนเวียนอยู่นานและ จะทำให้เกิดแรงบิดห้ามล้อ (BREAKING TORQUE) พลังงานส่วนใหญ่จะสูญเสียเป็นความร้อนใน ความต้านทานของขดลวด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

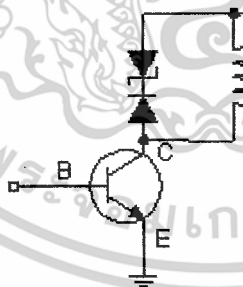
2. ใช้ไดโอดและตัวต้านทานชัฟเฟรสเซอร์



รูปที่ 2.15 ป้องกันอันตรายต่อทรานซิสเตอร์ด้วยไดโอดและตัวต้านทานชัฟเฟรสเซอร์

ถ้าค่าตัวต้านทาน R_S ยิ่งมากกระแสหมุนเวียนก็จะลดลงเร็วขึ้นแต่ศักดาของ คอลเลกเตอร์ในขณะกระแสไหลย้อนกลับ จะมีค่าสูงขึ้นพลังงานส่วนใหญ่สูญเสียในตัวต้านทาน R_S

3. ใช้ซีเนอร์ไดโอดชัฟเฟรสเซอร์

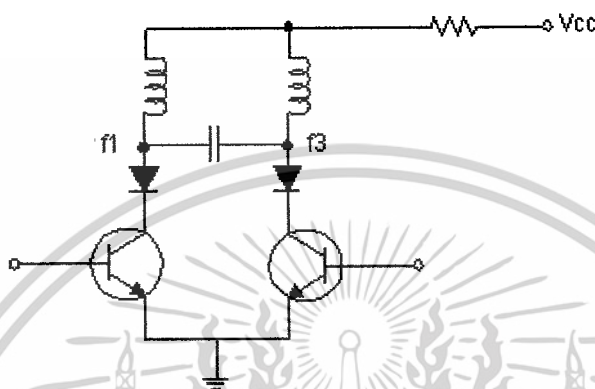


รูปที่ 2.16 ป้องกันอันตรายโดยต่อทรานซิสเตอร์ด้วยซีเนอร์ไดโอดชัฟเฟรสเซอร์

เมื่อทรานซิสเตอร์หยุดนำกระแส กระแสจะลดลงได้เร็วกว่า 2 แบบแรก และศักดาที่คอลเลกเตอร์ขณะกระแสไหลย้อนกลับ จะเท่ากับศักดาของซีเนอร์

บวกกับศักดาของแหล่งจ่าย ซึ่งเป็นอิสระต่อกระแสพลังงานส่วนใหญ่สูญเสียใน ซีเนอร์ไดโอด

4. ใช้ตัวเก็บประจุฟเพรสเซอร์



รูปที่ 2.17 ป้องกันอันตรายโดยต่อทรานซิสเตอร์ด้วยตัวเก็บประจุฟเพรสเซอร์

จะใส่ตัวเก็บประจุให้ $f1$ กับ $f3$ และ $f2$ กับ $f4$ เมื่อทรานซิสเตอร์ หยุด นำกระแสตัวเก็บประจุ C จะต่อกับทรานซิสเตอร์ โดยผ่านไดโอด และจะ ดูดกลืนกระแสที่ค่อยๆ ลดลงจากขั้วของมอเตอร์ เพื่อป้องกันทรานซิสเตอร์ เสีย และยังช่วยแคมป์ไฟ คือช่วยลดความร้อนที่เกิดขึ้นในขั้วของมอเตอร์ เนื่องจากการแกว่งของโรเตอร์

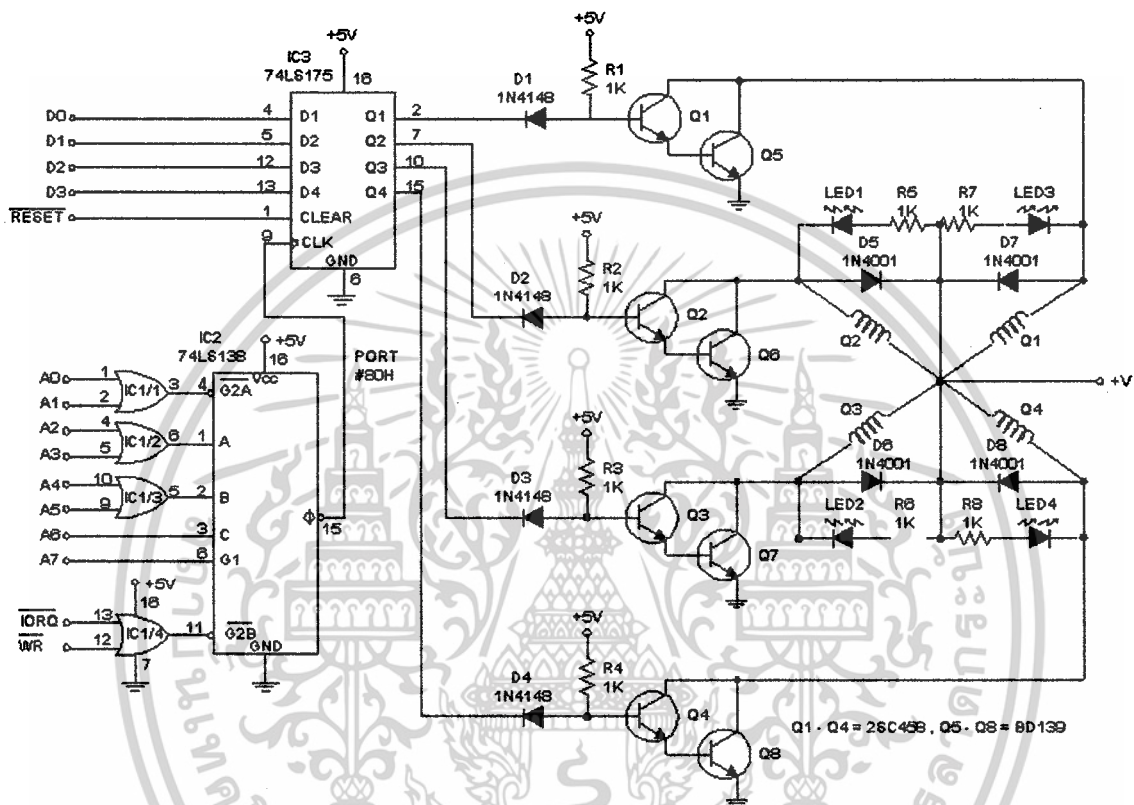
วงจรขับ

Q1-Q4 เป็นตัวขับร่วมกับ Q5-Q8 โดยการต่อแบบคาร์ลิงตัน วงจรนี้ สามารถใช้กับมอเตอร์ที่ดึงกระแสมากๆแต่ไม่ ควรเกิน 2A เพราะ Q5-Q8 อาจจะ ร้อนซึ่งวงจรที่ให้มานี้ออกแบบไว้สำหรับรุ่นที่ให้กระแสไม่เกิน 600 mA ถ้าจะ ใช้ กับรุ่นที่ดึงกระแสมากๆก็เพียงแค่เปลี่ยน Q5-Q8 เป็นเบอร์ที่สามารถทนกระแส ได้มากขึ้น D5-D8 เป็นตัวป้องกัน ความเสียหาย ทรานซิสเตอร์อันเกิดจากการ เปลี่ยนแปลงของสนามแม่เหล็ก ทำให้เกิดแรงดันย้อนกลับ LED1-LED5 ทำ

หน้าที่บอกสถานะเอาต์พุต (สว่างคือ "1" ดับคือ "0") R5-R8 ทำหน้าที่จำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระแส โดย LED ซึ่งแรงดันย้อนกลับสามารถทำให้ LED ติดสว่างได้เช่นกัน
แรงดัน +V นั้นควรเลือกให้เหมาะสมกับสแต็ปมอเตอร์ที่จะใช้ด้วยคือ ประมาณ
 $V+2.5$ โดย V คือค่าแรงดันที่กำกับไว้ที่ข้างตัวมอเตอร์ เช่น 12 V ก็จะใช้แรงดัน
เท่ากับ $12+2.5 = 14.5$ โวลต์ เป็นต้น



รูปที่ 2.18 วงจรสแต็ปมอเตอร์ต่อกับ Z-80 ซึ่งสามารถดัดแปลงไปใช้กับ
ไมโครโปรเซสเซอร์เบอร์อื่น ได้

2.4 เทคโนโลยี CAD CAM

CAD เป็นคำย่อมาจากคำว่า Computer Aided Design ซึ่งแปลตามศัพท์ได้ว่า การใช้คอมพิวเตอร์ช่วยออกแบบ

เทคโนโลยีของซอฟต์แวร์ CAD ได้ถูกพัฒนาเริ่มจากการเป็นซอฟต์แวร์ช่วยเขียนแบบ 2 มิติ(Drawing) เสมือนเป็นกระดานเขียนแบบอิเล็กทรอนิกส์ ซึ่งจะมีคำสั่งในการใช้งานซึ่งผู้ใช้สามารถเรียกใช้คำสั่ง โดยการใช้เมาส์เลือกที่เมนูบนจอภาพ หรือการป้อนคำสั่งจากแป้นพิมพ์ ซอฟต์แวร์ CAD มีหลายกลุ่มคำสั่ง ได้แก่ คำสั่งในการวาดองค์ประกอบต่างๆ ได้แก่ เส้นตรง(Line), ส่วนโค้ง(Arc), วงกลม(Circle), วงรี(Ellipse), รูปเหลี่ยม(Polygon) เช่น สามเหลี่ยม, สี่เหลี่ยม และยังมีคำสั่งในการช่วยวาดองค์ประกอบเพิ่มเติม เช่น การสะท้อนให้เกิดภาพ (Mirror), การสำเนาองค์ประกอบที่มีอยู่(Copy) นอกจากนี้ยังมีคำสั่งในการแก้ไขสิ่งที่ได้วาดลงไปแล้ว ได้แก่ คำสั่งลบออก(Erase), ตัดบางส่วน (Trim), เคลื่อนย้าย(Move), หมุนภาพ(Rotate), การจัดองค์ประกอบต่างๆ จำแนกอยู่ในชั้นต่างๆ(_layer_) เพื่อความสะดวกในการทำงานเสมือน มีแบบหลายๆ แผ่นมาซ้อนทับกันอยู่ เช่น ในอาคารหนึ่งหลังจะมีทั้งแบบโครงสร้างแบบไฟฟ้า, แบบผนัง ฯลฯ ซึ่งเป็นกระดานไข เมื่อต้องการใช้ก็จะนำมาทับกับคำสั่งดังกล่าวที่ซอฟต์แวร์ CAD 2 มิติ มีให้ นั่น ทำให้ผู้ใช้สะดวกและประหยัดทั้งเวลาและค่าใช้จ่าย คือเขียนแบบให้เสร็จในคอมพิวเตอร์ จากนั้นค่อยพลอตออกทางเครื่องพลอต(Plotter) ทำให้ประหยัดกระดาษ, ประหยัดเวลาที่จะต้องเขียนแบบใหม่หมดหากเกิดข้อผิดพลาด, ไม่ต้องใช้ใบมีดขีดแบบเพื่อลบเส้นที่ผิด แต่ด้วยการใช้ซอฟต์แวร์ CAD ที่เป็น 2 มิติ นี้ ผู้ใช้ยังคงต้องใช้จินตนาการและประสบการณ์ เพื่อวาดให้ได้แบบที่ถูกต้อง เช่น การวาดรูปด้านข้างของอาคารหรือของชิ้นส่วนที่มีความโค้งมน เหล่านี้ อาจทำให้แบบที่ออกมามีความผิดพลาดไป ซอฟต์แวร์ CAD อีกประเภทหนึ่งซึ่งเป็นซอฟต์แวร์ที่ใช้เทคโนโลยีในการพัฒนาสูงขึ้น คือ ซอฟต์แวร์ CAD ที่มีการทำงานในระบบ 3 มิติ ซึ่งจะมีคุณสมบัติพื้นฐาน คือ

1.ออกแบบหรือสร้างแบบจำลอง ลักษณะ 3 มิติ คือ มีขนาดทั้งความกว้าง

ความยาว และความสูง(ความหนา)

2.หมุนดูได้ทุกมุมมองที่อยากดู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.สร้างแบบ(Drawing) 2 มิติหลังจากการเสร็จสิ้นออกแบบเพื่อนำไปผลิต

4.แก้ไขได้ทันทีที่ต้องการ

CAM ย่อมาจาก Computer Aided Manufacturing ซึ่งแปลว่า คอมพิวเตอร์ช่วยในการผลิต ซึ่งจะใช้ซอฟต์แวร์เพื่อควบคุมเครื่องจักร ให้สามารถสร้างชิ้นงานได้ตามที่ได้ออกแบบไว้แล้ว โดยใช้ข้อมูลทางรูปร่างจาก CAD จากความก้าวหน้าทางเทคโนโลยี IT CAM สามารถใช้ข้อมูลจาก CAD ในการกำหนดว่าจะใช้เครื่องจักรใดในการผลิต วัสดุชิ้นงานมีขนาดเท่าใด วางตำแหน่งอ้างอิงอย่างไร ใช้เครื่องมืออะไรในการตัดเฉือน ระบบการใช้คอมพิวเตอร์ช่วยในการผลิตนี้ มีองค์ประกอบหลักๆ คือ

1.เครื่องจักรซีเอ็นซี (CNC:Computer Numerical Controlled) คือ เครื่องจักรที่ใช้สำหรับกัด ก้อนวัตถุดิบ(โลหะ, ไม้ ,พลาสติกสังเคราะห์) ให้ได้รูปร่าง ตามแบบชิ้นงานที่ได้ออกแบบไว้แล้ว

2.ซอฟต์แวร์สำหรับงาน CAM ซึ่งมีมากมายหลายยี่ห้อให้เลือกใช้ ซึ่งควรมีคุณสมบัติ ดังนี้รับข้อมูล 3 มิติจากซอฟต์แวร์ CAD ได้ในรูปแบบมาตรฐาน (IGES, STEP, STL) เลือก Tool หรือ หัวกัดชิ้นงาน ตามขนาดที่ต้องการ กำหนด การกัดงานด้วยรูปแบบต่างๆ ได้แก่ การกัดหยาบ, กัดละเอียด ทดสอบการกัดชิ้นงาน บนจอภาพเพื่อตรวจสอบก่อนการกัดงานจริง (ส่วนนี้ซอฟต์แวร์บางตัวอาจยังไม่มีให้ใช้งาน) สร้าง G-code ซึ่งเป็นรหัสเพื่อบอกให้เครื่องจักรทำงานตามขั้นตอนที่กำหนดไว้ได้ถูกต้อง ซึ่งซอฟต์แวร์ CAM นั้นจะต้องสร้าง G-code ให้มีรูปแบบตรงกับ รูปแบบที่เครื่องจักรรุ่นนั้น ๆ รู้จัก ตัวอย่างรายชื่อซอฟต์แวร์ เช่น CAM MasterCAM, Delcam, Esprit, EdgeCAM, Gibbscam, Solidcam

บทที่ 3

โปรแกรม K_EM_CAM

3.1 คอนเซปต์ของโปรแกรม

- ทำงานในคอส
- ใช้ภาษาซีในการเขียน
- ง่ายต่อการศึกษาใช้งาน
- มีความคล่องตัวเนื่องจากสามารถประยุกต์ใช้กับ สเต็ปมอเตอร์ในเครื่องกลอื่นๆ ได้

3.2 แนวความคิดการทำงานของโปรแกรม

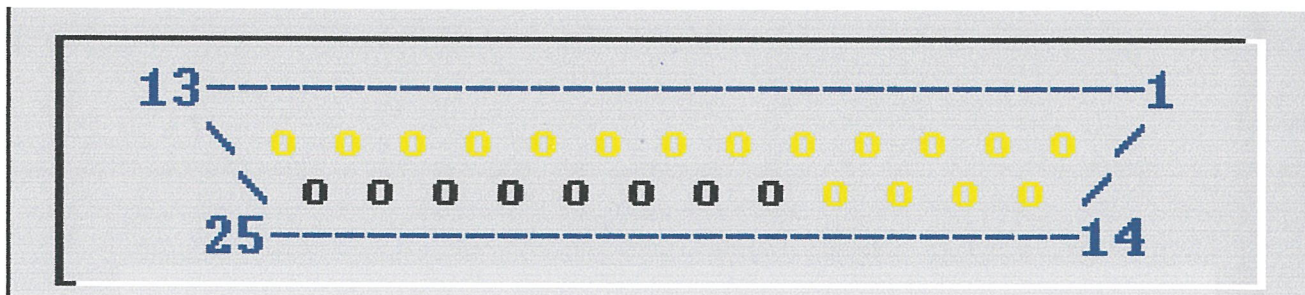
- โปรแกรมทำหน้าที่ควบคุมความเร็วและทิศทางการหมุนของสเต็ปมอเตอร์ ที่ต่างกันทั้งสามแกน เพื่อให้หัวกัดเลื่อนไปในทิศทางตาม G-code ที่รับมา



รูปที่ 3.1 การทำงานของเครื่อง CNC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การเชื่อมต่อผ่าน พอร์ตขนาน



รูปที่3.2 แผนผังพอร์ตขนาน

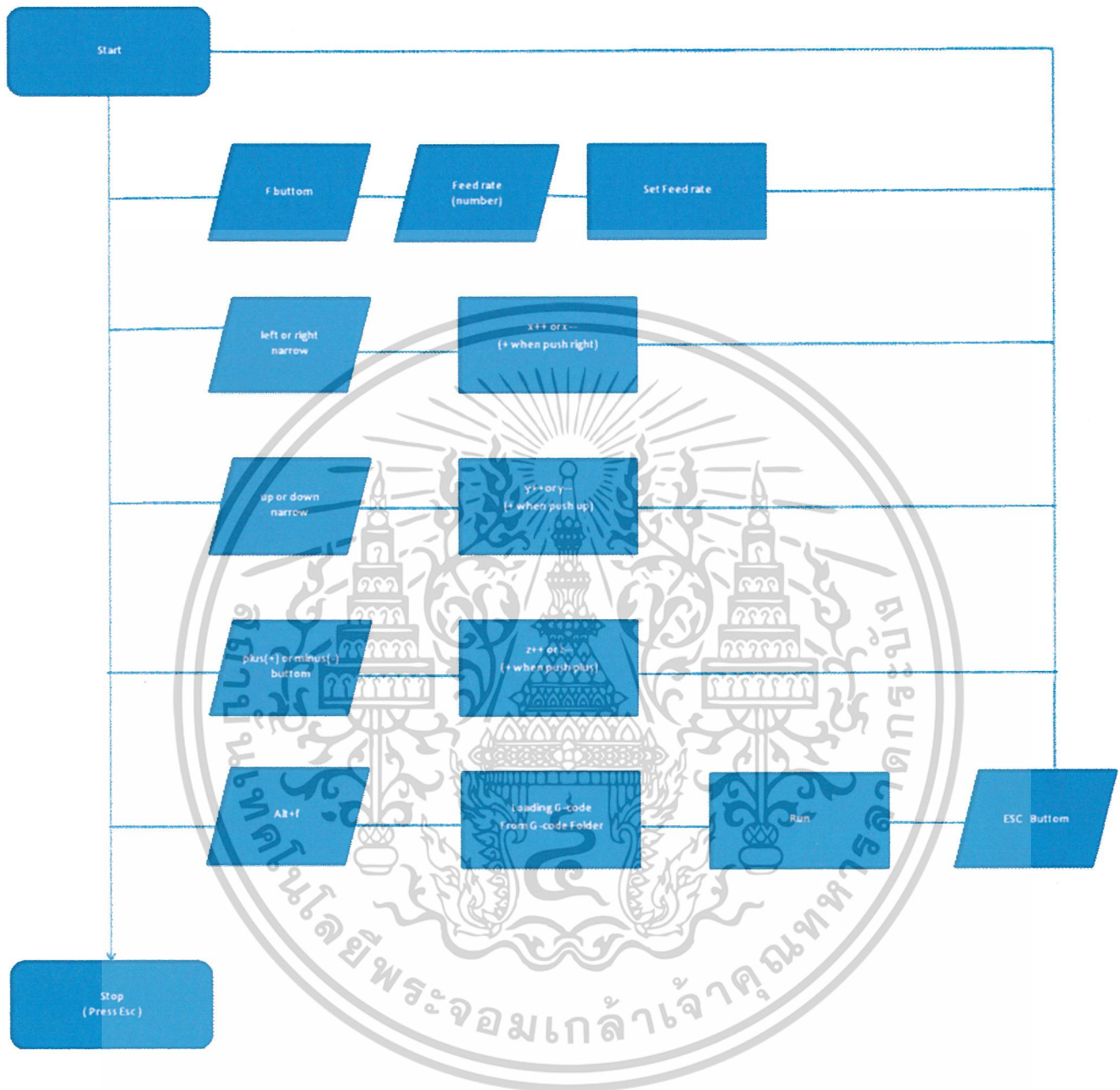
ใน พอร์ตขนานจะใช้พินหมายเลข 1-17 ในการส่งข้อมูล ในการเชื่อมต่อกับ ซีเอ็นซี เรา
จะใช้พอร์ต 2-9 ในการควบคุมการเคลื่อนที่ของมอเตอร์ในแกนต่างๆ แล้วใช้พิน 10-13
และ 15 ในการรับค่าจากลิมิตสวิตช์ ที่เหลือจะยังไม่ได้ใช้งานในโปรแกรมของเรา



รูปที่3.3 พอร์ตขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 หลักการทำงานของโปรแกรม(โฟลว์ชาร์ท)



รูปที่ 3.4 Flow Chart ของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

เริ่มด้วยการเปิดคอส



รูปที่ 4.1 แสดงหน้าต่างของโปรแกรมคอส

เรียกโปรแกรมจากหน้าของคอส

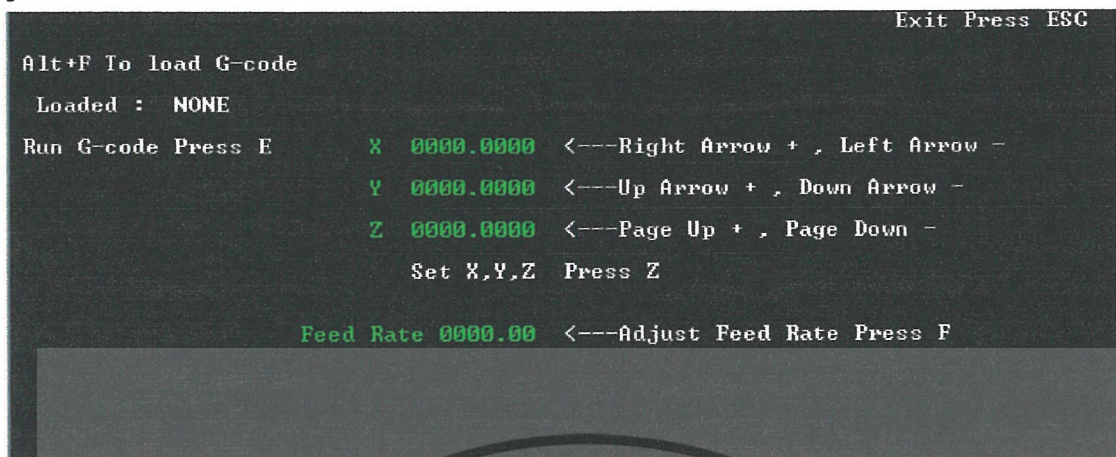


รูปที่ 4.2 แสดงหน้าต่างของโปรแกรมคอสขณะผู้ใช้ทำการเรียกโปรแกรม

K_EM_cam.exe

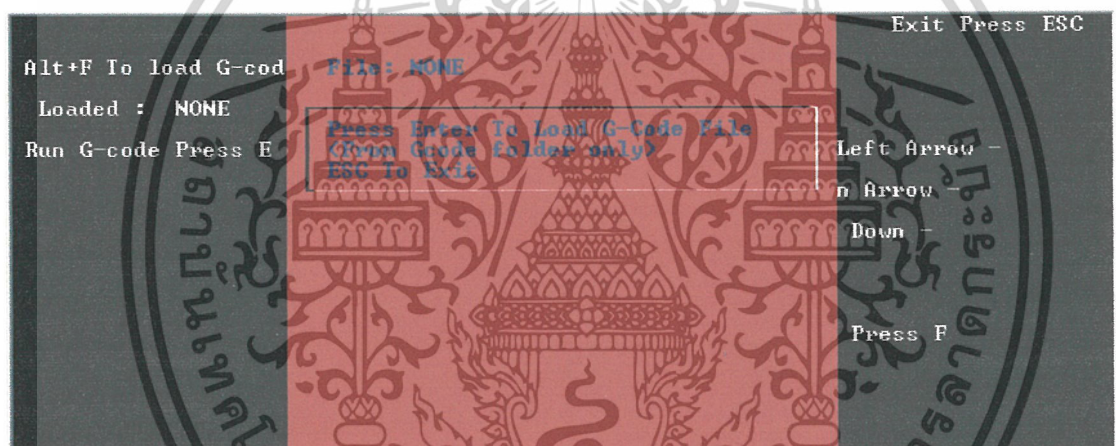
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปหน้าต่างโปรแกรม



รูปที่4.3 แสดงหน้าต่างของโปรแกรม K_EM_cam.exe

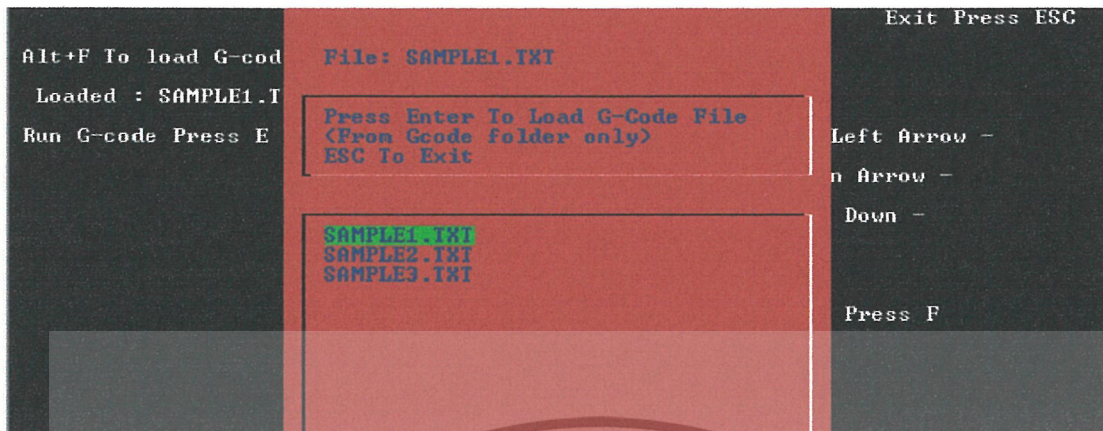
กด Alt+F เพื่อทำการโหลด G-code (จากโฟลเดอร์เดียวที่เก็บตัวไฟล์โปรแกรม)



รูปที่4.4 แสดงหน้าต่างของโปรแกรม K_EM_cam.exe ขณะทำการเรียกไฟล์ G-code

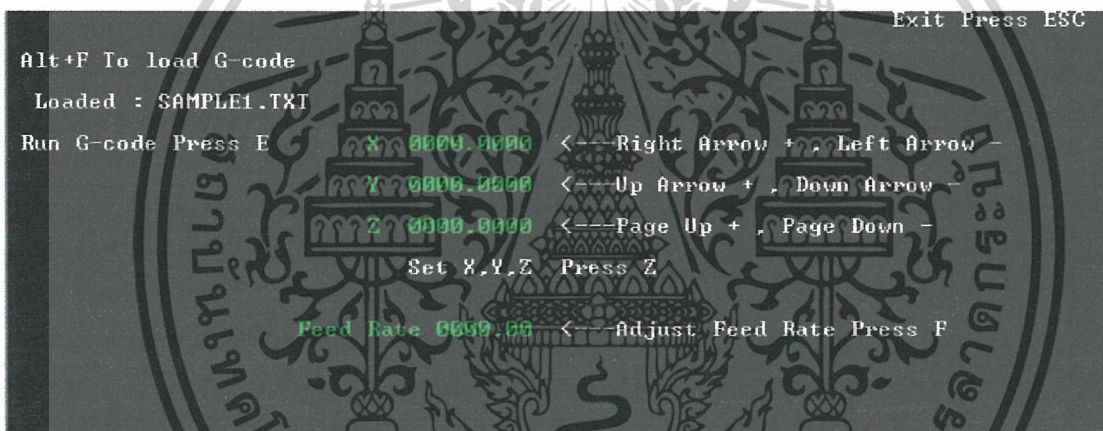
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกไฟล์ที่ต้องเซฟ G-code ไว้



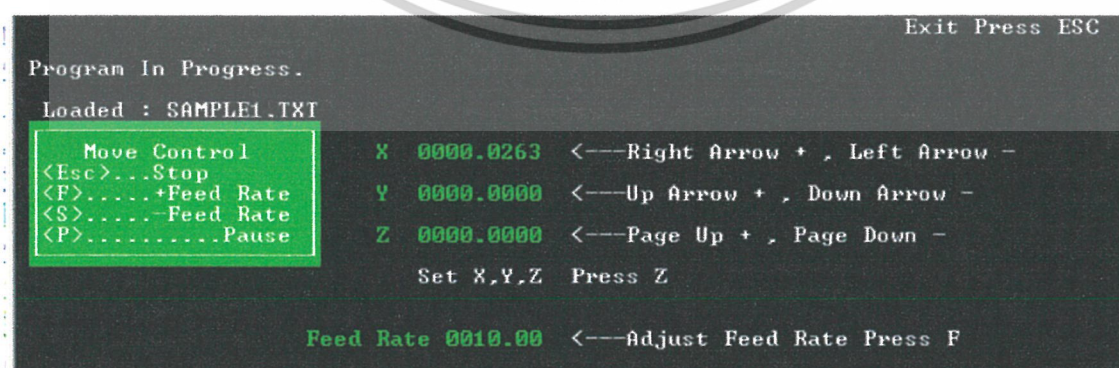
รูปที่4.5 แสดงหน้าต่างของโปรแกรม K_EM_cam.exe ขณะทำการเรียกไฟล์ G-code

กด E เพื่อ เริ่มโปรแกรม



รูปที่4.6 แสดงหน้าต่างของโปรแกรม K_EM_cam.exe ขณะทำการรันไฟล์ G-code

รันโปรแกรม



รูปที่4.7 แสดงหน้าต่างของโปรแกรม K_EM_cam.exe ขณะทำการรันไฟล์ G-code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์ผลการทดลอง

5.1 สรุปผลการทดลอง

จากการทดลองโปรแกรมที่สร้างขึ้น นักศึกษาได้พบว่า โปรแกรมสามารถรับค่า G-code ที่ถูกเก็บไว้ในไฟล์เดียวกันกับไฟล์ที่เก็บโปรแกรม และสามารถทำการอ่าน G-code ซึ่งอยู่ในลักษณะเท็กซ์ไฟล์ (ไฟล์ที่เปิดจากโปรแกรม Notepad และมีนามสกุล txt) เพื่อทำการสั่งมอเตอร์แบบขั้นเคลื่อนที่หวั้กัด หรือ ปากกา หรือ เพื่อสร้างชิ้นงานได้ตามต้องการ

5.2 วิจารณ์ผลการทดลอง

ผลที่ได้จากการทดลองเป็นไปตามคาด เนื่องจากโปรแกรมสามารถทำการอ่านค่า G-code และ สั่งมอเตอร์แบบขั้นให้เคลื่อนหวั้กัดไปได้ทั้งสามแกน โดยโปรแกรมทำหน้าที่ได้เสมือนโปรแกรมของมืออาชีพ แต่ยังมีหน้าต่างที่ดูค่าสมย และยังไม่สามารถรันตัวโปรแกรมได้ใน วินโดว์ และบางฟังก์ชัน เช่น การสั่งหวั้กัดยังไม่มี จึงต้องทำให้ต่อไฟเลี้ยงตรงให้หวั้กัด(สว่าง)

5.3 ปัญหาที่พบในการทำโครงการนี้

ปัญหาที่พบเกิดจากการที่แหล่งข้อมูล เข้าถึงได้ยากมาก เนื่องจากไม่มีเนื้อหาที่เกี่ยวข้องโดยตรงซะทีเดียว ทำให้การศึกษาในเมืองค้นเป็นไปอย่างยากลำบาก จนกระทั่งต้องขวนขวายในอินเทอร์เน็ต จึงพบกับแหล่งข้อมูล ซึ่งก็ยังคงเป็นข้อมูลที่ไม่ประติดประต่อกัน แต่ก็สามารถช่วยผู้จัดทำได้มากเป็นอย่างยิ่ง (โดยในที่นี้ผู้จัดทำจะลงเว็บไซต์ที่เป็นที่มาของแหล่งความรู้ในการทำโครงการครั้งนี้ เพื่อเป็นประโยชน์แก่รุ่นน้องในปีการศึกษาถัดไปในการนำโปรแกรมมาพัฒนาเพื่อให้เกิดความก้าวหน้ายิ่งขึ้น)

ปัญหาใหญ่ที่พบในการทำโครงการชิ้นนี้คือ การที่ไม่มีซอร์จโค้ดตัวอย่าง จึงเปรียบเสมือนไม่มีโปรแกรมให้ศึกษา ผู้จัดทำจึงต้องศึกษาซอร์จโค้ดของโปรแกรมน้อยๆ เช่น โปรแกรมที่นำมาขับมอเตอร์แบบขั้น โปรแกรมเพื่อใช้รับค่าไฟล์ G-code แล้วนำ

โปรแกรมมาต่อกัน แล้วลองคอมไพล์ เพื่อทดสอบหาข้อผิดพลาด และนี่เองทำให้การทำ
โครงการล่าช้า

5.4 แนวทางในการพัฒนา

แนวทางที่ผู้จัดทำคาดหวังให้ผู้สนใจหรือรุ่นน้องมาพัฒนาต่อยอดก็คือ การ
เพิ่มเติมโปรแกรม และการศึกษารายละเอียดของโค้ดจนมีความเข้าใจแตกฉาน และ
สามารถประยุกต์พลิกแพลงได้(ต้องใช้เวลาอย่างมาก) เพื่อที่จะสามารถใส่ฟังก์ชันการ
ทำงาน เช่น การนำห้วงกุดกลับสู่ตำแหน่งเริ่มต้น(Move To Origin, Move to Home) และ
การสั่งการสว่านจากตัวเครื่องคอมพิวเตอร์



ภาคผนวก

```

#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <string.h>
#include <dir.h>
#include <stdlib.h>
#include <math.h>
#include "definitions.h"
#include "kernel.h"
#include "setup.h"
#include "utils.h"
#include "gcode.h"
#include "file.h"
#include "control.h"
#include "help.h"
#include "operator.h"
int version;
int main(int argc, char *argv[])
{
    float cur_speed=0, dist_x=0, dist_y=0, dist_z=0, dist_a=0, tfloat;
    double num, num2, loopmax;
    int msg=COMPLETE, i, ch, ch2, kp=0, tint, gosub=OFF, goloop=OFF, loopnum;
    unsigned long slow_freq, fast_freq;
    long place_holder;
    static char drive[MAXDRIVE], dir[MAXDIR], tstring[20];
    static char fstring[200];
    static BYTE buffer[1000], buffer2[1500], buffer3[100], buffer4[400], buffer5[1000];
    static struct hard_setup hardware;
    static struct soft_setup software, temp_software;
    static struct gcode_setup gcode;
    static struct control ctrl;
    static struct filepos sub[100];
    static struct pos pt, world;
    FILE *ifp, *ifp2;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fpos_t fpos1, fpos2, fpos3, fpostemp, subpos, mainpos;
```

```
BYTE s_port_read;
```

```
if(argc>0)
```

```
{fnsplit(argv[0],drive,dir,NULL,NULL);
```

```
if(strlen(dir)>0) dir[strlen(dir)-1]=0;
```

```
system(drive); chdir(dir);
```

```
}
```

```
version=FULL;
```

```
clrscr();
```

```
_setcursortype(_NOCURS);
```

```
box(20,4,60,19, RED, GREEN, GREEN, BLACK);
```

```
gotoxy(4,4); cprintf(" ");
```

```
gotoxy(4,5); cprintf(" This program written by ");
```

```
gotoxy(4,7); cprintf(" Surasit Suntikasemsuk ");
```

```
gotoxy(4,6); cprintf(" ");
```

```
gotoxy(4,9); cprintf(" KMITL 4EM ");
```

```
gotoxy(4,10); cprintf(" ");
```

```
sleep(1);
```

```
world.x_pos=world.y_pos=world.z_pos=world.a_pos=
```

```
pt.x_pos=pt.y_pos=pt.z_pos=pt.a_pos=0;
```

```
ctrl.jog_type=FAST; ctrl.jog_inc=0; ctrl.frate=0;
```

```
sprintf(ctrl.file, "NONE");
```

```
ctrl.cs=0; ctrl.pty=ABS;
```

```
ctrl.M00=ctrl.M03=ctrl.M07=ctrl.M08=ctrl.M10=ctrl.cvv=OFF;
```

```
load_screen_setup(&world,&pt,&ctrl);
```

```
end_realtime();
```

```
load_setup(&hardware, &software, "<DEFAULT>");
```

```
load_setup(&hardware, &software, "SETUP.INI");
```

```
if(software.update_after==ON) begin_realtime(&software, fast_freq);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else begin_realtime(&software, slow_freq);

load_setup(&hardware, &software, "<DEFAULT>");

load_setup(&hardware, &software, "SETUP.INI");

for(i=0;i<100;i++) sprintf(gcode.tool_dis[i],"NONE");

load_gcode_setup(&gcode);

//reassign port bits

clear_data_bits(&hardware);

if(hardware.io1_on==HIGH) clear_io_bit(&hardware, hardware.io1_bit);

else set_io_bit(&hardware, hardware.io1_bit);

if(hardware.io2_on==HIGH) clear_io_bit(&hardware, hardware.io2_bit);

else set_io_bit(&hardware, hardware.io2_bit);

if(hardware.io3_on==HIGH) clear_io_bit(&hardware, hardware.io3_bit);

else set_io_bit(&hardware, hardware.io3_bit);

if(hardware.io4_on==HIGH) clear_io_bit(&hardware, hardware.io4_bit);

else set_io_bit(&hardware, hardware.io4_bit);

//print control screen

control_screen(&software, &pt,&ctrl, &gcode);

for(i=0; i<20; i++) if(kbhit()) getch();

message("Alt+F To load G-code");

while(kp!=27)

{for(i=0;i<20;i++) if(kbhit()) getch();

kp=getch();

if(kp==0) kp=getch()+1000; cur_speed=0;

switch(kp)

{case 35+1000: //alt h

case 59+1000: //F1

help(1);

break;

case 'v':

case 'V':

if(ctrl.cvv==ON) ctrl.cvv=OFF;

else ctrl.cvv=ON;

break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(software.e_stop!=OFF) s_port_read=(248&inportb(hardware.port+1))^128;
if(software.e_stop!=OFF                &&                hardware.e_trip*hardware.e_bit-
(s_port_read&hardware.e_bit)==0) {msg=E_STOP; break;}
if(ctrl.M03==OFF)
    {if(hardware.io1_on==HIGH) set_io_bit(&hardware, hardware.io1_bit);
    else clear_io_bit(&hardware, hardware.io1_bit);
    ctrl.M03=ON;}
else
    {if(hardware.io1_on==HIGH) clear_io_bit(&hardware, hardware.io1_bit);
    else set_io_bit(&hardware, hardware.io1_bit);
    ctrl.M03=OFF;}
break;
case 64+1000: //F6
if(software.e_stop!=OFF) s_port_read=(248&inportb(hardware.port+1))^128;
if(software.e_stop!=OFF                &&                hardware.e_trip*hardware.e_bit-
(s_port_read&hardware.e_bit)==0) {msg=E_STOP; break;}
if(ctrl.M07==OFF)
    {if(hardware.io2_on==HIGH) set_io_bit(&hardware, hardware.io2_bit);
    else clear_io_bit(&hardware, hardware.io2_bit);
    ctrl.M07=ON;}
else
    {if(hardware.io2_on==HIGH) clear_io_bit(&hardware, hardware.io2_bit);
    else set_io_bit(&hardware, hardware.io2_bit);
    ctrl.M07=OFF;}
break;
case 65+1000: //F7
if(software.e_stop!=OFF) s_port_read=(248&inportb(hardware.port+1))^128;
if(software.e_stop!=OFF                &&                hardware.e_trip*hardware.e_bit-
(s_port_read&hardware.e_bit)==0) {msg=E_STOP; break;}
if(ctrl.M08==OFF)
    {if(hardware.io3_on==HIGH) set_io_bit(&hardware, hardware.io3_bit);
    else clear_io_bit(&hardware, hardware.io3_bit);
    ctrl.M08=ON;}
else
    {if(hardware.io3_on==HIGH) clear_io_bit(&hardware, hardware.io3_bit);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else set_io_bit(&hardware, hardware.io3_bit);

ctrl.M08=OFF;

break;

if(software.e_stop!=OFF) s_port_read=(248&inportb(hardware.port+1))^128;

if(software.e_stop!=OFF          &&          hardware.e_trip*hardware.e_bit-
(s_port_read&hardware.e_bit)==0) {msg=E_STOP; break;}

if(ctrl.M10==OFF)

    {if(hardware.io4_on==HIGH) set_io_bit(&hardware, hardware.io4_bit);

    else clear_io_bit(&hardware, hardware.io4_bit);

    ctrl.M10=ON;};

else

    {if(hardware.io4_on==HIGH) clear_io_bit(&hardware, hardware.io4_bit);

    else set_io_bit(&hardware, hardware.io4_bit);

    ctrl.M10=OFF;};

break;

case 'o':

case 'O':

    tint=ctrl.cs; ctrl.cs++;

    if(ctrl.cs==7) ctrl.cs=0;

    pt.x_pos=pt.x_pos+(gcode.x_coord[tint]-gcode.x_coord[ctrl.cs])*ctrl.pty;

    pt.y_pos=pt.y_pos+(gcode.y_coord[tint]-gcode.y_coord[ctrl.cs])*ctrl.pty;

    pt.z_pos=pt.z_pos+(gcode.z_coord[tint]-gcode.z_coord[ctrl.cs])*ctrl.pty;

    pt.a_pos=pt.a_pos+(gcode.a_coord[tint]-gcode.a_coord[ctrl.cs])*ctrl.pty;

    break;

case 'u':

case 'U':

    if(software.units==MM) {ctrl.frate=change_unit(&software, ctrl.frate, IN);

        change_pt_units(&software, &world, IN, IN);

        change_gcode_units(&software, &gcode, IN, IN);

        change_units(&hardware, &software, IN, IN);}

    else {ctrl.frate=change_unit(&software, ctrl.frate, MM);

        change_pt_units(&software, &world, MM, MM);

        change_gcode_units(&software, &gcode, MM, MM);

        change_units(&hardware, &software, MM, MM);}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    break;
case 'r':
case 'R':
    if(ctrl.pty==REL)
        {pt.x_pos=world.x_pos-gcode.x_coord[ctrl.cs]; pt.y_pos=world.y_pos-gcode.y_coord[ctrl.cs];
        pt.z_pos=world.z_pos-gcode.z_coord[ctrl.cs]; pt.a_pos=world.a_pos-gcode.a_coord[ctrl.cs];
        ctrl.pty=ABS;
        }
    else
        {pt.x_pos=pt.y_pos=pt.z_pos=pt.a_pos=0;
        ctrl.pty=REL;
        }
    break;
case 'w':
case 'W':
    if(software.a_status==OFF) break;
    if(software.a_follow==INDEPENDANT) software.a_follow=FOLLOW_X;
    else if(software.a_follow==FOLLOW_X) software.a_follow=FOLLOW_Y;
    else if(software.a_follow==FOLLOW_Y) software.a_follow=FOLLOW_Z;
    else if(software.a_follow==FOLLOW_Z) software.a_follow=INDEPENDANT;
    break;
case 'f':
case 'F':
    gettext(22,19,59,24,buffer);
    box(22,19,59,24, RED,BLACK,WHITE,WHITE);
    gotoxy(12,1); cprintf("<Esc> To Exit");
    gotoxy(10,3); cprintf("Enter Feed Rate:");
    _setcursortype(_NORMALCURSOR);
    tfloat=0;
    gotoxy(16,4); get_value(&tfloat);
    if(tfloat>9999) {message("F/R Must Be <=9999."); puttext(22,19,59,24,buffer); break;}
    if(software.time==MIN) ctrl.frate=tfloat/60;
    else ctrl.frate=tfloat;
    _setcursortype(_NOCURSOR);
    puttext(22,19,59,24,buffer);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    break;
case 't':
case 'T':
    gettext(22,19,59,24,buffer);
    box(22,19,59,24, RED,BLACK,WHITE,WHITE);
    gotoxy(13,1); cprintf("<Esc> To Exit");
    gotoxy(10,3); gotoxy(3,8); cprintf("Enter New Tool Number:");
    _setcursortype(_NORMALCURSOR);
    gotoxy(18,4); tint=tfloat=0; get_value(&tfloat); tint=tfloat;
    _setcursortype(_NOCURSOR);
    puttext(22,19,59,24,buffer);
    if(tint<=gcode.tnum_max)
    {
        pt.z_pos=(pt.z_pos-(gcode.tool_len[tint]-gcode.tool_len[gcode.tnum]))*ctrl.pty;
        gcode.tnum=tint;
    }
    else
    {
        gettext(22,19,59,24,buffer);
        box(22,19,59,24, RED,BLACK,WHITE,WHITE);
        gotoxy(4,3); cprintf("    Tool Number Not Valid");
        gotoxy(4,4); cprintf("    <Any Key>");
        getch();
        puttext(22,19,59,24,buffer);
    }
    break;
case 'j':
case 'J':
    if(ctrl.jog_type==FAST) ctrl.jog_type=SLOW;
    else if(ctrl.jog_type==SLOW) ctrl.jog_type=INC;
    else if(ctrl.jog_type==INC) ctrl.jog_type=FAST;
    break;
case 'i':
case 'I':
    gettext(22,19,59,24,buffer);
    box(22,19,59,24, RED,BLACK,WHITE,WHITE);
    gotoxy(13,1); cprintf("<Esc> To Exit");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gotoxy(7,3); cprintf("Enter New Jog Increment:");
_setcursortype(_NORMALCURSOR);
gotoxy(16,4); get_value(&ctrl.jog_inc);
_setcursortype(_NOCURSOR);
puttext(22,19,59,24,buffer);

break;

case 71+1000: //home
    if(software.limit==OFF)
        {message("Limit Switches Must Be On To Home.");
        break;
        }
    gettext(22,19,59,24,buffer);
    box(22,19,59,24, RED,BLACK,WHITE,WHITE);
    gotoxy(2,1); cprintf("<Esc> To Exit  <Enter> To Home All");
    gotoxy(4,3); cprintf("<%c>>Home %c",software.x_label, software.x_label);
    gotoxy(21,3); cprintf("<%c>>Home %c",software.y_label, software.y_label);
    gotoxy(4,4); cprintf("<%c>>Home %c",software.z_label, software.z_label);
    if(software.a_status==ON) {gotoxy(21,4); cprintf("<%c>>Home %c",software.a_label,
software.a_label);}
    gettext(2,6,22,12,buffer4);
    while((i=getch())!=27)
        {if(i==software.x_label || i==software.x_label+32)
            {move_menu(1); if((msg=homing(&hardware, &software, &world, X_AXIS, NULL, NULL,
NULL))!=COMPLETE) break;}
            if(i==software.y_label || i==software.y_label+32)
                {move_menu(1); if((msg=homing(&hardware, &software, &world, Y_AXIS, NULL, NULL,
NULL))!=COMPLETE) break;}
            if(i==software.z_label || i==software.z_label+32)
                {move_menu(1); if((msg=homing(&hardware, &software, &world, Z_AXIS, NULL, NULL,
NULL))!=COMPLETE) break;}
            if((i==software.a_label || i==software.a_label+32) && software.a_status==ON)
                {move_menu(1); if((msg=homing(&hardware, &software, &world, A_AXIS, NULL, NULL,
NULL))!=COMPLETE) break;}
            if(i==13)
                {move_menu(1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if((msg=homing(&hardware, &software, &world, Z_AXIS, POS, A_AXIS,
POS))!=COMPLETE) break;

        if((msg=homing(&hardware, &software, &world, X_AXIS, NULL, Y_AXIS,
NULL))!=COMPLETE) break;

        if(software.home_z==NEG && software.home_a==NEG)
            if((msg=homing(&hardware, &software, &world, Z_AXIS, NULL, A_AXIS,
NULL))!=COMPLETE) break;

        if(software.home_z==NEG && software.home_a==POS)
            if((msg=homing(&hardware, &software, &world, Z_AXIS, NULL, NULL,
NULL))!=COMPLETE) break;

        if(software.home_a==NEG && software.home_z==POS)
            if((msg=homing(&hardware, &software, &world, A_AXIS, NULL, NULL,
NULL))!=COMPLETE) break;

        break;
    }
    puttext(2,6,22,12,buffer4);
    pt.x_pos=(world.x_pos-gcode.x_coord[ctrl.cs])*ctrl.pt;
    pt.y_pos=(world.y_pos-gcode.y_coord[ctrl.cs])*ctrl.pt;
    pt.z_pos=(world.z_pos-gcode.z_coord[ctrl.cs])*ctrl.pt;
    pt.a_pos=(world.a_pos-gcode.a_coord[ctrl.cs])*ctrl.pt;
    update_display(&software, pt.x_pos, pt.y_pos, pt.z_pos, pt.a_pos, 0, 0, ctrl.frate, ALL);
}
puttext(2,6,22,12,buffer4);
puttext(22,19,59,24,buffer);
break;
case 31+1000: //alt s
    temp_software=software;
    end_realtime();
    setup(&hardware, &software);
    //reset for new freq if necessary
    load_setup(&hardware, &software, software.file);
    if(software.update_after==ON) begin_realtime(&software, fast_freq);
    else begin_realtime(&software, slow_freq);
    load_setup(&hardware, &software, software.file);
    change_units(&hardware, &software, software.units, software.units);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ctrl.frate=change_unit(&software, ctrl.frate, software.units);
change_pt_units(&temp_software, &world, software.units, software.a_units);
change_gcode_units(&temp_software, &gcode, software.units, software.a_units);
save_setup(&hardware, &software, "SETUP.INI");
break;
case 33+1000: //alt f
    if(ctrl.M00==ON) {do_gcode(&hardware, &software, &gcode, &ctrl, &pt, 'M', 30);
        ctrl.M00=OFF; message("");}
    if(strcmp(ctrl.file, "NONE")!=0) fclose(ifp);
    temp_software=software;
    end_realtime();
    get_file(ctrl.file, software.file);
    //reset for new freq if neccessary
    load_setup(&hardware, &software, software.file);
    if(software.update_after==ON) begin_realtime(&software, fast_freq);
    else begin_realtime(&software, slow_freq);
    load_setup(&hardware, &software, software.file);
    change_units(&hardware, &software, software.units, software.units);
    ctrl.frate=change_unit(&software, ctrl.frate, software.units);
    change_pt_units(&temp_software, &world, software.units, software.a_units);
    change_gcode_units(&temp_software, &gcode, software.units, software.a_units);
    save_setup(&hardware, &software, "SETUP.INI");

    if(strcmp(ctrl.file, "NONE")==0) {window(1,1,80,25); break;}
    sprintf(tstring, "GCODE\\%s", ctrl.file);
    if((ifp=fopen(tstring, "r"))!=NULL)
    {
        while((ch=fgetc(ifp))!=EOF)
        {if(ch=='(')
            {while((ch=fgetc(ifp))!=EOF && ch!='\n');
                if(ch==EOF) break;
            }
            if(ch=='M' || ch=='m')
                {if(NULL==fscanf(ifp, "%lf", &num)) continue;
                    if(num==30) break;
                }
        }
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}

//clear old subs from memory and get new subs id's and file positions
for(i=0;i<100;i++) sub[i].id=-1;
while(ch!=EOF && (ch=fgetc(ifp))!=EOF)
{
    if(ch=='(')
    {
        while((ch=fgetc(ifp))!=EOF && ch!='\n');
        if(ch==EOF) break;
    }
    if(ch=='$')
    {
        if(NULL==fscanf(ifp, "%lf", &num)) continue;
        if(num>=0) for(i=0;i<100;i++) if(sub[i].id==1) {sub[i].id=num; fgetpos(ifp,&sub[i].pos);
break;}
    }
}

//reset file
rewind(ifp);
//clear all variables
get_var(0,0,CLEAR);
//set up file positions
fgetpos(ifp,&fpos1);
for(i=0;i<8;i++)
{
    if(fgets(fstring, 200, ifp)!=NULL)
    {fstring[36]=0; textcolor(WHITE); textbackground(BLACK); gotoxy(1,8+i); }
}
fgetpos(ifp,&fpos2); fsetpos(ifp, &fpos1);
}

else sprintf(ctrl.file,"NONE");

window(1,1,80,25);

break;

case 24+1000: //alt o
    temp_software=software;
    operator_setup(&hardware, &software, &gcode);
    ctrl.frate=change_unit(&software, ctrl.frate, software.units);
    change_pt_units(&temp_software, &world, software.units, software.a_units);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

save_gcode_setup(&gcode);

break;

if((kp==72+1000    &&    software.invert_arrows==OFF)    ||    (kp==77+1000    &&
software.invert_arrows==ON))

    msg=jog(&hardware, &software, &pt, ctrl.jog_type, ctrl.jog_inc, Y_AXIS, POS);
else msg=jog(&hardware, &software, &pt, ctrl.jog_type, ctrl.jog_inc, X_AXIS, POS);

break;

case 80+1000:
case 75+1000:

    if((kp==80+1000    &&    software.invert_arrows==OFF)    ||    (kp==75+1000    &&
software.invert_arrows==ON))

        msg=jog(&hardware, &software, &pt, ctrl.jog_type, ctrl.jog_inc, Y_AXIS, NEG);
    else msg=jog(&hardware, &software, &pt, ctrl.jog_type, ctrl.jog_inc, X_AXIS, NEG);

    break;

case 73+1000:

    msg=jog(&hardware, &software, &pt, ctrl.jog_type, ctrl.jog_inc, Z_AXIS, POS);

    break;

case 81+1000:

    msg=jog(&hardware, &software, &pt, ctrl.jog_type, ctrl.jog_inc, Z_AXIS, NEG);

    break;

case '+':
case '=':

    msg=jog(&hardware, &software, &pt, ctrl.jog_type, ctrl.jog_inc, A_AXIS, POS);

    break;

case '-':

    msg=jog(&hardware, &software, &pt, ctrl.jog_type, ctrl.jog_inc, A_AXIS, NEG);

    break;

case 'z':
case 'Z':

    if(ctrl.cs!=0)

        {message("Must Be In MCS/G53 To Adjust Axis.");

        break;

        }

    gettext(22,19,59,24,buffer);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

box(22,19,59,24, RED,BLACK,WHITE,WHITE);

gotoxy(2,1); cprintf("<Esc> To Exit  <Enter> To Zero All");

gotoxy(4,3); cprintf("<%c> -> Set %c",software.x_label, software.x_label);

gotoxy(21,3); cprintf("<%c> -> Set %c",software.y_label, software.y_label);

gotoxy(4,4); cprintf("<%c> -> Set %c",software.z_label, software.z_label);

if(software.a_status==ON) {gotoxy(21,4);  cprintf("<%c>  ->  Set  %c",software.a_label,
software.a_label);}

while((i=getch())!=27)

    {if(i>96 && i<123) i=i-32;

        if(i==software.x_label || i==software.y_label || i==software.z_label || (i==software.a_label &&
software.a_status==ON))

            {_setcursortype(_NORMALCURSOR);

                gotoxy(6,6); cprintf("Enter New '%c' Value:",i); gotoxy(27,6);

                if(i==software.x_label) {tfloat=pt.x_pos; get_value_all(&tfloat); pt.x_pos=tfloat;}

                if(i==software.y_label) {tfloat=pt.y_pos; get_value_all(&tfloat); pt.y_pos=tfloat;}

                if(i==software.z_label) {tfloat=pt.z_pos; get_value_all(&tfloat); pt.z_pos=tfloat;}

                if(i==software.a_label) {tfloat=pt.a_pos; get_value_all(&tfloat); pt.a_pos=tfloat;}

                gotoxy(6,6); cprintf("                "); _setcursortype(_NOCURSOR);

            }

            if(i==13) {pt.x_pos=pt.y_pos=pt.z_pos=pt.a_pos=0; break;}

            update_display(&software, pt.x_pos, pt.y_pos, pt.z_pos, pt.a_pos, 0, 0, ctrl.frate, ALL);

        }

        puttext(22,19,59,24,buffer);

        break;

case 'm':

case 'M':

    if(ctrl.frate<.00009) {message("F/R Must Be >0 To Move."); break;}

    gettext(22,19,59,24,buffer);

    box(22,19,59,24, RED,BLACK,WHITE,WHITE);

    gotoxy(2,1); cprintf("<Esc> To Exit      <Enter> To Move");

    gotoxy(4,3); cprintf("<%c> -> %s",software.x_label, prn_val_all(dist_x));

    gotoxy(21,3); cprintf("<%c> -> %s",software.y_label, prn_val_all(dist_y));

    gotoxy(4,4); cprintf("<%c> -> %s",software.z_label, prn_val_all(dist_z));

    if(software.a_status==ON) {gotoxy(21,4);  cprintf("<%c>  ->  %s",software.a_label,
prn_val_all(dist_a));}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while((i=getch())!=27 && i!=13)
{
    if(i>96 && i<123) i=i-32;

    if(i==software.x_label || i==software.y_label || i==software.z_label || (i==software.a_label &&
software.a_status==ON))
    {
        _setcursortype(_NORMALCURSOR);

        gotoxy(6,6); cprintf("Enter New '%c' Value:",i); gotoxy(27,6);

        if(i==software.x_label)    {get_value_all(&dist_x);    gotoxy(4,3);    cprintf("<%c>    ->
%s",i,prn_val_all(dist_x));}

        if(i==software.y_label)    {get_value_all(&dist_y);    gotoxy(21,3);    cprintf("<%c>    ->
%s",i,prn_val_all(dist_y));}

        if(i==software.z_label)    {get_value_all(&dist_z);    gotoxy(4,4);    cprintf("<%c>    ->
%s",i,prn_val_all(dist_z));}

        if(i==software.a_label)    {get_value_all(&dist_a);    gotoxy(21,4);    cprintf("<%c>    ->
%s",i,prn_val_all(dist_a));}

        gotoxy(6,6); cprintf("    "); _setcursortype(_NOCURSOR);
    }
}

puttext(22,19,59,24,buffer);
if(i==27) break;
gettext(2,6,22,12,buffer4);
move_menu(2);
message("Machine Move In Progress.");
msg=move(&hardware, &software, &pt, &cur_speed, &ctrl.frate,
dist_x-(world.x_pos-gcode.x_coord[ctrl.cs])*ctrl.pt,
dist_y-(world.y_pos-gcode.y_coord[ctrl.cs])*ctrl.pt,
dist_z-(world.z_pos-gcode.z_coord[ctrl.cs])*ctrl.pt,
dist_a-(world.a_pos-gcode.a_coord[ctrl.cs])*ctrl.pt,
NULL,LINE_ENT,0,0,0,0,NULL,LINE_ENT);

message("");
puttext(2,6,22,12,buffer4);

break;

case 'p':
case 'P':

    if(strcmp(ctrl.file,"NONE")==0)

        {gettext(22,19,59,24,buffer);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

box(22,19,59,24, RED,BLACK,WHITE,WHITE);

gotoxy(4,3); cprintf("    No Program Loaded");

gotoxy(4,4); cprintf("    <Any Key>");

getch();

window(1,1,80,25);

puttext(22,19,59,24,buffer);

break;

}

gettext(22,19,59,24,buffer);

box(22,19,59,24, RED,BLACK,WHITE,WHITE);

gotoxy(5,1); cprintf("Program Control <Esc> To Exit");

gotoxy(4,3); cprintf("<?c> Skip Forward  <?c> Skip Back",25,24);

gotoxy(4,4); cprintf("<R> Rewind And Reset Program");

window(1,1,80,25);

while(1)

{while((ch=getch())!=27 && ch!=80 && ch!=72 && ch!='r' && ch!='R');

if(ch==27) break;

if(ch==80)

{while((ch2=fgetc(ifp))!='\n' && ch2!=EOF);

if(ch2==EOF) {message("End Of Program"); fsetpos(ifp,&fpos1); continue;}

fgetpos(ifp,&fpos1); fsetpos(ifp,&fpos2);

if(fgets(fstring, 200, ifp)!=NULL)

{fstring[36]=0; gotoxy(43,17); textcolor(WHITE); textbackground(BLACK);}

fgetpos(ifp,&fpos2); fsetpos(ifp,&fpos1);

message("Skipped Forward");

goloop=OFF;

}

if(ch==72)

{place_holder=ftell(ifp);

rewind(ifp);

if(ftell(ifp)==place_holder) {message("Beginning Of Program"); continue;}

fgetpos(ifp,&fpos3); i=0;

while(ftell(ifp)!=place_holder)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{fgetpos(ifp,&fpos1);
if(i<=6) i++;
if(i==7) {fsetpos(ifp,&fpos3); fgets(fstring, 200, ifp); fgetpos(ifp,&fpos3);
fsetpos(ifp,&fpos1);}
while((ch2=fgetc(ifp))!='\n' && ch2!=EOF && ftell(ifp)!=place_holder);
if(ch2==EOF) break;
}

if(i==7) {fstring[36]=0; }
fsetpos(ifp,&fpos1);
for(i=0;i<8;i++) fgets(fstring, 200, ifp);
fgetpos(ifp,&fpos2); fsetpos(ifp,&fpos1);
message("Skipped Back");
goloop=OFF;
}
if(ch=='r' || ch=='R')
{do_gcode(&hardware, &software, &gcode, &ctrl, &pt, 'M', 30);

rewind(ifp); fgetpos(ifp,&fpos1);
get_var(0,0,CLEAR);
for(i=0;i<8;i++)
{if(fgets(fstring, 200, ifp)!=NULL)
{fstring[36]=0; textcolor(WHITE); textbackground(BLACK); gotoxy(1,8+i);}
}
fgetpos(ifp,&fpos2); fsetpos(ifp, &fpos1);
window(1,1,80,25);
message("Program Start");
goloop=OFF;
}
}

puttext(22,19,59,24,buffer);
break;
case 18+1000:
case 'e':
case 'E':

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gettext(22,19,59,24,buffer);
box(22,19,59,24, RED,BLACK,WHITE,WHITE);
if(strcmp(ctrl.file,"NONE")==0)
{
    gotoxy(4,3); cprintf("    No Program Loaded");
    gotoxy(4,4); cprintf("    <Any Key>");
}
else
{
    gotoxy(4,3);
    if(kp==1018) cprintf("    Line By Line Execute?");
    else    cprintf("    Run Program");
    gotoxy(4,4); cprintf("<Enter>To Accept <Esc>To Cance");
}
window(1,1,80,25);
while(1)
{
    ch=getch();
    if(strcmp(ctrl.file,"NONE")==0) break;
    else if(ch==27 || ch==13 || ch==32) break;
}
puttext(22,19,59,24,buffer);
if(ch==27 || strcmp(ctrl.file,"NONE")==0) break;
BEGIN_EXEC:
msg=COMPLETE;
gettext(41,2,80,2,buffer);

gettext(2,6,22,12,buffer4);
move_menu(3);
if(ctrl.M00!=ON) {ctrl.M00=OFF; GMTtext(&software, &ctrl, &gcode);}
message("Program In Progress.");
if(kp==1018) do_gcode(&hardware, &software, &gcode, &ctrl, &pt, 1, 0); //line by line execute
else do_gcode(&hardware, &software, &gcode, &ctrl, &pt, 2, 0); //normal execute
while(1)
{
    ch=fgetc(ifp);
    if(ch=='(') while((ch=fgetc(ifp))!='\n' && ch!=EOF);
    if(ch==EOF) {msg=do_gcode(&hardware, &software, &gcode, &ctrl, &pt, 'M', 30); break;}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(ch>=97 && ch<=122) ch=ch-32;
if(ch=='V' && version!=LITE)
{
    if(NULL==fscanf(ifp, "%lf", &num)) continue;
    while((ch=fgetc(ifp))==' ' && ch!=EOF);
    if(ch==EOF) {msg=do_gcode(&hardware, &software, &gcode, &ctrl, &pt, 'M', 30); break;}
    if(ch=='=')
    {
        if(NULL==fscanf(ifp, "%lf", &num2)) continue;
        get_var(num,num2,STORE);
    }
}

if(ch=='L' && version!=LITE)
{
    if(goloop==ON || NULL==fscanf(ifp, "%lf", &loopmax)) {continue;}
    goloop=ON;
    loopnum=0;
}

if(ch=='X' || ch=='Y' || ch=='Z' || ch==software.a_label || ch=='I' || ch=='J' || ch=='K' ||
    ch=='R' || ch=='Q' || ch=='F' || ch=='G' || ch=='M' || ch=='P' || ch=='\n')
{
    if(ch=='\n')
    {
        num=NULL;
        if(goloop==ON) {loopnum++; if(loopnum>=abs(loopmax)) goloop=OFF; else
fsetpos(ifp,&fpos1);}
    }
    else
    {
        fgetpos(ifp,&fpostemp);
        if(NULL==fscanf(ifp, "%lf", &num))
        {
            fsetpos(ifp,&fpostemp);
            while((ch2=fgetc(ifp))==' ' && ch2!=EOF);
            if(ch2==EOF) {msg=do_gcode(&hardware, &software, &gcode, &ctrl, &pt, 'M', 30);
break;}}

        if((ch2=='V' || ch2=='v') && version!=LITE)
        {
            if(NULL==fscanf(ifp, "%lf", &num2)) continue;
            num=get_var(num2,0,GET);
        }
        else continue;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

//check for go to subroutine command
if(ch=='G' && num==22 && version!=LITE)
{
while((ch2=fgetc(ifp))==' ' && ch2!=EOF);
if(ch2==EOF) {msg=do_gcode(&hardware, &software, &gcode, &ctrl, &pt, 'M', 30);
break;}

if(ch2=='P' || ch2=='p')
{
if(NULL==fscanf(ifp, "%lf", &num2)) continue;
if(num2<0) continue;
for(i=0;i<100;i++) if(sub[i].id==num2) {gsub=ON; subpos=sub[i].pos;}
}
else continue;
}

//check for file chain command
else if(ch=='G' && num==95)
{
while((ch2=fgetc(ifp))==' ' && ch2!=EOF);
if(ch2=='#' && fscanf(ifp, "%s", tstring)!=NULL && (ifp2=fopen(tstring, "r"))!=NULL)
{
fclose(ifp);
ifp=ifp2;
while((ch=fgetc(ifp))!=EOF)
{
if(ch=='(')
{
while((ch=fgetc(ifp))!=EOF && ch!='\n');
if(ch==EOF) break;
}
if(ch=='M' || ch=='m')
{
if(NULL==fscanf(ifp, "%lf", &num)) continue;
if(num==30) break;

msg=do_gcode(&hardware, &software, &gcode, &ctrl, &world, 'M', 99);
}
}
else msg=do_gcode(&hardware, &software, &gcode, &ctrl, &world, ch, num);
//go to subroutine now if needed
if(ch=='\n' && gsub==ON)
{
gsub=OFF;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

fgetpos(ifp,&mainpos);
fsetpos(ifp,&subpos);
//do subroutine now
while(1)
{ch=fgetc(ifp);
if(ch=='(') while((ch=fgetc(ifp))!='\n' && ch!=EOF);
if(ch==EOF) break;
if(ch>=97 && ch<=122) ch=ch-32;
if(ch=='V')
{if(NULL==fscanf(ifp, "%lf", &num)) continue;
while((ch=fgetc(ifp))==' ' && ch!=EOF);
if(ch==EOF) {msg=do_gcode(&hardware, &software, &gcode, &ctrl, &pt, 'M', 30);
break;}}

if(ch=='=')
{if(NULL==fscanf(ifp, "%lf", &num2)) continue;
get_var(num,num2,STORE);
}
}

if(ch=='X' || ch=='Y' || ch=='Z' || ch==software.a_label || ch=='I' || ch=='J' || ch=='K' ||
ch=='R' || ch=='Q' || ch=='F' || ch=='G' || ch=='M' || ch=='P' || ch=='\n')
{if(ch=='\n') num=NULL;
else
{fgetpos(ifp,&fpostemp);
if(NULL==fscanf(ifp, "%lf", &num))
{fsetpos(ifp,&fpostemp);
while((ch2=fgetc(ifp))==' ' && ch2!=EOF);
if(ch2==EOF) {msg=do_gcode(&hardware, &software, &gcode, &ctrl, &pt, 'M', 30);
break;}}

if(ch2=='V' || ch2=='v')
{if(NULL==fscanf(ifp, "%lf", &num2)) continue;
num=get_var(num2,0,GET);
}
else continue;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if(ch=='M' && num==2) break;

        msg=do_gcode(&hardware, &software, &gcode, &ctrl, &world, ch, num);
    }

    if(msg==M30 || msg==M99 || msg==M0 || msg==QUIT || msg==E_STOP ||
(msg>=X_PLIMIT && msg<=A_NLIMIT)) break;
    }

    //end subroutine

    fsetpos(ifp,&mainpos);

    ch='\n';
}

}

if(ch=='\n' && goloop==OFF)
{
    fgetpos(ifp,&fpos1); fsetpos(ifp,&fpos2);
    if(fgets(fstring, 200, ifp)!=NULL)
        {fstring[36]=0; }
    fgetpos(ifp,&fpos2); fsetpos(ifp,&fpos1);
    if(kp==1018 && msg!=M30 && msg!=M0 && msg!=QUIT && msg!=E_STOP &&
(msg<X_PLIMIT || msg>A_NLIMIT))
    {update_display(&software, 0, 0, 0, 0, 0, 0, 3);
    update_display(&software, 0, 0, 0, 0, 0, 0, 5);
    if(ctrl.pty==REL)
        for(i=0;i<4;i++)
            {if(i==3) i++;
            update_display(&software, 0, 0, 0, 0, 0, 0, i);
            }
    puttext(2,6,22,12,buffer4);
    gettext(22,19,59,24,buffer5);
    box(22,19,59,24, RED,BLACK,WHITE,WHITE);
    gotoxy(4,3); cprintf("      Continue?");
    gotoxy(4,4); cprintf("<Enter> To Accept <Esc> To Abort");
    window(1,1,80,25);
    while(1)
        {ch=getch();
        if(ch==27 || ch==13 || ch==32) break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    puttext(22,19,59,24,buffer5);

    move_menu(3);

    if(ch==27) msg=do_gcode(&hardware, &software, &gcode, &ctrl, &pt, 'M', 0);
}

}

if(msg==M30 || msg==M99 || msg==M0 || msg==QUIT || msg==E_STOP ||(msg>=X_PLIMIT
&& msg<=A_NLIMIT)) break;
}

if(msg==M30 || msg==M99)
{window(43,3,78,18); textcolor(WHITE); textbackground(BLACK); clrscr();
rewind(ifp); fgetpos(ifp,&fpos1);
get_var(0,0,CLEAR);
for(i=0;i<8;i++)
{if(fgets(fstring, 200, ifp)!=NULL)
{fstring[36]=0; textcolor(WHITE); textbackground(BLACK); gotoxy(1,8+i); }
}
fgetpos(ifp,&fpos2); fsetpos(ifp, &fpos1);
window(1,1,80,25);
message("Program Start");
}

if(msg==M0) {ctrl.M00=ON; message("Program Paused...<E>/<Alt-E> To Cont.");}
else gotoop=OFF;

puttext(2,6,22,12,buffer4);
puttext(41,2,80,2,buffer);
if(msg==M99) goto BEGIN_EXEC;

break;

case 'a':
case 'A':

    gettext(22,18,59,25,buffer);

    box(22,18,59,25, RED,BLACK,WHITE,WHITE);

    gotoxy(12,1); cprintf("<Esc> To Exit");

    gotoxy(4,3); cprintf("<1> +90 About %c",software.x_label);

    gotoxy(21,3); cprintf("<2> -90 About %c",software.x_label);

    gotoxy(4,4); cprintf("<3> +90 About %c",software.y_label);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gotoxy(21,4); cprintf("<4> -90 About %c", software.y_label);
gotoxy(4,5); cprintf("<5> +90 About %c",software.z_label);
gotoxy(21,5); cprintf("<6> -90 About %c", software.z_label);
gotoxy(4,6); cprintf("<7> Switch %c With %c", software.z_label, software.a_label);
gotoxy(3,8); cprintf("Rotate Coord. System About An Axis");
while((i=getch())!=27)
{
    if(i>='1' && i<='7')
    {
        gettext(22,19,59,25,buffer2);
        box(22,19,59,25, RED,BLACK,WHITE,WHITE);
        gotoxy(7,3); cprintf("Are You Sure You Want To");
        gotoxy(6,4);
        if(i=='1') cprintf("Rotate +90 Degrees About %c",software.x_label);
        if(i=='2') cprintf("Rotate -90 Degrees About %c",software.x_label);
        if(i=='3') cprintf("Rotate +90 Degrees About %c",software.y_label);
        if(i=='4') cprintf("Rotate -90 Degrees About %c",software.y_label);
        if(i=='5') cprintf("Rotate +90 Degrees About %c",software.z_label);
        if(i=='6') cprintf("Rotate -90 Degrees About %c",software.z_label);
        if(i=='7') cprintf("Switch %c and %c",software.z_label,software.a_label);
        gotoxy(4,5); cprintf("<Enter> To Accept <Esc> To Abort");
        while((ch=getch())!=27 && ch!=13 && ch!=32);
        if(ch==13 || ch==32)
        {
            if(i=='1') rotate_x(&software, &hardware, &pt);
            if(i=='3') rotate_y(&software, &hardware, &pt);
            if(i=='5') rotate_z(&software, &hardware, &pt);
            if(i=='2') {rotate_x(&software, &hardware, &pt); rotate_x(&software, &hardware, &pt);
rotate_x(&software, &hardware, &pt);}
            if(i=='4') {rotate_y(&software, &hardware, &pt); rotate_y(&software, &hardware, &pt);
rotate_y(&software, &hardware, &pt);}
            if(i=='6') {rotate_z(&software, &hardware, &pt); rotate_z(&software, &hardware, &pt);
rotate_z(&software, &hardware, &pt);}
            if(i=='7') switch_z_a(&software, &hardware, &pt);
            update_display(&software, pt.x_pos, pt.y_pos, pt.z_pos, pt.a_pos, 0, 0, ctrl.frate, ALL);
        }
        puttext(22,19,59,25,buffer2);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    puttext(22,18,59,25,buffer);

    break;

case 27:

    gettext(22,19,59,24,buffer);

    box(22,19,59,24, RED,BLACK,WHITE,WHITE);


    gotoxy(4,4); printf("<Enter> To Accept <Esc> To Abort");

    while(1)

        {ch=getch();

        if(ch==27 || ch==13 || ch==32) break;

        }

    if(ch==27) kp=0;

    else if(version==DEMO) msg=QUIT;

    puttext(22,19,59,24,buffer);

    break;

}

switch(kp)

{case 31+1000: //alt s

case 33+1000: //alt f

case 24+1000: //alt o

case 71+1000: //home

case 18+1000: //alt e

case 'e': case 'E':

case 'u': case 'U':

    pt.x_pos=(world.x_pos-gcode.x_coord[ctrl.cs])*ctrl.pty;

    pt.y_pos=(world.y_pos-gcode.y_coord[ctrl.cs])*ctrl.pty;

    pt.z_pos=(world.z_pos-gcode.z_coord[ctrl.cs])*ctrl.pty;

    pt.a_pos=(world.a_pos-gcode.a_coord[ctrl.cs])*ctrl.pty;

    break;


    world.x_pos=world.x_pos-(world.x_pos-gcode.x_coord[ctrl.cs])*ctrl.pty+pt.x_pos;

    world.y_pos=world.y_pos-(world.y_pos-gcode.y_coord[ctrl.cs])*ctrl.pty+pt.y_pos;

    world.z_pos=world.z_pos-(world.z_pos-gcode.z_coord[ctrl.cs])*ctrl.pty+pt.z_pos;

    world.a_pos=world.a_pos-(world.a_pos-gcode.a_coord[ctrl.cs])*ctrl.pty+pt.a_pos;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    pt.x_pos=pt.x_pos*ctrl.pty; pt.y_pos=pt.y_pos*ctrl.pty;
    pt.z_pos=pt.z_pos*ctrl.pty; pt.a_pos=pt.a_pos*ctrl.pty;
    break;
}
if(msg>=X_PLIMIT && msg<=A_NLIMIT) msg=limit_msg(&hardware, &software, &pt, msg);
if(msg==E_STOP)
{if(ctrl.M03==ON)
    {if(hardware.io1_on==HIGH) clear_io_bit(&hardware, hardware.io1_bit);
    else set_io_bit(&hardware, hardware.io1_bit);
    ctrl.M03=OFF;}}
if(ctrl.M07==ON)
    {if(hardware.io2_on==HIGH) clear_io_bit(&hardware, hardware.io2_bit);
    else set_io_bit(&hardware, hardware.io2_bit);
    ctrl.M07=OFF;}}
if(ctrl.M08==ON)
    {if(hardware.io3_on==HIGH) clear_io_bit(&hardware, hardware.io3_bit);
    else set_io_bit(&hardware, hardware.io3_bit);
    ctrl.M08=OFF;}}
if(ctrl.M10==ON)
    {if(hardware.io4_on==HIGH) clear_io_bit(&hardware, hardware.io4_bit);
    else set_io_bit(&hardware, hardware.io4_bit);
    ctrl.M10=OFF;}}
estop_msg();
}
if(msg==BREAK) message("Machine Break Commanded.");
if(msg==QUIT) break;
control_text(&software,&pt,&ctrl,&gcode);
msg=COMPLETE;
}

```

//end realtime control

normvideo();

_setcursortype(_NORMALCURSOR);

window(1,1,80,25);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

clrscr();

if(strcmp(ctrl.file,"NONE")!=0) fclose(ifp);

save_setup(&hardware, &software, "SETUP.INI");

save_screen_setup(&world,&pt,&ctrl);

return 0;
}

void control_screen(struct soft_setup *software, struct pos *pt,
                    struct control *ctrl, struct gcode_setup *gcode)
{
    int i;

    window(1,1,80,25); textbackground(BLACK); clrscr();
    window(1,1,80,25); textbackground(BLACK); textcolor(WHITE);

    textbackground(BLACK);
    gotoxy(40,7); cprintf(" <---Right Arrow + , Left Arrow -",26);

    gotoxy(40,9); cprintf(" <---Up Arrow + , Down Arrow -",24);

    gotoxy(40,11); cprintf(" <---Page Up + , Page Down -");

    gotoxy(30,13); cprintf("Set X,Y,Z Press Z");
    gotoxy(2,7); cprintf("Run G-code Press E");

    gotoxy(65,1); cprintf("Exit Press ESC");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gotoxy(42,10); putch(26);

control_text(software, pt, ctrl, gcode);
}

void control_text(struct soft_setup *software, struct pos *pt,
                 struct control *ctrl, struct gcode_setup *gcode)
{
    window(1,1,80,25); textbackground(BLACK); textcolor(WHITE);
    update_display(software, (*pt).x_pos, (*pt).y_pos, (*pt).z_pos, (*pt).a_pos, 0, 0, (*ctrl).frate, ALL);

    textcolor(GREEN);
    gotoxy(19,16); cprintf("  Feed Rate");
    textcolor(WHITE);
    gotoxy(40,16); cprintf(" <--Adjust Feed Rate Press F");
    gotoxy(2,5); cprintf(" Loaded : %5s",(*ctrl).file);

    gotoxy(2,23); cprintf("      +++Gu Surasit Suntikasem Tum For Puak Mung End Na Kuy+++");
    if((*software).a_status==OFF)

    if((*software).a_status==OFF)
    textcolor(GREEN);
    gotoxy(27,7); cprintf("X");
    gotoxy(27,9); cprintf("Y");
    gotoxy(27,11); cprintf("Z");

    GMTtext(software,ctrl,gcode);
}

void GMTtext(struct soft_setup *software, struct control *ctrl, struct gcode_setup *gcode)
{
    window(1,1,80,25);
    textcolor(WHITE); textbackground(BLACK);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(version==LITE) {gotoxy(44,20);}

if((*software).a_status==ON)
{if((*software).a_follow==INDEPENDANT)                                {gotoxy(6 0 , 2 4 ) ;
cprintf("%c(Independent)",(*software).a_label);}
    if((*software).a_follow==FOLLOW_X)        {gotoxy(6 0 , 2 4 ) ;        cprintf("%c(Follow      X)
",(*software).a_label);}
    if((*software).a_follow==FOLLOW_Y)        {gotoxy(6 0 , 2 4 ) ;        cprintf("%c(Follow      Y)
",(*software).a_label);}
    if((*software).a_follow==FOLLOW_Z)        {gotoxy(6 0 , 2 4 ) ;        cprintf("%c(Follow      Z)
",(*software).a_label);}
}
}

int limit_msg(struct hard_setup *hardware, struct soft_setup *software, struct pos *pt, int msg)
{
    int xp=1, xn=1, yp=1, yn=1, zp=1, zn=1, ap=1, an=1, ch, temp_a_follow=(*software).a_follow;
    static BYTE buffer[1000];
    char letter;
    gettext(22,19,59,24,buffer);
    box(22,19,59,24, RED,BLACK,WHITE,WHITE);
    gotoxy(8,3); cprintf("All Processes Aborted");
    gotoxy(5,4); cprintf("Use Jog Keys To Clear Limits");
    if(msg==X_PLIMIT) {xp=0; letter='X';} if(msg==X_NLIMIT) {xn=0; letter='X';} if(msg==Y_PLIMIT)
{yp=0; letter='Y';} if(msg==Y_NLIMIT) {yn=0; letter='Y';}
    if(msg==Z_PLIMIT) {zp=0; letter='Z';} if(msg==Z_NLIMIT) {zn=0; letter='Z';} if(msg==A_PLIMIT)
{ap=0; letter=(*software).a_label;} if(msg==A_NLIMIT) {an=0; letter=(*software).a_label;}
    textcolor(YELLOW+BLINK); gotoxy(8,1); cprintf("<<°c Axis Limit Override>>",letter);
    (*software).a_follow=INDEPENDANT;
    while(xp==0 || xn==0 || yn==0 || yp==0 || zn==0 || zp==0 || an==0 || ap==0)
    {ch=getch(); if(ch==0) ch=getch()+1000;
    if(ch==1014) break;
    switch(ch)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{case 72+1000: //up
case 77+1000: //right
    if((ch==72+1000 && (*software).invert_arrows==OFF) || (ch==77+1000 &&
(*software).invert_arrows==ON))
        {if((msg=jog(hardware, software, pt, SLOW, NULL, Y_AXIS*yp, POS))==UNLIMIT) yn=1;}
        else if((msg=jog(hardware, software, pt, SLOW, NULL, X_AXIS*xp, POS))==UNLIMIT) xn=1;
        break;
case 80+1000: //down
case 75+1000: //left
    if((ch==80+1000 && (*software).invert_arrows==OFF) || (ch==75+1000 &&
(*software).invert_arrows==ON))
        {if((msg=jog(hardware, software, pt, SLOW, NULL, Y_AXIS*yn, NEG))==UNLIMIT) yp=1;}
        else if((msg=jog(hardware, software, pt, SLOW, NULL, X_AXIS*xn, NEG))==UNLIMIT) xp=1;
        break;
case 73+1000: //page up
    if((msg=jog(hardware, software, pt, SLOW, NULL, Z_AXIS*zp, POS))==UNLIMIT) zn=1;
    break;
case 81+1000: //page down
    if((msg=jog(hardware, software, pt, SLOW, NULL, Z_AXIS*zn, NEG))==UNLIMIT) zp=1;
    break;
case '+':
case '=':
    if((msg=jog(hardware, software, pt, SLOW, NULL, A_AXIS*ap, POS))==UNLIMIT) an=1;
    break;
case '-':
    if((msg=jog(hardware, software, pt, SLOW, NULL, A_AXIS*an, NEG))==UNLIMIT) ap=1;
    break;
}

if(msg==E_STOP || msg==QUIT) break;

if(msg==X_PLIMIT) xp=0; if(msg==X_NLIMIT) xn=0; if(msg==Y_PLIMIT) yp=0;
if(msg==Y_NLIMIT) yn=0;
if(msg==Z_PLIMIT) zp=0; if(msg==Z_NLIMIT) zn=0; if(msg==A_PLIMIT) ap=0;
if(msg==A_NLIMIT) an=0;
msg=COMPLETE;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(*software).a_follow=temp_a_follow;

puttext(22,19,59,24,buffer);

return msg;
}

void estop_msg(void)
{
    static BYTE buffer[1000];
    gettext(22,19,59,24,buffer);
    box(22,19,59,24, RED,BLACK,WHITE,WHITE);
    gotoxy(10,3); cprintf("All Processes Aborted");
    gotoxy(16,4); cprintf("<Any Key>");
    textcolor(YELLOW+BLINK); gotoxy(11,1); cprintf("<<Emergeny Stop>>");
    getch();
    puttext(22,19,59,24,buffer);
}

void message(char msg[40])
{
    textcolor(WHITE);
    textbackground(BLACK);
    window(1,1,80,25);
    gotoxy(2,3); cprintf(" ");
    gotoxy(2,3); cprintf("%s",msg);
}

void move_menu(int id)
{
    int i;
    window(2,6,22,12);
    textbackground(GREEN); textcolor(WHITE); clrscr();
    window(1,1,80,25);
    gotoxy(2, 6);  putch(218);
    for(i=3; i<22; i++) putch(196);
    putch(191);
    for(i=7; i<12; i++)
    {gotoxy(2, i);  putch(179);
      gotoxy(22, i); putch(179);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    gotoxy(2, 12); putchar(192);
    for (i=3; i<22; i++) putchar(196);
    putchar(217);

    gotoxy(3,7); cprintf("  Move Control");
    gotoxy(3,8); cprintf("<Esc>...Stop");
    if(id==2 || id==3) {gotoxy(3,9); cprintf("<F>.....+Feed Rate");
        gotoxy(3,10); cprintf("<S>.....-Feed Rate");}
    if(id==3)      {gotoxy(3,11); cprintf("<P>.....Pause");}
}

double get_var(double num, double num2, int action)
{
    static double V[101];
    int i;
    if(num>100 || num<0) return 0;
    if(action==CLEAR) for(i=0;i<101;i++) V[i]=0;
    if(action==STORE) V[num]=num2;
    if(action==GET) return V[num];
    return 1;
}

#include <stdio.h>
#include <stdlib.h>
#include <dos.h>
#include <math.h>
#include <conio.h>
#include "definitions.h"
#include "kernel.h"

extern int version;

void video_write(BYTE, BYTE, double, int, int);

int move_machine(struct hard_setup *, struct soft_setup *, struct pos *,
    float *, float *, float, double, double, double, double, double);

double get_unit_vect(struct unit_vect *, double, double, double, double, double, int);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

float max_speed_sq(struct hard_setup *,
                  float, float, float, float, float,
                  struct unit_vect *);

float max_speed_sq_line(struct hard_setup *, double, double, double, double, double);

int backlash(struct hard_setup *, struct soft_setup *, struct unit_vect*, struct unit_vect*);

int backlash_to_come(struct hard_setup *, struct unit_vect *, struct unit_vect *, int, int);

float get_max_speed(struct hard_setup *, float, float, float, float, float);

float get_max_accel(struct hard_setup *, float, float, float, float, float);

float get_min_speed(struct hard_setup *, float, float, float, float, float);

float get_max_4(float, float, float, float);

int sign(float);

double total_vect(double, double, double, double);

void _interrupt_far new_direct_timer(void);

void set_timer_freq(unsigned long);

void _interrupt_far MyKeyboardRoutine(void);

void init_key(void);

void restore_key(void);

//hardware bit manipulation functions

void clear_data_bits(struct hard_setup *hard_config)
{
    outportb((*hard_config).port, 0);
}

void set_io_bit(struct hard_setup *hard_config, int bit)
{
    outportb((*hard_config).port+2, ((inportb((*hard_config).port+2)^11)|bit)^11);
}

void clear_io_bit(struct hard_setup *hard_config, int bit)
{
    BYTE iobits;

    iobits=(inportb((*hard_config).port+2)^11)|bit;
    outportb((*hard_config).port+2, (iobits&(iobits-bit))^11);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

void toggle_io_bit(struct hard_setup *hard_config, int bit)
{
    bit=pow(2,bit);
    outportb((*hard_config).port+2, ((inportb((*hard_config).port+2)^11)^bit^11);
}

//update the display
void update_display(struct soft_setup *soft_config, double x, double y, double z,
                    double a, float speed, float a_speed, float f_rate, int item)
{
    if ((*soft_config).display_x && (item == 0 || item == ALL))
        video_write((*soft_config).x_xpos,(*soft_config).x_ypos, x, RED, LONG);
    if ((*soft_config).display_y && (item == 1 || item == ALL))
        video_write((*soft_config).y_xpos,(*soft_config).y_ypos, y, RED, LONG);
    if ((*soft_config).display_z && (item == 2 || item == ALL))
        video_write((*soft_config).z_xpos,(*soft_config).z_ypos, z, RED, LONG);
    if ((*soft_config).display_s && (item == 3 || item == ALL))
        video_write((*soft_config).s_xpos,(*soft_config).s_ypos, speed*(*soft_config).time, BLACK,
SHORT);
    if ((*soft_config).display_a && (item == 4 || item == ALL))
        video_write((*soft_config).a_xpos,(*soft_config).a_ypos, a, RED, LONG);
    if ((*soft_config).display_as && (item == 5 || item == ALL))
        video_write((*soft_config).as_xpos,(*soft_config).as_ypos, a_speed*(*soft_config).a_time, BLACK,
SHORT);
    if ((*soft_config).display_fr && (item == 6 || item == ALL))
        video_write((*soft_config).fr_xpos,(*soft_config).fr_ypos, f_rate*(*soft_config).time, BLACK,
SHORT);
}

void video_write(BYTE x, BYTE y, double dnum, int bkg, int lgth)
{
    BYTE_far *video=(BYTE_far *)0xB8000000L; /* Ptr to base of VGA text RAM */
    BYTE unit;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

WORD yoff=(y-1)*160, xoff=(x-1)*2;
int s=sign(dnum), i;
double temp;
unsigned long lnum=(unsigned long)(temp=s*dnum*(10000)), den=10000000;

if (temp-lnum >= .5)
    lnum++;
if (lgth == SHORT && lnum/100.0-lnum/100>=.5)
    lnum=lnum+100;
if (lnum == 0)
    s=1;
if (lnum>99999999)
    lnum=0;
for (i=1; i<17+lgth*4; i+=2)
    *(video+(WORD)(xoff+i+yoff))=WHITE+bkg*16;
*(video+(WORD)(xoff+10+yoff))='.';
if (s<0)
    *(video+(WORD)(xoff+yoff))='-';
else
    *(video+(WORD)(xoff+yoff))=' ';
for (i=2; i<16+lgth*4; i+=2) {
    if (i == 10)
        i+=2;
    unit=lnum/den;
    lnum=lnum-unit*den;
    den=den/10;
    *(video+(WORD)(xoff+i+yoff))=unit+48;
}
}

//realtime function for moving device
volatile int tick;
volatile int key, jogging;
int move_machine(struct hard_setup *hard_config, struct soft_setup *soft_config, struct pos *current,
    float *cur_speed, float *sp_limit, float max_end_speed_sq,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        double dist_x, double dist_y, double dist_z, double dist_a, double total_vect)
{
    double current_x=0, current_y=0, current_z=0, current_a=0, current_vect=0, temp;
    static double inc_x, inc_y, inc_z, inc_a;
    static float cur_debo_x, cur_debo_y, cur_debo_z, cur_debo_a;
    static float display_count;
    static float cur_x_sp, cur_y_sp, cur_z_sp, cur_a_sp;
    float speed, max_accel, cur_decel, max_speed, min_speed, speed_inc;
    static int x_limit=ON, y_limit=ON, z_limit=ON, a_limit=ON, display;
    static int pxs=1, pys=1, pzs=1, pas=1, x_step=LOW, y_step=LOW, z_step=LOW, a_step=LOW,
act_step=HIGH;
    int xs=sign(dist_x), ys=sign(dist_y), zs=sign(dist_z), as=sign(dist_a);
    int msg=COMPLETE, step_flag=OFF, break_flag=OFF, accelerate=NORMAL;
    int acc_x=NORMAL, acc_y=NORMAL, acc_z=NORMAL, acc_a=NORMAL;
    BYTE dir=0, s_port_read, c_port_read;
    static BYTE step;

    //check step activate parameter and change if needed
    if (act_step != (*hard_config).act_step) {
        act_step=(*hard_config).act_step;
        if (x_step == LOW)
            x_step=HIGH;
        else
            x_step=LOW;
        if (y_step == LOW)
            y_step=HIGH;
        else
            y_step=LOW;
        if (z_step == LOW)
            z_step=HIGH;
        else
            z_step=LOW;
        if (a_step == LOW)
            a_step=HIGH;
        else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

a_step=LOW;
if ((step & (*hard_config).x_sbit) == 0)
    step=step+(*hard_config).x_sbit;
else
    step=step-(*hard_config).x_sbit;
if ((step & (*hard_config).y_sbit) == 0)
    step=step+(*hard_config).y_sbit;
else
    step=step-(*hard_config).y_sbit;
if ((step & (*hard_config).z_sbit) == 0)
    step=step+(*hard_config).z_sbit;
else
    step=step-(*hard_config).z_sbit;
if ((step & (*hard_config).a_sbit) == 0)
    step=step+(*hard_config).a_sbit;
else
    step=step-(*hard_config).a_sbit;
outportb((*hard_config).port, step);
}

//set up auxiliary channel for move
if ((*soft_config).a_status == OFF)
    dist_a=0;
else
    if (dist_a == 0) {
        if ((*soft_config).a_follow == FOLLOW_X) {
            dist_a=dist_x;
            as=xs;
        }
        if ((*soft_config).a_follow == FOLLOW_Y) {
            dist_a=dist_y;
            as=ys;
        }
        if ((*soft_config).a_follow == FOLLOW_Z) {
            dist_a=dist_z;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    as=zs;
}

if ((*soft_config).a_follow != INDEPENDANT) {
    if ((*soft_config).units == IN && (*soft_config).a_units == MM)
        dist_a=dist_a*25.4;
    else
        if ((*soft_config).units == MM && (*soft_config).a_units == IN)
            dist_a=dist_a/25.4;
}
}

//return if vector is zero length
if (dist_x == 0 && dist_y == 0 && dist_z == 0 && dist_a == 0)
    return COMPLETE;

//reset limit switches to original state if limits are turned off
if ((*soft_config).limit == OFF) {
    x_limit=y_limit=z_limit=a_limit=ON;
    cur_debo_x=cur_debo_y=cur_debo_z=cur_debo_a=0;
}

//check for invalid speed parameter
if (fabs(*sp_limit)<.00001)
    return INVALID_PARAMETER;

//define directions
if ((*hard_config).x_pdir == HIGH && dist_x>0)
    dir=dir+(*hard_config).x_dbit;
if ((*hard_config).x_pdir == LOW && dist_x<0)
    dir=dir+(*hard_config).x_dbit;
if ((*hard_config).y_pdir == HIGH && dist_y>0)
    dir=dir+(*hard_config).y_dbit;
if ((*hard_config).y_pdir == LOW && dist_y<0)
    dir=dir+(*hard_config).y_dbit;
if ((*hard_config).z_pdir == HIGH && dist_z>0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    dir=dir+(*hard_config).z_dbit;
if ((*hard_config).z_pdir == LOW && dist_z<0)
    dir=dir+(*hard_config).z_dbit;
if ((*hard_config).a_pdir == HIGH && dist_a>0 && (*soft_config).a_status == ON)
    dir=dir+(*hard_config).a_dbit;
if ((*hard_config).a_pdir == LOW && dist_a<0 && (*soft_config).a_status == ON)
    dir=dir+(*hard_config).a_dbit;
if ((*soft_config).a_status == OFF)
    dir=dir+(*hard_config).a_dbit;

```

```

//set up increment for new direction if sign change

```

```

if (pxs != xs) {
    if (act_step == HIGH && (step&(*hard_config).x_sbit) == (*hard_config).x_sbit)
        step=step-(*hard_config).x_sbit;
    if (act_step == LOW && (step&(*hard_config).x_sbit) == 0)
        step=step+(*hard_config).x_sbit;
    if (x_step == HIGH)
        x_step=LOW;
    else
        x_step=HIGH;
    inc_x=(*hard_config).x_per_step/2-inc_x;
}
if (pys != ys) {
    if (act_step == HIGH && (step&(*hard_config).y_sbit) == (*hard_config).y_sbit)
        step=step-(*hard_config).y_sbit;
    if (act_step == LOW && (step&(*hard_config).y_sbit) == 0)
        step=step+(*hard_config).y_sbit;
    if (y_step == HIGH)
        y_step=LOW;
    else
        y_step=HIGH;
    inc_y=(*hard_config).y_per_step/2-inc_y;
}
if (pzs != zs) {
    if (act_step == HIGH && (step&(*hard_config).z_sbit) == (*hard_config).z_sbit)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    step=step-(*hard_config).z_sbit;
if (act_step == LOW && (step&(*hard_config).z_sbit) == 0)
    step=step+(*hard_config).z_sbit;
if (z_step == HIGH)
    z_step=LOW;
else
    z_step=HIGH;
inc_z=(*hard_config).z_per_step/2-inc_z;
}

if (pas != as) {
    if (act_step == HIGH && (step&(*hard_config).a_sbit) == (*hard_config).a_sbit)
        step=step-(*hard_config).a_sbit;
    if (act_step == LOW && (step&(*hard_config).a_sbit) == 0)
        step=step+(*hard_config).a_sbit;
    if (a_step == HIGH)
        a_step=LOW;
    else
        a_step=HIGH;
    inc_a=(*hard_config).a_per_step/2-inc_a;
}

pxs=xs;
pys=ys;
pzs=zs;
pas=as;
outportb((*hard_config).port, dir+step);

if (*sp_limit>0) {
//define speed inc.
    if (dist_x == 0 && dist_y == 0 && dist_z == 0)
        speed_inc=(*soft_config).a_speed_inc;
    else
        speed_inc=(*soft_config).speed_inc;

//define maximum acceleration
    max_accel=get_max_accel(hard_config, dist_x, dist_y, dist_z, dist_a, total_vect);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//define maximum speed
max_speed=get_max_speed(hard_config, dist_x, dist_y, dist_z, dist_a, total_vect);

//define minimum speed
min_speed=get_min_speed(hard_config, dist_x, dist_y, dist_z, dist_a, total_vect);

//reset speeds accordingly
if (*cur_speed<min_speed && *sp_limit>=min_speed)
    *cur_speed=min_speed;
if (*cur_speed<*sp_limit && *sp_limit<min_speed)
    *cur_speed=min_speed=*sp_limit;
}
else
    *cur_speed=0;

//begin control loop
while (1) {
    while(tick){}
    tick=1;

    //read status and control ports
    if ((*soft_config).limit == ON || (*soft_config).e_stop != OFF)
        s_port_read=(248&inportb((*hard_config).port+1))^128;
    if ((*soft_config).io1_cmm == ON || (*soft_config).io2_cmm == ON ||
        (*soft_config).io3_cmm == ON || (*soft_config).io4_cmm == ON)
        c_port_read=(15&inportb((*hard_config).port+2))^11;

    //check for e-stop button or hold line trip
    if (((*soft_config).e_stop != OFF && (*hard_config).e_trip*(*hard_config).e_bit-
(s_port_read&(*hard_config).e_bit) == 0) {
        if ((*soft_config).e_stop == E_STOP_L) {
            cur_x_sp=cur_y_sp=cur_z_sp=cur_a_sp=0;
            msg=E_STOP;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
    while ((*hard_config).e_trip*(*hard_config).e_bit-(s_port_read&(*hard_config).e_bit) == 0) {
        s_port_read=(248&inportb((*hard_config).port+1))^128;
        //check for ESC key
        if (key == 0x01) {
            key=0;
            msg=E_STOP;
            break;
        }
    }
    if (msg == E_STOP)
        break;
}
if ((*soft_config).limit == ON) { //check for limit switch trip
    if (x_limit == ON) {
        if ((*hard_config).x_ltrip*(*hard_config).x_lbit-(s_port_read&(*hard_config).x_lbit) == 0) {
            if (*sp_limit>0)
                cur_debo_x=cur_debo_x+cur_speed*(*soft_config).dt*dist_x*xs/total_vect;
            else
                cur_debo_x=cur_debo_x+cur_x_sp*(*soft_config).dt;
            if (cur_debo_x>(*hard_config).debo_x) {
                cur_x_sp=0;
                cur_debo_x=0;
                x_limit=OFF;
                if (xs>0)
                    msg=X_PLIMIT;
                else
                    msg=X_NLIMIT;
                break;
            }
        }
    }
    else
        cur_debo_x=0;
}
if (y_limit == ON) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if ((*hard_config).y_ltrip*(*hard_config).y_lbit-(s_port_read&(*hard_config).y_lbit) == 0) {
    if (*sp_limit>0)
        cur_debo_y=cur_debo_y+cur_speed*(*soft_config).dt*dist_y*ys/total_vect;
    else
        cur_debo_y=cur_debo_y+cur_y_sp*(*soft_config).dt;
    if (cur_debo_y>(*hard_config).debo_y) {
        cur_y_sp=0;
        cur_debo_y=0;
        y_limit=OFF;
        if (ys>0)
            msg=Y_PLIMIT;
        else
            msg=Y_NLIMIT;
        break;
    }
}
else
    cur_debo_y=0;
}

if (z_limit == ON) {
    if ((*hard_config).z_ltrip*(*hard_config).z_lbit-(s_port_read&(*hard_config).z_lbit) == 0) {
        if (*sp_limit>0)
            cur_debo_z=cur_debo_z+cur_speed*(*soft_config).dt*dist_z*zs/total_vect;
        else
            cur_debo_z=cur_debo_z+cur_z_sp*(*soft_config).dt;
        if (cur_debo_z>(*hard_config).debo_z) {
            cur_z_sp=0;
            cur_debo_z=0;
            z_limit=OFF;
            if (zs>0)
                msg=Z_PLIMIT;
            else
                msg=Z_NLIMIT;
            break;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

else

    cur_debo_z=0;

}

if (a_limit == ON && (*soft_config).a_status == ON) {

    if ((*hard_config).a_ltrip*(*hard_config).a_lbit-(s_port_read&(*hard_config).a_lbit) == 0) {

        if (*sp_limit>0)

            cur_debo_a=cur_debo_a+cur_speed*(*soft_config).dt*dist_a*as/total_vect;

        else

            cur_debo_a=cur_debo_a+cur_a_sp*(*soft_config).dt;

        if (cur_debo_a>(*hard_config).debo_a) {

            cur_a_sp=0;

            cur_debo_a=0;

            a_limit=OFF;

            if (as>0)

                msg=A_PLIMIT;

            else

                msg=A_NLIMIT;

            break;

        }

    }

    else

        cur_debo_a=0;

}

//check for limit switch release

if (x_limit == OFF) {

    if ((*hard_config).x_ltrip*(*hard_config).x_lbit-(s_port_read&(*hard_config).x_lbit) != 0) {

        if (*sp_limit>0)

            cur_debo_x=cur_debo_x+cur_speed*(*soft_config).dt*dist_x*xs/total_vect;

        else

            cur_debo_x=cur_debo_x+cur_x_sp*(*soft_config).dt;

        if (cur_debo_x>(*hard_config).debo_x) {

            cur_debo_x=0;

            x_limit=ON;

            msg=UNLIMIT;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break;
    }
}
else
    cur_debo_x=0;
if (y_limit == OFF) {
    if ((*hard_config).y_ltrip>(*hard_config).y_lbit(s_port_read&(*hard_config).y_lbit) != 0) {
        if (*sp_limit>0)
            cur_debo_y=cur_debo_y+cur_speed>(*soft_config).dt*dist_y*ys/total_vect;
        else
            cur_debo_y=cur_debo_y+cur_y_sp>(*soft_config).dt;
        if (cur_debo_y>(*hard_config).debo_y) {
            cur_debo_y=0;
            y_limit=ON;
            msg=UNLIMIT;
            break;
        }
    }
    else
        cur_debo_y=0;
}
if (z_limit == OFF) {
    if ((*hard_config).z_ltrip>(*hard_config).z_lbit(s_port_read&(*hard_config).z_lbit) != 0) {
        if (*sp_limit>0)
            cur_debo_z=cur_debo_z+cur_speed>(*soft_config).dt*dist_z*zs/total_vect;
        else
            cur_debo_z=cur_debo_z+cur_z_sp>(*soft_config).dt;
        if (cur_debo_z>(*hard_config).debo_z) {
            cur_debo_z=0;
            z_limit=ON;
            msg=UNLIMIT;
            break;
        }
    }
    else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    cur_debo_z=0;
}
if (a_limit == OFF && (*soft_config).a_status == ON) {
    if ((*hard_config).a_ltrip*(*hard_config).a_lbit-(s_port_read&(*hard_config).a_lbit) != 0) {
        if (*sp_limit>0)
            cur_debo_a=cur_debo_a+cur_speed*(*soft_config).dt*dist_a*as/total_vect;
        else
            cur_debo_a=cur_debo_a+cur_a_sp*(*soft_config).dt;
        if (cur_debo_a>(*hard_config).debo_a) {
            cur_debo_a=0;
            a_limit=ON;
            msg=UNLIMIT;
            break;
        }
    }
    else cur_debo_a=0;
}
}
if (*sp_limit>0){
    //set acceleration flag to decel if applicable
    if (accelerate == NORMAL && total_vect>current_vect+.0001) {
        cur_decel=(max_end_speed_sq*cur_speed*cur_speed)/(2*(total_vect-current_vect));
        if (cur_decel<=max_accel)
            accelerate=DECEL;
    }
    //do acceleration routine and define speed
    if (accelerate == NORMAL) {
        if (*cur_speed<=*sp_limit) {
            *cur_speed=*cur_speed+max_accel*(*soft_config).dt;
            if (*cur_speed>min(*sp_limit, max_speed))
                *cur_speed=min(*sp_limit, max_speed);
        }
        else {
            *cur_speed=*cur_speed-max_accel*(*soft_config).dt;
            if (*cur_speed<min(*sp_limit, max_speed))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        *cur_speed=min(*sp_limit, max_speed);
    }
}

if (accelerate == DECEL) {
    *cur_speed=*cur_speed-max_accel*(*soft_config).dt;
    if (*cur_speed<min_speed)
        *cur_speed=min_speed;
}

//increment position
current_vect=current_vect+*cur_speed*(*soft_config).dt;
temp=*cur_speed*(*soft_config).dt/total_vect;
if (current_x*xs<dist_x*xs) {
    current_x=current_x+temp*dist_x;
    inc_x=inc_x+temp*dist_x*xs;
}
if (current_y*ys<dist_y*ys) {
    current_y=current_y+temp*dist_y;
    inc_y=inc_y+temp*dist_y*ys;
}
if (current_z*zs<dist_z*zs) {
    current_z=current_z+temp*dist_z;
    inc_z=inc_z+temp*dist_z*zs;
}
if (current_a*as<dist_a*as) {
    current_a=current_a+temp*dist_a;
    inc_a=inc_a+temp*dist_a*as;
}

if (dist_x*xs<=current_x*xs) {
    inc_x=inc_x-(current_x*xs-dist_x*xs);
    current_x=dist_x;
}
if (dist_y*ys<=current_y*ys) {
    inc_y=inc_y-(current_y*ys-dist_y*ys);
    current_y=dist_y;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (dist_z*zs<=current_z*zs) {
    inc_z=inc_z-(current_z*zs-dist_z*zs);
    current_z=dist_z;
}

if (dist_a*as<=current_a*as) {
    inc_a=inc_a-(current_a*as-dist_a*as);
    current_a=dist_a;
}

}

else { //if fast non-coordinated move
    //set acceleration flag to decel if applicable
    if (acc_x == NORMAL && dist_x*xs>current_x*xs+.0001) {
        cur_decel=((*hard_config).min_vel_x*(*hard_config).min_vel_x-
cur_x_sp*cur_x_sp)/(2*(dist_x*xs-current_x*xs));
        if (cur_decel<=(*hard_config).accel_x)
            acc_x=DECEL;
    }

    if (acc_y == NORMAL && dist_y*ys>current_y*ys+.0001) {
        cur_decel=((*hard_config).min_vel_y*(*hard_config).min_vel_y-
cur_y_sp*cur_y_sp)/(2*(dist_y*ys-current_y*ys));
        if (cur_decel<=(*hard_config).accel_y)
            acc_y=DECEL;
    }

    if (acc_z == NORMAL && dist_z*zs>current_z*zs+.0001) {
        cur_decel=((*hard_config).min_vel_z*(*hard_config).min_vel_z-
cur_z_sp*cur_z_sp)/(2*(dist_z*zs-current_z*zs));
        if (cur_decel<=(*hard_config).accel_z)
            acc_z=DECEL;
    }

    if (acc_a == NORMAL && dist_a*as>current_a*as+.0001) {
        cur_decel=((*hard_config).min_vel_a*(*hard_config).min_vel_a-
cur_a_sp*cur_a_sp)/(2*(dist_a*as-current_a*as));
        if (cur_decel<=(*hard_config).accel_a)
            acc_a=DECEL;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//do acceleration routine and define speed
if (acc_x == NORMAL && cur_x_sp < (*hard_config).vel_x && dist_x*xs > current_x*xs) {
    cur_x_sp = max(cur_x_sp + (*hard_config).accel_x * (*soft_config).dt, (*hard_config).min_vel_x);
    if (cur_x_sp > (*hard_config).vel_x)
        cur_x_sp = (*hard_config).vel_x;
}

if (acc_x == DECEL && cur_x_sp > (*hard_config).min_vel_x && dist_x*xs > current_x*xs) {
    cur_x_sp = cur_x_sp - (*hard_config).accel_x * (*soft_config).dt;
    if (cur_x_sp < (*hard_config).min_vel_x)
        cur_x_sp = (*hard_config).min_vel_x;
}

if (acc_y == NORMAL && cur_y_sp < (*hard_config).vel_y && dist_y*ys > current_y*ys) {
    cur_y_sp = max(cur_y_sp + (*hard_config).accel_y * (*soft_config).dt, (*hard_config).min_vel_y);
    if (cur_y_sp > (*hard_config).vel_y)
        cur_y_sp = (*hard_config).vel_y;
}

if (acc_y == DECEL && cur_y_sp > (*hard_config).min_vel_y && dist_y*ys > current_y*ys) {
    cur_y_sp = cur_y_sp - (*hard_config).accel_y * (*soft_config).dt;
    if (cur_y_sp < (*hard_config).min_vel_y)
        cur_y_sp = (*hard_config).min_vel_y;
}

if (acc_z == NORMAL && cur_z_sp < (*hard_config).vel_z && dist_z*zs > current_z*zs) {
    cur_z_sp = max(cur_z_sp + (*hard_config).accel_z * (*soft_config).dt, (*hard_config).min_vel_z);
    if (cur_z_sp > (*hard_config).vel_z)
        cur_z_sp = (*hard_config).vel_z;
}

if (acc_z == DECEL && cur_z_sp > (*hard_config).min_vel_z && dist_z*zs > current_z*zs) {
    cur_z_sp = cur_z_sp - (*hard_config).accel_z * (*soft_config).dt;
    if (cur_z_sp < (*hard_config).min_vel_z) cur_z_sp = (*hard_config).min_vel_z;
}

if (acc_a == NORMAL && cur_a_sp < (*hard_config).vel_a && dist_a*as > current_a*as) {
    cur_a_sp = max(cur_a_sp + (*hard_config).accel_a * (*soft_config).dt, (*hard_config).min_vel_a);
    if (cur_a_sp > (*hard_config).vel_a)
        cur_a_sp = (*hard_config).vel_a;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (acc_a == DECEL && cur_a_sp > (*hard_config).min_vel_a && dist_a*as > current_a*as) {
    cur_a_sp = cur_a_sp - (*hard_config).accel_a * (*soft_config).dt;
    if (cur_a_sp < (*hard_config).min_vel_a)
        cur_a_sp = (*hard_config).min_vel_a;
}

//increment position
current_x = current_x + cur_x_sp * (*soft_config).dt * xs;
current_y = current_y + cur_y_sp * (*soft_config).dt * ys;
current_z = current_z + cur_z_sp * (*soft_config).dt * zs;
current_a = current_a + cur_a_sp * (*soft_config).dt * as;
inc_x = inc_x + cur_x_sp * (*soft_config).dt;
inc_y = inc_y + cur_y_sp * (*soft_config).dt;
inc_z = inc_z + cur_z_sp * (*soft_config).dt;
inc_a = inc_a + cur_a_sp * (*soft_config).dt;
if (dist_x*xs <= current_x*xs) {
    inc_x = inc_x - (current_x*xs - dist_x*xs);
    cur_x_sp = 0;
    current_x = dist_x;
}
if (dist_y*ys <= current_y*ys) {
    inc_y = inc_y - (current_y*ys - dist_y*ys);
    cur_y_sp = 0;
    current_y = dist_y;
}
if (dist_z*zs <= current_z*zs) {
    inc_z = inc_z - (current_z*zs - dist_z*zs);
    cur_z_sp = 0;
    current_z = dist_z;
}
if (dist_a*as <= current_a*as) {
    inc_a = inc_a - (current_a*as - dist_a*as);
    cur_a_sp = 0;
    current_a = dist_a;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//check to see if time to take a step
if (inc_x > (*hard_config).x_per_step/2) {
    step_flag=ON;
    inc_x=inc_x-(*hard_config).x_per_step/2;
    if (act_step == HIGH) {
        if ((step & (*hard_config).x_sbit) == (*hard_config).x_sbit)
            step=step-(*hard_config).x_sbit;
        else
            if (x_step == LOW)
                step=step+(*hard_config).x_sbit;
    }
    else {
        if ((step & (*hard_config).x_sbit) == 0)
            step=step+(*hard_config).x_sbit;
        else
            if (x_step == HIGH)
                step=step-(*hard_config).x_sbit;
    }
    if (x_step == HIGH)
        x_step=LOW;
    else
        x_step=HIGH;
}
if (inc_y > (*hard_config).y_per_step/2) {
    step_flag=ON;
    inc_y=inc_y-(*hard_config).y_per_step/2;
    if (act_step == HIGH) {
        if ((step & (*hard_config).y_sbit) == (*hard_config).y_sbit)
            step=step-(*hard_config).y_sbit;
        else
            if (y_step == LOW)
                step=step+(*hard_config).y_sbit;
    }
    else {
        if ((step & (*hard_config).y_sbit) == 0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        step=step+(*hard_config).y_sbit;
    else
        if (y_step == HIGH)
            step=step-(*hard_config).y_sbit;
    }
    if (y_step == HIGH)
        y_step=LOW;
    else
        y_step=HIGH;
}
if (inc_z>(*hard_config).z_per_step/2) {
    step_flag=ON;
    inc_z=inc_z-(*hard_config).z_per_step/2;
    if (act_step == HIGH) {
        if ((step&(*hard_config).z_sbit) == (*hard_config).z_sbit)
            step=step-(*hard_config).z_sbit;
        else
            if (z_step == LOW)
                step=step+(*hard_config).z_sbit;
    }
    else {
        if ((step&(*hard_config).z_sbit) == 0)
            step=step+(*hard_config).z_sbit;
        else
            if (z_step == HIGH)
                step=step-(*hard_config).z_sbit;
    }
    if (z_step == HIGH)
        z_step=LOW;
    else
        z_step=HIGH;
}
if (inc_a>(*hard_config).a_per_step/2) {
    step_flag=ON;
    inc_a=inc_a-(*hard_config).a_per_step/2;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (act_step == HIGH) {
    if ((step & (*hard_config).a_sbit) == (*hard_config).a_sbit)
        step = step - (*hard_config).a_sbit;
    else
        if (a_step == LOW)
            step = step + (*hard_config).a_sbit;
}
else {
    if ((step & (*hard_config).a_sbit) == 0)
        step = step + (*hard_config).a_sbit;
    else
        if (a_step == HIGH)
            step = step - (*hard_config).a_sbit;
}
if (a_step == HIGH)
    a_step = LOW;
else
    a_step = HIGH;
}
if (step_flag) {
    step_flag = OFF;
    outportb((*hard_config).port, dir + step);
}
//check for completion of move
if (dist_x*xs <= current_x*xs && dist_y*ys <= current_y*ys &&
    dist_z*zs <= current_z*zs && dist_a*as <= current_a*as) {
    cur_x_sp = cur_y_sp = cur_z_sp = cur_a_sp = 0;
    if (*cur_speed <= min_speed && max_end_speed_sq <= min_speed*min_speed)
        *cur_speed = sqrt(max_end_speed_sq);
    break;
}
//check for keyboard speed control
if ((*soft_config).keyboard == ON && *sp_limit > 0) {
    if (key == 0x21) {
        key = 0;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    *sp_limit=*sp_limit+speed_inc;
    update_display(soft_config, 0, 0, 0, 0, 0, *sp_limit, 6); //'F' key
}
if (key == 0x1f) {
    key=0;
    if (*sp_limit-speed_inc>0)
        *sp_limit=*sp_limit-speed_inc;
    update_display(soft_config, 0, 0, 0, 0, 0, *sp_limit, 6); //'S' key
}
}
//check for machine pause
if (key == 0x19) {
    key=0;
    msg=PAUSE;
}
if (*sp_limit>0) {
    //check for io trip
    if ((*soft_config).io1_cmm == ON && (c_port_read&(*hard_config).io1_bit) == LOW) {
        msg=IO1;
        break;
    }
    if ((*soft_config).io2_cmm == ON && (c_port_read&(*hard_config).io2_bit) == LOW) {
        msg=IO2;
        break;
    }
    if ((*soft_config).io3_cmm == ON && (c_port_read&(*hard_config).io3_bit) == LOW) {
        msg=IO3;
        break;
    }
    if ((*soft_config).io4_cmm == ON && (c_port_read&(*hard_config).io4_bit) == LOW) {
        msg=IO4;
        break;
    }
}
//check for keyboard trip (space bar or esc key)
if ((*soft_config).keyboard == ON && (key == 0x39 || key == 0x01)) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    key=0;
    msg=BREAK;
    break;
}

//check for jog release
if ((*soft_config).keyboard == JOG && jogging == OFF)
    break;
}

if (*sp_limit<0 && break_flag == OFF) {
    //check for io trip
    if ((*soft_config).io1_cmm == ON && (c_port_read&(*hard_config).io1_bit) == LOW)
        msg=IO1;
    if ((*soft_config).io2_cmm == ON && (c_port_read&(*hard_config).io2_bit) == LOW)
        msg=IO2;
    if ((*soft_config).io3_cmm == ON && (c_port_read&(*hard_config).io3_bit) == LOW)
        msg=IO3;
    if ((*soft_config).io4_cmm == ON && (c_port_read&(*hard_config).io4_bit) == LOW)
        msg=IO4;

    //check for keyboard trip on space bar or ESC
    if (key == 0x39 || key == 0x01) {
        key=0;
        msg=BREAK;
    }

    if (msg == BREAK || msg == IO1 || msg == IO2 || msg == IO3 || msg == IO4) {
        break_flag=ON;
        if (msg != BREAK) {
            (*current).pt_x=(*current).x_pos+current_x;
            (*current).pt_y=(*current).y_pos+current_y;
            (*current).pt_z=(*current).z_pos+current_z;
            (*current).pt_a=(*current).a_pos+current_a;
        }

        dist_x=current_x+sign(dist_x)*min(fabs(dist_x-
current_x),cur_x_sp*cur_x_sp/(2*(*hard_config).accel_x));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    dist_y=current_y+sign(dist_y)*min(fabs(dist_y-
current_y),cur_y_sp*cur_y_sp/(2*(*hard_config).accel_y));

    dist_z=current_z+sign(dist_z)*min(fabs(dist_z-
current_z),cur_z_sp*cur_z_sp/(2*(*hard_config).accel_z));

    dist_a=current_a+sign(dist_a)*min(fabs(dist_a-
current_a),cur_a_sp*cur_a_sp/(2*(*hard_config).accel_a));

    }
}

if ((*soft_config).update_after == OFF) { //update screen every .016 seconds
    display_count=display_count+(*soft_config).dt;
    if (display_count>.015) {
        display_count=0;
        if (*sp_limit>0) {
            if (dist_x == 0 && dist_y == 0 && dist_z == 0)
                speed=0;
            else
                speed=*cur_speed;
            update_display(soft_config, (*current).x_pos+current_x, (*current).y_pos+current_y,
                (*current).z_pos+current_z, (*current).a_pos+current_a,
                speed, (dist_a*as**cur_speed/total_vect), 0, display);
        }
        else {
            update_display(soft_config, (*current).x_pos+current_x, (*current).y_pos+current_y,
                (*current).z_pos+current_z, (*current).a_pos+current_a,
                sqrt(cur_x_sp*cur_x_sp+cur_y_sp*cur_y_sp+cur_z_sp*cur_z_sp),
                cur_a_sp, 0, display);
        }
        display++;
        if (display == 6)
            display=0;
    }
}

if (msg == UNLIMIT)
    cur_x_sp=cur_y_sp=cur_z_sp=cur_a_sp=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if ((msg == COMPLETE || msg == PAUSE) && (*soft_config).keyboard != JOG) {
    (*current).x_pos=(*current).x_pos+dist_x;
    (*current).y_pos=(*current).y_pos+dist_y;
    (*current).z_pos=(*current).z_pos+dist_z;
    (*current).a_pos=(*current).a_pos+dist_a;
}
else {
    (*current).x_pos=(*current).x_pos+current_x;
    (*current).y_pos=(*current).y_pos+current_y;
    (*current).z_pos=(*current).z_pos+current_z;
    (*current).a_pos=(*current).a_pos+current_a;
}
return msg;
}

int sign(float num)
{
    if (num<0)
        return -1;
    else
        return 1;
}

double total_vect(double x, double y, double z, double a)
{
    if (x == 0 && y == 0 && z == 0)
        return fabs(a);
    return sqrt(x*x+y*y+z*z);
}

float get_max_accel(struct hard_setup *hard_config, float x, float y, float z, float a, float total)
{
    if (total == 0)
        return 0;
    return total/get_max_4(fabs(x)/(*hard_config).accel_x, fabs(y)/(*hard_config).accel_y,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        fabs(z)/(*hard_config).accel_z, fabs(a)/(*hard_config).accel_a);
    }

float get_max_speed(struct hard_setup *hard_config, float x, float y, float z, float a, float total)
{
    if (total == 0)
        return 0;
    return total/get_max_4(fabs(x)/(*hard_config).vel_x, fabs(y)/(*hard_config).vel_y,
        fabs(z)/(*hard_config).vel_z, fabs(a)/(*hard_config).vel_a);
}

float get_min_speed(struct hard_setup *hard_config, float x, float y, float z, float a, float total)
{
    if (total == 0)
        return 0;
    return total/get_max_4(fabs(x)/(*hard_config).min_vel_x, fabs(y)/(*hard_config).min_vel_y,
        fabs(z)/(*hard_config).min_vel_z, fabs(a)/(*hard_config).min_vel_a);
}

float get_max_4(float a, float b, float c, float d)
{
    if (a>=b&&a>=c&&a>=d)
        return a;
    if (b>=a&&b>=c&&b>=d)
        return b;
    if (c>=a&&c>=b&&c>=d)
        return c;
    return d;
}

int pause(struct hard_setup *hardware, struct soft_setup *software, float t, float *frate)
{
    float count=0;
    int msg=CONTINUE;
    BYTE s_port_read;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

init_key();
while (count<t) {
    while(tick) {
        if (key == 0x39 || key == 0x01) {
            key=0;
            msg=BREAK;
            break;
        }
        if (key == 0x21) {
            key=0;
            *frate=*frate+(*software).speed_inc;
            update_display(software, 0, 0, 0, 0, 0, *frate, 6);
        } // 'F' key
        if (key == 0x1f) {
            key=0;
            if (*frate-(*software).speed_inc>0)
                *frate=*frate-(*software).speed_inc;
            update_display(software, 0, 0, 0, 0, 0, *frate, 6);
        } // 'S' key
        if (key == 0x19) {
            key=0;
            msg=PAUSE;
        }
    }
    if (msg == BREAK)
        break;
    tick=1;
    count=count+(*software).dt;
    if ((*software).e_stop != OFF)
        s_port_read=(248&inportb((*hardware).port+1))^128;
    if ((*software).e_stop != OFF && (*hardware).e_trip>(*hardware).e_bit-
(s_port_read&(*hardware).e_bit) == 0) {
        msg=E_STOP;
        break;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}
restore_key();
return msg;
}

```

```

unsigned long guage(struct soft_setup *software)

```

```

{
    float count=0;
    unsigned long i=0;

    while (count<1.0) {
        while (tick){
            sqrt(3);
            i++;
        }
        tick=1;
        count=count+(*software).dt;
    }
    i=i*.0083;
    if (i>100000)
        i=100000;
    return i;
}

```

```

//software interrupt functions

```

```

volatile int tick_limit;      /* The number of ticks before we call DOS ISR */

```

```

volatile int tick_count;      /* Current DOS tick counter */

```

```

void set_timer_freq(unsigned long freq)

```

```

{
    unsigned int new_timer_number ;
    _asm cli      //disable interrupts

```

```

    if (freq<18)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    freq=18;
    tick_limit = freq/18; //number of ticks to give original
    tick_count = 0;

    /* Calculate the 'count down' value for the timer chip itself... */
    new_timer_number=(freq == 18) ? 0 : ((unsigned int) (1193180L/freq));
    /* Re-program the timer chip... */
    _asm mov dx, 43H
    _asm mov al, 34H
    _asm out dx, al
    _asm mov dx, 40H
    _asm mov ax, new_timer_number
    _asm out dx, al
    _asm mov al, ah
    _asm out dx, al

    _asm sti //enable interrupts
}

void (interrupt far *old_direct_timer)(void);

void _interrupt_far new_direct_timer(void)
{
    tick=0;
    if (--tick_count<=0) {
        tick_count=tick_limit;
        old_direct_timer();
    }
    else
        outp(0x20,0x20);
}

void begin_realtime(struct soft_setup *software, unsigned long freq)
{
    if (freq<18)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    freq=18;
    (*software).dt=1.0/freq;

    _asm cli //disable interrupts

    old_direct_timer = _dos_getvect(0x08);
    set_timer_freq(freq);
    _dos_setvect(0x08,new_direct_timer);

    _asm sti //enable interrupts
}

void end_realtime(void)
{
    _asm cli //disable interrupts

    set_timer_freq(18); /* Ensures correct rate! */
    _dos_setvect(0x08,old_direct_timer);

    _asm sti //enable interrupts
}

//keyboard interrupt functions
void (interrupt far *OldKeyboardRoutine)(void);

void _interrupt_far MyKeyboardRoutine(void)
{
    key = inportb(0x60);
    if (key == 140 || key == 141 || key == 206 || key == 202 || key == 200 ||
        key == 203 || key == 205 || key == 208 || key == 201 || key == 209)
        jogging=OFF;
    outportb(0x20, 0x20); /* Issue EOI */
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void init_key(void)
{
    _asm cli //disable interrupts

    OldKeyboardRoutine = _dos_getvect(9);
    _dos_setvect(0x09, MyKeyboardRoutine);
    key=0;

    _asm sti //enable interrupts
}

void restore_key(void)
{
    _asm cli //disable interrupts

    _dos_setvect(0x09, OldKeyboardRoutine);

    _asm sti //enable interrupts
}

//more high level user functions
int move(struct hard_setup *hard_config, struct soft_setup *soft_config, struct pos *current,
        float *cur_speed, float *sp_limit,
        double x2, double y2, double z2, double a2, double angle2, int object2,
        double x3, double y3, double z3, double a3, double angle3, int object3)
{
    double total_vect2, total_vect3, total_vect2_end, factor, total, temp;
    double radius, currentang, endang, incang, a_inc, z_inc, x_inc, y_inc;
    double x_start, y_start, z_start, a_start, x, y, z, a=0, slow_down_ang, i;
    double back_x_slow_down_ang, back_y_slow_down_ang, back_z_slow_down_ang, currentang_frac;
    float max_speed_2, max_accel, max_speed, high_speed=HIGH_SPEED;
    int msg, temp1, temp2, temp3, temp4, temp5;
    int acceleration=NORMAL, calcang=ON, break_flag=OFF;
    static int fd_fwd_msg=START_OVER, fd_fwd_msg_a=START_OVER;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

struct unit_vect vect2, vect3, vect2_end;

static struct unit_vect vect1;

// static int demo_inc;

// if (version == DEMO)
// {demo_inc++;
// if (demo_inc>40) return QUIT;
// }

if (*sp_limit<0 && object2 != LINE_ENT && object2 != LINE_G0_ENT)
    return INVALID_PARAMETER;

//hook keyboard interrupt
init_key();

switch (object2) {
case LINE_ENT:
    //get the max speed if HIGH_SPEED is selected
    if (*sp_limit<0)
        *sp_limit=get_max_speed(hard_config,x2,y2,z2,a2,total_vect(x2,y2,z2,a2));

    //get unit vector and total length for current object
    total_vect2=get_unit_vect(&vect2, x2, y2, z2, a2, angle2, object2);

    //do backlash
    msg=backlash(hard_config, soft_config, &vect1, &vect2);
    if (msg != COMPLETE)
        break;

    //save current backlash status
    if (vect2.x != 0)
        vect1.x=vect2.x;
    if (vect2.y != 0)
        vect1.y=vect2.y;
    if (vect2.z != 0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    vect1.z=vect2.z;
if (vect2.a != 0)
    vect1.a=vect2.a;

//get unit vector and total length for next object
total_vect3=get_unit_vect(&vect3, x3, y3, z3, a3, angle3, object3);

//see if backlash at end of move
if (backlash_to_come(hard_config, &vect2, &vect3, fd_fwd_msg, fd_fwd_msg_a) || object3 ==
LINE_G0_ENT)
    max_speed_2=0;
else { //get maximum ending speed squared
    if (object3 == LINE_ENT)
        max_speed_2=min(max_speed_sq(hard_config, x2, y2, z2, a2, total_vect2, &vect3),
            max_speed_sq_line(hard_config, x3, y3, z3, a3, total_vect3));
    else
        max_speed_2=max_speed_sq(hard_config, x2, y2, z2, a2, total_vect2, &vect3);
}

//move machine accordingly
msg=move_machine(hard_config, soft_config, current, cur_speed, sp_limit, max_speed_2,
    x2, y2, z2, a2, total_vect2);

//continue to stop if break or io trip
if (msg == BREAK || msg == IO1 || msg == IO2 || msg == IO3 || msg == IO4) {
    if (msg != BREAK) {
        (*current).pt_x=(*current).x_pos;
        (*current).pt_y=(*current).y_pos;
        (*current).pt_z=(*current).z_pos;
        (*current).pt_a=(*current).a_pos;
    }
    temp1=(*soft_config).keyboard;
    temp2=(*soft_config).io1_cmm;
    temp3=(*soft_config).io2_cmm;
    temp4=(*soft_config).io3_cmm;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

temp5=(*soft_config).io4_cmm;

(*soft_config).keyboard=(*soft_config).io1_cmm=(*soft_config).io2_cmm=
(*soft_config).io3_cmm=(*soft_config).io4_cmm=OFF;

max_accel=get_max_accel(hard_config, x2, y2, z2, a2, total_vect2);

total=*cur_speed**cur_speed/(2*max_accel);

factor=total/total_vect2;

move_machine(hard_config, soft_config, current, cur_speed, sp_limit, 0,
             x2*factor, y2*factor, z2*factor, a2*factor, total);

(*soft_config).keyboard=temp1;
(*soft_config).io1_cmm=temp2;
(*soft_config).io2_cmm=temp3;
(*soft_config).io3_cmm=temp4;
(*soft_config).io4_cmm=temp5;
}
for (temp1=0; temp1<4; temp1++) {
    if (temp1 == 3)
        temp1++;
    update_display(soft_config, (*current).x_pos, (*current).y_pos,
                  (*current).z_pos, (*current).a_pos, 0, 0, 0, temp1);
}
break;

case LINE_G0_ENT:
    //get backlash vector for current object
    vect2.x=x2;
    vect2.y=y2;
    vect2.z=z2;
    vect2.a=a2;

    //do backlash
    msg=backlash(hard_config, soft_config, &vect1, &vect2);
    if (msg != COMPLETE)
        break;

    //save current backlash status

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (vect2.x != 0)
    vect1.x=vect2.x;
if (vect2.y != 0)
    vect1.y=vect2.y;
if (vect2.z != 0)
    vect1.z=vect2.z;
if (vect2.a != 0)
    vect1.a=vect2.a;

//get unit vector and total length for next object
total_vect3=get_unit_vect(&vect3, x3, y3, z3, a3, angle3, object3);

//set backlash to come vector
backlash_to_come(hard_config, &vect2, &vect3, fd_fwd_msg, fd_fwd_msg_a);

//move machine accordingly
msg=move_machine(hard_config, soft_config, current, cur_speed, &high_speed, 0,
    x2, y2, z2, a2, total_vect2);
for (temp1=0;temp1<4;temp1++) {
    if (temp1 == 3)
        temp1++;
    update_display(soft_config, (*current).x_pos, (*current).y_pos,
        (*current).z_pos, (*current).a_pos, 0, 0, 0, temp1);
}
break;

case XY_ARC_ENT:
    //get unit vector and total length end of arc and next object
    radius=sqrt(x2*x2+y2*y2);
    if (radius == 0)
        currentang=endang=angle2=0;
    else {
        currentang=atan2(-y2,-x2);
        endang=currentang+angle2;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

total_vect2_end=get_unit_vect(&vect2_end, -radius*cos(endang), -radius*sin(endang), z2, a2,
angle2, XY_ARC_ENT);

total_vect3=get_unit_vect(&vect3, x3, y3, z3, a3, angle3, object3);

//see if backlash at end of move

if (backlash_to_come(hard_config, &vect2_end, &vect3, START_OVER, START_OVER) || object3
==LINE_G0_ENT)

    max_speed_2=0;

else { //get maximum ending speed squared

    if (object3 == LINE_ENT)

        max_speed_2=min(max_speed_sq(hard_config, vect2_end.x, vect2_end.y,
vect2_end.z, vect2_end.a, total_vect2_end, &vect3),
max_speed_sq_line(hard_config, x3, y3, z3, a3, total_vect3));

    else

        max_speed_2=max_speed_sq(hard_config, vect2_end.x, vect2_end.y, vect2_end.z,
vect2_end.a, total_vect2_end, &vect3);

}

//max acceleration due to end vector

max_accel=get_max_accel(hard_config, vect2_end.x, vect2_end.y, vect2_end.z, vect2_end.a,
total_vect2_end);

//get loop increments

if (angle2 != 0) {

    incang=2*atan(10*min((*hard_config).x_per_step, (*hard_config).y_per_step)/(2*radius));

    if (angle2<0)

        incang=-incang;

    a_inc=a2*incang/angle2;

    z_inc=z2*incang/angle2;

}

else

    incang=1;

//initialize ang begin angle loop

x_start=(*current).x_pos;

y_start=(*current).y_pos;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

z_start=(*current).z_pos;
a_start=(*current).a_pos;
for (i=0;i<labs(angle2/incang);i++) {
    currenttang=currenttang+incang;
    x=cos(currenttang)*radius+x2-((*current).x_pos-x_start);
    y=sin(currenttang)*radius+y2-((*current).y_pos-y_start);
    z=z_inc*(i+1)-((*current).z_pos-z_start);
    if (a2 != 0)
        a=a_inc*(i+1)-((*current).a_pos-a_start);
    total_vect2=get_unit_vect(&vect2, x, y, z, a, NULL, LINE_ENT);

    //do backlash
    msg=backlash(hard_config, soft_config, &vect1, &vect2);
    if (msg != COMPLETE)
        goto DONE;

    //save backlash settings
    if (vect2.x != 0)
        vect1.x=vect2.x;
    if (vect2.y != 0)
        vect1.y=vect2.y;
    if (vect2.z != 0)
        vect1.z=vect2.z;
    if (vect2.a != 0)
        vect1.a=vect2.a;

    if (break_flag != OFF)
        acceleration=DECEL;
    else
        acceleration=NORMAL;

    temp=*cur_speed**cur_speed*fabs(incang);
    //determine if backlash slow down angles have been exceeded
    if ((*hard_config).back_x>.000009) {
        currenttang_frac=fmod(currenttang,PI);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (currentang_frac<0)
    currentang_frac=currentang_frac+PI;
if (angle2>0)
    currentang_frac=PI-currentang_frac;
if (calcang == ON)
    back_x_slow_down_ang=temp/(2*(*hard_config).accel_y*total_vect2);
if (back_x_slow_down_ang>=currentang_frac)
    acceleration=DECEL;
}

if ((*hard_config).back_y>.000009) {
    currentang_frac=fmod(currentang+PI/2,PI);
    if (currentang_frac<0)
        currentang_frac=currentang_frac+PI;
    if (angle2>0)
        currentang_frac=PI-currentang_frac;
    if (calcang == ON)
        back_y_slow_down_ang=temp/(2*(*hard_config).accel_x*total_vect2);
    if (back_y_slow_down_ang>=currentang_frac)
        acceleration=DECEL;
}

//determine if ending slow down angle has been exceeded
if (calcang == ON)
    slow_down_ang=(max_speed_2-
*cur_speed*cur_speed)*fabs(incang)/(2*max_accel*total_vect2);
    if (slow_down_ang>=fabs(angle2-incang*(i+1)))
        acceleration=DECEL;

if (acceleration == NORMAL) {
    max_speed=get_max_speed(hard_config, x, y, z, a, total_vect2);
    max_speed=max_speed*max_speed;
    msg=move_machine(hard_config, soft_config, current, cur_speed, sp_limit, max_speed, x, y, z,
a, total_vect2);
    calcang=ON;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (acceleration == DECEL) {
    msg=move_machine(hard_config, soft_config, current, cur_speed, sp_limit, 0, x, y, z, a,
total_vect2);
    calcang=OFF;
}
if (msg != COMPLETE) {
    if ((msg == BREAK || msg == IO1 || msg == IO2 || msg == IO3 || msg == IO4) && break_flag ==
OFF) {
        break_flag=msg;
        if (msg != BREAK) {
            (*current).pt_x=(*current).x_pos;
            (*current).pt_y=(*current).y_pos;
            (*current).pt_z=(*current).z_pos;
            (*current).pt_a=(*current).a_pos;
        }
        temp1=(*soft_config).keyboard;
        temp2=(*soft_config).io1_cmm;
        temp3=(*soft_config).io2_cmm;
        temp4=(*soft_config).io3_cmm;
        temp5=(*soft_config).io4_cmm;
        (*soft_config).keyboard=(*soft_config).io1_cmm=(*soft_config).io2_cmm=
(*soft_config).io3_cmm=(*soft_config).io4_cmm=OFF;
        //reset angle for stop
        max_accel=get_max_accel(hard_config, x, y, z, a, total_vect2);
        slow_down_ang=*cur_speed**cur_speed*fabs(incang)/(2*max_accel*total_vect2);
        z2=z_inc*(i+1)+z2*slow_down_ang/fabs(angle2);
        a2=a_inc*(i+1)+a2*slow_down_ang/fabs(angle2);
        if (angle2<0) {
            angle2=incang*(i+1)-slow_down_ang;
            endang=currentang-slow_down_ang;
        }
        else {
            angle2=incang*(i+1)+slow_down_ang;
            endang=currentang+slow_down_ang;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    else
        goto DONE;
    }
}

x=cos(endang)*radius+x2-((*current).x_pos-x_start);
y=sin(endang)*radius+y2-((*current).y_pos-y_start);
z=z2-((*current).z_pos-z_start);
if (a2 != 0)
    a=a2-((*current).a_pos-a_start);
total_vect2=get_unit_vect(&vect2, x, y, z, a, NULL, LINE_ENT);

//do backlash
msg=backlash(hard_config, soft_config, &vect1, &vect2);
if (msg != COMPLETE)
    goto DONE;

//save backlash settings
if (vect2.x != 0)
    vect1.x=vect2.x;
if (vect2.y != 0)
    vect1.y=vect2.y;
if (vect2.z != 0)
    vect1.z=vect2.z;
if (vect2.a != 0)
    vect1.a=vect2.a;

msg=move_machine(hard_config, soft_config, current, cur_speed, sp_limit, max_speed_2, x, y, z, a,
total_vect2);
if ((msg == BREAK || msg == IO1 || msg == IO2 || msg == IO3 || msg == IO4) && break_flag ==
OFF) {
    break_flag=msg;
    if (msg != BREAK) {
        (*current).pt_x=(*current).x_pos;
        (*current).pt_y=(*current).y_pos;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    (*current).pt_z=(*current).z_pos;
    (*current).pt_a=(*current).a_pos;
}

temp1=(*soft_config).keyboard;
temp2=(*soft_config).io1_cmm;
temp3=(*soft_config).io2_cmm;
temp4=(*soft_config).io3_cmm;
temp5=(*soft_config).io4_cmm;
(*soft_config).keyboard=(*soft_config).io1_cmm=(*soft_config).io2_cmm=
(*soft_config).io3_cmm=(*soft_config).io4_cmm=OFF;
max_accel=get_max_accel(hard_config, vect2.x, vect2.y, vect2.z, vect2.a, total_vect2);
total=*cur_speed**cur_speed/(2*max_accel);
factor=total/total_vect2;
move_machine(hard_config, soft_config, current, cur_speed, sp_limit, 0,
             x2*factor, y2*factor, z2*factor, a2*factor, total);
}

DONE :
if (break_flag != OFF) {
    (*soft_config).keyboard=temp1;
    (*soft_config).io1_cmm=temp2;
    (*soft_config).io2_cmm=temp3;
    (*soft_config).io3_cmm=temp4;
    (*soft_config).io4_cmm=temp5;
    msg=break_flag;
}

for(temp1=0;temp1<4;temp1++) {
    if (temp1 == 3)
        temp1++;
    update_display(soft_config, (*current).x_pos, (*current).y_pos,
                  (*current).z_pos, (*current).a_pos, 0, 0, 0, temp1);
}

break;

```

case YZ_ARC_ENT:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//get unit vector and total length end of arc and next object
radius=sqrt(y2*y2+z2*z2);
if (radius == 0)
    currentang=endang=angle2=0;
else {
    currentang=atan2(-z2,-y2);
    endang=currentang+angle2;
}
total_vect2_end=get_unit_vect(&vect2_end, x2, -radius*cos(endang), -radius*sin(endang), a2,
angle2, YZ_ARC_ENT);
total_vect3=get_unit_vect(&vect3, x3, y3, z3, a3, angle3, object3);

//see if backlash at end of move
if (backlash_to_come(hard_config, &vect2_end, &vect3, START_OVER, START_OVER) || object3
==LINE_G0_ENT)
    max_speed_2=0;
else { //get maximum ending speed squared
    if (object3 == LINE_ENT)
        max_speed_2=min(max_speed_sq(hard_config, vect2_end.x, vect2_end.y,
        vect2_end.z, vect2_end.a, total_vect2_end, &vect3),
        max_speed_sq_line(hard_config, x3, y3, z3, a3, total_vect3));
    else
        max_speed_2=max_speed_sq(hard_config, vect2_end.x, vect2_end.y, vect2_end.z,
        vect2_end.a, total_vect2_end, &vect3);
}

//max acceleration due to end vector
max_accel=get_max_accel(hard_config, vect2_end.x, vect2_end.y, vect2_end.z, vect2_end.a,
total_vect2_end);

//get loop increments
if (angle2 != 0) {
    incang=2*atan(10*min((*hard_config).y_per_step, (*hard_config).z_per_step)/(2*radius));
    if (angle2<0)
        incang=-incang;
    a_inc=a2*incang/angle2;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    x_inc=x2*incang/angle2;
}

else

    incang=1;

//initialize ang begin angle loop
x_start=(*current).x_pos;
y_start=(*current).y_pos;
z_start=(*current).z_pos;
a_start=(*current).a_pos;
for (i=0;i<labs(angle2/incang);i++) {
    currentang=currentang+incang;
    y=cos(currentang)*radius+y2-((*current).y_pos-y_start);
    z=sin(currentang)*radius+z2-((*current).z_pos-z_start);
    x=x_inc*(i+1)-((*current).x_pos-x_start);
    if (a2 != 0)
        a=a_inc*(i+1)-((*current).a_pos-a_start);
    total_vect2=get_unit_vect(&vect2, x, y, z, a, NULL, LINE_ENT);

    //do backlash
    msg=backlash(hard_config, soft_config, &vect1, &vect2);
    if (msg != COMPLETE)
        goto DONE;

    //save backlash settings
    if (vect2.x != 0)
        vect1.x=vect2.x;
    if (vect2.y != 0)
        vect1.y=vect2.y;
    if (vect2.z != 0)
        vect1.z=vect2.z;
    if (vect2.a != 0)
        vect1.a=vect2.a;

    if (break_flag != OFF)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    acceleration=DECEL;
else
    acceleration=NORMAL;

temp=*cur_speed**cur_speed*fabs(incang);
//determine if backlash slow down angles have been exceeded
if ((*hard_config).back_y>.000009) {
    currentang_frac=fmod(currentang,PI);
    if (currentang_frac<0)
        currentang_frac=currentang_frac+PI;
    if (angle2>0)
        currentang_frac=PI-currentang_frac;
    if (calcang == ON)
        back_y_slow_down_ang=temp/(2*(*hard_config).accel_z*total_vect2);
    if (back_y_slow_down_ang>=currentang_frac)
        acceleration=DECEL;
}
if ((*hard_config).back_z>.000009) {
    currentang_frac=fmod(currentang+PI/2,PI);
    if (currentang_frac<0)
        currentang_frac=currentang_frac+PI;
    if (angle2>0)
        currentang_frac=PI-currentang_frac;
    if (calcang == ON)
        back_z_slow_down_ang=temp/(2*(*hard_config).accel_y*total_vect2);
    if (back_z_slow_down_ang>=currentang_frac)
        acceleration=DECEL;
}

```

```

//determine if ending slow down angle has been exceeded

```

```

if (calcang == ON)

```

```

    slow_down_ang=(max_speed_2-
*cur_speed**cur_speed)*fabs(incang)/(2*max_accel*total_vect2);

```

```

    if (slow_down_ang>=fabs(angle2-incang*(i+1)))

```

```

        acceleration=DECEL;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (acceleration == NORMAL) {
    max_speed=get_max_speed(hard_config, x, y, z, a, total_vect2);
    max_speed=max_speed*max_speed;
    msg=move_machine(hard_config, soft_config, current, cur_speed, sp_limit, max_speed, x, y, z,
a, total_vect2);
    calcang=ON;
}

if (acceleration == DECEL) {
    msg=move_machine(hard_config, soft_config, current, cur_speed, sp_limit, 0, x, y, z, a,
total_vect2);
    calcang=OFF;
}

if (msg != COMPLETE) {
    if ((msg == BREAK || msg == IO1 || msg == IO2 || msg == IO3 || msg == IO4) && break_flag ==
OFF) {
        break_flag=msg;
        if (msg != BREAK) {
            (*current).pt_x=(*current).x_pos;
            (*current).pt_y=(*current).y_pos;
            (*current).pt_z=(*current).z_pos;
            (*current).pt_a=(*current).a_pos;
        }
        temp1=(*soft_config).keyboard;
        temp2=(*soft_config).io1_cmm;
        temp3=(*soft_config).io2_cmm;
        temp4=(*soft_config).io3_cmm;
        temp5=(*soft_config).io4_cmm;
        (*soft_config).keyboard=(*soft_config).io1_cmm=(*soft_config).io2_cmm=
(*soft_config).io3_cmm=(*soft_config).io4_cmm=OFF;
        //reset angle for stop
        max_accel=get_max_accel(hard_config, x, y, z, a, total_vect2);
        slow_down_ang=*cur_speed*cur_speed*fabs(incang)/(2*max_accel*total_vect2);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

x2=x_inc*(i+1)+x2*slow_down_ang/fabs(angle2);
a2=a_inc*(i+1)+a2*slow_down_ang/fabs(angle2);
if (angle2<0) {
    angle2=incang*(i+1)-slow_down_ang;
    endang=currentang-slow_down_ang;
}
else {
    angle2=incang*(i+1)+slow_down_ang;
    endang=currentang+slow_down_ang;
}
}
else
    goto DONE2;
}
}
y=cos(endang)*radius+y2-((*current).y_pos-y_start);
z=sin(endang)*radius+z2-((*current).z_pos-z_start);
x=x2-((*current).x_pos-x_start);
if (a2 != 0)
    a=a2-((*current).a_pos-a_start);
total_vect2=get_unit_vect(&vect2, x, y, z, a, NULL, LINE_ENT);

//do backlash
msg=backlash(hard_config, soft_config, &vect1, &vect2);
if (msg != COMPLETE)
    goto DONE2;

//save backlash settings
if (vect2.x != 0)
    vect1.x=vect2.x;
if (vect2.y != 0)
    vect1.y=vect2.y;
if (vect2.z != 0)
    vect1.z=vect2.z;
if (vect2.a != 0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

vect1.a=vect2.a;

msg=move_machine(hard_config, soft_config, current, cur_speed, sp_limit, max_speed_2, x, y, z, a,
total_vect2);

if ((msg == BREAK || msg == IO1 || msg == IO2 || msg == IO3 || msg == IO4) && break_flag ==
OFF) {
    break_flag=msg;
    if (msg != BREAK) {
        (*current).pt_x=(*current).x_pos;
        (*current).pt_y=(*current).y_pos;
        (*current).pt_z=(*current).z_pos;
        (*current).pt_a=(*current).a_pos;
    }
    temp1=(*soft_config).keyboard;
    temp2=(*soft_config).io1_cmm;
    temp3=(*soft_config).io2_cmm;
    temp4=(*soft_config).io3_cmm;
    temp5=(*soft_config).io4_cmm;
    (*soft_config).keyboard=(*soft_config).io1_cmm=(*soft_config).io2_cmm=
    (*soft_config).io3_cmm=(*soft_config).io4_cmm=OFF;
    max_accel=get_max_accel(hard_config,vect2.x, vect2.y, vect2.z, vect2.a, total_vect2);
    total=cur_speed*cur_speed/(2*max_accel);
    factor=total/total_vect2;
    move_machine(hard_config, soft_config, current, cur_speed, sp_limit, 0,
        x2*factor, y2*factor, z2*factor, a2*factor, total);
}

```

DONE2 :

```

if (break_flag != OFF) {
    (*soft_config).keyboard=temp1;
    (*soft_config).io1_cmm=temp2;
    (*soft_config).io2_cmm=temp3;
    (*soft_config).io3_cmm=temp4;
    (*soft_config).io4_cmm=temp5;
    msg=break_flag;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
for (temp1=0;temp1<4;temp1++) {
    if (temp1 == 3)
        temp1++;
    update_display(soft_config, (*current).x_pos, (*current).y_pos,
        (*current).z_pos, (*current).a_pos, 0, 0, 0, temp1);
}
break;

case ZX_ARC_ENT:
    //get unit vector and total length end of arc and next object
    radius=sqrt(z2*z2+x2*x2);
    if (radius == 0)
        currentang=endang=angle2=0;
    else {
        currentang=atan2(-x2,-z2);
        endang=currentang+angle2;
    }
    total_vect2_end=get_unit_vect(&vect2_end, -radius*sin(endang), y2, -radius*cos(endang), a2,
    angle2, ZX_ARC_ENT);
    total_vect3=get_unit_vect(&vect3, x3, y3, z3, a3, angle3, object3);

    //see if backlash at end of move
    if (backlash_to_come(hard_config, &vect2_end, &vect3, START_OVER, START_OVER) || object3
    == LINE_G0_ENT)
        max_speed_2=0;
    else { //get maximum ending speed squared
        if (object3 == LINE_ENT)
            max_speed_2=min(max_speed_sq(hard_config, vect2_end.x, vect2_end.y,
                vect2_end.z, vect2_end.a, total_vect2_end, &vect3),
                max_speed_sq_line(hard_config, x3, y3, z3, a3, total_vect3));
        else
            max_speed_2=max_speed_sq(hard_config, vect2_end.x, vect2_end.y, vect2_end.z,
                vect2_end.a, total_vect2_end, &vect3);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//max acceleration due to end vector

max_accel=get_max_accel(hard_config, vect2_end.x, vect2_end.y, vect2_end.z, vect2_end.a,
total_vect2_end);

//get loop increments
if (angle2 != 0) {
    incang=2*atan(10*min((*hard_config).z_per_step, (*hard_config).x_per_step)/(2*radius));
    if (angle2<0)
        incang=-incang;
    a_inc=a2*incang/angle2;
    y_inc=y2*incang/angle2;
}
else
    incang=1;

//initialize ang begin angle loop
x_start=(*current).x_pos;
y_start=(*current).y_pos;
z_start=(*current).z_pos;
a_start=(*current).a_pos;
for (i=0;i<labs(angle2/incang);i++) {
    currentang=currentang+incang;
    z=cos(currentang)*radius+z2-((*current).z_pos-z_start);
    x=sin(currentang)*radius+x2-((*current).x_pos-x_start);
    y=y_inc*(i+1)-((*current).y_pos-y_start);
    if (a2 != 0)
        a=a_inc*(i+1)-((*current).a_pos-a_start);
    total_vect2=get_unit_vect(&vect2, x, y, z, a, NULL, LINE_ENT);

//do backlash
msg=backlash(hard_config, soft_config, &vect1, &vect2);
if (msg != COMPLETE)
    goto DONE;

//save backlash settings

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (vect2.x != 0)
    vect1.x=vect2.x;
if (vect2.y != 0)
    vect1.y=vect2.y;
if (vect2.z != 0)
    vect1.z=vect2.z;
if (vect2.a != 0)
    vect1.a=vect2.a;

if (break_flag != OFF)
    acceleration=DECEL;
else
    acceleration=NORMAL;

temp=*cur_speed**cur_speed*fabs(incang);
//determine if backlash slow down angles have been exceeded
if ((*hard_config).back_z>.000009) {
    currentang_frac=fmod(currentang,PI);
    if (currentang_frac<0)
        currentang_frac=currentang_frac+PI;
    if (angle2>0)
        currentang_frac=PI-currentang_frac;
    if (calcang == ON)
        back_z_slow_down_ang=temp/(2*(*hard_config).accel_x*total_vect2);
    if (back_z_slow_down_ang>=currentang_frac)
        acceleration=DECEL;
}

if ((*hard_config).back_x>.000009) {
    currentang_frac=fmod(currentang+PI/2,PI);
    if (currentang_frac<0)
        currentang_frac=currentang_frac+PI;
    if (angle2>0)
        currentang_frac=PI-currentang_frac;
    if (calcang == ON)
        back_x_slow_down_ang=temp/(2*(*hard_config).accel_z*total_vect2);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if (back_x_slow_down_ang >= currentang_frac)
        acceleration = DECEL;
}

//determine if ending slow down angle has been exceeded
if (calcang == ON)
    slow_down_ang = (max_speed_2 -
*cur_speed**cur_speed)*fabs(incang)/(2*max_accel*total_vect2);
    if (slow_down_ang >= fabs(angle2-incang*(i+1)))
        acceleration = DECEL;

if (acceleration == NORMAL) {
    max_speed = get_max_speed(hard_config, x, y, z, a, total_vect2);
    max_speed = max_speed*max_speed;
    msg = move_machine(hard_config, soft_config, current, cur_speed, sp_limit, max_speed, x, y, z,
a, total_vect2);
    calcang = ON;
}
if (acceleration == DECEL) {
    msg = move_machine(hard_config, soft_config, current, cur_speed, sp_limit, 0, x, y, z, a,
total_vect2);
    calcang = OFF;
}
if (msg != COMPLETE) {
    if ((msg == BREAK || msg == IO1 || msg == IO2 || msg == IO3 || msg == IO4) && break_flag ==
OFF) {
        break_flag = msg;
        if (msg != BREAK) {
            (*current).pt_x = (*current).x_pos;
            (*current).pt_y = (*current).y_pos;
            (*current).pt_z = (*current).z_pos;
            (*current).pt_a = (*current).a_pos;
        }
        temp1 = (*soft_config).keyboard;
        temp2 = (*soft_config).io1_cmm;
        temp3 = (*soft_config).io2_cmm;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

temp4=(*soft_config).io3_cmm;
temp5=(*soft_config).io4_cmm;
(*soft_config).keyboard=(*soft_config).io1_cmm=(*soft_config).io2_cmm=
(*soft_config).io3_cmm=(*soft_config).io4_cmm=OFF;

//reset angle for stop
max_accel=get_max_accel(hard_config, vect2.x, vect2.y, vect2.z, vect2.a, total_vect2);
slow_down_ang=*cur_speed**cur_speed*fabs(incang)/(2*max_accel*total_vect2);
y2=y_inc*(i+1)+y2*slow_down_ang/fabs(angle2);
a2=a_inc*(i+1)+a2*slow_down_ang/fabs(angle2);
if (angle2<0) {
    angle2=incang*(i+1)-slow_down_ang;
    endang=currentang-slow_down_ang;
}
else {
    angle2=incang*(i+1)+slow_down_ang;
    endang=currentang+slow_down_ang;
}
}
else
    goto DONE3;
}
}
z=cos(endang)*radius+z2-((*current).z_pos-z_start);
x=sin(endang)*radius+x2-((*current).x_pos-x_start);
y=y2-((*current).y_pos-y_start);
if (a2 != 0)
    a=a2-((*current).a_pos-a_start);
total_vect2=get_unit_vect(&vect2, x, y, z, a, NULL, LINE_ENT);

//do backlash
msg=backlash(hard_config, soft_config, &vect1, &vect2);
if (msg != COMPLETE)
    goto DONE3;

//save backlash settings

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (vect2.x != 0)
    vect1.x=vect2.x;
if (vect2.y != 0)
    vect1.y=vect2.y;
if (vect2.z != 0)
    vect1.z=vect2.z;
if (vect2.a != 0)
    vect1.a=vect2.a;

msg=move_machine(hard_config, soft_config, current, cur_speed, sp_limit, max_speed_2, x, y, z, a,
total_vect2);
if ((msg == BREAK || msg == IO1 || msg == IO2 || msg == IO3 || msg == IO4) && break_flag ==
OFF) {
    break_flag=msg;
    if (msg != BREAK) {
        (*current).pt_x=(*current).x_pos;
        (*current).pt_y=(*current).y_pos;
        (*current).pt_z=(*current).z_pos;
        (*current).pt_a=(*current).a_pos;
    }
    temp1=(*soft_config).keyboard;
    temp2=(*soft_config).io1_cmm;
    temp3=(*soft_config).io2_cmm;
    temp4=(*soft_config).io3_cmm;
    temp5=(*soft_config).io4_cmm;
    (*soft_config).keyboard=(*soft_config).io1_cmm=(*soft_config).io2_cmm=
    (*soft_config).io3_cmm=(*soft_config).io4_cmm=OFF;
    max_accel=get_max_accel(hard_config, x, y, z, a, total_vect2);
    total=cur_speed**cur_speed/(2*max_accel);
    factor=total/total_vect2;
    msg=move_machine(hard_config, soft_config, current, cur_speed, sp_limit, 0,
        x2*factor, y2*factor, z2*factor, a2*factor, total);
}

```

DONE3 :

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (break_flag != OFF) {
    (*soft_config).keyboard=temp1;
    (*soft_config).io1_cmm=temp2;
    (*soft_config).io2_cmm=temp3;
    (*soft_config).io3_cmm=temp4;
    (*soft_config).io4_cmm=temp5;
    msg=break_flag;
}

for (temp1=0;temp1<4;temp1++) {
    if (temp1 == 3)
        temp1++;
    update_display(soft_config, (*current).x_pos, (*current).y_pos,
        (*current).z_pos, (*current).a_pos, 0, 0, 0, temp1);
}
break;
}

//reset backlash to come messages accordingly
if ((x3 == 0 && y3 == 0 && z3 == 0) || msg != COMPLETE || msg != PAUSE)
    fd_fwd_msg=START_OVER;
else
    fd_fwd_msg=CONTINUE;
if (a3 == 0 || msg != COMPLETE || msg != PAUSE)
    fd_fwd_msg_a=START_OVER;
else
    fd_fwd_msg_a=CONTINUE;

//restore keyboard interrupt
restore_key0;
return msg;
}

```

//backlash function

```

int backlash(struct hard_setup *hard_config, struct soft_setup *soft_config,
    struct unit_vect *vect1, struct unit_vect *vect2)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    double x=0, y=0, z=0, a=0, total;
    float max_speed, cur_speed=0;
    int msg, temp1, temp2, temp3, temp4, temp5, temp6, temp7, temp8, temp9, temp10, temp11, temp12;
    struct pos pt;
    pt.x_pos=pt.y_pos=pt.z_pos=pt.a_pos=0;

    //determine if backlash takes place
    if (((*vect1).x*(*vect2).x>=0 || (*hard_config).back_x<=.000009) &&
        ((*vect1).y*(*vect2).y>=0 || (*hard_config).back_y<=.000009) &&
        ((*vect1).z*(*vect2).z>=0 || (*hard_config).back_z<=.000009) &&
        ((*vect1).a*(*vect2).a>=0 || (*hard_config).back_a<=.000009))
        return COMPLETE;

    //determine backlash
    if ((*vect1).x*(*vect2).x<0 && (*hard_config).back_x>.000009) {
        if ((*vect2).x<0)
            x=(*hard_config).back_x;
        else
            x=(*hard_config).back_x;
    }
    if ((*vect1).y*(*vect2).y<0 && (*hard_config).back_y>.000009) {
        if ((*vect2).y<0)
            y=(*hard_config).back_y;
        else
            y=(*hard_config).back_y;
    }
    if ((*vect1).z*(*vect2).z<0 && (*hard_config).back_z>.000009) {
        if ((*vect2).z<0)
            z=(*hard_config).back_z;
        else
            z=(*hard_config).back_z;
    }
    if ((*vect1).a*(*vect2).a<0 && (*hard_config).back_a>.000009) {
        if ((*vect2).a<0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    a=(*hard_config).back_a;
else
    a=(*hard_config).back_a;
}

//determine total backlash vector
total=total_vect(x, y, z, a);

//determine max speed
max_speed=get_max_speed(hard_config, x, y, z, a, total);

//do backlash compensation
temp1=(*soft_config).keyboard;
temp2=(*soft_config).io1_cmm;
temp3=(*soft_config).io2_cmm;
temp4=(*soft_config).io3_cmm;
temp5=(*soft_config).io4_cmm;
temp6=(*soft_config).display_x;
temp7=(*soft_config).display_y;
temp8=(*soft_config).display_z;
temp9=(*soft_config).display_s;
temp10=(*soft_config).display_as;
temp11=(*soft_config).display_a;
temp12=(*soft_config).display_fr;
(*soft_config).keyboard=(*soft_config).io1_cmm=(*soft_config).io2_cmm=
(*soft_config).io3_cmm=(*soft_config).io4_cmm=
(*soft_config).display_x=(*soft_config).display_y=(*soft_config).display_z=(*soft_config).display_a=
(*soft_config).display_s=(*soft_config).display_as=(*soft_config).display_fr=OFF;
msg=move_machine(hard_config, soft_config, &pt, &cur_speed, &max_speed, 0, x, y, z, a, total);
(*soft_config).keyboard=temp1;
(*soft_config).io1_cmm=temp2;
(*soft_config).io2_cmm=temp3;
(*soft_config).io3_cmm=temp4;
(*soft_config).io4_cmm=temp5;
(*soft_config).display_x=temp6;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(*soft_config).display_y=temp7;
(*soft_config).display_z=temp8;
(*soft_config).display_s=temp9;
(*soft_config).display_as=temp10;
(*soft_config).display_a=temp11;
(*soft_config).display_fr=temp12;
return msg;
}

```

```

double get_unit_vect(struct unit_vect *vect, double x, double y, double z, double a, double angle, int
object)
{
double total, temp;

if (object == XY_ARC_ENT) {
temp=x;
x=y;
y=temp;
if (angle<0) {
x=-x;
y=-y;
}
if (angle != 0) {
z=z/fabs(angle);
a=a/fabs(angle);
}
}

if (object == YZ_ARC_ENT) {
temp=y;
y=z;
z=temp;
if (angle<0) {
y=-y;
z=-z;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if (angle != 0) {
        x=x/fabs(angle);
        a=a/fabs(angle);
    }
}

if (object == ZX_ARC_ENT) {
    temp=z;
    z=x;
    x=temp;
    if (angle<0) {
        z=-z;
        x=-x;
    }
    if (angle != 0) {
        y=y/fabs(angle);
        a=a/fabs(angle);
    }
}
total=total_vect(x, y, z, a);
if (total == 0) {
    (*vect).x=(*vect).y=(*vect).z=(*vect).a=0;
    return total;
}
(*vect).x=x/total;
(*vect).y=y/total;
(*vect).z=z/total;
(*vect).a=a/total;
if (object != LINE_ENT)
    total=1;
return total;
}

//find maximum ending speed squared
float max_speed_sq(struct hard_setup *hard_config,
    float x1, float y1, float z1, float a1, float total1,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    struct unit_vect *vect2)
{
    float vx2, vy2, vz2, va2, den;

    float max_speed;

    if (total1 == 0)
        return 0;

    //define maximum speed squared
    max_speed=get_max_speed(hard_config, x1, y1, z1, a1, total1);
    max_speed=max_speed*max_speed;

    //determine maximum speed squared entering new vector
    if ((den=fabs((*vect2).x-x1/total1)*(fabs((*vect2).x)+fabs(x1/total1)))<=.00000001)
        vx2=max_speed;
    else
        vx2=(*hard_config).accel_x*(*hard_config).x_per_step/den;
    if ((den=fabs((*vect2).y-y1/total1)*(fabs((*vect2).y)+fabs(y1/total1)))<=.00000001)
        vy2=max_speed;
    else
        vy2=(*hard_config).accel_y*(*hard_config).y_per_step/den;
    if ((den=fabs((*vect2).z-z1/total1)*(fabs((*vect2).z)+fabs(z1/total1)))<=.00000001)
        vz2=max_speed;
    else
        vz2=(*hard_config).accel_z*(*hard_config).z_per_step/den;
    if ((den=fabs((*vect2).a-a1/total1)*(fabs((*vect2).a)+fabs(a1/total1)))<=.00000001)
        va2=max_speed;
    else
        va2=(*hard_config).accel_a*(*hard_config).a_per_step/den;

    //return lowest speed
    if (vx2<=vy2 && vx2<=vz2 && vx2<=va2 && vx2<max_speed)
        return vx2;
    if (vy2<=vx2 && vy2<=vz2 && vy2<=va2 && vy2<max_speed)
        return vy2;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (vz2<=vx2 && vz2<=vy2 && vz2<va2 && vz2<max_speed)
    return vz2;
if (va2<=vx2 && va2<=vy2 && va2<vz2 && va2<max_speed)
    return va2;
return max_speed;
}

//get maximum ending speed due to length of next line
float max_speed_sq_line(struct hard_setup *hard_config, double x, double y, double z, double a, double
total)
{
    float max_accel, min_speed;

    if (total == 0)
        return 0;
    max_accel=get_max_accel(hard_config, x, y, z, a, total);
    min_speed=get_min_speed(hard_config, x, y, z, a, total);
    return total*2*max_accel+min_speed*min_speed;
}

//check for backlash to come in next move
int backlash_to_come(struct hard_setup *hard_config, struct unit_vect *vect1, struct unit_vect *vect2,
    int fd_fwd_msg, int fd_fwd_msg_a)
{
    static struct unit_vect prev;
    int msg=FALSE;

    if (fd_fwd_msg == START_OVER) {
        prev.x=(*vect1).x;
        prev.y=(*vect1).y;
        prev.z=(*vect1).z;
    }
    if (fd_fwd_msg_a == START_OVER)
        prev.a=(*vect1).a;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//check for backlash to come
if ((prev.x*(*vect2).x<0 && (*hard_config).back_x>.000009) ||
    (prev.y*(*vect2).y<0 && (*hard_config).back_y>.000009) ||
    (prev.z*(*vect2).z<0 && (*hard_config).back_z>.000009) ||
    (prev.a*(*vect2).a<0 && (*hard_config).back_a>.000009))
    msg=TRUE;

if ((*vect2).x != 0)
    prev.x=(*vect2).x;
if ((*vect2).y != 0)
    prev.y=(*vect2).y;
if ((*vect2).z != 0)
    prev.z=(*vect2).z;
if ((*vect2).a != 0)
    prev.a=(*vect2).a;
return msg;
}

int jog(struct hard_setup *hardware, struct soft_setup *software, struct pos *pt, int type, double inc, int
axis, int dir)
{
    float j_speed=0, cur_speed=0;
    double inc_x, inc_y, inc_z, inc_a;
    int x=0, y=0, z=0, a=0, msg=COMPLETE;

    if (axis == X_AXIS)
        x=dir;
    if (axis == Y_AXIS)
        y=dir;
    if (axis == Z_AXIS)
        z=dir;
    if (axis == A_AXIS)
        a=dir;
    if (type == FAST || type == INC) {
        if (axis == X_AXIS) j_speed=(*software).fjog_x;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (axis == Y_AXIS) j_speed=(*software).fjog_y;
if (axis == Z_AXIS) j_speed=(*software).fjog_z;
if (axis == A_AXIS) j_speed=(*software).fjog_a;
}
else {
    if (axis == X_AXIS)
        j_speed=(*software).sjog_x;
    if (axis == Y_AXIS)
        j_speed=(*software).sjog_y;
    if (axis == Z_AXIS)
        j_speed=(*software).sjog_z;
    if (axis == A_AXIS)
        j_speed=(*software).sjog_a;
}
if (type == INC) {
    if (fabs(inc)<.000001)
        inc=0;
    inc_x=inc_y=inc_z=inc_a=inc;
    msg=move(hardware, software, pt, &cur_speed, &j_speed,
        x*inc_x, y*inc_y, z*inc_z, a*inc_a, NULL, LINE_ENT, 0, 0, 0, 0, NULL, LINE_ENT);
}
else {
    (*software).keyboard=JOG;
    jogging=ON;
    msg=move(hardware, software, pt, &cur_speed, &j_speed,
        x*9999, y*9999, z*9999, a*9999,
        NULL, LINE_ENT, 0, 0, 0, 0, NULL, LINE_ENT);
    (*software).keyboard=ON;
    if (type == FAST && msg == COMPLETE)
        msg=move(hardware, software, pt, &cur_speed, &j_speed,
            x*cur_speed*cur_speed/(2*(*hardware).accel_x),
            y*cur_speed*cur_speed/(2*(*hardware).accel_y),
            z*cur_speed*cur_speed/(2*(*hardware).accel_z),
            a*cur_speed*cur_speed/(2*(*hardware).accel_a),
            NULL, LINE_ENT, 0, 0, 0, 0, NULL, LINE_ENT);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

return msg;
}

int homing(struct hard_setup *hardware, struct soft_setup *software, struct pos *pt,
           int id1, int dir1, int id2, int dir2)
{
    int i, msg=COMPLETE, x=0, y=0, z=0, a=0, temp_a_follow;
    float old_x_sp, old_y_sp, old_z_sp, old_a_sp, cur_speed=0, sp_limit=0;

    old_x_sp=(*hardware).vel_x;
    old_y_sp=(*hardware).vel_y;
    old_z_sp=(*hardware).vel_z;
    old_a_sp=(*hardware).vel_a;
    temp_a_follow=(*software).a_follow;

    (*hardware).vel_x=(*hardware).h_spd_x;
    (*hardware).vel_y=(*hardware).h_spd_y;
    (*hardware).vel_z=(*hardware).h_spd_z;
    (*hardware).vel_a=(*hardware).h_spd_a;
    (*software).a_follow=INDEPENDANT;

    if (id1 == X_AXIS) {
        if (dir1 == NULL)
            x=(*software).home_x;
        else
            x=dir1;
    }

    if (id1 == Y_AXIS) {
        if (dir1 == NULL)
            y=(*software).home_y;
        else y=dir1;
    }

    if (id1 == Z_AXIS) {
        if (dir1 == NULL)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        z=(*software).home_z;
    else
        z=dir1;
}
if (id1 == A_AXIS && (*software).a_status == ON) {
    if (dir1 == NULL)
        a=(*software).home_a;
    else
        a=dir1;
}
if (id2 == X_AXIS) {
    if (dir2 == NULL)
        x=(*software).home_x;
    else
        x=dir2;
}
if (id2 == Y_AXIS) {
    if (dir2 == NULL)
        y=(*software).home_y;
    else
        y=dir2;
}
if (id2 == Z_AXIS) {
    if (dir2 == NULL)
        z=(*software).home_z;
    else
        z=dir2;
}
if (id2 == A_AXIS && (*software).a_status == ON) {
    if (dir2 == NULL)
        a=(*software).home_a;
    else
        a=dir2;
}
for(i=0; i<4; i++) {
    msg=move(hardware, software, pt, &cur_speed, &sp_limit,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    x*9999, y*9999, z*9999, a*9999,
    NULL, LINE_G0_ENT, 0, 0, 0, 0, NULL, LINE_ENT);
if (msg == X_PLIMIT || msg == X_NLIMIT)
    x=0;
if (msg == Y_PLIMIT || msg == Y_NLIMIT)
    y=0;
if (msg == Z_PLIMIT || msg == Z_NLIMIT)
    z=0;
if (msg == A_PLIMIT || msg == A_NLIMIT)
    a=0;
if (msg == BREAK || msg == E_STOP || msg == QUIT)
    goto DONE;
}
for(i=0; i<4; i++) {
    cur_speed=sp_limit=x=y=z=a=0;
    if (id1 == X_AXIS && i == 0) {
        if (dir1 == NULL)
            x=(*software).home_x;
        else
            x=dir1;
        sp_limit=(*hardware).min_vel_x;
    }
    if (id2 == X_AXIS && i == 0) {
        if (dir2 == NULL)
            x=(*software).home_x;
        else
            x=dir2;
        sp_limit=(*hardware).min_vel_x;
    }
    if (id1 == Y_AXIS && i == 1) {
        if (dir1 == NULL)
            y=(*software).home_y;
        else
            y=dir1;
        sp_limit=(*hardware).min_vel_y;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
if (id2 == Y_AXIS && i == 1) {
    if (dir2 == NULL)
        y=(*software).home_y;
    else
        y=dir2;
    sp_limit=(*hardware).min_vel_y;
}
if (id1 == Z_AXIS && i == 2) {
    if (dir1 == NULL)
        z=(*software).home_z;
    else
        z=dir1;
    sp_limit=(*hardware).min_vel_z;
}
if (id2 == Z_AXIS && i == 2) {
    if (dir2 == NULL)
        z=(*software).home_z;
    else
        z=dir2;
    sp_limit=(*hardware).min_vel_z;}
if (id1 == A_AXIS && (*software).a_status == ON && i == 3) {
    if (dir1 == NULL)
        a=(*software).home_a;
    else
        a=dir1;
    sp_limit=(*hardware).min_vel_a;
}
if (id2 == A_AXIS && (*software).a_status == ON && i == 3) {
    if (dir2 == NULL)
        a=(*software).home_a;
    else
        a=dir2;
    sp_limit=(*hardware).min_vel_a;}
msg=move(hardware, software, pt, &cur_speed, &sp_limit,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    x*9999, y*9999, z*9999, a*9999,

    NULL, LINE_ENT, 0, 0, 0, 0, NULL, LINE_ENT);

if (msg == BREAK || msg == E_STOP || msg == QUIT)
    goto DONE;
if (x != 0)
    (*pt).x_pos=0;
if (y != 0)
    (*pt).y_pos=0;
if (z != 0)
    (*pt).z_pos=0;
if (a != 0)
    (*pt).a_pos=0;
}
msg=COMPLETE;

DONE:
(*hardware).vel_x=old_x_sp;
(*hardware).vel_y=old_y_sp;
(*hardware).vel_z=old_z_sp;
(*hardware).vel_a=old_a_sp;
(*software).a_follow=temp_a_follow;

return msg;
}

```

เอกสารอ้างอิง

- [1] อภิชาติ ภู่วลัฒ. เริ่มต้นการเขียนโปรแกรมติดต่อ และควบคุมฮาร์ดแวร์ด้วย Visual Basic. กรุงเทพมหานคร:คิธีวี บุค.2546.
- [2] <http://www.editecnc.com/ProgramExamples.html>
- [3][http://nptel.iitm.ac.in/courses/Webcourse-contents/IIT-Delhi/Computer%20Aided%20Design%20&%20ManufacturingII/Module%20F/Module%20F\(5\)/p2.htm](http://nptel.iitm.ac.in/courses/Webcourse-contents/IIT-Delhi/Computer%20Aided%20Design%20&%20ManufacturingII/Module%20F/Module%20F(5)/p2.htm)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้